Replicating Blair and Sambanis: Forecasting Civil Wars - JCR 2020

Introduction

While reading Blair and Sambanis (2020), a few issues stuck out. We downloaded the replication files authors provide from https://journals.sagepub.com/doi/suppl/10.1177/0022002720918923/suppl_file/sj-zip-1-jcr-10.1177_0022002720918923.zip. The original folder name of is `sj-zip-1-jcr-10.1177_0022002720918923/replication-3`. For simplicity, we extracted this `replication-3` folder and saved to the `Replication Files` folder of this repo. To ensure that we have a clean copy of the original replication scripts, I copied all files to `replication-3_BM` I'm (RM) currently using the sripts in this folder to recreate figure 1 and table 1 with and without smoothing – see below

Before listing the primary issues we identified, we should note that we are partial to their approach. Theory is important. It is helpful for forecasting problems, especially with regards to identifying which variables to include in a model. This is particularly true when the goal is to assess how ...

This document lists the primary issues we identified.

1. Using AUC for smoothed ROC curves. Does the interpretation of Table 1, and superiority of the escalation model, depend on the choice to use smoothed ROC curves?
2. In Table 4, are the forecasts evaluated against incidence or onset? The data from 2001 to 2015 have 20 civil war onsets in total, yet Table 4 appears to show 15 and 16 positive events for 2016 only. If this is indeed incidence, but the original models and forecasts were for onset, then this is a incorrect assessment of the forecasts' accuracy.

- Table 4!
  - There are only 20 onsets from 2001 to 2015 But there are 15 "Persistence" cases (top of table 4) and 16 "Change" cases (bottom of table 4)

3. Unorthodox RF hyperparameter choices that may be driving the OOS test prediction results. There are two issues here. One is that the RF models are so unorthodox. I'm not sure that this in itself is a problem. But the second potential problem is that the hyperparameter values may not have been tuned in a way that works the best for the non-escalation model, and especially the CAMEO model with 1,100+ features.

- Improperly tuning of models
  - RM: In a twitter DM, I asked Blair to clarify – "[I] was wondering how you all came to your tuning procedure for the RF model hyper-parameters?" He responded "mostly trial and error, honestly. ... Trial and error was on early data. Forecasts were for much later data"
- Using a regression framework in a classification problem
- Rounding of AUC/ROC scores
  - AB: smoothing or rounding? I think rounding is ok.
- Treating the output of a RF regression model as Pr() – (RM: I guess this is akin to a Linear Probability Model, but I'm not sure if this is possible in a RF framework
  - It's hacky but it works. The resulting forecasts are within the 0-1 interval.

The table below shows the randomForest package default hyperparameter values for a binary classification problem like the one at hand and compares them to the B&S base settings. Some hyperparameters have heuristics that determine the default value based on characteristics of the input data; we note these in the second column and what the realized default setting would be for the basic Escalation model (first row in Table 1 in the paper) in the 3rd column. The last column has the settings B&S use.

Hyperparameter	Default heuristic	Default values (Escalation)	B&S value
type		classification	regression
ntree		500	100,000
mtry	`floor(sqrt(ncol(x)))`	3	3
replace		true	false
sampsize	`nrow(x)` if replace, else `ceiling(.632*nrow(x))`	11,869	100
nodesize	1 for classification	1	1
maxnodes		null	5

It is worth noting that commonly the ntree, mtry, and nodesize parameters are the main parameters tuned; of these three in the B&S specification only 1, the ntree parameter, deviates from the default settings.

In any case, there is a stark contrast in the default RF settings and the way B&S use the RF models. The default approach is to train a relatively small number (ntree; 500) of classification trees, but where each tree is fairly big in that it is trained on data that has the same number of rows as the training data, albeit sampled with replacement (replace is true; sampsize is 11,869), and allowed to grow fairly deep (this is governed by nodesize (1), which is the minimum size a terminal node must have). In contrast, B&S grow very extensive forests with a large number of trees (100,000 compared to 500), but each tree is very small and shallow; only a 100 rows are sampled from the training data for each tree, and the trees are constrained to at most 5 terminal nodes (maxnodes).

This approach only works due to the other unorthodox choice, which is to use regression, not classificaton, trees. Trying to use classification trees with the other parameter settings in fact does not work at all because it is almost guaranteed that a sample of 100 from the 11,869 training data rows with 9 positive cases will only include 0 (negative) outcomes in the sample. As it is, using regression with a 0 or 1 outcome produces warnings when estimating the models:

```
Warning message:
In randomForest.default(y = as.integer(train_df$incidence_civil_ns_plus1 ==  :
  The response has five or fewer unique values.  Are you sure you want to do regression?
```

While unorthodox, the approach does work. It produces predictions that are within the 0 to 1 interval. The concern is that the settings work well only for the escalation model, and only for the particular test set at hand. Specifically, we wonder (1) if the hyperparameter settings only work well for the particular test set chosen, and (2) if the CAMEO model with 1,100+ features, compared to 10 or less for the other specifications, would perform better with a more explicit tuning procedure.

Are the RF hyperparameters overfit to the test set?

Does the base specification's good performance in the test set generalize?

One initial piece of evidence is already available from Table 1. In addition to the test set starting 2008, B&S also evaluate test sets starting in 2009, 2010, and 2011 (sets 1, 2, and 3 in Table 1). For the 1 month models, when we compare the base specification performance of each model to the performance in the alternate test sets, the AUC-ROC increases in 2 (of 15) cases, is the same in 0 cases, and decreases in 13 cases. For the 6 month models, it increases in 1 case, is the same in 3 cases, and decreases in 11 cases. Altering the test set thus generally shows reduced performance. The table below shows cross-validated training and OOS test AUC-ROC for the original 1 month escalation model, a RF model with default hyperparameter settings, and a tuned RF model with 10,000 trees and otherwise the default settings. The first set of results are out of sampel results from repeated cross-validations performed on the training data. We show the average AUC-ROC, it's lower and upper 95% CI, obtained via bootstrapping, and the standard deviation of the distribution of resampled AUC-ROC values.

Because there are only 9 positive cases in the whole training data, and thus to ensure that any given split will include at least 1 positive case in each data partition, we use 2-fold CV, i.e. splitting the original training data into equally-sized new training and validation sets. This is repeated 21 times for a total of 42 OOS

performance samples for each model. The last column shows the OOS performance on the original test set. The first value in this column corresponds to the base escalation model results reported Table 1 in the B&S paper when *not* using smoothed ROC curves.

```
tbl <- structure(list(Model = c("Escalation, 1mo", "Modified Escalation, 1mo",
"Tuned Escalation, 1mo"), Avg_CV_ROC_AUC = c(0.677707841149067,
0.619719616721681, 0.671630392456263), ci_lower = c(0.634101138978023,
0.5835673668958, 0.634713167892701), ci_upper = c(0.722598486545989,
0.65584632967561, 0.706310478133188), SD_CV_ROC_AUC = c(0.147542574525528,
0.122337186539073, 0.117973980721395), Test_ROC_AUC = c(0.783352194140576,
0.586147564308735, 0.582927991423296)), row.names = c(NA, -3L
), class = c("tbl_df", "tbl", "data.frame"))

knitr::kable(tbl, digits = 2)
```

Model	Avg_CV_ROC_AUC	ci_lower	ci_upper	SD_CV_ROC_AUC	Test_ROC_AUC
Escalation, 1mo	0.68	0.63	0.72	0.15	0.78
Modified Escalation, 1mo	0.62	0.58	0.66	0.12	0.59
Tuned Escalation, 1mo	0.67	0.63	0.71	0.12	0.58

The results show that:

- Both of the alternative RF models are able to achieve roughly similar OOS performance in the training data split. T-tests comparing model 1 to the other models fail to reject the null hypothesis at a 95% confidence level. (A t-test comparing model 1 and 2 average AUC-ROC just barely fails to reject the null hypothesis (p slightly above 0.05).)
- Only the base escalation model is able to achive good test performance; the other two models, despite achieving similar training data performance, have significantly lower test performance.

This suggests that the base RF specifications are (over-)fit to the test data.

4. Design choices

Rick, I think the points below are more subjective choices. The first two points above are IMO potentially objectively incorrect technical errors that could undermine the original B&S findings. The 3rd point about the RF models is somewhere between objective error and subjective choice. I'm not sure. But the points below I would say are subjective choices that could alter their findings and that I think are justifiable criticisms, but I think it'll be easier for someone to push back on this if they wanted to.

- Their train/test approach
  - 5-year forecasts for sep plots and AUC scores)
- Smoothing of AUC/ROC scores
- Lack of yearly test forecasts in favor of a single 5-year test forecast
- The lack of procedures to account for rare events in an RF model

To-do

1. Full replication
   - Can we get their results
   - RM: maybe I can go through their code and bring it "up-to-date" i.e. make it tidy...
     - AB: that sounds like a lot of work
   - Figure out how they built table 4 (It looks like this is done in STATA)
   - Check to see if it matter that they are using regression for a classification problem
     - AB: i tried switching the models as they have them just from regression to classification and it breaks because the data samples end up being all 0's.
2. Address tuning

- Does properly tuning the CAMEO model improve its performance?

3. Address AUC/ROC smoothing...
    - How drastic is the change; does it change their conclusions?
4. Run a yearly test forecast – as is the current standard in TSCS forecasts (Is it the current standard?)
5.