THE UNIVERSITY OF UTAH

Bayesian Streaming Sparse Tucker Decomposition

Shikai Fang, Robert M. Kirby, Shandian Zhe
Presenter: Shikai Fang
School of computing, The University of Utah

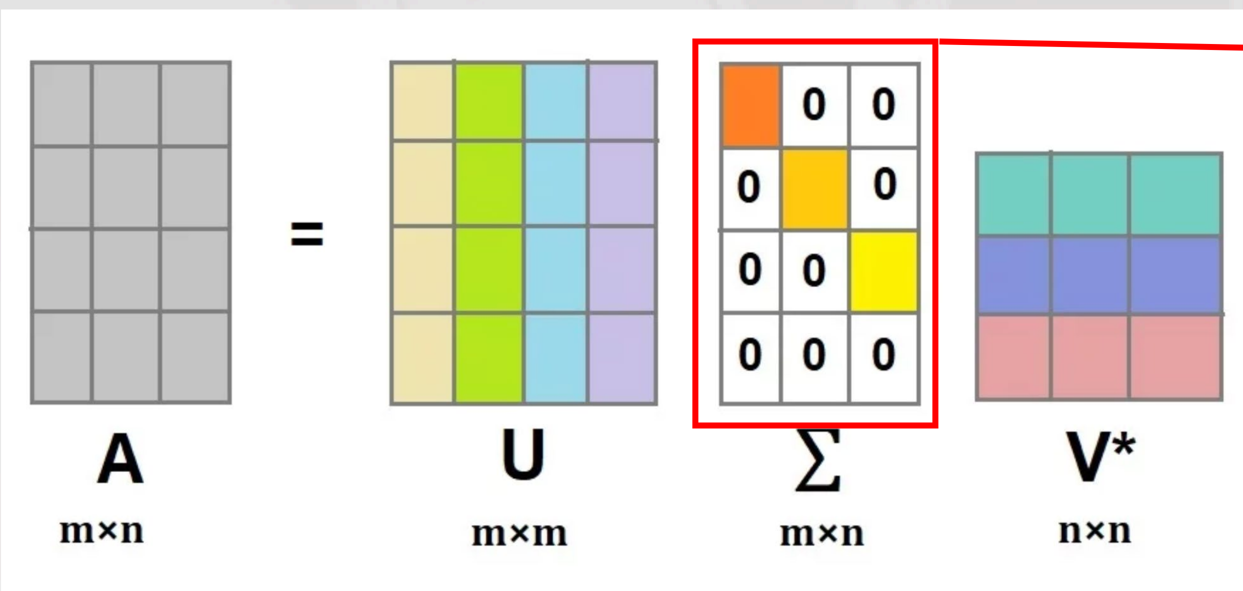
For UAI 2021

Outline

1. Background
2. Motivation
3. Bayesian Sparse Tucker Model
4. One-shot & Streaming Inference
5. Experiments on Real-world Data

1. Background

- Tucker tensor decomposition: Generalization of the matrix SVD (also called *Higher Order Singular Value Decomposition* (HOSVD))



SVD core:
•2-D diagonal matrix
•only model interactions of embeddings on same dim

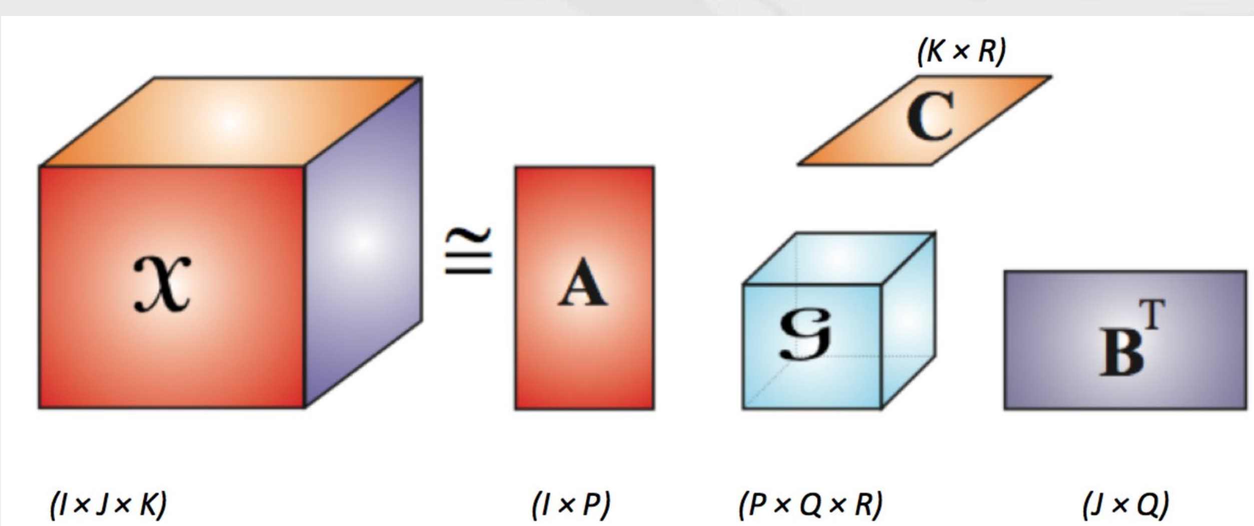
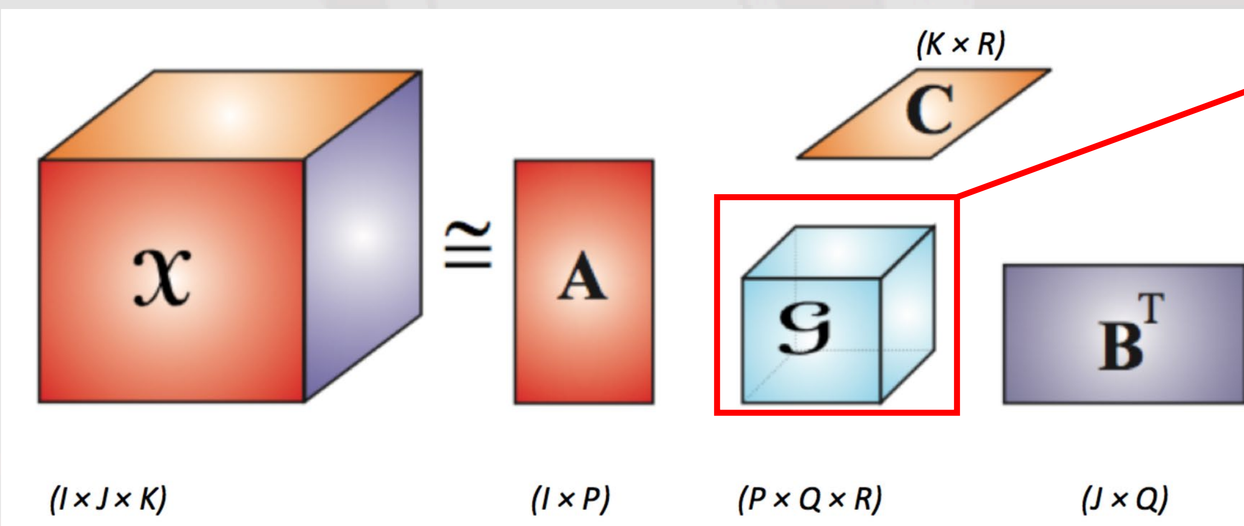


Image from https://iksinc.online/2018/05/02/understanding-tensors-and-tensor-decompositions-part-3/.

1. Background

- Tucker tensor decomposition: Generalization of the matrix SVD (also called *Higher Order Singular Value Decomposition* (HOSVD))



Tucker core (3-mode example):
•3-d dense tensor
•model all possible interactions of embeddings at every dim

- Element-wise form for a K-mode tensor Y:

Core tensor element: interaction weight

$$y_i \approx \boxed{\sum_{r_1=1}^{R_1} \cdots \sum_{r_K=1}^{R_K}} \underbrace{\left[\boxed{w_{(r_1,\ldots,r_K)}} \cdot \prod_{k=1}^{K} \boxed{u^k_{i_k,r_k}} \right]}_{\text{One interaction}} .$$

Embeddings

Traverse each dim per mode

Image from https://iksinc.online/2018/05/02/understanding-tensors-and-tensor-decompositions-part-3/.

1. Background

- Probabilistic/Bayesian version of tucker decomposition:
  everything is random variable (distribution)
- For uncertainty measure and robustness

- Element-wise form for a K-mode tensor Y:

$$y_{\mathbf{i}} \approx \sum_{r_1=1}^{R_1} \cdots \sum_{r_K=1}^{R_K} \left[\boxed{w_{(r_1,\ldots,r_K)}} \cdot \prod_{k=1}^{K} \boxed{u^k_{i_k,r_k}} \right].$$

All random variables:
place priors and do inference!

2. Motivation:

- Classical Tucker tensor decomposition is featured as
  I. Flexible: model all possible interactions
  II. Interpretable: core tensor indicates interaction strengths

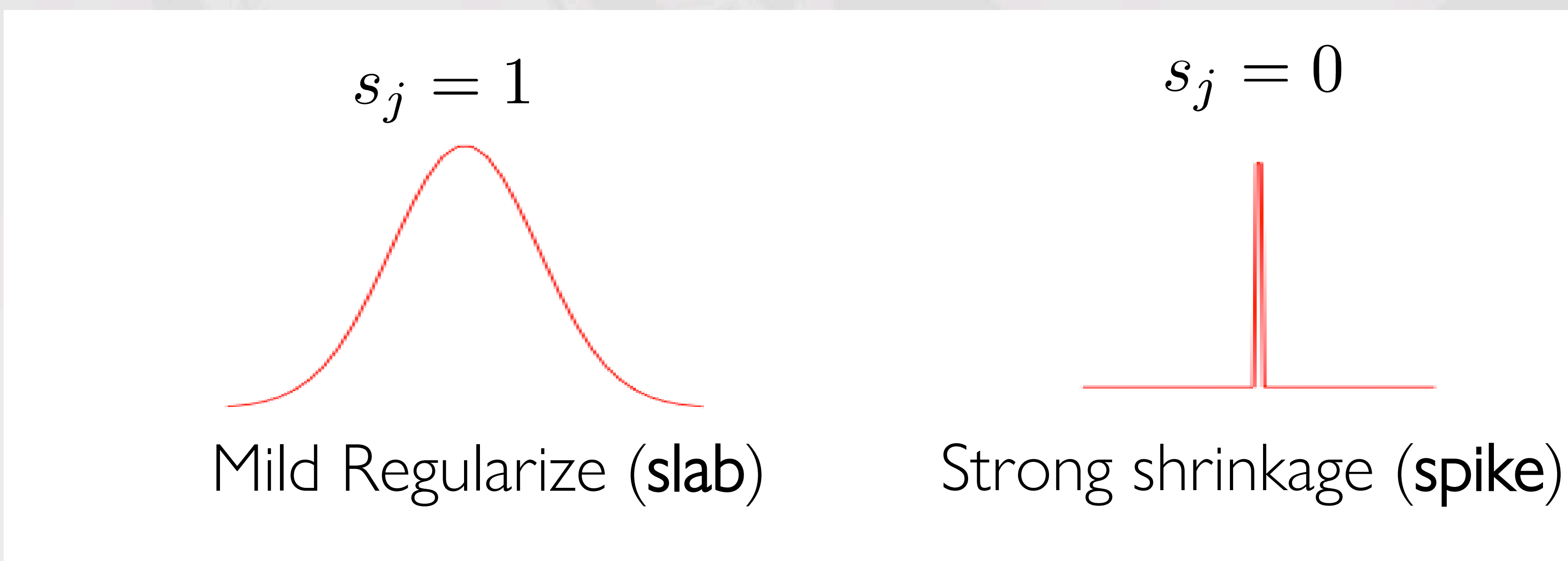
  but suffers from:
  I. Estimating core-tensor is memory & computationally intensive
  II. Overparameterizing and Overfitting risk, esp. for sparse data
  III. The two problems are more severe for streaming data!

- Goal
  I. Alleviate over-parameterization: automatic selecting meaningful interactions
  II. Efficient streaming posterior inference

3. Bayesian Sparse Tucker Model

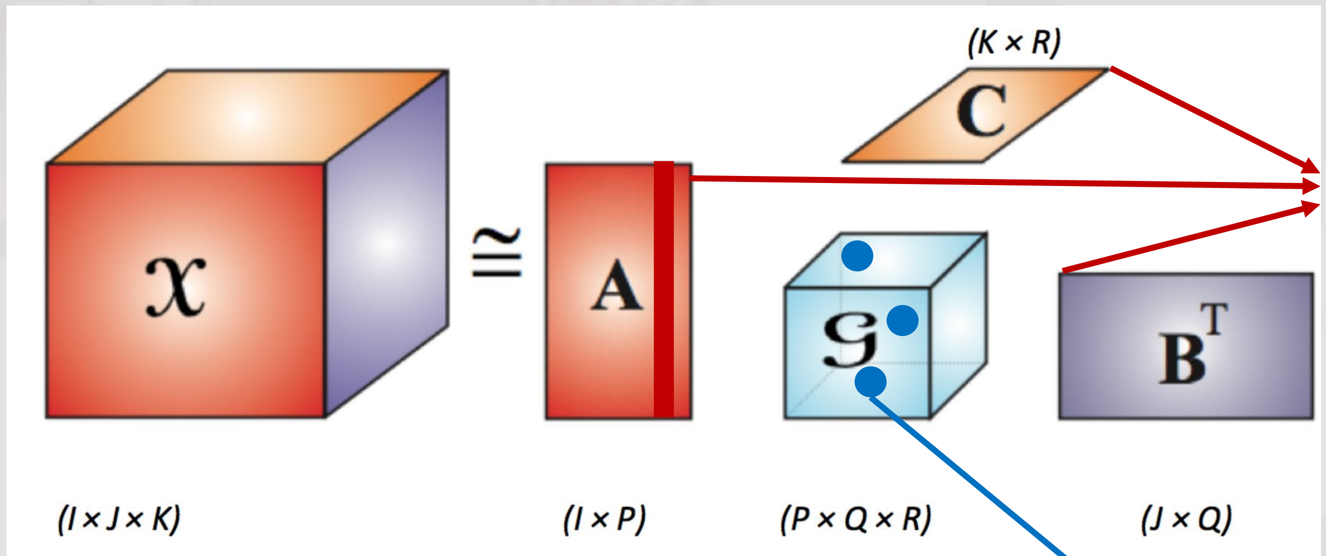
- Spike and Slab priors :

Introduce binary selection indicators on each core tensor element ! $s_1, s_2, \ldots, s_d$



$$s_{\mathbf{j}} \mathcal{N}\left(w_{\mathbf{j}} \mid 0, \sigma_0^2\right) \qquad \left(1 - s_{\mathbf{j}}\right) \delta\left(w_{\mathbf{j}}\right)$$

3. Bayesian Sparse Tucker Model



Standard Gaussian prior

S&S Prior over each core tensor element

$$p(\mathcal{S},\mathcal{W},\mathcal{U},\mathcal{Y},\tau)=\prod_{k}\prod_{j}\mathcal{N}\left(\mathbf{u}_j^k \mid \mathbf{0},\mathbf{I}\right)\mathrm{Gam}\left(\tau \mid a_0,b_0\right)$$

$$\cdot\prod_{\mathbf{j}}\boxed{\mathrm{Bern}\left(s_{\mathbf{j}} \mid \rho_0\right)}\boxed{\left(s_{\mathbf{j}}\mathcal{N}\left(w_{\mathbf{j}} \mid 0,\sigma_0^2\right)+\left(1-s_{\mathbf{j}}\right)\delta\left(w_{\mathbf{j}}\right)\right)}$$

Binary indicators

Selective shrinkage priors on core based on indicator

$$\cdot\prod_{\mathbf{i}\in\mathcal{F}}\mathcal{N}\left(y_{\mathbf{i}} \mid \mathcal{W}\times_1\left(\mathbf{u}_{i_1}^1\right)^\top\times_2\ldots\times_K\left(\mathbf{u}_{i_K}^K\right)^\top,\tau^{-1}\right)$$

3. Bayesian Sparse Tucker Model

- <u>Exact posterior distribution</u>: Intractable!
- Approximation with distributions in <u>exponential family</u>:

$$q_{\text{cur}}(\mathcal{W},\mathcal{U},\tau) \propto p(\mathcal{S})\underbrace{\boxed{\xi(\mathcal{W},\mathcal{S})}}_{\text{Approximation of SS priors}} \cdot \prod_{k=1}^{n}\prod_{j=1}^{a_k}\mathcal{N}\left(\mathbf{u}_j^k \mid \boldsymbol{\mu}_j^k, \mathbf{V}_j^k\right)$$

$$\cdot\, \mathcal{N}(\text{vec}(\mathcal{W}) \mid \mathbf{m}, \boldsymbol{\Sigma})\,\text{Gam}(\tau \mid a, b)$$

Approximation of data likelihood

where:

$$\xi(\mathcal{W},\mathcal{S}) = \prod_{\mathrm{j}} \xi_{\mathbf{j}}\left(w_{\mathrm{j}}, s_{\mathrm{j}}\right)$$

$$= \prod_{\mathrm{j}} \text{Bern}\left(s_{\mathrm{j}} \mid c\left(\rho_{\mathrm{j}}\right)\right)\mathcal{N}\left(w_{\mathrm{j}} \mid m_{\mathrm{j}}, \eta_{\mathrm{j}}\right) \propto p(\mathcal{W} \mid \mathcal{S}),$$

3. Streaming & One-shot inference

- Streaming: data come, model update, data drops
- Incremental Bayesian rule:

Exact / Approx posterior on old data

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

3. Streaming & One-shot inference

- Streaming: data come, model update, data drops
- Incremental Bayesian rule:

Exact / Approx posterior on old data

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

Data likelihood on current model

3. Streaming & One-shot inference

- Streaming: data come, model update, data drops
- Incremental Bayesian rule:

Exact / Approx posterior on old data

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

Exact / Approx posterior on all data

Data likelihood on current model

3. Streaming & One-shot inference

- Streaming: data come, model update, data drops
- Incremental Bayesian rule:

Exact / Approx posterior on old data

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

Exact / Approx posterior on all data

Data likelihood on current model

- Classical ADF: integrating data points one by one via **moment matching** --- inefficient (esp for core tensor update); many approximations
- Our goal: assimilating a **batch of streaming** data points at a time: -- more efficient and improve the quality

3. Streaming & One-shot inference

- For tractable moment in ADF, we made 3 tech-contributions

I. Conditional Expectation Propagation(CEP)

$$\mathbb{E}_{\tilde{p}}[\phi(\mathcal{W})] = \mathbb{E}_{\tilde{p}\left(\Theta_{\backslash w}\right)}\left[\boxed{\mathbb{E}_{\tilde{p}\left(\mathcal{W}|\Theta_{|w}\right)}\phi(\mathcal{W}) \mid \Theta_{\backslash \mathcal{W}}}\right]$$

Tractable Conditional Moment!

II. Delta method: Expectation on first-order Taylor approximation

$$\mathbb{E}_{q_{\text{cur}}\left(\Theta_{\backslash \mathcal{W}}\right)}\left[\mathbf{h}\left(\Theta_{\backslash \mathcal{W}}\right)\right] \approx \boxed{\mathbf{h}\left(\mathbb{E}_{q_{\text{cur}}}\left[\Theta_{\backslash \mathcal{W}}\right]\right)}$$

h : first-order approx. at the mean

III. Repeated update of S&S prior approx. to ensure sparsity inducing effect

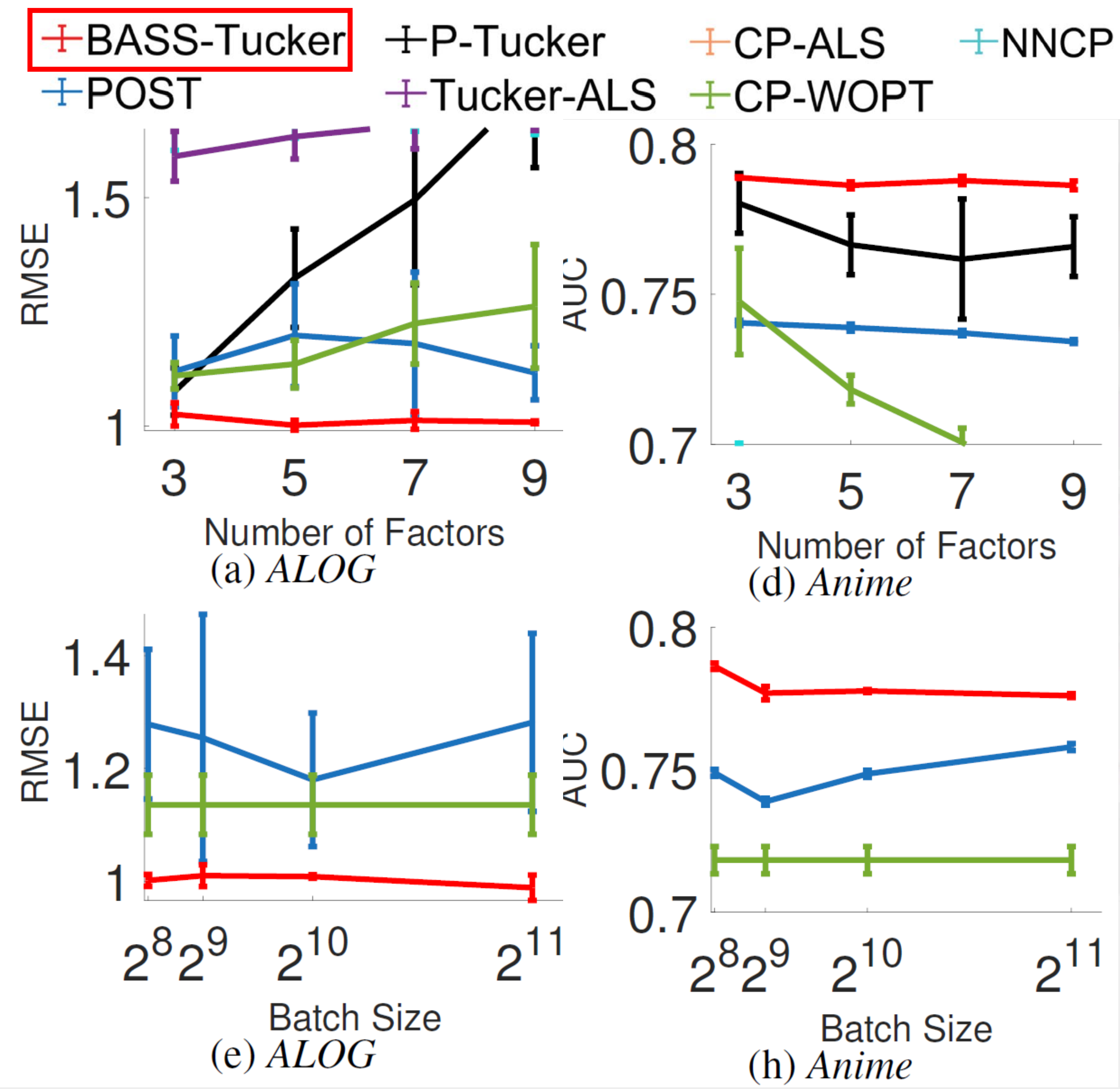
$$q^{*}\left(w_{\mathbf{j}}, s_{\mathbf{j}}\right) = \operatorname{Bern}\left(s_{\mathbf{j}} \mid c\left(\rho_{\mathbf{j}}^{*}\right)\right) \mathcal{N}\left(w_{\mathbf{j}} \mid \mu_{\mathbf{j}}^{*}, v_{\mathbf{j}}^{*}\right)$$

THE UNIVERSITY OF UTAH

5. Experiments on real-world data

- Predictive performance on large real-world datasets
- With different factors / streaming batch size

Our method!



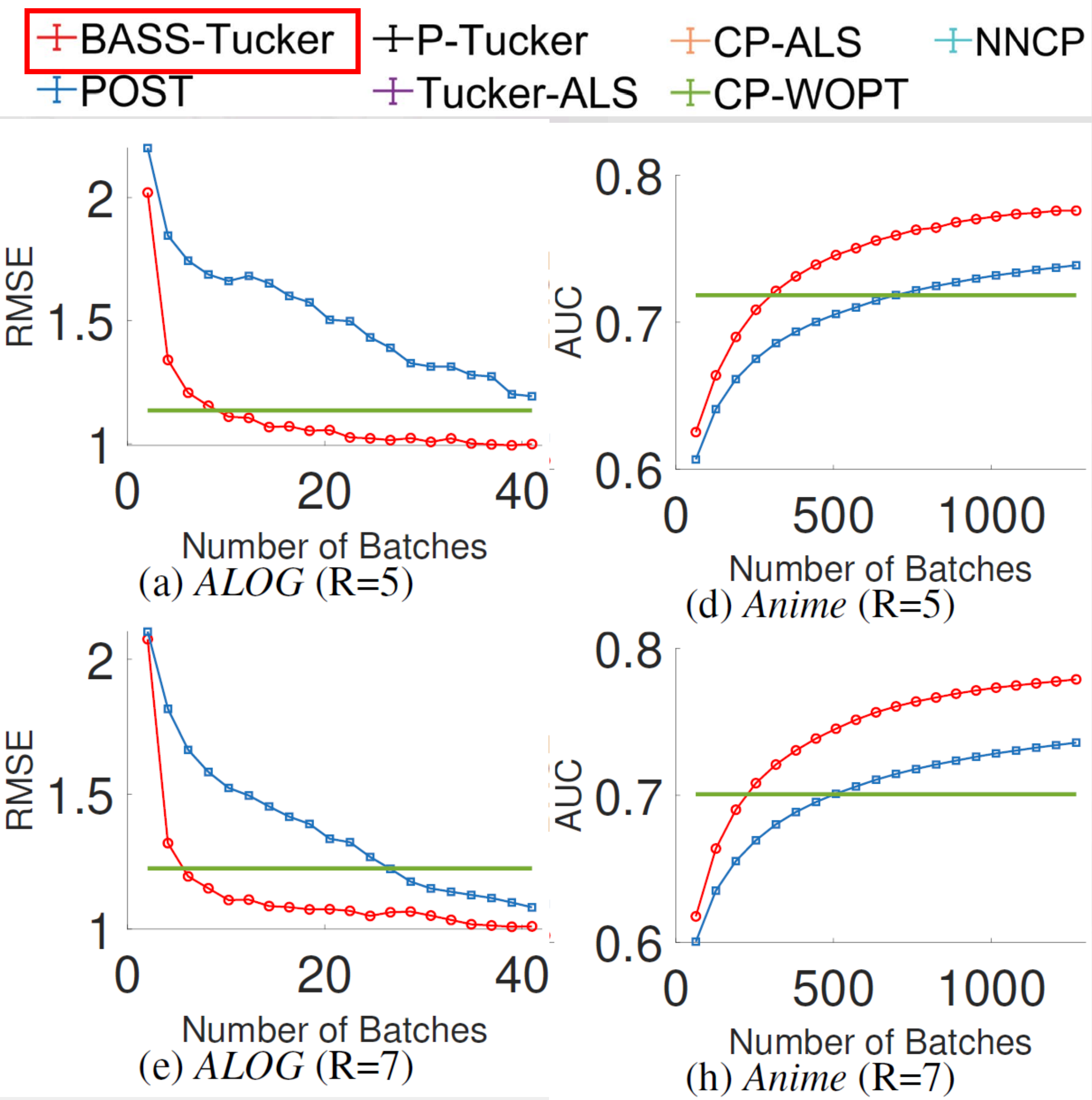
(a) *ALOG* (d) *Anime* (e) *ALOG* (h) *Anime*

THE UNIVERSITY OF UTAH

5. Experiments on real-world data

- Running prediction large real-world datasets
- With different number of factors

Our method!



(a) *ALOG* (R=5)

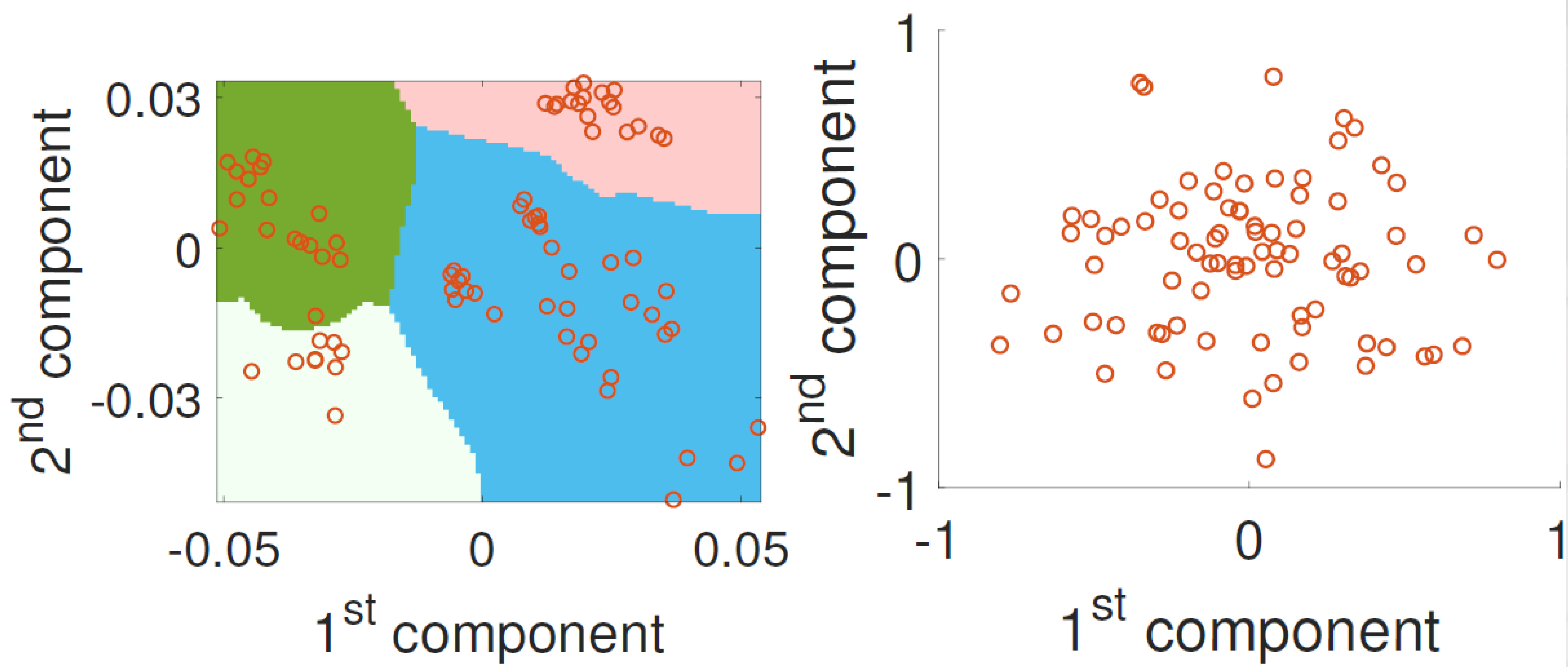
(d) *Anime* (R=5)

(e) *ALOG* (R=7)

(h) *Anime* (R=7)

5. Experiments on Real-world Data

- More significant Core tensor structure



(a) BASS-Tucker (b) P-Tucker

THE UNIVERSITY OF UTAH

Thanks for attention
Q&A Time

Authors' email: shikai.fang@utah.edu, {kirby, zhe}@cs.utah.edu

Focus: Probabilistic model, Bayesian machine learning and its application