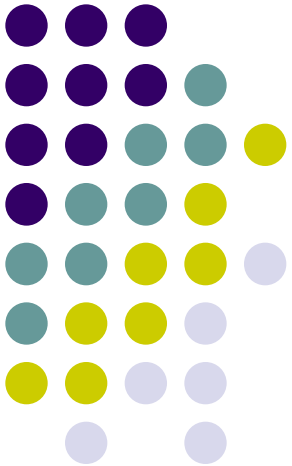
School of Computer Science
Carnegie Mellon

Spectral Learning for Graphical Models

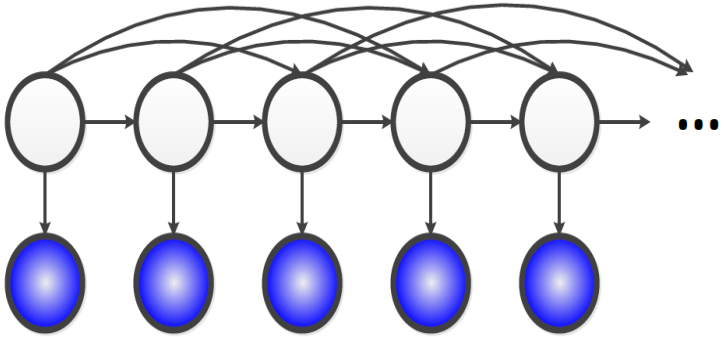
An Intuitive Introduction with a Focus on NLP

Ankur Parikh

July 11, 2013

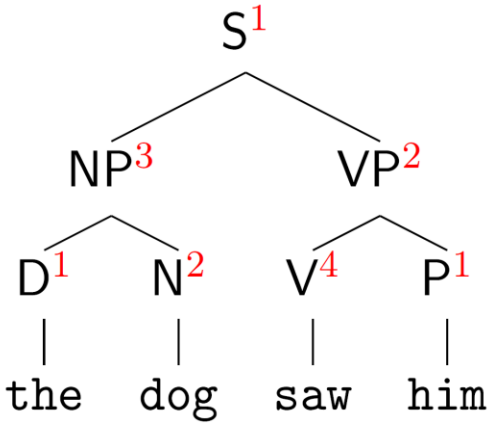


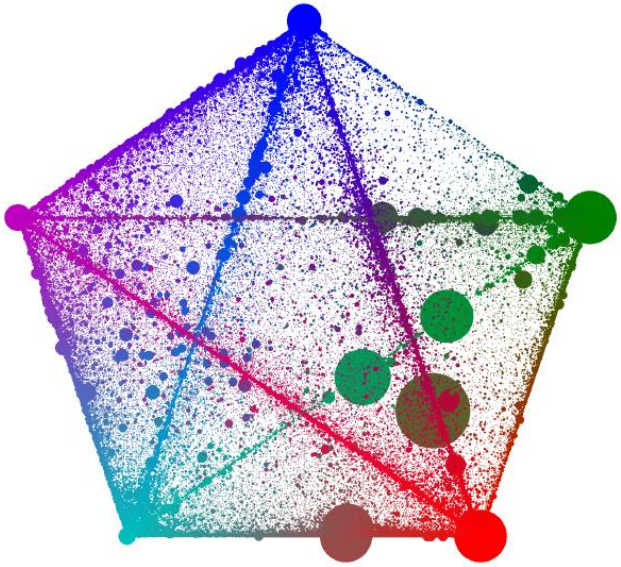
Latent Variable Models



Sequence models

Parsing

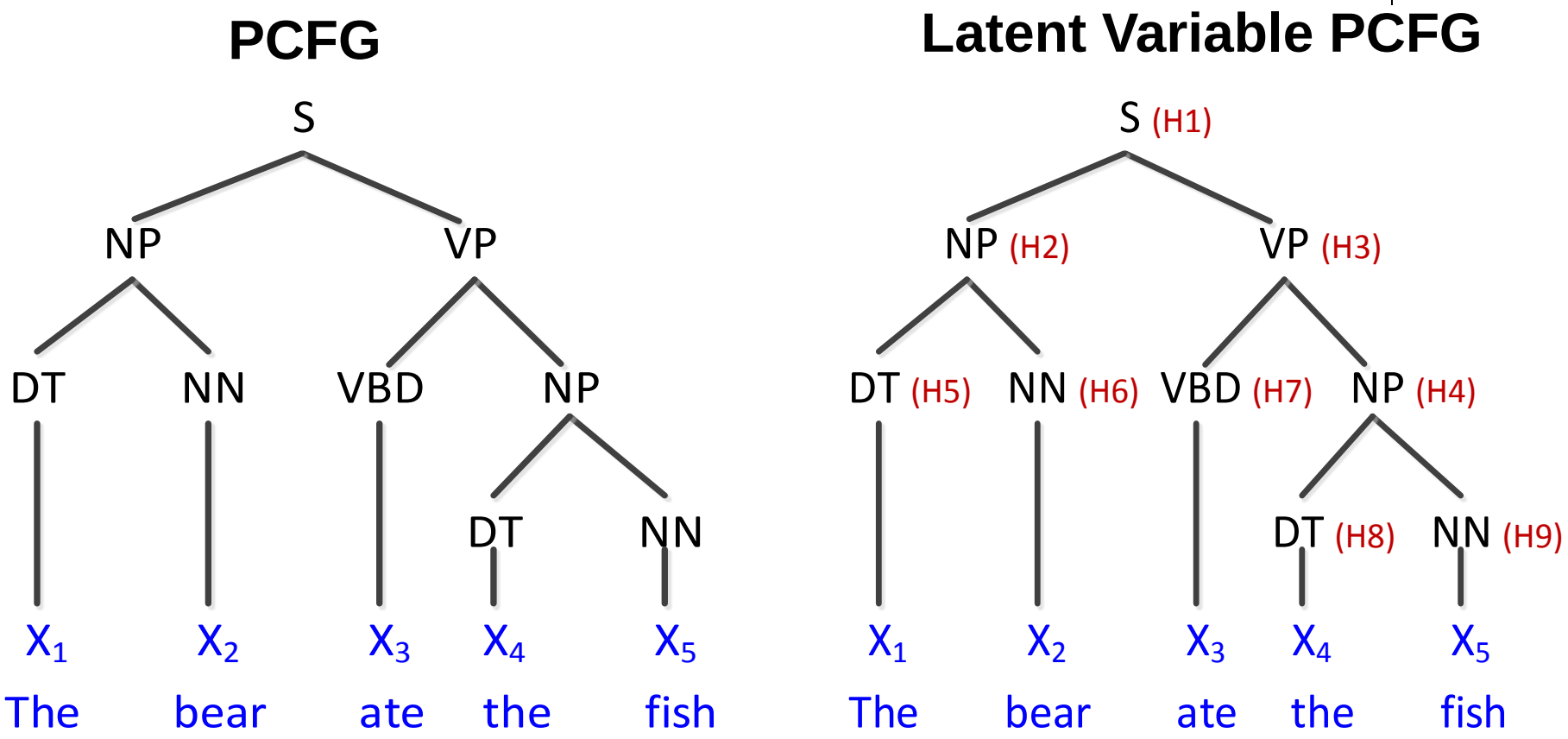




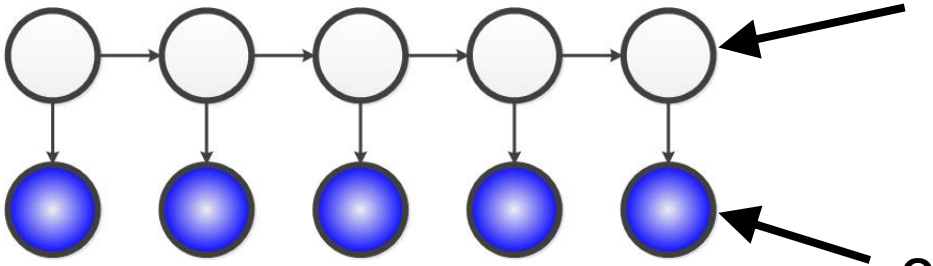
Ho. et al. 2012

Mixed membership models

Latent Variable PCFG [Matsuzaki et al., 2005, Petrov et al. 2006]



Learning Parameters (EM)

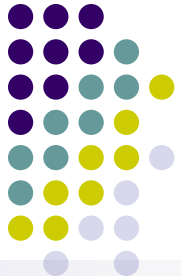


$$\mathbb{P}[X_1, \ldots, X_5, H_1, \ldots, H_5] = \mathbb{P}[H_1] \prod_{i=2}^{5} \mathbb{P}[H_i | H_{i-1}] \prod_{i=1}^{5} \mathbb{P}[X_i | H_i]$$

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

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

Spectral Learning

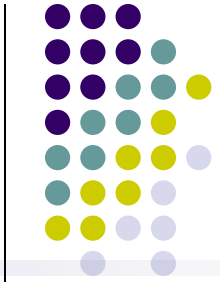


- Different paradigm of learning in latent variable models based on linear algebra

- Theoretically,
  - Provably consistent
  - Can offer deeper insight into the identifiability

- Practically,
  - Local minima free
  - As if now, performs comparably to EM with 10-100x speed-up
  - Can also model non-Gaussian continuous data using kernels (usually performs much better than EM in this case)

Tutorial Outline



- The Spectral View of Graphical Models
- Small (HMM-like) example
- How to make Spectral Learning Work in Practice
- Intuition to why this works for trees / latent PCFGs
- Discussion of Empirical Aspects
- Detailed derivation for latent PCFG

Related References

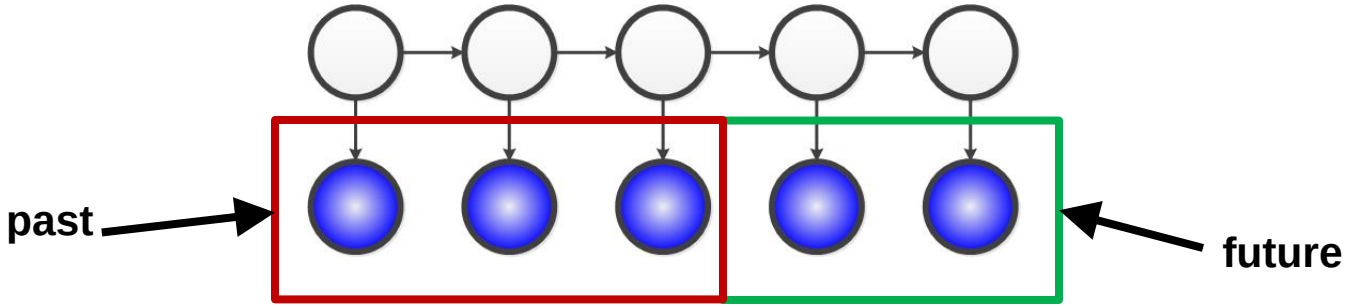
- Relevant works
  - **Hsu et al. 2009** – Spectral HMMs (also Bailly 2009)
  - **Siddiqi et al. 2009** – Features in Spectral Learning
  - **Parikh et al. 2011/2012** –Tensors to Generalize to Trees/Low Treewidth Graphs
  - **Cohen et al. 2012 / 2013** – Spectral Learning of latent PCFGs

- Will present it from "matrix factorization" view:
  - **Balle et al. 2012** – Connection between Spectral Learning / Hankel Matrix Factorization
  - **Song et al. 2013** – Spectral Learning as Hierarchical Tensor Decomposition

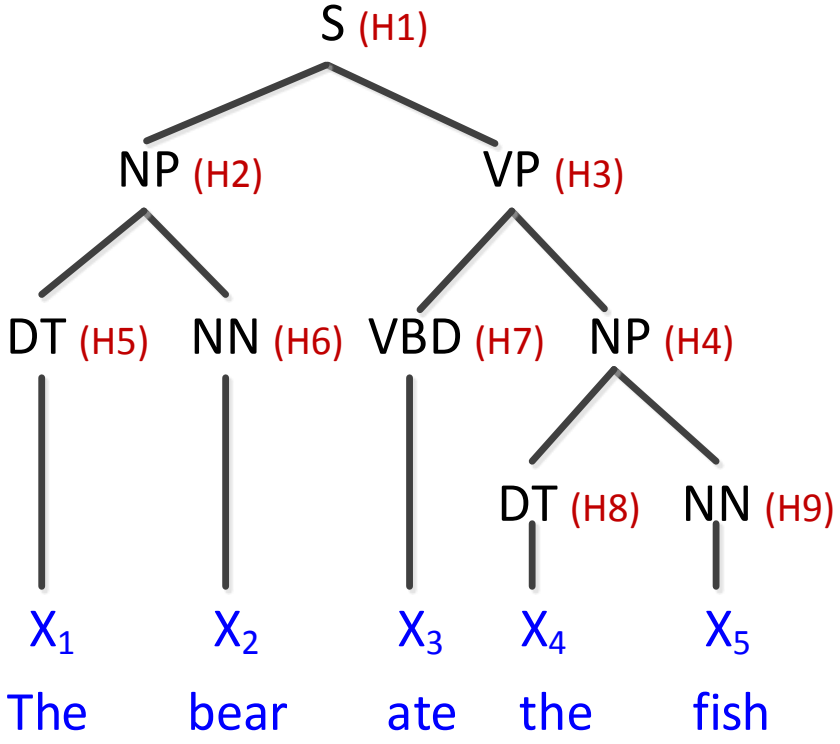
Focusing on Prediction

- 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



Latent Variable PCFG [Matsuzaki et al., 2005, Petrov et al. 2006]



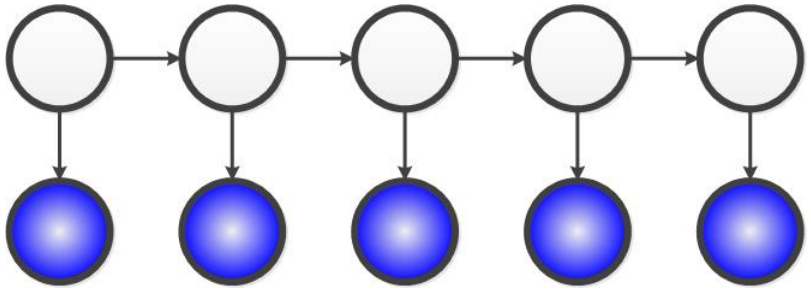
$$\mathbb{P}[tree] = \sum_{H_1,\ldots,H_5} \mathbb{P}[tree, H_1, \ldots, H_5]$$

Focusing on Prediction

- We will only be concerned with quantities related to the observed variables:

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

- We do not care about the latent variables explicitly.

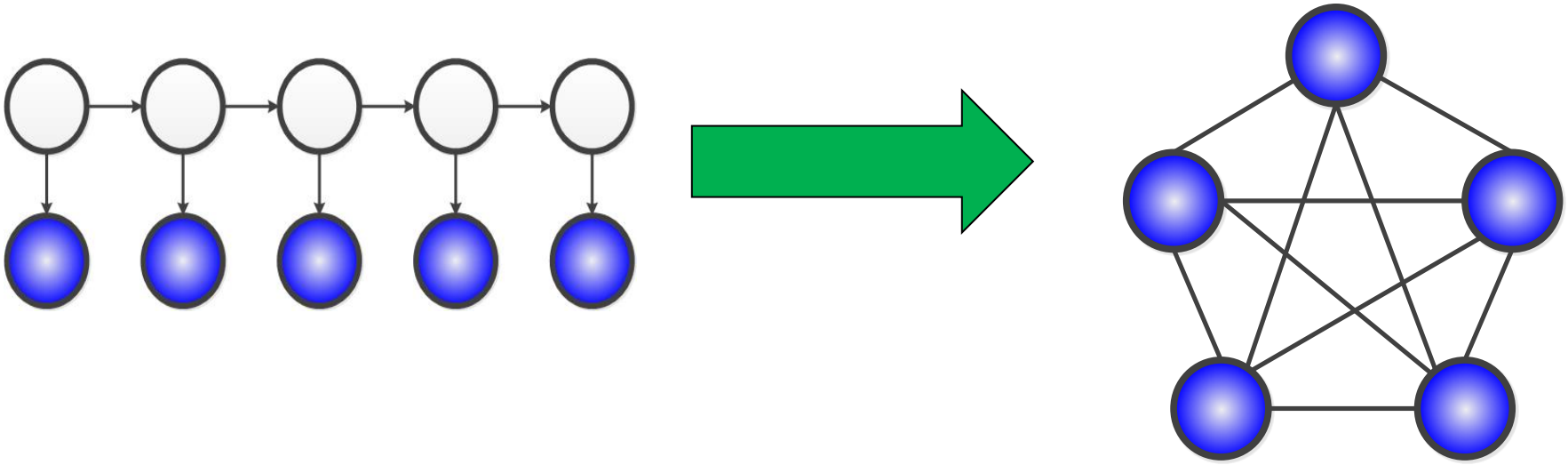


- **Do we still need EM to learn the parameters?**

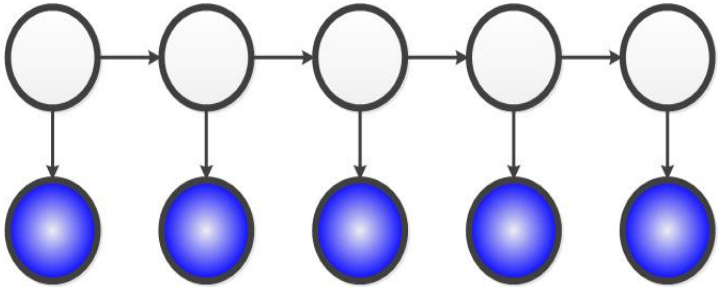
But if we don't care about the latent variables....

- Why don't we just integrate them out?

- Because integrating them out results in a clique ☹



Marginal Does Not Factorize



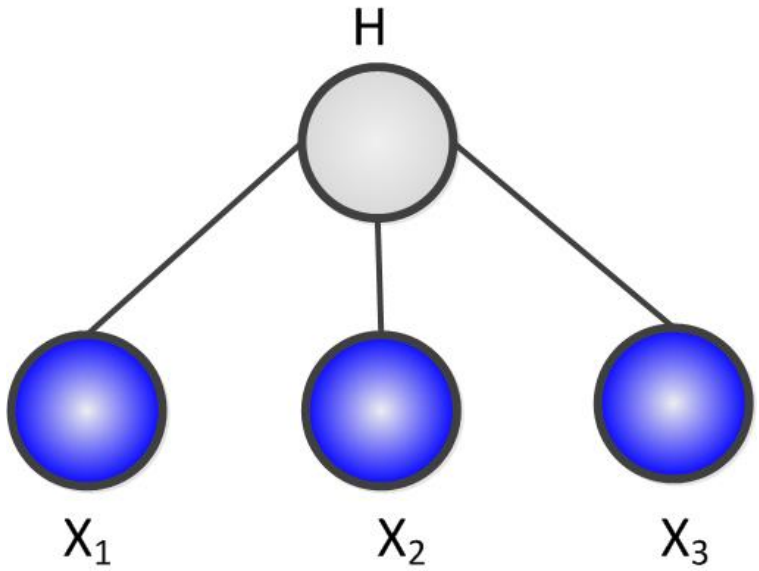
$$\mathbb{P}[X_1, X_2, X_3, X_4, X_5] = \sum_{H_1,\ldots,H_5} \mathbb{P}[H_1]\mathbb{P}[H_1] \prod_{i=2}^{5} \mathbb{P}[H_i|H_{i-1}] \prod_{i=1}^{5} \mathbb{P}[X_i|H_i]$$

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

But isn't an HMM different from a clique?

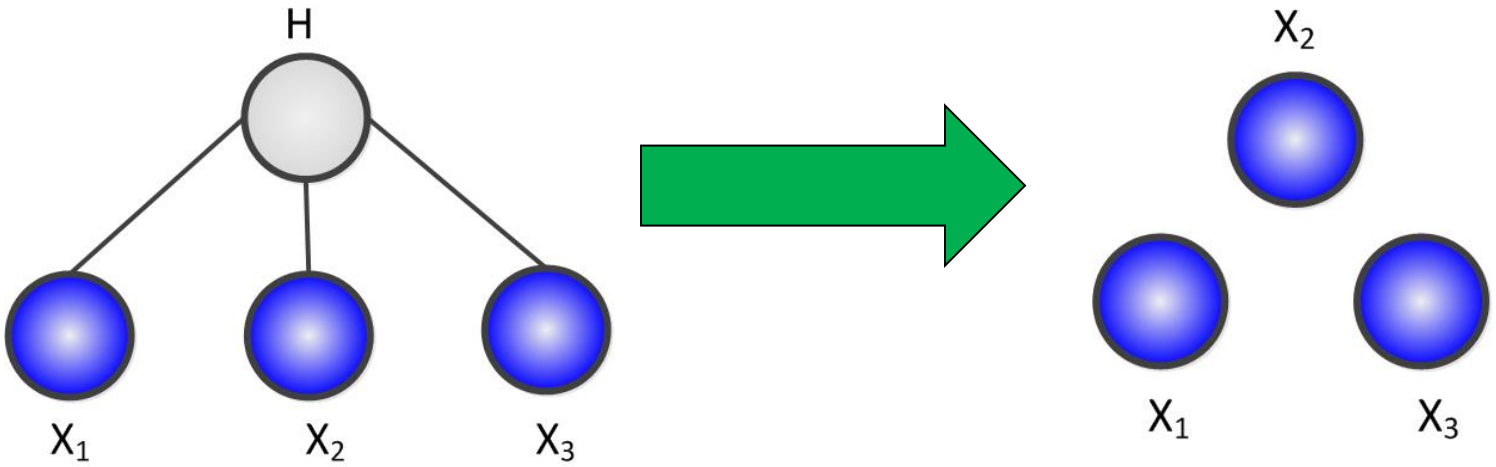
- It depends on the number of latent states.

- Consider the following model.



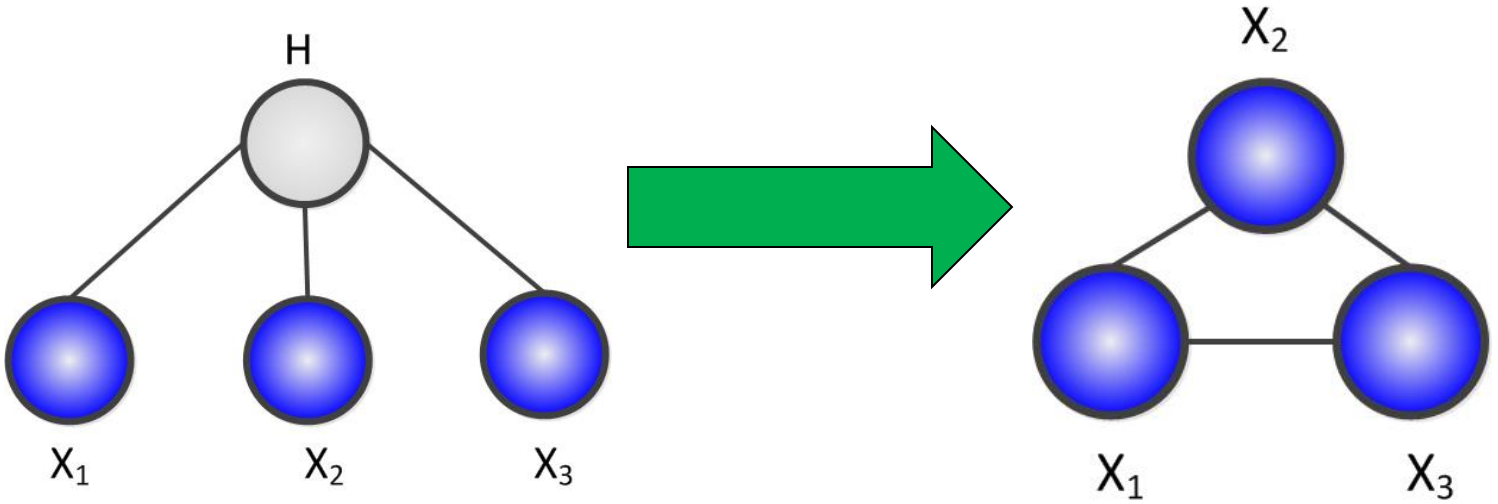
If H has only one state.....

- Then the observed variables are independent!



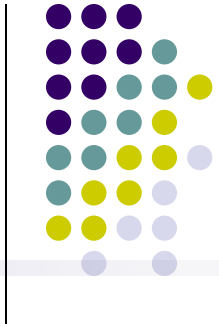
What if H has many states?

- 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).



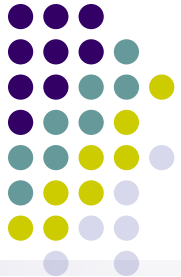
- But what about all the other cases?

The Question



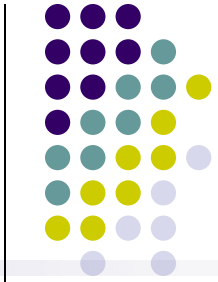
- Under existing methods, latent models all require EM to learn regardless of the number of hidden states.
- However, is there a formulation of latent variable models where the difficulty of learning is a function of the number of latent states?
- This is the question that the *spectral view* will answer.

Tutorial Outline



- **The Spectral View of Graphical Models**

Sum Rule (Matrix Form)



- Sum Rule

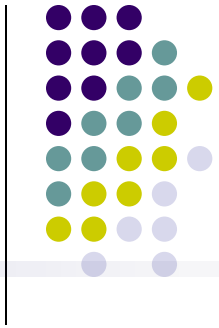
$$\mathbb{P}[X] = \sum_{Y} \mathbb{P}[X|Y]\mathbb{P}[Y]$$

- Equivalent view using Matrix Algebra

$$\boldsymbol{\mathcal{P}}[X] = \boldsymbol{\mathcal{P}}[X|Y] \times \boldsymbol{\mathcal{P}}[Y]$$

$$\begin{pmatrix} \mathbb{P}[X=0] \\ \mathbb{P}[X=1] \end{pmatrix} = \begin{pmatrix} \mathbb{P}[X=0|Y=0] & \mathbb{P}[X=0|Y=1] \\ \mathbb{P}[X=1|Y=0] & \mathbb{P}[X=1|Y=1] \end{pmatrix} \times \begin{pmatrix} \mathbb{P}[Y=0] \\ \mathbb{P}[Y=1] \end{pmatrix}$$

Important Notation



- Calligraphic P to denotes that the probability is being treated as a matrix/vector/tensor

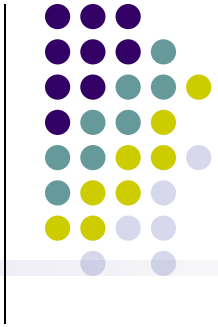
- Probabilities

$$\mathbb{P}[X, Y] = \mathbb{P}[X|Y]\mathbb{P}[Y]$$

- Probability Vectors/Matrices/Tensors

$$\boldsymbol{\mathcal{P}}[X] = \boldsymbol{\mathcal{P}}[X|Y]\boldsymbol{\mathcal{P}}[Y]$$

Chain Rule (Matrix Form)



- Chain Rule

$$\mathbb{P}[X,Y] = \mathbb{P}[X|Y]\mathbb{P}[Y] = \mathbb{P}[Y|X]\mathbb{P}[Y]$$

- Equivalent view using Matrix Algebra

Means on diagonal

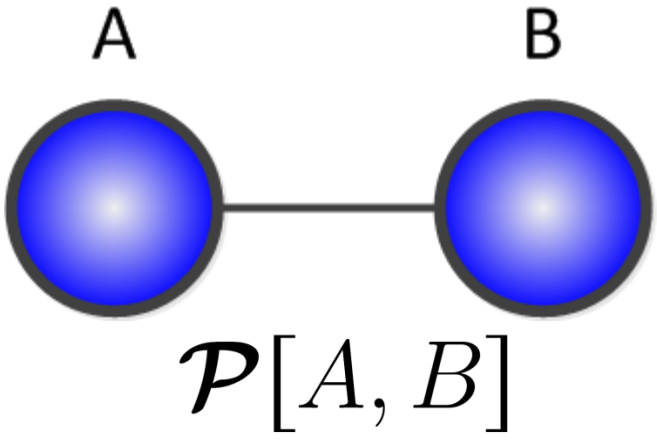
$$\boldsymbol{\mathcal{P}}[X,Y] = \boldsymbol{\mathcal{P}}[X|Y] \times \boldsymbol{\mathcal{P}}[\oslash Y]$$

$$\begin{pmatrix} \mathbb{P}[X=0,Y=0] & \mathbb{P}[X=0,Y=1] \\ \mathbb{P}[X=1,Y=0] & \mathbb{P}[X=1,Y=1] \end{pmatrix} =$$

$$\begin{pmatrix} \mathbb{P}[X=0|Y=0] & \mathbb{P}[X=0|Y=1] \\ \mathbb{P}[X=1|Y=0] & \mathbb{P}[X=1|Y=1] \end{pmatrix} \times \begin{pmatrix} \mathbb{P}[Y=0] & 0 \\ 0 & \mathbb{P}[Y=1] \end{pmatrix}$$

- Note how diagonal is used to keep ***Y*** from being marginalized out.

Graphical Models: The Linear Algebra View



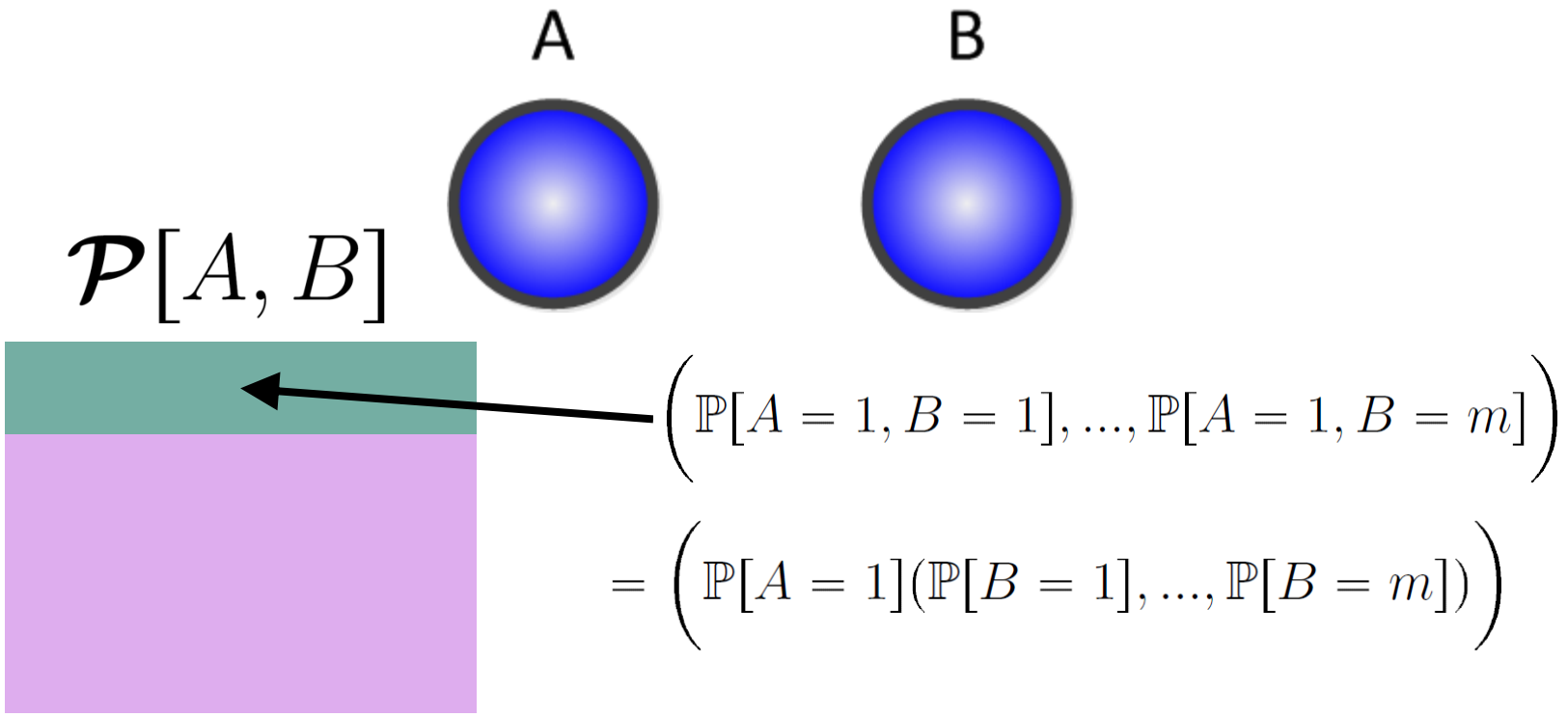
$$\mathcal{P}[A, B]$$

A and B have m states each.

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

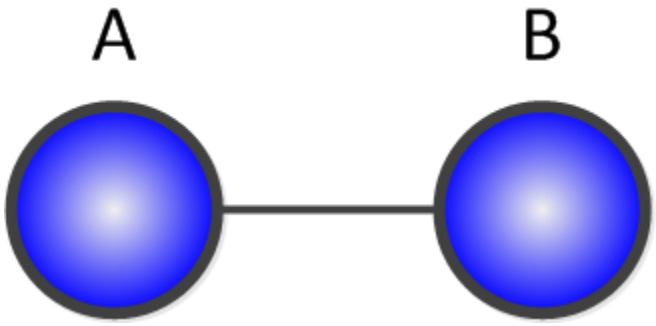
Independence: The Linear Algebra View

- What if we know A and B are independent?

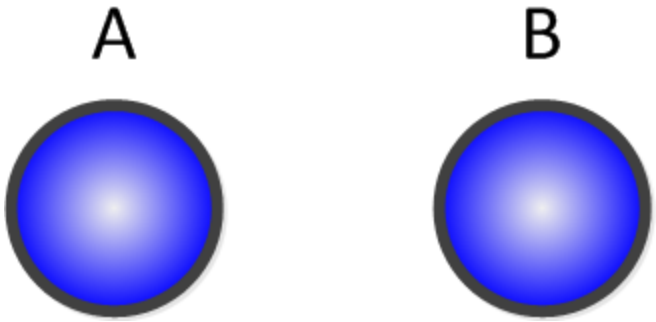


- Joint probability matrix is rank one, since all rows are multiples of one another!!

Independence and Rank



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

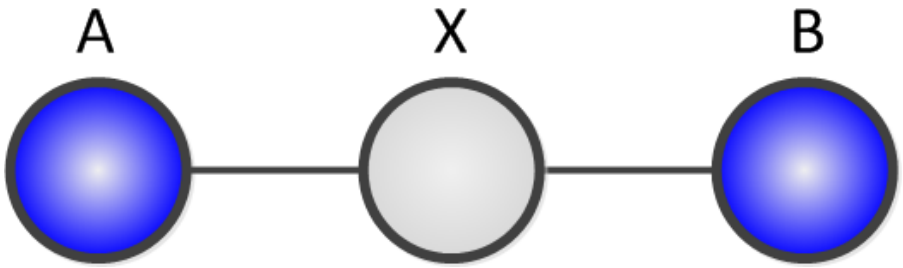


$\mathcal{P}[A, B]$ **has rank 1**

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

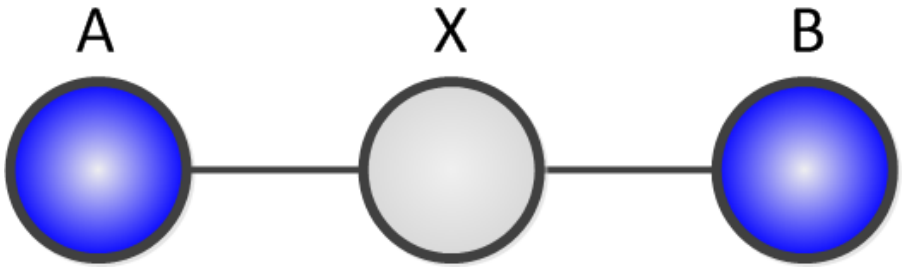
Low Rank Structure

- ***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?

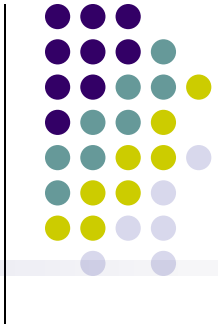
Low Rank Structure



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

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

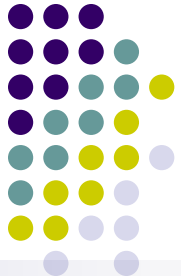
The Spectral View



- 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

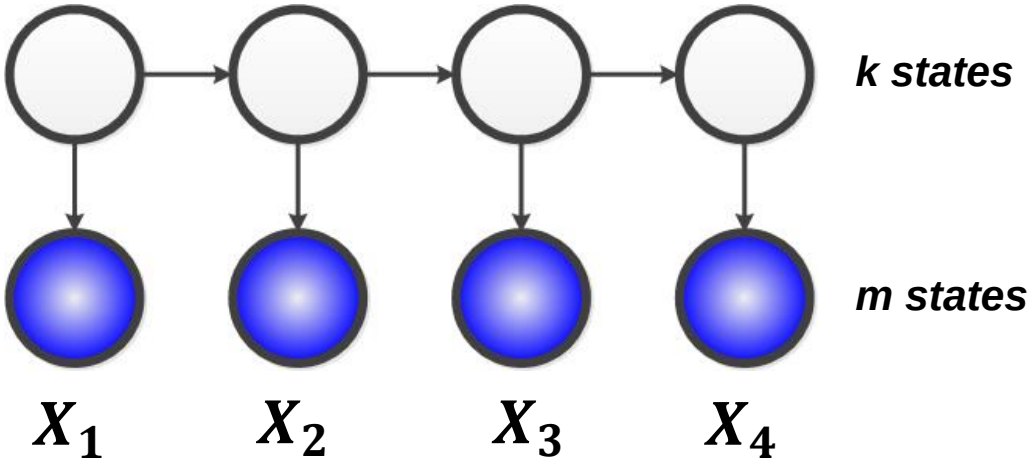
Tutorial Outline



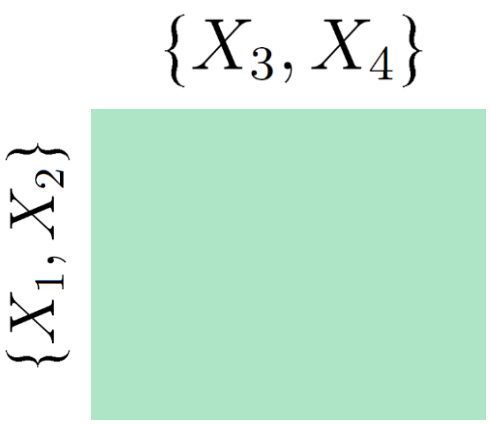
- The Spectral View of Graphical Models

- Small (HMM-like) example

A More Interesting Example

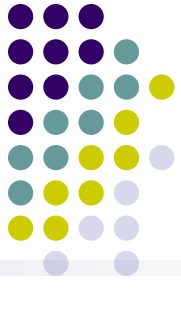


$$\mathcal{P}[X_{\{1,2\}}, X_{\{3,4\}}]$$



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!!!

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

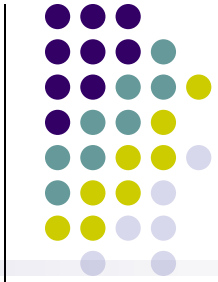
Factor of 4 variables

Factor of 3 variables

Factor of 1 variable

Factor of 3 variables

Alternate Factorizations



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

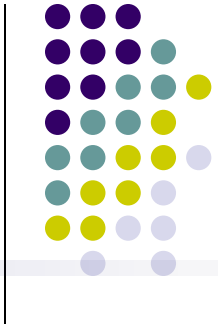
- Consider Matrix Factorization. Can add any invertible transformation:

$$M = RL$$

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

- **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

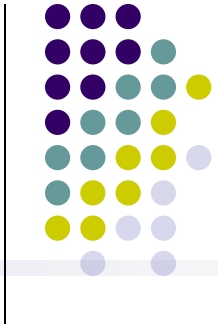
$$\mathcal{P}[X_{\{1,2\}}, X_{\{3,4\}}]$$

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

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

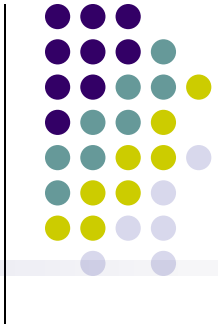
$$\mathcal{P}[X_2, X_{\{3,4\}}] = \mathcal{P}[X_2|H_2]\mathcal{P}[\oslash H_2]\mathcal{P}[X_{\{3,4\}}|H_2]^\top$$

- Product of green terms (in some order) is

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

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

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**

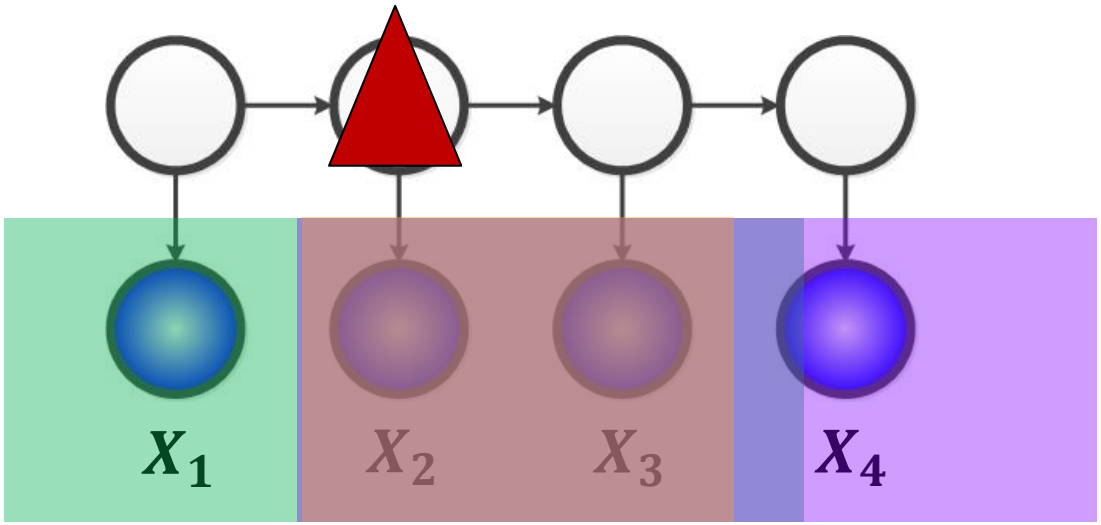
Advantage: Factors are only functions of observed variables! Can be directly computed from data without EM!!!!

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

We will call this factorization the observable factorization.

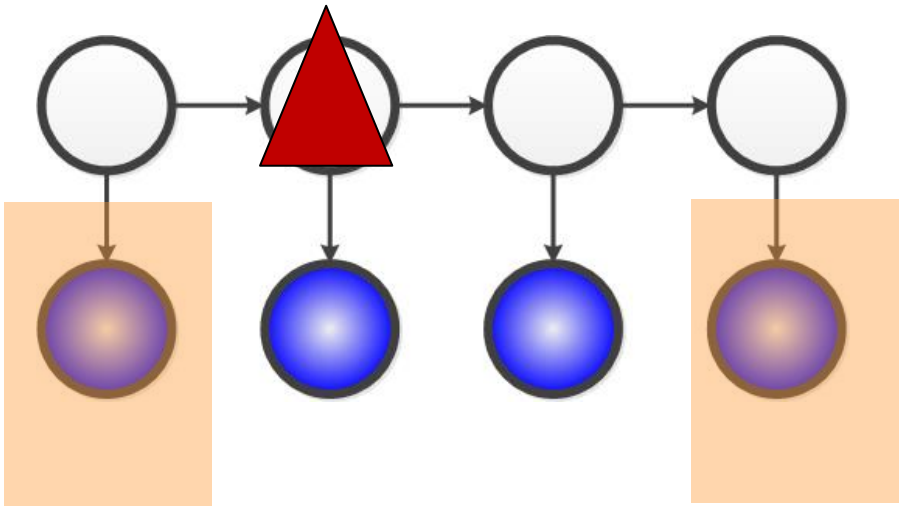
Graphical Relationship

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



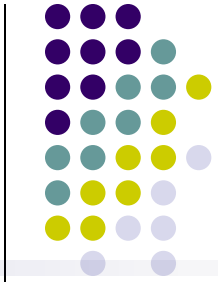
Another Factorization

$$\mathcal{P}[X_{\{1,2\}}, X_{\{3,4\}}] = \mathcal{P}[X_{\{1,2\}}, X_4]\mathcal{P}[X_1, X_4]^{-1}\mathcal{P}[X_1, X_{\{3,4\}}]$$



- Seems we would do better empirically if you could "combine" both factorizations. Will come back to this later.

Relationship to Original Factorization



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

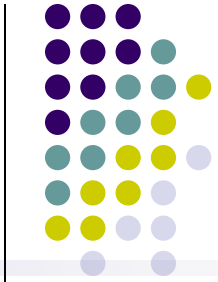
$$\underbrace{\boldsymbol{\mathcal{P}}[X_{\{1,2\}}, X_{\{3,4\}}]}_{\boldsymbol{M}} = \underbrace{\boldsymbol{\mathcal{P}}[X_{\{1,2\}}|H_2]\boldsymbol{\mathcal{P}}[\oslash H_2]}_{\boldsymbol{R}}\underbrace{\boldsymbol{\mathcal{P}}[X_{\{3,4\}}|H_2]^\top}_{\boldsymbol{L}}$$

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

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

Can I choose S to get the observable factorization?

Relationship to Original Factorization

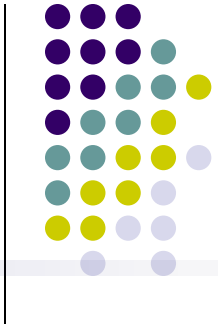


- Let

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

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

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 3 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 ***V*** variables has such a factorization where:
  - All factors are of size 3
  - All factors are only functions of observed variables

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?

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

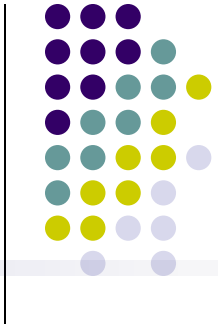
When does this inverse exist?

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)

When m > k



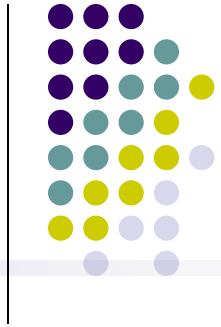
- The inverse cannot exist, but this situation is easily fixable (project onto lower dimensional space)

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

$$\boldsymbol{\mathcal{P}}[X_{\{1,2\}}, X_3]\boldsymbol{V}(\boldsymbol{U}^\top \boldsymbol{\mathcal{P}}[X_2, X_3]\boldsymbol{V})^{-1}\boldsymbol{U}^\top \boldsymbol{\mathcal{P}}[X_2, X_{\{3,4\}}]$$

- Where ***U***, ***V*** are the top left/right **k** singular vectors of $\boldsymbol{\mathcal{P}}[X_2, X_3]$

When k > m

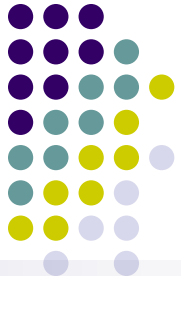


- The inverse does exist. But it no longer satisfies the following property, which we used to derive the factorization

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

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

What does k>m mean?



- Intuitively, large ***k***, small ***m*** means long range dependencies

- Consider following generative process:

  (1) With probability 0.5, let ***S= X***, and with probability 0.5 let ***S=Y***.

  (2) Print ***A n*** times.

  (3) Print ***S***

  (4) Go back to step (2)

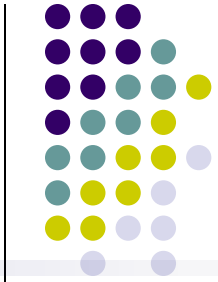
With ***n=1*** we either generate:

AXAXAXA…… or AYAYAYA…..

With ***n=2*** we either generate:

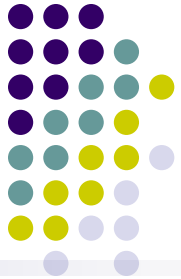
AAXAAXAA….. or AAYAAYAA…….

How many hidden states does HMM need?



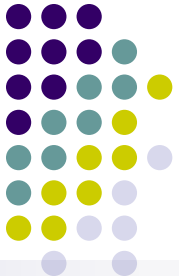
- HMM needs **2n** states.
- Needs to remember count as well as whether we picked ***S=X*** or ***S=Y***
- However, number of observed states ***m*** does not change, so our previous spectral algorithm will break for ***n > 2***.
- How to deal with this in spectral framework?

Tutorial Outline



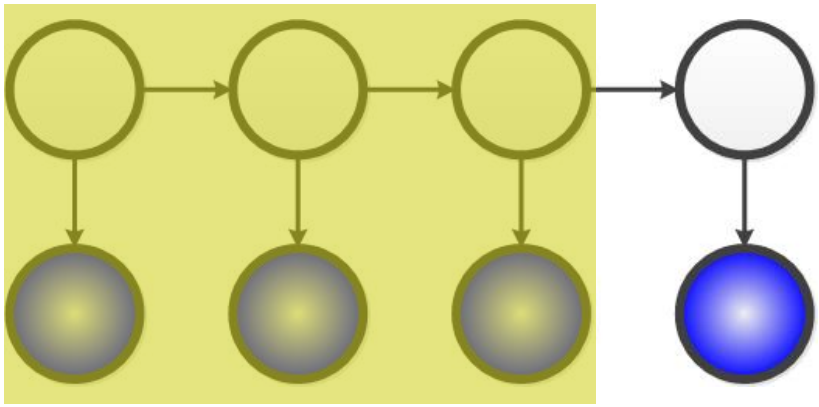
- The Spectral View of Graphical Models
- Small (HMM-like) example
- How to make Spectral Learning Work in Practice

Making Spectral Learning Work In Practice



- We are only using marginals of pairs/triples of variables to construct the full marginal among the observed variables.

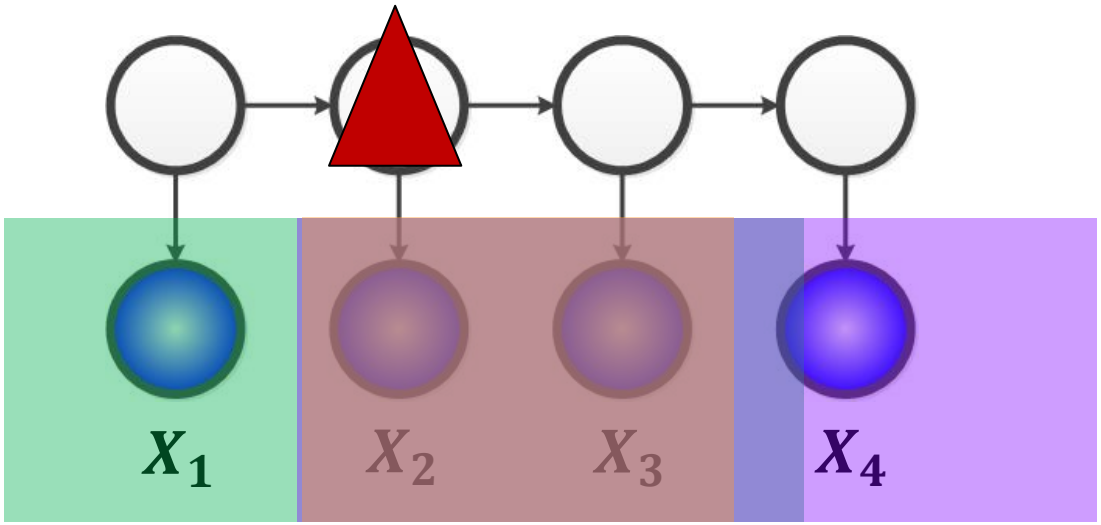
- Only works when $k < m$.



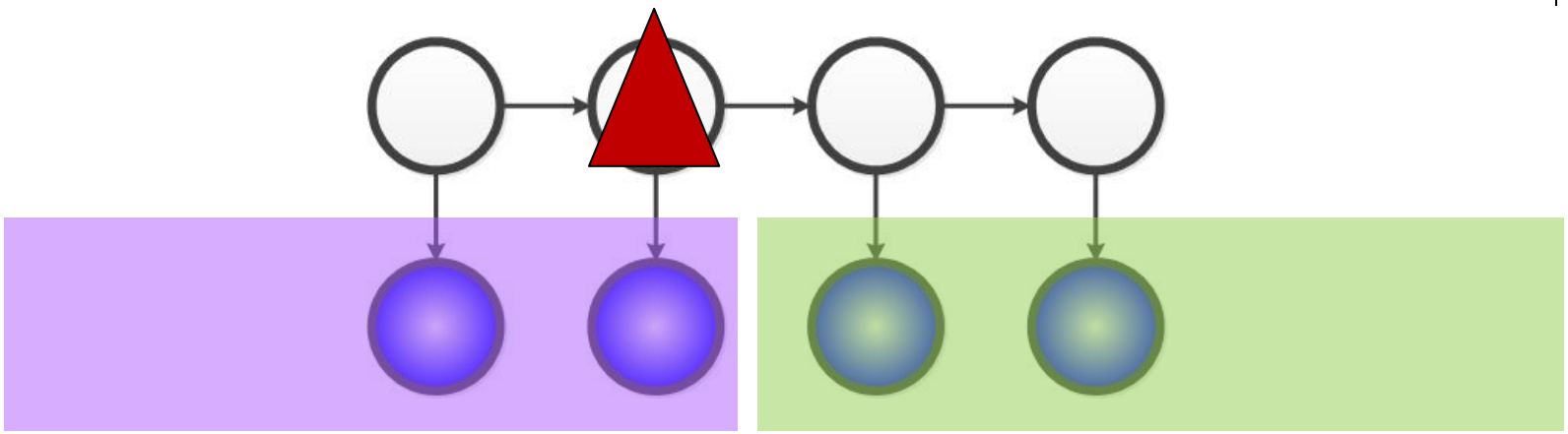
- However, in real problems we need to capture longer range dependencies.

Recall our factorization

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



Key Idea: Use Long-Range Features



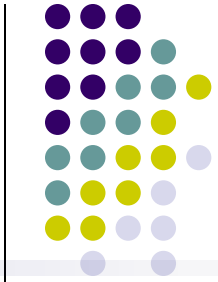
Construct feature vector of left side

$$\phi_L$$

Construct feature vector of right side

$$\phi_R$$

Observable Factorization Works With Features Too



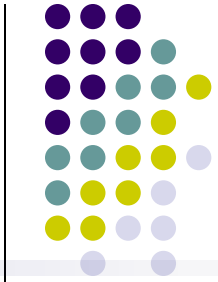
$$\mathcal{P}[X_2, X_3] = \mathbb{E}[\boldsymbol{\delta}_2 \otimes \boldsymbol{\delta}_3] := \mathbb{E}[\boldsymbol{\delta}_2 \boldsymbol{\delta}_3^\top]$$

Use more complex feature instead:

$$\mathbb{E}[\boldsymbol{\phi}_L \otimes \boldsymbol{\phi}_R]$$

$$\mathcal{P}[X_{\{1,2\}}, X_{\{3,4\}}] = \mathbb{E}[\boldsymbol{\delta}_{1\otimes 2}, \boldsymbol{\delta}_{3\otimes 4}]$$
$$= \mathbb{E}[\boldsymbol{\delta}_{1\otimes 2}, \boldsymbol{\phi}_R]\boldsymbol{V}(\boldsymbol{U}^\top \mathbb{E}[\boldsymbol{\phi}_L \otimes \boldsymbol{\phi}_R]\boldsymbol{V})^{-1}\boldsymbol{U}^\top \mathcal{P}[\boldsymbol{\phi}_L, X_{\{3,4\}}]$$

Interesting Engineering Trick

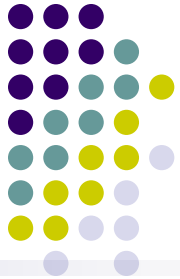


- SVD may be very sensitive to very common features.

- Normalize features based on how many times they appear in the dataset.

$$\tilde{\phi}_L^{(i)} = \phi_L^{(i)} \times \sqrt{\frac{1}{\text{count}(i) + \kappa}}$$

Other (Simple) Engineering Tricks

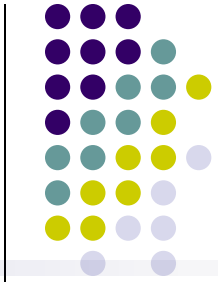


- Sometimes, spectral learning will produce negative probabilities. There are two options to deal with this.
  - Set to zero (or very small probability)
  - Flip the sign (i.e. take absolute value)

- Flipping the sign works much, much better!

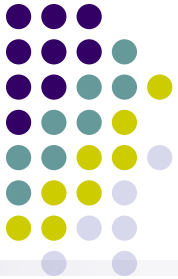
- Sometimes the matrices/cubes are sparse. Can interpolate with lower order models to make them more dense.

Tutorial Outline

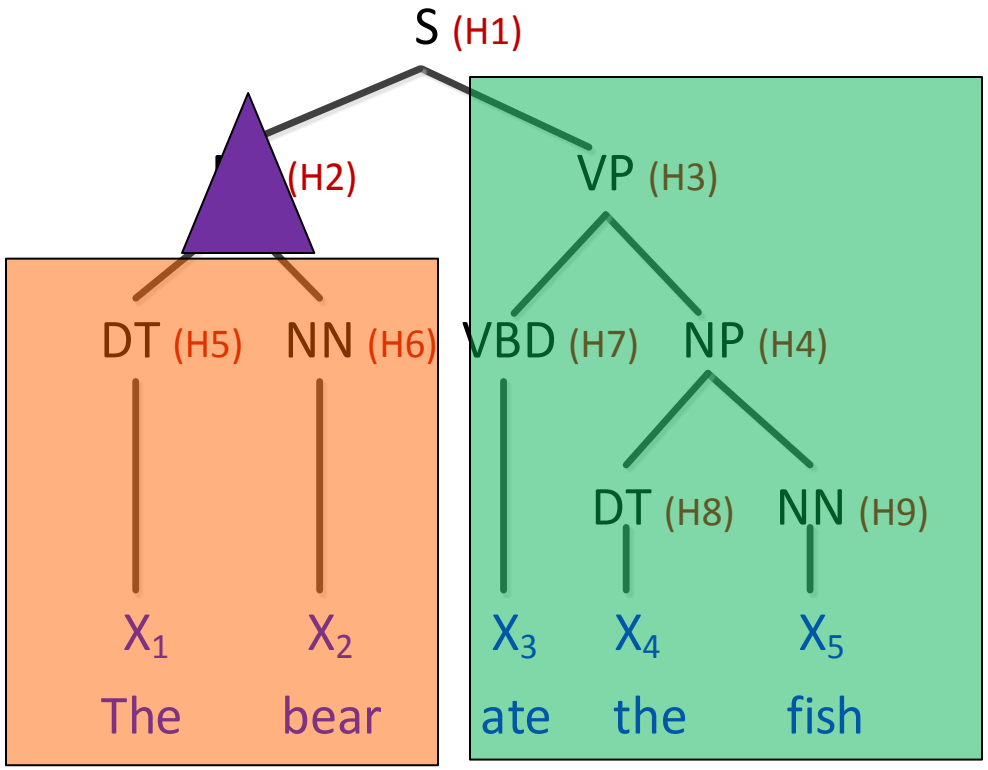


- The Spectral View of Graphical Models
- Small (HMM-like) example
- How to make Spectral Learning Work in Practice
- Intuition to why this works for trees / latent PCFGs
- Discussion of Empirical Aspects

Same General Ideas work with Latent Variable PCFGs

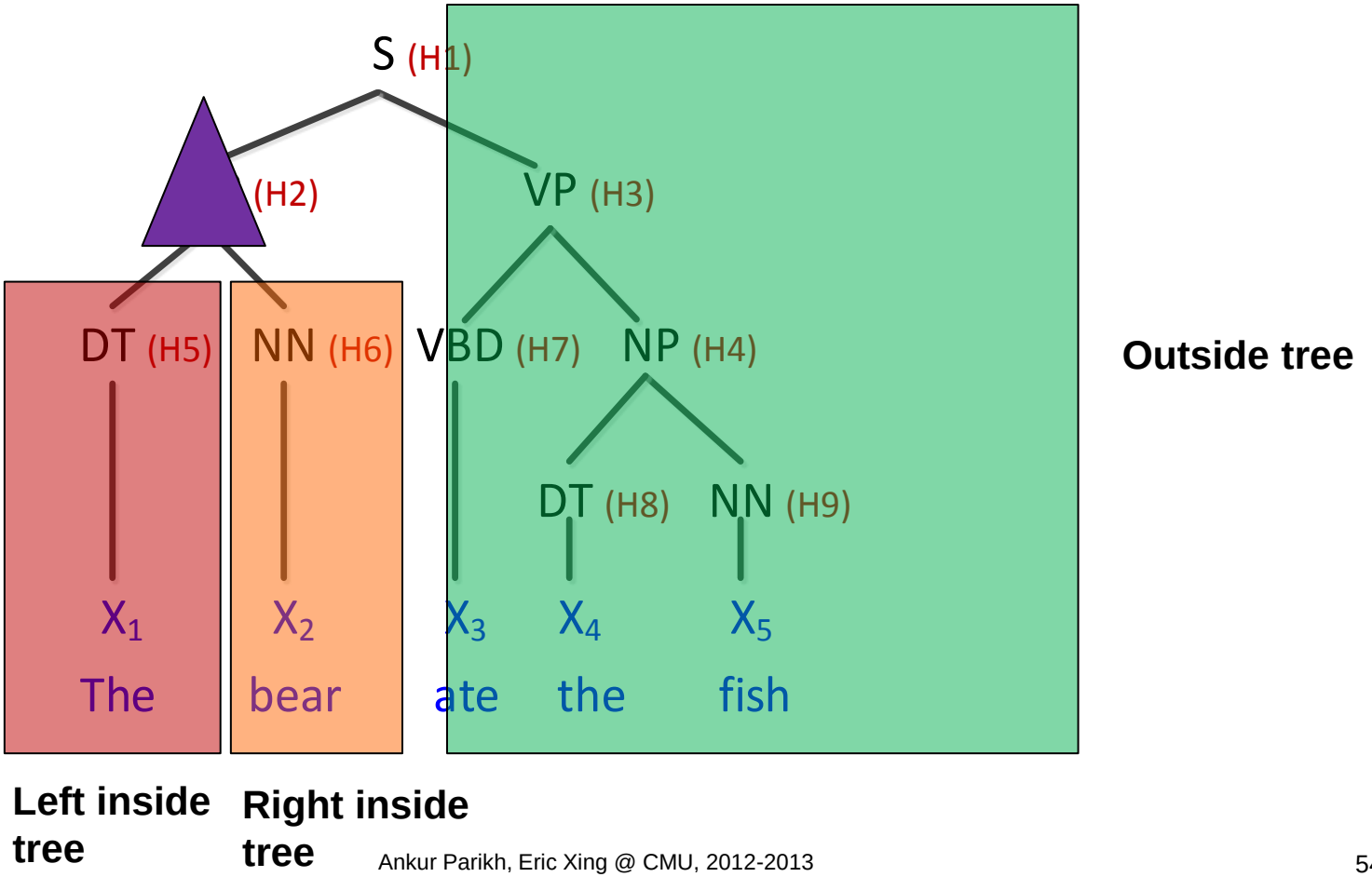


- General Idea is to use same strategy as before partitioning inside/outside trees.

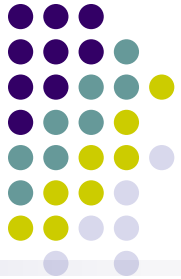


Features

- Use Features of Inside/Outside Trees Instead



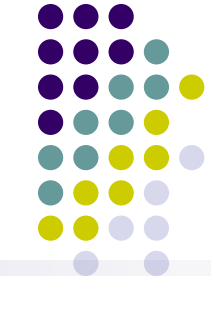
Empirical Results for Latent PCFGs



	section 22		section 23	
	EM	spectral	EM	spectral
$m = 8$	86.87	85.60	—	—
$m = 16$	88.32	87.77	—	—
$m = 24$	88.35	88.53	—	—
$m = 32$	88.56	88.82	87.76	88.05

Results from Cohen et al. 2013

Spectral Learning is Much Faster

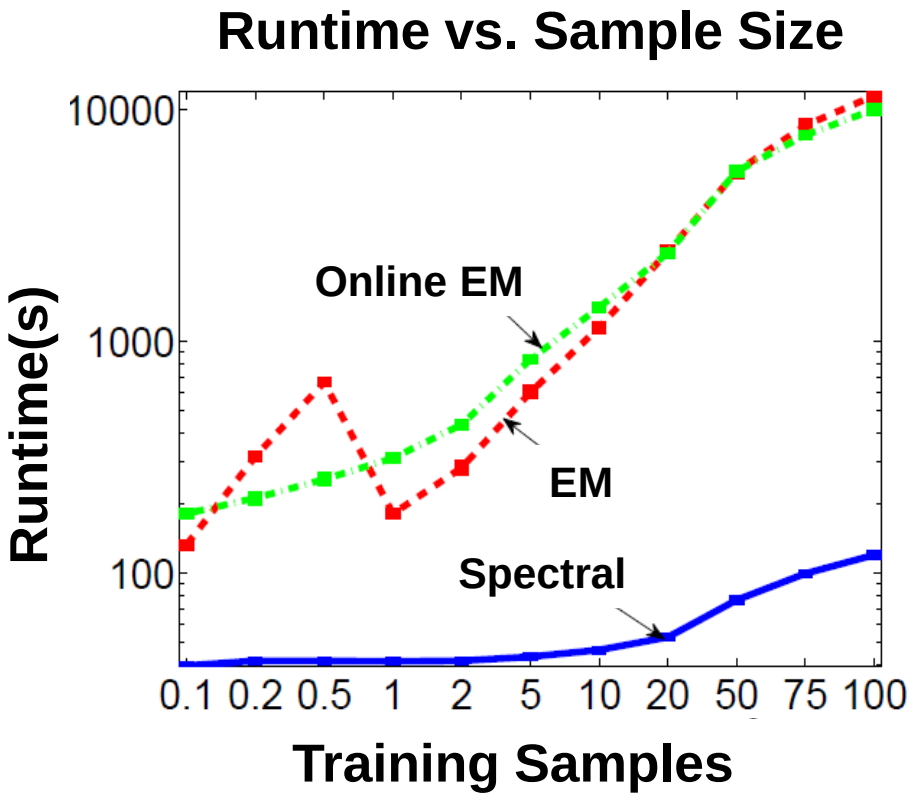


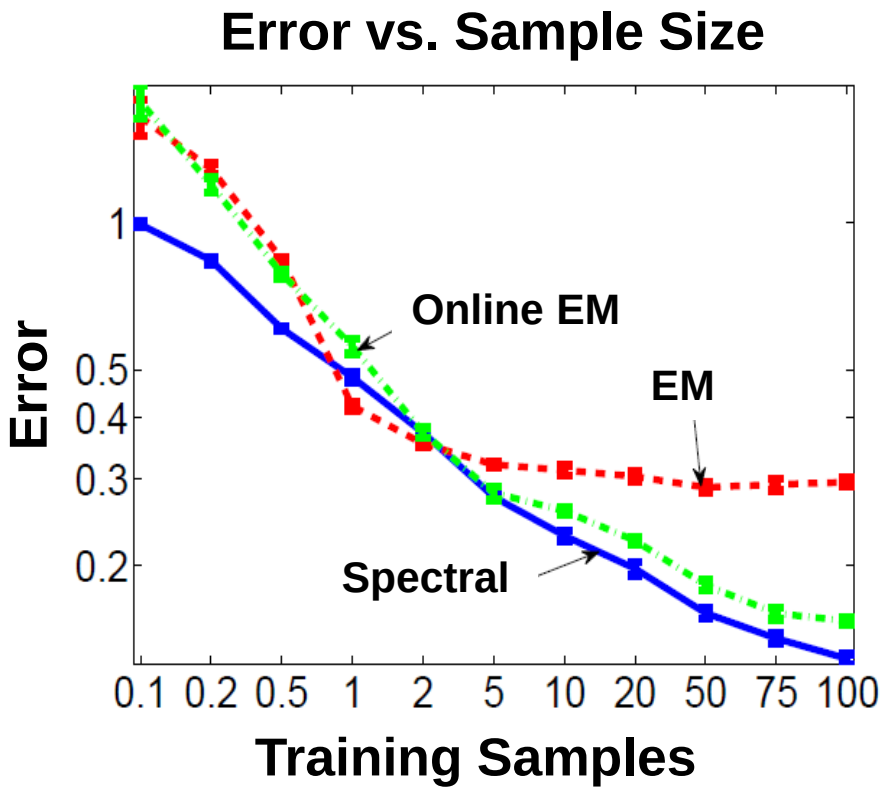
	single EM iter.	EM best model	spectral algorithm total	feature	transfer + scaling	SVD	$a \rightarrow b\ c$	$a \rightarrow x$
$m = 8$	6m	3h	3h32m	22m	49m	36m	1h34m	10m
$m = 16$	52m	26h6m	5h19m			34m	3h13m	19m
$m = 24$	3h7m	93h36m	7h15m			36m	4h54m	28m
$m = 32$	9h21m	187h12m	9h52m			35m	7h16m	41m

Results from Cohen et al. 2013

Spectral Learning Scales Better with Training Sample Size

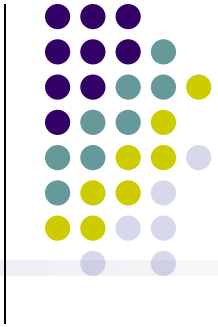
- Synthetic 3rd order HMM Example (Spectral/EM/Online EM):





Results from Parikh et al. 2012

EM vs Spectral (Part I)



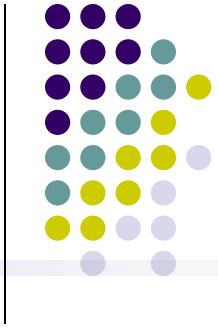
EM

- Aims to Find MLE so more "statistically" efficient
- Can get stuck in local-optima
- Lack of theoretical guarantees
- Slow
- Easy to derive for new models

Spectral

- Does not aim to find MLE so less statistically efficient.
- Local-optima-free
- Provably consistent
- Very fast
- Challenging to derive for new models (Unknown whether it can generalize to arbitrary loopy models)

EM vs Spectral (Part II)



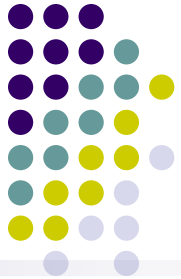
EM

- No issues with negative numbers
- Allows for easy modelling with conditional distributions
- Difficult to incorporate long-range features (since it increases treewidth).
- Generalizes poorly to non-Gaussian continuous variables.

Spectral

- Problems with negative numbers. Requires explicit normalization to compute likelihood.
- Allows for easy modelling with marginal distributions
- Easy to incorporate long-range features.
- Easy to generalize to non-Gaussian continuous variables via Hilbert Space Embeddings

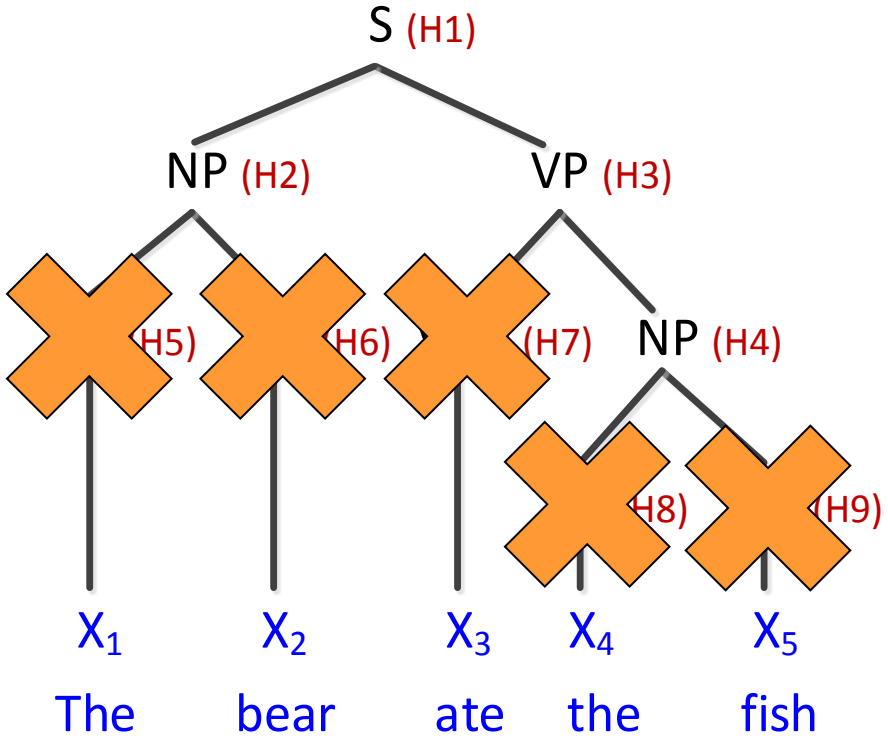
Other "Spectral" Directions in NLP

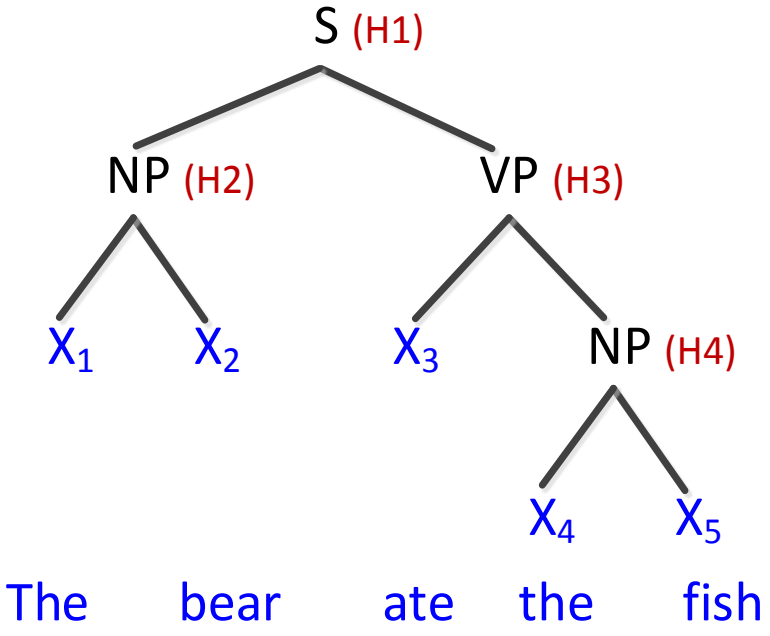


- Recovering the original parameters using tensor decomposition.
  - **Anandkumar et al. 2012** (HMMs, Mixture of Gaussians)
  - **Anandkumar et al. 2012** (Latent Dirichlet Allocation)

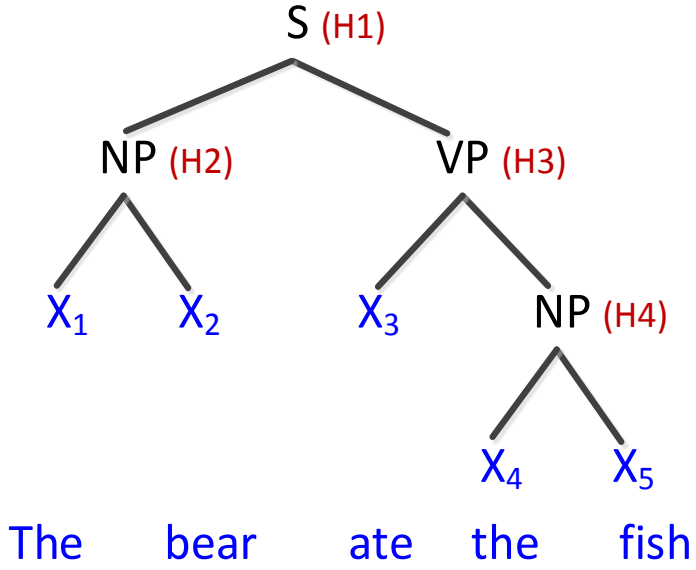
- "Spectral Inspired" Methods
  - **Dhillon et al. 2011/2012** – Word Embeddings using CCA
  - **Parikh et al. 2013 (Hopefully)** – Local-Optima-Free Unsupervised Parsing

Latent PCFGs (Simplification)





Latent PCFGs (Simplification)

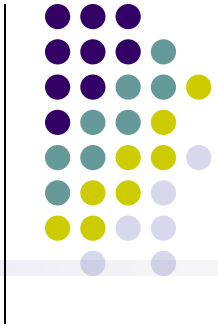


Rules

$$\mathtt{S} \rightarrow \mathtt{NP}, \mathtt{VP}$$
$$\mathtt{NP} \rightarrow \mathtt{X}_1, \mathtt{X}_2$$
$$\mathtt{VP} \rightarrow \mathtt{X}_3, \mathtt{NP}$$
$$\mathtt{NP} \rightarrow \mathtt{X}_4, \mathtt{X}_5$$

$$\mathbb{P}[X_1, ..., X_5, \mathtt{S}_1, \mathtt{NP}_2, \mathtt{VP}_3, \mathtt{NP}_4, H_1, ..., H_5] =$$
$$\mathbb{P}[X_1, X_2 | \mathtt{NP}_2, H_2] \times \mathbb{P}[X_4, X_5 | \mathtt{NP}_4, H_4] \times \mathbb{P}[X_3, \mathtt{NP}_4, H_4 | \mathtt{VP}_3, H_3]$$
$$\times \mathbb{P}[\mathtt{NP}_2, H_2, \mathtt{VP}_3, H_3 | \mathtt{S}_1, H_1] \times \mathbb{P}[\mathtt{S}_1, H_1]$$

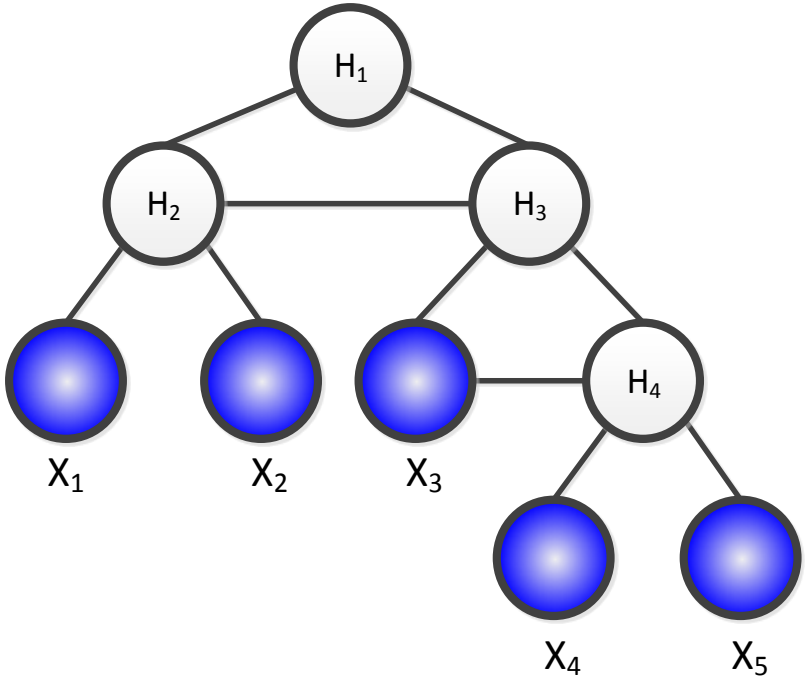
Latent PCFGs (Simplification)



- For now ignore the observed, non-leaf variables (i.e. the rules). They can easily be added later.

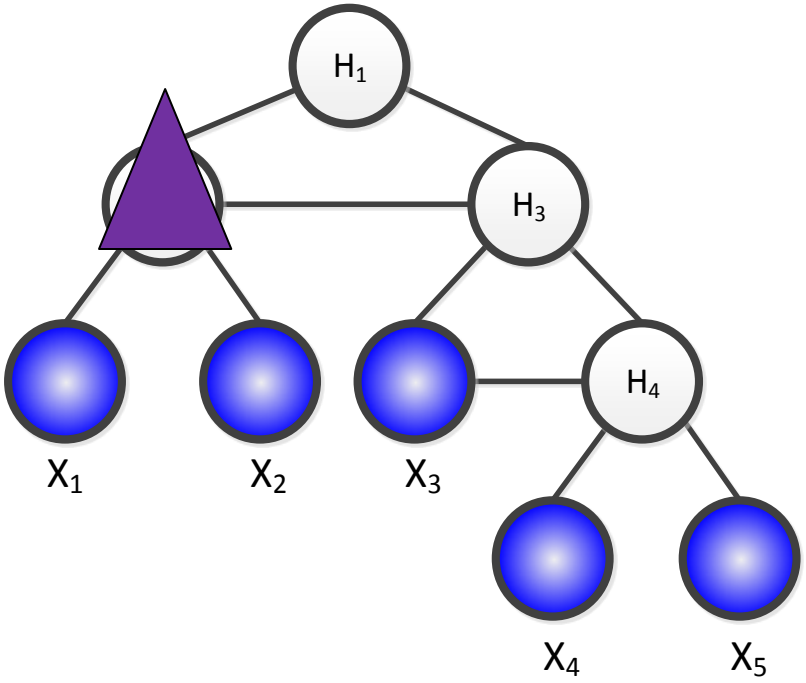
$$\mathbb{P}[X_1, ..., X_5, H_1, ..., H_5] =$$
$$\mathbb{P}[X_1, X_2|H_2] \times \mathbb{P}[X_4, X_5|H_4] \times \mathbb{P}[X_3, H_4|H_3]$$
$$\times \mathbb{P}[H_2, H_3|H_1] \times \mathbb{P}[H_1]$$

Now Our Graphical Model Looks Like This



$$\mathbb{P}[X_1, \ldots, X_5, H_1, \ldots, H_5] =$$
$$\mathbb{P}[X_1, X_2|H_2] \times \mathbb{P}[X_4, X_5|H_4] \times \mathbb{P}[X_3, H_4|H_3]$$
$$\times \mathbb{P}[H_2, H_3|H_1] \times \mathbb{P}[H_1]$$

Constructing The Observable Factorization

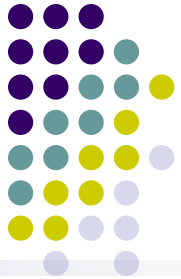


Like we did before

$$\mathcal{P}[X_{\{1,2\}}, X_{\{3,4,5\}}] = \mathcal{P}[X_{\{1,2\}}, X_3]\mathcal{P}[X_2, X_3]^{-1}\mathcal{P}[X_2, X_{\{3,4,5\}}]$$

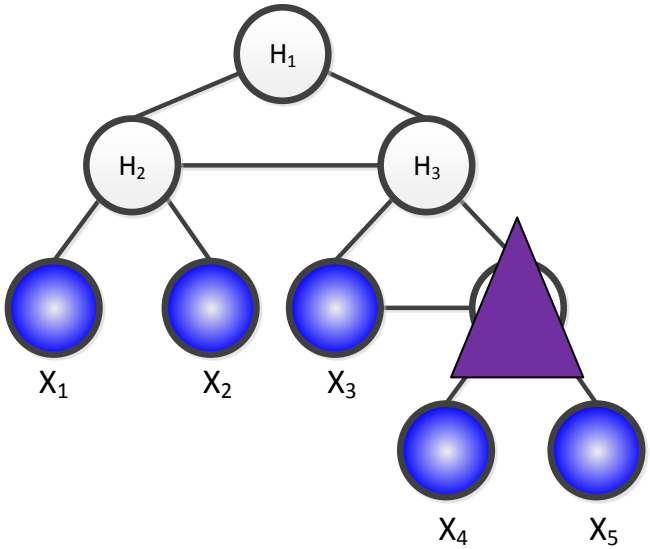
Decompose this recursively

Constructing The Observable Factorization



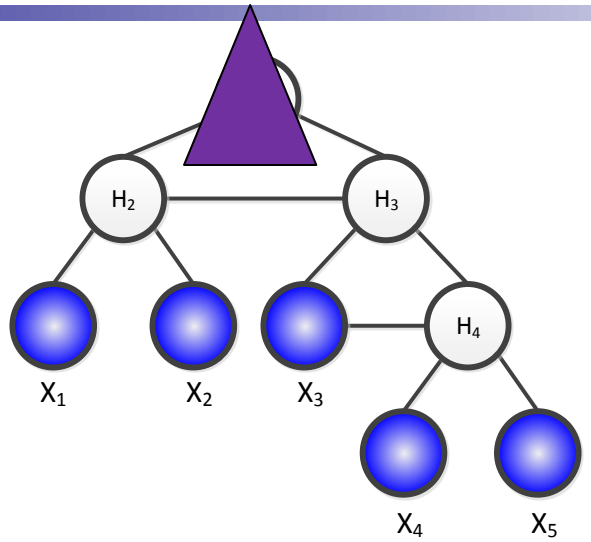
But first reshape:

$$\mathcal{P}[X_2, X_{\{3,4,5\}}] \Rightarrow \mathcal{P}[X_{\{2,3\}}, X_{\{4,5\}}]$$



$$\mathcal{P}[X_{\{2,3\}}, X_{\{4,5\}}] = \mathcal{P}[X_{\{2,3\}}, X_4]\mathcal{P}[X_3, X_4]^{-1}\mathcal{P}[X_3, X_{\{4,5\}}]$$

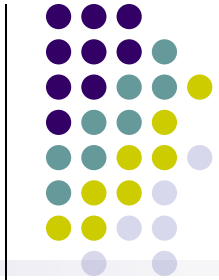
This is Redundant But….



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

- Having the additional factor makes some other comparisons easier

Our Observable Factorization

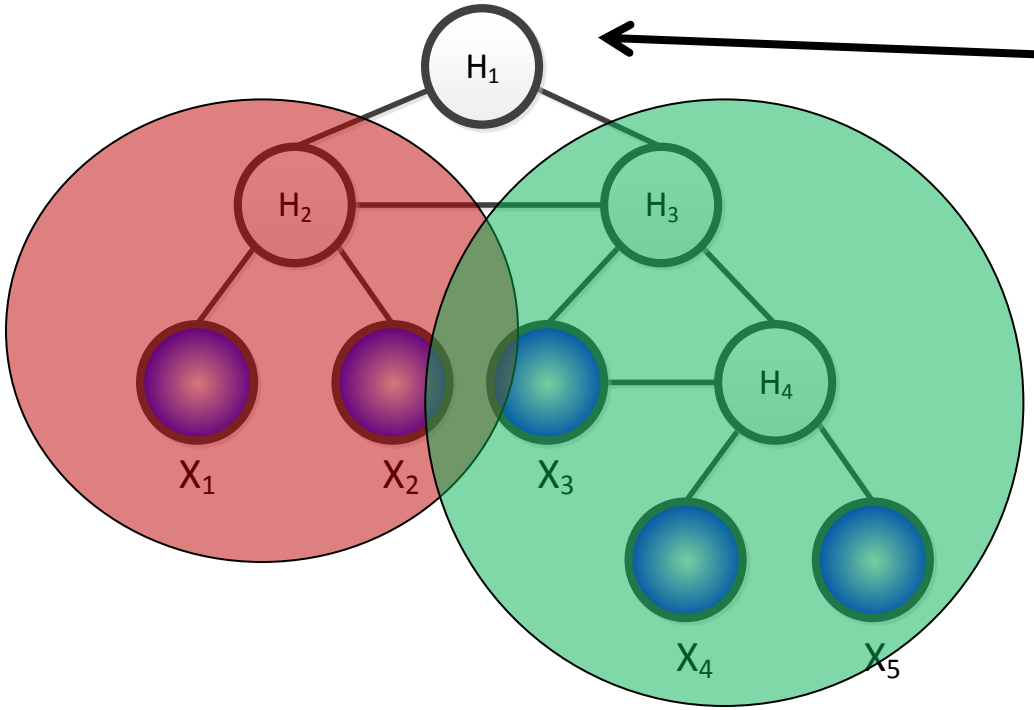


$$\mathcal{P}[X_{\{1,2\}}, X_{\{3,4,5\}}] = \mathcal{P}[X_{\{1,2\}}, X_3]\mathcal{P}[X_2, X_3]^{-1}\mathcal{P}[X_2, X_{\{3,4,5\}}]$$

$$\mathcal{P}[X_{\{2,3\}}, X_{\{4,5\}}] = \mathcal{P}[X_{\{2,3\}}, X_4]\mathcal{P}[X_3, X_4]^{-1}\mathcal{P}[X_3, X_{\{4,5\}}]$$

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

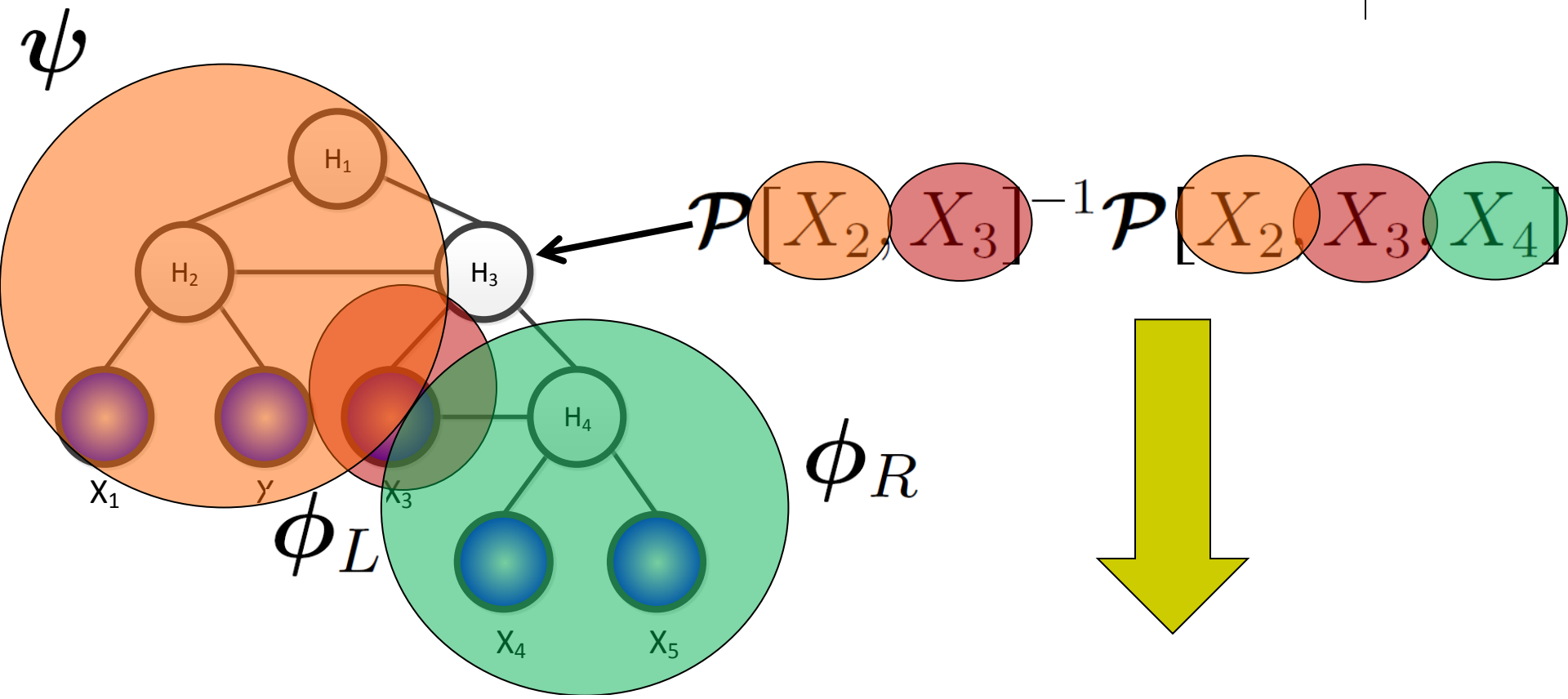
Intuitively,



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

Corresponds to parameter for root

Intuitively,



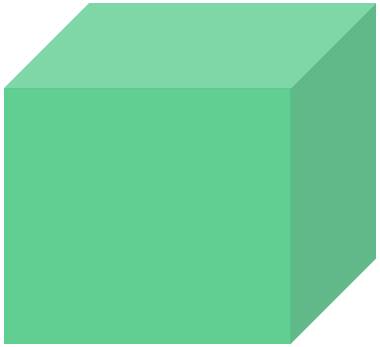
How To Add Back In The Rules

$$\mathtt{a} \rightarrow \mathtt{b}, \mathtt{c}$$

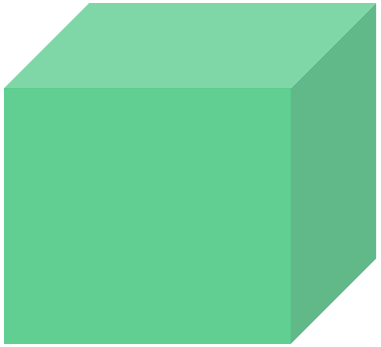
$$\mathbb{P}[X_3, b = \texttt{NP}, H_4 | a = \texttt{VP}, H_3]$$

Instead of thinking of this as a factor of 5 variables, think of it as several factors of 3 variables, one for each combination of a,b

$$\boldsymbol{\mathcal{P}}[X_3, b = \texttt{NP}, H_4 | a = \texttt{VP}, H_3]$$



$$\boldsymbol{\mathcal{P}}[X_3, b = \texttt{NP}, H_4 | a = \texttt{NP}, H_3]$$



How To Add Back In The Rules

- As a result instead of just one

$$C = \mathbb{E}[\psi \otimes \phi_L]^{-1}\mathbb{E}[\psi \otimes \phi_L \otimes \phi_R]$$

- We will have one for every choice of **a,b** i.e.

$$C^{a,b} = \mathbb{E}[\psi^{a,b} \otimes \phi_L^{a,b}]^{-1}\mathbb{E}[\psi^{a,b} \otimes \phi_L^{a,b} \otimes \phi_R^{a,b}]$$