Time Series Modelling and Forecasting with Applications in R

Prof. Sudeep Bapat

Shailesh J. Mehta School of Management

Indian Institute of Technology Bombay

Department of Applied Statistics/Finance

Week 12

Lecture 56: Machine Learning in Time Series

Hello all, welcome to this course on time series modeling and forecasting using R. Now we are entering the last week of this entire 30-hour course, and as you see in front of you, the very last topic will be a new idea here—or rather a recent one—which is machine learning integration in time series. I understand that some of you may not have a background in machine learning, but you need not worry. The entire topics this week will be more from an introductory perspective rather than delving deeper into each and every idea. We'll try to develop some understanding of the different machine learning ideas that can be implemented and integrated into a time series setup, okay?

On the other hand, I am very sure that many of you might already have some understanding of machine learning. For example, the usual regression you do, classification problems, decision trees, random forests. All these form the crux of what you call machine learning, but over the next week or so—especially this week—we'll focus not only on machine learning but also on a few advancements, such as neural networks in time series or deep learning ideas in time series, and so on. Of course, depending on time.

I've kept this at the very end because it is not typically introduced in a traditional time series course. But due to recent demands or the integration of ideas like ML, artificial intelligence in general, or deep learning and neural networks, I thought of including at least some topics and giving a flavor of ML integration into the time series domain. Again, the first session will be, as I said, more from an introductory perspective. We'll try to understand different ML ideas that could be implemented in a time series framework.

And of course, as we go deeper into the week, we'll try to focus on some individual ideas, such as regression, random forests, decision trees, etc. And then we'll explore some applications of those in time series and so on. And of course, there will be one session specifically about neural networks. There, I'll try to explain neural networks from scratch. So again, if nobody is very familiar with the idea of neural networks,

And of course, we will try to integrate neural networks into time series by developing a very famous model called the neural network autoregressive model. So, of course, all of that will follow, but the first session will be more introductory. So, we'll start with a broad idea about machine learning in time series and a brief introduction to get things started. Machine learning models have gained popularity in time series analysis due to their flexibility. So, a couple of points here.

First is flexibility and the ability to capture complex patterns. So again, I'd like to pause here and go back to some of the initial sessions of this course to understand where we started. So, we started by developing some very basic time series models. Again, if you remember, we covered autoregressive, moving average, then combined AR and MA into ARMA, then ARIMA, SARIMA, right? But one particular feature there was linearity.

So, we sort of assumed very strongly that let us say the time series is linear or at the most it can be transformed to a linear sort of a setting, right. But again, many, many practical examples are, of course, not linear. So linearity goes for a toss. Right. And then how do you capture non-linearity?

So, of course, in the last few sessions, we've discussed non-linear time series as well. So if you remember, we discussed about threshold models and so on and so forth. But what next? Right. So let's say if you have like really complex patterns, then, of course, all these models may not do good.

And you require some sort of a black box method. So the entire idea about machine learning, we should always understand that ML is always a black box kind of a method. So what do you mean by black box method? So black box method means that if you give the ML model or ML technique some inputs, then you actually don't know that what's going on inside the model, right? So you can't explain what the model is.

So let's say decision trees, random forest or support vector machines, right? So you can't really enter into the model setting and then trying to explain each and every point and hence the black box. So you don't know what's going inside the modeling sort of the arena. And if you throw in some inputs, then the ML model has to give some outputs and the outputs are what you visualize or what you see towards the end of the day. So, in that sense the entire area of ML is has its own pros and cons.

So, why pros because it is able to very well manage the complex patterns as written here, but at the same time if you want an explainable model or if you want to see more in depth as to what is going on inside the model, then you do step back a little bit and then really try to sort of explain it using some traditional time series models or be it linear regression sort of a model and so on and so forth. So the idea of black box one should always understand not just for this course but let us say if you are sitting for any other ML course or you know introductory ML course, deep learning course, neural networks course or data science course. But the crux of all the ML models are that the ML models are nothing but black box models. Alright anyway so again just to repeat the first point that ML models have gained very much popularity in time series analysis due to their flexibility. And on the other hand, the ability to capture some complex patterns.

And secondly, depending on the nature of your data and task you want to achieve, you can choose among several models. So again, ML, as we discussed a short while back, that a lot of models sort of form the crux of the ML model, sort of an umbrella. And then you can choose among several models, each with its own strengths and weaknesses. So each ML model, let's say regression, classification, SVM, random forest, decision trees, etc. Each commits own pros and cons.

So, the following slides will cover the different kinds of models initially which one can develop and then how to integrate each and every model into a time series setting. Because again, at the end of the day, you should not forget that the data we have is nothing but yt sort of data where it depends on the time point. Ok. So, how do you pick up any practical time series data, which is let us say yt, yt plus 1, yt plus 2, etcetera, and apply some of the well-known or traditional ML models to integrate the two, ok? So, what exactly are all the classical ML models or classical machine learning models?

So, we will again spell out a few. So, let us say these models are often applied with feature engineering techniques, right? So, again, do not worry. So, we will try to explain what we mean by feature engineering techniques where lag variables, rolling statistics, or Fourier transforms are used to create the features. So, again, I will pause here for a second.

So, again, many of the terms or terminologies might be new to many of you. So, let us say features. So, features are nothing but independent variables or nothing but the inputs that you provide to the model, ok? So, this is a very classical terminology pertaining to the ML setting that features are nothing but all the independent x variables or independent inputs that you provide to the ML model, ok? Now, how do you apply the feature engineering techniques or, by the way, what exactly would be the features, right?

So, these are some of the ideas. So, let us say lagged variables. So, X t minus 1, X t minus 2, X t minus 3, etcetera, or rolling statistics. Or, at the same time, Fourier transforms are used to create all these features. Now, what exactly are these?

So, let us say linear regression. So, again, I am very sure that almost all of you should have some understanding of linear regression, right? So, linear regression forms the base of any of the advanced ML models also. So, linear regression, how do you use it? So, it is used for forecasting when the relationship between the dependent variable, let us say y, and its lagged variables is linear, right?

Now, again, one important point to note here is that if you are sitting in a non-time-series kind of course or if you are sitting in a regression course, then these lagged variables would be nothing but the independent variables, and these are nothing but the x variables, right? But since we are trying to integrate the regression idea into time series, so what we can do is we can have, let us say, one dependent variable which is y t, and one very easy thing to achieve or sort of get all the independent variables is simply look at all the lagged variables of y t. So, y t minus 1, y t minus 2, etcetera, okay. So, all these would be my independent variables. And, of course, y t is the dependent variable, alright.

So, this is a very, very easy sort of integration or a sort of link between a linear regression model and a usual time series setting, okay. So, again, linear regression is used for forecasting when the relationship between the dependent variable and its lagged variables is linear. So, again, linearity is sort of the important aspect here. If linearity is not met, then again, you would go for some of the advanced ML models and so on and so forth. Now, the second would be SVM.

So, the full form of SVM is support vector machines. So, again, these are applied in time series classification and regression tasks, especially with non-linear relationships. So, again, anything apart from regression, you are slowly entering into the non-linear setting, okay, one by one. So, again, another important terminology here in this slide is classification. So, classification stands for, let us say, if your dependent variable is a binary variable, let us say 0 or 1.

I can give you some standard examples, let us say yes versus no or males versus females, right? Or again, let us say high versus low, something like that. So, whenever the dependent variable has categories which are only two, then this is a classification problem or rather a binary problem, okay? So, support vector machines are applied in time series classification problems or regression tasks. But especially with non-linear relationships.

So, if you want to explore more in a non-linear setting, then of course, linear regression would not be useful, right, since the term itself contains 'linear.' But as I said a short while back, one can actually explore some advanced techniques, such as SVMs, random forests, gradient boosting, etc. Okay. And thirdly, random forests and gradient boosting. So, all these are capable of handling non-linear relationships.

Right. So, I should highlight that they perform effectively and work well with engineered lag features. All right. So, this is a small example about random forests and gradient boosting. So, these are capable of handling non-linear relationships effectively and perform well with engineered lag features.

And some examples might include, let us say, highly non-linear time series structures. So, let us say if a time series structure is highly non-linear. So, linearity goes for a toss. So, you have lots of ups and downs, lots of complex patterns, lots of, you know... So, you have a trend plus seasonality, plus again, the trend is not a continuous trend.

So, let us say it is upwards and downwards. So, let us say you have a highly non-linear sort of time series, a practical time series. Then, of course, linear regression might not be useful. SVMs might not be useful again. So, you may require some

ideas like random forests or gradient boosting, etc. Now, on the other hand, we have a plethora of different models that connect the neural networks idea to time series. And again, as I said a short while back, you need not worry. There will be one extra session—not extra, there will be one session this week pertaining only to neural networks integrating with time series. So, what exactly are neural networks?

So, again, deep learning models are useful for capturing non-linear dependencies and complex temporal structures, right? So, again, neural networks are capable of handling non-linearity very well, as well as temporal aspects. So, again, this is slightly newer terminology; we did not use this quite often in the earlier lectures. So, temporal means anything that depends on time, okay? So, something varying temporally, or if something varies across time, we sort of say that you have a temporal sort of structure in that.

And these are some of the very well-known or widely applied neural networks in literature. So, the first one is recurrent neural network or RNN, and even within RNN, we have these two, which are very widely applicable. So, the first one is LSTM, or long short-term memory. And these are effective for long-term dependencies in sequential data. So, let us say if you want to preserve the memory within the neural network method or methodology, then one can actually go with an LSTM sort of structure or long short-term memory structure.

And on the other hand, let us say GRU. So, GRUs mean gated recurrent units. So, GRU is a simpler alternative to LSTMs with comparable performance in many cases. So, again, just to repeat, you need not worry at all as of now. So, I am just trying to give you an overview of what ML techniques or deep learning techniques—neural networks—are there, where integration of all these ML models into a time-free setting is possible.

But down the line, as I said, we will explore the idea of neural networks a bit more in a different session, okay? But again, just to repeat, these are some of the widely applicable ones when it comes to neural networks. So, again, both of these fall under or come under the RNN. So, RNN is recurrent neural network. So, the first one is LSTM, which is long short-term memory.

So, if you want to preserve let us say memory over a longer time then one can actually use LSTM. On the other hand, one can use GRU or gated recurrent units. Then the third one is convolutional neural network or CNN. So, again we are under the neural networks heading. So, the third one could be CNN or convolutional neural networks and these are again useful for extracting local patterns from time series especially in combination with RNN.

So, let us say if you want to explore more locally. So, locally means in a short neighborhood or rather than exploring for the overall time series. Let us say if you want to divide the time series into batches into smaller batches and then extract some useful information from all the small batches one by one. Then rather than applying RNN one can actually go with CNN. ok or convolutional neural network and transformers.

So, again ah I am I am pretty sure that. So, again I am sure that all of you must have heard all these terminologies somewhere right. So, neural networks, deep learning, transformers, CNN, RNN, LSTM, ML techniques, random forest etcetera, but again this this particular session would be again useful to give a overview or sort of an idea about what exactly all of these mean. in basic terms and later on of course we will see that how do you integrate each one of them with a time series setting ok so under transformers what we have so first one is attention mechanism so effective for long sequence modeling right and then second one when it comes to integration with a time series setting you have a time series transformer or in short THT And then adaptations of the transformer architecture for time series forecasting and classification.

So, transformers could be used again in a two-fold mechanism. So, one is forecasting, and the second is classification. So, we want to classify an object into different categories, like we discussed a short while back. Then one can again use a time-series sort of transformer or TST for classification problems or the usual regression sort of setting to forecast. And then next is the autoencoder.

So, what do you mean by that autoencoder? So, it is used for anomaly detection and feature extraction in time series. So, again, I will pause here for a second and try to understand or try to make you understand what anomaly detection is. So, again. I think we introduced this idea long back in this course when we were seeing some examples, right, of different time series applications.

So, one very important application was anomaly detection. So, anomalies are any situations which sort of change the usual path of the time series. So, I will give you an example. So, let us say you have a time series. which is behaving like this, and suddenly you see that you have a sharp drop, right.

So, maybe a stock price or something like that, or let us say gas prices, right. So, this could be some hypothetical example, but let us say suddenly the time series moves like this, and suddenly you see a sharp drop, and then again, let us say the time series goes up and then starts behaving as earlier. So, these could be anomalies. So, anomalies are all those which are slightly different from the usual pattern, right. So, how do you detect that?

So, it is called anomaly detection, then estimation forecasting, right? But if you do not detect the anomalies, then if you simply blindly put forward an ML model or some time series model, it will not do a good job, right? Because you have to capture all these anomalies in the data. And for that, autoencoders could be useful, and again, autoencoders are inside the neural network framework also, okay. Then the next kind of model is called a hybrid model. So, again, as the name suggests, hybrid should be a combination of two or three different things, right. So, hybrid models combine traditional statistical models or machine learning techniques with neural networks, okay.

So, on one hand, you have the traditional statistical models or ML models combined with some idea about neural networks, okay. Now, again, I will try to explain a bit more as to what you mean by a statistical model. So, a statistical model is any model which is explainable. So, let us say ARIMA or regression, right, where you can actually write down the model. So, if I tell you that YT follows ARIMA or YT has an ARMA structure or AR structure, MA structure, whatever, then one can actually write down the model using a pen and paper, isn't it?

So, all such models which are explainable are called as statistical models. So statistical models are not black box models, right? Because you know what is going on inside the model. You know the complete model structure. You can play around with the parameters.

You can estimate the parameters, right? So it's not quite a blind model, right? So all such models are called as statistical models. On the other hand, any model which is not explainable are ML models. So let's say random forest, decision trees, gradient boosting, SVMs, etc.

And the third idea is neural networks, which we just discussed. So any sort of a combination between statistical techniques or statistical models or ML techniques with neural networks are called as hybrid models. And all these are sort of a standard example. So, first one could be ARIMA plus neural networks. So, combines the statistical models, let us say ARIMA with neural nets to capture residual patterns.

So, again, the idea is exactly the same. So, why would one want to combine all these? So, let us say if you want some explainability in the model. So, if you do not want a completely black-box model, then why not just combine them? So, that you have some room to play around with explainability.

And a black-box sort of output, right? So, you are introducing or rather combining an ARIMA setting, which is explainable, along with some of the advanced neural network settings. So, you can actually explain the model very well using the ARIMA structure as well as try to capture the non-linearity in the existing model. Okay? Because, again, just to repeat, ARIMA cannot handle non-linearity very well, right? So, if you combine

ARIMA with neural networks, you get a strong sort of hybrid model. So, again, it combines the statistical model, which is ARIMA in this case, with neural nets or neural networks to capture the residual patterns. The second set of examples would be CNN-RNN hybrids. So, use CNNs to extract the features from the input and RNNs for temporal modeling. So, again, this is a combination of CNN and RNN.

So, one can actually use convolutional neural networks to extract the features from the input. And one can use recurrent neural networks for temporal modeling. So, temporal modeling is nothing but time-series modeling. So, let us say, to define the actual RNN model, one can use RNN, and to extract the features and information from all the features, one can use CNN. And the third could be Deep AR.

So, a probabilistic forecasting model was developed by Amazon using RNNs again. So, you have a special name for that. So, these models are called Deep AR models. And then, Deep AR is nothing but a probabilistic forecasting model developed by Amazon using RNNs. The same sort of idea, which is the RNN idea.

Now, the next set of models is called probabilistic and Bayesian models. So, again, I will try to explain very briefly what I mean by Bayesian. So, Bayesian is a slightly different category in statistical models that includes prior information. So, whenever somebody says Bayesian, right? So, Bayesian always relies on some prior information.

So, what do you mean by prior information? I will give you a very simple example. Let us say you want to estimate a simple parameter theta, right? So, again, forget about time series for the time being, right? Let us say the idea is to estimate this unknown parameter theta, and for some reason, you know some information about theta.

So, that information is called prior information. Usually using some historical studies or some other, let us say, news for example, right. So, you have some prior information about this parameter theta. So, how do you apply or how do you use this prior information and the likelihood of the data that you have to get some idea about theta, ok? So, this route—using some prior information to improve the estimation of theta—is called the Bayesian school.

So, any sort of Bayesian technique or Bayesian estimation technique, Bayesian forecasting technique, or Bayesian predicting technique always makes use of some sort of prior information. So, on the other hand, we have probabilistic and Bayesian models.

What do you mean by that? So, probabilistic approaches provide uncertainty estimates in forecasts. And these two are widely applicable.

One, the first one is Gaussian process. So, non-parametric models suited for smaller data sets are called Gaussian processes. And on the other hand, we have Bayesian structural time series, or in short, BSTS, useful for modeling trends, seasonality, causal inference, etc. So, again, what do you mean by probabilistic and Bayesian models is that it sort of uses some prior information first. On the other hand, there should be some defined probability.

So, when it comes to Gaussian processes, the term 'Gaussian' stands for a normal distribution, by the way. So, a normal distribution—the other name for it is Gaussian distribution, by the way. So, Gaussian processes explore the idea of normality, but again, one can also have non-parametric models suited for smaller datasets, though there must be some probabilistic idea. So, one can propose some probabilities. On the other hand, BSTS models, or Bayesian Structural Time Series Models, are useful for modeling trends, seasonality, causal inference, etc.

Okay, then the next set of models. So again, if you follow, we are trying to explore a bit more each time and then delve deeper into the advancements of all the models, right? Now, this slide presents some specialized models for time series. So, the first one is called the Facebook Prophet Model. It is designed for business forecasting, handling seasonality, holidays, trends, etc.

By the way, all these are specifically suited for modeling time series data. So, Facebook Prophet. The second one is the N-BEATS model. So, Neural Basis Expansion Analysis for Interpretable Forecasting. So again.

Let us not delve deeper into each and every model; just try to understand what all models are in place that one can apply or deploy, right, depending on the dataset we have or the goal we have—be it forecasting, handling seasonality, or handling trends, right. So, accordingly, one can choose or rather pick the suitable model and deploy. Work using that. So, Facebook Prophet is the second one, and NB is the third. The third one could be, let us say, Temporal Fusion Transformer or TFT.

So, this TFT model combines attention mechanisms with interpretable forecasting. So again, the basic idea is how you move from linearity to non-linearity, then to even more non-linearity, which is called complex patterns, and so on and so forth. Now, the next one is called ensemble learning. So again, many of you might have heard about bagging techniques and boosting techniques. So, all these are part of ensemble learning ideas.

So, combining models often improves performance, right? Let us say bagging and boosting techniques. So, what do you mean by bagging and boosting? So, aggregating predictions from multiple models. Let us say, for example, Random Forest, XGBoost, etc.

On the other hand, one can have stacking. So, what do you mean by stacking? So, layering models where predictions of one serve as input to another. So, all these models are nothing but a combination of different models, which is called ensemble learning. So, ensemble means a combination of different techniques or a combination of different models, right?

So, one can actually combine some models to improve performance. So, let us say random forest, XGBoost, or stacking. So, stacking means layering several models where predictions of one can serve as input to another model. Now, a bit more on classical ML models, and of course, in later sessions, we will delve deeper into each one of them, probably, or if not each one, then at least some of the usual ones. But classical ML models and time series analysis rely on structured feature engineering to transform the sequential nature of time series data into a tabular format suitable for all these algorithms.

So, let us say linear regression. Decision trees, random forests, gradient boosting models, k-nearest neighbors. So, so far, we have not discussed k-nearest neighbors. SVMs we discussed very briefly, right. K-means clustering techniques, naive Bayes.

So, by the way, under the ML umbrella or ML models, you have two different kinds of models, right. So, one is supervised, and the other is unsupervised. So, supervised means that if you have labeled data, It is called a supervised model, or unsupervised models are where the inputs are not labeled. So, you do not know which input belongs to which category.

So, clustering techniques are usually unsupervised ML models, among all the other ones. So, linear regression, decision trees, and random forests are supervised models because the inputs that you have or inputs that you feed the ML model or ML technique with are all labeled data sets. So, probably I will stop here for this session, and again we will pick up some of the widely applicable ones in the next session and then try to delve slightly deeper into each one of them. Let us say regression, decision trees, random forests—at least some of the well-known ones. Thank you.