BayOTIDE: Bayesian Online Multivariate Time Series Imputation with Functional Decomposition

Spotlight Paper of ICML 2024

Presenter: Shikai Fang

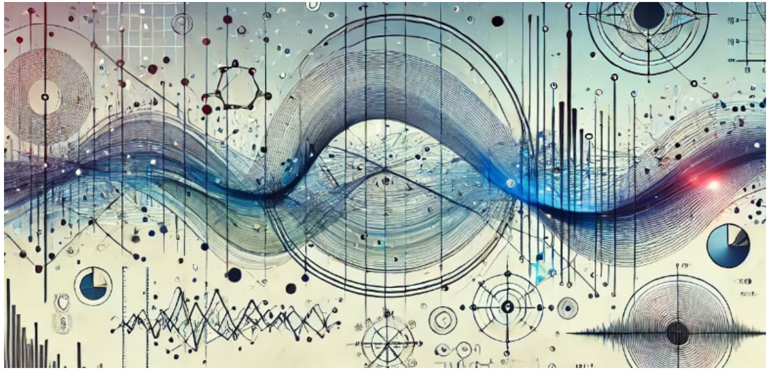
Shikai Fang, Qingsong Wen,
Yingtao Luo, Shandian Zhe, Liang Sun

Alibaba DAMO Academy

ICML International Conference On Machine Learning

THE UNIVERSITY OF UTAH

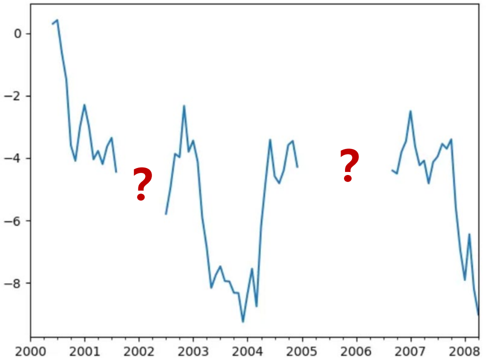
CARNEGIE MELLON UNIVERSITY PITTSBURGH PENNSYLVANIA 1900

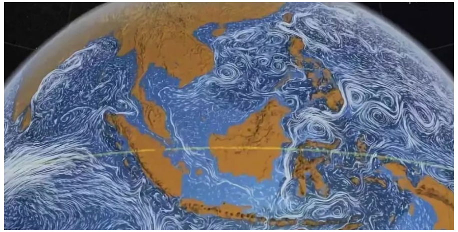


https://github.com/xuangu-fang/BayOTIDE

Imputation of Multi-Var. Time Series

- Time series data is ubiquitous
- The same with missing values…
- **Robust and efficient imputation** is crucial





Climate

Energy

Traffic

Finance

Limitations of current methods

- Regulate timestamp -> underuse **continuous** temporal info.
- Sensitive to noise -> call for **uncertainty-aware** model
- Black-box model -> lack of interpretability
- **Offline** infer. -> not efficient for fast-generated streaming seq.

BayOTIDE: Bayesian online TS Imputation

Properties / Methods	**BayOTIDE**	TIDER	Statistic-based	DNN-based	Diffusion-based
Uncertainty-aware	✓	✗	✗	✓✗	✓
Interpretability	✓	✓	✓	✗	✗
Continuous modeling	✓	✗	✗	✗	✓✗
Inference manner	**online**	offline	offline	offline	offline

Table 1: Comparison of *BayOTIDE* and main-stream multivariate time series imputation methods. ✓✗ means only partial models in the family have the property, or it's not clear in the original paper. For example, only deep models with probabilistic modules can offer uncertainty quantification, such as GP-VAE (Fortuin et al., 2020), but most deep models cannot. The diffusion-based CSDI (Tashiro et al., 2021) and CSBI (Chen et al., 2023) take timestamps as input, but the model is trained with discretized time embedding.

Method: Function Decomposition

- **Imputation ⇔ Low-rank function approximation**

$$\mathbf{X}(t) = \mathbf{U}\mathbf{V}(t) = [\mathbf{U}_{\text{trend}}, \mathbf{U}_{\text{season}}] \begin{bmatrix} \mathbf{v}_{\text{trend}}(t), \\ \mathbf{v}_{\text{season}}(t) \end{bmatrix}$$

$t \to \mathbb{R}^D$ $\quad \mathbb{R}^{D\times D_r}$ $\quad \mathbb{R}^{D\times D_s}$

Weights

$D_r + D_s << D$

Latent temporal functions (trend + seasonal factors)

$$\mathbf{v}_{\text{trend}}(t) = \text{concat}[v^i_{\text{trend}}(t)]_{i=1\ldots D_r},$$
$$\mathbf{v}_{\text{season}}(t) = \text{concat}[v^j_{\text{season}}(t)]_{j=1\ldots D_s},$$

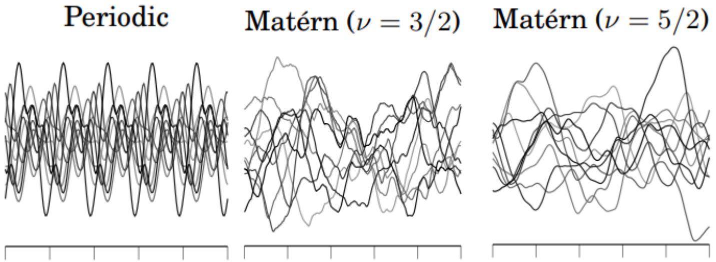
Gaussian Processes (GP) as function estimator

$$\begin{cases} \mathbf{v}_{\text{trend}}(t) = \text{concat}[v^i_{\text{trend}}(t)]_{i=1\ldots D_r}, \\ \mathbf{v}_{\text{season}}(t) = \text{concat}[v^j_{\text{season}}(t)]_{j=1\ldots D_s}, \end{cases}$$

$$\begin{cases} v^i_{\text{trend}}(t) \sim \mathcal{GP}(0, \kappa_{\text{Matérn}}) \\ v^j_{\text{season}}(t) \sim \mathcal{GP}(0, \kappa_{\text{periodic}}) \end{cases}$$

Facts of GPs:

- Powerful prob. functional model
- Characterized by kernel:



- Non-scalable : O(N^3) time cost

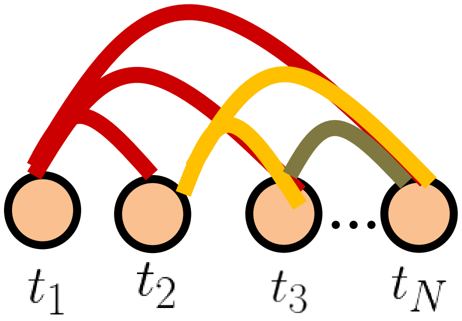
Linear-Cost GP with Chain-Structure

Temporal GPs

Temporal States:

LTI-SDE

$$\frac{\mathrm{d}\gamma_{\mathbf{r}}(t)}{\mathrm{d}t} = \mathbf{F}\gamma_{\mathbf{r}} + \mathbf{L}\xi(t)$$



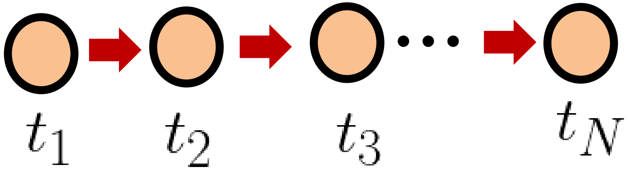
Space: $\mathcal{O}(N^2)$

Time: $\mathcal{O}(N^3)$

discrete form

$$p\left(\gamma_{\mathbf{r}}\left(t_{n+1}\right) \mid \gamma_{\mathbf{r}}\left(t_n\right)\right) = \mathcal{N}\left(\gamma_{\mathbf{r}}\left(t_{n+1}\right) \mid \mathbf{A}_n\gamma_{\mathbf{r}}\left(t_n\right), \mathbf{Q}_n\right)$$

State Space Model



Space: $\mathcal{O}(N)$

Time: $\mathcal{O}(N)$

Joint Prob. of BayOTIDE

Noise

Channel-wise weight

Joint prob. $p(\mathbf{Y}, \mathbf{V}(t), \mathbf{U}, \tau) = \text{Gam}(\tau \mid a_0, b_0) \prod_{d=1}^{D} \mathcal{N}(\mathbf{u}^d \mid \mathbf{0}, \mathbf{I})$

O(N^3) $\prod_{j=1}^{D_s} \mathcal{GP}(0, \kappa_{\text{periodic}}) \prod_{i=1}^{D_r} \mathcal{GP}(0, \kappa_{\text{Matérn}}) \cdot p(\mathbf{Y}|\mathbf{U}, \mathbf{V}(t), \tau)$

Likelihood

O(N)! $P(\mathbf{Z}(t_1)) \prod_{i=1}^{N-1} P(\mathbf{Z}(t_{n+1})|\mathbf{Z}(t_n))$

States of temporal factors.

Bayesian Online Learning

- **Online learning/ Straming Inference:**
  data come, model update, date drops

- Principnple: Incremental version of Bayes' rule:

Posterior on old data

$$p(\boldsymbol{\theta} \mid \mathcal{D}_{\text{old}} \cup \mathcal{D}_{\text{new}}) \propto p(\boldsymbol{\theta} \mid \mathcal{D}_{\text{old}})\, p(\mathcal{D}_{\text{new}} \mid \boldsymbol{\theta})$$

Posterior on all data **Likelihood on current model**

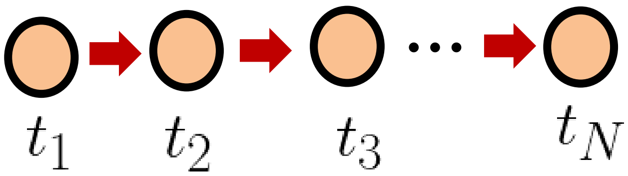
Online Learning ⇔ Kalman Filter!

Incremental version of Bayes' rule:

$$p(\boldsymbol{\theta} \mid \mathcal{D}_{\text{old}} \cup \mathcal{D}_{\text{new}}) \propto p(\boldsymbol{\theta} \mid \mathcal{D}_{\text{old}})\, p(\mathcal{D}_{\text{new}} \mid \boldsymbol{\theta})$$

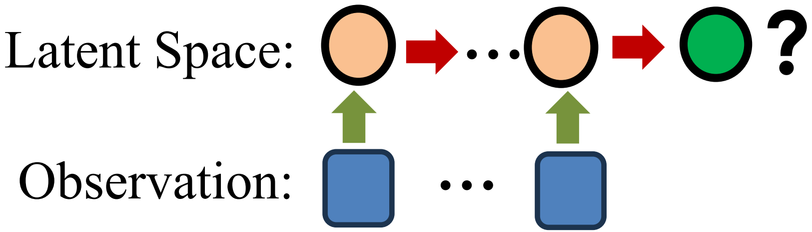
+

Chain-Struture of latent factors



→

Kalman Filter!

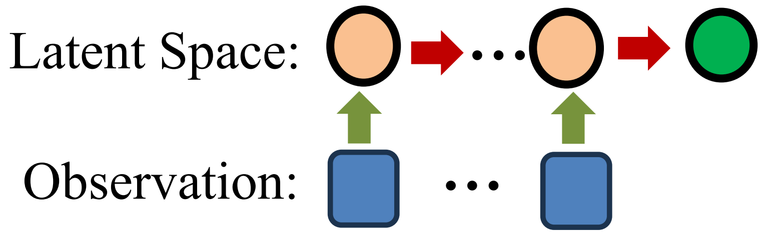


Moment Matching & Message Passing

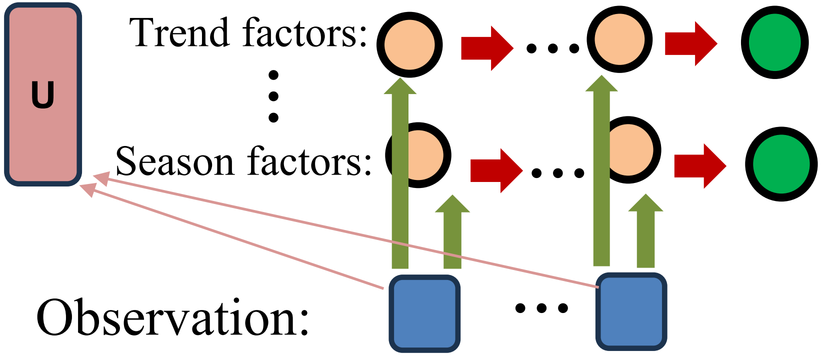
Last Challenge:

- One obvervation ⟷ ? States of multi-factors + weights + noise

- Classical Kalman Filter



- BayOTIDE



Moment Matching & Message Passing

- Conditional Moment-Matching

$$p(y_{n+1}^d \mid \Theta) \approx Z f_{n+1}^d(\mathbf{Z}(t_{n+1})) f_{n+1}^d(\mathbf{u}_d) f_{n+1}^d(\tau)$$

Observation llk. (Gaussian)

Msgs to **multi**-factors (Gaussian)

Msgs to **weights and noise** (Gaussian & Gamma)

- Message passing and merging

$$q(\tau|\mathcal{D}_{t_{n+1}}) = q(\tau|\mathcal{D}_{t_n}) \prod_{d=1}^{D} f_{n+1}^d(\tau)$$

$$q(\mathbf{u}^d|\mathcal{D}_{t_{n+1}}) = q(\mathbf{u}^d|\mathcal{D}_{t_n}) f_{n+1}^d(\mathbf{u}^d)$$

$$q(\mathbf{Z}(t_{n+1})) = q(\mathbf{Z}(t_n)) p(\mathbf{Z}(t_{n+1})|\mathbf{Z}(t_n)) \prod_{d=1} f_{n+1}^d(\mathbf{Z}(t_{n+1}))$$

All closed-form update!

Algorithm of BayOTIDE

Algorithm 1 *BayOTIDE*

Input: observation $\mathbf{Y} = \{\mathbf{y}_n\}_{n=1}^N$ over $\{t_n\}_{n=1}^N$, D_s, D_r, the kernel hyperparameters.

Initialize $q(\tau), q(\mathcal{W}), \{q(\mathbf{Z}(t_n))\}_{n=1}^N$.

for $t = 1$ **to** N **do**

Approximate messages by (12) for all observed channels in parallel.

Update posterior of τ and $\mathbf{U}$ by (13) and (14) for all observed channels in parallel.

Update posteriors of $\mathbf{Z}(t)$ using Kalman filter by (15).

end for

Run RTS smoother to obtain the full posterior of $\mathbf{Z}(t)$.

Return: $q(\tau), q(\mathcal{W}), \{q(\mathbf{Z}(t_n))\}_{n=1}^N$

Could be irregulate

- **Parallel over channels**
- **Online update for all para.**

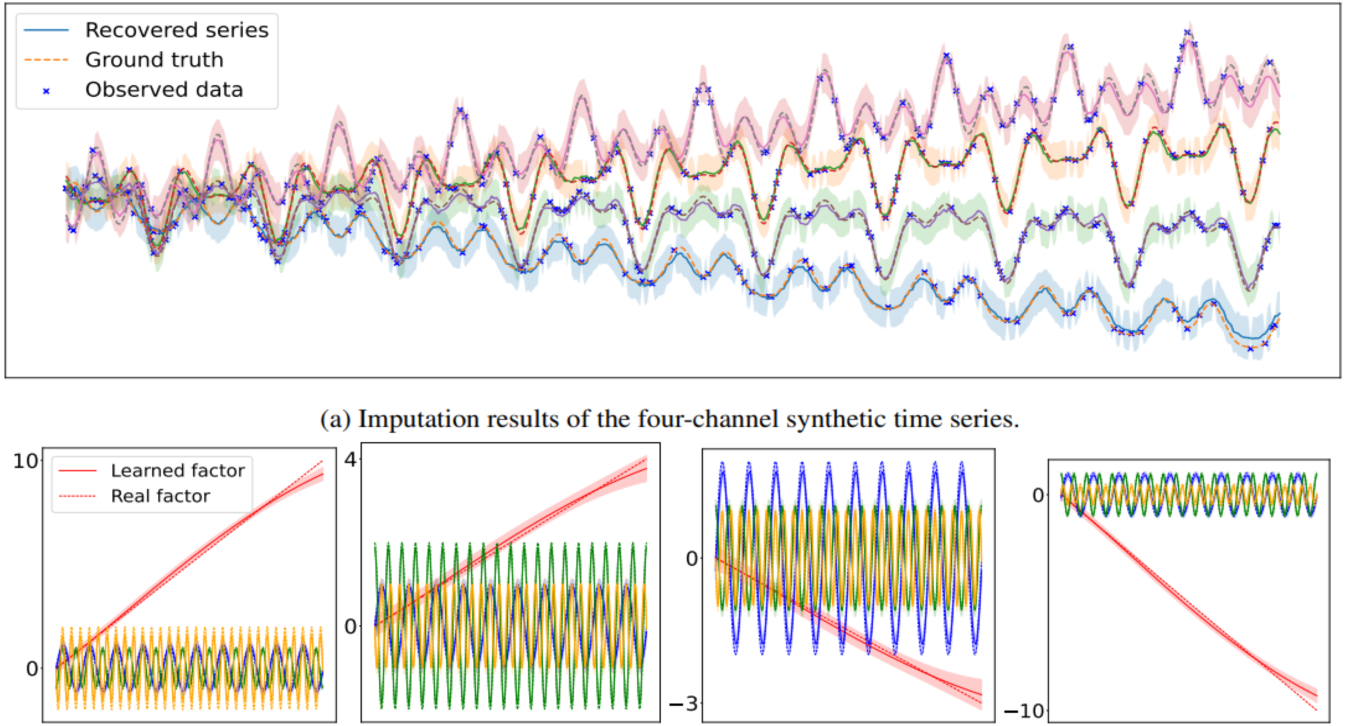
Allow prob. imputation over **arbitrary timestamp** (even never seen in training)

Experiments: Simulation

- **20% noisy observations**

$$\mathbf{U} = \begin{pmatrix} 1 & 1 & -2 & -2 \\ 0.4 & 1 & 2 & -1 \\ -0.3 & 2 & 1 & -1 \\ -1 & 1 & 1 & 0.5 \end{pmatrix},$$

$$\mathbf{V}(t) = \begin{pmatrix} 10t, \\ \sin(20\pi t), \\ \cos(40\pi t), \\ \sin(60\pi t) \end{pmatrix}.$$



(a) Imputation results of the four-channel synthetic time series.

(b) Channel#1's factors (c) Channel#2's factors (d) Channel#3's factors (e) Channel#4's factors

Experiments on real-world tasks

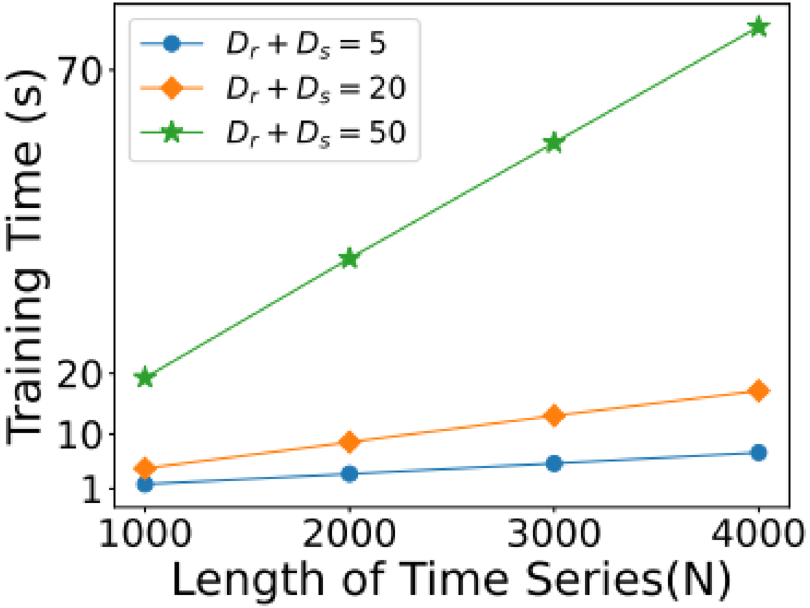
Online beats offline!

Observed-ratio=50%	*Traffic-GuangZhou*			*Solar-Power*			*Uber-Move*		
Metrics	RMSE	MAE	CRPS	RMSE	MAE	CRPS	RMSE	MAE	CRPS
Deterministic & Offline									
SimpleMean	9.852	7.791	-	3.213	2.212	-	5.183	4.129	-
BRITS	4.874	3.335	-	2.842	1.985	-	2.180	1.527	-
NAOMI	5.986	4.543	-	2.918	2.112	-	2.343	1.658	-
SAITS	4.839	3.391	-	2.791	1.827	-	1.998	1.453	-
TIDER	4.708	3.469	-	**1.679**	0.838	-	1.959	1.422	-
Probabilistic & Offline									
Multi-Task GP	4.887	3.530	0.092	2.847	1.706	0.203	3.625	2.365	0.121
GP-VAE	4.844	3.419	0.084	3.720	1.810	0.368	5.399	3.622	0.203
CSDI	4.813	3.202	0.076	2.276	0.804	0.166	1.982	1.437	0.072
CSBI	4.790	3.182	0.074	2.097	1.033	0.153	1.985	1.441	0.075
Probabilistic & Online									
BayOTIDE-fix weight	11.032	9.294	0.728	5.245	2.153	0.374	5.950	4.863	0.209
BayOTIDE-trend only	4.188	2.875	0.059	1.789	0.791	0.132	2.052	1.464	0.067
BayOTIDE	**3.820**	**2.687**	**0.055**	1.699	**0.734**	**0.122**	**1.901**	**1.361**	**0.062**

Table 2: RMSE, MAE and CRPS scores of imputation results of all methods on three datasets with observed ratio = 50%.

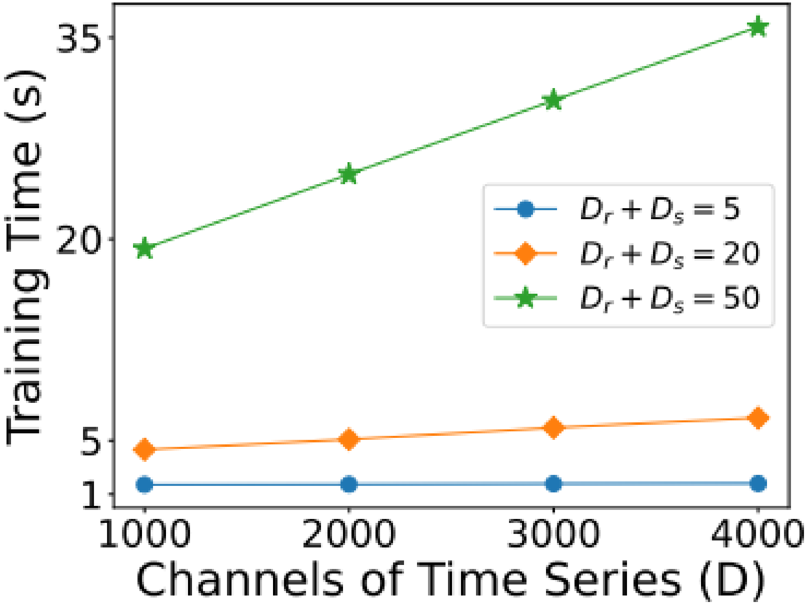
Scalability of BayOTIDE

D fixed as 1000



(b) Scalability over time series length.

N fixed as 1000



(c) Scalability over the channels number.

Thanks.

Github Repo

https://github.com/xuangu-fang/BayOTIDE