CSE 258 - Assignment 2
Amazon Sports Product Rating Prediction System

Yun-Yi Lin A59002388
Kai-Ning Keng A59002368
Yu-Mu Chen A53311606

Introduction

In recent years, online shopping has become more and more popular for its convenience; being able to get whatever you need with just one simple click on the mouse is truly a wonderful feat of living in modern world. When doing online shopping, one feature that people often use without noticing its significance is the recommending column that comes along with the product page. It serves as a bridge between customers and providers, giving buyers more things that they might want, creating more business opportunities for merchants to sell products, while netting the online service provider more profit in the process, and thus creating a triple win situation. Therefore, it is crucial for the owner of such online service to build a reliable recommending system, and with the help of machine learning techniques, we can let the system "learns" by itself what to recommend to a certain customer based on various information; one way of doing this is to predict the rating that a user would give to an item, and only recommend it if it passes a certain threshold.

There are numerous things that can effect the ratings that a product receives. For example, if a trend for skiing arises, there will be a higher demand for corresponding equipment, and the ratings for related products could change more drastically than other times. Therefore, we believe that the time of the purchase can have its effect on ratings as well. This can be potentially useful when the current trend of purchase fits some time in the past. For instance, in some Asian countries, people might have more demand on face masks under the COVID-19 influence, just like back in 2004 when the SARS crisis happened. Or, we can assume that the demand for jackets and coats will be higher in winter months than in summer.

In this project, our main goal is to build a system that can achieve the prediction of ratings. We attempt this task with a simple latent factor model that considers the bias of users and items. Furthermore, we do some experiment on taking the time of the purchase into consideration, exploring the possibility to generate a more interesting system that can predict more accurately based on current state in the world. Among the various choices of online shopping services, Amazon is easily one of the most reliable and most widely used website with robust reviewing system, thus we use their data as the dataset for this problem. For this project, we focus on the sports products.

1 Dataset

The dataset we choose to use in this project is the small 5-core Amazon "Sports and Outdoors" product review dataset provided by Prof. Julian McAuley. It contains 2839940 reviews of 222146 products given by 12982 users. The distribution of the ratings in these reviews are shown in figure 1. It can be easily noticed that users tend to give a higher ratings in general. Therefore, when using the latent factor model, we can expect that the biases will drift towards a higher result.

In order to know a little more about the purchase time distribution and its relationship with product ratings, we calculate the average rating of each year. Firstly, figure 2 shows the number of reviews in the

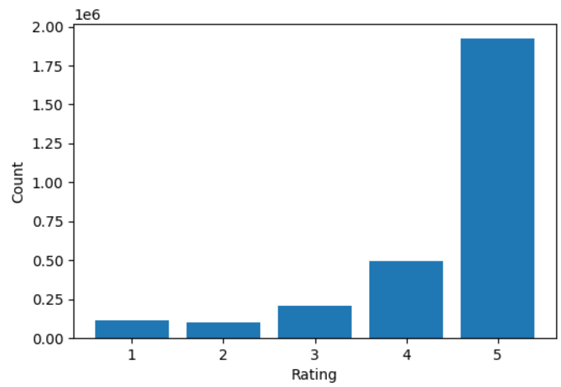


Figure 1: Distribution of ratings

dataset with respect to time. As we can see in the figure, the numbers of reviews before the year 2012 are significantly lower than those of 2012 onward, which means that we cannot reliably estimate the bias caused by purchase year if it's before 2012. Therefore, in the process of training, we only consider the bias caused by year of purchase if it's 2012 or after. After deciding the effective time range, we draw the plot of the average ratings of reviews each year, as shown in figure 3. In the plot we can see that as time moves on, the average rating generally rises, but not indefinitely.

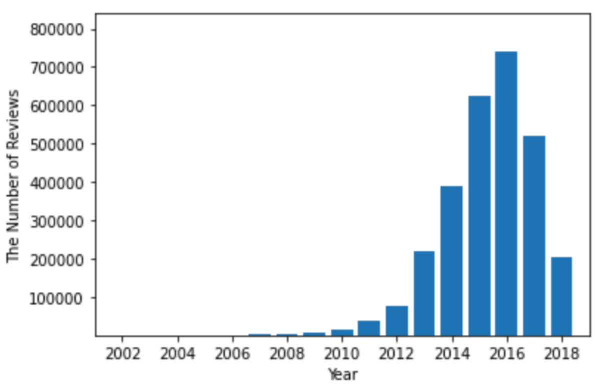


Figure 2: Number of reviews each year

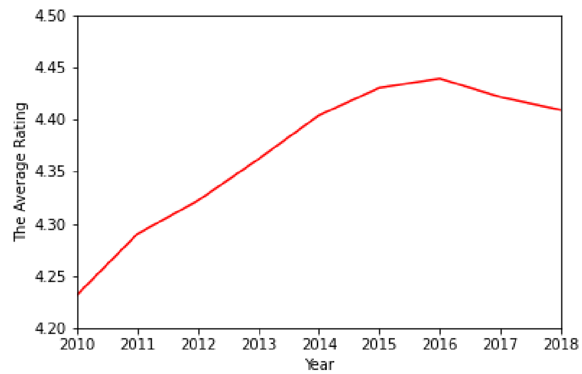


Figure 3: Average rating of each year

After exploring the properties of year distribution, we consider the month properties. Same as before, we check the distribution of reviews over different months in the year, shown in figure 4. From the graph we can see that the dataset is fairly evenly distributed between months, so we don't need to do additional pre-processing beforehand. Drawing the average ratings in each month gives figure 5. The ratings fluctuate as time goes by, peaking at May and August, according to the plot.

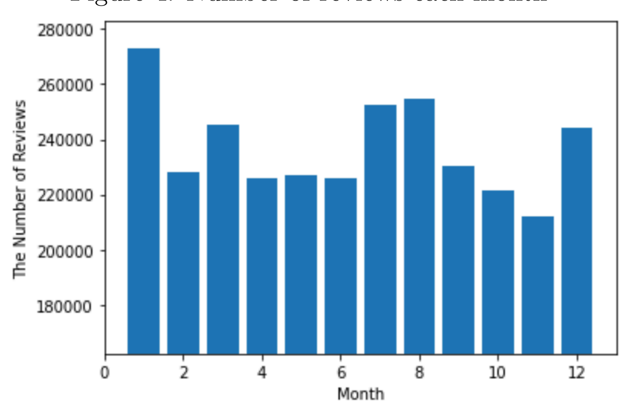
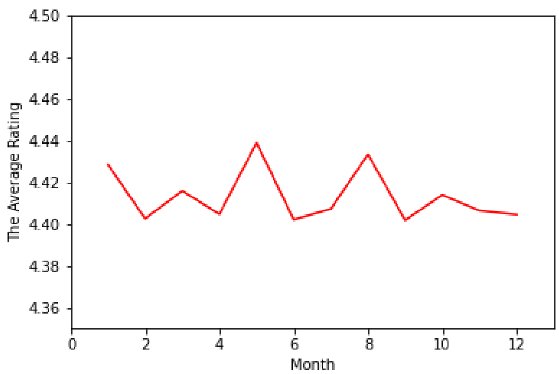


Figure 4: Number of reviews each month

Figure 5: Average rating of each month



2 Predictive Task

The task for this model is to predict a rating based on the user, the item, and the year and month of purchase. We believe that a latent factor model should handle this task well, starting from the mean rating and taking biases caused by user, item, and time into consideration to adjust the prediction, because this can effectively capture the hidden relation between these features. In this project, we try out three variants of latent factor model: one that only considers user and item biases, another that adds the year into consideration, and the other that involves biases of user, item, and month. For the baseline model, we use a model that always predicts the mean rating of training data, and we use mean squared error to evaluate the models and verify whether time is actually a factor that matters.

3 Model

As stated before, we try to use three different latent factor model to attempt at the task. The first one only considers the biases caused by the user and the item, while the second and the third takes year and month into account, respectively. The optimization problem for the three models looks like this:

$$\text{Model 1: } \arg\min_{\alpha,\beta} \sum_{u,i} (\alpha + \beta_u + \beta_i - R_{u,i})^2 + \lambda \left[\sum_u \beta_u^2 + \sum_i \beta_i^2 \right]$$

$$\text{Model 2: } \arg\min_{\alpha,\beta} \sum_{u,i,y} (\alpha + \beta_u + \beta_i + \beta_y - R_{u,i,y})^2 + \lambda \left[\sum_u \beta_u^2 + \sum_i \beta_i^2 + \sum_y \beta_y^2 \right]$$

$$\text{Model 3: } \arg\min_{\alpha,\beta} \sum_{u,i,m} (\alpha + \beta_u + \beta_i + \beta_m - R_{u,i,m})^2 + \lambda \left[\sum_u \beta_u^2 + \sum_i \beta_i^2 + \sum_m \beta_m^2 \right]$$

Note that if year is out of the effective range of [2012, 2018] in model 2, we will ignore λ_y because we don't want the few data in certain years effect our prediction too much. In addition, to improve the performance of these models, we also try to tune two different regularization parameters for different biases for each model, since we believe that the biases caused by different factors will likely have different impact on the rating. For the models that consider time, we think that time will have a more different effect, so we train them with one λ on user and item, and another on time. Therefore, there are 6 models in total, the rest of them look as follows:

$$\text{Model 4: } \arg\min_{\alpha,\beta} \sum_{u,i} (\alpha + \beta_u + \beta_i - R_{u,i})^2 + \lambda_1 \sum_u \beta_u^2 + \lambda_2 \sum_i \beta_i^2$$

$$\text{Model 5: } \arg\min_{\alpha,\beta} \sum_{u,i,y} (\alpha + \beta_u + \beta_i + \beta_y - R_{u,i,y})^2 + \lambda_1 \left[\sum_u \beta_u^2 + \sum_i \beta_i^2 \right] + \lambda_2 \sum_y \beta_y^2$$

$$\text{Model 6: } \arg\min_{\alpha,\beta} \sum_{u,i,m} (\alpha + \beta_u + \beta_i + \beta_m - R_{u,i,m})^2 + \lambda_1 \left[\sum_u \beta_u^2 + \sum_i \beta_i^2 \right] + \lambda_2 \sum_m \beta_m^2$$

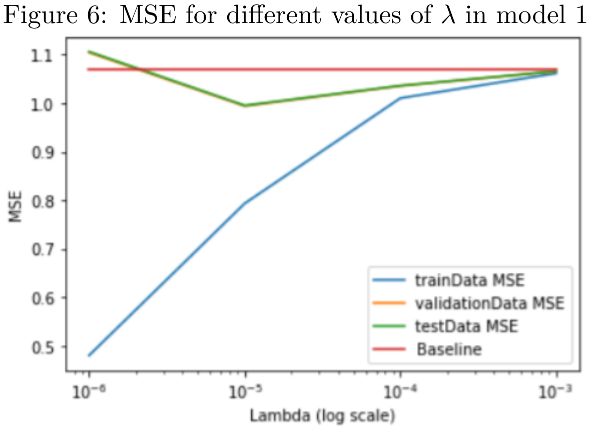
In the fitting process, we shuffle the data randomly first, and then use the first 500000 data as training, next 500000 for validation, and the rest for testing. We found that the MSE converges fairly quickly after about 25 iterations, therefore for all the training experiments we set the max iteration to be 30.

4 Literature

5 Results

For all the single λ models, we try out four values for λ: 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}. Plotting the results for the first model yields figure 6. As we can see, while the MSE on the training data keeps decreasing as λ gets smaller, when λ hits 10^{-6} the performance on testing data lost even to our baseline model, meaning that a serious overfitting occurred. Therefore the best value for λ that we found is 10^{-5}, yielding a MSE of 0.9952315429428911 on testing data.

Figure 6: MSE for different values of λ in model 1



For model 2, ...

As for model 3, ...

Based on the observations above, unfortunately, we find that the models that consider time generally perform worse than the one who doesn't. However, having two different λs for different biases might change this, therefore we try tuning two parameters and here are the results:

For

References

[1] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In Proceedings of the 25th International Conference on World Wide Web (WWW '16). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 507–517. DOI:https://doi.org/10.1145/2872427.2883037

[2] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15). Association for Computing Machinery, New York, NY, USA, 43–52. DOI:https://doi.org/10.1145/2766462.2767755