Mercedes Benz Kaggle Winner's Solution

The Competition Problem

Before modelling, I try to understand the process behind the data and figure out what could affect the response. For this task I had very good aids aids: listen to others in the forum and view wonderful EDAs in kernels. For this learning stack I had some good time.

I set up some working hypotheses:

1. Lack of synchronization between manufacturing subprocesses leads to time delays.
2. Mercedes process must be a highly controlled one, only focalized perturbances.
3. Test time changes attributable to a small set of variables and/or short parts of few subprocesses.

I consider that in any modelling problem, feature extraction and selection are the primary issues one has to consider. What kind of new features would be interesting to explore?

The Feature Exploration Problem

Data features are:

1. Anonymized (no hints with names)
2. Only binary and categorical (prone to overfitting)
3. Just 2-way interactions of binary variables: 67528
4. Small data set relative to number of features
5. High quality data: apparently no need to deal with contaminated data

I used some hints to explore for more features:

Parameters in kernels Low value of max_depth in XGBoost suggested me that perhaps a few 2- or 3-way interactions and a small set of variables would be relevant in the sense that test time changes seemed to be attributable to a small set of variables and/or parts of few subprocesses.
Sequencing of feature names: 9 sequences, 8 categorical features I imagined that X0 to X8, the categorical features, were some sort of summary of parts of the manufacturing process. The holes in the sequencing of the binary features took me to define nine groups of binary variables, consistent with the eight categorical ones.
Interactions I found that discussions and kernels tended to confirm my view of the process, but notably, besides one kernel dealing specifically with interactions, I found no reference to any 2 or n-way interactions. The kernel on this subject did not deal with the features I used later.
Initial Features in Training Method
Search for interesting interactions looking at patterns in preliminary XGBoost runs. Pairs of variables (X314, X315), (X118, X119) and (X47, X48) appeared always "near" in the variable importance reports. So, these interactions were included and, additionally (X118, X314, X315). This last interaction did not conform to my understanding of the process, but I could be totally wrong.
As a tool to try to identify subprocesses and catch some information of them related to test time, cumulative sums of binary variables within each of the nine groups were formed. So in a first step the number of variables increased substantially. This was a risky decision, since adding correlated features to data can also add too much noise. Decision tree algorithms are relatively robust to these features.
To include or not ID was a question I tried to answer in preliminary runs. Discussions in the forum suggested that including ID was totally consistent with my thoughts on the Mercedes process. I detected modest improvements in preliminary runs and so it was included.
After some playing with the data, I decided to recode eleven of the levels of X0 (trigger of the process?) into new ones and added the recoded variable named recX0

One-hot encoding (dummies) of categorical features was applied, that is, the original X0 through X8 and the ones created for interaction variables. One-hot encoding variables were kept if sum of ones exceeded 50. Since this value looks reasonable, but arbitrary, it is subject to tests.

It is known that decision tree algorithms can handle categorical features transformed to numerical, something that makes no sense in other models. These features were also included, so extra weight is given to categorical features.

Training Method

Two models were trained with XGBoost, named hereafter Model A and Model B. Both are built in a sequence of feature selection steps, like backward elimination. Model B uses a stacked predictor formed in a step of Model A. Any decision point in this sequence is preceded by a 30-fold cross validation (CV) to find the best rounds. The steps are simple:

1. Preliminary model with all features included, Model A, 900 features and Model B, 900+1, the stacked predictor.
2. Feature selection. Keep the variables used by XGBoost as seen on variable importance reports (229 in Model A, 208 in Model B).
3. Feature selection. Include features with gains above a cut value in the models; 0.1%, in percentage, was the cut value used, 53 in Model A, 47 in Model B.

Both models use XGBoost and a 30-fold CV through all the model building process. The rationale for a 30-fold validation was to use them for a 30-fold stacking in Model B. With 30-fold the CV-mean results in different runs behaved too optimistic (0.59xx) in contrast with a 5-fold CV (0.57xx). On the other hand, CV-standard deviations increased substantially going from (0.04xx) in 5-fold CV runs to (0.11xx) in 30-fold CV runs.

Fortunately, last three of the final five submissions would have won. First two are previous versions of Model A with private scores 0.55292 and 0.55351; the third submission comes from Model A (0.55530). Fourth is from Model B, (0.55536) and the last, a simple average of A and B (0.55551).

In the models proposed Interactions played an important role

```
a)    By far, pair (X314, X315), jointly and pair levels

b)    3-way interaction (X118, X314, X315)

c)   X314

d)   (X118,X314,X315), levels (1,1,0)

e)   Individual features: X279, X232, X261, X29

f)   Two levels of recoded X0 and recoded X0

g)   Sum of X122  to  X128

h)   X127
```

Note: Fifteen features account for 85% of gain

Tools and Process Time

Tools: R Version 3.4.0, Windows version and required packages

Time to train both models in a desktop I7-3770 @3.40 GHz, 8 cores, 16 MB RAM: ~8 minutes

Time to process new data: Load packages and prepare model, ~6 seconds; predictions, for 4209 observations, ~2 seconds

Conclusions

1. Try including more two or three-way interactions
2. Model B does not really add value since both models are at the same R2 level of 0.555xx. Discard Model B; better ways can be found to improve results.
3. Try reducing features from the ones in Model A. Single test with 24 features: 0.5553
4. Some sensitivity checks on parameters for XGBoost might lead to slight gains.
5. The variables used are quite different from others in discussions, so there is a good chance to do some blending to capture synergies with one or both of the 2nd and 3rd places solutions.
6. Using feature ID in the model may result in poor results with new data. Time must be relevant, but needs special treatment. Single test without ID: Public LB, 0.55799, Private LB: 0.55421
7. As far as I have seen, the variables in the model developed are quite different from others in discussions, so there is a good chance to do some blending to capture synergies with one or both of the 2nd and 3rd places solutions.