EECS 349 Project Status Report – Group 21

Qiping Zhang (qze7487) Han Wang (hwm4792) Ji Lin (jle2795)

1. Task Definition

Our task is to predict whether a user will download an app after clicking a mobile app advertisement, based on a vast collection of users' clicking record on app advertisement as our original training data, along with other attributes such as ip, os, device and etc.

Recent years have witnessed an inevitable phenomenon where more and more users simply click on the ad link of apps without downloading them, which results in considerable wasted ad investment from app service providers. Therefore, it is worthwhile for us to build a model so that such problem could be solved or its latency could be minimized to utmost extent. If the result of our model was with high accuracy, it would definitely help those companies advertising online cast less investment and meanwhile make more profit.

2. Data Set Description

We would acquire the original data from an ongoing featured competition on Kaggle, which is provided by TalkingData. The given dataset covers approximately 200 million clicks over 4 days, including 8 attributes in total: ip,app,device,os,channel,click_time,attributed_time (if the app is downloaded),is_attributed (whether the click results in a download, it is the target of prediction). Since this dataset is too large to be completely utilized for training, and the positive results take too small proportion in the whole set, we had to make a sampling on the raw data. We first selected the first 10,000,000 examples in the list, keeping all the 18717 positive examples with is_attributed = 1. Then, we iterated through all negative examples, randomly picking 1 from every 450 examples. This could ensure our selected negative examples are more scattered and separated, instead of gathering in a short period like an hour/minute. In this way, we have sampled approximately 38,000 pieces of data to perform as our training/testing set to be used. And this would not conspicuously affect the overall accuracy compared with the original data. To process the data, we first converted the click_time and attributed_time to the number of seconds starting from 2017-11-06 00:00:00 (slightly earlier than all the time attributes given), and then we partition the values of ip, click_time and attributed_time into several equal-sized intervals such that these "continuous" variables could be denoted more efficiently and representatively. Since the other attributes like os, device, channel have already been encoded into discrete integers, we need not make further operations on them and could use them directly. To distribute the data, currently we assigned 10,000 examples for training, 5,000 examples for validation, and the rest of them are used for testing. But this number is quite flexible, and we also switched our way of partitioning data to observe their different effect on the quality of model and testing accuracy.

3. Preliminary Results

After we got the smaller-size equally distributed sampling data from the raw data, we tried some ML techniques such as Decision Tree, Random Forest and REPTree to build model. As we mentioned in the proposal, we used 10-folds cross validation to test and evaluate. Among all the initial approaches we plan to use, the accuracy of Random Forest is 84.47% and the accuracy of REPTree in weka is 84.59%. As for Decision Tree, we tried to test with different training data

size from 500 to 10000. The accuracy went from 81.77% to 85.74% with the increase of training data size. Thus, decision tree got the best accuracy result currently.

4. Plans and Concerns

According to the current results we have obtained, we are going to improve our model from the following two perspectives. On the one hand, it would be reasonable for us to create a bunch of new valuable features based on the current raw categorical features (ip, os, app, channel, device, click-time). For example, we could try to count the click times within next 30 minutes or calculate the pass time between the click time and the time when user begin downloading. After that, we could use the 60% new attributes along with the raw features to train the model. On the other hand, it is also wise for us to adopt other learning algorithms such as Light Gradient Boosting or Extreme Gradient Boosting, which are derivatives of Decision Tree we originally used. Last but not the least, what we are going to cast effort on is the final visualization. We plan to convert our training data and testing data as well as their results into graphs for simple comparison. Based on the preliminary results and training data we have picked, we have confidence that there is still great potential for our model to improve.

As for questions or concerns, we actually have some concerns upon our project. Firstly, we currently have no specific idea about how we should use the attribute named ‘attribute_time’ effectively, which all exists in the positive examples. But we are trying to figure out create a new feature via combine this feature with some other attributes like ‘click_time’. What’s more, we are not sure the new features we are going to create would be beneficial to our model. But all in all, these concerns would be finally eliminated in the following couple of weeks.