Tensors and graphical models

Mariya Ishteva with Haesun Park, Le Song

Dept. ELEC, VUB

Georgia Tech, USA

INMA Seminar, May 7, 2013, LLN

Outline

Tensors

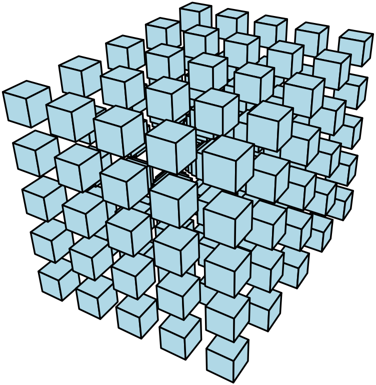
Random variables and graphical models

Tractable representations

Structure learning

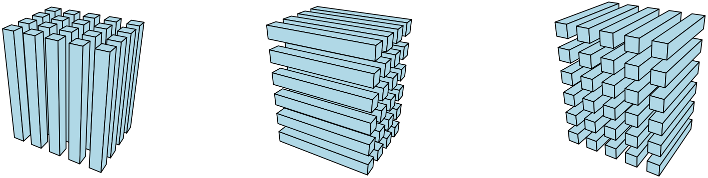
Tensors

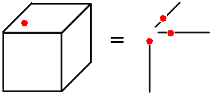
$\mathbb{R}^{M \times N \times P}$



Ranks

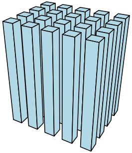
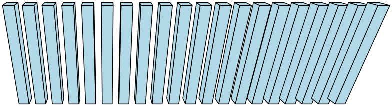
- Multilinear rank (R_1, R_2, R_3)



- Rank-R
    - Rank-1 tensor: 
    - $R = \min(r)$, s.t. $\mathcal{A} = \sum_{i=1}^{r} \{\text{rank-1 tensor}\}_i$

Matrix representations of tensors

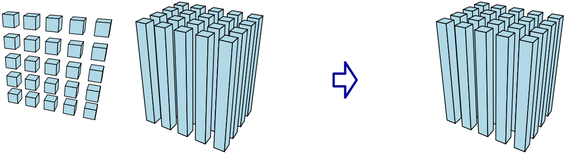
Mode-1

$\mathcal{A} =$  ⇨ $\mathbf{A}_{(1)} =$ 

- Mode-2
- Mode-3
- Multilinear rank: $(\text{rank}(\mathbf{A}_{(1)}), \text{rank}(\mathbf{A}_{(2)}), \text{rank}(\mathbf{A}_{(3)}))$

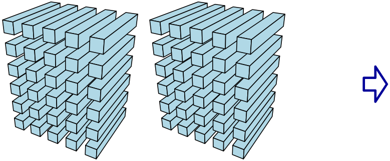
Tensor-matrix multiplication

- Tensor-matrix product



- Contraction $\mathcal{A} \in \mathbb{R}^{I \times J \times M}$ $\mathcal{B} \in \mathbb{R}^{K \times L \times M}$

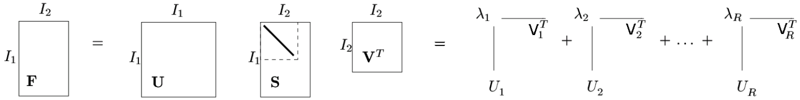
$$\mathcal{C} = \langle \mathcal{A}, \mathcal{B} \rangle_3 \qquad \mathcal{C}(i,j,k,l) = \sum_{m=1}^{M} a_{ijm} b_{klm}$$



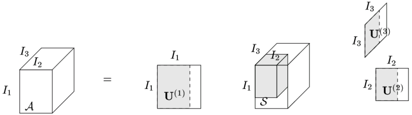
4th order tensor
$\mathcal{C} \in \mathbb{R}^{I \times J \times K \times L}$

Basic decompositions

Singular value decomposition (SVD)



MLSVD / HOSVD



CP / CANDECOMP / PARAFAC

Outline

Tensors

Random variables and graphical models

Tractable representations

Structure learning

Discrete random variables

- Random variable

$$X;\quad 1,\quad \ldots,\quad n$$
$$P_x(1),\quad \ldots,\quad P_x(n) \qquad\qquad P_x \in \mathbb{R}^n,\quad \mathbb{R}^n_+,\quad [0,1]$$

- $X_1, X_2;\ P(X_1, X_2)$ $\qquad P_{12} \in \mathbb{R}^{n\times n}$

	1	$\cdots$	n
1	$P_{12}(1,1)$	$\cdots$	$P_{12}(1,n)$
$\vdots$			
n	$P_{12}(n,1)$	$\cdots$	$P_{12}(n,n)$

- $P(x_1, x_2) := P(X_1 = x_1, X_2 = x_2)$

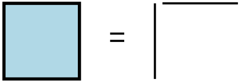
2 random variables

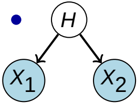
- $X_1, X_2; \quad P(X_1, X_2)$

$P_{12} \in \mathbb{R}^{n \times n}$

- $X_1 \perp X_2$
  $P(x_1, x_2) = P(x_1)P(x_2)$

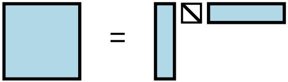
rank-1 matrix



- 

$$P(x_1, x_2) = \sum_h P(x_1|h)P(x_2|h)P(h)$$

low-rank matrix
rank-k matrix, $k < n$



Conditional probability tables (CPTs) $P(X_1|H), P(X_2|H)$

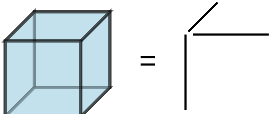
3 random variables

- $X_1, X_2, X_3; \quad P(X_1, X_2, X_3)$

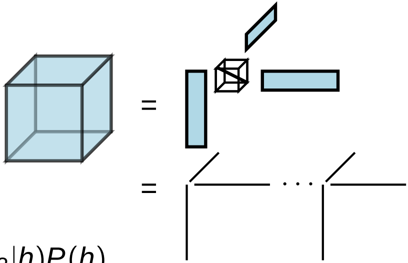
$P_{123} \in \mathbb{R}^{n \times n \times n}$

- X_1, X_2, X_3 independent
  $P(x_1, x_2, x_3) = P(x_1)P(x_2)P(x_3)$

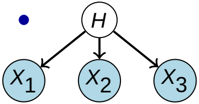
rank-1 tensor



rank-k tensor, $k < n$



-

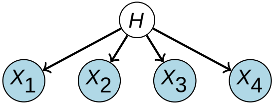


$$P(x_1, x_2, x_3) = \sum_h P(x_1|h)P(x_2|h)P(x_3|h)P(h)$$

4 random variables

- $X_1, X_2, X_3, X_4; \quad P(X_1, X_2, X_3, X_4) \qquad P_{1234} \in \mathbb{R}^{n \times n \times n \times n}$

- X_1, X_2, X_3, X_4 independent

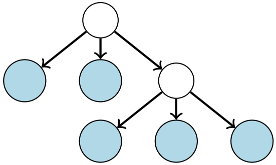
- 

$$P(x_1, x_2, x_3, x_4) = \sum_h P(x_1|h)P(x_2|h)P(x_3|h)P(x_4|h)P(h)$$

- more variables
- more hidden variables

Challenges

- 10 variables, 10 states each $\longrightarrow 10^{10}$ entries
- We need tractable representations
  - Latent variable models / low-rank factors
  - # parameters: exponential $\longrightarrow$ polynomial



- Challenges:
  - Choose a good representation ✓
  - Learn the correct structure ✓
  - Estimate the parameters ×

Outline

Tensors

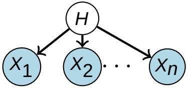
Random variables and graphical models

Tractable representations

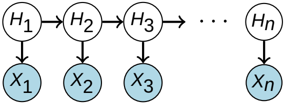
Structure learning

Tensors and graphical models

- CP / CANDECOMP / PARAFAC

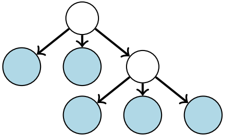


- Tensor train



HMM

- Hierarchical Tucker

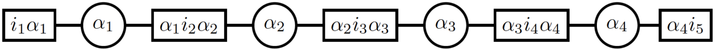


Latent tree model

- Tucker / MLSVD
- Block term decomposition

×

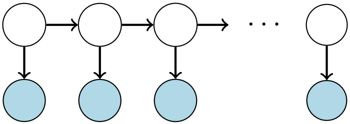
Tensor train (TT) decomposition



$$A(i_1,\ldots,i_d) = \sum_{\alpha_0,\ldots,\alpha_d} G_1(\alpha_0, i_1, \alpha_1) G_2(\alpha_1, i_2, \alpha_2) \ldots G_d(\alpha_{d-1}, i_d, \alpha_d)$$

[I. V. Oseledets, SIAM J. Scientific Computing, 2011]

- Avoids curse of dimensionality
- Small number of parameters, compared to Tucker model
- Slightly more parameters than CP but more stable
- $G_k(\alpha_{k-1}, n_k, \alpha_k)$ has dimensions $r_{k-1} \times n_k \times r_k, \quad r_0 = r_d = 1$
- r_k are called compression ranks:
  $A_k = A_k(i_1, \ldots, i_k; i_{k+1}, \ldots, i_d)$, $\mathrm{rank}(A_k) = r_k$
- Computation based on SVD
- Computation: top $\rightarrow$ bottom



Hierarchical Tucker decomposition

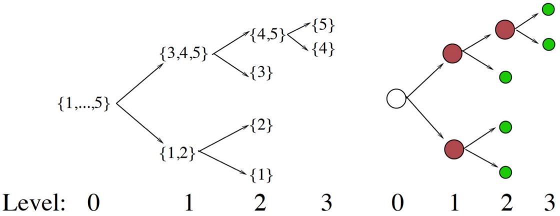
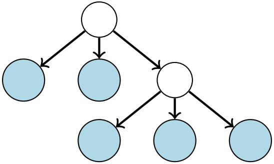


FIG. 3.1. *Left: A dimension tree for $d = 5$. Right: The interior nodes $\mathcal{I}(T_\mathcal{I})$ are colored dark (brown), the leaves $\mathcal{L}(T_\mathcal{I})$ are light (green) and the root is white.*

[L. Grasedyck, SIMAX, 2010]

- Similar properties as TT decomposition
- Computation: bottom $\rightarrow$ top



Potential advantages of tensor approach

- Real data are often multi-way
- Provides higher-level view
- Flexibility: different ranks in each mode: Tucker
- Uniqueness: CP, Block term decomposition
- No curse of dimensionality: Tensor train, hierarch. Tucker

Outline

Tensors

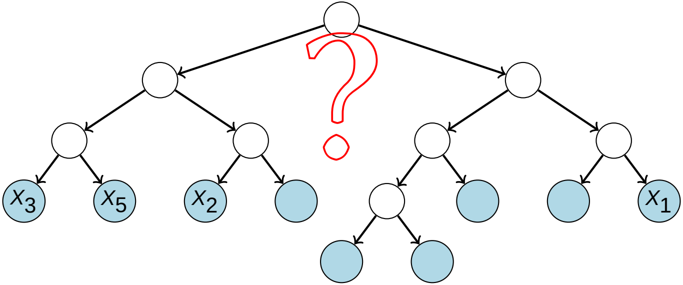
Random variables and graphical models

Tractable representations

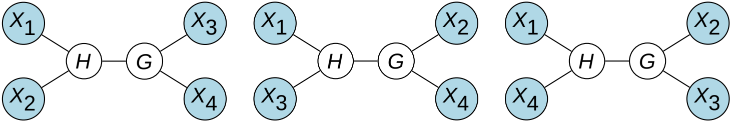
Structure learning

Structure learning

- Given: (samples of) observed variables
- Assumption: the variables can be connected via hidden variables in a tree structure in a meaningful way
- Find: the tree / the relationships between the variables
- Additional difficulty: unknown number of hidden states

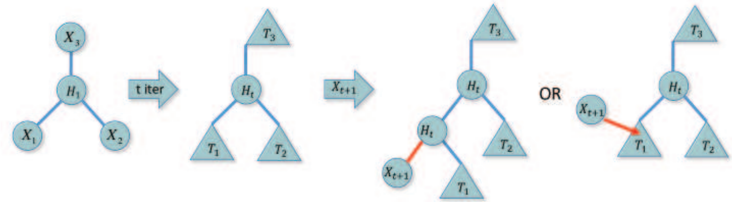


Quartet relationships: topologies



$$P(x_1, x_2, x_3, x_4) = \sum_{h,g} P(x_1|h)P(x_2|h)P(h,g)P(x_3|g)P(x_4|g)$$

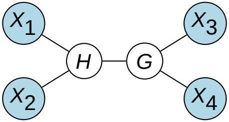
Building trees based on quartet relationships



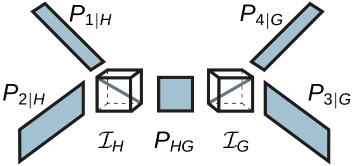
- Choose 3 variables and form a tree
- Add all other variables, one by one
  - Split the current tree into 3 subtrees
  - Choose 3 variables from different subtrees
  - Resolve the quartet relation with current and chosen variables
  - Insert the current variable in a subtree or connect to the tree

[For simplicity, assume each latent variable has 3 neighbors]

Tensor view of quartets



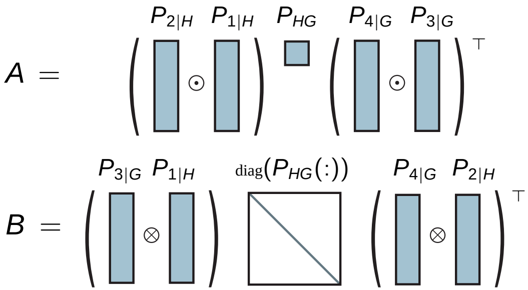
$\mathcal{P}(X_1,X_2,X_3,X_4) \quad =$



$$
\begin{aligned}
A &= \text{reshape}(\mathcal{P}, n^2, n^2);\\
B &= \text{reshape}(\text{permute}(\mathcal{P}, [1,3,2,4]), n^2, n^2);\\
C &= \text{reshape}(\text{permute}(\mathcal{P}, [1,4,2,3]), n^2, n^2).
\end{aligned}
$$

Notation: $P_{1|H}$, $P_{2|H}$, etc. stand for $P(X_1|H)$, $P(X_2|H)$, etc.

Rank properties of matrix representations



- $\mathrm{rank}(A) = \mathrm{rank}(P_{HG}) = k$
  $\mathrm{rank}(B) = \mathrm{rank}(C) = \mathrm{nnz}(P_{HG})$

$$\boxed{\mathrm{rank}(A) \ll \mathrm{rank}(B) = \mathrm{rank}(C)}$$

- Sampling noise ⇨ Nuclear norm relaxation

$$\|A\|_* = \sum_{i=1}^{n^2} \sigma_i(A)$$

Resolving quartet relations

Algorithm 1 $i^* = \text{Quartet}(X_1, X_2, X_3, X_4)$

1: Estimate $\widehat{\mathcal{P}}(X_1, X_2, X_3, X_4)$ from a set of m *i.i.d.* samples.

2: Unfold $\widehat{\mathcal{P}}$ into matrices $\widehat{A}$, $\widehat{B}$ and $\widehat{C}$, and compute

$$a_1 = \|\widehat{A}\|_*, \ a_2 = \|\widehat{B}\|_* \text{ and } a_3 = \|\widehat{C}\|_*.$$

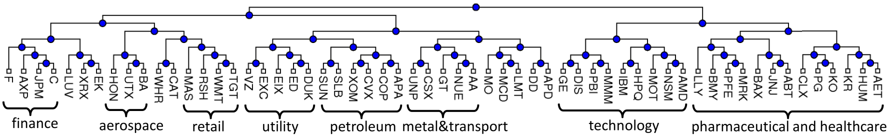
3: Return $i^* = \arg\min_{i \in \{1,2,3\}} a_i$.

- Easy to compute
- Recovery conditions
- Finite sample guarantees
- Agnostic to the number of hidden states
- Compares favorably to alternatives

Example: stock data

Given: stock prices (25 years, discretized into 10 values)

Find: relations between stocks



- Finance:
  - C (Citigroup)
  - JPM (JPMorgan Chase)
  - AXP (American Express)
  - F (Ford Motor: Automotive and Financial Services)
- Retailers:
  - TGT (Target)
  - WMT (WalMart)
  - RSH (RadioShack)

Conclusions

- Tensor decompositions are related to graphical models
- A common goal: tractable representations
- Tensors can be used for structure learning

Thank you!

mariya.ishteva@vub.ac.be