Facebook V: Predicting Check Ins

Rongting Zhang

1 Abstract

The Kaggle competition is about select the three most possible check place ids from more than 100,000 candidates on a 10km by 10km artifical map for the future. The models trained and tested during the competitions by our team include weighted majority vote on various grids, various decision tree algorithms including xgboost and random forest trained with different features on various grids and variation of knn with grid splitting adopted from the forum. Models included in the first submission are various xgboost and a varition of knn trained on grids.

2 Understanding Features

The only two features in the data sets that do not have immediate interpretation are the time and accuracy. After plotting the distribution of the time feature for several specific place ids, it is not hard to conclude that the time feature is in terms of minute as we see strong periodicity when taking modulo 1440 and 10080. After doing some research online on how location accuracy is estimated as well as testing on the data, it seems the most plausible explanation for accuracy is that this is the estimated location accuracy of the device used for check in, i.e. an accuracy of x is likely to mean that the device estimate the true location is within certain distance from the reported location with a fixed probability. However this estimate may not be accurate as well and could potentially different for different devices. Naive testing on the deviation does not suggest that lower accuracy would actually imply a good location accuracy which might due to the shape of the place etc, but for different place id it does show some difference in the distribution of accuracy, thus we could still benefit from accuracy even its interpretation is not immediately helpful to the estimation of spatial location.

3 Training and Validation

3.1 Splitting the Data into Local Grids

As the size of the training set is very large, directly training over the whole training set using common machine learning algorithms would not be very likely. Here we split the training set based on (x, y) into smaller grids, we training the similar models on the local grids and make prediction. For nearest neighbor related models we augmented the training by 6 percent in the four directions.

3.2 Selection of Validation Set

As the task is to predict for the future, so we split the training set according to the time into two sets. We select the time 655200 as the line of separation for validation. The validation score turns out to be quite consistent though a little bit different for different models.

3.3 Description of Models in Final Ensemble

There are mainly two types of models we used for our final ensemble. Nearest neighbor based models and xgboost models.

1. **KNN Model** The nearest neighbor models we trained is almost identical to the models on the forum. The only difference is that to deal with the periodic nature of the time variable like hours and day etc, we did an encoding of the time feature. For example for a periodic data with possible values $0, 1, 2, 3$ we would map $0, 1, 2, 3$ to the rows of the following the matrix A to achieve a periodic l_1 distance. The only reason to do the encoding is because the sklearn package allows user-defined distance by defining a python function however it is much slower than the built-in distances.

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}. \tag{1}$$

We also did a weighted log average of the knn probability with the place id based probability estimate using histogram with the time of day, day of week and binned accuracy. We trained the knn model on 20 by 40 grid while the histogram probability estimation are done on 10 by 20 grid.

2. **XGBoost Models** The xgboost models we trained consists of models trained with three feature sets.
   (a) x, y, accuracy, time, (time mod 1440), (time mod 10080)
   (b) x, y, accuracy, time of day, day of week, week of year, year
   (c) x, y, accuracy, time of day, day of wee, month of year, year, knn probability as mentioned above for potential place ids with appearance above certain level on the local grid.

   We trained the xgboost models on grids 100 by 200, 71 by 141, 90 by 223, 63 by 159 and with grid shifted by half the grid size in x, y and (x, y) direction.

3.4 Ensemble and Leader Board Scores

For each model, we recorded the probability predicted by the algorithm of the top 20 places. In the end, we did a weighted sum of the probability predicted by different models, the weights are selected empirically based on the location validation score and leader board score.
The leader score for several different models are listed in Table 1, Here xgboost (a), (b), (c) are the xgboost models

Model	CV	Public LB
nearest neighbor	0.546	0.5889
xgboost (a)	0.523	0.5675
xgboost (b)	0.536	0.5752
xgboost (c)	0.544	0.5849
ensemble	0.563	0.6053

Table 1: Score of different models.

introduced above trained on a single 100 by 200 grid. However we need to mention that average over shifted grids usually would have a gain in score of around 0.006 and scores for 71 by 141 or 63 by 159 grids would be about 0.004 higher.

4 Code Description

The code included in the models consists of model files in Python and validation and ensemble files in C++. To avoid duplication, only files on 63 by 159 grid for the xgboost models are included.