RNN with Particle Flow for Probabilistic Spatio-temporal Forecasting

Soumyasundar Pal[1], Liheng Ma[1], Yingxue Zhang[2], Mark Coates[1]

1. Dept. of Electrical and Computer Engineering, McGill University
2. Huawei Noah's Ark Lab, Montreal Research Center

July 17, 2021

McGill HUAWEI Computer Networks Research Laboratory

Introduction

- Exploit underlying graph structure for time series forecasting

Introduction

- Exploit underlying graph structure for time series forecasting
- Applications: road traffic, wireless networks

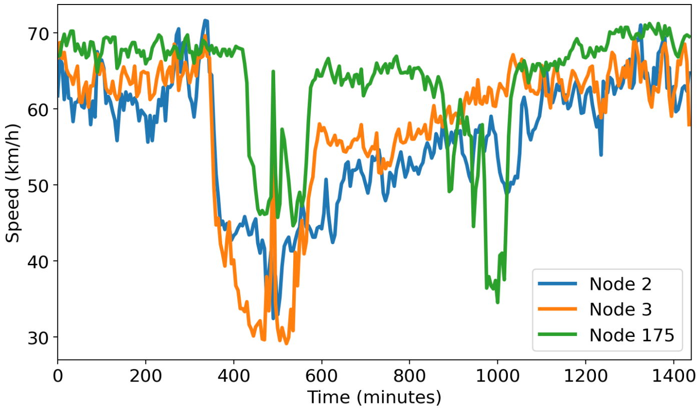
Introduction

- Exploit underlying graph structure for time series forecasting
- Applications: road traffic, wireless networks

reproduced from
https://www.tomtom.com/blog/traffic-and-travel-information/road-traffic-prediction/

Introduction

- Exploit underlying graph structure for time series forecasting
- Applications: road traffic, wireless networks



Introduction

- *State-of-the-art*
  - Graph convolution + recurrent networks[1]
  - Temporal convolution[2]
  - Attention mechanism[3]

[1] Li et al. 2018, Bai et al. 2020
[2] Yu et al. 2018, Huang et al. 2020
[3] Guo et al. 2019, Zheng et al. 2020

Introduction

- *State-of-the-art*
  - Graph convolution + recurrent networks[1]
  - Temporal convolution[2]
  - Attention mechanism[3]
- Provide point forecast, no measure of uncertainty

[1] Li et al. 2018, Bai et al. 2020
[2] Yu et al. 2018, Huang et al. 2020
[3] Guo et al. 2019, Zheng et al. 2020

Introduction

- *State-of-the-art*
  - Graph convolution + recurrent networks[1]
  - Temporal convolution[2]
  - Attention mechanism[3]
- Provide point forecast, no measure of uncertainty
- Existing probabilistic models[4] cannot process a graph.

[1] Li et al. 2018, Bai et al. 2020
[2] Yu et al. 2018, Huang et al. 2020
[3] Guo et al. 2019, Zheng et al. 2020
[4] Salinas et al. 2020, Wang et al. 2019, Rasul et al. 2021

Introduction

- *State-of-the-art*
  - Graph convolution + recurrent networks[1]
  - Temporal convolution[2]
  - Attention mechanism[3]
- Provide point forecast, no measure of uncertainty
- Existing probabilistic models[4] cannot process a graph.
- This work: Bayesian framework to assess forecast uncertainty

[1] Li et al. 2018, Bai et al. 2020
[2] Yu et al. 2018, Huang et al. 2020
[3] Guo et al. 2019, Zheng et al. 2020
[4] Salinas et al. 2020, Wang et al. 2019, Rasul et al. 2021

Problem formulation

State-space model

Initial state distribution: $\mathrm{x}_1 \sim p_1(\cdot, \mathrm{z}_1, \rho)$,

State transition model: $\mathrm{x}_t = g_{\mathcal{G},\psi}(\mathrm{x}_{t-1}, \mathrm{y}_{t-1}, \mathrm{z}_t, \mathrm{v}_t)$, for $t > 1$,

Emission model: $\mathrm{y}_t = h_{\mathcal{G},\phi}(\mathrm{x}_t, \mathrm{z}_t, \mathrm{w}_t)$, for $t \geqslant 1$.

Problem formulation

State-space model

Initial state distribution: $\mathrm{x}_1 \sim p_1(\cdot, \mathrm{z}_1, \rho)$,

State transition model: $\mathrm{x}_t = g_{\mathcal{G},\psi}(\mathrm{x}_{t-1}, \mathrm{y}_{t-1}, \mathrm{z}_t, \mathrm{v}_t)$, for $t > 1$,

Emission model: $\mathrm{y}_t = h_{\mathcal{G},\phi}(\mathrm{x}_t, \mathrm{z}_t, \mathrm{w}_t)$, for $t \geqslant 1$.

– y_t: time series, x_t: hidden state, z_t: known covariate(s)

Problem formulation

State-space model

Initial state distribution: $\mathrm{x}_1 \sim p_1(\cdot, \mathrm{z}_1, \rho)$,

State transition model: $\mathrm{x}_t = g_{\mathcal{G},\psi}(\mathrm{x}_{t-1}, \mathrm{y}_{t-1}, \mathrm{z}_t, \mathrm{v}_t)$, for $t > 1$,

Emission model: $\mathrm{y}_t = h_{\mathcal{G},\phi}(\mathrm{x}_t, \mathrm{z}_t, \mathrm{w}_t)$, for $t \geqslant 1$.

- y_t: time series, x_t: hidden state, z_t: known covariate(s)
- $\mathrm{v}_t \sim p_v(\cdot | \mathrm{x}_{t-1}, \sigma)$: dynamic noise

Problem formulation

State-space model

Initial state distribution: $x_1 \sim p_1(\cdot, z_1, \rho)$,

State transition model: $x_t = g_{\mathcal{G},\psi}(x_{t-1}, y_{t-1}, z_t, v_t)$, for $t > 1$,

Emission model: $y_t = h_{\mathcal{G},\phi}(x_t, z_t, w_t)$, for $t \geqslant 1$.

- y_t: time series, x_t: hidden state, z_t: known covariate(s)
- $v_t \sim p_v(\cdot | x_{t-1}, \sigma)$: dynamic noise
- $w_t \sim p_w(\cdot | x_t, \gamma)$: measurement noise

Problem formulation

State-space model

Initial state distribution: $\mathrm{x}_1 \sim p_1(\cdot, \mathrm{z}_1, \rho)$,

State transition model: $\mathrm{x}_t = g_{\mathcal{G},\psi}(\mathrm{x}_{t-1}, \mathrm{y}_{t-1}, \mathrm{z}_t, \mathrm{v}_t)$, for $t > 1$,

Emission model: $\mathrm{y}_t = h_{\mathcal{G},\phi}(\mathrm{x}_t, \mathrm{z}_t, \mathrm{w}_t)$, for $t \geqslant 1$.

- y_t: time series, x_t: hidden state, z_t: known covariate(s)
- $\mathrm{v}_t \sim p_v(\cdot|\mathrm{x}_{t-1}, \sigma)$: dynamic noise
- $\mathrm{w}_t \sim p_w(\cdot|\mathrm{x}_t, \gamma)$: measurement noise
- $g_{\mathcal{G},\psi}$: GNN+RNN (e.g. AGCGRU[5], DCGRU[6])

[5] Bai et al. 2020, [6] Li et al. 2018

Problem formulation

State-space model

Initial state distribution: $\mathrm{x}_1 \sim p_1(\cdot, \mathrm{z}_1, \rho)\,,$

State transition model: $\mathrm{x}_t = g_{\mathcal{G},\psi}(\mathrm{x}_{t-1}, \mathrm{y}_{t-1}, \mathrm{z}_t, \mathrm{v}_t),\;$ for $t > 1\,,$

Emission model: $\mathrm{y}_t = h_{\mathcal{G},\phi}(\mathrm{x}_t, \mathrm{z}_t, \mathrm{w}_t),\;$ for $t \geqslant 1\,.$

- y_t: time series, x_t: hidden state, z_t: known covariate(s)
- $\mathrm{v}_t \sim p_v(\cdot|\mathrm{x}_{t-1}, \sigma)$: dynamic noise
- $\mathrm{w}_t \sim p_w(\cdot|\mathrm{x}_t, \gamma)$: measurement noise
- $g_{\mathcal{G},\psi}$: GNN+RNN (e.g. AGCGRU[5], DCGRU[6])
- $h_{\mathcal{G},\phi}$: NN (e.g. linear layer)

[5] Bai et al. 2020, [6] Li et al. 2018

Problem formulation

State-space model

Initial state distribution: $\mathrm{x}_1 \sim p_1(\cdot, \mathrm{z}_1, \rho)$,

State transition model: $\mathrm{x}_t = g_{\mathcal{G},\psi}(\mathrm{x}_{t-1}, \mathrm{y}_{t-1}, \mathrm{z}_t, \mathrm{v}_t)$, for $t > 1$,

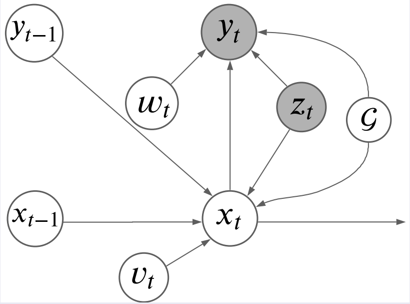
Emission model: $\mathrm{y}_t = h_{\mathcal{G},\phi}(\mathrm{x}_t, \mathrm{z}_t, \mathrm{w}_t)$, for $t \geqslant 1$.

- y_t: time series, x_t: hidden state, z_t: known covariate(s)
- $\mathrm{v}_t \sim p_v(\cdot|\mathrm{x}_{t-1}, \sigma)$: dynamic noise
- $\mathrm{w}_t \sim p_w(\cdot|\mathrm{x}_t, \gamma)$: measurement noise
- $g_{\mathcal{G},\psi}$: GNN+RNN (e.g. AGCGRU[5], DCGRU[6])
- $h_{\mathcal{G},\phi}$: NN (e.g. linear layer)
- Unknown model parameters: $\Theta = \{\rho, \psi, \sigma, \phi, \gamma\}$

[5] Bai et al. 2020, [6] Li et al. 2018

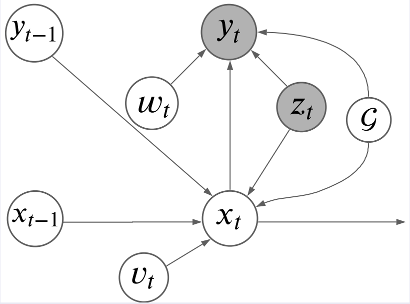
Problem formulation

Graphical model representation



Problem formulation

Graphical model representation

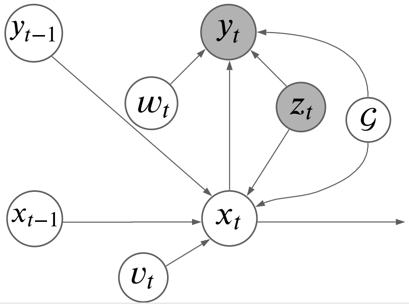


Task

Predict $\mathrm{y}_{t_0+P+1:t_0+P+Q}$ based on $\mathrm{y}_{t_0+1:t_0+P}$, $\mathrm{z}_{t_0+1:t_0+P+Q}$, and (possibly) $\mathcal{G}$

Problem formulation

Graphical model representation



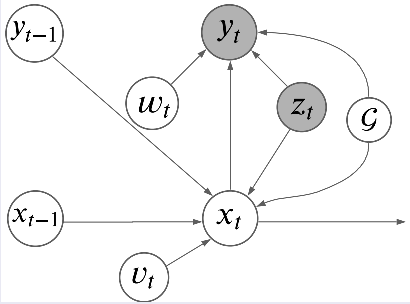
Task

Predict $y_{t_0+P+1:t_0+P+Q}$ based on $y_{t_0+1:t_0+P}$, $z_{t_0+1:t_0+P+Q}$, and (possibly) $\mathcal{G}$

- Train the model to learn Θ

Problem formulation

Graphical model representation



Task

Predict $\mathrm{y}_{t_0+P+1:t_0+P+Q}$ based on $\mathrm{y}_{t_0+1:t_0+P}$, $\mathrm{z}_{t_0+1:t_0+P+Q}$, and (possibly) $\mathcal{G}$

- Train the model to learn Θ
- Approximate $p_\Theta(\mathrm{y}_{P+1:P+Q}|\mathrm{y}_{1:P}, \mathrm{z}_{1:P+Q})$ for test data

Computing forecast distribution

$$p_\Theta(y_{P+1:P+Q} | y_{1:P}, z_{1:P+Q}) = \int \prod_{t=P+1}^{P+Q} \Big(p_{\phi,\gamma}(y_t | x_t, z_t) \, p_{\psi,\sigma}(x_t | x_{t-1}, y_{t-1}, z_t) \Big) \, p_\Theta(x_P | y_{1:P}, z_{1:P}) dx_{P:P+Q} \, .$$

Computing forecast distribution

$$p_\Theta(y_{P+1:P+Q}|y_{1:P}, z_{1:P+Q}) = \int \prod_{t=P+1}^{P+Q} \Big(p_{\phi,\gamma}(y_t|x_t, z_t) \, p_{\psi,\sigma}(x_t|x_{t-1}, y_{t-1}, z_t) \Big) \, p_\Theta(x_P|y_{1:P}, z_{1:P}) dx_{P:P+Q} \, .$$

– Intractable, Monte Carlo approximation

Computing forecast distribution

$$p_\Theta(y_{P+1:P+Q} | y_{1:P}, z_{1:P+Q}) = \int \prod_{t=P+1}^{P+Q} \Big(p_{\phi,\gamma}(y_t | x_t, z_t) \\ p_{\psi,\sigma}(x_t | x_{t-1}, y_{t-1}, z_t) \Big) \\ p_\Theta(x_P | y_{1:P}, z_{1:P}) dx_{P:P+Q} \,.$$

- Intractable, Monte Carlo approximation
- $p_\Theta(x_P | y_{1:P}, z_{1:P})$: posterior distribution of the state

Computing forecast distribution

$$p_\Theta(y_{P+1:P+Q}|y_{1:P}, z_{1:P+Q}) = \int \prod_{t=P+1}^{P+Q} \Big(p_{\phi,\gamma}(y_t|x_t, z_t) \, p_{\psi,\sigma}(x_t|x_{t-1}, y_{t-1}, z_t) \Big) \, p_\Theta(x_P|y_{1:P}, z_{1:P}) dx_{P:P+Q} \,.$$

- Intractable, Monte Carlo approximation
- $p_\Theta(x_P|y_{1:P}, z_{1:P})$: posterior distribution of the state
- Need particle filter/particle flow for approximation

Computing forecast distribution

$$p_\Theta(y_{P+1:P+Q}|y_{1:P}, z_{1:P+Q}) = \int \prod_{t=P+1}^{P+Q} \Big(p_{\phi,\gamma}(y_t|x_t, z_t)$$
$$p_{\psi,\sigma}(x_t|x_{t-1}, y_{t-1}, z_t)\Big)$$
$$p_\Theta(x_P|y_{1:P}, z_{1:P}) dx_{P:P+Q} \,.$$

- Intractable, Monte Carlo approximation
- $p_\Theta(x_P|y_{1:P}, z_{1:P})$: posterior distribution of the state
- Need particle filter/particle flow for approximation
- $p_{\psi,\sigma}(x_t|x_{t-1}, y_{t-1}, z_t)$: state transition using $g_{\mathcal{G},\psi}$

Computing forecast distribution

$$p_\Theta(y_{P+1:P+Q}|y_{1:P}, z_{1:P+Q}) = \int \prod_{t=P+1}^{P+Q} \Big(p_{\phi,\gamma}(y_t|x_t, z_t) \\ p_{\psi,\sigma}(x_t|x_{t-1}, y_{t-1}, z_t) \Big) \\ p_\Theta(x_P|y_{1:P}, z_{1:P}) dx_{P:P+Q} \, .$$

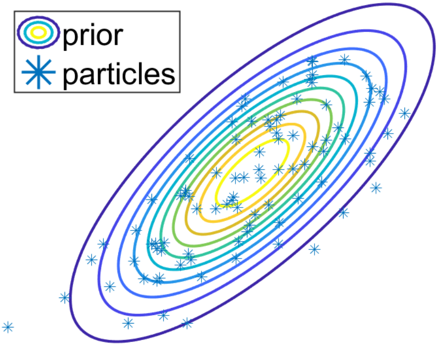
- Intractable, Monte Carlo approximation
- $p_\Theta(x_P|y_{1:P}, z_{1:P})$: posterior distribution of the state
- Need particle filter/particle flow for approximation
- $p_{\psi,\sigma}(x_t|x_{t-1}, y_{t-1}, z_t)$: state transition using $g_{\mathcal{G},\psi}$
- $p_{\phi,\gamma}(y_t|x_t, z_t)$: sampling forecast using $h_{\mathcal{G},\phi}$

Particle filter: weight degeneracy

Particle filter suffers from weight degeneracy for high dimensional state/ informative observations.

Particle filter: weight degeneracy

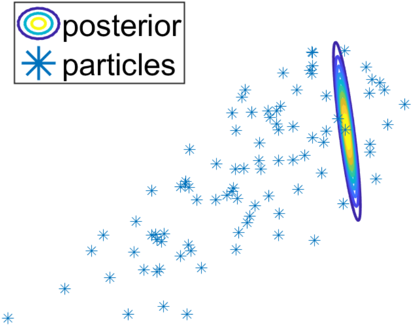
Particle filter suffers from weight degeneracy for high dimensional state/ informative observations.



Contours of the prior distribution

Particle filter: weight degeneracy

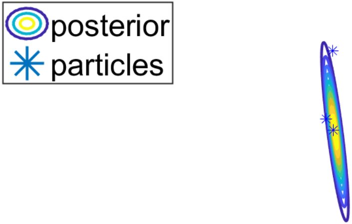
Particle filter suffers from weight degeneracy for high dimensional state/ informative observations.



Contours of the posterior distribution

Particle filter: weight degeneracy

Particle filter suffers from weight degeneracy for high dimensional state/ informative observations.



Resampling of the particles

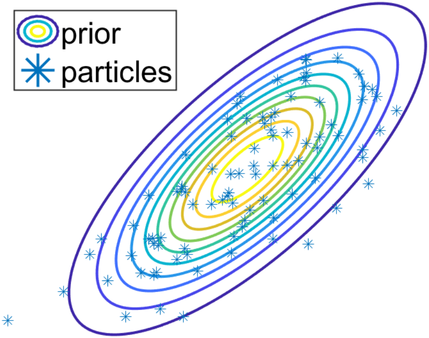
Particle flow

Particles flow[7] migrates particles from the prior to the posterior distribution.

[7] F. Daum and J. Huang, "Nonlinear filters with log-homotopy," in *Proc. SPIE Signal and Data Proc. Small Targets*, Sep. 2007.

Particle flow

Particles flow[7] migrates particles from the prior to the posterior distribution.



[7] F. Daum and J. Huang, "Nonlinear filters with log-homotopy," in *Proc. SPIE Signal and Data Proc. Small Targets*, Sep. 2007.

Computing forecast distribution

$$p_\Theta(y_{P+1:P+Q}|y_{1:P}, z_{1:P+Q}) = \int \prod_{t=P+1}^{P+Q} \Big(p_{\phi,\gamma}(y_t|x_t, z_t) \\ p_{\psi,\sigma}(x_t|x_{t-1}, y_{t-1}, z_t) \Big) \\ p_\Theta(x_P|y_{1:P}, z_{1:P}) dx_{P:P+Q} \, .$$

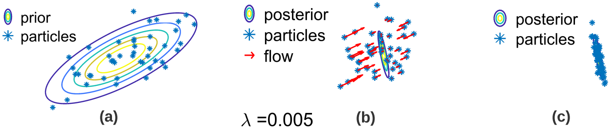
Computing forecast distribution

$$p_\Theta(y_{P+1:P+Q}|y_{1:P}, z_{1:P+Q}) = \int \prod_{t=P+1}^{P+Q} \Big(p_{\phi,\gamma}(y_t|x_t, z_t) \; p_{\psi,\sigma}(x_t|x_{t-1}, y_{t-1}, z_t) \Big) \; p_\Theta(x_P|y_{1:P}, z_{1:P}) dx_{P:P+Q} \, .$$

State transition model $g_{\mathcal{G},\psi}(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \mathbf{z}_t, \mathbf{v}_t)$

Emission model $h_{\mathcal{G},\phi}(\mathbf{x}_t, \mathbf{z}_t, \mathbf{w}_t)$

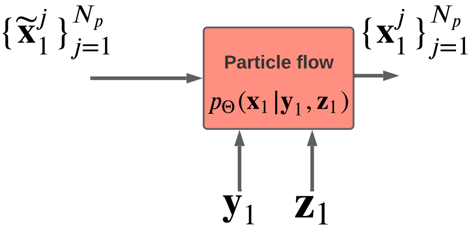
Particle flow



(a) Samples (asterisk) from the prior distribution
(b) Contours of the posterior distribution and the direction of flow for the particles at an intermediate step
(c) Particles after the flow, approximately distributed according to the posterior distribution

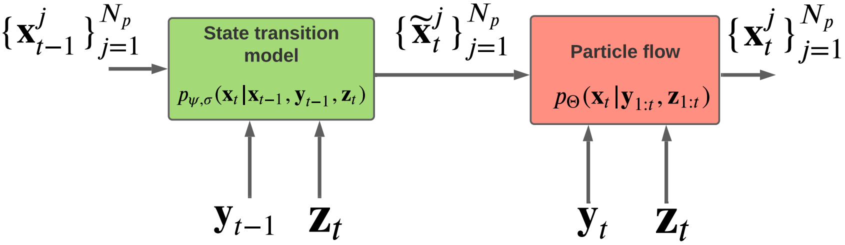
$$t = 1$$

$$\{\widetilde{\mathbf{x}}_1^j\}_{j=1}^{N_p} \sim p_1(\cdot, \mathbf{z}_1, \rho)$$

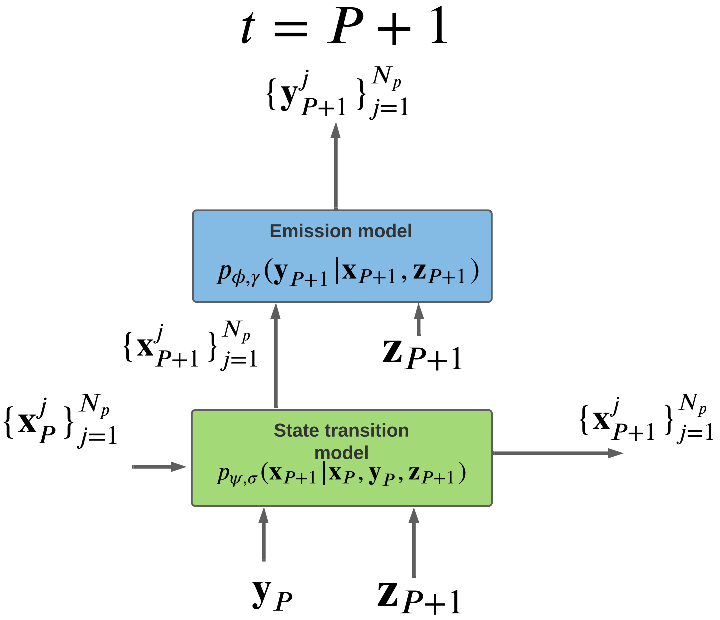


Computing forecast distribution

$$2 \leqslant t \leqslant P$$

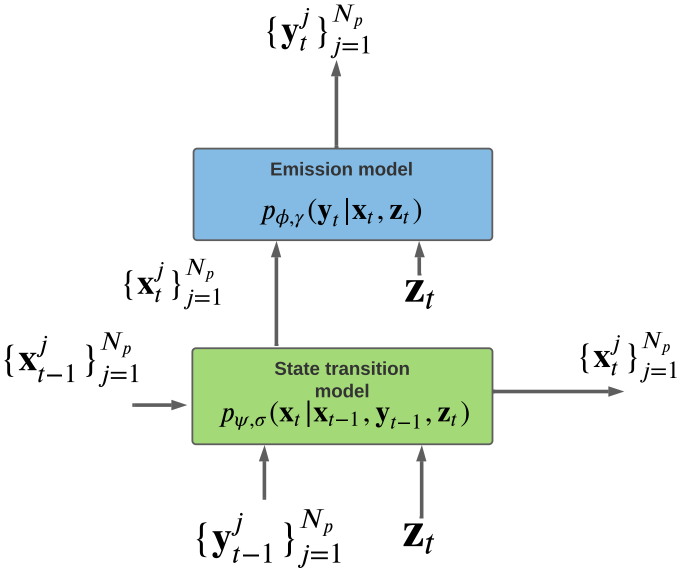


Computing forecast distribution

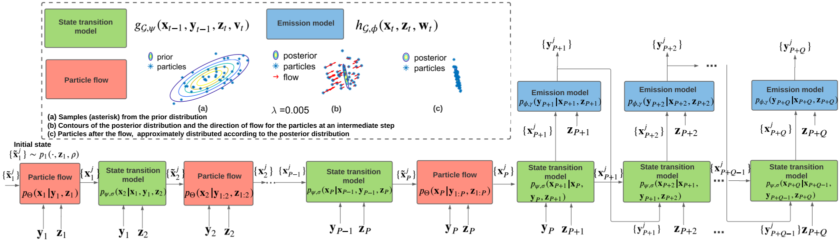


Computing forecast distribution

$$P + 2 \leqslant t \leqslant P + Q$$



Computing forecast distribution



Approximation of the joint posterior distribution of the forecasts

Loss function

- For point forecasting: MAE, MSE
- For probabilistic forecasting: negative log posterior probability

Loss function

- For point forecasting: MAE, MSE
- For probabilistic forecasting: negative log posterior probability

$$\mathcal{L}_{\text{prob}}(\Theta, \mathcal{D}) = -\frac{1}{|\mathcal{D}|} \sum_{n \in \mathcal{D}} \log p_{\Theta}(\mathrm{y}_{P+1:P+Q}^{(n)} | \mathrm{y}_{1:P}^{(n)}, \mathrm{z}_{1:P+Q}^{(n)}),$$

$$\widehat{p}_{\Theta}(\mathrm{y}_{P+1:P+Q} | \mathrm{y}_{1:P}, \mathrm{z}_{P+1:P+Q}) = \prod_{t=P+1}^{P+Q} \left[\frac{1}{N_p} \sum_{j=1}^{N_p} p_{\phi,\gamma}(\mathrm{y}_t | \mathrm{x}_t^j, \mathrm{z}_t) \right].$$

Experiments

- Road traffic datasets: PeMSD3/4/7/8[8]

[8] Chen et al. 2000

Experiments

- Road traffic datasets: PeMSD3/4/7/8[8]
- Node: loop detector, time series: speed, interval: 5 minutes

[8] Chen et al. 2000

Experiments

- Road traffic datasets: PeMSD3/4/7/8[8]
- Node: loop detector, time series: speed, interval: 5 minutes
- predicting one hour from an hour of historical data ($P = Q = 12$)

[8] Chen et al. 2000

Experiments

- Road traffic datasets: PeMSD3/4/7/8[8]
- Node: loop detector, time series: speed, interval: 5 minutes
- predicting one hour from an hour of historical data ($P = Q = 12$)
- 70/10/20% data for training/validation/testing

[8] Chen et al. 2000

Experiments

- Road traffic datasets: PeMSD3/4/7/8[8]
- Node: loop detector, time series: speed, interval: 5 minutes
- predicting one hour from an hour of historical data ($P = Q = 12$)
- 70/10/20% data for training/validation/testing
- Performance metrics for point forecasting:
  - MAE, RMSE, and MAPE

[8] Chen et al. 2000

Experiments

- Road traffic datasets: PeMSD3/4/7/8[8]
- Node: loop detector, time series: speed, interval: 5 minutes
- predicting one hour from an hour of historical data ($P = Q = 12$)
- 70/10/20% data for training/validation/testing
- Performance metrics for point forecasting:
  - MAE, RMSE, and MAPE
- Performance metrics for probabilistic forecasting:
  - Continuous Ranked Probability Score (CRPS)[9]
  - P10, P50, and P90 Quantile Losses[10]

[8] Chen et al. 2000

[9] Gneiting & Raftery 2007

[10] Wang et al. 2019

Baselines

- Statistical and ML point forecast models:
  - HA, ARIMA[11], VAR[12], SVR[13], FNN, FC-LSTM[14]

[11] Makridakis & Hibon 1997, [12] Hamilton 1994, [13] Chun-Hsin et al. 2004, [14] Sutskever et al. 2014

Baselines

- Statistical and ML point forecast models:
  - HA, ARIMA[11], VAR[12], SVR[13], FNN, FC-LSTM[14]
- Spatio-temporal point forecast models:
  - DCRNN[15], STGCN[16], ASTGCN[17], GWN[18], GMAN[19], AGCRN[20], LSGCN[21]

[11] Makridakis & Hibon 1997, [12] Hamilton 1994, [13] Chun-Hsin et al. 2004, [14] Sutskever et al. 2014
[15] Li et al. 2018, [16] Yu et al. 2018, [17] Guo et al. 2019, [18] Wu et al. 2019, [19] Zheng et al. 2020,
[20] Bai et al. 2020, [21] Huang et al. 2021

Baselines

- Statistical and ML point forecast models:
  - HA, ARIMA[11], VAR[12], SVR[13], FNN, FC-LSTM[14]
- Spatio-temporal point forecast models:
  - DCRNN[15], STGCN[16], ASTGCN[17], GWN[18], GMAN[19], AGCRN[20], LSGCN[21]
- Graph agnostic point forecast models:
  - DeepGLO[22], N-BEATS[23], FC-GAGA[24]

[11] Makridakis & Hibon 1997, [12] Hamilton 1994, [13] Chun-Hsin et al. 2004, [14] Sutskever et al. 2014

[15] Li et al. 2018, [16] Yu et al. 2018, [17] Guo et al. 2019, [18] Wu et al. 2019, [19] Zheng et al. 2020, [20] Bai et al. 2020, [21] Huang et al. 2021

[22] Sen et al. 2019, [23] Oreshkin et al. 2020, [24] Oreshkin et al. 2021

Baselines

- Statistical and ML point forecast models:
  - HA, ARIMA[11], VAR[12], SVR[13], FNN, FC-LSTM[14]
- Spatio-temporal point forecast models:
  - DCRNN[15], STGCN[16], ASTGCN[17], GWN[18], GMAN[19], AGCRN[20], LSGCN[21]
- Graph agnostic point forecast models:
  - DeepGLO[22], N-BEATS[23], FC-GAGA[24]
- Graph agnostic probabilistic forecast models:
  - DeepAR[25], DeepFactors[26], MQRNN[27]

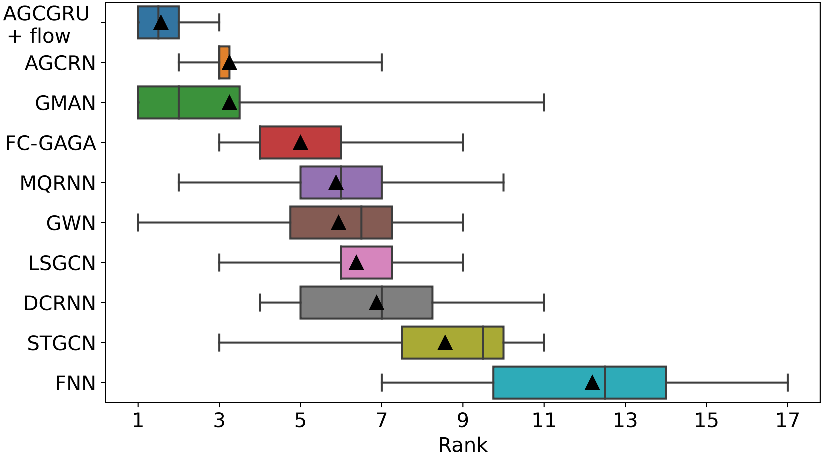
[11] Makridakis & Hibon 1997, [12] Hamilton 1994, [13] Chun-Hsin et al. 2004, [14] Sutskever et al. 2014

[15] Li et al. 2018, [16] Yu et al. 2018, [17] Guo et al. 2019, [18] Wu et al. 2019, [19] Zheng et al. 2020, [20] Bai et al. 2020, [21] Huang et al. 2021

[22] Sen et al. 2019, [23] Oreshkin et al. 2020, [24] Oreshkin et al. 2021

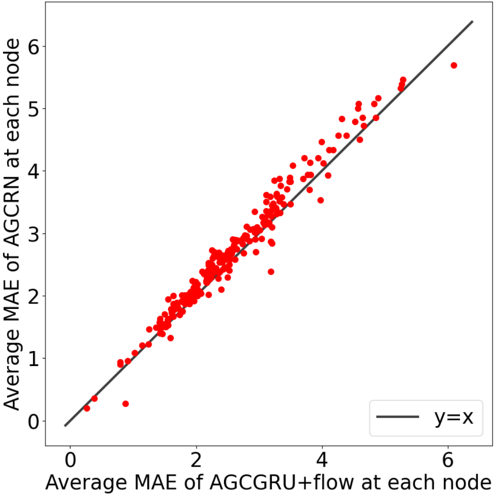
[25] Salinas et al. 2020, [26] Wang et al. 2019, [27] Wen et al. 2017

Experimental results: point forecasting



AGCGRU+flow achieves the best average rank.

Experimental results: node by node comparison



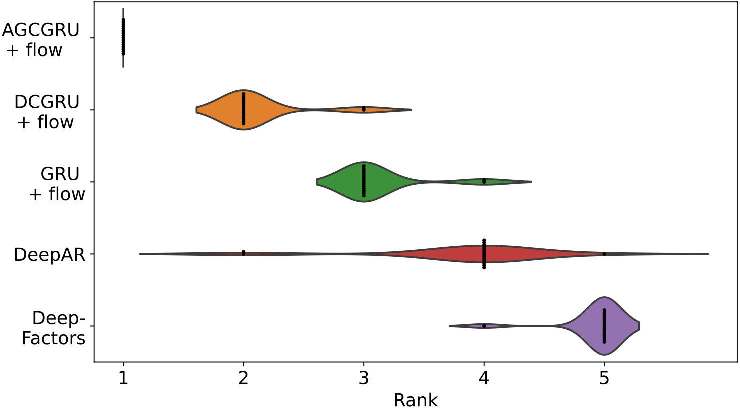
AGCGRU+flow outperforms AGCRN at majority of nodes in PeMSD7

Experimental results: probabilistic forecasting

$$\mathrm{CRPS}(F, x) = \int_{-\infty}^{\infty} \Big(F(z) - 1\{x \leqslant z\} \Big)^2 dz$$

Experimental results: probabilistic forecasting

$$\mathrm{CRPS}(F, x) = \int_{-\infty}^{\infty} \Big(F(z) - 1\{x \leqslant z\} \Big)^2 dz$$



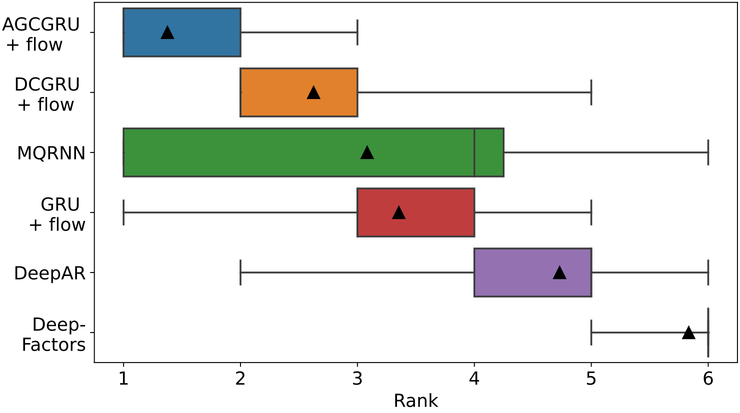
Our approaches obtain lower average CRPS.

Experimental results: quantile estimation

$$\mathrm{QL}\big(x, \hat{x}(\alpha)\big) = 2\Big(\alpha\big(x - \hat{x}(\alpha)\big)1\{x > \hat{x}(\alpha)\} + (1-\alpha)\big(\hat{x}(\alpha) - x\big)1\{x \leqslant \hat{x}(\alpha)\}\Big)$$

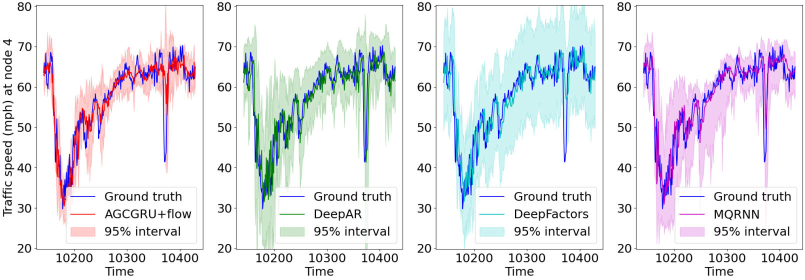
Experimental results: quantile estimation

$$\mathrm{QL}\big(x, \hat{x}(\alpha)\big) = 2\Big(\alpha\big(x - \hat{x}(\alpha)\big)1\{x > \hat{x}(\alpha)\} + (1-\alpha)\big(\hat{x}(\alpha) - x\big)1\{x \leqslant \hat{x}(\alpha)\}\Big)$$



AGCGRU+flow has the lowest quantile error on average.

Experimental results: confidence intervals



Confidence intervals for 15 minutes ahead predictions at node 4 of PeMSD7 for the first day in the test set.

Conclusion

– General Bayesian framework to represent forecast uncertainty

Conclusion

- General Bayesian framework to represent forecast uncertainty
- Can incorporate various RNNs, sophisticated inference tools

Conclusion

- General Bayesian framework to represent forecast uncertainty
- Can incorporate various RNNs, sophisticated inference tools
- Univariate/multivariate forecasting with/without graphs

Conclusion

- General Bayesian framework to represent forecast uncertainty
- Can incorporate various RNNs, sophisticated inference tools
- Univariate/multivariate forecasting with/without graphs
- Comparable point forecasting to *state-of-the-art*

Conclusion

- General Bayesian framework to represent forecast uncertainty
- Can incorporate various RNNs, sophisticated inference tools
- Univariate/multivariate forecasting with/without graphs
- Comparable point forecasting to *state-of-the-art*
- Better characterization of prediction uncertainty

Conclusion

- General Bayesian framework to represent forecast uncertainty
- Can incorporate various RNNs, sophisticated inference tools
- Univariate/multivariate forecasting with/without graphs
- Comparable point forecasting to *state-of-the-art*
- Better characterization of prediction uncertainty
- Results for non-graph data, component analyses in the paper

Conclusion

- General Bayesian framework to represent forecast uncertainty
- Can incorporate various RNNs, sophisticated inference tools
- Univariate/multivariate forecasting with/without graphs
- Comparable point forecasting to *state-of-the-art*
- Better characterization of prediction uncertainty
- Results for non-graph data, component analyses in the paper
- Code: `https://github.com/networkslab/rnn_flow`