Machine Learning in Finance: 1

Overview

- Econometric Models: Recap
- The GARCH Model and Conditional Volatility
- VAR's and Big Data
- Nonlinear Estimation: Opportunities and Pitfalls
- Why Machine Learning

Econometric Models: Recap

- We examined **simple** and **multivariate** regression models
- We examined **linear** and **nonlinear** regression models
- Linear models have **closed form** solutions, whereas nonlinear models do not.
- Multivariate models can be single equation models or s set of simultaneous equations with several dependent variables.
- In forecasting, we usually start with the single-equation simple linear regression model:

$$y_t = \sum \beta_k x_{k,t} + \epsilon_t \tag{1}$$

$$\epsilon_t \sim N(0, \sigma^2) \tag{2}$$

- ϵ_t is a random disturbance term, usually assumed to be normally distributed with mean zero and constant variance σ^2 , and $\{\beta_k\}$ represent the parameters to be estimated.

Parameter Estimation

- The goal is to select $\{\widehat{\beta}_k\}$ in order to minimize the sum of squared differences between the actual observations y and the observations predicted by the linear model, $\widehat{y}$.
- The estimation problem is posed in the following way:

$$\underset{\widehat{\beta}}{Min} \Psi = \sum_{t=1}^{T} \widehat{\epsilon}_t^2 = \sum_{t=1}^{T} (y_t - \widehat{y}_t)^2 \tag{3}$$

$$\text{s.t.} \quad y_t = \sum \beta_k x_{k,t} + \epsilon_t \tag{4}$$

$$\widehat{y}_t = \sum \widehat{\beta}_k x_{k,t} \tag{5}$$

$$\epsilon_t \sim N(0, \sigma^2) \tag{6}$$

- The symbol $N()$is the normal distribution function.

The ARX Model

- Commonly used linear model for forecasting is the Autoregressive X (ARX) model with one dependent variable y depending on its lags and a set of exogenous X-variables:

$$y_t = \sum_{i=1}^{k*} \beta_i y_{t-i} + \sum_{j=1}^{k} \gamma_j x_{j,t} + \epsilon_t \tag{7}$$

- k independent x variables, with coefficient γ_j for each x_j, and k^* lags for the dependent variable y, with, of course $k + k^*$ parameters,$\{\beta\}$ and $\{\gamma\}$, to estimate.
- Thus, the longer the lag structure, the larger the number of parameters to estimate, and the smaller the degrees of freedom of the overall regression estimates.
- The number of output variables, of course, may be more than one. Then we would call this a VARX (Vector Autoregerssive Model with X exogenous variables)
- But in the benchmark linear model, one may estimate and forecast each output variable $y_j, j = 1, ..., j^*$, with a series of J^* independent linear models. For j^* output or dependent variables, we estimate $(J^* \cdot K)$ parameters.
- The linear model has the advantage of having a **closed form** solution.
- Coefficient vector is a straightforward generalization of the simple estimator above.
- For short-run forecasting, the linear model is a reasonable starting point, or "benchmark", since in many markets, one observes only small symmetric changes in the variable to be predicted, around a long-term trend.

The VARX Model and Granger Causality

- The multi-equation VARX (Vector Autoregressive X model) is a generalization of the ARX single-equation multivariate model
- It is a model of several equations with a set of dependent variables, with each variable depending on their own and each other's lags as well as a common set of exogenous X-variables.
- For a two variable model for $y_{1,t}, y_{2,t}$ we can write the following system:

$$y_{1,t} = \sum_{i=1}^{k*} \beta_i y_{1,t-i} + \sum_{i=1}^{k*} \delta_i y_{2,t-i} + \sum_{j=1}^{k} \gamma_j x_{j,t} + \epsilon_t \tag{8}$$

$$y_{2,t} = \sum_{i=1}^{k*} \kappa_i y_{1,t-i} + \sum_{i=1}^{k*} \lambda_i y_{2,t-i} + \sum_{j=1}^{k} \rho_j x_{j,t} + \epsilon_t \tag{9}$$

- $y_{1,t}$ is independent of $y_{2,t}$, then the set of coefficients δ_i are jointly insignificant. Similarly if $y_{2,t}$ is independent of $y_{1,t}$ then the set of coefficients κ_i are jointly insignificant
- If δ_i coefficients are significant, but κ_i are not significant, then y_2 is a Granger cause of y_1.
- If δ_i coefficients are insignificant, but κ_i are significant, then y_1 is a Granger cause of y_2.
- If both sets of coefficients, δ_i and κ_i are significant, then there is feedback between y_1,y_2
- This approach is widely used to examine tests of causality among key macroeconomic variables

The GARCH Model: Conditional Volatility

- We are often not only interested in forecasting returns of a variable but also its **risk**
- We proxy risk by conditional volatility
- It comes from the Generalized Autoregressive Conditional Heterosexuality (GARCH) model.For an asset return y_t, we specify and estimate the following model:

$$y_t = \alpha + \epsilon_t \tag{10}$$

$$\epsilon_t \sim N(0, \sigma_t^2) \tag{11}$$

$$\sigma_t^2 = \delta_0 + \delta_1 \sigma_{t-1}^2 + \delta_2 \epsilon_{t-1}^2 \tag{12}$$

- The target depends only on a constant α and a disturbance term ϵ which as mean zero and **a conditional variance** σ_t^2
- The variance is not constant but changes through time. It depends on its own lag at time (t-1) and the squared disturbance term at time (t-1)
- Since the distribution of the shock is "normal" we can use maximum likelihood estimation to come up with estimates for $\alpha, \beta, \delta_0, \delta_1$,and δ_2.
- The likelihood function L is the joint probability function for $\widehat{y_t} = y_t$, for $t = 1, ... T$.
- For the GARCH models, the likelihood function has the following form:

$$L_t = \prod_{t=1}^{T} \sqrt{\frac{1}{2\pi \widehat{\sigma}_t^2}} \exp\left[-\frac{(y_t - \widehat{y_t})^2}{2\widehat{\sigma}_t^2}\right] \tag{13}$$

$$\widehat{y_t} = \widehat{\alpha} \tag{14}$$

$$\widehat{\epsilon_t} = y_t - \widehat{y_t} \tag{15}$$

$$\widehat{\sigma}_t^2 = \widehat{\delta}_0 + \widehat{\delta}_1 \widehat{\sigma}_{t-1}^2 + \widehat{\delta}_2 \widehat{\epsilon}_{t-1}^2 \tag{16}$$

Log-likelihood Functions

- The usual method for obtaining the parameter estimates maximizes the sum of *logarithm* of the likelihood function, or log-likelihood function, over the entire sample T, from $t = 1$ to $t = T$, with respect to the choice of coefficient estimates.
- We impose the the restriction that the variance is greater than zero, given the initial condition $\widehat{\sigma}_0^2$ and $\widehat{\epsilon}_{t-1}^2$.

$$\underset{\{\widehat{\alpha},\delta,\widehat{\delta}_1,\widehat{\delta}_2\}}{Max} \sum_{t=1}^{T} \ln(L_t) = \sum_{t=1}^{T} \left(-.5\ln(2\pi) - .5\ln(\widehat{\sigma}_t) - .5 \left[\frac{(y_t - \widehat{y}_t)^2}{\widehat{\sigma}_t^2} \right] \right) \tag{17}$$

$$\text{s.t.:} \widehat{\sigma}_t^2 > 0, t = 1, 2, \ldots T \tag{18}$$

$$\widehat{\sigma}_t^2 = \widehat{\delta}_0 + \widehat{\delta}_1 \widehat{\sigma}_{t-1}^2 + \widehat{\delta}_2 \widehat{\epsilon}_{t-1}^2 \tag{19}$$

$$\widehat{y}_t = \widehat{\alpha} \tag{20}$$

- In most optimization software, we maximize by minimizing the negative value of the log-Likelihood function

Pitfalls of GARCH Estimation

- The appeal of the GARCH approach is that it pins down the source of the non-linearity in the process.
- The **conditional variance** is a nonlinear (quadratic) transformation of past values, in the same way that the variance measure is a nonlinear transformation of past prediction errors.
- One of the major drawbacks of the GARCH method is that minimization of the log-likelihood functions is often very difficult to achieve.
- Specifically, if we are interested in evaluating the statistical significance of the coefficient estimates, $\widehat{\alpha}, \widehat{\delta}_0, \widehat{\delta}_1$, and $\widehat{\delta}_2$, we may find it difficult to obtain estimates of the confidence intervals.
- All of these difficulties are common to maximum likelihood approaches to parameter estimation.
- However, the restrictiveness of the GARCH approach is also its drawback: we are limited to a well-defined set of parameters, a well-defined distribution, a specific nonlinear functional form, and an estimation method which does not always "converge" to parameter estimates which make sense.
- With specific nonlinear models, we thus lack the flexibility to capture alternative nonlinear processes. We will see this flexibility with Neural Net models for deep learning

Why Machine Learning?

- Machine Learning Methods have made a big comeback.
- Neural networks were big in the late 90s and early 2000s. See my book, Neural Networks in Finance: Gaining Predictive Edge in the Market [Elsevier, 2005]. Matlab code for the chapters is available on my GitHub page: www.github.com/McNelis-CMML.
- Now Neural Network analysis is a part of Machine Learning called Deep Learning
- Machine Learning also includes LASSO/Elastic Net methods for parameter reduction and Random Forests
- All help us cope with large number of regressors (wide data sets).
- We will cover neural nets, for sure, but also other methods such as Random Forests and Clustering Methods
- We are interested in forecasting, classification and clustering (how to partition large data sets to smaller classifications of data)
- We also have faster hardware and better solution algorithms to handle big data with nonlinear models

Wide and Deep Data Sets

- Data sets are big in two senses: deep and wide
- Deep data sets mean we have large, very large numbers of observations so degrees of freedom is not an issue
- Wide data sets mean we have lage numbers of characteristics or covariates for forecasting
- Even with a deep data set, how can we do a regression with several hundred regressors?
- We need to figure out how to reduce the dimension of wide data sets
- But we want to exploit meaningful information from these sets.

Interface with Econometrics

- In ML the dependent variable y_i is called the **target**
- The set of regressors,$x_{i,k}$, k =1,...K is now called the set of **covariates** or attributes
- The sample of data used for estimation is called the **training** set
- The coefficients of the covariates are called weights, constant terms are called **biases**.
- The data set for out-of-sample performance tests is called the **test** set
- Estimation is called **learning**
- When we try to use covariates to predict or classify a target, we have supervised learning
- When we try to cluster or partition data sets, we call this unsupervised learning.
- In general we are not interested in tests of significance of parameters

Drawbacks of Linear Models

- If we have a lot of regressors, the big problem with linear models is lack of independence of regressors
- Most linear models look for parsimony, few regressors
- Rarely do we studies with more than a few regressors
- If we have too many, there is a high likelihood of **multicollinearity**
- The model cannot be solved, or it generates absurd results.
- Even if the interdependence of the regressors is not very high, hard to make sense of results

The Twin Curses: Dimensionality and Multicollinearity

- We may get garbage for our regression results. Or no results, just "Inf" or "NaN"
- The key assumption of basic linear regression is that the regressors are independent
- The likelihood of statistical dependent falls as we add more regressors. Problem of abandoning linear methods
- The larger the number of regressors, the more likely we have a high degree of multicollinearity
- We thus have to throw out a lot of information (discard variables) or conflate many variables
- For example: income and tax payments as regressors. They are co linear, taxes depend on income.
- So we define a new regressor: disposable income, equal to income less taxes
- Still the wider the data sets, the more information we can extract.

Advantages of Linear Models

- Linear models have exact **closed form** solutions
- Once we solve for the regression coefficients, they are unique.
- Anyone using the same data will get the same result
- The result is based on minimization of the sum of squared errors with respect to the coefficient vector β

$$\overset{Min}{\hat{\beta}} \sum_{i=1}^{N} (y_i - x_i \tilde{\beta})^2$$

- The closed form solution is $\beta = (\hat{x}'x)^{-1}$x'y. This is known as the Ordinary Least Squares (OLS) estimator β for β.
- Solving for the coefficient vector is also very fast
- As you see, the coefficient estimation requires that we can invert the matrix $(x'x)$
- The larger the dimension of the matrix x, the harder it is to invert $(x'x)$.
- The curse of multicollinearity is closely related to the **curse of dimensionality**.

Costs of Going "Nonlinear"

- We do not need invert a matrix to get the coefficient estimates
- Nonlinear models can handle a higher degree of multicollinearity
- Instead we take a guess of the solution vector of coefficients and iterate on the coefficients
- This goes all the way back of Isaac Newton, no less.
- As noted above, we want to minimize the sum of squared errors:
$\underset{\hat{\beta}}{Min} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$
- Now $\hat{y}_i = f(x_i; \hat{\beta})$,since we have a nonlinear (unspecified model.
- So the Sum of Squared Residuals, SSR, is a nonlinear function of $\hat{\beta}$.
- So to minimize the SSR we have to iterate based on initial guesses of $\hat{\beta}$

Nonlinear Optimization

- Issue is to minimize a Sum of Squared Errors with respect to coefficients: $SSE(\beta) = \overset{Min}{\hat{\beta}} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$
  - This function is more complex, it is not a simple quadratic function
  - To find the vector of coefficients, we need to take an initial guess and then iterate:
  - $SSE(\beta_1) = SSE(\beta_0) + (\beta_1 - \beta_0) SSE'(\beta_0) + .5(\beta_1 - \beta_0) SSE''(\beta_0).(\beta_1 - \beta_0)$
  - $SSE'(\beta_0)$ is the gradient or Jacobian of the error function, $SSE''(\beta_0)$ is the Hessian (matrix of second derivatives
  - Minimizing the error function, given a guess of β_0, gives the following recursion formula: $\beta_1 = \beta_0 - \frac{SSE'(\beta_0)}{SSE''(\beta_0)}$