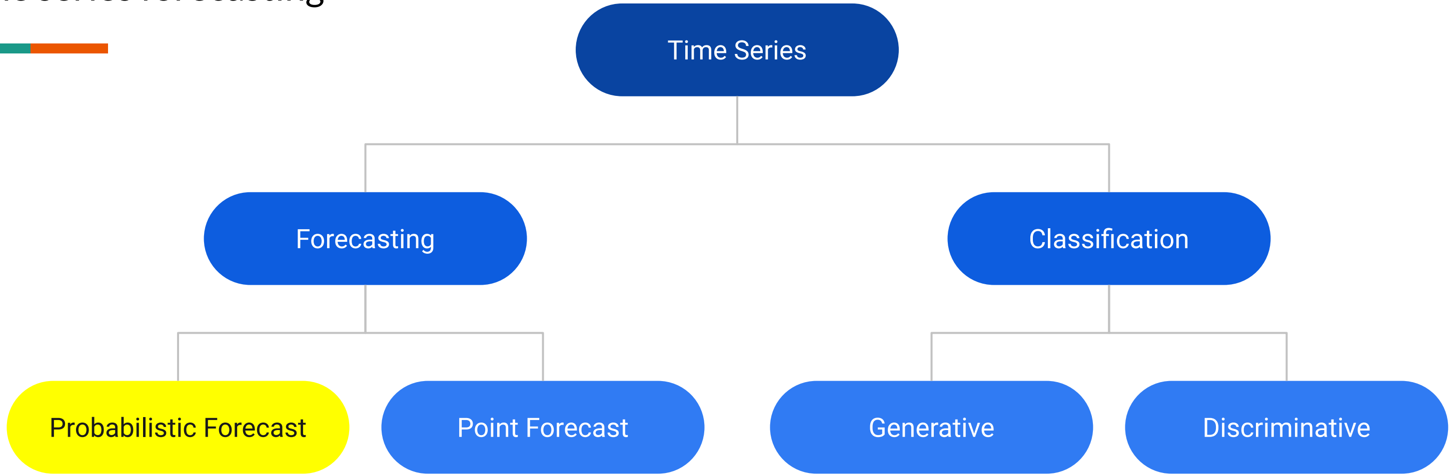
Probabilistic Multivariate Time-series Forecasting

Ali Izadi

Agenda

- Why probabilistic and multivariate?
- Familiar with the progress in the state of the art methods and their categories
- Future works.

Time series forecasting



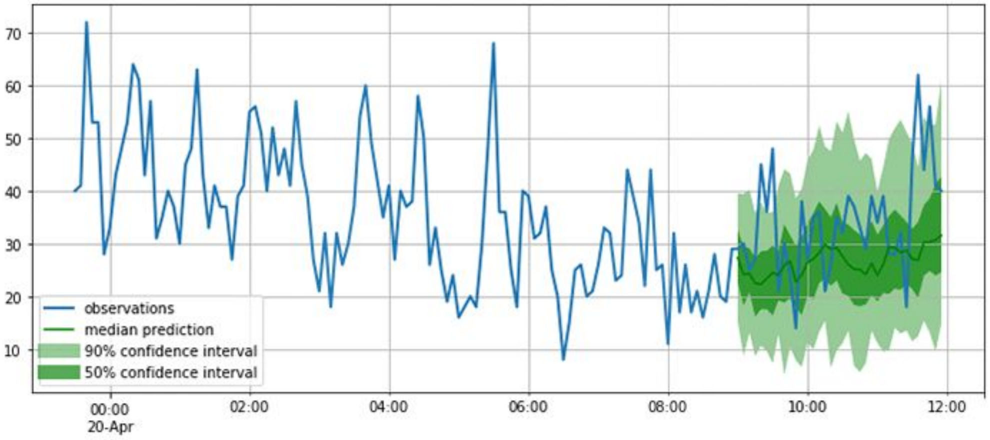
- Point Forecast: Seq2Seq - RNN - LSTM - Transformers

[1] Benidis, K., Rangapuram, S.S., Flunkert, V., Wang, B., Maddix, D., Turkmen, C., Gasthaus, J., Bohlke-Schneider, M., Salinas, D., Stella, L. and Callot, L., 2020. **Neural forecasting: Introduction and literature overview.** *arXiv preprint arXiv:2004.10240*.

Probabilistic forecasting

- Prediction **uncertainty** for assessing how much to trust the predictions.
- This problem is challenging, especially during **high variance** segments.
- **Extreme event** prediction depends on numerous **external factors**.

Forecast Type	Model
Point	$\hat{\mathbf{z}}_{i,T_i+1:T_i+\tau} = f(\mathbf{z}_{i,1:T_i}, \mathbf{x}_{i,1:T_i+1}; \Phi)$
Probabilistic	$p(\mathbf{z}_{i,T_i+1:T_i+\tau} \mid \mathbf{z}_{i,1:T_i}, \mathbf{x}_{i,1:T_i+\tau}; \Phi) = f(\mathbf{z}_{i,1:T_i}, \mathbf{x}_{i,1:T_i+1}; \Phi)$

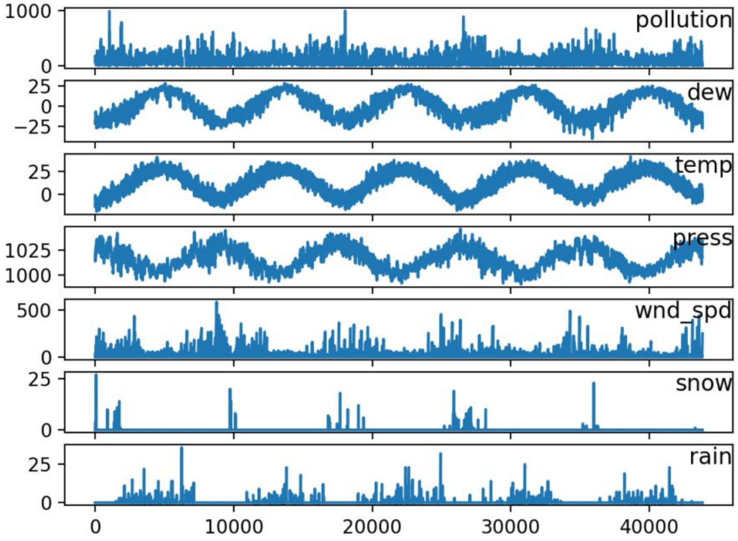


Multivariate forecasting

- Forecasting **thousands or millions** of related time series.

    - Energy consumption of individual households
    - The demand for all products that a large retailer offers
    - The load for servers in a data center

- The **computational** and **numerical difficulties** of estimating time-varying and **high-dimensional <u>dependencies</u>**



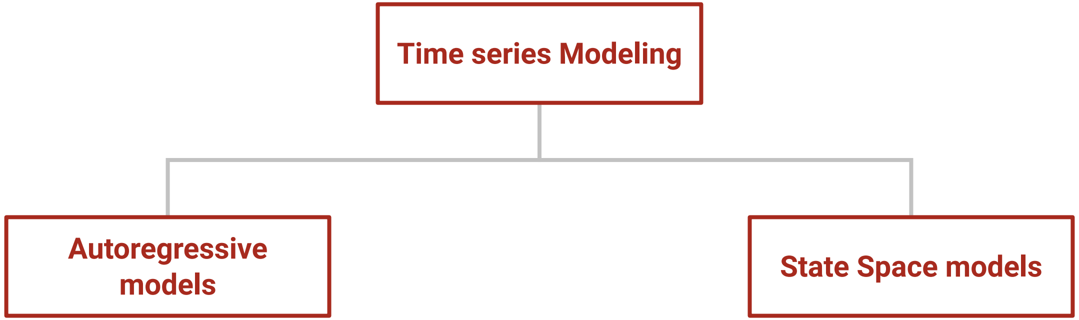
Definition

Let $y_t \in \mathbb{R}^N$ denote the value of a multivariate time series at time t, with $y_{t,i} \in \mathbb{R}$ the value of the corresponding i-th univariate time series. Further, let $x_{t,i} \in \mathbb{R}^k$ be time varying covariate vectors associated to each univariate time series at time t, and $x_t := [x_{t,1}, \dots, x_{t,N}] \in \mathbb{R}^{k \times N}$

Data-sets

- **Exchange rate**: daily exchange rate between 8 currencies
-
- **Solar**: hourly photovoltaic production of 137 stations in Alabama State
- **Electricity**: hourly time series of the electricity consumption of 370 customers
- **Traffic**: hourly occupancy rate, between 0 and 1, of 963 San Francisco car lanes
- **Taxi**: spatio-temporal traffic time series of New York taxi rides taken at 1214 locations every 30 minutes in the months of January 2015 (training set) and January 2016 (test set)
- **Wikipedia**: daily page views of 2000 Wikipedia pages

State space models vs **Autoregressive** models



State space models vs **Autoregressive** models

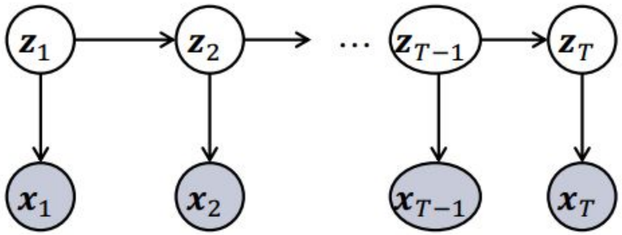
- Autoregressive:

$$Q_\Theta(\mathbf{z}_{i,t_0:T}|\mathbf{z}_{i,1:t_0-1}, \mathbf{x}_{i,1:T}) = \prod_{t=t_0}^{T} Q_\Theta(z_{i,t}|\mathbf{z}_{i,1:t-1}, \mathbf{x}_{i,1:T})$$

- State space

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

State space models



Linear Gaussian model

Gaussian State Space model:

$$z_t \sim \mathcal{N}(G_\alpha(z_{t-1}, \Delta_t), S_\beta(z_{t-1}, \Delta_t)) \quad \textit{(Transition)} \quad (1)$$
$$x_t \sim \Pi(F_\kappa(z_t)) \quad \textit{(Emission)} \quad (2)$$

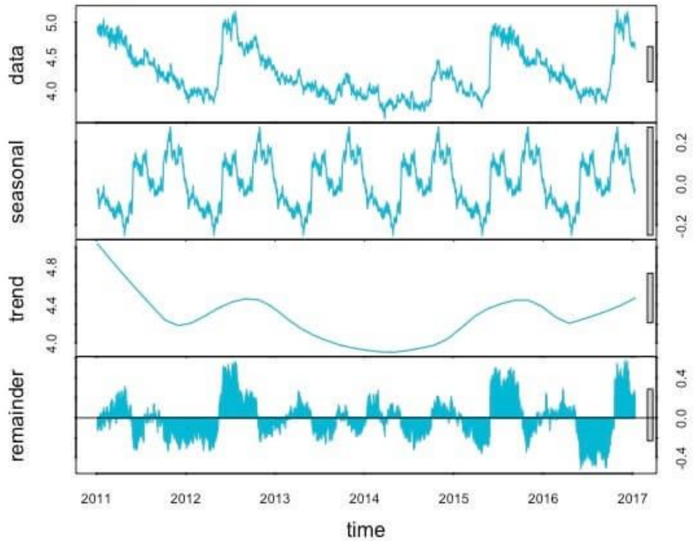
linear State Space model:

$$G_\alpha(z_{t-1}) = G_t z_{t-1}, S_\beta = \Sigma_t, F_\kappa = F_t z_t,$$

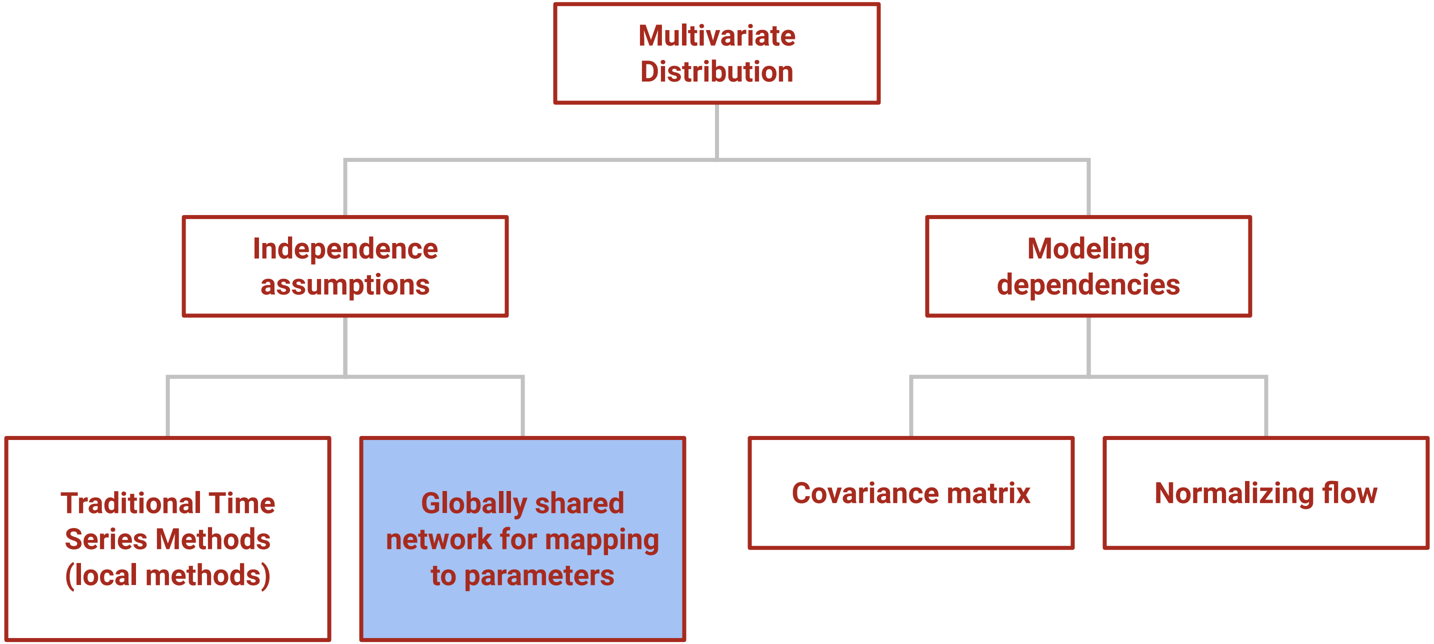
State space models vs **Autoregressive** models

- **Data efficiency**:
  forecasting time series with missing or noisy data irrespective of whether the data regime is sparse or dense

- **Structural assumptions**:
  Interpretability with composition of level-trend and seasonality model



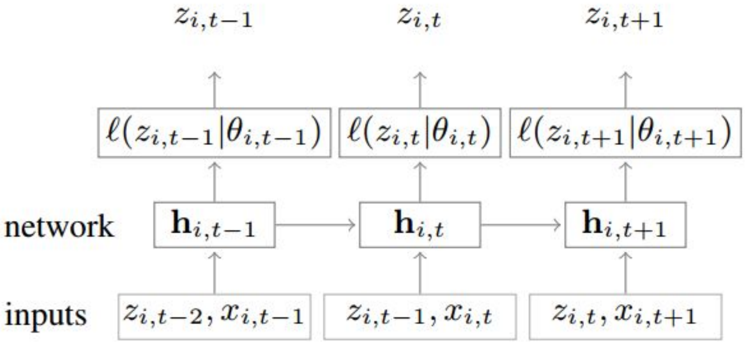
Multivariate Methods



DeepAr

$$Q_\Theta(\mathbf{z}_{i,t_0:T}|\mathbf{z}_{i,1:t_0-1},\mathbf{x}_{i,1:T}) = \prod_{t=t_0}^{T} Q_\Theta(z_{i,t}|\mathbf{z}_{i,1:t-1},\mathbf{x}_{i,1:T}) = \prod_{t=t_0}^{T} \ell(z_{i,t}|\theta(\mathbf{h}_{i,t},\Theta))$$

$$\mathbf{h}_{i,t} = h\left(\mathbf{h}_{i,t-1}, z_{i,t-1}, \mathbf{x}_{i,t}, \Theta\right)$$



$$\ell_{\mathrm{G}}(z|\mu,\sigma) = (2\pi\sigma^2)^{-\frac{1}{2}}\exp(-(z-\mu)^2/(2\sigma^2))$$

$$\mu(\mathbf{h}_{i,t}) = \mathbf{w}_\mu^T\mathbf{h}_{i,t} + b_\mu \quad \text{and} \quad \sigma(\mathbf{h}_{i,t}) = \log(1+\exp(\mathbf{w}_\sigma^T\mathbf{h}_{i,t} + b_\sigma))$$

[2] Salinas, D., Flunkert, V., Gasthaus, J. and Januschowski, T., 2020. **DeepAR: Probabilistic forecasting with autoregressive recurrent networks.** *International Journal of Forecasting*, *36*(3), pp.1181-1191.

Deep State Space

$$\boldsymbol{l}_t = \boldsymbol{F}_t \boldsymbol{l}_{t-1} + \boldsymbol{g}_t \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0,1).$$

$$z_t = y_t + \sigma_t \epsilon_t, \qquad y_t = \boldsymbol{a}_t^\top \boldsymbol{l}_{t-1} + b_t, \qquad \epsilon_t \sim \mathcal{N}(0,1),$$

$$\Theta_t = (\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \boldsymbol{F}_t, \boldsymbol{g}_t, \boldsymbol{a}_t, b_t, \sigma_t),$$

likelihood:

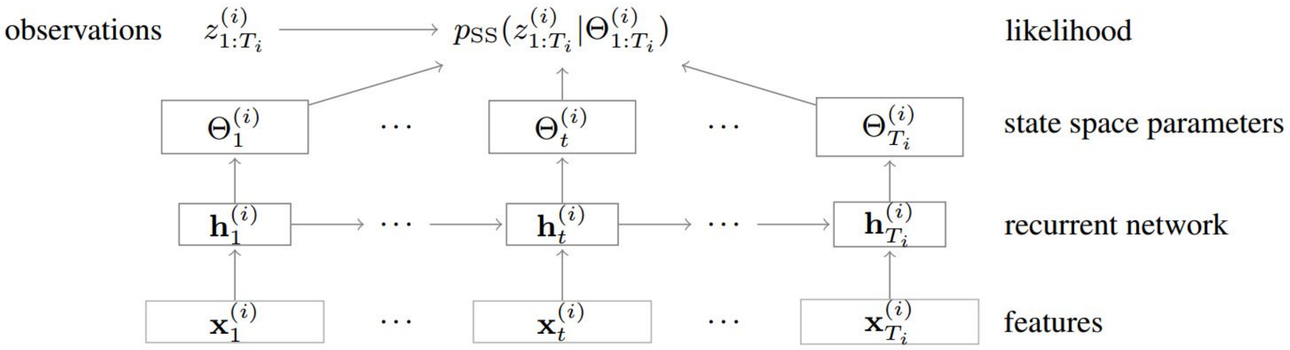
$$p_{SS}(z_{1:T}|\Theta_{1:T}) := p(z_1|\Theta_1) \prod_{t=2}^{T} p(z_t|z_{1:t-1}, \Theta_{1:t}) = \int p(\boldsymbol{l}_0) \left[\prod_{t=1}^{T} p(z_t|\boldsymbol{l}_t) p(\boldsymbol{l}_t|\boldsymbol{l}_{t-1}) \right] \mathrm{d}\boldsymbol{l}_{0:T}$$

[3] Rangapuram, S.S., Seeger, M.W., Gasthaus, J., Stella, L., Wang, Y. and Januschowski, T., 2018. **Deep state space models for time series forecasting**. *Advances in neural information processing systems*, *31*, pp.7785-7794.

Deep State Space

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

- State space parameters learned by recurrence network with **independent** assumptions between **dimensions**.



[3] Rangapuram, S.S., Seeger, M.W., Gasthaus, J., Stella, L., Wang, Y. and Januschowski, T., 2018. **Deep state space models for time series forecasting**. *Advances in neural information processing systems*, *31*, pp.7785-7794.

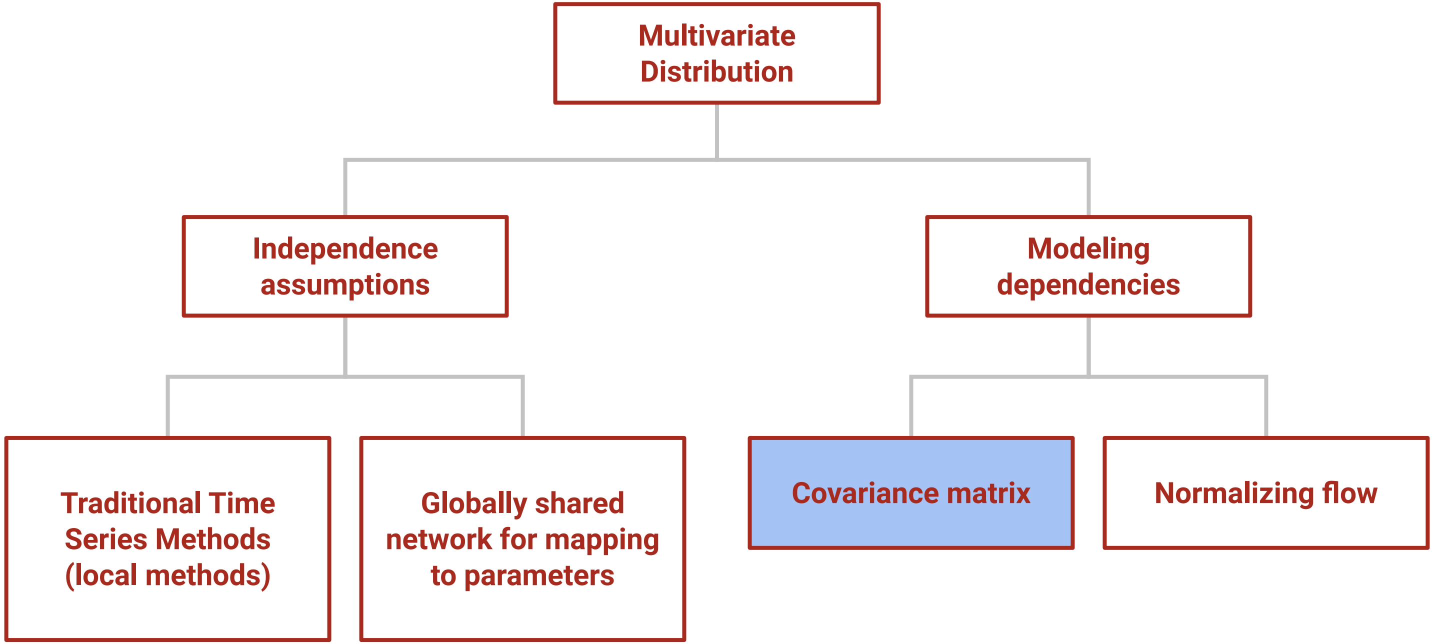
Deep State Space

Better result than **DeepAr** (autoregressive model)

Datasets	Methods	2-weeks p50Loss	2-weeks p90Loss	3-weeks p50Loss	3-weeks p90Loss	4-weeks p50Loss	4-weeks p90Loss
electricity	auto.arima	0.283	0.109	0.291	0.112	0.30	0.11
	ets	0.121	0.101	0.130	0.110	0.13	0.11
	DeepAR	0.153	0.147	0.147	0.132	0.125	0.080
	DeepState	**0.087**	**0.05**	**0.085**	**0.052**	**0.085**	**0.057**
traffic	auto.arima	0.492	0.280	0.492	0.289	0.501	0.298
	ets	0.621	0.650	0.509	0.529	0.532	0.60
	DeepAR	0.177	0.153	**0.126**	**0.096**	0.219	0.138
	DeepState	**0.168**	**0.117**	0.170	0.113	**0.168**	**0.114**

[3] Rangapuram, S.S., Seeger, M.W., Gasthaus, J., Stella, L., Wang, Y. and Januschowski, T., 2018. **Deep state space models for time series forecasting**. *Advances in neural information processing systems*, *31*, pp.7785-7794.

Multivariate Methods



Multivariate forecasting with low rank gaussian copula

a low-rank covariance structure to reduce computational complexity and handle non-Gaussian marginal distributions.

$$p(\mathbf{z}_1, \ldots \mathbf{z}_{T+\tau}) = \prod_{t=1}^{T+\tau} p(\mathbf{z}_t | \mathbf{z}_1, \ldots, \mathbf{z}_{t-1}) = \prod_{t=1}^{T+\tau} p(\mathbf{z}_t | \mathbf{h}_t).$$

$$\mathbf{h}_{i,t} = \varphi_{\theta_h}(\mathbf{h}_{i,t-1}, z_{i,t-1}) \qquad i = 1, \ldots, N,$$

$$p(\mathbf{z}_t | \mathbf{h}_t) = \mathcal{N}([f_1(z_{1,t}), f_2(z_{2,t}), \ldots, f_N(z_{N,t})]^T \mid \boldsymbol{\mu}(\mathbf{h}_t), \Sigma(\mathbf{h}_t)).$$

[4] Salinas, D., Bohlke-Schneider, M., Callot, L., Medico, R. and Gasthaus, J., 2019. **High-dimensional multivariate forecasting with low-rank gaussian copula processes.** *arXiv preprint arXiv:1910.03002*.

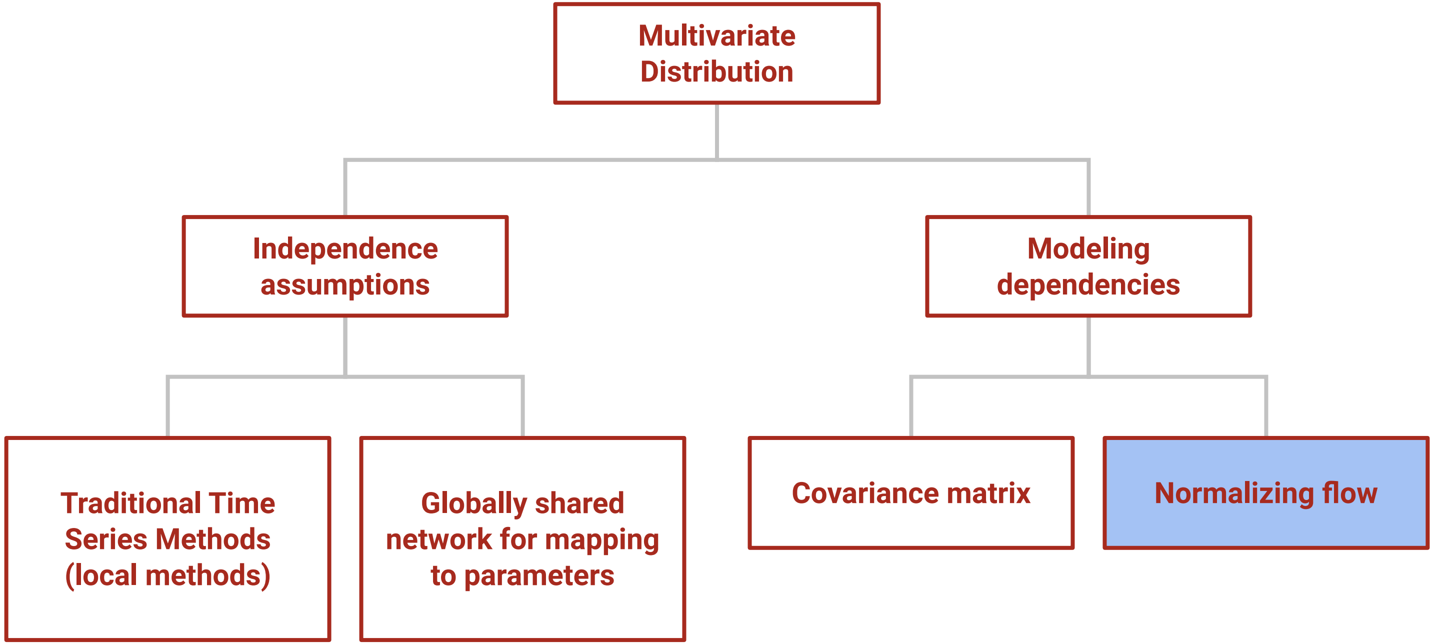
Multivariate forecasting with low rank gaussian copula

- Only methods that are able to produce **correlated samples** are considered in their comparisons.

	CRPS-Sum					
dataset estimator	exchange	solar	elec	traffic	taxi	wiki
VAR	0.010+/-0.000	0.524+/-0.001	0.031+/-0.000	0.144+/-0.000	0.292+/-0.000	3.400+/-0.003
GARCH	0.020+/-0.000	0.869+/-0.000	0.278+/-0.000	0.368+/-0.000	-	-
Vec-LSTM-ind	0.009+/-0.000	0.470+/-0.039	0.731+/-0.007	0.110+/-0.020	0.429+/-0.000	0.801+/-0.029
Vec-LSTM-ind-scaling	0.008+/-0.001	0.391+/-0.017	0.025+/-0.001	0.087+/-0.041	0.506+/-0.005	**0.133+/-0.002**
Vec-LSTM-fullrank	0.646+/-0.114	0.956+/-0.000	0.999+/-0.000	-	-	-
Vec-LSTM-fullrank-scaling	0.394+/-0.174	0.920+/-0.035	0.747+/-0.020	-	-	-
Vec-LSTM-lowrank-Copula	**0.007+/-0.000**	**0.319+/-0.011**	0.064+/-0.008	0.103+/-0.006	0.326+/-0.007	0.241+/-0.033
GP	0.011+/-0.001	0.828+/-0.010	0.947+/-0.016	2.198+/-0.774	0.425+/-0.199	0.933+/-0.003
GP-scaling	0.009+/-0.000	0.368+/-0.012	**0.022+/-0.000**	**0.079+/-0.000**	**0.183+/-0.395**	1.483+/-1.034
GP-Copula	**0.007+/-0.000**	**0.337+/-0.024**	**0.024+/-0.002**	**0.078+/-0.002**	**0.208+/-0.183**	**0.086+/-0.004**

[4] Salinas, D., Bohlke-Schneider, M., Callot, L., Medico, R. and Gasthaus, J., 2019. **High-dimensional multivariate forecasting with low-rank gaussian copula processes.** *arXiv preprint arXiv:1910.03002*.

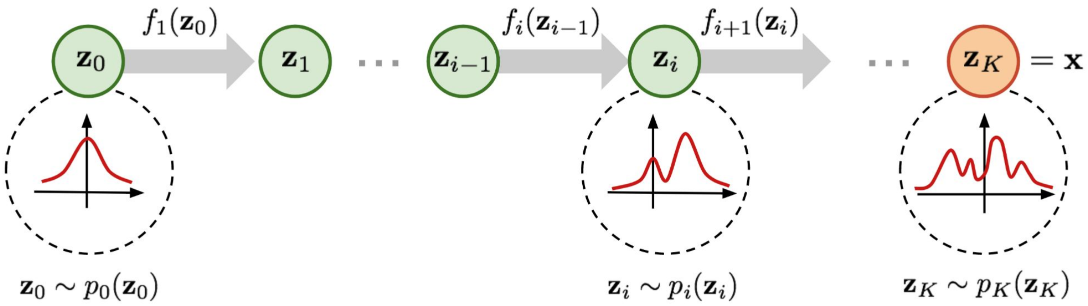
Multivariate Methods



Normalizing flow

We can transform a probability distribution using an invertible mapping (*i.e.* bijection). Let $\mathbf{z} \in \mathbb{R}^d$ be a random variable and $f : \mathbb{R}^d \mapsto \mathbb{R}^d$ an invertible smooth mapping. We can use f to transform $\mathbf{z} \sim q(\mathbf{z})$. The resulting random variable $\mathbf{y} = f(\mathbf{z})$ has the following probability distribution:

$$q_y(\mathbf{y}) = q(\mathbf{z}) \left| \det \frac{\partial f^{-1}}{\partial \mathbf{z}} \right| = q(\mathbf{z}) \left| \det \frac{\partial f}{\partial \mathbf{z}} \right|^{-1}. \quad (1)$$



[5] Rezende, D. and Mohamed, S., 2015, June. **Variational inference with normalizing flows.** In *International conference on machine learning* (pp. 1530-1538). PMLR.

Autoregressive model + Normalizing flow

- The model is autoregressive it can be written as a product of factors

$$p_{\mathcal{X}}(\mathbf{x}_{t_0:T}|\mathbf{x}_{1:t_0-1}, \mathbf{c}_{1:T}; \theta) = \Pi_{t=t_0}^{T} p_{\mathcal{X}}(\mathbf{x}_t|\mathbf{h}_t; \theta),$$

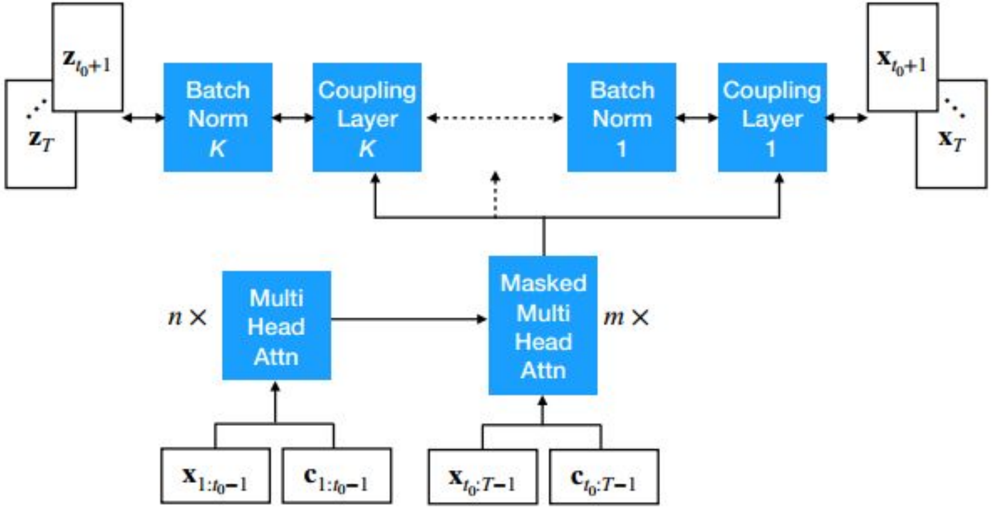
$$\mathbf{h}_t = \mathrm{RNN}(\mathtt{concat}(\mathbf{x}_{t-1}, \mathbf{c}_{t-1}), \mathbf{h}_{t-1}).$$

- To get a powerful and general **emission distribution** model, we stack K layers of a **conditional flow**

$$\log p_{\mathcal{X}}(\mathbf{x}) = \log p_{\mathcal{Z}}(\mathbf{z}) + \log|\det(\partial \mathbf{z}/\partial \mathbf{x})| = \log p_{\mathcal{Z}}(\mathbf{z}) + \sum_{i=1}^{K} \log|\det(\partial \mathbf{y}_i/\partial \mathbf{y}_{i-1})|.$$

[6] Rasul, K., Sheikh, A.S., Schuster, I., Bergmann, U. and Vollgraf, R., 2020. **Multivariate probabilistic time series forecasting via conditioned normalizing flows.** *arXiv preprint arXiv:2002.06103*.

Autoregressive model + Normalizing flow



[6] Rasul, K., Sheikh, A.S., Schuster, I., Bergmann, U. and Vollgraf, R., 2020. **Multivariate probabilistic time series forecasting via conditioned normalizing flows.** *arXiv preprint arXiv:2002.06103*.

Autoregressive model + Normalizing flow

- Better results than **GP copula (covariance)**

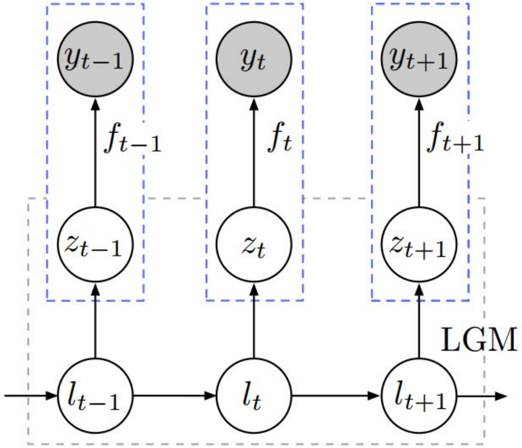
Data set	Vec-LSTM ind-scaling	Vec-LSTM lowrank-Copula	GP scaling	GP Copula	LSTM Real-NVP	LSTM MAF	Transformer MAF
Exchange	0.008±0.001	0.007±0.000	0.009±0.000	0.007±0.000	**0.0064**±0.003	**0.005**±0.003	**0.005**±0.003
Solar	0.391±0.017	0.319±0.011	0.368±0.012	0.337±0.024	0.331±0.02	**0.315**±0.023	**0.301**±0.014
Electricity	0.025±0.001	0.064±0.008	0.022±0.000	0.024±0.002	0.024±0.001	**0.0208**±0.000	**0.0207**±0.000
Traffic	0.087±0.041	0.103±0.006	0.079±0.000	0.078±0.002	0.078±0.001	**0.069**±0.002	**0.056**±0.001
Taxi	0.506±0.005	0.326±0.007	0.183±0.395	0.208±0.183	**0.175**±0.001	**0.161**±0.002	0.179±0.002
Wikipedia	0.133±0.002	0.241±0.033	1.483±1.034	0.086±0.004	0.078±0.001	**0.067**±0.001	**0.063**±0.003

[6] Rasul, K., Sheikh, A.S., Schuster, I., Bergmann, U. and Vollgraf, R., 2020. **Multivariate probabilistic time series forecasting via conditioned normalizing flows.** *arXiv preprint arXiv:2002.06103*.

State space model + Normalizing flow

$$
\begin{aligned}
&\mathbf{l}_1 \sim \mathcal{N}(\mu_1, \Sigma_1) && && \text{(initial state)}\\
&\mathbf{l}_t = F_t \mathbf{l}_{t-1} + \boldsymbol{\epsilon}_t, && \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \Sigma_t), && \text{(transition dynamics)}\\
&\mathbf{y}_t = f_t(A_t^T \mathbf{l}_t + \boldsymbol{\varepsilon}_t), && \boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, \Gamma_t). && \text{(observation model)}
\end{aligned}
$$

$$
p(y_t|l_t; \Theta, \Lambda) = p_z(f_t^{-1}(y_t)|l_t; \Theta) \left|\det\left[\mathrm{Jac}_{y_t}(f_t^{-1})\right]\right|,
$$



The resulting model still retaining many of the attractive **properties of state space models**, inference is tractable

[7] de Bézenac, E., Rangapuram, S.S., Benidis, K., Bohlke-Schneider, M., Kurle, R., Stella, L., Hasson, H., Gallinari, P. and Januschowski, T., 2020, January. **Normalizing Kalman Filters for Multivariate Time Series Analysis.** In *NeurIPS*.

State space model + Normalizing flow

- Better results than **GP copula and DeepAr**

	method	exchange	solar	elec	wiki	traffic
	VES	**0.005 ± 0.000**	0.9 ± 0.003	0.88 ± 0.0035	-	0.35 ± 0.0023
	VAR	**0.005 ± 0.000**	0.83 ± 0.006	0.039 ± 0.0005	-	0.29 ± 0.005
	VAR-Lasso	0.012 ± 0.0002	0.51 ± 0.006	0.025 ± 0.0002	3.1 ± 0.004	0.15 ± 0.002
	GARCH	0.023 ± 0.000	0.88 ± 0.002	0.19 ± 0.001	-	0.37 ± 0.0016
	DeepAR	0.006±0.001	0.336±0.014	0.023±0.001	0.127±0.042	0.055±0.003
	GP-Copula	0.007±0.000	0.363±0.002	0.024±0.000	0.092±0.012	**0.051±0.000**
	KVAE	0.014 ± 0.002	0.34 ± 0.025	0.051 ± 0.019	0.095 ± 0.012	0.1 ± 0.005
	NKF(Ours)	**0.005 ± 0.000**	**0.320±0.020**	**0.016±0.001**	**0.071±0.002**	0.10±0.002
ablation study	f_t = id	0.005±0.000	0.415±0.002	0.026±0.000	0.082±0.000	0.123±0.000
ablation study	f_t Local	0.005±0.000	0.405±0.005	0.018±0.001	0.068±0.004	0.102±0.013

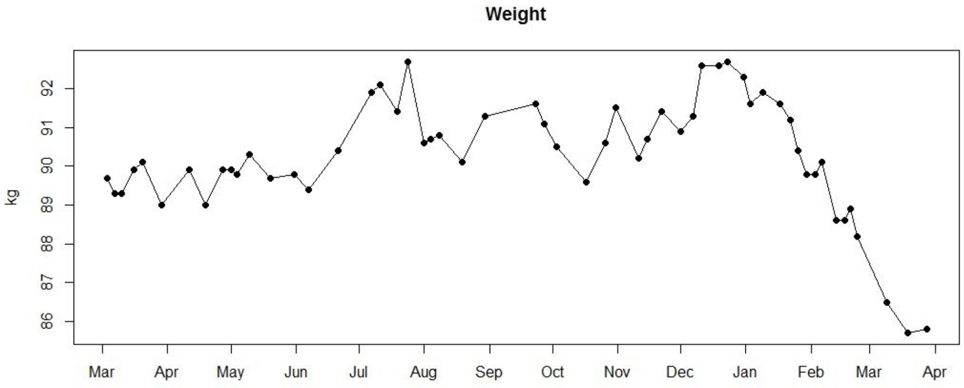
[7] de Bézenac, E., Rangapuram, S.S., Benidis, K., Bohlke-Schneider, M., Kurle, R., Stella, L., Hasson, H., Gallinari, P. and Januschowski, T., 2020, January. **Normalizing Kalman Filters for Multivariate Time Series Analysis.** In *NeurIPS*.

Continuous time series models

- necessary for **irregular** time series:
  Time between observations isn't constant.

- Tasks:

  - Interpolation(missing values)
  - Exterapolation



Continuous state space model + normalizing flow

- inherits many of the appealing **properties of its base process** such as efficient **computation of likelihoods**.

- Wiener process (continuous)

$$p_{\mathbf{W}_t|\mathbf{W}_s}(\boldsymbol{w}_t|\boldsymbol{w}_s) = \mathcal{N}(\boldsymbol{w}_t; \boldsymbol{w}_s, (t-s)\boldsymbol{I}_d),$$

- Continuous normalizing flow

$$\log p_{\mathbf{X}}(\boldsymbol{h}(t_1)) = \log p_{\mathbf{Z}}(\boldsymbol{h}(t_0)) - \int_{t_0}^{t_1} \operatorname{tr}\left(\frac{\partial f}{\partial \boldsymbol{h}(t)}\right) dt.$$

[8] Deng, R., Chang, B., Brubaker, M.A., Mori, G. and Lehrmann, A., 2020. **Modeling continuous stochastic processes with dynamic normalizing flows**. *arXiv preprint arXiv:2002.10516*.
[9] Chen, R.T., Rubanova, Y., Bettencourt, J. and Duvenaud, D., 2018. **Neural ordinary differential equations.** *arXiv preprint arXiv:1806.07366*.

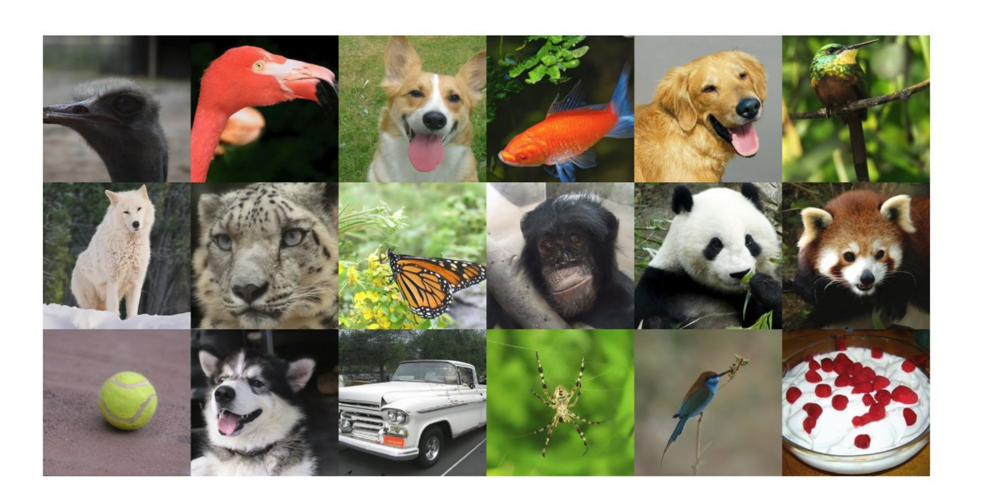
Future Works

- Enrich multivariate Distribution with probabilistic graphical models (sparser representation) or finding casual dependencies.

[10] Wehenkel, A. and Louppe, G., 2021, March. **Graphical normalizing flows**. In *International Conference on Artificial Intelligence and Statistics* (pp. 37-45). PMLR.

Future Works

- Diffusion models (state of the art generative models)
    - Diffusion Models Beat GANs on Image Synthesis
    - Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting



Thank you.

aliizadi2030@gmail.com