Large Scale Nonparametric Tensor Analysis

Zenglin Xu
http://smilelab.uestc.edu.cn
Big Data Research Center
University of Electronic Science & Technology of China

MLA, Nanjing, 2015

电子科技大学
University of Electronic Science and Technology of China

大数据挖掘与推理研究所

- **异构多源大数据处理与建模**

  实时数据处理、多源数据处理、时间空间数据分析、复杂网络数据分析、金融大数据建模、媒体大数据建模、医学大数据建模、移动大数据建模

- **大数据智能计算与分析技术**

  分布式大数据查询技术、机器学习算法研究、分布式机器学习、随机化算法与在线学习、社会网络分析、推荐系统、深度学习算法

- **大数据分布式计算模型与系统**

  大数据机器学习平台研究、面向行业应用（如医疗、教育、安全、移动数据）的大数据分析与学习平台设计等

- **大数据知识表示与推理技术研究**

  大型本体知识库构建方法和本体映射等知识深层理解的关键处理算法、知识的深层表示、知识图谱、大型知识库上逻辑推理机制和机器学习

Join Us

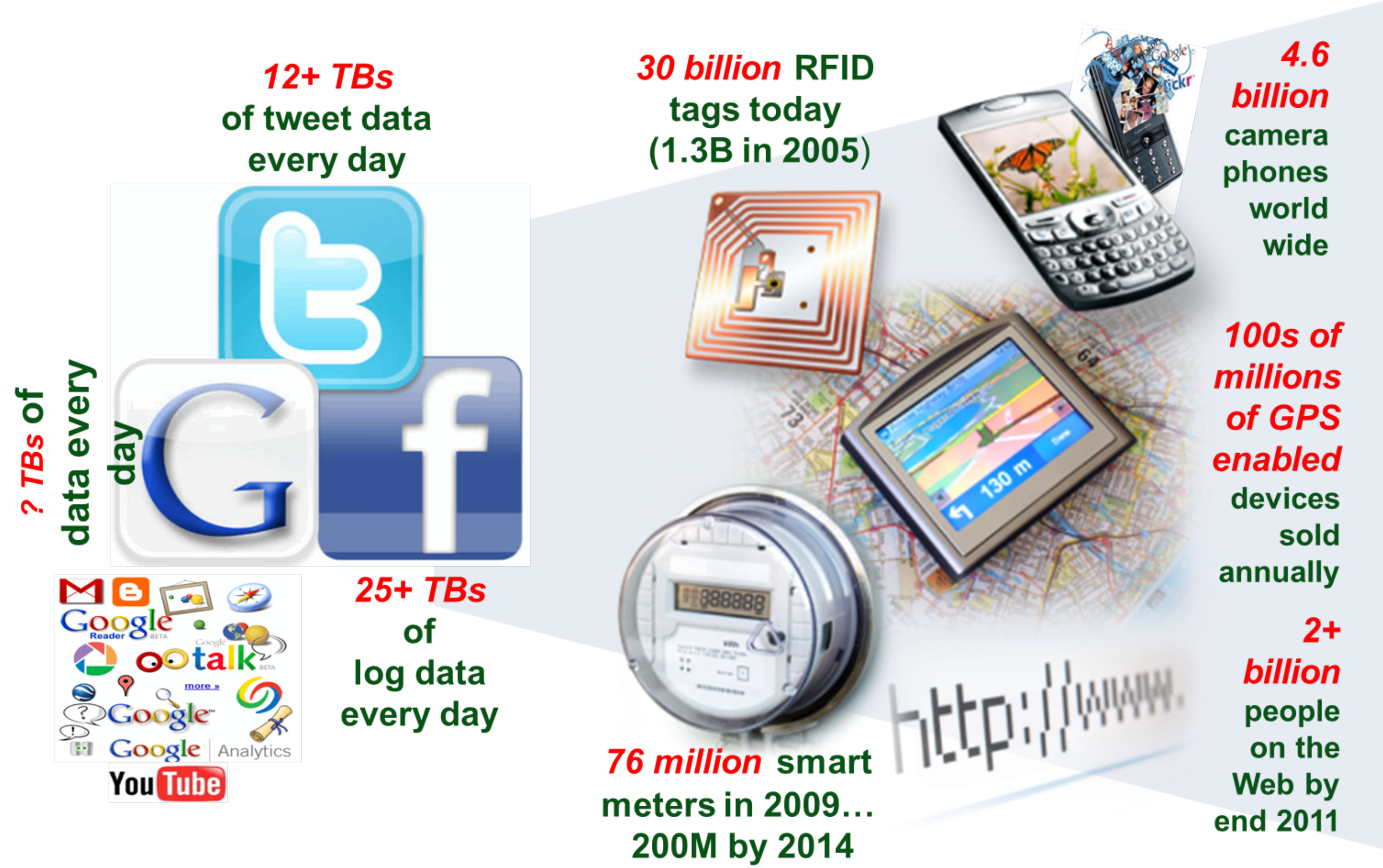
中组部“青年千人计划”入选者徐增林教授团队，因科研和教学工作需要，面向海内外诚聘优秀青年学者加盟。团队的研究着重于机器学习、统计学习、数据挖掘技术及其在社会网络分析、医学图像处理、空间安全数据分析、自然语言处理、神经信息学等方面的应用。
http://smilelab.uestc.edu.cn/index.php?title=HOME

- **特聘教授/特聘副教授/讲师**
- **在职和专职博士后／研究助理**
- **博士生/硕士生**

Outline

- Motivation
- Multilinear tensor decomposition
- Nonparametric nonlinear tensor models
- Large scale models

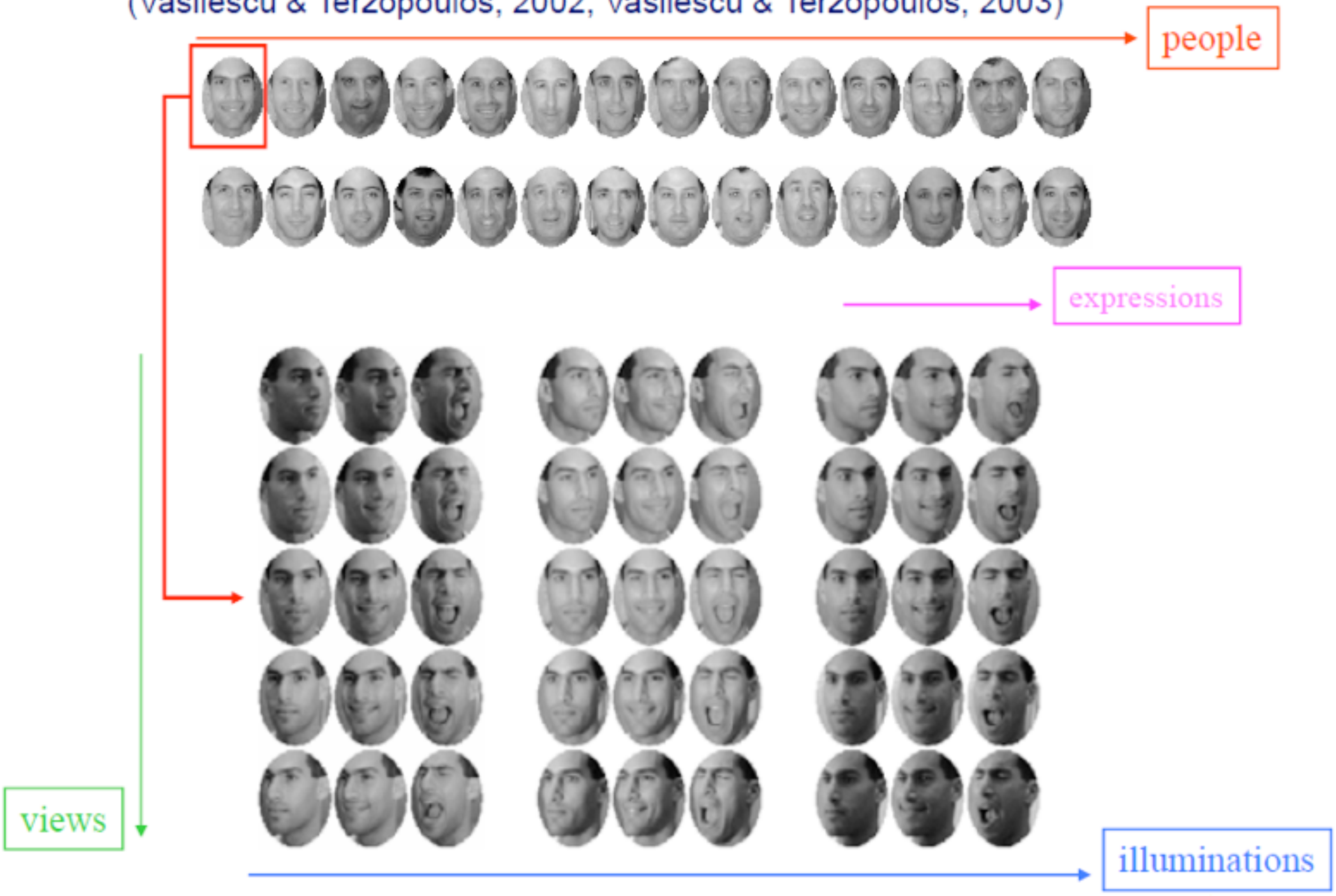
Big Data is Everywhere & Data have multiple persectives



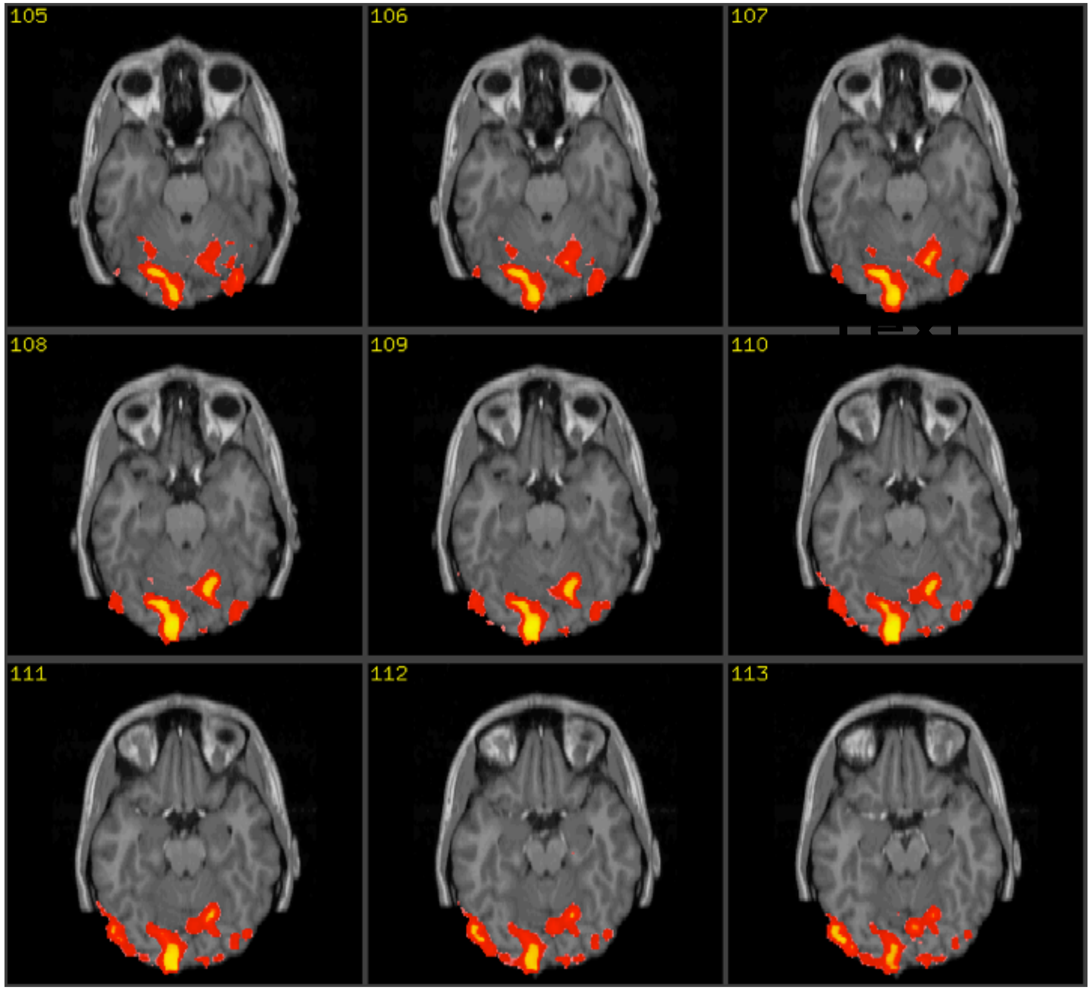
电子科技大学
University of Electronic Science and Technology of China

Tensor data: facial expressions

(Vasilescu & Terzopoulos, 2002; Vasilescu & Terzopoulos, 2003)



Tensor data: fMRI

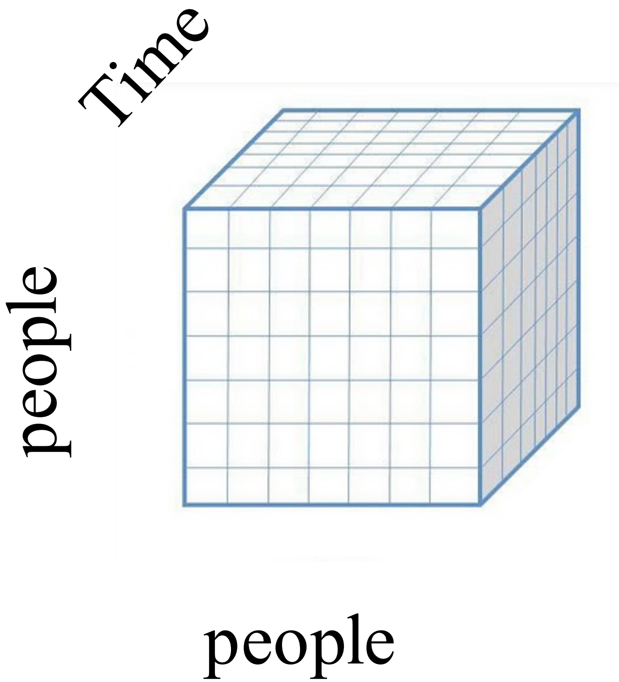


Activities of brain regions time at different time points

Discover the interactions of different regions

电子科技大学
University of Electronic Science and Technology of China

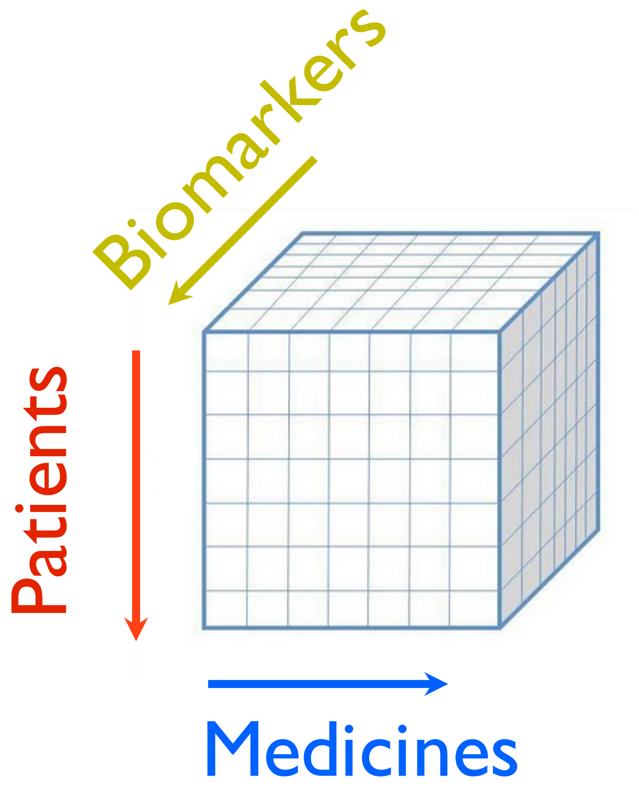
Tensor data: dynamic networks



$Y_{i,j,k}$: 1 if node i is linked to node j at time k; 0 otherwise.

Who will be your friends on facebook tomorrow?

Tensor data: drug responses



$Y_{i,j,k}$: the value of the k-th biomarker (i.e., cell population) for the j-th patient after taking the i-th medicine

Predict drug response

Goals

- Predict unknown elements (e.g., drug response and network interactions)

- Identify latent multi-aspect groups (communities)

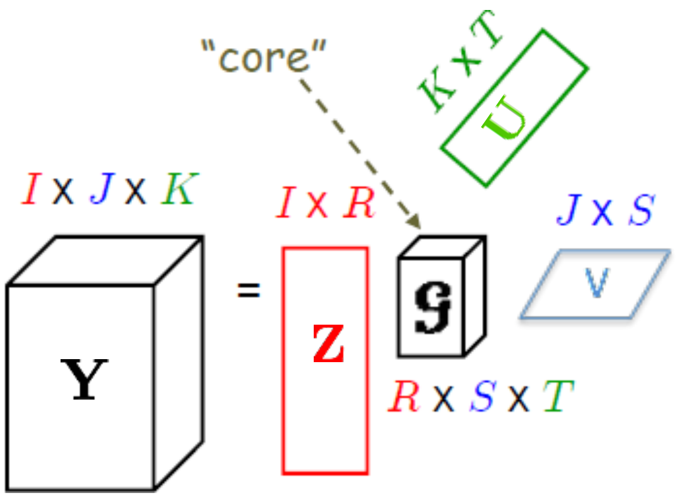
Outline

- Motivation
- Multilinear tensor decomposition
- Nonparametric nonlinear tensor models
- Large scale models

Classical Tucker decomposition

Generalization of matrix factorization

3D case:



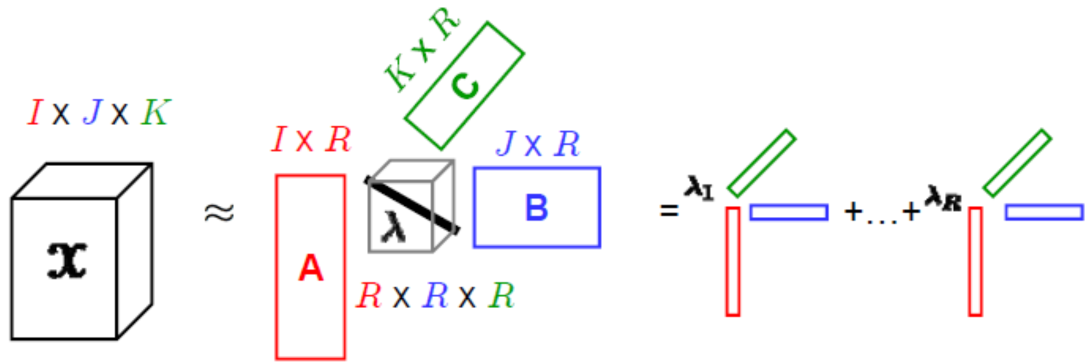
core tensor loading matrices

$$\mathbf{Y} = \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{Z} \times_3 \mathbf{V}$$

$$y_{ijk} = \sum_r \sum_s \sum_t g_{rst} u_{ir} z_{js} v_{ks}$$

Sun et al. 2008

PARAFAC (canonical polyadic decomposition) (CPD))



diagonal core tensor

loading matrices

$$\mathbf{Y} = \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{Z} \times_3 \mathbf{V}$$

$$y_{ijk} = \sum_r \sum_s \sum_t g_{rst} u_{ir} z_{js} v_{ks}$$

$$g_{rst} \neq 0 \text{ if } r = s = t;\ g_{rst} = 0, \text{ otherwise.}$$

电子科技大学
University of Electronic Science and Technology of China

PARAFAC vs. Tucker Decomposition

- Pros
  - Less prune to overfitting
  - Faster computation
- Cons
  - Less representation power

Limitations

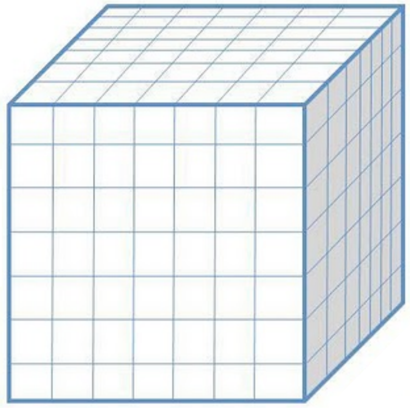
- Complete
- Continuous
- Multi-linear

Outline

- Motivation
- Multilinear tensor decomposition
- Nonparametric nonlinear tensor models
- Large scale models

电子科技大学
University of Electronic Science and Technology of China

Tensor-variate Gaussian Processes



Sparse latent Gaussian processes on tensors

Why Gaussian Process?

- Bayesian methods
- Graphical model
- Non-parametric methods

Bayesian Methods

- The Bayesian approach treats the parameters themselves as random variables having distributions:

1. Beliefs about our parameter values θ --- encoded in the *prior distribution* $P(\theta)$.
2. Treating the parameters θ as random variables --- the likelihood of data X as a conditional probability: $P(X|\theta)$.
3. Update our beliefs about θ based on the data by obtaining $P(\theta|X)$, the *posterior distribution*.

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

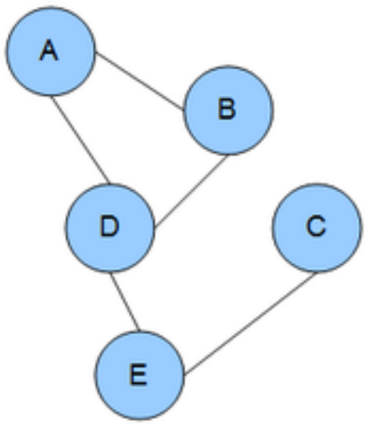
where

$$P(X) = \int P(X|\theta)P(\theta)d\theta$$

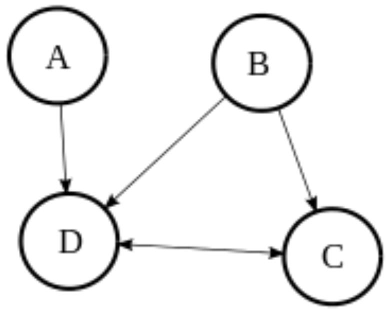
Graphical Model

- A **graphical model** is a probabilistic model (**Probabilistic Graphical Model**, or **PGM** for short) for which a graph denotes the conditional dependence structure between random variables.

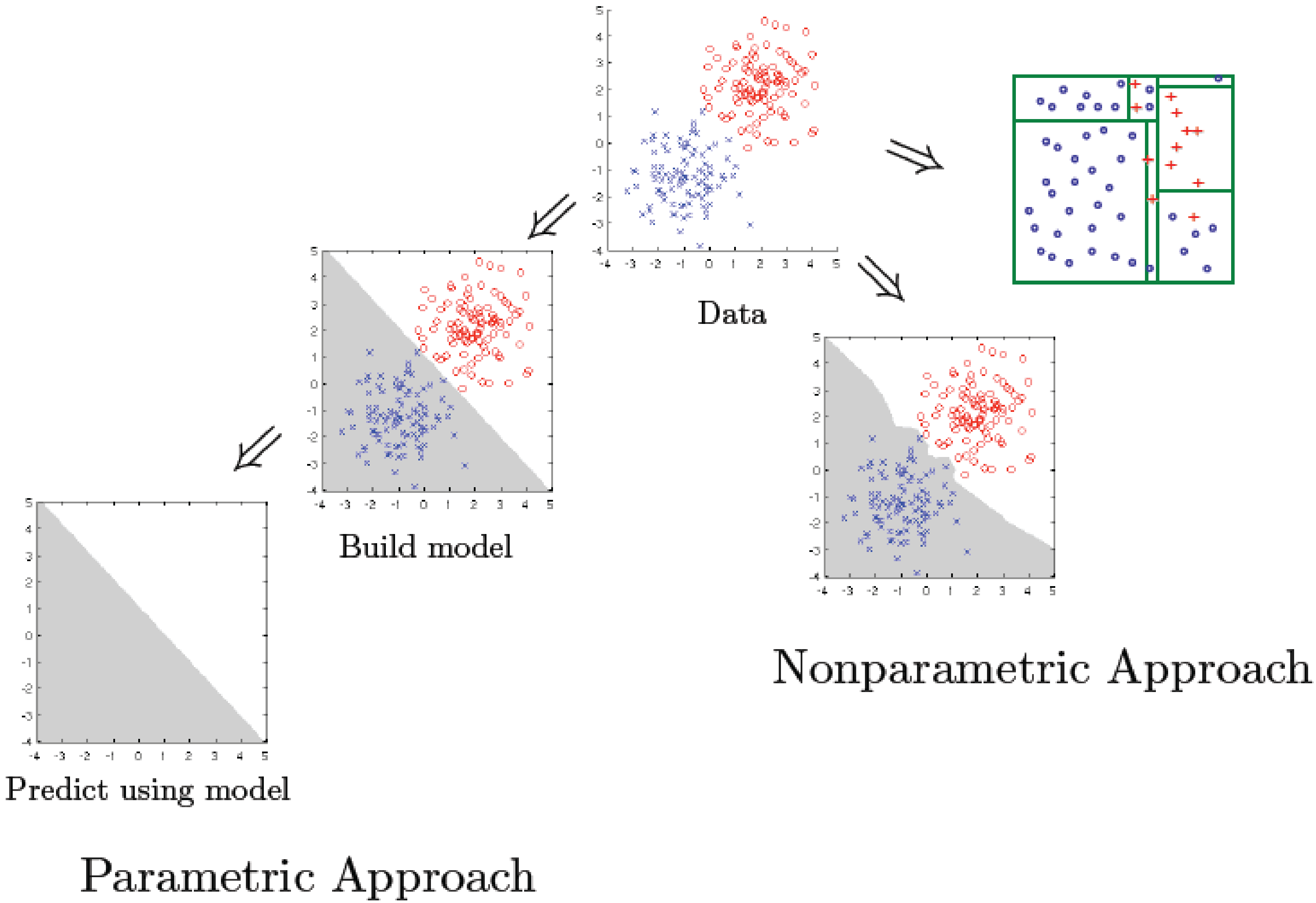
Markov Random Field



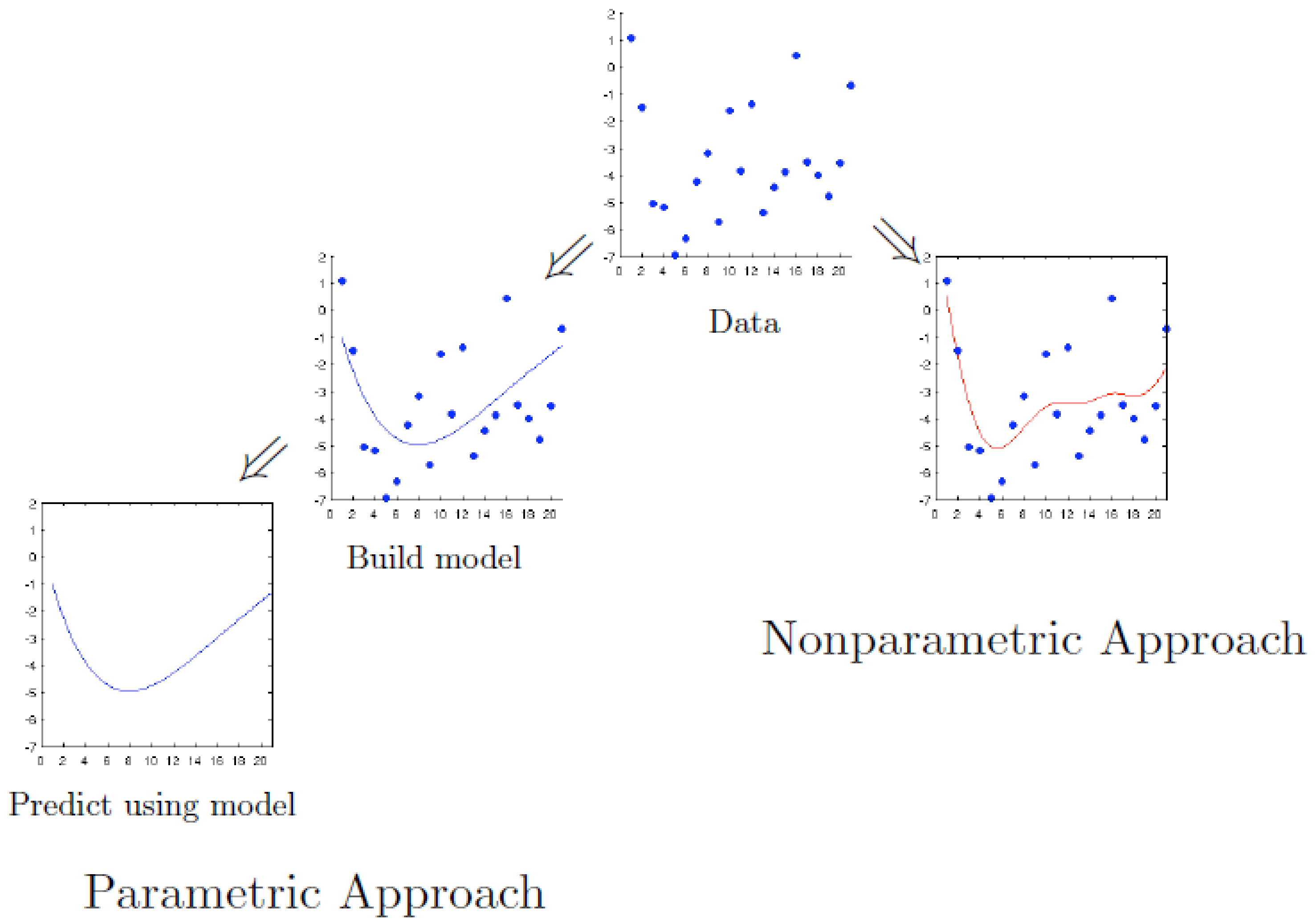
Bayesian Network



Nonparametric Classification



Non-parametric Regression



Nonparametric Bayesian Methods Examples

- Dirichlet Process/Chinese Restaurant Process
  - Latent class models - often used in the clustering context
- Beta Process/Indian Buffet Process
  - Latent feature models
- Gaussian Process
  - Regression
- Today we focus on the Gaussian Process!

Gaussian Process

- A Gaussian process is a stochastic process whose realizations consist of random values associated with every point in a range of times (or of space) such that each such random variable has a normal distribution.
- Gaussian processes (GPs) extend multivariate Gaussian distributions to infinite dimensionality.
- Formally, a Gaussian process generates data located throughout some domain, such that any finite subset of the range follows a multivariate Gaussian distribution.

An example of Gaussian Process

- Given {x_i, y_i}, predict $y_*|x_*$

$$y = f(\mathbf{x}) + \varepsilon,$$

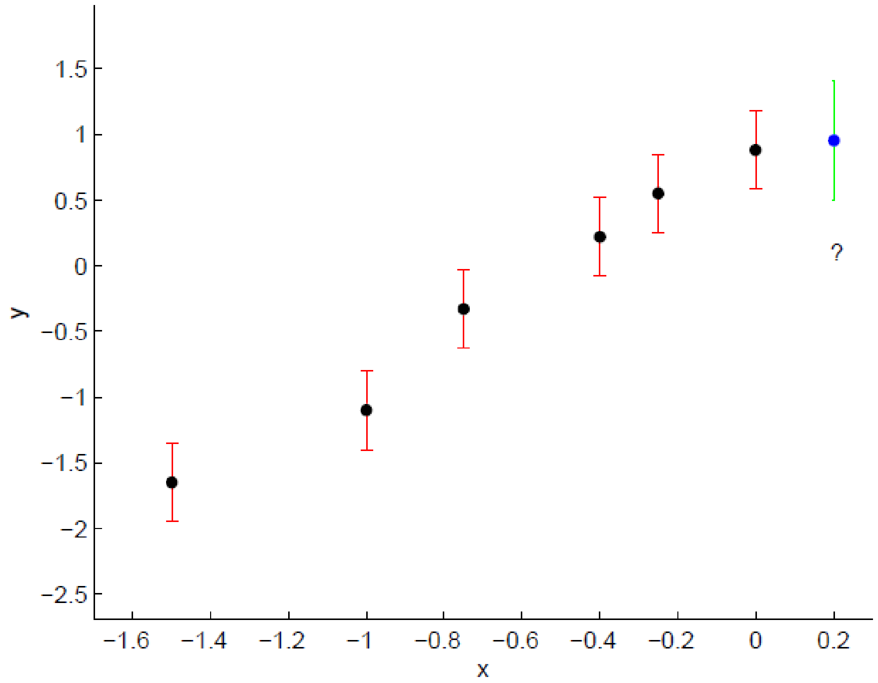
- Gaussian noise model

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big).$$

$$\varepsilon \sim \mathcal{N}(0,\ \sigma_n^2).$$

- Covariance function

$$k(x, x') = \sigma_f^2 \exp\left[\frac{-(x - x')^2}{2l^2}\right],$$



An example of Gaussian Process

- Covariance Matrix

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix}$$

$$K_* = \begin{bmatrix} k(x_*, x_1) & k(x_*, x_2) & \cdots & k(x_*, x_n) \end{bmatrix} \qquad K_{**} = k(x_*, x_*).$$

- Joint distribution – Normal distribution

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K & K_*^{\mathrm{T}} \\ K_* & K_{**} \end{bmatrix}\right)$$

An example of Gaussian Process

- Predictive distribution

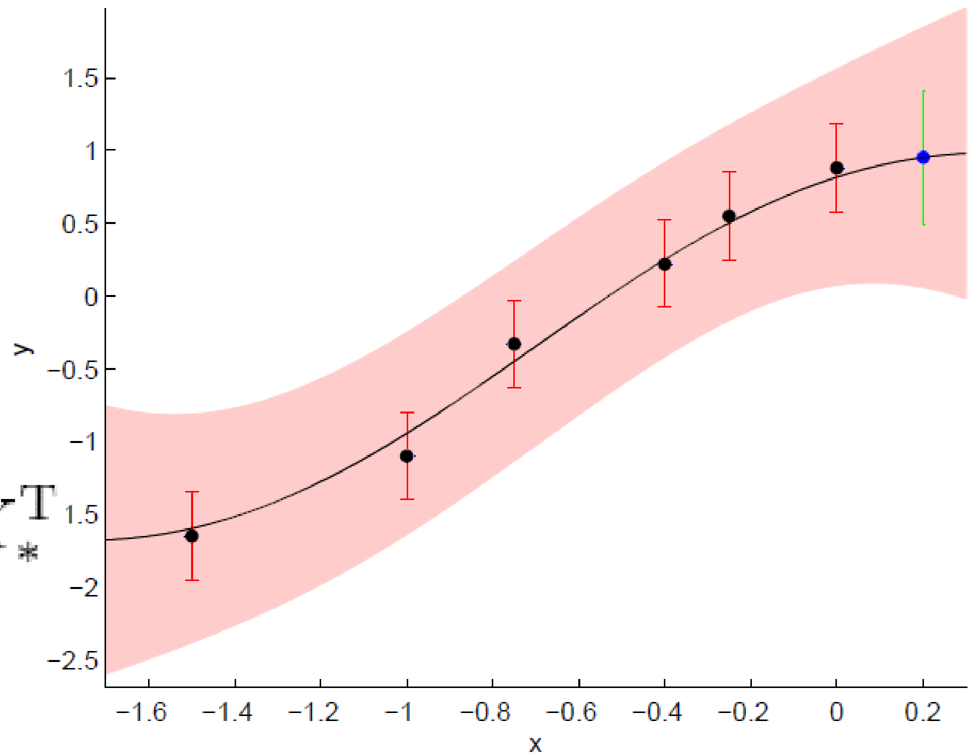
$$y_* | \mathbf{y} \sim \mathcal{N}(K_* K^{-1} \mathbf{y},\ K_{**} - K_* K^{-1} K_*^{\mathrm{T}})$$

- Mean

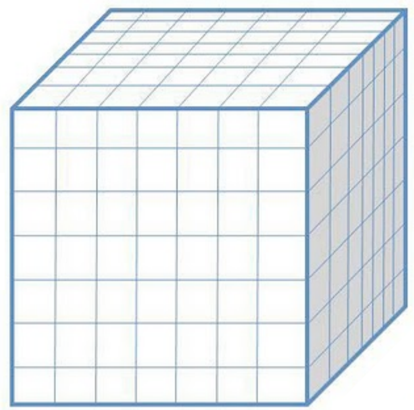
$$\overline{y}_* = K_* K^{-1} \mathbf{y}$$

- Variance

$$\mathrm{var}(y_*) = K_{**} - K_* K^{-1} K_*^{\mathrm{T}}$$

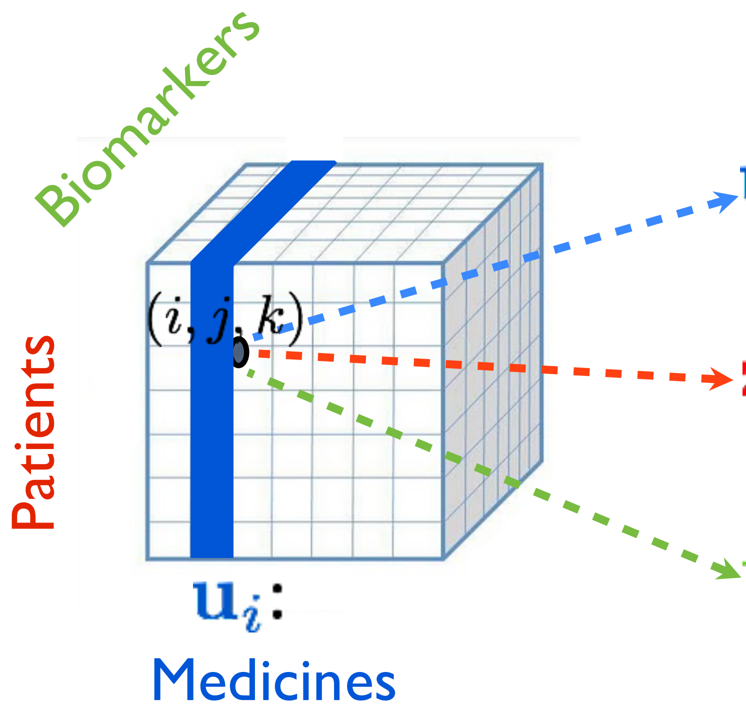


Our Solutions: Infinite Tucker Decomposition



Sparse latent Gaussian processes on tensors

Latent sparse GP on tensors



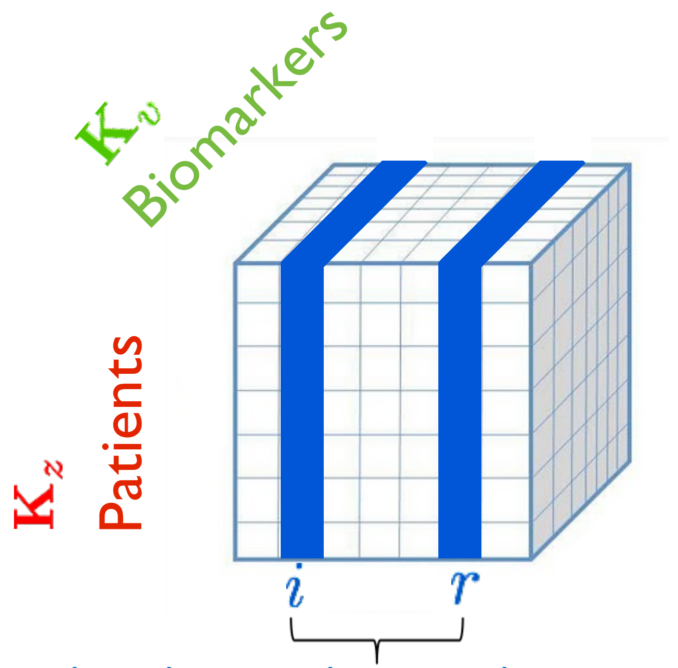
Element $(i,\ j,\ k)$ is characterized by

$\mathbf{u}_i$: Sparse loading vector in latent medicine groups

$\mathbf{z}_j$: Sparse loading vector in latent patient groups

$\mathbf{v}_k$: Sparse loading vector in latent biomarker groups

Separate covariance for each dimension



$$\mathbf{K}_u(i, r) = k(\mathbf{u}_i, \mathbf{u}_r)$$

Medicines

Nonlinear relationship between medicines i and r

-Separate covariance/kernel function for each dimension

-The more similar loading vectors, the larger the covariance function value

GP on tensors

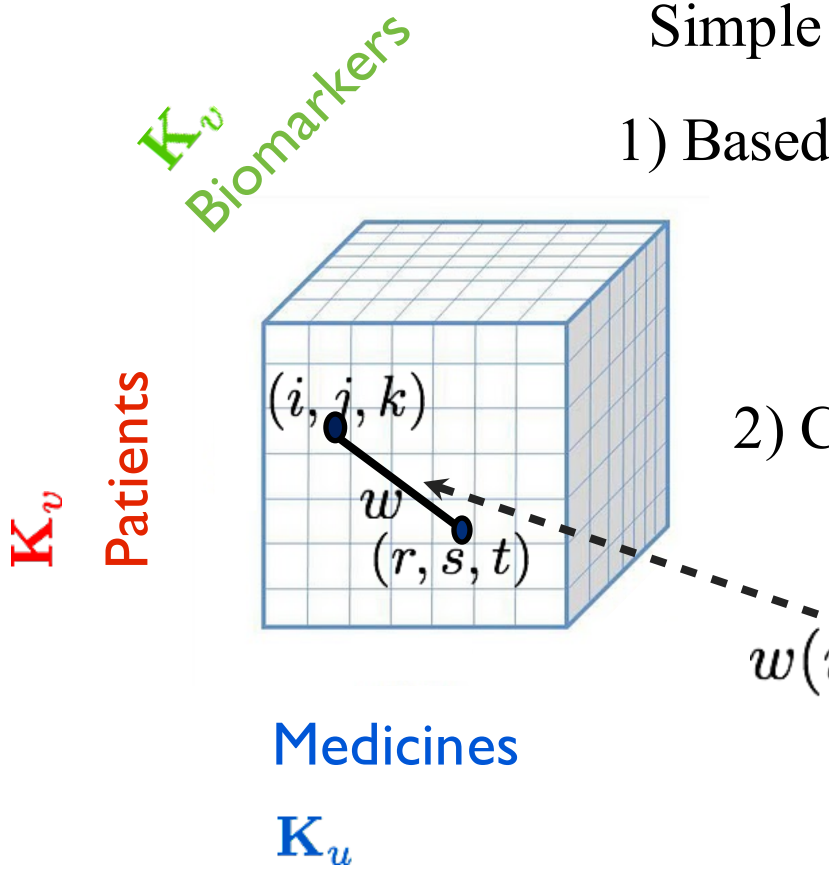
- GP on a tensor: stochastic process in an infinite tensor space
- Evaluations of GP on any tensor of finite size is a tensor-valued Gaussian distribution

Tensor

$$\mathcal{N}(\mathbf{F}|\mathbf{0};\mathbf{K}_u,\mathbf{K}_v,\mathbf{K}_z) = (2\pi)^{-\frac{3N}{2}} \prod_{s=u,v,z} |\mathbf{K}_s|^{-\frac{N^2}{2}}$$

$$\cdot \exp\{-\frac{1}{2}||(\mathbf{F} \times_1 \mathbf{K}_u^{-1} \times_2 \mathbf{K}_v^{-1} \times_3 \mathbf{K}_z^{-1}) \circ \mathbf{F}||^2\}$$

Predict unknown tensor elements



Simple illustration:

1) Based on observed data, estimate loading vectors:

$$[\mathbf{u}_i, \mathbf{z}_j, \mathbf{v}_k]$$

2) Compute weights (similarities) between unknown and observed elements:

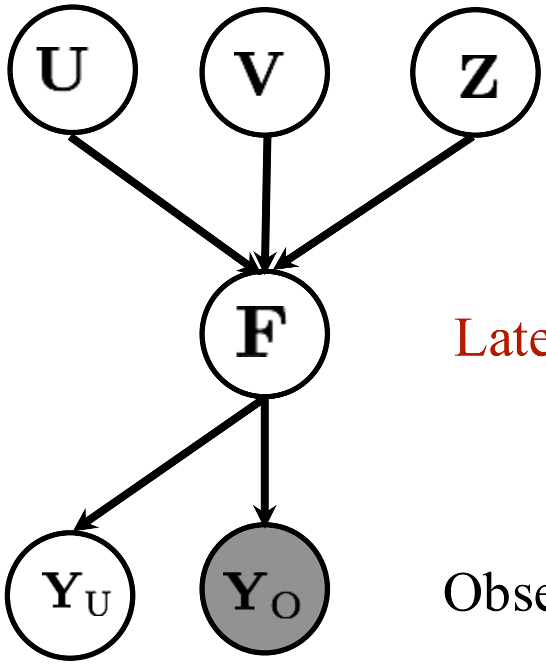
$$w(ijk, rst) \equiv w([\mathbf{u}_i, \mathbf{z}_j, \mathbf{v}_k], [\mathbf{u}_r, \mathbf{z}_s, \mathbf{v}_t])$$

3) Predict the unknown element:

$$y_{i,j,k} = \sum_{r,s,t} w(ijk, rst) y_{r,s,t}$$

Graphical model representation

$\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$



Unknown data

Sparse loading vectors $\mathbf{u}_i \sim \exp(-\lambda |\mathbf{u}_i|)$

Similarly, sample $\mathbf{V}$ and $\mathbf{Z}$

Latent tensor $\mathbf{F} \sim \mathcal{N}(\mathbf{0}; \mathbf{K}_u, \mathbf{K}_v, \mathbf{K}_z)$

Observed data

$$\mathbf{Y}_{\mathrm{O}} \sim \prod_{(i,j,k)\in O} p(y_{ijk}|f_{ijk})$$

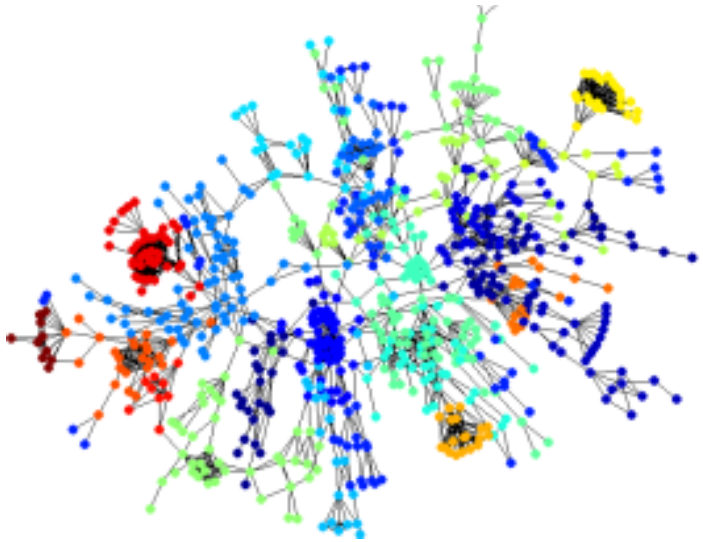
$\boxed{p(y_{ijk}|f_{ijk})}$ Gaussian for continuous data

Probit for binary data

Possion for count data

Benefits

- Handle binary and missing data
- Discover block/group structures
- Avoid overfitting: adaptive nonparametric model complexity
- Model prediction uncertainty
- Incorporate additional side information



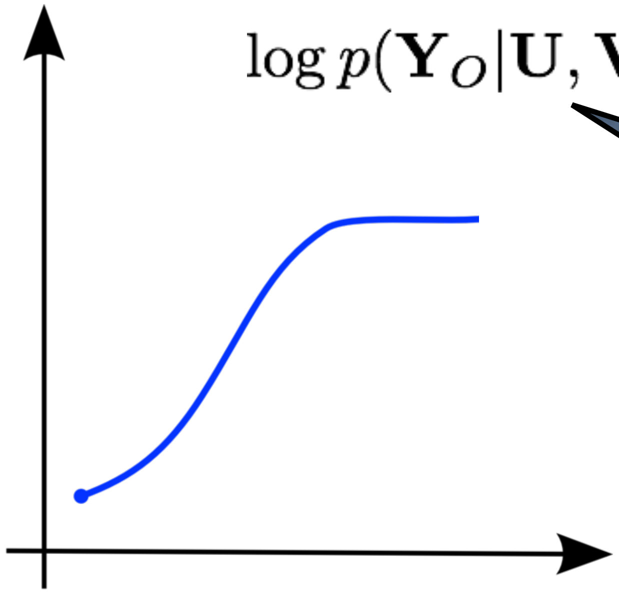
Yan, Xu & Qi, 2011; Xu, Yan & Qi 2011; Xu, Yan & Qi 2015

Algorithm: Variational EM

Marginal likelihood

$$\log p(\mathbf{Y}_O|\mathbf{U},\mathbf{V},\mathbf{Z}) + \log p(\mathbf{U},\mathbf{V},\mathbf{Z})$$

Variational approximation



Iterations

Algorithm: explore model structures

Example: $$\mathrm{Trace}\{\underbrace{(\mathbf{I} + \mathbf{K}_u \otimes \mathbf{K}_v \otimes \mathbf{K}_z)}_{N^3 \text{ by } N^3}^{-1}\}$$

Direct computation:
Matrix inversion

$O(N^9)$

Kronecker product operation:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

Properties of Kronecker product

Properties: $(AB) \otimes (CD) = (A \otimes B)(C \otimes D)$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

Property: Eigen-decomposition $\mathbf{K} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^{\mathrm{T}}$

If $\mathbf{K} = \mathbf{K}_u \otimes \mathbf{K}_v \otimes \mathbf{K}_z$ then

$$\mathbf{W} = \mathbf{W}_u \otimes \mathbf{W}_v \otimes \mathbf{W}_z$$

$$\mathbf{\Lambda} = \mathbf{\Lambda}_u \otimes \mathbf{\Lambda}_v \otimes \mathbf{\Lambda}_z$$

$$\mathbf{W}_u : \text{eigenvectors of } \mathbf{K}_u$$

$$\mathrm{diag}\{\mathbf{\Lambda}_u\} : \text{eigenvalues of } \mathbf{K}_u$$

Reduced computational complexity

Example: $\mathrm{Trace}\{(\mathbf{I}+\mathbf{K}_u \otimes \mathbf{K}_v \otimes \mathbf{K}_z)^{-1}\}$

Direct computation

$O(N^9)$

Using the new theorem and trace properties

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\frac{1}{1+\lambda_i^u\lambda_j^v\lambda_k^z} \quad O(N^3)$$

$$=\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M}\frac{1}{1+\lambda_i^u\lambda_j^v\lambda_k^z} \quad O(N^2M)$$

$M << N$

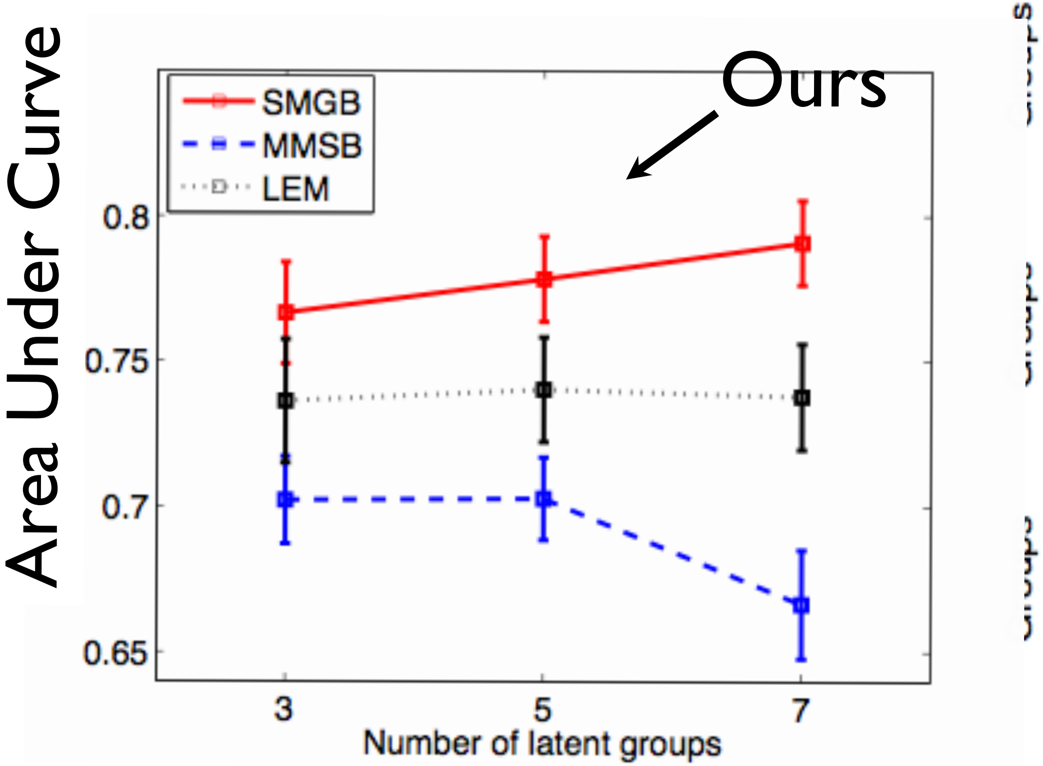
2D case: GP stochastic blockmodels

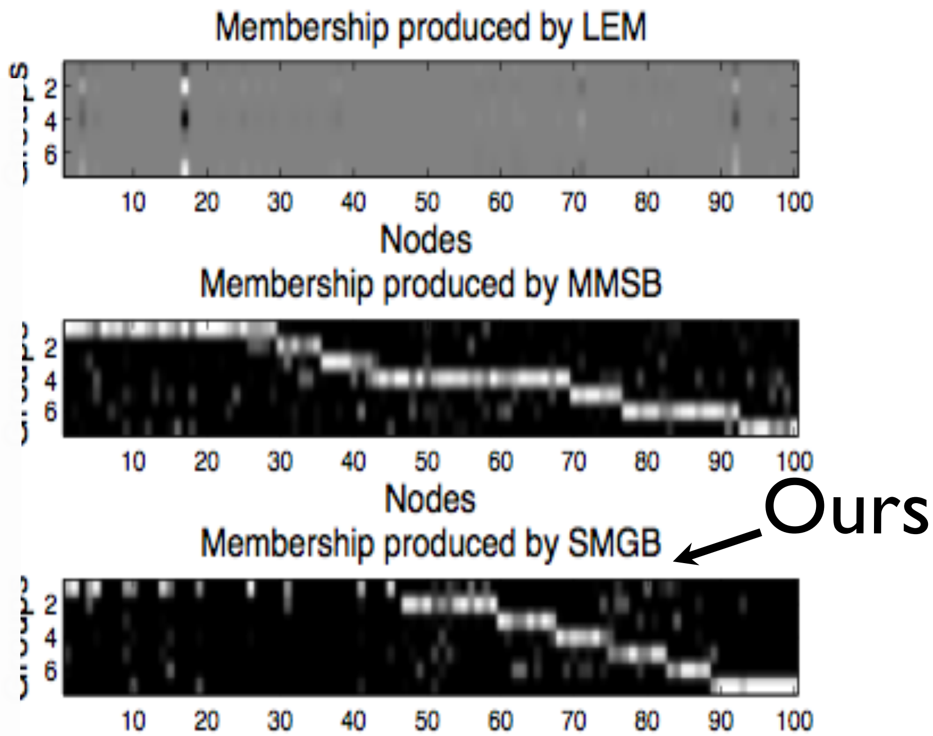
- Undirected networks (friend relationships and protein-protein interactions)
- Represented by symmetric adjacent matrices

Yan, Xu & Qi, UAI 2011; Xu, Yan & Qi AAAI 2011

电子科技大学
University of Electronic Science and Technology of China

2D: Coauthor networks





NIPS authors

Co-authorship dataset: co-authorship links from100 authors who have the largest number of co-authors from NIPS 1-17.

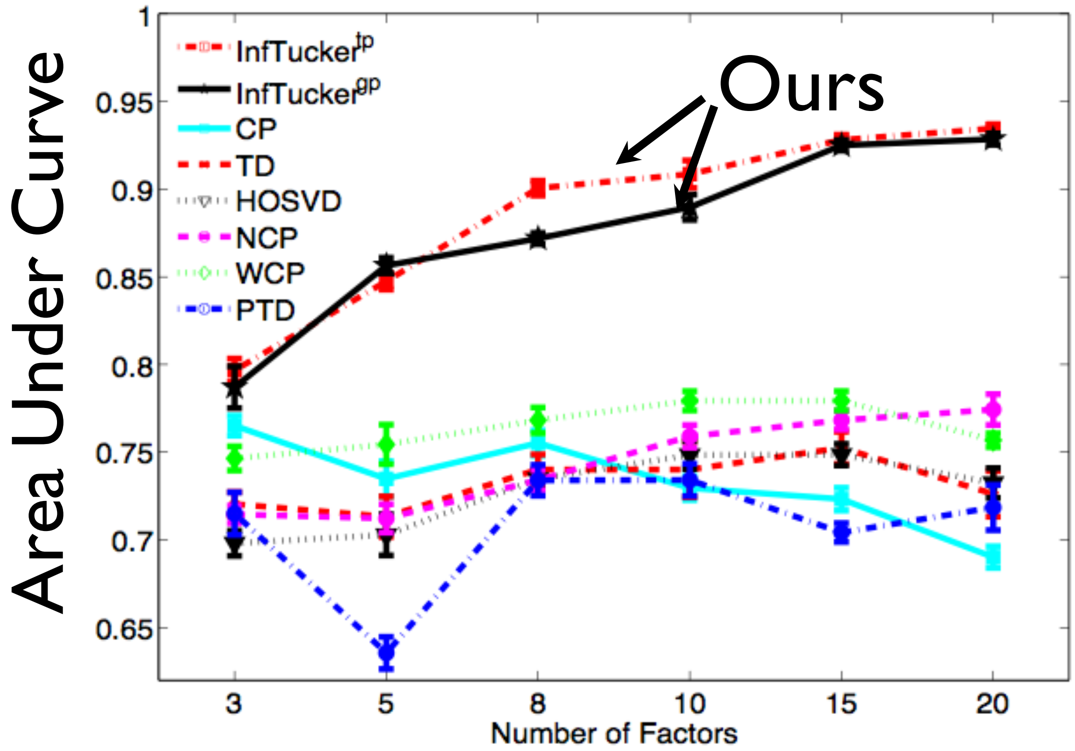
Comparison methods

- CANDECOMP/PARAFAC (CP)
- Tucker decomposition (TD)
- None-Negative CP (NCP)
- High Order Singular Value Decomposition (HOSVD)
- Weighted CP (WCP)

Implemented by the tensor matlab toolbox
http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/

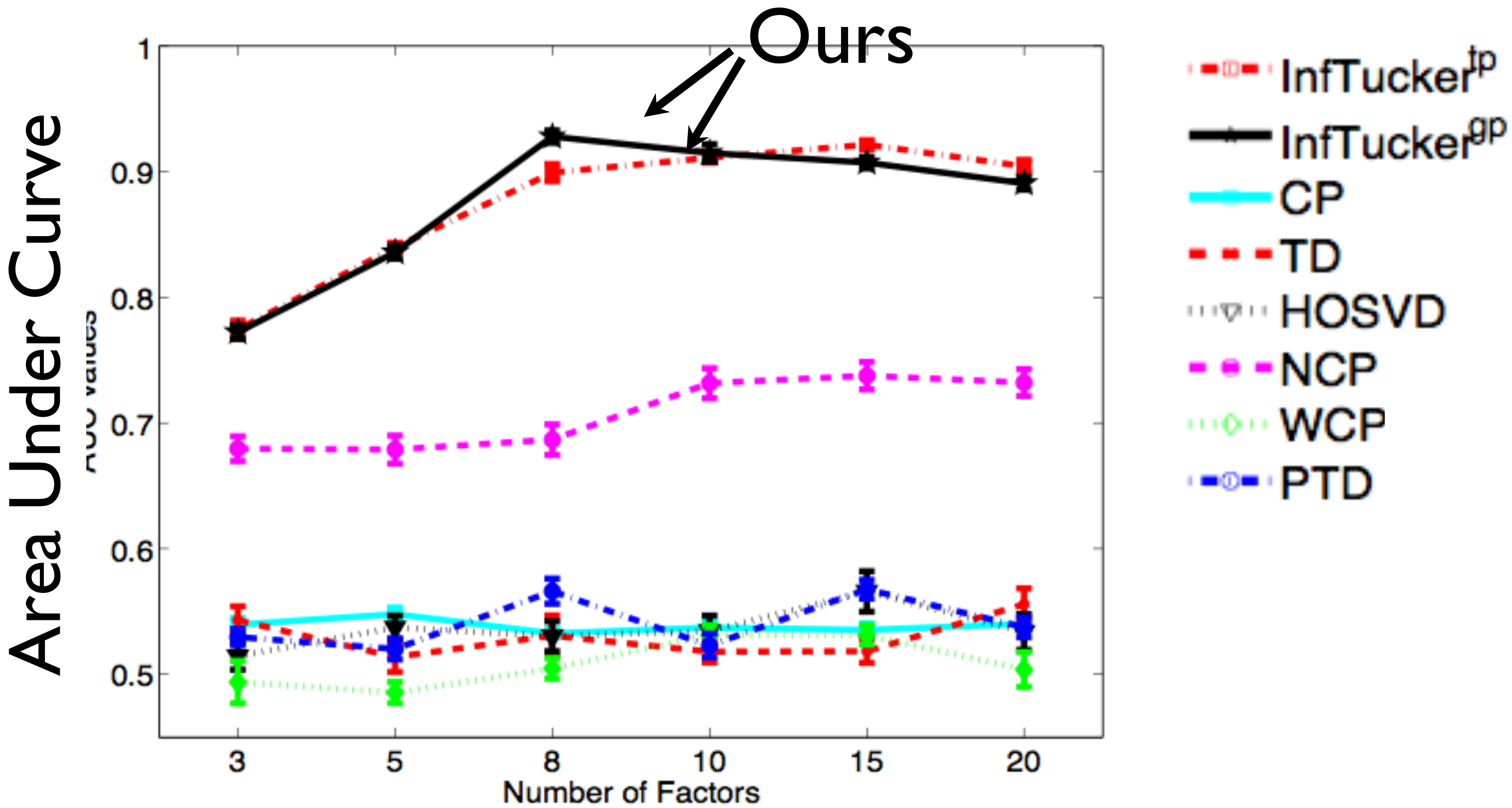
Enron emails (3D)

- Enron dataset: emails from senior management of Enron before its bankruptcy in 2001.
- 3D tensor representation: Sender-Recipient-Subject



Xu, Yan & Qi, ICML 2012, TPAMI 2015

Digg (4D)



- Digg dataset: Social news from digg.com
- 4D tensor representation: user-news-keywords-category

Outline

- Motivation
- Multilinear tensor decomposition
- Nonparametric nonlinear tensor models
- Large scale models

电子科技大学
University of Electronic Science and Technology of China

Limitations of InfTucker

Tensor-variate Gaussian Process

$$p(\mathcal{M}|\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(K)}) = \mathcal{N}(\mathrm{vec}(\mathcal{M}); \mathbf{0}, \Sigma^{(1)} \otimes \ldots \otimes \Sigma^{(K)})$$

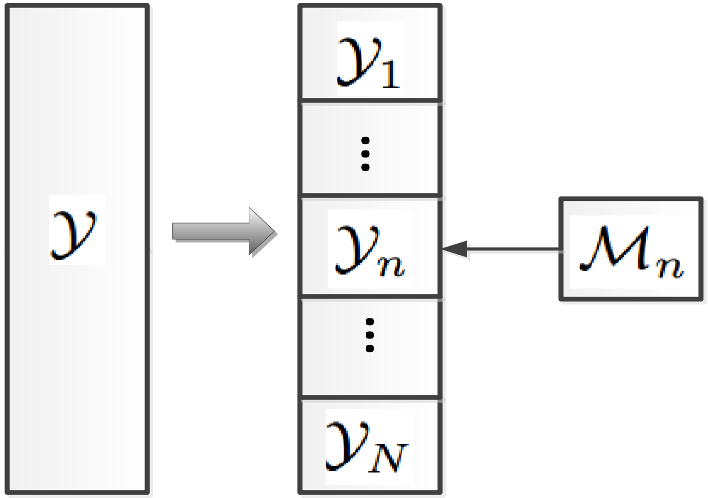
- Zero and nonzero imbalance
- Needs improvement in discovering latent cluster structures
- Scalability issue (expensive computation on the Kronecker-product of the covariance matrices)

Solutions

- Latent cluster structure
  - Dirichlet Process Mixture(DPM) prior (Zhe, Xu, Zhu&Qi, AISTATS 2015)
- Scalablility
  - Online learning (Zhe, Xu, Zhu& Qi, AISTATS 2015)
  - Distributed Computing (Zhe et al., Submitted to AAAI 2016)
- Sparsity of tensor data (imbalanced non-zero entries)
  - Random function (Zhe et al., Submitted to AISTATS 2016)

Local GP

- Slice array $\mathcal{Y}$ into many subarrays $\{\mathcal{Y}_1, \ldots, \mathcal{Y}_N\}$



- Rewrite the latent GP as：

$$p(\mathcal{Y}_n, \mathcal{M}_n | \mathcal{U}) = p(\mathcal{M}_n | \mathcal{U}_n) p(\mathcal{Y}_n | \mathcal{M}_n)$$
$$= \mathcal{N}(\mathrm{vec}(\mathcal{M}_n); \mathbf{0}, \Sigma_n^{(1)} \otimes \ldots \otimes \Sigma_n^{(K)})) p(\mathcal{Y}_n | \mathcal{M}_n)$$

Dirichlet Process Mixtures Priors

- Assign DPM prior over latent factors to capture an undetermined number of latent clusters

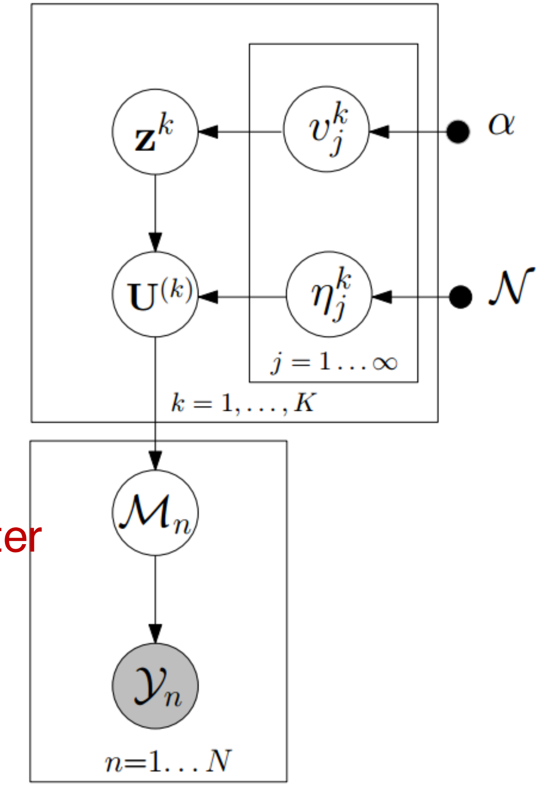
$$p(\mathbf{u}_t^k, z_t^k | \mathbf{v}^k, \boldsymbol{\eta}^k) = p(z_t^k | \mathbf{v}^k) p(\mathbf{u}_t^k | z_t^k, \boldsymbol{\eta}^k)$$

$$= \underbrace{\prod_{j=1}^{\infty} (\pi_j(\mathbf{v}_k))^{\mathbb{1}(z_t^k = j)}}_{\text{Cluster assignments}} \cdot \underbrace{\mathcal{N}(\mathbf{u}_t^k | \boldsymbol{\eta}_{z_t^k}^k, \lambda_k \mathbf{I})}_{\text{Closeness to cluster center}}$$

Stick-breaking construction of DPM

$$p(\mathbf{v}^k | \alpha) = \prod_{j=1}^{\infty} \mathrm{Beta}(v_j^k | 1, \alpha), \quad p(\boldsymbol{\eta}^k) = \prod_{j=1}^{\infty} \mathcal{N}(\boldsymbol{\eta}_j^k | \mathbf{0}, \mathbf{I})$$

Where $\mathbf{v}^k = \{v_1^k, v_2^k, \ldots\}$ $\quad \boldsymbol{\eta}^k = \{\boldsymbol{\eta}_1^k, \boldsymbol{\eta}_2^k, \ldots\}$

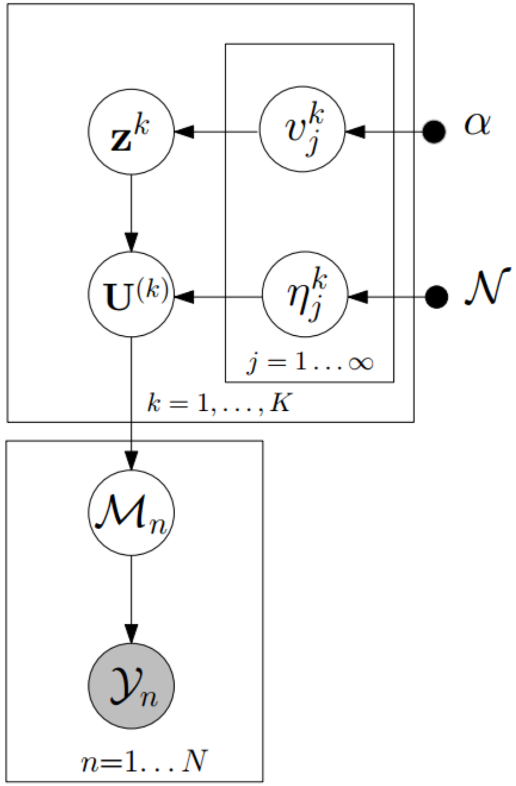


Inftucker with Dirichlet Process Mixtures Priors

- The joint probability of DPM model

$$
\begin{aligned}
&p(\mathcal{U}, \{\mathbf{z}^k, \mathbf{v}^k, \boldsymbol{\eta}^k\}_{k=1}^K, \{\mathcal{M}_n, \mathcal{Y}_n\}_{n=1}^N) \\
&= \prod_{k=1}^{K} p(\mathbf{v}^k|\alpha)p(\boldsymbol{\eta}^k) \prod_{t=1}^{m_k} p(z_t^k|\mathbf{v}^k)p(\mathbf{u}_t^k|z_t^k, \boldsymbol{\eta}^k) \\
&\cdot \prod_{n=1}^{N} p(\mathcal{M}_n|\mathbf{U}_n^{(1)}, \ldots, \mathbf{U}_n^{(K)})p(\mathcal{Y}_n|\mathcal{M}_n).
\end{aligned}
$$

Local sub-tensors

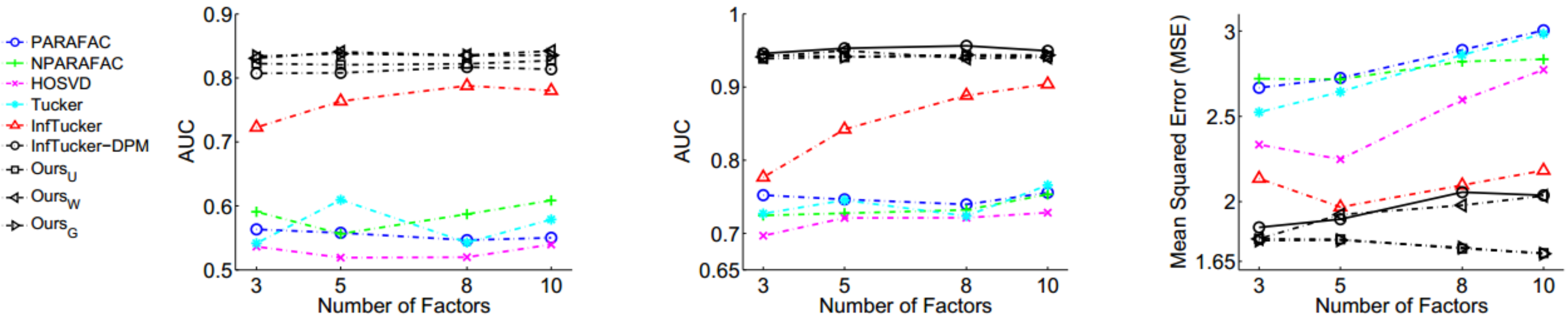


Where $\mathbf{v}^k = \{v_1^k, v_2^k, \ldots\}$ $\boldsymbol{\eta}^k = \{\boldsymbol{\eta}_1^k, \boldsymbol{\eta}_2^k, \ldots\}$

Sample a sub-tensors and conduct Inftucker

Evaluation(1)

- Missing value prediction
  - Binary datasets: Digg(581*124*48) and Enron(203* 203*200)
  - One continuous dataset: Alog(200*100*200)



(a) *digg* (b) *enron* (c) *alog*

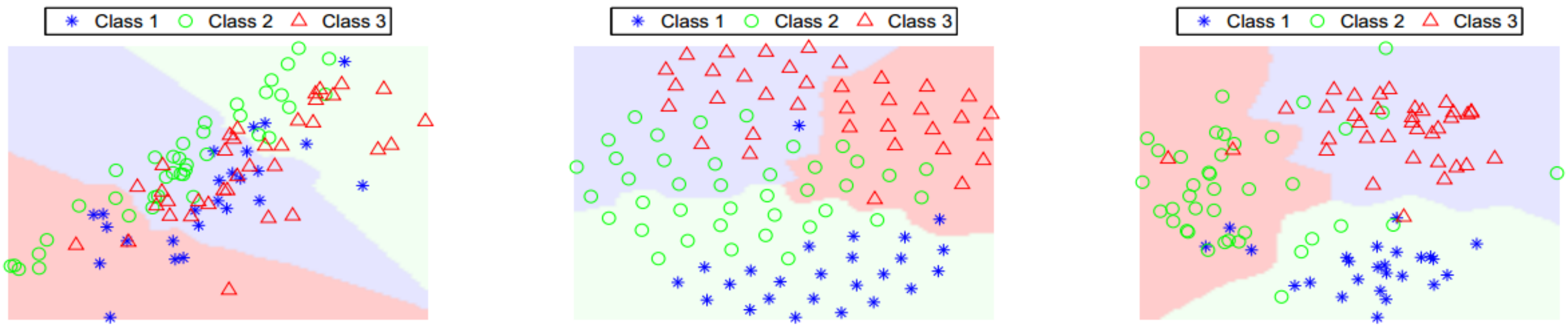
Ours_u, Ours_w, and Ours_G refer to our method based on the uniform, weighted, and grid sampling strategies, respectively

Enron(*www.cs.cmu.edu/~./enron/)*
Digg(*digg.com)*

电子科技大学
University of Electronic Science and Technology of China

Evaluation(2)

- Latent cluster discovery
  - Test on synthetic tensor of size(100*100*100)
  - Subarray size (10*10*10)



(a) PARAFAC (b) InfTucker (c) Our model

Estimated cluster structures in Mode 1

电子科技大学
University of Electronic Science and Technology of China

Evaluation(3)

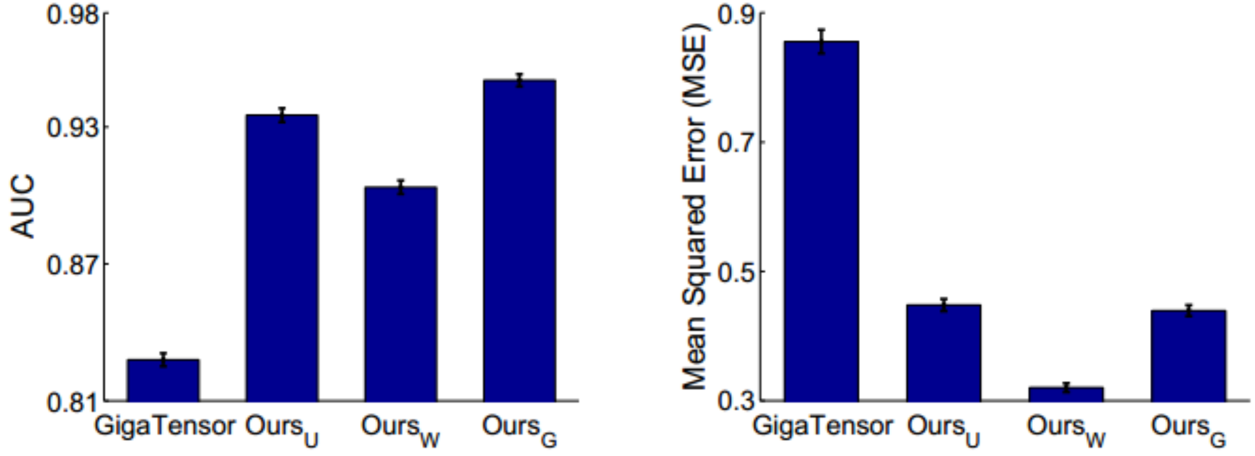
- Latent cluster discovery
  - Test on synthetic tensor of size(100*100*100)
  - Subarray size (10*10*10)

Table 1: The purity of the estimated clusters.

Method	Mode 1	Mode 2	Mode 3
PARAFAC	0.42	0.44	0.42
InfTucker	0.62	0.69	0.75
Our model	**0.84**	**0.84**	**0.88**

Evaluation(4)

- Large multiway array analysis
  - DBLP, of size (10K*200*10K), contain 0.001% nonzero elements
  - ACC, of size (3K*150*30K) ,0.009% nonzero elements

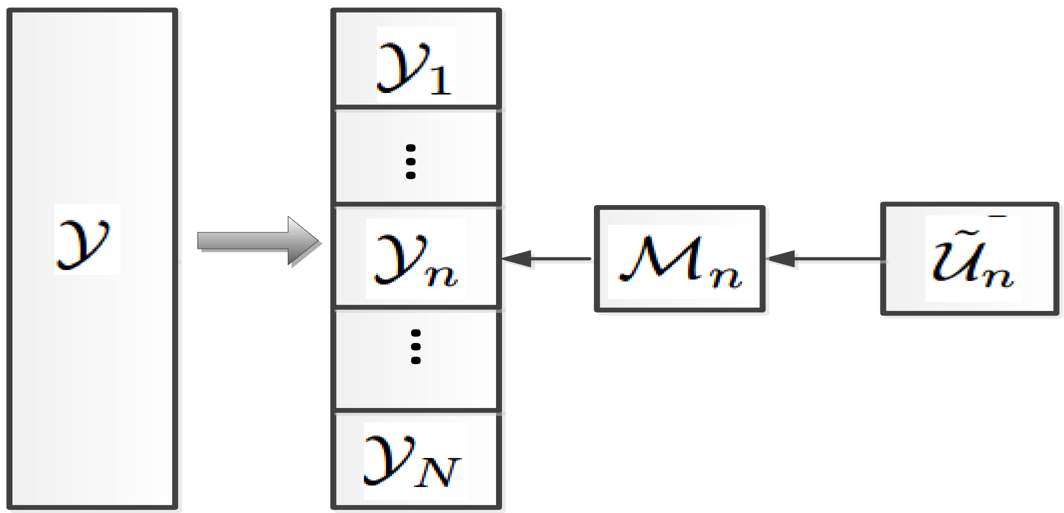


(a) DBLP (b) ACC

DBLP(*(http://dblp.uni-trier.de/xml/)*)

Distributed Bayesian Nonlinear Tensor Decomposition(1)

- Sample $\mathcal{Y}_n$ from tensor-variate GPs based on local latent factors $\tilde{\mathcal{U}}_n = \{\tilde{\mathbf{U}}_n^{(1)}, \ldots, \tilde{\mathbf{U}}_n^{(K)}\}$



$$p(\tilde{\mathcal{U}}_n|\mathcal{U}) = \prod_{k=1}^{K} \mathcal{N}(\mathrm{vec}(\tilde{\mathbf{U}}_n^{(k)})|\mathrm{vec}(\mathbf{U}^{(k)}), \lambda\mathbf{I})$$

Distributed Bayesian Nonlinear Tensor Decomposition(2)

- To stochastic gradient descent to optimize $\{\tilde{\mathcal{U}}_n\}$and $\mathcal{U}$, we slice $\mathcal{Y}_n$ as:

$$\mathcal{Y}_n = \{\mathcal{Y}_{n1}, \ldots, \mathcal{Y}_{nT_n}\}$$

- The joint probability of our model is:

$$p(\mathcal{U}, \{\tilde{\mathcal{U}}_n, \mathcal{M}_n, \mathcal{Y}_n\}_{n=1}^{N}) = \prod_{n=1}^{N} p(\tilde{\mathcal{U}}_n|\mathcal{U}) \prod_{t=1}^{T_n} \boxed{p(\mathcal{M}_{nt}|\tilde{\mathcal{U}}_n)} p(\mathcal{Y}_{nt}|\mathcal{M}_{nt})$$

Only depends on the corresponding subfactors, which can be computed efficiently.

Distributed Bayesian Nonlinear Tensor Decomposition(3)

- Algorithm implemented on HADOOP
  - Estimating the group-specific latent factor $\{\tilde{\mathcal{U}}_n\}$via MAPPER

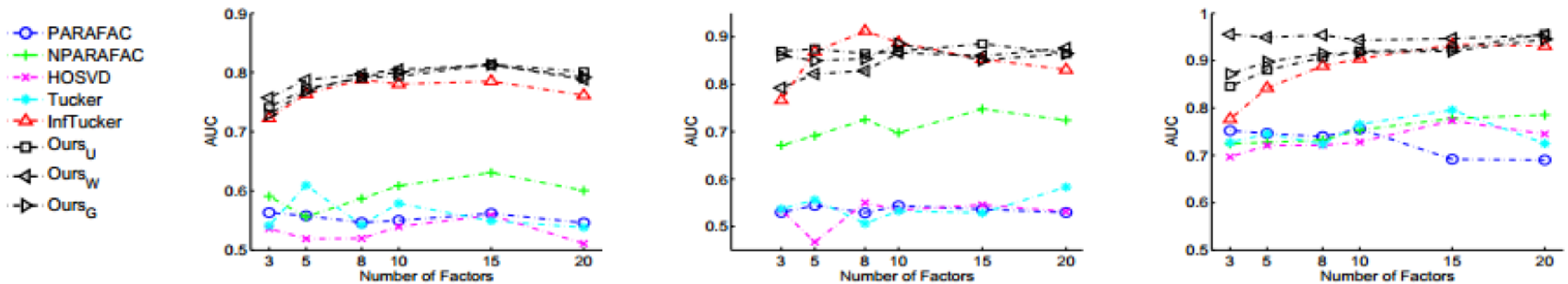
$$f(\tilde{\mathcal{U}}_n) = \log(p(\tilde{\mathcal{U}}_n|\mathcal{U})) + \sum_{t=1}^{T_n} \mathbb{E}_q \left[\log(p(\mathcal{M}_{nt}|\tilde{\mathcal{U}}_n))\right] + \sum_{t=1}^{T_n} \mathbb{E}_q \left[\log(p(\mathcal{Z}_{nt}|\mathcal{M}_{nt}))\right].$$

  - Estimating the latent factor $\mathcal{U}$

$$\mathbf{U}^{(k)} = \frac{1}{N}\tilde{\mathbf{U}}_n^{(k)}.$$

Evaluation(1)

- Missing value prediction
  - Binary datasets: Digg1(581*124*48) , Digg2(22*109*330*30) a
  - Enron(203* 203*200)

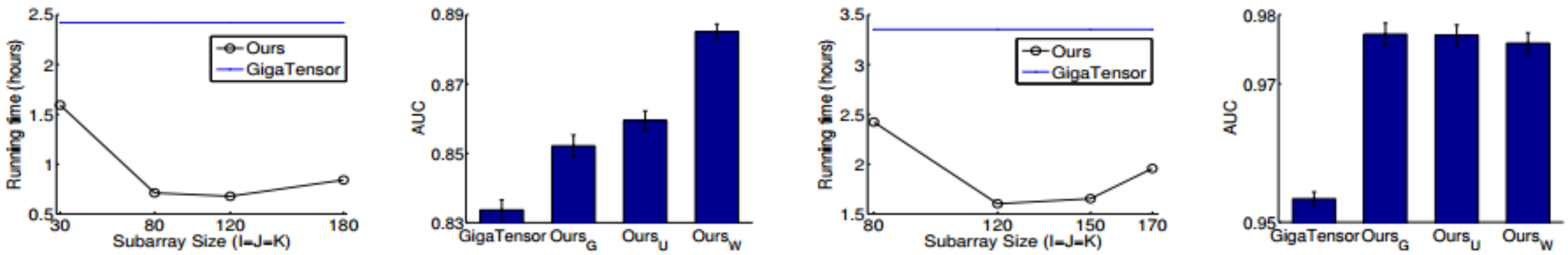


(a) *digg1* (b) *digg2* (c) *enron*

Ours_u, Ours_w, and Ours_G refer to our method based on the uniform, weighted, and grid sampling strategies, respectively

Evaluation(2)

- Running time and prediction accuracy
  - NELL, of size (10K*20K*10K), contain 0.001% nonzero elements
  - ACC, of size (3K*150*30K) ,0.009% nonzero elements



(a) NELL: running time (b) NELL: prediction (c) ACC: running time (d) ACC: prediction

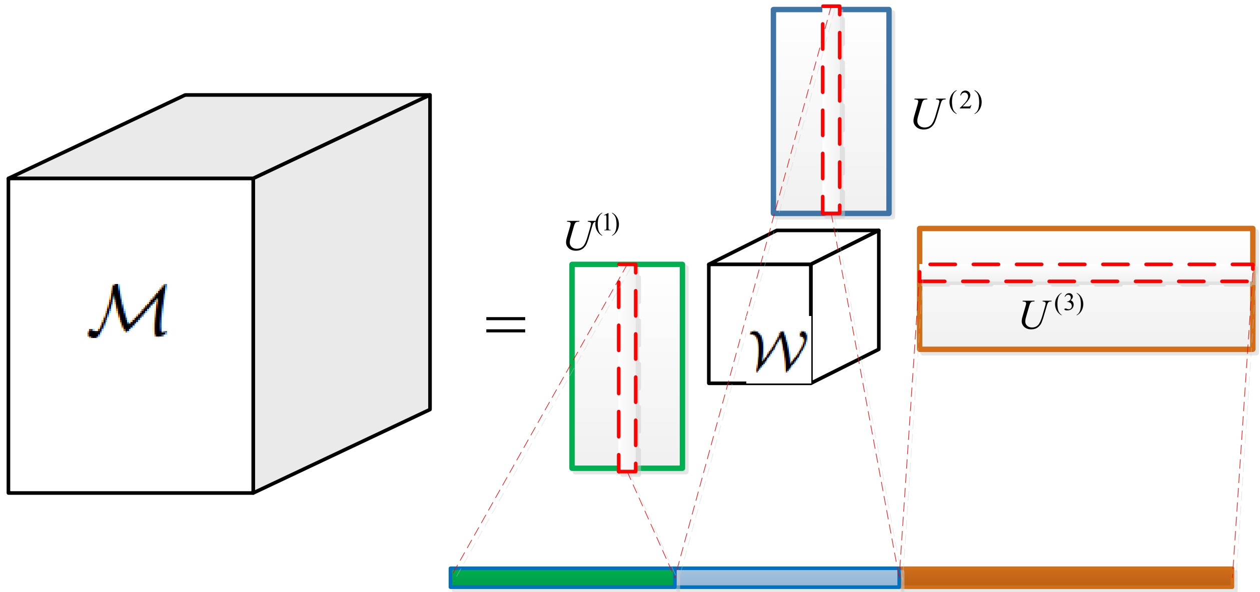
Distributed Flexible Nonlinear Tensor Decomposition(1)

- Sparse Tensor (nonzero imbalance)
  - The Kronecker product between the covariance matrices are calculated over all the modes.
  - Many zero elements are meaningless. Biased prediction will be caused if using them.

Distributed Flexible Nonlinear Tensor Decomposition(2)

- Our model

  - For each tensor entry $m_{\mathbf{i}}$ $(\mathbf{i} = (i_1, \ldots, i_K))$, we construct an input $\mathbf{x}_i$ as following:



$$\mathbf{x_i} = [\mathbf{u}_{i_1}^{(1)}, \ldots, \mathbf{u}_{i_K}^{(K)}] \in \mathbb{R}^{1 \times \sum_{j=1}^{K} m_j}$$

Distributed Flexible Nonlinear Tensor Decomposition(3)

- Our model

  - We assume that there is an underlying function :

$$f : \mathbb{R}^{\sum_{j=1}^{K} m_j} \rightarrow \mathbb{R}$$

$$m_{\mathbf{i}} = f(\mathbf{x}_{\mathbf{i}}) = f([\mathbf{u}_{i_1}^{(1)}, \ldots, \mathbf{u}_{i_K}^{(K)}])$$

  - For any set of tensor entries : $S = \{\mathbf{i}_1, \ldots, \mathbf{i}_N\}$

the $\mathbf{f}_S = \{f(\mathbf{x}_{\mathbf{i}_1}), \ldots, f(\mathbf{x}_{\mathbf{i}_N})\}$ are distributed according to a multivariate Gaussian distribution with mean 0 and covariance decided by :

$$p(\mathbf{f}_S|\mathcal{U}) = \mathcal{N}(\mathbf{f}_S|\mathbf{0}, k(\mathbf{X}_S, \mathbf{X}_S))$$

Where $k(\mathbf{x}_{\mathbf{i}}, \mathbf{x}_{\mathbf{j}}) = k([\mathbf{u}_{i_1}^{(1)}, \ldots, \mathbf{u}_{i_K}^{(K)}], [\mathbf{u}_{j_1}^{(1)}, \ldots, \mathbf{u}_{j_K}^{(K)}])$

$$\mathbf{X}_S = \{\mathbf{x}_{\mathbf{i}_1}, \ldots, \mathbf{x}_{\mathbf{i}_N}\}$$

University of Electronic Science and Technology of China

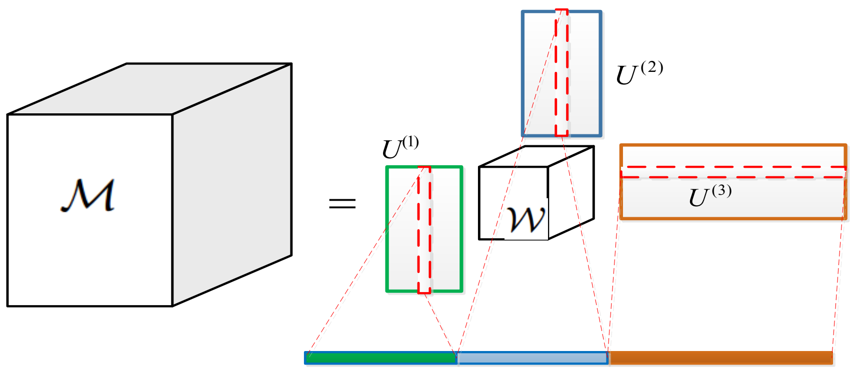
Distributed Flexible Nonlinear Tensor Decomposition(4)

- Our model

  - By assigning a standard normal prior over the latent factors $\mathcal{U}$, we get the joint probability:

$$p(\mathbf{y}, \mathbf{m}, \mathcal{U}) = \prod_{t=1}^{K} \mathcal{N}(\mathrm{vec}(\mathbf{U}^{(t)})|\mathbf{0}, \mathbf{I}) \cdot \mathcal{N}(\mathbf{m}|\mathbf{0}, k(\mathbf{X}_S, \mathbf{X}_S))\mathcal{N}(\mathbf{y}|\mathbf{m}, \beta\mathbf{I})$$

where $S = [\mathbf{i}_1, \ldots, \mathbf{i}_N]$



$$\mathbf{x_i} = [\mathbf{u}_{i_1}^{(1)}, \ldots, \mathbf{u}_{i_K}^{(K)}] \in \mathbb{R}^{1 \times \sum_{j=1}^{K} m_j}$$

Distributed Flexible Nonlinear Tensor Decomposition(5)

- Binary model
  - For binary data, an augmented variables $\mathbf{z} = [z_1, \ldots, z_N]$ has been introduced:

$$p(z_j|m_{\mathbf{i}_j}) = \mathcal{N}(z_j|m_{\mathbf{i}_j}, 1)$$

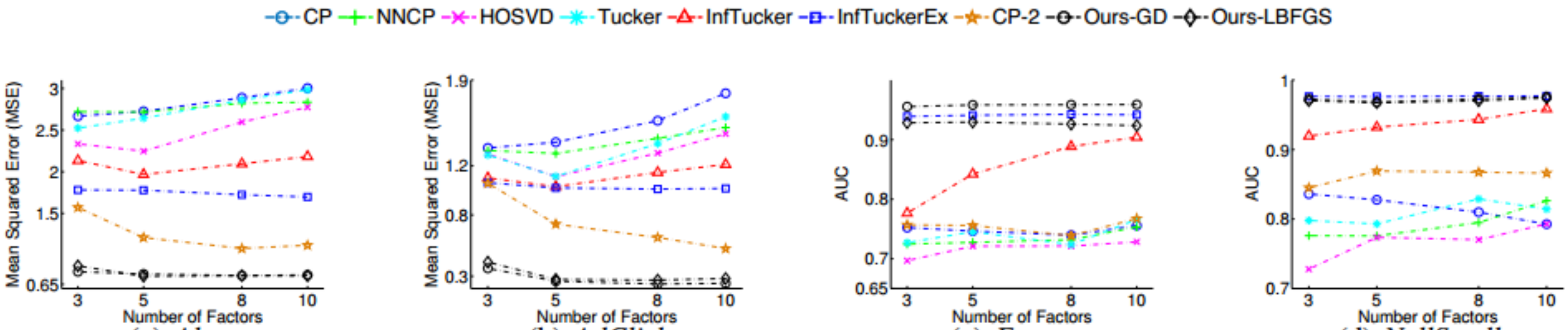
$$p(y_{\mathbf{i}_j}|z_j) = \mathbb{1}(y_{\mathbf{i}_j} = 0)\mathbb{1}(z_j \le 0) + \mathbb{1}(y_{\mathbf{i}_j} = 1)\mathbb{1}(z_j > 0)$$

- Then the joint model for binary data is :

$$p(\mathbf{y}, \mathbf{z}, \mathbf{m}, \mathcal{U}) = \prod_{t=1}^{K} \mathcal{N}(\mathrm{vec}(\mathbf{U}^{(t)})|\mathbf{0}, \mathbf{I})$$
$$\cdot \mathcal{N}(\mathbf{m}|\mathbf{0}, k(\mathbf{X}_S, \mathbf{X}_S))\mathcal{N}(\mathbf{z}|\mathbf{m}, \mathbf{I})$$
$$\cdot \prod_j \mathbb{1}(y_{\mathbf{i}_j} = 0)\mathbb{1}(z_j \le 0) + \mathbb{1}(y_{\mathbf{i}_j} = 1)\mathbb{1}(z_j > 0)$$

Evaluation(1)

- Missing value prediction
  - Binary datasets: and Enron(203* 203*200), NellSmall(295* 170*94)
  - Continuous dataset: AdClick(80*100*100), Alog(200*100*200)



(a) *Alog* (b) *AdClick* (c) *Enron* (d) *NellSmall*

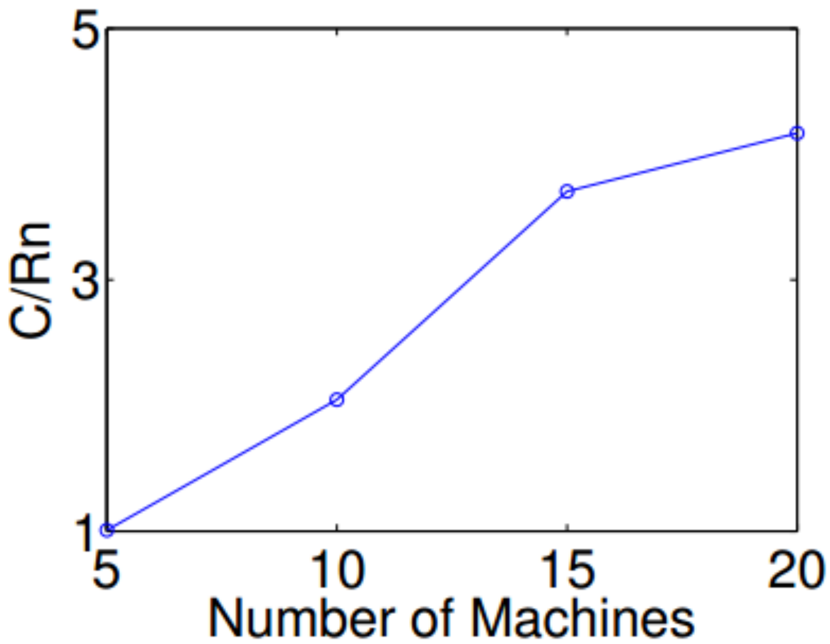
Mean Square Error (The lower the better)

Area Under Curve (The higher the better)

电子科技大学
University of Electronic Science and Technology of China

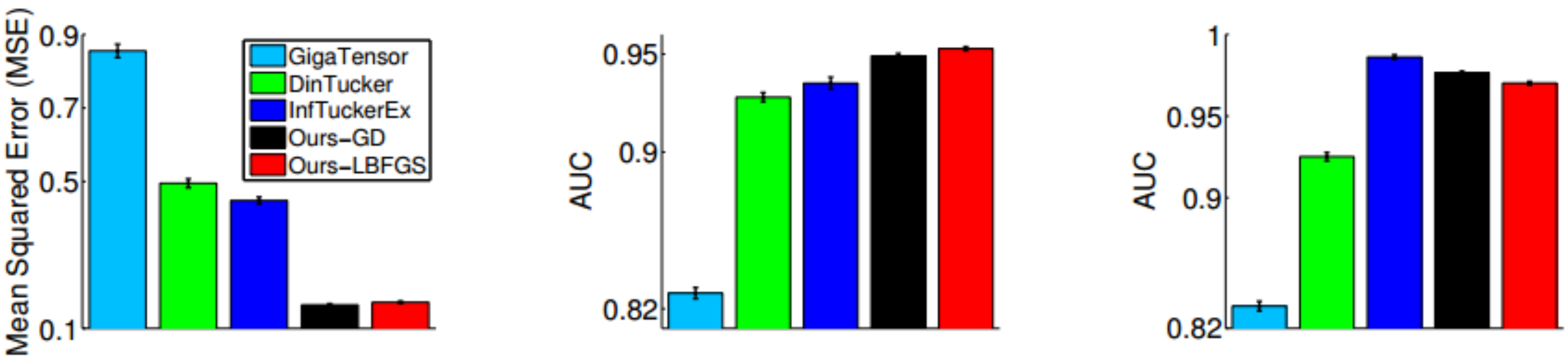
Evaluation(2)

- Scalability with regard to the Number of Machines
  - DBLP, of size (10K*20K*10K), contain 0.001% nonzero elements
  - ACC, of size (3K*150*30K) , 0.009% nonzero elements



Evaluation(3)

- Large multiway array analysis
  - DBLP, size (10K*200*10K), 0.001% nonzero elements
  - ACC, size (3K*150*30K) , 0.009% nonzero elements
  - NELL, size (20K*12.3K*280), 0.0001% nonzero elements



(a) ACC (b) DBLP (c) NELL

Mean Square Error (The lower the better)

Area Under Curve (The higher the better)

电子科技大学
University of Electronic Science and Technology of China

Evaluation(4)

- Click-Through-Rate Prediction
  - Online ads click log from a major Internet company--four mode tensor (user, advertisement, publisher, page-section).
  - Size of the extracted tensors for the three days: 179K*81K*35*355, 167K*78K*35*354, 213K*82K *37*354

Method	1-2	2-3	3-4
Logistic regression	0.7360	0.7337	0.7538
Linear SVM	0.7414	0.7332	0.7540
Our model	**0.8907**	**0.8897**	**0.9036**

CTR prediction accuracy on the first three days of May 2015. "1-2" means using May 1st's data for training and May 2nd's data for testing; similar are "2-3" and "3-4".

Summary

	Methods	Priors/Constraints
Latent factors $\mathbf{Y} \approx \mathbf{U}\mathbf{V}$ Loading matrix	SVD	Minimize MSE
	NMF	$u_{ij} \geq 0 \quad \& \quad v_{ij} \geq 0$
	PPCA	Gaussian priors on $\mathbf{U} \ \& \ \mathbf{V}$
	Sparse PPCA	Laplace prior on $\mathbf{V}$
	GP-LVM	Marginalization of $\phi(\mathbf{U})$ with Gaussian prior
Interaction matrix $\mathbf{Y} \approx \mathbf{U}\mathbf{W}\mathbf{V}$ Membership matrix	Latent Eigen Model	Gaussian priors on $\mathbf{U} \ \& \ \mathbf{W}$
	Infinite Relational Model	CRPs on $\mathbf{U} \ \& \ \mathbf{V}$, Bern on $\mathbf{W}$
	MMSB	LDAs on $\mathbf{U} \ \& \ \mathbf{V}$
	SMGB	GPs on $\mathbf{X} = \mathbf{U}\mathbf{W}\mathbf{V}$
Interaction tensor $\mathbf{Y} \approx \mathbf{W} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$ Membership matrix	Tucker	No constraints
	CP	Diagonal matrix $\mathbf{W}$
	Nonnegative CP	non-negativity constraints
	InfTucker (GP on tensors)	GPs on $\mathbf{X} = \mathbf{W} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$

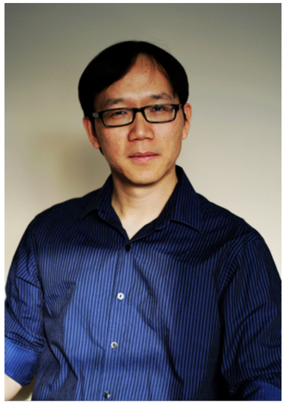
	Methods	Priors/Constraints
$\mathbf{Y} \approx \mathbf{UV}$	SVD	Minimize MSE
	NMF	$u_{ij} \geq 0 \quad \& \quad v_{ij} \geq 0$
	PPCA	Gaussian priors on $\mathbf{U} \;\&\; \mathbf{V}$
	Sparse PPCA	Laplace prior on $\mathbf{V}$
	GP-LVM	Marginalization of $\phi(\mathbf{U})$ with Gaussian prior
$\mathbf{Y} \approx \mathbf{UWV}$	Latent Eigen Model	Gaussian priors on $\mathbf{U} \;\&\; \mathbf{W}$
	Infinite Relational Model	CRPs on $\mathbf{U} \;\&\; \mathbf{V}$, Bern on $\mathbf{W}$
	MMSB	LDAs on $\mathbf{U} \;\&\; \mathbf{V}$
	SMGB	GPs on $\mathbf{X} = \mathbf{UWV}$
$\mathbf{Y} \approx \mathbf{W} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$	Tucker	No priors
	CP	Diagonal matrix $\mathbf{W}$
	Nonnegative CP	non-negativity constraints
	InfTucker (GP on tensors)	GPs on $\mathbf{X} = \mathbf{W} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$

Probabilistic matrix and tensor models

- Pros:
  - High prediction accuracy
  - Sparsity: easy to interpret
  - Model selection or non-parametric Bayes: e.g., learn the number of latent groups and hyperparameters
  - Handling missing data
  - Various noise types: continuous, binary, counts
- Improve the scalability:
  - Potentially high computational cost: require clever inference algorithms
  - Parallel computing or online learning

Acknowledgements

My co-authors: Alan(Yuan) Qi, Shandian Zhe, Feng Yan



My fellow colleagues/students of SMILE Lab (Statistical Machine Intelligence & LEarning Lab) @ UESTC

- 成员：教师： 4名
  博士/硕士研究生：14名
- [网址： http://smilelab.uestc.edu.cn/index.php?title=HOME](http://smilelab.uestc.edu.cn/index.php?title=HOME)

电子科技大学
University of Electronic Science and Technology of China

Questions?