Slope One on the Sparse Yelp Review Dataset

Maxwell Omdal

Fall 2019

1 Slope One on the Yelp Challenge Data Set

Slope one is an algorithm proposed by Daniel Lemire and Anna Maclachlan in 2009 as an item-based collaborative filtering algorithm [2]. The algorithm is often used instead of more complex recommendation algorithms for its simplicity. In this paper, I will outline the implementation of the standard Slope One algorithm, Weighted Slope one, and a modified version of standard Slope One that attempts to utilize more data provided by our yelp dataset.

For this problem, we will be predicting user ratings from a data set of Yelp reviews. The dataset, provided by Yelp is provided as a collection of JSON files defining reviews for businesses by ID, user info by ID, and business info by ID [1]. First, the data will have to be organized in a table with users as the rows and businesses as the columns. Because the data set has nearly 7 million reviews, and over 100,000 businesses across the US, we will reduce this to businesses in a single state: Arizona. About one third of this dataset is reviews of businesses in Arizona. Other states represented in the dataset include North Carolina, Pennsylvania, Ohio, Illinois, and Nevada. To further handle the size of the data set, our data restructuring tool will run multiple processes in parallel, and our user-rating matrix will be stored as a sparse column matrix. In order to build the user-business table structure, we will define 3 files: a sparse matrix file hosting all the ratings, a hash-map where each key is a user ID and the value is the row associated with that user in our sparse matrix, and a hash-map where each key is a business ID and the value is the column associated with that business in our sparse matrix. Then, we can lookup user's ratings for businesses by their IDs. We can also then predict user's ratings for businesses by their IDs.

2 Standard Slope One

The standard Slope One algorithm predicts that a user rates everything as the average of all its ratings plus the user's average deviation from the mean for that item in question:

$$P^{S1}(u)_j = \bar{u} + \frac{1}{card(S_i(\chi))} \sum_{i \in R_j(\chi)} dev_{j,i} \tag{1}$$

where χ is all users in the set and $dev_{j,i} = \sum_{u \in S_{j,i}(\chi)} \frac{u_j - u_i}{card(S_{j,i}(\chi))}$.

And $\bar{u}$ is the average rating for user u, $card(S)$ is the total number of items in the Set s, χ is the set of all users, R_j is all ratings for a user excluding j, where j is the business we'd like to predict a rating for, S is the sample set. Assuming we don't try to predict a rating for a business that is already rated for the user, $R_j = S(u)$.

3 Weighted Slope One

Weighted slope one is a slightly more sophisticated version of slope one that takes into account the number of ratings observed. In analysis, I will explain why this is not the optimal choice for our data set.

$$P^{wS1}(u)_j = \frac{\sum_{i \in S(u)-j} (dev_{j,i} + u_i) c_{j,i}}{\sum_{i \in S(u)-j} c_{j,i}} \tag{2}$$

where $c_{j,i} = card(S_{j,i}(\chi)$

On the occasion that $\sum_{i \in S(u)-j} c_{j,i} = 0$, we will default to predicting $\bar{u}$, the average rating for the user, u that we would like to predict a business rating for.

4 Modified Slope One

Associated with each review in our dataset is a rating of its 'usefulness' as voted on by other users. Assuming users up-vote posts that provide highly accurate descriptions of a business, we would assume these ratings to be more accurate than ratings with low or no 'usefulness' votes. The following modified slope one algorithm takes into account the weight $w_{u,i}$ as the usefulness score for the rating of business i by user u.

$$P^{mS1}(u)_j = \bar{u} + \frac{1}{card(S_i(\chi)) \times \sum_{w_{u,i} \in U(u)} w_{u,i}} \sum_{i \in R_j(\chi)} dev_{j,i} \tag{3}$$

where $dev_{j,i} = \sum_{u \in S_{j,i}(\chi)} \frac{u_j - u_i}{card(S_{j,i}(\chi)} \times w_{u,i}$

5 The Data set

The Yelp data set was sparse, with 99.993% of the user-business table made up of unrated values. Users reviewed on average 8.72 businesses. There are more than 50,000 businesses and 506,000 users in the set. Figures 1 and 2 provide some more data on user's rating patterns.

6 Results and Analysis

The results for a MSE of 1000 random samples can be seen in fig. 3.

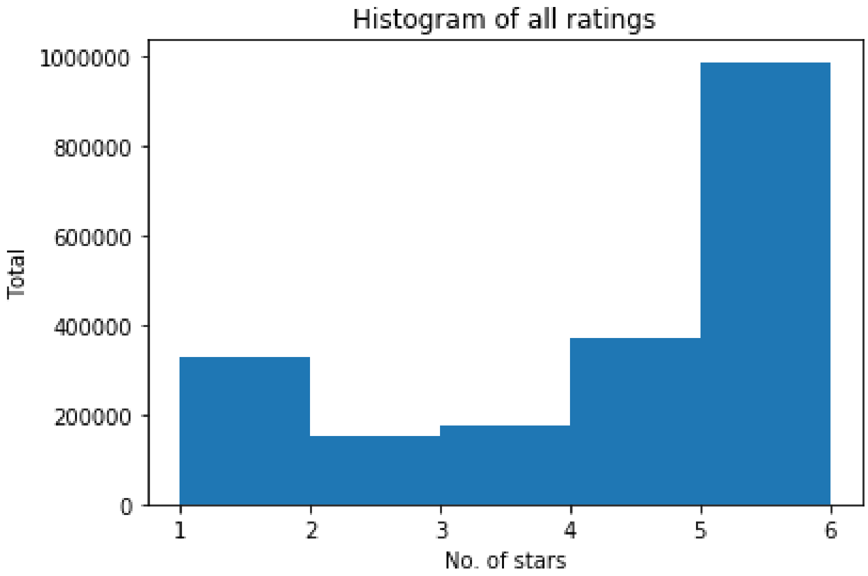

Figure 1: A histogram of all ratings. Note that the most popular rating is 5 stars, followed by 4 stars. This makes sense, considering the overall average rating is 3.763 stars.

For control, I compared all actual values to the median possible rating of 2. Slope one is also compared to *PER USER AVERAGE*, which always predicts the average of a user's ratings. Mean Squared Error is used as the error value, and is calculated by predicting items that already have ratings for a user. We can use a single data set for testing and training because Slope one does not learn specific data points. The algorithm we implemented is online, and nothing is pre-computed.

It should be clear that Slope One is doing something intelligent. It's performance improves upon the control and PER USER AVERAGE, if even only slightly. Weighted Slope One, however is worse than both PER USER AVERAGE and Standard Slope One. This most likely has to do with the sparsity of our model. When running our algorithm, we find users who have reviewed both businesses i and j, where j is the business we would like to predict a rating for, i is in the set of businesses the user, u has already rated. Since our data is so sparse, the likelihood we find more than a couple users who have reviewed both business j and any other business i that u has rated is very low.

Unfortunately, no improvements were found for the Modified Slope One algorithm. Its results were similar to that of Slope One and *PER USER AVERAGE*. In the section on future work, I will describe how other data from each review could be used to bolster our ratings data.

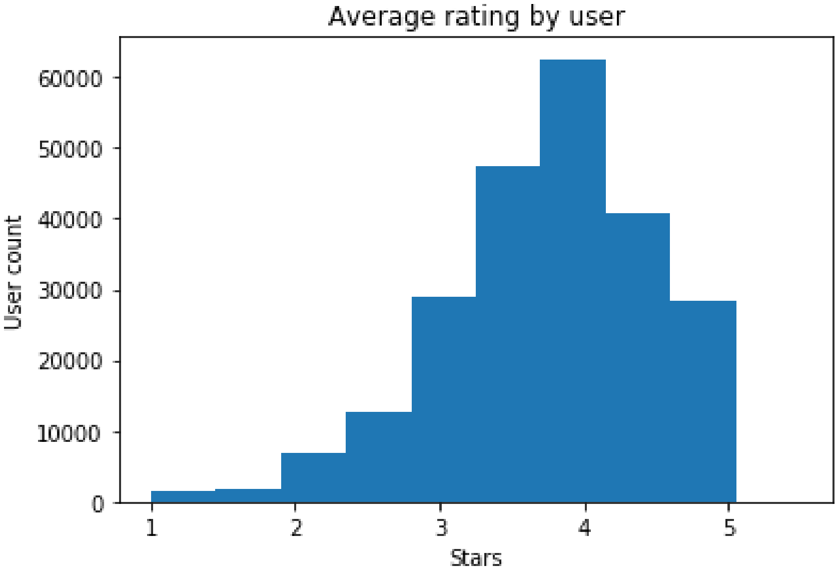

Figure 2: A histogram of the average ratings by user. The overall average is 3.763 stars. It shows people generally have positive sentiment about the businesses they rate.

Algorithm	MSE
Control	5.45
Per User Average	2.7461
Slope One	2.5824
Weighted Slope One	4.3628
Modified Slope One	2.6364

Figure 3: Mean Squared Error Results

7 Future Work

This was an introduction to the slope one algorithm for predicting ratings for users. Although the results were not a big improvement over PER USER AVERAGE, this is a guide for future investigation. In Lemire and Machlaclan's paper, a third Slope One algorithm, Bi-Polar Slope One would be the next logical algorithm to test. Bi-Polar essentially runs Weighted Slope One twice, once considering "positive" ratings, and once considering "negative" ratings. In our case, we could either use the middle of our range, 3 as the split between positive and negative ratings. Alternatively, we could consider the average rating, 3.7 as the middle point.

A follow-up experiment would be to incorporate more data in our prediction algorithm. To overcome data sparsity, logically we would want to incorporate as much useful data as possible. Another experiment would be to run TF-IDF, or a similar keyword identifying algorithm on user reviews, and define user-similarity weights by the keywords found in their reviews [4, 3]. Users who share many keywords in their reviews likely have similar interests, and thus would be more likely to rate businesses in similar ways. For example, two users who regularly review coffee shops and use keywords like *quiet, study-spot* or *latte* would likely have a strong correlation in what they look for in a coffee shop: a quiet place to study with good lattes.

References

[1] Y. Inc. Yelp dataset, 2019. https://www.yelp.com/dataset/challenge.

[2] D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 471–475. SIAM, 2005.

[3] J. Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. Piscataway, NJ, 2003.

[4] M. Sun, H. Zhang, S. Song, and K. Wu. Uso-a new slope one algorithm based on modified user similarity. In *2012 International Conference on Information Management, Innovation Management and Industrial Engineering*, volume 2, pages 335–340. IEEE, 2012.