Capstone Proposal

October 22, 2019

Machine Learning Engineer Nanodegree Timo Meiendresch

1 Domain Background

1.1 Time Series Forecasting and Machine Learning

Time series forecasting is a key challenge in business, economics, and many other areas. Improving forecast quality is of great advantage to make accurate business decisions.

In many academic areas, machine learning methods have already become part of the standard toolkit. Common examples include classification tasks, image processing, or text analysis. Yet, in the area of time series forecasting, machine learning methods are rarely considered.

The reason for this is that traditional methods often outperformed highly complex machine learning methods historically. A widely shared perception among researchers was that complex methods for time series forecasting were not performing better than traditional ones (e.g. Hyndman, 2019). Among others, Makridakis et al. (2018) noted that there is only limited scientific evidence which suggests that neural networks for time series forecasting are an essential tool for time series forecasting.

1.2 Paradigm Shift

But, advances in the recent years seem to challenge this notion. For example, in the recognized M4 forecasting competition a hybrid model based on a **recurrent neural network (RNN)** outperformed all other approaches and, thus, showed the potential of RNN-based methods (Smyl, 2019, Makridakis et al., 2019).

Moreover, the M4 competition highlights an ongoing paradigm shift in the forecasting community. Traditional methods, in particular **ARIMA** or **exponential smoothing methods**, are applied to individual time series. These **local** methods estimate a number of parameters within the limited model space and are referred to as **model-based** (Wang et al., 2019). These methods focus on independent models for individual time series.

As the availability of large sets of related data increasead, a new type of forecasting problem emerged. Instead of forecasting a single seies independently, it is often beneficial to forecast big collections of related series. Examples of these type of data can be found in diverse areas, such as web traffic, household electricity consumption, or product demand of online retailers.

In contrast to local models, RNN-based methods enable the use of **cross-series learning**, i.e. the training process can use all available series, yielding a **global** representation of the data. This approach uses possible dependencies between series which may improve forecasting accuracy.

The conditions under which these models actually perform better than local ones is a crucial part of this project.

In the aftermath of the M4 competition, various RNN-based methods have been published that are based on the idea of *cross-series learning*, sometimes combining local methods with recurrent networks. In this project I will focus on three recently developed RNN-based methods:

- **DeepAR** - Salinas et al. (2017)
- **DeepFactor** - Wang et al. (2019)
- **DeepState** - Rangapuram et al. (2018)

To the best of my knowledge no results of the accuracy of these algorithms applied to the M4 data have been published yet with the exception of the winning method ES-RNN (Smyl, 2019).

1.3 Personal motivation

My personal motivation to carry out this project is my background in Statistics & Econometrics, in particular time series analysis. During my studies I covered the traditional methods, such as ARIMA, exponential smoothing methods and various other exotic models. The superiority of local methods was never questioned by the lecturer and machine learning models for time series forecasting was not covered at all. Given my interest and background, I followed the developments and the M4 competition closely and take this project as a chance to keep up with current research, recently developed methods, as well as new frameworks for time series forecasting (i.e. GluonTS).

2 Problem Statement

Aforementioned algorithms have not been applied to the M4 data. For practitioner's and researchers this would be a valuable insight into how accurate these methods perform in comparison to traditional, **local** methods as well as in comparison to the ranked approaches.

2.1 Problem area 1 - RNN-based algorithms in practice:

- How accurate are the three RNN-based algorithms on the M4 data?
- Are the RNN-based algorithms applicable in *real-world* forecasting scenarios?
- What is their relative performance compared to benchmark methods used in the M4 competition?

Moreover, the M4 competition data were designed for the purpose of resembling "real-world" forecasting practice. Hence, we can we can use these data to answer some of the many questions regarding the RNN-based algorithms in practice.

2.2 Problem area 2 (possible extension of the project if there is enough time left):*

- Are there significant performance differences across *frequencies* (Yearly, Quarterly, Monthly, Weekly, Daily, Hourly) or *domain* (Micro, Industry, Macro, Finance, Demographic, Other)?

Another problem area concerns the requirements data requirements. The aforementioned papers vaguely addresses this by describing the dataset to consist of **large** and **related** series (for example Salinas et al., 2017; Wang et al., 2019).

2.3 Problem area 3 - Size and relatedness requirements (possible extension of the project if there is enough time left):*

- Do models that are trained on larger series of the same frequency perform better? Possibele experimental Design: Using random subsets of N={100, 500, 1000, n_i}
- Do models that are trained on the same domain perform better compared to models that are trained cross-sectoral (keeping N constant and vary domain/cross-domain)?

Here, "better" refers to "more accurate" according to the accuracy metrics outlined in the competition.

Please note that answering all these questions is well beyond the extent of this project. I will therefore focus on problem area 1 and proceed if enough time is left to work on the other problem areas. Also, Spiliotis et al. (2019) argues that a subset of 1,000 series should be sufficient to reach similar conclusions about the data. Hence, I will use smaller subsets of the M4 data to alleviate time and computational capacity restrictions.

2.4 A potential solution - Quantifiable, measurable, and replicable

A potential solution presents the results of the three **global** algorithms using the M4 data and compares them to the **local** benchmark methods.

Performance will be measured by comparing estimated forecasts with the realized observations (ground truth) using evaluation metrics of the competition. These metrics will be described in a later section.

All code and datasets will be provided on github making this project fully replicable.

3 Datasets and Inputs

The M4 competition data (Makridakis et al., 2019) contains 100,000 real-world time series with a frequency-specific lower limit of available observations. Also, the data is divided according to six domains (Economic, Finance, Demographics, Industry, and Other), as well as by frequency (yearly, quarterly, monthly, weekly, daily, and hourly).

Data are publicly available on github and can also be used with the respective R package or the GluonTS API for Python. They are separated by type (train or test data), where the test data length is equal to the frequency-specific forecasting horizon. In the M4 competition the forecasting horizons were given as follows:

- Yearly (frequency) - 6 (forecast horizon)
- Quarterly - 8
- Monthly - 18
- Weekly - 13
- Daily - 14
- Hourly - 48

According to Spiliotis et al. (2019) the M4 data is diverse and the closest to what can be perceived as "real-world" among a wide variety of competition datasets. Moreover, results suggest that random samples of 1,000 series could be enough to resemble the overall feature space of the entire dataset. Accordingly, due to computational resource limitations, I will restrict myself in this project to a subset of 1,000 series per domain or frequency at maximum.

4 Solution statement

The main problem is to indicate whether aforementioned RNN-based algorithms are useful in a *real-world* forecasting scenario. As a proxy for *real-world* the M4 competition data are used with the established reasoning of Spiliotis et al. (2019).

The project quantifies the performances of each algorithm using three evaluation metrics:

- Symmetric mean absolute percentage error (sMAPE)
- Mean absolute scaled error (MASE)
- Overall weighted average (OWA)

Accuracy performance can be quantified using these metrics. Replicability will be secured by making the code available on github.

A solution delivers basic takeaways regarding the applicability of the algorithms and accuracy results in comparison to widely known benchmark methods.

Extension: Additionally, the solution presents how size, relatedness, and frequency affects the performance (for small samples).

5 Benchmark models

The M4 competition used a wide variety of benchmark methods. For this project I will limit myself to the main one and add two other widely used methods.

- Comb benchmark method of the M4 competition - Arithmetic average of simple exponential smoothing, Holt method, and damped Holt method
- Auto ARIMA - automatic framework for ARIMA methods
- ETS - Automatic framework for exponential smoothing methods

ARIMA and ETS are among the most widely known and used **local** methods in time series forecasting.

6 Evaluation metrics

For comparability with the competition results I will use the same evaluation metrics that were used during the competition.

Ranks in the competition were determined by the overall weighted average (OWA) which is a composite measure of the symmetric mean absolute percentage error (sMAPE) and mean absolute scaled error (MASE). The formulas are described in Makridakis et al. (2019).

7 Project Design

The workflow will be focused around the interaction between the GluonTS API in Python and using the computational power of AWS instances (with GPU). Gluon Time Series (GluonTS) toolkit for probabilistic time series modeling is based on the Apache MXNet deep learning framework. This project requires to extend the knowledge on model deployment using PyTorch and SageMaker to working with MXNet on an AWS instance to train the models on a GPU instance.

The algorithms are built-in algorithms of GluonTS and/or AWS SageMaker. Data preprocessing will be a crucial part of the project as these algorithms require the data in a specific format. Based on Spiliotis et al. (2019) and computational complexity of these models I will use subsets of the data with an upper limit of 1,000 time series per experiment. Therefore, I need to preprocess the data to randomly choose 1,000 series (setting seeds). Moreover, the data needs to be divided by frequency and domain for problem areas 2 and 3.

Lastly, the estimates have to be translated into the secondary measures MASE and sMAPE from which the OWA can be calculated.

- Step 1 (Data processing) - Preprocess the data in aforementioned way
- Step 2 (Data modeling) - Use GluonTS and AWS SageMaker to train the models on the subsets of the M4 data and get forecast estimations
- Step 3 (Inference) - Evaluate the forecasts using the outlined accuracy measures.

8 References

- Hyndman, Rob J. "A brief history of forecasting competitions." *International Journal of Forecasting (2019).*
- Rangapuram, Syama Sundar, et al. "Deep state space models for time series forecasting." *Advances in Neural Information Processing Systems. 2018.*
- Salinas, David, Valentin Flunkert, and Jan Gasthaus. "DeepAR: Probabilistic forecasting with autoregressive recurrent networks." *arXiv preprint arXiv:1704.04110 (2017).*
- Smyl, Slawek. "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting." *International Journal of Forecasting (2019).*
- Spiliotis, Evangelos, et al. "Are forecasting competitions data representative of the reality?." *International Journal of Forecasting (2019).*
- Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "Statistical and Machine Learning forecasting methods: Concerns and ways forward." *PloS one 13.3 (2018): e0194889.*
- Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M4 competition: 100,000 time series and 61 forecasting methods." *International Journal of Forecasting (2019).*
- Wang, Yuyang, et al. "Deep Factors for Forecasting." *arXiv preprint arXiv:1905.12417 (2019).*