Deep State Spaces Models for Time Series Forecasting
ESC

그

수리남 (김민준, 김용진, 신동빈)

Department of Mathematics
Yonsei University
Seoul 03722
Korea, Republic of

November 27, 2025

수리남 **(Yonsei University)**   **Deep SSMs**   **November 27, 2025   1 / 26**

Introduction

- Main reference: [Rangapuram et al., 2018]
- SSM operates well when the structure of the time series is well known. This requires that the size of the series be large. Also, SSM cannot infer patterns of similar time series.
- The feasible alternative is the deep learning model because it is able to infer complex patterns and high-dimensional data sets. But it has an issue called black-box problems, requiring large time series and making it hard to add assumptions.
- By fusing SSM and RNN(multi-layer LSTM), we can effectively solve this problem well.

Backgrounds

- Let $\left(z_{1:T_i}^{(i)}\right)_{i=1}^N$ be the target time series, where

$$z_{1:T_i} = \left(z_1^{(i)}, z_2^{(i)}, \cdots, z_{T_i}^{(i)}\right)$$

  and $z_t^{(i)}$ denotes the value of ith time series at time t.

- Let $\left\{\mathbf{x}_{1:T_i+\tau}^{(i)}\right\}_{i=1}^N$ be a set of associated covariate vectors with $\mathbf{x}_t^{(i)} \in \mathbb{R}^D$.

- Our goal is to estimate the state

$$p\left(z_{T_i+1:T_i+\tau}^{(i)} \,\middle|\, z_{1:T_i}^{(i)}, \mathbf{x}_{1:T_i+\tau}^{(i)}; \Phi\right).$$

To estimate

$$p\left(z_{T_i+1:T_i+\tau}^{(i)} \,\middle|\, z_{1:T_i}^{(i)}, \mathbf{x}_{1:T_i+\tau}^{(i)}; \Phi\right),$$

we refer:

- $z_{1:T_i}^{(i)}$: target time series
- $\{1, 2, \cdots, T_i\}$: training range
- $\{T_i + 1, T_i + 2, \ldots, T_i + \tau\}$: prediction range
- $T_i + 1$: forecast start time
- $\tau \in N$: forecast horizon

Note that we assume that the covariate vectors $\mathbf{x}_t^{(i)}$ are given also in the prediction range.

Backgrounds

- We consider the *linear state space models* where

$$\mathbf{l}_t = F_t \mathbf{l}_{t-1} + g_t \varepsilon_t \quad \text{with} \quad \mathbf{l}_0 \sim N\left(\boldsymbol{\mu}_0, \operatorname{diag}\left(\sigma_0^2\right)\right) \quad \varepsilon_t \sim N(0,1)$$

and the univariate Gaussian observation model

$$z_t = y_t + \sigma_t \varepsilon_t, \quad y_t = a_t^\top \mathbf{l}_{t-1} + b_t, \quad \varepsilon_t \sim N(0,1).$$

Backgrounds

- Note that the state space model is fully described by parameters

$$\Theta_t = \Theta = (F_t, g_t, \boldsymbol{\mu}_0, \boldsymbol{\sigma}_0, a_t, \boldsymbol{\sigma}_t).$$

- In *classical* way, we estimate Θ_t by

$$\Theta^*_{1:T} = \arg\max_{\Theta_{1:T}} p_{SS}(z_{1:T}\Theta_{1:T})$$

where

$$
\begin{aligned}
p_{SS}(z_{1:T} \,|\, \Theta_{1:T}) &= p(z_1 \mid \Theta_1) \prod_{t=2}^{T} p\left(z_t \mid z_{1:t-1}, \Theta_{1:t}\right) \\
&= \int p(\mathbf{l}_0 \mid \Theta_1) \left[\prod_{t=1}^{T} p(z_t \mid \mathbf{l}_t, \Theta_t) p(\mathbf{l}_t \mid \mathbf{l}_{t-1}, \Theta_t) \right] d\mathbf{l}_{0:T}
\end{aligned}
$$

Deep State Space Models: Structure

- The mapping

$$\Theta_t^{(i)} = \Psi(\mathbf{x}_{1:t}^{(i)}, \Phi)$$

is a function of $\mathbf{x}_{1:t}^{(i)}$ and a set of shared parameters Φ.
We parametrise the map Ψ using a deep recurrent neural network(RNN).

- Via a recurrent function h, we represent the features by

$$\mathbf{h}_t^{(i)} = h(\mathbf{h}_{t-1}^{(i)}, \mathbf{x}_t^{(i)}, \Phi.)$$

By $\mathbf{h}_t^{(i)}$, we directly compute our desired values by

$$\mathbf{h}_t^{(i)} \overset{\Psi \circ \mathbf{h}^{-1}}{\longmapsto} \Theta_t^{(i)}$$

without using $\mathbf{x}$.

Deep State Space Models: Training
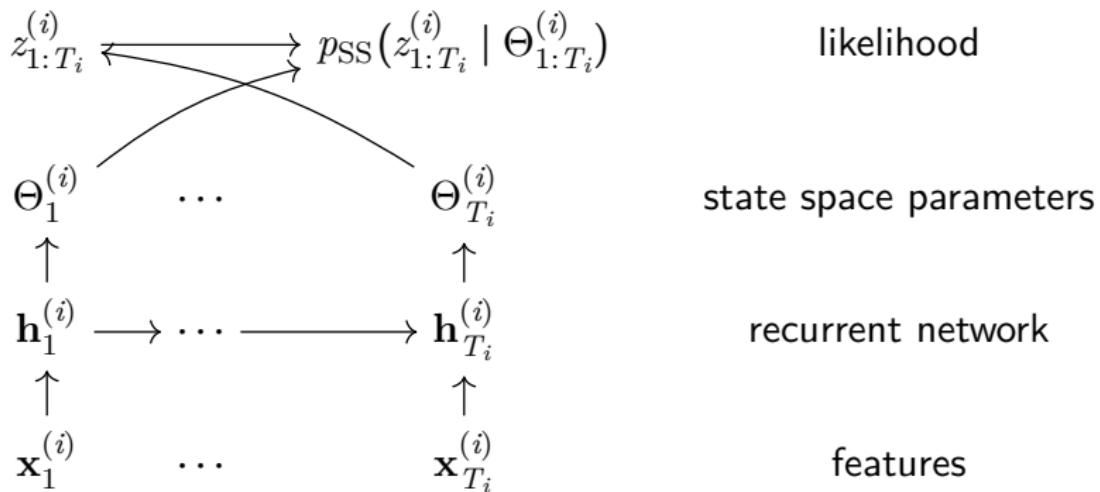
- The model parameters Φ are learned by maximising the log-likelihood, that is,

$$\Phi^* = \arg\max_{\Phi} \mathcal{L}(\Phi)$$

where

$$\mathcal{L}(\Phi) = \sum_{i=1}^{N} \log p\left(z_{1:T_i}^{(i)} \,\middle|\, \mathbf{x}_{1:T_i}^{(i)}, \Phi \right) = \sum_{i=1}^{N} \log p\left(z_{1:T_i}^{(i)} \,\middle|\, \Theta_{1:T_i}^{(i)} \right).$$

Deep State Space Models: Training

$$z_{1:T_i}^{(i)} \xleftrightarrow{\qquad} p_{\mathrm{SS}}(z_{1:T_i}^{(i)} \mid \Theta_{1:T_i}^{(i)}) \qquad \text{likelihood}$$

$$\Theta_1^{(i)} \qquad \cdots \qquad \Theta_{T_i}^{(i)} \qquad \text{state space parameters}$$

$$\uparrow \qquad\qquad\qquad \uparrow$$

$$\mathbf{h}_1^{(i)} \longrightarrow \cdots \longrightarrow \mathbf{h}_{T_i}^{(i)} \qquad \text{recurrent network}$$

$$\uparrow \qquad\qquad\qquad \uparrow$$

$$\mathbf{x}_1^{(i)} \qquad \cdots \qquad \mathbf{x}_{T_i}^{(i)} \qquad \text{features}$$

Deep State Space Models: Prediction

- object: prediction for 1115 independent stores
- covariate: num of customers, DayofWeeks, state holiday, promo, school holiday
- optimization: adam
- Set $F_t = F = I$
- performing prediction using median of 32 samples drawn from a 200-dimensional pdf via MCMC

Non-Gaussian Likelihoods

- Recall that we have used the the univariate Gaussian observation model

$$z_t = y_t + \sigma_t \varepsilon_t, \quad y_t = a_t^\top \mathbf{l}_{t-1} + b_t, \quad \varepsilon_t \sim N(0, 1).$$

- Now we extend above to

$$z_t \sim \mathbb{P}(\cdot \mid u_t), \quad u_t = y_t + \sigma_t \varepsilon_t.$$

- We optimise the marginal likelihood of $z_{1:T}$ by

$$
\begin{aligned}
\log p(z_{1:T}) &= \log \int p(z_{1:T}, \mathbf{u}, \mathbf{l}) \\
&\geq \mathbb{E}_{q_\phi(\mathbf{u}, \mathbf{l})} \log \left[\frac{p(z_{1:T}, \mathbf{u}, \mathbf{l})}{q_\phi(\mathbf{u}, \mathbf{l})} \right] \\
&\approx \frac{1}{L} \left(\log \left[\frac{p(z_{1:T}, \tilde{\mathbf{u}}_j, \mathbf{l})}{q_\phi(\tilde{\mathbf{u}}_j)} \right] + \log p(\tilde{\mathbf{u}}_j) \right)
\end{aligned}
$$

where $\tilde{\mathbf{u}}_j \sim q_\phi(\mathbf{u})$.

- Read [Rangapuram et al., 2018] for the details.

Deep SSM with Attention

- We integrated a self-attention mechanism into the output of the standard DeepSSM framework.
- We modeled the attention mechanism by linearly projecting the LSTM outputs to generate Queries, Keys, and Values.
- Future work: Indeed $\exists$ blackbox since we have not used the transformer.

Deep SSM

- We denote

$$(\# \text{ of RNN layers}, \# \text{ of samples}, \# \text{ of hidden units}).$$

- Root mean squared error, abbreviated by RMSE is defined as $\text{RMSE} = \sqrt{\text{MSE}}$.

- $(2, 16, 512)$: $\text{RMSE} = 8659$
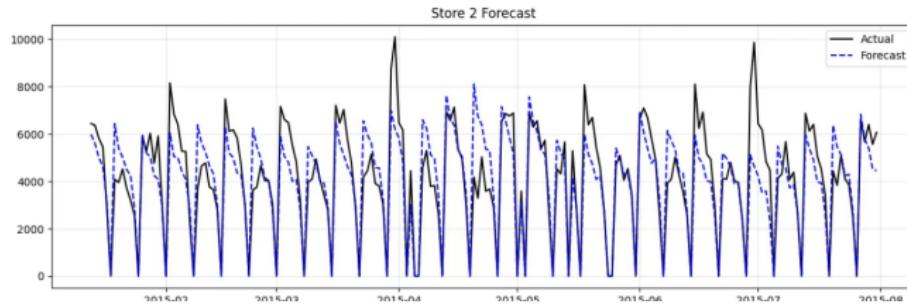
Figure: 2, 16, 512

Deep SSM

- $(2, 32, 2048)$: RMSE $= 5936$
- $(3, 32, 384)$ RMSE $= 17962$
- $(2, 32, 768)$: RMSE $= 3300$
- $(2, 32, 768)$ again: RMSE $= 378262$
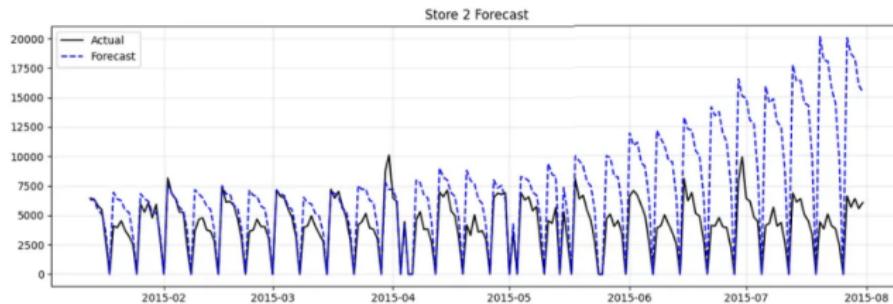
Figure: 3, 32, 384

Deep SSM

- $(2, 32, 512)$ (0s removed): RMSE $= 21806$
- $(2, 32, 384)$ (0s and trends removed): RMSE $= 2258$
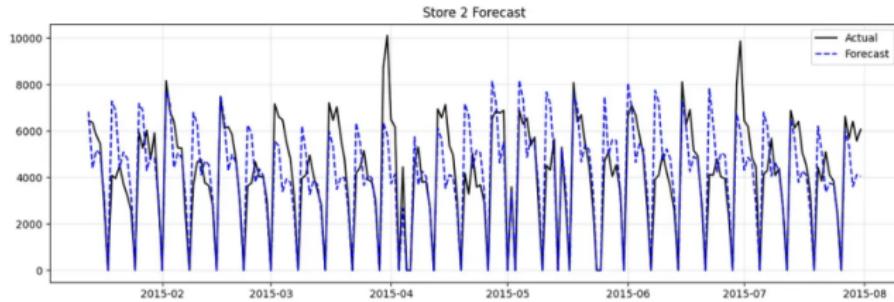
Figure: 2, 32, 384

Deep SSM with Non-Gaussian likelihoods

> **Recall**
>
> Note that we reconstructed
>
> $$z_t \sim \mathbb{P}(\cdot \mid u_t), \quad u_t = y_t + \sigma_t \varepsilon_t.$$
>
> and observed
>
> $$\log p(z_{1:T}) \approx \frac{1}{L} \left(\log \left[\frac{p(z_{1:T}, \tilde{\mathbf{u}}_j, \mathbf{l})}{q_\phi(\tilde{\mathbf{u}}_j)} \right] + \log p(\tilde{\mathbf{u}}_j) \right).$$

- We employ the Poisson distribution for $\mathbb{P}$ and Gaussian distribution for q_ϕ.
- Why?

Deep SSM with Non-Gaussian likelihoods

- $(2, 32, 512)$: RMSE $= 6518$
- A significant decrease in the computational time occurred.
- Why?
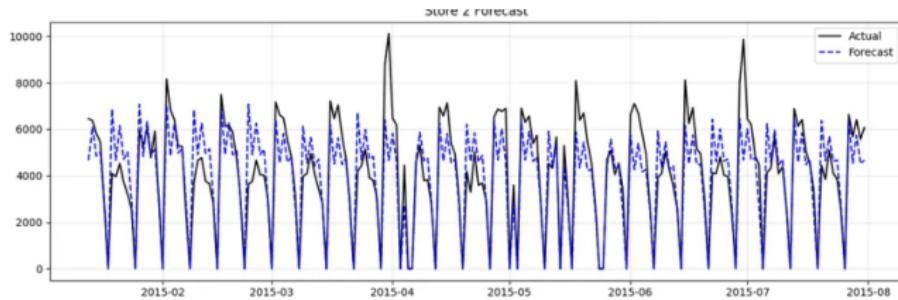
Figure: 2, 32, 512

Deep SSM with Attention

- $(2, 32, 256, 4)$ (0s removed): RMSE $= 2733$
- Why?
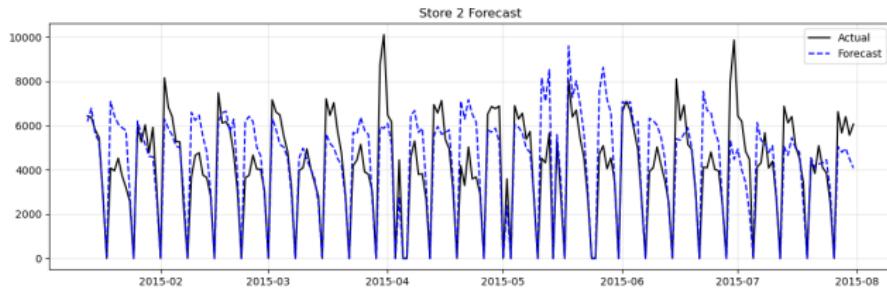
Figure: 2, 32, 256, 4

References

Rangapuram, S. S., Seeger, M. W., Gasthaus, J., Stella, L., Wang, Y., and Januschowski, T. (2018).

Deep state space models for time series forecasting.

In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Thank You