THE UNIVERSITY OF UTAH®

Streaming Factor Trajectory Learning for Temporal Tensor Decomposition

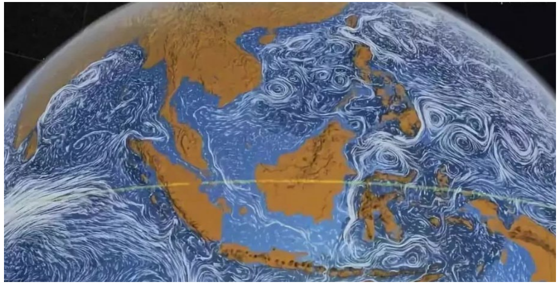
Shikai Fang, Xin Yu, Shibo Li, Zheng Wang, Robert Kirby, Shandian Zhe

NeurIPS 2023

Github Repo: https://github.com/xuangu-fang/Streaming-Factor-Trajectory-Learning

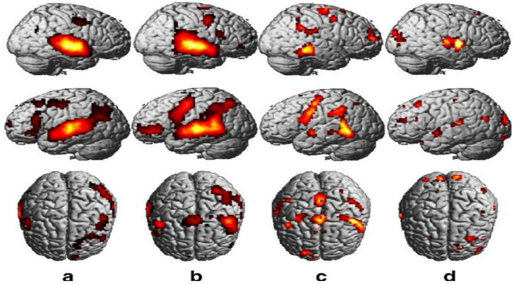
Tensor Date

- Multi-dim Array
- Represent interactions of multiple objects/entitiles
- Each entry: (index1, index2..)-> value



(location, region, time, climate)

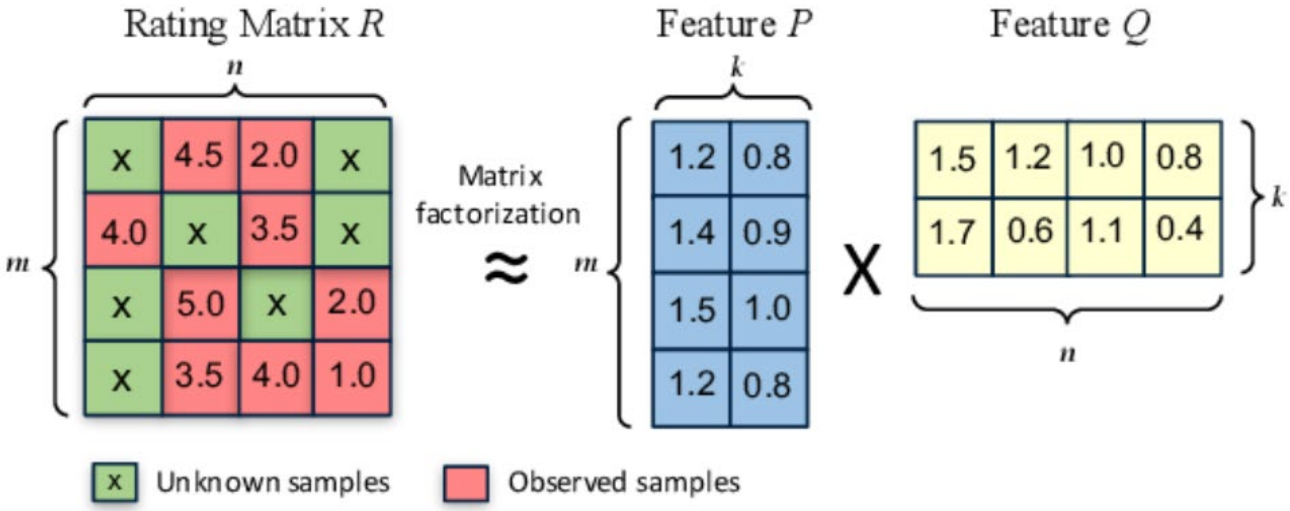
(user, movie, episode)



(subject, voxel, electrode)

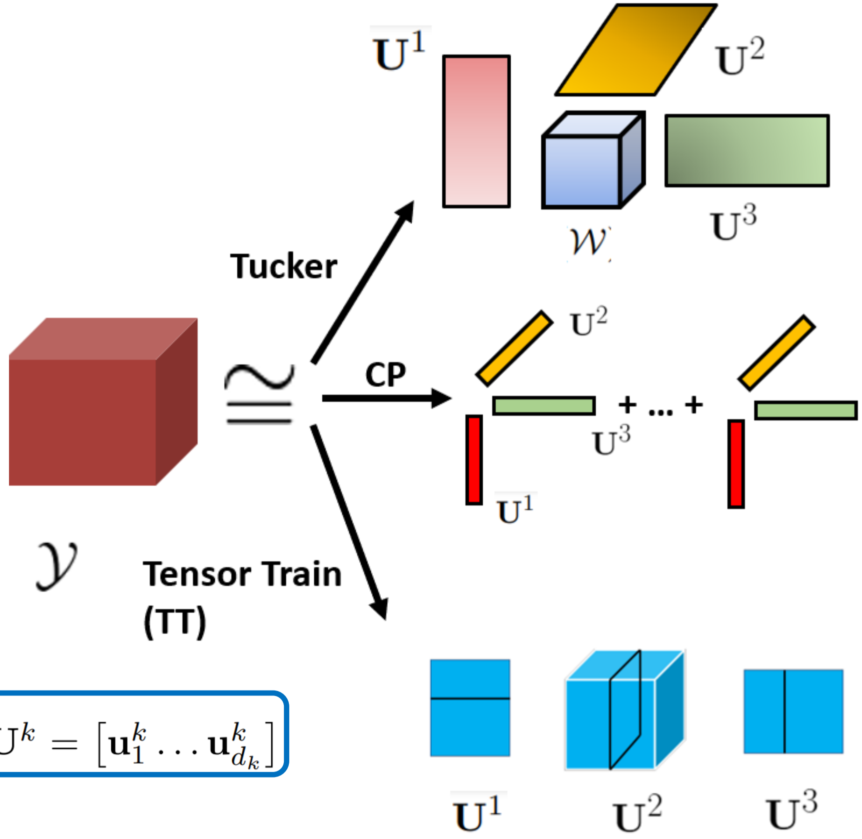
Tensor Decomposition

- Learning representation of latent factors
- Simple case: **Collaborative Filterling** (Matrix Factorization)



NETFLIX

CP / Tucker / TT Decomposition



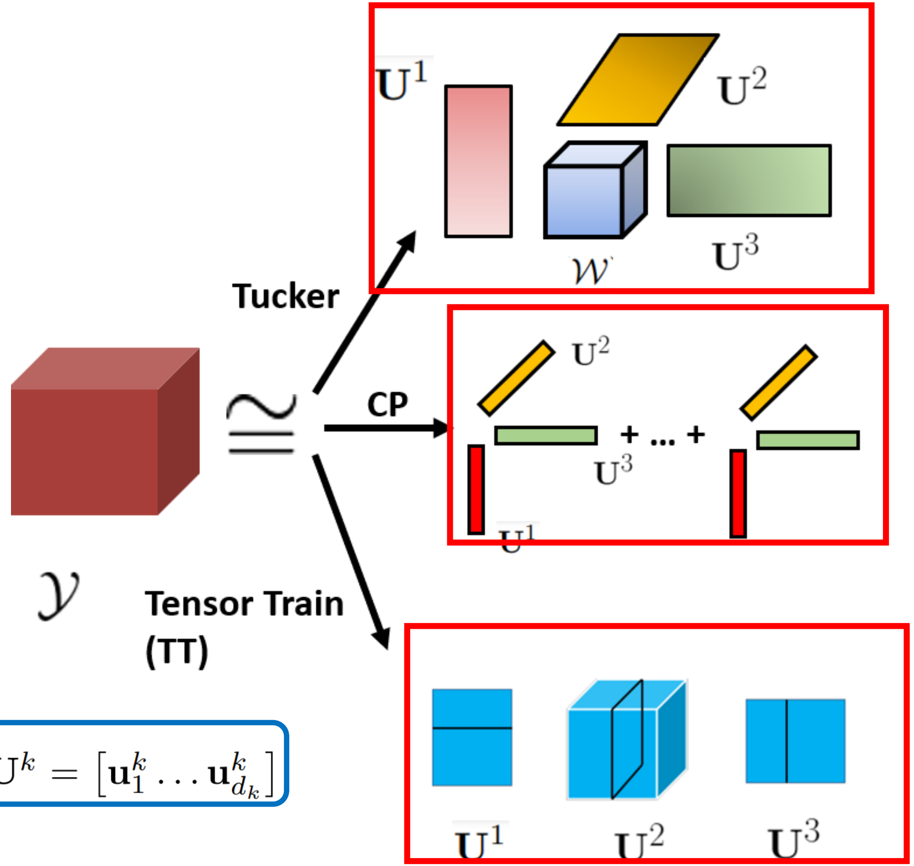
$\mathbf{U}^k = [\mathbf{u}^k_1 \dots \mathbf{u}^k_{d_k}]$

$$y_{\mathbf{i}} \approx \mathrm{vec}(\mathcal{W})^\top \left(\mathbf{u}^1_{i_1} \otimes \dots \otimes \mathbf{u}^K_{i_K}\right)$$

$$y_{\mathbf{i}} \approx \left(\mathbf{u}^1_{i_1} \circ \dots \circ \mathbf{u}^K_{i_K}\right)^\top \mathbf{1}$$

$$y_{\mathbf{i}} \approx \underbrace{\mathbf{u}^1_{i_1}}_{r_0 \times r_1} \underbrace{\mathbf{u}^2_{i_2}}_{r_1 \times r_2} \cdots \underbrace{\mathbf{u}^K_{i_K}}_{r_{K-1} \times r_K}$$

Bayesian Tensor Decomposition



$$\mathbf{U}^k = [\mathbf{u}_1^k \dots \mathbf{u}_{d_k}^k]$$

Deterministic factors:
Vectors

Probabilistic factors :
Distributions

$$\mathcal{N}\left(\mathbf{u}_s^k \mid \mathbf{u}_s^k, v\mathbf{I}\right)$$

Uncertainty counts!

Example: Bayesian CP Decomposition

Joint Distribution

Priors of factors and noise

$$p\left(\{y_{\mathbf{i}}\}_{\mathbf{i}\in S}, \mathcal{U}, \tau\right) = \boxed{\mathrm{Gam}\left(\tau \mid a_0, b_0\right) \prod_{k=1}^{K} \prod_{s=1}^{d_k} \mathcal{N}\left(\mathbf{u}_s^k \mid \mathbf{m}_s^k, v\mathbf{I}\right)}$$

$$\prod_{i\in S} \boxed{\mathcal{N}\left(y_{\mathbf{i}} \mid \mathbf{1}^\top \left(\mathbf{u}_{i_1}^1 \circ \ldots \circ \mathbf{u}_{i_K}^K\right), \tau^{-1}\right)}$$

CP likelihood

Approx. Posterior

$$p\left(\mathcal{U}, \tau \mid \{y_{\mathbf{i}}\}_{\mathbf{i}\in S}\right) \approx q(\mathcal{U}, \tau) = q(\tau) \prod_{k=1}^{K} \prod_{s=1}^{d_k} q\left(\mathbf{u}_s^k\right)$$

Inference!

$$= \mathrm{Gamma}(\tau \mid \boxed{a^\star, b^\star}) \prod_{k=1}^{K} \prod_{s=1}^{d_k} \mathcal{N}\left(\mathbf{u}_s^k \mid \boxed{\boldsymbol{\mu}_s^{k\,\star}, \boldsymbol{\Sigma}_s^{k\,\star}}\right)$$

Bayesian inference toolbox

Variational inference(VI):

- minimize the **KL divergence** of exact posterior and approx. posterior

$$\mathcal{L} = \int q^*(\mathcal{U}, \tau) \log \frac{p\left(\{y_i\}_{i \in S_t} \mid \mathcal{U}, \tau\right) q(\mathcal{U}, \tau)}{q^*(\mathcal{U}, \tau)} \mathrm{d}\mathcal{U} \mathrm{d}\tau$$

Expectation propagation(EP):

- **moment match** when L(reverse order) has **closed form solution**

Assumed density filtering(ADF):

- **moment match** when having **tractable normalization term**

Reparameterization trick (SVI):

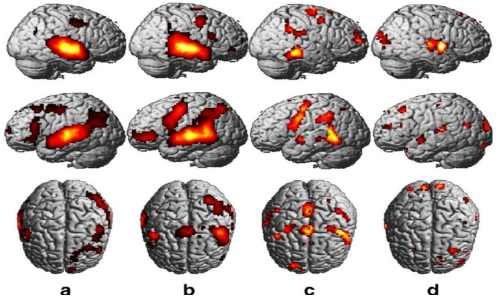
- **Probabilistic version SGD** when **L is totally intractable**

…(sampling based methods)

Temporal Tensor Data

- Tensor-valued time series
- Represent time-varing and high-order interactions

(region, site, weather)



(subject, voxel, electrode)

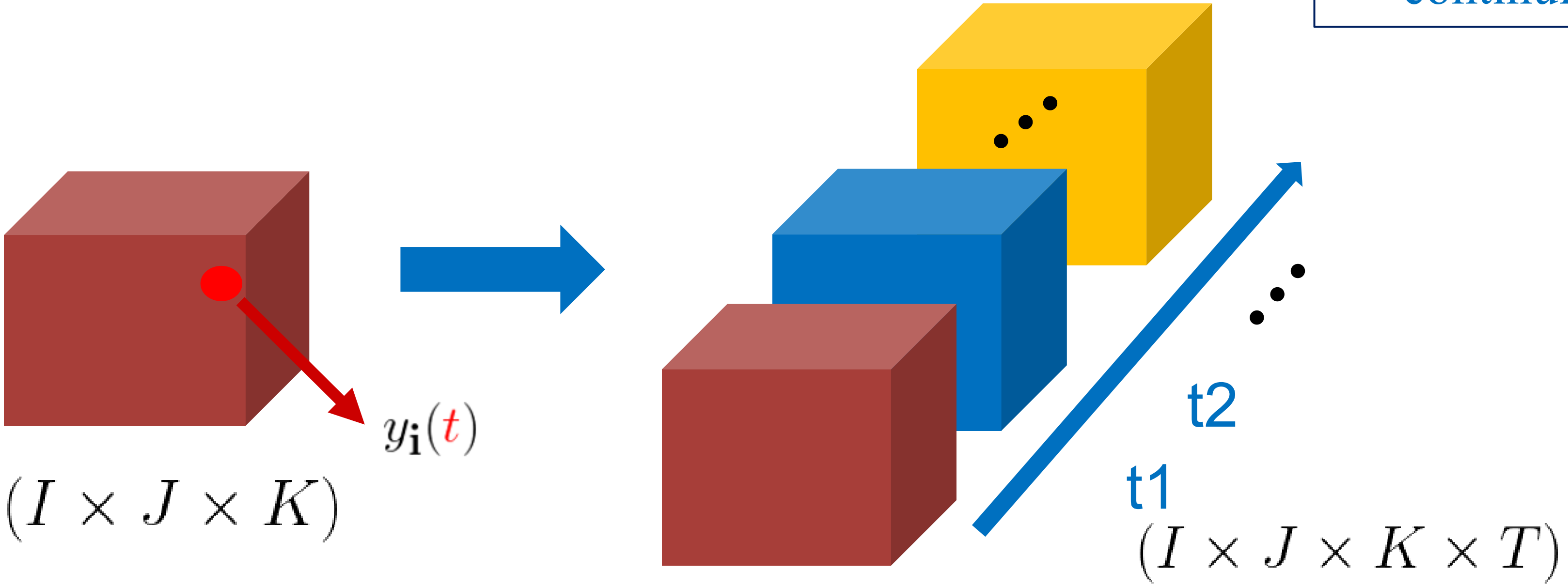
(user, user,location, message)

Tensor structure are evolving through time!

Temporal Tensor: trival solutions

- Drop timestamps
- Augment tensor with discrete time mode

- Too Sparse!
- Loss temporal continuity



Temporal Tensor Decomposition

Most existing work[1][2][3]:

Evolving weights + Static Factors

$$y_{\mathbf{i}}(t) \approx \mathbf{w}(t)^\top \left(\mathbf{u}^1_{i_1} \circ \ldots \circ \mathbf{u}^K_{i_K}\right)$$

- Over-Simplistic!
- **Evolving factors** dominate in many cases

[1] Zhang, Yanqing, et al. "Dynamic tensor recommender systems." *The Journal of Machine Learning Research* 22.1 (2021): 3032-3066.
[2] Fang, Shikai, et al. "Bayesian Continuous-Time Tucker Decomposition." *International Conference on Machine Learning*. PMLR, 2022.
[3] Li, Shibo, et al.. "Decomposing Temporal High-Order Interactions via Latent ODEs." *International Conference on Machine Learning*. PMLR, 2022.

SFTL

SFTL(ours): Learning functional factor trajectories!

CP: $$y_{\mathbf{i}}(t) \approx \mathbf{w}^\top \left(\mathbf{u}^1_{i_1}(t) \circ \ldots \circ \mathbf{u}^K_{i_K}(t)\right)^\top$$

Tucker: $$y_{\mathbf{i}}(t) \approx \text{vec}(\mathcal{W})^\top \left(\mathbf{u}^1_{i_1}(t) \otimes \ldots \otimes \mathbf{u}^K_{i_K}(t)\right)$$

Gaussian Process for Trajectory Learning

Gaussian Process(GP): Bayesian Functional Prior

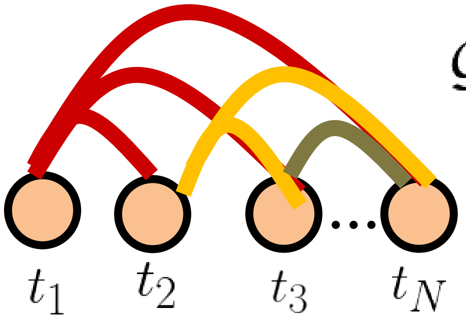
$$y_{\mathbf{i}}(t) \approx \mathbf{w}^\top \underbrace{\left(\boxed{\mathbf{u}^1_{i_1}(t)} \circ \boxed{\ldots} \circ \boxed{\mathbf{u}^K_{i_K}(t)} \right)}_{GP(0,\, k(t,t'))}^\top$$

- Non-parametric: strong and flexiable
- **Non-scalable : O(N^3) time cost**

Linear-Cost GP with Chain-Structure

Temporal GPs

Temporal States:



$$\mathcal{GP}\left(\mathbf{w_r} \mid \mathbf{0}, \mathbf{K_r}(t, t')\right)$$

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

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

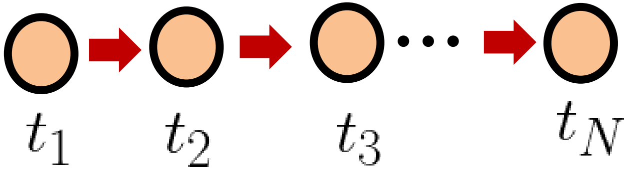
LTI-SDE

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

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 SFTL (CP)

noise

states of factor trajectory

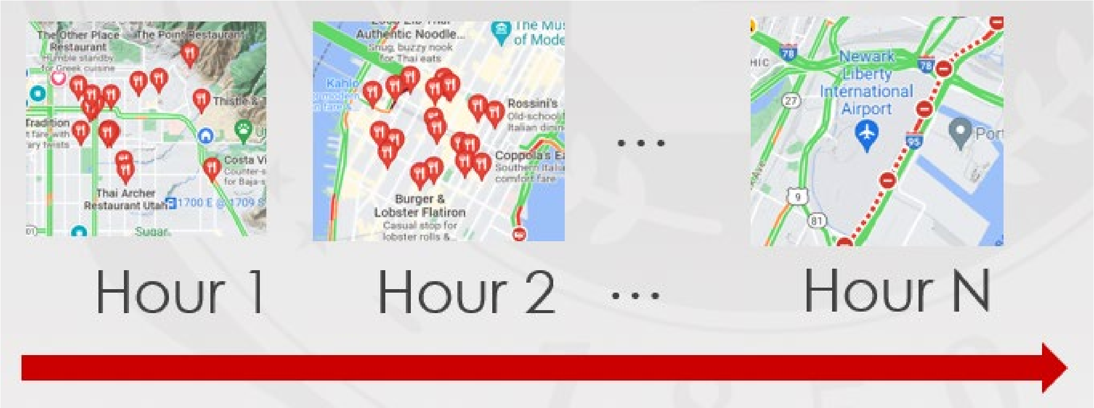
$$p(\{\mathbf{h}_{j,k}^m\}, \tau, \mathbf{y}) = \underbrace{\boxed{p(\tau)}}_{\text{noise}} \prod_{m=1}^{M} \prod_{j=1}^{d_m} \underbrace{\boxed{p(\mathbf{h}_{j,1}^m) \prod_{k=1}^{c_j^m - 1} p(\mathbf{h}_{j,k+1}^m | \mathbf{h}_{j,k}^m)}}_{\text{states of factor trajectory}}$$

$$\cdot \prod_{n=1}^{N} \underbrace{\boxed{\mathcal{N}(y_n | \mathbf{1}^\top (\mathbf{u}_{\ell_{n1}}^1(t_n) \circ \ldots \circ \mathbf{u}_{\ell_{nM}}^M(t_n)), \tau^{-1})}}_{\text{likelihood}}.$$

likelihood

Inference Challenges

- One obvervation ⟷ interweaving of multi-trajectories
- Datasets are collected in **fast-generating & streaming** manner



Inference Challenges

Security / Privacy / Safety-constrained senarios

- Data are **NOT** allowed to be stored/revisited

Classical Offline Algorithems

- **Offline learning** of classical tensor models:

$$\min_{U\in\mathbb{R}^{m\times d}, V\in\mathbb{R}^{n\times d}} \sum_{(i,j)\in \text{ obs}} \boxed{(A_{ij} - \langle U_i, V_j\rangle)^2} + w_0 \boxed{\sum_{(i,j)\notin \text{ obs}} (\langle U_i, V_j\rangle)^2}$$

- Need to collect **<u>all observations</u>** at first
- Then go through data for **<u>multi-epochs</u>** to update

Can not fit **streaming** tesnor data!

Bayesian Straeming Inference

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

- Princinple: Incremental version of Bayes' rule:

Posterior on old data

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

Posterior on all data

Likelihood on current model

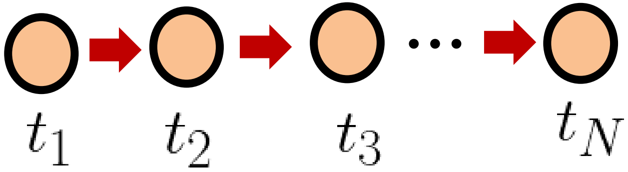
Streaming Inference ⇔ Kalman Filter!

Incremental version of Bayes' rule:

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

$+$

Chain-Struture of factor trajactoy

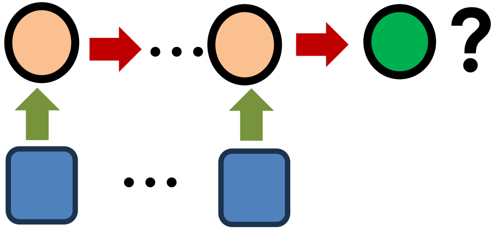


$\longrightarrow$

Kalman Filter!

Latent Space:

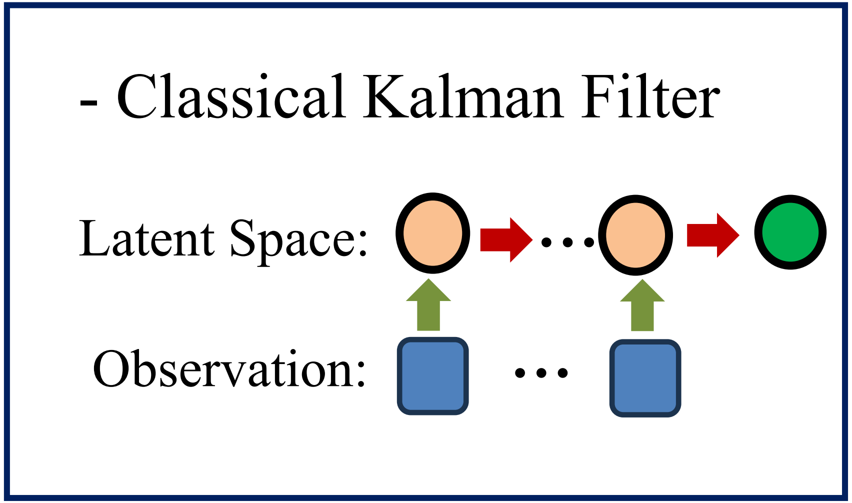
Observation:

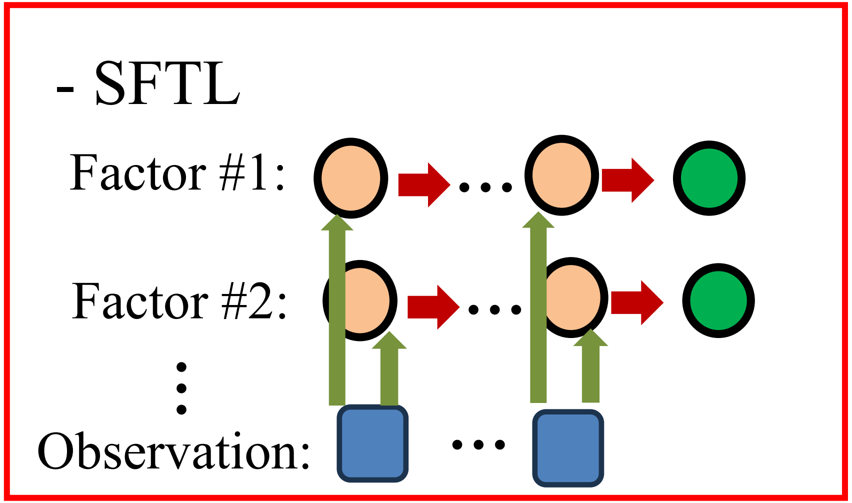


Message Passing & Moment Matching

Last Challenge:

- One obvervation ⟷(?) States of multi-trajectories





Message Passing & Moment Matching

- Conditional Moment-Matching

$$\boxed{\mathcal{N}\left(y|\mathbf{1}^\top\left(\mathbf{u}^1_{\ell_1}(t_{n+1})\circ\cdots\circ\mathbf{u}^M_{\ell_M}(t_{n+1})\right)\right)} \propto \boxed{\prod_{m=1}^{M}\mathcal{N}\left(\mathbf{u}^m_{\ell_m}(t_{n+1})|\boldsymbol{\gamma}^m_{\ell_m},\boldsymbol{\Sigma}^m_{\ell_m}\right)}\mathrm{Gam}(\tau|\alpha_{\boldsymbol{\ell}},\omega_{\boldsymbol{\ell}})$$

Observation

Msgs to **multi**-trajatories

- Message Passing (**Expectation Propagation**)

$$\mathbb{E}_q[\mathbf{v}^{\setminus m}_{\boldsymbol{\ell}}] = \mathbb{E}_q[\mathbf{v}^1_{\ell_1}]\circ\ldots\circ\mathbb{E}_q[\mathbf{v}^{m-1}_{\ell_{m-1}}]\circ\mathbb{E}_q[\mathbf{v}^{m+1}_{\ell_{m+1}}]\circ\ldots\circ\mathbb{E}_q[\mathbf{v}^M_{\ell_M}],$$

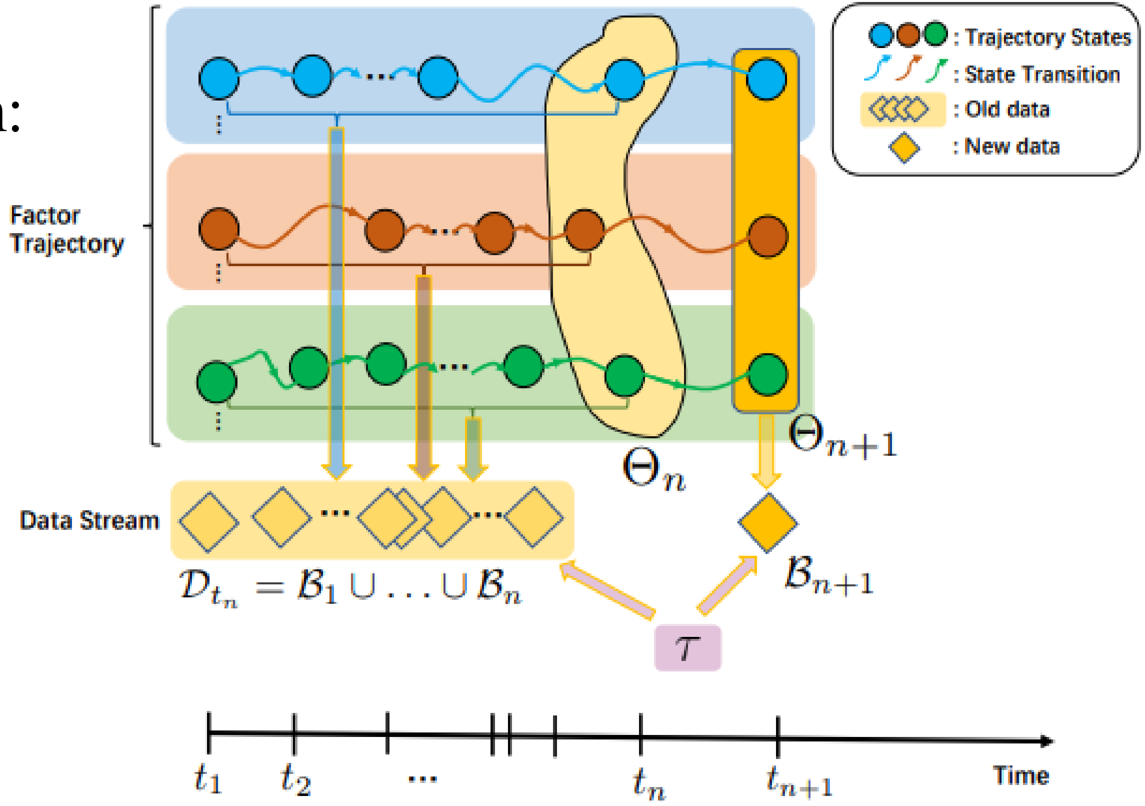
$$\mathbb{E}_q[\mathbf{v}^{\setminus m}_{\boldsymbol{\ell}}\left(\mathbf{v}^{\setminus m}_{\boldsymbol{\ell}}\right)^\top] = \mathbb{E}_q[\mathbf{v}^1_{\ell_1}\left(\mathbf{v}^1_{\ell_1}\right)^\top]\circ\ldots\circ\mathbb{E}_q[\mathbf{v}^{m-1}_{\ell_{m-1}}\left(\mathbf{v}^{m-1}_{\ell_{m-1}}\right)^\top]$$

$$\circ\,\mathbb{E}_q[\mathbf{v}^{m+1}_{\ell_{m+1}}\left(\mathbf{v}^{m+1}_{\ell_{m+1}}\right)^\top]\circ\ldots\circ\mathbb{E}_q[\mathbf{v}^M_{\ell_M}\left(\mathbf{v}^M_{\ell_M}\right)^\top].$$

Inference Algorithem of SFTL

Efficient Online algorithem:

- Closed-form Updates
- Conditional Moment Match
- Kalman Filter
  + (RTS Smoother)



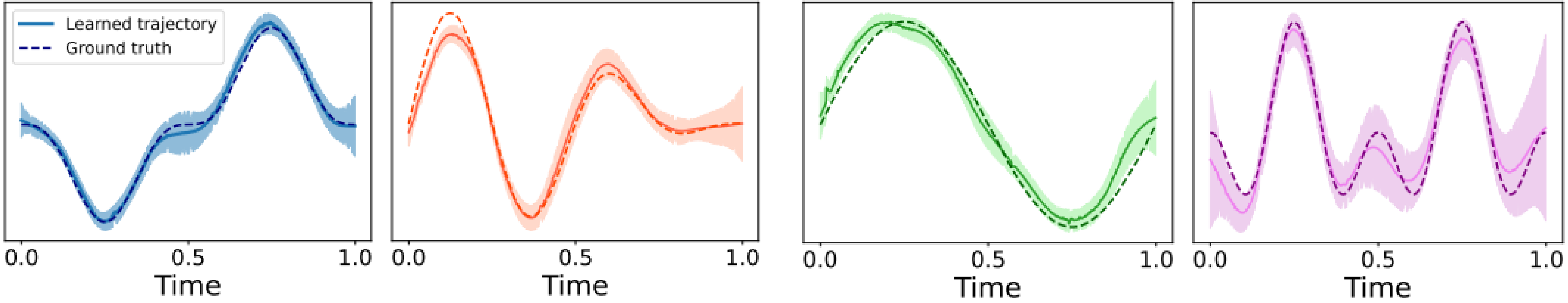
Inference Algorithem of SFTL

Algorithm 1 Streaming Factor Trajectory Learning (SFTL)

1: **Input:** kernel hyper-parameters a, ρ, $\nu = p + \frac{1}{2}$ $(p \in \{0, 1, 2, \dots\})$
2: $n \leftarrow 0$
3: **while** Receiving a new batch of entreis $\mathcal{B}_{n+1}$ **do**
4:     **if** $n = 0$ **then**
5:         Set $a_n = a_0$, $b_n = b_0$, $\widehat{\boldsymbol{\mu}}^m_{j, c^m_{j,n+1}} = \mathbf{0}$, and $\widehat{\mathbf{V}}^m_{j, c^m_{j,n+1}} = \overline{\mathbf{P}}_\infty$ in (8).
6:         Goto 9.
7:     **end if**
8:     Retrieve the involved preceding factor states $\Theta_n = \{\mathbf{h}^m_{j, c^m_{j,n}} | j \in \mathcal{I}^m_{n+1}\}_m$ and their running posterior, $p(\Theta_n, \tau | \mathcal{D}_{t_n}) \approx \text{Gam}(\tau | a_n, b_n) \prod_{m=1}^M \prod_{j \in \mathcal{I}^m_{n+1}} \mathcal{N}(\mathbf{h}^m_{j, c^m_{j,n}} | \widehat{\boldsymbol{\mu}}^m_{j, c^m_{j,n}}, \widehat{\mathbf{V}}^m_{j, c^m_{j,n}})$.
9:     According to (8) and (9), use conditional Expectation Propagation to calculate the running posterior of the current factor states, $p(\Theta_{n+1}, \tau | \mathcal{D}_{t_{n+1}}) \approx \text{Gam}(\tau | a_{n+1}, b_{n+1}) \prod_{m=1}^M \prod_{j \in \mathcal{I}^m_{n+1}} \mathcal{N}(\mathbf{h}^m_{j, c^m_{j,n+1}} | \widehat{\boldsymbol{\mu}}^m_{j, c^m_{j,n+1}}, \widehat{\mathbf{V}}^m_{j, c^m_{j,n+1}})$.
10:     **if** Needed **then**
11:         Run RTS smoothing on any factor state chain $\{\mathbf{h}^m_{j,k} | k = 1, 2, \dots\}$ of interest.
12:     **end if**
13:     $n \leftarrow n + 1$
14: **end while**
15: Run RTS smoothing for every factor state chain $\{\mathbf{h}^m_{j,k} | k = 1, 2, \dots\}$.
16: **Return:** $\{q(\mathbf{h}^m_{j,k} | \mathcal{D}) | k = 1, 2, \dots\}_{1 \le m \le M, 1 \le j \le d_m}$, $q(\tau | \mathcal{D})$, where $\mathcal{D}$ is all the data received.

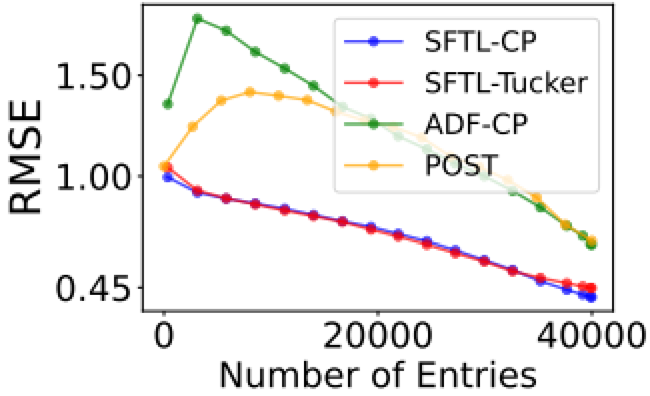
Experiments: Simulation

$$y_{(i,j)}(t) \sim \mathcal{N}\left(u_i^1(t)u_j^2(t), 0.05\right)$$

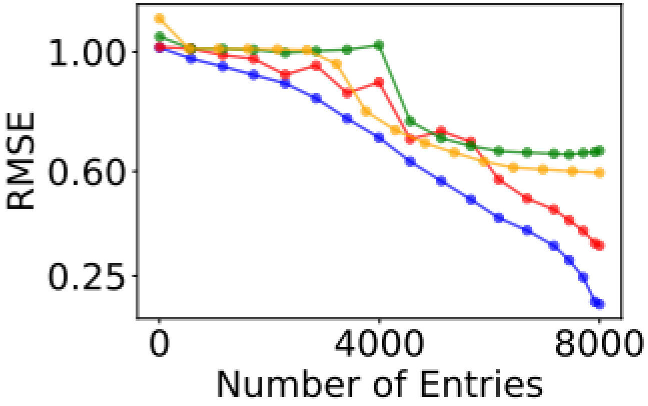


(a) $u_1^1(t)$ (b) $u_2^1(t)$ (c) $u_1^2(t)$ (d) $u_2^2(t)$

Experiments: Online Prediction



(a) *FitRecord*



(b) *ServerRoom*

Experiments: Prediction Accuracy

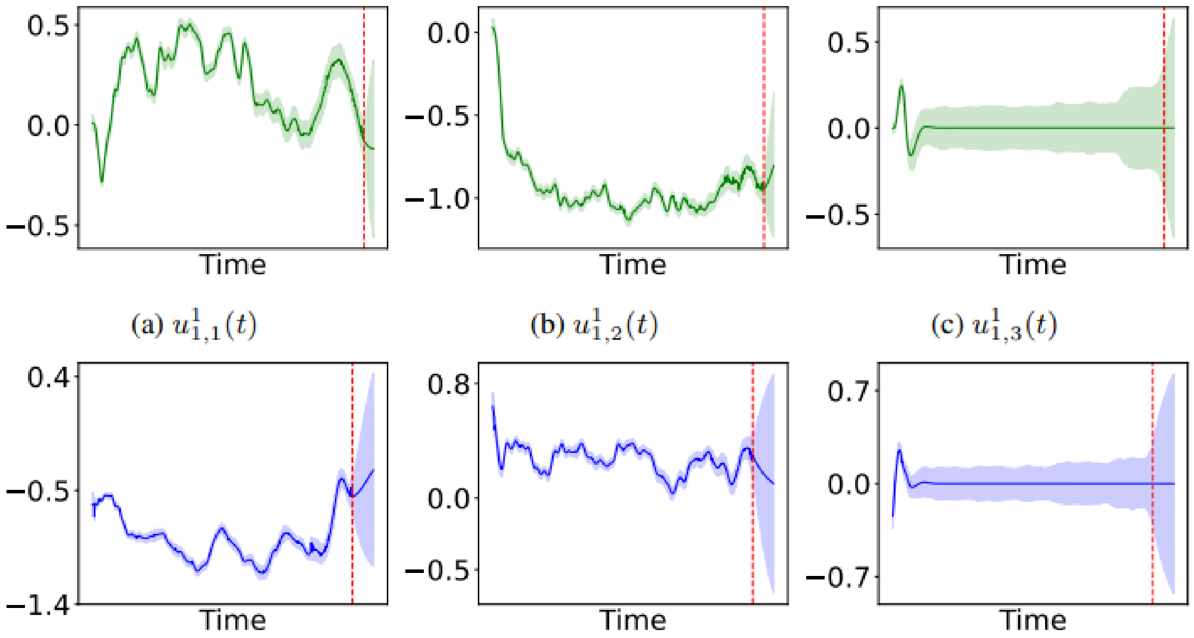
- **Online beat Offline!**

	RMSE	*FitRecord*
Static	PTucker	0.656 ± 0.147
	Tucker-ALS	0.846 ± 0.005
	CP-ALS	0.882 ± 0.017
	CT-CP	0.664 ± 0.007
	CT-GP	0.604 ± 0.004
	BCTT	0.518 ± 0.007
	NONFAT	0.503 ± 0.002
	THIS-ODE	0.526 ± 0.004
Stream	POST	0.696 ± 0.019
	ADF-CP	0.648 ± 0.008
	BASS-Tucker	0.976 ± 0.024
	SFTL-CP	$\mathbf{0.424 \pm 0.014}$
	SFTL-Tucker	0.430 ± 0.010

	MAE	
Static	PTucker	0.369 ± 0.009
	Tucker-ALS	0.615 ± 0.006
	CP-ALS	0.642 ± 0.012
	CT-CP	0.46 ± 0.004
	CT-GP	0.414 ± 0.001
	BCTT	0.355 ± 0.005
	NONFAT	0.341 ± 0.001
	THIS-ODE	0.363 ± 0.004
Stream	POST	0.478 ± 0.014
	ADF-CP	0.449 ± 0.006
	BASS	0.772 ± 0.031
	SFTL-CP	$\mathbf{0.242 \pm 0.006}$
	SFTL-Tucker	0.246 ± 0.001

Experiments: Learned Trajectories

- **Reasonable uncertainty**
- **Indicate tensor rank?**



(a) $u^1_{1,1}(t)$ (b) $u^1_{1,2}(t)$ (c) $u^1_{1,3}(t)$

(d) $u^2_{2,1}(t)$ (e) $u^2_{2,2}(t)$ (f) $u^2_{2,3}(t)$

Thanks.

Github Repo