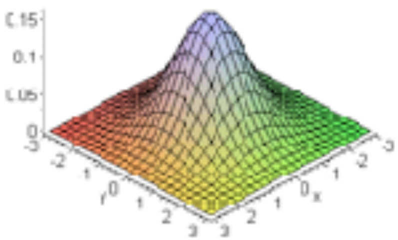
Scalable Nonparametric Tensor Decomposition

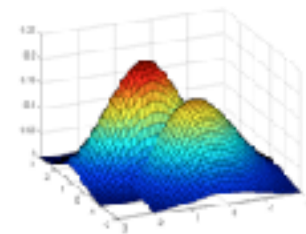
Shandian Zhe(zhe@cs.utah.edu)
School of Computing
University of University
07/30/2019

Agenda

- Bayesian learning
- Bayesian nonparametric tensor analysis
  - Distributed infinite Tucker decomposition
  - Distributed flexible nonlinear tensor decomposition
  - Nonparametric event-tensor decomposition

Bayesian Learning





Prior distribution

$$p(\boldsymbol{\theta})$$

Data likelihood

$$p(\mathbf{D}|\boldsymbol{\theta})$$

Posterior distribution

$$p(\boldsymbol{\theta}|\mathbf{D})$$

Bayes' Rule

$$p(\boldsymbol{\theta}|\mathbf{D}) = \frac{p(\boldsymbol{\theta}, \mathbf{D})}{p(\mathbf{D})} = \frac{p(\boldsymbol{\theta})p(\mathbf{D}|\boldsymbol{\theta})}{\int p(\boldsymbol{\theta})p(\mathbf{D}|\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}$$

Advantages

- Unified, principled mathematical framework

Priors + Data = Posteriors

$\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$ $\qquad$ $\mathbf{D}|\boldsymbol{\theta} \sim p(\mathbf{D}|\boldsymbol{\theta})$ $\qquad$ $p(\boldsymbol{\theta}|\mathbf{D})$

- Seamless, flexible uncertainty reasoning

Asthma: 60%

Heart disease: 30%

Healthy: 10%

Raining: 70%

Sunny: 30%

Computational Challenge

$$p(\boldsymbol{\theta}|\mathbf{D}) = \frac{p(\boldsymbol{\theta})p(\mathbf{D}|\boldsymbol{\theta})}{\boxed{\int p(\boldsymbol{\theta})p(\mathbf{D}|\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}}$$

Infeasible to compute!

High dimensional integration

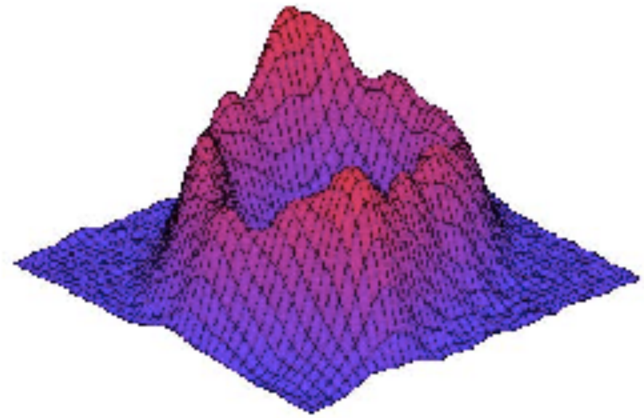
Complicated forms

Approximate Inference

- MCMC Sampling
- Variational Approximation
- Belief Propagation

My Research

- Bayesian Nonparametrics: ***Complex patterns***



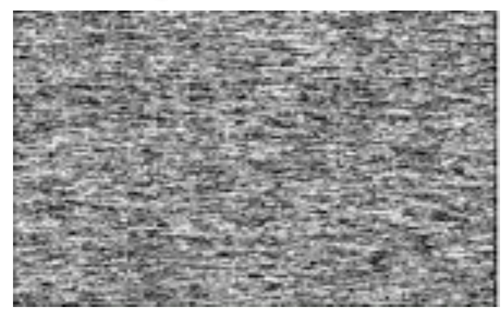
- Bayesian Sparse Learning: ***Succinct patterns***

Big Data Analytics

Rich Knowledge

Big Noise

Bayesian Nonparametrics



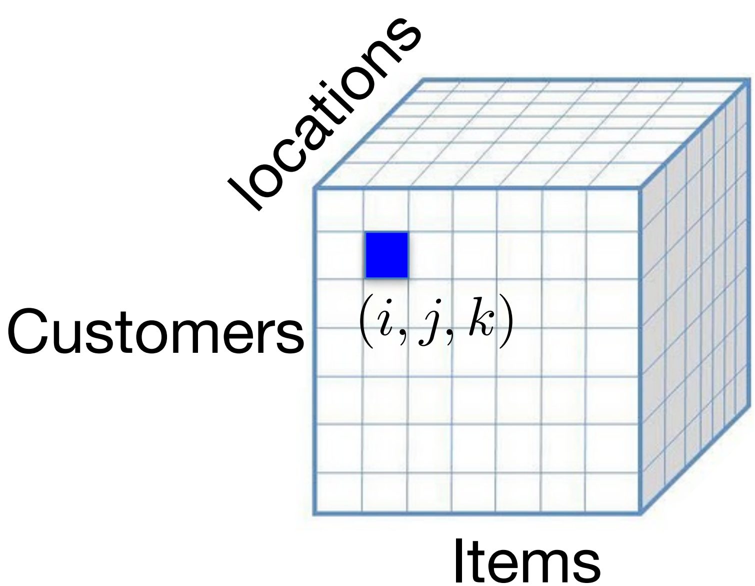
Bayesian Sparse Learning

Bayesian Nonparametric Tensor Analysis

Tensors


amazon
ebay
WHICH ONE CAN MAXIMIZE YOUR SALES
?

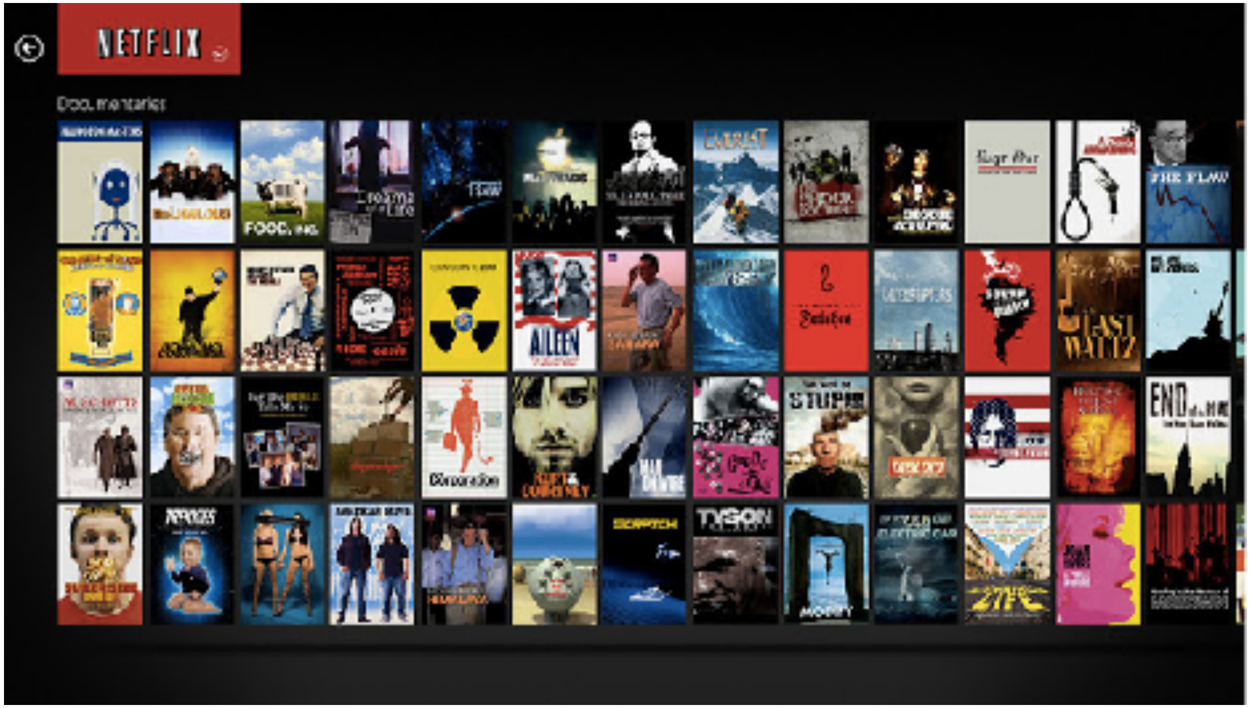

facebook
Facebook helps you connect and share with the people in your life.



$$(i,\ j,\ k)$$

Whether customer i purchased item j at store-location k ?

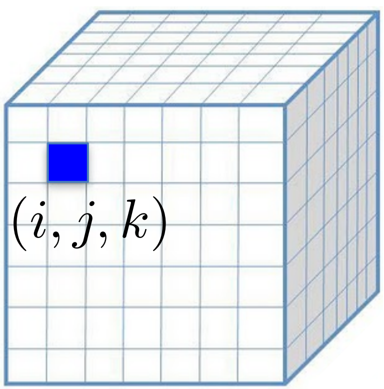
(User, Item, Online-store)



(User, Movie, TV Series, Month)

Tensor: Interaction records between multiple entities

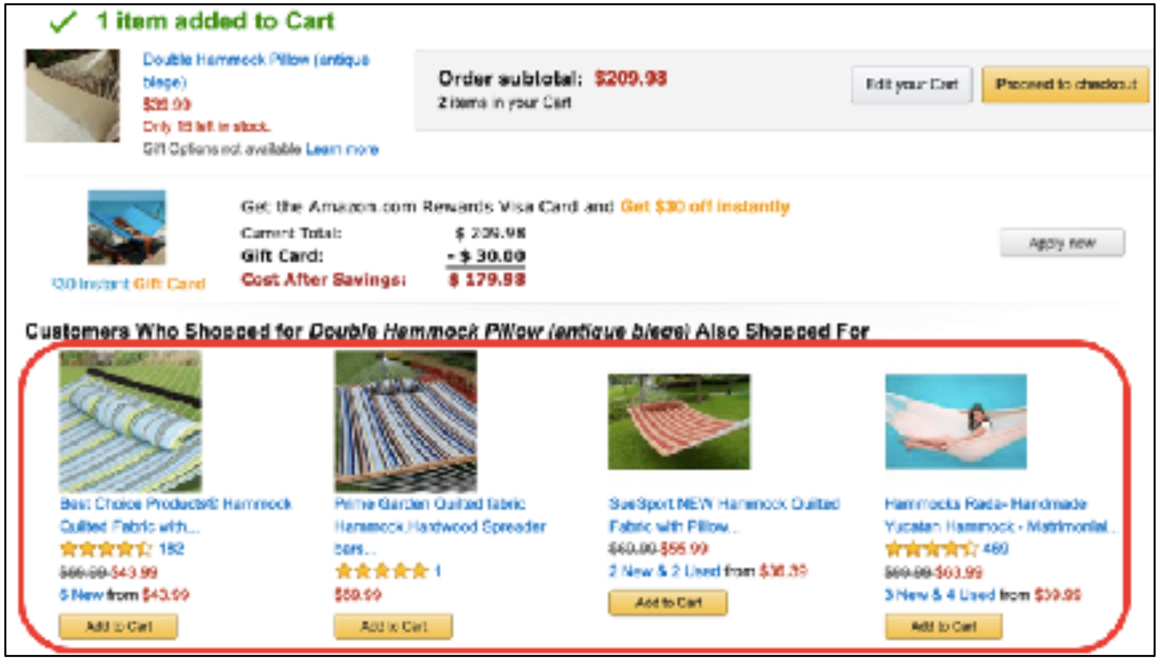
Key Problem: How to infer the **underlying multiway relationships** between the entities?



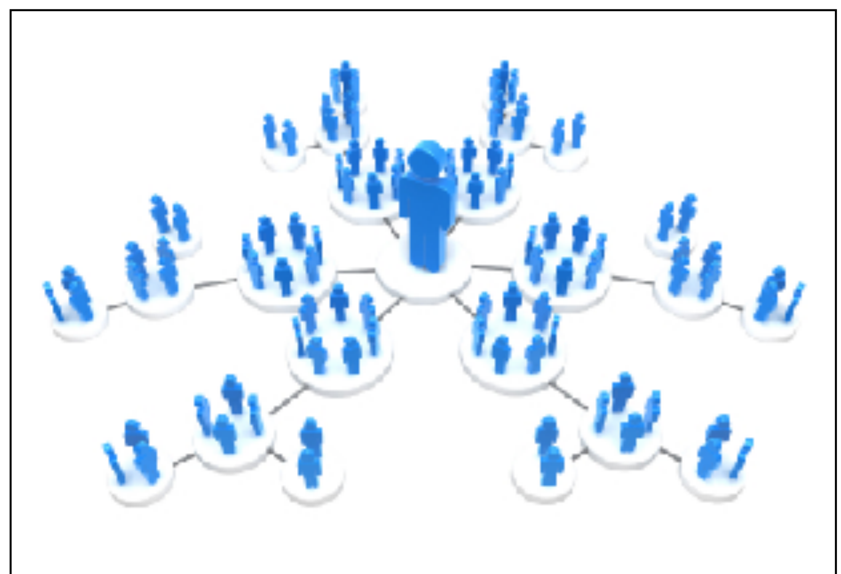
Relationship (Customer, Item, Online-store)

Relationship (User, Movie, TV Series, Month)

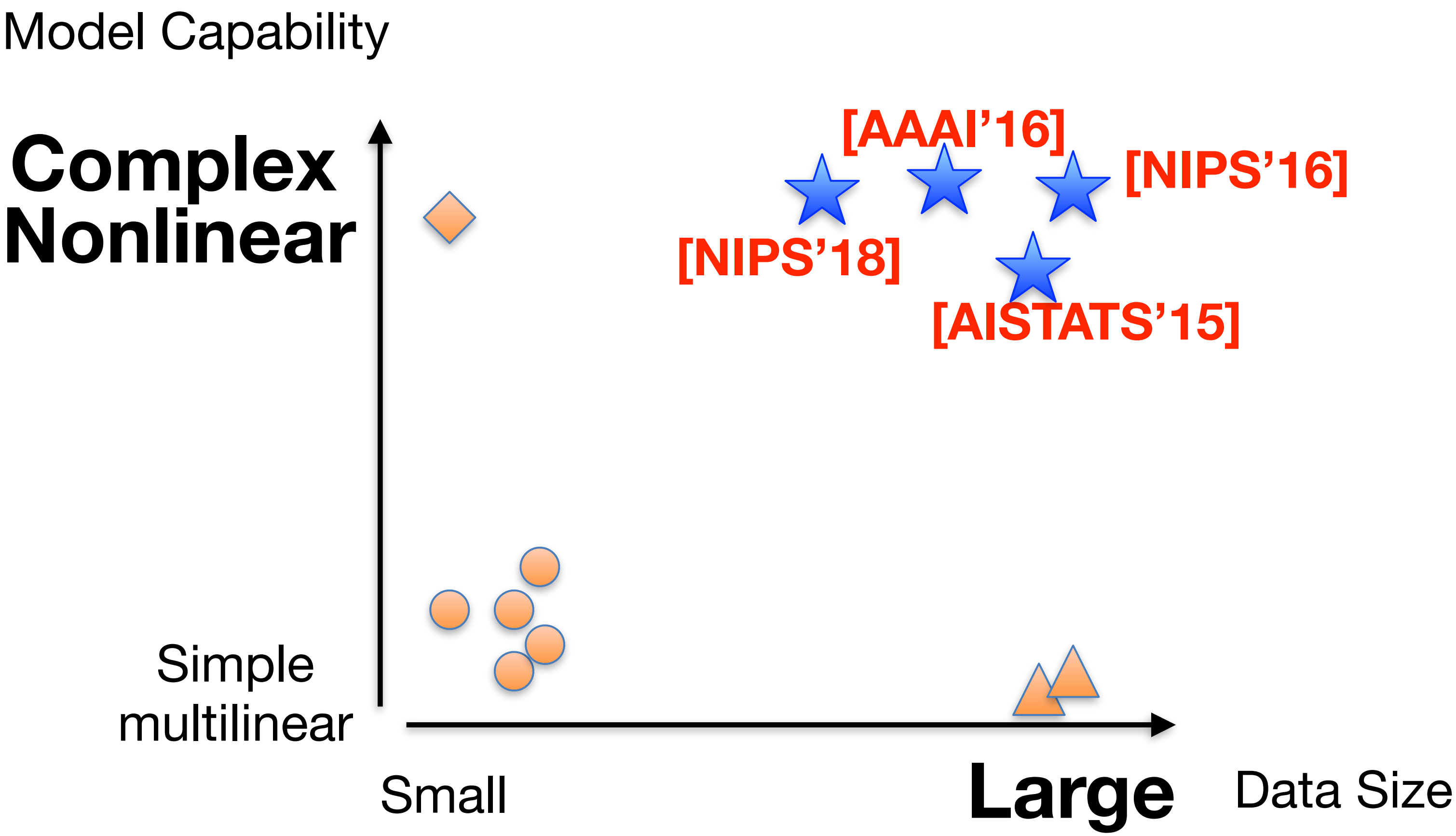
......


1 item added to Cart
Order subtotal: $209.98
2 items in your Cart
Edit your Cart
Proceed to checkout
Get the Amazon.com Rewards Visa Card and Get $30 off instantly
Current Total: $ 209.98
Gift Card: - $ 30.00
Cost After Savings: $ 179.98
Customers Who Shopped for Double Hammock Pillow (antique beige) Also Shopped For
Add to Cart

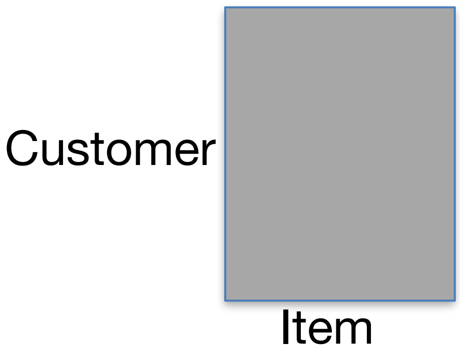

Other Movies You Might Enjoy
Add
Not Interested



Overview of Tensor Analysis



Tensor Analysis — Factorization



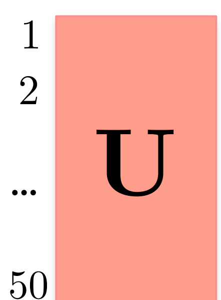
Matrix factorization review

1. Factor representation

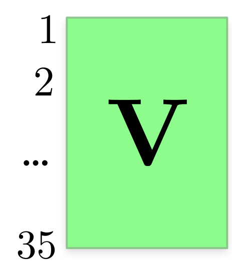
Customer 1	?	?	?
Interests:	***I***	***II***	***III***

Item 2	?	?	?
Attributes:	***a***	***b***	***c***

Customer factor matrix

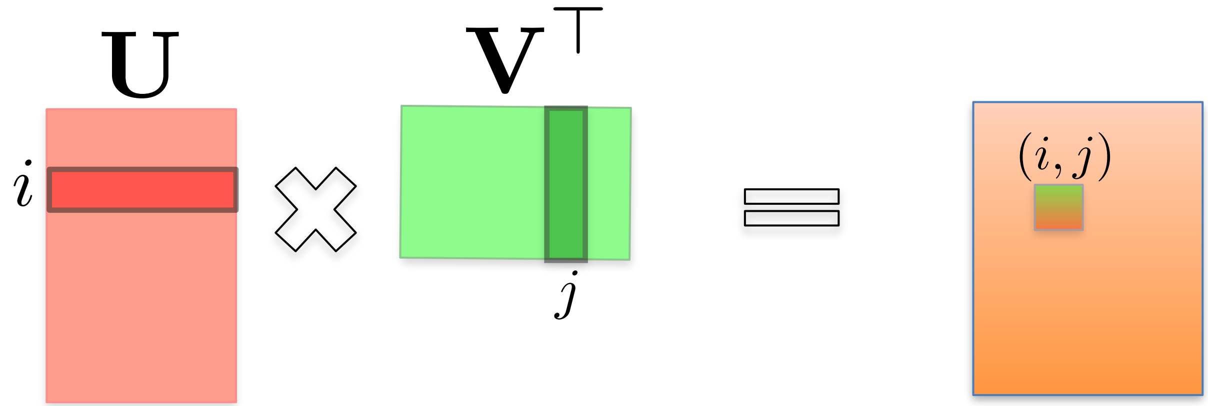


Item factor matrix



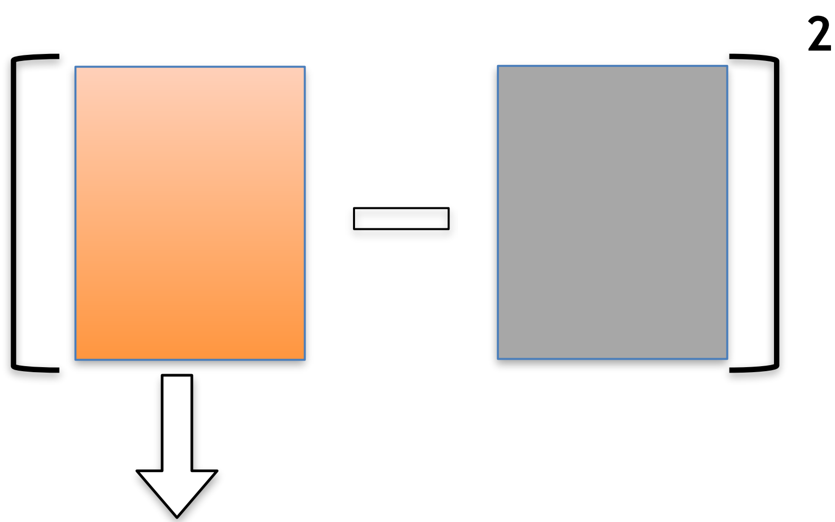
Matrix Factorization

2.Construction model



3. Latent factor estimation

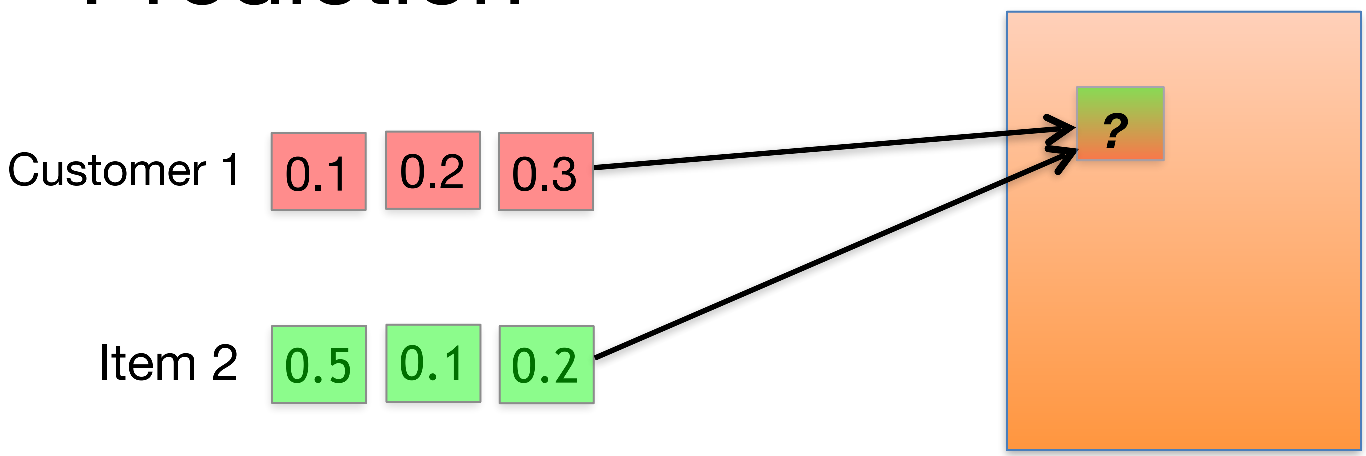
Minimize



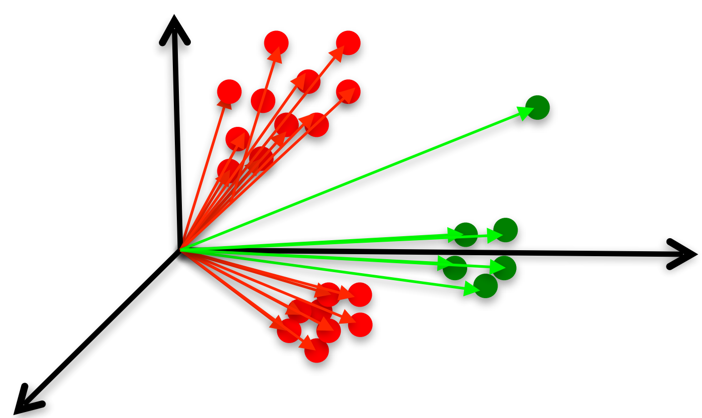
$\mathbf{U}^*, \mathbf{V}^*$

Usage of Factors

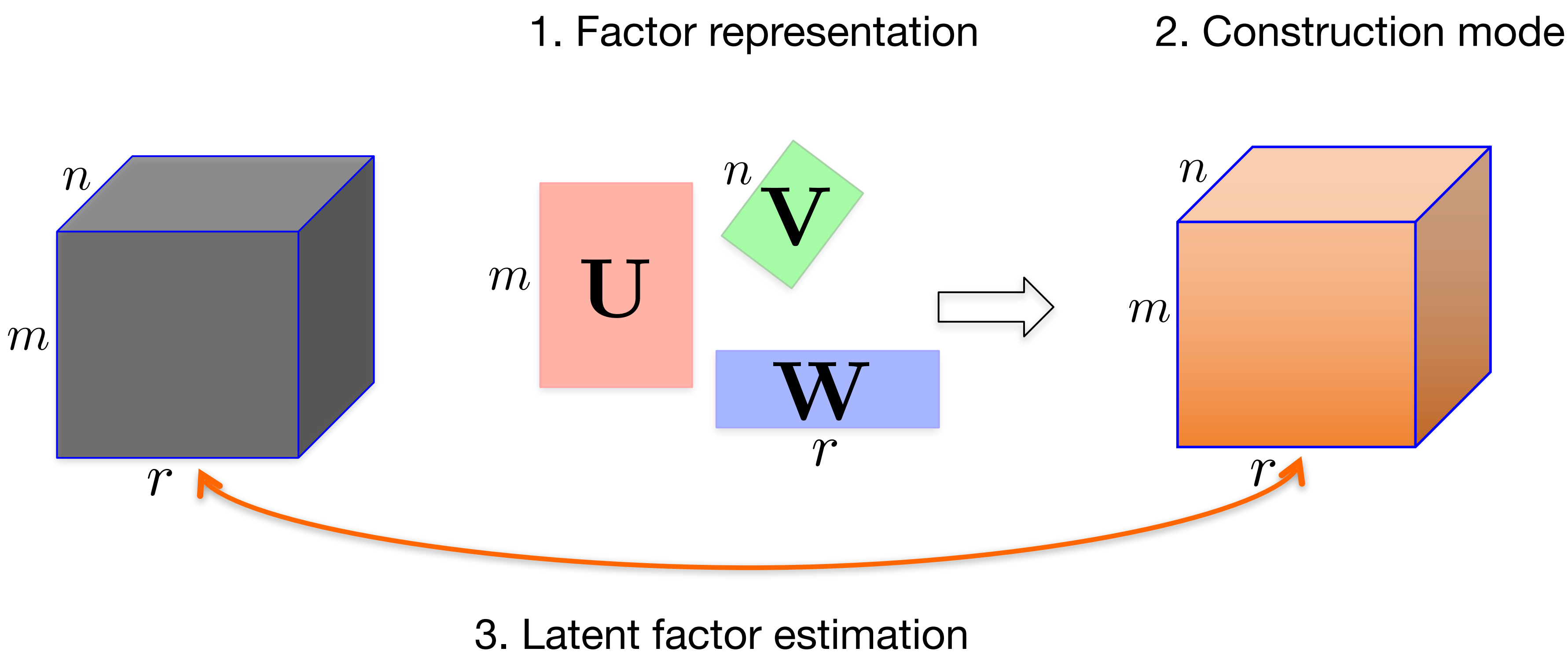
- Prediction



- Patterns



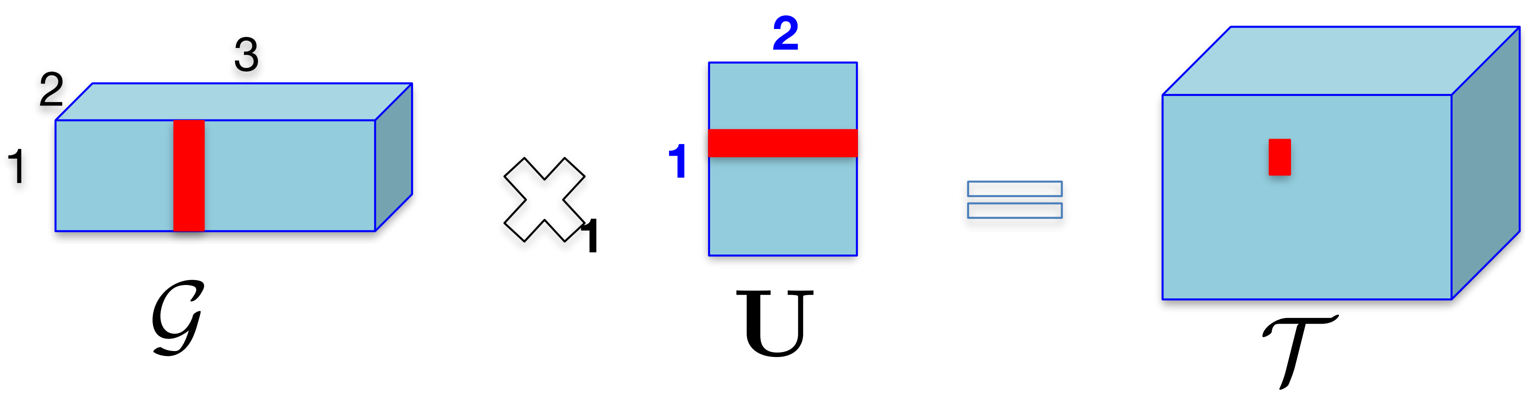
Tensor Factorization



Multilinear Tensor Factorization

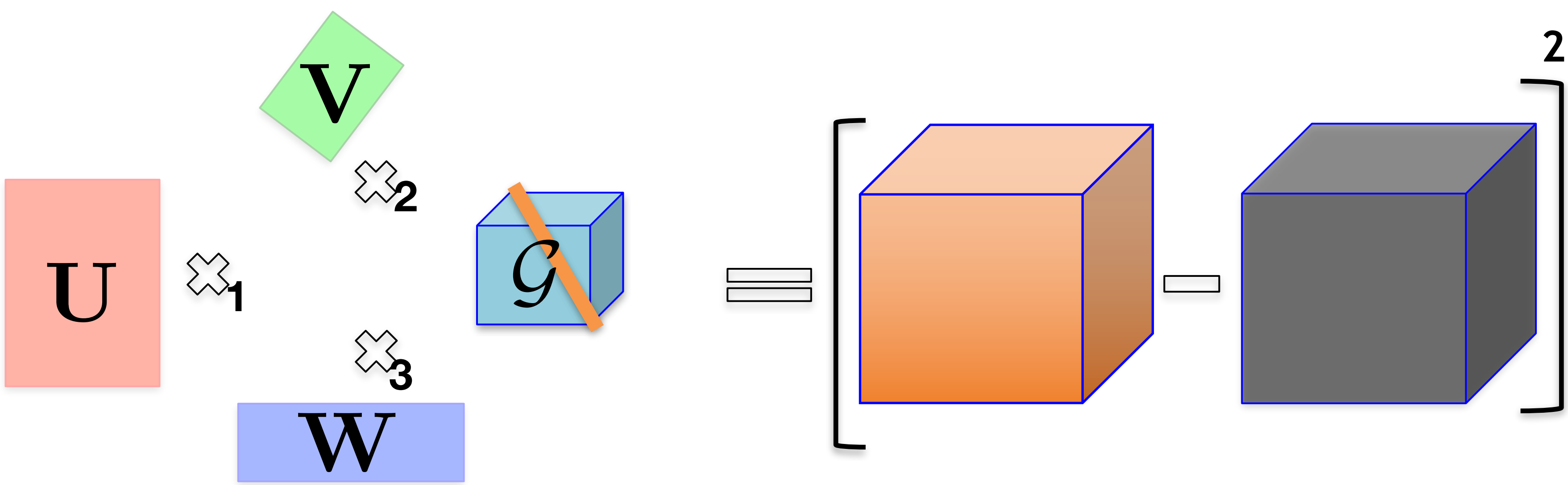
Tensor-matrix multiplication

$$\mathcal{G} \times_1 \mathbf{U} = \mathcal{T}$$

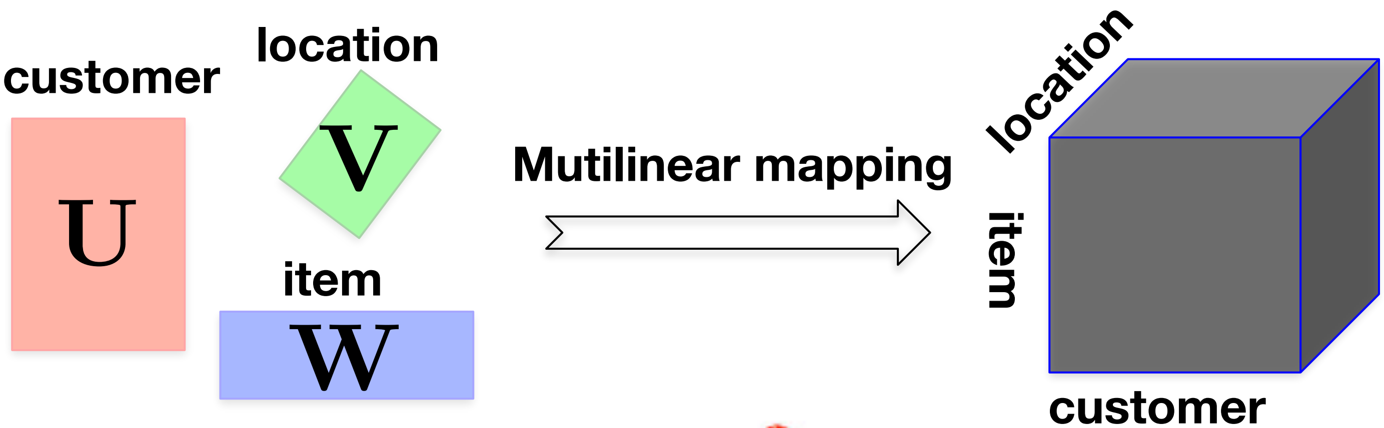


Multilinear Tensor Factorization

Tucker decomposition $\Longrightarrow$ CP decomposition



Multilinear Factorization: Limitation



Complex interactions/Relationships: ***Customer, location, items***

Infinite Tucker decomposition

Gaussian Process

Complex interactions/Relationships

Gaussian Process Models: Nonlinear Mapping Estimator

Observations

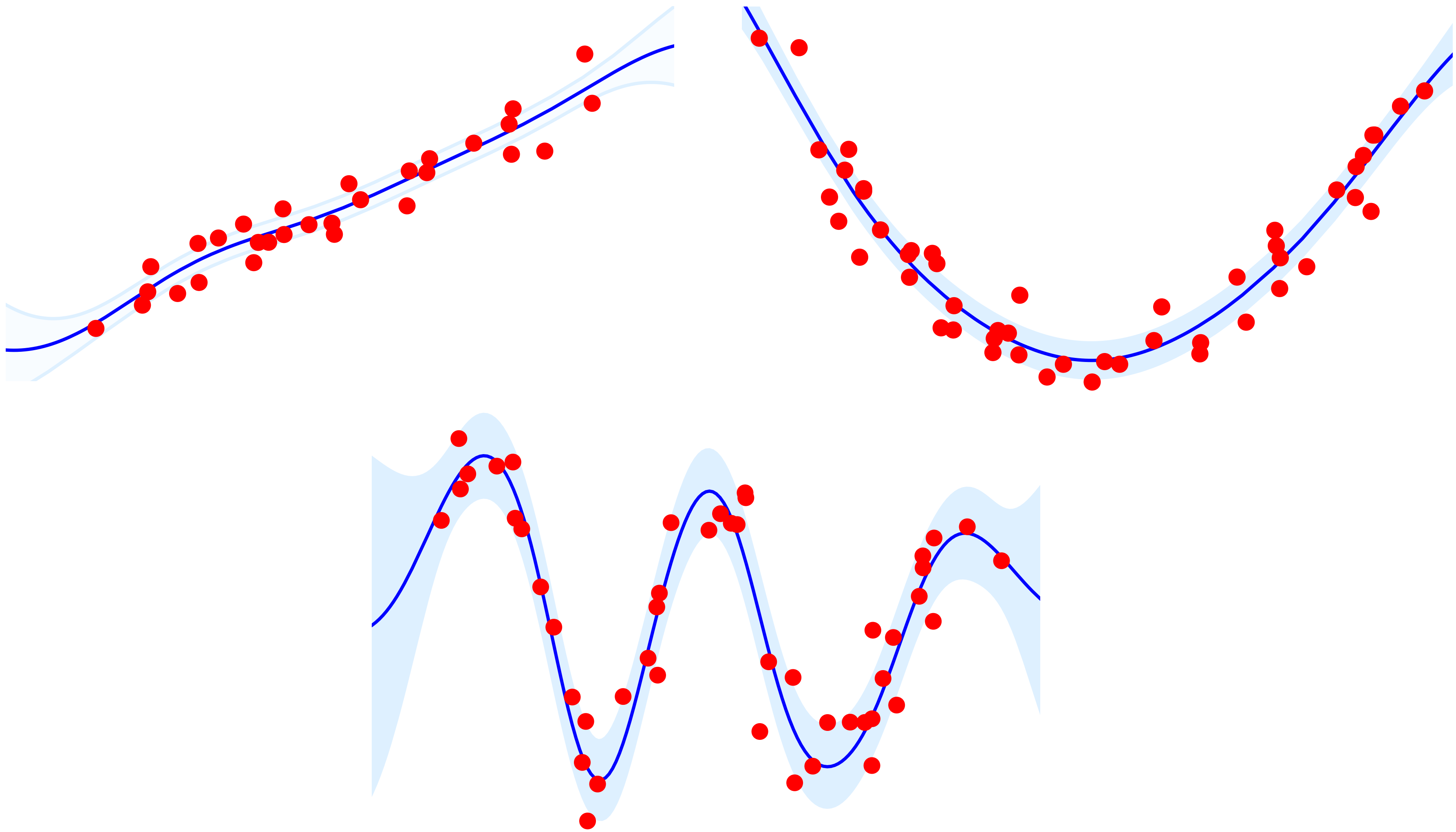
$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \cdots \\ \mathbf{x}_n^\top \end{bmatrix} \qquad Y = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \cdots \\ \mathbf{y}_n \end{bmatrix}$$

$$p(Y|\mathbf{X}) = \mathcal{N}\big(Y|\mathbf{0}, k(\mathbf{X}, \mathbf{X})\big)$$

$$k(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \boxed{k(\mathbf{x}_1, \mathbf{x}_2)} & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}$$

e.g., $k(\mathbf{x}_i, \mathbf{x}_j) = \sigma e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}$

Gaussian Process Learning Examples



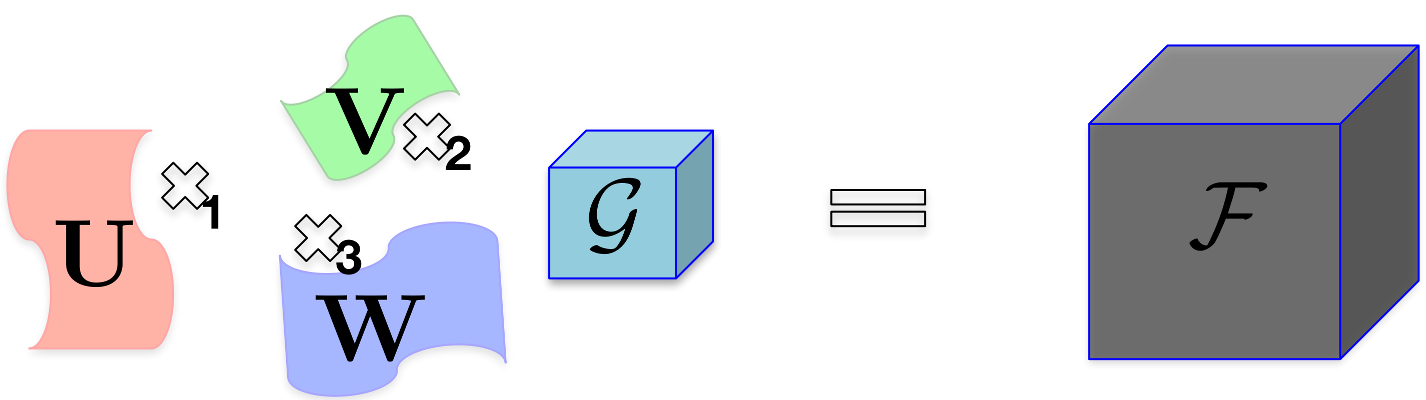
From Tucker to Infinite Tucker

Probabilistic sampling procedure

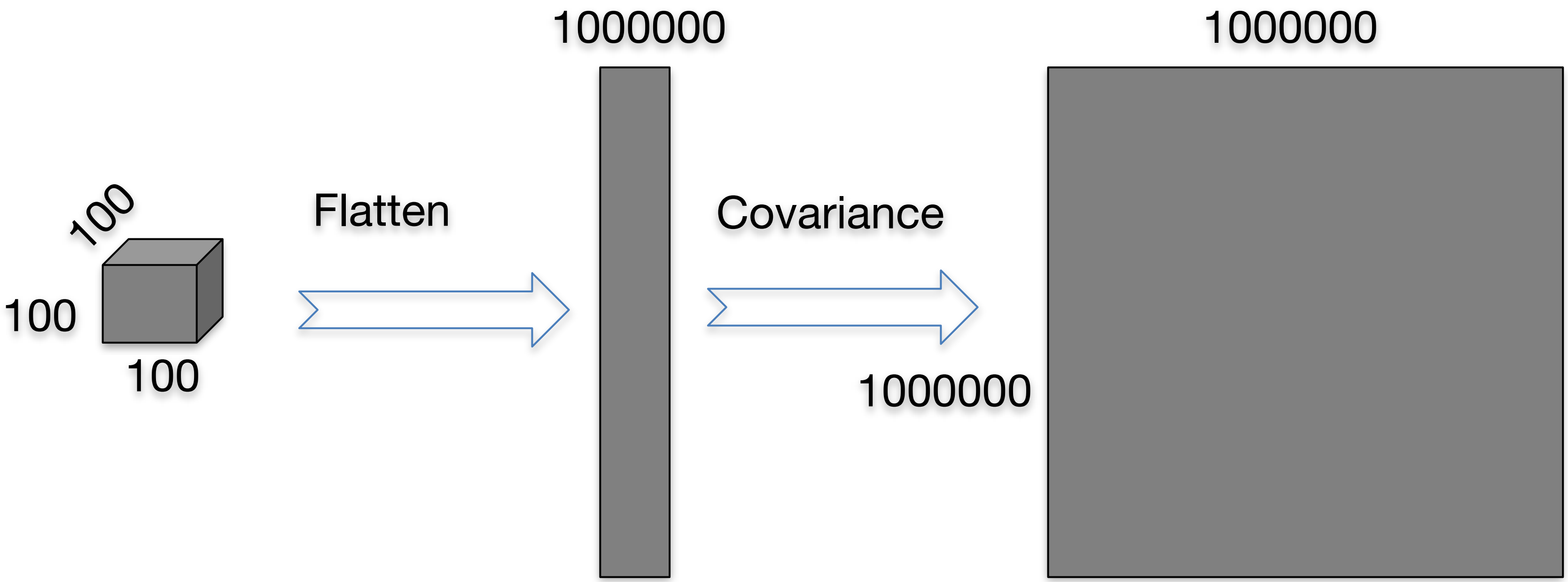
Applying kernel function

Tensor-Variate Gaussian Process

$$\mathcal{N}\left(\text{vec}(\mathcal{F})|\mathbf{0}, k(\mathbf{U},\mathbf{U}) \otimes k(\mathbf{V},\mathbf{V}) \otimes k(\mathbf{W},\mathbf{W})\right)$$



Infinite Tucker: Scalability Problem

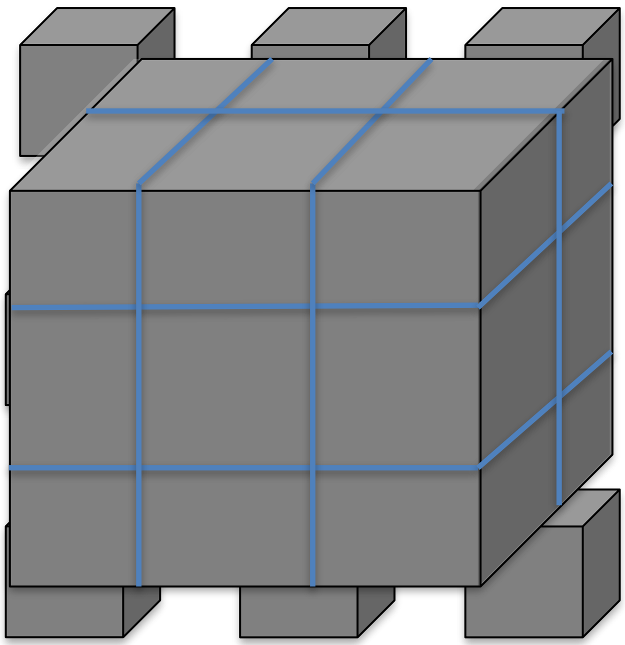


$$\mathcal{N}\left(\mathrm{vec}(\mathcal{F})|\mathbf{0}, \underline{k(\mathbf{U},\mathbf{U})} \otimes \underline{k(\mathbf{V},\mathbf{V})} \otimes \underline{k(\mathbf{W},\mathbf{W})}\right)$$

Global GP ➔ Huge Covariance Matrix

Distributed Infinite Tucker Decomposition

[*Zhe et. al.*, AAAI'16]

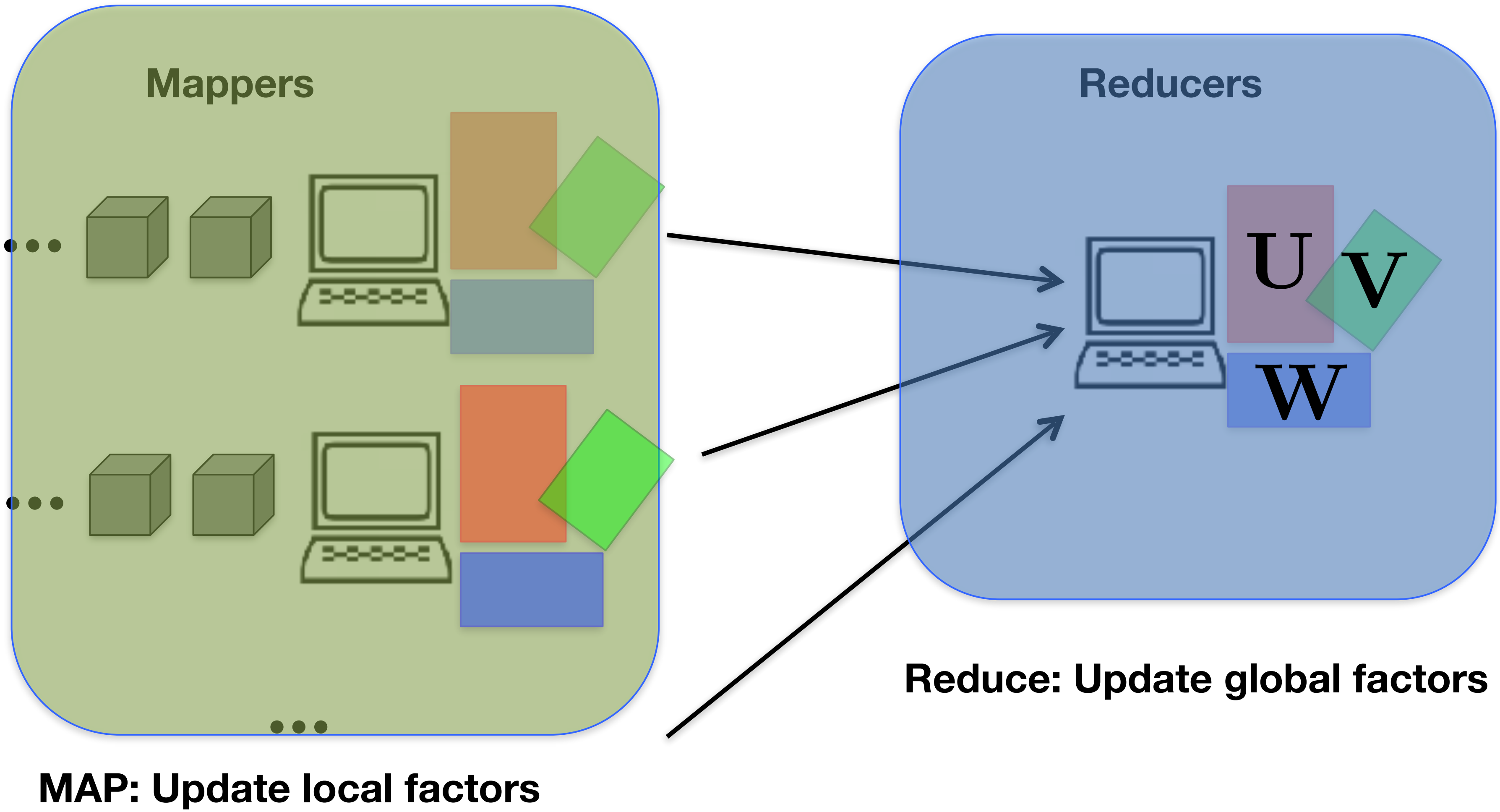


Divide and Conquer

Think globally: Assume ***globally shared*** latent factors

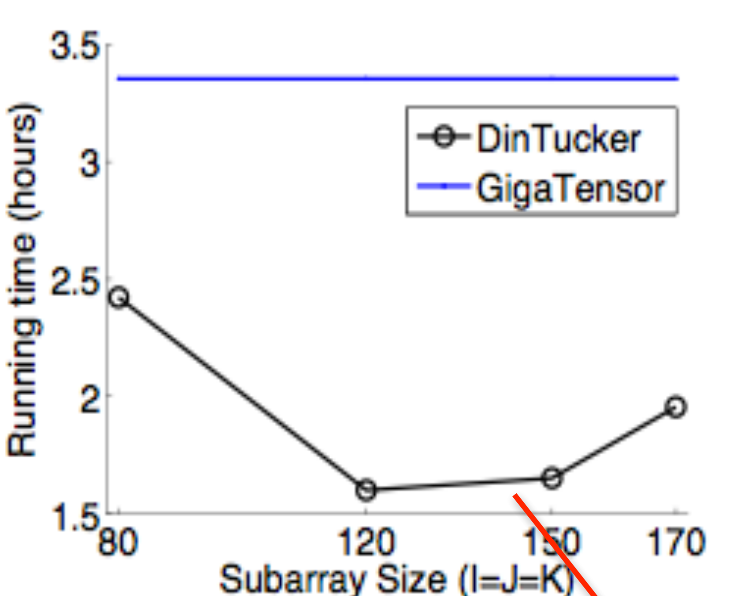
Act locally: Each subtensor is sampled from a ***local*** tensor-varaite GP

DinTucker: Distributed Infinite Tucker on Map-Reduce

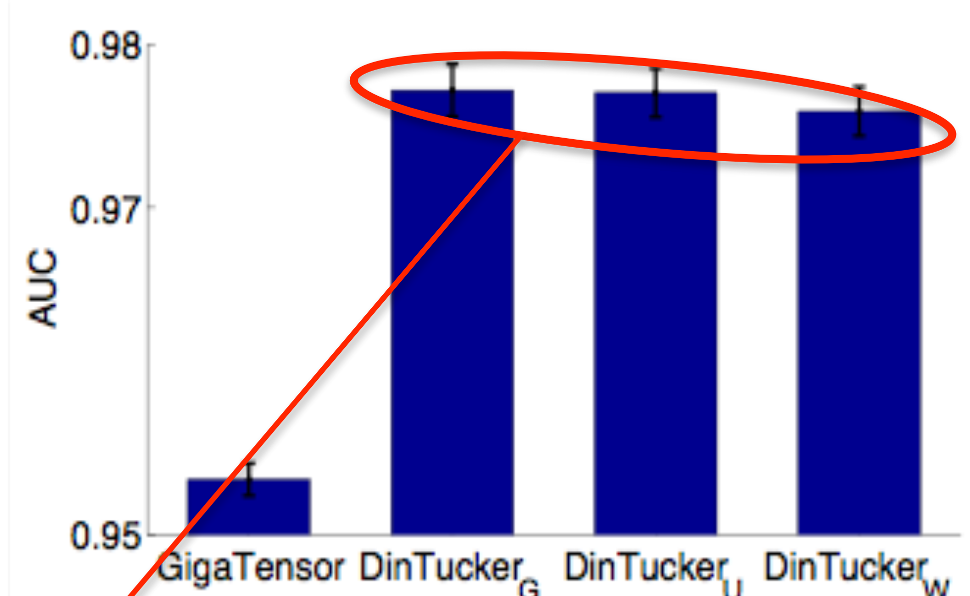


Hadoop Implementation on Large Data

Data	*I*	*J*	*K*	# of entries
Access Log	2,000	179	199,800	71.5 billions



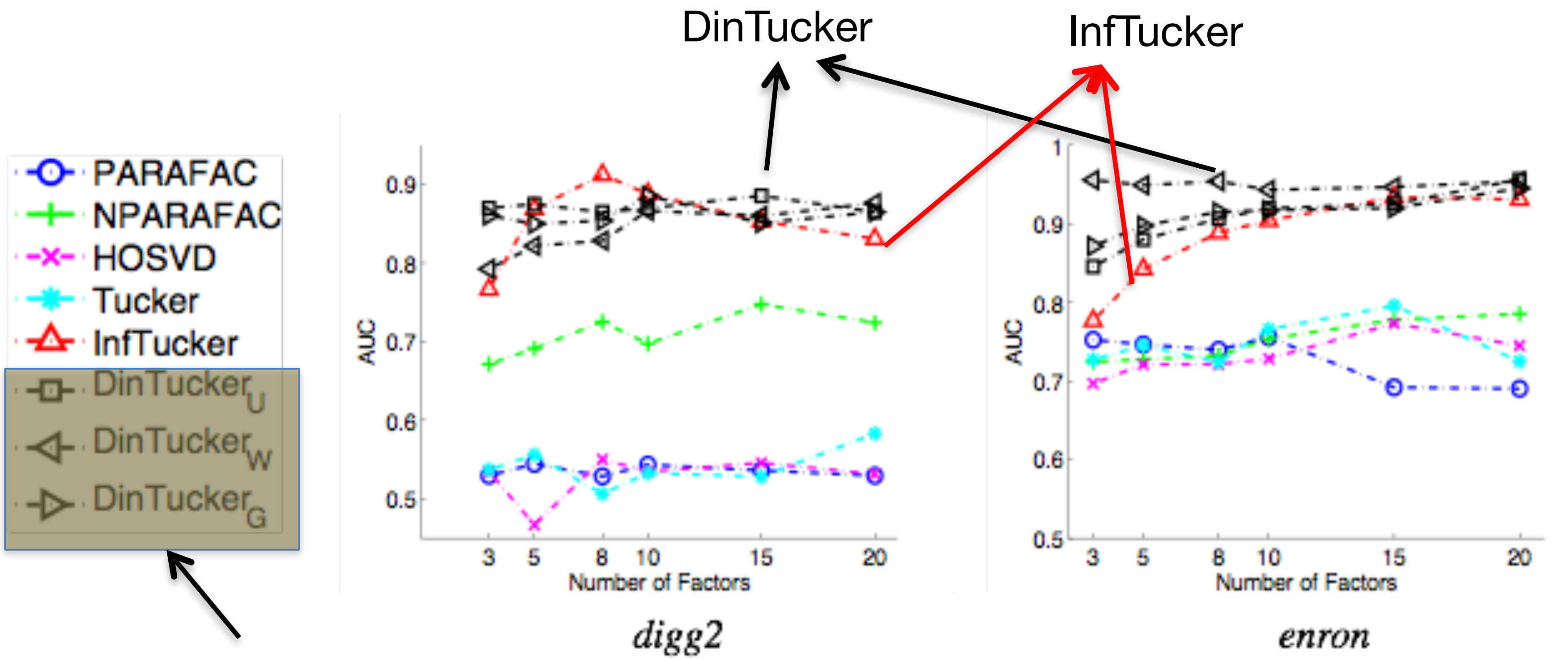
ACC: running time



ACC: prediction

DinTucker

DinTucker (Local GP) *vs.* InfTucker (Global GP)



DinTucker with a specific sampling strategy

Local GP *vs.* Global GP? A Theoretical Analysis

- **Covariance Structure**

infTucker — Global GP

$$\begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1n} \\ \Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2n} \\ \vdots & & \ddots & \vdots \\ \Sigma_{n1} & \Sigma_{n2} & \cdots & \Sigma_{nn} \end{bmatrix}$$

DinTucker — Local GP

$$\begin{bmatrix} \Sigma_{11} & & & \\ & \Sigma_{22} & & \\ & & \ddots & \\ & & & \Sigma_{nn} \end{bmatrix}$$

$\Downarrow$ Block matrix decomposition + Jensen's inequality

- **Model Evidence (- Loss Function)**

global / local

$$\underbrace{\log\left(p_I(\mathrm{vec}(\mathcal{Y})|\mathcal{U})\right)}_{\text{global}} \geq \alpha \cdot \underbrace{\log\left(p_D(\mathrm{vec}(\mathcal{Y})|\mathcal{U})\right)}_{\text{local}} + O(\alpha)$$

(under certain conditions)

- **Conclusion**

Local training $\Longleftrightarrow$ Global training ***with lower bound surrogate***

DinTucker

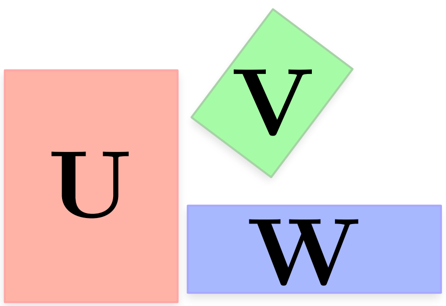
Local GP

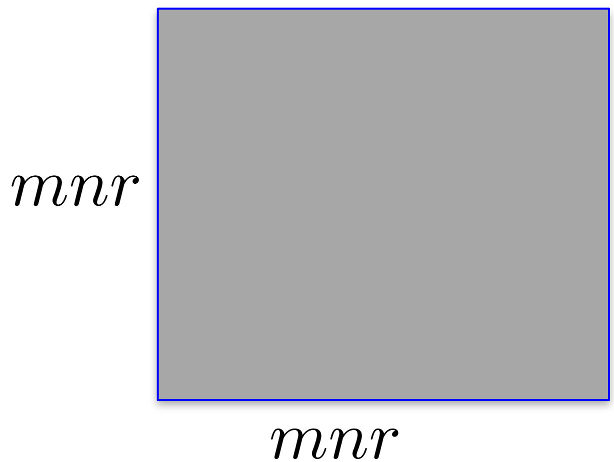
Divide & Conquer

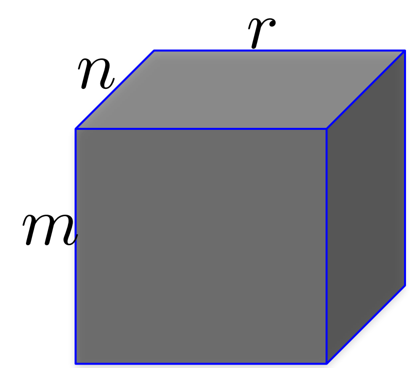
Tensor-Variate GP:
Fully observed assumption

Kronecker product in covariance

$$\underset{m \times m}{k(\mathbf{U}, \mathbf{U})} \otimes \underset{n \times n}{k(\mathbf{V}, \mathbf{V})} \otimes \underset{r \times r}{k(\mathbf{W}, \mathbf{W})}$$





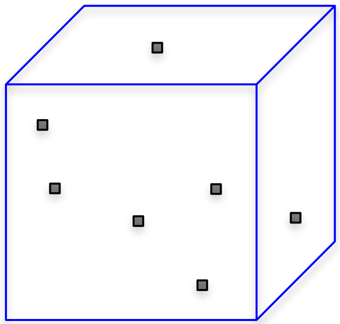


mnr
elements in total

Every element is used in training

Every element is observed

Real-World Tensor: Partially observed



e.g., (customer , item, location)
nonzeros < 1%

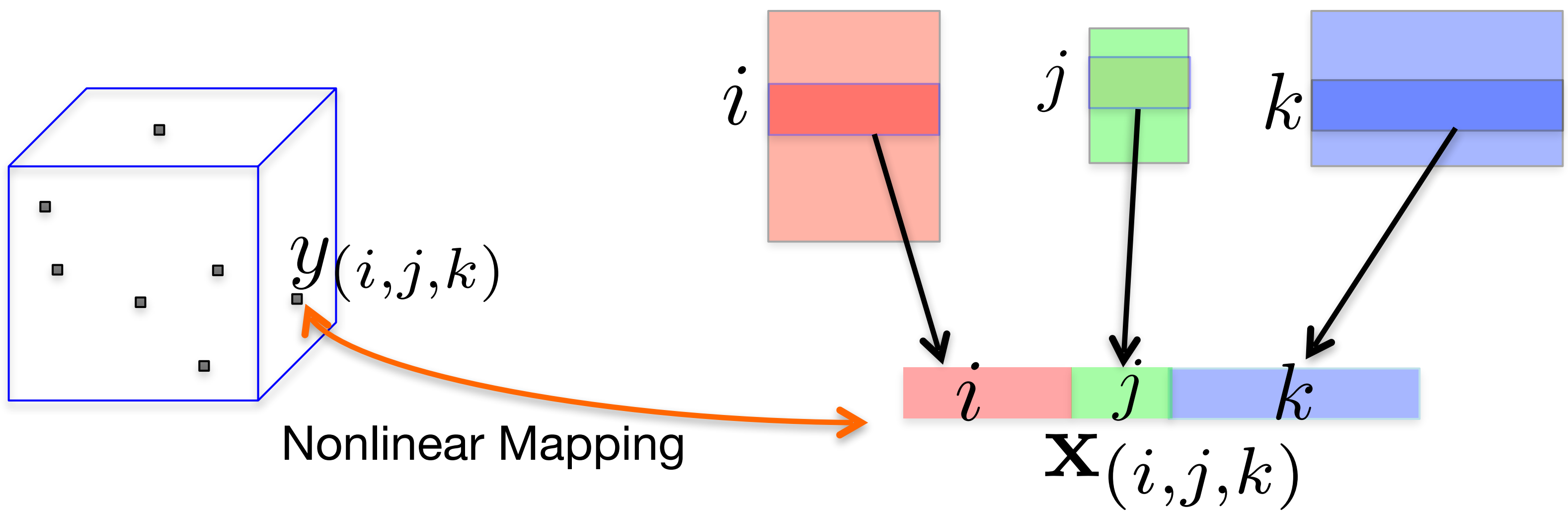
Many 0s: Missing/Unobserved

Use all —> bias results

Flexibility

Flexible Gaussian Process Factorization Model

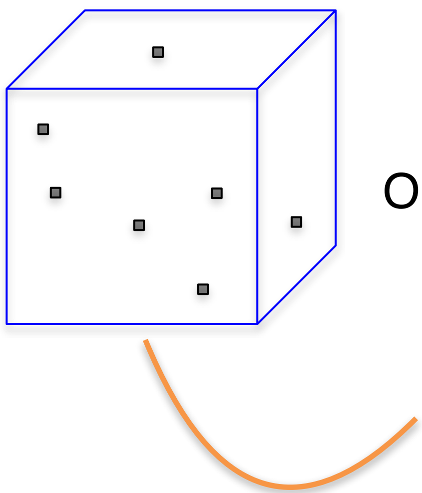
[*Zhe et. al.*, NIPS'16]



Entry input vector: **Factor concatenation**

$$y_{(i,j,k)} = f\left(\mathbf{x}_{(i,j,k)}\right)$$

Flexibility in Using Arbitrary Entries

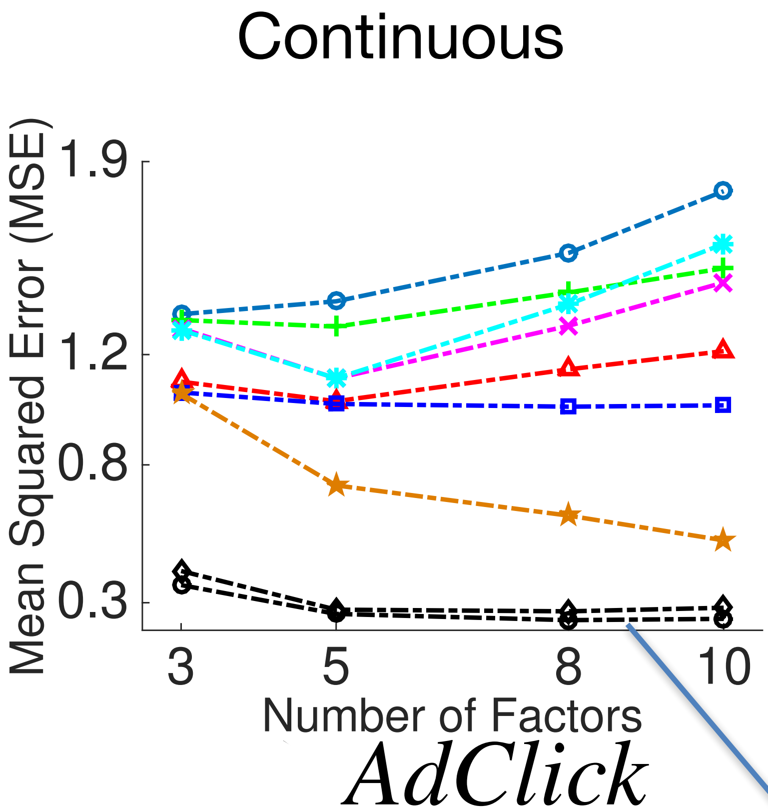


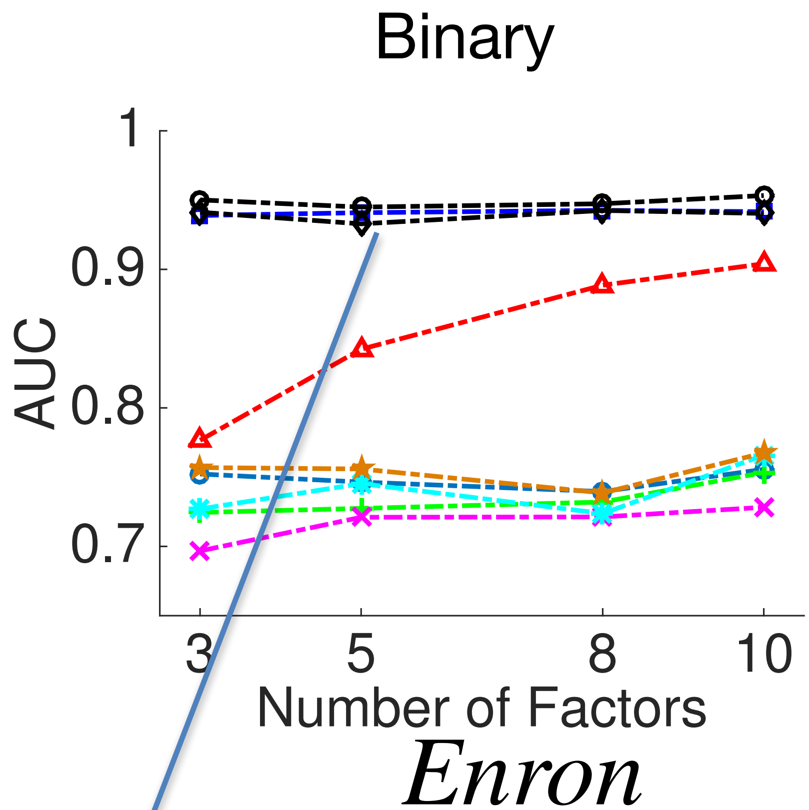
Observed entries	Entry input vectors		
$y_{(i_1,j_1,k_1)}$	i_1	j_1	k_1
$y_{(i_2,j_2,k_2)}$	i_2	j_2	k_2
...	...	...	...
$y_{(i_n,j_n,k_n)}$	i_n	j_n	k_n
$\mathbf{y}$	$\mathbf{X}$		

$k(\cdot,\cdot)$

$$\mathcal{N}\big(\mathbf{y}|\mathbf{0}, k(\mathbf{X},\mathbf{X})\big)$$ Kronecker Product Structure

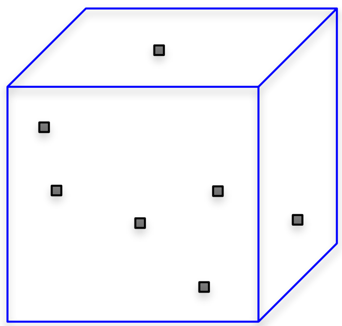
Small Data





Flexible GP factorization

Large Data: Scalability Issue



n entries

e.g., 1 million

$$\mathcal{N}\left(\mathbf{y}|\mathbf{0}, \underbrace{k(\mathbf{X},\mathbf{X})}_{n \times n}\right)$$

Model Estimation: Infeasible

Variational Estimation Procedure

(*i.e.*, Objective function)

Maximize **Model Evidence** w.r.t **latent factors**

$$\underbrace{\log\left(p(\mathbf{y})\right)}_{\mathcal{N}\left(\mathbf{y}|\mathbf{0},k(\mathbf{X},\mathbf{X})\right)} \qquad \mathcal{U} = \{\mathbf{U},\mathbf{V},\mathbf{W}\}$$

Variational Model Evidence $\leq$ **Model Evidence**

Continuous $L_1(\mathcal{U},\mathbf{B})$

Binary $L_2(\mathcal{U},\mathbf{B},\lambda)$

Variational Model Evidence

Continuous $L_1(\mathcal{U}, \mathbf{B})$

Binary $L_2(\mathcal{U}, \mathbf{B}, \lambda)$

- **Decomposed mathematical structure**
- **Parallel computation** ✔

Maximize **Variational Model Evidence**

(*i.e.*, Objective function)

Variational Form

$$
\begin{aligned}
L_1(\mathcal{U}, \mathbf{B}) = \frac{1}{2}\log|\mathbf{K}_{BB}| - \frac{1}{2}\log|\mathbf{K}_{BB} + \beta\mathbf{A}_1| - \frac{1}{2}\beta\boldsymbol{a_2} - \frac{1}{2}\beta\boldsymbol{a_3} \\
- \frac{1}{2}\sum_{k=1}^{K}\|\mathbf{U}^{(k)}\|_F^2 + \frac{1}{2}\beta^2\mathbf{a_4}^\top(\mathbf{K}_{BB} + \beta\mathbf{A}_1)^{-1}\mathbf{a}_4 \\
+ \frac{\beta}{2}\mathrm{tr}(\mathbf{K}_{BB}^{-1}\mathbf{A}_1) + \mathrm{CONST}
\end{aligned}
$$

Additive Structure Over Tensor Entries

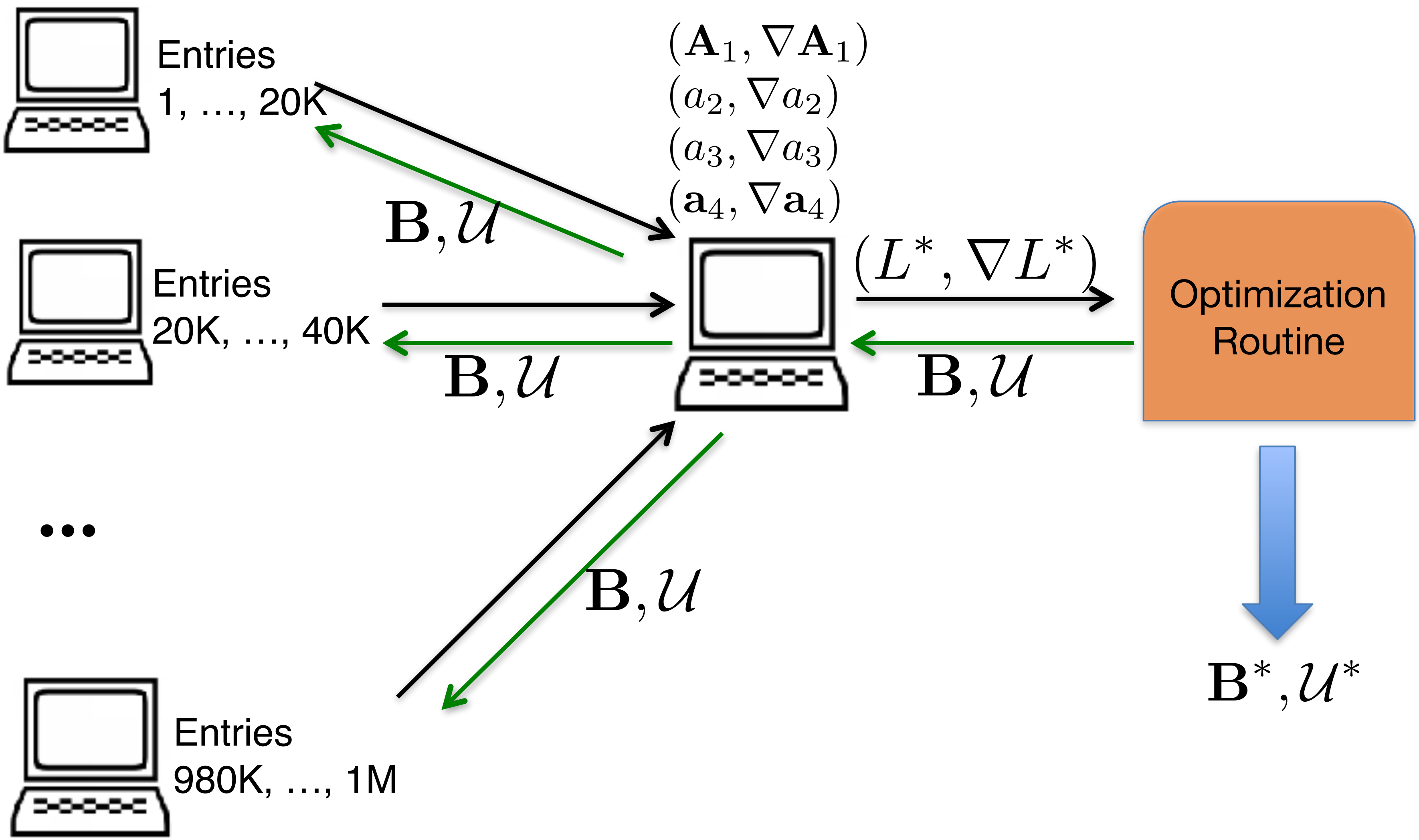
$$\mathbf{A}_1 = \sum_j k(\mathbf{B}, \mathbf{x}_{\mathbf{i}_j}) k(\mathbf{x}_{\mathbf{i}_j}, \mathbf{B})$$

$$a_2 = \sum_j y_{\mathbf{i}_j}^2$$

$$a_3 = \sum_j k(\mathbf{x}_{\mathbf{i}_j}, \mathbf{x}_{\mathbf{i}_j})$$

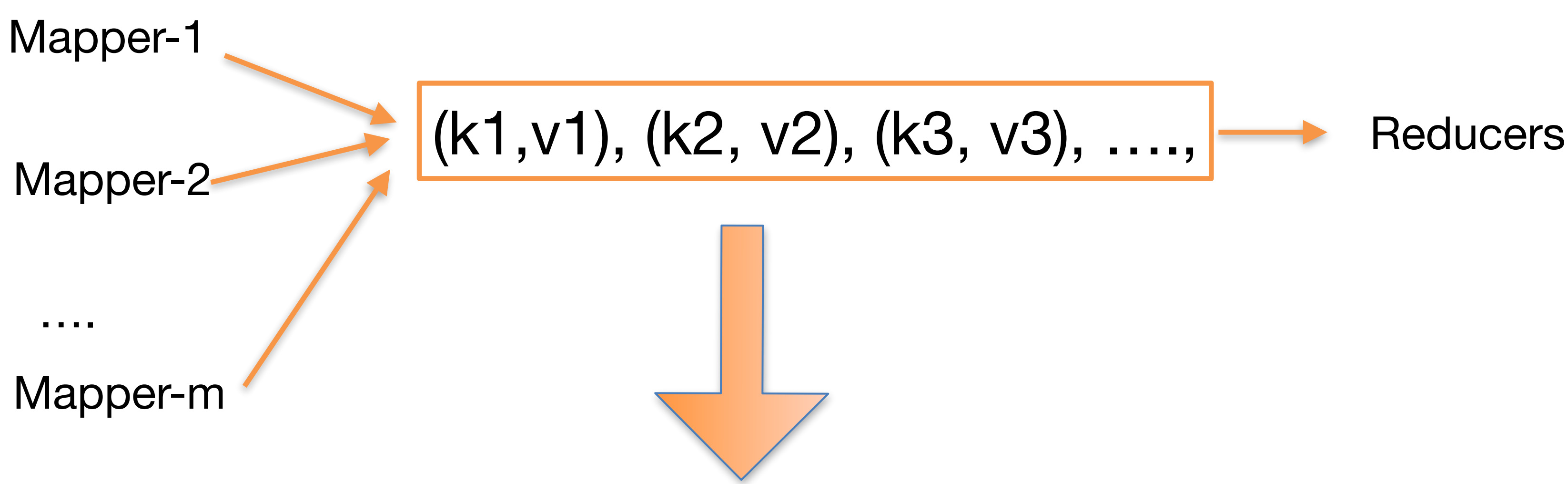
$$\mathbf{a}_4 = \sum_j k(\mathbf{B}, \mathbf{x}_{\mathbf{i}_j}) y_{\mathbf{i}_j}$$

Distributed Optimization Procedure



Map-Reduce Implementation

Standard Key-Value Map-Reduce

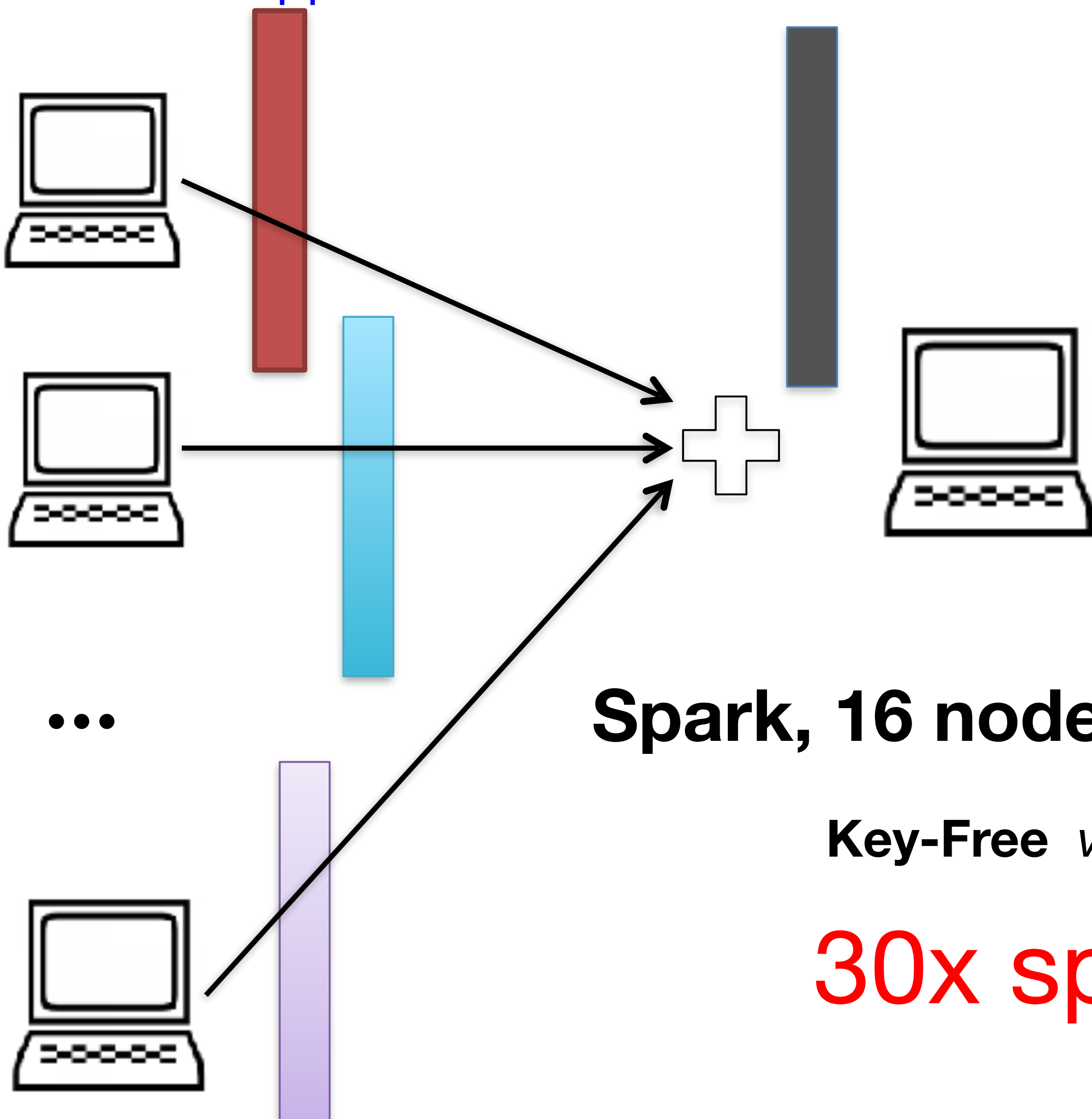


Key-Value sorting: Disk sorting

Key-Free Map-Reduce

Calculate a full gradient on each mapper

Simple summation



No sorting

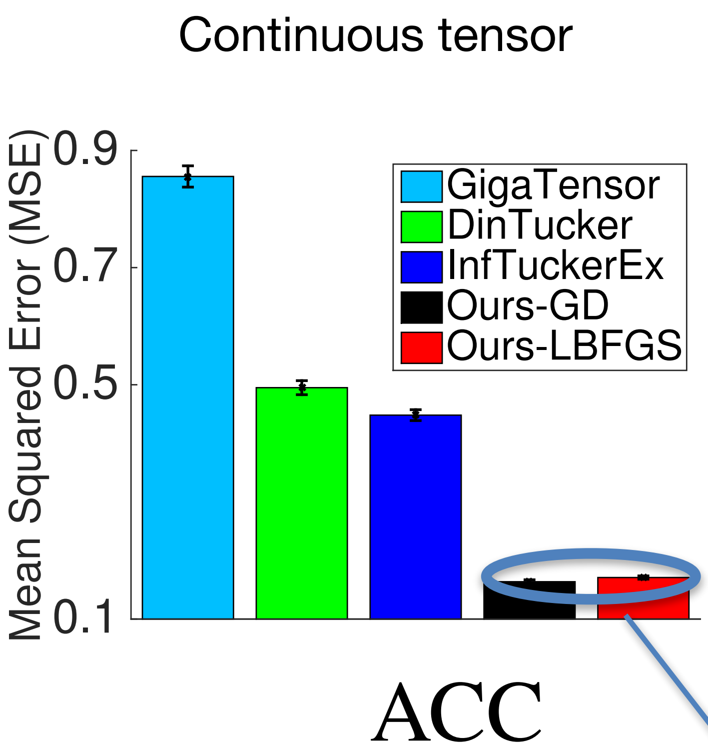
Fully use memory cache

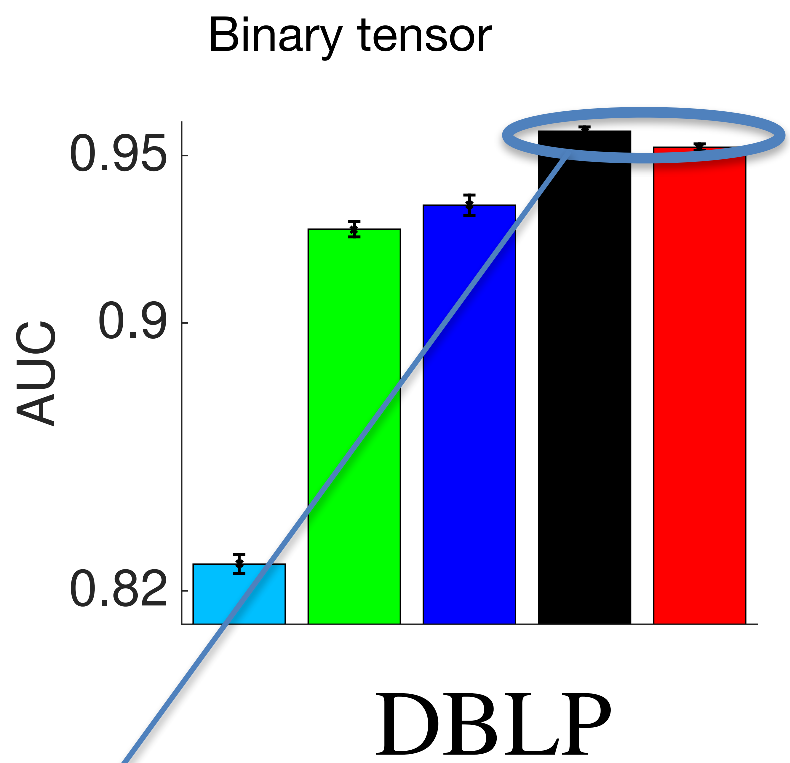
Spark, 16 nodes $100 \times 100 \times 100$

Key-Free *vs* **Key-Value**

30x speed up

Large Tensor with Millions of Nonzero Elements





Flexible GP factorization with Spark

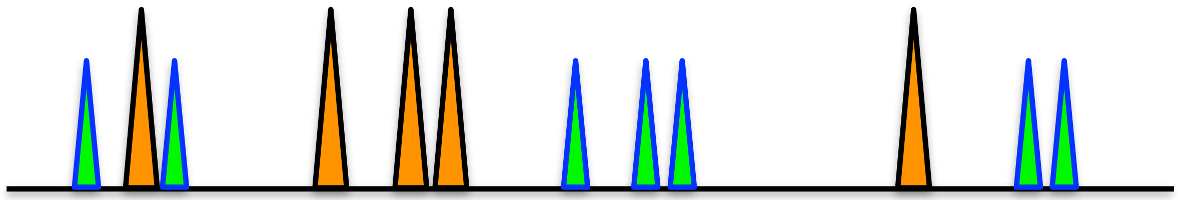
Flexible GP Factorization

1. Factor Concatenation

2. Decomposed Variational Form

3. Key-Free Map-Reduce

How to Deal with Temporal Information



3:10pm

10:10am

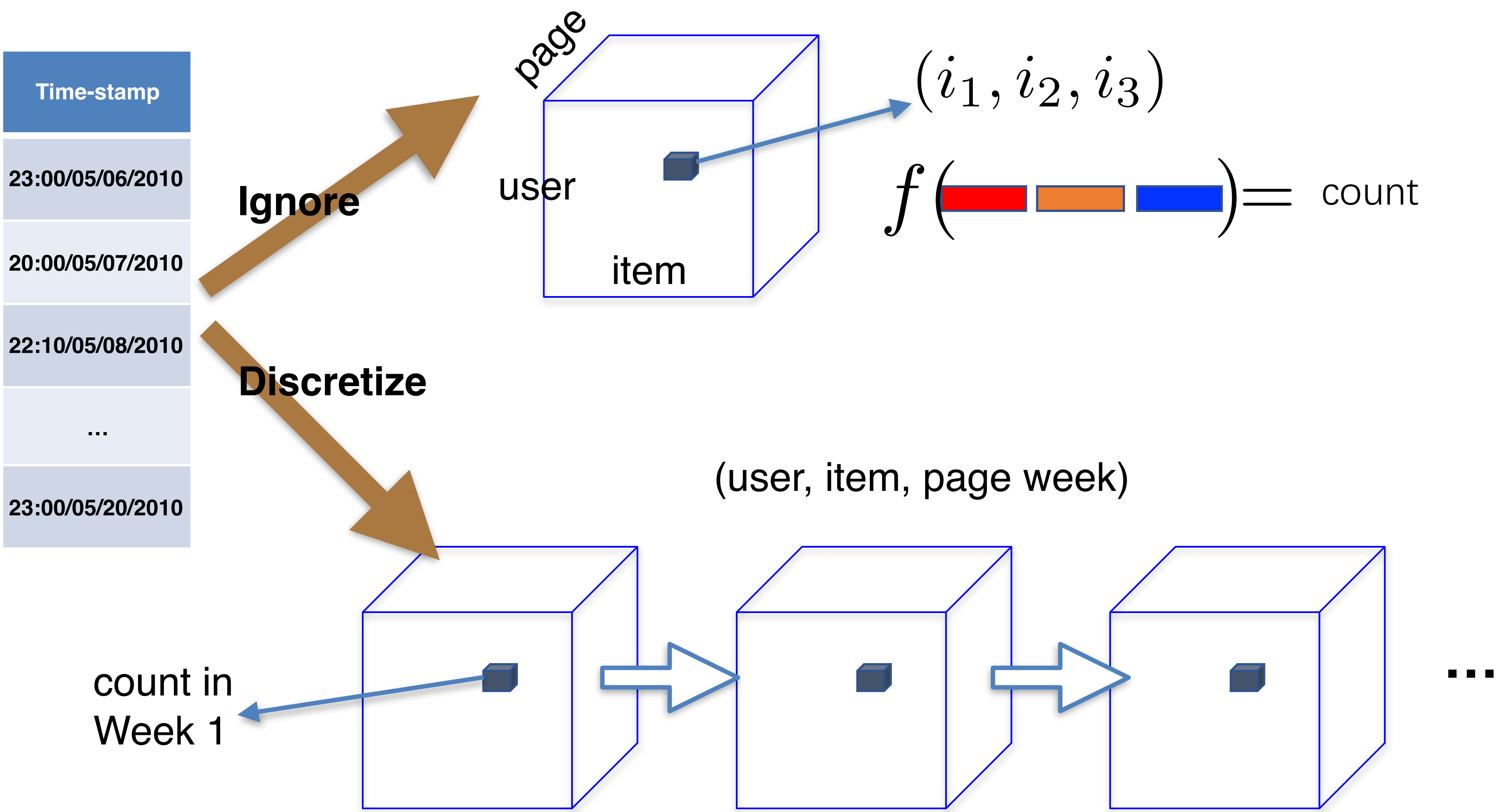
6:20pm

Interaction time series (events)

An Example

User id	Item id	Page id	Buy	Time-stamp
100	25	35	1	23:00/05/06/2010
23	21	56	0	20:00/05/07/2010
100	25	32	1	22:10/05/08/2010
...	...	...	...	...
32	33	46	0	23:00/05/20/2010

Existing Approaches



Existing Approaches

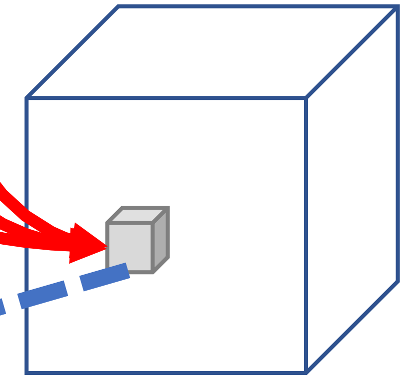
Ignore or over-simplify temporal inferences!

Nonparametric Event-Tensor Decomposition

[***Zhe et. al.***, NIPS'18 spotlight]

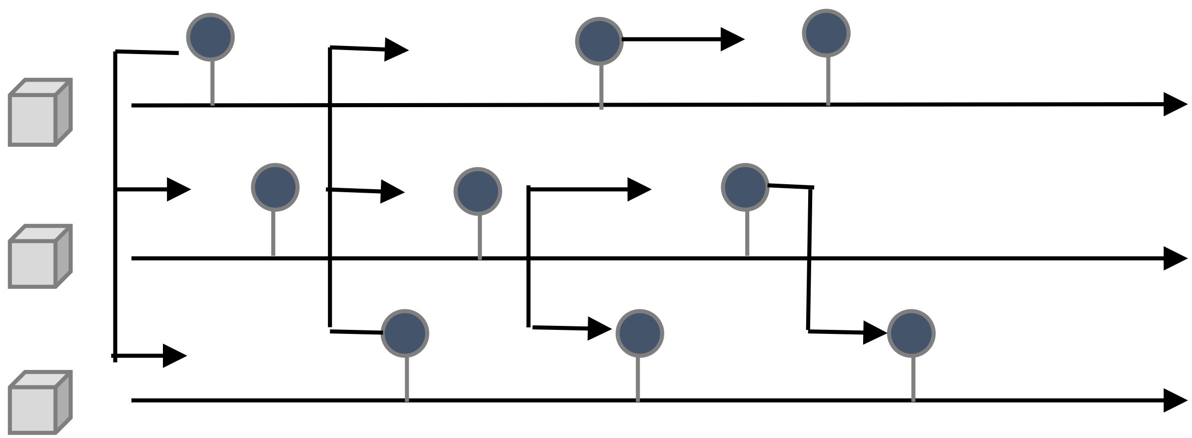
user	item	page	time-stamp
100	25	35	23:00/05/06/2010
23	21	56	20:00/05/07/2010
100	25	35	22:10/05/08/2010
..	..		..
32	33	46	23:00/05/20/2010

Event-Tensor



index	**time-stamps**
(100,25,35)	$\{s_1, s_2 \ldots s_n\}$

Mutually Excited Hawkes Processes



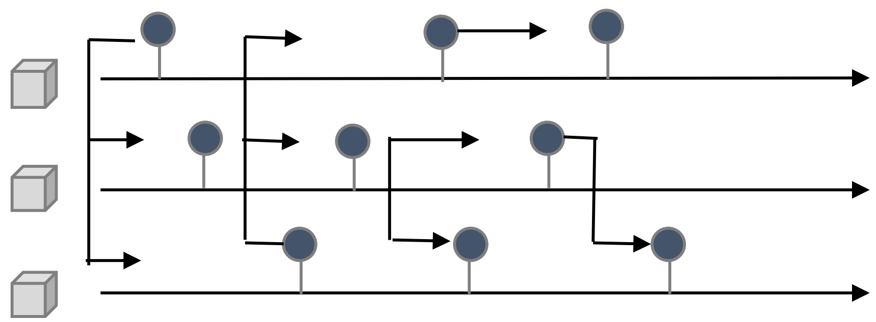
rate function $$\lambda(t) = \lambda_0 + \sum_{s_i < t} h(t - s_i)$$

background rate

triggering-kernel, e.g.,

$$h(\Delta) = \beta \exp(-\frac{\Delta}{\tau})$$

Mutually Excited Hawkes Processes (HPs)



Entry ***i***: $$\lambda_{\mathbf{i}}(t) = \lambda_{\mathbf{i}}^{0} + \sum_{s_n < t} h_{\mathbf{i}_n \to \mathbf{i}}(t - s_n)$$

A function of the factors with GP prior

Static nonlinear relationships

Mutually Excited Hawkes Processes (HPs)

Entry ***i***:

$$\lambda_{\mathbf{i}}(t) = \lambda_{\mathbf{i}}^0 + \sum_{s_n < t} h_{\mathbf{i}_n \to \mathbf{i}}(t - s_n)$$

$$h_{\mathbf{i}_n \to \mathbf{i}}(t - s_n) = k(\mathbf{x}_{\mathbf{i}_n}, \mathbf{x}_{\mathbf{i}}) h_0(t - s_n)$$

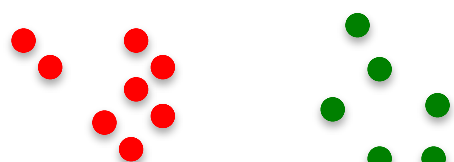
associated factors
with the entry

base triggering kernel

$$\mathbb{1}(s_n \in A_t)\beta e^{-\frac{1}{\tau}(t - s_n)}$$

Help discover triggering clusters!

Stronger mutual excitations
within in the cluster



Hybrid of GPs and HPs

HP $$\lambda_{\mathbf{i}}(t) = \lambda_{\mathbf{i}}^{0}(\mathbf{x}_{\mathbf{i}}) + \sum_{s_n < t} k(\mathbf{x}_{\mathbf{i}_n}, \mathbf{x}_{\mathbf{i}}) h_0(t - s_n)$$

GP

static nonlinear relationships

temporal excitation effects

Latent factors

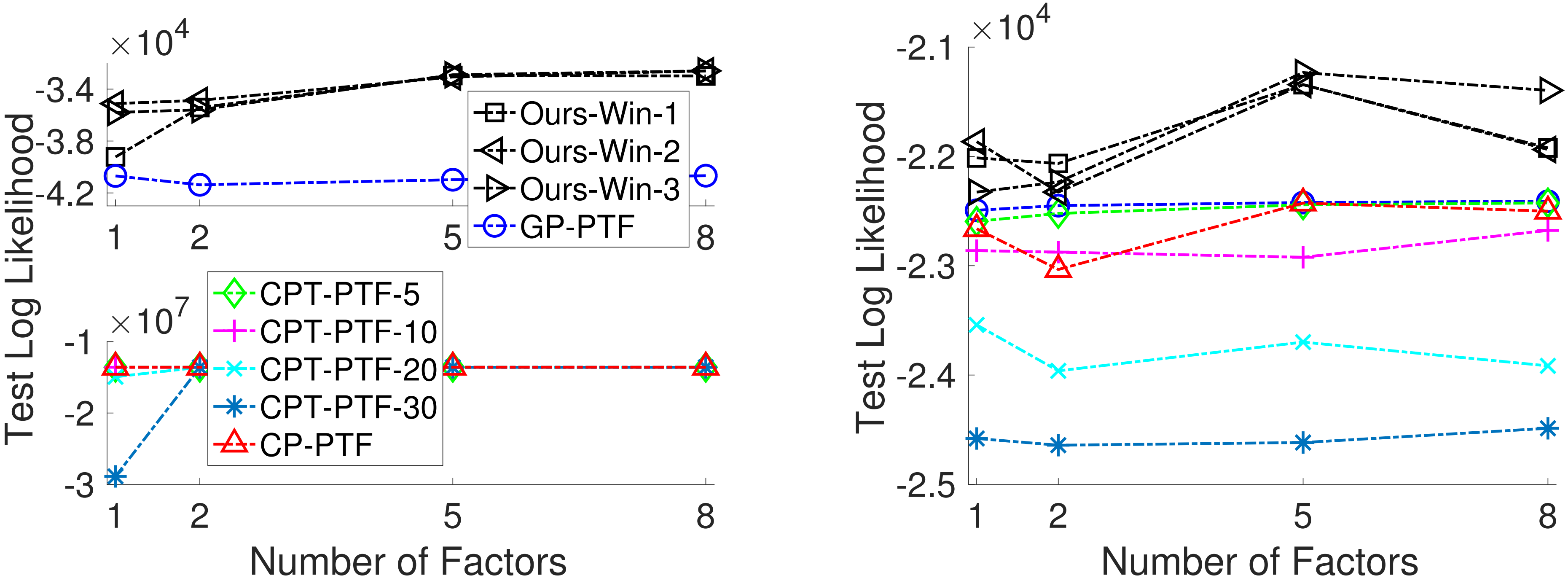
Scalable Inference

Poisson process super-position theorem + variational ***sparse GP framework***

$$\mathcal{L} = \mathbb{E}_{p(k),p(l)}\big(\tilde{\mathcal{L}}_{k,l}\big) \quad \tilde{\mathcal{L}}_{k,l} = \mathbb{E}_{q(\mathbf{g})}\Big(\log\frac{p(\mathbf{g})}{q(\mathbf{g})}\Big) + \sum_{j\in N_k}\phi_{s_j,\bar{A}_{s_j}}\frac{N}{|N_k|} + \underbrace{\boxed{\sum_{j\in N_k}}}_{\text{entries}}\underbrace{\boxed{\sum_{\mathbf{i}\in M_l}}}_{\text{events}}\psi_{s_j,\mathbf{i},\mathbf{i}_j}\frac{N}{|N_k|}\frac{M}{|M_l|}$$

Develop a ***doubly stochastic optimization*** algorithm to maximize the bound

Predictive Performance

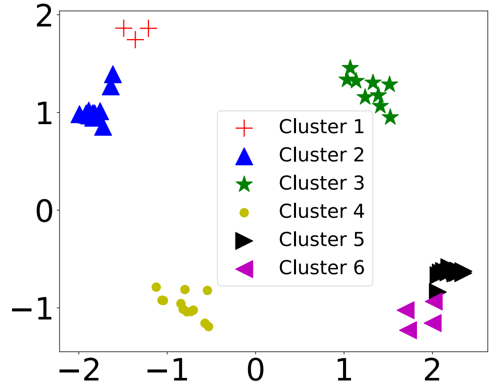


(a) *Article* (b) *911*

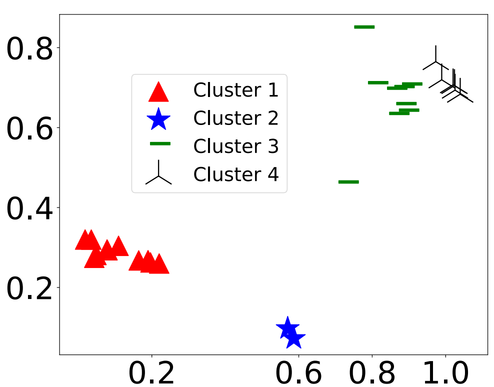
Structure Discovery

- 911 EMS dataset (EMS title, township) 12/10/2015 - 04/10/2017 in Montgomery County, PA.
- UFO sightings (UFO shape, city) in last century

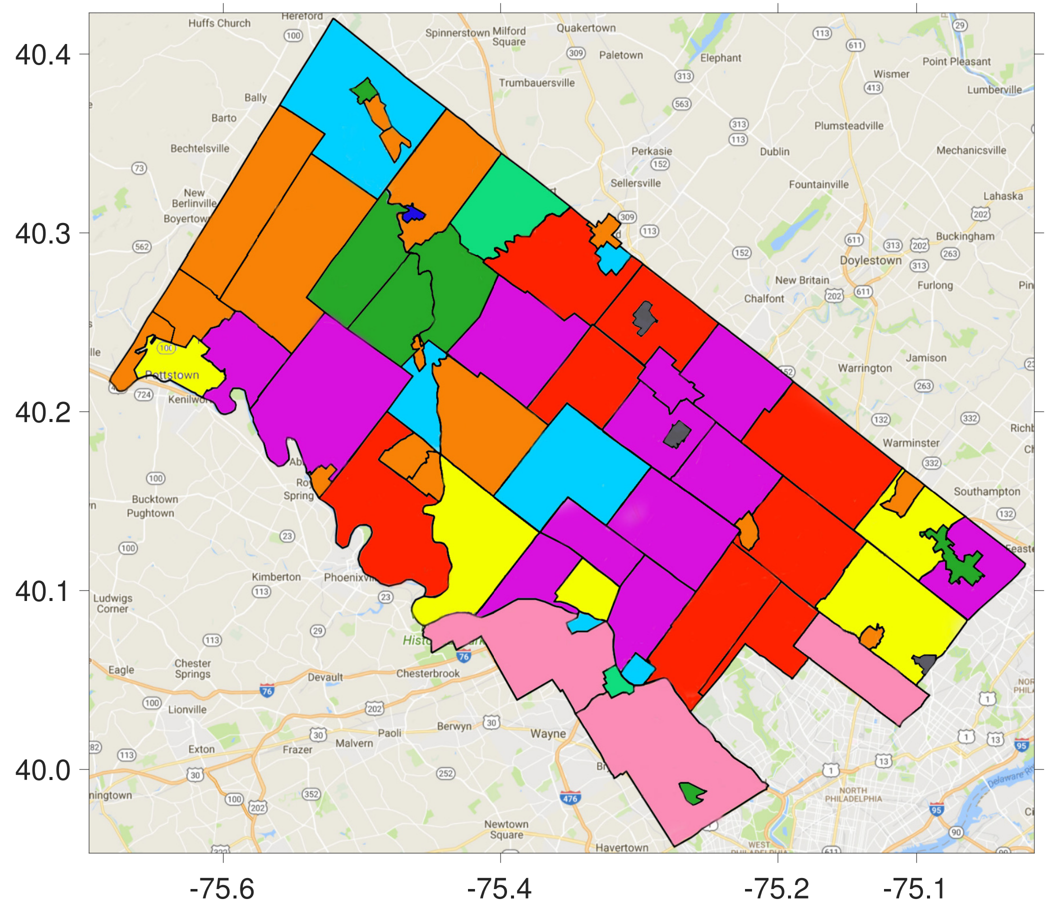
Structure Discovery



(a) EMS titles



(b) UFO shapes



(c) Townships

Nonparametric Event-Tensor Decomposition

1. Hybrid of GP and HP

2. Superposition for decomposable variational bound

Bayesian Learning

As an elegant mathematical framework

- Intuitive model design
- Convenient prior knowledge incorporation
- Excellent interpretation
- Flexible uncertainty reasoning

Could be useful for **numerous** applications

Collaborative filtering, social activities analysis, anomaly detection, community discovery, intelligent decision, disease diagnosis, computational forensic tools, personalized medicine….

Thanks!

Shandian Zhe
zhe@cs.utah.edu
University of Utah