HIS PROJECT - TSA

Task 09

Author
Alberto Mejia

Contents

1 Chapter 17: Multi-Step Forecasting

1.1 Why multi-step forecasting?

Most real-world applications of time series forecasting demand multi- step forecasting, whether it is the energy consumption of a household or the sales of a product. This is because forecasts are never created to know what will happen in the future, but to take some action using the visibility we get.

Despite a more prevalent use case, multi-step forecasting has not received the attention it deserves. One of the reasons for that is the existance of easy to implement classical statistical models or econometrics models such as *ARIMA* and *exponential smoothing* which include the multi-step strategy bundled within what we call a model.

Another reason for the low popularity of multi-step forecasting is that is simply harder to implement. The more steps we extrapolate into the future, the more uncertainty in the predictions.

1.2 Recursive strategy

The oldest strategy, most intuitive, and most popular technique to generate multi-step forecast. After training a model for one-step-ahead predictions, we employ it recursively to produce forecasts for H future timesteps. Initially, the model uses the window $\mathcal{W}(Y_T)$, which incorporates the most recent timestamp from the training data, to forecast the immediate next value, $\hat{y}_{T+1}$. This forecast, $\hat{y}_{T+1}$, is then integrated into the historical data, leading to a new window $\mathcal{W}(Y_T; \hat{y}_{T+1})$ that includes this forecast. The updated window serves as input for the model to predict the subsequent value, $\hat{y}_{T+2}$. This iterative process continues, extending the forecast horizon until we have obtained predictions for all H timesteps. This is the strategy that classical models like *ARIMA* and *exponential smoothing* use internally.

1.3 Direct strategy

Under the direct strategy, we train H distinct models, each designed to forecast a specific future timestep. These models utilize the same input window $\mathcal{W}(Y_t)$ but are independently trained to predict outcomes at different points within the forecast horizon. Consequently, this approach enables us to learn a unique set of parameters for each timestep, allowing the ensemble of models to establish a direct and independent mapping from the input window to each point in the forecast horizon H. This method has gained popularity alongside the rise of machine learning (ML)-based forecasting techniques. Practical implementations of this strategy within the ML context can be categorized into two methods:

- **Shifting Targets:** Involves training each model to forecast a future step by shifting the target output by the corresponding horizon length

- **Eliminating Features:** Involves training models by selectively using input features that are valid according to the forecasting rules, such as excluding certain lagged features that would not be available at the time of prediction.

NOTE: The two ways mentioned in the preceding list work nicely if we only have lags as features. For instance, for eliminating features, we can just drop the offending lags and train the model. But in cases where we are using rolling features and other more sophisticated features, simple dropping doesn't work because lag 1 is already used in calculating the rolling features. This leads to data leakage.

1.4 Joint strategy

In the realm of multi-step time series forecasting, traditional models typically generate a single output, adhering to a Multiple Input, Single Output (MISO) framework. However, advancements in Deep Learning (DL) allow the configuration of models to produce multiple outputs simultaneously, leading to the development of the Joint Strategy or Multiple Input, Multiple Output (MIMO). This approach trains a singular model to predict the entire forecast horizon in one operation.

Training Regime: The Joint Strategy involves the training of a multi-output model to forecast all timesteps within the horizon concurrently. Utilizing a window function, $\mathcal{W}(Y_t)$, the model is trained to predict a sequence of future values, $\hat{y}_{t+1}, \ldots, \hat{y}_{t+H}$, based on inputs drawn from the historical data. A loss function evaluates the model's predictions against actual future values, optimizing the model's parameters for accurate horizon-wide forecasting.

Forecasting Regime: Once trained, the model applies the same window function to new or unseen data, $\mathcal{W}(Y_T)$, to forecast the entire horizon in a single step. This capability is particularly leveraged in DL models by designing the last layer to output H scalars, representing the forecast for each future timestep, rather than a single output.

This strategy's utilization predominantly in DL models showcases its compatibility with the architectural flexibility of neural networks, enabling the simultaneous prediction of multiple future values, enhancing forecasting efficiency and coherence across the forecast horizon.

1.5 Hybrid strategies

Over the year there has been efforts to implement hybrid strategies which combine the advatanges of each of the strategies previously mentioned. Some of them are: *DirRec Strategy*, *Iterative block-wise strategy*, *Rectify Strategy*, and *RecJoint*. That being said, it is possible for each user to come up with their own strategy.

1.5.1 DirRec Strategy

As the name suggest, it is a combination of direct and recursive strategies.

- **Trainig Regime:** The DirRec strategy, standing for Direct-Recursive, innovatively blends aspects of both direct and recursive forecasting methodologies for multi-step forecasting over a horizon of H. Unlike conventional approaches that either forecast each step independently (Direct) or recursively use each forecast as input for the next (Recursive), DirRec employs H distinct models. Each model is tailored to forecast a specific future step, starting with $\mathcal{W}(Y_t)$ for the immediate next step. As a novel twist, models for steps beyond the first not only utilize the initial input window but also incorporate forecasts from all previous steps. Formally, for any given step $h < H$, the model inputs include $\mathcal{W}(Y_t)$ alongside all intermediary forecasts up to h, thereby enriching the predictive context. This methodical incorporation of preceding forecasts into each subsequent model's training regime aims to capture a more comprehensive temporal pattern, potentially enhancing the accuracy and robustness of the overall forecasting strategy.

- **Forecasting Regime:** Similar to Training regime, but instead of training the models, we use the H trained models to generate the forecast in a recursive manner.

1.5.2 Iterative block-wise strategy

Also called the **iterative multi-SVR strategy**, tries to tackle the shortcomings of the direct strategy, which requires H different models to train, making it difficult for long scale-horizon forecasting. This is achieved by using a block-wise iterative style of forecasting.

- **Trainig Regime:** Forecast horizon H is split into R blocks of lenght L such that $H = L \times R$. And instead of training H direct models, we train L direct models.

- **Forecasting Regime:** We use the L trained models to generate a forecasts for the initial L timesteps ($T + 1$ to $T + L$) within the forecast horizon H, employing the window $\mathcal{W}(Y_T)$. The first series of forecasts, denoted Y_{T+L}, combined with the dataset Y_T, is used to form a new window, $\mathcal{W}(Y_T; Y_{T+L})$. This newly formed window then serves to predict the next L timesteps ($T + L$ to $T + 2L$). This iterative approach is continued to cover the complete forecast horizon.

1.5.3 Rectify Strategy

It's another way to combine direct and recursive strategies by forming a two stage training and inferencing methodology. It can be seen as a model stacking approach but between different multi-step forecasting strategies.

- **Trainig Regime:** It is done in two steps. Initially, a recursive approach forecasts the entire horizon H, producing a preliminary set of predictions, $\tilde{Y}_{t+H}$. Following this, for each horizon, direct models are trained, harnessing both the authentic historical data Y_t and the recursively derived forecasts $\tilde{Y}_{t+H}$ as inputs.
- **Forecasting Regime:** Similar to the training, recursive forecasts are generated first and are used to generate the final forecast along the original history.

1.5.4 RecJoint

Is a mashup between the recursive and joint strategies, but applicable for single output models.

- **Trainig Regime:** It trains a single model to recursively predict future values, utilizing the forecast at time $t + 1$ as input for the subsequent prediction at $t + 2$, and so on. *RecJoint* generates forecasts for the entire horizon H, conducting a joint optimization of these forecasts during training. This forces the model to concurrently consider all H timesteps, thereby achieveing a comprehensive horizon-wide optimization. Notably implemented in Seq2Seq models with RNN encoders and decoders.
- **Forecasting Regime:** It is the same as for the recursive strategy.

1.6 How to choose a multi-step forecasting strategy

When choosing a which strategy to use it is important to take into consideration multiple perspectives, such as:

- **Engineering complexity:** Recursive, Joint, RecJoint << IBD << Direct, DirRec << Rectify
- **Training time:** Recursive << Joint (typically $T_{mo} > T_{so}$) << RecJoint << IBD << Direct, DirRec << Rectify
- **Inference time:** Joint << Direct, Recursive, DirRec, IBD, RecJoint << Rectify

It also helps us to decide the kind of model we can use for each strategy. For instance, a joint strategy can only be implemented with a model that supports multi-output, such as a DL model.

Some othe important guidelines to take into consideration when choosing the best multi-step forecasting strategy for each use case are the following:

- In the recursive forecasting strategy, error components such as bias and variance at step $h = 1$ directly influence the error at step $h = 2$, leading to an accumulation of errors as the forecast horizon extends. This phenomenon results in increasing error variance over time, a characteristic that was empirically confirmed through studies showing significant variance escalation in recursive forecasts. In contrast, the direct strategy lacks this step-wise error dependence, thereby avoiding the recursive approach's error accumulation and ensuring more consistent forecast quality over the horizon.

- The direct forecasting strategy ensures that error components like bias and variance at step $h = 1$ do not influence subsequent steps, as each h is forecasted independently. While this method avoids cumulative error, it may yield forecasts that lack coherence across the horizon, failing to account for potential dependencies between consecutive forecasts. This independence might lead to unrealistic predictions, particularly in time series characterized by non-linear trends, potentially resulting in a discontinuous forecast curve.
- Practically, in most cases, a direct strategy produces coherent forecasts.
- In recursive forecasting, models that generate highly variable forecasts can lead to an increased bias. This effect is particularly noticeable in complex models, which, despite their inherent low bias, introduce significant variability into forecasts. Such variability amplifies the bias within the recursive strategy, underscoring the nuanced relationship between model complexity, forecast variation, and bias amplification.
- In the context of large datasets, the bias term for the direct forecasting strategy approaches zero, contrasting with the recursive strategy, where bias remains nonzero. Empirical studies have shown that the direct strategy consistently outperforms the recursive strategy in scenarios involving long time series. This phenomenon is explained through learning theory, which posits that the direct strategy requires learning H distinct functions using the same dataset—a more complex task than learning a single function as in the recursive strategy. This difficulty is especially magnified in situations of data scarcity.
- Although the recursive strategy seems inferior to the direct strategy theoretically and empirically, it is not without some advantages:
  - For highly non-linear and noisy time series, learning direct functions for all the horizons can be hard. And in such situations, recursive can work better.
  - If the underlying **data-generating process (DGP)** is very smooth and can be easily approximated, the recursive strategy can work better.
  - When the time series is shorter, the recursive strategy can work better.
- The joint strategy mitigates the challenge of generating potentially unrelated forecasts across the forecast horizon—an issue inherent to the direct strategy. By evolving from the direct strategy's framework of H distinct

models for H forecasts, the joint strategy utilizes a single model to yield H outputs. This consolidation to learning a single function from the data circumvents the direct strategy's drawback in short time series, ensuring a more unified forecasting approach.

- A limitation of the joint (and RecJoint) strategy lies in its bias towards longer forecast horizons, particularly noticeable with very short horizons (e.g., $H = 2$, $H = 3$). This strategy involves optimizing a model to predict across all H timesteps using a unified loss function, such as mean squared error, which does not differentiate between the varying error scales across the horizon. As a result, the larger errors anticipated in distant forecasts lead to a model bias that favors accuracy in these later stages, potentially at the detriment of short-term forecast precision.
- Both the joint and RecJoint strategies show comparable variance in forecasting, yet the joint strategy often results in lower bias. This distinction arises because the joint strategy fully leverages the forecasting model's capabilities to directly predict the entire horizon. Conversely, the RecJoint strategy, which focuses on learning a recursive function, may not possess sufficient flexibility to discern intricate patterns, thus potentially yielding a higher bias.

Hybrid forecasting strategies, including DirRec and IBD, endeavor to synthesize the benefits while alleviating the shortcomings of the core methods—direct, recursive, and joint. This synthesis enables the creation of an informed experimentation framework, which is instrumental in determining the optimal strategy for a given forecasting challenge, thereby customizing the solution to enhance prediction performance.

1.7 Summary

This discourse delves into multi-step forecasting, a crucial yet underexplored facet of forecasting with significant real-world applicability. Initially, it underscores the imperative for multi-step forecasting, subsequently exploring prevalent methodologies encompassing direct, recursive, and joint strategies. The narrative progresses to evaluate hybrid models such as DirRec and rectify, culminating in a comprehensive assessment of these strategies' strengths and weaknesses. Conclusively, it furnishes pragmatic guidelines for the strategic selection of an optimal forecasting approach, tailored to address specific challenges.