Discussion of "Nowcasting US Business Cycles with Machine Learning"

Philippe Goulet Coulombe

University of Pennsylvania

Modelling with Big Data and Machine Learning: Interpretability and Model Uncertainty
November 5th, 2019

Summary

- **The goal**: knowing if we are in recession/expansion in real time.
- Usual issues:
  - We cannot wait for GDP data to come out.
  - Standard data is subject to multiple revisions, which can be quite misleading in real-time (think 2008...)
- The authors propose another way
  1. Use S&P 500 firms returns data. Big advantage: no revisions.
  2. Estimate a network at different frequencies
  3. Create a bunch of statistics describing the network Disadvantage: like any statistic extracted from the data, it changes with additional data.
  4. Take these statistics as new data.
  5. Use Recursive Feature Elimination to select less than 20 series out of >30k of them
  6. Fit a simple model to GDP_t using the selected subset of series.

The last two steps are in fact the Machine Learning part.

Open Questions about the Network and Data Reduction Steps

- Why using pairwise Granger causality rather than one huge network estimated with LASSO à la Diebold-Yilmaz?
- Recursive Feature Elimination, somewhat unheard of. It would be nice to justify it wrt to the many available alternatives.
- How does it differ from Backward Stepwise Regression?
- RFE uses SVMs, which operates quite differently than NB. Does it makes sense to do that? Have ML people done that? Would be nice to know.
- The full model must be estimated in the first step, how is that done, with 30k features? Could thee elimination path could depend on the original regularization?

Why Naïve Bayes?

- Standard econometric approach would be to fit a probit/logit. Logistic Regression (LR) is also widespread in ML.
- NB is a surprising choice: it amounts to fit a linear model $X\beta$ where the β's are all obtained by separate regressions. If features are correlated, we obviously get biased estimates. Authors seems to select regressors in order to satisfy loosely this condition.
- Ng and Jordan (2001) look into the NB vs LR debate.
- They point out that NB can perform better when there's a shortage of training data, which is the point raised by the authors as well.
- However, LR should always outperform given enough data.
- Fitting less than 20 coeffcients with 150 data points does not sound like data shortage.
- Even if it was the case, doing ridge/lasso regularization seems more legit than resorting to NB.

Obvious econometrics things to look into

- Is the exercise done in quasi-real time? That is, the network statistics and the parameters of NB are estimated recursively?
- Compare results to a logistic regression. Big plus: we can even look at coefficients and attempt some post-selection inference.
- Compare the results to a standard factor model with standard macro data as a real benchmark to beat
- To satisfy NB conditional independence (or get closer to it) : extract factors out of a trimmed version (left to be defined) of their data set. Might work better because of (i) more data (ii) and (ii) factors are uncorrelated by construction.

Obvious ML things to look into

Again, using a trimmed version of the data set (for computational reasons), I'd be curious to see

- Using Random Forest, which is known to select quite well what features matter and is almost immune to overfitting and the abundance of useless features.
- Boosting, for similar reasons
- Both are super fast (in R)
- The point being that is much more one can do with that data than NB
- LASSO/Elastic Net will most likely perform poorly in such an environment but are rather hard to dismiss given their popularity (and interpretability).

In short: if the paper is truly about ML with new (wider) data, there many interesting avenues to look into.

thanks.