STA582: Datafest 2017

Hotel Recommendation for Expedia via Collaborative Filtering

Authors: Menglan Jiang

1 Motivation

In this project, we are interested in building up a recommendation system for Expedia. We are given two datasets. One is a destination-specific data, the other is a user-related hotel data. For the destination dataset, we have all kinds of information about a hotel, including its location, features, nearby sightseeing places and etc. For the other dataset, it is all records of the browsing history of users. The way how the data is collected is based on both accounts and IPs. Therefore, if different people use the same account to search for hotels, their browsing history will appear under the same user ID. In addition, we can also access the longitude and latitude of locations where the search is executed. Furthermore, all information concerning the search, such as the destination, the hotel, are recorded.

Based on the information given, we want to make predictions on the preference a user would give to each hotel in the candidate pool, and therefore, we can recommend hotels based on his or her preference. The broad idea is that we can use information of the two datasets to calculate similarities between hotels and users and then recommend based on similarities. The academic foundation for our project is user item collaborative filtering methods. There are a great number of researches that concentrate on the relevant area and it is widely explored in real business world. Therefore, we believe our method has a robust theoretical foundation and a practical usage in the industry.

2 Problem definition

The final objective of this project is that we can inference the score of a user for a hotel via the prediction function. A score matrix for each user among candidate hotels is one that indicates the preference of the user for various candidate hotels. The higher the score, the higher the probability that the user will prefer the hotel. As a result, hotels with higher score will go before than those with lower scores on the webpage. In order to achieve this goal, several steps are carried out.

First, similarities are estimated between different hotels, as well as users. Assuming there are n users and m hotels, the similarity matrices are n by n and m by m. Secondly, we assign scores to records in our datasets. For example, if a user has previously booked a hotel, then the score for that combination, that is user i on hotel j, will be high. Otherwise, the score will be lower if the user looks into the hotel but doesn't book the hotel. This is one of the most challenging part of the project as there are so limited number of successful bookings. In order to make the inference possible, we have to simulate some scores for existing browing history. For instance, we believe that a user looked into a hotel three times has higher preference compared to hotels that are only looked once. A precise estimation is a strong foundation for later inference. The last step is use a loss function to estimate the coefficients of our prediction function. The prediction function has two coefficients and a intercept. We try to minimize the sum of squared loss between the real data and the prediction. We use gradient descent to get the optimization of the loss function.

3 Models and methods

The item-item collaborative filtering methods consist two major part.

Firstly, we need to calculate the similarity. In the project, we need to estimate similarities for users and hotels. There are different methods for similarity calculation. We choose adjusted cosine distance, as this method will magnify both the similarity and difference. The formula is as follows,

$$sim(u_i, u_i') = \frac{(u_i - \bar{u})^T W_1 (u_i' - \bar{u})}{\sqrt{(u_i - \bar{u})^T W_1 (u_i - \bar{u})}\sqrt{(u_i' - \bar{u})^T W_1 (u_i' - \bar{u})}} \tag{1}$$

where $u_i \in R^m \quad i \in (1, ..., K)$; $v_j \in R^n \quad j \in (1, ..., L)$

and $W(i, i') = (1 + sim(i, i'))/2$.

In the formula above, u_i is a vector that storing the information of a user, such as location, destination, and etc. v_i is the corresponding vector for hotels. We also add an adjusted matrix, that is the weight matrix W.

Next, we create a prediction function for estimating the score based on existing neighboring score.

$$\hat{S}_{ij} = \beta_0 + \beta_1 \frac{\sum_k S_{kj}\sigma_{ik}}{\sum_k I_{kj}\sigma_{ik}} + \beta_2 \frac{\sum_k S_{kj}\lambda_{ik}}{\sum_k I_{kj}\lambda_{ik}} \tag{2}$$

where σ is the correlation between users and λ is the correlation between hotels, I_{ij} is the indication of a existing combination and S_{ij} is the corresponding score.

The loss function is as follows,

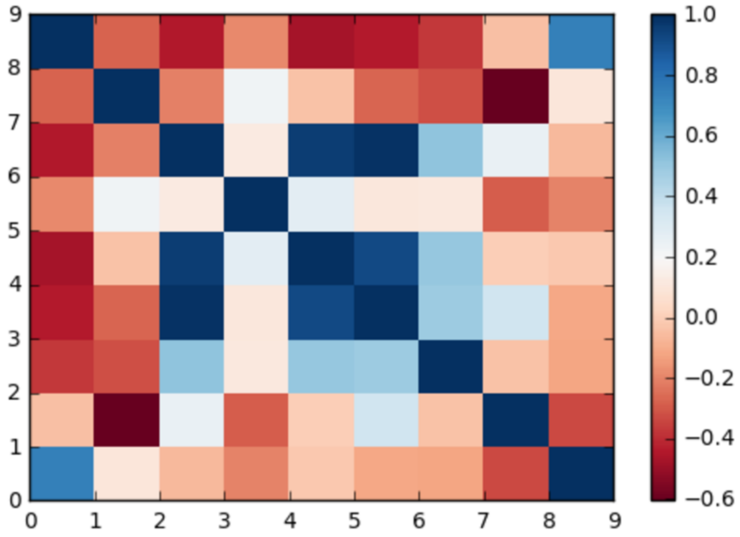
$$L(\hat{S}_{ij}, S_{ij}) = \sum_{ij} I_{ij}(\hat{S}_{ij} - S_{ij})^2 \tag{3}$$

We use the gradient descent method in python to optimize the function. The estimated coefficients and results are present in the next section. The prediction function above incorporates both user-based effect and hotel-based effect. We are trying to figure out the corresponding effects from similar users and hotels. In the real calculation, we only use a subset of users as the computational capacity is limited. We chose users whose locations are in the U.S and the top 50 booked hotels.

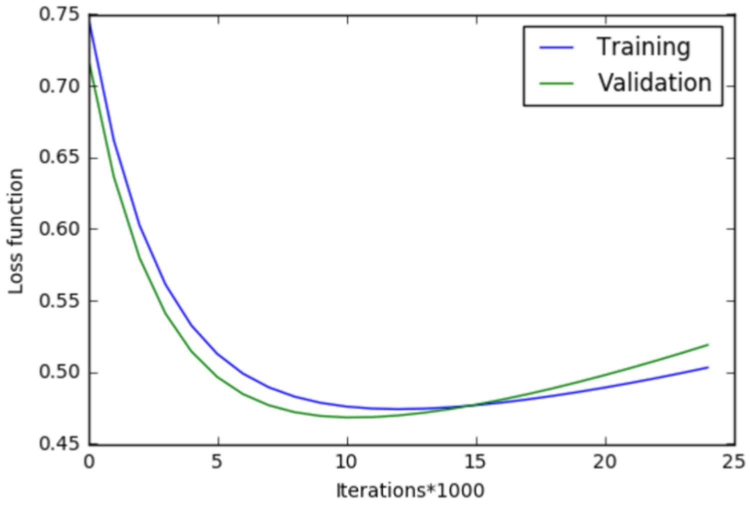
4 Results and validation

Based on the optimization result, we have $\beta_1 = 1.238$ and $\beta_2 = -0.365$. The estimation provides reasonable intuition, a similar user will increase the probability for preferring the same hotel while a similar hotel will result in negative effect. Specifically, user i will have more similar preference with user j who is more similar under the measure. In contrast, similar hotels are competitors and will lower the probability of a user preferring a specific hotel. The average loss is at around 0.6 and changes across number of iterations.

The graph below is a visual interpretation of similarity matrix between users. As can be seen from the graph, users' similarity is well calibrated. The cold color means that two users are highly similar while warm color means that two users are less similar. The graph shows different levels of similarities. The same patterns apply to hotels, but we will not show the graph in the paper.



The graph below shows the prediction capacity for both the training and test data. We use 30% of the sub dataset as the test data. As can be seen from the graph, the minimum loss converges at around 10^4 iterations. I think the prediction performance is satisfying given the information we have at hand.



At the current stage, we haven't tried other methods. Therefore, we can't argue the relative performance of our model compared to others. However, the loss function of the trainng and test data shows highly similar patterns, we believe our model is well calibrated. In terms of how to estimate uncertainty, we are considering using different sub dataset and run the same procedures and check whether the loss differs greatly when applying to different data.

In conclusion, similar users will provide positive impact on choosing the same hotel while similar hotels are negatively correlated due to competitions. The prediction performance of collaborative filtering is good in terms of the losses for both the training and test data.