Deep Factors with Gaussian Processes for Forecasting

Danielle C. Maddix
Amazon Web Services
Palo Alto, CA 94303
dmmaddix@amazon.com

Yuyang Wang
Amazon Web Services
Palo Alto, CA 94303
yuyawang@amazon.com

Alex Smola
Amazon Web Services
Palo Alto, CA 94303
smola@amazon.com

Abstract

A large collection of time series poses significant challenges for classical and neural forecasting approaches. Classical time series models fail to fit data well and to scale to large problems, but succeed at providing uncertainty estimates. The converse is true for deep neural networks. In this paper, we propose a hybrid model that incorporates the benefits of both approaches. Our new method is data-driven and scalable via a latent, global, deep component. It also handles uncertainty through a local classical Gaussian Process model. Our experiments demonstrate that our method obtains higher accuracy than state-of-the-art methods.

1 Introduction

Some prevalent forecasting methods in statistics and econometrics have been developed for forecasting individual or small groups of time series. These methods consist of complex models designed and tuned by domain experts [1]. Recently, there has been a paradigm shift from model-based to fully-automated data-driven approaches. This shift can be attributed to the availability of large and diverse time series datasets in a wide variety of fields [2]. A substantial amount of data consisting of past behavior of related time series can be leveraged for making a forecast for an individual time series. Use of data from related time series allows for fitting of more complex and potentially more accurate models without overfitting.

Classical time series methods, such as Autoregressive Integrated Moving Average (ARIMA) [3], exponential smoothing [4] and general Bayesian time series [5], excel at modeling the complex dynamics of individual time series of sufficiently long history. These methods are computationally efficient, e.g. via a Kalman filter, and provide uncertainty estimates. Uncertainty estimates are critical for optimal downstream decision making. These methods are local, that is, they learn one model per time series. As a consequence, they cannot effectively extract information across multiple time series. These classical methods also have challenges with cold-start problems, where more time series are added or removed over time.

Deep neural networks (DNNs), in particular, recurrent neural networks (RNNs), such as LSTMs [6] have been successful in time series forecasting [7, 8]. DNNs are generally effective at extracting patterns across multiple time series. Without a combination with probabilistic methods, such as variational dropout [9] and deep Kalman filters [10], DNNs can be prone to overfitting and have challenges in modeling uncertainty [11].

The combination of probabilistic graphical models with deep neural networks has been an active research area recently [10, 12, 13, 14]. In the time series forecasting domain, a recent example is [15], where the authors combine RNNs and State-Space Models (SSM) for scalable time series forecasting. Our work in this paper follows a similar theme: we propose a novel and scalable global-local method, Deep Factors with Gaussian Processes. It is based on a global DNN backbone and local Gaussian Process (GP) model for computational efficiency. The global-local structure extracts complex non-linear patterns globally while capturing individual random effects for each time series locally. The

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

main idea of our approach is to represent each time series as a combination of a global time series and a corresponding local model. The global part is given by a linear combination of a set of deep dynamic factors, where the loading is temporally determined by attentions. The local model is a stochastic Gaussian Process (GP), which allows for the uncertainty to propagate forward in time.

2 Deep Factor Model with Gaussian Processes

We first define the forecasting problem that we are aiming to solve. Let $\mathcal{X} \subset \mathbb{R}^d$ denote the input features space and $\mathcal{Z} \subset \mathbb{R}^k$ the space of the observations. We are given a set of N time series with the i^{th} time series consisting of $(x_{i,t}, z_{i,t}) \in \mathcal{X} \times \mathcal{Z}, t = 1, \cdots, T$, where $x_{i,t}$ are the input co-variates, and $z_{i,t}$ is the corresponding observation at time t. Given a forecast horizon $\tau \in \mathbb{N}^+$, our goal is to calculate the joint predictive distribution of future observations,

$$p(\{z_{i,T+1:T+\tau}\}_{i=1}^N | \{x_{i,T+1:T+\tau}\}_{i=1}^N, \{\mathcal{D}_i\}_{i=1}^N),$$

where $\mathcal{D}_i = \{(x_i, z_i)\}$ denotes the i^{th} time series with corresponding features. For concreteness, we restrict ourselves to univariate time series ($k = 1$).

2.1 Generative Model

We assume that each time series z_i is governed by the following two components: fixed and random.

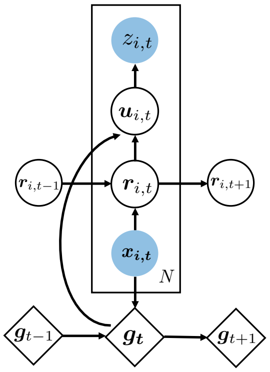


Figure 1: Plate graph of the proposed Deep Factors with Gaussian Processes model. The diamond nodes represent deterministic states.

Fixed effects are common patterns that are given by linear combinations of K latent global deep factors, $g_{k,t}$. These deep factors can be thought of as dynamic principal components or eigen time series that drive the underlying dynamics of all the time series.

Random effects, r_i, are the local fluctuations that are chosen to be the Gaussian Process [16], i.e., $r_i \sim \mathbf{GP}(0, \mathcal{K}_i(\cdot, \cdot))$, where the covariance $\mathcal{K}_i$ is a kernel matrix and $r_{i,t} = r_i(x_{i,t})$.

The observed value $z_{i,t}$ at time t, or more generally, its latent function $u_{i,t}$ such that $z_{i,t} \sim p(\cdot | u_{i,t}(x_{i,t}))$, can be expressed as a sum of the weighted average of the global patterns and its local fluctuations. The summary of this generative model is given in Eqn. (1), and is illustrated in Figure 1. For simplicity, we consider $w_i(x_{i,t}) := w_i$ to be the embedding of time series i.

$$\begin{aligned} \text{random effect}: \quad & r_i \sim \mathbf{GP}(0, \mathcal{K}_i(\cdot, \cdot)), \\ \text{fixed effect}: \quad & f_{i,t} = w_i^\top g_t(x_{i,t}), \\ \text{emission}: \quad & z_{i,t} \sim p(\cdot | u_{i,t}), \ u_{i,t} = f_{i,t} + r_{i,t}. \end{aligned} \tag{1}$$

We use a global dynamics factors RNN or a set of K univariate-valued RNNs to generate $g_t \in \mathbb{R}^K$. The RNNs are learned globally to capture the common patterns from all time series. For each time series at time t, we use attention networks to assign stationary attentions $w_i \in \mathbb{R}^K$ to the dynamic factors g_t. This determines the group of the global factors to focus on and the relevant segment of histories. At a high level, the weighting gives temporal attention to different global factors.

2.2 Inference and Learning

Given a set of N time series generated by Eqn. (1), our goal is to estimate $\mathbf{\Theta}$, the parameters in the global RNNs, attention network and the hyperparameters in the kernel function. To do so, we use maximum likelihood estimation, where $\mathbf{\Theta} = \text{argmax} \sum_i \log p(z_i)$, Computing the marginal likelihood may require doing inference over the latent variables. In our case, $p(\cdot | u_{i,t})$ is Gaussian, and the marginal likelihood can be computed easily as,

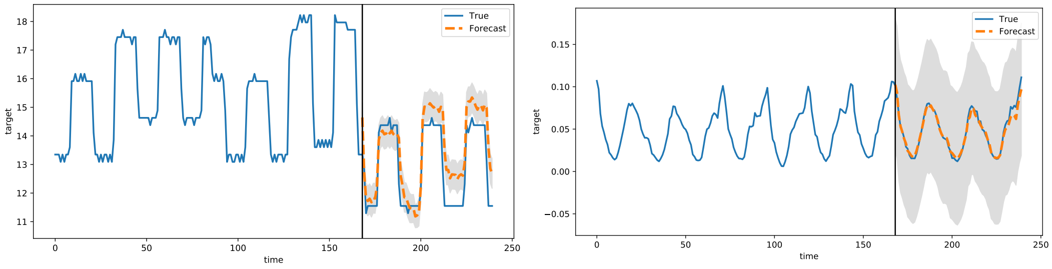
$$p(z_i) = \mathcal{N}(f_i, \mathcal{K}_i + \sigma_i^2 \mathbb{I}).$$

For non-Gaussian likelihoods, classical techniques, such as Box-Cox transform [17] or variational inference in the framework of Variational Auto Encoder (VAE) [18, 19], can be used. This is a direction of future work.

3 Experiments

The model is implemented in MXNet Gluon [20] with a RBF kernel [23] using the `mxnet.linalg` library [21, 22]. We use a p3.4xlarge SageMaker instance in all our experiments. The global factor network is chosen to be LSTM with 1 hidden layer and 50 hidden units. We fix the number of factors to be 10.

To assess the quality of the proposed model, we limit the training, sometimes artificially by pruning the data, to only one week of time series. This results in 168 observations per time series. Figures 2a-2b show that the forecasts qualitatively on the publicly available datasets `electricity` and `traffic` from the UCI data set [24, 25].



(a) `electricity`

(b) `traffic`

Figure 2: The dashed orange curve shows the forecast of the proposed global LSTM with GP local model. The black vertical line marks the division between the training and prediction regions.

We use the quantile loss to evaluate the probabilistic forecast. For a given quantile $\rho \in (0, 1)$, a target value z_t and ρ-quantile prediction $\widehat{z}_t(\rho)$, the ρ-quantile loss is defined as

$$\mathrm{QL}_\rho[z_t, \widehat{z}_t(\rho)] = 2\big[\rho(z_t - \widehat{z}_t(\rho))\mathbb{I}_{z_t - \widehat{z}_t(\rho) > 0} + (1-\rho)(\widehat{z}_t(\rho) - z_t)\mathbb{I}_{z_t - \widehat{z}_t(\rho) \leqslant 0}\big].$$

We use a normalized sum of quantile losses, $\sum_{i,t} \mathrm{QL}_\rho[z_{i,t}, \widehat{z}_{i,t}(\rho)] / \sum_{i,t} |z_{i,t}|$, to compute the quantile losses for a given span across all time series. We include results for $\rho = 0.5, 0.9$, which we abbreviate as the P50QL (mean absolute percentage error (MAPE)) and P90QL, respectively. We also report the root mean square error (RMSE), which is the square root of the aggregated squared error normalized by the product of number of time series and the length of the time series in the evaluation segment.

Table 1 compares with DeepAR (DA), a state-of-art RNN-based forecasting algorithm on the publicly available AWS SageMaker [7, 26] and Prophet (P), a Bayesian structural time series model [27]. To ensure a fair comparison, we set DeepAR to have the same 1-layer 50 hidden units network configuration, with the number of epochs set to be 2000. The results show that our model outperforms the others, in particular with respect to the P90 quantile loss. This shows that we are better at capturing uncertainty.

DS	HRZN	P50QL			P90QL			RMSE		
		DA	P	DFGP	DA	P	DFGP	DA	P	DFGP
`elec`	3d	0.216	0.149	**0.109**	0.182	0.103	**0.061**	1194.421	902.724	**745.175**
	24hr	0.132	0.124	**0.103**	0.100	0.091	**0.074**	2100.927	783.598	**454.307**
`traf`	3d	0.348	0.457	**0.137**	0.162	0.207	**0.093**	0.028	0.032	**0.021**
	24hr	0.268	0.380	**0.131**	0.149	0.191	**0.090**	0.024	0.028	**0.019**

Table 1: Results for short-term (3-day forecast) and near-term (24-hour forecast) scenario with one week of training data on `electricity, traffic`.

4 Conclusion

We propose a novel global-local model, Deep Factors with Gaussian Processes, for forecasting a collection of related time series. Our method differs from other global-local models by combining classical Bayesian probabilistic models with deep learning techniques that scale. We show promising experiments that demonstrate the effectiveness and potential of our method in learning across multi-time series and propagating uncertainty.

References

[1] Andrew C Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge university press, 1990.

[2] Matthias W Seeger, David Salinas, and Valentin Flunkert. Bayesian intermittent demand forecasting for large inventories. In *Advances in Neural Information Processing Systems*, pages 4646–4654, 2016.

[3] Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer Science & Business Media, 2013.

[4] Rob Hyndman, Anne B Koehler, J Keith Ord, and Ralph D Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

[5] David Barber, A Taylan Cemgil, and Silvia Chiappa. *Bayesian time series models*. Cambridge University Press, 2011.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[7] Valentin Flunkert, David Salinas, and Jan Gasthaus. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *arXiv preprint arXiv:1704.04110*, 2017.

[8] Ruofeng Wen, Kari Torkkola, and Balakrishnan Narayanaswamy. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.

[9] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. *arXiv preprint arXiv:1512.05287*, 2016.

[10] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[11] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International Conference on Machine Learning*, pages 1690–1699, 2018.

[12] Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *AAAI*, pages 2101–2109, 2017.

[13] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in neural information processing systems*, pages 2199–2207, 2016.

[14] Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in Neural Information Processing Systems*, pages 3604–3613, 2017.

[15] Syama Sundar Rangapuram, Matthias Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In *Advances in Neural Information Processing Systems*, 2018.

[16] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian process for machine learning*. MIT press, 2006.

[17] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

[19] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

[20] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*, 2015.

[21] Matthias Seegar, Asmus Hetzel, Zhenwen Dai, Eric Meissner, and Neil D. Lawrence. Auto-differentiating linear algebra. *arXiv preprint arXiv:1710.08717*, 2017.

[22] Dai Zhenwen, Eric Meissner, and Neil D. Lawrence. Mxfusion: A modular deep probabilistic programming library. *NIPS 2018 Workshop MLOSS*, 2018.

[23] Gardner J.R., Pleiss G., Bindel D., Weinberger K.Q., and Wilson A.G. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *32nd Conference on Neural Infromation Processing Systems (NIPS 2018) arXiv:1809.11165v2*, 2018.

[24] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2017. University of California, Irvine, School of Information and Computer Sciences.

[25] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in neural information processing systems*, pages 847–855, 2016.

[26] Tim Januschowski, David Arpin, David Salinas, Valentin Flunkert, Jan Gasthaus, Lorenzo Stella, and Paul Vazquez. Now available in amazon sagemaker: Deepar algorithm for more accurate time series forecasting. *https://aws.amazon.com/blogs/machine-learning/now-available-in-amazon-sagemaker-deepar-algorithm-for-more-accurate-time-series-forecasting/*, 2018.

[27] Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 2017.