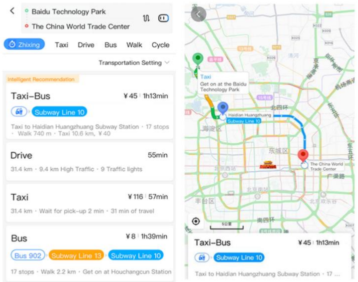
Context-Aware Transportation Mode Recommendation for Navi Apps

Meixin Zhu, Jingyun Hu

Motivation

Classical navigation apps offer mode choices based on recent, frequent and labeled destinations. For example, if given an already recorded destination, the app will recommend the same mode clicked by the user last time. However, if the origin and destination pair of the trip are new, the recommended mode may not be appropriate.



(App Interface. Our Goal: Predict Which Choice Will Be Clicked.)

Problem Definition

Given a user ***u***, an origin-destination pair *o* and ***d,*** and the situational context, we want to recommend the most proper transport mode ***m ϵ M*** for user ***u*** to travel between ***o*** and ***d,*** considering user's preferences (e.g., costs, time) reveled in their historical trip data and trip characteristics (e.g., purpose, distance).

Data

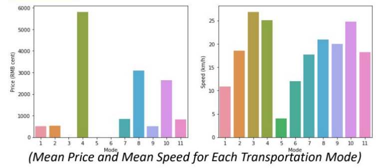
Source: Baidu Map data of Beijing for KDD Cup 2019.

- **500 k** queries in total. Oct 1, 2018 to Dec 7, 2018.
- Training set: Oct 1 to Oct 30.
- Testing set: Dec 1 to Dec 7.

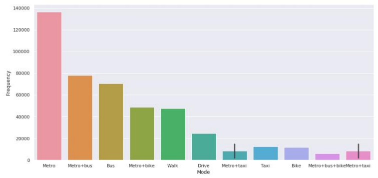
sid	pid	req_time	o	d	plan_time	click_time	click_mode	plans
973273	100000	2018-10-24 20:58:45	116.37,40.	116.39,39.	2018-10-24 20:58:45	2018-10-24 20:59:2	10	distance price eta transport_mode 24205 1800 3807 10 17624 2306 3 17624 6400 2606 4 24332 500 4158 9 21859 700 4612 7 17374 500 5392 1
1203367	100000	2018-11-22 08:50:46	116.45,39.	116.39,39.	2018-11-22 08:50:46	2018-11-22 08:56:5	2	distance price eta transport_mode 11184 400 2629 2 13978 2596 2 9617 2674 3 9617 4900 2914 4
1695455	100000	2018-10-23 17:36:19	116.36,40.	116.37,40.	2018-10-23 17:36:19			distance price eta transport_mode 1735 524 6 2046 428 3 2046 1300 548 4 1445 1255 5 1892 200 1420 1 2491 200 1694 1

(Data Example after Preprocess and Organization)

Data Exploration



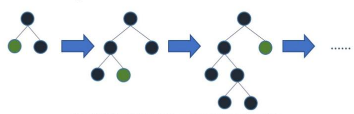
(Mean Price and Mean Speed for Each Transportation Mode)



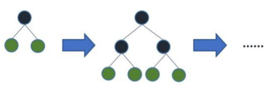
(Distribution of Mode Choice)

Main Method--Light GBM

Light GBM is a gradient boosting framework that uses tree based learning algorithm. It grows tree **vertically in a leaf-wise way** by choosing the leaf with maximum loss reduction to grow.



(Leaf-Wise Tree Growth of Light GBM)



(Level-Wise Tree Growth)

Key Point in This Method: Feature Engineering

304 features in the final model.

- Request time and plan time.(weekday, hour)
- Longitudes and latitudes of ***od*** pair.
- Distance, price, time, **price*time** and speed for candidate modes.
- Descriptive statistics of distance, price, time, speed for all the modes in each plan list.
- Features of distances to bus/subway stations, train stations and airports for each ***o*** and ***d***.
- Number of nearby POIs for each ***o*** and ***d*** (33 POI categories, including auto service, transit port, tourist area, etc.) .

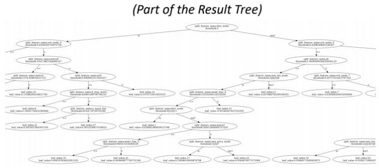
Evaluation Metric

The **F1 score** is used as the evaluation metric for transportation mode prediction.

$$F_1 = \frac{2\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Results

(Part of the Result Tree)



Hyper Parameters:

num_leaves = 40 max_depth=8
learning_rate = 0.1 subsample = 0.8
colsample_bytree = 0.8 min-_child_samples = 60

Final train F1 score：0.7
Final validation F1 score：0.6932
Final testing F1 score：0.6951

(Validation Results)

Transportation Mode	Sample ratio	F1 score	Precision	Recall
No click	0.0874	0.3552	**0.9453**	0.2187
Bus	0.1446	0.6804	0.6372	0.7299
Metro	**0.3133**	**0.9019**	0.8543	**0.9551**
Drive	0.0446	0.1424	0.3997	0.0867
Taxi	**0.0245**	**0.0829**	**0.1836**	**0.0535**
Walk	0.0976	0.8452	0.7859	0.9143
Bike	0.0199	0.2187	0.2529	0.1927
Metro+bus	0.1779	0.7883	0.7112	0.8840
Bus+taxi	0.0046	0.3288	0.2568	0.4567
Metro+bike	0.0499	0.5148	0.5803	0.4626
Metro+taxi	0.0285	0.5424	0.4643	0.6521
Metro+bus+bike	0.0072	0.4187	0.3731	0.4771
Validation F1 score for all modes: 0.693168348				

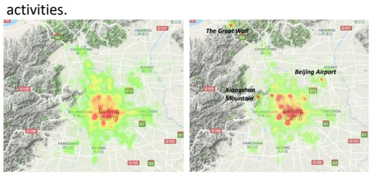
Model	Weighted F1 score
Lightgbm with smote oversample	**0.6951**
Lightgbm	0.6920
Lightgbm with backward feature selection	0.6912
Lightgbm with pid	0.6905
Lightgbm with random oversample	0.6884
Lightgbm learning to rank	0.6883
Xgboost	0.6877
Random forest	0.6851
Auto ML	0.6836
Catboost	0.6835
Xgboost learning to rank	0.6661
Lightgbm train with single pid (if not have enough data, grouped with others)	0.6645
Multinomial logit model	0.4971
Shallow neural network (6 layers)	0.0178

Discussion

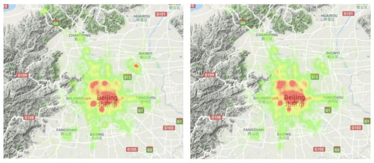
- Currently, the highest score in the leaderboard is 0.7043, still far from perfect.
- Not sure the ceiling of the F1 score for this problem. Maybe human's behavior is too hard to predict, and the F1 score is impossible to be larger than 80%.
- Oversampling gives a stable boosting for the model performance.
- Individual heterogeneity does exist.
- Things tried but not having significant effects (or even making the performance worse):
  - Do PCA before learning
  - Expand the plan list and do binary classification
  - Adding class weights for unbalanced data
  - Incorporate historical mode choice probability as features
  - Scale the data
  - Normalize the data
  - Scale the time, price, distance to 0~1 within the candidate plans
  - SVM (too slow)
  - Train with individual's data. Turned out that some pids are just too hard to predict
  - Adding weather as features

Some Thoughts

App data can be used to learn and predict people's travel behavior. For example, to identify trends of users' activities.



Origin Points Destination Points
(Distribution of ***O*** and ***D*** in the Morning Peak Hours)



Origin Points Destination Points
(Distribution of ***O*** and ***D*** in the Evening Peak Hours)

Other things we could do:
Analyze activity pattern of different user groups.
The process of urban center change.