Nonparametric Latent Variable Models via Kernel Embedding of Distributions

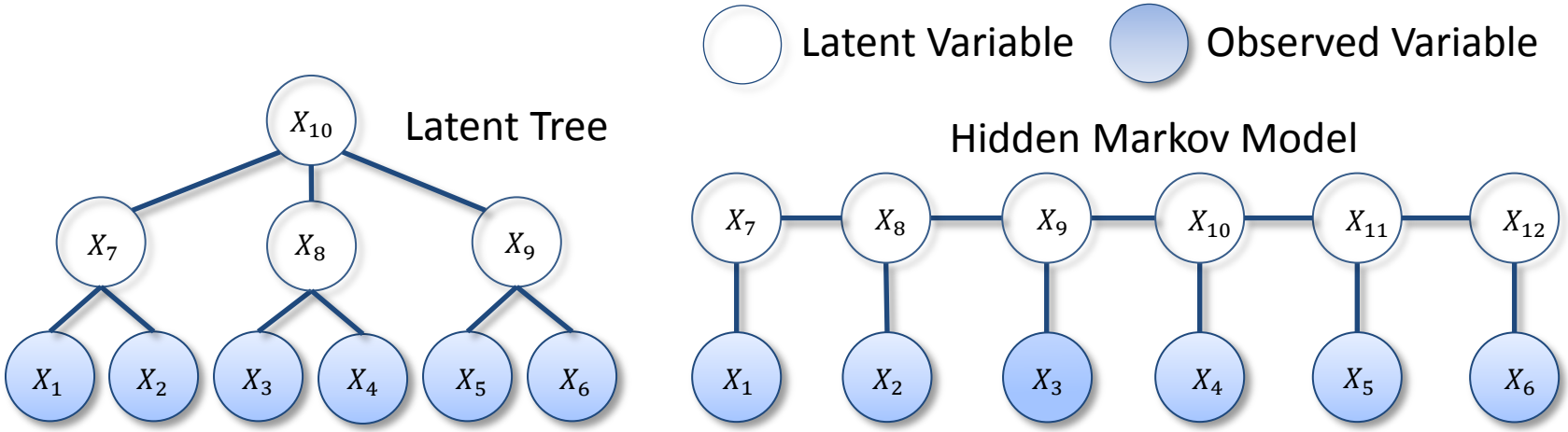
Le Song

College of Computing
Georgia Institute of Technology

Latent Tree Graphical Models

- Graphical model: nodes represent variables, edges represent conditional independence relation

- Latent tree graphical models: latent and observed variables are arranged in a tree structure



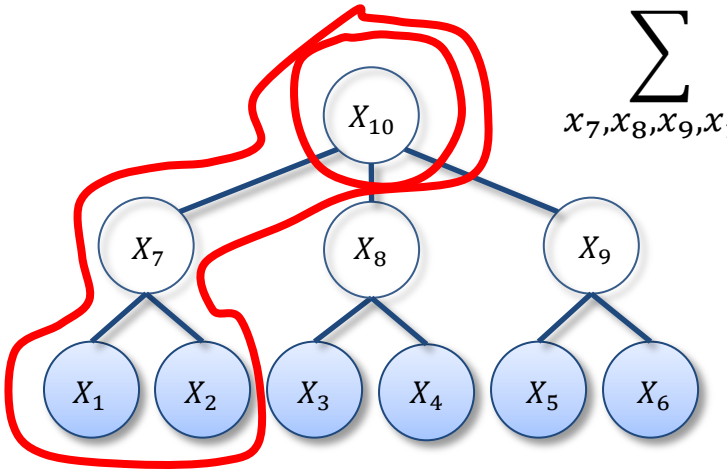
- Many real world applications, eg., time-series prediction, topic modeling

Benefit of Latent Variables

- Exponential number of entries in $P(X_1, X_2, \dots, X_6)$
  - Discrete variable taking n possible values, P has $O(n^6)$ entries!

- Latent tree reduces the number of parameters

$$P(X_1, X_2, \dots, X_6) = \sum_{x_7, x_8, x_9, x_{10}} \sum_{x_7, x_8, x_9, x_{10}}$$



$$P(X_1, X_2, \dots, X_6, x_7, \dots, x_{10})$$

$O(n)$ params

$O(3n^2)$ params

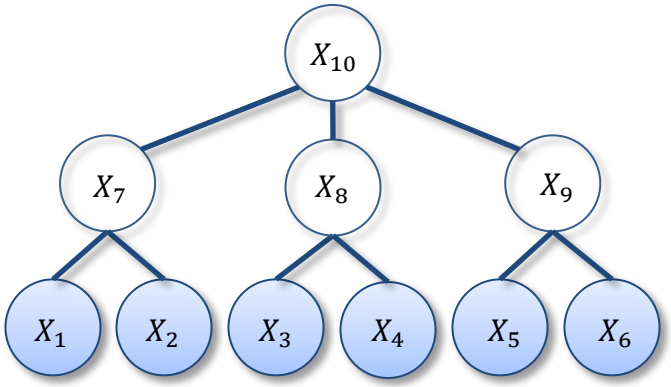
$$P(X_{10})$$
$$P(x_7|x_{10})P(X_1|x_7)P(X_2|x_7)$$
$$P(x_8|x_{10})P(X_3|x_8)P(X_4|x_8)$$
$$P(x_9|x_{10})P(X_5|x_9)P(X_6|x_9)$$

Latent tree has $O(9n^2)$ params

Significant saving!

EM algorithm

- Do not observe latent variables, need to estimate the structure and parameters, eg., $P(X_7|X_{10})$ and $P(X_1|X_7)$



$i = 1$	x_1^1	x_2^1	x_3^1	x_4^1	x_5^1	x_6^1
$\vdots$	$\vdots$					$\vdots$
$i = m$	x_1^m	x_2^m	x_3^m	x_4^m	x_5^m	x_6^m

Spectral algorithm: Estimate structure or parameters in a local-minimum-free fashion

- Expectation maximization: maximize likelihood of observations
  - $\max_\theta \log \prod_{i=1}^m P(x_1^i, \dots, x_6^i)$
- Drawback: local maxima, slow to converge, difficult to analyze

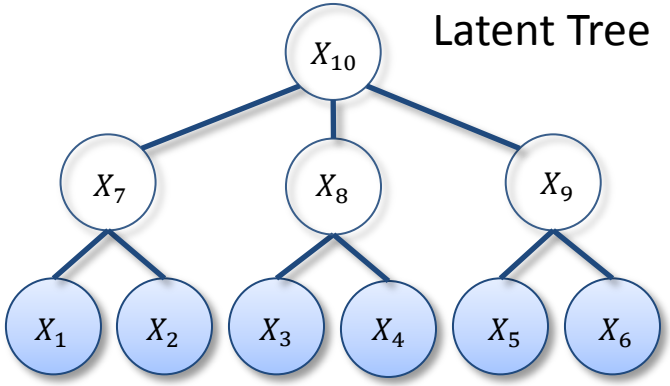
Learning latent tree with spectral algorithms

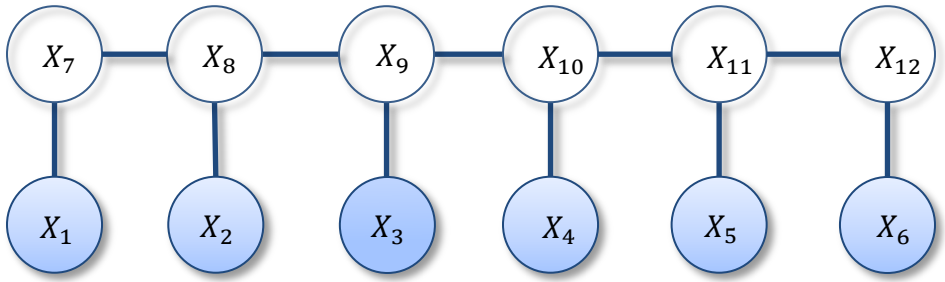
- Estimating marginal distribution of the observed variables
  - Spectral HMMs (Hsu et al. COLT'09, Bailly et al. ICML'09)
  - Spectral latent tree (Parikh et al. ICML'11)
  - Spectral dimensional reduction for HMMs (Foster et al. Arxiv)
  - More recent: Cohen et al. ACL'12, Balle et al. ICML'12
- Estimating latent parameters
  - PCA approach (Mossel & Roch AOAP'06)
  - SVD and tensor approach (Anandkumar et al. …)
- Estimating the structure of latent variable models
  - Recursive grouping (Choi et al. JMLR'11)
  - Spectral short quartet (Anandkumar et al. NIPS'11)

Tensor view of marginal probability

- Marginal probability table $\mathcal{T} = P(X_1, X_2, \dots, X_6)$
  - Discrete variable taking n possible values $\Omega = \{1, \dots, n\}$
    - 6-way table, or 6th order tensor
    - Dimension labeled by the variable
  - Value of the variable is the index to the corresponding dimension, need 6 indexes to access a single entry
    - $P(X_1 = 1, X_2 = 4, \dots, X_6 = 3)$ is the entry $\mathcal{T}[1,4, \dots, 3]$

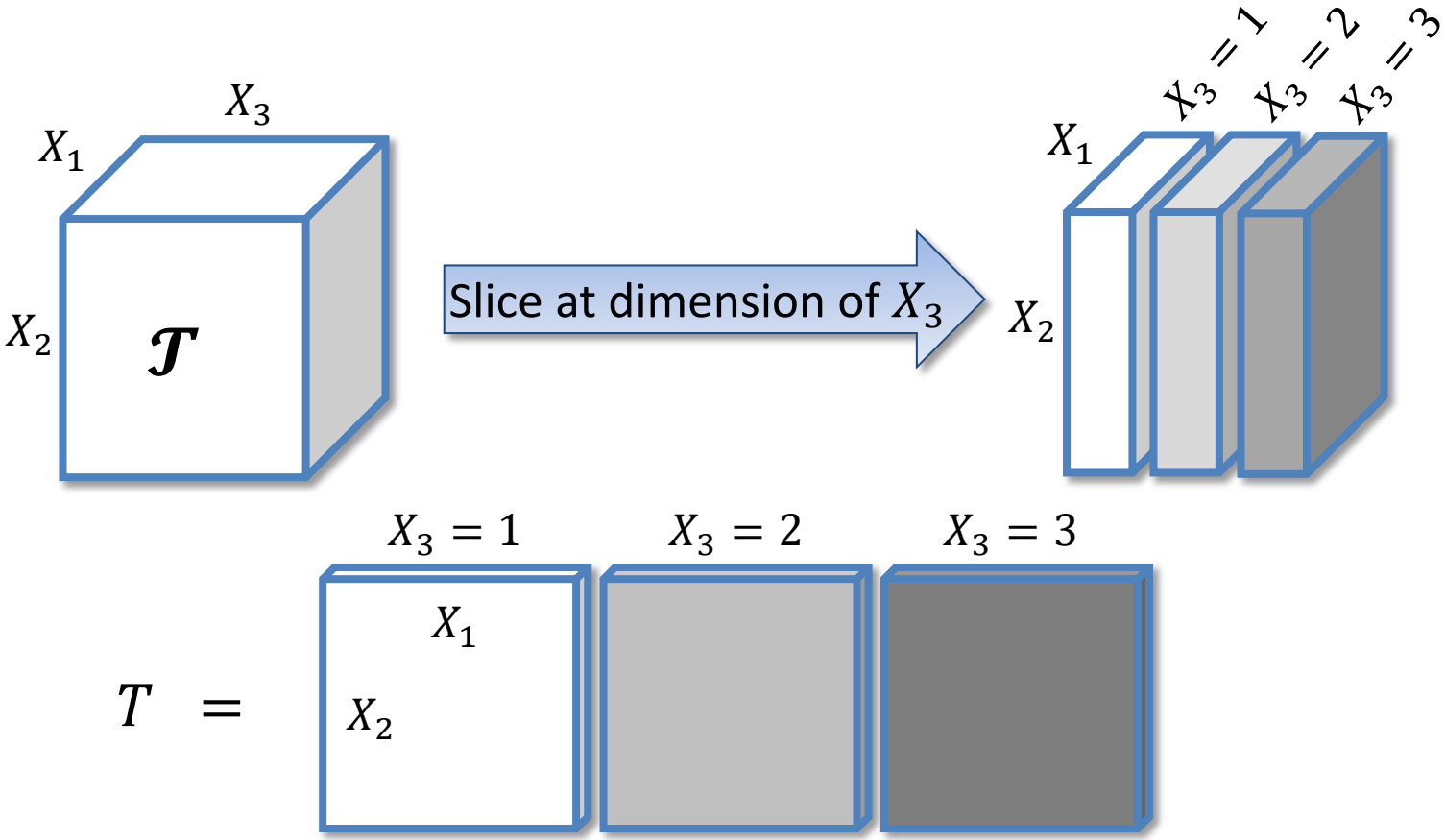
Examples:





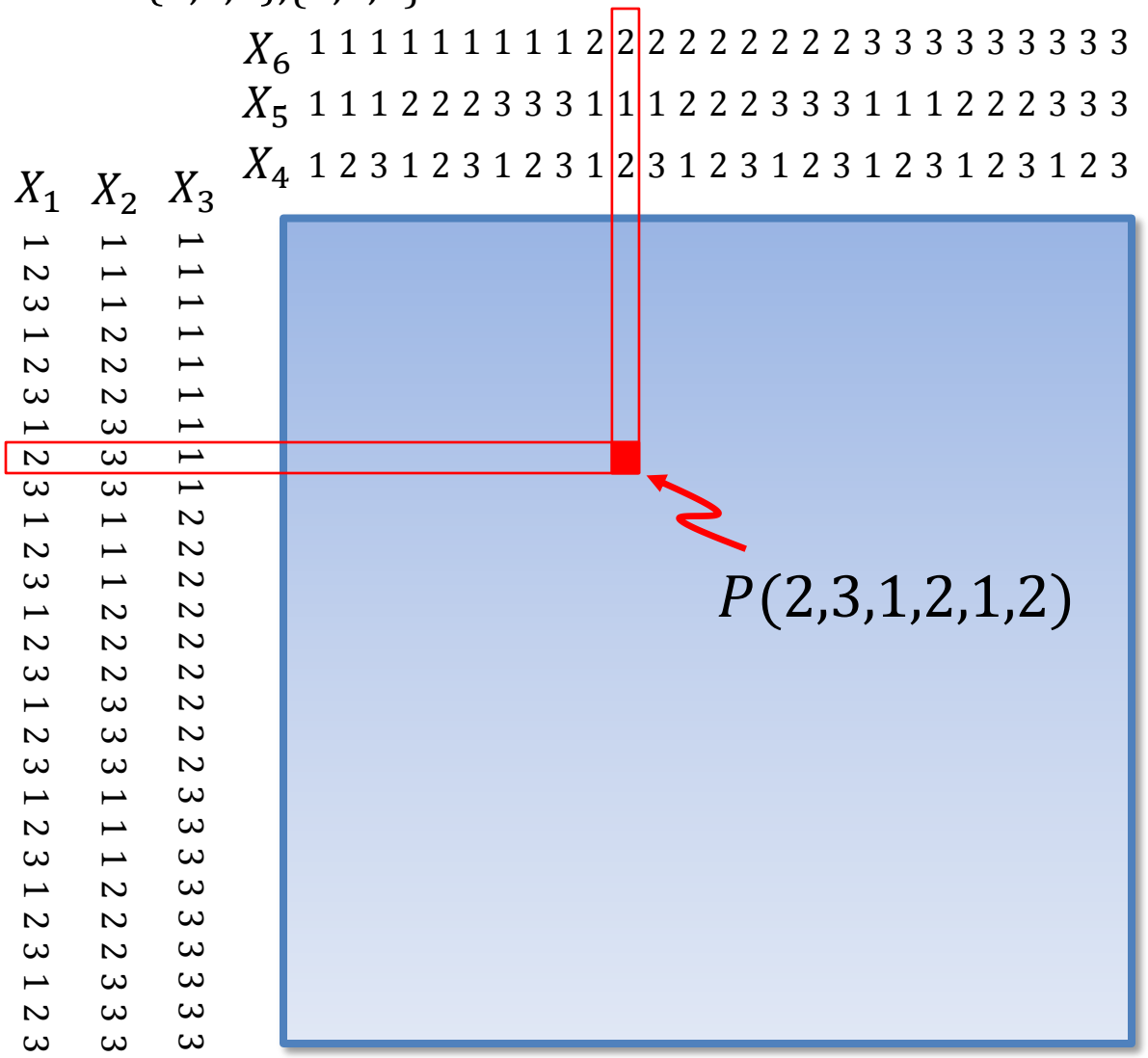
Reshaping tensor into matrices

- $T = Reshape(\mathcal{T}, \mathcal{C})$: multi-index $\mathcal{C}$ mapped into row index, and the remaining indexes into column index
  - Eg. $\mathcal{T} = P(X_1, X_2, X_3)$, a 3rd order tensor and $n = 3$
  - $P_{\{2\};\{1,3\}} = Reshape(\mathcal{T}, \{2\})$ turns the dimension of X_2 into row



Reshaping 6th order tensor

- $T = P_{\{1,2,3\};\{4,5,6\}} = Reshape(P(X_1, \dots, X_6), \{1,2,3\})$

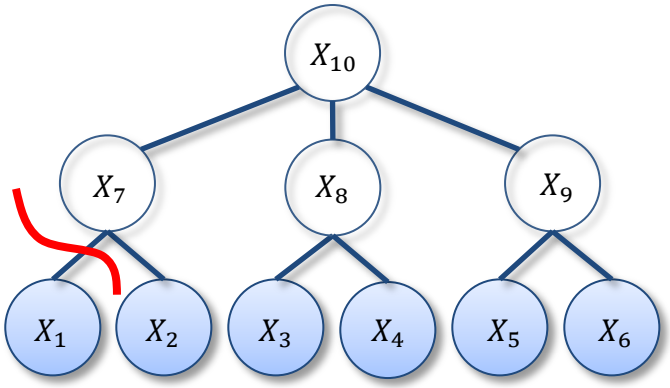


Each entry is the probability of a unique assignment to $X_1, \dots, X_6$

Reshaping according to latent tree structure

- For marginal $\boldsymbol{\mathcal{P}} = P(X_1, X_2, \ldots, X_6)$ of a latent tree model, reshape it according to the edges in the tree

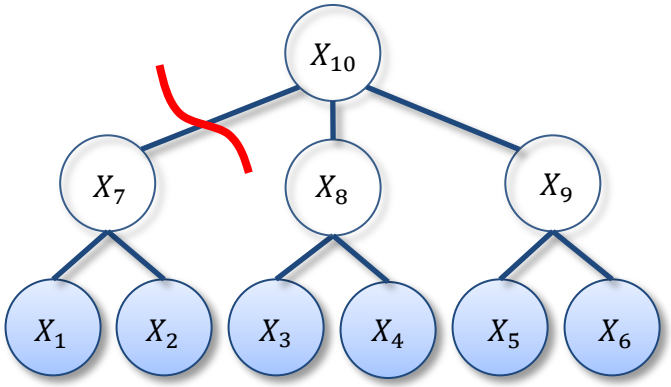
- $P_{\{1\};\{2,3,4,5,6\}} = Reshape(\boldsymbol{\mathcal{P}}, \{1\})$



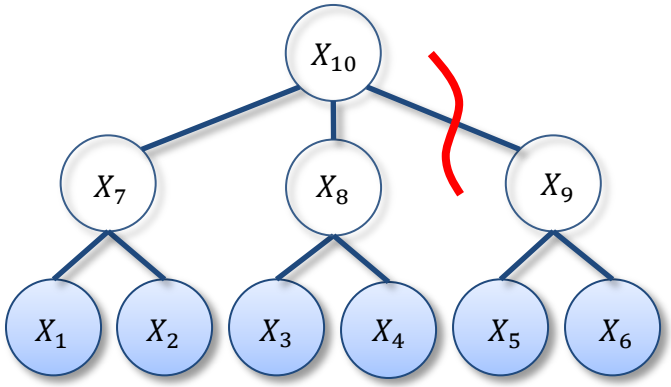
$P_{\{1\};\{2,3,4,5,6\}}$

- $P_{\{1,2\};\{3,4,5,6\}} = Reshape(\boldsymbol{\mathcal{P}}, \{1,2\})$

- $P_{\{1,2,3,4\};\{5,6\}} = Reshape(\boldsymbol{\mathcal{P}}, \{1,2,3,4\})$



$P_{\{1,2\};\{3,4,5,6\}}$



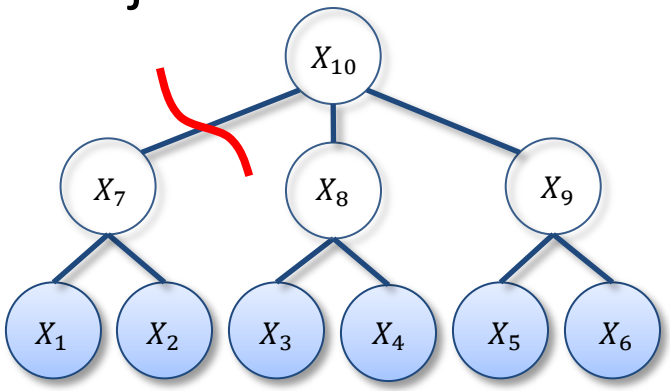
$P_{\{1,2,3,4\};\{5,6\}}$

Low rank structure after reshaping

- Size of $P_{\{1,2\};\{3,4,5,6\}}$ is $n^2 \times n^4$, but its rank is just n

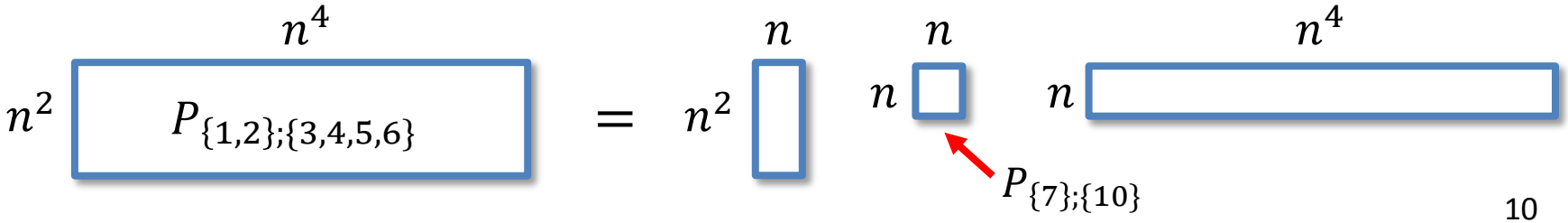
- $P(X_1, X_2, \dots, X_6) =$

$$\sum_{x_7, x_{10}} P(X_1, X_2 | x_7)\ P(x_7, x_{10})\ P(X_3, X_4, X_5, X_6 | x_{10})$$



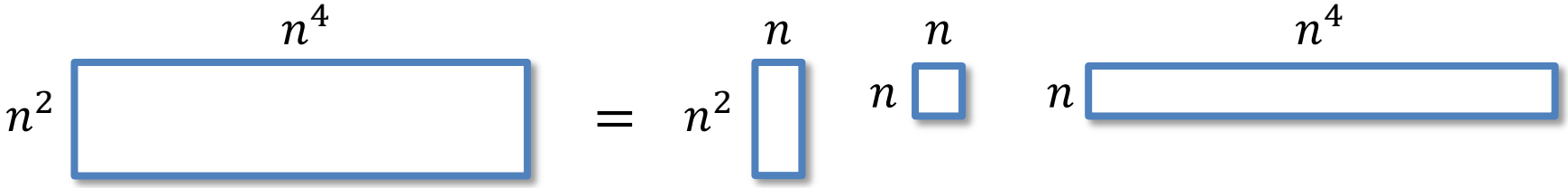
$P_{\{1,2\};\{3,4,5,6\}}$

- Use matrix multiplications to express summation over X_7, X_{10}
  - $P_{\{1,2\};\{3,4,5,6\}} = P_{\{1,2\}|\{7\}}\ P_{\{7\};\{10\}}\ P^{\top}_{\{3,4,5,6\}|\{10\}}$
  - $P_{\{1,2\}|\{7\}} \coloneqq Reshape(P(X_1, X_2 | X_7), \{1,2\})$
  - $P_{\{3,4,5,6\}|\{10\}} \coloneqq Reshape(P(X_3, X_4, X_5, X_6 | X_{10}), \{3,4,5,6\})$

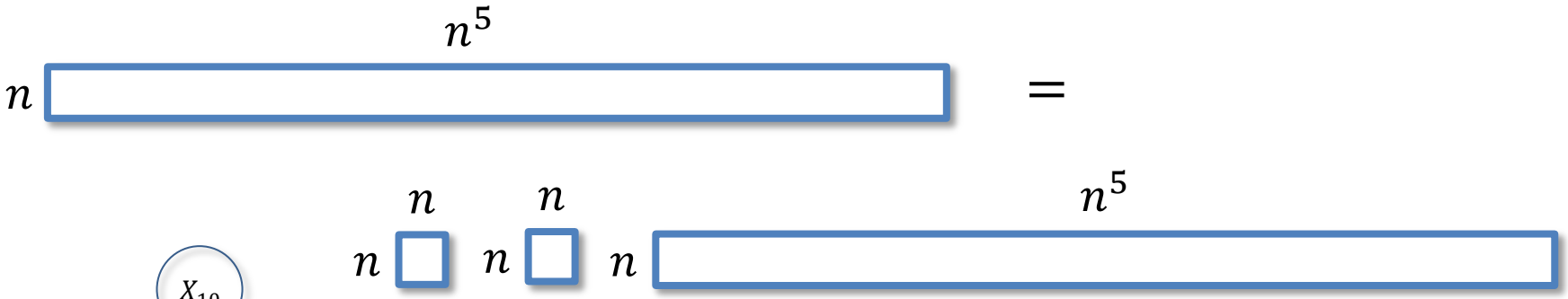


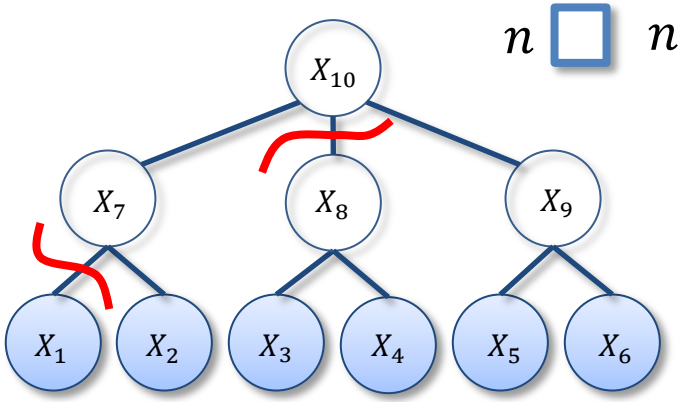
Low rank structure of latent tree models

- $P_{\{3,4\};\{1,2,5,6\}} = P_{\{3,4\}|\{8\}}\, P_{\{8\};\{10\}}\, P^{\top}_{\{1,2,5,6\}|\{10\}}$



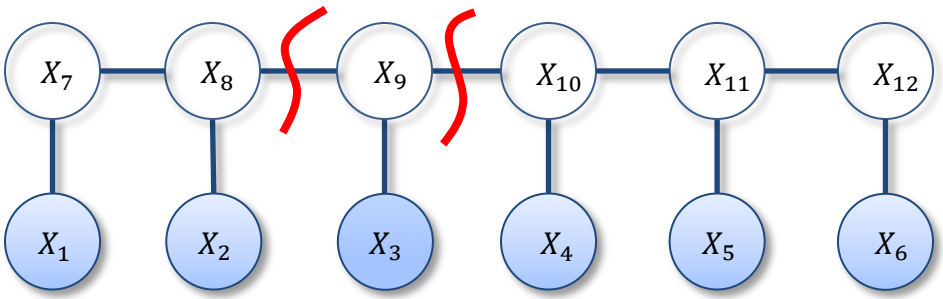
- $P_{\{1\};\{2,3,4,5,6\}} = P_{\{1\}|\{7\}}\, P_{\{7\};\{7\}}\, P^{\top}_{\{2,3,4,5,6\}|\{7\}}$



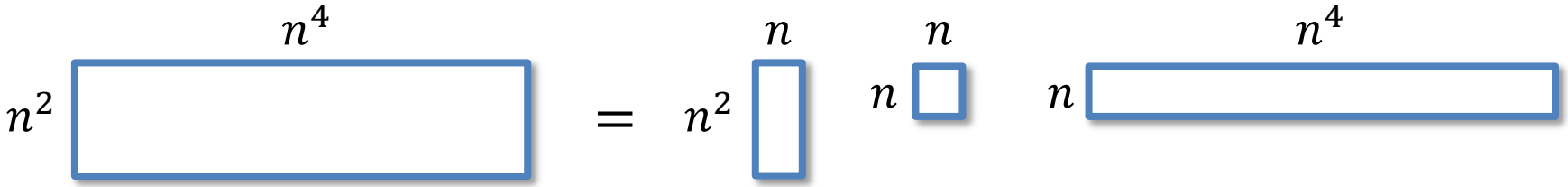


All these reshapings are low rank, and with rank n

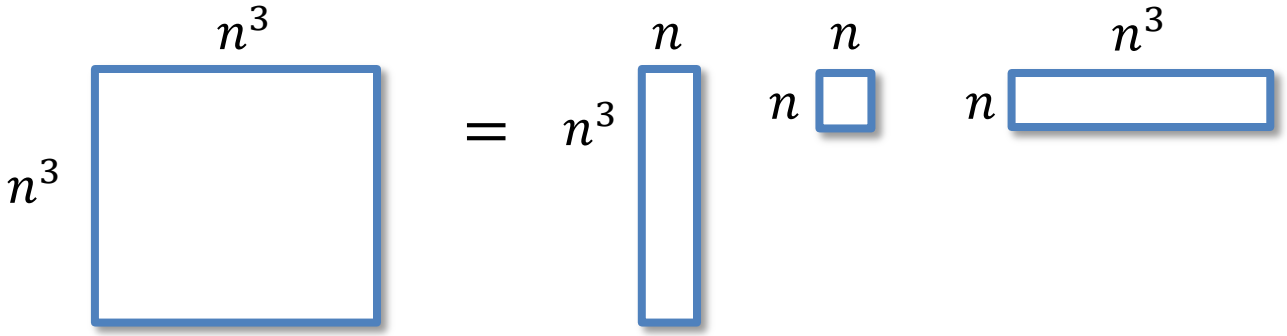
Low rank structure of hidden Markov models



- $P_{\{1,2\};\{3,4,5,6\}} = P_{\{1,2\}|\{8\}}\, P_{\{8\};\{9\}}\, P^{\top}_{\{3,4,5,6\}|\{9\}}$



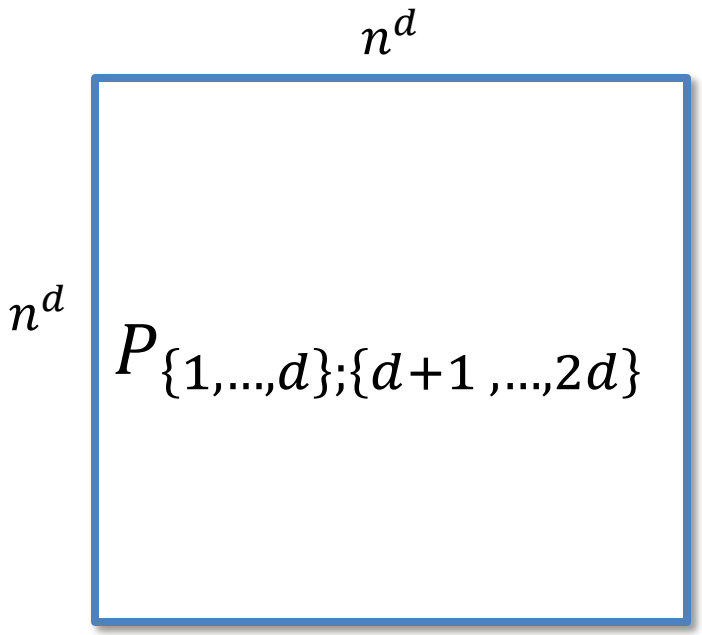
- $P_{\{1,2,3\};\{4,5,6\}} = P_{\{1,2,3\}|\{9\}}\, P_{\{9\};\{10\}}\, P^{\top}_{\{4,5,6\}|\{10\}}$



Key features of spectral algorithms

- Represent joint probability table of observed variables with low rank factorization

- Eg. $P_{\{1,\ldots,d\};\{d+1,\ldots,2d\}} = Reshape(P(X_1,\ldots,X_{2d}),\{1,\ldots,d\})$

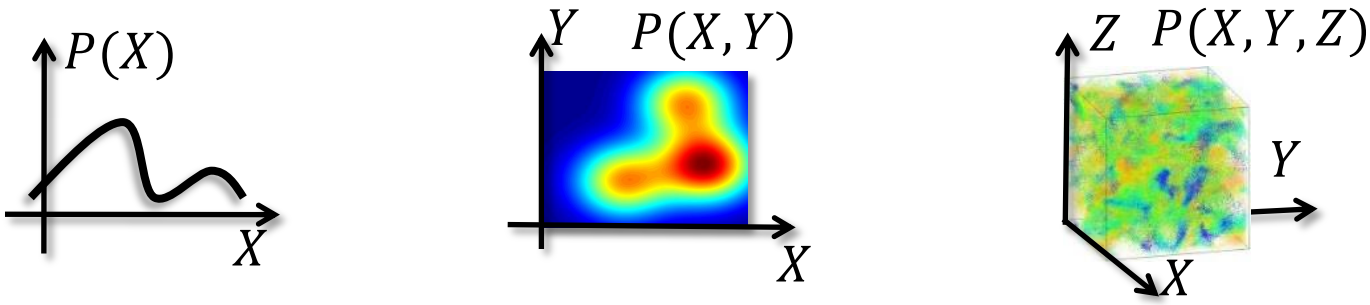


- Represent it by low rank factors to avoid exponential blowup
- Use clever decomposition technique to avoid directly using all entries from the table
- Use spectral decomposition

Non-discrete, non-Gaussian case

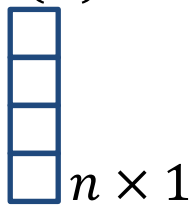
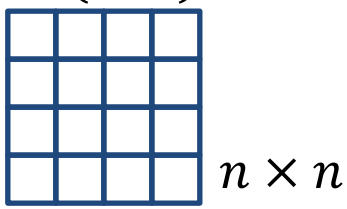
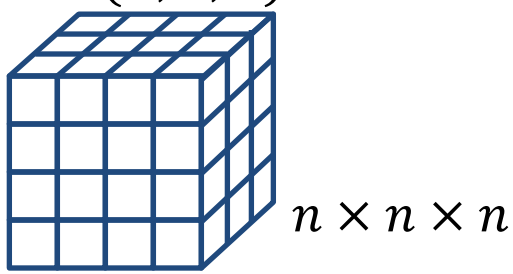
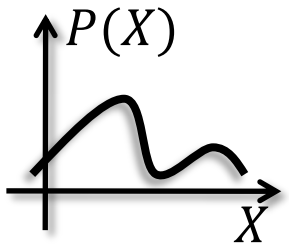
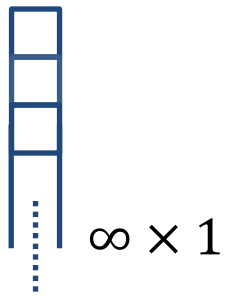
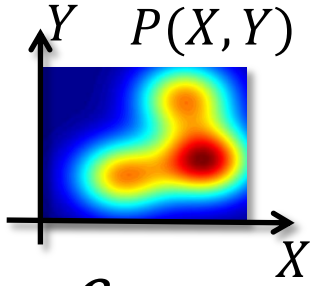
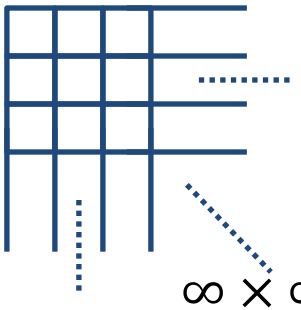
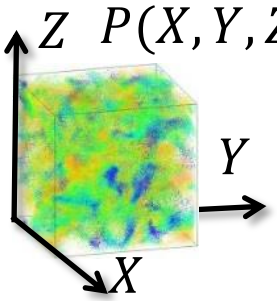
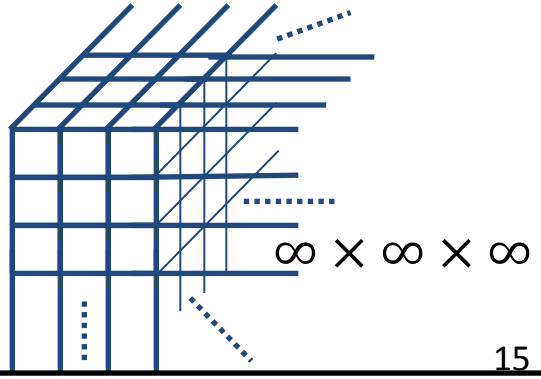
- Previous approach mostly about discrete variables

- Real world data can be continuous, and have multimodal distribution and other rich statistical features



- Replace discrete probabilities by kernel embedding of distributions
  - $k(x, x') = \langle \phi(x), \phi(x') \rangle$, eg., $\exp(-s\|x - x'\|^2)$
  - Expected feature of distribution $\mu_X = \mathbb{E}_X[\phi(X)]$ (can be infinite dimension)
  - Discrete case: use one-of-n feature representation

Kernel embedding and covariance operator

Discrete	$P(X)$  $n \times 1$	$P(X,Y)$  $n \times n$	$P(X,Y,Z)$  $n \times n \times n$
Kernel Embedding	 $\mu_X := \mathbb{E}_X[\phi(X)]$  $\infty \times 1$	 $\mathcal{C}_{XY} := \mathbb{E}_{XY}[\phi(X) \otimes \phi(Y)]$  $\infty \times \infty$	 $\mathcal{C}_{XYZ} := \mathbb{E}_{XYZ}[\phi(X) \otimes \phi(Y) \otimes \phi(Z)]$  $\infty \times \infty \times \infty$

Infinite dimensional tensors

- Eg. $\boldsymbol{\mathcal{C}}_{XYZ} \coloneqq \mathbb{E}_{XYZ}[\phi(X) \otimes \phi(Y) \otimes \phi(Z)] \in \mathcal{F}^3$ can be viewed as a multi-linear operator mapping from $\mathcal{F} \times \mathcal{F} \times \mathcal{F}$ to R

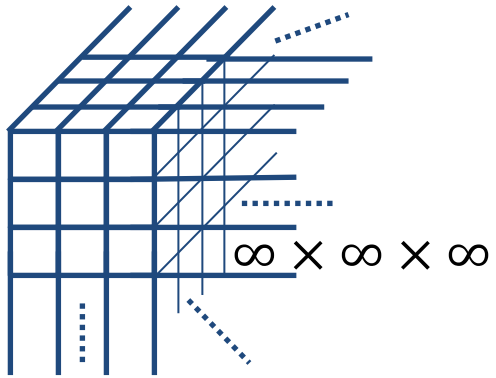
$$\boldsymbol{\mathcal{C}}_{XYZ} \circ_1 f_1 \circ_2 f_2 \circ_3 f_3 = \langle T, f_1 \otimes f_2 \otimes f_3 \rangle_{\mathcal{F}^3}$$
$$\coloneqq \mathbb{E}_{XYZ}[\langle \phi(X), f_1 \rangle_{\mathcal{F}} \langle \phi(Y), f_2 \rangle_{\mathcal{F}} \langle \phi(Z), f_3 \rangle_{\mathcal{F}}]$$

- Given an orthonormal basis $\{e_1, e_2, \dots, e_\infty\}$ of $\mathcal{F}$, define generalized Hilbert-Schmidt norm

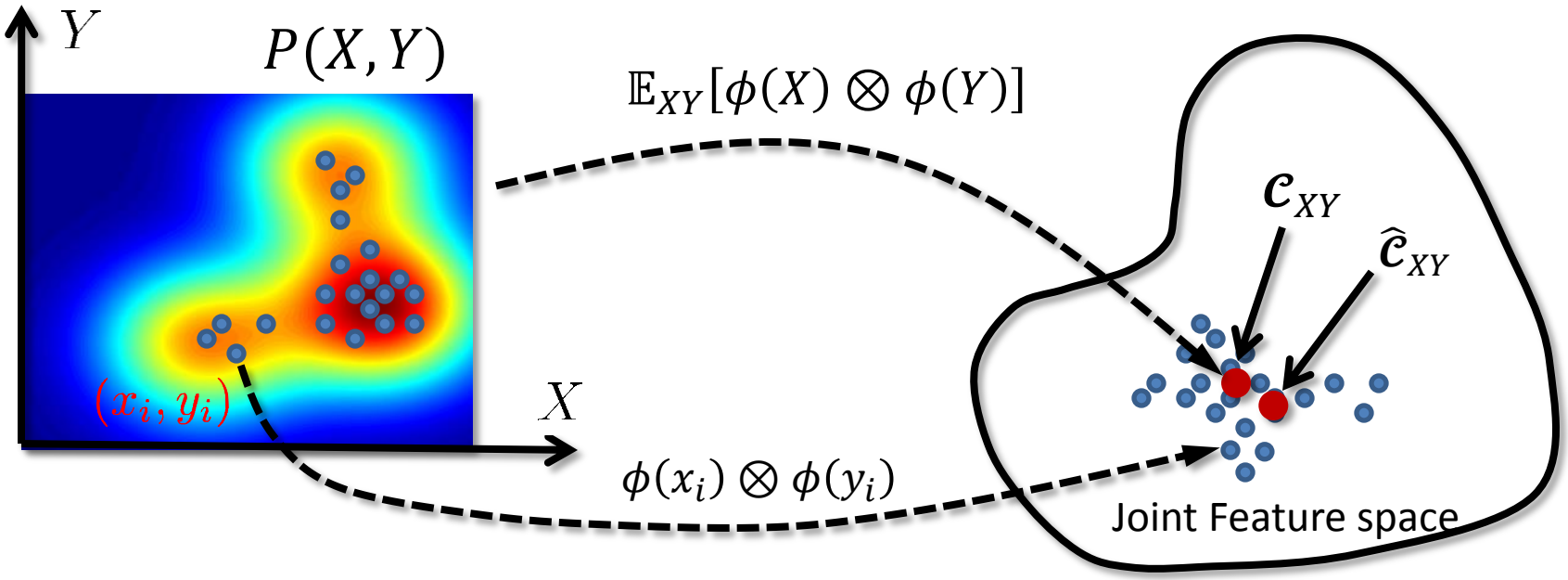
$$\|\boldsymbol{\mathcal{C}}_{XYZ}\|_{HS}^2 \coloneqq \sum_{i_1} \sum_{i_2} \sum_{i_3} \left(\boldsymbol{\mathcal{C}}_{XYZ} \circ_1 e_{i_1} \circ_2 e_{i_2} \circ_3 e_{i_3}\right)^2$$

- For second order tensor $\boldsymbol{\mathcal{C}}_{XY}$, similar to matrix,
- We can reshape the tensor to matrix

$$\boldsymbol{\mathcal{C}}_{\{XY\};\{Z\}} = reshape(\boldsymbol{\mathcal{C}}_{XYZ}, \{XY\})$$



Kernel embedding with finite sample



$$\mathcal{C}_{XY} = \mathbb{E}_{XY}[\phi(X) \otimes \phi(Y)] \approx \widehat{\mathcal{C}}_{XY} = \frac{1}{m}\sum_{i=1}^{m} \phi(x_i) \otimes \phi(y_i)$$

Use finite sample mean to approximate expectation,
But in the end work with kernel matrices

How to deal with infinite features?

- Kernel trick: never explicitly compute features, always turn it into inner product $k(x, x') = \langle \phi(x), \phi(x') \rangle$

- Eg. kernel singular value decomposition
  - $\widehat{\boldsymbol{C}}_{XY} = \frac{1}{m} \sum_{i=1}^{m} \phi(x_i) \otimes \phi(y_i) = U \Sigma V^{\top}$
  - Run kernel principal component analysis on $\widehat{\boldsymbol{C}}_{XY} \widehat{\boldsymbol{C}}_{XY}^{\top}$
  - Eigenvector lies the span of data $U = \sum_{i=1}^{m} \alpha_i \phi(x_i)$
  - Solve a generalized eigenvalue problem
    - $KGK\alpha = \lambda K\alpha$
    - Kernel matrix $K_{ij} = k(x_i, x_j)$ and $G_{ij} = k(y_i, y_j)$

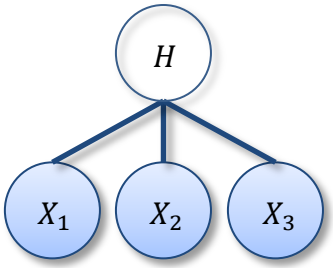
Kernel embedding of latent tree variable models

- Low rank structure of kernel embeddings
- Estimating latent parameters
  - Working with 2nd and 3rd order tenors
  - Tensor power methods
- Estimating the structure of latent variable models
  - Kernel tree metric
  - Neighbor joining algorithm
- Estimating marginal distribution of the observed variables
  - Tensor train decomposition
  - Low rank density estimator

Low rank structure of the kernel embeddings

- Three observed $X_1, X_2, X_3 \in R$, discrete hidden $H \in \{1,2,..,k\}$, joint density
  - $p(X_1, X_2) = \sum_{h=1}^{k} p(X_1|h)p(X_2|h)P(h)$
  - $p(X_1, X_2, X_3) = \sum_{h=1}^{k} p(X_1|h)p(X_2|h)p(X_3|h)P(h)$

- Kernel embedding (let $\pi_h := P(h), \mu_{X|h} := \mathbb{E}_{X|h}[\phi(X)]$)
  - $C_{X_1X_2} = \sum_{h=1}^{k} \pi_h * \mu_{X_1|h} \otimes \mu_{X_2|h}$
  - $C_{X_1X_2X_3} = \sum_{h=1}^{k} \pi_h * \mu_{X_1|h} \otimes \mu_{X_2|h} \otimes \mu_{X_3|h}$



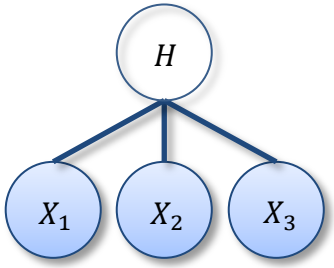
- Tensor form (let $C_{X|H} := (\mu_{X|h=1}, \mu_{X|h=2}, \cdots, \mu_{X|h=k})$)
  - $C_{X_1X_2} = C_{X_1|H} C_{HH} C_{X_2|H}^{\top}$ $\quad (C_{HH} := \mathrm{diag}(\pi_h))$
  - $C_{X_1X_2X_3} = C_{HHH} \circ_1 C_{X_1|H} \circ_2 C_{X_2|H} \circ_2 C_{X_3|H}$ $\quad (C_{HHH} := \mathrm{tdiag}(\pi_h))$

Recovering latent parameters

- Suppose tree structure is given, want to recover parameters $C_{X_1|H}$, $C_{X_2|H}$, $C_{X_3|H}$ and C_{HH} using $C_{X_1X_2}$ and $C_{X_1X_2X_3}$
  - $C_{X_1X_2} = C_{X_1|H} C_{HH} C_{X_2|H}^{\top}$
  - $C_{X_1X_2X_3} = C_{HHH} \circ_1 C_{X_1|H} \circ_2 C_{X_2|H} \circ_2 C_{X_3|H}$

- For simplicity: assume $C_{X_1|H}$, $C_{X_2|H}$, $C_{X_3|H}$ are the same

- Key idea:
  - Use $C_{X_1X_2}$ to obtain some whitening operator
  - Whiten the 3rd order tensor $C_{X_1X_2X_3}$
  - Run orthogonal tensor decomposition
  - Unwhiten to recover parameters



Convert to orthogonal tensor decomposition

- **Step 1**: perform eigen-decomposition of $C_{X_1X_2}$

$$C_{X_1X_2} = C_{X_1|H} C_{HH} C_{X_2|H}^{\top} = U_k S_k U_k^{\top}$$

define whitening matrix $W := U_k S_k^{-1/2}$

$$W^{\top} C_{X_1X_2} W = W^{\top} C_{X_1|H} C_{HH}^{1/2} C_{HH}^{1/2} C_{X_2|H}^{\top} W = MM^{\top} = I$$

M: $k \times k$ orthogonal matrix

- **Step 2**: Whiten tensor $C_{X_1X_2X_3} = C_{HHH} \circ_1 C_{X_1|H} \circ_2 C_{X_2|H} \circ_3 C_{X_3|H}$

$$T = C_{X_1X_2X_3} \circ_1 W^{\top} \circ_2 W^{\top} \circ_3 W^{\top}$$
$$= C_{HHH}^{-1/2} \circ_1 \left(W^{\top} C_{X_1|H} C_{HH}^{1/2}\right) \circ_2 \left(W^{\top} C_{X_2|H} C_{HH}^{1/2}\right) \circ_3 \left(W^{\top} C_{X_3|H} C_{HH}^{1/2}\right)$$
$$= C_{HHH}^{-1/2} \circ_1 M \circ_2 M \circ_3 M$$

T: tensor with orthogonal factor M

Tensor power method

- **Step 3**: use tensor power method to find eigenvector of T
  - Iterate till convergence

$$v \leftarrow T \circ_1 I \circ_2 v \circ_3 v$$

$$v \leftarrow \frac{v}{\|v\|}$$

  - Eigenvalue: $\lambda = T \circ_1 v \circ_2 v \circ_3 v$
  - Deflate $T \leftarrow T - \lambda * v \otimes v \otimes v$ and find the next v

- **Step 4**: recover the latent parameters, let $M = (v_1, v_2, \ldots, v_k)$

$$\pi_i := P(h = i) = \frac{1}{\lambda_i^2}$$

$$C_{X|H} := \left(\mu_{X|h=1}, \mu_{X|h=2}, \ldots, \mu_{X|h=k}\right) = W^{\dagger} M \, diag\left(\pi_i^{-1/2}\right)$$

Kernel algorithm

- Given m observation $\left\{\left(x_1^i, x_2^i, x_3^i\right)\right\}_{i\in[m]}$ drawn iid from a multi-view model $p(X_1, X_2, X_3)$

- **Step 1**: feature matrix with kernel $k(x, x') = \langle \phi(x), \phi(x') \rangle$
  - $\Phi \coloneqq \left(\phi(x_1^1), \ldots, \phi(x_1^m), \phi(x_2^1), \ldots, \phi(x_2^m)\right)$, $K = \Phi^\top \Phi$
  - $\Psi \coloneqq \left(\phi(x_2^1), \ldots, \phi(x_2^m), \phi(x_1^1), \ldots, \phi(x_1^m)\right)$, $L = \Psi^\top \Psi$

- Empirical covariance: $\hat{C}_{X_1 X_2} \coloneqq \frac{1}{2m} \Phi \Psi^\top$

- Generalized eigen-decomposition: $\frac{1}{4m^2} KLK\beta = \hat{\sigma}^2 K\beta$

Kernel algorithm (cont.)

- **Step 2**: Compute third order tensor

$$\hat{T} = \frac{1}{3m}\sum_{i=1}^{m}\begin{pmatrix}\xi(x_1^i)\otimes\xi(x_2^i)\otimes\xi(x_3^i)+\\ \xi(x_3^i)\otimes\xi(x_1^i)\otimes\xi(x_2^i)+\\ \xi(x_3^i)\otimes\xi(x_1^i)\otimes\xi(x_2^i)\end{pmatrix}$$

where $\xi(x_1^i) = \hat{S}_k^{-\frac{1}{2}}(\beta_1,\dots,\beta_k)^\top\Phi^\top\phi(x_1^i)$, and $\hat{S}_k = diag(\hat{\sigma}_1,\dots,\hat{\sigma}_k)$

- **Step 3**: apply tensor power method to $\hat{T}$ and obtain k leading eigenvectors

$$\widehat{M} := (\hat{v}_1,\dots,\hat{v}_k)$$

- **Step 4**: $\hat{C}_{X|H} = \Phi(\beta_1,\dots,\beta_k)\hat{S}_k^{\frac{1}{2}}\widehat{M}$

Guarantees

- Population case: consistent
- m samples: $\pi_{min} := \min_{i \in [k]} \pi_i$, $\theta := \max\left(\frac{k^2}{\sigma_k(C_{X_1X_2})}, \frac{k^{2/3}}{\pi_{min}^{1/3}}\right)$, if

$$m > O\left(\frac{\theta \log \frac{\delta}{2}}{\sigma_k(C_{X_1X_2})}\right)$$

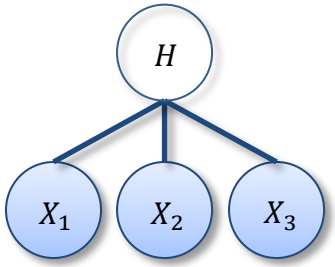
and run tensor power long enough, then with probability $1 - 4\delta$

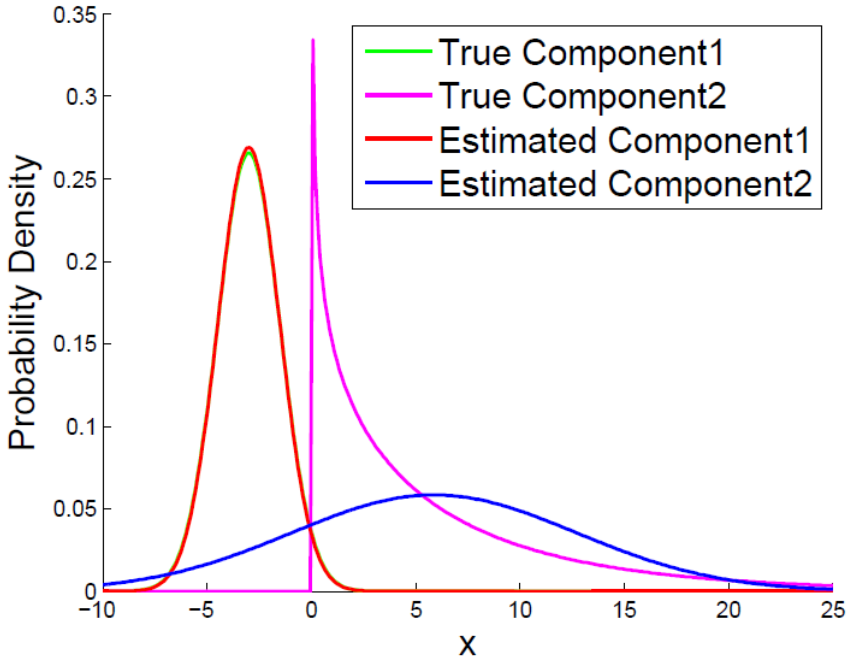
$$\left|\pi_i^{-1/2} - \hat{\pi}_i^{-1/2}\right| \leq O\left(\frac{\sqrt{\log \frac{\delta}{2}}}{\sqrt{m}\,\sigma_k(C_{X_1X_2})}\right)$$

$$\left|\pi_i^{-1/2}\mu_{X|h=i} - W^{\dagger}v_i\right| \leq O\left(\frac{\pi_i^{-1/2}\sqrt{\log \frac{\delta}{2}}}{\sqrt{m}\,\sigma_k(C_{X_1X_2})}\right)$$

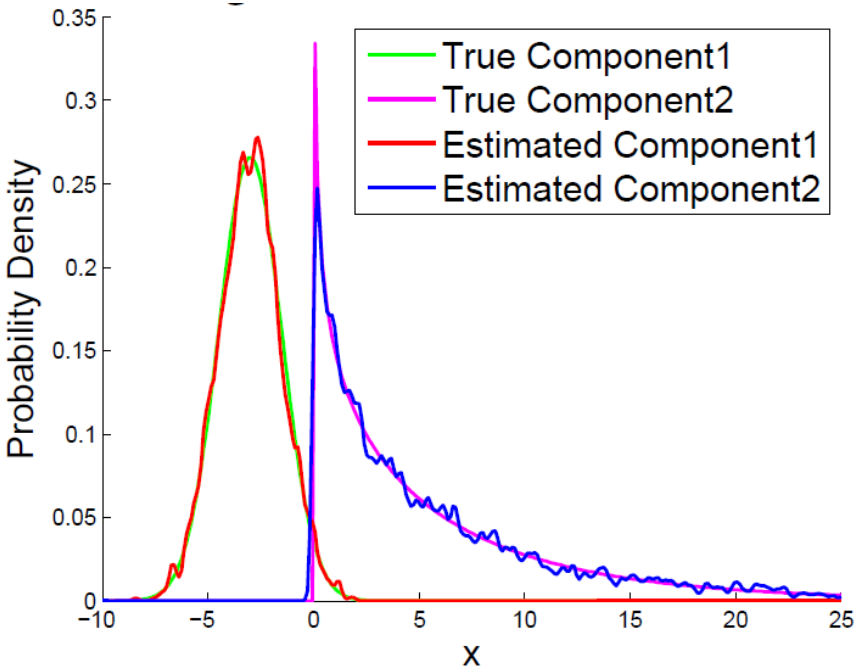
Experiment

- 3 dimensional multi-view latent variable model
- Mixture of Gaussian and shifted Gamma in each view





(a) EM using Mixture of Gaussians Model

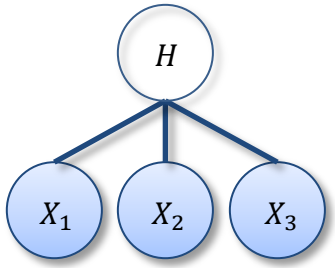


(b) Kernel Spectral

Structure recovery: tree metric

- A metric over the variables in a latent tree model is called a tree metric if the distance d_{st} between $X_s \; and \; X_t$ satisfies

$$d_{st} = \sum_{(u,v) \in Path(s,t)} d_{uv}$$



Eg. $d_{13} = d_{1h} + d_{h3}$

- Let $C_{st} \coloneqq \mathbb{E}_{X_1 X_2}[\phi(X_1) \otimes \phi(X_2)]$, define determinant@$k$

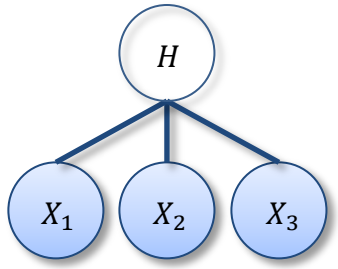
$$\|C_{st}\|_k \coloneqq \prod_{i=1}^{k} \sigma(C_{st})$$

- Nonparametric tree metric:

$$d_{st} = -\frac{1}{2}\log\|C_{st}C_{st}^{\top}\|_k + \frac{1}{4}\log\|C_{ss}C_{ss}^{\top}\|_k + \frac{1}{4}\log\|C_{tt}C_{tt}^{\top}\|_k$$

Tree metric: proof

- Low rank: $\left\|C_{13}C_{13}^{\top}\right\|_k \Rightarrow \left\|C_{1|H}C_{HH}C_{3|H}^{\top}C_{3|H}C_{HH}C_{1|H}^{\top}\right\|_k$
- Sylvester's theorem:$\Rightarrow \left\|C_{1|H}^{\top}C_{1|H}C_{HH}C_{3|H}^{\top}C_{3|H}C_{HH}\right\|_k$
- Split:$\Rightarrow \left\|C_{1|H}^{\top}C_{1|H}C_{HH}\right\|_k\left\|C_{3|H}^{\top}C_{3|H}C_{HH}\right\|_k$



- Multiply and divide: $\Rightarrow \dfrac{\left\|C_{1|H}^{\top}C_{1|H}C_{HH}C_{HH}\right\|_k\left\|C_{3|H}^{\top}C_{3|H}C_{HH}C_{HH}\right\|_k}{\|C_{HH}\|_k\|C_{HH}\|_k}$
- Sylvester again: $\Rightarrow \dfrac{\left\|C_{1|H}C_{HH}C_{HH}C_{1|H}^{\top}\right\|_k\left\|C_{3|H}C_{HH}C_{HH}C_{3|H}^{\top}\right\|_k}{\|C_{HH}C_{HH}\|_k}$

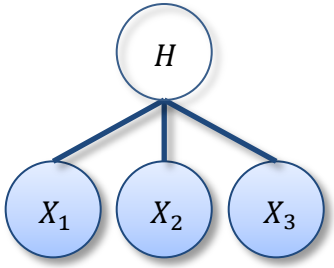
Tree metric: proof II

- Simplify: $\Rightarrow \left\|C_{13}C_{13}^{\top}\right\|_k = \frac{\left\|C_{1H}C_{1H}^{\top}\right\|_k \left\|C_{3H}C_{3H}^{\top}\right\|_k}{\left\|C_{HH}C_{HH}\right\|_k}$

- Plug expansion of $\left\|C_{13}C_{13}^{\top}\right\|_k$ into the distance definition

$$d_{13} = -\frac{1}{2}\log\left\|C_{13}C_{13}^{\top}\right\|_k + \frac{1}{4}\log\left\|C_{11}C_{11}^{\top}\right\|_k + \frac{1}{4}\log\left\|C_{33}C_{33}^{\top}\right\|_k$$
$$= -\frac{1}{2}\log\left\|C_{1H}C_{1H}^{\top}\right\|_k - \frac{1}{2}\log\left\|C_{3H}C_{3H}^{\top}\right\|_k$$
$$+\frac{1}{4}\log\left\|C_{HH}C_{HH}^{\top}\right\|_k + \frac{1}{4}\log\left\|C_{HH}C_{HH}^{\top}\right\|_k$$
$$+\frac{1}{4}\log\left\|C_{11}C_{11}^{\top}\right\|_k + \frac{1}{4}\log\left\|C_{33}C_{33}^{\top}\right\|_k$$

- $\Rightarrow d_{13} = d_{1H} + d_{3H}$



Recover latent tree structure

- Compute d_{st} for all pairs of n observed variables, run neighbor joining algorithm

- While $n > 3$, do
  - $Q_{st} = (n-2)d_{st} - \sum_{i=1}^{n} d_{si} - \sum_{i=1}^{n} d_{ti}$
  - introduce hidden node h to connect s^* and t^*

$$(s^*, t^*) = \text{argmin}_{(s,t)} Q_{st}$$

  - Update distance to the new node h

$$d_{s^*h} = \frac{1}{2} d_{s^*t^*} + \frac{1}{2(n-2)} \left[\sum_{i=1}^{n} d_{s^*i} - \sum_{i=1}^{n} d_{t^*i} \right]$$

$$d_{t^*h} = d_{s^*t^*} - d_{s^*h}$$

$$d_{hu} = \frac{1}{2} [d_{s^*u} + d_{t^*u} - d_{s^*t^*}]$$

  - $n \Leftarrow n - 1$

Property of the algorithm

- Population case: consistent

- m iid. samples: $t_{min} \coloneqq \inf_{(s,t)} \sigma_k(C_{st})$

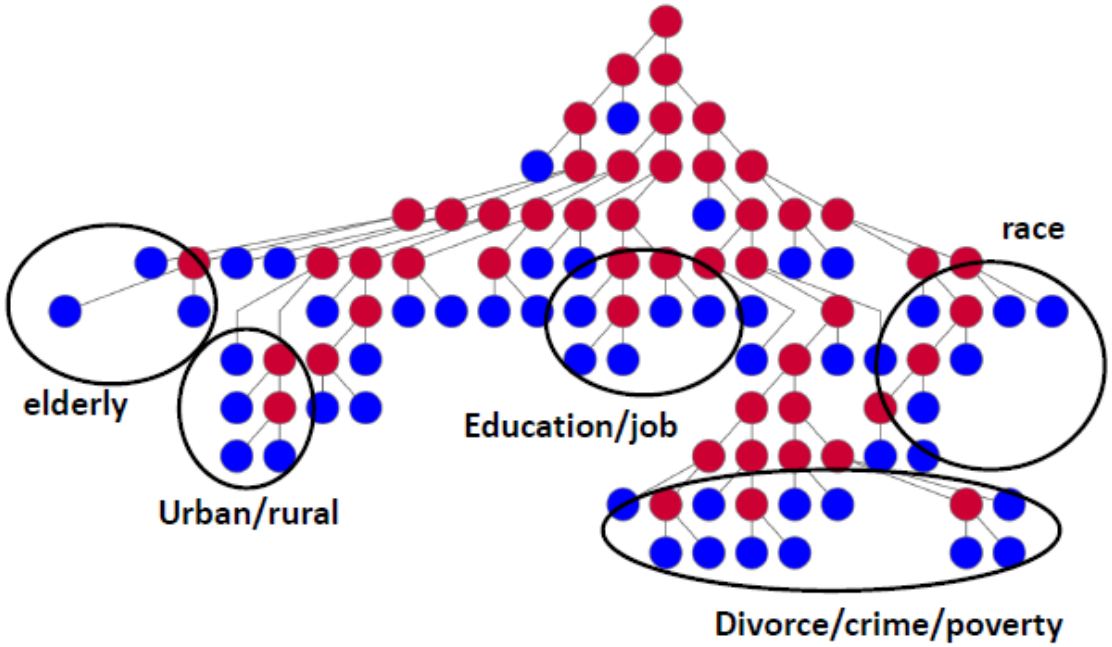
$$Pr\left\{\sup_{(s,t)} \left|\hat{d}_{st} - d_{st}\right| \leq \frac{8C}{t_{min}}\sqrt{\frac{2k\delta \log n}{m}}\right\} \geq 1 - o(1)$$

- Sparsistency: $I_{min} \coloneqq \inf_{(s,t)} d_{st}$, if $\frac{8C}{t_{min}}\sqrt{\frac{2k\delta \log n}{m}} \leq \frac{I_{min}}{4}$

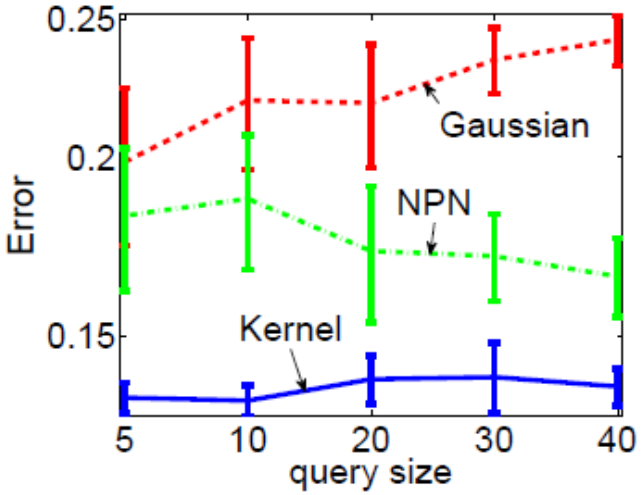
$$Pr\{\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{T}}^*\} \geq 1 - o(1)$$

Experiments on crime data

- High dimensional attribute of different suburbs
- Continuous attributes with non-Gausssian marginals
- Compared to Gaussian and nonparanormal latent variable models



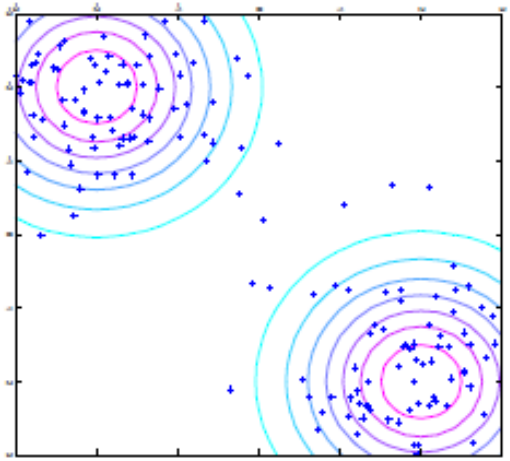
(a)



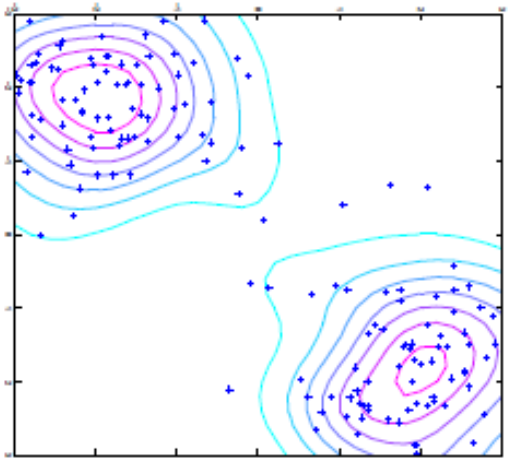
(b)

Low rank structure in data

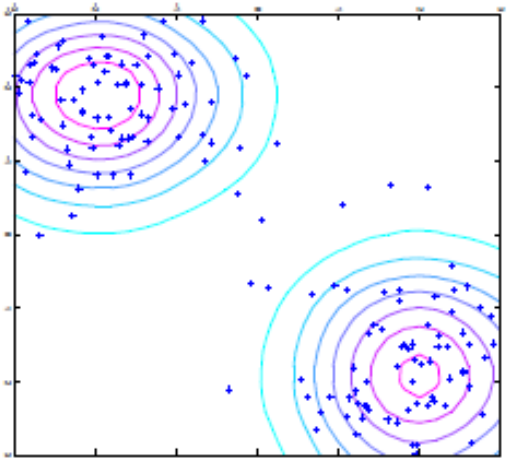
- Many datasets have low rank structure (clusters or manifolds)
- Parametric density estimators makes use of that (eg. Mixture of Gaussians)
- Can we nonparametric methods make use of that?



(a) Ground truth



(b) Ordinary KDE



(c) Low Rank KDE

Density estimator with kernel embeddings

- Eg. two dimensional density $p^*(X_1, X_2)$

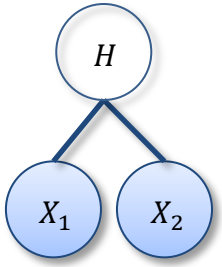
- Kernel density estimator with product kernel $k(x_1, x_1')k(x_2, x_2')$

$$p(x_1', x_2') = \mathbb{E}_{X_1X_2}[k(X_1, x_1)k(X_2, x_2)]$$
$$= \mathbb{E}_{X_1X_2}[\langle \phi(X_1), \phi(x_1') \rangle_{\mathcal{F}} \langle \phi(X_2), \phi(x_2') \rangle_{\mathcal{F}}]$$
$$= \mathbb{E}_{X_1X_2}[\langle \phi(X_1) \otimes \phi(X_2), \phi(x_1') \otimes \phi(x_2') \rangle_{\mathcal{F}^2}]$$
$$= \left\langle C_{X_1X_2}, \phi(x_1') \otimes \phi(x_2') \right\rangle_{\mathcal{F}^2}$$

- In the presence of latent variable H,

$$C_{X_1X_2} = C_{X_1|H} C_{HH} C_{X_2|H}^{\top}$$

- Perform low rank decomposition of $C_{X_1X_2}$



Low rank kernel density estimation

- Given m iid sample, perform rank k SVD approximation, $\hat{C}_{X_1X_2} \approx \hat{U}_k \hat{S}_k \hat{V}_k^\top$; each singluar vector in the span of the data

$$\hat{u}_i = \sum_{j=1}^m \alpha_i^j \phi\left(x_1^j\right), \hat{v}_i = \sum_{j=1}^m \beta_i^j \phi\left(x_2^j\right), \hat{s}_i = (\hat{s}_i - \lambda)_+$$

- Then the density estimate is

$$\hat{p}(x_1', x_2') = \left\langle \hat{U}_k \hat{S}_k \hat{V}_k^\top, \phi(x_1') \otimes \phi(x_2') \right\rangle_{\mathcal{F}^2}$$

$$= \sum_{i=1}^k s_i * \left(\sum_{j=1}^m \alpha_i^j k\left(x_1^j, x_1'\right) \right) \left(\sum_{j=1}^m \beta_i^j k\left(x_2^j, x_2'\right) \right)$$

- Compare to traditional KDE

$$\hat{p}(x_1', x_2') = \frac{1}{m} \sum_{j=1}^m k\left(x_1^j, x_1'\right) k\left(x_2^j, x_2'\right)$$

Property of low rank KDE

- Eg. Gaussian RBF kernel $k(x, x') = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-x')^2}{2\sigma^2}\right)$

- Given kernel bandwidth σ, low rank KDE has the same bias as the traditional KDE (due to the finite bandwidth)

- Low rank KDE can have smaller variance. (work in progress: shrinkage estimator in high dimension has lower variance)

$$
\begin{aligned}
&|p(x_1, x_2) - \hat{p}(x_1', x_2')| \\
&\leq \left\| C_{X_1 X_2} - \hat{U}_k \hat{S}_k \hat{V}_k \right\|_{HS} \|\phi(x_1') \otimes \phi(x_2')\|_{HS} \\
&= \frac{\rho}{\sigma} \left\| U_k S_k V_k^\top - \hat{U}_k \hat{S}_k \hat{V}_k^\top \right\|_{HS}
\end{aligned}
$$

Experiment

- 11 uci datasets with continuous attributes
- Cross-validated kernel bandwidth ($\left\{\frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1,2,4,8\right\}$ of median trick) and the rank (2 to 30)

Table 1: Negative log-likelihood on held-out data (the lower the better).

			Method			
Data Set	# Sample	Dim.	Gaussian mixture	KDE	Spectral	Low rank
australian	690	14	17.97±0.26	18.32±0.64	33.50 ±2.17	**15.88±0.11**
bupa	345	6	8.17±0.30	8.36±0.17	25.01±0.66	**7.57±0.14**
german	1000	24	31.14 ± 0.41	30.57 ± 0.15	28.40 ± 11.64	**22.89 ± 0.26**
heart	270	13	17.72 ±0.23	18.23 ±0.18	21.50 ± 2.39	**16.95 ± 0.13**
ionosphere	351	34	47.60 ±1.77	43.53 ± 1.25	54.91±1.35	**35.84 ± 1.00**
pima	768	8	11.78 ± 0.04	10.38 ± 0.19	31.42 ± 2.40	**10.07 ± 0.11**
parkinsons	195	22	30.13± 0.24	30.65 ± 0.66	33.20 ± 0.70	**28.19 ± 0.37**
sonar	208	60	107.06 ± 1.36	96.17 ± 0.27	89.26 ± 2.75	**57.96 ± 2.67**
wpbc	198	33	50.75 ± 1.11	49.48 ± 0.64	48.66 ± 2.56	**40.78 ± 0.86**
wine	178	13	19.59 ± 0.14	19.56 ± 0.56	19.25 ± 0.58	**18.67 ± 0.17**
yeast	208	79	146.11 ± 5.36	137.15 ± 1.80	76.58 ± 2.24	**72.67 ±4.05**

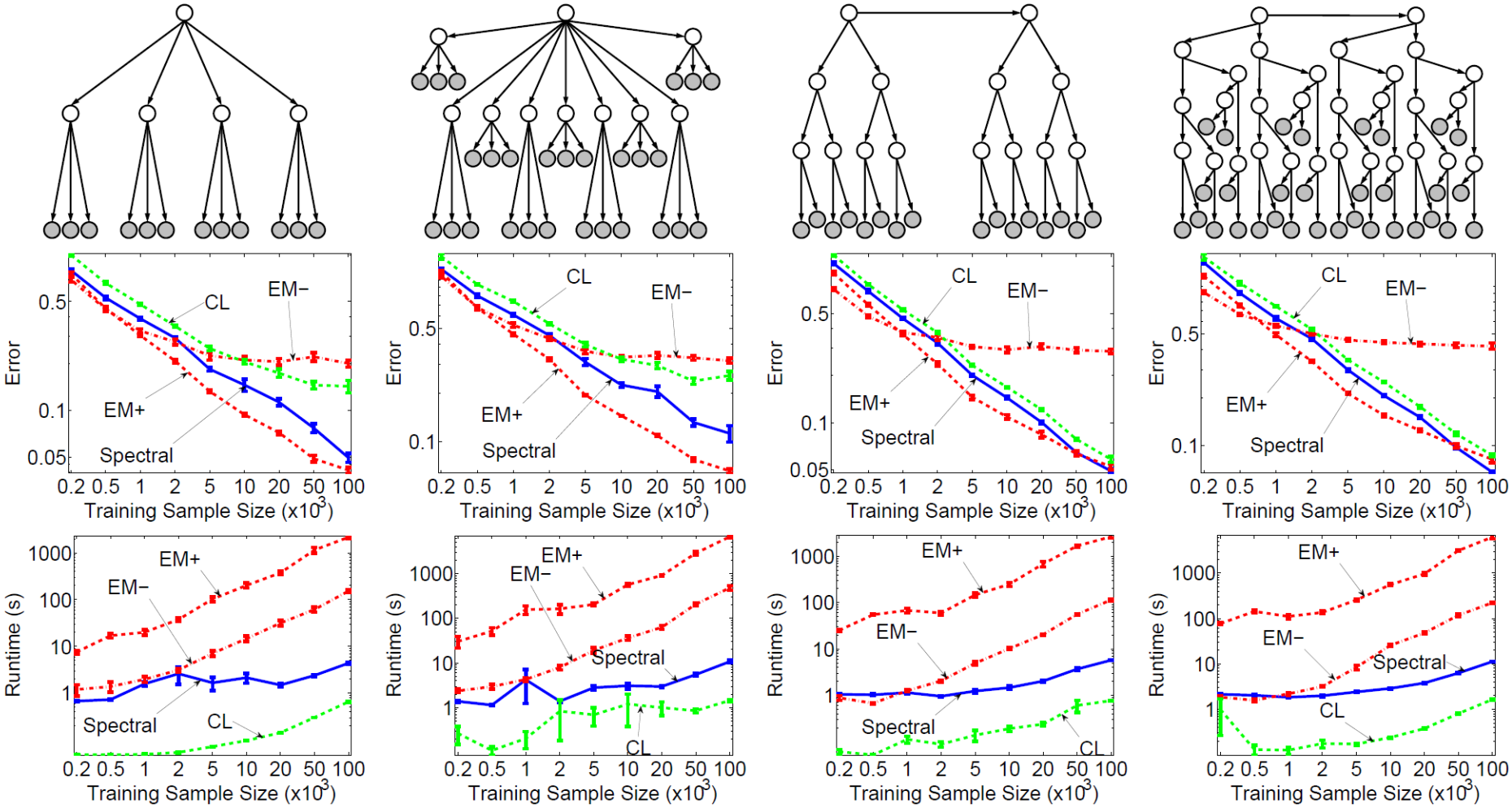
Conclusion

- Low rank structure of the kernel embeddings
  - Estimating latent parameters
    - Working with 2nd and 3rd order tenors
    - Tensor power methods
  - Estimating the structure of latent variable models
    - Kernel tree metric
    - Neighbor joining algorithm
  - Estimating marginal distribution of the observed variables
    - Tensor train decomposition
    - Low rank density estimator

- Future work
  - Extend to other latent variable structures
  - Robust algorithms
  - Algorithms for big datasets

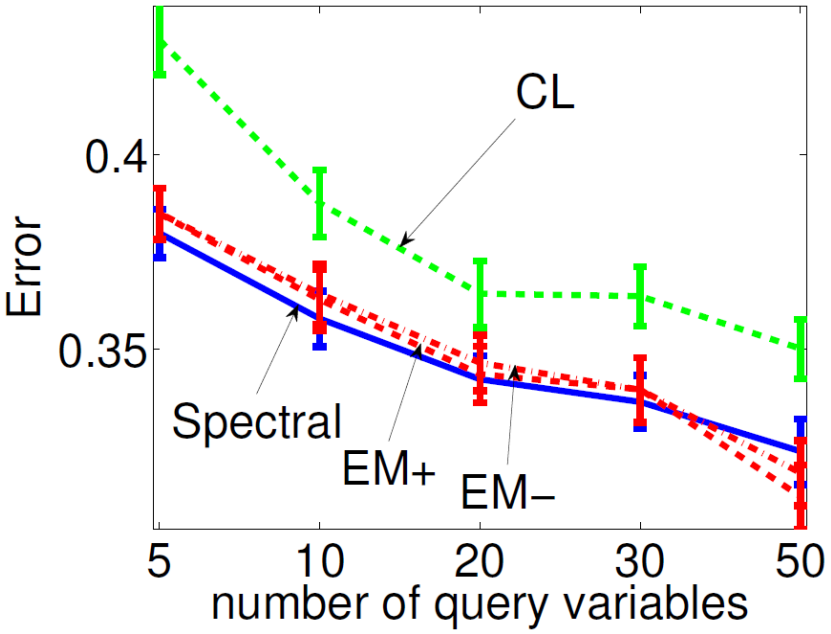
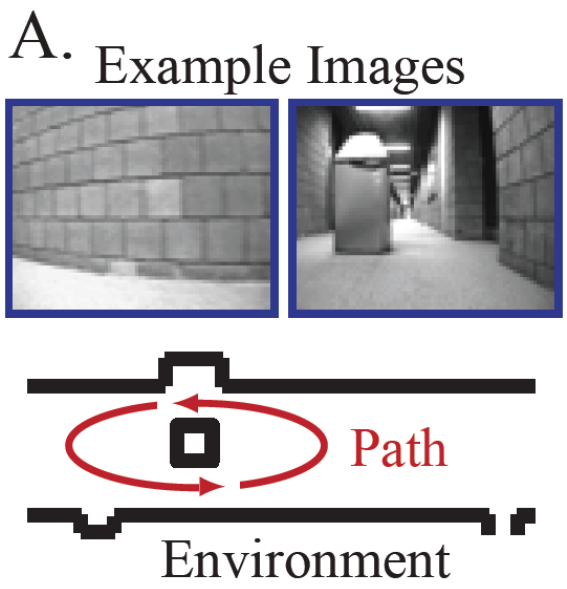
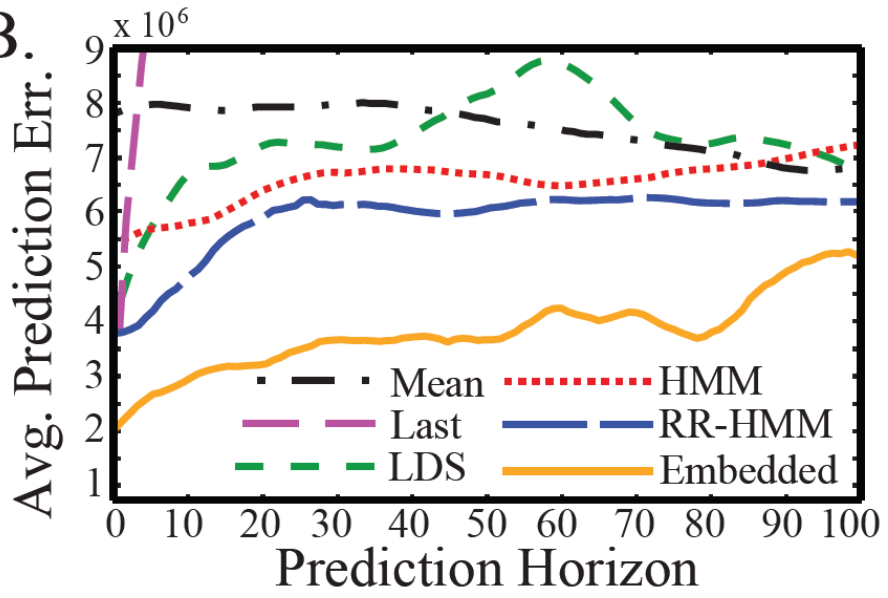
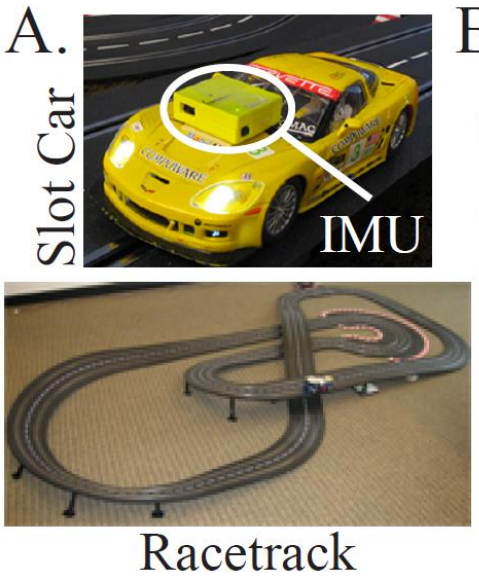
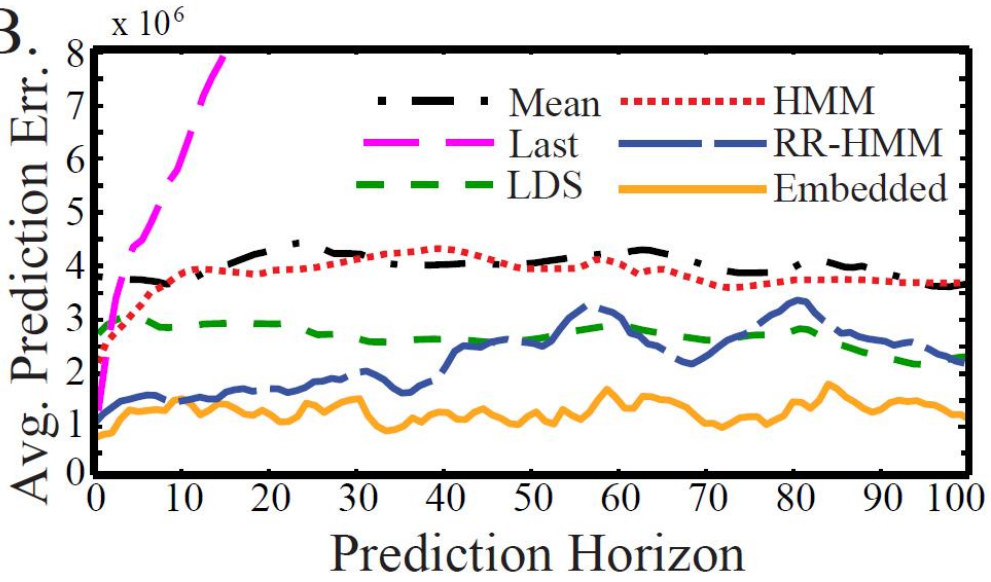
The end

Synthetic Data



Stock Trend Prediction Data

- 59 stocks, 6800 samples, learn latent structure first and then estimate the marginal
- MAP prediction task $x_i = argmax\ P(X_i|x_1, x_2, \ldots, x_{i-1})$ (query i variables)
- Absolute error $|x_i - x_i^\star|$
- Also compared with Chow-Liu tree (fully observed model)



Video and Slot Car Senor Prediction