Predicting Business Ratings on Yelp

Travis Gingerich
travisg@stanford.edu

Yevhen Bochkov
ebochkov@stanford.edu

***Abstract*—Matrix factorization is an extensible method for predicting ratings in a user-item matrix. This paper explores the use of matrix factorization in predicting business ratings on Yelp.**

I. Introduction

Recommender systems are vital for many modern services. Good personalized recommendations add another dimension to the user experience, enhancing user satisfactory and loyalty.

Most recommender systems are based on one of two strategies. The first is called content filtering and is based on analyzing an explicit product profile. An alternative to content filtering is collaborative filtering, which relies only on past user behavior – previous transactions or product ratings, without requiring the creation of explicit profiles and called collaborative filtering. Collaborative filtering analyzes relationships between users and interdependencies among products to identify new user-item associations.

There are two approaches in collaborative filtering – neighborhood methods and latent factor models. Neighborhood methods are focused on computing the relationships between items or between users. Latent factor models try to explain user preferences (ratings) by characterizing both items and users on some limited number of factors inferred from the rating patterns. These factors can be fetched from the some low-dimensional user-item space representation.

Some of the most successful realizations of latent factor models are based on matrix factorization. Such methods combines good scalability with predictive accuracy and in this work we will focus on them.

The problem can be formalized in following way. Given matrix $R \in \mathbb{R}^{m \times n}$ of user ratings of items, where m - number of users, n - number of items, r_{ui}, $(u, i) \in K$, rating of u user of i item. K - set of given ratings, $u \in [1, m]$, $i \in [1, n]$. Required give estimation $\hat{r}_{ui}$, such that $(u, i) \notin K$.

II. Related Work

A general overview of approaches to matrix factorization method is described in [2], including information on basic variations such as including bias terms and additional features.

Increasing of types of information available for analysis leads to new methods that try to leverage these rich information sources to improve performance. Some works try to design a specific model for each scenario, which demands great efforts in developing and modifying models. Others try to extend matrix factorization model by incorporating different sources of information into it. Thus, an abstract framework called feature-based matrix factorization model is presented in [1]. It allows the creation of additional matrix factorization models that utilize new types of information by defining new features, without modifying the underlying algorithm or code.

Recently, the incorporation of social relationships within the framework of recommender systems using collaborative filtering and matrix factorization has emerged. A method that represents social constraints on recommender systems is discussed in [5]. It shows how to design a matrix factorization objective function that includes social regularization term, the goal of which is to limit the amount of variation in ratings between users that have meaningful social connections.

Another approach that allows the combination of different information sources is described in [7]. To take advantage of the heterogeneity of the information network, the authors first diffuse user preferences along different meta-paths in the network, or with other attribute based item similarity semantics. Then matrix factorization techniques are used to calculate the latent features for users and items accordingly. Each set of latent features represent one recommendation factor with a specific semantic. A Bayesian ranking based recommendation model is then used to combine these recommendation factors.

III. Dataset and Features

We choose the Yelp dataset [6] for our research as a modern, information-rich dataset. It includes following entities: businesses (items), users, reviews (ratings), tips, and check-ins.

Businesses are characterized by location information (city, state, address, coordinates), neighborhoods, average rating, review count, working schedule, category (what kind of business - restaurant, dental clinic) and various additional attributes (these vary greatly in type; for example, they range from "Accepts credit cards" to "Good for kids").

Users are characterized by review counts and average ratings, votes (votes of user's reviews by other users), lists of friends, and lists of compliments.

A review is a rating of a business by a user. It also contains review text and a count of different kind of votes of this review by other users.

We used a subset of the Yelp Dataset Challenge data to perform our analysis and training of the models, allowing us to iterate over models more quickly and allowing easier development and experimentation. In order to ensure reasonable overlap between users and businesses and to maintain a reasonable number of social connections within the subset, we chose a portion of reviews by geographic location, taking the set of all reviews in the state of Pennsylvania. Some statistic about selected subset is presented in table I. Review

TABLE I
WORKING DATASET STATISTIC

Number of Users	17799
Number of Businesses	3041
Number of Ratings	66116
Avg number of Ratings per Business	21.74153239
Avg number of Ratings per User	3.714590707
Avg number of Friends per User	0.719929468

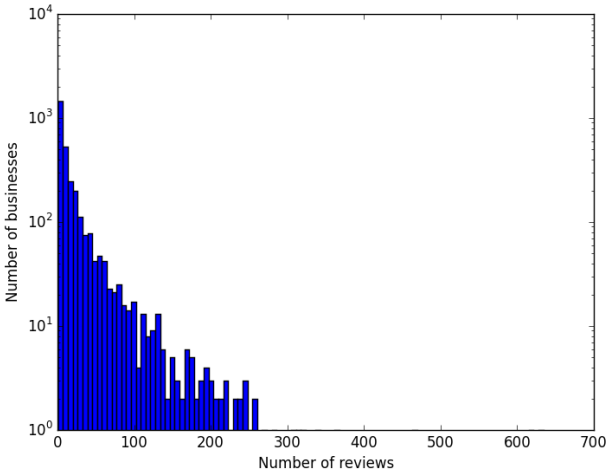

Fig. 1. Number of reviews per business

distributions and friend count distributions can be found in figures 1 and 2. As we can clearly see, a key feature of the data is sparseness of reviews.

IV. METHODS

As discussed in the introduction, business ratings are formulated as a $R \in \mathbf{R}^{m \times n}$ matrix, where m is the number of users and n is the number of businesses in the dataset. We

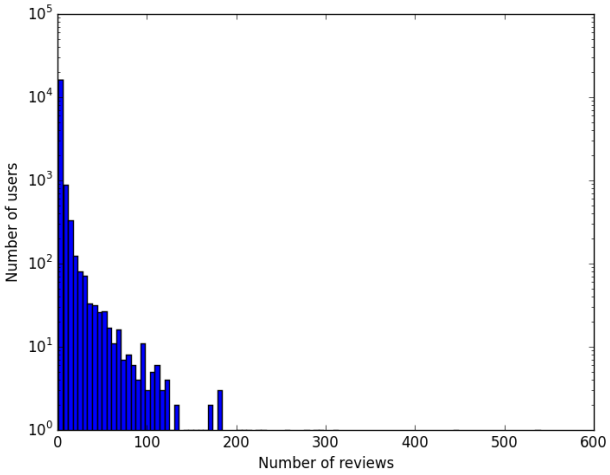

Fig. 2. Number of reviews per user

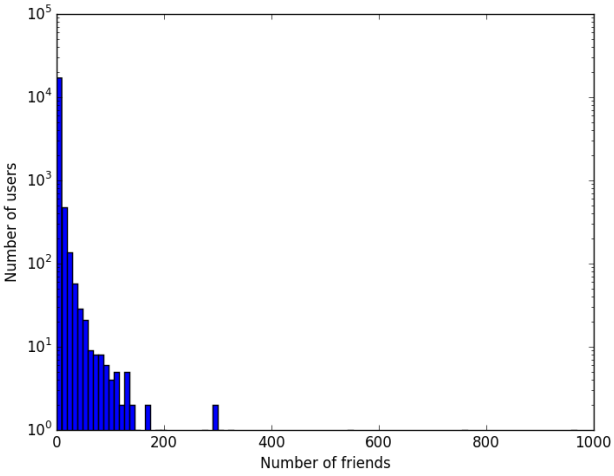

Fig. 3. Number of friends per user

hide a number of known ratings from this matrix, and attempt to reproduce them using several methods.

A. Baseline method

In order to interpret the success of our algorithms, we use a relatively simple baseline metric that we expect our more advanced methods to out-perform. A first-order approximation of the ratings of all businesses would be to predict the average rating over all reviews. Beyond this, one would expect each business to have an average rating around which individual users' ratings could vary slightly. Additionally, it's reasonable to expect that individual users would tend to have a bias in terms of how positively or negatively they rate businesses, across all businesses that they rate.

Taking this into account, the following simple baseline metric, suggested in [2]:

$$\hat{r}_{ui} = \mu + b_u + b_i \tag{1}$$

This states that the predicted rating $\hat{r}_{ui}$ for business/item i by user u is given by the sum of the global review average score μ, plus bias terms b_u for each user, and b_i for each business. These are given by the following equations:

$$\begin{aligned} \mu &= \frac{1}{|K|} \sum_{r_{ui} \in K} r_{ui} \\ b_u &= \frac{1}{|K_u|} \sum_{r_{ui} \in K_u} r_{ui} - \mu \\ b_i &= \frac{1}{|K_i|} \sum_{r_{ui} \in K_i} r_{ui} - \mu \end{aligned} \tag{2}$$

where K is the set of all known ratings, K_i is taken to be the set of all known ratings of business i, and K_u is taken to be the set of all known ratings by user u.

B. Basic matrix factorization and matrix factorization with biases

Matrix factorization is a technique that aims to produce a more nuanced prediction of the ratings of businesses by users by identifying a set of latent factors that describe both each user's preferences and each business' characteristics. The

general idea is that the known entries in the user-item rating matrix R can be approximated by the product of two matrices, $P \in \mathbf{R}^{m \times f}$ and $Q^T \in \mathbf{R}^{f \times n}$ where f is the number of chosen latent factors. The matrices P and Q are chosen to minimize some error function on the predicting rating matrix, which is found as follows:

$$\hat{R} = PQ^T \tag{3}$$

While there is a strong conceptual case for this model, it does fail to account for the biases that result in the success of the baseline metric identified in equation 1. Incorporating these biases results in the following formulation:

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i \tag{4}$$

where once again, μ is the average review score over all known reviews, and b_u and b_i are bias terms for each user and item/business, respectively. In this model, b_u and b_i can either be calculated as given in equation 2, or they can also be fit to the data to minimize some error measure.

C. Feature-based matrix factorization

1) Problem formulation: A useful framework for conceptualizing the model given in equation 4 is feature-based matrix factorization, as presented in [1]. The model is formulated as follows:

$$\hat{r}_{ui} = \mu + (\sum_j b_j^{(g)} \gamma_j + \sum_j b_j^{(u)} \alpha_j + \sum_j b_j^{(i)} \beta_j) + (\sum_j p_j \alpha_j)^T (\sum_j q_j \beta_j) \tag{5}$$

Similarly, in this formulation μ is the global average, and p_j and q_j are user and business/item latent factors. The b_j terms represent biases. This formulation introduces the concept of additional features for users and items, as well as "global" features. The global, user, and item features are represented by the γ_j, α_j, and β_j terms, respectively.

The advantage of this formulation is that it can be used to represent many variations of matrix factorization. For example, in order to implement the basic matrix factorization equation (with bias) given in equation 4, the following parameterizations of γ_j, α_j, and β_j suffice:

$$\gamma = \emptyset, \alpha_k = \begin{cases} 1 & k = u \\ 0 & k \neq u \end{cases}, \beta_k = \begin{cases} 1 & k = i \\ 0 & k \neq i \end{cases} \tag{6}$$

We use this formulation to create a model using basic matrix factorization (with bias).

Two additional models we evaluate use this same framework, with the addition of additional item/business and user features. In adding additional features concerning the business, we take the top 10 most frequently occurring business categories in the training set and add additional business features β corresponding to these categories. That is, for business i, we set

$$\beta_k = \begin{cases} 1 & k = i \\ w \frac{1}{|C_i|} & k \in C_i \\ 0 & \end{cases} \tag{7}$$

TABLE II
MODEL PERFORMANCE

Model	Train	Test	1-5	5-10	10-20	20-50	50+
Baseline	0.877	1.125	1.648	1.194	1.078	1.002	0.894
MF	0.930	1.037	1.347	1.190	1.011	0.983	0.884
RMF	0.932	1.037	1.348	1.190	1.011	0.984	0.880
BMF	0.993	1.036	1.378	1.210	1.005	0.982	0.884
UMF	0.993	1.037	1.381	1.212	1.004	0.982	0.884

where w is some weight controlling relative contribution of the business' latent factors and the latent factors representing business categories, and C_i is a set of indices corresponding to business categories.

The third model we implement incorporates information about the social network present on Yelp into our minimization formulation and our predictions. Taking inspiration from [5], for user u we set

$$\alpha_k = \begin{cases} 1 & k = u \\ w \frac{1}{|F(u)|} & k \in F(u) \\ 0 & \end{cases} \tag{8}$$

where $F(u)$ is the set of friends of user u and w is some weight. In a similar manner to the method taken in [5], this models the assumption that friends have similar tastes by introducing factors that encourage friends' predicted scores to not vary too far from each other.

2) Fitting parameters: In order to find the biases and latent factors, we use batch stochastic gradient descent as suggested in [1]. We also incorporate a momentum term (incorporating a fraction of the previous update vector in each iteration) to speed learning, as suggested in [4]. In addition, we incorporate regularization terms to mitigate overfitting. An additional step we take to avoid overfitting is stopping the stochastic gradient descent process early, as suggested in [3]. Parameters for learning rate, momentum, regularization, and the number of iterations after which to halt gradient descent were set by testing performance on a subset of the data not used for training or final testing. Batch size was set at 3000 by profiling the program and optimizing for execution time.

D. Evaluation

Model performance is evaluated using the RMSE:

$$RMSE = \sqrt{\frac{1}{n} \sum_{(u,i) \in S} (\hat{r}_{ui} - r_{ui})^2} \tag{9}$$

where S is the set of ratings in the test set. Simple cross validation was used to train/test the data. We select approximately 20% of the ratings to be withheld as a test set. Instead of selecting at random across all known reviews, we selected 20% of reviews on a per-user basis, ensuring even distribution of the test set across users with different numbers of reviews.

V. RESULTS

A. Model error

The final results from the 4 methods discussed are given Table II. The models shown are the baseline model (Baseline),

the basic matrix factorization model with bias but without regularization (MF), the basic matrix factorization model with bias and regularization (RMF), matrix factorization with business features (BMF), and matrix factorization with user friendship features (UMF). We evaluated each method on the training set, the test set, and several subsets of the training set that only include users and businesses with certain numbers of reviews.

B. Parameters

As mentioned in section IV, parameters were set by evaluation on a subset of the data. For all matrix factorization methods, the number of latent factors was set at 5

1) Baseline: The baseline metric has no parameters to set.

2) Matrix factorization: The learning rate was set to 0.0005, with a momentum term of 0.5. Batch stochastic gradient descent was run for 200 iterations.

3) Matrix factorization with regularization: The learning rate was set to 0.0005, with a momentum term of 0.5. The regularization term for updates to P and Q was set at 0.02, and the regularization term for updates to the bias terms was set at 0.001. Batch stochastic gradient descent was run for 200 iterations.

4) Matrix factorization with business category features: The learning rate was set to 0.0002, with a momentum term of 0.5. The regularization parameter for updates to P and Q was set at 0.02, and the regularization for the bias terms was set to 0.001. Batch stochastic gradient descent was run for 300 iterations. The parameter w weighting business features in equation 7 was set to 0.05.

5) Matrix factorization with user friendship features: The learning rate was set to 0.0002, with a momentum term of 0.5. The regularization parameter for updates to P and Q was set at 0.02, and the regularization for the bias terms was set to 0.001. Batch stochastic gradient descent was run for 300 iterations. The parameter w weighting user features in equation 7 was set to 0.1.

C. Discussion

1) Baseline: The baseline metric performs surprisingly well on the data set as a whole, especially when error is measured on reviews for which both the user and business have at least 50 known reviews. This illustrates that bias explains a surprisingly large portion of reviews. Although the baseline model performs worse than all other models evaluated, it performs extremely poorly when predicting ratings for businesses and users with low numbers of reviews (1-5 reviews).

An interesting characteristic of the baseline model is that it performs much better on the training set than it does on the test set. In general, this indicates that a model is overfitting the data. Initially, this appears to be a surprising finding, as the baseline model has a relatively low number of parameters to fit to the data. However, the high error on predicting ratings with a low number of known ratings for the target user or business hints that overfitting may still be possible, as even though there are very few parameters to fit to the data, there are also very few data points to fit the parameters to.

2) Matrix factorization: The matrix factorization method performs significantly better than the baseline model across all portions of the test set for which error was evaluated. It performs significantly better than the baseline method for users and businesses with low numbers of known reviews. It also overfits to the data to a lesser degree. Although this model has more parameters in terms of the latent factors that can be fitted to the test set, stopping gradient descent after a fixed number of iterations seems to mitigate their effect. In addition, the process through which latent factors are set allows the model to more accurately represent users interests and business characteristics even with a low number of known ratings for a particular user or business.

3) Regularized matrix factorization: Surprisingly, adding regularization to the basic matrix factorization model did not significantly alter performance. It did increase performance slightly when predicting ratings for users/businesses with a large number of known ratings. It may be that the strategy of halting gradient descent early already provides enough protection against overfitting.

4) Matrix factorization with business features: When fitting parameters to the matrix factorization with business features model, it became apparent that the model is prone to overfitting. Initially, the top 15 business categories were used, and the category features were given a weight of 1 in equation 7 (equal to the weight of features specific to the business), resulting in a training error of 0.899 and test error of 1.077, clearly indicating that overfitting was occurring. The number of business categories and the weight given to the business category features were both reduced in order to reduce overfitting; the final results shown in Table II show that the model is likely no longer overfitting to the test data, as the testing and training error are very close.

Unfortunately, these features did not improve the model significantly. Overall test error was reduced slightly, as was test error on ratings for businesses and users with 10-20 known ratings. It is likely that the latent factors captured by the model already account for a significant portion of the rating variation that could be attributed to business category.

5) Matrix factorization with user friendship features: Similarly to matrix factorization with business features, adding features representing users' friendships did not significantly improve results of the model. This could be attributed tos several reasons. One reason is sparsity of the friendship graph; the majority of users have a very low number of friends, as shown in Section III. Another potential reason is that social connections on Yelp are not particularly meaningful and may not reflect real-world associations that would influence users' tastes; the authors of [5] similarly found that social ties themselves were not particularly useful, and more complex measurements of social connections were required to improve results meaningfully.

VI. Conclusion and Future Work

Matrix factorization methods clearly provide a highly extensible, useful method to predict user-item ratings. On this

particular dataset, matrix factorization provided a clearly superior method of predicting user ratings of businesses than a baseline model taking into account global, user, and business rating biases. Incorporating business category features into the model resulted in a marginal improvement in rating prediction, while incorporating information about user friendships did not significantly alter the model's performance.

Future work could take several directions. It is likely worthwhile to continue investigating improving the model using additional business and user features. While business categories were somewhat useful in predicting ratings, the dataset provides many additional features about businesses that could potentially be included in the model. In addition, while raw user friendship information was not useful in improving results, more nuanced analysis of social connections could still prove to be useful. One area that could be investigated is the inclusion of tie strength between users in terms of metrics such as embeddedness or other metrics that attempt to characterize how strong a particular social tie is. An additional source of features that was not evaluated in the scope of this paper is the inclusion of textual features from reviews written by users for businesses. Methods such as topic modeling could be investigated to extract features about user preferences and business characteristics from the review text itself; these features could prove to be useful in predicting ratings.

REFERENCES

[1] T. Chen, Z. Zheng, Q. Lu, W. Zhang, and Y. Yu, "Feature-Based Matrix Factorization," arXiv:1109.2271 [cs], Sep. 2011.

[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, vol. 42, no. 8, pp. 30?37, 2009.

[3] S. Funk, "Netflix Update: Try This at Home," 11-Dec-2006. [Online]. Available: http://sifter.org/~simon/journal/20061211.html. [Accessed: 11-Dec-2015].

[4] "Optimization: Stochastic Gradient Descent." Deep Learning Tutorial. [Online]. Available: http://ufldl.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent/. [Accessed: 11-Dec-2015].

[5] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender Systems with Social Regularization," in Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, New York, NY, USA, 2011, pp. 287-296.

[6] "Yelp Dataset Challenge.' 'Yelp. [Online]. Available: http://www.yelp.com/dataset_challenge. [Accessed: 11-Dec-2015].

[7] Xiao Yu, Xiang Ren, Yizhou Sun, Bradley Sturt, Urvashi Khandelwal, Quanquan Gu, Brandon Norick, Jiawei Han, "Recommendation in Heterogeneous Information Networks with Implicit User Feedback," in Proceedings of the 7th ACM conference on Recommender systems, New York, NY, USA, 2013, pp. 347-350.