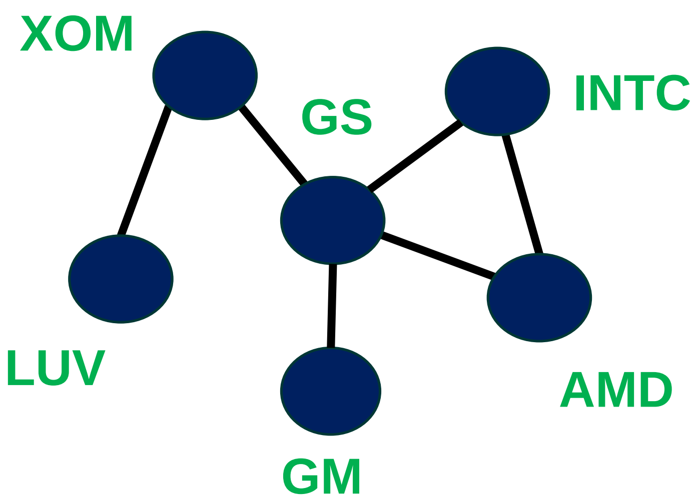
Laboratory for Statistical Artificial InteLligence & INtegrative Genomics

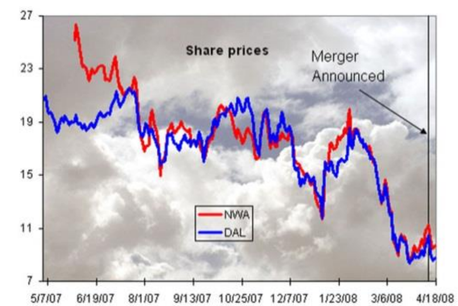
A Spectral Algorithm for Latent Tree Graphical Models

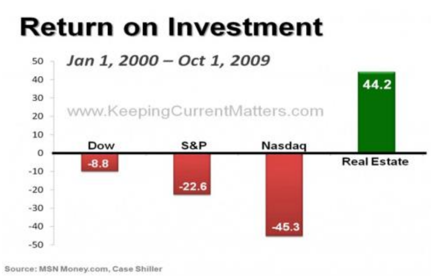
Ankur P. Parikh, Le Song, Eric P. Xing
School of Computer Science
Carnegie Mellon University

Probabilistic Graphical Models

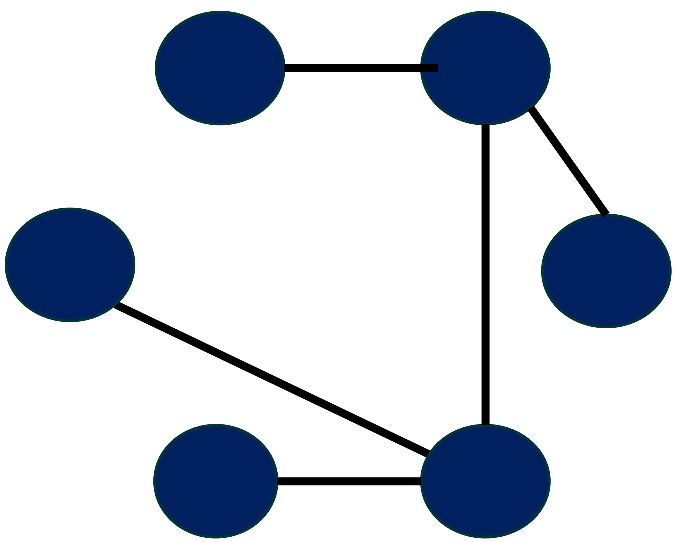
Ubiquitous in many applications, where it is necessary to model structure and dependencies among a set of variables.





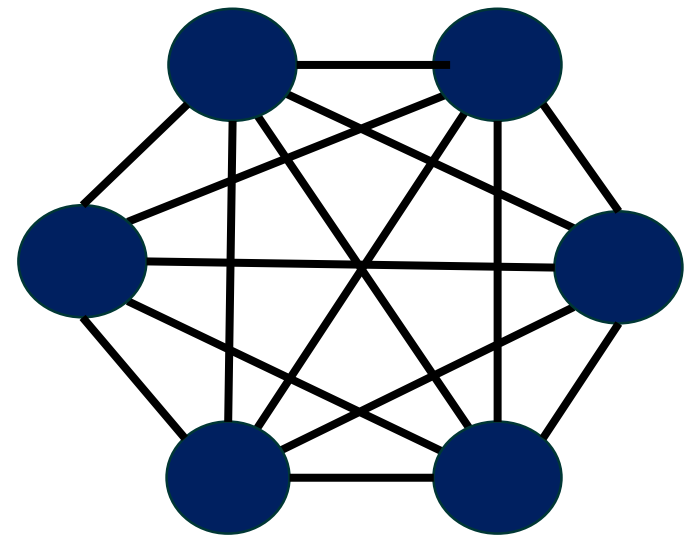


Tree Graphical Models



- Very restrictive model
- Structure Learning – Easy
- Parameter learning – Easy
- Inference – Easy

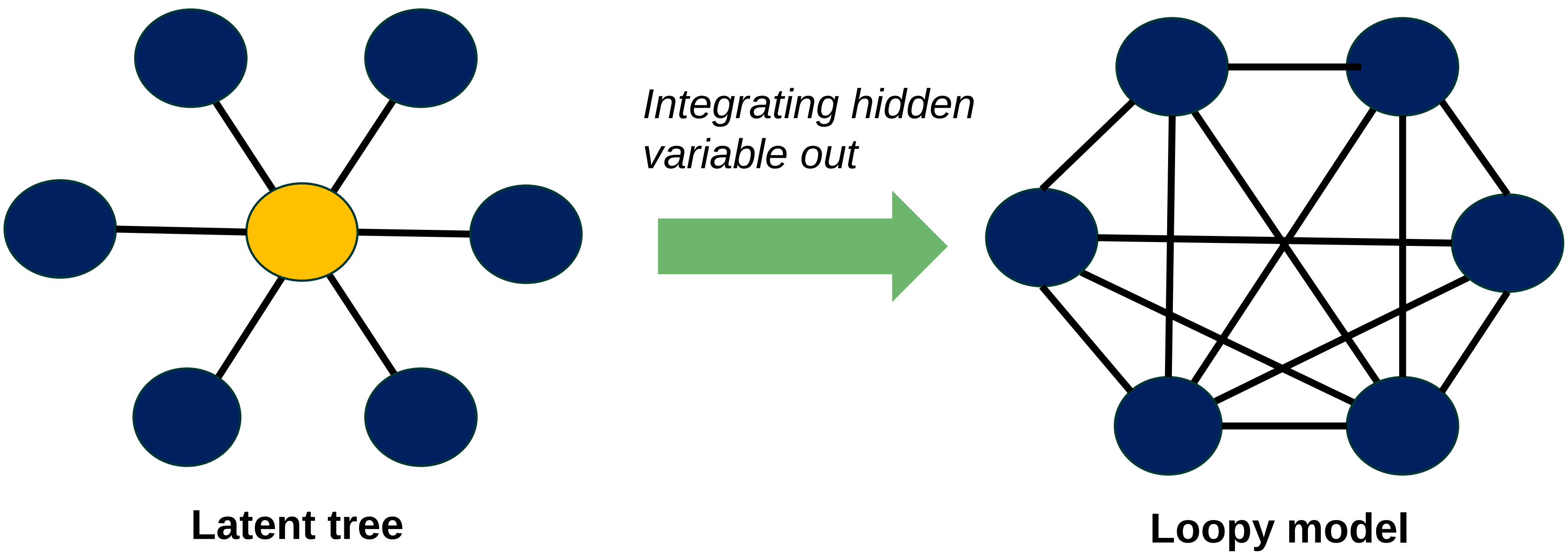
Loopy Graphical Models



- Very rich model
- Structure Learning – Hard
- Parameter learning – Hard
- Inference – Hard

Latent Tree Graphical Models

Add additional unobserved variables to enrich flexibility of model



- **Reasonably rich model**
- **Structure Learning – Tractable (Choi et al. 2010)**
- **Parameter learning/Inference – ???**

Expectation Maximization

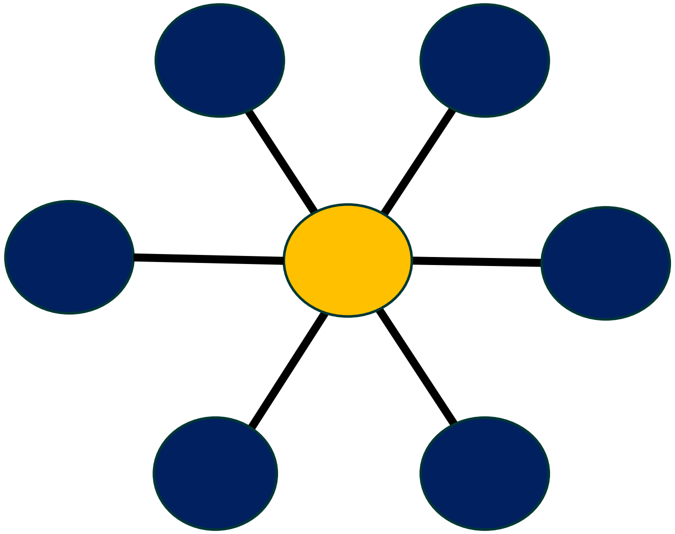
- Recovers parameters explicitly and therefore can recover hidden states
- Slow – First Order Optimization Method
- Local Minima
- Lack of Theoretical Guarantees

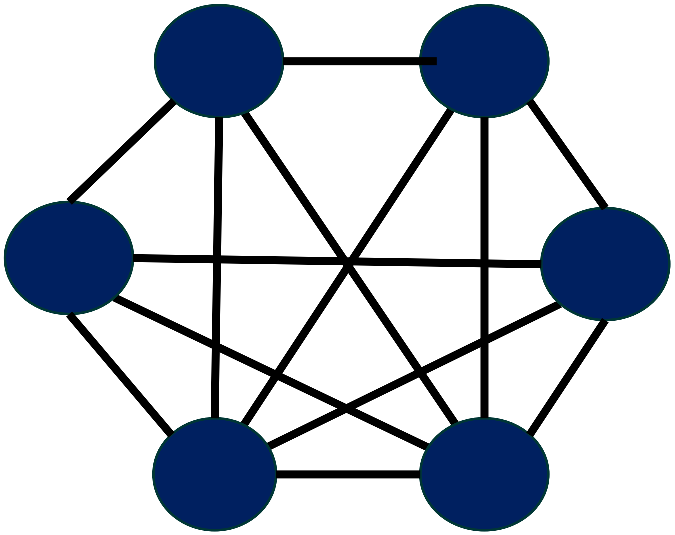
Spectral Algorithm

- Does not explicitly recover parameters, so cannot recover hidden states (We can only compute observed marginals).
- (Very) Fast – No optimization needed
- Local Minima Free
- Consistent

Focusing on Inference

- Explicitly recovering the hidden states makes the problem fundamentally non-convex.
- But in many applications the goal is to simply do prediction (i.e. compute marginals among observed variables)





Spectral Algorithm

- Do NOT explicitly learn latent parameters
- Instead learn “transformed” version of parameters.

$$P = AB$$

A and B depend on hidden variables

$$P = ASS^{-1}B$$

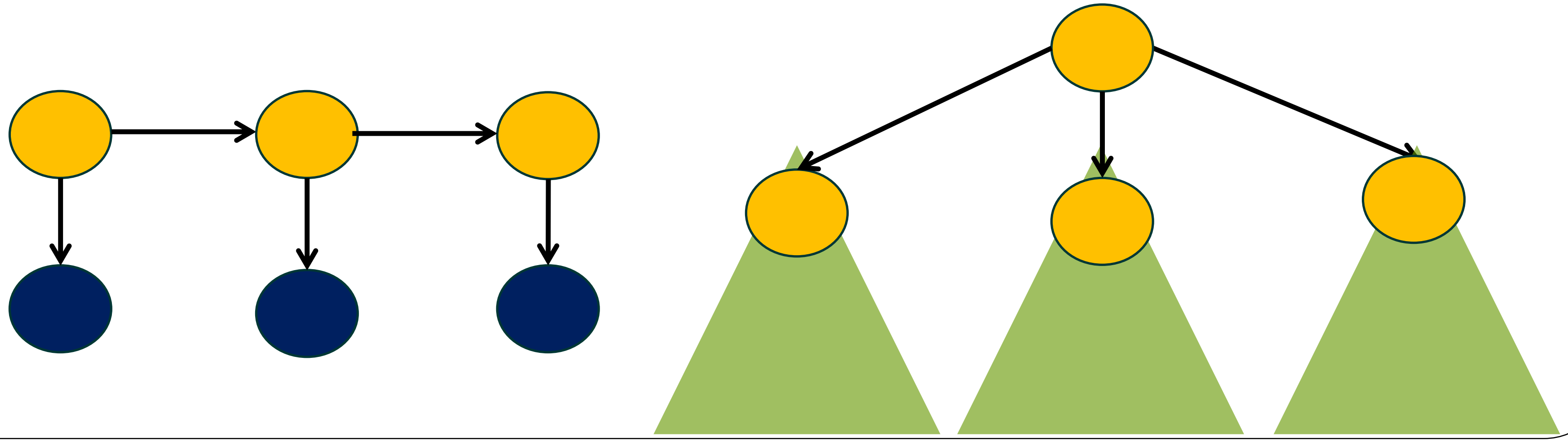
Key: Construct **S** such that **AS** and **S^{-1} B** only depend on observed variables. Then we can easily compute **P** (without ever learning **A**, **B**, or **S** individually)

Underlying dependence on spectral properties is what gives the method the name spectral algorithm.

Related Work

- Hsu et al. – Spectral Algorithm for HMMs
- Boots et al. - Reduced Rank HMMs
- Song et al. - Kernelized Spectral Method for nonparametric HMMs

- **Challenges** for Latent Trees
  - Topology significantly more complex
  - Not every hidden variable has an observed neighbor



Algorithm Overview

Latent Tree Representation

Compute joint probability as sequence of matrix multiplications

Transformed Representation

Insert transform matrices so that we can estimate transformed quantities instead of actual quantities

Observable Representation

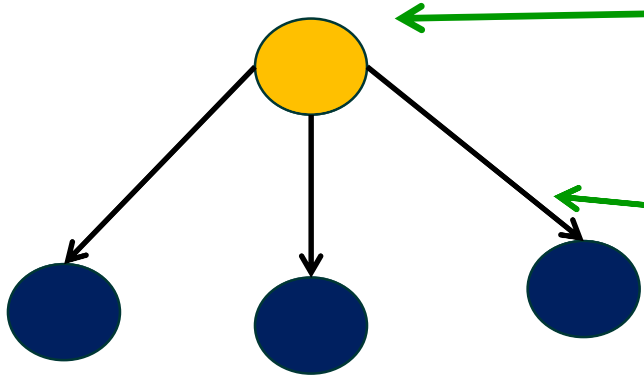
Prove that these transformed quantities are functions of observed variables.

Algorithm Overview

Latent Tree Representation

Compute joint probability as sequence of matrix multiplications

Parametrizing Latent Trees



Marginal root vector

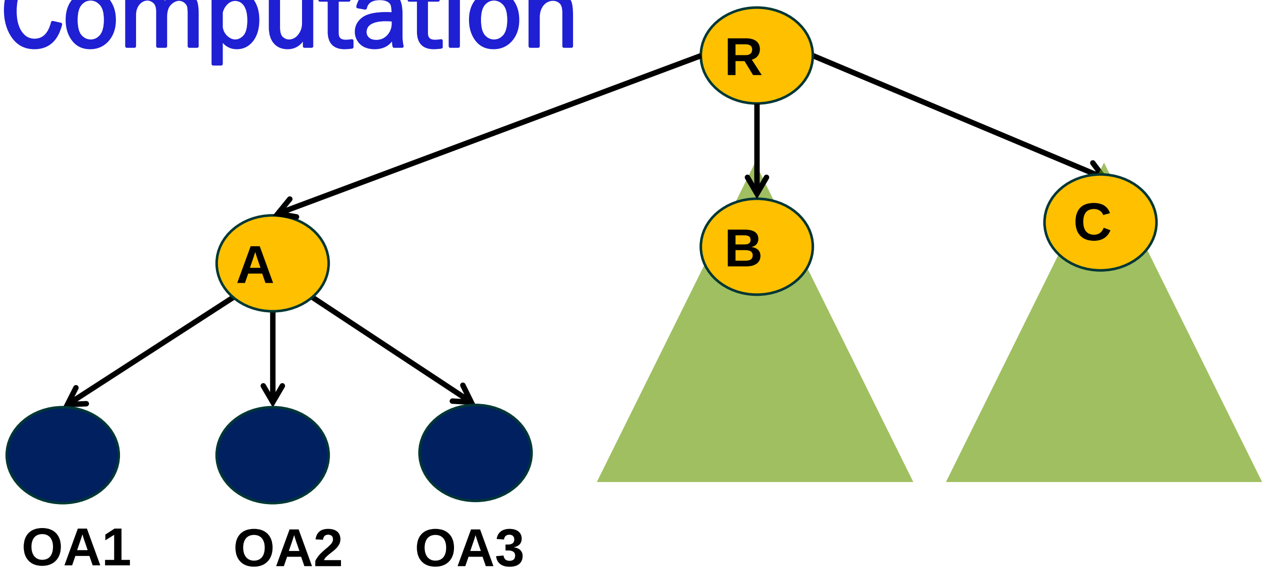
$$\begin{pmatrix} Pr[R=k] \\ \cdots \\ \cdots \end{pmatrix}$$

Conditional probability table

$$\begin{pmatrix} Pr[X=i|R=j] & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix}$$

- All hidden nodes are internal
- Conditional Probability Tables (CPTs) **cannot be directly estimated from data**

Representing Joint Probability Computation

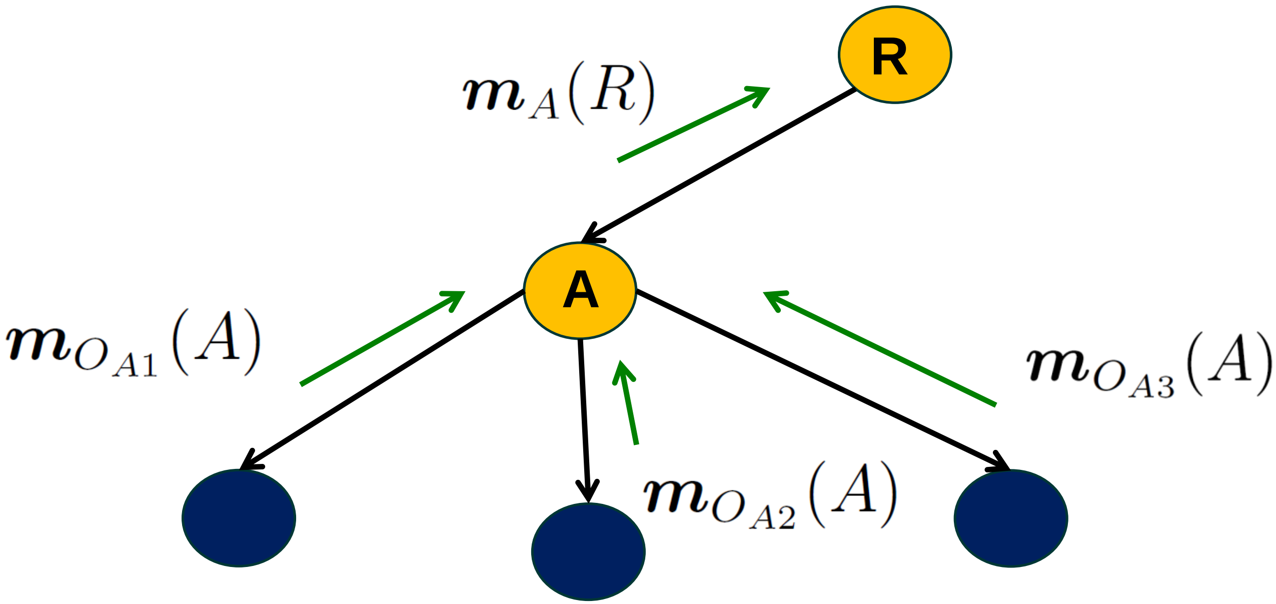


We would like to compute the following joint probability of all observed variables:

$$\mathbb{P}[O_{A1} = o_{A1}, O_{A2} = o_{A2}, O_{A3} = o_{A3}, \ldots\ldots\ldots]$$

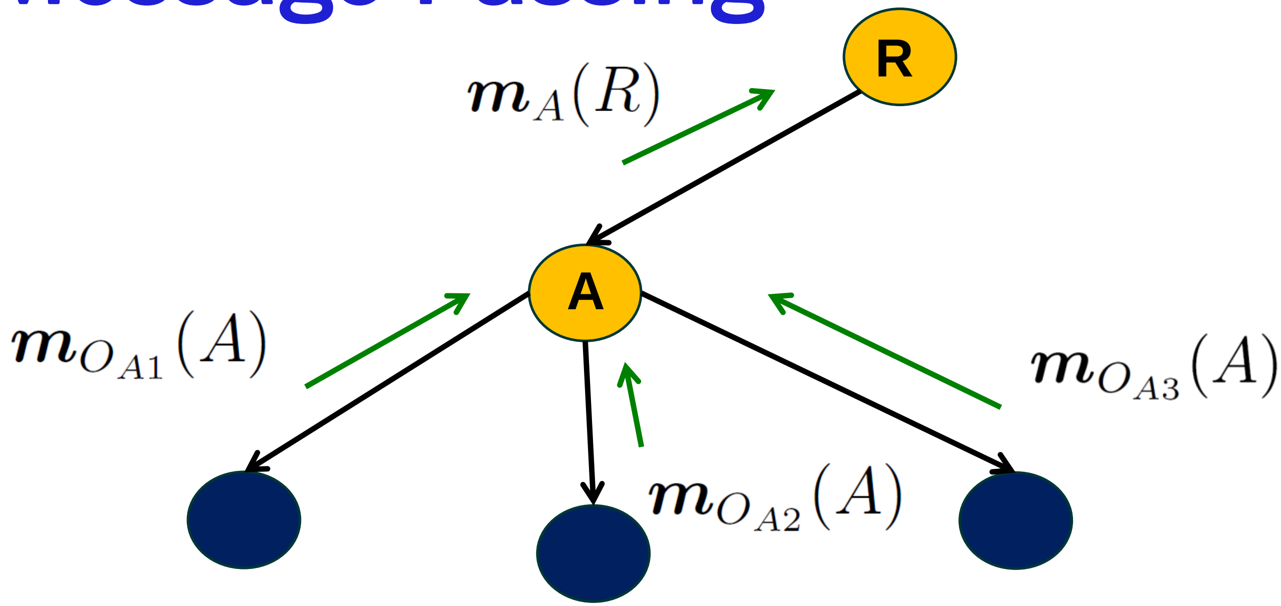
Compute this using message passing

Message Passing



$$\boldsymbol{m}_A(R) = \sum_A \mathbb{P}[A|R]\boldsymbol{m}_{O_{A1}}(A)\boldsymbol{m}_{O_{A2}}(A)\boldsymbol{m}_{O_{A3}}(A)$$

Message Passing



a vector

$$\boldsymbol{m}_{o_{A1}}(A) = \mathbb{P}[O_{A1} = o_{A1} | A] =$$

(remember we don't know how to explicitly compute this since it depends on A…)

Representing the Product

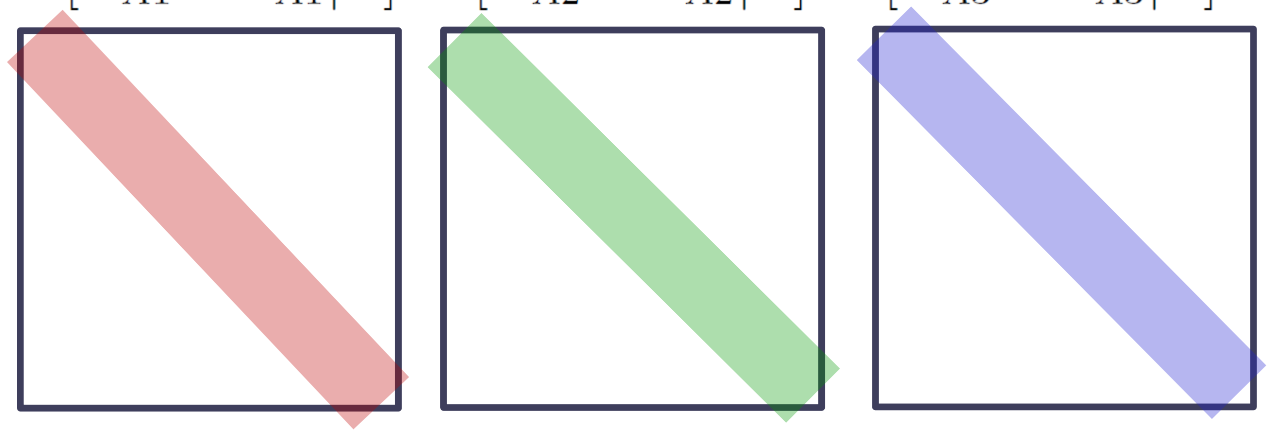
$$\boldsymbol{m}_{o_{A1},o_{A2},o_{A3}}(A)$$

$$\boldsymbol{m}_A(R) = \sum_A \mathbb{P}[A|R]\boldsymbol{m}_{O_{A1}}(A)\boldsymbol{m}_{O_{A2}}(A)\boldsymbol{m}_{O_{A3}}(A)$$

$$\boldsymbol{m}_{o_{A1},o_{A2},o_{A3}}(A) = \mathbb{P}[O_{A1} = o_{A1}|A] \cdot \mathbb{P}[O_{A2} = o_{A2}|A] \cdot \mathbb{P}[O_{A3} = o_{A3}|A]$$

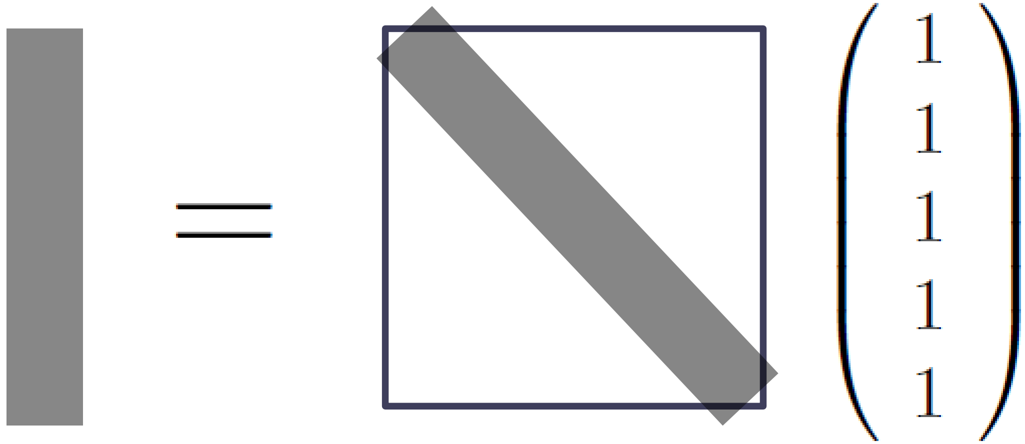
Instead let Message be Diagonal Matrix

$$\mathbb{P}[O_{A1} = o_{A1}|A] \cdot \mathbb{P}[O_{A2} = o_{A2}|A] \cdot \mathbb{P}[O_{A3} = o_{A3}|A]$$



$$\boldsymbol{M}_A(R) \quad = \quad \boldsymbol{M}_{A1}(A) \quad \boldsymbol{M}_{A2}(A) \quad \boldsymbol{M}_{A3}(A)$$

Equivalence with Original Message



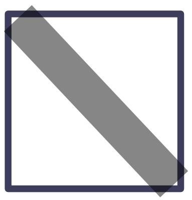
$$\boldsymbol{m}_A(R) = \boldsymbol{M}_{A1}(A)\mathbf{1}_A$$

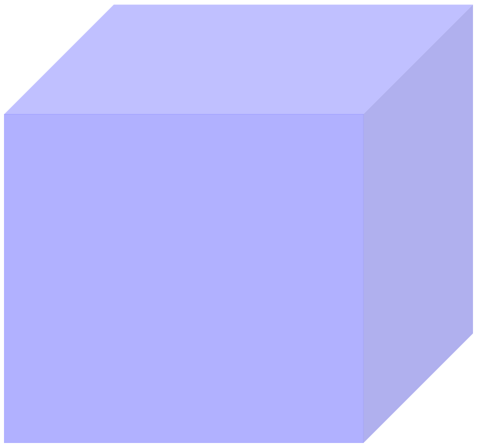
Representing the Sum

- Represent conditional probability table with a cube

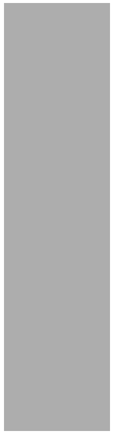
Representation of Pr[R | A] as a cube

$$M_A(R) =$$



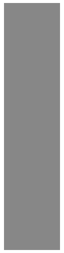


$$\bar{\times}_1$$



$$\mathcal{T}_{A|R}$$

$$M_{o_{A1}}(A) M_{o_{A2}}(A) M_{o_{A3}}(A) \mathbf{1}_A$$

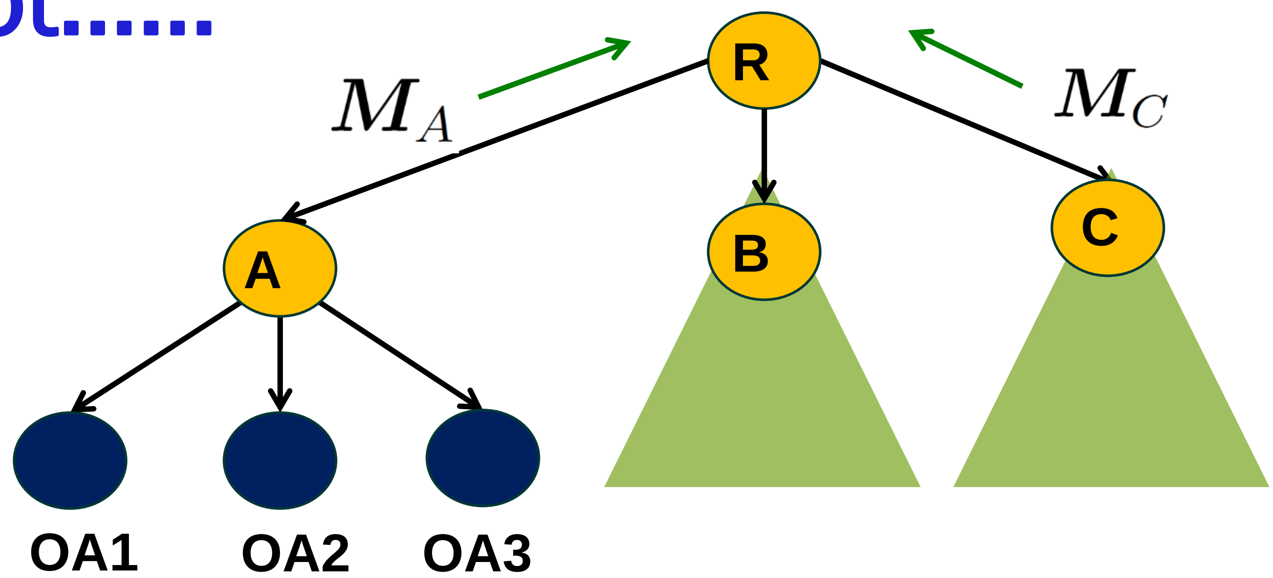


$$m_A(R) = \sum_A \mathbb{P}[A|R] m_{O_{A1}}(A) m_{O_{A2}}(A) m_{O_{A3}}(A)$$

SAILING LAB
Laboratory for Statistical Artificial InteLligence & INtegrative Genomics

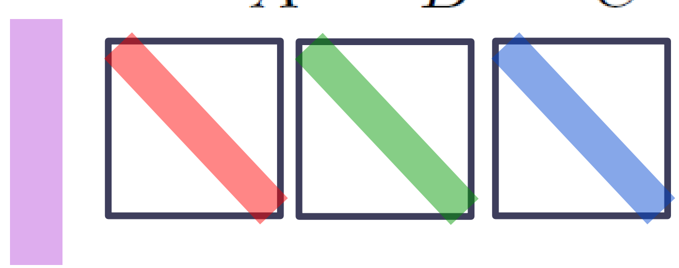
At the root……

$$\boldsymbol{r} = \mathbb{P}[X_r]$$



$$\mathbb{P}[O_{A1} = o_{A1}, O_{A2} = o_{A2}, O_{A3} = o_{A3}, \dots\dots\dots] =$$

$$\boldsymbol{r}^\top \boldsymbol{M}_A \boldsymbol{M}_B \boldsymbol{M}_C \boldsymbol{1}_r$$



Computation of Joint Probability

$$\mathbb{P}[O_{A1} = o_{A1}, O_{A2} = o_{A2}, O_{A3} = o_{A3}, \ldots\ldots\ldots] =$$

$$\boldsymbol{r}^\top \boldsymbol{M}_A \boldsymbol{M}_B \boldsymbol{M}_C \mathbf{1}_r$$

$$\boldsymbol{M}_A = \boldsymbol{\mathcal{T}}_{A|R} \bar{\times}_1 \boldsymbol{M}_{o_{A1}} \boldsymbol{M}_{o_{A2}} \boldsymbol{M}_{o_{A3}} \mathbf{1}_A$$

Sequence of matrix multiplications

Algorithm Overview

Latent Tree Representation

Compute joint probability as sequence of matrix multiplications

Transformed Representation

Insert transform matrices so that we can estimate transformed quantities instead of actual quantities

Transformed Representation

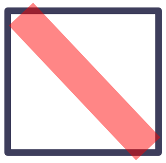
Transform matrices $RL^{-1} = I$

$$\mathbb{P}[O = o] = r^\top M_A M_B M_C 1_r = r^\top M_A R L^{-1} M_B M_C 1_r$$

$$M_A = \mathcal{T}_{A|R} \bar{\times}_1 L_{o_{A1}} L^{-1}_{o_{A1}} M_{o_{A1}} R_{o_{A1}} L^{-1}_{o_{A2}} M_{o_{A2}} R_{o_{A2}} L^{-1}_{o_{A3}} M_{o_{A3}} R_{o_{A3}} 1_A$$

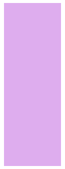
$$\tilde{M}_{o_{A1}}$$

Transformed Representation

Original Quantity: $M_{\mathcal{O}_{A1}}$ 

Transformed Quantity: $\tilde{M}_{\mathcal{O}_{A1}} = L_{\mathcal{O}_{A1}}^{-1} M_{\mathcal{O}_{A1}} R_{\mathcal{O}_{A1}}$

Estimate this instead!

Original Quantity: $r^\top$ 

Transformed Quantity: $\tilde{r}^\top = r^\top L_A$

Estimate this instead!

And similarly for the cube and one vector…..

Algorithm Outline

Latent Tree Representation

Compute joint probability as sequence of matrix multiplications

Transformed Representation

Insert transform matrices so that we can estimate transformed quantities instead of actual quantities

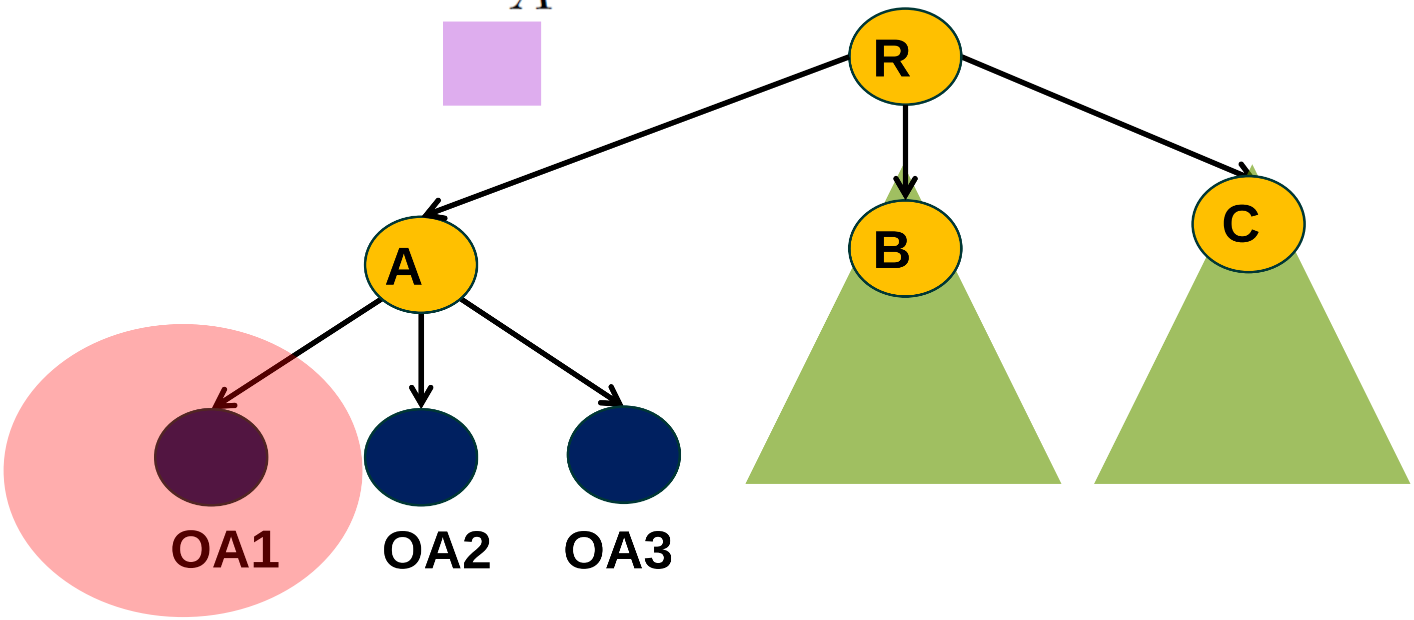
Observable Representation

Prove that these transformed quantities are functions of observed variables.

Observable Representation

Consider the root:

$$\tilde{\boldsymbol{r}}^{\top} = \boldsymbol{r}^{\top} \boldsymbol{L}_A$$



Consider the following choice for **L** :

$$\boldsymbol{Z}(k,l) = \mathbb{P}[O_{A1} = k | R = l]$$

Observable Representation

$$\boldsymbol{r}^\top = \mathbb{P}[R]^\top$$

$$\boldsymbol{r}^\top =$$

Not a function of observed variables

$$\mathbb{P}[O_{A1} = k | R = l]$$

$$\boldsymbol{r}^\top = \mathbb{P}[R]^\top$$

$$\mathbb{P}[O_{A1} = o | R]$$

$$\tilde{\boldsymbol{r}}^\top = \quad \boldsymbol{L} \quad = \quad \mathbb{P}[O_{A1}]^\top$$

function of observed variables

$$\tilde{\boldsymbol{r}}[o] = \sum_R \mathbb{P}[O_{A1} = o | R]\mathbb{P}[R] = \mathbb{P}[O_{A1} = o]$$

R integrated out by the matrix multiplication!

Observable Representation

- If **L = Z**^{T} then **L**$^{-1}$ does not exist since **Z** is not square!

$$\boldsymbol{Z}(k,l) = \mathbb{P}[O_{A1} = k | R = l]$$

- **Solution:** Project **Z** down to the subspace of hidden variables with a matrix **U**

$$\boldsymbol{L} = \boldsymbol{Z}^\top \boldsymbol{U} \qquad \boldsymbol{L}^{-1} = (\boldsymbol{Z}^\top \boldsymbol{U})^{-1}$$

$$\tilde{\boldsymbol{r}}^\top = \boldsymbol{r}^\top \boldsymbol{L}_A \quad \Longrightarrow \quad \boldsymbol{r}^\top = \mathbb{P}[O_{A1}]^\top \boldsymbol{U}_{O_{A1}}$$

Observable Representation (Message)

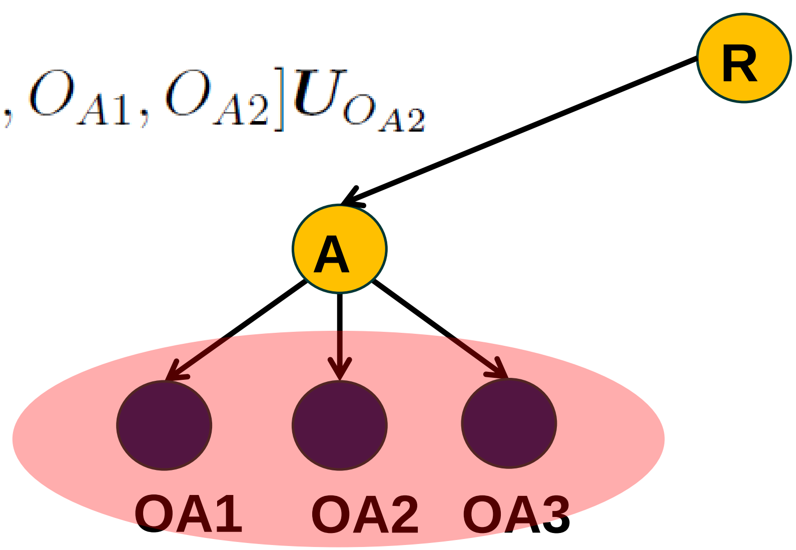
$$M_A = \square$$

Not a function of observed variables

$$\tilde{M}_{o_{A1}} = L^{-1}_{o_{A1}} \; M_A \; R_{o_{A1}}$$

function of observed variables

$$\begin{aligned}\tilde{M}_{o_{A1}} &= f(O_{A1}, O_{A2}, O_{A3}) \\ &= (\mathbb{P}[O_{A3}, O_{A1}] U_{O_{A1}})^{\dagger} \mathbb{P}[O_{A3}, O_{A1}, O_{A2}] U_{O_{A2}}\end{aligned}$$



Algorithm Overview

Latent Tree Representation

Compute joint probability as sequence of matrix multiplications

Transformed Representation

Add transform matrices to estimate transformed quantities instead of actual quantities

Observable Representation

Prove that these transformed quantities are functions of observed variables.

Sample Complexity

- When empirical estimate of transformed quantities equals true transformed quantities, joint probability estimate is equal to the true joint probability.

- Aggregate the errors across the quantities to get a bound.

With high probability,

$$\sum_{x_1,\ldots,x_O} \left| \widehat{\mathbb{P}}[x_1,\ldots,x_O] - \mathbb{P}[x_1,\ldots,x_O] \right| \leq O\left(\sqrt{\frac{(d_{max} S_H)^{2\ell+1} S_O}{N}} \right)$$

Max degree → d_{max}

Number of hidden states → S_H

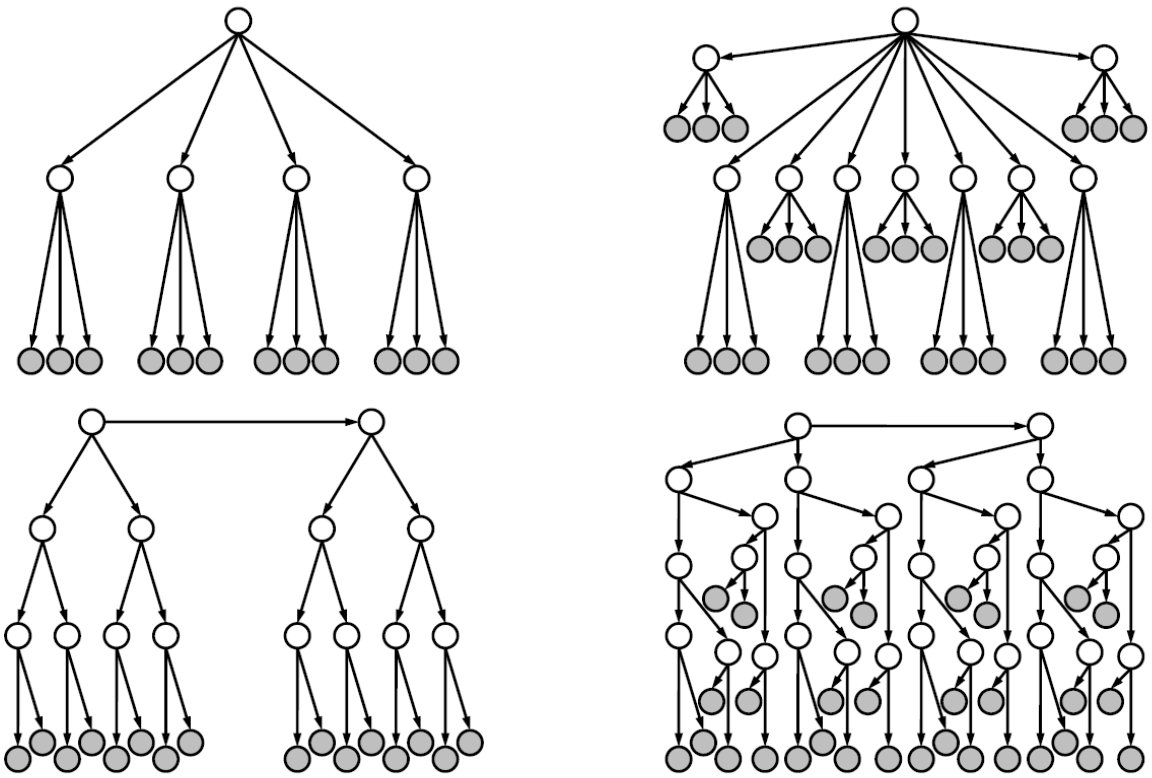
depth → ℓ

Number of samples → N

Number of observed states → S_O

Simulations

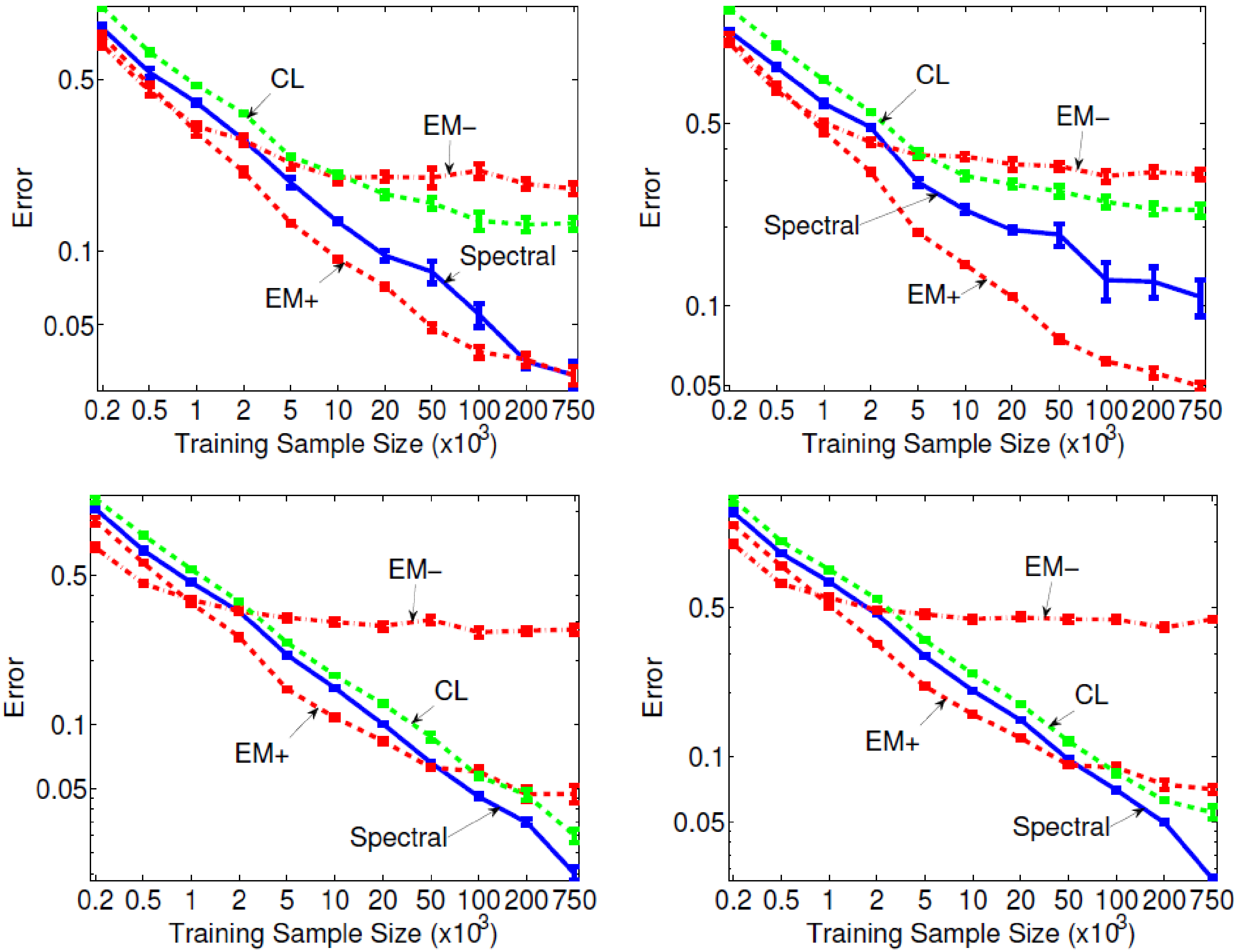
- **4 types of trees:**



Compare with:

- **EM (high precision), EM (low precision) on latent tree**
- **Chow liu tree on best fully observable tree – *more restricted model*).**

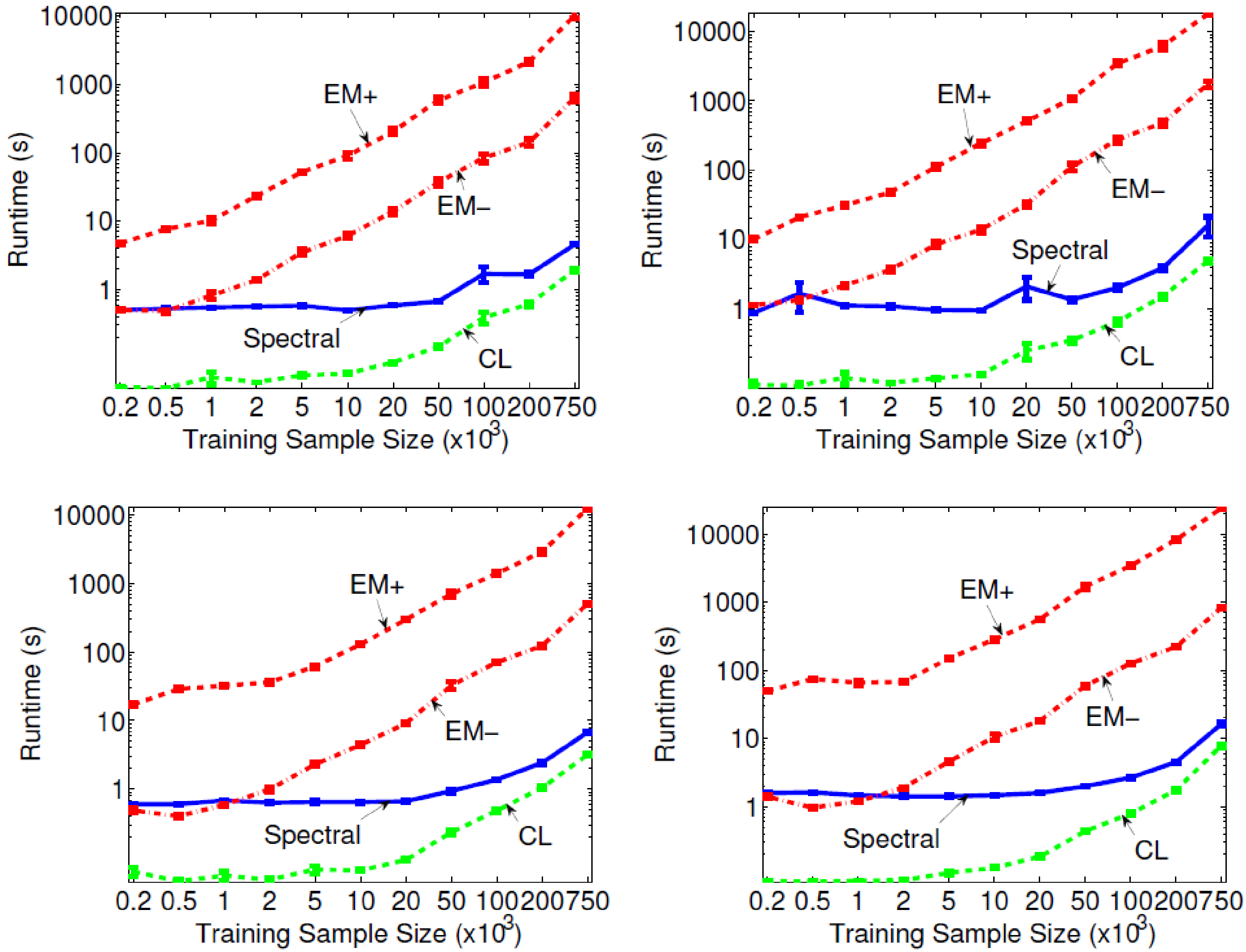
Simulations-Error



Spectral
EM
Chow Liu

Spectral
EM
Chow Liu

Simulations-Speed



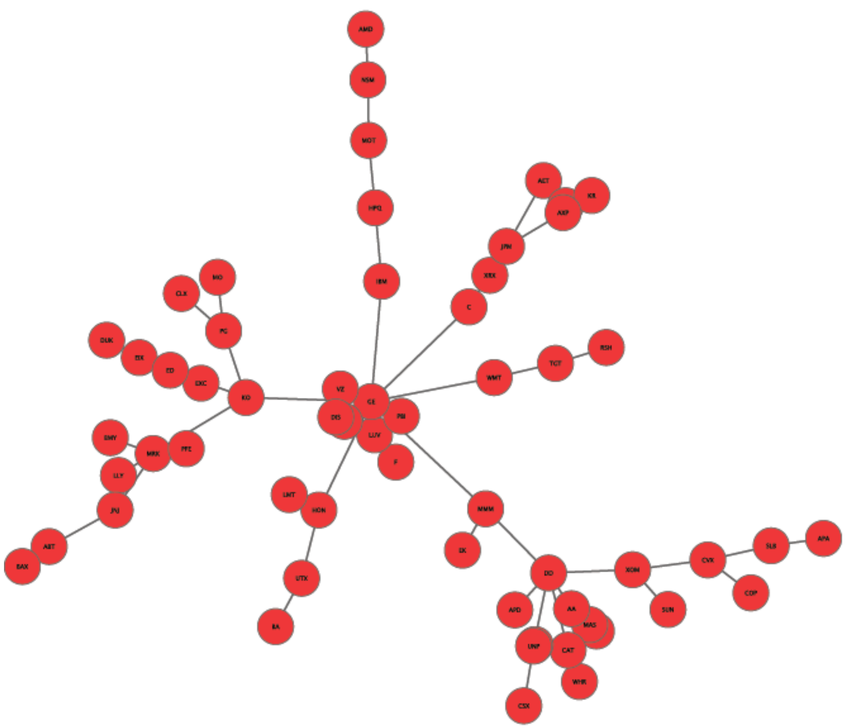
Spectral
EM
Chow Liu

Spectral
EM
Chow Liu

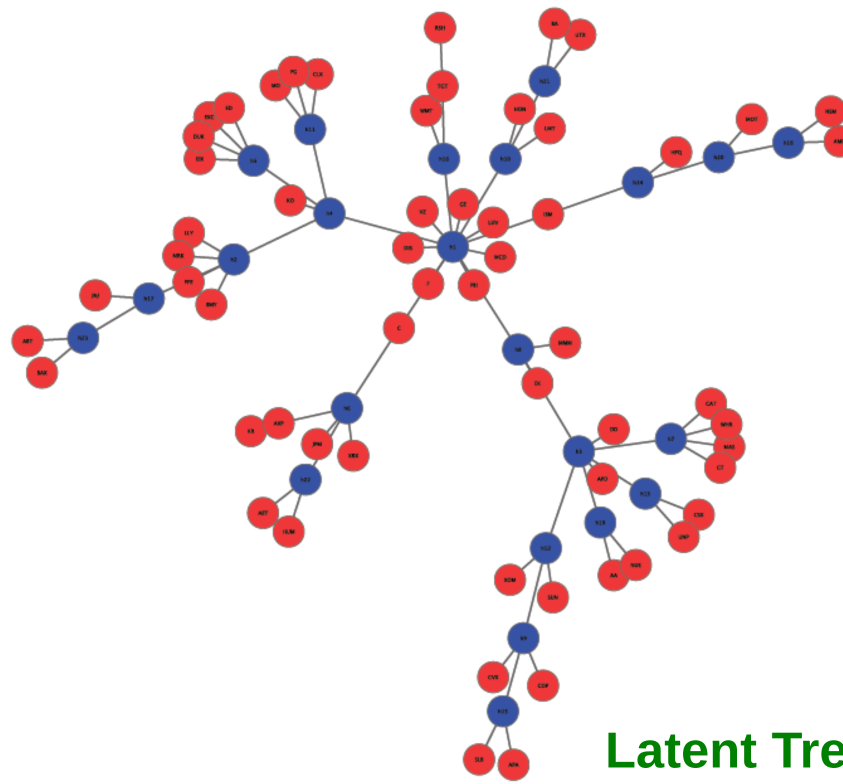
Stock Data Experiment

Acquired closing prices for 59 stocks from 1984 to 2011.Goal is to condition on a few stocks and see how well they predict another stock.

Latent tree structure learned using algorithm of Choi et al. 2010

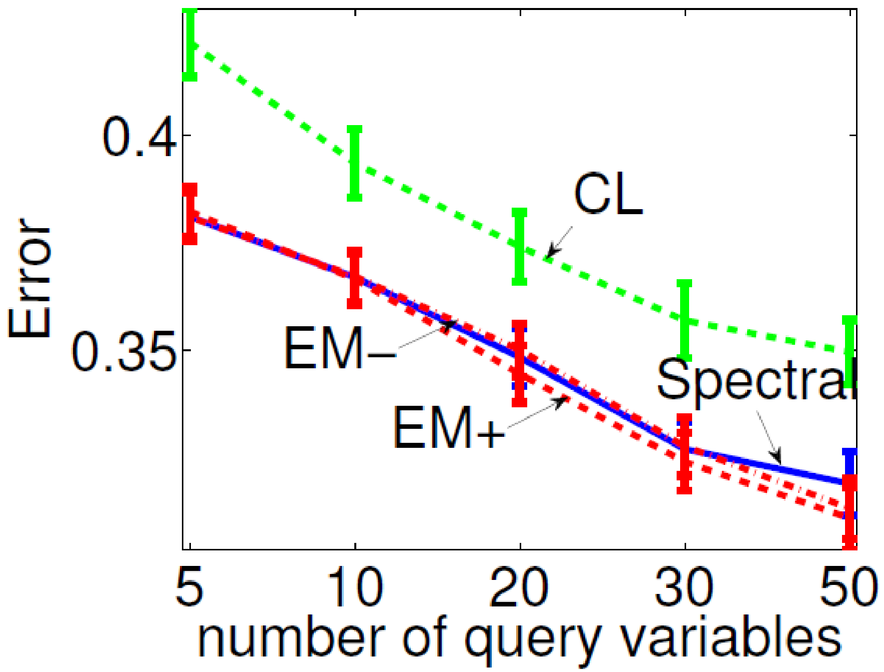


Chow Liu Tree



Latent Tree

Stock Data Results



Spectral
EM
Chow Liu

All the approaches that use the estimated latent tree perform better than message passing on the fully observable estimated Chow Liu tree.

Conclusion

- Latent trees are a **powerful** as well as **tractable** way to model relationships among variables
- Our spectral algorithm presents a fast, consistent, and local-minima-free approach for parameter learning/inference in latent trees.
- Future directions include spectral algorithms for loopy graphs and kernelized spectral algorithms.