Hotel Hybrid Recommender System

Matthew Zhang

February 2021

Introduction

- Develops a hybrid system for hotel recommendations; representing content-based and collaborative filtering methods for better recommendation.
- Model considers user and item features to predict for new users.
- The system additionally implements centroid based clustering and classification techniques to aggregate a successful hybrid.
- Scalability, sparsity within utility matrix and the cold start problem.

Data

Source:

Kaggle Competition on Expedia Hotels.

Each row represents a user’s search query.

Clicked or booked, # of adults, # of rooms, etc.

Details:

Dataset had predefined hotel (item) clusters based on price, ratings, etc.

37M rows, and 1M unique users.

Cleaning and manipulation: 3M rows and 100,000 unique users.

Interactions:

Setting a metric to proxy ratings. Receiving 1 if clicked and 5 if booked, otherwise unrated.

Helps create the utility matrix with user cluster as users and hotel cluster as items.

K-Means Clustering

- Regarding scalability, I clustered the users into 1,000 clusters.
- Most users have little booking history leading to a lots of sparsity; now, you get an average of interactions with each hotel
- What this means is reduced computational cost and efficiently running matrix factorization
- Finally, a compressed (clustered) utility matrix.

hotel_cluster	0	1	2	3	4
cluster					
0	0.074074	0.133333	0.037037	0.111111	0.333333
1	1.149392	0.045010	0.047945	0.054795	0.184682
2	0.033898	0.003390	0.010169	0.023729	0.000000
3	0.127273	0.072727	0.000000	0.027273	0.036364
4	0.776543	0.340000	0.277778	0.466667	0.244444

Intermediary: Decision Tree Classifier

- To address the cold start problem, I introduced an ontology model.
- Ontology theory states that user profile represents user behavior, to an extent.
- In a general case, explore the user profile: gender, age, occupation, etc. to predict user behavior.
- For the system, by inputting user profile data, I can predict the user cluster they will be in and create recommendations.
- Decision tree- my features are: country, region, is_mobile, is_package to classify the user into user cluster.

SVD and Baseline

- Used just the clustered utility matrix without user/item features.
- Tuned/final Surprise SVD model MAE of 0.38.
- SVD predicts ratings from 0-5 with an MAE of 0.38.
- Still collaborative filtering but can be used as a part of the system for old users because it performed so well.

Baseline (LightFM):

- Purely Collaborative Filtering and user-item features
- Clustered utility matrix
- AUC score of 0.36
- Hyper parameters tuned: epochs, loss, learning_rate, learning_schedule

Evaluating RMSE, MAE of algorithm SVD on 3 split(s).

	Fold 1	Fold 2	Fold 3	Mean	Std
RMSE (testset)	0.4532	0.4577	0.4608	0.4572	0.0031
MAE (testset)	0.3876	0.3884	0.3883	0.3881	0.0003
Fit time	3.10	3.06	3.08	3.08	0.02
Test time	0.16	0.26	0.16	0.19	0.05

Hybrid Model: LightFM

- Hybrid representation that learns embeddings of user and item features capturing user preference; optimized through SGD.
- Reformat entire dataset into LightFM dataset reader class.
- Create user-item features in the format of tuples (id, [list of features])
- LightFM's train-test split

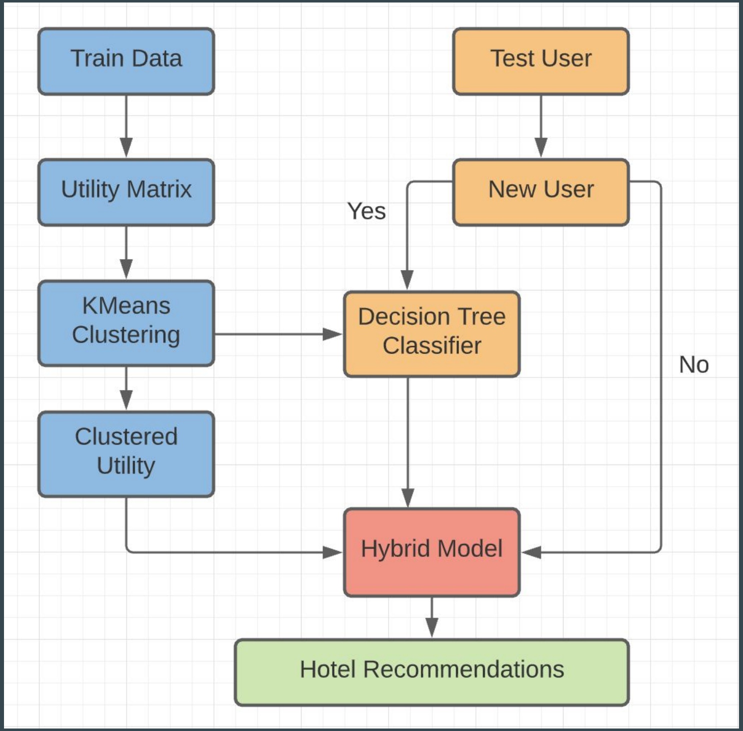
```
AUC: 0.65.
CPU times: user 77.3 ms, sys: 2.04 ms, total: 79.3 ms
Wall time: 79.1 ms
```

LightFM approach for predictions on new users:

- A function that takes in user features and converts it to Scipy sparse matrix to feed into LightFM.

```
Out[93]: array([-27.64719 , -27.84846 , -27.789427, -27.955019, -27.692997,
               -27.59371 , -27.75037 , -27.845528, -27.578558, -27.689655,
               -27.783152, -27.722939, -27.871927, -28.050377, -27.736067,
               -27.613546, -27.766125, -27.792068, -27.520163, -27.70413 ,
               -27.571182, -28.034004, -27.569946, -27.73175 , -27.705309,
               -27.728773, -27.723295, -27.623236, -28.084944, -27.603584,
               -27.952738, -27.630623, -27.669527, -27.707672, -27.858715,
               -27.979872, -27.903032, -27.794706, -27.647255, -27.590855,
               -27.747269, -27.743156, -27.842402, -28.048119, -27.907158,
               -27.64153 , -27.821224, -27.531546, -28.002157, -27.823479,
               -27.84626 , -27.75831 , -27.932985, -27.65263 , -27.894604,
               -27.825382, -28.114437, -27.66928 , -27.748356, -27.92019 ,
               -27.651892, -27.898592, -27.934505, -27.905355, -27.660353,
               -27.859673, -28.104322, -27.846777, -28.120453, -27.650785,
               -27.775906, -27.65272 , -27.64584 , -27.613794, -27.724989,
               -27.832443, -27.76451 , -27.992283, -27.765013, -28.061031,
               -27.698301, -28.019169, -27.76381 , -27.737213, -27.813599,
               -28.070532, -27.825714, -28.039589, -27.937046, -27.735723,
               -27.844881, -27.92392 , -28.052881, -27.86714 , -27.881466,
               -27.88933 , -27.885582, -27.982077, -27.588787, -27.708693],
              dtype=float32)
```

Conclusions



- I have an aggregate of many algorithms that make up my recommender system.
- User preference and Item features are vital in recommender systems.
- Hybrid model successfully compensates the shortcomings of individual CF and Content
- Hybrid model produces an AUC score of 0.65
- Flow chart of the entire system.

Future Work

- Consider Hierarchical Clustering where each point is a cluster to capture uniqueness of users as opposed to arbitrary cluster points.
- PCA of additional dataset with arbitrary numbers representing reviews.
- Combine user preference with hotel destination input. Individual user may not go to a high rating hotel within a cluster, which produces a great error if recommending the top hotel by purely clustered utility matrix.
- Hybrid Model through Neural Networks where layers are represented by CF and Content-Based methods

Thank You

GitHub Repository	• https://github.com/mzcode98/hybrid-recommender-system
Email	• mattzhang989@gmail.com
Acknowledgements	• Instructor Yish Lim, Classmates, Flatiron School
Reference	• Jing Wang, Jiajun Sin, Zhendong Lin: Hotel Recommendations Based on Hybrid Model • Xinxing Jiang, Yao Xiao, and Shunji Li: Personalized Expedia Hotel Searches