Machine Learning for Central Banking

Daily Takeaways from BSP Lectures

Paul D. McNelis, S.J.

October 2023

Outline

1. October 16
2. October 17
3. October 18
4. October 19
5. October 20
6. October 23
7. October 24
8. October 25
9. October 26
10. October 27

Introduction

- No free lunch in regression analysis:
- We have to filter our data.
- We have to know the questions we are asking.
- We have to check the assumptions of the regression model.

Regression

- Single equation is not the way to go.
- We have to take seriously the assumptions of Gauss-Markov.
- Are the regressors exogenous? Is the disturbance term IID?
- Moving to first-differenced weekly data does not get rid of serial dependence.
- Ergodicity: we can move to monthly first differences or quarterly.
- No free lunch: we lose observations.

Macroeconomics and Reality

- Chris Sims developed the Vector Autoregressive (VAR) model in the 1980s.
- It is one of the principal workhorses of policy analysis.
- As Sargent notes, it is a state-space model that brings together good dynamic econometrics with good dynamic economics.
- All variables depend on other *lagged* endogenous variables.
- Adding more lags removes higher-order serial correlation, as shown by the Ljung-Box Q statistic.

Interpreting the VAR

- We can use Granger causality to see if one variable is a cause or significant predictor of another variable.
- We can use impulse response functions to see how one-time changes in one variable affect the dynamic response of other variables.
- We can use Forecast Error Variance Decomposition (FEVD) to see the relative importance of one variable for the overall variance of other variables.
- We can use the FEVD matrix to see if one variable has more outward or inward connectedness to other variables in the system.
- The relative strength of bivariate connectedness can be visualized with Directional Graphics.

Questions about VAR Models

- Are the results of VAR regressions robust to the choice of the number of lags?
- As we increase the dimensions of the VAR, or lags, or both, we rapidly increase the number of parameters.
- For a VAR system of 10 variables with a lag structure of 5, we have 510 parameters, if we also include constant terms.
- So a VAR rapidly consumes degrees of freedom.
- There is also the ever-present danger of *overfitting*.
- Another way of putting things: we encounter the bias-variance trade-off.

Selection Criteria

- Need for *regularization* criteria
- After getting rid of serial correlation, one can add more lags and get a better fit
- So we need to handicap our models: adjust the Likelihood L by the number of parameters k for a given number of observations n:
- Akaike: $AIC = -2\ln(L) + 2k$
- Schwartz: $BIC = -2\ln(L) + k\ln(n)$
- Hannan-Quinn: $HQIC = -2\ln(L) + 2k\ln(\ln(n))$

VAR a la Sims

- We learned that one can derive information from Granger causality, IRF, and FEVD
- When doing IRF and FEVD, for small VAR's, we use the Cholesky decomposition to orthogonalyze the residuals
- In this way, each shock is independent of the other shocks so we can interpret the effects of a shock to one variable
- No free lunch: the results depend crucially on the ordering of the variables
- We can use the Pesaran Generalized Forecast Error Variance Decomposition
- Results do not depend on the ordering of the variables but interpretation of the shock is less clear-cut
- We can also *bootstrap* the regression results and obtain confidence intervals for the IRF and FEVD estimates

Regularization Criteria

- We can use the Akaike, Schwartz and Hannan-Quinn criteria for model comparison for different numbers of parameters
- Basically idea is to handicap the inverse of the Likelihood by 2K, ln(K) and ln(ln(K)), where K is the number of parameters
- Select the model which delivers the lowest values of the information criteria
- Often we get different ranking of models by different criteria.
- Broader issue is over-fitting and the *bias vs. variance trade-off*

Elastic Net and Cross Validation

- With EN we handicap the Sum of Squared Residuals by a factor λ, α for the sum of the absolute values of the coefficients or the sum of squared values of the coefficients
- We find the optimal values of the parameter λ by *Cross Validation*
- We start with grids on λ, α and choose a percentage of observations to pull out of the sample and use as test or validation sets
- We select the values of λ, α, which deliver the lowest out-of-sample mean prediction errors.
- We showed that the Elastic Net with Cross Validation is a ruthless killer of coefficients
- The ones that survive are important
- The FEVD results can be used to asses the relative inward and outward connectedness of the state variables

Volatility

- The GARCH frame is the most widely used way of estimating time-varying volatility.
- Such volatility is a proxy for the latent uncertainty or risk process.
- GARCH model led to the development of VaR analysis (Value at Risk).
- Problem: risk in this setup has no independent drivers, it is only a function of the lagged prediction errors.
- Stochastic volatility models have emerged to compensate for this drawback of GARCH.
- We can estimate such models with Maximum Likelihood or Generalized Method of Moments.
- GMM allows us to simulate the artificial data for longer periods than actual data.

Limits of Linear VAR models

- We revised how returns can be calendar adjusted for days of the week and months of the year. Very important to do so.
- The VAR model is linear.
- We have to figure out how to destroy nuisance parameters.
- There is no free lunch: to interpret the results, one has to use the Cholesky decomposition.
- This means we have to order the variables in a special way.
- In order of importance?

GARCH Models

- The GARCH model due to Engle is another workhorse of financial empirical work.
- It allows for a time-varying risk or volatility but the risk only depends on the shocks to the return.
- There are no shocks to volatility (aka uncertainty shocks) which are different from shocks to mean return forecasting errors.
- Is this realistic?

SVJD Model

- Start with a model which allows shocks to the standard deviation apart from the shocks to the mean return.
- Assumption is that they are correlated by a factor $\rho < 0$. The idea is that negative shocks to mean return will increase volatility or uncertainty.
- We make use of the continuous-time Bates SVJD (Stochastic Volatility Jump Diffusion) model.
- It incorporates both types of shocks, to mean and conditional variance, but also permits a Poisson shock to mean returns.
- We thus have four stochastic processes, for Normal shocks to mean and variance, a Poisson shock for jump occurrence, and a shock for the size of the Poisson shock.
- We can estimate such a model with seven parameters.

Estimation

- Estimating the highly nonlinear stochastic model is challenging.
- Newton's Method: $\Delta\Omega_t = -\frac{J_{t_1}}{H_{t-1}}$
- Starting with guess at time t= 0, we iterate till convergence
- Problem: the Hessian H is often impossible to invert and the Gradient J often vanishes
- It is easier to obtain parameters that deliver local rather than global optima.
- Choice between Maximum Likelihood and Generalized Method of Moments (GMM).
- We see the asset returns have a high degree of kurtosis relative to the Normal Distribution.
- Makes sense to do GMM over Maximum Likelihood estimation for getting the coefficients.

Methods of Estimation

- We can start with *global* methods such as Genetic Algorithm.
- Then go to less global, more local, but stochastic methods like Simulated Annealing (SA) and Particle Swarm (PS).
- Then go to local gradient descent methods like ADAM.

Hyper-Parameters

- We have to choose hyper parameters
- For a feedforward neural net, we choose the number of hidden layers and number of neurons in each layer
- Shallow vs. deep neural networks
- We also choose the activation functions for the neurons: sigmoid, tansig, RELU, Leaky RELU
- Different layers can have different activation functions, why not?
- We can choose the learning rate and the momentum parameter for the gradient solution method
- We can also set the L1 regularization parameter and number of iterations for stopping.
- We also choose our objective function: sum of squared errors, mean absolute error.

Train-Validation-Test Splits

- We usually break the data set in three, a training set for estimation, validation set to peek at for tuning hyper-parameters
- We also have a test set to evaluate the performance after we finish up our best model.
- We can do our own cross-validation, by altering the L1 parameter to give the best performance in the validation set.
- Idea is to work with the specification which gives the best out-of-sample performance by the test-set evaluation

Uses of the Neural Net

- **Supervised Learning**: forecasting and classification
- **Unsupervised Learning**: data compression with auto-encoding networks
- **Reinforcement Learning**: real-time decisions to get a reward or avoid a cost (buy-sell-hold decisions
- Neural nets can be used to *approximate* or forecast variables which are obtained from more complex processes
- Neural nets outperform polynomial and orthogonal polynomial approximators (Chebeycheff, Legendre, LaGuerre, and Hermite)

Quantile Regression

- *Quantile regression* is a useful way to evaluation the sources of system market risk
- It is basically a regression (linear or nonlinear) through a given quantile of the dependent variable rather than the mean
- We basically examine how a set of regressions explain deviations of a dependent variable from its value at quantile τ.
- The $\Delta CoVar$ Method due to Adrian and Brunnermaier does a regression, for re-ordered negative returns, for predicted values from $\tau = .95$ less the predicted values for $\tau = .50$.
- The method tells us how much the returns for one bank, given a set of controls contribute to the market falling 45 percent below the median value of the market
- It thus tells us which banks or firms put the market at significant risk.
- The regression can be done with Neural Nets as well as with linear and polynomial regression

Understanding Risk and the Use of Shallow or Deep Networks?

- We reviewed accuracy for quantile regression with linear models and shallow vs. deep networks
- We found that the shallow network outperformed the deep networks for in-sample accuracy
- As a transmitter of risk we found that HSBC was the leader of the pack for the weighted returns of the rest of the 20 GSIB's (with returned weighted by market capitalization
- By contrast, STT was the leading risk transmitter when we used range volatility
- These results may be complimentary not contradictory
- Quantile regression captures extreme tail risk while range volatility captures ordinary risk
- There is RISK and risk.

Use of Data

- We discussed that the use of training and test data may be problematic
- If we are doing a classification, what if the percentage of one category is very small relative to the other one?
- We can bootstrap from both categories an equal number and then evaulate the out-of-sample performance on the remaining data
- This is similar to the .632 bootstrap of Blake LeBaron
- Our problem is the relative size of the different categories if we are doing classification

Credit Card Default and Texas Banks

- We examined the use of Discriminant Analysis, Logit, Probit, Weibull and Neural Nets for classification
- For the Credit Card and the Texas banks for out-of-sample evaluation, we found that the Logit and Neural Net were equally accurate
- We discussed that the classification evaluation depends on how on evaluates the False Positive and False Negatives
- Putting equal weight on both errors may not be wise in most situations
- Error probabilities should be weighted by disutility payoffs.

Corporate Bond Ratings

- We looked at bond ratings [AAA to C] as the dependent variable for 4000 firms
- We looked at Logit, Random Forests, XGboost, ADAboost for both out-of-sample accuracy and relative importance
- The most important feature for predicting the rating turned out to be the ratio of the market value of equity to the book value of total debt.
- For evaluating the tests, we found it is harder to accurately predict the top winners (AAA vs AA) that to predict the bottom of the pack (C rated bond issuers)

Interpreting Classification

- We need to carefully assess the relative importance of False Positives and False Negatives
- Only if both have equal distuility payoffs should be average them
- Classification involves careful assessment of the risks
- We used logistic, Random Forecast, Multilayer Perceptron (neural net) and XGboost
- We found that the results are sensitive to the specificaiton of hyperparameters
- Good reason to work in teams
- As Tim Kehoe once said, the secret of doing good research is finding good co-authors.

Banks

- We looked at daily share price data for a dozen banks
- We applied the SVJD jump diffusion process
- We found common patterns, even though some were more volatile than others
- Big increase in the latent uncertainty around 2018
- This was a time of policy change for banks
- This method is a good way to asses, ex post, how policy changes affect the processes of underlying latent variables
- One idea: see how volatility in the exchange rate interacts with banking-sector volatility

Classification and Compression

Default Predictions

- We examined data on defaults of 30+ countries over 25 years
- We compared the accuracy rates of Logistic, RF, XGboost, and MLP (Neural Net)
- Large number of hyper-parameters to explore
- Good discussion of the difference between "confusion matrix" and the percentage of TP, TN, FP, FN
- Good discussion about splitting the sample into training and test sets
- Class consensus was that both the training and text should have equal percentages of 0 1 as in the full data set
- We found that public debt to total debt ratios were important. Monetary factors were less important for default.

More on Hyper-Parameters

- One issue is how to compress data. We found not much difference between standardization and normalization between zero and one.
- We found in the default case, that the MLP network did better with more hidden layers and more neurons, with Logistic function and Adam as optimizer
- This is useful.
- The model was estimated as a Panel Data set.
- We can construct separate dummies for each country as well as year dummies.
- This will add more than 50 coefficients to the model
- But we have regularization terms so the irrelevant ones will vantage.

Compression

- We reviewed origins of PCA.
- Idea was to do Reduced Form estimation for 2SLS to get around *simultaneous equations bias*, very serious
- If the model was big there were too many exogenous variables
- One idea is to take a few Principal Components of all of the exogenous variables for created the instruments for the RHS endogenous variables in the model
- Other uses of PCA were to amend the Captial Asset Pricing model
- In both the Auto Encoding Network and the PCA, the components are called *latent features*
- Factor Analysis is close to PCA but conceptually different. There is also Dynamic Factor Analysis used for Now-Casting.
- Consensus is that PCA's and Auto Encoders do the job.

FRED Data and Forecasting

- We showed how to use the FRED data base and to chart properties of data
- We then showed how to do forecasting of key variables in the FRED database with VAR and Lasso
- The data base consists of month macroeconomic variables from 1959 to the present

Data Compression

- We discussed the program for compressing data for Hong Kong Swap Volatilities
- The first example was Hong Kong interest-rate swap volatilities
- Idea is to identify periods of idiosyncratic risk for HK since the risk is not from any exchange-rate uncertainty
- We first used the traditional Principal Component Analysis
- We did both in-sample and out-of-sample performance evaluation

Auto-Encoder Network

- We then examined the performance of the Auto Encoder Network with alternative network "architectures"
- We found that a small network with only a few hidden layers and sigmoid functions performed almost as well as a large network the RELU functions
- Both beat the PCA in terms of out-of-sample performance and the Diebold-Mariano Statistic.
- We looked at several other methods, such as K-Nearest Neighbor, but noted that the PCA and Auto Encoder are the most widely used

Bank Stock Returns

- We then looked at the share market returns of ten Philippine banks
- We found that most of the action was captured by two "latent vectors" which explain about 85 percent of the total data variation
- We see that there is a switch in the direction of the two latent vectors asfter 2016.
- We interpreted the two vectors as two financial stress indicators, one of low-stress with relatively high and stable returns, and the other of high stress (with lower returns)
- We identified several banks which are more highly correlated with the high and low stress indicators.
- Not surprisingly, the banks correlated the low stress indicator were among the high-performers in returns, while the opposite was true for the banks correlated with the high stress index

Coming Attraction

- We finished the day with a review of *Recurrent Neural Networks*
- We looked at the equations for Elman, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks for time series
- The lags are on the hidden layers of the network.
- Networks are allowed to forget, in part and over time, some of the hidden lagged neurons
- We reviewed simple codes provided by chatGPT.

Banking Stability-Capital Asset Ratios

- We discussed how to use Linear and Auto Encoder methods for obtaining leading indicators of financial stability
- We first looked at the Excess Capital Ratios of 30+ banks since 2008 with quarterly data.
- We found that most of the variation of the data can be explained by two components or "latent factors"
- We found that one component moved upwards after 2016 and the other moved downward.
- One speculation: there are two segments, one with bullish spirits and one with bearish spirits
- We identified several banks highly correlated with one and several correlated with the other component.
- Results suggest that this can be a useful tool for identifying which banks are the leaders of the bears and leaders of the bulls.

Non-Performing Loan Ratios

- We looked at data for 39 banks with anonymous names with NLP ratios
- We noticed the variation of the ratios in the data
- Only a few banks in the sample had a high Coefficient of Variation of this variable.
- Main issue was lack of a balanced data set.
- Do we throw out banks from the sample if their series started later than others?
- One suggestion was to treat the exercise as a panel data estimation with unbalanced panel
- Issue is that we have many more parameters to estimate.

Inflation Forecasting

- For the last session we looked at inflation forecasting
- We used quarterly data since 2005 with inflation (four quarter ahead) as the dependent variable
- The regressors were inflation measured over four quarters, gdp growth or output gap, world commodity inflation, the excess return on currency
- We split the sample in two, leaving out the last seven quarters as the out-of-sample test set for measuring forecasting performance
- There is strong autocorrelation in the model but we are just interested in forecasting, not significance of parameters

Estimation Performance

- We estimated the model with simple OLS as well as with Elman, LSTM and GRU Recurrent networks
- These models build in lagged memory neurons as well as an optimal "forget rate"
- We used alternative hyperparameters for the Recurrent Networks.
- Measuring the out of sample performance by the Root Mean Squared Error of the out-of-sample forecast errors, the GRU performed best.
- However, all of the models under-predicted the inflation rates coming from the COVID-19 crisis.
- This should not be surprising

Extensions for Inflation Forecasting

- We can build inflation forecast *intervals* for the final seven quarters based on various permutations of the models
- Starting with the GRU we can trying different seeds for initializing the parameters of the model and compare the forecast performance
- We can vary the hyperparameres of the different models and find an out-of-sample inflation forecast confidence interval.
- How so? Run the model 100 times with different configurations of hyper parameters and then order the inflation forecasts for each period from highest to lowest. Delete the highest and lowers 2.5 percent and then plot a 95 percent confidence interval
- Another idea: include in the interval DSGE forecasts for the same periods to assess broad model consistency.
- In other words, let the Neural Nets be a check on the DSGE and let the DSGE be a check on the Neural Nets.

Concluding Remarks

- Doing work with ML methods requires a lot of work with computational methods, learning from mistakes, assessing data quality, looking for outliers.
- At times this seems like a tedious waste of time. It is not. Top people in the professor spend a lot of time doing stuff like this.
- I asked Tom Sargent once about this, why does he not leave this work to research assistants. He told me he learns a lot from the nitty-gritty of the data and the programming. So one can still have big thoughts and do work like this. They are not incompatible activities.
- Sometimes I felt that I was a research assistant to chatGPT. But think of chatGPT as your language instructor, correcting your grammar and helping you with pronunciation and vocabulary lessons. Or a music teacher giving us practice scales and cords. How we put the scales and cords together for great music is our task.

Research Strategy

- We need a language to develop big ideas but the language teacher does not produce the big ideas. We do, but having good language ability helps a great deal. So chatGPT is there to help us with our Python language and, for me, LATEX writing.
- As in jogging or golf practice, do not overdo it. Spending a half hour a day on learning Python coding for various problems will have a high payof and save you time as you prepare presentations. Also look at some good math books from time to time.
- Some references:
  1. Rashka and Mirjalili, "Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn and TensorFlow 2, 3rd Edition."
  2. Angel de la Fuente, "Mathematical Methods and Models for Economists."
- Both are good for browsing and fine-tuning your skills and refreshing your memories of math concepts.
- GitHub will be available, with the sample codes