Hotel Recommendation Based on Hybrid Model

Jing WANG, Jiajun SUN, Zhendong LIN

Abstract: This project develops a hybrid model that combines content-based with collaborative filtering (CF) for hotel recommendation. This model considers both hotel popularity in input destination and users preference. It produces the prediction with 53.6% accuracy on test data-4% improvement on purely content-based model. Addtionally, three issues are well-resolved when implementing CF: sparsity in utility matrix, cold-start, and scalability.

Keywords: Collaborative filtering, Content-based, SVD, Hierarchical clustering, Decision tree.

I. Introduction

The goal of the project is to develop a hybrid model for better hotel recommendation. At this moment, the majority of the recommendation systems are content-based models, which only consider the searching paramaters input by customers but not the users preference. For instance, Expedia focuses on the searching criterion and recommends the top popular local hotels. Personalizing the user search by their preference is a burning need for better hotel recommendation. Collaborative filtering is considered as the starting point of this project. It has been widely used in recommendation systems but rarely in hotel recommendation.

Nevertheless, there are still related works. *Ryosuke Saga. et al*[1] created a preference transition-based network system to recommend hotels. By tranversing user booking history, a transition network of user preference is constructed to do recommendation. But the network is too specific to accept new users and detect the further changes of old customers. *Xiong Yu-ning.et al*[2] came up with a personalized intelligent hotel recommendation system for online reservation. This research firstly extracts Hotel Characteristic factors, attempts to analyze customers browsing and purchasing behaviors and secondly constructs a personalized online hotel marketing recommendation system polymerization model for Multi-level customers. They combined user-item sytem and achieved positive outcomes. But it does not expand for new users.

In this project, hybrid model is applied to combine user preference and item properties. Based on the final comparison of accuracy, the model achieves good results. More Details are as follows.

II. Methodology

In this part, three models will be introduced. Content-based model and collaborative filtering are traditional methods in recommendation system. Hybrid model compensates the shortcomings in two models by combining these two models successfully. At the same time, it introduces new methods.

A. Content-based Model

Content-based filtering is a common approach in recommendation system. The features of the items previously rated by users and the best-matching ones are recommended. In our case, the local popularity of the hotel clusters based on ratings by users is used to be the main feature in the content-based model. More details will be explained later.

There are three main shortcomings of this approach[3]:

(1). It is limited by the number and the types of features associated with the objects for recommendation.

(2). It may involve the issue of over-specialization as no inherent method is included for finding something unexpected.

(3). It may not predict precisely for new users. Usually, a content-based recommendation system need enough ratings to provide accurate recommendations.

In our project, we used content-based filtering as a reference for result comparison.

B. Collaborative Filtering

The philosophy of collaborative filtering is to identify similar users and give recommendation based on the preference of similar user. But collaborative Filtering have the following issues. First of all, user behavior are chaotic, such as the gray sheep problem. The gray sheep refers to the users whose preference do not consistently agree or disagree with any group and itself. In our dataset

of the next experiment, we find some users choose different hotel cluster every time. This makes collaborative filtering extremely ineffective on those "gray sheep" users. Secondly, Collaborative filtering assumes that users make decisions purely due to their preference. However, we find their choices are highly correlated with hotel destination. A destination sometimes only have certain types of hotel; and a certain type of hotel is very famous or popular in that destination. Thus, the users' choices will be significantly limited by destination or influenced by destination. Besides the chaotic user behavior and hotel destination influence, collaborative filtering have utility matrix sparsity and data scalability issue that we will address them in detail in next Hybrid Model section.

C. Hybrid Model

The goal of hybrid model is to resolve two big problems. First of all, we need to work out three big issues of CF mentioned in part B. On the other hand, we would like to combine users preference and popularity of hotels to recommend.

Utility Matrix

The utility matrix gives each user-item pair; a value represents the degree of preference of that user for that item. In the later experiments, we will use user ID representing user; and hotel cluster representing item. When a hotel cluster is viewed by a user, rating of 1 is given; when a hotel is booked, rating of 5 is given.

Hierarchical Clustering

In terms of scalability problem, hierarchical clustering is applied to cluster the large number of users into different clusters. Most of the users have little booking history (less than 5 booking history) in the data. This leads to a very sparse utility matrix. Also, the number of user in the data is massive, which makes it impossible to implement Matrix Factorization on the original utility matrix. Therefore, users are classified into user cluster and utility matrix is compressed based on that. The hierarchical clustering method builds a hierarchy of clusters, by moving up this hierarchy, similar pairs of clusters are merged as a cluster. To be more specific, cosine distance is applied to measure similarity between users. Normally, cosine distance requires data normalisation. The original rating is substracted by the average rating of that user cluster:

$$M'_{i,k} = M_{i,k} - \sum_{k}^{K} M_{i,k}$$

where, i represents user cluster, k represents hotel cluster and K is the total number of hotel cluster. $M_{i,k}$ means the rating of user cluster i on hotel cluster k and $M'_{i,k}$ is the normalised rating (M represents it in the rest of paper). Cosine distance is written as follow:

$$\text{cosine similarity} = \frac{\sum_{k=1}^{K} A_k B_k}{\sqrt{\sum_{k=1}^{K} A_k^2}\sqrt{\sum_{k=1}^{n} B_k^2}}$$

where A_k and B_k is the ratings on hotel cluster k by user cluster A, B, respectively and K is the total number of hotel cluster.

SVD (singular value decomposition) Method

After clustering the users based on their preference in utility matrix, the utility matrix might still be super sparse because it is also rare to a cluster of users to rate most of the hotel. We would like to find a method to fill the unrated entries in utility matrix by smallest error. Here SVD is applied to do that.

SVD seeks a low-rank matrix $X = UV^T$, where $U \in \mathbb{R}^{N\times C}$ and $V \in \mathbb{R}^{K\times C}$ (N is the total number of distinct customers, K is the number of distinct items and C is the dimension factor), that minimizes the sum-squared distance to the fully obseved target matrix M (here is clustered utility matrix) with the dimension $\mathbb{R}^{N\times K}$. Matrices U and V are initialized with small random values sampled from a zero-mean normal distribution with standard deviation 0.01. We minimized the following objective function[4],

$$\sum_{(i,k)} (M_{ik} - V_k^T U_i)^2 + \lambda(||V_k||^2 + ||U_i||^2)$$

where λ is the regularization parameter in order to avoid overfitting. To solve this objective, we can use stochastic gradient descent (SGD). After taking derivatives of the objective with respect to U and V, and the following is the update rules:

$$U_i := U_i + \alpha((M_{ik} - V_k^T U_i)V_k - \lambda U_i)$$

$$V_k := V_k + \alpha((M_{ik} - U_i^T V_k)U_i - \lambda V_k)$$

where α is the learning rate parameter.

After estimating U and V by iterating over the known (i, k) pairs in the data, user i's recommendation for product k can be estimate by computing

$U_i V_k^T$.

Decision Tree Classifier

In order to resolve the cold-start problem, ontology model is introduced. The ontology decision model[5] is making up by user ontology characteristics and results of sub-comities, which are regarded as attributes and classes, respectively. The ontology theory demonstrates that users profile determines the users behavior, to some extent. Generally speaking, we explore the users profile data, such as age, gender, occupation, class, location and so forth, to predict the users behavior. That is to say, if the users have similar or same profile information, we think they have same or similar preference so that they might do the same behavior.

Decision tree is to predict the cluster label of a new user by inputting the user's profile data. Decision tree is a high-level overview of all the sample data, which not only can accurately identify all categories of the sample, but also can effectively identify the class of the new customer. In order to avoid overfitting, cross-validation method is adopted to obtain the best decision tree. A procedure to do that in hotel recommendation is in Fig. 1.

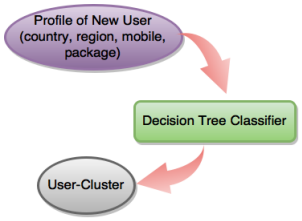


Figure 1: Decision tree classifier

Combination

The second problem we would like to resolve is combining the user preference with the item properties. Take the hotel recommendation as an example. Set a evaluation metric. If the hotel was booked, it is rated as 5 points. If it is just clicked, it gets 1 point. Otherwise, it is unrated. Based on booking history we have, a ranking matrix in terms of hotel properties (i.e, destination) can be created by average all ratings based on counts of booking or clicking, shown as follows.

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots d_{1K} \\ d_{21} & d_{22} & \cdots d_{2K} \\ \vdots & \vdots & \ddots \quad \vdots \\ d_{J1} & d_{J2} & \cdots d_{JK} \end{bmatrix}_{J \times K}$$

where J is the total number of hotel destinations, K is the total number of hotel type or hotel cluster.

From the previous procedures, a clustered SVD-utitlity matrix M is attained. We created a new matrix R with the dimension $N \times K$ by M and D. That is,

$$R_{ik} = M_k^{(i)} \cdot D_k^{(j)}$$

where i is customer ID, j is destination ID, k is hotel cluster ID, N is the total number of users that have booking histories.

M reflects the user preference, and D represents the popularity of hotel in local destination. Matrix R connects two attributes. The other benefit is for a single user's perspective. By clustering customers, booking histories from several users are combined together. But for a single user, he or she might not go to the hotel with the high rating of that cluster, which produces a great error if recommending the top hotel by purely clustered utitliy matrix. It is about 13% accuracy in the later experiment. Furthermore, the top cluster recommended by utility matrix might not in that destination input by the customer, which also leads to big bias. However, in the end, matrix R can get better tradeoff.

All the process is shown in the flow chart in Fig. 2.

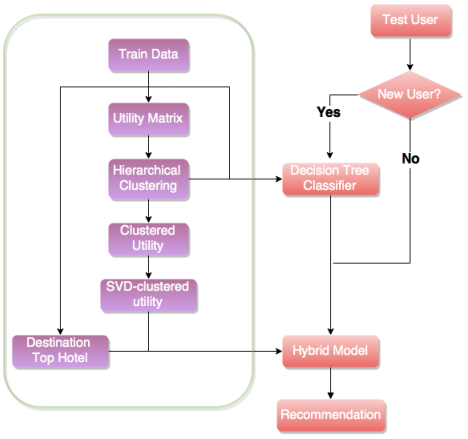


Figure 2: Hybrid Model Flow Chart

III. Experimental Results

In order to test the validation of hybrid model, datasets from random 20000 customers on Expedia are accepted to do experiments. The dataset is shown in Tab.1.

Table 1: Dataset Head

user_location_country	user_location_region	user_location_city	user_id	is_mobile	is_package	srch_destination_id	is_booking	hotel_cluster
66	258	5888	8798	0	0	25946	0	73
66	258	5888	8798	0	0	25946	0	73
66	258	5888	8798	0	0	25946	0	96
66	258	5888	8798	0	0	25946	0	48
66	258	5888	8798	0	0	25946	0	42

When applying hierarchical, in order to control whether two clusters should be merged or not, a

distance threshold should be specified. Here we set the threshold based on Fig.3. It is found that after 0.75, the number of hotel cluster starts to converge. Therefore, we set threshold = 0.75 resulting 112 user cluster. After clustering utility matrix, SVD is applied on the clustered utility matrix. Fig.4 shows how stochastic gradient descent converges. However, it is found that SVD does not converge on dimension. Fig.5 shows the RMSE linearly decreases with a increase of dimension (number of eigenvalue).

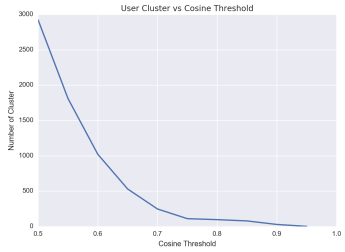


Figure 3: Cosine threshold

Figure 4: SVD converge

Figure 5: SVD rank and RMSE

It is also found that utility matrix become significantly less sparse after applying SVD. The left figure of Fig.6 is the utility matrix after clustering users; while the right one is the SVD utility matrix. It can be found that, after applying SVD, utility matrix become less sparse.

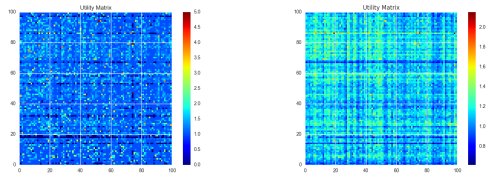


Figure 6: Clustered utility matrix (left) and utility matrix after SVD

Based on data we have, user country, region, city, is-mobile and is-package (the users booked the hotel with a flight) are regarded as the features of user profiles in this case. But we would like to detect if all these attributes show obvious users preference.

From the Fig.7, it is easy to see that users in different cities are inclined to choose the range of small hotel cluster ID, which means it has obvious relation with users preference. It is same with the attribute that the user used mobiles to book or not because the figures show the same distribution. Nevertheless, country, region and package factors show the different density for users in different factor values. For example, package users prefer to choose hotel cluster ID 65, while not-package users would like to book ID 91 hotel cluster.

Hybrid model results in prediction with 53.6% accuracy on testing data-4% improvement on content-base model. This result is consistent with our hypothesis: both user preference and hotel popularity are vital in recommendation system. In Kaggle, the benchmark content-based model (Data Leak method) has 49.8% accuracy.

Table 2: Accuracy Comparison

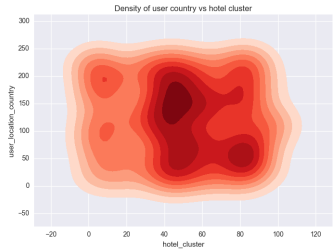
Accuracy	Hybrid Model	Content-based
Train	61.29%	61.29%
Test	53.62%	51.03%

IV. Future Work

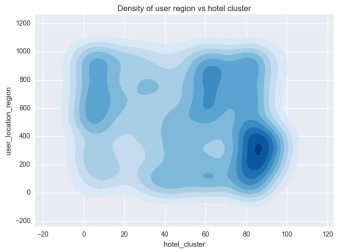
Our hybrid model can be further improved in these two aspects:

1. Larger dataset will be applied in this model so density-based clutesring method should be used instead of hierarchical clustering.

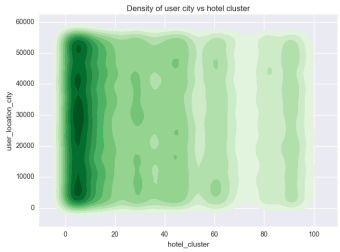
2. More features such as hotel country and hotel market might be included to test their impacts in prediction.



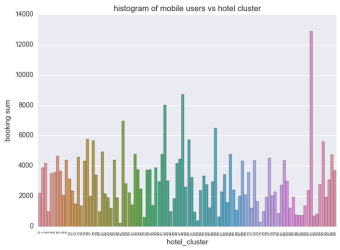
(a) Density of users country



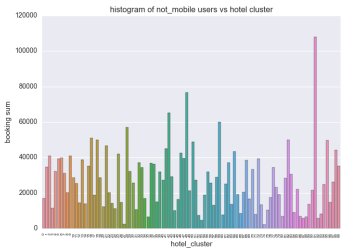
(b) Density of users region



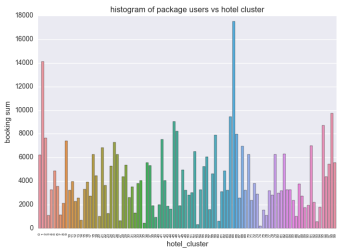
(c) Density of users city



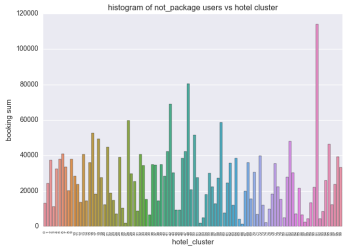
(d) Density of mobile users



(e) Density of not-mobile users



(f) Density of package users



(g) Density of not-package users

Figure 7: User Profile Detection (package user is the user booked the hotel with the flight)

V. Reference

[1] Saga R, Hayashi Y, Tsuji H. Hotel recommender system based on user's preference transition[C]//Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on. IEEE, 2008: 2437-2442.

[2] Yuning X, Li xiao G. Personalized Intelligent Hotel Recommendation System for Online Reservation–A Perspective of Product and User Characteristics[C]//Management and Service Science (MASS), 2010 International Conference on. IEEE, 2010: 1-5.

[3] Lops P. and Gemmis M., Content-based Recommender Systems: State of the Art and Trends, Springer Science+Business Media, 2011

[4] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering[C]//Proceedings of the 24th international conference on Machine learning. ACM, 2007: 791-798.

[5] Meng C, Cheng Y, Jiechao C, et al. A Method to Solve Cold-Start Problem in Recommendation System based on Social Network Sub-community and Ontology Decision Model[C]//3rd International Conference on Multimedia Technology (ICMT-13). Atlantis Press, 2013.