Time Series Forecasts with Random Forest

The questions:

* How to use random forest with time series?
  + Is it possible to overfit?
* How to tune the parameters?
  + Feature fraction
  + Minimum size vs. other criteria (for leafs)
* Applications
  + Significant variables (and how these change with horizon)

For tuning:

* Pick a few hyperparameters
* Select values for those hyperparameters
* Use cross-validation to choose best values

**Introduction**

Predicting the future is hard. It makes sense, then, to explore all of the resources at our disposal when we set out to forecast. We should not limit ourselves to a certain class of models, as if only these could reasonably be expected to forecast well. Indeed, some classical models do a fair job, but no model is perfect, and many models even fare worse than a naïve forecast.

Given the rise of machine learning methods and their success in modeling cross-sectional data, and given the difficulty of forecasting time series data even with sophisticated classical models, it makes sense to at least consider applying some machine learning techniques to time series forecasting. Classical models such as the ARIMA and the VAR are valuable, and they often perform significantly better than a naïve forecast. But machine learning is a valuable and underexplored resource when it comes to time series data.

With this in mind, this paper offers a brief exploration of one particular machine learning technique, the random forest, and its success in forecasting a particular set of time series data, the US seasonally-adjusted monthly inflation rate.

**The Random Forest**

The random forest was first developed by Tin Kam Ho in 1995. It refers to a collection of regression trees, which are each trained on a randomly selected subsample of features. This method allows the “forest” (the collection of trees) to grow increasingly accurate while also remaining resistant to overfitting as it grows in size. Ho’s work represented a major development in statistical modelling: “an increase in classifier complexity” did not lead to “overtraining” (Ho)

**Tuning**

The bulk of this project consists in appropriately tuning the forest and the trees. In preparation for this task, we anticipate a number of difficulties that may arise when using the random forest method on inflation data. First, inflation data is a time series. Time series data present a bundle of unique challenges and quirks, which the random forest method is not explicitly designed to handle. Second, an especially prominent challenge that results from the time-series nature of our data is the problem of cross-validation. Any tuning technique will require some form of cross-validation; with cross-sectional data, it is straightforward to use a k-fold cross-validation technique. With time series data, however, the process requires more careful consideration. Third, we realize that our tuned trees will only be used to forecast in the context of a random forest that we also tune. Therefore, a perfectly tuned standalone tree may not actually be the best tree to use in the context of the forest. How do we tune the trees and the forest together? This is something we need to consider.

These are all preliminary considerations, conceived before the process of tuning is even begun. In addition to these considerations, other difficulties may arise as we embark upon the project in earnest. Such difficulties, if they do indeed arise, will be detailed below. There are impossibly many parameters to consider, so we will need to identify the most important ones and tune those. For example, even in the process of just randomly sampling the data for each tree in the forest, should we sample with replacement? Randomly sample without replacement from a fixed point in time? From a random point in time? How big would this random sample without replacement be? And none of these parameters can be tuned in a vacuum; each of these questions would have to be answered simultaneously with each other and with dozens of other such questions, regarding for instance the number of trees in the forest, the feature fraction for the forest, the splitting penalty for each tree, etc. Thus, to perfectly tune the forest would take an extraordinary amount of computation, and could risk overfitting (over-tuning?) the tree and forest itself, such that the model would not be effective over different horizons or for different types of time-series data, and suggesting that the model may not be especially adept at predicting inflation as the underlying process evolves. In other words, the random forest model that is so perfectly tuned for data from 1960-2020 may fall out of tune as the year 2025 approaches, and its predictions may get worse over time as it fails to sufficiently adapt to changes in the inflationary process.

The random forest and regression tree methods are explicitly cross-sectional models. They are not designed to handle time trends or seasonal variation or evolving processes. The theory with time-series data is that it is motivated by some underlying process, typically an autoregressive one. This process, however, is not perfectly consistent over time, and may include a time trend or seasonal variation, or may simply evolve over time, so that the process in the 1960s differs from the process in the 1990s. The random forest method is not equipped to handle this sort of data. However, the random forest method can be very effective at predicting cross-sectional data, and it is not impossible that with a few adjustments and with some careful handling, this method could present an improvement over a classical ARIMA model. To realize that potential improvement is the goal of this project.

Random forest and cross-validation are inherently cross-sectional techniques. It doesn’t quite make sense to k-fold the time series data, because you’ll end up validating past data on future data. This seems problematic because the future data theoretically has no effect on the past data. For all we know, the time series is fundamentally different in the 2010s as opposed to the 1960s. Maybe there’s a structural break, maybe there’s some kind of non-linear time trend; at any rate, it would be best to avoid validating past predictions on future data.

To minimize the computational complexity of this project, we will consider two parameters. One of these, the penalty function, relates to the regression tree itself; the other, the feature fraction, relates to the random forest.

When tuning the forest, it’s important to consider the trees and the forest itself as complementary entities. In other words, neither the tree nor the forest should be tuned in a vacuum. Instead, the tree should be tuned with the understanding that it will be used in a forest explicitly designed to accommodate overfitting. Thus, we may allow the tree to fit the data more tightly than we would if we planned to use the tree alone for forecasting.

Tin Kam Ho, "Random decision forests," Proceedings of 3rd International Conference on Document Analysis and Recognition, Montreal, Quebec, Canada, 1995, pp. 278-282 vol.1, doi: 10.1109/ICDAR.1995.598994.