

Recommendation Algorithms for User First Booking on Airbnb

Zhao Wang
Northeastern University
Seattle, WA
wang.zhao2@husky.neu.edu

Zerui Ma
Northeastern University
Seattle, WA
zeruima1989@gmail.com

Heng Xu
Northeastern University
Seattle, WA
xu.he@husky.neu.edu

ABSTRACT

Airbnb has become more and more popular under the trend of sharing economy. In order to attract new users to place their first booking on Airbnb and offer a more personalized experience, we aim to build a recommendation system by using collaborative filtering approach, to predict which country these new users will make their first booking. Also the hosting data for various countries will be analyzed to provide better insight around booking area.

Keywords

Learning Algorithms; Recommendation Systems; Collaborative Filtering; Cluster Models

1. INTRODUCTION

Data mining is the process of discovering interesting patterns from massive amounts of data. As a knowledge discovery process, it typically involves data cleaning, data integration, data selection, data transformation, pattern discovery, pattern evaluation, and knowledge presentation[1]. The science of learning plays a key role in the fields of data mining, statistics and artificial intelligence, intersecting with areas of engineering and other disciplines[2]. In a typical scenario, a quantitative or categorical outcome measurement is predicted based on a set of features.

Recommendation algorithms are a kind of learning algorithms which are widely used on e-commerce web sites. they use input about a customer's interests to generate a list of recommended items. Most recommendation algorithms are designed for finding similar customers, where they aggregate items from the similar customers, eliminates items the user has already purchased or rated, and recommends the remaining items to the user. The popular versions of these algorithms are collaborative filtering and cluster models. In the collaborative filtering, the similarity of customers is measured by the cosine of the angle between the two vectors which represent users' interests[3]. Using this algorithm to generate recommendations is computationally ex-

pensive, but it can be released by dimensionally reduction techniques[4]. To find the similar customers to the user, cluster models divide the customer base into many segments and treat the task as a classification problem[5]. Some algorithms classify users into multiple segments and describe the strength of each relationship[6]. Besides grouping the user to the similar customers, other algorithms such as search-based methods and item-to-item collaborative filtering focus on finding similar items[7]. Search- or content-based methods treat the recommendations problem as a search for related items[8].

User experience is now a critical factor to keep users and attract new users among web applications. That is the reason Airbnb wants to provide personalized and unique experience for its new users, thus Airbnb need an effective recommendation system to recommend a country for first-time booking.

However, the main challenge here is that Airbnb doesn't have the travel history or other type of the traveling data of new users, the only data available here is basic feature such as age, gender, session log etc., basically like a white paper to a recommendation system. While a typical recommendation system might make recommendation based on a few strongly related features, the system designed here need to focus on correctly classify similar users first, then trying to make recommendation with some relatively strong features. And that is why we choose collaborative filtering as our first-step approach.

2. DATASET DESCRIPTION

In this section, the data sets which are applied for the recommendation algorithms are introduced. The data is consisted of two parts: the first one is the list of users' first booking destinations as well as their personal information and web session records; the other is the aggregated public host information dataset which is sourced from the Airbnb site. By analyzing the statistics of the source data, we can have an overview of the entire data and decide how to do the process and apply the data for the recommendation algorithms.

2.1 User data set

In the list of user first booking, each user is specified by a unique string id and each record contains multiple types of properties for that user, including account created time, age, sign up device, etc. There exists missing values in each fields, which should be addressed before training recommendation model. The users whose destination countries are

unavailable (indicated as NDF) are also need to be filtered out since we are going to predict the destination countries for those users. The details of user properties are described as Table.1.

Table 1: User Data Statistics

field	type	description
id	string	unique for each user
date_account_created	date	
timestamp_first_active	timestamp	
date_first_booking	date	
gender	categorical	FEMALE, MALE, OTHER, unknown
age	numerical	1 to 150
signup_method	categorical	basic, facebook, google
signup_flow	numerical	0 to 25
language	categorical	en, zh, fr, es, ko, de, etc.
affiliate_channel	categorical	api, content, direct, etc
affiliate_provider	categorical	bing, facebook, google, etc
first_affiliate_tracked	categorical	linked, local ops, product, etc.
signup_app	categorical	Android, iOS, Moweb, Web
first_device_type	categorical	Android Phone, iPad, iPhone, Mac Desktop, etc.
first_browser	categorical	Chrome, Safari, Firefox, etc.

After dropping the records without specific destinations, the numbers of the users for each country are shown as Table.2 described. The instances with specific destinations are the data we should focus on to build the predication model for new users. From the Table.2 we can conclude that most users choose U.S. as their first booking destination since Airbnb is a company in U.S. At sometime, there is a lot of users traveling to the countries which are listed in those specific countries (indicated as other).

Table 2: Users First Booking Destination

Destination Country	Population
AU	537
CA	1425
DE	1059
ES	2243
FR	5013
GB	2318
IT	2827
NL	757
PT	217
US	62263
other	10075

The Figure.1 describe the age distribution of the users in the list.

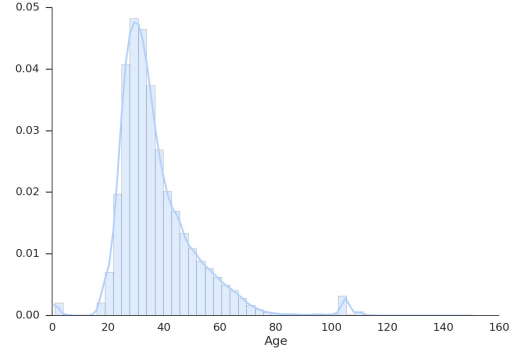


Figure 1: Age Distribution for Users on Airbnb

2.2 Hosts data set

The other data set is about detailed information of host listings among 16 countries available on Airbnb, although it covers almost all the countries in new user data set, but PT(Portugal) listing data is missing here.

This listing data set contains most of the public information available on Airbnb, and lots of features such as listing price, listing rating score, host registration information and neighbourhood etc., may help recommendation system identify better listings among target country for new users. However, due to the limitation of new user data, the recommendation on this part is most likely to be based on features weighted by common sense: highest rated, best available price etc..

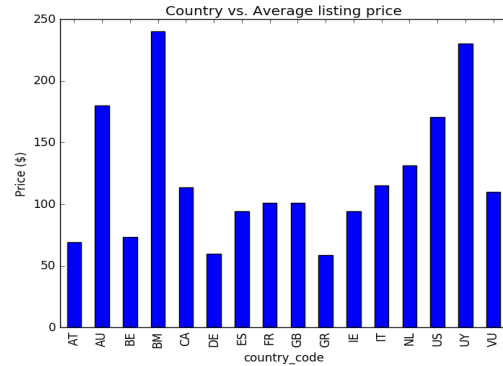


Figure 2: Average price distribution among countries

Since listing price is showed in dollar, so the average listing price distribution is influenced by currency exchange rate in some extent, but besides that fact, the average listing price definitely reflects the popularity of these destination countries. BM(Bermuda) is a typical high-end vacation hot spot, but it's surprised to see UY(Uruguay) has such a high average price, maybe because the retrived UY listings is very limited. Other than these two countries, AU and US are the most expensive countries to go on Airbnb. It's interesting to see AU has higher average listing price over US, a possible explanation for this could be the number of listings in US is much greater than AU's, thus competetion over the host has lowered the average listing price in US.

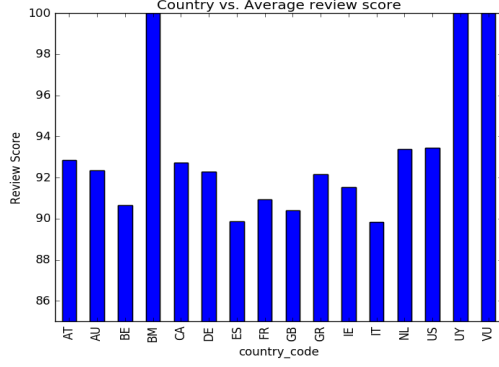


Figure 3: Average rating score distribution among countries

When comes to average review scores, most of countries get over score of 90. BM, UY and VU receives average score of 100, but since the number of listing of these 3 countries in this data set is very limited, it probably has large bias here. Besides these 3 countries, US and NL is probably best reviewed place to go on Airbnb, IT and ES is the worst reviewed, but still has average score of over 89.

3. PROPOSED ALGORITHMIC APPROACH

In this section, we introduce the learning algorithms we have used to implement the recommendation and evaluate their performance. At first, some basic classifiers are used and compared to find the optimal one which could do the predictions with the lowest error rate. Based on that some ensemble technicals such as boosting algorithm, random forest and feature selection are involved to improve the performance. Meanwhile we also use the collaborative filtering to build the recommendation model, in which the training dataset is clustered into k clusters and each testing instance is label by its closest cluster. A 3-layer ensemble classifier framework is also introduced to achieve performance improvements.

3.1 Data preprocess

The original properties for each users involve different data types, includes date and timestamp, so we need to convert all the features into numerical or categorical. As described in Table 3, we compute the lag days between *date_first_booking* and *date_account_created* and divide the result into four categories [$=0$, <0 , >0 , NA]. Some operation is also taken on *date_first_booking* and *timestamp_first_active*. For the age attribute, the missing values are replaced by the conditional mean and all the values are scaled by mean and standard deviation. Since the values in *signup_flow* are integers from 0 to 25, this attribute is considered as categorical when training the model. The training and testing set occupy 80% and 20% of entire data set.

3.2 Basic and Ensemble Classifiers

Initially we use *WEKA* package to implement the basic classifiers including decision tree, naive Bayes and support vector machine (SVM). The comparison of error rates shows that C4.5 decision tree has the best performance among these classifiers, whose error rate is much lower than other

Table 3: Data Preprocess

Processed Attribute	Description
Lag between <i>date_first_booking</i> and <i>date_account_created</i>	Divided into 4 categories $=0$, <0 , >0 , NA.
Lag between <i>date_first_booking</i> and <i>timestamp_first_active</i>	Divided into 3 categories $=0$, >0 , NA
Age	Replace the missing values with the conditional mean; scale the age with mean and standard deviation
<i>signup_flow</i>	Categorical Attribute
Others	Original value

classifiers. To verify the results we also implemented the same classifiers in *Scikit Learn* and obtain the error rates based on the training set size as Figure.4. In the basic classifier, Naïve Bayes starts at an error rate of 49% and drops drastically when instance number reaches about 8K. It achieves 70% accuracy at 50K and have not changed as the size of dataset increases. For Support Vector Machine and Logistic Regression, the error rates remain at around 29% and have not changed as the number of instances increases. Decision Tree starts at an accuracy rate of 70% and gradually decreases as the number of instances increases, and finally reaches around 74% accuracy when the number of instances is 70K.

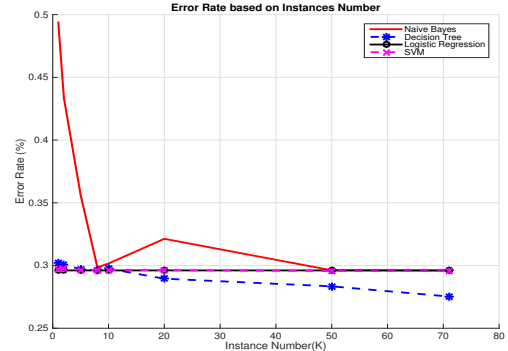


Figure 4: Error Rates for Basic Classifier

To improve the performance of the basic classifier, random forests and gradient boosting with decision tree are introduced and have shown improvements in accuracy as Figure.5 indicated. The baseline is decision tree which starts at an error rate of 39% and decreases as the training instances increase. It reaches an error rate of 33% when the data set is at 70K. Compared to decision tree, Random Forests get about 5% lower on error rate, starting at 30% and declining to 27.5%. Gradient Boosting with Decision Tree shown great performance on the training data, which is initially at 24.5% and decreases dramatically within the first 5K. Its error rate can achieve 22.5%, which is much better than Random Forest and any other classifiers.

In order to enhance the accuracy further, we also performed feature selection within PCA implemented by *Scikit Learn*. By using feature selection, we hope to reduce the less relevant information and get higher accuracy. Among the current 11 features, we use PCA to select 8 features be-

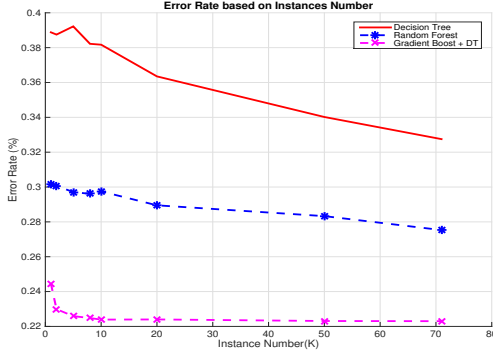


Figure 5: Error Rates for Ensemble Classifier

fore training the mode and get the results shown in Figure.6. By comparing to the performance of the original classifiers without PCA, the feature selection increase the error rate using PCA, which means some important information have been dropped. The only exception is adaboosting within decision tree, in which the feature selection cancelled the overfitting.

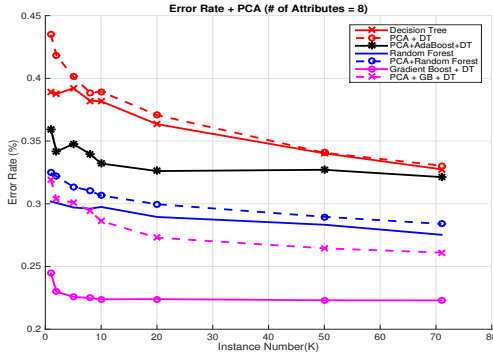


Figure 6: PCA with Classifiers

3.3 Collaborative Filtering

To find the optimal recommendation algorithm with the lowest error rate, collaborative filtering is also implemented. In this approach, we consider the *destination_country* as an attribute and use k means to aggregate the similar users based on all the properties. After clustering each cluster is labeled with the majority destination among all its members. To evaluate the model, each test case is predicted as the label of its closest cluster.

The distance between each pair of users is computed by Euclidean Distance and the Within Group Sum of Square (WGSS) of distance is used as a measurement. To find the optimal cluster number should be, we define a threshold ϵ . When the change rate of WGSS is smaller than the threshold, the best value of k is achieved. As Table.4 shown, as the threshold get smaller, the more clusters need to be aggregated and the lower rate can be achieved. When ϵ is 0.001%, the error rate can achieve 21.37% which is better than the ensemble classifiers and basic classifiers. The trend of WGSS with the cluster is indicated in Figure.7, where the

WGSS falls rapidly until at the cluster number is about 400, the best value of k is reached.

Table 4: Clustering Parameters

Percentage Change of WGSS	Cluster Num	WGSS	Error Rate
< 0.3%	10	114013	0.3036
< 0.1%	17	106895	0.2978
< 0.03%	67	84909	0.2700
< 0.003%	163	73112	0.2330
< 0.001%	424	55412	0.2137

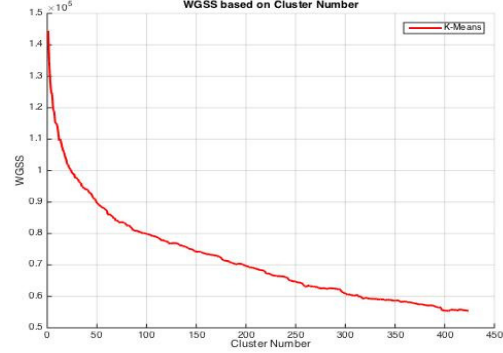


Figure 7: WGSS based on Clusters

4. CONCLUSION

5. REFERENCES

- [1] Han, J., Kamber, M., and Pei, J. , "Data Mining: Concepts and Techniques", 3rd Edition, 2011.
- [2] Hastie T., Tibshirani R., and Friedman, J. , "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", 2nd Edition, 2009.
- [3] B.M. Sarwarm et al., "Analysis of Recommendation Algorithms for E-Commerce", ACM Conf. Electronic Commerce, ACM Press, 2000, pp.158-167.
- [4] K. Goldberg et al., "Eigentaste: A Constant Time Collaborative Filtering Algorithm", Information Retrieval J., vol. 4, no. 2, July 2001, pp. 133-151.
- [5] P.S. Bradley, U.M. Fayyad, and C. Reina, "Scaling Clustering Algorithms to Large Databases", Knowledge Discovery and Data Mining, Kluwer Academic, 1998, pp. 9-15.
- [6] L. Ungar and D. Foster, "Clustering Methods for Collaborative Filtering", Proc. Workshop on Recommendation Systems, AAAI Press, 1998.
- [7] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering", Internet Computing 7:1, 2003, pp. 76-80.
- [8] Leskovec, J., Rajaraman, A., and Ullman, "Mining of Massive Datasets", 2nd Edition, 2004, pp. 307-341.