Link Prediction in the Yelp Social and Review Networks

Lucas Finn
Team 35
BAE Systems
6 New England Executive Park
Burlington, MA 01803
lucas.finn@gmail.com

ABSTRACT

In this project, we apply two predictive network algorithms to the Yelp academic challenge dataset. The first algorithm is adapted from Leskovec, Huttenlocher and Kleinberg (2010) to predict the rating that a user will assign to a business [1]. We adapt the notion of signed edges in the network to account for both user-user edges (friendships) as well as user-business edges (reviews). A logistic regression model is trained from features in the social and review network data, and experiments are carried out using ten-fold cross validation. We find that the ability to predict a user's rating of a business depends highly on the prior ratings given by the user and the prior ratings received by the business. While a user's rating is impacted by the ratings given by his or her friends, this influence appears to be a second-order effect. In addition, the logistic regression classifier achieves a moderate improvement over a baseline classifier.

The second algorithm is adapted from Backstrom and Leskovec (2011) to predict the friends that a user will link to in the future [2]. We extract eleven features from the Yelp dataset and separate friendships into training and testing sets. The training edges remain in the social network; the testing edges are removed. We then optimize a personalized weight vector for each user with gradient descent using a supervised random walk. We find that 56% of the testing edges are ranked in the top 20 recommendations for 300 randomly selected users. Because the candidate set can contain thousands of possible edges, this result is moderately significant.

The applications of these predictive models in the Yelp data are twofold: the first algorithm enables a user to predict their future evaluation of a business; equivalently a business can search for potential new clients. The second algorithm enables a friendship recommendation engine, which improves the user experience of the Yelp service, which benefits both users and businesses.

Keywords

Yelp, link recommendation, link prediction, edge sign prediction, supervised random walk, logistic regression

1. INTRODUCTION

1.1 Predicting Review Ratings

As the amount of information available in network problem domains increases, reducing the amount of information presented to a user becomes increasingly important [3]. Recommendation engines attempt to solve this problem by presenting the user with a subset of all possibilities. An example from the Yelp network is to recommend businesses to a user given all the historical data collected from Yelp [7]. The available data includes information about the user, the business, the social network surrounding the user, etc. In this project, we formulate the problem as the following: if the user reviews a particular business, what is the probability that the user will review the business favorably?

Multiple techniques have been applied in several network contexts to create recommendations, including collaborative filtering [3] and support vector machines (SVMs) [4]. Collaborative filtering is a matrix factorization approach wherein new user preferences are inferred from existing users with similar preferences. The recommendation matrix is factored into a low-dimensional representation with latent features [5]. The process is computationally expensive: traditional collaborative filtering requires $O(MN)$ operations, where M is the number of users and N is the number of produces (e.g. businesses). Optimizations that exploit the sparse nature of the network and subsampling techniques can improve runtime, but the process is still expensive on large networks. However, this technique has been applied successfully to several problems, including NetFlix Prize competition for product recommendations [6].

Support vector machines are an additional technique to predict structures in a network [4]. A set of features is identified from nodes and edges, and data is separated into training and testing sets. A binary classifier can be evaluated with several kernels to identify the highest training and testing accuracy. This technique has the benefit of finding optimal weights for the given set of features; it has also been applied to several problems, including predicting co-authorship network edge creation. Note that both collaborative filtering and SVMs traditionally exploit features extracted from the network and not the actual network structure itself [8]. We elaborate on this distinction in the next section.

1.2 Predicting Future Friendships

The problem of recommending friends to a user in a social network is similar to predicting the favorability of a review as described in the previous section. The desire for this capability is also the same: reducing the amount of information presented to a user to improve their experience. Recommendations engines are responsible for creating a significant fraction of connections in social networks, for example in the Facebook network [2].

There are several approaches to the link prediction problem, which is also referred to as network completion [10]. The first approach is machine learning based, and extracts features to train a model, similar to review prediction. The main drawbacks of these approaches are the lack of exploitation of network structure, and the lack of training data. For example, a typical user links to a tiny fraction of an entire social network. Recently, extensions to these machine learning formulations exploit additional elements in the network, such as geographic location [11] to improve accuracy.

Hybrid approaches which exploit both node and edge features, as well as network structure appear to improve prediction accuracy in social networks [2, 9]. By combining learned features with PageRank-type graph algorithms, personalized models can be trained for each user.

1.3 Project Outline

In this project, we apply two predictive network algorithms to the Yelp academic challenge dataset. Section 2 provides an overview of the dataset, including some interesting trends of the Yelp service over time, as well as user trends. Section 3 introduces an algorithm to predict the favorability of a user review of a business using a logistic regression classifier. Section 4 adapts a hybrid supervised random walk approach to the Yelp dataset to recommend friends for a subset of users. Section 5 discusses general conclusions.

2. DATASET DESCRIPTION

2.1 Yelp Dataset Overview

The Yelp challenge dataset contains a rich set of features across a large number of entities [7]. In this project, we focus on users, businesses and reviews. The dataset contains additional data not considered here, such as check-ins of when a user is visiting a business. The user data also contains friendship information, which creates the user social network. Reviews are treated as directed edges from one user to one business and represent the opinion of the user of that business. Table 1 summarizes the data published by the Yelp service.

Table 1 – Yelp dataset statistics indicate that both the social and business review networks are sparse.

Users	252,898	Businesses	42,153
User-user edges (friendships)	955,999	User-business edges (reviews)	1,125,458
User-user-user triads	8,977,967	User-user-business triads	666,238
90% effective social diameter	5.0	Largest social CC size	119,839

There are 253K users, but only 956K friendships, indicating that the social network is sparse: 0.001% of entries in the connectivity matrix exist. Additionally, there are 1.1M reviews: 0.01% of entries in the user-business matrix, which is slightly denser than the social network. Note that this data represents a subset of the entire Yelp database, e.g. not every friendship created or review written is included. Additionally, a user can review the same business multiple times; roughly 3.1% of reviews are duplicates. However, these reviews are not redundant; they may occur several years apart, or indicate a change or reinforcement of an opinion.

The structure of the social network in the provided dataset strongly implies that roughly half of users joined the Yelp service to review businesses. Over 51% of users (131K) do not take part in the social network at all; these users only review businesses. However, the remaining 49% of users (123K) form nearly a single connected component in the social network (119K). The remaining users (4K) all belong to connected components with nine or fewer users, most subgraphs contain only 2-3 users.

The social network itself is nearly undirected: if user *u* links to user *v* as a friend, then *v* also links to *u* as a friend. However, there are 42 instances (0.004% of occurrences) where friendship is not reciprocated in the dataset; this is likely an artifact caused by publishing a subset of the available data. As a result, we will treat the social network as undirected in this project.

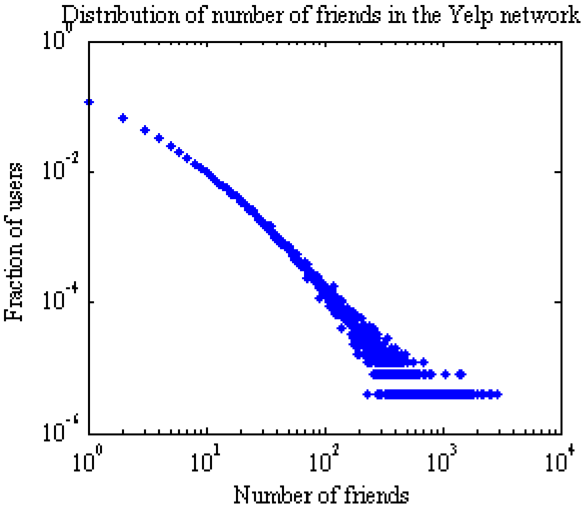

Figure 1 – The distribution of number of friends in the Yelp network follows a power law with a slight exponential cutoff.

Interestingly, while only about half of the users participate in the social network, the 90% effective diameter of the largest connected component (119K users) is only 5.0. The approximate full diameter is 11.0. There is insufficient evidence to conclude if this is a social phenomenon, or caused by the specific choice of users to include in the Yelp dataset. For users which connect to at least one friend, the frequency of friendships follows a common power law as shown in Figure 1.

In addition to the network structure, all entities contain an extensive list of attributes. The user data is partially anonymized, but the review and business data are not. For example, the user data does not include age, gender, or residential location, but does include the date the user joined Yelp, the number of reviews written, average rating, and the first name of the user. The business data includes fields such as the full name, address, type and hours. Users can annotate the reviews written by other users with attributes such as *useful* or *funny*.

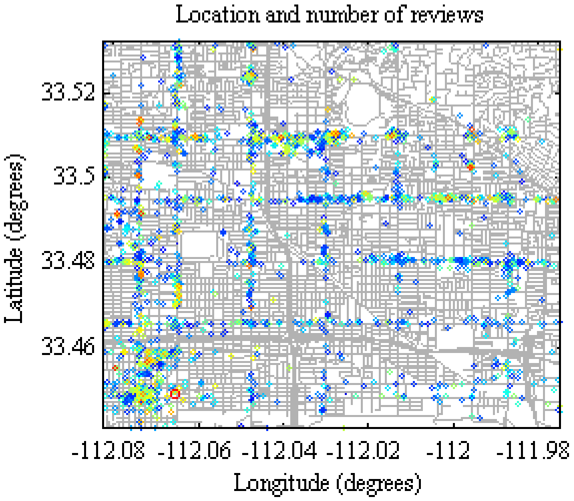

Figure 2 – Locations of businesses colored by number of reviews. The surrounding road network is shown in gray for reference.

The dataset includes businesses from several cities in the United States, including Arizona, Wisconsin, Massachusetts and Nevada. Figure 2 shows the location of some reviewed businesses in Phoenix, AZ. The city road network is included for scale; the rectangular spatial layout of businesses in the city is clearly visible. Each business is represented by a circle whose color indicates the number of reviews associated with that business. Blue indicates few reviews; red indicates many reviews.

2.2 User Trends

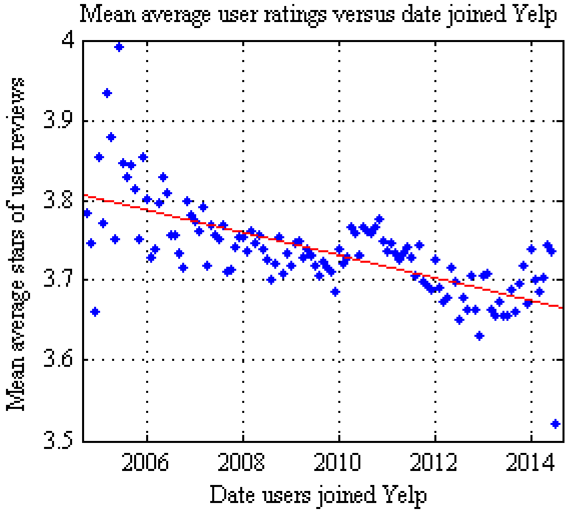

Figure 3 – Users who joined the Yelp service early tend to provide more favorable reviews of businesses.

An analysis of the raw user data indicates two interesting trends. The first trend shows that the earlier that a user joined the Yelp service, the higher the user tends to rate businesses. Since the dataset spans the ten years from 2004 to 2014, we see a decrease in the mean average rating of businesses from approximately 3.8 to 3.7 in Figure 3. The outlier in 2014 is likely caused by the dataset cutoff in July. Additionally, the variance in average user reviews increased significantly over the same time period, as shown in Figure 4.

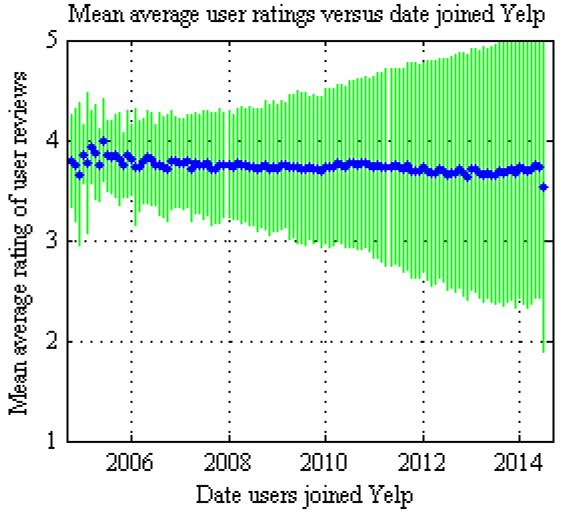

Figure 4 – One-sigma standard deviation in the average rating of user reviews increases significantly from 2004 to 2014.

One possible explanation is that early adopters of a service tend to be more optimistic about that service and hence review more favorably. Late adopters may join specifically to provide targeted feedback with a particularly positive or negative experience. This conjecture is partially supported by the large number of users who do not participate in the social network, and instead only review a small number of businesses.

The second trend is shown by analyzing the average rating versus the number of reviews written by a user (Figure 5). The ratings given by users that write few reviews have a large variance, with bimodal peaks at one and five stars. As a user writes more reviews, the variance in average ratings decreases. It is interesting to note that as a population, the average user rating is around 3.75 regardless of the number of reviews written. This weakly implies that there exists a common notion of a favorable business, independent of the number of reviews that a user has written.

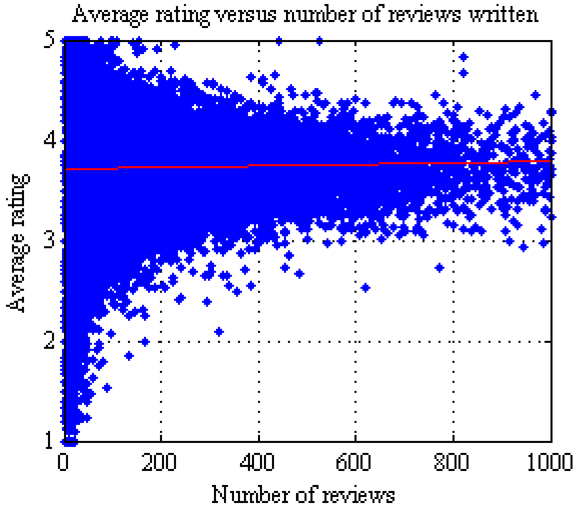

Figure 5 – The variance of the average review rating decreases with number of reviews written, but the mean remains near 3.75.

3. REVIEW PREDICTION

In this section, we adapt the algorithm described in [2] to predict the favorability of a user review. Specifically, given a user *u*, business *b*, and review *r*=(*u*,*b*), we seek to predict if the review is favorable or not. Recall that the mean user rating is 3.75; the median rating is 4 stars. We therefore define a favorable rating as 4 or 5 stars; an unfavorable rating is 1, 2 or 3 stars. With this definition, we find that roughly 34% of reviews in the dataset are negative and 66% are positive.

We apply logistic regression using two classes of features: node features and social network features. The methodology requires adaptation from [2] for several reasons. First, the underlying social network is undirected and unsigned, i.e. friendships are bidirectional and positive. Second, the Yelp network contains two entity types: users and businesses. The user-business edges are directed, and businesses only have incoming edges.

As a result, the Yelp network supports only two types of triads: user-user-user and user-user-business. Of the two, only the user-user-business triads can contain a negative sign, when the user unfavorably reviews a business. As shown in Table 1, there are nearly 9M user triads, but only 666K triads involving reviews. This implies that while many users are socially connected, they do not often review the same business. This result mildly surprising, since friends may frequent businesses together (e.g. dine together at a restaurant). This discrepancy (if any) may be caused by the choice of published Yelp data or the low probability of two friends reviewing the same business after a joint visit.

Table 2 – Fourteen features extracted from the Yelp user, business and review data.

No.	Feature	No.	Feature
1	| *u*.friends |	8	mean(*b*.reviews)
2	| *u*.reviews = + |	9	| *b*.reviews |
3	| *u*.reviews = - |	10	| *u*.reviews |
4	| *b*.reviews = + |	11	mean(*u*.ratings)
5	| *b*.reviews = - |	12	*u*.elite
6	| $u \to +(w,b)$ |	13	| *u*.fans |
7	| $u \to -(w,b)$ |	14	*b*.stars

Table 2 shows the fourteen features extracted to train the logistic regression classifier. In words, these features are (1) the number of friends the user has, (2-3) the number of [un]-favorable reviews made by the user, (4-5) the number of [un]-favorable reviews of the business, (6-7) the number of [un]-favorable triads, (8) the average rating of the business, (9) the number of times the business was reviewed, (10) the number of reviews made by the user, (11) the total average user rating, (12) if the user was elite, (13) the number of fans, (14) the total average rating of the business.

To clarify the user-user-business triads, suppose we are given a user *u* and a business *b*. We find all users *w* such that *u* and *w* are friends and the (*w*,*b*) review exists. If *w* favorably reviewed *b*, we say that the triad is positive (feature 6). Otherwise, we say that the triad is negative (feature 7). These correspond to the balance theory "the friend of my friend is my friend" and "the enemy of my friend is my enemy" [12].

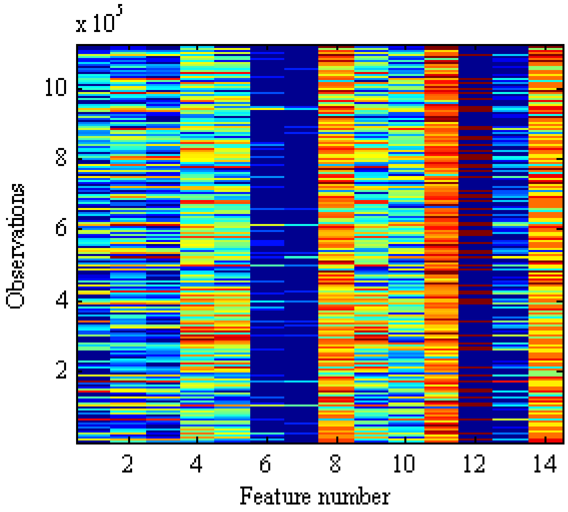

Figure 6 – Color map of fourteen features for each user-business review illustrates the distribution of each feature value.

Features 9, 10, 11 and 14 come directly from the Yelp dataset. For example, feature 9 is the total number of reviews that *b* has received, while features 4 and 5 represent the total number of reviews included in the provided subset of Yelp data. Therefore, features 4, 5 and 9 are linearly independent quantities. A similar argument holds for features 10, 11 and 14.

As a side note, the Yelp data also admits a connection to the theory of status [1]. Users can nominate themselves to obtain *elite* status for one calendar year. The Yelp service grants users this attribute; their reviews are featured more prominently. The evaluation of status theory is beyond the scope of this project; we leave this evaluation for future work.

While the individual features contain significant variations, each feature is rescaled to contain values between 0 and 1. For each review, Figure 6 shows the feature values: blue represents a value near 0; red represents values near 1. For example, column 1 visually illustrates the power law scaling of the number of friends in the social network. Note that columns 6 and 7 represent the user-user-business triad ratings, and contain significantly less variation than the other features. This shows the lack of available data to create these triads in the Yelp review network. Feature 12 is a binary value indicating elite status, and so only appears as blue (not elite) or red (elite).

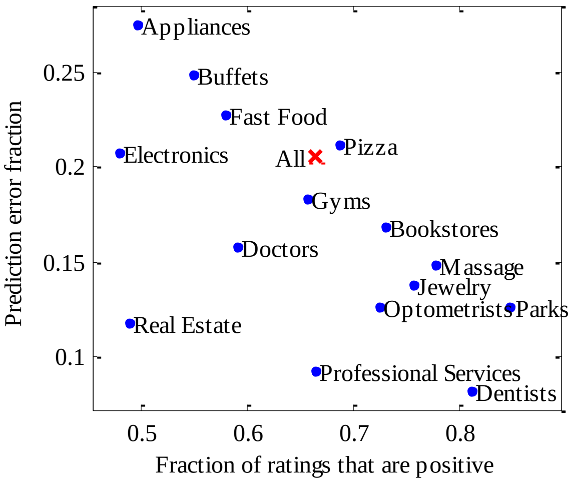

Figure 7 – Error rate versus fraction of ratings that are positive per business type. The All rating (red X) includes all businesses.

We train a logistic regression classifier to evaluate the probability P(+|x) of a favorable rating, given the user and business features:

$$P(+|x)=\left(1+e^{b_0+x\cdot b}\right)^{-1},$$

where *x* is the feature vector, + indicates a favorable rating, and *b* is the trained model coefficients. We use 10-fold cross-validation for training and testing with logistic regression. We train a classifier on all business categories jointly, as well as classifiers for specific business categories, e.g. fast food, appliances and dentists. Figure 7 shows the error rate in classifier prediction versus the fraction of ratings for businesses that are positive. For example, of all businesses listed in the "Doctor" category, 59.0% are reviewed favorably. Hence a classifier that always chooses favorable would have a 31.0% error rate; the logistic regression classifier achieves 15.8% error rate.

The highest improvement over a constant classifier for the various business types tested was Real Estate, where 48.8% of reviews were favorable and the classifier achieved an error rate of 11.8%. Training a single classifier on all businesses resulted in an error rate of 20.5%, where a constant favorable classifier would achieve

33.4% error rate. These results demonstrate less success than the applications to Facebook and other social networks as described in [1], although the reasons are not immediately clear.

To investigate the cause of the high error prediction rate, Table 3 shows the logistic regression coefficients for each of the fourteen features. The constant term b_0 (not shown) is -2.3. While the exact magnitude of these coefficients changes when training different business types, the relative ranking is nearly constant. Qualitatively, the most important features are consistently 2, 3, 4, 5 and 11. This seems to imply that the strongest predictor of the favorability of a review is the historical precedent of the user to give [un-]favorable reviews, and the business to receive [un]-favorable reviews. Note that the triad features 6-7 are next in terms of magnitude, but still far less indicative of a rating than features 2-5. This is possibly due to the lack of training triad data. In this case, a friend's rating appears to be a less important factor to a user's review than would be expected from [1].

Table 3 – Logistic regression coefficients trained from reviews indicate that personal user and business attributes are significant.

#	Feature	Cf.	#	Feature	Cf.
1	| *u*.friends |	-0.5	8	mean(*b*.reviews)	1.0
2	| *u*.reviews = + |	9.6	9	| *b*.reviews |	-0.8
3	| *u*.reviews = - |	-10	10	| *u*.reviews |	0.6
4	| *b*.reviews = + |	7.7	11	mean(*u*.ratings)	2.0
5	| *b*.reviews = - |	-6.7	12	*u*.elite	*-0.1*
6	| *u* → +(*w*,*b*) |	1.5	13	| *u*.fans |	0.5
7	| *u* → −(*w*,*b*) |	-1.3	14	*b*.stars	-0.1

There are three additional results of note. First, the signs of features 2-7 are as expected: e.g. a friend's favorable review correlates with a user's favorable review; similarly for unfavorable reviews. Second, the business category strongly influences the average rating of a given business. For example, electronics, real estate and appliances have some of the lowest average favorable ratings (near 50%). On the other hand, jewelers, dentists and optometrists have some of the highest (70-80%). Thus a constant classifier will perform differently on each type of business. The third interesting point is that the more reviews that a user writes, the lower the error rate becomes in predicting reviews for that user. For example, the error rate drops from 20.5% to 15.0% if we evaluate users who have written at least 25 reviews, with 67.4% of reviews being positive.

4. LINK PREDICTION

4.1 Supervised Random Walk Description

In this section, we execute a supervised random walk on the Yelp social graph to predict the future friends *v* for a given user *u*. We extract eleven features drawn from both the social and business review graphs. Table 4 summarizes these features, which are meant to model the personality of a user, i.e. their status, their disposition, and their social proclivities.

In words, the features are: the number of years that *u* and *v* have obtained elite status on Yelp (1-2), the average rating given for a business (3-4), the total number of reviews given (5-6), the total number of friends (7-8), the number of one-hop friends that *u* and *v* have in common (9), the number of reviewed businesses in common (10), and the number of business categories reviewed in common (11). All features are normalized to have zero mean and unit variance.

Table 4 – Features for the supervised random walk combine information from the Yelp social and business review graph

No.	Feature
1-2	*u* and *v* years of elite status
3-4	*u* and *v* mean business rating
5-6	*u* and *v* review counts
7-8	*u* and *v* friend counts
9	*u* and *v* number of friends in common
10	number of common businesses reviewed
11	number of common business categories reviewed

We form an optimization problem as described in [2] to find the set of feature weights *w* that result in the smallest loss function,

$$\min_{w} F(w) = \|w\|^2 + \lambda \sum_{d\in D, l\in L} h(p_l - p_d),$$

where *F* is the objective function, *h* is the loss function, p_i is the ranking of node *i*, *D* is the set of nodes that will be connected in the future, and *L* is the set of nodes that will not be connected to in the future. The equations used are explained next, however the intuition is straightforward: we optimize *F* by finding the weight vector that ranks nodes in *D* higher than nodes in *L*. Note that because the objective function is not convex, we are not guaranteed to find a global optimum.

One additional interesting note is that if all the rankings are satisfied, i.e. $p_d > p_l \; \forall d \in D, l \in L$, then the loss term h is identically zero. In this case, the optimization simply minimizes the magnitude of the weight vector *w* such that the relative rankings remain fixed. At the optimum, we should observe that $p_d \sim p_l$ for some $l \in L, d \in D$. This makes for a simple check on a subset of the algorithm implementation (i.e. it does not test all of the loss function calculation).

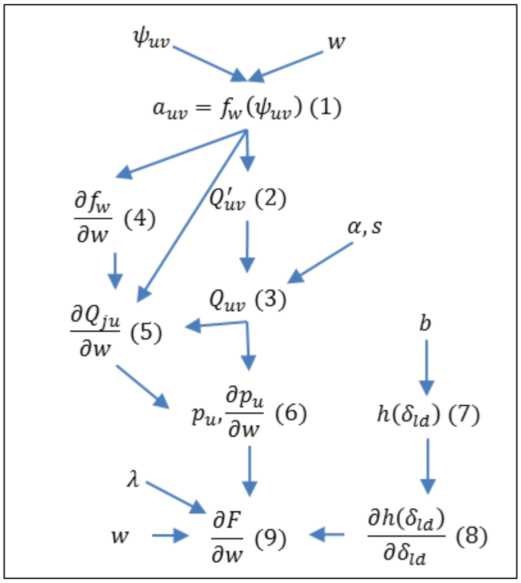
ψ_{uv}
w
$a_{uv} = f_w(\psi_{uv})$ (1)
$\frac{\partial f_w}{\partial w}$ (4)
Q'_{uv} (2)
α, s
$\frac{\partial Q_{ju}}{\partial w}$ (5)
Q_{uv} (3)
b
$p_u, \frac{\partial p_u}{\partial w}$ (6)
$h(\delta_{ld})$ (7)
λ
w
$\frac{\partial F}{\partial w}$ (9)
$\frac{\partial h(\delta_{ld})}{\partial \delta_{ld}}$ (8)

Figure 8 – Symbol dependency diagram to implement gradient descent of *F* as a function of *w*.

We implement gradient descent to iteratively compute *w*. Figure 8 shows the symbol dependencies needed to compute $\partial F / \partial w$, which is used to update w. The number next to each symbol in the figure corresponds to an equation below.

The feature vector for edge (u,v) is denoted ψ_{uv}, and is extracted from both node and edge information in the graph (see Table 4).

$$a_{uv} = f_w(\psi_{uv}) = (1 + \exp(-\psi_{uv} \cdot w))^{-1} \qquad (1)$$

Equation 1 shows how to compute the strength of the edge for the supervised random walk. As the feature vector aligns more with the weight vector, the edge strength increases.

$$Q'_{uv} = \begin{cases} \dfrac{a_{uv}}{\Sigma_w a_{uw}} & \text{if } (u,v) \in E \\ 0 & o.w. \end{cases} \qquad (2)$$

$$Q_{uv} = (1-\alpha)Q'_{uv} + \alpha \mathbf{1}(v = s) \qquad (3)$$

Equation 2 normalizes edge strengths to compute the transition probabilities in the random walk. Equation 3 includes the teleport probability α. As the authors note in [2], the choice of α can vary depending on the application. In this project, we examine values of $\alpha = 0.0, 0.1, 0.2$. Note that $\alpha = 0$ corresponds to a PageRank proportional to the node degree.

$$\frac{\partial f_w(\psi_{uv})}{\partial w} = \psi_{uv}(1 - a_{uv})a_{uv} \qquad (4)$$

Equation 4 is a straightforward partial derivative calculation, and interestingly can be rewritten in terms of the pre-calculated edge strengths a_{uv}. This slight optimization improves overall runtime. Equation 5 is also a straightforward partial derivative calculation, but the equation is not included here.

Equation 6 calculates the PageRank probabilities p and the partial derivatives $\partial p / \partial w$ using a power-iteration algorithm described in [2] and [9]. Note that the transition matrix Q (as opposed to Q') is never actually instantiated, which is necessary since it is a large and dense matrix. First, the probabilities p are calculated as the stationary distribution of Q, then the partial derivatives are calculated iteratively until convergence.

$$p_u = \sum_j p_j^{t-1} Q_{ju}$$
$$\frac{\partial p_u^t}{\partial w} = \sum_j Q_{ju} \frac{\partial p_j}{\partial w} - p_j^{t-1} \frac{\partial Q_{ju}}{\partial w} \qquad (6)$$

Equation 7 calculates the Wilcoxon-Mann-Whitney loss function h with width b. If $x \leq 0$, then $h(x) = 0$. Following the algorithm description in [1], we use $b = 0.1$. The calculation of the partial derivative in Eq. 8 is straightforward.

$$\delta_{ld} = \frac{p(l) - p(d)}{\sum_{v \in D \cup L} p(v)}$$
$$h(x) = (1 + \exp(-x/b))^{-1} \qquad (7)$$

Finally, we combine the partial derivative $\partial F / \partial w$ with the above quantities to update the weight vector w,

$$\frac{\partial F}{\partial w} = 2w + \lambda \sum_{l \in L, d \in D} \frac{\partial h(\delta_{ld})}{\partial \delta_{ld}} \frac{\partial \delta_{ld}}{\partial w}$$
$$w \leftarrow w - \gamma \frac{\partial F}{\partial w} \qquad (9)$$

The additional parameters are the Lagrange multiplier λ and the gradient descent step size γ. From numerical stability tests, we set $\lambda = 1$ and $\gamma = 1e - 3$.

4.2 Supervised Random Walk Implementation and Results

The implementation of the above algorithm was validated in four ways: (1) as mentioned previously, if the rankings are satisfied then the algorithm simply minimizes the weight vector magnitude, (2) the objective function F must strictly decrease due to gradient descent, (3) small graph testing (e.g. line graphs, 12-node random graphs given in lecture) for intuitive results, and (4) the algorithm must improve the global rankings of nodes in D, which must eventually come at the cost of rankings of nodes in L.

As noted in [2], the vast majority of new edges created in social networks are friend-of-friends. Therefore, we evaluate the algorithm against the Yelp social network using two-hop subgraphs centered at randomly chosen users. This approach is similar to [2], and is also for computational efficiency purposes; the entire graph contains nearly two million edges. Figure 9 shows the cardinality distribution of two-hop subgraph sizes on a logarithmic scale. The largest subgraph contains 30,000 nodes, but many subgraphs are 1000 nodes or fewer (and many are singletons).

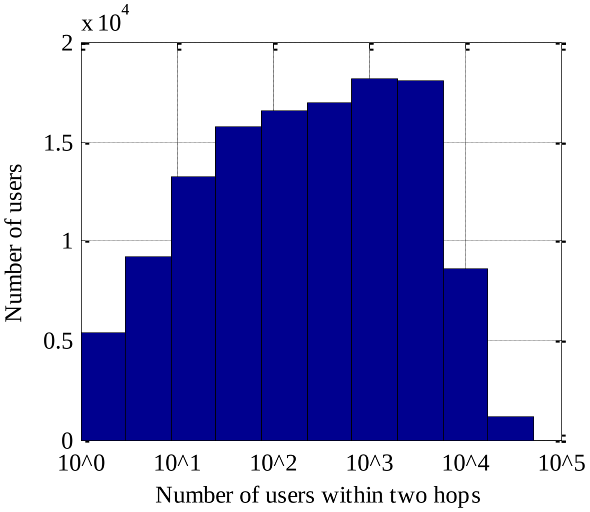

Figure 9 – The cardinality of two-hop subgraph induced by a user contains between 1 and 20,000 nodes.

For a given user s, nodes in the two-hop subgraph of s can be classified into three categories: (1) two-hop nodes that also connect to s with a direct edge, (2) two-hop nodes that do not connect to s with a direct edge, and (3) one-hop nodes (the remaining set of nodes in the two-hop subgraph). To validate the algorithm with Yelp data, we create an experiment where D is defined as the nodes of category type (1), and L is defined as the nodes of category type (2). Note that in this experiment there is no testing data yet, we seek only to validate the evolution of rankings nodes in the sets D and L.

As required by gradient descent, we observe that F is a strictly decreasing function of the algorithm iteration. The algorithm stops when the change in F is below a predefined threshold, which for this dataset was set to $\tau = 1e - 4$. As F decreases, we also see an improvement in the rankings for nodes in D, as desired (lower rank is better). For this example user, the PageRank values for nodes in D are shown in Figure 10. It is interesting to note that some rankings can worsen in the short term; tuning the weight vector unfavorably for one target user improves the overall ranking for nodes in *D* and leads to a global improvement in *F*. Overall the general trend for each node improves with iteration, especially at higher iterations.

Table 5 – Initial and final rankings in an example subgraph of the supervised random walk gradient descent algorithm.

Initial ranking	2	9	16	39	74	91	104	138	490
Final ranking	1	2	3	4	14	23	35	41	486

Table 5 shows the initial and final rankings for the nine nodes in *D* in the above experiment. Note that all improved, several improved significantly. In this subgraph, there were 544 nodes total. It is interesting to note that the outlier with initial rank 490 did not improve much; this is likely due to a combination of (1) the feature vector conflicts with other data in the subgraph, (2) the local graph structure near this node, and (3) the non-convex optimization identifying a local maximum.

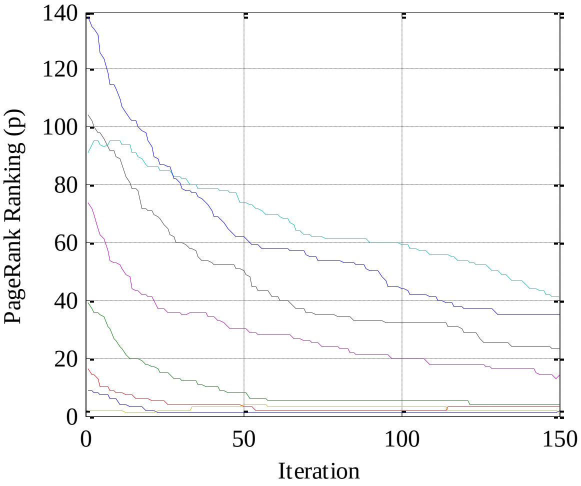

Figure 100 – The ranking of users to which edges will be created in the future improves with the number of SRW iterations.

We next carry out a similar experiment with 300 seed nodes, but with a slight modification. This time, when we choose a seed node *s*, we randomly choose a two-hop node *u* that also connects to *s* with a direct edge (a category 1 type node). We remove the (*s*, *u*) edge from the subgraph. In this way, we can evaluate the ability of the personalized weight vector to predict a future connection for *s*. Because the Yelp data does not timestamp when friend edges are created, we must choose the target user at random. This experiment mimics a personalized recommendation engine, to predict if we can recover the social edge linking the two users.

After carrying out this experiment with 300 randomly chosen users, Figure 11 shows the mean value of weight vector features over the 300 nodes. Features 8 and 9 appear to have the most weight, which correspond to the number of friends that *u* has, as well as the number of friends that *s* and *u* have in common. Slightly weaker is feature 6, which is the number of reviews that *u* has written, which is unsurprising since we have seen that the friend count and review count are correlated. Interestingly, elite status and average business rating have less impact on predicting future changes to the social network.

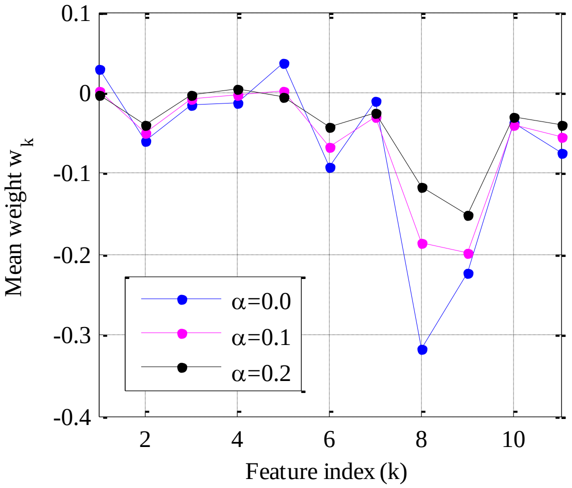

Figure 11 – The mean training weight of each feature in the SRW algorithm, showing a strong influence of features 8 and 9.

To put these results into context for friendship recommendation, Figure 12 shows the personalized PageRank threshold needed to recover the destination users *u*. For example in the 300 trials, the destination user is ranked in the top 20 approximately 56% of the time. The curve grows logarithmically and does not reach 100% until we grow the threshold to several thousand users. This indicates that with extracted features and optimized weight vector, the social network centered at certain users does not match this model well. However, many users do match the model, and this result is somewhat surprising.

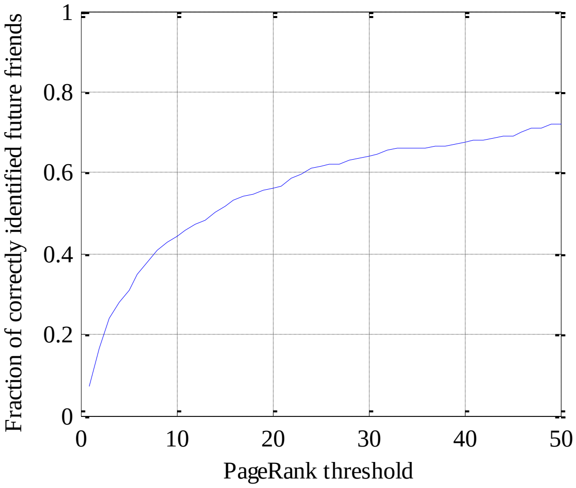

Figure 12 – Testing data indicates that a future friend is correctly ranked within the top 20 recommendations 56% of the time

Because the PageRank threshold metric does not normalize by the total number of candidates, this result may be potentially overstated in terms of significance. To investigate this, we plot the PageRank ranking of the 300 destination users *u* (marked *correct* in the figure) as well as the ranking of the other candidate users in Figure 13 (marked as *incorrect*). We note that the correct destination users consistently have a high PageRank, while the incorrect destination users do not. In other words, not only do the correct destination users frequently appear with high ranking, there are thousands of incorrect candidates that are correctly assigned a lower ranking. The figure is a zoomed-in plot and does not show the thousands of incorrect future connections.

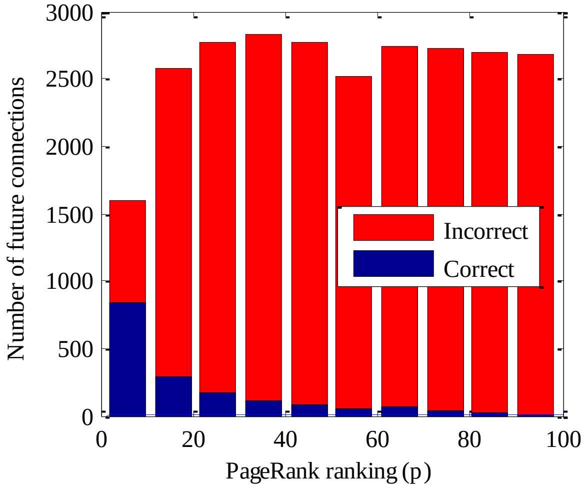

Figure 12 – Rankings for training data show that many testing future connections are identified within the top 20 rankings.

5. CONCLUSIONS

In this project, we investigated the Yelp academic challenge dataset and the application of two predictive algorithms. The Yelp dataset contains a rich set of features for tens of thousands of users and businesses, and more than one million reviews written by users. The friendship network indicates that half of Yelp users only review businesses, while the other half participate in the social network as well as review businesses. Several interesting trends are visible in the data, for example, the date that a user joins the Yelp service reduces the variance in the user's average rating. Also, users who write a single review are more likely to be polarized (i.e. give one or five stars to a business) than a user who writes dozens of reviews. Moreover, the average rating of a user review of a business is negatively correlated with when the user joined the Yelp service. These results appear to indicate that long-time users provide more balanced reviews than new users, or users who write few reviews.

To improve the user experience and enable targeted advertising for businesses, we applied a logistic regression classifier to predict the favorability of a review. We extracted features from the user, the business and the social network to learn the coefficients for the classifier. By looking at user-user-business triads, we can employ the theory of balance, e.g. the enemy of my friend is my enemy. We found that the favorability of a user's review is strongly impacted on historical reviews made by that user, as well as the historical reviews received by the business. Whether or not the user's friends reviewed the business favorably did not seem to strongly impact the user's review. The results may be affected by the lack of triads in the Yelp data, which may be the result of the Yelp dataset containing only a subset of the social network. In addition, the logistic classifier achieved only a small improvement over a baseline classifier. From the experiment in its entirety, one possible conclusion is that a user's experience depends on his or her tastes, and less on the influence of a friend in the Yelp service.

We also investigated a friendship recommendation engine for the Yelp social network. As with the logistic regression classifier, we extract features from network, but in this case we focus on users and friendships. We employed a hybrid approach that combines these features with the graph structure using a supervised random walk to optimize the feature weight vector. Thus, we exploit the actual graph structure, rather than simply extracting features. We tested the ability to predict future friends for 300 randomly chosen users in the Yelp network using personalized weight vectors for each user. We found that 56% of the time, the correct friend was identified in the top 20 recommendations. Because the network examined contains thousands of possible recommendations, these results are of moderate significance.

As networks continue to grow, prioritizing the information presented to a user becomes increasingly important. The ability to predict how a user will review a business and the ability to recommend friends in a social network both improve the user experience. These efforts benefit the users, the businesses, and the Yelp service itself.

6. REFERENCES

[1] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting Positive and Negative Links in Online Social Networks. In Proc WWW. Raleigh, USA. 2010.

[2] L. Backstrom and J. Leskovec. Supervised Random Walks: Predicting and Recommending Links in Social Networks. WSDM Hong Kong, China, 2011, ACM Press.

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. Proc. of International Conference on WWW, pages 285-295, 2001.

[4] M.A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In Proc. of SDM 06 workshop on Link Analysis, 2006.

[5] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1):76-80, January 2003.

[6] M. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. Computer, 42(8):30-37, August 2009.

[7] (2014) *Yelp Dataset Challenge*. Retrieved from http://www.yelp.com/dataset_challenge

[8] Y. Lu, P. Tsaparas, A. Ntoulas, and L. Polanyi. Exploiting social context for review quality prediction. In Proc. of the World Wide Web, WWW, NY, USA, 2010. ACM.

[9] A. Agarwal, S. Chakrabarti, and S. Aggarwal. Learning to rank networked entities. In Proc. ACM SIGKDD, 2006.

[10] M. Kim and J. Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In Proceedings of the 2011 SIAM ICDM, pages 47-58, 2011.

[11] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In Proc. of the ACM SIGKDD, NY, USA, 2011. ACM.

[12] S. H. Yang, A. Smola, B. Long, H. Zha, and Y. Chang. Friend or frenemy: predicting signed ties in social networks. In proc. of the ACM SIGIR, NY, USA, 2012. ACM.