Improving Business Rating Predictions Using Graph Based Features

Amit Tiroshi*†, Shlomo Berkovsky*‡, Mohamed Ali Kaafar*, David Vallet*, Terence Chen*, Tsvi Kuflik†
*NICTA, Australia †University of Haifa, Israel ‡CSIRO, Australia
*firstname.lastname@nicta.com.au †atiroshi,tsvikak@is.haifa.ac.il

ABSTRACT
Many types of recommender systems rely on a rich ensemble of user, item, and context features when generating recommendations for users. The features can be either manually engineered or automatically extracted from the available data, such that feature engineering becomes an important part of the recommendation process. In this work, we propose to leverage graph based representation of the data in order to generate and automatically populate features. We represent the standard user-item rating matrix and some domain metadata, as graph vertices and edges. Then, we apply a suite of graph theory and network analysis metrics to the graph based data representation, in order to populate features that augment the original user-item ratings data. The augmented data is fed into a classifier that predicts unknown user ratings, which are used for the generation of recommendations. We evaluate the proposed methodology using the recently released Yelp business ratings dataset. Our results indicate that the automatically populated graph features facilitate more accurate and robust predictions, with respect to both the variability and sparsity of ratings.

ACM Classification Keywords
H.3.3 Information Storage and Retrieval: Information Filtering

Author Keywords
Recommender Systems, Graph-Based Recommendations, Feature Extraction.

INTRODUCTION
Many widely-used recommendation approaches, e.g., collaborative filtering and matrix factorization, rely – in their base form – on statistical correlations in the available user ratings for items. However, prior research has shown that the accuracy of recommendations can be improved through augmenting the ratings with a variety of user and item features [5, 2]. Examples of systems that exploit data features in the recommendation process include content-based [22], knowledge-based [26], conversational [9], and context-aware recommenders [1], to name a few. Augmenting the data and incorporating additional features allow the recommender to address a range of issues, such as contextual dependencies, explanations and persuasion, bootstrapping and cold-start, diversity, and others.

Generating features (often referred to as *feature engineering*), populating their values, and incorporating them in the recommendation process is, however, not a straightforward process. Firstly, features that shed a new light on the data and encompass a new knowledge, should to be conceived. It is not clear a priori what features are more promising than others, and have the potential to lead to the new knowledge. Secondly, the new features need to be populated for as large as possible portions of the data. This may be a tedious task that is either done by human experts, e.g., through crowdsourcing or focus groups, or requires a substantial domain knowledge, e.g., ontologies or domain-specific databases like IMDB. Thirdly, the contribution of the new features to the recommender should be evaluated, in order to assess to what extent each of the features improves the system's performance and in which conditions.

Previous research into automatic feature generation focused primarily on combining multiple sets of features together [18, 13]. A more recent work proposed to extract new features from the available Social Network user profiles, and leverage these features in the recommendation process [25]. In here, we extend and thoroughly evaluate the ideas presented in [25], and consider a scenario, where new features are extracted and populated through looking at the data from a cardinally different perspective. Specifically, we represent a fairly standard collaborative filtering dataset of user ratings for items (containing also limited metadata: item location and category) using a graph-based structure. The users, items, and metadata entities are considered as the graph vertices, whereas the available user-item ratings and item categorization are the graph edges. Then, we apply a suite of widely-used graph theory and network analysis metrics

IUI'14, February 24–27, 2014, Haifa, Israel.

http://dx.doi.org/10.1145/2557500.2557526

[10], to automatically generate and populate additional features for the users and items. Finally, we feed both the original rating data and the newly populated graph features into a Random Forest regression model [7], in order to predict unknown user ratings and inform the recommendations.

We evaluate our approach using a publicly available dataset of user reviews for businesses, recently released by Yelp for the ACM Recommender Systems Conference 2013 challenge.[1] The dataset contains thousands of user reviews for businesses, which are accompanied by numeric ratings. We focus primarily on the ratings and model the dataset using two representations: as a bipartite (user and business vertices) and a tripartite[2] (user, business, and metadata vertices) non-directional graph. We extract and populate a set of user and business related features and use these features to predict unknown user ratings for businesses. Our results show that augmenting the rating data with the graph features improves the accuracy of the generated recommendations. We also investigate which features, combinations of features, and graph representation contribute most to the accuracy of the recommender, and how this contribution is affected by various parameters of the input data, such as the variability and the sparsity of ratings.

In summary, the main contributions of our work are three-fold. Firstly, we present and demonstrate an approach for augmenting the collaborative user-item rating data with automatically populated graph-based features. Secondly, we provide a strong empirical evidence in favor of incorporating these features into the prediction and recommendation process. Thirdly, we evaluate the accuracy of the rating predictions for various degrees of variability and sparsity in the data.

METHODOLOGY

In this section we present our method for predicting business ratings. This consists of two components. Firstly, we represent the data using a graph model and use this model to augment and populate features used by the prediction mechanism. Secondly, we apply the Random Forest regression method to predict user ratings for businesses. In the following sub-sections we present these components.

Graph Model and Feature Extraction

The original Yelp dataset (will be elaborately presented and characterized in the next section) inherently contains a limited number of features, e.g., average user/business rating and business category, which can be leveraged to predict unknown ratings. In order to enrich the set of features used by the predictor, we represent the data using a graph model. Two models were implemented and evaluated: a *bipartite* model with vertices U and B representing users and businesses, and a *tripartite* model with vertices U, B, and M representing users, businesses, and metadata items, respectively. The models are illustrated in Figure 1.

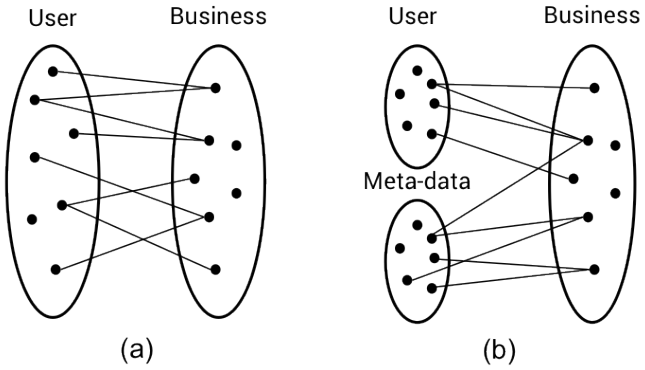


Figure 1. Two graph models: (a) bipartite graph (b) tripartite graph

More formally, the bipartite graph is defined as G = {U, B, E}, where U = $\{u_i \mid i$ is a user$\}$ and B = $\{b_j \mid j$ is a business$\}$ are the vertex sets in the two partitions of G. Vertices u_i and b_j are connected by edge e_{ij} if j was reviewed by i, i.e., E = $\{e_{ij} \mid i$ reviewed $j\}$. Similarly, the tripartite graph is defined by G = {U, B, M, E}. In this case, the vertex sets represent users U = $\{u_i \mid i$ is a user$\}$, businesses B = $\{b_j \mid j$ is a business$\}$, and metadata items M = $\{M_c \cup M_l\}$, where $M_c = \{c_m \mid m$ is the category(ies) of $j\}$, e.g., shopping, food, automotive, and $M_l = \{l_n \mid n$ is the location of $j\}$. As for the edges E of the graph, e_{ij} edges represent, in similar to the bipartite graph, user reviews for businesses, but e_{jc} represent business categories and e_{jn} represent business locations. All the edges in both types of graph are not labeled.

Every review in the dataset provided by user i for business j contributes to multiple features. We aggregate the features into three groups (see Figure 2).

- *Basic* features include only the unique identifiers of i and j.
- *Extended* features include: number of reviews by i, average rating of i, number of reviews for j, number of categories $|\{m\}|$ with which j is associated, average number of businesses in $\{m\}$, average rating of businesses in $\{m\}$, and location n of j.
- *Graph* features include: degree centrality [6], average neighbor degree [4], PageRank score [20], clustering coefficient [15], and node redundancy [15]. These five features are populated for both user nodes u_i and business nodes b_j, whereas an additional shortest path feature is computed for the pairs of nodes (u_i,b_j). Note that the graph features are populated separately for the bipartite and tripartite graph representations.

[1] http://recsys.acm.org/recsys13/recsys-2013-challenge/

[2] Our use of the "tripartite graph" notation is slightly inconsistent with the canonic definition, such that the "bipartite graph with metadata nodes" notation would be more appropriate. However, for the sake of brevity we stick to the bipartite and tripartite terminology.

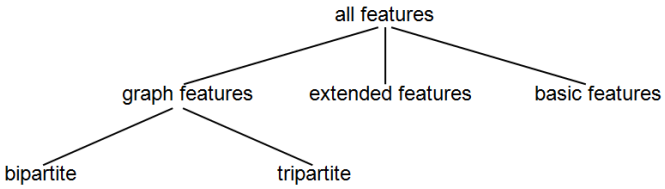


Figure 2. Feature classes

Degree centrality (or, simply, node degree) quantifies the importance of a node through the number of other nodes, to which it is connected. Hence, in the bipartite graph, degree centrality of a user node u_i is the activity of i, i.e., the number of businesses that i reviewed, and for a business node b_j it is the popularity of j, i.e., the number of users who reviewed it. In the tripartite graph, the number of categories $|\{m\}|$ of j plus 1 (business location) are added to degree centrality of b_j.

As the name suggests, *average neighbor degree* measures the average degree of nodes, to which a node is connected. In the bipartite graph, this metric communicates for u_i – the average popularity of businesses that i reviewed, and for b_j – the average activity of users who reviewed j. Note that in the tripartite graph, the average neighbor degree of b_j also incorporates the popularity of categories $\{m\}$ of j and the number of businesses sharing the same location n as j.

PageRank is a widely-used recursive metric that quantifies the importance of nodes in a graph. For a user node u_i, the PageRank score is computed through PageRank of the businesses $\{b_j\}$ which i reviewed, and, likewise, for a business node b_j – through PageRank of the users $\{u_i\}$ who reviewed j, of the categories with which j is associated, and of the location of j. In the tripartite graph, the PageRank scores of the business category nodes $\{c_m\}$ and of the location node l_n also affect PageRank of a business node b_j.

Clustering coefficient measures the density of a node's immediate subgraph as the ratio between the observed and possible number of cliques. Since cliques are impossible in the bipartite graph, clustering coefficient measures the density of "squares" in the graph, i.e., the portion of pairs of businesses j_x and j_y that are both reviewed by a pair of users i_a and i_b, and, respectively, the portion of pairs of users i_a and i_b that reviewed a pair of businesses j_x and j_y. Since no edges between u_i and $\{c_m\}$ (or, u_i and l_n) exist in the tripartite graph, clustering coefficient is meaningful only for the b_j nodes, where it is reduced to the bipartite variant.

Similarly, *node redundancy* shows what fraction of a node's pairs of neighbors are linked to the same other node. In the bipartite graph, node redundancy communicates for u_{i_a} - the portion of pairs of businesses that i_a reviewed that were both reviewed by another user i_b, and for b_{j_x} - the portion of pairs of users who reviewed j_x and also both reviewed another business j_y. In the tripartite graph, redundancy of business nodes also incorporates pairs of categories of j, with which some other businesses are associated as well.

Note that the graph based features are extracted and generated offline, i.e. not as part of the recommendation process, and, as such, the added computational overhead does not affect online recommendations.

Predicting Business Ratings

We apply the Random Forest regression model for the generation of the predictions of user ratings for businesses [7]. Random Forest is a popular ensemble classification algorithm that combines a set of binary decision trees. At the training stage, each tree is constructed using a portion of the training data and a subset of data features. Given a fixed set of features F that model the training data, log |F| features and about 2/3 of the training data are randomly selected by the algorithm to construct each tree. Within the forest trees, each node uses for the decision making only one feature $f \in$F, which is the top performing feature out of the selected subset of features.

At the classification stage, the test data items are run through all the trees in the trained forest. The class of a test item is determined by the majority voting of the terminal nodes reached when traversing the trees. In case of a regression model, which is applied for predictions of continuous values rather than discrete class labels, the predicted score is computed as a linear combination of the scores of the terminal nodes.

It should be noted that the ensemble of trees in Random Forest and the selection of the best performing feature in each node inherently eliminate the need for feature selection. Since every node uses for decision making a single top performing feature, the accurate predictive features get naturally selected in many nodes. Hence, these features have a strong impact on the classifier, such that the ensemble of multiple trees virtually substitutes the feature selection process. We refer the readers to [7] for an elaborate presentation of the Random Forest algorithm.

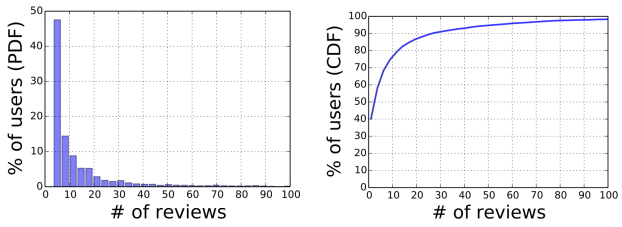
DATASET

The dataset used in this work is a public dataset released by Yelp for the ACM Recommender Systems Conference 2013 challenge. For our analysis, we filtered out users with less than 5 reviews (representing 21% of users), which results in 9,464 users providing 171,003 reviews and the corresponding ratings for 11,197 businesses.

Table 1 summarizes the basic statistics of users and businesses in the dataset. The average number of reviews per user stands at 18.07 (median=9) and the average number of reviews per business is 15.27 (median=5). Despite the high number of categories in the dataset, the average number of categories with which a business is associated is only 2.68. Every business is also associated with a single location.

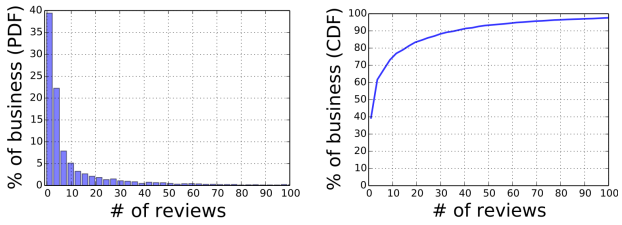
	mean	std	min	25%	50%	75%	max
#reviews per user	18.07	29.60	5	6	9	17	588
#reviews per business	15.27	32.43	1	3	5	13	528
#categories per business	2.68	1.14	0	2	2	3	10

Table 1. Basic data statistics



(a) (b)

Figure 3. PDF and CDF of number of reviews per user



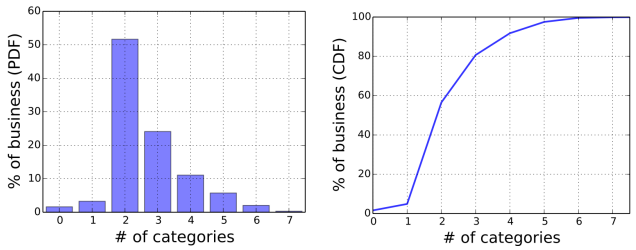
(a) (b)

Figure 4. PDF and CDF of number of reviews per business

Figure 3 illustrates the distribution of the number of reviews (and ratings) per user. The Cumulative Distribution Function (CDF) plot reveals a long tail distribution of the user degree with more than 75% of the users providing less than 10 reviews. Likewise, we observe in Figure 4 the distribution of the number of reviews per business. Only 24% of businesses attract more than 10 reviews, while only a few businesses (less than 2%) have a relatively higher number of reviews (more than 100). Hence, the cold-start problem manifests to some extent even in the filtered dataset.

Now, we characterize the distribution of business categories and locations in the dataset, which are considered as metadata information. Figure 5 illustrates the Probability Distribution Function (PDF) and CDF of the number of categories per business. We observe that the majority of businesses in the dataset are associated with less than three categories: more than half of the businesses have two categories and almost 25% have three. We also observe in Figure 6 that 97.6% of the businesses are located in the top-20 cities, with Phoenix alone being associated with 36% of businesses.

In order to evaluate whether the business category has any impact on the number of reviews and on the average rating, we show in Table 2 the top-25 categories and the corresponding average rating. The "Restaurants" category is by far the most popular with 4,467 businesses, where we observe on average 26.52 reviews per business with average rating of 3.45. Several other categories (marked in bold) obtain a higher average rating,



(a) (b)

Figure 5. PDF and CDF of number of categories per business

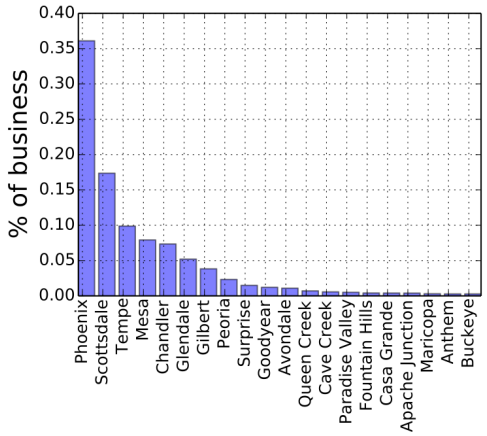


Figure 6. Location distribution of businesses across top-20 cities (overall, there are 62 cities)

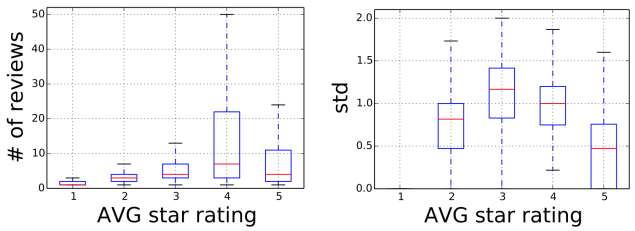
but have a substantially lower number of reviews per business. We computed the Spearman's Correlation coefficient between the average number of reviews per business and the average rating within the categories that are associated with 50 businesses or more. The computation shows a moderate correlation of 0.38, which suggests that the number of reviews per business influences the average rating assigned by the reviewers.

We then further study the impact of the number and the variability of reviews on the rating values. We examine in Figure 7(a) the distribution of the average rating as a function of the number of reviews that each business receives. We observe that although there is a trend slightly increasing from 1 to 4 with the increase of the median number of reviews, the distribution of the number of reviews is too skewed to argue for a clear impact. However, we observe in Figure 7(b), which shows the standard deviation of the average ratings, that both very high and low ratings generally imply a low deviation of ratings. This suggests that users mainly provide consistent ratings when reviewing businesses on the extreme sides of the scale, i.e., either very good or very bad businesses.

We also depict in Figure 8 the distributions of the average rating for users and for businesses, and the overall distribution of ratings in the dataset. More than 55% of the average user ratings range between 3 and 4. Combining this observation with the previously discussed distri-

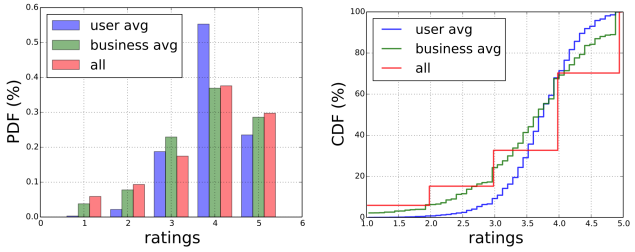
order	category	# business	avg reviews/business	avg rating
1	Restaurants	4467	26.52	3.45
2	Shopping	1646	7.15	3.75
3	Food	1606	16.53	3.77
4	Beauty & Spas	721	4.54	**3.97**
5	Nightlife	637	36.67	3.5
6	Mexican	623	23.25	3.47
7	Automotive	529	3.83	3.67
8	Bars	515	40.58	3.46
9	Active Life	504	8.73	**3.99**
10	American (Traditional)	476	27.52	3.32
11	Fashion	474	7.23	3.74
12	Pizza	453	23.96	3.44
13	Health & Medical	428	2.86	**3.99**
14	Event Planning & Services	419	9.34	3.66
15	Fast Food	384	7.26	3.1
16	Sandwiches	380	23.12	3.6
17	Home Services	369	3.01	3.66
18	Hotels & Travel	345	10.45	3.44
19	American (New)	339	52.79	3.58
20	Grocery	332	13.61	3.6
21	Coffee & Tea	322	20.09	3.77
22	Arts & Entertainment	298	22.06	3.87
23	Local Services	296	3.89	3.88
24	Chinese	287	19.16	3.36
25	Burgers	259	26.71	3.34

Table 2. Average number of reviews and average rating for top-25 business categories.



(a) (b)

Figure 7. Distribution of the number of review and of standard deviation of ratings



(a) (b)

Figure 8. PDF and CDF of user average ratings, business rating and all ratings

bution of ratings (almost 20% of user ratings are 3 and more than 35% are 4), we conclude that about half of the users consistently provide ratings between 3 and 4. This indicates that the average user rating could be an important indicator for rating predictions.

The observed average business rating distribution follows closely the overall rating distribution, with only a few businesses receiving a low rating. In fact, more than 80% of the average business scores are greater than 3. While in this case we do not observe a consistent behavior of users across rating different businesses, we posit that the average business rating could also be a valuable indicator to consider when predicting ratings.

RESULTS

In this section, we evaluate the accuracy of the recommendations and analyze the impact of graph features on the accuracy of the recommender. We use in the evaluation the Yelp dataset characterized in the previous section (recall that users who provided less than 5 ratings were excluded from the evaluation).

We perform a 5-fold cross validation and, therefore, split the ratings into 80% training and 20% test sets. For each fold, we train the predictive model using both the original features encapsulated in the review data and the new graph features. The basic and extended features are populated directly from the reviews. The graph features are populated from the bipartite and tripartite graph representations of the data and they are used to augment the basic and extended features.

We use an offline evaluation to optimize the parameters of the Random Forest regressor and set the number of trees to 100, and the number of features to select from at each node to 1. We measure the predictive accuracy of various combination of features (will be detailed in the next sub-section) using the RMSE metric and apply a paired t-test to validate statistical significance [24].

Graph Features Effectiveness

In this section, we study the contribution of the graph based features to the overall accuracy of the recommendations. We analyze how different types of features affect rating predictions and how they complement each other. Table 3 compares the RMSE values obtained by the recommender when different groups of features are used to train the Random Forest model. We show in the table four feature groups (basic, extended, bipartite, and tripartite), as well as some of their combinations.

First and foremost, we would like to highlight the closeness of the RMSE scores obtained by various combinations: the difference between the best and the worst performing combination is less than 10%. This is explained primarily by the low variance of user ratings, which was discussed in the previous section. Since most ratings given by a user are similar, they are highly predictable using simple methods like user/business average that perform reasonably accurately. This phenomenon is not peculiar to the Yelp dataset and got widely recognized in the Netflix Prize challenge [14]. As such, the complex mechanism of Random Forest has only a confined space for improvement.

Directly comparing the standalone performance of the two groups of graph features, bipartite vs. tripartite, we observe that the bipartite features produce more accurate predictions than the tripartite ones. When both bipartite and tripartite features are combined into graph features, there is a further slight improvement in performance over each of the two groups individually. Although the improvement is modest, it is statistically significant, $p<0.001$. This suggests that the bipartite fea-

features combination	features	RMSE
all_features	basic ∪ extended ∪ graph	1.076667
allexcept_tripartite	basic ∪ extended ∪ bipartite	1.077535
allexcept_basic	extended ∪ graph	1.082222
extended_and_bipartite	extended ∪ bipartite	1.085074
allexcept_bipartite	basic ∪ extended ∪ tripartite	1.089689
extended_and_tripartite	extended ∪ tripartite	1.107377
allexcept_extended	basic ∪ graph	1.109540
graph	bipartite ∪ tripartite	1.114891
allexcept_graph	basic ∪ extended	1.117572
bipartite		1.118867
tripartite		1.132601
basic		1.180921
extended		1.185396

Table 3. RMSE per selected feature combinations

tures may benefit, albeit minimally, from the availability of the tripartite features.

We now analyze how graph features contribute to the overall accuracy of the predictions. As expected, the best performance is achieved when all the groups of features are combined. By observing the accuracy differences when various groups of features are combined, there is a number of findings that support our hypothesis that graph features overall contribute to the accuracy of the recommendations. Firstly, when analyzing the performance of each group of features, we conclude that the two graph features (bipartite and tripartite) perform noticeably better than the other two groups features (basic and extended). The combination of graph features outperforms slightly, although statistically significantly, the combination of the basic and extended features.

When analyzing the impact of each feature group as a whole, we exclude a group from the overall set of features and measure the difference in performance with respect to the all_features run. We refer to these variants as allexcept_*group*, where *group* is the combination of features that is excluded from the computation. We observe that when graph features are excluded, the predictions are less accurate than when the basic and/or extended features are excluded. This indicates that graph features do provide additional information, which is not covered by the basic and extended features, and this information improves the accuracy of the generated recommendations.

There is also further evidence that the combination of bipartite and tripartite features is beneficial: we observe that the exclusion of tripartite features has a minor impact on the accuracy (allexcept_tripartite), whereas the exclusion of bipartite features (allexcept_bipartite) has a stronger impact. However, it should be noted that when both the groups of graph features are excluded (allexcept_graph), the impact on the accuracy is much stronger, which suggests that their combination benefits the system more than each one of them individually.

In order to evaluate the significance of the results, we perform a paired t-test using the RMSE values obtained for the various group combinations. The vast majority of differences are significant, $p<0.001$, whereas those that are not significant are highlighted in Figure 9. Two conclusions can be drawn from the failed tests: (1) the performance of bipartite features is not different from the performance of graph features, which indicates the prevalence of the bipartite features over the tripartite ones, when these two groups are considered individually; and (2) the combination of extended and bipartite features produces results that are not different from a combination of all the groups of features, which indicates that extended and bipartite features are the most important groups of features for the recommender.

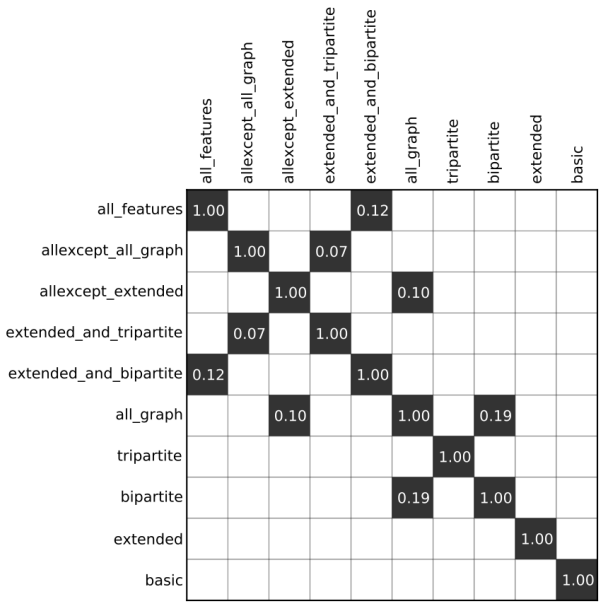


Figure 9. Significance of various combinations (white cells - significant, dark cells - not significant with p-value)

Delving one level deeper, we analyze the contribution of individual features to the accuracy of the predictions. For this, we analyze the importance scores of the individual features, as computed by the Random Forest model. Table 4 summarizes the importance scores of the top features. We observe that two most important features are, as expected, user and business average ratings. We would like to highlight that these two features account together to more than 43% of the overall feature importance. The third and fourth features are related, respectively, to the tripartite and bipartite graph representations. This is an important finding, which suggests that features from both representations are within the short list of features considered by Random Forest. The fifth feature, business review count, is related to business popularity, which is also important in the context of rating predictions. Note that the PageRank score in the bipartite and tripartite graphs are ranked fourth and sixth, respectively, which means that the topography of the graphs provides a valuable information.

feature	import.	feature group
business_avg_rating	0.2228	extended
user_average_stars	0.2085	extended
business_tripartite_avg_ne_deg	0.0467	tripartite
business_bipartite_pagerank	0.0457	bipartite
business_review_count	0.0410	extended
business_tripartite_pagerank	0.0394	tripartite
business_bipartite_clustering_coeff.	0.0337	bipartite
business_main_category_degree	0.0334	extended
business_main_category_avg_stars	0.0313	extended
business_main_category	0.0311	extended
business_avg_degree_of_categories	0.0296	extended
business_bipartite_degree_centrality	0.0283	bipartite
business_avg_stars_of_categories	0.0264	extended
business_tripartite_degree_centrality	0.0255	tripartite
business_bipartite_avg_ne_deg	0.0203	bipartite
business_bipartite_node_redundancy	0.0152	bipartite
user_bipartite_avg_ne_deg	0.0147	bipartite
user_bipartite_node_redundancy	0.0129	bipartite

Table 4. Relative importance of individual features for all_features combination

While feature importance scores in Table 4 show that the average user and business ratings are pivotal for accurate predictions, we posit that these may not perform well when predicting ratings for businesses with high variability of ratings. We will investigate this question in the following subsection.

Robustness to Variability

In this experiment we assess the robustness of various groups of features when predicting ratings for businesses with a high variability of ratings. For this, we split the businesses to equally sized buckets based on the standard deviation (STD) of the business ratings, and compute the RMSE scores for each feature combination and each bucket of businesses. The results of this experiment are summarized in Figure 10, where only a small selection of most interesting for our analysis feature combinations is shown. The left columns refers to buckets with low STD of ratings and right columns to buckets with high STD.

The experiment shows that the behavior of the graph features differs from the behavior of the basic and extended features extracted from the original rating data. While graph features achieve a low accuracy in the low STD buckets, i.e., they struggle to generate accurate predictions for businesses with stable ratings, they perform remarkably well in the high STD buckets, where dominant features like average user/business rating included in the extended features group, struggle to generate accurate predictions. The all_features combination manages to effectively balance between the benefits of the extended and graph features and across the board achieves the highest overall accuracy. The difference in performance in the high STD buckets between all_features and allexcept_graph features clearly shows that the graph features complement data-driven features by allowing for more accurate predictions to be generated for businesses with highly variable ratings.

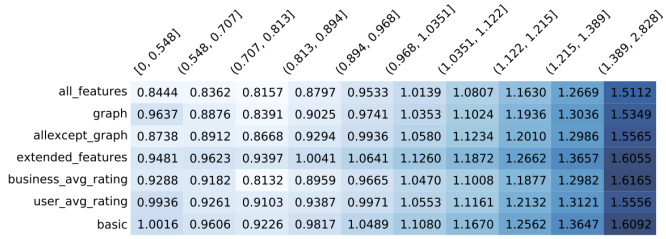


Figure 10. RMSE of the buckets (by STD) for selected combinations

In summary, graph features seem more robust when predicting ratings for businesses with variable ratings. However, how is their performance affected by the rating data sparsity? In the next section we will investigate the performance of the graph features at various levels of data sparsity, specifically, when a little rating data is available about a user/business.

Robustness to Sparsity

In this experiment we evaluate the performance of the recommender when predicting ratings for users/businesses with a low number of training ratings. We follow a methodology similar to the one used in the previous sub-section and split users/businesses into equally sized buckets. But this time, the split is done according to the number of ratings available. Figure 11 summarizes the RMSE scores obtained for each feature combination and each bucket of users, while Figure 12 focuses on businesses. The left columns refers to buckets with high sparsity of ratings and right columns to buckets with low sparsity.

Generally, the performance of the recommender improves as more ratings are available for a user/business, and this trend is consistent across all the evaluated combinations of features. Again, we can clearly see the contribution of the graph features. Comparing the performance of the all_features and the allexcept_graph combinations, we observe that the graph features complement other data-driven features and improve the performance of the recommender. Adding graph features yields a noticeable improvement in the business split, while also in the user split the effect is positive. Although other features such as the business and user average ratings perform well overall, they perform poorly when there is little training data. Focusing on the important predictive features, we notice that graph features outperform across the board individual features like user and business averages, which can be observed both for the user and business split.

Finally, we carry out an experiment that focuses on the business cold-start use case.[3] To this end, we split the businesses into 2 groups: those having more training ratings than test ratings and vice versa. We measure the accuracy of the ratings predictions using all the features

[3]The user cold-start evaluation is impossible, as we filtered from our dataset users with fewer than 5 ratings.

	[4, 5]	(5, 8]	(8, 11]	(11, 17]	(17, 25]	(25, 39]	(39, 58]	(58, 83]	(83, 140]	(140, 471]
all_features	1.2020	1.1693	1.1345	1.1181	1.0813	1.0599	1.0219	1.0177	0.9962	0.9130
graph	1.2447	1.2107	1.1694	1.1600	1.1158	1.0981	1.0555	1.0530	1.0305	0.9561
allexcept_graph	1.2545	1.2243	1.1807	1.1632	1.1230	1.0976	1.0650	1.0540	1.0349	0.9547
extended_features	1.3026	1.2812	1.2498	1.2287	1.1867	1.1700	1.1411	1.1253	1.1091	1.0111
business_avg_rating	1.2695	1.2206	1.1893	1.1765	1.1441	1.1209	1.0837	1.0839	1.0630	0.9914
user_avg_rating	1.2755	1.2462	1.1862	1.1789	1.1355	1.1156	1.0738	1.0807	1.0485	0.9839
basic	1.3000	1.2671	1.2276	1.2220	1.1810	1.1724	1.1212	1.1299	1.1081	1.0330

Figure 11. RMSE of the buckets (by number of user ratings) for selected combinations

	[0, 3]	(3, 9]	(9, 17]	(17, 27]	(27, 38]	(38, 53]	(53, 74]	(74, 103]	(103, 169]	(169, 440]
all_features	1.2894	1.1258	1.0727	1.0711	1.0700	1.0473	1.0238	1.0350	1.0125	0.9697
graph	1.3334	1.1983	1.1352	1.1164	1.1097	1.0801	1.0499	1.0551	1.0261	0.9867
allexcept_graph	1.3861	1.1513	1.0980	1.1078	1.1060	1.0802	1.0658	1.0813	1.0577	1.0109
extended_features	1.4514	1.1965	1.1536	1.1718	1.1746	1.1576	1.1388	1.1511	1.1330	1.0647
business_avg_rating	1.5984	1.1805	1.1050	1.0939	1.0931	1.0667	1.0419	1.0531	1.0206	0.9745
user_avg_rating	1.3434	1.2155	1.1604	1.1379	1.1392	1.1049	1.0797	1.0818	1.0587	1.0119
basic	1.3976	1.2656	1.2007	1.1703	1.1686	1.1469	1.1233	1.1304	1.0901	1.0617

Figure 12. RMSE of the buckets (by number of business ratings) for selected combinations

(all_features) and all the features except for the graph features (allexcept_graph). The results of this experiment are shown in Figure 13.

We observe that in the setting, where there are more training than test ratings, i.e., enough training data for informed predictions, the accuracy of the predictions in the all_features and allexcept_graph cases is comparable. However, when there are more test ratings than training ratings, i.e., in the business cold-start setting, the exclusion of graph features degrades the accuracy of the predictions, as shown by the lower CDF curve starting from approximately RMSE=1.

Hence, we conclude that in addition to strengthening the robustness to rating variability, graph features also strengthen the robustness to data sparsity and unavailability of sufficient training data. This is an important finding, which indicates that graph features have the potential to alleviate the cold-start problem of recommender systems.

RELATED WORK

Applying machine learning and data mining techniques to user modeling and recommender systems applications has been the subject of many early studies [23, 28, 21, 27]. The two major challenges for applying machine learning techniques to those tasks, small datasets and lack of labeled data, are less of an issue nowadays [3]. Large and labeled datasets are being publicly released quite often, e.g., the Netflix Prize dataset, Movielens, Yelp dataset that was used in this work, and many other datasets.

Larger and richer datasets bring with them the possibility of using more complex machine learning techniques

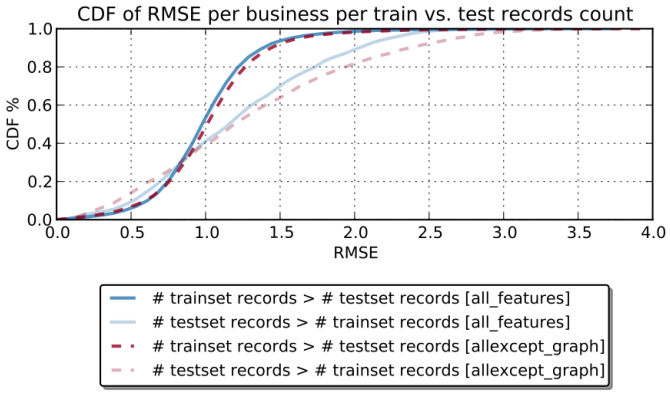


Figure 13. CDF of RMSE for business predictions. Businesses split based on the number of ratings in training and test set

that take into consideration a large set of features. The Netflix Prize winning team, for example, modeled the temporal dynamics, confidence levels, and implicit feedback features using the supplied dataset, and applied matrix factorization [14]. An ensemble method was then used to combine those features into a single model. The effectiveness of having more data and more features facilitates machine learning techniques achieving accurate results, which was discussed in [11]. Since extracting features from the data is considered a challenge [3], studies looked into ways of automatically engineering and populating features.

Previous work has investigated automatic feature generation by combining existing features using arithmetic functions such as min, max, average, and others [18]. In that work, a specific language for defining features was presented, where a feature was described by a set of inputs, their types, construction blocks, and the produced output. A framework for generating a feature space using the feature language as input was evaluated. The evaluation showed that the framework outperformed legacy feature generation algorithms in terms of accuracy. The main difference between the presented framework and its predecessors was that the framework was generic and applicable to multiple tasks and machine learning approaches. A review of other key feature generation methods was also provided, including task-oriented feature generation approaches and other construction methods (for instance, using boolean combinations [13]).

Only a few previous works incorporated graph features into machine learning and data mining applications [16, 17]. In [16], the recommendation problem was defined as a link prediction problem, and a similarity score between users and items nodes was computed using random walks. Items were then ranked based on their similarity scores, and top scoring items were recommended to users. Compared to other non-graph based similarity ranking methods, this approach was shown to outperform others using the True Positives vs. False Negatives

metric. A similar random walk metric was used in [17], complemented by additional graph metrics based on the graph structure. These metrics were used for the purpose of link prediction and property values prediction in semantic descriptive graphs (RDF) using an SVM-based machine learning technique. Experimental results showed that the graph structure features in use were competitive to other graph structure features. It was also noted that the new defined features were not dataset-specific but could be applied to any RDF graph, while previously known structure based features were dataset-specific.

Our work also defines dataset agnostic graph-based features. However, the studied features are generic and not dependent on a known graph schema, such as RDF. In addition, we extend the set of generated features beyond random walks and tree structures, with a variety of metrics based on local neighborhood and global popularity. Finally, our evaluation focuses on the effect of the generated features on rating prediction and recommendation. Since the recommendation domain is tightly connected to graphs (e.g., social recommendations, people recommenders, nearest neighbors notion of collaborative filtering, and so forth), it is natural to evaluate the contribution of the graph based features to the accuracy of ratings predictions, and, in turn, of the generated recommendations.

Regarding the dataset used in this work, Yelp's service and other published or proprietary datasets have been explored in several works. Although these works are not directly related to predicting ratings or recommending, they may shed more light on the service and the released datasets. In [12] the motivations for using Yelp was discussed. Key usage patterns that emerged from an online user study were: to retrieve information regarding businesses and to serve as a form of entertainment (by engaging in a collaborative reviewing of businesses). Another work explored a different aspect of Yelp, by studying its mechanism for filtering out fake reviews [19]. According to the evaluation results, which were subsequently confirmed by Yelp, approximately a quarter of the submitted reviews were filtered due to suspicion of being fake. Another study examined the effect that a site like Groupon had on business reviews in Yelp [8]. Their results showed that, contrary to the common belief, users of Groupon provided more balanced and detailed reviews than other Yelp users.

CONCLUSIONS

In this work, we examined how additional features can be extracted from a graph-based representation of a user-to-item rating dataset. Using the state-of-the-art machine learning methods and widely-used graph theory metrics, we designed, implemented, and evaluated a model that was applied to predict user ratings. We validated our approach using a publicly available dataset of user reviews for businesses.

The evaluation showed that when augmenting basic data-driven features with graph features (considering both bipartite or tripartite graph representation models) improved the accuracy of the generated predictions. We verified that the studied graph features were robust to data variability and sparsity. This could be credited to the augmentation of the graph structure, be it a user vertex, a business vertex, or an edge, that captured an intrinsic valuable information that might not be uncovered otherwise due to data sparsity.

We observed that bipartite graph features were superior to the tripartite ones and led to a higher accuracy of the rating predictions, which suggested that the use of a more complex graph representation might also introduce noise. Combining the two representations, however, yield a higher accuracy, such that the best overall results were achieved when combining the graph features with other data-driven features.

Although, as observed in this paper, the ratings in the Yelp dataset in use did not exhibit variance high enough to expect a substantial improvement in accuracy, it was interesting to note that graph features did improve the accuracy of predictions for businesses with a high variability of ratings.

The graph features used in this work were deliberately kept simple, although more complex features can potentially yield more accurate predictions. Despite this, the obtained results demonstrate the effectiveness of the graph representation of the ratings data and the benefits of augmenting the original data-driven features with the graph features. The graph representation of the data is generic enough to allow the applicability of the proposed technique to other datasets, which argues in favor of the generalization of the graph features and their necessity to improve the recommendation accuracy. Nevertheless, more sophisticated graph features extraction may result in a better representativeness of the data, and, in turn, in a better predictive accuracy. Analyzing how different types of data are influenced by the graph features is left as future work.

Another modification that remains beyond the scope of this work pertains to graph representation incorporating user-to-user and business-to-business relationships. Although neither the bipartite nor the tripartite graph allow edges within the user/business subgraphs, these edges naturally exist, e.g., user friendship and business similarity. A new method for modeling these edges should be developed and their contribution should be evaluated.

Also, in our work we did not label any of the graph edges. However, additional data pertaining to the graph edges is available, e.g., numeric scores of the reviews, strength of ties between users, or domain metadata relationships. If incorporated into the prediction mechanism, both the within-partition edges and their type

may affect the graph features and potentially improve the accuracy of the predictions.

REFERENCES

1. G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.
2. X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol. Data mining methods for recommender systems. In *Recommender Systems Handbook*, pages 39–71. 2011.
3. M. Anderson, D. Antenucci, V. Bittorf, M. Burgess, M. J. Cafarella, A. Kumar, F. Niu, Y. Park, C. Ré, and C. Zhang. Brainwash: A data system for feature engineering. In *CIDR*, 2013.
4. A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani. The Architecture of Complex Weighted Networks. *PNAS*, 2004.
5. S. Berkovsky, T. Kuflik, and F. Ricci. Cross-technique mediation of user models. In *AH*, pages 21–30, 2006.
6. S. P. Borgatti and D. S. Halgin. Analyzing Affiliation Networks. *The Sage handbook of social network analysis*, 2011.
7. L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
8. J. W. Byers, M. Mitzenmacher, and G. Zervas. The groupon effect on yelp ratings: A root cause analysis. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 248–265. ACM, 2012.
9. L. Chen and P. Pu. Critiquing-based recommenders: survey and emerging trends. *User Model. User-Adapt. Interact.*, 22(1-2):125–150, 2012.
10. A. Hagberg, P. Swart, and D. S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, LANL, 2008.
11. A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *Intelligent Systems, IEEE*, 24(2):8–12, 2009.
12. A. Hicks, S. Comp, J. Horovitz, M. Hovarter, M. Miki, and J. L. Bevan. Why people use yelp. com: An exploration of uses and gratifications. *Computers in Human Behavior*, 2012.
13. Y.-J. Hu and D. Kibler. Generation of attributes for learning algorithms. In *AAAI/IAAI, Vol. 1*, pages 806–811, 1996.
14. Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
15. M. Latapy, C. Magnien, and N. D. Vecchio. Basic notions for the analysis of large two-mode networks. *Social Networks*, 2008.
16. X. Li and H. Chen. Recommendation as link prediction: a graph kernel-based machine learning approach. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 213–216. ACM, 2009.
17. U. Lösch, S. Bloehdorn, and A. Rettinger. Graph kernels for rdf data. In *The Semantic Web: Research and Applications*, pages 134–148. Springer, 2012.
18. S. Markovitch and D. Rosenstein. Feature generation using general constructor functions. *Machine Learning*, 49(1):59–98, 2002.
19. A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance. What yelp fake review filter might be doing. In *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.
20. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999.
21. M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, 27(3):313–331, 1997.
22. M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.
23. W. Pohl. Labour-machine learning for user modeling. In *HCI (2)*, pages 27–30. Citeseer, 1997.
24. G. Shani and A. Gunawardana. Evaluating recommendation systems. In *Recommender Systems Handbook*, pages 257–297. Springer, 2011.
25. A. Tiroshi, S. Berkovsky, M. A. Kaafar, T. Chen, and T. Kuflik. Cross social networks interests predictions based on graph features. In *RecSys*, 2013.
26. S. Trewin. Knowledge-based recommender systems. *Encyclopedia of library and information science*, 69(Supplement 32):69, 2000.
27. G. I. Webb, M. J. Pazzani, and D. Billsus. Machine learning for user modeling. *User modeling and user-adapted interaction*, 11(1-2):19–29, 2001.
28. I. Zukerman and D. W. Albrecht. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11(1-2):5–18, 2001.