Model Manual

By Lyman, Flora

In this documentary, a brief introduction will be given about our work, including why we choose xgboost classifier, how to reshape datasets, process for feature engineering and other details.

1. Why xgboost classifier

Since trajectories are recorded at different time, it's not easy to apply models for time series such as Markov Chain and RNN to solve the problem.

Xgboost is a powerful machine learning method developed from random forest, which can find an excellent way to choose and split features, and in this case, time of trajectories can be considered as a feature to build the model.

Final positions of the trajectories are given in the data_train. It seems that regression can better make use of the information. However, the loss of regression is determined by MSE. If a position isn't within the city center while another one stays in the area, they can still indicate same loss in a regression model, which is not helpful for our target, predicting whether or not a device is in the city center.

2. How to reshape datasets

Since xgboost can only fed by records one by one, we need to reshape the dataset. Notice that simply merging trajectories associated to one device can loss much information about time. Thus, before merging, we should reshape the form of time recorded in the primary dataset .

Trajectory1 of device1	Time coefficient	Other features …
Trajectory2 of device1	Time coefficient	Other features …
Trajectory3 of device1	Time coefficient	Other features …

Device1	T1	…	T2	…	T3	…

3. Feature engineering

There are a lot of information behind the data worth digging deep. One of the most efficient way to explore the information is feature engineering. By constructing innovate factors, xgboost can better deal with dataset since these factors can indicate some key information. Here are the factors.

Possible position based on previous speed. We can calculate several possible

positions based on the past speeds. These estimated position can provide supports to predict the final position. There's the calculating process below.

Confidence factor based on time. If the time you start to move is close to the final time, it's more likely to adapt the estimated position.

Confidence factor based on position. When the estimated position is very close to the bond of city center , it's reasonable to doubt whether it's in the center.

4. Other details

We can see the scale of the records of 0 is larger than that of records of 1, we need to oversample records of 1 until they have the same scale.

There are many parameters in xgboost. To find the best parameters for the model, we build a dictionary and use GridsearchCV to look through it.

5. Motivation and outcome for each code piece

In many cases, each device has several records, but there are also some devices that just have one record. We put them into **_1trace.csv** file, then we train this dataset with an independent xgboost model, which are much simpler than the model we use to fit **_count5.csv** dataset. The final result is a combination of predictions made by the 2 models.

Code	Motivation	Datasets needed	Datasets generated
eytrain	Feature engineering for data_train	Data_train.csv	Train.csv, Train_pre.csv Train_1trace.csv
eytest	Feature engineering for data_test	Data_test.csv	Test.csv, Test_pre.csv Test_1trace.csv
eymerge	Merge records associated to the same device into one records	Train.csv, Train_pre.csv Test.csv, Test_pre.csv	Train_count5.csv Test_count5.csv
eymodel	Xgboost model for fitting and predicting	Train_count5.csv Test_count5.csv	T5_os_estim.csv
eyresult	Fitting and predicting 1trace.csv dataset; merge it with t5_os_estim to get final result	T5_os_estim.csv Train_1trace.csv Test_1trace.csv	**T5_1trace_estim.csv**