4.5 Model Adequacy and Model Evaluation

Ming Lu

2022-04-02

Modeling time series involves a certain amount of subjective judgement, and what’s desirable is to keep this subjectiveness as low as possible. With that in mind we can apply (and we should apply) various statistical tests to guide the selection of appropriate models. To fit an ARMA model we use the ACF and PACF functions to reveal a possible autoregressive relation on returns, and determine the order of this autoregression from the plot of those functions, or information criteria as the AIC (Eq. (4.29)). We have a test for the ARCH effect which helps to decide for fitting a volatility model. To select input features for the machine learning models we can run causality tests (Sect. 3.2) and feed the models with those time series that present a significant cause–effect relation with the target series (e.g., lags of the target series, its volume, or any other feature from the series or other). In a more general, and possibly primer stage of selection of models, there is to determine the possibility of the time series having a nonlinear relation with its lagged values, or with another time series, so that we focus our model selection to nonlinear models. Let us review some commonly used tests for nonlinearity relation among time series.

# 4.5.1 Tests for Nonlinearity

There are various statistical test to assess if a group of time series are nonlinearly related. Two of these tests, based on neural networks, are Teraesvirta’s test and White’s test for neglected nonlinearity. What this means is the following.

Given the general regression model, , where we view the information set as a -dimensional vector that may contain lagged values of , and is the true unknown regression function. is linear in mean conditional on , if for some , where is the parameter vector of the optimal linear least squares approximation to . The null hypothesis of linearity and the alternate hypothesis of interest are as follows:

When the null hypothesis is rejected the model is said to suffer from neglected nonlinearity, meaning that a nonlinear model may provide better forecasts than those obtained with the linear model. The neural network test for neglected nonlinearity, formulated by White, uses a single hidden layer feedforward neural network with nonlinear activation functions capable of approximating an arbitrary nonlinear mapping. Teraesvirta’s test is based on, and improves, White’s neural network test by using a Taylor series expansion of the activation function to obtain a suitable test statistic. Both tests are implemented in the R package tseries by the functions terasvirta.test () and white.test (), and returns -values of a Chi-squared test (by default) or an F-test (if type is set to ). Example 4.7 We test for nonlinearity among the and Shiller’s ratio for this market index, namely the series PE 10, analyzed before. We will use Teraesvirta test. The data, as we know, is in a . csv table, and so we want to transform it

to s form for handling the data by dates, since we want to consider the series from 1900 to 2012 (just for illustration purposes). Note that we specify the format of the date variable, as it appears in the file. This is to prevent errors in loading the data. > library(tseries); library (xts) inData as .xts (read.zoo(‘SP500\_shiller.csv’, sep , header , format ) data=inData [‘1900/2012’] na.omit (data$SP500) na. omit (data$P. E10) terasvirta.test The result of the test that you should see is: Teraesvirta Neural Network Test data: sp500PE and sp500 -squared -value The small value indicates that a nonlinear relation holds from PE 10 series to the . Thus we can try fitting a nonlinear model to the S& P 500, for example a NNET or SVM, using the PE 10 series as part of the known information. We will do that in the last section of this chapter.

# 4.5.2 Tests of Model Performance

After fitting a time series model the next and final step is to assess the performance of the model; that is, its accuracy as a predictor. Time series prediction evaluation is usually done by holding out an independent subset of data (the testing set) from the data used for fitting the model (the training set), and measuring the accuracy of predicting the values in the testing set by the model. Let be the testing set from the time dependent data that’s been modeled, and let be the values predicted by the model in the same time period of the testing set. Some common measures to determined how close are the predicted values to the actual values in the testing set are the mean squared error (MSE), the mean absolute error (MAE) and the root mean squared error (RMSE), defined as follows:

Note that these are scale-dependent measures, and hence, are useful when comparing different methods applied to the same data, but should not be used when comparing

across data sets that have different scales. The RSME is often preferred to the MSE since it is on the same scale as the data, but both are more sensitive to outliers than MAE. An experimental comparison of these and other scale dependent, and independent, measures of forecast accuracy is done in Hyndman and Koehler (2006). These prediction error measures, and variants, are all available in the package of evaluation metrics for machine learning, Metrics.

Alternatively, to evaluate the models, instead of measuring the accuracy of the predicted values, one can look at how well the model predicts the direction of the series, that is, if it has gone up or down with respect to the preceding value. This is the same as treating the forecaster as a binary classifier, and consequently appropriate measures of its accuracy are obtained from the information recorded in a contingency table of hits and misses of predicted against actual upward or downward moves. The general form of a contingency table for recording predictions of direction is like the one shown below,

where is the size of the testing set (equal to the number of predicted ups and downs), (resp. ) is the number of hits in predicting upward (resp. downward) moves, (resp. ) is the number of misses in predicting upward (resp. downward) moves, etc. The performance of the tested model can then be measured with the information given by this table with three different metrics:

Accuracy: computed as a simple percentage of correct predictions: . Cohen’s Kappa (Cohen, 1960): this measure takes into account the probability of random agreement between the predicted and the actual observed values and it is computed as , where is the observed agreement and is the probability of random agreement, that is, the probability that the actual and the predicted coincide assuming independence between predictions and actual values. Therefore, when the predicted and the actual values completely agree. Directional Measure: it is computed out from the contingency table as . Similar to Cohen’s Kappa, large values of tell us that the model outperforms the chance of random agreement. behaves like a chi-squared distribution with 1 degree of freedom, and we can use this information to compute the quantile with respect to a given significance level (Tsay 2010).

The models with best performance, according to one or a combination of these measures, are selected, and then their parameters can be re-calibrated and the new models tested again until some threshold in the measure valuations is reached or these values stabilize.