Collaborative Filtering on Sparse Rating Data for Yelp.com

Jason Fennell
jfennell@yelp.com

December 10, 2009

Abstract

We examine the problem of building a recommendation engine for Yelp.com, particularly the problem of extremely sparse rating data. We show that while click data does not directly model rating data well, it can be used to improve and extend the reach of neighborhood based rating interpolation methods.

1 Introduction

Yelp.com is a website that aggregates user reviews of local businesses ranging from restaurants to flower shops. In order to build a community and grow as a business, Yelp must ensure that users are exposed the most relevant businesses for their needs. This is primarily accomplished through active user actions like searching on the site. However, as Netflix, Amazon and other sites have shown, this misses the powerful avenue of passive suggestions proposed to the user.

The focus of this paper is to create a recommendation engine for Yelp based on the extensive public research that has been done as part of the Netflix Challenge. At first glance, Yelp seems nearly identical to Netflix in the applicability of a recommendation engine. Both Yelp and Netflix revolve around users rating entities; local businesses for Yelp, movies for Netflix. However, there is a fundamental difference in approach that changes the framing of this problem. The whole of Netflix.com pushes users toward casually rating a large number of movies. Yelp, on the other hand, is focused on gathering high quality *review content*, and thus strongly discourages rating a business without a well-thought review. This higher-per-rating-cost means the Yelp rating dataset is far more sparse than that of Netflix. Even restricting to users and businesses with a significant number of reviews, only .01% of possible ratings are known (this number is 1% for the Netflix dataset). Dealing with this sparsity issue is the focus of this paper.

We take a three-step approach to the problem of sparsity. First, we build a Netflix-style recommendation engine for the subset of "active" users/businesses: those that have at least 20 ratings. Next we investigate the direction relationship between ratings and clicks. Finally, we build a recommendation engine that leverages click data to improve its recommendation accuracy over the pure-rating model. This click-based model will also open the door to making Yelp's recommendation engine much more general, giving personalized ratings to even to users

who have few or no ratings.

2 Related Work

There are essentially two approaches to the problem of predicting unknown ratings. The first family of approachs, called *neighborhood methods*, predict an unknown rating for a user as a weighted average of the known ratings of similar users. The other family of approachs, called *laten factor methods*, predict unknown ratings as a product of auto-detected latent features generally produced by some sort of rank-reducing matrix decomposition. We focus on the neighborhood methods in this paper as they are easier to implement and, as we will see later, can have their dependence on rating data weakened. For more information on collaborative filtering and the Netflix callenge, see [2] which lists the papers of the team that won the Netflix Challenge.

3 Experiments

For all of our experiments we restrict ourselves to users and business on Yelp that have at least 20 reviews. We train on 70% of our data and test on the remaining held-out 30%. Each user u is represented as a vector that maps from business b to the rating/num clicks u had for b. Businesses are defined similarly. We measure performance with root mean squared error.

3.1 Baseline Model

To establish a performance baseline we build several naïve heuristic models. Our first model simply predicts the global average rating for every unknown rating. Next we predict the average user rating for each user, then the average business rating for each business, giving a substantial performance increase over global average by accounting for user or business bias. We see from the results in Table 1 that business averages have more variability and thus explanatory power than user averages.

For all of our further models in this paper we first subtract the user average from each user rating, then subtract the business average from each business rating. We can then work with the residuals without worrying about user/business biases. Our final heuristic model, which serves as a reasonable baseline to compare later models against, predicts the sum of the user and business averages. We can see in Table 1 that this "double centering" of results produces a substantial (11%) improvement over the global average model. These results are are somewhat higher than comparable number in the Neflix challenge, which is expected given Yelp's sparser data. As we previously mentioned, even restricting to users and businesses with a substantial number of reviews, Yelp has data on less than .01% of user, business pairs whereas Netflix has data on about 1% of user, movie pairs.

Method Name	Test RMSE
Global avg	1.1058
User's avg	1.0834
Biz's avg	.9976
User & biz avg	.9844

Table 1: Baseline Models

3.2 Rating-based Model

Now that we have a baseline model and are working with double-centered residuals we consider

predicting ratings as a weighted sum of ratings by similar users on a given business. That is, the predicted rating of user u on business b is

$$\hat{r}_{u,b} = \frac{\sum_{u' \in N(u,b)} w_{u',u} r_{u',b}}{\sum_{u' \in N(u,b)} w_{u',u}},$$

where $\hat{r}$ is the estimated rating, r are known ratings, w are similarity scores between pairs of users, and $N(u, b)$ is the set of the k users most similar to u. Pearson correlation coefficients are used to measure similarities. We experimented with shrinking correlations toward zero based on their support, as outlined in [1], but we found that it actually hurt our performance.

Note that the above explanation takes a user-centric approach, where ratings are predicted as a weighted sum of the ratings from a neighborhood of users. It is equally valid and defined similarly to take a business-centric approach where ratings are a weighted sum of ratings from a neighborhood of businesses. We try both approaches.

We list our results for rating-based neighborhood models in Table 2 for both user and business centric approaches and a range of neighborhood sizes, from which we can make several observations. First, the business-centric approach outperforms the user-centric approach for all by the trivial size neighborhood. We think that this is due to there being greater variation in the set of businesses rated by a user than there is in the set of users that rate a business. Users that like Thai restaurants will have rated a bunch of Thai restaurants, which will make business-centric predictions of Thai restaurants more reliable because it is easy to find similar businesses rated by a given user. However, the exact set of Thai restaurants rated will vary a lot even among other users that like Thai. Thus, even if similar users are identified it is harder to find ones that have actually rated the business in question.

The other main observation from this data is that it does not outperform the baseline model by very much at all. In fact it takes a neighborhood of size 50 for the user-centric approach to outperform the baseline by .0001. This seems to be a data-sparsity problem. Ratings are sparse enough that finding any similar users in the first place is a challenge, let alone similar users that rated a particular business. However, we do see decreasing RMSE with increasing neighborhood size, and the business-centric approach does beat out the baseline by for neighborhoods of $k > 10$, so neighborhood methods seem to have at least some power.

k-NN	User	Biz
1	.9936	.9942
5	.9860	.9857
10	.9851	.9846
20	.9846	.9840
35	.9844	.9838
50	.9843	.9837

Table 2: Rating-based model RMSE by neighborhood size

3.3 Regression on Clicks

We will now begin to examine the application of click data to the recommendation problem. Yelp encouraging casual rating of businesses has the potential to de-emphasize quality reviews, which are the foundational to Yelp's value as a website. Clicks, on the other hand, are a source of revenue. Yelp thus has an incentive to gather as many clicks as possible, so while they are noisier than ratings we may be able to leverage clicks to

address data sparsity problems.

With this in mind we have investigated the relationship between the clicks and ratings of the set of users that we used for previous recommendation models. In particular we looked at clicks from search results to individual businesses. We found that about 30% of a user's clicks were on businesses that they had rated from which we conclude two things. First, clicks and ratings have a substantial enough overlap that clicks can be useful for the prediction of ratings. Second, adding clicks to our model will bring in a substantial amount of extra information.

Given that there seems to be a relationship between ratings and clicks we tried to estimate ratings from clicks by using least squares to train a linear model that predicts rating with the number of times a user clicked on a business. The resulting set of models had terrible predictive power, with an average R^2 value of .062. Looking into the individual models it appears that linear term of the model tends to have a small magnitude (.22 average absolute value) while the constant term of the model tended to the average rating of the user. Essentially the regression tried to minimize the effect of number of the bad explanatory variable. While it is not particularly surprising that this model performed poorly, this negative result does suggest that approaches that directly model ratings from clicks will be less than fruitful.

3.4 Click-based Model

Based on the failure of a direct rating-from-clicks model, we decided to use clicks in a more indirect manner to help with the computation of similar neighbors. Note that in the original rating-based model sparsity of ratings hurts us in two ways. First, there are just are not many ratings of a given user or business to interpolate between. Second, the similarity scores and interpolation weights are less well-determined with less data. We showed in the previous section that clicks cannot really help us address the first of these problems. However, what we can do is use clicks instead of ratings to calculate similarity scores. This will allow for a better estimation of a neighborhood of similar users or businesses and their relative importances.

For the click-based approach we normalize away the absolute number of clicks and only use the presence vs. absence of clicks in the user or business vectors. This is because, for instance, the similarity scores of users that have 12 and 8 clicks on a business, should not really be any different than the similarity scores of users that have 5 and 15 clicks on a business.

We show our results for the click based model in Table 3. The most important thing to notice about this table is that the user-centric model is now beating out the baseline by a substantial, if not large, margin. This shows the validity of clicks as a similarity metric, which means that this feature can be extended to users who have not rated any Yelp businesses at all. Given only click data we will be able to compute, based on highly-active users with similar click profiles, businesses that a user would not find otherwise.

A more puzzling trend in the click-based data is that the business-centric approach not only does worse than the user-centric approach, but actually does worse than the rating-based business-centric approach. It seems most likely that the relative cost of making a click versus making a rating will reverse the logic we outlined in the rating-based model section. It is easy to make clicks so users that like Thai places will probably have clicked on many Thai places and provide a better basis for finding similar users.

Businesses, however, are subjected to a great deal more clicks from a variety of users that we cannot now discriminate between positive and negative interest.

k-NN	User	Biz
10	.9737	.9848
20	.9734	.9845
35	.9734	.9843
50	.9733	.9843

Table 3: Click based model RMSE by neighborhood size

4 Conclusion

In this paper we examined building a recommendation engine on extremely sparse data, particularly the application of click data to the problem. We showed that clicks do not model ratings well directly, but can be used to effectively boost the performance of some neighborhood methods. This result is particularly exciting is that it not only increases the performance of the recommendations, but it also allows them to be made to a broader range of users.

References

[1] Robert M. Bell and Yehuda Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 43–52, Washington, DC, USA, 2007. IEEE Computer Society.

[2] Volinsky Bell, Koren. http://www2.research.att.com/~volinsky/netflix/.