EECS 349 Machine Learning
Northwestern University
Predicting Yelp Restaurant Ratings

Robert Luo

June 13, 2018

Author

Robert Luo is an Applied Mathematics undergraduate at Northwestern University (graduation Spring 2018). Please feel free to send comments or feedback to robertluo@u.northwestern.edu.

Abstract

Yelp is an online platform where users can submit reviews about restaurants and businesses. A business' Yelp rating and its written reviews can attract or repel visitors, and consequently determine the success of the business. Many consumers depend on Yelp as well. In today's day and age, access to crowd-sourced reviews can highly influence consumer decisions on what stores to shop at and restaurants to dine in. A model that can predict when a Yelp reviewer will rate a restaurant highly (4 or 5 stars) can serve as a valuable recommender system, increasing user engagement and retention as well as customer traffic to new restaurants.

This project evaluates the performance of User-Item Collaborative Filtering, Item-Item Collaborative Filtering, and K-Nearest Neighbor Collaborative Filtering in predicting Yelp user ratings of restaurants in the United States.

I used data from the Yelp dataset challenge, (which can be found at www.yelp.com/dataset/challenge) and selected datasets consisting of user reviews and business features. I created a matrix of user-restaurant-ratings to use with User-Based and Item-Based CF, which I also used for K-NN CF in conjunction with a one-hot matrix of restaurants and their attributes. Due to the size of the Yelp dataset, I chose to only consider users and restaurants in the US with at least 15 reviews in order to avoid the cold-start problem related to recommender systems (this remains unaddressed throughout the project). Some of the features considered in the one-hot matrix used for K-NN CF include: cuisine type, restaurant attire, ambience, alcohol availability, and price range. Categorical variables were converted to one-hot attributes, for a total of 170 attributes (a comprehensive list of attributes can be found in Appendix A). In total, my processed dataset considered 5186 restaurants and 10634 users. There were 248838 ratings used for training, 31170 used for validation, and 10390 held out for testing (roughly an 85/10/5 split). I used scikit-learn to assist with my project, along with some implementation of my own.

Methods

Both User-Based CF and Item-Based CF fall under Memory-Based CF methods of predicting user ratings of restaurants. They essentially use a user-v-restaurant matrix of ratings to compute a user-user similarity metric and a restaurant-restaurant similarity metric, respectively. I use cosine distance in this report. Then, a rating is predicted from a weighting of other users' ratings of the item or the specific user's ratings of previously rated items.

K-NN CF falls under a Model-Based approach to predict user ratings. To implement K-NN CF, I use a one-hot matrix of restaurants and 170 attributes. This matrix is used to find a user's K most similar previously-rated restaurants based on cosine distance. Ratings are then predicted by taking a weighted sum of these ratings. A predicted rating is prescribed "high" if it is 4 stars or higher, and thus should be recommended to a user.

User-Item Collaborative Filtering

User-Item CF can be explained as "users similar to you liked...". Similarity between users a and b is computed via cosine distance:

$$similarity(u_a, u_b) = \frac{u_a \cdot u_b}{\|u_a\|_2 \|u_b\|_2} \tag{1}$$

Then, in order to create a prediction for a user's rating n, use the following formula:

$$\hat{x}_{a,n} = \bar{x}_a + \frac{\sum_{u_a} similarity(u_a, u_b)(x_{b,n} - \bar{x}_{u_b})}{\sum_{u_a} |similarity(u_a, u_b)|} \tag{2}$$

In equation 2 the $\bar{x}_{u_b}$ in the numerator adjusts for the difference in two users' average rating, since some users may tend to give higher or lower scores to their restaurants ratings, on average. The denominator normalizes the scores, and the $\bar{x}_a$ takes into account user $a's$ average rating to determine the prediction.

Item-Item Collaborative Filtering

Item-Item CF can be explained as "users who liked this restaurant also liked...". The steps to implement this method are identical to those in the User-Item CF process, except the user-restaurant-ratings input matrix is transposed so as to calculate similarities between restaurants rather than users.

K-NN Collaborative Filtering

K-NN CF can be explained as "you might like this item based on your ratings of K similar items...". The method I use to predict user $a's$ rating of a specific restaurant:

1. Use the user-restaurant ratings matrix to determine the restaurants that user a has previously rated
2. Among the user's previously-rated restaurants, use cosine distance to find the K most similar
3. Take a uniformly-weighted vote among the K restaurants to classify determine the restaurant's rating
4. If the predicted rating is at least 4 out of 5 stars, determine the rating "high" and recommend the restaurant to the user

In order to determine the optimal K value, I evaluate the K-NN model with a range of K's against the validation set and calculate precision, recall, and accuracy.

Results

User-Item & Item-Item Collaborative Filtering

Table 1 below shows errors for both User and Item-Based CF methods.

Evaluation Measure	User-Item	Item-Item
RMSE	3.907	3.921
MAE	3.731	3.746

Table 1: Root Mean Squared Error and Mean Absolute Error of User and Item Based CF

Although these results are not stellar, it is promising that we can predict user ratings off of similarity measures.

K-NN Collaborative Filtering: Determining K

Figure 1 shows the K-NN performance evaluated against the validation set with a range of K values. Although the accuracy of prediction is not stellar, we can evaluate two other metrics. Precision is the ability of the classifier not to incorrectly label an observation positive that is actually negative. It is important to have a high precision score in order to not recommend a would-be-low-rated restaurant to a user. Recall is the ability of the classifier to find all the positive samples. A recommender with a high recall score is able to recommend users most of the restaurants he or she would rate highly.

It can be observed in figure 1 that the K-NN model achieves high recall and precision as K is increased, suggesting that creating a well-functioning recommender with K-NN CF is indeed feasible. Furthermore, all three metrics plateau at K = 20, which is chosen as the optimal K for the K-NN CF model.

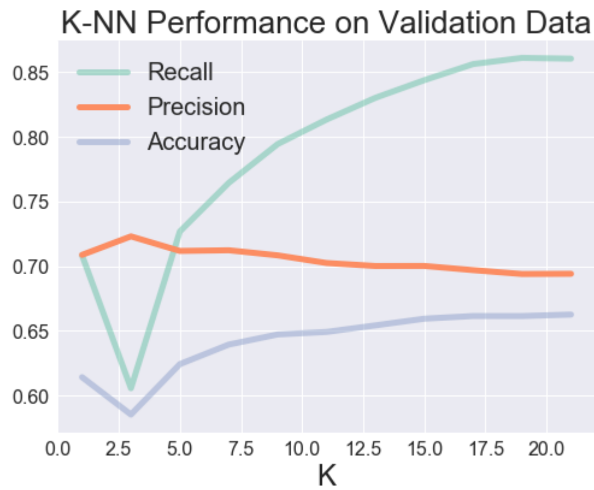

Figure 1: K-NN recall, precision, and accuracy on validation data

K-NN Collaborative Filtering: Testing Optimal K

After selecting K = 20 from K-NN CF results on validation data, I ran 20-NN CF on the held-out test data set and produced the results shown in table 2. These results are consistent with those obtained by running K-NN CF on validation data. High recall suggests that a recommender created with 20-NN CF would be able to recommend users most of their to-be-highly-rated restaurants, and could potentially be a step towards a valuable recommendation tool for Yelp.

Evaluation Measure	20-NN CF Results
Recall	0.870
Precision	0.688
Accuracy	0.663

Table 2: 20-NN CF Recall, Precision, and Accuracy

Further Work

In order to improve User-Item and Item-Item CF results, I would suggest looking at Top-K Collaborative Filtering, where only the most similar K ratings are taken into account. This is different from the K-NN CF method discussed in this report, as Top-K still works in the user-restaurant-rating space. Furthermore, it could be interesting to experiment with different similarity metrics such as Pearson correlation. To better improve K-NN CF, I would suggest experimenting with various distance metrics and methods of weighting the K-selected neighbors. Furthermore, as each user-rating vector and restaurant-feature vector has many attributes, it would be interesting to use Neural Net methods to predict ratings.

A large limitation of using the methods in this project to create a recommender system is the cold-start problem. In context, this arises when users or restaurants have few reviews. In order to implement the methods discussed in this report, I included restaurants rated at least 15 times and users who provided at least 15 ratings. Addressing this problem is crucial to implement a recommender system that will accommodate new users.

Appendix A: One-hot Attributes

The attributes which were initially boolean are listed in figure 2 below.

Boolean:	HasTV
	RestaurantsGoodForGroups
	GoodForKids
	RestaurantsReservations
	RestaurantsTableService

Figure 2: Boolean attributes

The attributes which were initially categorical which were transformed into one-hot representation are listed in figure 3 below.

Categorical:	Best Meals:	latenight	Cuisine:	Italian	Japanese	Malaysian	Lebanese
		breakfast		Seafood	Asian Fusion	Szechuan	British
		dessert		Fast Food	Vietnamese	Steakhouses	Laotian
		lunch		Burgers	Breakfast & Brunch	Barbeque	Irish
		brunch		Gluten-Free	American (New)	Filipino	Cambodian
		dinner		Indian	Chicken Wings	Kosher	Indonesian
	Ambience:	trendy		American (Traditional)	Noodles	Creperies	Kebab
		casual		Buffets	Vegetarian	Middle Eastern	Conveyor Belt Sushi
		touristy		Sushi Bars	Thai	Turkish	Brasseries
		intimate		Chinese	Diners	Cafes	Live/Raw Food
		romantic		Pizza	Korean	Brazilian	Izakaya
		classy		Salad	Southern	Taiwanese	Honduran
		upscale		Greek	Latin American	Hot Pot	Burmese
		hipster		Mediterranean	Chicken Shop	Fondue	Shanghainese
		divey		Tex-Mex	Delis	Caribbean	Japanese Curry
	Noise Level:	averageNoise		Mexican	Tacos	Puerto Rican	Bangladeshi
		loud		Sandwiches	Hawaiian	Ramen	Supper Clubs
		quiet		Hot Dogs	Soup	Cajun/Creole	Ukrainian
		very_loud		Venezuelan	Modern European	Hungarian	Afghan
	Alcohol:	full_bar		Food Stands	Falafel	New Mexican Cuisine	Dominican
		NoAlcohol		Tapas Bars	Basque	Sri Lankan	Polish
		beer_and_wine		Spanish	Haitian	Pop-Up Restaurants	Singaporean
	Attire:	casualAttire		Dim Sum	Cafeteria	Poutineries	Bulgarian
		dressy		Trinidadian	Belgian	Ethiopian	Austrian
		formal		Argentine	Gastropubs	Cantonese	Nicaraguan
	Price:	1Price		Cheesesteaks	Pan Asian	Salvadoran	Fish & Chips
		2Price		Persian/Iranian	Tapas/Small Plates	Arabian	Soul Food
		3Price		French	Wraps	Moroccan	Portuguese
		4Price		African	German	Russian	Mongolian
		5Price		Peruvian	Comfort Food	Uzbek	Vegan
	Overall Rating:	lowStars		Teppanyaki	Pakistani	Scandinavian	Colombian
		medStars		Halal	Food Court	Armenian	Cuban
		highStars		Waffles	Himalayan/Nepalese	Egyptian	

Figure 3: Categorical attributes - turned one-hot