Predicting Yelp User's Rating Based on Previous Reviews

Yue Li (yulelee@stanford.edu)
Haomiao Song (hmsong@stanford.edu)

Motivation

Currently, interactions between users and Yelp is mainly initiated by the users searching for some keywords, and then go through a list of ranked items. While this approach could be effective in many ways, personalized recommendation is also crucial for a better user experience. In this project, we aim to build a hybrid recommendation system to predict the ratings and recommend new places to users.

Data

The data is from Yelp Dataset Challenge[1]. We filtered out users or restaurants with less than 20 reviews associated with them. The final dataset contains 259143 reviews. Each review contains a user ID, a restaurant ID, the rating, and the original text of the review. Among those, 25% of the data is randomly picked out to be the testing set. There are 62958 reviews in the testing set, and 196185 in the training set.

Metric

We output the predicted the rating of a user to a restaurant, and use the root mean squared error as the metric:

$$RMSE = \sqrt{\frac{1}{N}\sum(r_{ub} - \hat{r}_{ub})^2}$$

Models

Various predicting models have been implemented:

(1) **Baseline**: Compute the average ratings for each user $\bar{r}_u^U$ and each restaurant $\bar{r}_b^B$, and output the average.

(2) **Model-based Collaborative Filtering**[2]: Construct the rating matrix X where X_{ub} is the rating user u has given to restaurant b. Fill in the missing values in X by the baseline estimate, then factorize X by singular value decomposition $X = USV^\top$. We pick the largest k singular values and use the product to estimate the missing values in X.

Models

(3) **Memory-based Collaborative Filtering**[3]: Compute the user similarity matrix S^U and restaurant similarity matrix S^B:

$$S_{ij}^U = \frac{(x_i^U)^\top (x_j^U)}{\|x_i^U\|_2 \|x_j^U\|_2} \qquad S_{ij}^B = \frac{(x_i^B)^\top (x_j^B)}{\|x_i^B\|_2 \|x_j^B\|_2}$$

Then make prediction by the user similarity:

$$\hat{r}_{ub} = \bar{r}_u^U + \frac{\sum_{i:\, X_{ib} \neq 0} S_{ui}^U (X_{ib} - \bar{r}_i^U)}{\sum_{i:\, X_{ib} \neq 0} |S_{ui}^U|}$$

or by the restaurant similarity:

$$\hat{r}_{ub} = \frac{\sum_{i:\, X_{ui} \neq 0} S_{bi}^B X_{ui}}{\sum_{i:\, X_{ui} \neq 0} |S_{bi}^B|}$$

(4) **Collaborative Filtering by Gradient Descent**: Factorize X into the product of two arbitrary matrices $X = PQ$, define the loss function s:

$$L = \sum_{(u,b)} (X_{ub} - P_u^\top Q_b)^2 + \lambda\,(\|P\|_2^2 + \|Q\|_2^2)$$

Take the derivative and get the update rule:

$$Q_b \coloneqq Q_b - \eta\,[(X_{ub} - P_u^\top Q_b)P_u + \lambda Q_b]$$
$$P_u^\top \coloneqq P_u^\top - \eta\,[(X_{ub} - P_u^\top Q_b)Q_b + \lambda P_u^\top]$$

(5) **Review-based Recommendation:** First concatenate all the reviews a restaurant has received, vectorize using the bag-of-words technique. Transform to tf-idf to get the feature vector for each restaurant. Also, construct the preference vector for each user, by a linear combination of the feature vectors of all the restaurants, with the ratings as the weight. Rank the restaurants based on the similarities between feature vectors of the restaurants and preference vectors of the users.

Then we use locally linear regression to predict ratings. Use the rankings as input and ratings as the labels.

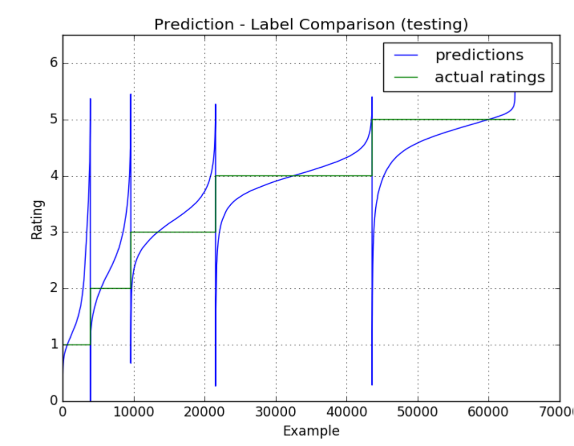
(6) **Text-based Recommendation**: Same infrastructure as the review-based recommendation, but build the preference vector for each user directly by concatenating the reviews written by him/her.

Results

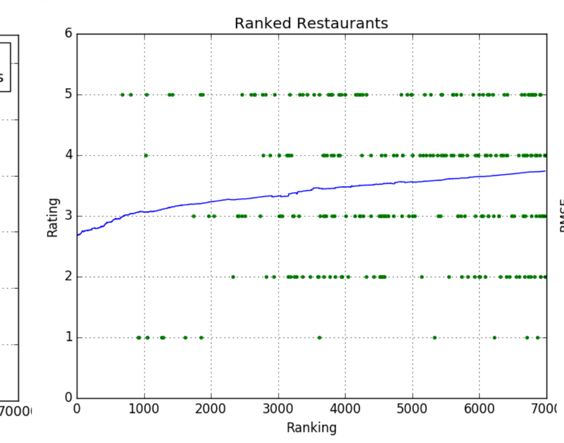
The RMSE for all of the models mentioned above:

Model	Training RMSE	Testing RMSE
Baseline	1.532	1.587
Model-based Collaborative Filtering (SVD)	0.061	1.128
Memory-based Collaborative Filtering (user)	1.099	1.107
Memory-based Collaborative Filtering (restaurant)	1.132	1.133
Gradient Descent	0.509	1.112
Review-based Recommendation	1.255	1.380
Text-based Recommendation	1.367	1.392
Hybrid (linear combination of all above)	0.598	0.618

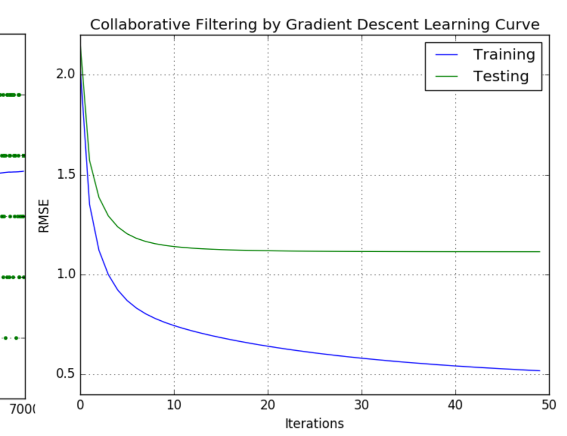
There are also a few plots we would like to show:



For the hybrid system, the actual ratings and the predictions are plotted. We can see that the predictions are centered around the actual ratings.



Regression step of the text-based recommendation is plotted. The x-axis is the rankings, y-axis is the actual ratings.



Learning curve for the SGD. We use the L2 regularization to prevent over-fitting.

Discussion and Future Work

Among the others, the text-based recommendation performs slightly worse. However, the interesting part about this model is that it doesn't require any previous rating of the user. All it need is some text associated with the user. For example, now that the users can use their Facebook account to sign up for Yelp, the preference vectors can be constructed using the text from their Facebook status. We hope that this might, to some extent, relieve the cold start problem. For the future, we would like to continue working on the hybrid system. If each individual model can capture one particular aspect of the data, it would be much more advantageous to combine them in order to produce predictions with higher accuracy.

[1] https://www.yelp.com/dataset_challenge
[2] B.M. Sarwar, G. Karypis, J.A. Konstan, and J.Reidl. Application of dimensionality reduction in recommender system - a case study. In ACM WebKDD 2000 Web Mining for E-Commerce Workshop, 2000.
[3] Breese, John S., David Heckerman, and Carl Kadie. "Empirical analysis of predictive algorithms for collaborative filtering." Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 1998.