SMU Data Science Review

Volume 1 | Number 4 Article 1

2018

A Comparative Evaluation of Recommender Systems for Hotel Reviews

Ryan Khaleghi
Southern Methodist University, rkhaleghi@smu.edu

Kevin Cannon
Southern Methodist University, cannonk@smu.edu

Raghuram Srinivas
Southern Methodist University, rsrinivas@mail.smu.edu

A Comparative Evaluation of Recommender Systems for Hotel Reviews

Ryan Khaleghi, KevinCannon, and Raghuram Srinivas

Southern Methodist University, Dallas, TX 75275 USA

Abstract. There has been increasing growth in deployment of recommender systems across Internet sites, with various models being used. These systems have been particularly valuable for review sites, as they seek to add value to the user experience to gain market share and to create new revenue streams through deals. Hotels are a prime target for this effort, as there is a large number for most destinations and a lot of differentiation between them. In this paper, we present an evaluation of two of the most popular methods for hotel review recommender systems: collaborative filtering and matrix factorization. The accuracy of these systems has become a focus, as more accurate recommendations can lead to increases in profits through various means. Also, given the rapid growth of big data, processing speed to calculate recommendations is an important issue. Using hotel reviews from the TripAdvisor website, we measure the speed and accuracy of these two recommender system methods to determine which method is superior, or the trade-offs between them. The result of the evaluation is a 10.58 times difference in speed of the collaborative filtering method over the matrix factorization recommender system method, but with significantly better accuracy with the matrix factorization method.

1 Introduction

Review websites have become an integral part of the online marketplace, often heavily influencing consumer behavior. If a consumer is traveling to a new or unfamiliar location, or even looking for a new type of experience, the internet is usually the first stop for information. Whether a consumer is looking for a restaurant, barber, or hotel, customer reviews can sway a potential customer's opinion before they even step on the premises. Websites like TripAdvisor, where users can leave reviews for hotels, restaurants, and other tourist activities, are commonplace for decision-making and contain ratings on various aspects of a business.

The landscape and conversation around business ratings and reputation management has changed with the advent of technology. Consumer focus has shifted from conventional rating systems to personalized ratings based on individual users and their experiences, thanks to widespread access to the internet. Conventional ratings that were once in the hands of a select few have now given way to consumer ratings, giving customers a say in driving future business success

and reputation. This paradigm shift is reflected in statistics provided by TripAdvisor, where their site says that 60% of travelers worldwide use TripAdvisor for travel planning, and 94% of users use a business with a rating of 4 or higher.

This change and accompanying explosion of data has resulted in continuing efforts to develop methods to process this data and monetize it. Processing the text reviews into something machine interpretable, for example, has led to development of text evaluation algorithms for sentiment analysis and natural language processing. Monetizing the data has resulted in a feedback loop where sites seek to engage users to provide content, such as ratings and reviews, but also use it to suggest hotels or restaurants based on the data in those ratings and reviews – all while selling ads and selling favorable placement in the display of the suggestions. This has led to the development of recommender systems and the algorithm and methods that allow them to function.

In this continuing development of recommender systems, numerous methods have been developed and researched, including regression, matrix factorization, and collaborative filtering, among others. The challenge for any business or site is then deciding what recommender system to implement. Obviously, accurate recommendations are important to users, as poor suggestions will lead to users going to another site. Speed is also a factor, as slowing down webpage loads while the machine learning algorithm works is not an acceptable solution in today's internet. It is difficult to determine what to use in order to maximize accuracy of predictions for users while managing the processing speed and complexity that goes with an ever-growing amount of data.

Hotels are a prime candidate for this analysis. There are a wide variety of hotels, from bed and breakfasts to multinational chains, with a lot of segmentation in the market. There is also a large amount of data, even compared to restaurants, which are similarly reviewed. A hotel experience lasts much longer than a dinner at a restaurant, and there are more facets of the experience to review. Rather than just wait service and food quality, a hotel has a check in experience, location, room quality, cleanliness, amenities, etc. Given this large amount of feature-rich data available for hotels, hotel data is a prime candidate for our examination. We selected a publically available dataset of hotel reviews from the TripAdvisor website to use for our analysis.

In this paper, we will examine recommender systems using the Turi Create package, which includes multiple algorithms for said systems. In 2016, Apple bought GraphLab Inc., a software company, and their machine learning software package GraphLab Create, and later released an updated version of GraphLab Create as Turi Create. We will test the two most popular recommender system methods to determine which is superior, in terms of accuracy and speed, for hotel recommendations, or what the trade-offs are between the two for each model.

2 Recommender Systems

Recommender systems are systems that seek to actively suggest items to a user by predicting a user's rating or preference for different things and displaying

those that the user prefers. A commonly seen example is the "Users who bought this also bought…" feature on Amazon.com. They are frequently used on websites for suggesting items to purchase or services to use, including hotels. Recommender systems can be effectively used to increase user satisfaction and sales by personalizing what is essentially advertising.

Many recommender system algorithms exist, from clustering methods, Bayesian classifiers, regression based methods, and more. Lee et al. [1] explored multiple algorithms, and found that performance differs substantially based on parameters, such as number of users, number of items, and sparsity of the data. They found that a method similar to Turi Create's matrix factorization method, using single value decomposition, was generally the most accurate, with the exception of very sparse datasets. At the time of their publishing, algorithms like Turi Create's collaborative filtering recommender were not in use. As the TripAdvisor data set is very sparse, we sought to compare this now-popular algorithm against the previous best-in-class.

Collaborative Filtering is a recommender system method that makes predictions for a user based on ratings and data from other users. The basic idea is that two users who share the same opinions on some items will be more likely to share opinions on other items. This can be extended to unknown items and predict a user's preference for items that they have not evaluated. As a user adds reviews or other opinions online, this data forms the basis for a profile that will be compared to others users' profiles. The more data a user adds through reviews, the better the predictive capability will be, as the profile can be more specifically defined and compared to those of other users.

A key point in collaborative filtering in Turi Create is that the recommendations are solely based on the user's ratings. They do not include any "side data," meaning any personal classifying information such as user age, location, etc. In the case of the hotel review dataset we are using, the only features that matter for collaborative filtering are then the users' ratings of hotels.

The item similarity recommender model in Turi Create computes the similarity between two hotels using the reviews of users who have reviewed both hotels. The similarity between two hotels i and j, $S(i,j)$, is calculated using the Cosine similarity we used as

$$S(i,j) = \frac{\sum_{u \in U_i j} r_{ui} r_{uj}}{\sqrt{\sum_{u \in U_i} r_{ui}^2} \sqrt{\sum_{u \in U_j} r_{uj}^2}} \tag{1}$$

where U_i is the set of users who rated item i, U_j is the set of users who rated item j, and $U_i j$ is the set of users who rated both item i and item j. Predictions are then made, rating an item j for a user u using a weighted average of the user's ratings I_u with the equation

$$y_{uj} = \frac{\sum_{i \in I_u} S(i,j) r_{ui}}{\sum_{i \in I_u} S(i,j)}. \tag{2}$$

One of the drawbacks of collaborative filtering is the "cold start problem." This is where, due to an inability to build an accurate user profile due to limited user data (such as few ratings), the recommendations may be poor. Until a user has rated many hotels, as in the case of many users in this dataset, the predictive capability may be weak. Research by Moshfeghi et al. [2] identify this as the most important problem with the collaborative filtering method.

Matrix factorization is a recommender system method that makes predictions for a user based on combinations of users and items. The model learns latent factors for each user/item, and uses them make predictions. Unlike collaborative filtering, this model uses side data as part of its learning. As users review more hotels, as in our dataset, this changes the user/item combinations and latent factors that will be used in the model. Users and items are represented by weights and factors, where the weights account for a user or item's bias towards lower or higher ratings, and the factors model the interactions between users and items. For example, a hotel that is rated highly across many reviews would have a higher weight, or a user who likes beach hotels and dislikes urban hotels would have that reflected in the factors.

The Turi Create Matrix Factorization Recommender predicts a score as

$$score(i,j) = \mu + \omega_i + \omega_j + a^T x_i + b^T y_i + u_i v_j \tag{3}$$

where μ is a global bias term, ω_i is the weight for user i, ω_j is the weight for user j, x_i and y_j are the user and item side feature vectors, and a and b are the weight vectors for those side features. The latent factors are u_i and v_j.

The model is trained by optimizing the equation

$$min_{w,a,b,v,U} \frac{1}{|D|} = \sum_{i,j,r_{ij} \in D} L(score(i,j), r_{ij}) + \lambda_1(||w||_2^2 + ||a||_2^2 + ||b||_2^2) + \lambda_2(||U||_2^2 + ||V||_2^2) \tag{4}$$

where D is the dataset, r_ij is the rating that a use i gave to item j, U is the set of the user's latent factors and V is the item latent factors. λ_1 is the linear regularization parameter and λ_2 is the regularization parameter. L is the loss function, which is given by $(\hat{y} - y)^2$. The Turi Create model we used computes two solvers and reports the best result. The two solvers are Stochastic Gradient Descent and Alternating Least Squares. This model in Turi Create is a generalization of traditional Matrix Factorization, in that it learns latent factors for all variables, including side data, and not just user and item interactions.

3 Data

The TripAdvisor data set used was obtained from the Database and Information Systems (DAIS) Laboratory at the University of Illinois at Urbana-Champaign [3, 4]. This dataset consists of over 1.5 Million reviews for 12,774 hotels, scraped from the TripAdvisor.com website. The dataset reviews contain the user's username, written review, and ratings for different aspects of each hotel room. The

hotel and its location are also included. There are 9 aspect ratings: Overall, the overall hotel rating that is aggregated and displayed on the hotel page; Value, the perceived value of the room for the cost; Rooms, the quality of the hotel rooms; Location, the rating of the hotel's location (proximity to attractions, destinations, etc.); Cleanliness, a rating of how clean the hotel/rooms are; Check in/Front desk, a rating for the hotel lobby and check-in process; Service, a measure of the quality of the hotel's service and staff; Sleep Quality, which measures the beds and sleep environment (noise, etc.); and Business Service, the rating for business amenities (internet, conference rooms, etc.). While the Overall rating is available to all users leaving a review, the other individual ratings have been used or not at different times. At the time of this analysis, only Service, Sleep Quality, and Cleanliness were available for users writing a hotel review.

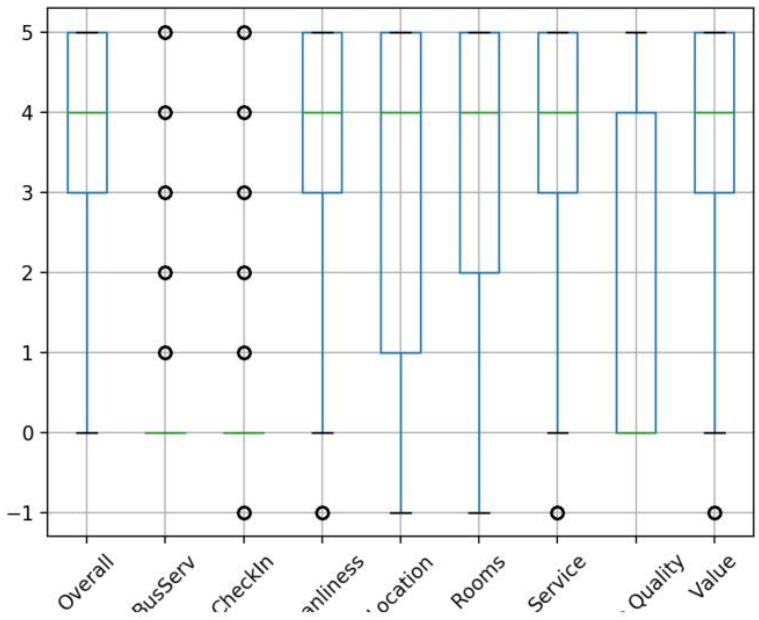

Fig. 1. Boxplot of Overall and Aspect Ratings features

As we see from the aspect ratings boxplot and correlation matrix, the aspect ratings for Business Service, Check In, and Sleep Quality are not strongly correlated to the Overall rating, and differ substantially in their distributions. Business Service and Check In have very few ratings in the data set. We expected that though these ratings are given individually, they would be correlated with each and the Overall rating. Chua et al. [5] investigated the reliability of TripAdvisor reviews and found that while text and scores usually were aligned, in some cases they found they were not. We did not implement a method of determining the reliability of the reviews, and as such, left the data that existed

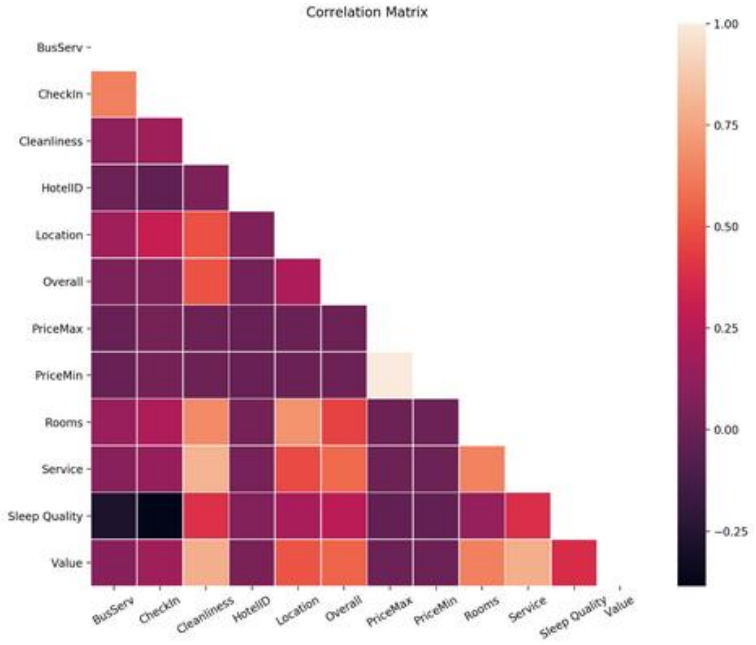

Fig. 2. Correlation Matrix of features

as-is. Previous work by Nilashi et al. [6] explored multi-criteria rating information, such as the aspect ratings side data in this dataset, and found that while it can potentially be used to improve prediction accuracy, the sparseness and noisiness of side data can prevent the realization of any accuracy improvements. We thus chose to replace missing data with the Overall score, as our focus is not on the absolute accuracy, but on the speed and relative accuracy of the models. Replacing the missing values this way provided significantly more features and reduced sparseness, as many of the aspect ratings were missing. If there is an advantage to be gained from the inclusion of this data, we want to see the matrix factorization model use it, under the assumption that real-world models will collect and include this data.

4 Methods

4.1 Models and Data Loading

The models were built and tested using the Turi Create package from Apple Inc.. The algorithms and math used in both packages is the same, with the main change being the addition of Apple's CoreML functionality. Graphlab has been extensively researched as a framework for machine learning by Low et al [7].

Due to computing challenges from loading and handling such a large data set, two approaches were attempted. The first was to use Amazon Web Services

(AWS) and hire a more powerful computing instance to run the analysis. This was only able to be used for the written text review processing and analysis, and the results are discussed below. As AWS does not have the type of operating systems available which Turi Create can be used with, it could not be used to build the recommender systems and compare them. The second approach, due to personal computer constraints, was to sample 1000 hotels, import those reviews (averaging around 120,000 reviews for 1000 hotels), create and compare the models, and repeat. The sampling method chosen was simple random sampling without replacement. This iterative process was repeated 10 times and the results recorded.

This approach was applied to two different tests. For the first, we simply processed the 1000 hotels and models, and recorded the results, as above. For the second, we wanted to examine the cold start problem, and did so by filtering the data set on the reviewers. Since the cold start problem is that users with few reviews are poorly matched, we first decided to remove any grouped, anonymous reviews, such as those by "A TripAdvisor member," as the model would think they are one user with a wide variety of tastes, since it uses the UserID for scoring. Next, we filtered out the users with only 1 review. This left us with a data set we then split in two: one group with exactly 2 written reviews, and one group with more than 2 reviews. We then processed the models, recorded the results, and compared them to both each other and to the original unfiltered analysis, to determine if having 2 reviews was enough to improve RMSE, or were more reviews needed.

4.2 Written Text Reviews

The written text reviews pose an interesting challenge, as they require some processing to be used in a recommender system. Much research has been done on using only the written text reviews, processing them, and building a model from this processed data, such as Leung et al. [8] using sentiment analysis in their approach. Other analyses simply remove the written text reviews before building the models, but this previous work by Liu et al. shows that processing of the written text and inclusion in the model can improve accuracy.

In order to include them in the model, we processed them using sklearns's TDF-IDF Vectorizer, then classified them to an aspect rating using a Linear Support Vector Machine (SVM). The SVM was chosen as previous research from Chen et al. [9], Xia et al. [10], and Duvvur [11] shows that this provides better accuracy than other models. This TF-IDF Vectorization process was used effectively by Liu et al. [12] to predict Overall scores from review text, but it was not incorporated with the rest of the data into a model like ours. The TF-IDF (term-frequency inverse document-frequency) Vectorizer is a bag of words method of handling text. The words are weighted using the TF-IDF method, where words are re-weighted according to the multiple of the term-frequency and inverse-document frequency. The TF, IDF, and TF-IDF are defined as:

$$idf(t) = log(\frac{Total number of documents}{Number of documents with term t in it}) \tag{5}$$

$$tf(t) = \frac{Number of times term t appears in documents}{Total number of terms in the document} \tag{6}$$

$$tf - idf = tf(t) - idf(t) \tag{7}$$

This serves to reduce the impact of frequent words ("the", "an", "is", etc.) and increase the impact of infrequent, rarer words. The Linear Support Vector Machine was used as noted above for classification with high accuracy, and we achieved a similar high-accuracy result. The main parameter to specify is the number of word combinations ,or ngrams, that will be calculated by the model. We found the highest accuracy with 1 and 2 word ngrams.

We processed and classified, using default parameters, the entirety of the text reviews (over 1.5 million), with a resulting 95% accuracy. Due to the high accuracy of the result, we moved forward with including this review text aspect rating in the models.

4.3 Evaluation Methods

All processing of results was performed on a MacBook Pro with 2.7 GHz Intel Core i5 processor, with 8 GB of 1867 MHz DDR3 RAM and a 128 GB solid state HDD. The measure we used for evaluating efficiency was the time used to process the recommend function of each model. For the accuracy, the metric we used is the Root Mean Squared Error (RMSE). RMSE is the square root of the Mean Squared Error (MSE), and is for this collaborative filtering analysis is defined as

$$RMSE = \sqrt{1/T \sum_{u,i}^{T} (\hat{r}_{ui} - r_{ui})^2} \tag{8}$$

where T is the total number of ratings in the data, $\hat{r}_{ui}$ is the predicted hotel rating, and r_{ui} is the actual hotel rating.

To separate into training and test sets, we use the Turi Create function "recommender.util.random_split_by_user." Typical methods of train/test split on data such as this may leave either the test or training data set with no data from any particular user, which would significantly degrade the predictive capabilities of the models. Instead, the Turi Create "recommender.util.random_split_by_user" function splits data for each user along the desired proportion. If a user has 10 reviews and an 80%/20% split for train/test, 8 randomly selected reviews from that user will go in the training data set, and 2 randomly selected reviews from that user will go in the test data set set. This way there is data for each user in both the test and training data set. We have selected 80%/20% as the train/test proportion for each user, and applied this to each model.

5 Results

We ran the analysis on 1000 randomly sampled hotels 10 times, and the results are shown for the time and RMSE of each model in Figure 3 and Figure 4.

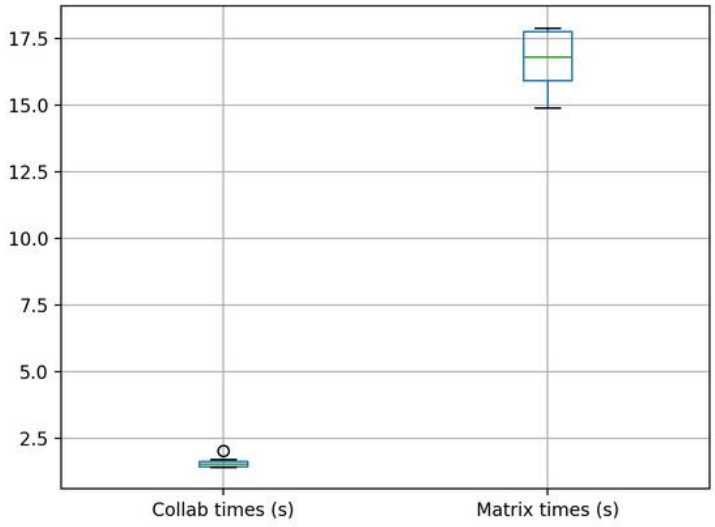

Fig. 3. Processing time boxplot of the two models

As above, we then filtered the dataset into two groups: reviews where the reviewer had reviewed more than 3 hotels, and reviews where the reviewers had reviewed 2 hotels. Those having reviewed only one hotel were dropped. We also ran this on samples of 1000 hotels, 10 times. Results for this analysis are in Figure 5 and Figure 6

Since the models for the two filtered datasets appear visually very similar to the original analysis, we decided to perform ANOVA against the null hypothesis that the means for all three datasets were equal. The result was an F-statistic of 2.91 and a p-value of .082.

6 Analysis

As we can see in Figure 3 and 4 above, the collaborative filtering model is significantly faster to process than the matrix factorization model. With a mean processing time for the 10 iterations of 1.58 seconds compared to the matrix factorization model's mean processing time of 16.73 seconds, we show that the collaborative filtering model is 10.58 times faster.

This is most likely due to how the two models process the data. The collaborative filtering model only uses the User ID, Hotel, and Overall rating, and

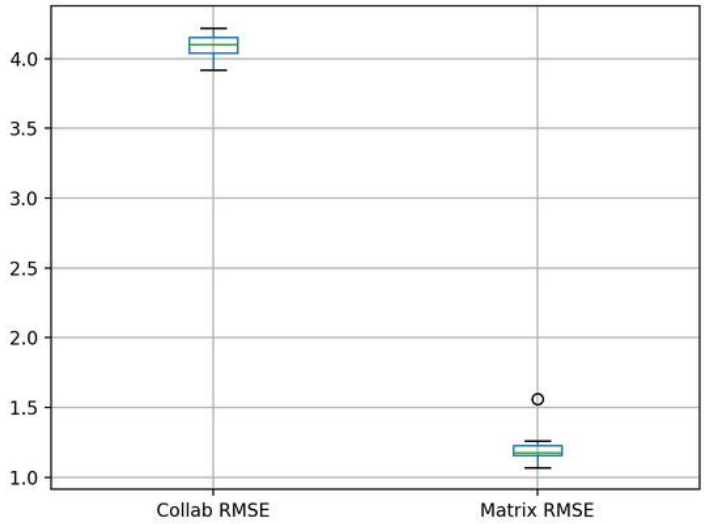

Fig. 4. RMSE boxplot of the two models

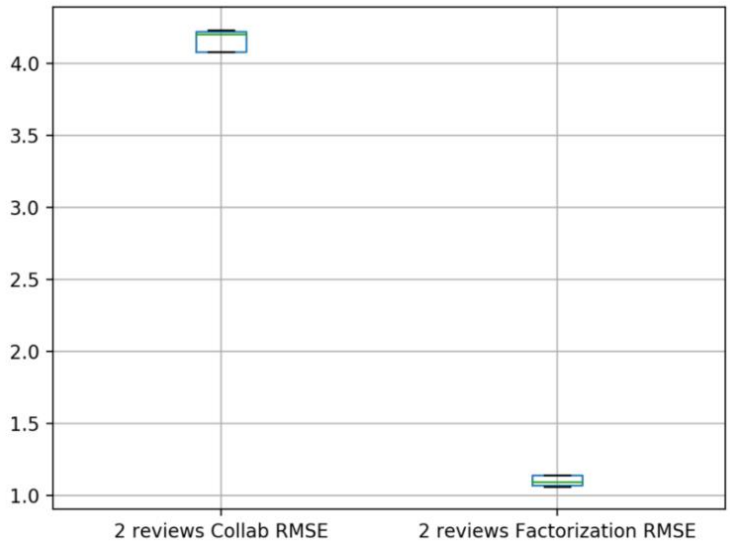

Fig. 5. RMSE boxplot of the two models from the 2 review dataset

generates a finite number of user profiles to match to other users. Each random sample of 1000 hotels typically has about 100,000 unique reviewers, with only a small percentage having written more than one review. That sets the limit of user profiles at that number of unique reviewers. In contrast, the matrix factoriza-

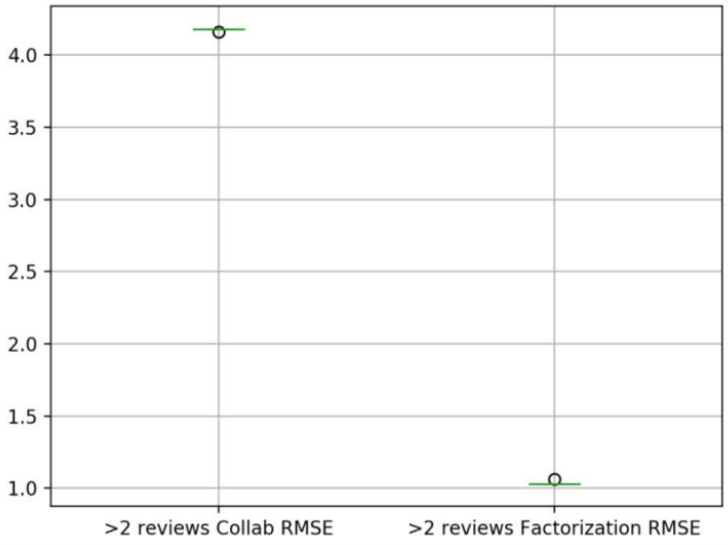

Fig. 6. RMSE boxplot of the two models from the ¿2 review dataset

tion model creates a much larger number of user-hotel pairs. If there are 100,000 unique reviewers and 1000 hotels, that creates a maximum of 100,000,000 user-hotel matches to evaluate, which is 1000 times larger than the number of user profiles created and evaluated in the collaborative filtering model.

The comparison of model processing times gives collaborative filtering as clearly superior, so next we look at the RMSE. With a mean RMSE of 1.21 for the matrix factorization model to the mean RMSE of 4.09 for collaborative filtering, matrix factorization is superior.

RMSE is notably difficult to interpret. Rehorek [13] gives the errors in the RMSE as the difference between the predicted rating and actual rating for recommender systems. For example, if a hotel was rated a 4 by a user, and the model predicts 2.5, this would give an error of 1.5. As given in the formula above, these errors are then squared, summed, divided by the number of reviews, and the square root taken. An RMSE value of 4.09 for the collaborative filtering model is quite poor, representing an average error per rating of over 4. Since the ratings only go from 1 to 5, this represents a significant error.

The next step of the analysis was to filter the dataset and examine whether having only reviewers with more reviews improved the RMSE. Since collaborative filtering theoretically performs better with more data for each user, we would expect an improvement over the original model. The matrix factorization model should be unaffected by this, but we ran it for consistency with the original analysis. As we can see in Figure 5 and 6 above, there appears to be no improvement. Visually, the results are very similar to the original analysis. Due to this result, we performed ANOVA, as above. The p-value of .082, being

higher than our threshold of .05, means we fail to reject the null hypothesis that the means are equal. Thus, we find no evidence that there is a statistically significant difference between the mean RMSE between the original model with the full dataset, the model which only included reviews where the reviewer had written exactly 2 reviews, and the model which only included reviews where the reviewer had written 3 or more reviews.

This result is surprising. The intent was to examine a facet of the cold start problem, and look at how many reviews are needed to build a more accurate collaborative filtering model. The likely explanation is that due to the number of hotels, our sampling approach, and some aspects of the data, these models are all similar because there would need to be a much larger number of reviews per reviewer to increase the accuracy. Given the large number of hotels, the ability to make a user profile from reviews that covers any number of them is small. Since the collaborative filtering model ignores all data but the hotel, user, and Overall rating, the user profile built likely matches very few other users well. This is compounded by the sampling approach we took due to computing constraints. A user may have 500 reviews in the system to build a profile, but if we only sample 2 of those reviews each time, the profile will be inaccurate, especially compared to using the entire dataset. As for the data, since most of the Overall review scores are high, with a mean of 4, it may be easier to build user-user profiles as a result, and the model may be more accurate due to this skew. Since almost all users rate highly, this has the effect of reducing the variation between users, and makes more users similar to others.

7 Ethics

Ethics are a significant concern in our research. The data provided freely to TripAdvisor may be used to build a detailed profile of a user, which could be dangerous. For example, a user who reviews different hotels every summer could be profiled as someone to rob while they are on vacation. The data has usernames and location data, but user data can also be searched on the site or Google. This allows relatively simple identification of users who use the same usernames across multiple sites.

We selected users with usernames containing unusual or potentially identifying information and multiple reviews as a starting point for our investigation. While many user reviews and data kept by TripAdvisor show the users first name and last initial, we found that this was not useful for identifying a user. For example, one sample tested for analysis had 13 reviews from a user “David B”, however the user location data shows this was likely 9 different users, and identifying each was difficult. After reviewing the data and pulling usernames from multiple samples, we found two users we will use to highlight the ethical concerns. We will show how the ACM Code of Ethics [14] General Ethical Principles 1.2 Avoid Harm and 1.6 Respect Privacy are violated.

The first user we will discuss is Pawel_EPWR, A search of EPWR on Google brings up multiple references to Wroclaw Strachowice airport, which confirms the

user's location of Wroclaw, Poland. Searching the username shows an Instagram with the user's full name, and this was verified by comparing the user's Instagram pictures to Pawel_EPWR's travel history via hotel reviews in the dataset. This Instagram also contains a picture of the user. So to summarize, with a few simple Google searches and an unusual username, we were able to find a real name and picture of a TripAdvisor user. Next, we then researched each of the hotels reviewed by the user, and identified that many of them are gay hotels, which TripAdvisor tags. This tagging metadata was not available in the dataset, but is readily available now on each hotel's page on the site. This was added in the time since the dataset was collected. With this information, we now have a breach of the user's privacy that is significant: it is highly likely the user is a gay man. While homosexuality is legal in Poland, there are more restrictions on gay people there than in most of the West, and this information could be used to discriminate against the user or otherwise be used against him. While this user may not suffer from this breach, it is easy to see how a user from a country with more restrictions on homosexuality could also be identified in a similar manner. Given the nature of travel, where many people seek new or different experiences, users may specifically travel for the purposes of engaging in acts or behavior that is illegal in their home countries. In the case of homosexuality, in some countries this can mean execution as a punishment. This adds a dimension of risk to the TripAdvisor review, where a user reviewing hotels tagged with "gay hotel" or other such identifiers could be identified and face punishment, or other harm, at home. There is no mention of any such risk on the TripAdvisor site, or any reference to these issues in their privacy policy.

The second user, stevied888, is slightly different. A google search of the username pulls up a candidate profile for someone who ran for public office, with their email as stevied888@gmail.com, as well as a full name, address and phone number. The user location for this candidate can be matched to the user location for the user profile on the TripAdvisor site, as well as the user location field in the dataset. This username also matches a Pinterest page with that same username, which also gives the user's matching full name. This case presents several challenges. The first is that if the user writes a negative review, a displeased hotel owner could easily find this information and harass the user via email, mail, or phone. The ability to find a phone number or address using an online profile is not an expected result, as most users assume some level of anonymity. Second, a user running for public office likely does not want their travel history and reviews able to be easily linked to their professional life. Even if a user has nothing to hide, this information could be used in attack ads or part of a smear campaign, possibly by misrepresenting the user's words from an online review.

Both of these users highlight how this data could be used in a negative way to harm an individual or violate their privacy, and is a completely unintended use. There is no warning on TripAdvisor that the reviews could be aggregated in a way that reveals hidden insights, such as a user's sexual orientation. Often, this is the actual value data scientists provide. It is easy to see how gay users would want to find hotels that cater to their needs, and that TripAdvisor would collect

data for that purpose. The challenge is in access to that data. A user who goes to a hotel tagged as "gay hotel", or one known for prostitution, may not want that association known publicly, so common sense would dictate that access to that information should be protected. Through the TripAdvisor data, though, we have shown how easy it can be for anyone to tie that kind of information to an actual person.

8 Conclusions and Further Work

As a result, given the much faster processing time for the collaborative filtering model with poor accuracy, and the comparatively very slow times but much higher accuracy for matrix factorization, it's difficult to say which model is superior. While much more accurate, the times required for the matrix factorization model are a significant challenge. Some of this processing time could be mitigated by more sophisticated computer hardware. Given that this model processing was run in relatively small samples, and not the entire data set, it's difficult to know what the effect of more computing power would have when all of the data is processed.

Still, we know that one laptop could effectively run and create the collaborative filtering model for some portion of the data in a real-world acceptable timeframe. On today's internet and with today's use, no user would accept a website that served recommendations after a 10+ second delay, as with the matrix factorization model. That itself shows the value of the collaborative filtering model. The accuracy problem would still need to be solved, however.

With this data, we can only build and evaluate the recommender system, and predict how a user would rate the hotel. The other side, evaluating the impact of these recommendations and how the user responds and they turn into bookings or sales, is unavailable to us. How the ratings affect which hotel the user will choose to book when presented with multiple recommendations would be interesting research to engage in. Whether or not it is the most accurate recommendation that is chosen is unknown.

The lack of improvement in accuracy through the filtering of the data set into different groups is surprising, and suggests a need to examine the cold start problem more deeply. It is difficult to say how many reviews are necessary to solve it, but even with multiple users having 20+ reviews in some samples, it did not affect the accuracy of either model. In the case of hotels with the collaborative filtering model, each user-user comparison is likely only to matter if two users have reviewed one or more of the same hotels, as there is no other distinguishing information used. Several methods have been proposed that would solve this problem without the need to visit and review many hotels before accurate recommendations could be given. Nadimi-Shahraki et al [15], for example proposed using an interview or questionnaire to classify the user. Hu et al [16] integrated a clustering approach into the pipeline prior to the collaborative filtering step to use side data that wouldn't be used in the model, and achieved improved accuracy with this method.

A future analysis could also be done using a different package and collaborative filtering model that uses more data to see how much of an impact that has on the accuracy of the model and these metrics, as well as the cold start problem. If the model uses more of the features than our package did, this may lead to improvements in accuracy for the collaborative filtering model that are still faster than matrix factorization due to the differences between user-user pair and user-hotel pair computation times.

References

1. Lee, J., Sun, M., Lebanon, G.: A Comparative Study of Collaborative Filtering Algorithms. (2012) https://arxiv.org/abs/1205.3193.
2. Moshfeghi, Y., Piwowarski. B., Jose, J.M: Handling data sparsity in collaborative filtering using emotion and semantic based features. (2011)
3. Wang, H., Lu, Y., Zhai, C.: Latent Aspect Rating Analysis without Aspect Keyword Supervision. 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. (2011) 618-626
4. Wang, H., Lu, Y., Zhai, C.: Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach. 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. (2010) 783-792
5. Chua, A., Banerjee, S.: Reliability of Reviews on the Internet: The Case of TripAdvisor. Proceedings of the World Congress on Engineering and Computer Science, Vol. I. (2013)
6. Nilashi, M., Jannach, D., bin Ibranim, O., Ithnin, N.: Clustering- and Regression-based Multi-Criteria Collaborative Filtering with Incremental Updates. Information Sciences. (2014)
7. Low, Y., Gonzalez, J., Kyrole, A., Nickson, D., Guestrin, C., and Hellerstein, J.:, GraphLab: A New Framework for Parallel Machine Learning. (2010) https://arxiv.org/abs/1006.4990
8. Leung, C., Chan, S., Chung, F.: Integrating Collaborative Filtering and Sentiment Analysis: A Rating Inference Approach. Proceedings of ECAI Workshop on Recommender Systems. (2006) 62-66
9. Chen, L, Zhang, J.: Prediction of Yelp Review Star Rating using Sentiment Analysis. (2014)
10. Xia, H., Peng, L.: SVM-Based Comments Classification and Mining of Virtual Community: For Case of Sentiment Classification of Hotel Reviews. Proceedings of the International Symposium on Intelligent Information Systems and Applications. (2009) 507-511
11. Duvvur, P.K.: Predicting Hotel Rating Based on User Reviews. (2016)
12. Liu, D, Chai, Y., Zheng. C., Zhang, Y.: Rating Prediction Based on TripAdvisor Reviews. (2017)
13. Rehorek, T.: Evaluating Recommender Systems: Choosing the best one for your business. (2016) https://medium.com/recombee-blog/evaluating-recommender-systems-choosing-the-best-one-for-your-business-c688ab781a35.
14. Association for Computing Machinery.: Code of Ethics and Professional Conduct. (2018) https://www.acm.org/code-of-ethics.
15. Nadimi-Shahraki, M-H., Bahadorpour, M.: Cold-start Problem in Collaborative Recommender Systems: Efficient Methods Based on Ask-to-rate Technique. Journal of Computing and Information Technology. (2014) 105-113

16. Hu, Y., Lee, P., Chen, K., Tam J., Dang, D.: Hotel Recommendation System Based on Review and Context Information: A Collaborative Filtering Approach. Pacific Asia Conference on Information Systems. (2016)