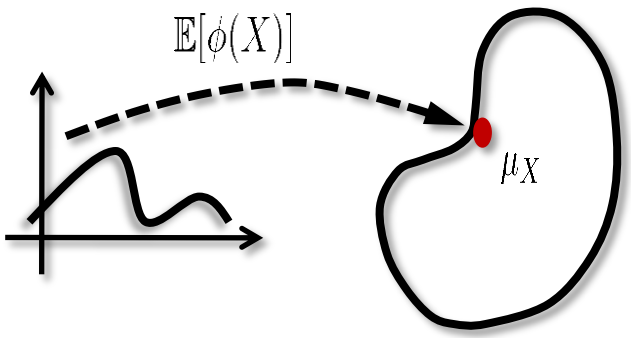
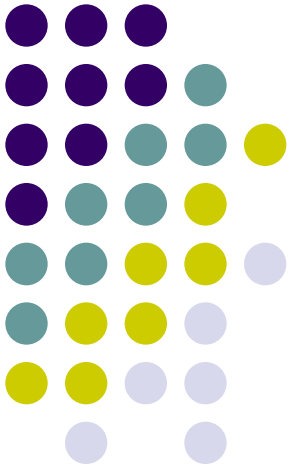
School of Computer Science
Carnegie Mellon

Probabilistic Graphical Models

Spectral Algorithms for Graphical Models

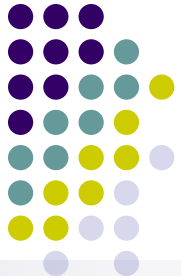




Ankur Parikh

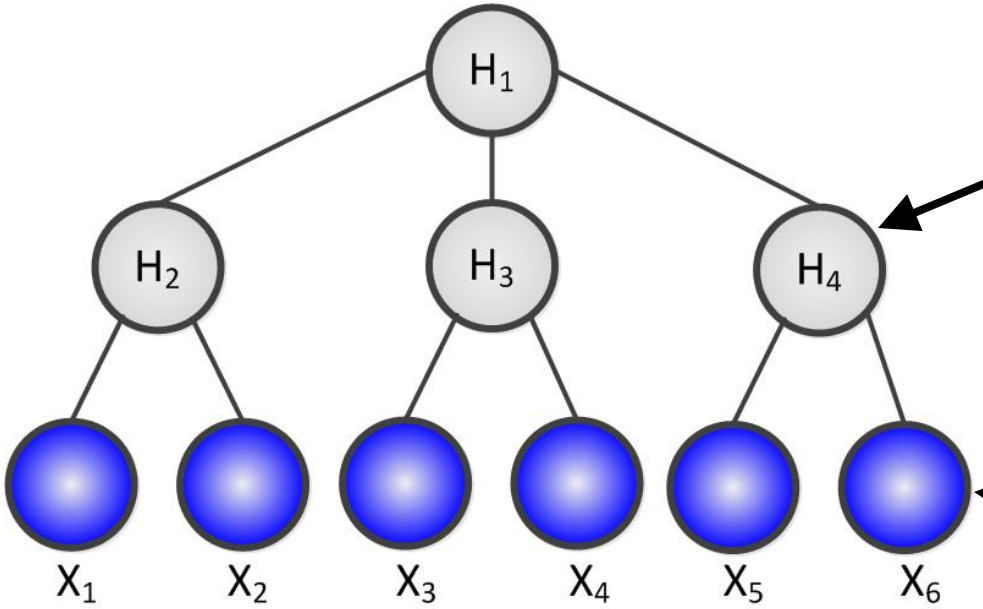
Lecture 21, April 3, 2013

The Linear Algebra View of Latent Variable Models



- Linear algebra can provide a different perspective of graphical models
- Specifically, linear algebra for latent variable models **(Spectral Algorithms)**
- As we will see in this lecture, there is a deep connection between the notion of low rank models and latent variable models.
- This connection will allow us to derive local minima free learning algorithms for latent variable models (that do not use EM).

Consider the following Tree

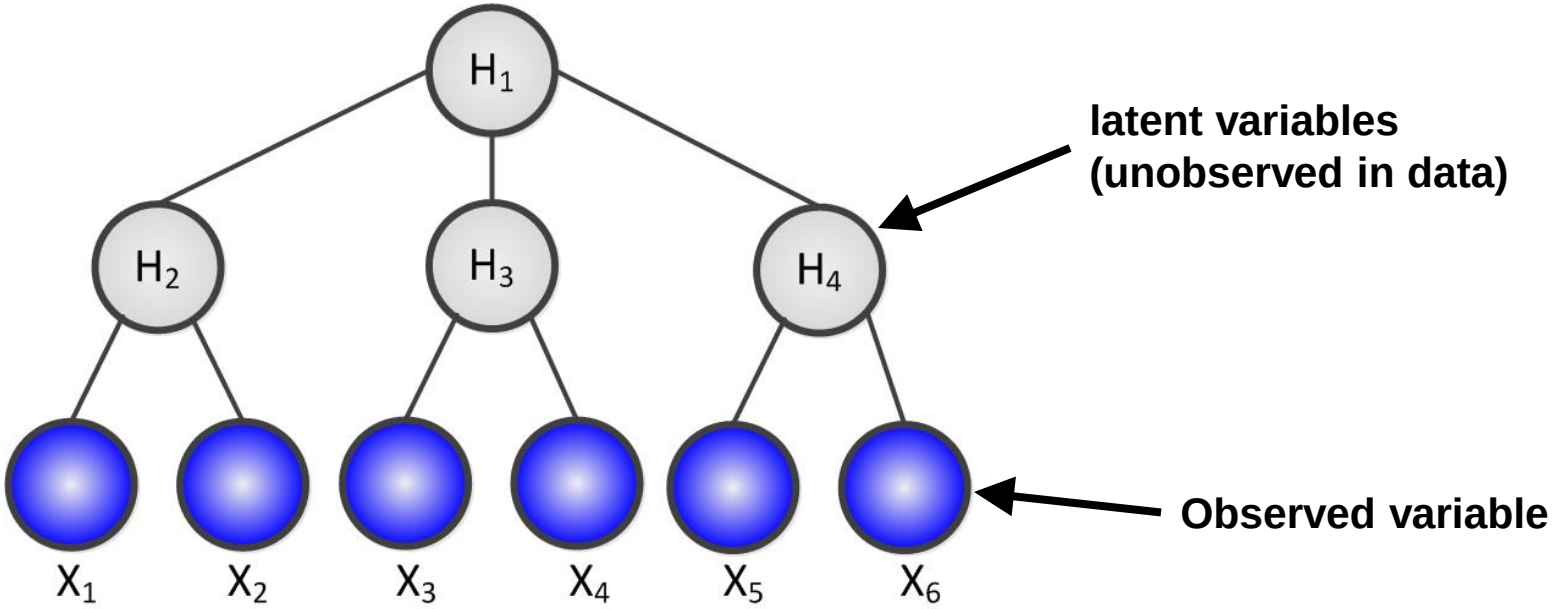


latent variables (unobserved in data)

Observed variable

$$\mathbb{P}[X_1, ..., X_6, H_1, ..., H_4] = \mathbb{P}[H_i] \prod_{i=2}^{4} \mathbb{P}[H_i | H_{\pi(H_i)}] \prod_{i=1}^{6} \mathbb{P}[X_i | H_{\pi(X_i)}]$$

Learning Parameters (EM)



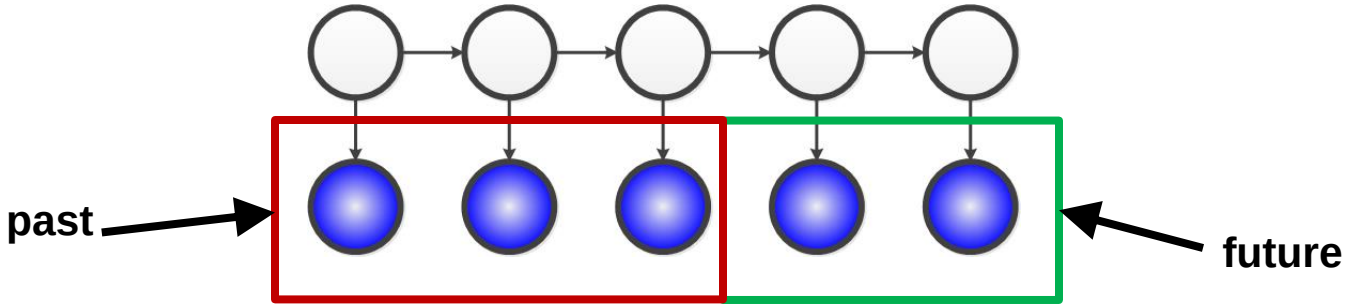
Since latent variables are not observed in the data, we have to use Expectation Maximization (EM) to learn parameters

- **Slow**
- **Local Minima**

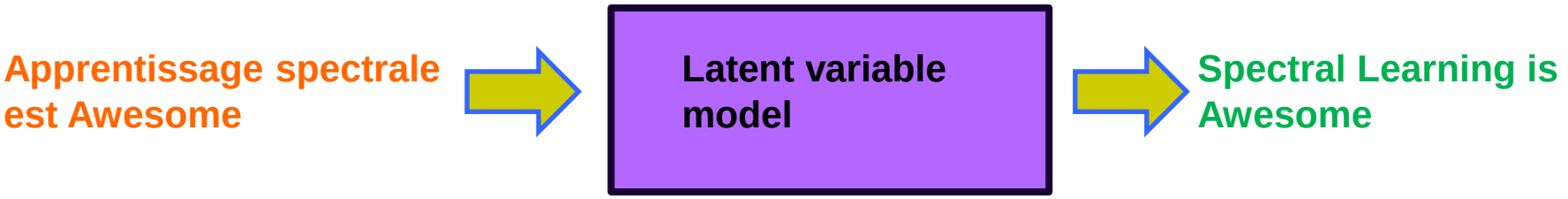
Focusing on Prediction

- In many applications that use latent variable models, the end task is not to recover the latent states, but rather to use the model for prediction among observed variables.

- Dynamical Systems – Predict future given past



- Machine Translation

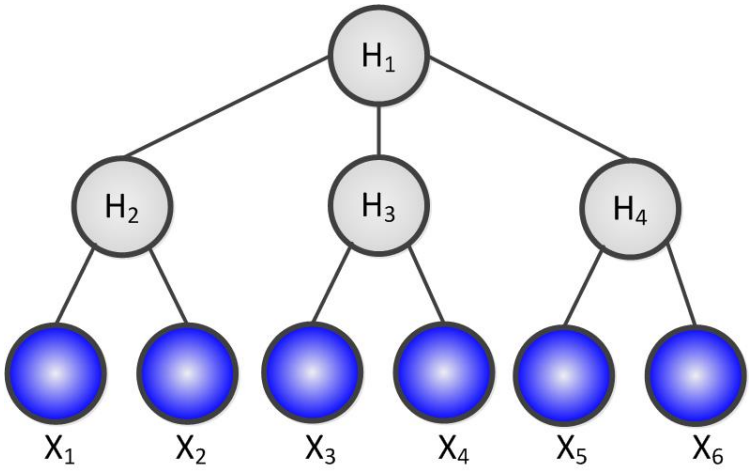


Focusing on Prediction

- As a result, for most of this lecture, we are concerned with quantities related to the observed variables:

$$\mathcal{P}[X_1, X_2, X_3, X_4, X_5, X_6]$$

- We do not care about the latent variables explicitly.

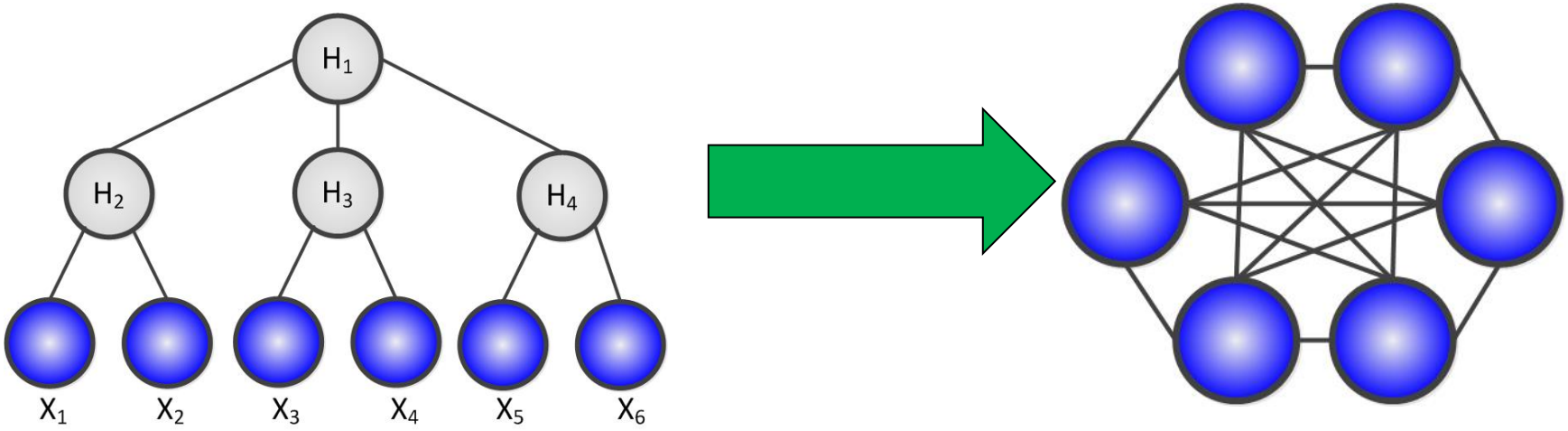


- **Do we still need EM to learn the parameters?**

But if we don't care about the latent variables....

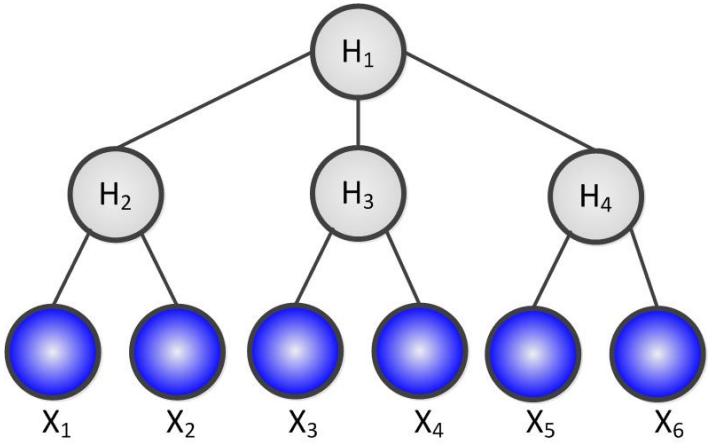
- Why don't we just integrate them out?

- Because integrating them out results in a clique ☹



- The clique is the minimal I-map.

Marginal Does Not Factorize

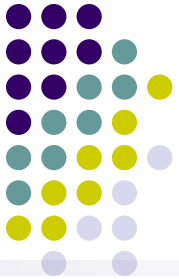


$$\mathbb{P}[X_1, X_2, X_3, X_4, X_5, X_6] = \sum_{H_1,\ldots,H_6} \prod_{i=2}^{4} \mathbb{P}[H_i | H_{\pi(H_i)}] \prod_{i=1}^{6} \mathbb{P}[X_i | H_{\pi(X_i)}]$$

Does not factorize due to the outer sum (Can somewhat distribute the sum, but doesn't solve problem)

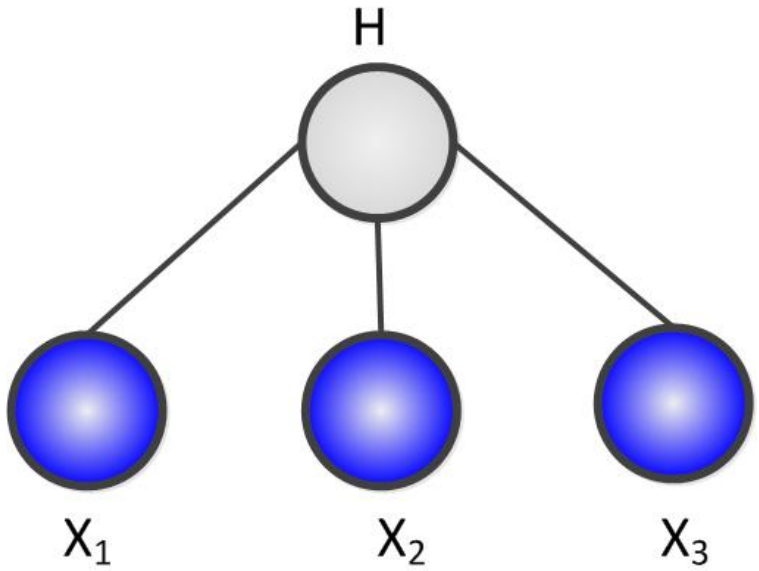
$$\mathbb{P}[X_1, X_2, X_3, X_4, X_5, X_6] = \sum_{H_1} \mathbb{P}[H_1] \prod_{i=2}^{4} \sum_{H_i} \mathbb{P}[H_i | H_{\pi(H_i)}] \prod_{i=1}^{6} \sum_{X_i} \mathbb{P}[X_i | H_{\pi(X_i)}]$$

But isn't a latent tree different from a clique?



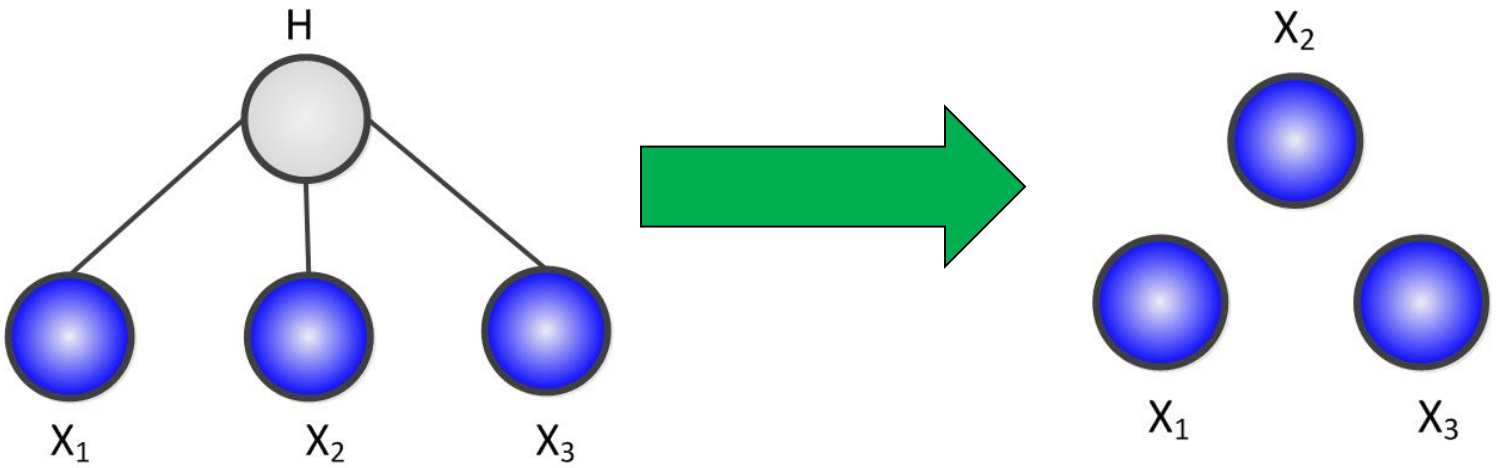
- It depends on the number of latent states.

- Consider the following model.



If H has only one state.....

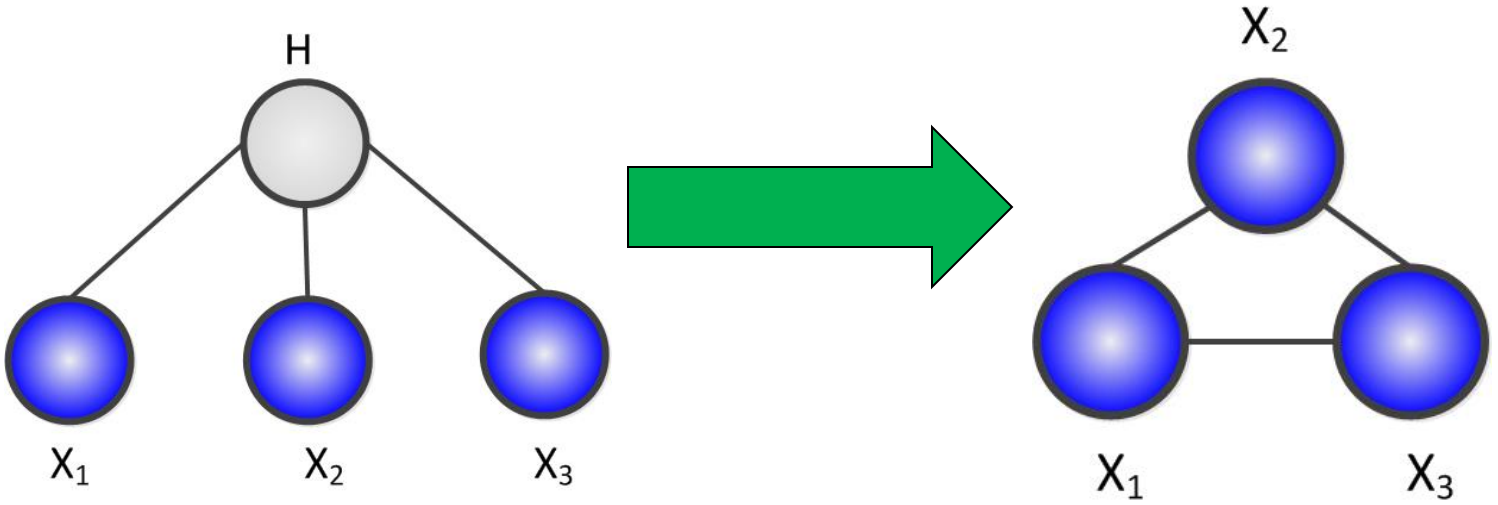
- Then the observed variables are independent!



What if H has many states?

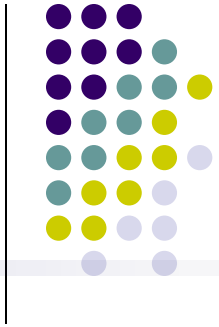
- Let us say the observed variables each have ***m*** states.

- Then if H has m^3 states then the latent model can be exactly equivalent to a clique (depending on how parameters are set).



- But what about all the other cases?

The Question

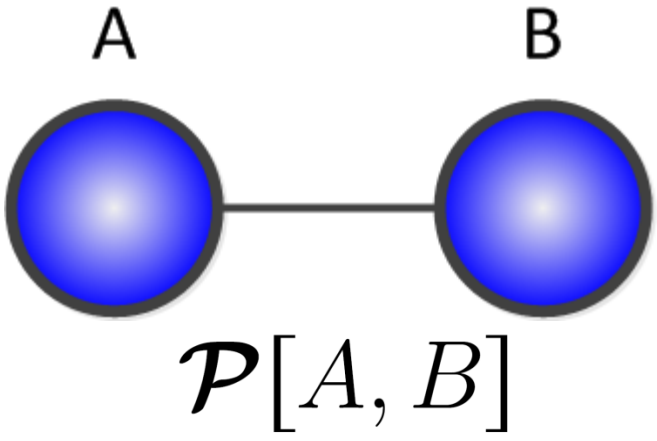


- Under existing methods, latent models all require EM to learn regardless of the number of hidden states.

- However, is there a formulation of latent variable models where the difficulty of learning is a function of the number of latent states?

- This is the question that the ***spectral view*** will answer.

Graphical Models: The Linear Algebra View



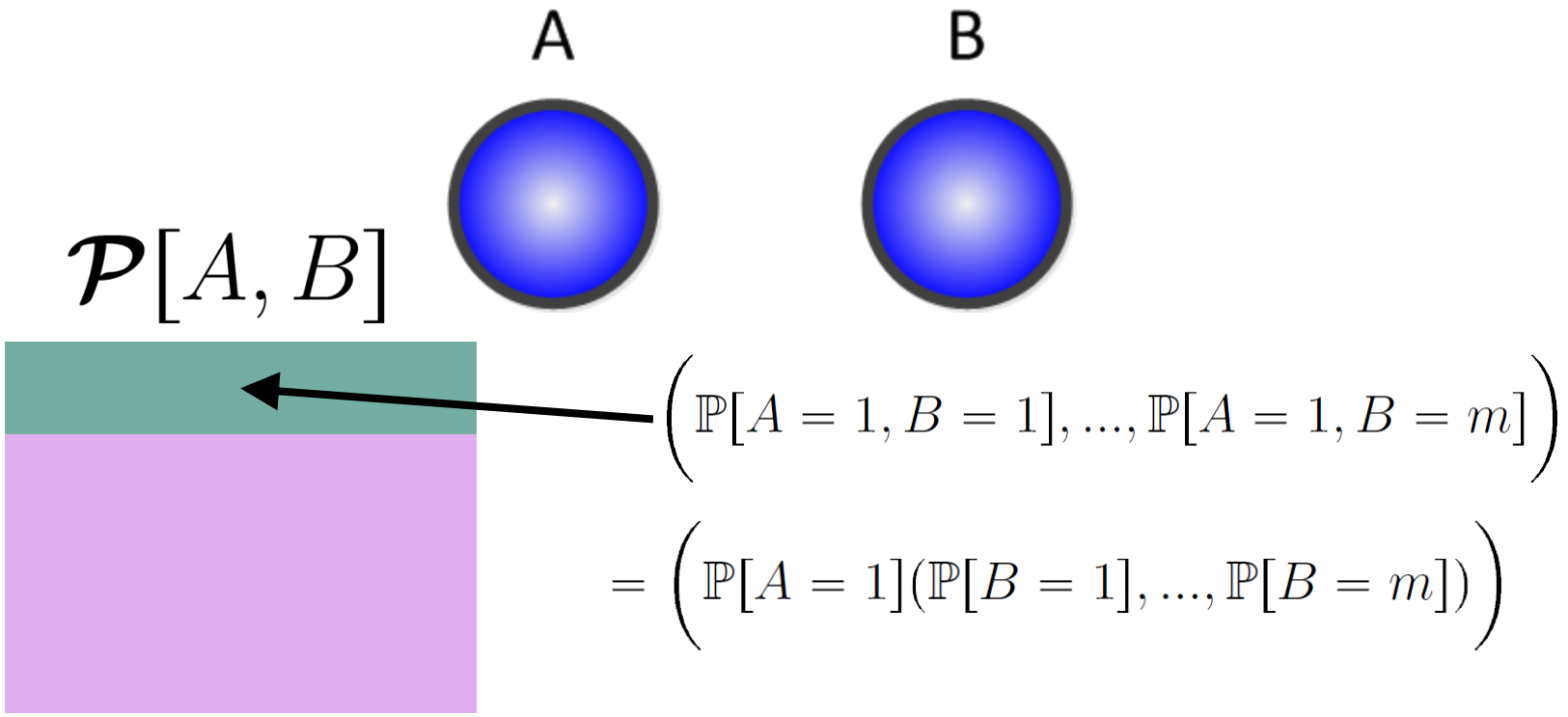
$$\mathcal{P}[A, B]$$

A and B have m states each.

- In general, nothing we can say about the nature of this matrix.

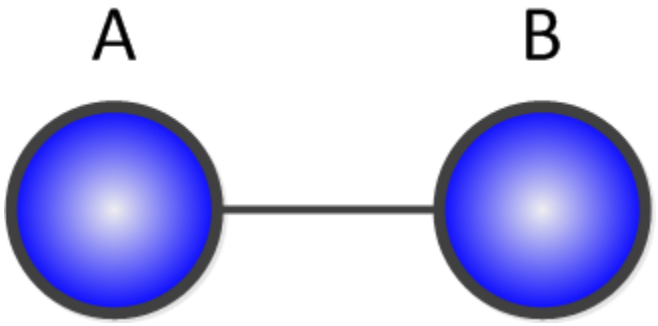
Independence: The Linear Algebra View

- What if we know A and B are independent?

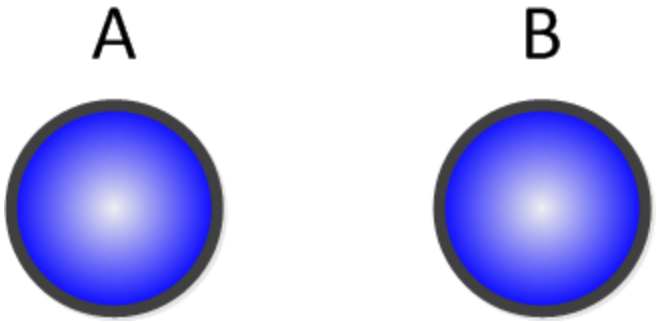


- Joint probability matrix is rank one, since all rows are multiples of one another!!

Independence and Rank



$\mathcal{P}[A, B]$ **has rank m (at most)**

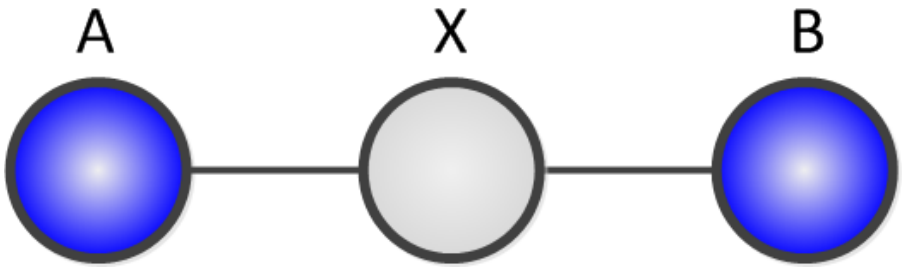


$\mathcal{P}[A, B]$ **has rank 1**

- **What about rank in between 1 and m?**

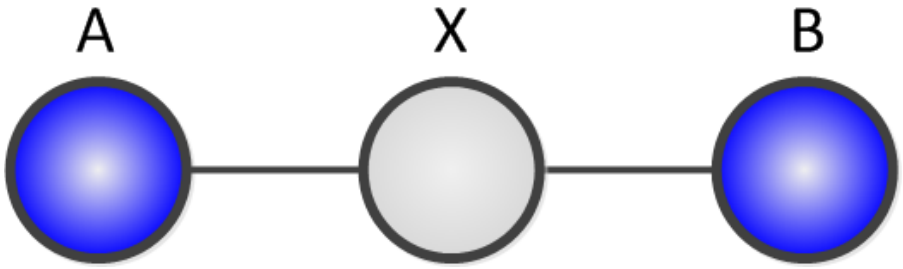
Low Rank Structure

- ***A*** and ***B*** are not marginally independent (They are only conditionally independent given ***X***).



- Assume ***X*** has ***k*** states (while ***A*** and ***B*** have ***m*** states).
- Then, $rank(\boldsymbol{\mathcal{P}}[A, B]) \leqslant k$
- Why?

Low Rank Structure

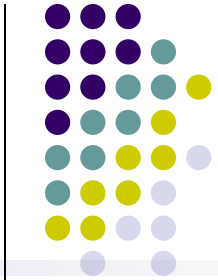


$\mathcal{P}[A, B]$ $\mathcal{P}[A|X]$ $\mathcal{P}(\oslash X)$ $\mathcal{P}[B|X]^{\top}$

$=$

$rank \leq k$ $rank \leq k$ $rank \leq k$ $rank \leq k$

The Spectral View

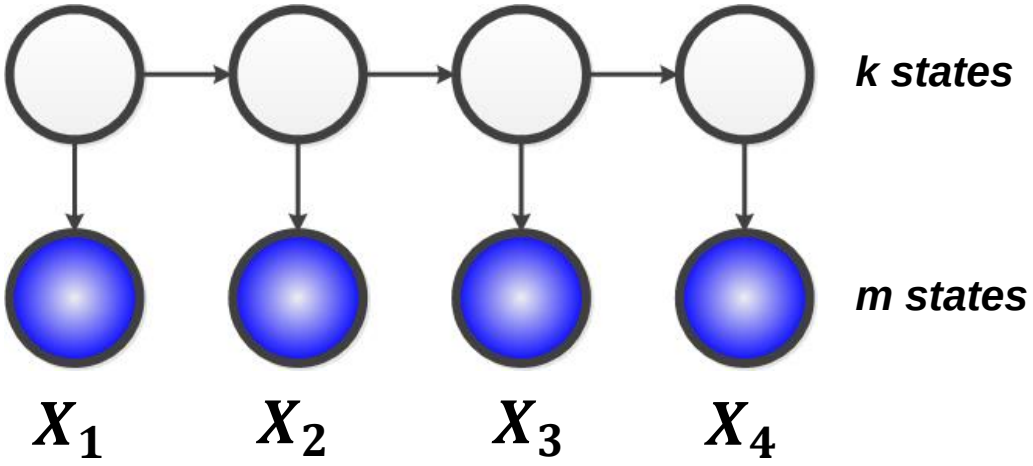


- Latent variable models encode **low rank dependencies** among variables ***(both marginal and conditional)***

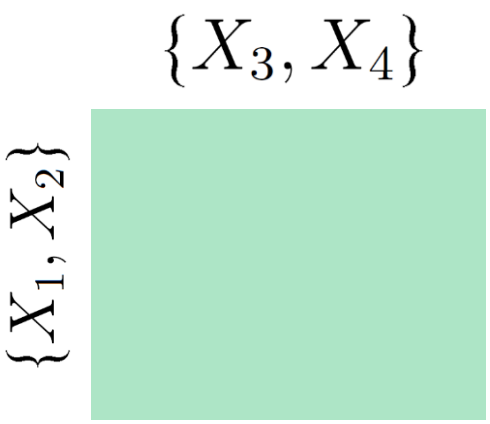
- Use tools from linear algebra to exploit this structure.
  - Rank
  - Eigenvalues
  - SVD
  - Tensors

- In the rest of the lecture we will focus on **parameter learning** in latent variable models.

A More Interesting Example

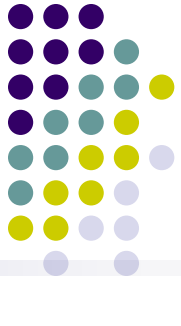


$\mathcal{P}[X_{\{1,2\}}, X_{\{3,4\}}]$



has rank k

Low Rank Matrices "Factorize"



$$M = RL$$

m by n m by k k by n

If M has rank k

We already know one factorization!!!

$$\mathcal{P}[X_{\{1,2\}}, X_{\{3,4\}}] = \mathcal{P}[X_{\{1,2\}}|H_2]\mathcal{P}[\oslash H_2]\mathcal{P}[X_{\{3,4\}}|H_2]^\top$$

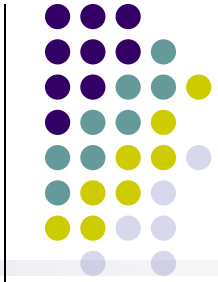
Factor of 4 variables

Factor of 3 variables

Factor of 1 variable

Factor of 3 variables

Alternate Factorizations



- The key insight is that this factorization is not unique.

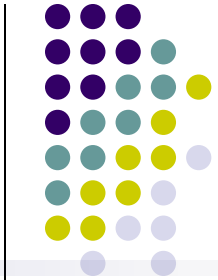
- Consider Matrix Factorization. Can add any invertible transformation:

$$M = RL$$

$$M = RSS^{-1}L$$

- **The magic of spectral learning is that there exists an alternative factorization that only depends on observed variables!**

An Alternate Factorization



- Let us say we only want to factorize this matrix of 4 variables

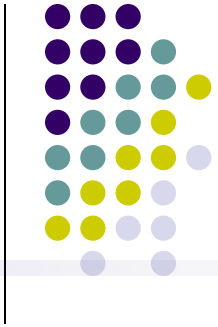
$$\mathcal{P}[X_{\{1,2\}}, X_{\{3,4\}}]$$

such that it is product of matrices that contain at most three ***observed*** variables e.g.

$$\mathcal{P}[X_{\{1,2\}}, X_3]$$

$$\mathcal{P}[X_2, X_{\{3,4\}}]$$

An Alternate Factorization



- Note that

$$\boldsymbol{\mathcal{P}}[X_{\{1,2\}}, X_3] = \boldsymbol{\mathcal{P}}[X_{\{1,2\}}|H_2]\boldsymbol{\mathcal{P}}[\oslash H_2]\boldsymbol{\mathcal{P}}[X_3|H_2]^\top$$

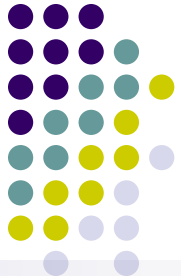
$$\boldsymbol{\mathcal{P}}[X_2, X_{\{3,4\}}] = \boldsymbol{\mathcal{P}}[X_2|H_2]\boldsymbol{\mathcal{P}}[\oslash H_2]\boldsymbol{\mathcal{P}}[X_{\{3,4\}}|H_2]^\top$$

- Product of green terms (in some order) is

$$\boldsymbol{\mathcal{P}}[X_{\{1,2\}}, X_{\{3,4\}}]$$

- Product of red terms (in some order) is $\boldsymbol{\mathcal{P}}[X_2, X_3]$

An Alternate Factorization



$$\mathcal{P}[X_{\{1,2\}}, X_{\{3,4\}}] = \mathcal{P}[X_{\{1,2\}}, X_3]\mathcal{P}[X_2, X_3]^{-1}\mathcal{P}[X_2, X_{\{3,4\}}]$$

factor of 4 variables **factor of 3 variables** **factor of 3 variables**

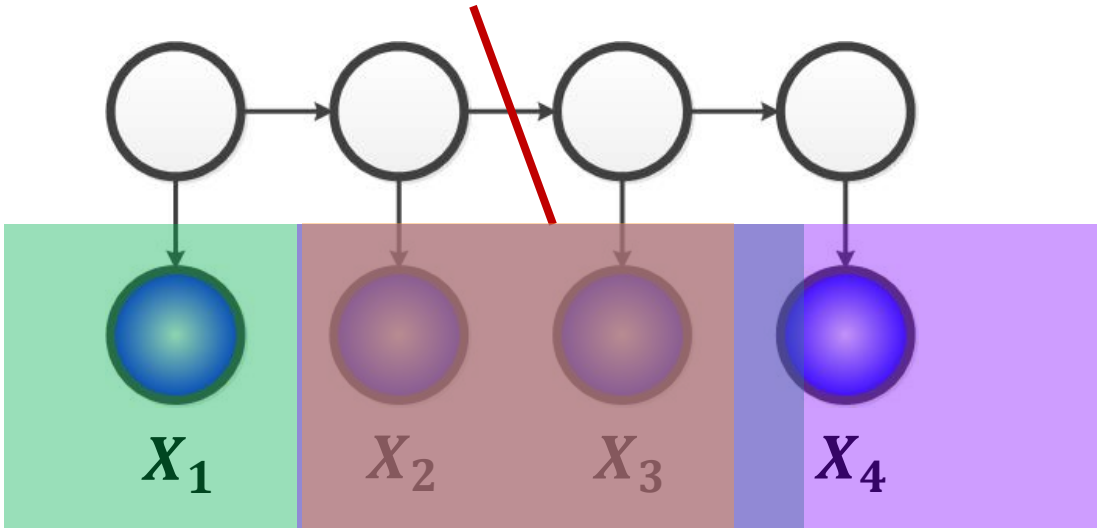
Advantage: Factors are only functions of observed variables! Can be directly computed from data without EM!!!!

Caveat: Factors are no longer probability tables (do not have to be non-negative)

We will call this factorization the observable factorization.

Graphical Relationship

$$\mathcal{P}[X_{\{1,2\}}, X_{\{3,4\}}] = \mathcal{P}[X_{\{1,2\}}, X_3]\mathcal{P}[X_2, X_3]^{-1}\mathcal{P}[X_2, X_{\{3,4\}}]$$



Our Alternate Factorization

$$\mathcal{P}[X_{\{1,2\}}, X_{\{3,4\}}] = \mathcal{P}[X_{\{1,2\}}, X_3]\mathcal{P}[X_2, X_3]^{-1}\mathcal{P}[X_2, X_{\{3,4\}}]$$

factor of 4 variables **factor of 3 variables** **factor of 4 variables**

- It may not seem very amazing at the moment (we have only reduced the size of the factor by 1)
- What is cool is that every latent tree of N variables has such a factorization where:
  - All factors are of size 3 (for simplicity we only consider binary trees)
  - All factors are only functions of observed variables
- We will prove why in a few slides.

Where's the Catch?

- Before we said that if the number of latent states was very large then the model was equivalent to a clique.

- Where does that scenario enter in our factorization?

$$\boldsymbol{\mathcal{P}}[X_{\{1,2\}}, X_{\{3,4\}}] = \boldsymbol{\mathcal{P}}[X_{\{1,2\}}, X_3]\boldsymbol{\mathcal{P}}[X_2, X_3]^{-1}\boldsymbol{\mathcal{P}}[X_2, X_{\{3,4\}}]$$

When does this inverse exist?

When Does the Inverse Exist

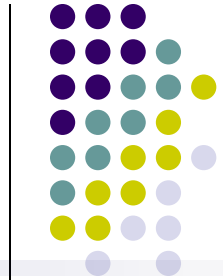
$$\boldsymbol{\mathcal{P}}[X_2, X_3] = \boldsymbol{\mathcal{P}}[X_2|H_2]\boldsymbol{\mathcal{P}}[\oslash H_2]\boldsymbol{\mathcal{P}}[X_3|H_2]^\top$$

- All the matrices on the right hand side must have full rank. (This is in general a requirement of spectral learning, although it can be somewhat relaxed, see Siddiqi et al. 2009)

- If ***m*** (number of observed states) > ***k*** (number of hidden states), then the inverse cannot exist, but this situation is easily fixable (project onto lower dimensional space)

- But what happens when ***k*** **>** ***m***?

When k > m

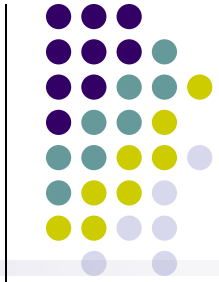


- The inverse does exist. But it no longer satisfies the following property, which we used to derive the factorization

$$\boldsymbol{\mathcal{P}}[X_2, X_3]^{-1} = (\boldsymbol{\mathcal{P}}[X_3|H_2]^{\top})^{-1}\boldsymbol{\mathcal{P}}[\oslash H_2]^{-1}\boldsymbol{\mathcal{P}}[X_2|H_2]^{-1}$$

- This is much more difficult to fix, and intuitively corresponds to how the problem becomes intractable if ***k >> m***.

Relationship to Original Factorization



- What is the relationship between the original factorization and the new factorization?

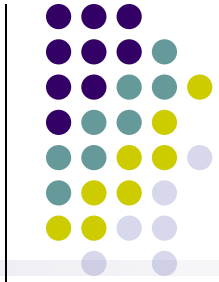
$$\underbrace{\boldsymbol{\mathcal{P}}[X_{\{1,2\}}, X_{\{3,4\}}]}_{\boldsymbol{M}} = \underbrace{\boldsymbol{\mathcal{P}}[X_{\{1,2\}}|H_2]\boldsymbol{\mathcal{P}}[\oslash H_2]}_{\boldsymbol{R}}\underbrace{\boldsymbol{\mathcal{P}}[X_{\{3,4\}}|H_2]^\top}_{\boldsymbol{L}}$$

$$\boldsymbol{M} = \boldsymbol{R}\boldsymbol{L}$$

$$\boldsymbol{M} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^{-1}\boldsymbol{L}$$

Can I choose S to get the observable factorization?

Relationship to Original Factorization

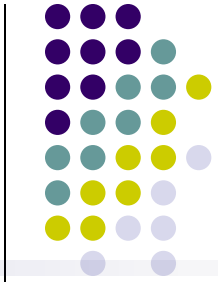


- Let

$$\boldsymbol{S} := \boldsymbol{\mathcal{P}}[X_3 | H_2]$$

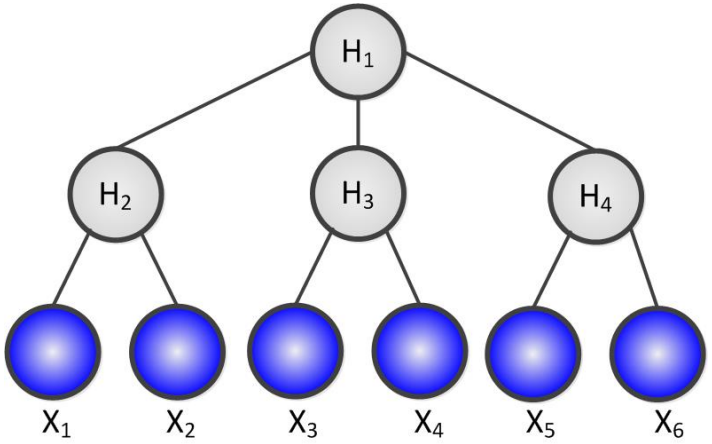
$$\boldsymbol{\mathcal{P}}[X_{\{1,2\}}, X_{\{3,4\}}] = \underbrace{\boldsymbol{\mathcal{P}}[X_{\{1,2\}}, X_3]}_{= \boldsymbol{L}\boldsymbol{S}} \underbrace{\boldsymbol{\mathcal{P}}[X_2, X_3]^{-1} \boldsymbol{\mathcal{P}}[X_2, X_{\{3,4\}}]}_{= \boldsymbol{S}^{-1}\boldsymbol{R}}$$

Overview of Spectral Learning Magic



- Represent original latent model as a decomposition of matrices (tensors in our case)
- Derive an alternate factorization that is only a function of observed variables (which we term the observable factorization).
- Learn parameters for alternate factorization directly from the data without using EM!
  - Local minima free and provably consistent.
- **For simplicity, assume number of hidden states is equal to the number of observed states and all CPTs have full rank.**

The Marginal Does Not Factorize



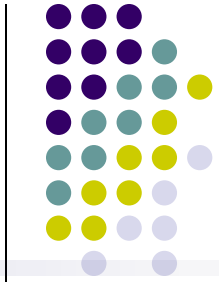
$$\mathbb{P}[X_1, X_2, X_3, X_4, X_5, X_6] = \sum_{H_1,\ldots,H_6} \prod_{i=2}^{4} \mathbb{P}[H_i | H_{\pi(H_i)}] \prod_{i=1}^{6} \mathbb{P}[X_i | H_{\pi(X_i)}]$$

Does not factorize due to the outer sum. Can somewhat distribute the sum.

$$\mathbb{P}[X_1, X_2, X_3, X_4, X_5, X_6] = \sum_{H_1} \mathbb{P}[H_1] \prod_{i=2}^{4} \sum_{H_i} \mathbb{P}[H_i | H_{\pi(H_i)}] \prod_{i=1}^{6} \sum_{X_i} \mathbb{P}[X_i | H_{\pi(X_i)}]$$

Sum product reminds us of matrix multiplication

Matrices are Insufficient



- Matrix factorization

$$\boldsymbol{M = RL}$$

- For example, we can do the following:

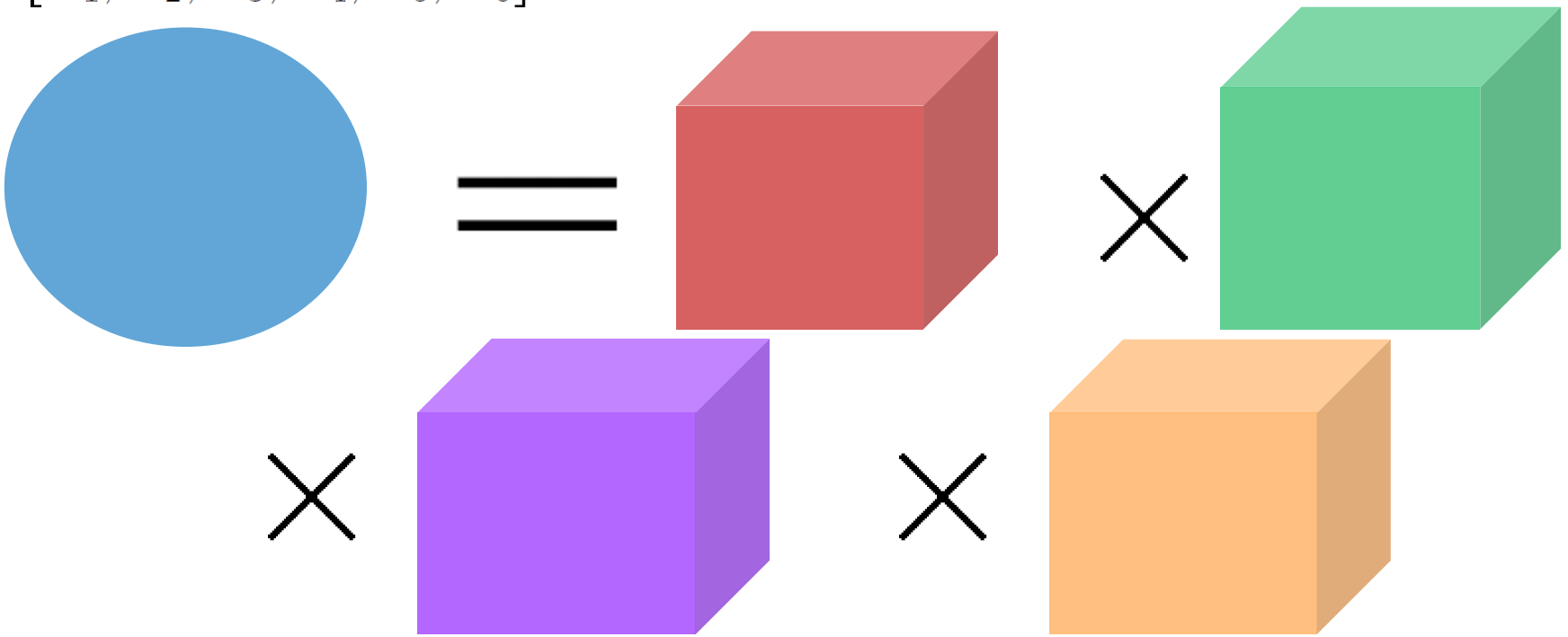
$$\boldsymbol{\mathcal{P}}[X_{\{1,2,3,4\}}, X_{\{5,6\}}] = \boldsymbol{\mathcal{P}}[X_{\{1,2,3,4\}}|H_1]\boldsymbol{\mathcal{P}}[\oslash H_1]\boldsymbol{\mathcal{P}}[X_{\{5,6\}}|H_1]^\top$$

- But then there is nothing more we can do.

Tensor Factorization

- Express higher order tensors as products of lower order tensors.

$$\mathcal{P}[X_1, X_2, X_3, X_4, X_5, X_6]$$

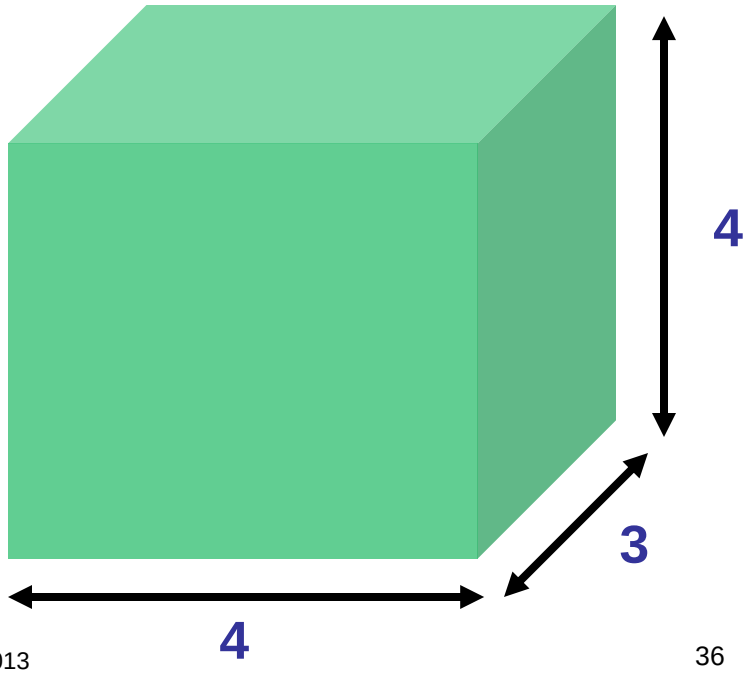


Tensors

- Multidimensional arrays
- A Tensor of order ***N*** has ***N*** modes (***N*** indices):

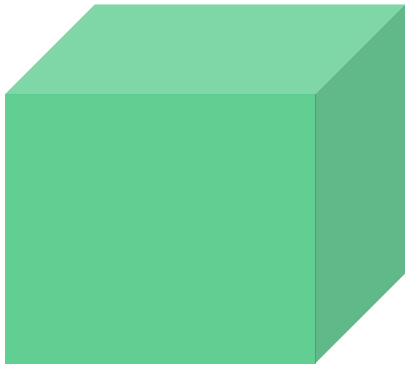
$$\mathcal{T}(i_1, \ldots, i_N)$$

- Each mode is associated with a dimension. In the example,
  - Dimension of mode 1 is 4
  - Dimension of mode 2 is 3
  - Dimension of mode 3 is 4



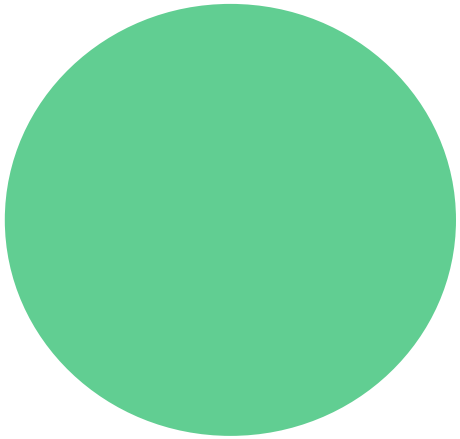
Tensors

1st order tensor (vector)



3rd order tensor (cube)

2nd order tensor (matrix)

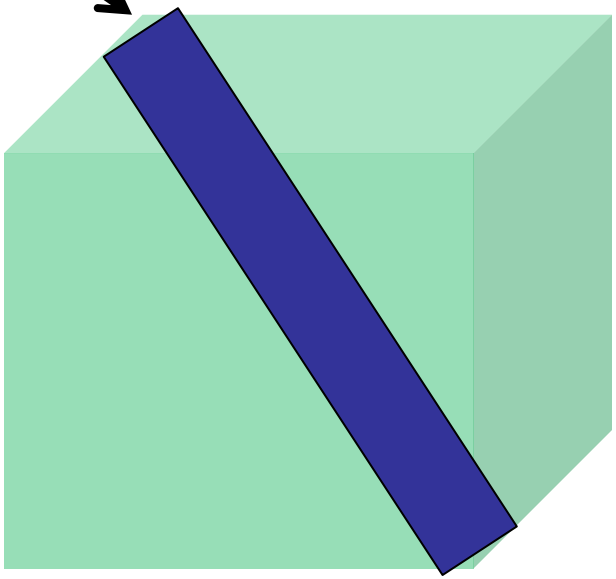


Higher order tensors

Diagonal Tensors

$$\mathcal{P}[X]$$

$$\mathcal{T}(i,j,k) = \begin{cases} \mathbb{P}[X=i] & \text{if } i=j=k \\ 0 & \text{otherwise} \end{cases}$$

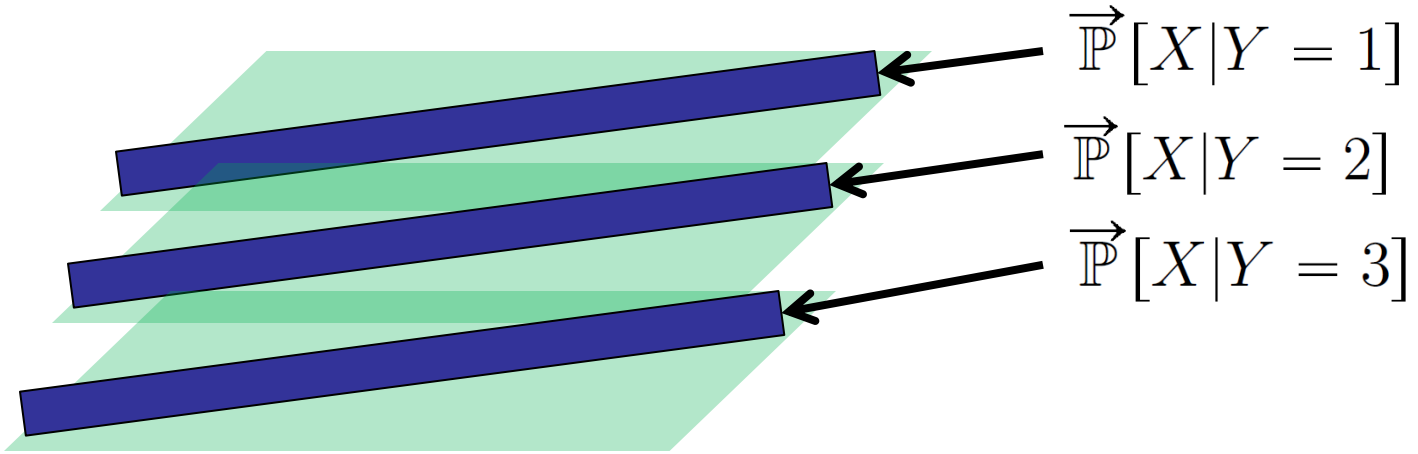


Notation for Diagonal Third Order Tensors:

$$\mathcal{P}[\oslash_3 X]$$

Partially Diagonal Tensors

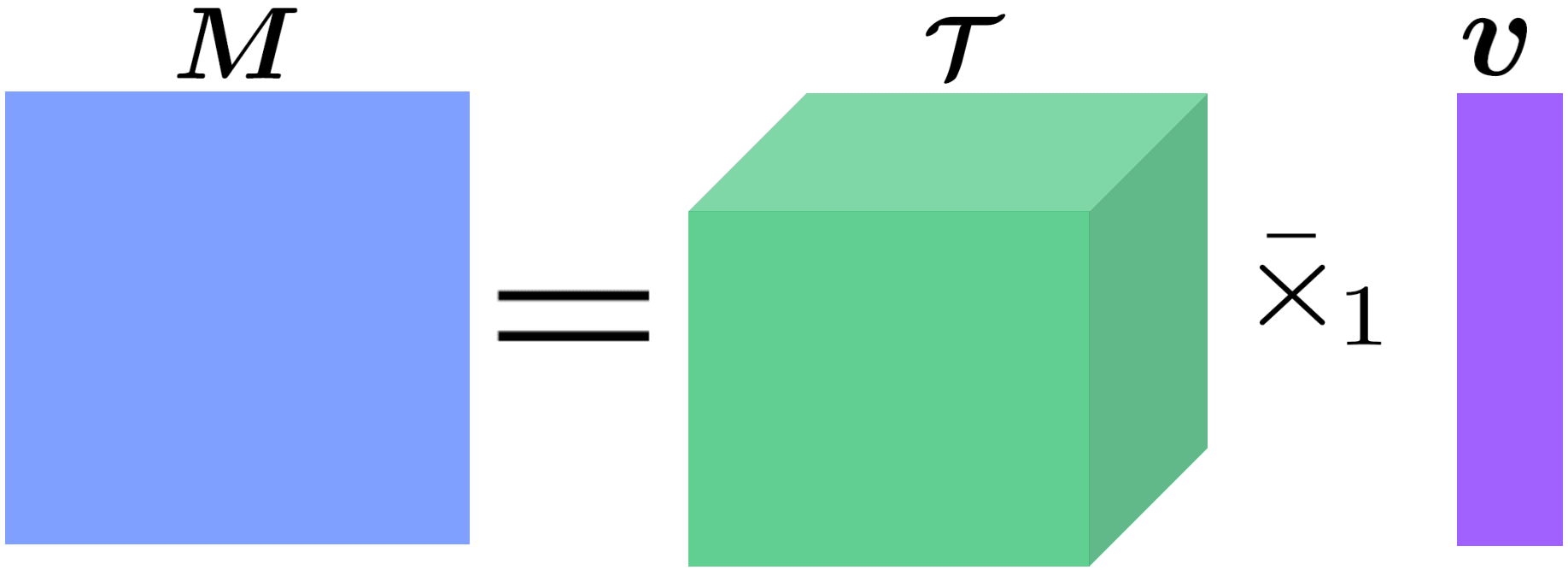
$$\boldsymbol{\tau}(i,j,k) = \begin{cases} \mathbb{P}[X = i | Y = k] & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$



Notation: $\boldsymbol{\mathcal{P}}[\oslash X | Y]$

Tensor Vector Multiplication

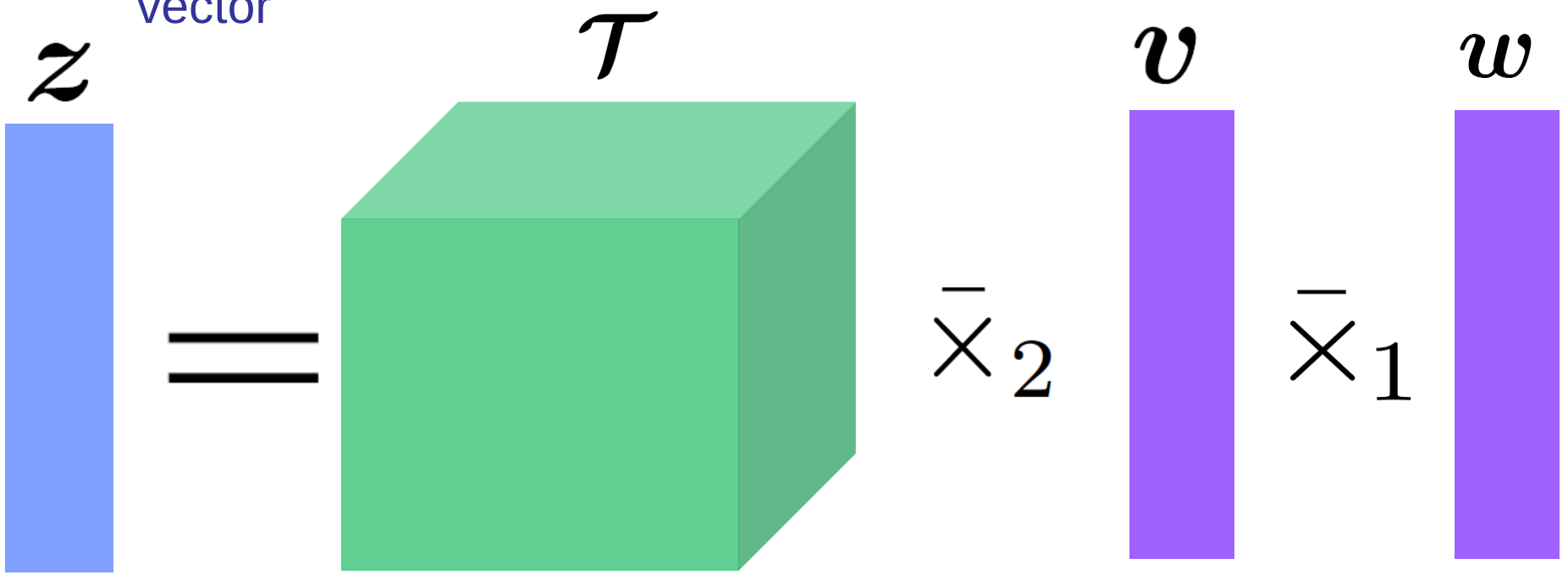
- Multiplying a 3rd order tensor by a vector produces a matrix



$$M(j,k) = \sum_i \mathcal{T}(i,j,k)v(i)$$

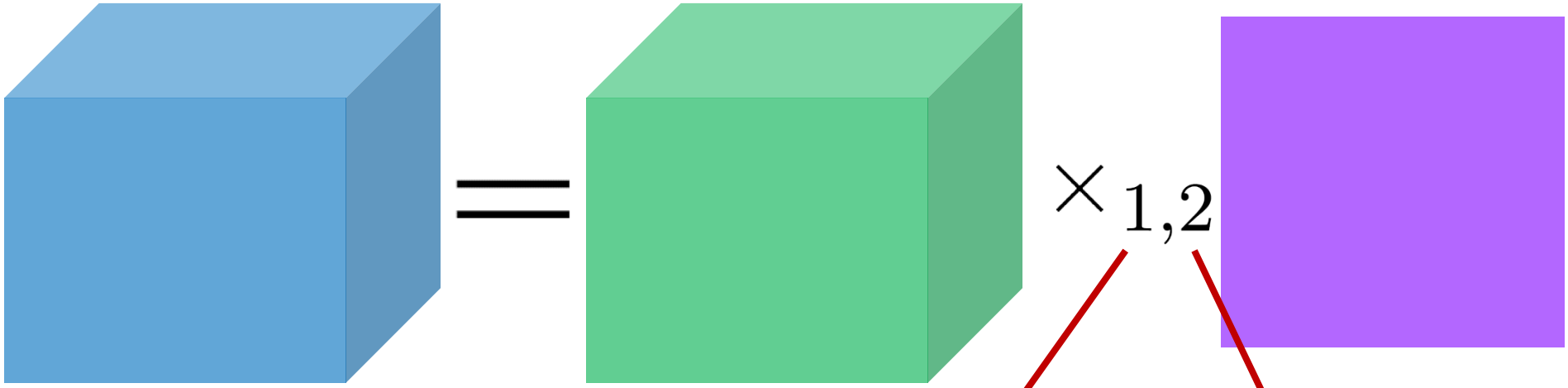
Tensor Vector Multiplication Cont.

- Multiplying a 3rd order tensor by two vectors produces a vector



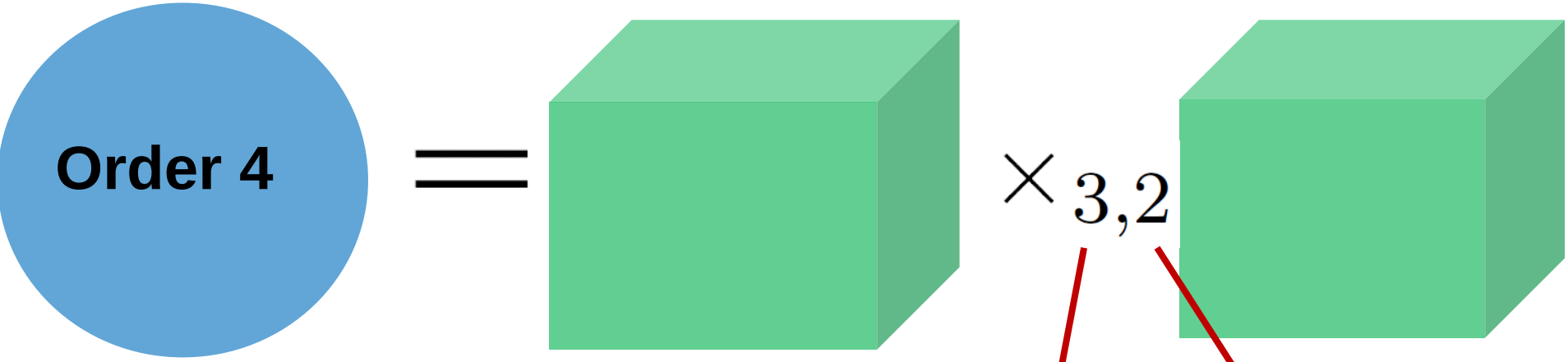
$$\boldsymbol{z}(k) = \sum_i \left(\sum_j \boldsymbol{\mathcal{T}}(i,j,k)\boldsymbol{v}(i) \right) \boldsymbol{w}(j) = \sum_{i,j} \boldsymbol{\mathcal{T}}(i,j,k)\boldsymbol{v}(i)\boldsymbol{w}(j)$$

Tensor Matrix Multiplication



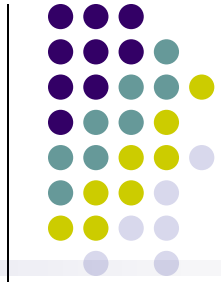
$$\boldsymbol{R}(i,j,k) = \sum_{m} \boldsymbol{\mathcal{T}}(m,j,k)\boldsymbol{M}(i,m)$$

Tensor Tensor Multiplication



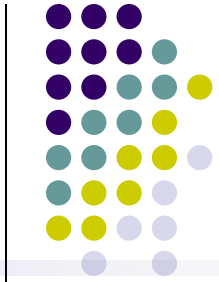
$$\boldsymbol{R}(i,j,k,l) = \sum_{m} \boldsymbol{\mathcal{T}}_1(i,j,m)\boldsymbol{\mathcal{T}}_2(k,m,l)$$

Tensor Tensor Multiplication



- Matrix multiplication is closed under multiplication (i.e. product of matrices is a matrix).
- Tensor multiplication is not closed under multiplication (e.g. product of two tensors of order ***n*** is a tensor of order ***2n-2***)
- **This is key to developing the factorization - multiplying lower order tensors will result in a higher order tensor**

Cleaner Notation



- This is clunky

$$\mathcal{P}[X_1, X_2, X_3] = \mathcal{P}[X_1, X_2, H_1] \times_{3,2} \mathcal{P}[X_3|H_1]$$

- Instead use variable names instead of modes

$$\mathcal{P}[X_1, X_2, X_3] = \mathcal{P}[X_1, X_2, H_1] \times_{H_1} \mathcal{P}[X_3|H_1]$$

The Matricization View

- You can also interpret tensor-tensor (or tensor-matrix) multiplication as first matricizing the tensors, and performing normal matrix multiplication and then reshaping back into a tensor.

- Example:

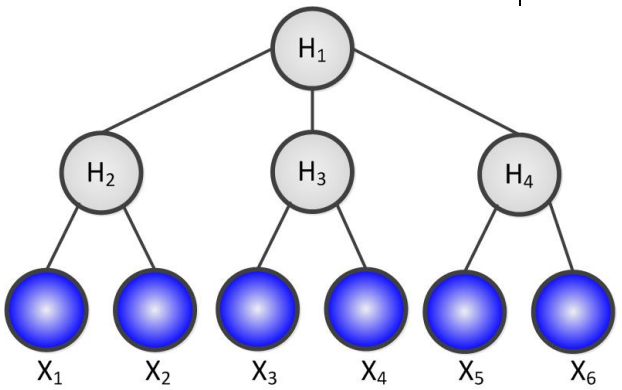
$$\mathcal{P}[X_1, X_2, X_3] = \mathcal{P}[X_1, X_2, H_1] \times_{H_1} \mathcal{P}[X_3|H_1]$$

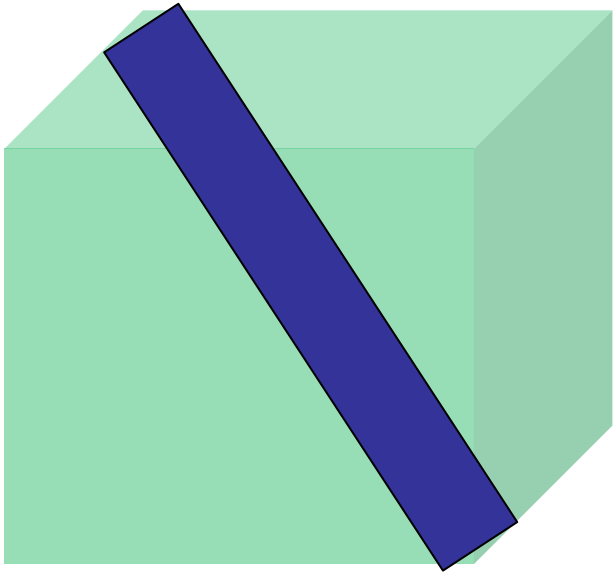
$$\mathcal{P}[X_{\{1,2\}}, X_3] = \mathcal{P}[X_{\{1,2\}}, H_1]\mathcal{P}[X_3|H_1]^\top$$

Can reshape this back into a tensor

The Tensor Form of The Root

$$\mathcal{P}(\oslash_3 H_1)$$



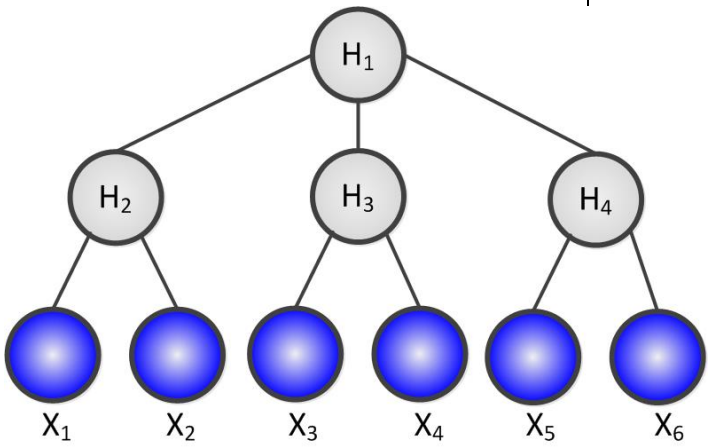


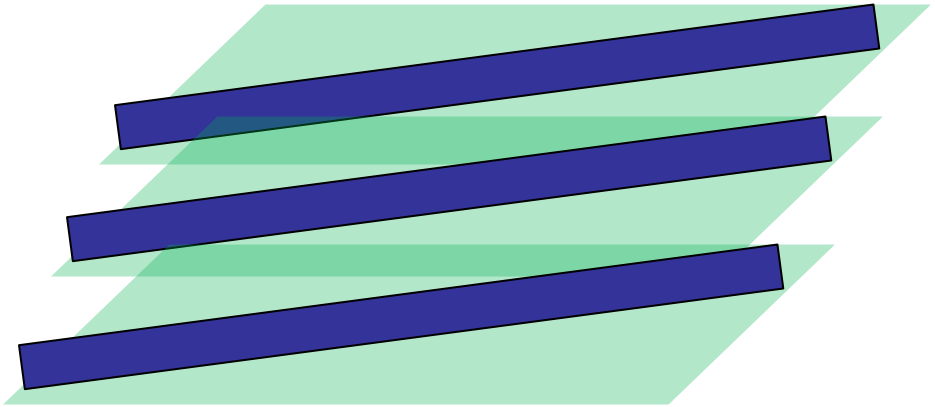
The Tensor Form of the Internal Nodes

$$\mathcal{P}(\oslash H_2|H_1)$$

$$\mathcal{P}(\oslash H_3|H_1)$$

$$\mathcal{P}(\oslash H_4|H_1)$$





The Tensor Form of the Factorization-Leaves

$\mathcal{P}(X_1|H_2)$

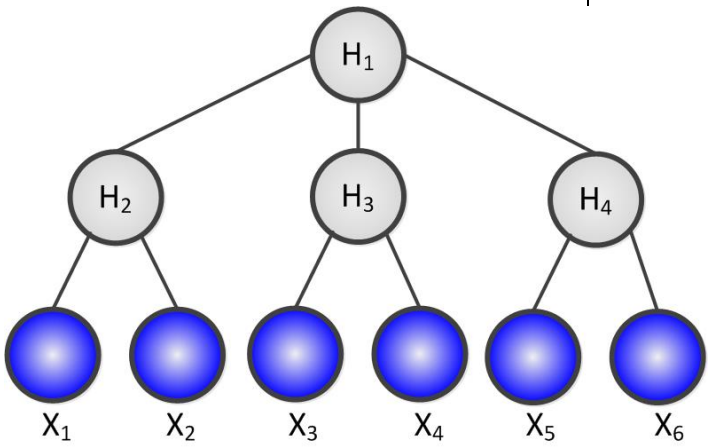
$\mathcal{P}(X_2|H_2)$

$\mathcal{P}(X_3|H_3)$

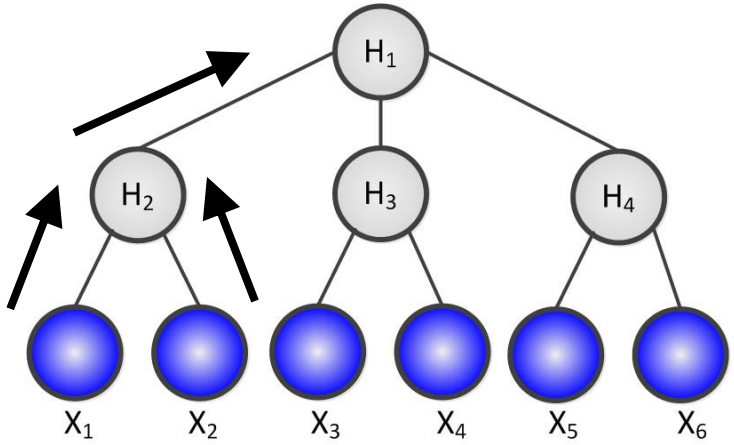
$\mathcal{P}(X_4|H_3)$

$\mathcal{P}(X_5|H_4)$

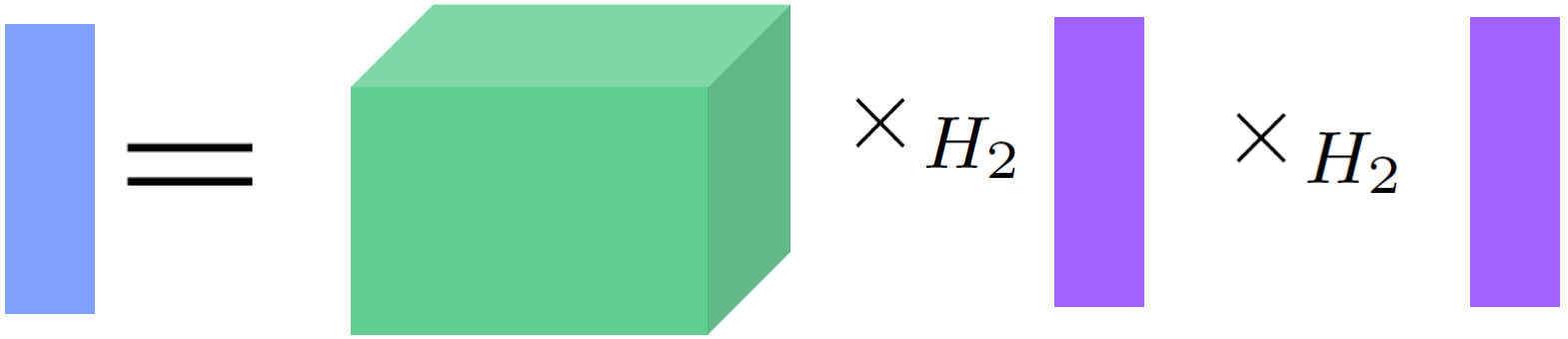
$\mathcal{P}(X_6|H_4)$



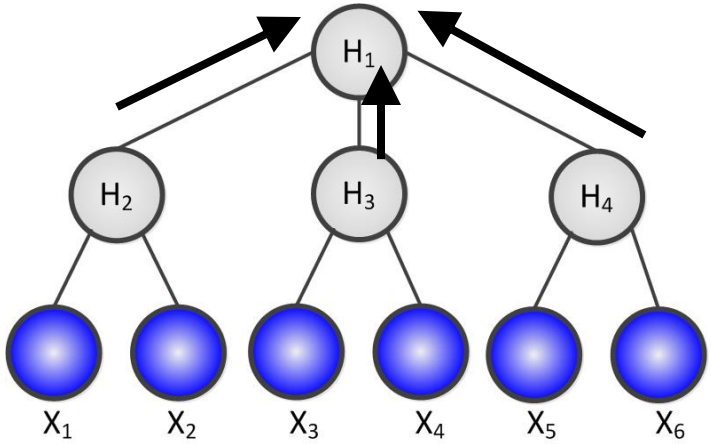
Tensor Message Passing



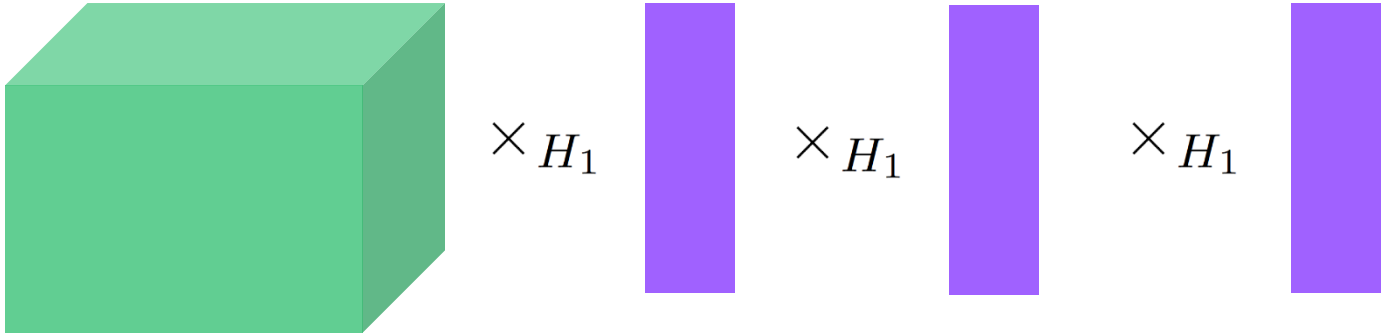
$\mathcal{P}[x_1, x_2|H_1]$ $\mathcal{P}(\oslash H_2|H_1)$ $\mathcal{P}[x_1|H_2]$ $\mathcal{P}[x_2|H_2]$



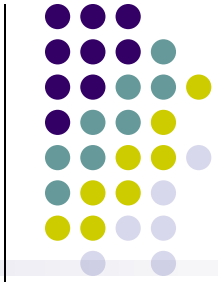
At the Root



$$\mathbb{P}[\text{evidence}] = \mathcal{P}(\oslash_3 H_1) \times_{H_1} \mathcal{P}[x_1, x_2|H_1] \times_{H_1} \mathcal{P}[x_3, x_4|H_1] \times_{H_1} \mathcal{P}[x_5, x_6|H_1]$$



To Compute the Marginal Tensor

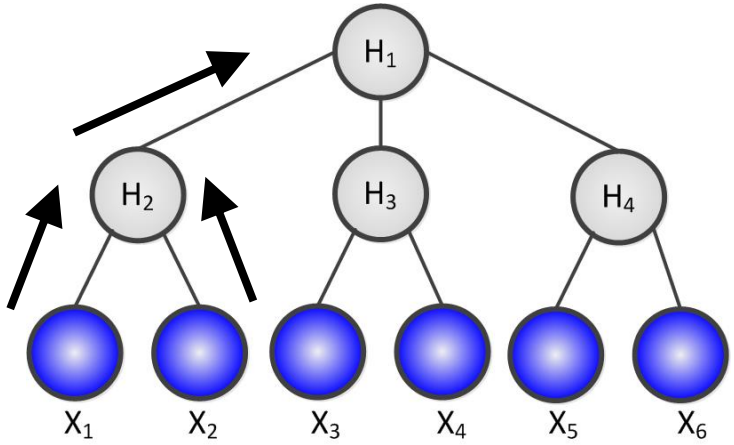


- What if we want to compute the whole marginal tensor?

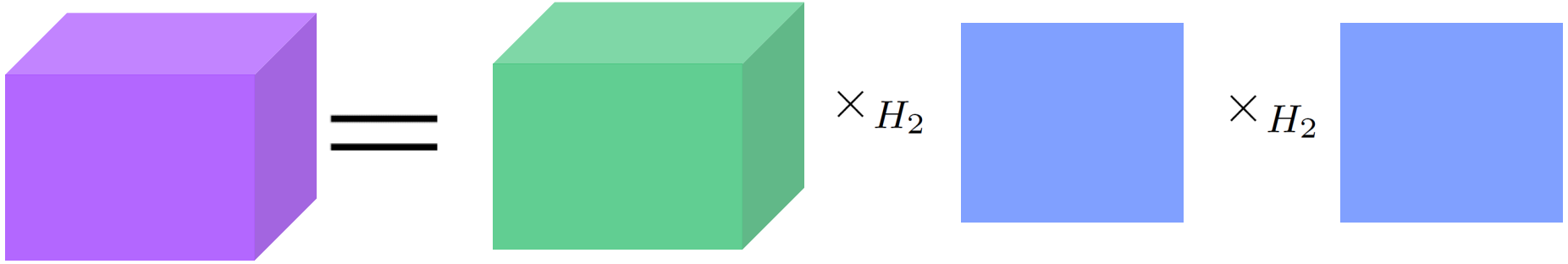
$$\mathcal{P}[X_1, X_2, X_3, X_4, X_5, X_6]$$

- Run message passing, except don't incorporate evidence at the leaves

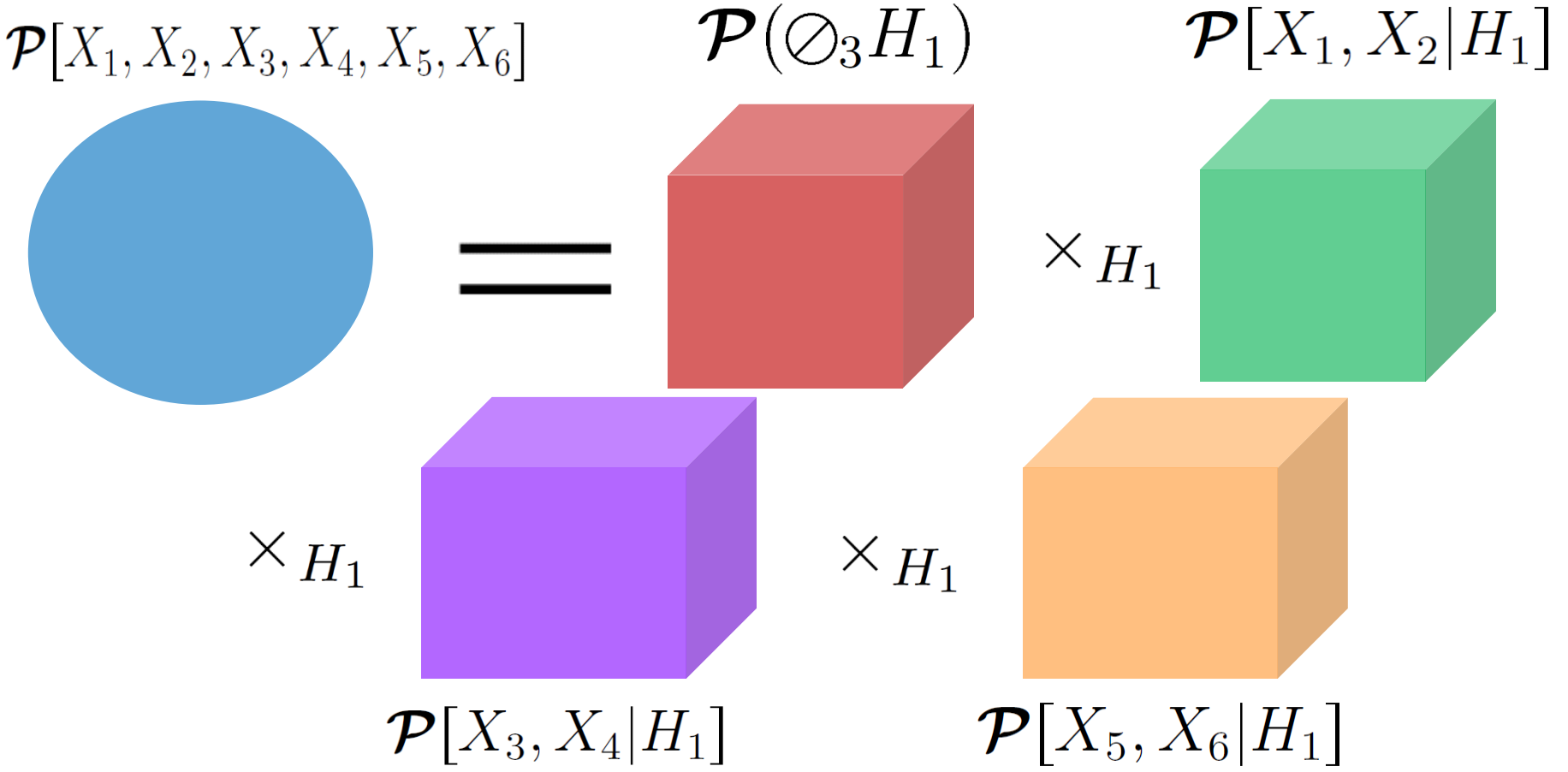
To Compute the Marginal Tensor



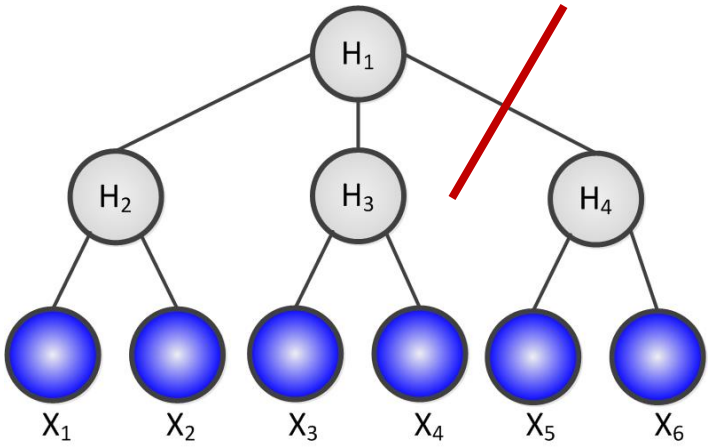
$$\mathcal{P}[X_1, X_2|H_1] \quad \mathcal{P}(\oslash H_2|H_1) \quad \mathcal{P}(X_1|H_2) \quad \mathcal{P}(X_2|H_2)$$



At the Root


$\mathcal{P}[X_1, X_2, X_3, X_4, X_5, X_6]$
$\mathcal{P}(\oslash_3 H_1)$
$\mathcal{P}[X_1, X_2|H_1]$
$\times_{H_1}$
$\times_{H_1}$
$\times_{H_1}$
$\mathcal{P}[X_3, X_4|H_1]$
$\mathcal{P}[X_5, X_6|H_1]$

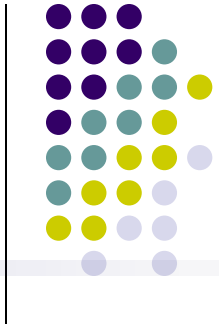
Constructing a Different Factorization



Like we did before

$$\boldsymbol{\mathcal{P}}[X_{\{1,2,3,4\}}, X_{\{5,6\}}] = \boldsymbol{\mathcal{P}}[X_{\{1,2,3,4\}}, X_5]\boldsymbol{\mathcal{P}}[X_4, X_5]^{-1}\boldsymbol{\mathcal{P}}[X_4, X_{\{5,6\}}]$$

The Tensor View



“Matricized” Way

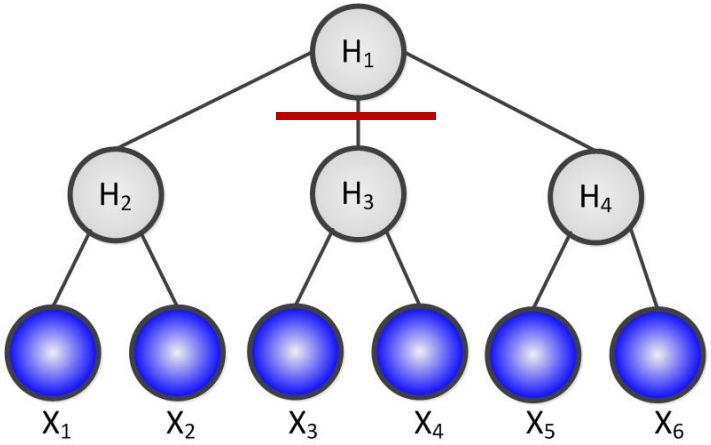
$$\mathcal{P}[X_{\{1,2,3,4\}}, X_{\{5,6\}}] = \mathcal{P}[X_{\{1,2,3,4\}}, X_5]\mathcal{P}[X_4, X_5]^{-1}\mathcal{P}[X_4, X_{\{5,6\}}]$$

“Tensor” Way

$$\begin{aligned} & \mathcal{P}[X_1, X_2, X_3, X_4, X_5, X_6] \\ = \; & \mathcal{P}[X_1, X_2, X_3, X_4, X_5] \times_{X_5} \mathcal{P}[X_4, X_5]^{-1} \times_{X_4} \mathcal{P}[X_4, X_5, X_6] \end{aligned}$$

Decompose this recursively

Constructing a Different Factorization

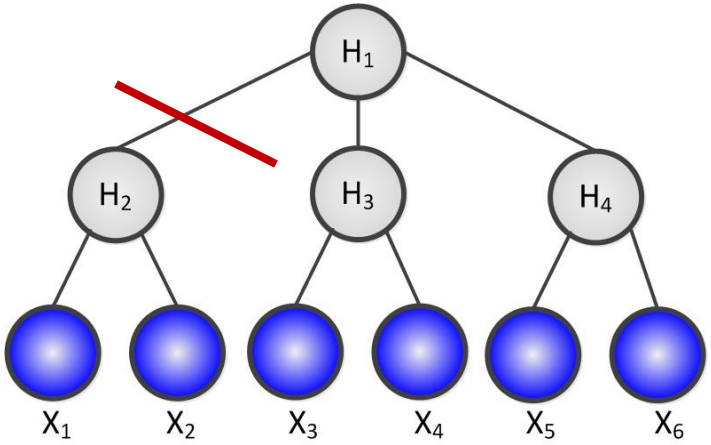


$$\mathcal{P}[X_{\{1,2,5\}}, X_{\{3,4\}}] = \mathcal{P}[X_{\{1,2,5\}}, X_3]\mathcal{P}[X_5, X_3]^{-1}\mathcal{P}[X_5, X_{\{3,4\}}]$$

$$\mathcal{P}[X_1, X_2, X_3, X_4, X_5]$$
$$= \mathcal{P}[X_1, X_2, X_3, X_5] \times_{X_3} \mathcal{P}[X_5, X_3]^{-1} \times_{X_5} \mathcal{P}[X_3, X_4, X_5]$$

Decompose this recursively

Constructing a Different Factorization



$$\mathcal{P}[X_{\{1,2\}}, X_{\{3,5\}}] = \mathcal{P}[X_{\{1,2\}}, X_3]\mathcal{P}[X_1, X_3]^{-1}\mathcal{P}[X_1, X_{\{3,5\}}]$$

$$\begin{aligned} & \mathcal{P}[X_1, X_2, X_3, X_5] \\ = \; & \mathcal{P}[X_1, X_2, X_3] \times_{X_3} \mathcal{P}[X_1, X_3]^{-1} \times_{X_1} \mathcal{P}[X_1, X_3, X_5] \end{aligned}$$

Our Observable Factorization

$$\mathcal{P}[X_1, X_2, X_3, X_4, X_5, X_6]$$

$$= \mathcal{P}[X_1, X_2, X_3, X_4, X_5] \times_{X_5} \mathcal{P}[X_4, X_5]^{-1} \times_{X_4} \mathcal{P}[X_4, X_5, X_6]$$

$$\mathcal{P}[X_1, X_2, X_3, X_4, X_5]$$

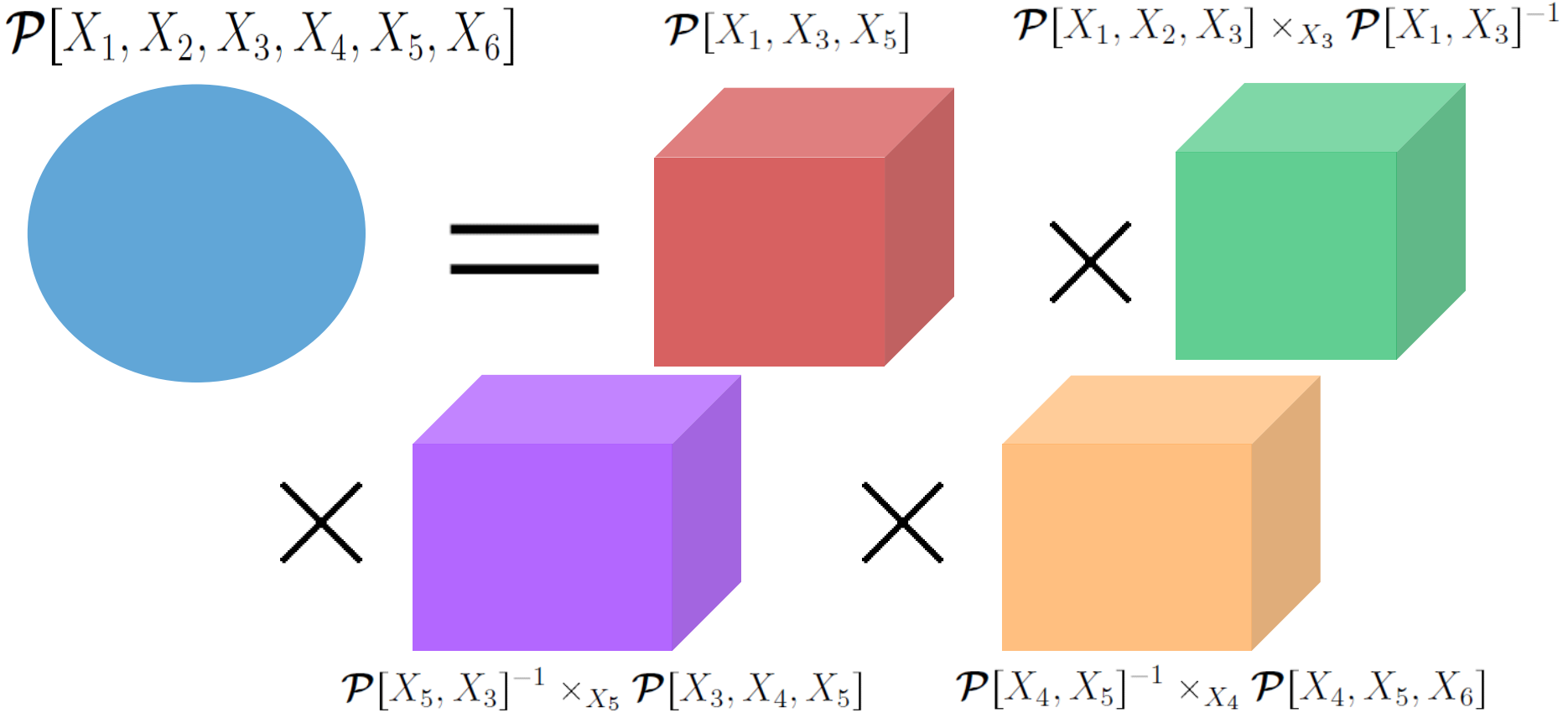
$$= \mathcal{P}[X_1, X_2, X_3, X_5] \times_{X_3} \mathcal{P}[X_5, X_3]^{-1} \times_{X_5} \mathcal{P}[X_3, X_4, X_5]$$

$$\mathcal{P}[X_1, X_2, X_3, X_5]$$

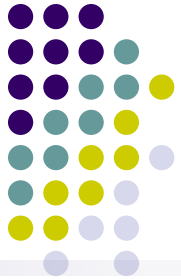
$$= \mathcal{P}[X_1, X_2, X_3] \times_{X_3} \mathcal{P}[X_1, X_3]^{-1} \times_{X_1} \mathcal{P}[X_1, X_3, X_5]$$

All Third Order Tensors

Observable Factorization



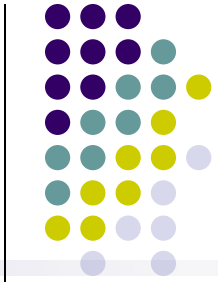
Can We Decompose the Third Order Tensors Further?



$$\mathcal{P}[X_1, X_3, X_5] \quad ?$$

Not to lower order tensors. Multiplying second order tensors doesn't increase the tensor order.

The Observable Factorization



Root

$$\mathcal{P}(\oslash_3 H_1) \rightarrow \mathcal{P}[X_1, X_3, X_5]$$

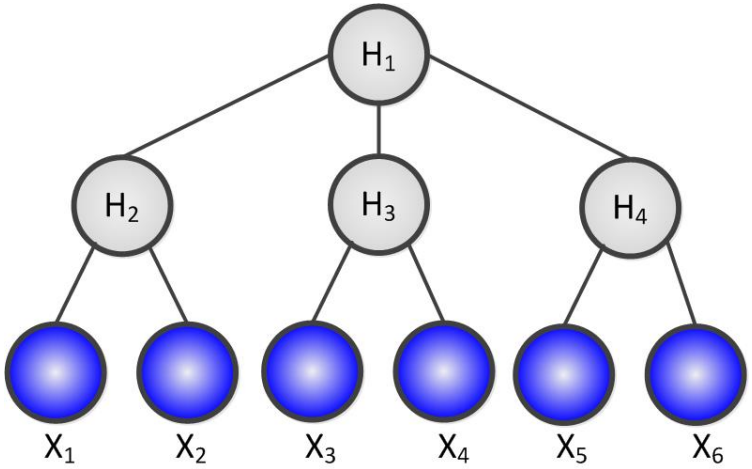
Internal Nodes (Example)

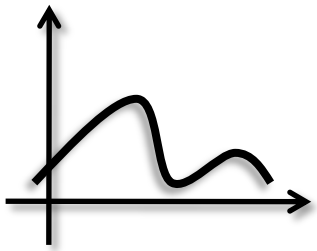
$$\mathcal{P}(\oslash H_2 | H_1) \rightarrow \mathcal{P}[X_1, X_2, X_3] \times_{X_3} \mathcal{P}[X_1, X_3]^{-1}$$

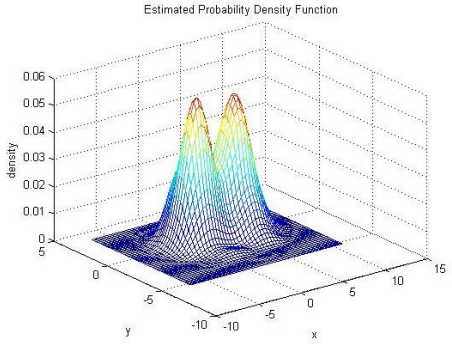
Leaves (Example)

$$\mathcal{P}(X_1 | H_2) \rightarrow$$ **Identity Matrix**

How To Deal with (Nonparametric) Continuous Variables?







Hilbert Space Embeddings!!!!

How To Deal with (Nonparametric) Continuous Variables?

Root

$$\mathcal{P}(\oslash_3 H_1) \rightarrow \mathcal{C}_{1,3,5}$$

Internal Nodes (Example)

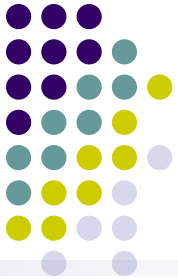
$$\mathcal{P}(\oslash H_2 | H_1) \rightarrow \mathcal{C}_{1,2,3} \times_3 \mathcal{C}_{1,3}^{-1}$$

Leaves (Example)

$$\mathcal{P}(X_1 | H_2) \rightarrow$$ **Identity Operator**

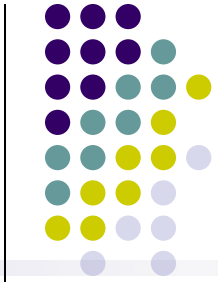
(something very similar to this is true)

Comparing the Factorizations

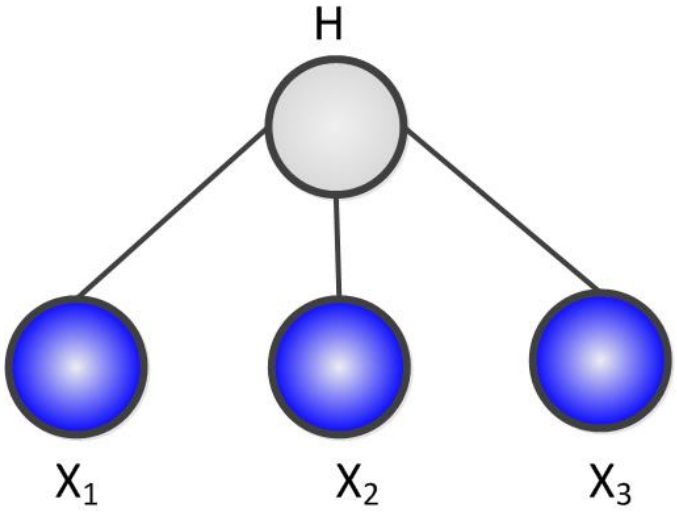


Traditional (CPT)	Observable
Requires EM to compute (local minima, slow)	Very fast, local minima free
No theoretical guarantees	Provably consistent
Aims to Find MLE	Less statistically efficient (does not aim to find MLE)
Allows for inference among latent variables	Does not allow for inference among latent variables

What If I Want to Extract the Actual Latent Parameters?



- Let's consider an even simpler example.



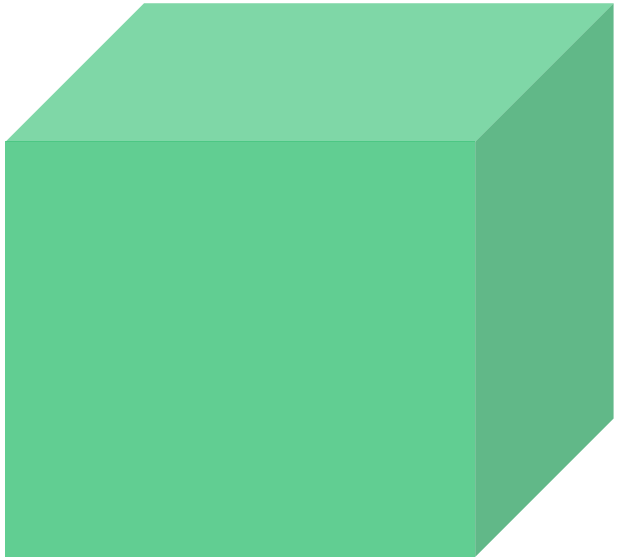
- Assume all the conditional probability tables are the same i.e.

$$\boldsymbol{O} := \boldsymbol{\mathcal{P}}[X_1|H] = \boldsymbol{\mathcal{P}}[X_2|H] = \boldsymbol{\mathcal{P}}[X_3|H]$$

Extracting Latent Parameters

- We only have three observed variables, so one observable parameterization is just their joint:

$$\mathcal{P}[X_3, X_2, X_1]$$



$\mathcal{P}[X_3, x_2, X_1]$ is a slice of the tensor

Extracting Latent Parameters

$$\mathcal{P}[X_3, x_2, X_1] = \boldsymbol{O}\,\mathrm{diag}(O_{x_2})P(\oslash H_1)\boldsymbol{O}^\top$$

I want to extract this....

Looks kind of like an eigenvalue decomposition, but not quite.....

$$\boldsymbol{M} = \boldsymbol{S}\Lambda\boldsymbol{S}^{-1}$$

What about the Related Quantity?

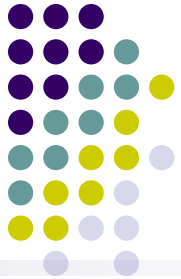
$$\boldsymbol{\mathcal{P}}[X_3, x_2, X_1] = \boldsymbol{O}\,\mathrm{diag}(O_{x_2})P(\oslash H_1)\boldsymbol{O}^\top$$

$$\boldsymbol{\mathcal{P}}[X_3, x_2, X_1] = \boldsymbol{O}\,\mathrm{diag}(O_{x_2})\boldsymbol{O}^{-1}\boldsymbol{O}P(\oslash H_1)\boldsymbol{O}^\top$$

This looks better!!!!

$$\boldsymbol{B}_{x_2} := \boldsymbol{O}\,\mathrm{diag}(O_{x_2})\boldsymbol{O}^{-1} = \boldsymbol{\mathcal{P}}[X_3, x_2, X_1]\boldsymbol{\mathcal{P}}[X_2, X_1]^{-1}$$

Eigenvalue Decomposition

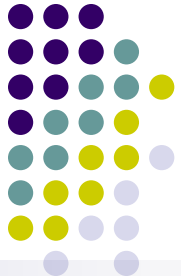


Run Eigenvalue Decomposition on this

$$\boldsymbol{B}_{x_2} := \boldsymbol{O}\,\mathrm{diag}(O_{x_2})\boldsymbol{O}^{-1} = \boldsymbol{\mathcal{P}}[X_3, x_2, X_1]\boldsymbol{\mathcal{P}}[X_2, X_1]^{-1}$$

Eigenvectors correspond to columns of $\boldsymbol{O}$
(need to be normalized)

Comments

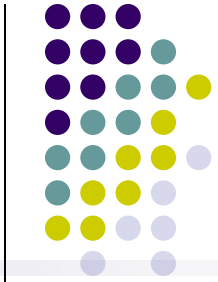


- Same technique is also the basis of recent advances in spectral LDA (Anandkumar et al. 2012)

- Not stable in practice in its naïve form.

- Anandkumar, Hsu, Kakade (2012) recently propose a tensor power iteration method that works better in practice (still current area of research)

Case Study: Supervised Parsing

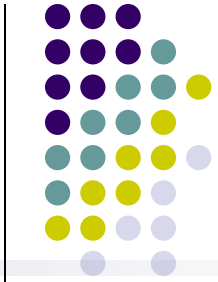


- Cohen et al. 2012 derived a spectral learning method for latent PCFGs.

- Latent PCFGs (Matsuzaki et al, Petrov et al.) are Probabilistic Context Free Grammars with additional latent nodes.

- The challenge in their setting is that the tree structure changes on every example.

Case Study: Supervised Parsing

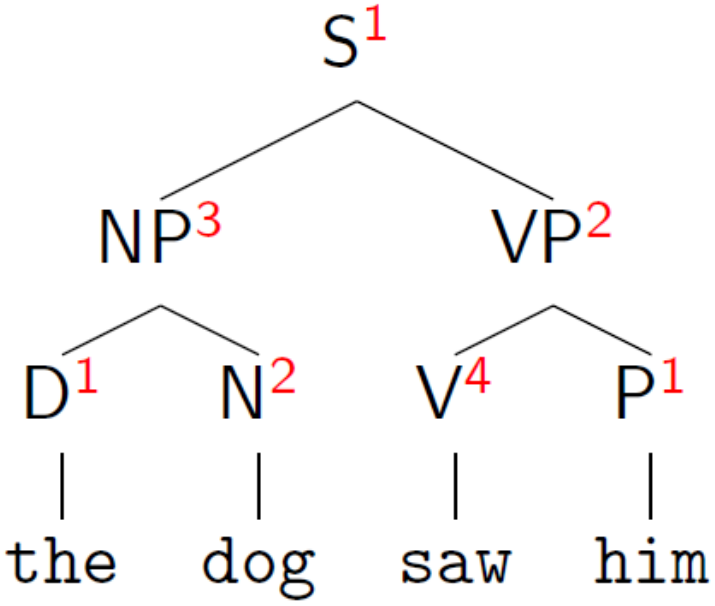


- Cohen et al. 2012 derived a spectral learning method for latent PCFGs.

- Latent PCFGs (Matsuzaki et al, Petrov et al.) are Probabilistic Context Free Grammars with additional latent nodes.

- The challenge in their setting is that the tree structure changes on every example.

Latent PCFG Supervised Parsing



$$
\begin{aligned}
& p(\text{tree}, 1\ 3\ 1\ 2\ 2\ 4\ 1) \\
&= \pi(\mathrm{S}^1) \times \\
&\quad t(\mathrm{S}^1 \rightarrow \mathrm{NP}^3\ \ \mathrm{VP}^2 | \mathrm{S}^1) \times \\
&\quad t(\mathrm{NP}^3 \rightarrow \mathrm{D}^1\ \ \mathrm{N}^2 | \mathrm{NP}^3) \times \\
&\quad t(\mathrm{VP}^2 \rightarrow \mathrm{V}^4\ \ \mathrm{P}^1 | \mathrm{VP}^2) \times \\
&\quad q(\mathrm{D}^1 \rightarrow \mathtt{the} | \mathrm{D}^1) \times \\
&\quad q(\mathrm{N}^2 \rightarrow \mathtt{dog} | \mathrm{N}^2) \times \\
&\quad q(\mathrm{V}^4 \rightarrow \mathtt{saw} | \mathrm{V}^4) \times \\
&\quad q(\mathrm{P}^1 \rightarrow \mathtt{him} | \mathrm{P}^1)
\end{aligned}
$$

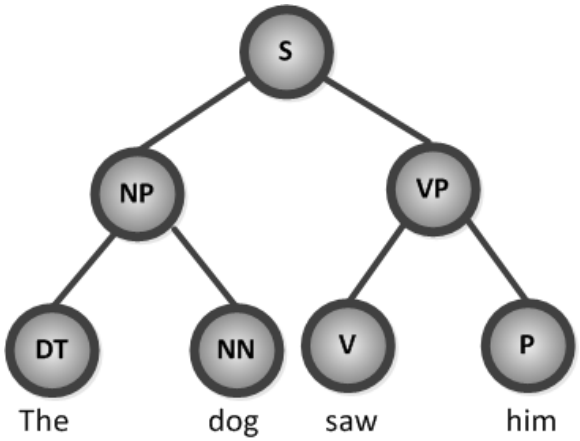
$$
p(\text{tree}) = \sum_{h_1 \ldots h_7} p(\textit{tree}, h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7)
$$

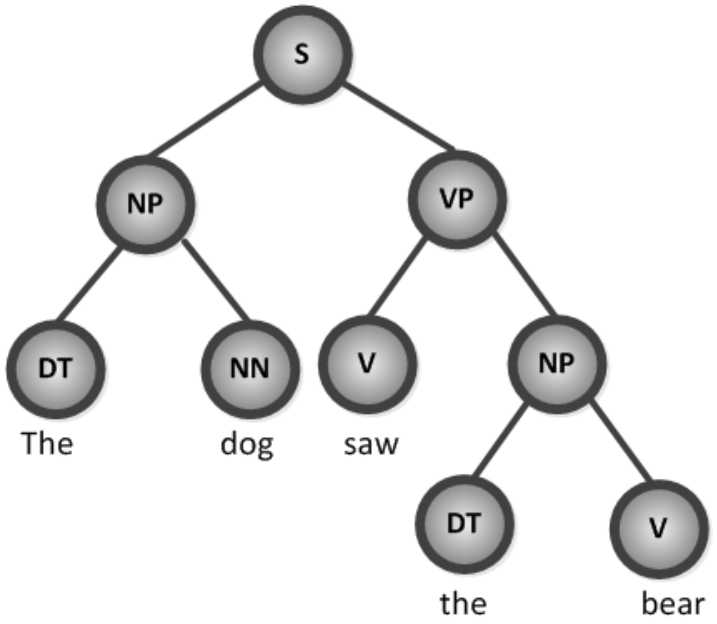
(from Karl Stratos ACL 2012 presentation)

Different Examples Have Different Trees

- If all examples had the same tree then the method would be very similar to spectral learning for trees.

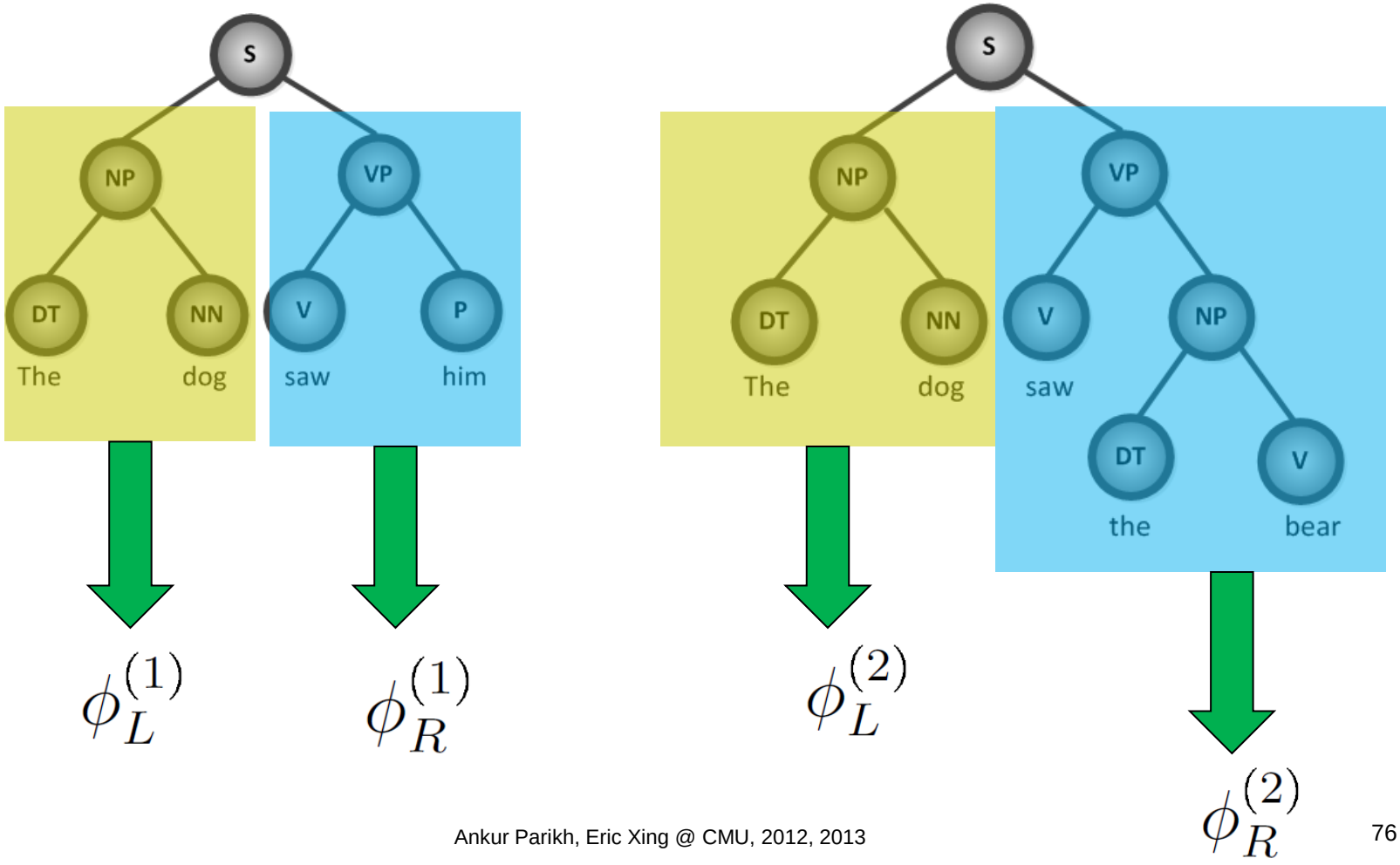
- However, different sentences have different parse trees.



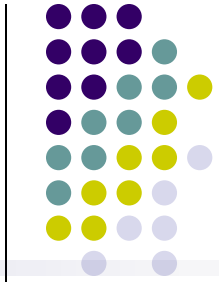


Construct Features of Subtrees

- Consider the root



Use Features of Subtrees



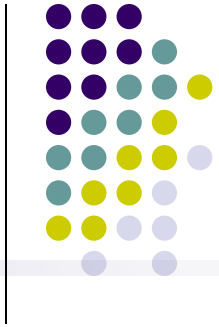
- In our previous case the parameter for the root looked something like this (one observation from each subtree):

$$\mathcal{P}[X_1, X_3, X_5]$$

- For Cohen et al. 2012, it will look something like

$$\mathbb{E}[\phi_L \phi_R^\top]$$

Empirical Results



- (Results are unpublished – will be presented at NAACL 2013).
- In general, the algorithm performs comparably with EM in accuracy.
- However, it is around 20x faster.

References

- Song, L., Ishteva, M. Parikh, A.P., Park, H. Xing, E.P. Under Review (2013)
- Mossel, E. and Roch, S. Learning nonsingular phylogenies and hidden markov models. Annals of Applied Probability, 16(2):583–614, 2006.
- Hsu, D., Kakade, S., and Zhang, T. A Spectral Algorithm for Learning Hidden Markov Models. Conference on Learning Theory, 2009.
- Siddiqi, S., Boots, B. Gordon, G., Reduced Rank Hidden Markov Models, Artificial Intelligence and Statistics (AISTATS), 2009.
- Parikh, A.P. Song, L., and Xing, E.P. A Spectral Algorithm for Latent Tree Graphical Models, International Conference of Machine Learning (ICML), 2011.
- Song, L., Boots, B., Siddiqi, S., Gordon, G., and Smola, A. Hilbert space embeddings of Hidden Markov Models. International Conference of Machine Learning (ICML), 2010..
- S. Cohen, K. Stratos, M. Collins, D. Foster, L. Ungar. Spectral Learning of Latent-Variable PCFGs. Association of Computational Linguistics (ACL) 2012.