Statistical Learning-Classification

Project Title: Expedia Hotel Recommendation

Project Number: 12

Group Members:

Surname, First Name	Student ID	STAT 441	STAT 841	Your Dept. e.g. STAT, ECE, CS
Alexander Wan	20520127	☑	☐	STAT
Thongchai Kevin Hu	20472052	☑	☐	ACTSC & STAT
Tahmid Mehdi	20469834	☑	☐	CM & STAT
Hanxin Zhang	20437677	☑	☐	CS & STAT

Your project falls into one of the following categories. Check the boxes which describe your project the best.

1. ☑ **Kaggle project.** Our project is a Kaggle competition.
   - This competition is active ☐ inactive ☑.
   - Our rank in the competition is *Expedia Hotel Recommendation*.
   - The best Kaggle score in this competition is ... ..., and our score is ....
2. ☐ **New algorithm.** We developed a new algorithm and demonstrated (theoretically and/or empirically) why our technique is better (or worse) than other algorithms.
3. ☐ **Application.** We applied known algorithm(s) to some domain.
   - ☐ We applied the algorithm(s) to our own research problem.
   - ☐ We tried to reproduce results of someone else's paper.
   - ☐ We used an existing implementation of the algorithm(s).
   - ☐ We implemented the algorithm(s) ourself.

Our most significant contributions are (List at most three):

(a) .

(b) .

(c) .

List the name of programming languages, tools, packages, and software that you have used in this project:
Python(numpy, sklearn, Rodeo), R(xgboost, caret), LaTeX, Bash

1

Background Introduction

Many companies have internal search engines on their websites where visitors can search for products or services. Popular keywords and search filters can provide insights into customer behaviour. Many companies want to harness the power of machine learning to suggest products to users based on their searches. One of these companies is Expedia. Their website allows users to search for hotels and plan vacations. They hosted a Kaggle competition, "Expedia Hotel Recommendation" in April 2016 so that competitors can recommend hotels to users based on their search data.

1.1 Data

The training set includes search data (approximately 37 million search events) from 2013 and 2014. The features are search parameters such as dates, locations and whether the search resulted in a hotel being booked. Expedia grouped hotels into 100 hotel clusters based on their similarity. The testing set consists of approximately 2.5 million search events from 2015 which resulted in hotel bookings. They have the same features as the training set. The goal is to predict which hotel cluster the user booked for each search event.

1.2 Evaluation

For each search event, up to five hotel clusters can be predicted. Submissions are scored using the Mean Average Precision at 5 (MAP@5). The formula is

$$MAP@5 = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{k=1}^{5} P(k)$$

where U is the testing set and P(k) is the precision at the k^{th} prediction. It calculates the mean precision of each search event and then averages them. For a given search event, if the correct class appears in the k^{th} position in the list of five predictions, the average precision for that event is $\frac{1}{k}$. If it is not in the list, the average precision is 0. For example, if the correct hotel cluster is 3 and the list of predicted clusters is "10

3 52 98 75", then it would receive an average precision of $\frac{1}{2}$ as the correct cluster is ranked second. The MAP@5 score from random guessing is 0.022.

2

Methods

2.1 Initial Approaches

2.1.1 Data Preprocessing

Since the training set is very large (4 GB), incomplete entries that contained missing values were filtered out. Furthermore, search events that did not lead to hotel bookings were removed because they may not reflect the search patterns of users who actually book hotels. Moreover, time stamp features such as check-in and check-out dates were split into different attributes for years, months and days. Lastly, highly correlated variables were removed. For example, a user's continent is correlated with their country so it is redundant. In the testing set, some entries were missing a destination distance so mean values were imputed.

2.1.2 Algorithms

Initially, we tried many popular machine learning algorithms to predict hotel clusters for the search events in the testing set. The following methods were used:

- Linear Discriminant Analysis
- k-Nearest Neighbours with Hamming distance as the distance metric
- Random Forests with various numbers of trees
- Naive Bayes
- XGBoost
- AdaBoost with various numbers of trees

However, none of these methods produced a MAP@5 score above 0.105. These methods were unsuccessful due to the fact that there was a large amount of classes in the problem. There are 100 clusters in this problem and basic machine learning methods are not usually accurate for datasets with large amounts of confusable classes (Gupta et al., 2014) . Another issue is the processing of the training and testing data. Many

search events were filtered out of the training set. Furthermore, although highly correlated variables were removed, the data is high-dimensional because of the manifold of categorical features.

2.2 New Approach for Data Engineering

2.2.1 Frequency Tables

Since the aforementioned machine learning algorithms failed to successfully classify most of the search events, it is evident that additional feature engineering techniques are required to pre-process the data. Thus, frequency tables were created for each categorical feature to utilize rule-based and count-based learning. For a given feature, the rows of its table correspond to values of the feature while each column corresponds to a hotel cluster. Each entry represents the number of events, in the training set, with a particular feature value and hotel cluster.

User ID	Cluster 0	Cluster 1	...	Cluster 99
1	1	70	...	5
...	...	...	...	...
n	4	1	...	10

For example, based on this table, user 1 has looked at hotel cluster 0 once, cluster 1 70 times and cluster 99 five times. The model assumes that the hotel cluster follows a conditional distribution such that

$$P(Y = j|X = i) = \frac{p_{ij}}{\sum_{k=0}^{99} p_{ik}}$$

where Y is a hotel cluster, X is the value of a particular feature and p_{ij} is the number of search events with feature value i and hotel cluster j.

2.2.2 Ranking Algorithm

The following algorithm predicts five classes for a search event.

1. Extract the selected features from the event.
2. For each feature, retrieve the n most frequent clusters from the row corresponding to the feature value of the event.
3. Pool the most frequent clusters from each table into a list. Clusters with high frequencies in multiple tables will be recorded multiple times in the list.
4. Rank the clusters based on how many times they occur in the list and select the top five.

Frequency tables for some features can hurt the predictive accuracy so only features that had a significant influence on the predictions were included. To determine which features to remove, a random forest was applied to the training set and the Mean Decrease in Accuracy (MDA) of each categorical feature was computed. It is calculated by generating permutations of the feature values and computing the average decrease in accuracy. The most influential features (with MDA ≥ 0.05) were determined to be the user id, the region of the user, the search destination id, the country of the hotel, the market of the hotel, and (to a lesser extent) the day of the week that a user checked-in and out of a hotel. The ranking algorithm was performed using the tables of these features with n=5 and 10 classes retrieved from each table. It was also run without the date features. The results are summarized in Table 1.

Features Included	n	MAP@5
user_id, user_location_region, srch_destination_id, hotel_market, hotel_country	5	0.22066
user_id, user_location_region, srch_destination_id, hotel_market, hotel_country	10	0.20653
user_id, user_location_region, srch_destination_id, hotel_market, hotel_country, srch_ci_week, srch_co_week	5	0.17164
user_id, user_location_region, srch_destination_id, hotel_market, hotel_country, srch_ci_week, srch_co_week	10	0.14837

Table 1: MAP@5 of Ranking Algorithm with different parameter settings

3

Discussion and Conclusions

The Ranking Algorithm with frequency tables outperforms the other methods that were tested. There are several advantages to using the Ranking Algorithm. The frequency tables are more compact than the raw training data and it is computationally efficient due to the low number of features. Other frequency based models like Term Frequency-Inverse Document Frequency (Ramos, 2013) and Bag of Features (O'Hara and Draper, 2011), have high predictive accuracy when the testing set contains information that is included in the training set. Similarly, the Ranking Algorithm was able to correctly predict the hotel clusters for active Expedia users because they often book the same hotels.

The Ranking Algorithm also has some drawbacks. It weighs all features equally but

some may be more important than others. Instead, all features have a democratic vote on which classes should be predicted. Moreover, Frequency based algorithms have difficulties identifying similarities between feature values (Ramos, 2013). The Ranking Algorithm faces the same disadvantage because it cannot measure the similarity between values of a categorical variable.

The algorithm had a higher MAP@5 score when it did not incorporate date features because the check-in and check-out days were not associated with the hotel cluster that a user booked. Furthermore, retrieving five classes from each table was more effective than retrieving ten because the five most frequent classes for a particular feature value usually had much higher counts than the following five.

Acknowledgements

We would like to thank Dr. Ali Ghodsi for offering suggestions to improve our feature engineering process. Moreover, we would like to thank Kaggle user ZFTurbo as his solution inspired the idea to use frequency tables ("Leakage Solution", 2016). Lastly, we would like to thank Amazon for allowing us to use their EC2 server to run the Ranking Algorithm.

References

Gupta, Maya R., Samy Bengio, and Jason Weston. "Training Highly Multiclass Classifiers." *Journal of Machine Learning Research,* vol. 15, 2014, pp. 1461-1492.

"Leakage Solution." Expedia Hotel Recommendation. Kaggle, 2016. kaggle.com/zfturbo/expedia-hotel-recommendations/leakage-solution/code.

O'Hara, Stephen, and Bruce A. Draper. "Introduction to the Bag of Features Paradigm for Image Classification and Retrieval." *Arxiv preprint arXiv:1101.3354*, 2011.

Ramos, Juan. "Using TF-IDF to Determine Word Relevance in Document Queries." *Proceedings of the first instructional conference on machine learning*, 2003.