Grab AI for S. E. A

Detect Dangerous Driving from Grab's Telematics Data

Wang Lu
17th June ,2019

Introduction

To push transportation in S.E.A safer, Grab launched this challenge and provided telematics data collected from phone Grab app. Timely and reliably identifying dangerous driving are two of the most important needs. Therefore, computational cost and model performance would be discussed and evaluated together in finalising candidate models.

Understanding features, and especially labels

The original features including acceleration and gyroscope on 3 directions, accuracy, and of special interest, speed and bearing. Some initial engineering efforts were made including compute change of bearing (derivatives), accumulated acceleration on each directions (integral), bearing times speed (product), Euclidean distance (acceleration) using 3 directions (x, y, z) and so on. Among which bearing X speed and Euclidean acceleration are most relevance and predictive. However, I run of time to generate new set of reference features just to integrate them into current solution. They are not presented here. (But if time allows, at least the two mentioned should be added.)

An especially important question to ask is how to define dangerous driving. An article from Uber **[1]** mentions user feedback is the main source for them to label dangerous driving. In this case, driving style is not the sole factor in determining dangerous driving, but also driver's attitude and personality, timestamp of the trip (passenger's mental sense of safety), area where the trip happen (congestion) and so on, although relevant information is not provided. Assuming Grab is using a similar methodology to generate labels, abnormality in features such as extremely long trip, negative speed, extremely inaccurate localization should be treated as valuable information instead of being filtered, as such abnormality most likely correlate with accident, unusual map or usual driver behaviour which generally led to passenger dissatisfaction.

Feature engineering

The task itself is by nature a supervised binary classification problem on imbalanced dataset (>70 % labelled as negative). However, it differs from many other classical classification problems in that feature extraction for a single trip has to be done from time series (sorted data), even though entries are disrupted and presented as snapshots in provided format. Therefore, signal processing techniques such as Fourier Transform, Continuous Wavelet Transform are reasonable ways to handle them besides statistical features, with FFT looking at this dataset from frequency domain solely, and CWT taking care of time-frequency coefficients. For time domain, there are two methods explored in this work. One way is simply calculating central tendency measures, dispersion measures (mean, mode, STD, …),

and absolute distribution features (by taking percentiles). Another way is to look at cross sectional distribution and statistics, that is to say, taking a relative perspective. This is justified by the intuition that understanding dangerous driving must be enhanced under context where individual behaviour (or trip) is compared with peer behaviour.

Back to the concrete problem itself. Detection of dangerous driving is an active research field. Some background checking on academia and industry methodology reveals that mainstream idea is to extract individual events and use them as features to characterise dangerous driving **[1, 2, 3].** For example, an abrupt acceleration/braking event, rapid turning and rapid lane change. More specifically, an abrupt braking can be extracted by hard thresholding, common cut-off is an accelerated velocity beyond 0.3G. A harsh left turn can be identified by sudden acceleration with simultaneous big negative bearing change. More ideally than hard thresholding, model can be trained to recognize fundamental events, which usually achieve much better performance. However, it would be too complex task altogether and require separating labelling for events. Due to time limits, industrial grade event extracting is not carried out, instead the method calculating peak features in this work is indeed derived from this idea, where peak statistics such as STD, frequency and intensity (assuming every set of peak is a group of notable events) are collected for different peak extracting thresholds.

Besides, an initial exploration shows the feature `bookingID` does have some predicting power on label at least for this provided training dataset, though vague. This most likely due to some unusual distribution of positive labels, for example, bookingID from 1709396983812 to 1709396983975 (about 20 consecutive ID) and from 17179869191 to 17179869363 (about 40 consecutive ID) are all labelled as positive, this is vey unusual considering rareness of positive labels. Assuming homogeneous generation of provided dataset and dataset to be tested, bookingID is also used as features in this model.

Pre-processing

Pre-processing is carried out but not intensively. As mentioned before, abnormality in features was treated as information. Scaling is therefore not implemented. Tree algorithms are especially good at handling dirty features (DecisionTree, GradientBoosting), thus they are chosen in both base estimators and final estimator for ensemble (GradientBoosting). Faulty entries in labels are detected, where 18 trips are labelled as both positive and negative. The second entries are simply ignored. PCA is used to compress CWT features into 1D. Otherwise all extracted features are equally fed in to base estimators to form predictions, which used by GradientBoosting later for final decision.

Model selection

The finalized model uses of extracted features from 6 methods as mentioned in the introduction. Without ensemble, a default GradientBoost model built on 513 raw features from get_idf (bookingID), get_fftf (features from fast fourier transform), get_cwt (features from continuous wavelet transform), get_stat(features from statistics) can give 0.78 AUC, which takes around 1 mins for data preprocessing (mainly concatenating and sorting) , and 6 mins for feature extraction, assuming 400MB testing data. With an ensemble process on all the same 513 features, GB performance can reliably improve to 0.80 AUC at extra cost of 10 mins. To ensemble on all 863 features, the whole computational cost will double (up to 30

mins), with 0.02 more improvement on AUC, reasonable but still a bit expensive for real-time need.

Feature selection

From cost effectiveness point of view, bookingID features, fft features, cwt features and statistic features in this model are more efficient than peak characteristics and cross-sectional features. However, GaussianNB does perform best on this two groups (up to 0.69 and 0.7 AUC in local CV), indicating underlying information is not fully exploited using current classification methods (no scaling, no multicollinearity checking and so on). If time allows, feature selection and pre-processing, especially on the last two categories should bring some performance bonus.

Optimization

Major efforts are made on feature engineering. Optimization is solely done to select appropriate base estimator depth for GB the ensemble estimator. Grid search is not carried out to make the model as simple as possible for business use. At the same time, it should not bring improvement on model improvement on the same features.

Discussion on the limitations

1) A general observation is that outliers and anomaly suggesting unusual context or behaviour instead of central tendency measures have stronger predicting power. Noticeably, the number of features generated from each category of method is empirically picked, but they in fact hugely impact performance of base estimators such as linear regression, KNN and Gaussian, which is more averse to dimensionality than Tree algorithms.

2) Estimators are chosen based on the heterogeneity, such as tree algorithms, neuron networks, linear algorithm and instance based KNN. Same family of algorithm at different implementation (parameters) are not explored with an assumption that similarity within family is greater than between families, which is unattractive for effective ensemble. SVM is also excluded concerning the computational cost.

3) As mentioned in beginning, some engineered features such as bearing X speed (potentially characterise an abrupt turn), acceleration calculated from Euclidean distance should be included but not.

4) Optimization is not fully explored, such as undifferentiated feeding of unscaled, unselected feature to dimensionality adverse algorithm, no hyper parameter tuning and so on.

Conclusion

In general, anomaly is valuable information and helpful in evaluating overall danger experienced by passenger. A medium expensive model can be fast implemented, from sorting, to extract ~500 features, to generate ensemble based on 6 estimators, cost around 15 mins on 400 MB data. However, this model should be easily improved by adding time-stamp or driver identification if the label to be predicted is based on user satisfaction.