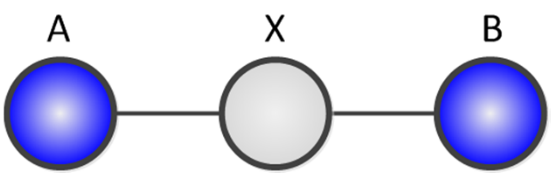
School of Computer Science
Carnegie Mellon

Probabilistic Graphical Models

Spectral Learning for Graphical Models

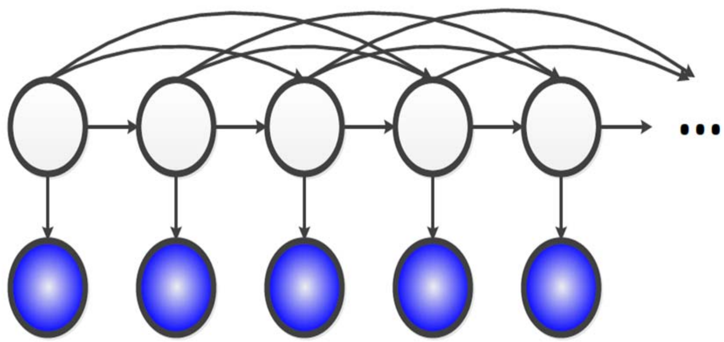
Eric Xing

Lecture 26, April 21, 2015

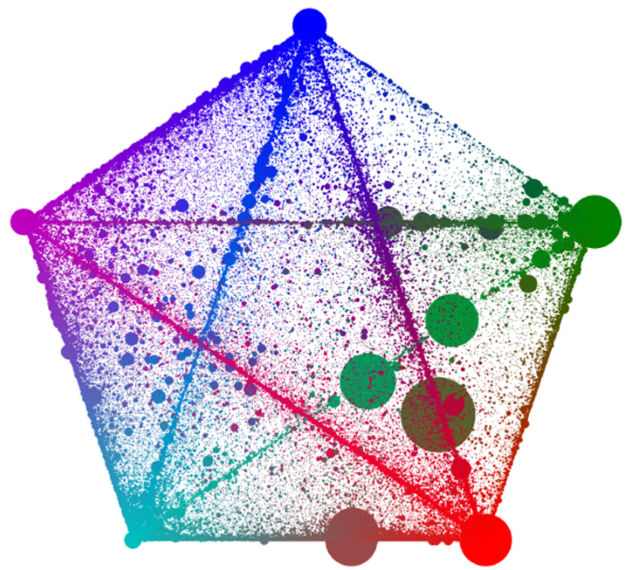


Acknowledgement: slides drafted by Ankur Parikh

Latent Variable Models



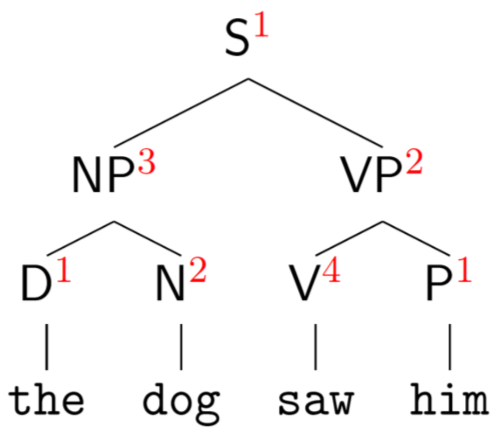
Sequence models



Ho. et al. 2012

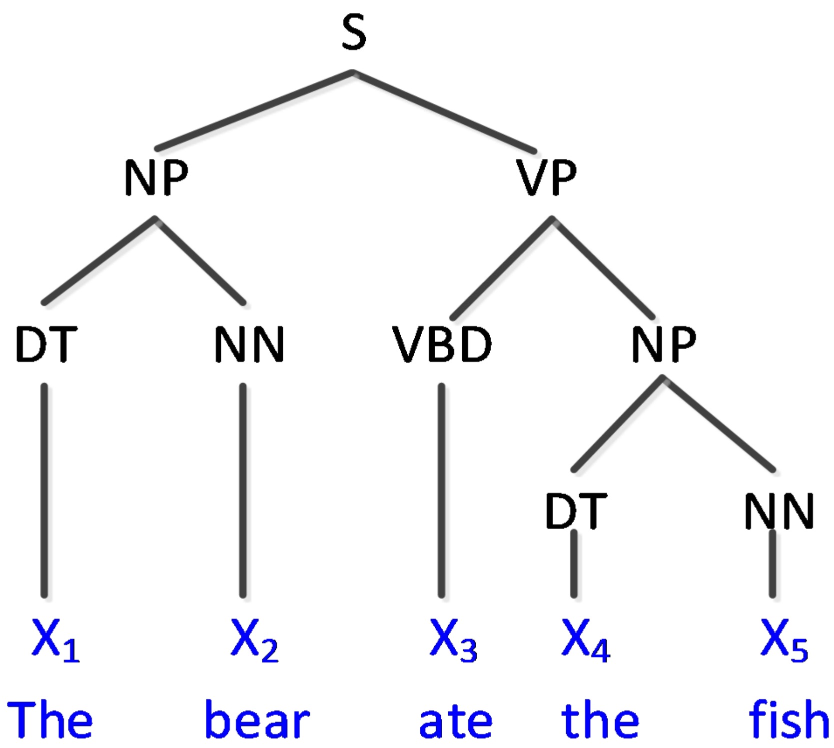
Mixed membership models

Parsing

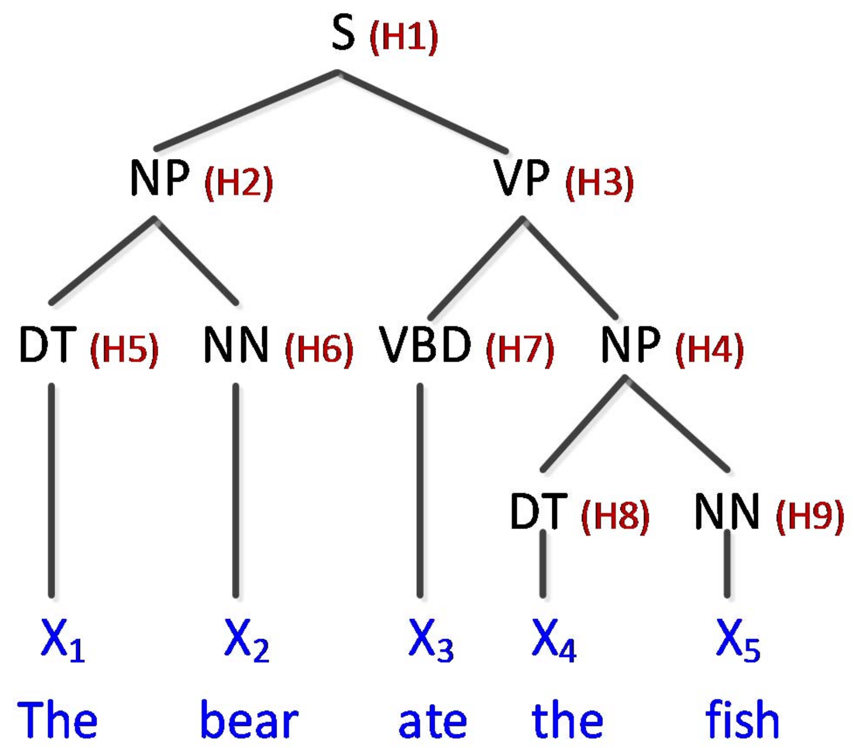


Latent Variable PCFG [Matsuzaki et al., 2005, Petrov et al. 2006]

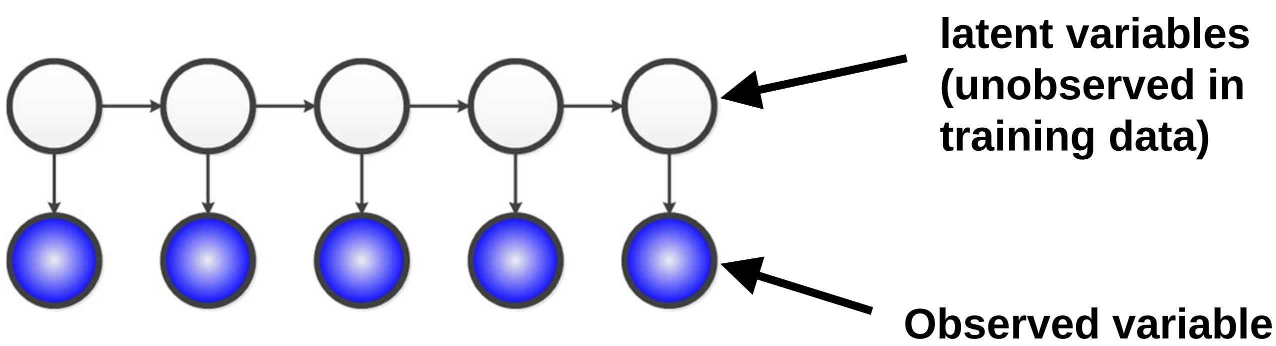
PCFG



Latent Variable PCFG



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 of 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)

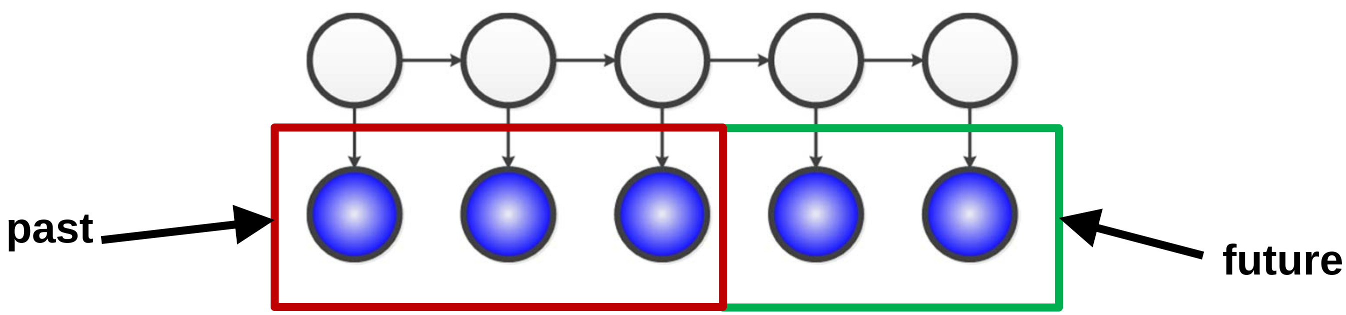
Related References

- 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

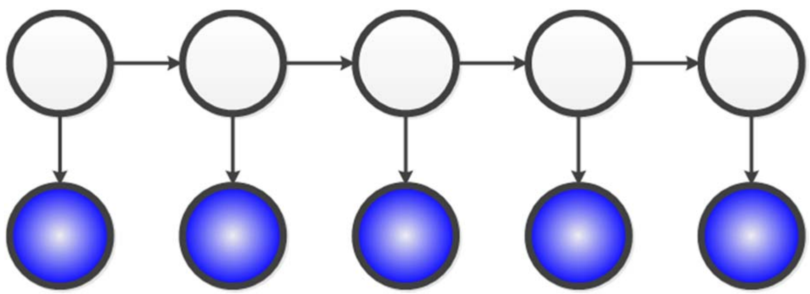


Focusing on Prediction

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

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

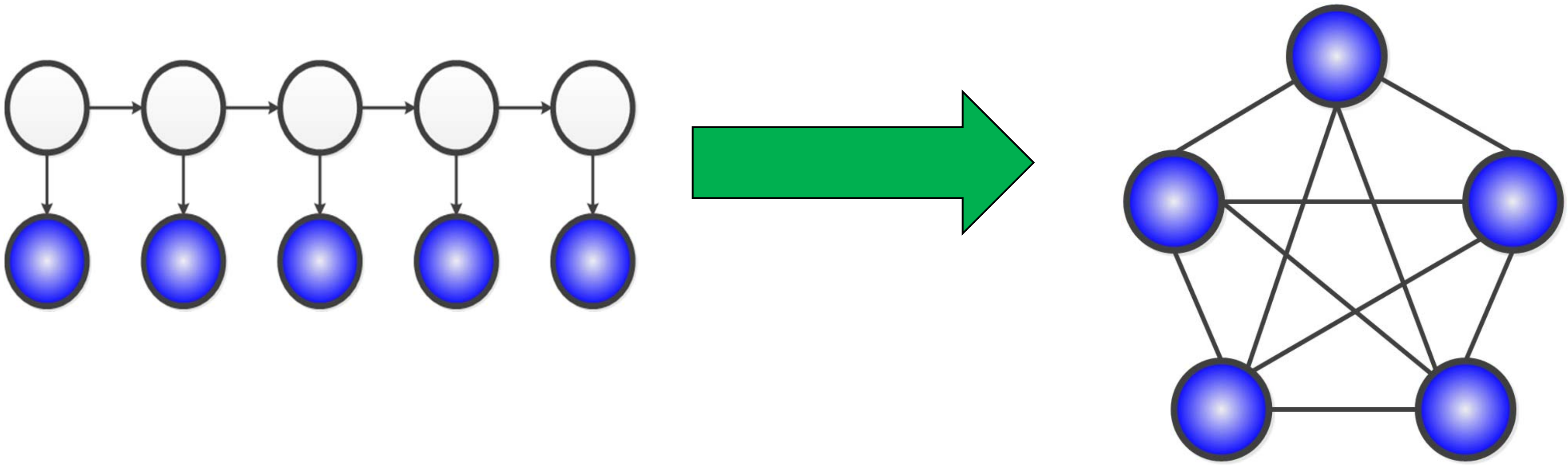
- We do not care about the latent variables explicitly.



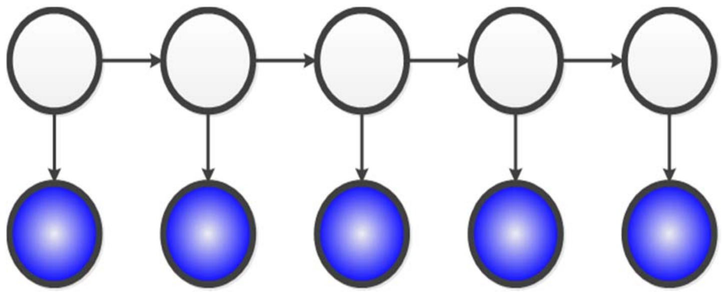
- **Do we still need EM to learn the parameters?**

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



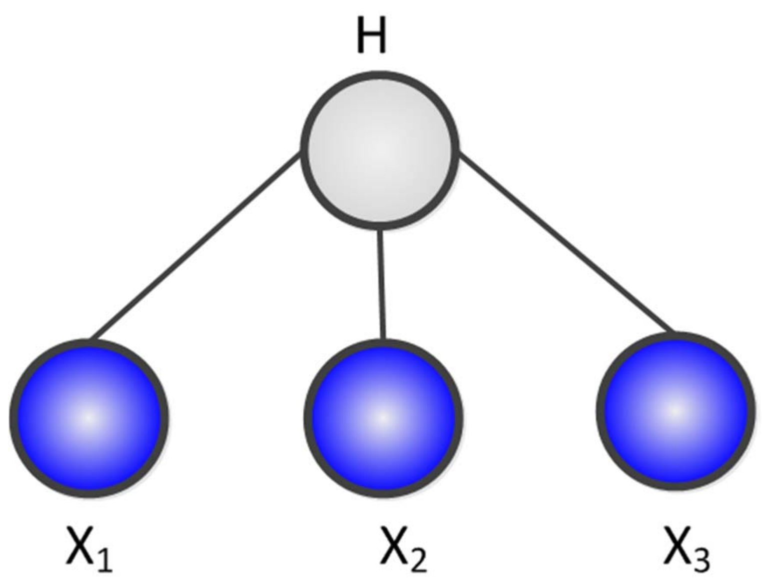
$$\mathbb{P}[X_1, X_2, X_3, X_4, X_5] = \sum_{H_1,\ldots,H_5} \mathbb{P}[H_1]\mathbb{P}[H_1] \prod_{i=2}^{5} \mathbb{P}[H_i|H_{i-1}] \prod_{i=1}^{5} \mathbb{P}[X_i|H_i]$$

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?

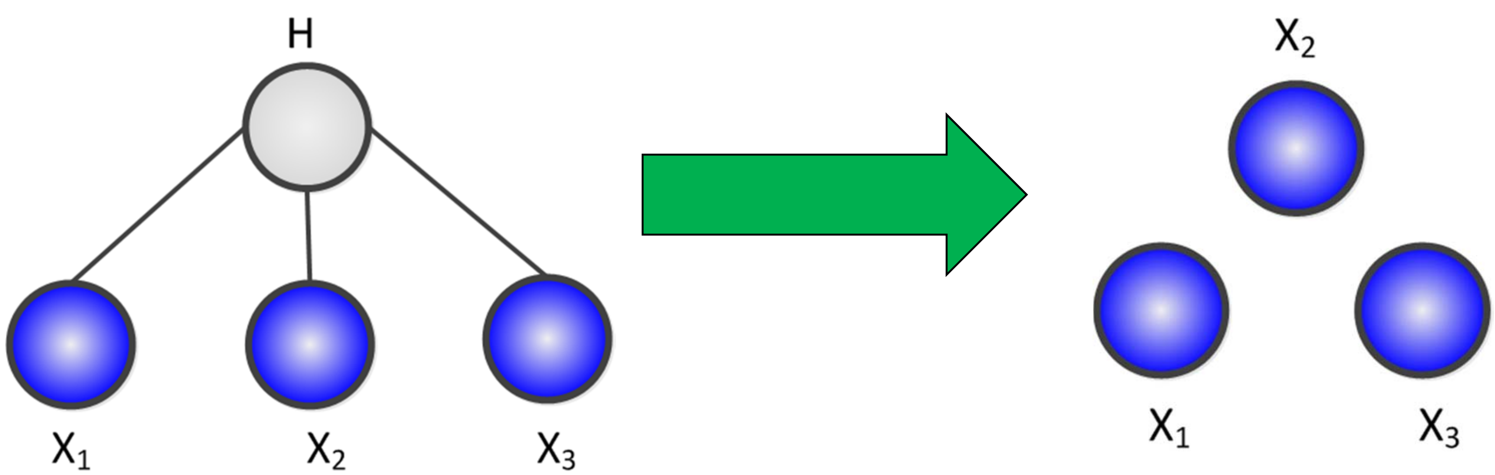
- It depends on the number of latent states.

- Consider the following model.



If H has only one state.....

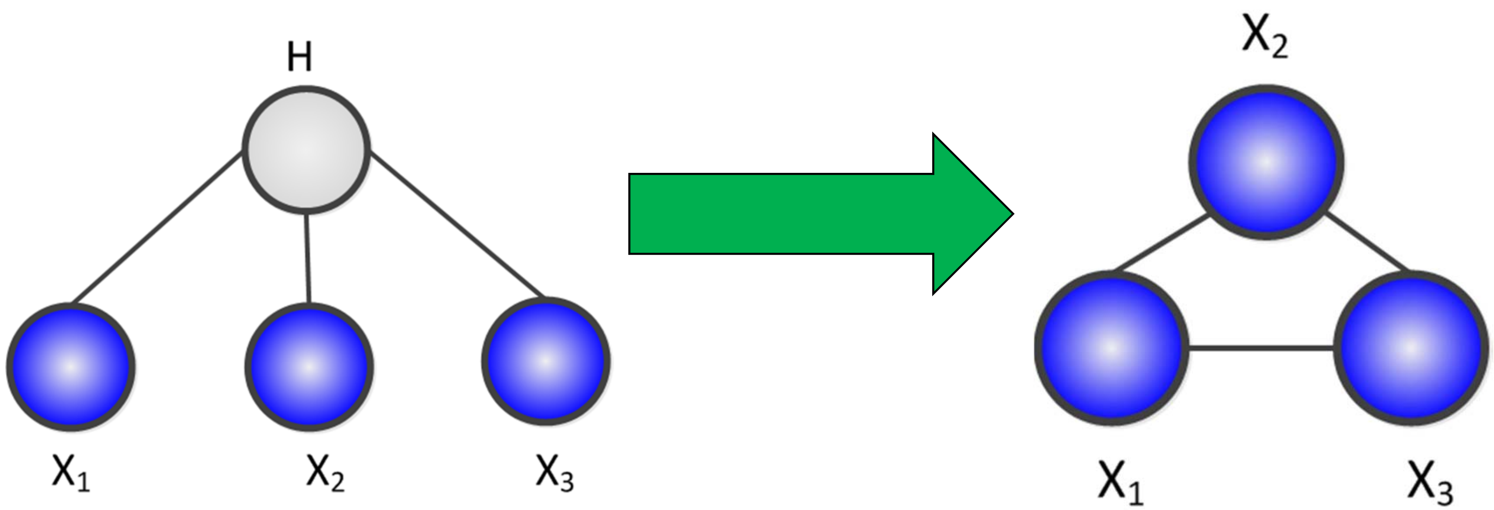
- Then the observed variables are independent!



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.

Sum Rule (Matrix Form)

- Sum Rule

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

- Equivalent view using Matrix Algebra

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

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

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

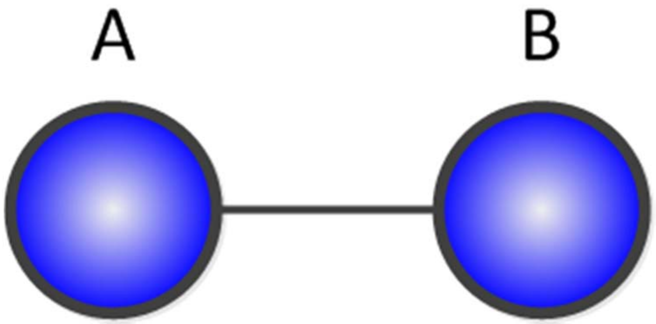
Means on diagonal

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

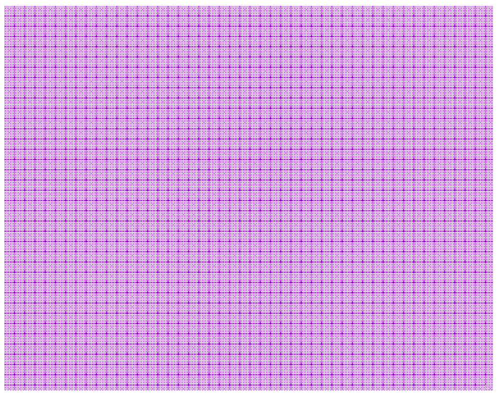
$$\begin{pmatrix} \mathbb{P}[X=0,Y=0] & \mathbb{P}[X=0,Y=1] \\ \mathbb{P}[X=1,Y=0] & \mathbb{P}[X=1,Y=1] \end{pmatrix} = \begin{pmatrix} \mathbb{P}[X=0|Y=0] & \mathbb{P}[X=0|Y=1] \\ \mathbb{P}[X=1|Y=0] & \mathbb{P}[X=1|Y=1] \end{pmatrix} \times \begin{pmatrix} \mathbb{P}[Y=0] & 0 \\ 0 & \mathbb{P}[Y=1]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

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

Graphical Models: The Linear Algebra View



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

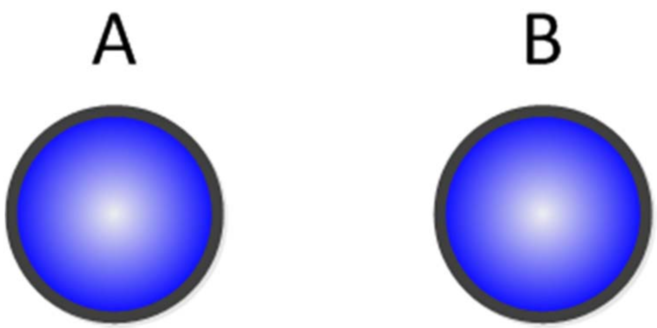


A and B have m states each.

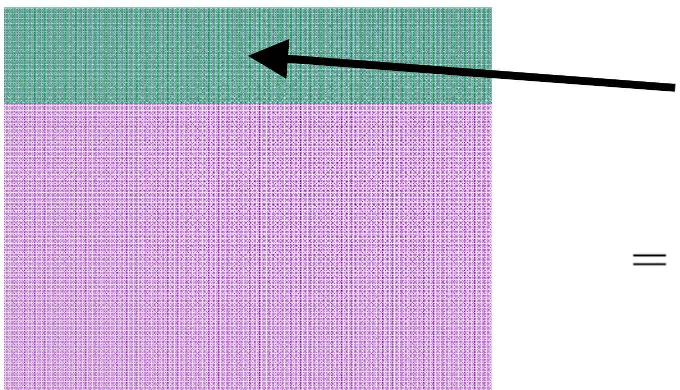
- In general, nothing we can say about the nature of this matrix.

Independence: The Linear Algebra View

- What if we know A and B are independent?



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

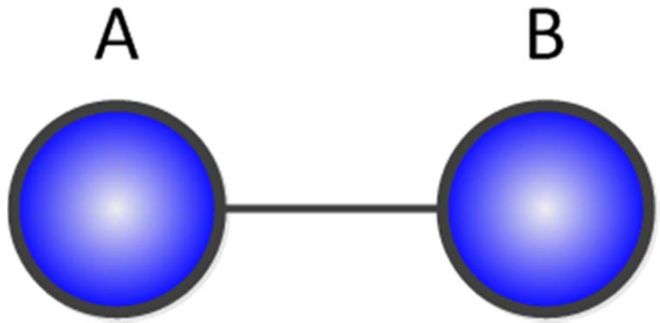


$$\left(\mathbb{P}[A=1, B=1], \ldots, \mathbb{P}[A=1, B=m]\right)$$

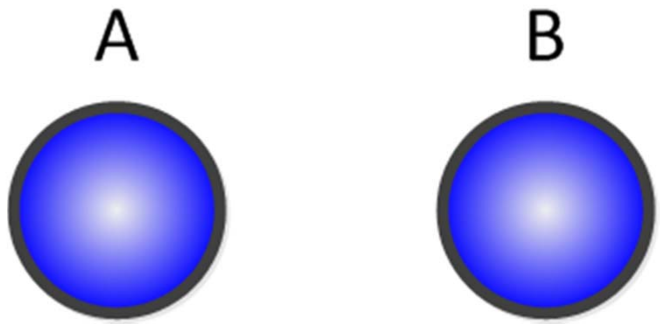
$$= \left(\mathbb{P}[A=1](\mathbb{P}[B=1], \ldots, \mathbb{P}[B=m])\right)$$

- 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)**

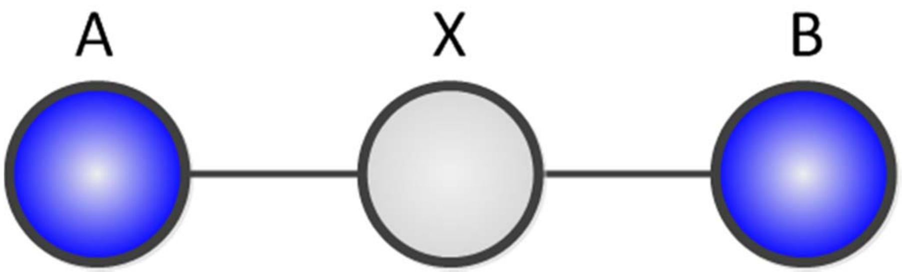


$\mathcal{P}[A, B]$ **has rank 1**

- **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***).

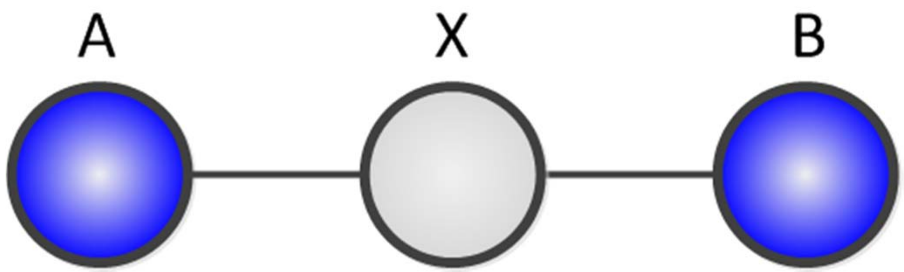


- Assume ***X*** has ***k*** states (while ***A*** and ***B*** have ***m*** states).

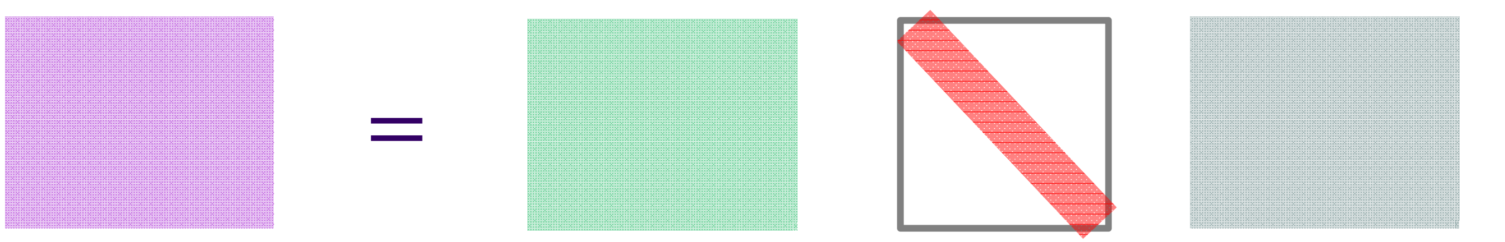
- Then, $rank(\boldsymbol{\mathcal{P}}[A, B]) \leqslant k$

- Why?

Low Rank Structure



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



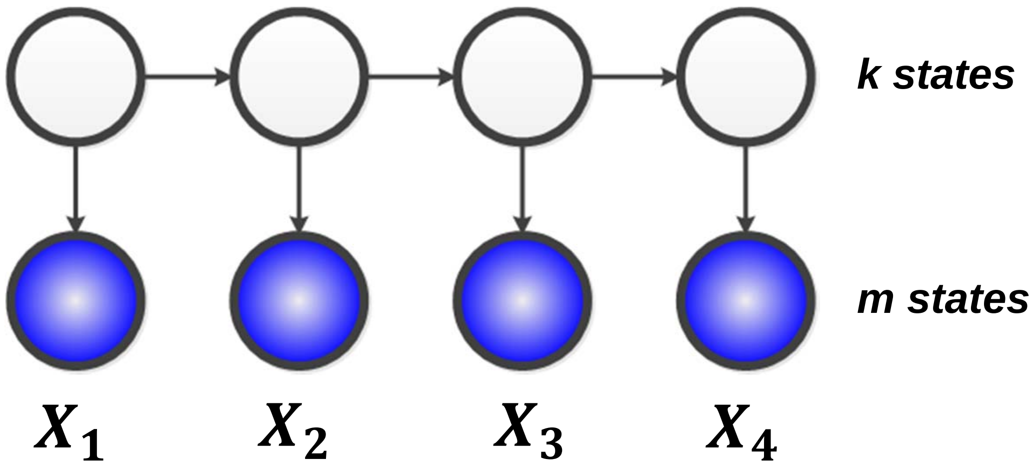
$rank \leq k$ $\quad$ $rank \leq k$ $\quad$ $rank \leq k$ $\quad$ $rank \leq k$

The Spectral View

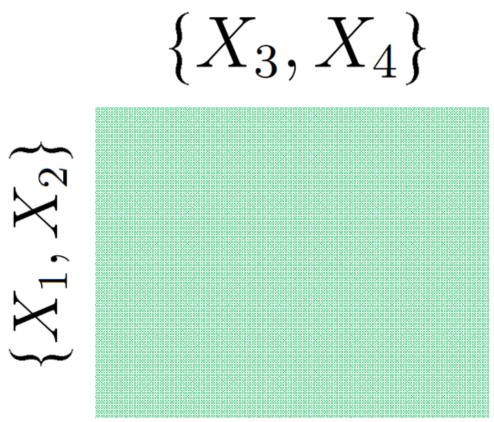
- Latent variable models encode low rank dependencies among variables *(both marginal and conditional)*

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

A More Interesting Example



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



has rank k

Low Rank Matrices "Factorize"

$$M = LR$$

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:

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

- **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

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

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

- 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

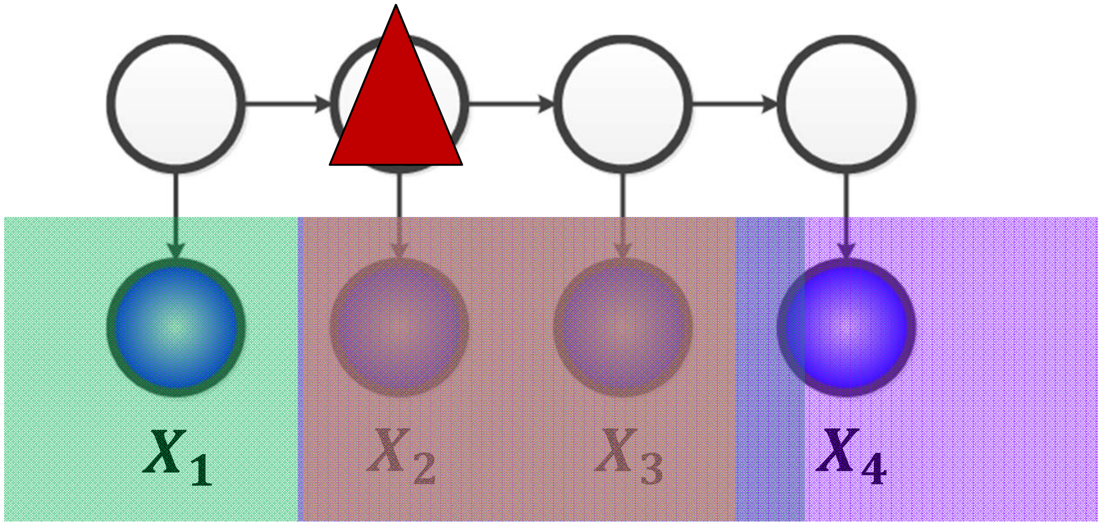
Advantage: Factors are only functions of observed variables! Can be directly computed from data without EM!!!!

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

We will call this factorization the observable factorization.

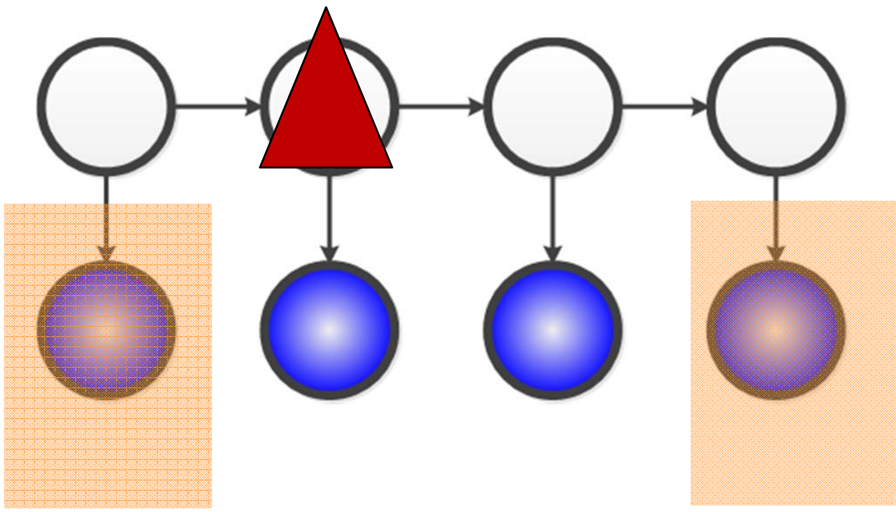
Graphical Relationship

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



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?

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

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

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

Can I choose S to get the observable factorization?

Relationship to Original Factorization

- Let

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

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

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

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

Training / Testing with Spectral Learning

- We have that

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

- In training, we compute estimates:

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

- In test time, we can compute probability estimates (let lowercase letters denote fixed evidence values):

$$\widehat{\mathbb{P}}_{spec}[x_1, x_2, x_3, x_4] = \mathcal{P}_{MLE}[x_{\{1,2\}}, X_3]\mathcal{P}_{MLE}[X_2, X_3]^{-1}\mathcal{P}_{MLE}[X_2, x_{\{3,4\}}]^\top$$

Generalizing To More Variables

- Consider HMM with 5 observations. Using similar arguments as before we will get that:

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

reshape and decompose recursively

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

Consistency

- A trivial consistent estimator is to simply attempt to estimate the “big” probability table from the data without making any conditional independence assumptions

$$\mathcal{P}_{MLE}[X_1, X_2; X_3, X_4] \rightarrow \mathcal{P}[X_1, X_2; X_3, X_4]$$

as number of samples increases

- While this is consistent, it is not very statistically efficient

Consistency

- A better estimate is to compute likelihood estimates of the factorization:

$$\mathcal{P}_{MLE}[X_{\{1,2\}}|H_2]\mathcal{P}_{MLE}[\oslash H_2]\mathcal{P}_{MLE}[X_{\{3,4\}}|H_2]^\top$$
$$\rightarrow \mathcal{P}[X_1, X_2; X_3, X_4]$$

- But this requires running EM, which will get stuck in local optima and is not guaranteed to obtain the MLE of the factorized model

Consistency

- In spectral learning, we estimate the alternate factorization from the data

$$\mathcal{P}_{MLE}[X_{\{1,2\}}, X_3]\mathcal{P}_{MLE}[X_2, X_3]^{-1}\mathcal{P}_{MLE}[X_2, X_{\{3,4\}}]$$
$$\rightarrow \mathcal{P}[X_1, X_2; X_3, X_4]$$

- This is consistent and computationally tractable (at some loss of statistical efficiency due to the dependence on the inverse)

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

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

- 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)

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

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

When k > m

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

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

- 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…….

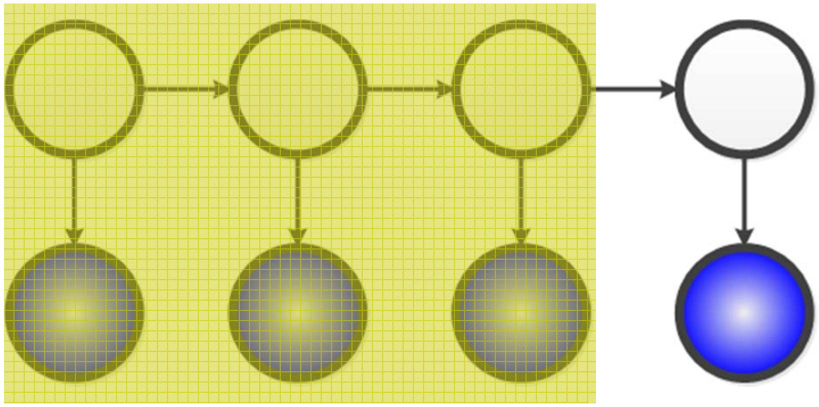
How many hidden states does HMM need?

- 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?

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.

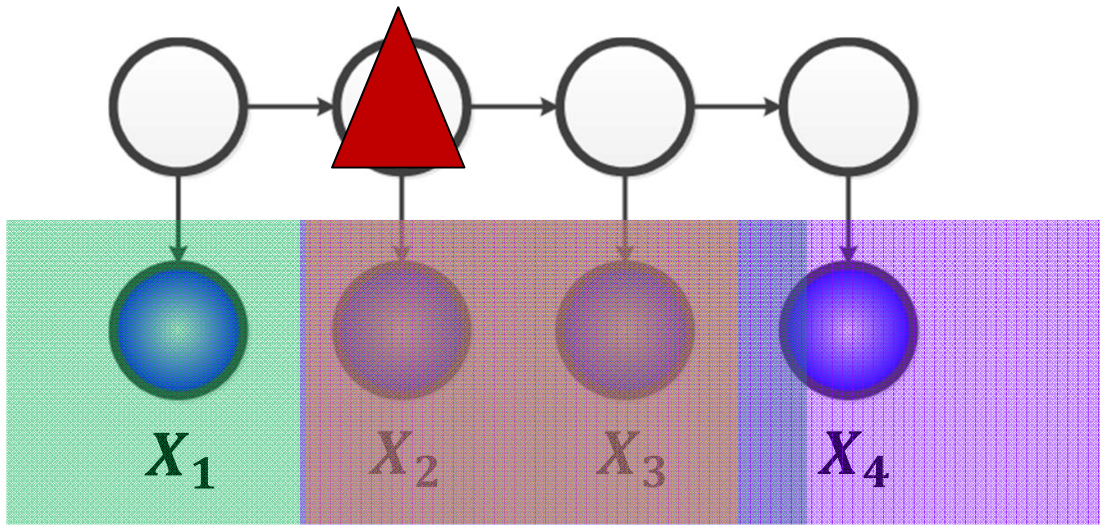
- Only works when ***k* < *m***.



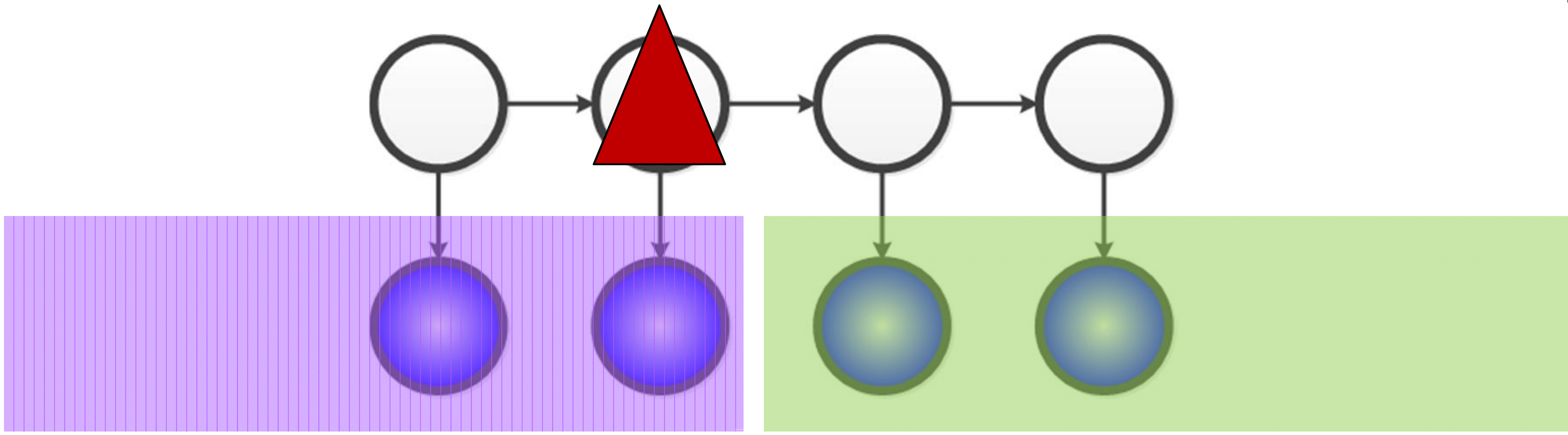
- However, in real problems we need to capture longer range dependencies.

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

$$\phi_R$$

Spectral Learning With Features

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

Use more complex feature instead:

$$\mathbb{E}[\phi_L \otimes \phi_R]$$

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

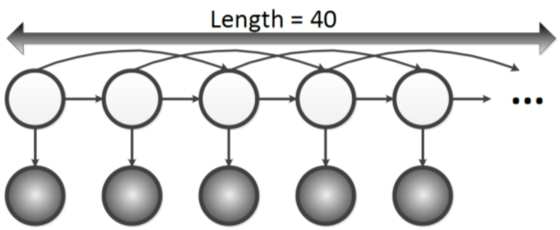
Experimentally,

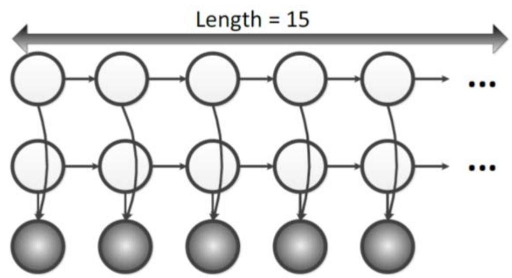
- Has been shown by many authors that (with some work) spectral methods achieve comparable results to EM but are 10-50x faster
  - Parikh et al. 2011 / 2012
  - Balle et al. 2012
  - Cohen et al. 2012 / 2013

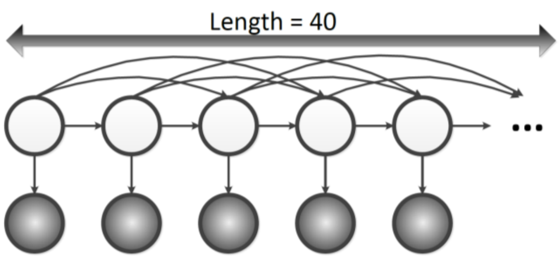
- The following are some synthetic and real data results demonstrating the comparison between EM and spectral methods.

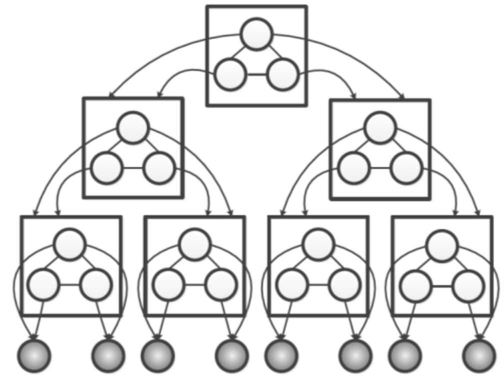
Synthetic Data [Parikh et al. 2012]

- Different latent variable models





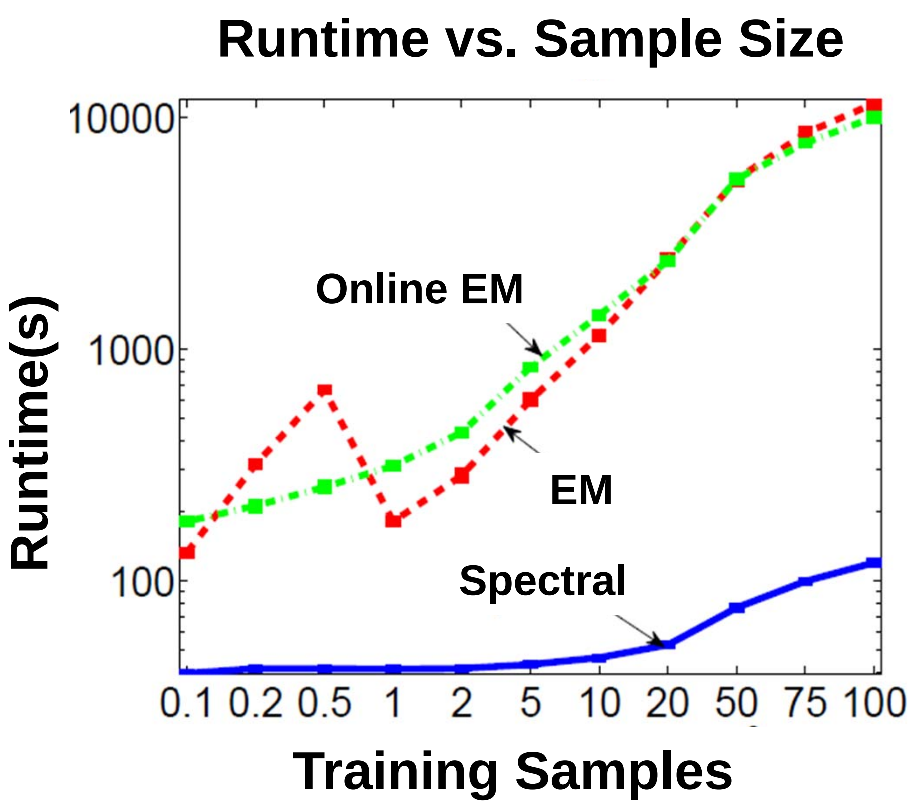


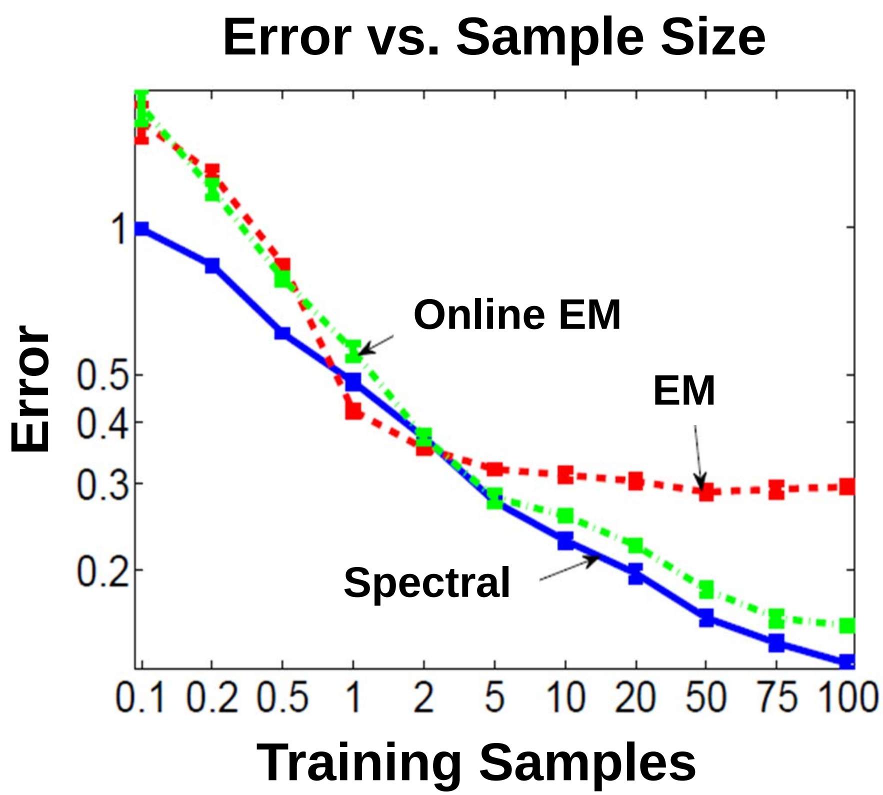


- Train: Learn parameters for a given model given samples of observed variables
- Test: Evaluate likelihood of random samples drawn from model and compare to the true likelihood

Synthetic Data [Parikh et al. 2012]

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

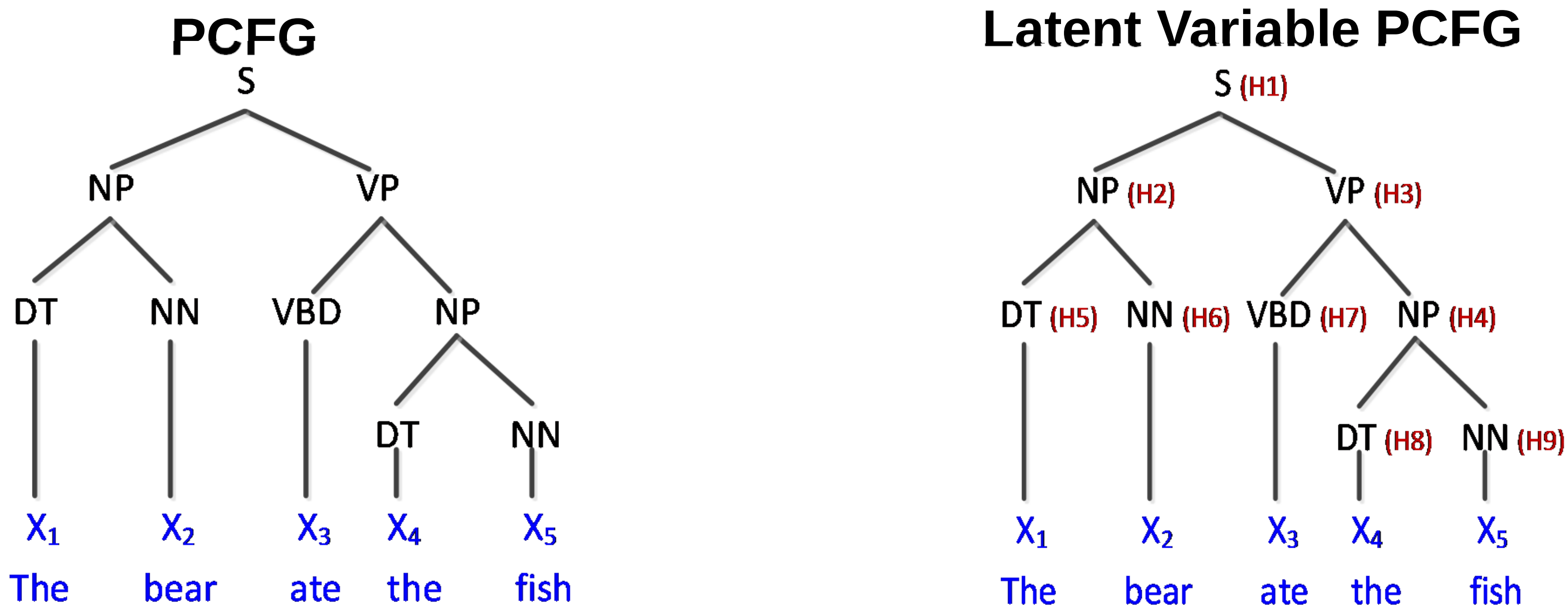




- Results for other structures look similar

Supervised Parsing [Cohen et al. 2012/2013]

- Learn a latent variable Probabilistic Context Free Grammar model (latent PCFG) which is a PCFG augmented with additional latent states



- Train: Learn parameters given parse trees on training examples.
- Test: Estimate most likely parse structure on test sentences

Empirical Results for Latent PCFGs [Cohen et al. 2013]

	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

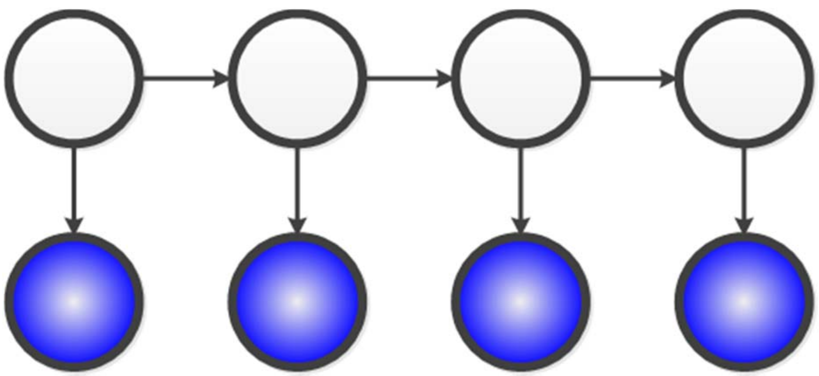
Evaluation Measure: *F1 bracketing score*

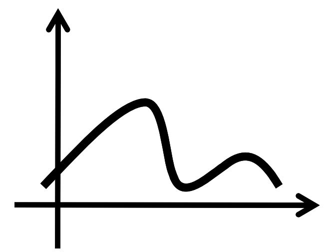
Timing Results on Latent PCFGs[Cohen et al. 2013]

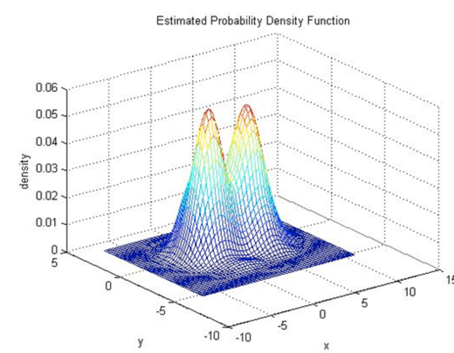
	single EM iter.	EM best model	spectral algorithm total	feature	transfer + scaling	SVD	$a \to b\ c$	$a \to x$
$m = 8$	6m	3h	3h32m	22m	49m	36m	1h34m	10m
$m = 16$	52m	26h6m	5h19m			34m	3h13m	19m
$m = 24$	3h7m	93h36m	7h15m			36m	4h54m	28m
$m = 32$	9h21m	187h12m	9h52m			35m	7h16m	41m

Dealing with Nonparametric, Continuous Variables

- It is difficult to run EM if the conditional/marginal distributions are continuous and do not easily fit into a parametric family.







- However, we will see that Hilbert Space Embeddings can easily be combined with spectral methods for learning nonparametric latent models.

Connection to Hilbert Space Embeddings

- Recall that we could substitute features for variables

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

Use more complex feature instead:

$$\mathbb{E}[\phi_L \otimes \phi_R]$$

Can Also Use Infinite Dimensional Features

- Replace

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

- with

$$\mathcal{C}[X_2, X_3] = \mathbb{E}[\phi_{X_2} \otimes \phi_{X_3}]$$

covariance operator

- (and similarly for other quantities)

Connection to Hilbert Space Embeddings

Discrete case:

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

Continuous case:

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

Summary - EM & Spectral (Part I)

EM	Spectral
Aims to Find MLE so more "statistically" efficient	Does not aim to find MLE so less statistically efficient.
Can get stuck in local-optima	Local-optima-free
Lack of theoretical guarantees	Provably consistent
Slow	Very fast
Easy to derive for new models	Challenging to derive for new models (Unknown whether it can generalize to arbitrary loopy models)

Summary - EM & 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