Reconstruction of Tree and Latent Tree Models: Consistency and Error Rates

Anima Anandkumar

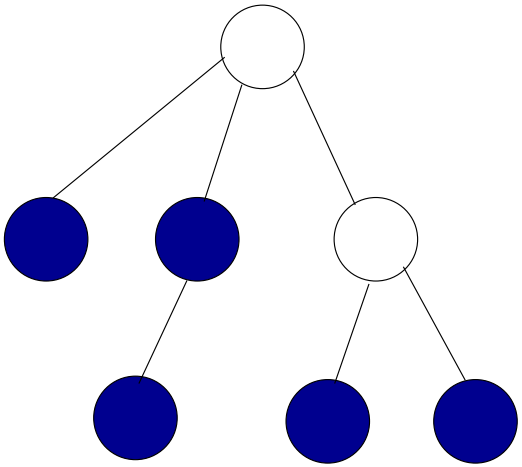
Electrical Engineering and Computer Science
MIT, Cambridge, MA 02139

Joint work with Myung Jin Choi, Vincent Tan, Lang Tong and Alan Willsky.

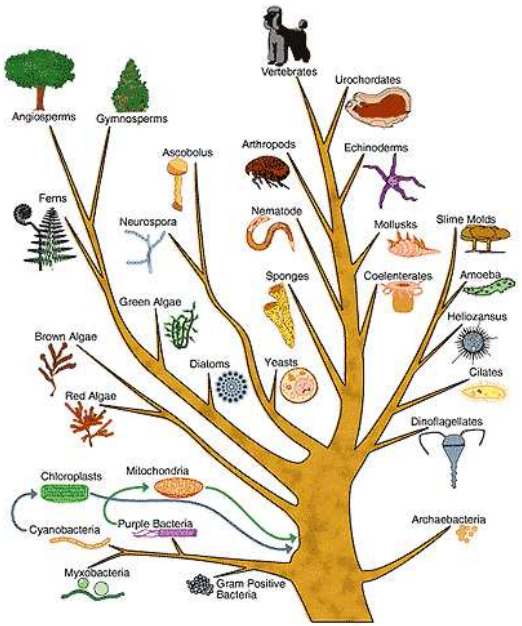
Yale University

Tree and Latent Tree Models

Latent Tree Model



Phylogenetic Tree*

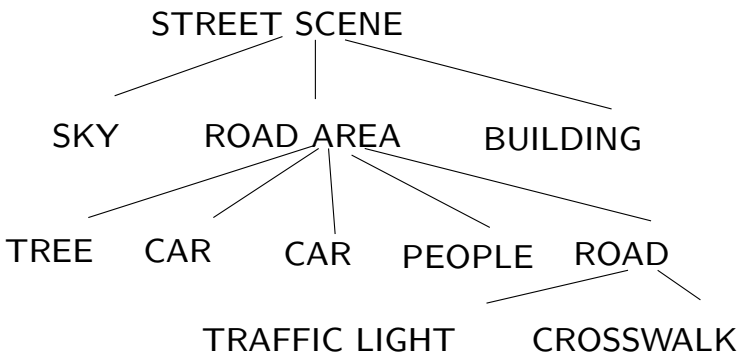


Object Recognition

*

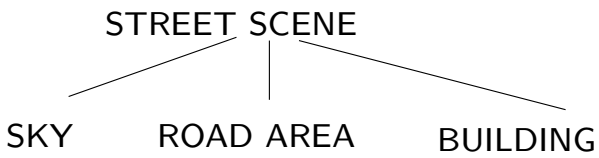
CREDIT, phylogenetic tree: CSS, Inc.

Hierarchical Context in Object Recognition



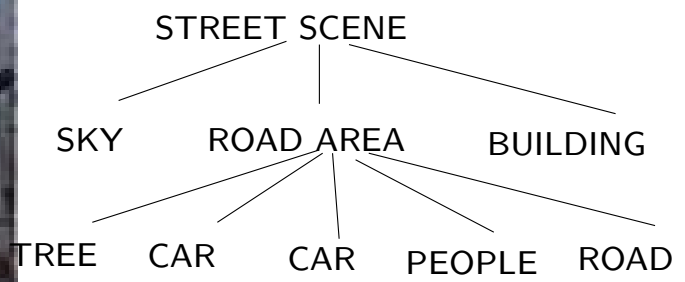
M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting Hierarchical Context on a Large Database of Object Categories", to appear at 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Hierarchical Context in Object Recognition



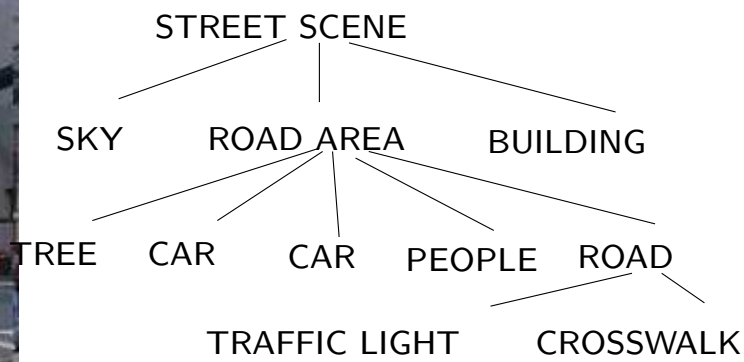
M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting Hierarchical Context on a Large Database of Object Categories", to appear at 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Hierarchical Context in Object Recognition



M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting Hierarchical Context on a Large Database of Object Categories", to appear at 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

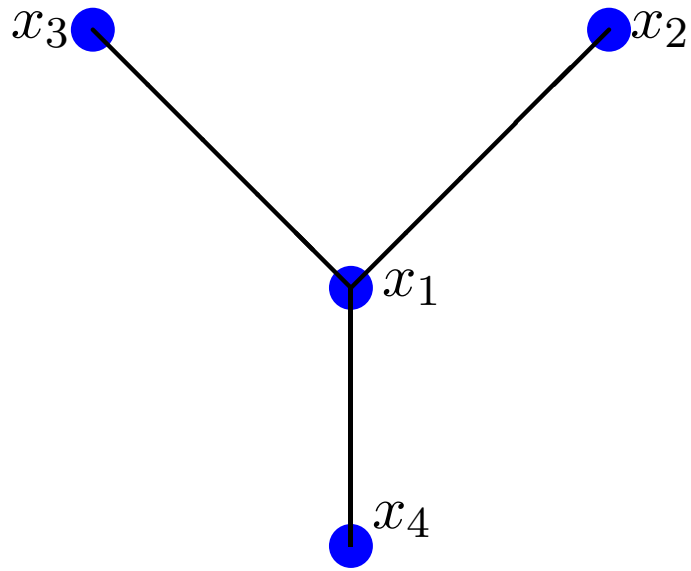
Hierarchical Context in Object Recognition



M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting Hierarchical Context on a Large Database of Object Categories", to appear at 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

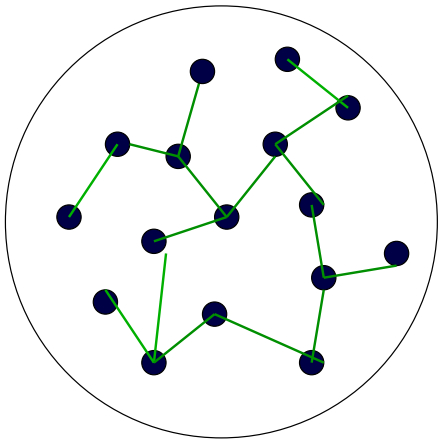
Tree Distributions

- A tree is an undirected acyclic graph.



- Inference via Belief Propagation is exact on trees.
- Maximum-likelihoood learning can be implemented in polynomial time via the Chow-Liu (1968) algorithm.
- But maximum-likelihood learning of latent trees is NP-hard (Roch 06)

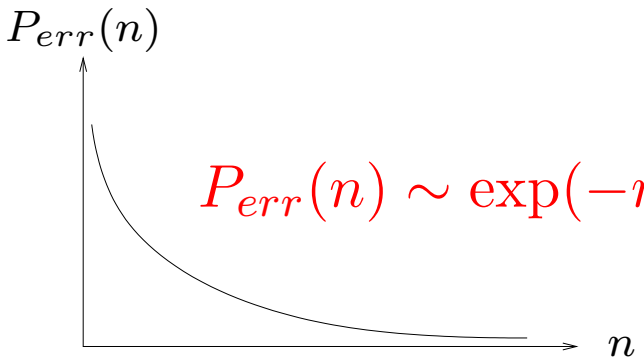
Error Exponent for ML-Learning of Trees



Dependency tree

Structure Learning

Given IID samples from MRF p, maximum-likelihood learning of dependency graph.



$$P_{err}(n) \sim \exp(-nK_p)$$

Error Exponent K_p

Rate of exponential decay of prob. that estimated tree $\neq$ true tree.

Influence of Graph Structure on Learning Error Exponent

Results on Error Exponents for Tree Learning

- Almost every (true) tree distribution has exponential decay.
- Provide the exact rate of decay for a given P.
- Rate of decay $\approx$ SNR for learning.
- Provide intuition as to how errors occur.

---

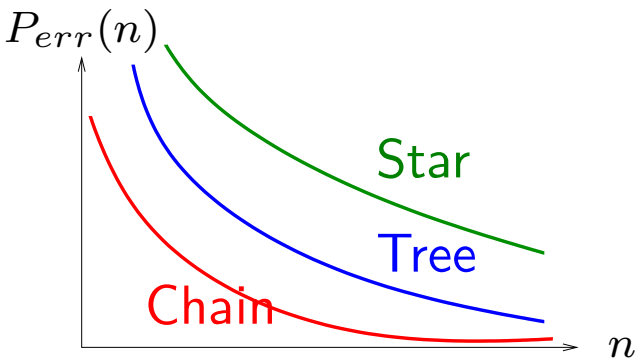
1. V. Tan, A. Anandkumar, A. Willsky " Learning Gaussian Tree Models: Analysis of Error Exponents and Extremal Structures", *IEEE Tran. on Signal Proc.*, Vol. 58, No. 5, May 2010, pp. 2701-2714.
2. V. Tan, A. Anandkumar, L. Tong, A. Willsky "A Large-Deviation Analysis of the Maximum-Likelihood Learning of Markov Tree Structures," *submitted to IEEE Tran. on Information Theory*, on Arxiv.

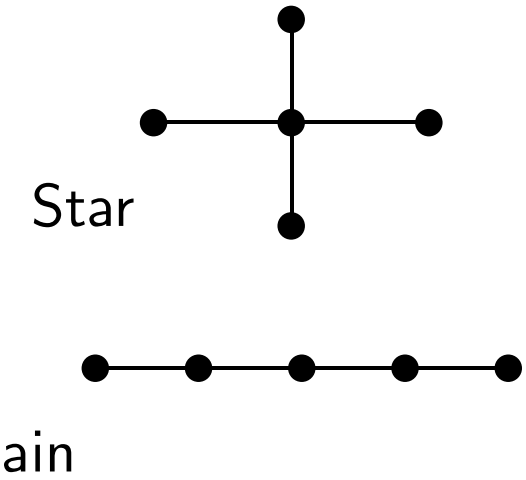
Summary of Results Contd.,

Extremal Tree Structures for Learning

For Gaussian distribution in very noisy learning regime

- Star graphs are hardest to learn, Markov chains are easiest to learn.
- Error exponent increases with tree diameter.
- Keeping the correlations on edges fixed.

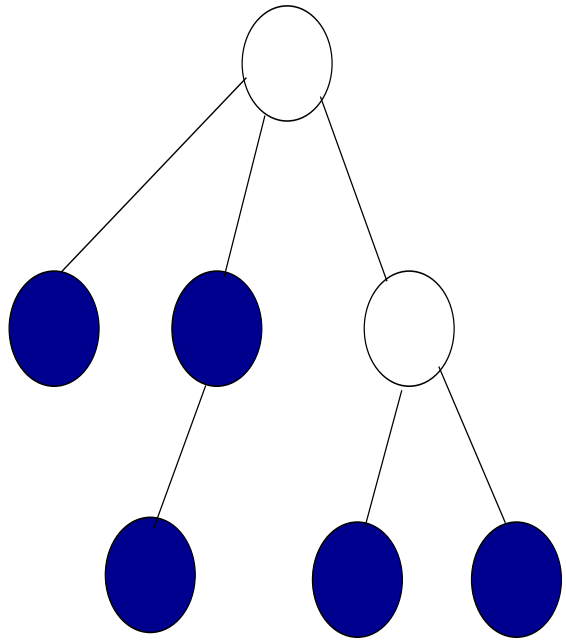




Error exponent related to correlation decay

Latent Tree Model

- Visible Nodes V, Hidden Nodes H and $W := V \cup H$
- $T = (W, E)$ is a tree on W



Latent Tree Reconstruction

Given n IID samples from node set V, reconstruct latent tree model

Results on Learning Latent Trees

Reconstruction of general latent tree models from samples

- Propose two novel algorithms under unified approach for Gaussian and discrete models
- Provide theoretical guarantees: consistency, computational and sample complexities
  - Structural and estimation consistency for any minimal latent tree
  - Sample complexity of $O(\log m)$ for m observed nodes when effective depth is constant
  - Low computational complexity
- Experimental results demonstrate efficiency of methods

M.J. Choi, V. Tan, A. Anandkumar & A. Willsky, "Consistent and Efficient Reconstruction of Latent Tree Models," Preprint.

Large Deviations for Learning Trees: Related Work

Learning Tree Distributions

- Efficient implementation of ML (Chow & Liu 68)
- High Dimensional Forests (Tan, Anandkumar & Willsky 10)

Learning Sparse Distributions

- ℓ_1 regularization (Dudik 2004, Wainwright 2006).
- Gaussian Graphical Models (Meinshausen and Buehlmann 2006).
- Logistic regression for Ising models (Ramkumar et. al. 10)

Error rates for learning

- Many algorithms have efficient sample complexities
- Bounds on error rate for learning Bayesian networks. (Zuk et al. 06)
- Euclidean information theory (Borade, Abbe, Zheng 2006-8) is used here to gain deeper insight.

Related Work on Latent Trees

Expectation Maximization

Greedy local structural search (Kemp & Tenenbaum 08, Zhang & Kocka 04, Elidan & Friedman 05)

Phylogenetic Tree Reconstruction

Focus on 3-complete trees with only leaves observed. (Erdos et. al 99, Attenson 99, Daskalakis et al. 06)

Network Tomography

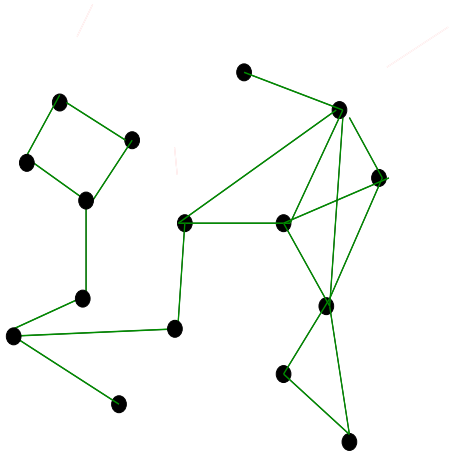
End-to-end delay measurements (Ni, Tatikonda 08, Bhamidi et.al. 08)

Outline

Definition of Graphical Model

- Parsimonious representation of a distribution $P(\mathbf{x})$ via a graph.
- Consider an undirected graph $G = (V, E)$, each vertex $i \in V = \{1, \ldots, d\}$ is associated with a random variable x_i.

Conditional-Independence

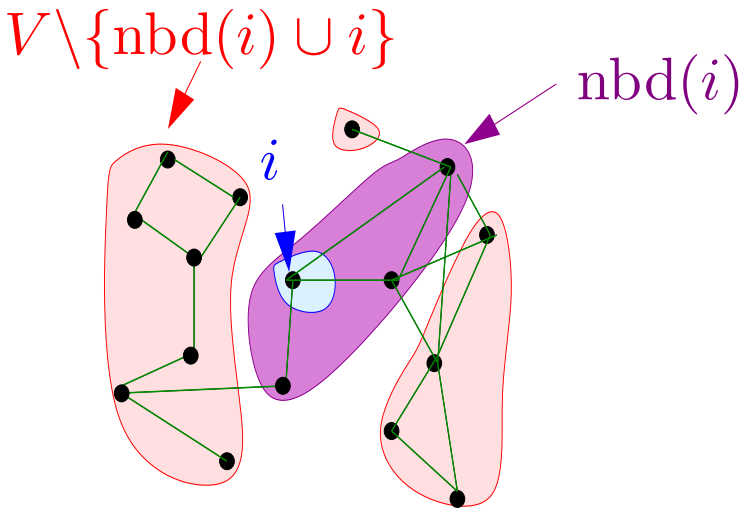


$$\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_{V \setminus \{\mathrm{nbd}(i) \cup i\}} \,|\, \mathbf{x}_{\mathrm{nbd}(i)}$$

Definition of Graphical Model

- Parsimonious representation of a distribution $P(\mathbf{x})$ via a graph.
- Consider an undirected graph $G = (V, E)$, each vertex $i \in V = \{1, \ldots, d\}$ is associated with a random variable x_i.

Conditional-Independence

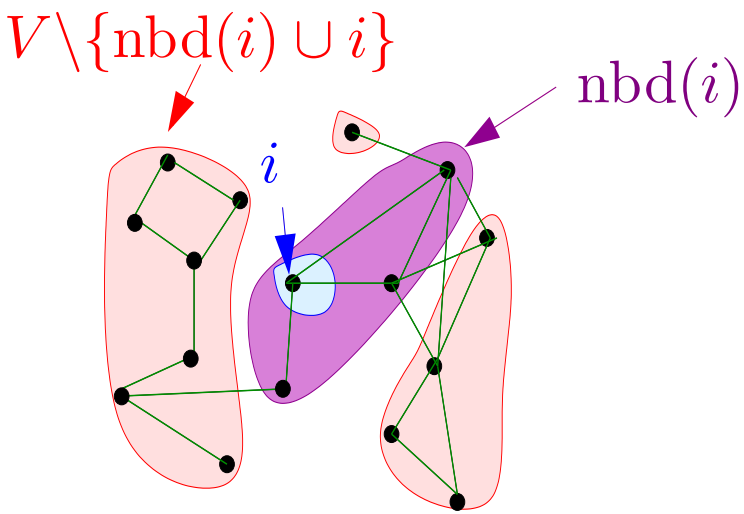


$$\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_{V \backslash \{\mathrm{nbd}(i) \cup i\}} \,|\, \mathbf{x}_{\mathrm{nbd}(i)}$$

Definition of Graphical Model

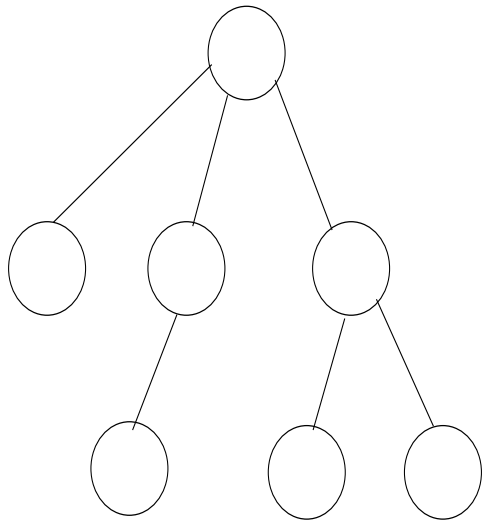
- Parsimonious representation of a distribution $P(\mathbf{x})$ via a graph.
- Consider an undirected graph $G = (V, E)$, each vertex $i \in V = \{1, \ldots, d\}$ is associated with a random variable x_i.

Conditional-Independence



$$\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_{V \backslash \{\text{nbd}(i) \cup i\}} \big| \mathbf{x}_{\text{nbd}(i)}$$

Tree Models



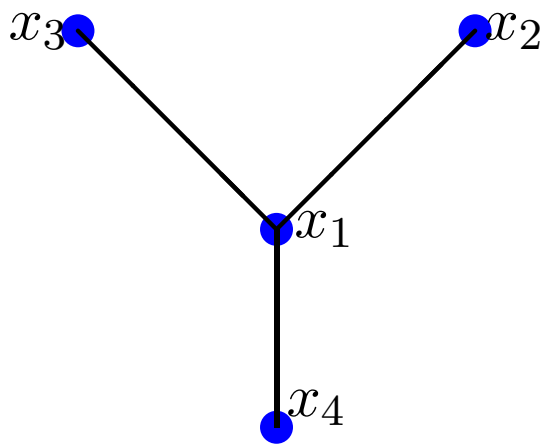
$$p(\mathbf{x}_V) = \prod_{i \in V} p_i(x_i) \prod_{(i,j) \in E} \frac{p_{i,j}(x_i, x_j)}{p_i(x_i) p_j(x_j)}$$

Notation and Background

Assume we have a set of samples $\mathbf{x}^n = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ each drawn i.i.d. from $P \in \mathcal{P}(\mathcal{X}^d)$, where $\mathcal{X}$ is a finite set.

Vector $\mathbf{x} = (x_1, \ldots, x_d) \in \mathcal{X}^d$. Define $V := \{1, \ldots, d\}$ and $E_P \subset \binom{V}{2}$.

- $P(\mathbf{x})$ is Markov on $T_P = (V, E_P)$, a tree.
- $P(\mathbf{x})$ factorizes according to T_P.
- Example for P with $d = 4$.

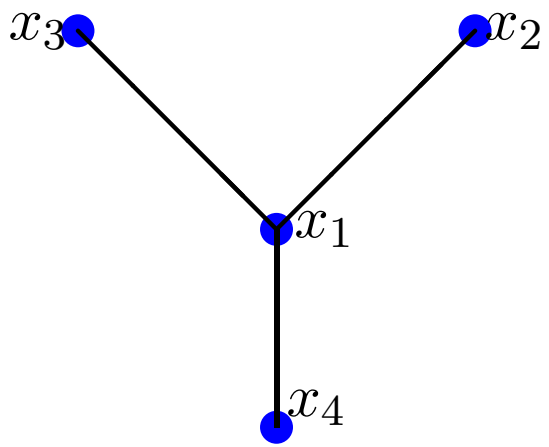


Notation and Background

Assume we have a set of samples $\mathbf{x}^n = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ each drawn i.i.d. from $P \in \mathcal{P}(\mathcal{X}^d)$, where $\mathcal{X}$ is a finite set.

Vector $\mathbf{x} = (x_1, \ldots, x_d) \in \mathcal{X}^d$. Define $V := \{1, \ldots, d\}$ and $E_P \subset \binom{V}{2}$.

- $P(\mathbf{x})$ is Markov on $T_P = (V, E_P)$, a tree.
- $P(\mathbf{x})$ factorizes according to T_P.
- Example for P with $d = 4$.



$$P(\mathbf{x}) \;=\; P_1(x_1) \times \frac{P_{1,2}(x_1, x_2)}{P_1(x_1)} \times \frac{P_{1,3}(x_1, x_3)}{P_1(x_1)} \times \frac{P_{1,4}(x_1, x_4)}{P_1(x_1)}.$$

ML Learning of Tree Distribution (Chow-Liu) I

- Solve the ML problem given $\mathbf{x}^n = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

$$P_{\text{ML}} = \underset{Q \in \text{Trees}}{\text{argmax}} \log Q^n(\mathbf{x}^n)$$

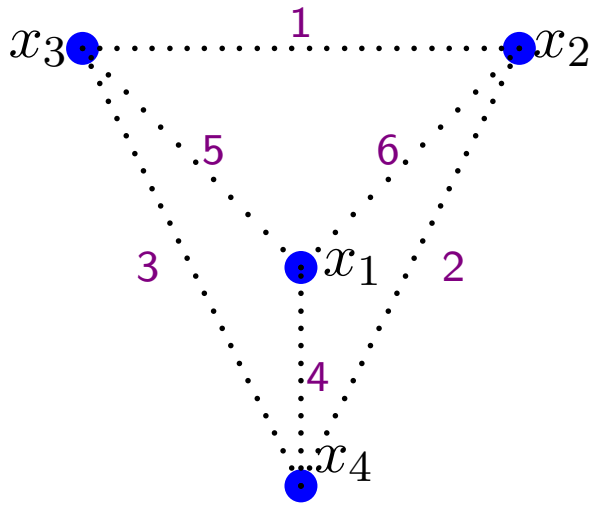
- Denote $\widehat{P} = \widehat{P}_{\mathbf{x}^n}$ as the empirical distribution of $\mathbf{x}^n$.
- Reduces to a max-weight spanning tree problem (Chow-Liu 1968)

$$E_{\text{ML}} = \underset{E_Q : Q \in \text{Trees}}{\text{argmax}} \sum_{e \in E_Q} I(\widehat{P}_e),$$

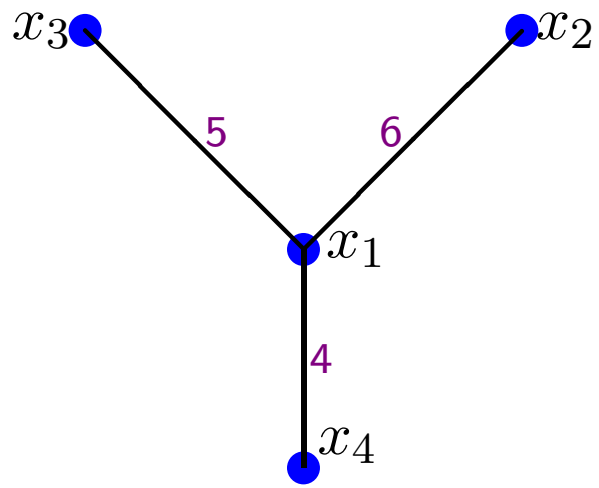
$$I(\widehat{P}_e) := \mathbb{E}_{\widehat{P}_{i,j}} \left[\log \frac{\widehat{P}_{i,j}(x_i, x_j)}{\widehat{P}_i(x_i)\widehat{P}_j(x_j)} \right],$$

where $\widehat{P}_e$ is the pairwise marginal of $\widehat{P}$ on edge $e = (i, j)$.

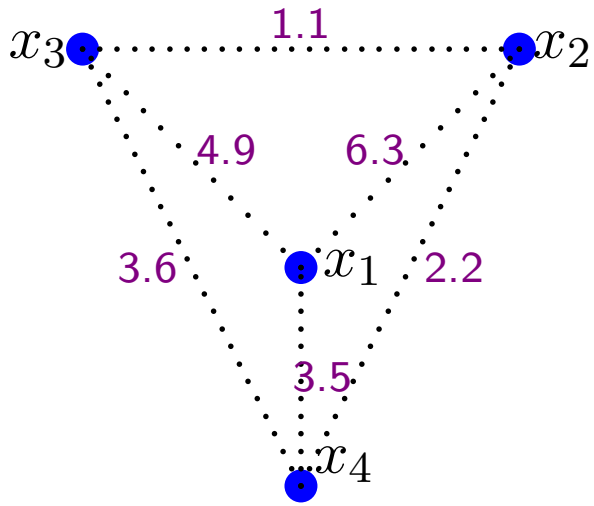
ML Learning of Tree Distribution (Chow-Liu) II



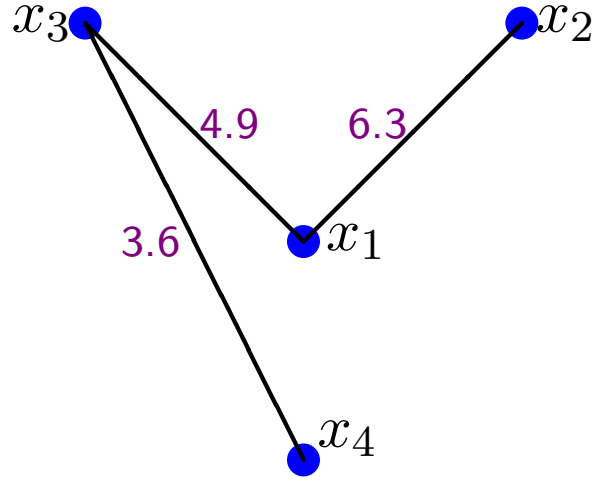
True MIs $\{I(P_e)\}$



Max-weight spanning tree E_P



Empirical MIs $\{I(\widehat{P}_e)\}$ from $\mathbf{x}^n$



Max-weight spanning tree $E_{\text{ML}} \neq E_P$

Outline

Problem Statement

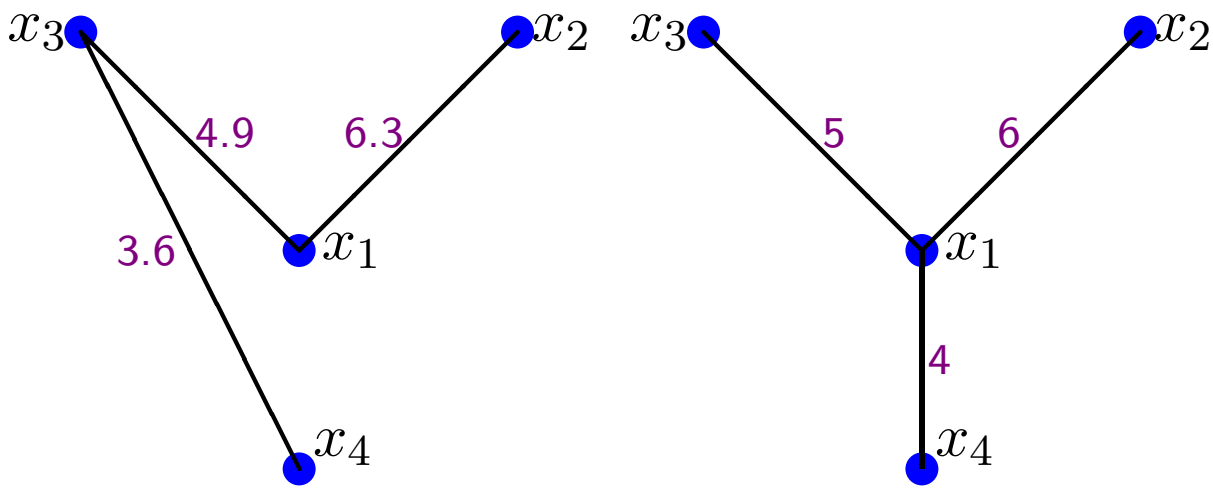
- Define P_{ML} to be ML tree structured distribution with edge set E_{ML} and the error event is

$$\{E_{\text{ML}} \neq E_P\}.$$

Problem Statement

- Define P_{ML} to be ML tree structured distribution with edge set E_{ML} and the error event is

$$\{E_{\text{ML}} \neq E_P\}\,.$$

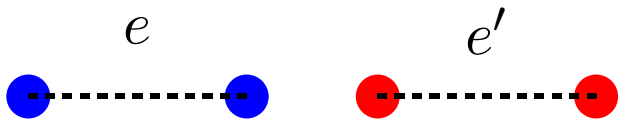


- Find the error exponent K_P:

$$K_P := \lim_{n\to\infty} -\frac{1}{n}\log \mathbb{P}\left(\{E_{\text{ML}} \neq E_P\}\right), \quad \mathbb{P}\left(\{E_{\text{ML}} \neq E_P\}\right) \doteq \exp(-nK_P)$$

- Naïvely, what could we do to compute the error exponent K_P?
- Easier to consider crossover events first.

The Crossover Rate I



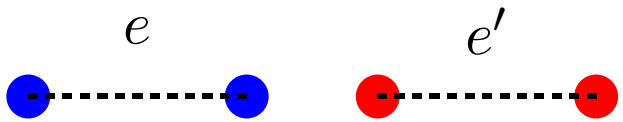
Given two node pairs $e, e' \in \binom{V}{2}$ with distribution $P_{e,e'} \in \mathcal{P}(\mathcal{X}^4)$, s.t.

$$I(P_e) > I(P_{e'}).$$

Consider the crossover event of the empirical mutual informations

$$\{I(\widehat{P}_e) \leq I(\widehat{P}_{e'})\}.$$

The Crossover Rate I



Given two node pairs $e, e' \in \binom{V}{2}$ with distribution $P_{e,e'} \in \mathcal{P}(\mathcal{X}^4)$, s.t.

$$I(P_e) > I(P_{e'}).$$

Consider the crossover event of the empirical mutual informations

$$\{I(\widehat{P}_e) \leq I(\widehat{P}_{e'})\}.$$

Def: Crossover Rate

$$J_{e,e'} := \lim_{n\to\infty} -\frac{1}{n} \log \mathbb{P}\left(\left\{I(\widehat{P}_e) \leq I(\widehat{P}_{e'})\right\}\right).$$

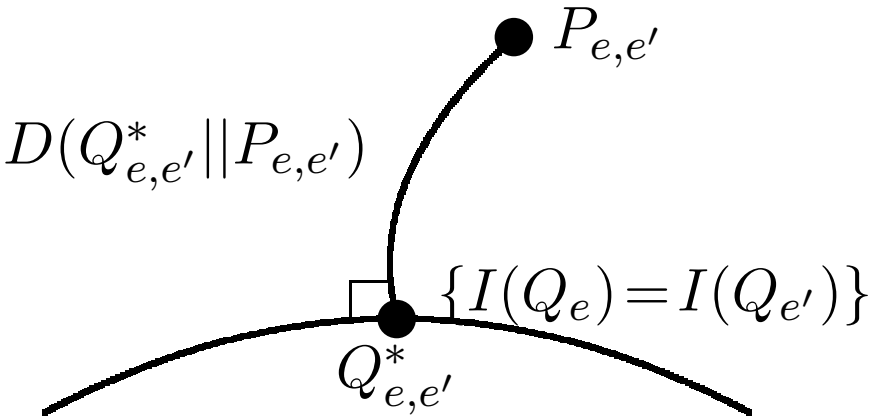
This event may potentially lead to an error in structure learning. Why?

The Crossover Rate II

Theorem

The crossover rate for empirical mutual informations is

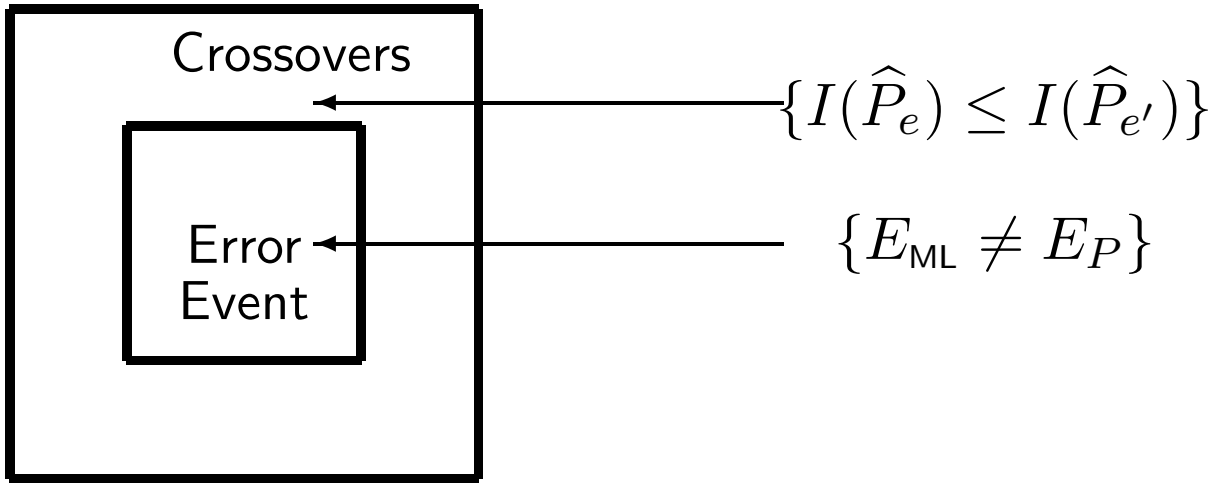
$$J_{e,e'} = \min_{Q \in \mathcal{P}(\mathcal{X}^4)} \left\{ D(Q \,||\, P_{e,e'}) : I(Q_{e'}) = I(Q_e) \right\}.$$



- Sanov's Theorem and The Contraction Principle.
- Exact but not very intuitive.
- Non-Convex.

Error Exponent for Structure Learning I

- We have characterized the rate for the crossover event $\left\{I(\widehat{P}_e) \leq I(\widehat{P}_{e'})\right\}$. We called the rate $J_{e,e'}$.



- How to relate this to K_P, the overall error exponent?

$$\mathbb{P}\left(\{E_{\mathsf{ML}} \neq E_P\}\right) \doteq \exp(-nK_P)$$

Error Exponent for Structure Learning II

Theorem

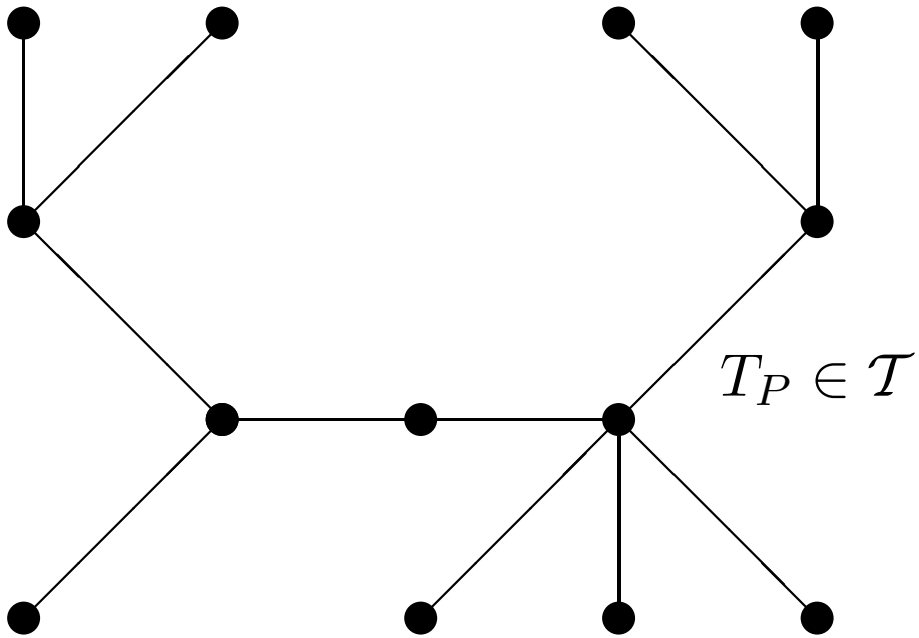
$$K_P = \min_{e' \notin E_P} \left(\min_{e \in \mathrm{Path}(e'; E_P)} J_{e,e'} \right).$$

Error Exponent for Structure Learning II

Theorem

$$K_P = \min_{e' \notin E_P} \left(\min_{e \in \mathrm{Path}(e'; E_P)} J_{e,e'} \right).$$

Identify the tree that dominates the error event.

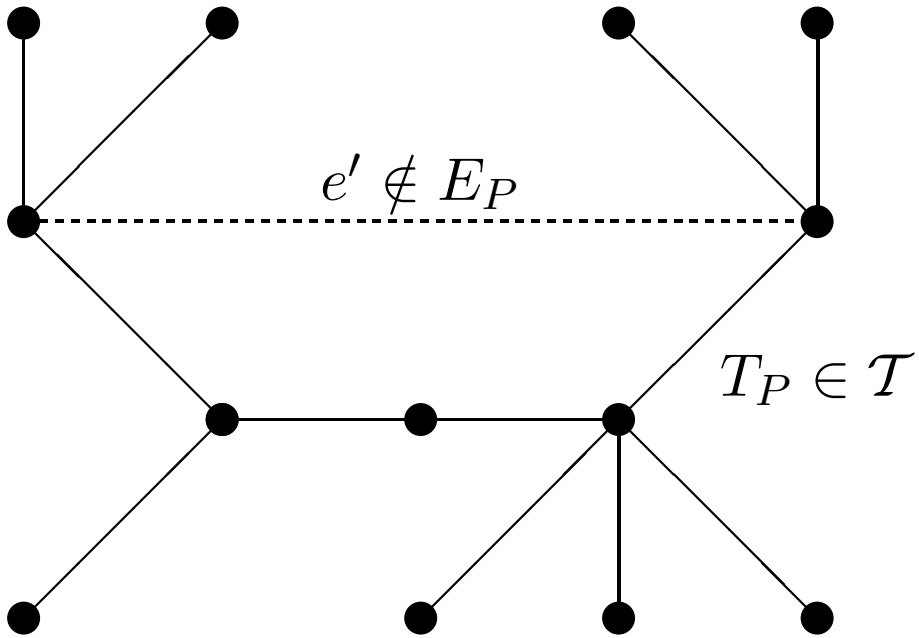


Error Exponent for Structure Learning II

Theorem

$$K_P \;=\; \min_{e' \notin E_P} \left(\min_{e \in \mathrm{Path}(e'; E_P)} J_{e,e'} \right).$$

Identify the tree that dominates the error event.

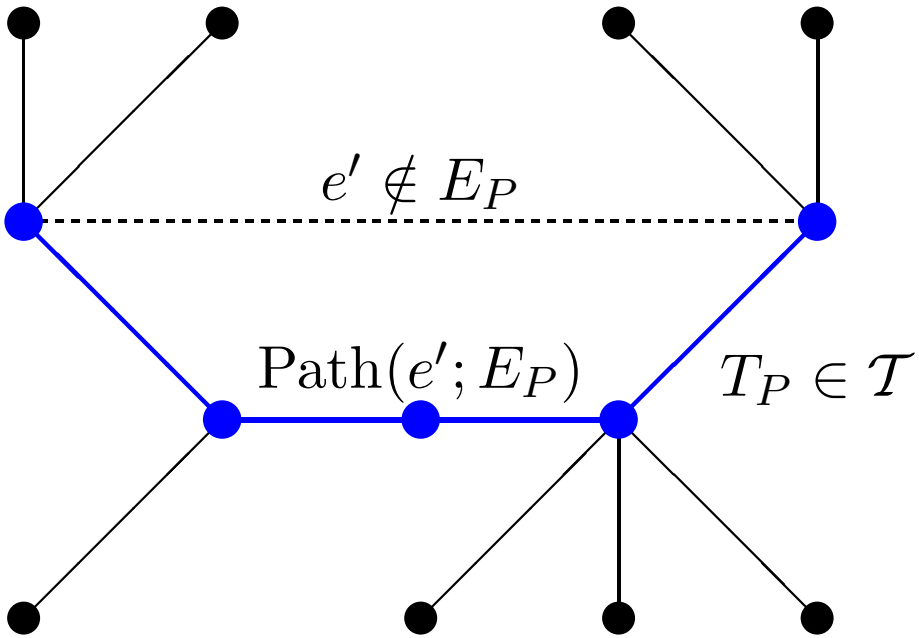


Error Exponent for Structure Learning II

Theorem

$$K_P \;=\; \min_{e' \notin E_P} \left(\min_{e \in \mathrm{Path}(e'; E_P)} J_{e,e'} \right).$$

Identify the tree that dominates the error event.

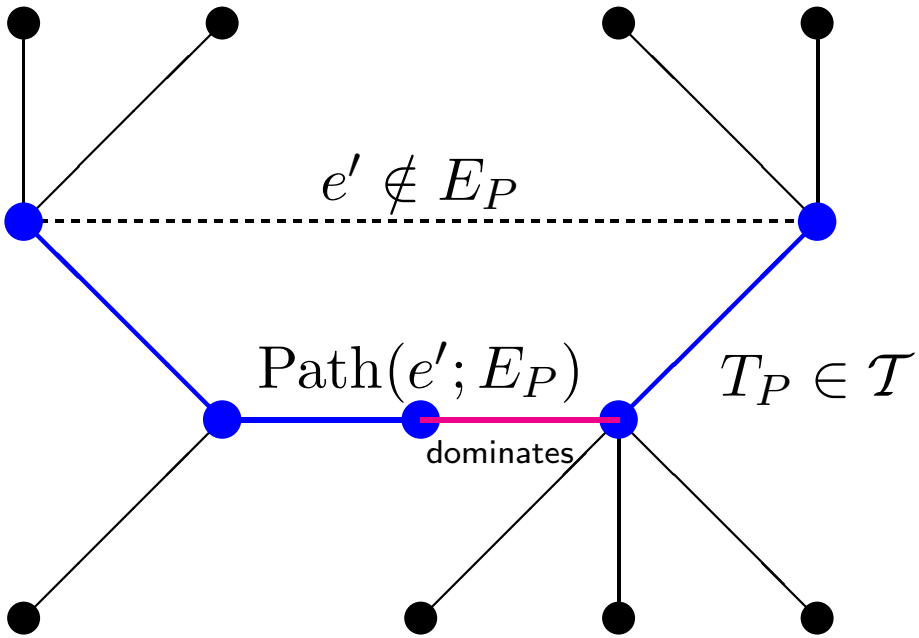


Error Exponent for Structure Learning II

Theorem

$$K_P \;=\; \min_{e' \notin E_P} \left(\min_{e \in \mathrm{Path}(e'; E_P)} J_{e,e'} \right).$$

Identify the tree that dominates the error event.

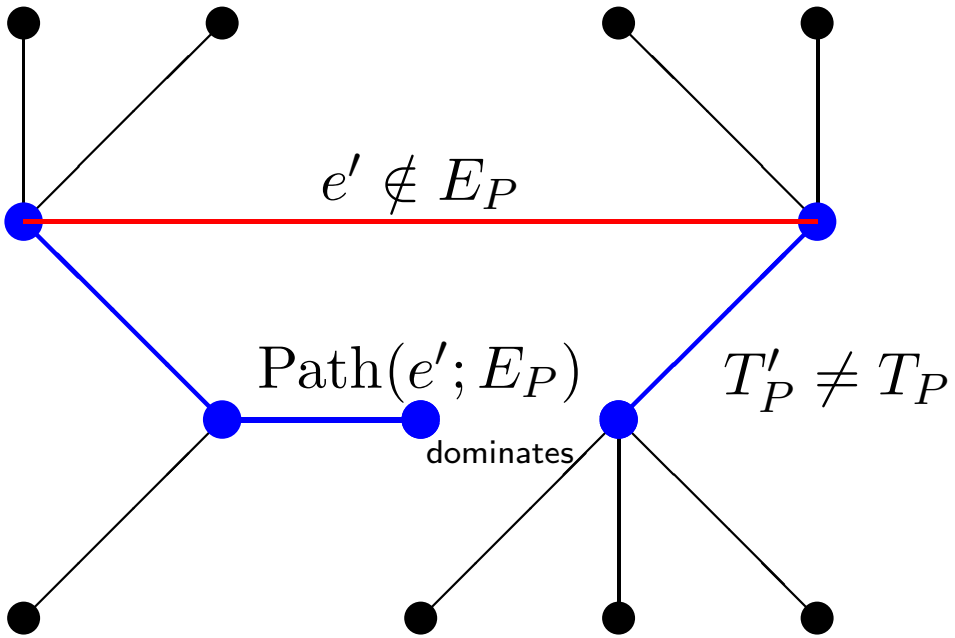


Error Exponent for Structure Learning II

Theorem

$$K_P = \min_{e' \notin E_P} \left(\min_{e \in \mathrm{Path}(e'; E_P)} J_{e,e'} \right).$$

Identify the tree that dominates the error event.



Error Exponent for Structure Learning III

$$\mathbb{P}(\{E_{\mathsf{ML}} \neq E_P\}) \doteq \exp\left[-n \min_{e' \notin E_P} \left(\min_{e \in \mathrm{Path}(e';E_P)} J_{e,e'} \right)\right].$$

- Proof Idea:
  - Dominant error tree only differs from true tree by 1 edge.
  - Union bound.
  - "Worst-exponent-wins" principle gives upper bound.

Proposition

The number of computations to compute K_P is upper bounded by

$$\frac{1}{2}\mathrm{diam}(T_P)(d-1)(d-2).$$

Compare to a brute force search $(d^{d-2}-1)$.

Positivity of the Error Exponent

Theorem

The following statements are equivalent:

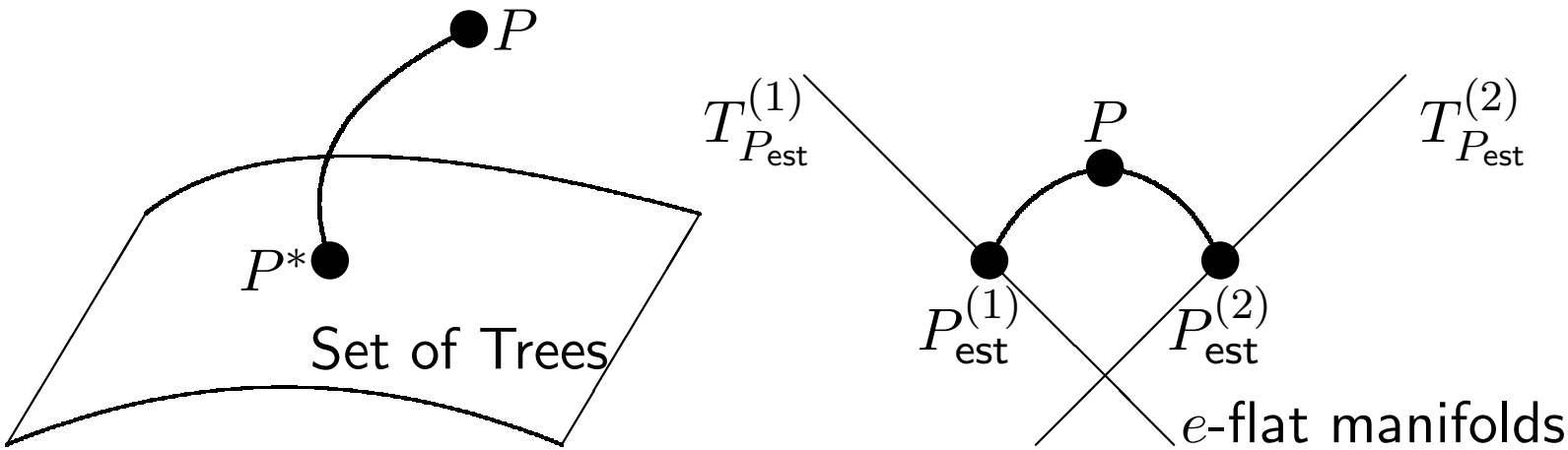
(a) *The error probability decays exponentially i.e.,*

$$K_P > 0.$$

(b) T_P *is a spanning tree, i.e., not a proper forest.*

Extensions

- Can also derive a LDP for non-tree distributions P by judiciously redefining the error event.
- One small problem: P may project onto multiple trees $\{T_1, T_2, \ldots, T_k\}$.



- Easy to show that the class of such P belong to a measure zero set in the probability simplex.

Outline

Approximating The Crossover Rate I

- Euclidean Information Theory [Borade & Zheng '08]:

$$Q_0 \approx Q_1 \quad \Rightarrow \quad D(Q_0||Q_1) \approx \frac{1}{2}\|Q_0 - Q_1\|_{Q_1}^2 = \frac{1}{2}\sum_x \frac{(Q_0(x) - Q_1(x))}{Q_1(x)^2}$$

- Def: Very noisy learning condition on $P_{e,e'}$

$$P_e \approx P_{e'} \qquad \Rightarrow$$

Approximating The Crossover Rate I

- Euclidean Information Theory [Borade & Zheng '08]:

$$Q_0 \approx Q_1 \quad \Rightarrow \quad D(Q_0||Q_1) \approx \frac{1}{2}\|Q_0 - Q_1\|_{Q_1}^2 = \frac{1}{2}\sum_x \frac{(Q_0(x) - Q_1(x))}{Q_1(x)^2}$$

- Def: Very noisy learning condition on $P_{e,e'}$

$$P_e \approx P_{e'} \qquad \Rightarrow \qquad I(P_e) \approx I(P_{e'}).$$

- Def: Given a $P_e = P_{i,j}$ the information density function is

$$s_e(x_i, x_j) \; := \; \log \frac{P_{i,j}(x_i, x_j)}{P_i(x_i)P_j(x_j)}, \quad \forall\, (x_i, x_j) \in \mathcal{X}^2.$$

- Note: $\mathbb{E}[s_e] = I(P_e)$.

Approximating The Crossover Rate II

$$J_{e,e'} := \lim_{n\to\infty} -\frac{1}{n}\log \mathbb{P}\left(\left\{I(\widehat{P}_e) \leq I(\widehat{P}_{e'})\right\}\right).$$

Approximating The Crossover Rate II

$$J_{e,e'} := \lim_{n\to\infty} -\frac{1}{n}\log \mathbb{P}\left(\left\{I(\widehat{P}_e) \leq I(\widehat{P}_{e'})\right\}\right).$$

Theorem

The approximate crossover rate is:

$$\widetilde{J}_{e,e'} = \frac{(I(P_{e'}) - I(P_e))^2}{2\,\mathrm{Var}(s_{e'} - s_e)} = \frac{(\mathbb{E}[s_{e'} - s_e])^2}{2\,\mathrm{Var}(s_{e'} - s_e)}$$

Signal-to-noise ratio for structure learning.

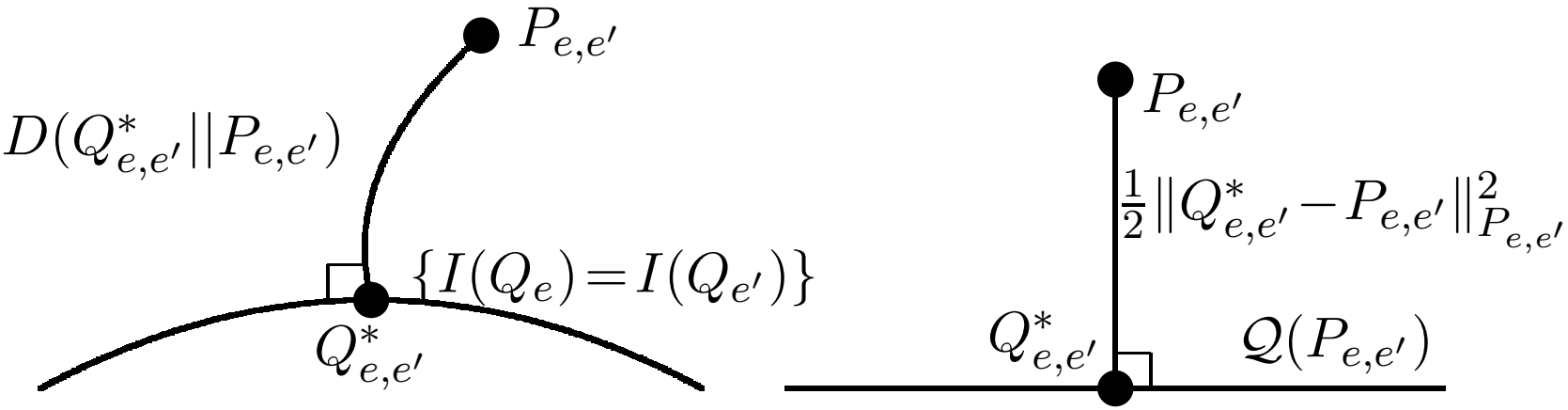
$$\mathrm{SNR} = \left(\frac{m}{\sigma}\right)^2.$$

Approximating The Crossover Rate III

Convexifying the optimization problem by linearizing the constraints.

Approximating The Crossover Rate III

Convexifying the optimization problem by linearizing the constraints.



Non-Convex problem becomes a Least-Squares problem.

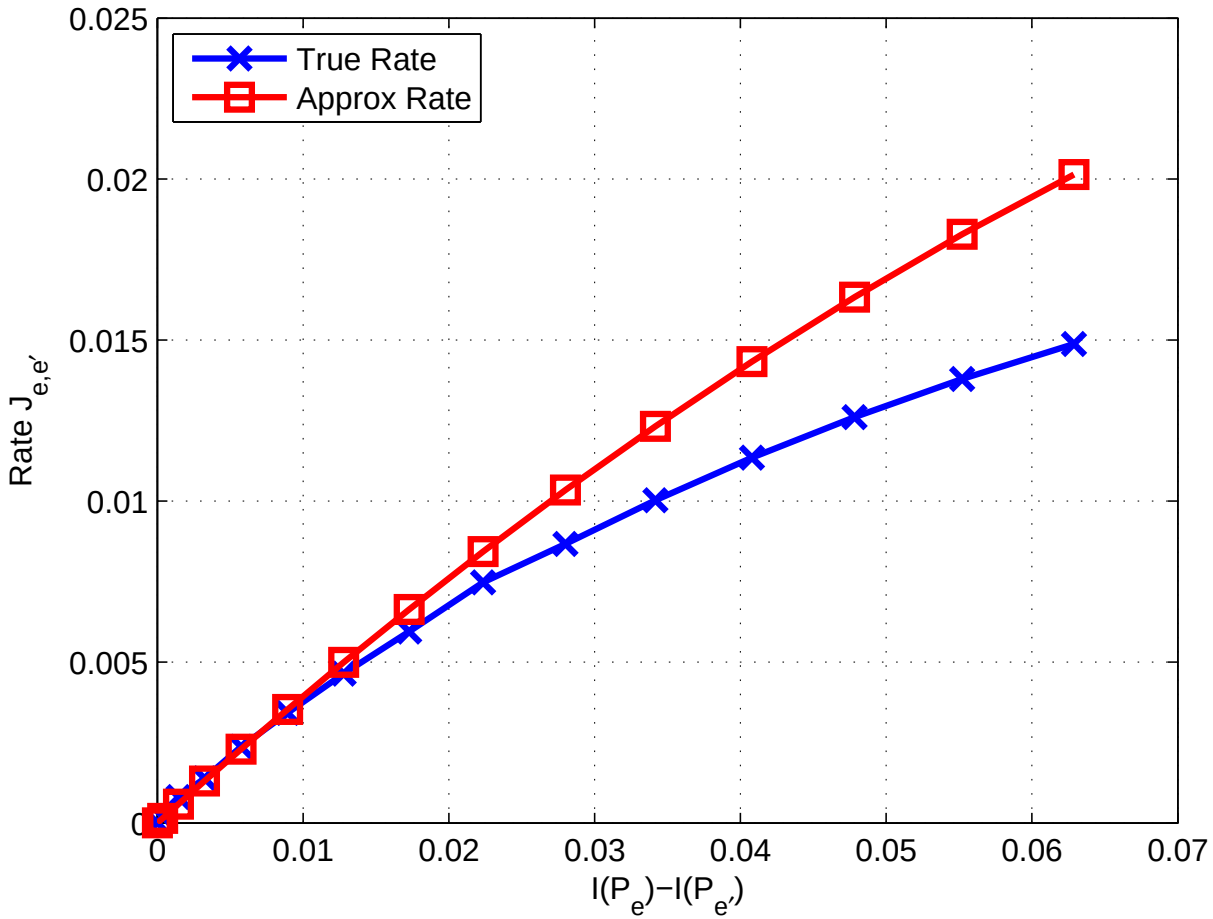
$$\widetilde{J}_{e,e'} = \frac{(\mathbb{E}[s_{e'} - s_e])^2}{2\,\mathrm{Var}[s_{e'} - s_e]}$$

is the signal-to-noise ratio for learning structure.

$$\widetilde{K}_P = \min_{e' \notin E_P} \min_{e \in \mathrm{Path}(e';E_P)} \widetilde{J}_{e,e'}.$$

The Crossover Rate IV

How good is the approximation? We consider a binary model.



Summary of Error Exponent Results

- Goal: Find the rate of decay of the probability of error:

$$K_P := \lim_{n\to\infty} -\frac{1}{n}\log \mathbb{P}\left(\{E_{\mathsf{ML}} \neq E_P\}\right).$$

- Provided the exact rate of decay for the crossover event.
- Employed tools from Large-Deviation theory and basic properties of trees.
- Used Euclidean Information Theory to obtain an intuitive signal-to-noise ratio expression for the crossover rate.
- Found the dominant error tree that relates crossover events to overall error event.

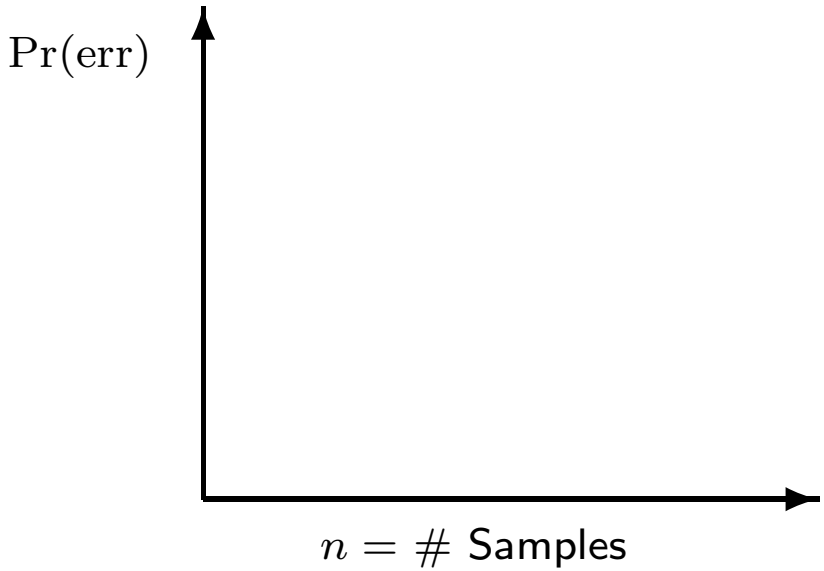
V. Tan, A. Anandkumar, L. Tong, A. Willsky "A Large-Deviation Analysis of the Maximum-Likelihood Learning of Markov Tree Structures," submitted to IEEE Trans. on Information Theory, on Arxiv.

The Gaussian Counterpart

Extremal Tree Structures for Learning

For Gaussian distribution in very noisy learning regime

- Star graphs are hardest to learn, Markov chains are easiest to learn.
- Error exponent increases with tree diameter.
- Keeping the correlations on edges fixed.

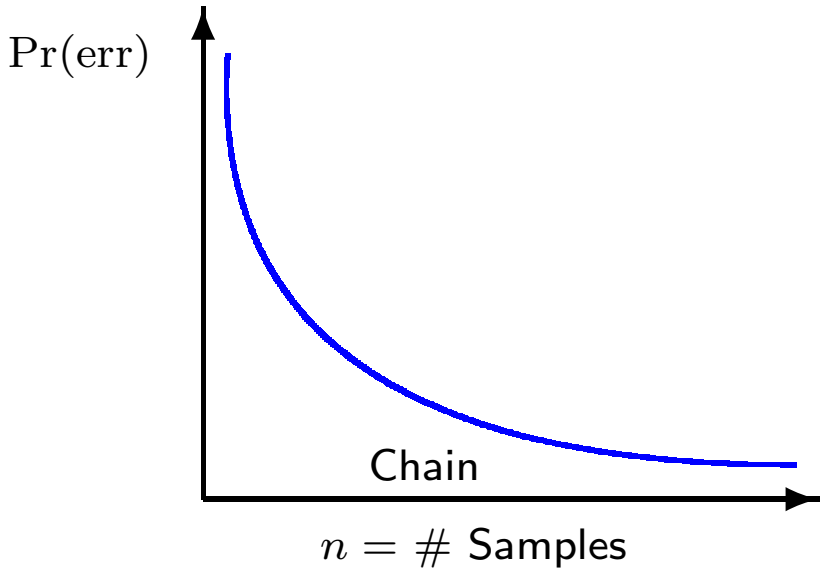


The Gaussian Counterpart

Extremal Tree Structures for Learning

For Gaussian distribution in very noisy learning regime

- Star graphs are hardest to learn, Markov chains are easiest to learn.
- Error exponent increases with tree diameter.
- Keeping the correlations on edges fixed.

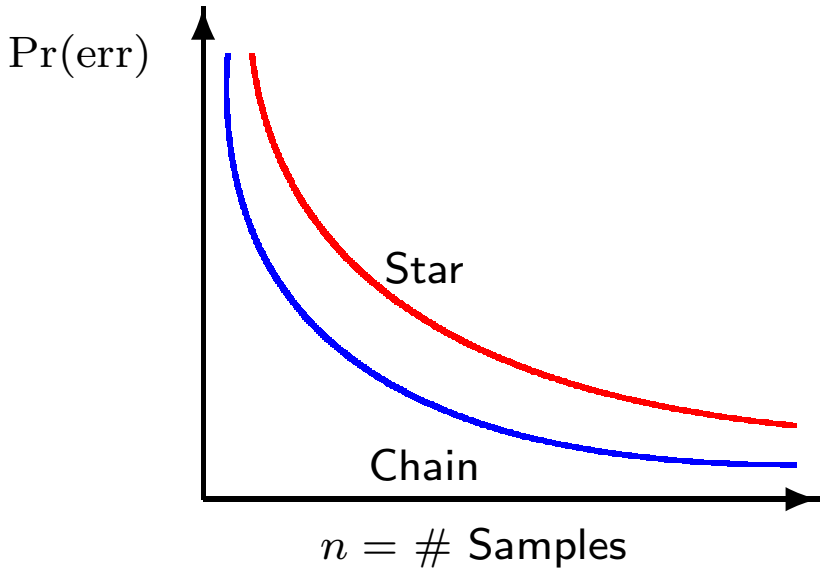


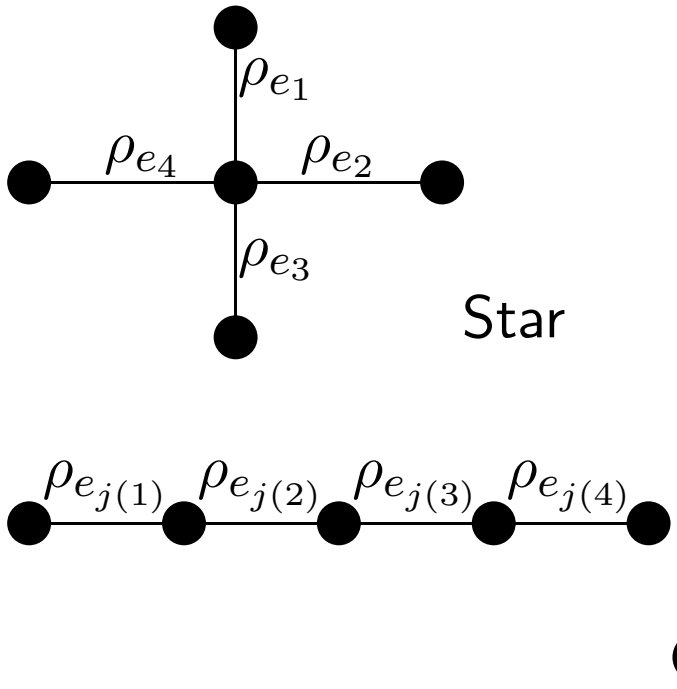
The Gaussian Counterpart

Extremal Tree Structures for Learning

For Gaussian distribution in very noisy learning regime

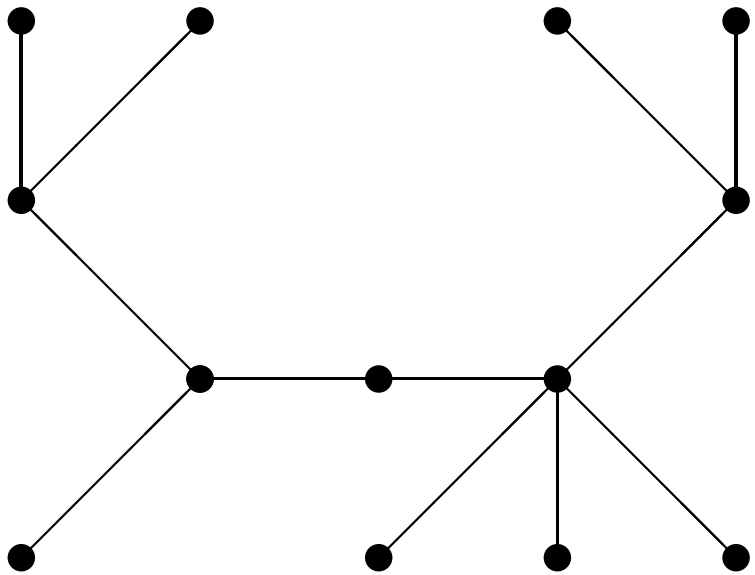
- Star graphs are hardest to learn, Markov chains are easiest to learn.
- Error exponent increases with tree diameter.
- Keeping the correlations on edges fixed.





Methodology

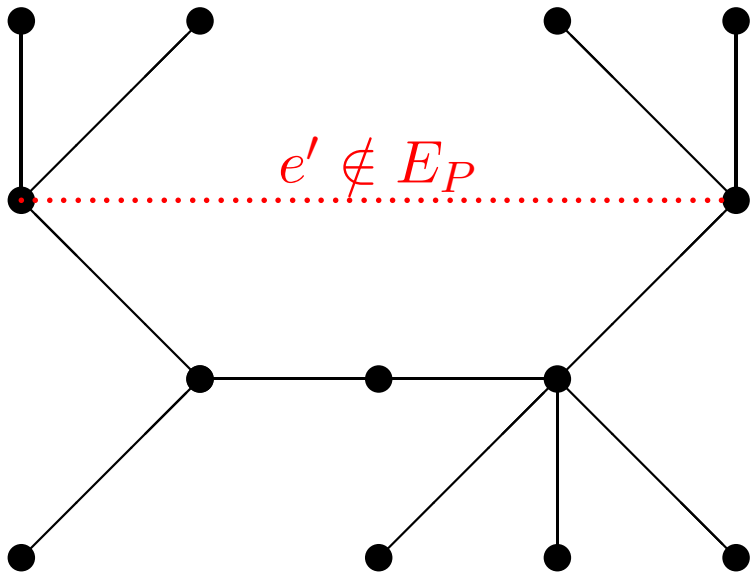
- Error exponent as optimization over a set of local error events.
- Closed-form solution in very-noisy learning regime or weak correlation regime: SNR for learning.
- Correlation decay: Only two-hop node pairs (cherries) likely to be mistaken as edges during estimation.



V. Tan, A. Anandkumar, A. Willsky "Learning Gaussian Tree Models: Analysis of Error Exponents and Extremal Structures", IEEE Tran. on Signal Proc., May 2010.

Methodology

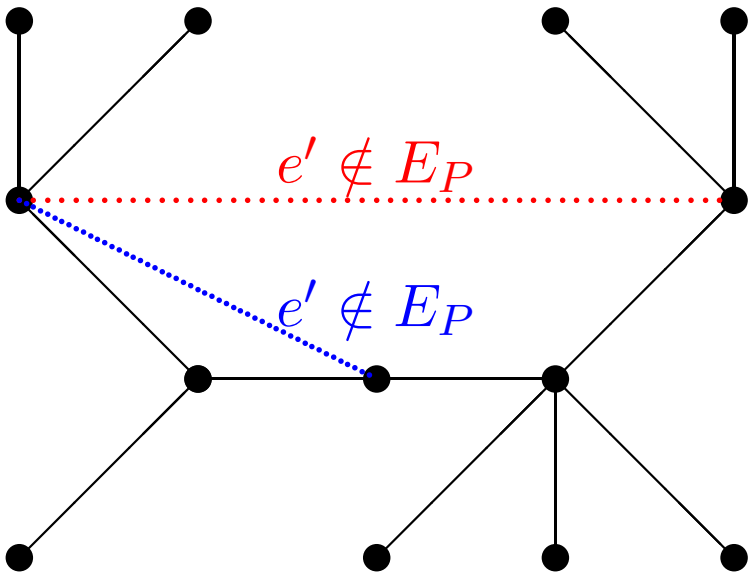
- Error exponent as optimization over a set of local error events.
- Closed-form solution in very-noisy learning regime or weak correlation regime: SNR for learning.
- Correlation decay: Only two-hop node pairs (cherries) likely to be mistaken as edges during estimation.



V. Tan, A. Anandkumar, A. Willsky "Learning Gaussian Tree Models: Analysis of Error Exponents and Extremal Structures", IEEE Tran. on Signal Proc., May 2010.

Methodology

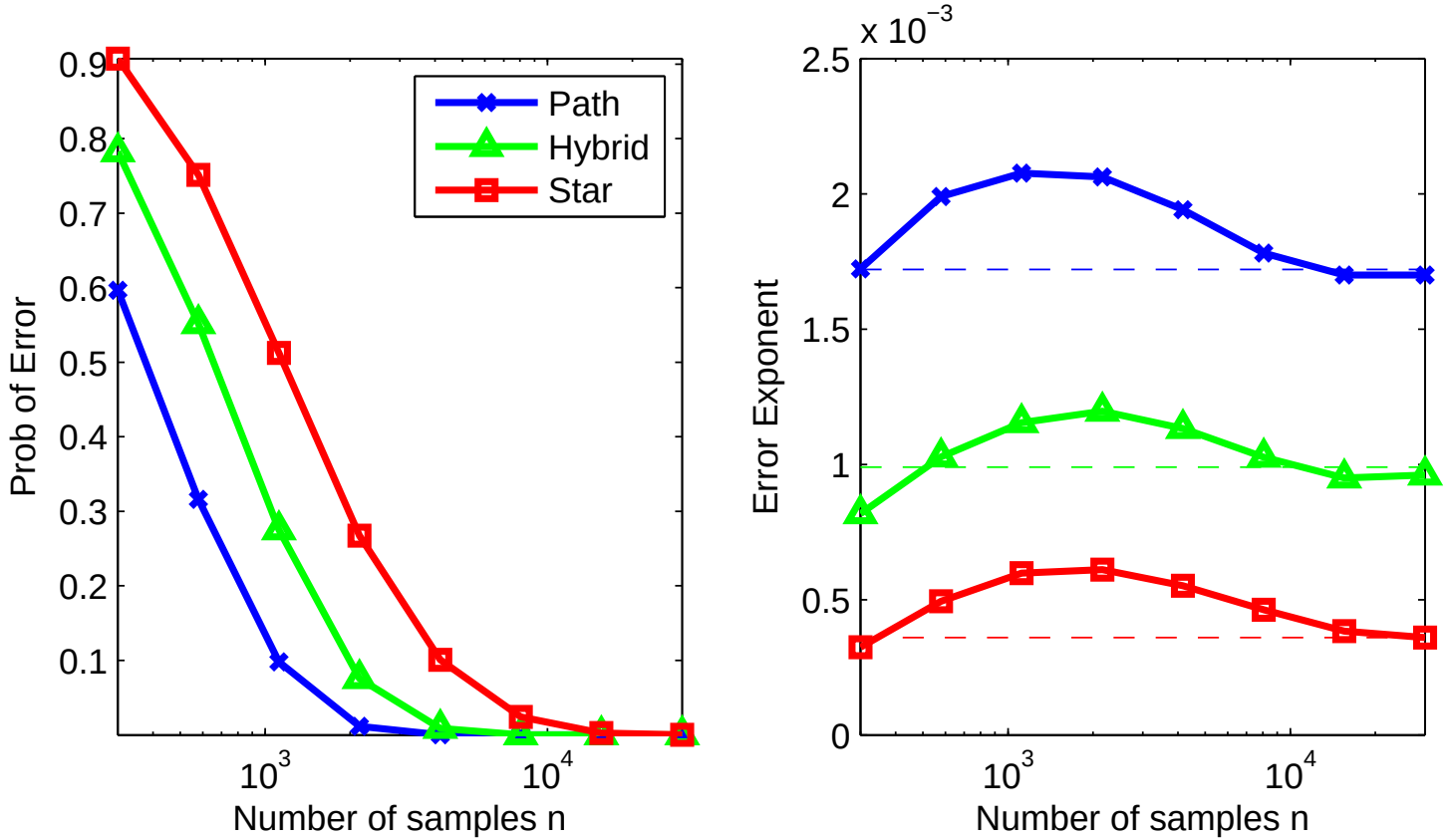
- Error exponent as optimization over a set of local error events.
- Closed-form solution in very-noisy learning regime or weak correlation regime: SNR for learning.
- Correlation decay: Only two-hop node pairs (cherries) likely to be mistaken as edges during estimation.



V. Tan, A. Anandkumar, A. Willsky "Learning Gaussian Tree Models: Analysis of Error Exponents and Extremal Structures", IEEE Tran. on Signal Proc., May 2010.

Extremal Structures III

Chain, Star and Hybrid Graphs for $d = 10$.

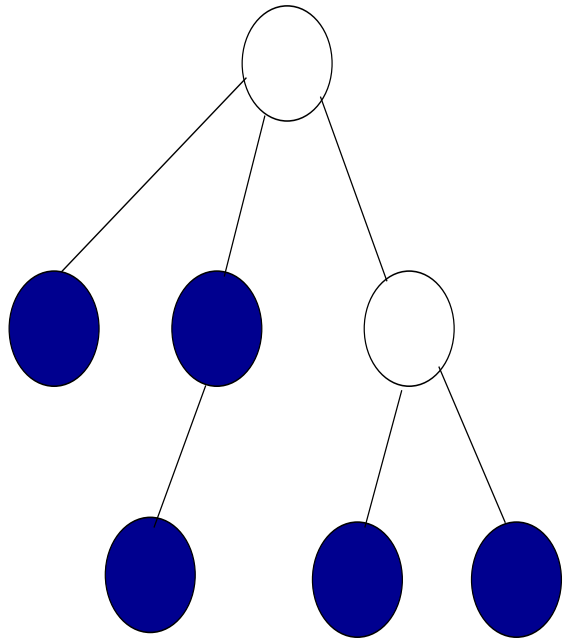


Plot of the error probability and error exponent for 3 tree graphs.

Outline

Latent Tree Model

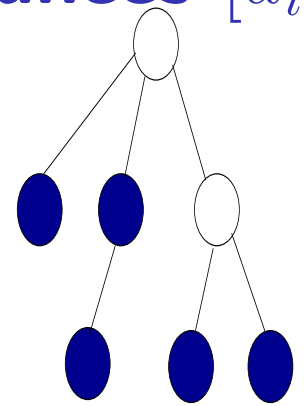
- Visible Nodes V, Hidden Nodes H and $W := V \cup H$
- $T = (W, E)$ is a tree on W



Latent Tree Reconstruction

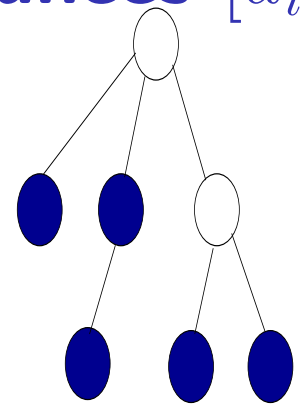
Given n IID samples from node set V, reconstruct latent tree model

Information Distances $[d_{i,j}]$ on Tree Models



Gaussian Model: $\mathbf{X}_W \sim N(\mathbf{0}, \mathbf{\Sigma})$, $d_{ij} := -\log|\rho_{ij}|$, $\rho_{ij} := \frac{\mathbf{\Sigma}_{ij}}{\sqrt{\mathbf{\Sigma}_{ii}\mathbf{\Sigma}_{jj}}}$

Information Distances $[d_{i,j}]$ on Tree Models

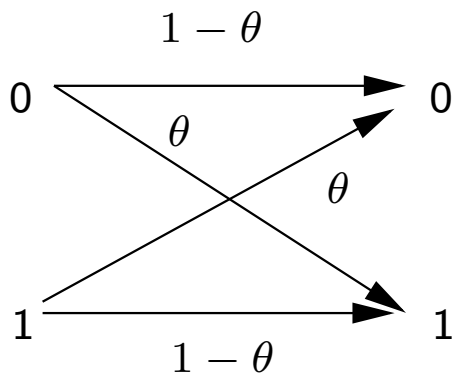


Gaussian Model: $\mathbf{X}_W \sim N(\mathbf{0}, \mathbf{\Sigma})$, $d_{ij} := -\log|\rho_{ij}|$, $\rho_{ij} := \frac{\mathbf{\Sigma}_{ij}}{\sqrt{\mathbf{\Sigma}_{ii}\mathbf{\Sigma}_{jj}}}$

Discrete Symmetric Model

- $X_i \in \{1, 2, \ldots, K\}$ and for $\theta_{ij} \in (0, 1/K)$,

$$p(x_i|x_j) = \begin{cases} 1-(K-1)\theta_{ij} & \text{if } x_i = x_j \\ \theta_{ij}, & \text{o.w.} \end{cases}$$



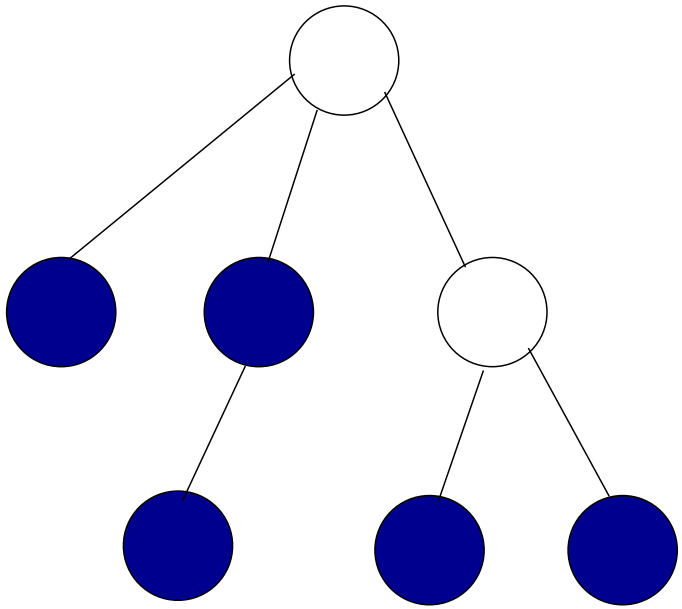
- node marginal is uniform
- Distance is $d_{i,j} := -\log(1 - K\theta_{ij})$.

Markov property on information distances

Markov Property on Trees: $[d_{i,j}]$ is a tree metric

$$d_{k,l} = \sum_{(i,j) \in \mathrm{Path}(k,l;E_p)} d_{i,j},$$

where $\mathrm{Path}(k, l; E_p)$ is the path from k to l along edges E_p of tree.

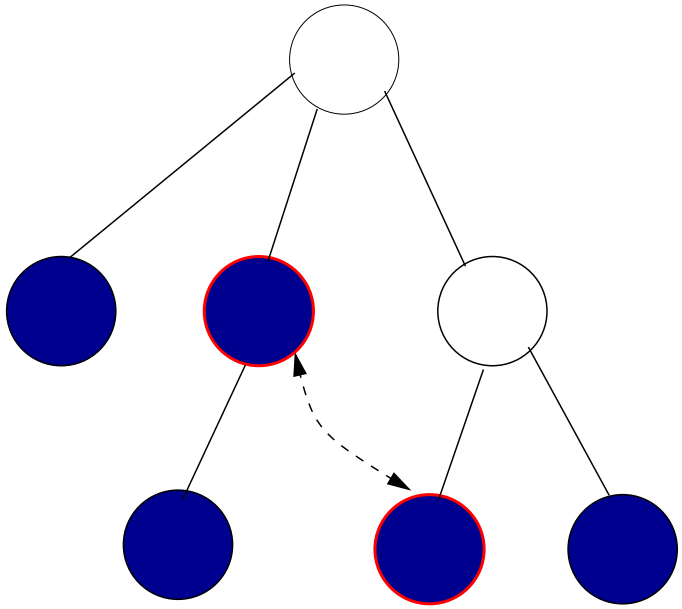


Markov property on information distances

Markov Property on Trees: $[d_{i,j}]$ is a tree metric

$$d_{k,l} = \sum_{(i,j) \in \mathrm{Path}(k,l;E_p)} d_{i,j},$$

where $\mathrm{Path}(k, l; E_p)$ is the path from k to l along edges E_p of tree.

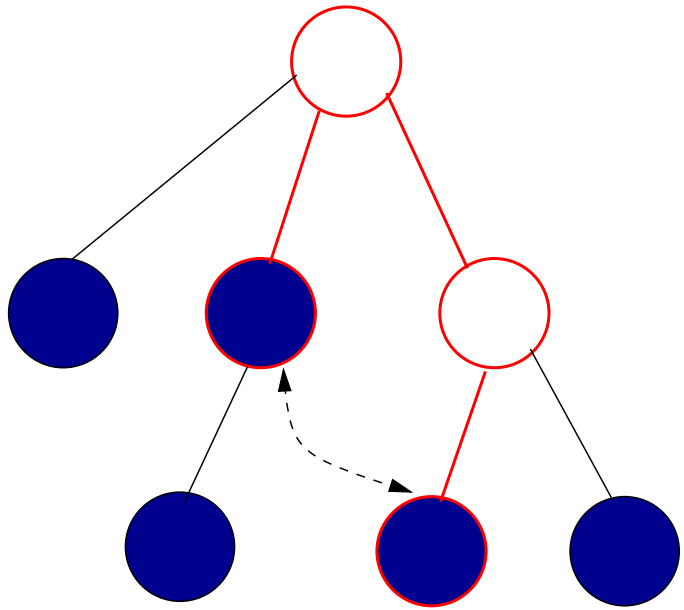


Markov property on information distances

Markov Property on Trees: $[d_{i,j}]$ is a tree metric

$$d_{k,l} = \sum_{(i,j) \in \mathrm{Path}(k,l;E_p)} d_{i,j},$$

where $\mathrm{Path}(k, l; E_p)$ is the path from k to l along edges E_p of tree.



Minimal Tree Extensions

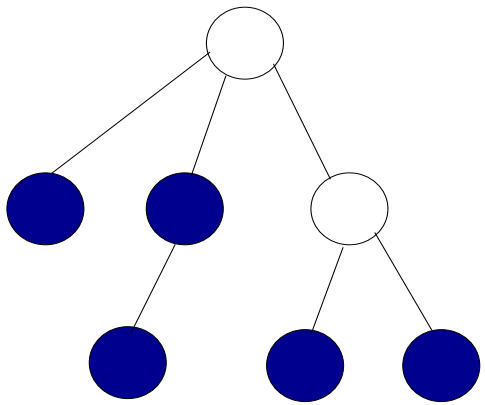
Minimal Tree Extension (Pearl 88)

Tree with least hidden variables explaining observed statistics

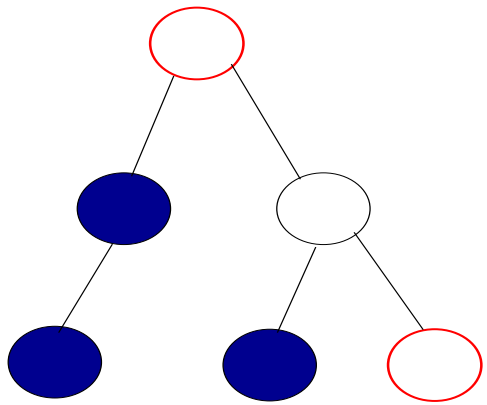
Conditions for Minimality

- Each hidden variable has at least three neighbors: Leaves are visible
- No two variables are perfectly dependent or independent:

$$0 < l \leq d_{i,j} \leq u < \infty, \quad \forall\ (i,j) \in E_p.$$



Minimal



Non-minimal

Definitions: Consistency and Sample Complexity

Structural Consistency

$$\lim_{n\to\infty} \Pr(\{\mathbf{x}_V^n : \widehat{T}^n \neq T_p\}) = 0.$$

Estimation Consistency

$$\lim_{n\to\infty} \Pr(\{\mathbf{x}_V^n : D(p \,||\, \widehat{p}^n) > \epsilon\}) = 0, \quad \forall\ \epsilon > 0.$$

Sample Complexity

An algorithm has sample complexity of $O(f(m))$, if for every $\delta > 0$, error probability $< \delta$ is achieved when number of samples $n > f(m; \delta)$, where m is number of observed nodes.

Summary of Contributions

Reconstruction of general latent tree models from samples

- Propose two distance-based algorithms: ML estimates of distances between observed nodes $[\widehat{\mathbf{d}}]_{i,j\in V}$
  - Recursive Grouping Algorithm
  - Chow-Liu Grouping Algorithm
- Provide theoretical guarantees
  - Structural and estimation consistency for any minimal latent tree
  - Sample complexity of $O(\log m)$ for m observed nodes when effective depth is constant
  - Low computational complexity
- Experimental results demonstrate efficiency of methods

M.J. Choi, V. Tan, A. Anandkumar & A. Willsky, "Consistent and Efficient Reconstruction of Latent Tree Models," Preprint.

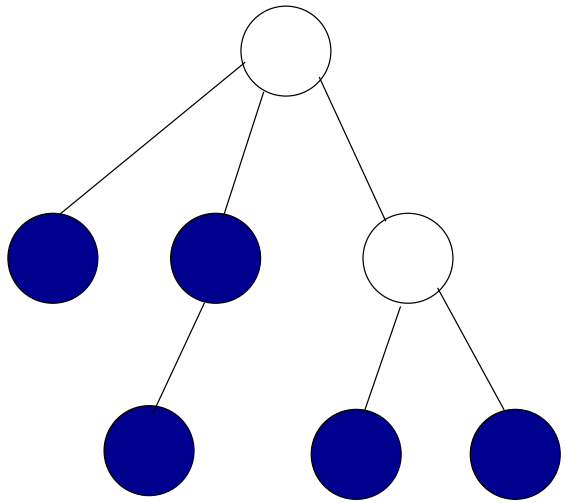
Outline

Siblings Test Based on Information Distances

Exact Statistics: Distances $[d_{i,j}]$

Let $\Phi_{ijk} := d_{i,k} - d_{j,k}$.

- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j}\ \forall\, k, k' \neq i, j, \iff i, j$ leaves with common parent
- $\Phi_{ijk} = d_{i,j},\ \forall\, k \neq i, j, \iff i$ is a leaf and j is its parent.



Sample Statistics: ML Estimates $[\hat{d}_{i,j}]$

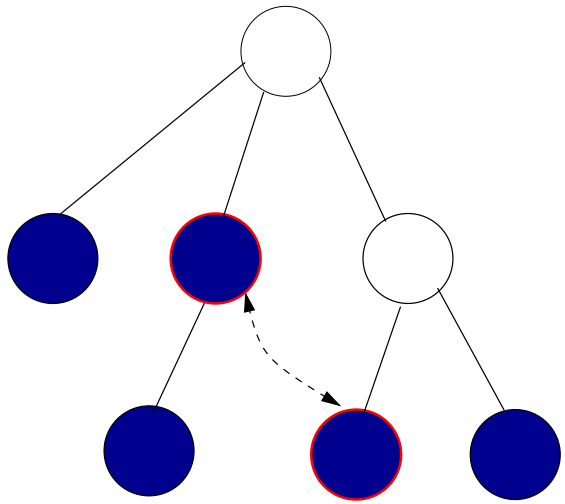
Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships

Siblings Test Based on Information Distances

Exact Statistics: Distances $[d_{i,j}]$

Let $\Phi_{ijk} := d_{i,k} - d_{j,k}$.

- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j}\ \forall\, k, k' \neq i, j, \iff i, j$ leaves with common parent
- $\Phi_{ijk} = d_{i,j},\ \forall\, k \neq i, j, \iff i$ is a leaf and j is its parent.



Sample Statistics: ML Estimates $[\hat{d}_{i,j}]$

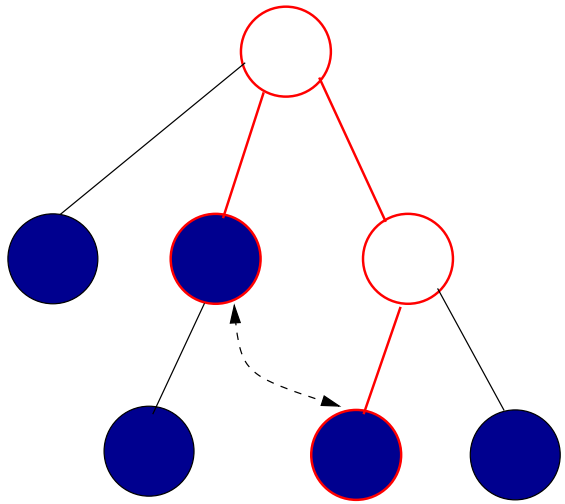
Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships

Siblings Test Based on Information Distances

Exact Statistics: Distances $[d_{i,j}]$

Let $\Phi_{ijk} := d_{i,k} - d_{j,k}$.

- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j}\ \forall\, k, k' \neq i, j, \iff i, j$ leaves with common parent
- $\Phi_{ijk} = d_{i,j},\ \forall\, k \neq i, j, \iff i$ is a leaf and j is its parent.



Sample Statistics: ML Estimates $[\hat{d}_{i,j}]$

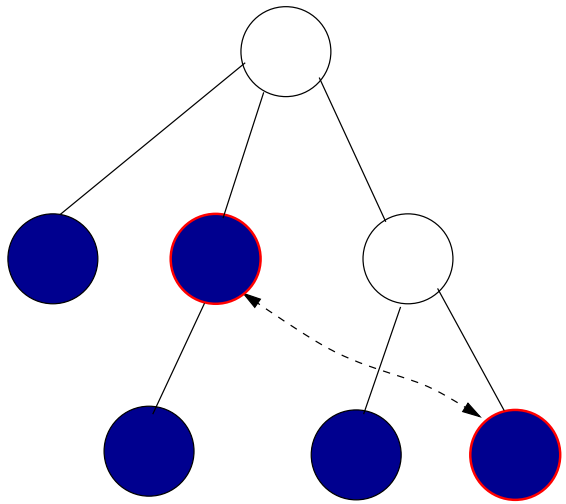
Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships

Siblings Test Based on Information Distances

Exact Statistics: Distances $[d_{i,j}]$

Let $\Phi_{ijk} := d_{i,k} - d_{j,k}$.

- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j}\ \forall\, k, k' \neq i, j, \iff i, j$ leaves with common parent
- $\Phi_{ijk} = d_{i,j},\ \forall\, k \neq i, j, \iff i$ is a leaf and j is its parent.



Sample Statistics: ML Estimates $[\hat{d}_{i,j}]$

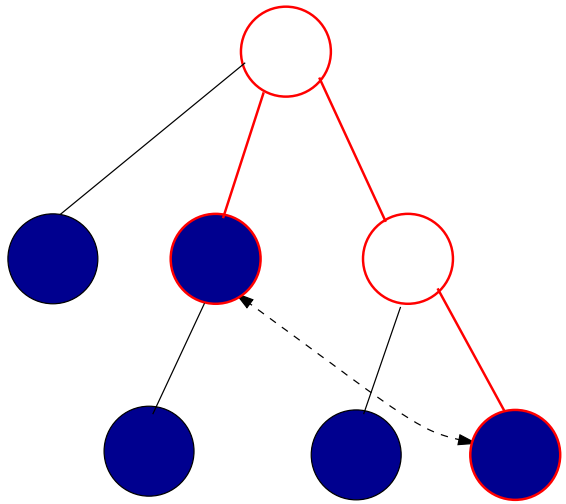
Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships

Siblings Test Based on Information Distances

Exact Statistics: Distances $[d_{i,j}]$

Let $\Phi_{ijk} := d_{i,k} - d_{j,k}$.

- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j}\ \forall\, k, k' \neq i, j, \iff i, j$ leaves with common parent
- $\Phi_{ijk} = d_{i,j},\ \forall\, k \neq i, j, \iff i$ is a leaf and j is its parent.



Sample Statistics: ML Estimates $[\hat{d}_{i,j}]$

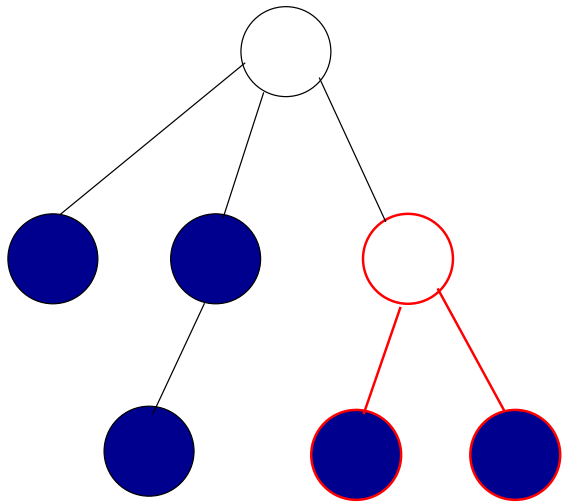
Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships

Siblings Test Based on Information Distances

Exact Statistics: Distances $[d_{i,j}]$

Let $\Phi_{ijk} := d_{i,k} - d_{j,k}$.

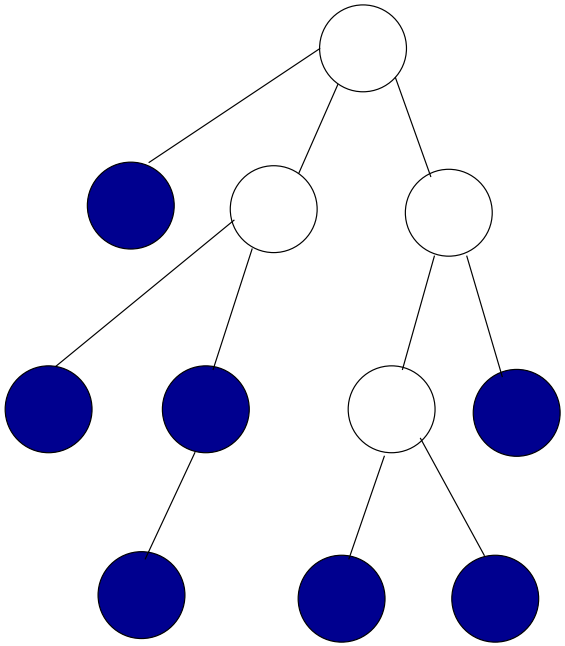
- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j}\ \forall\, k, k' \neq i, j, \iff i, j$ leaves with common parent
- $\Phi_{ijk} = d_{i,j},\ \forall\, k \neq i, j, \iff i$ is a leaf and j is its parent.



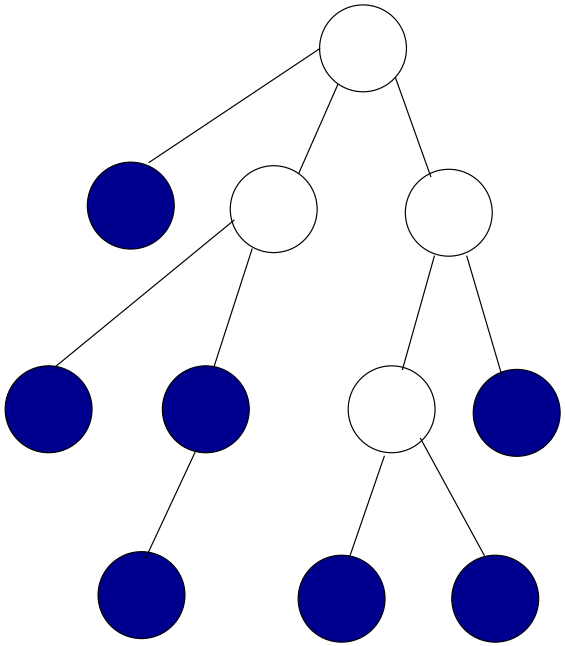
Sample Statistics: ML Estimates $[\hat{d}_{i,j}]$

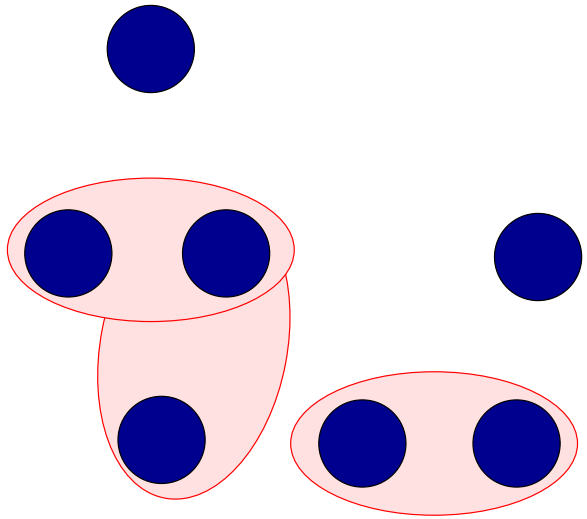
Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships

Recursive Grouping: Example and Guarantees

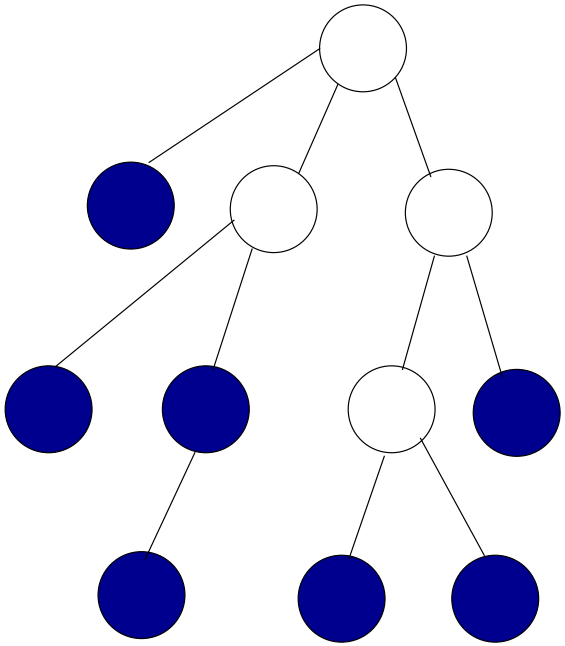


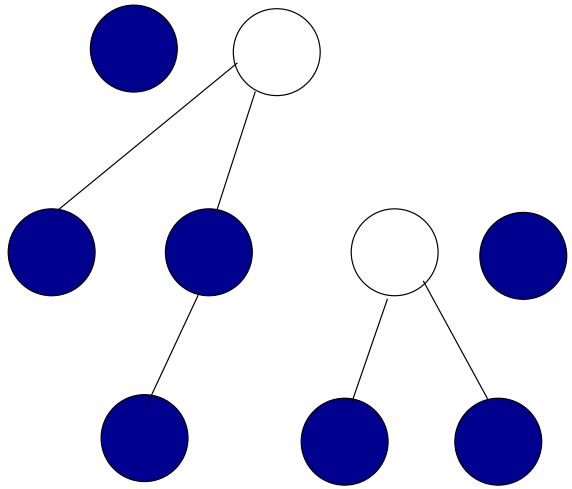
Recursive Grouping: Example and Guarantees



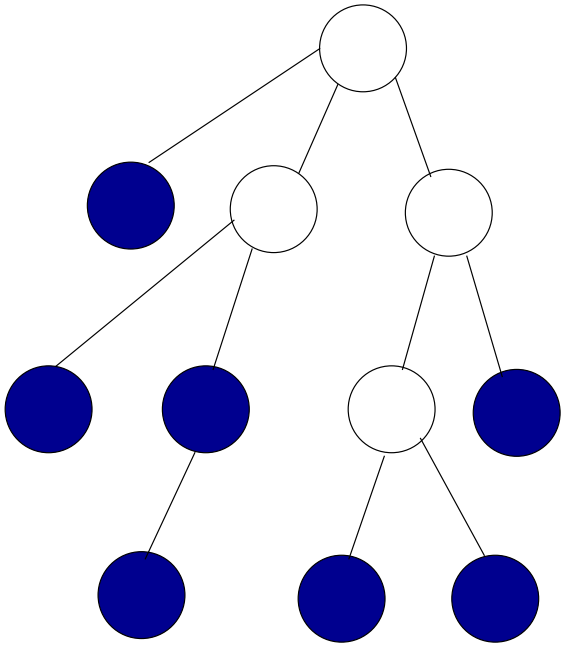


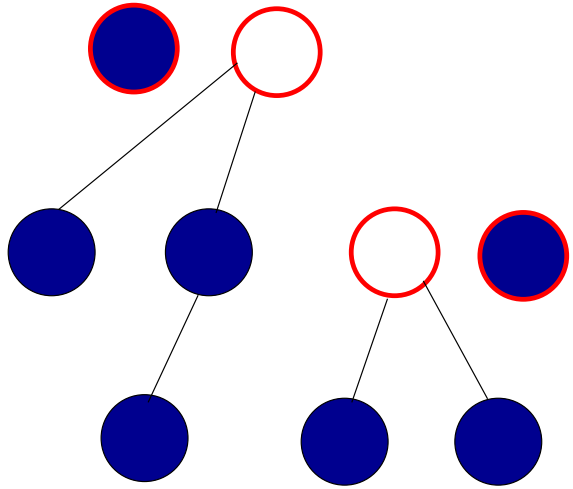
Recursive Grouping: Example and Guarantees



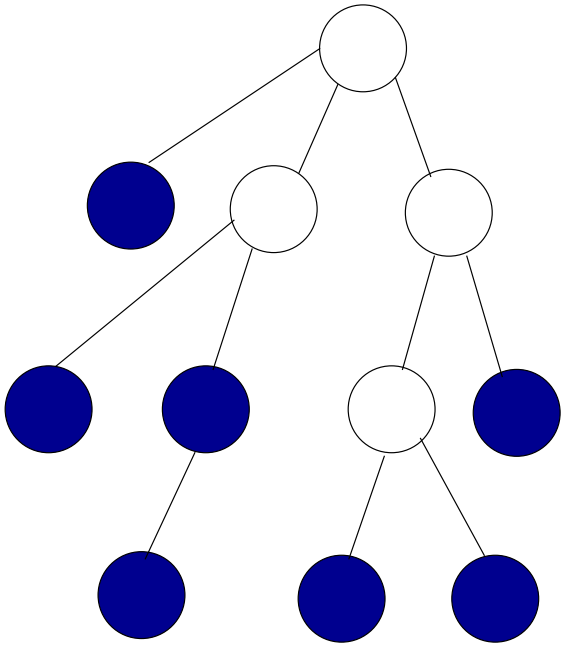


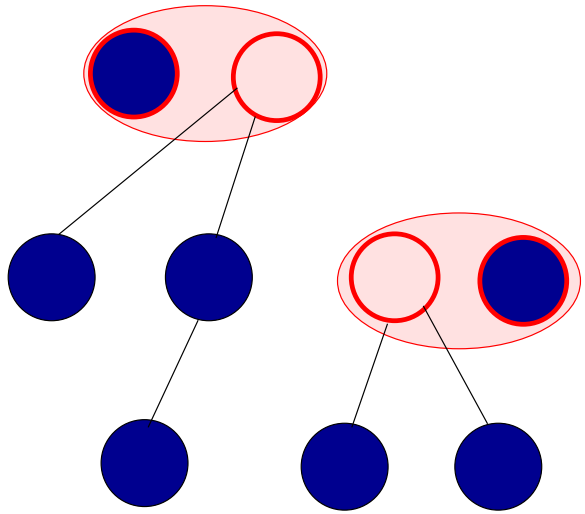
Recursive Grouping: Example and Guarantees



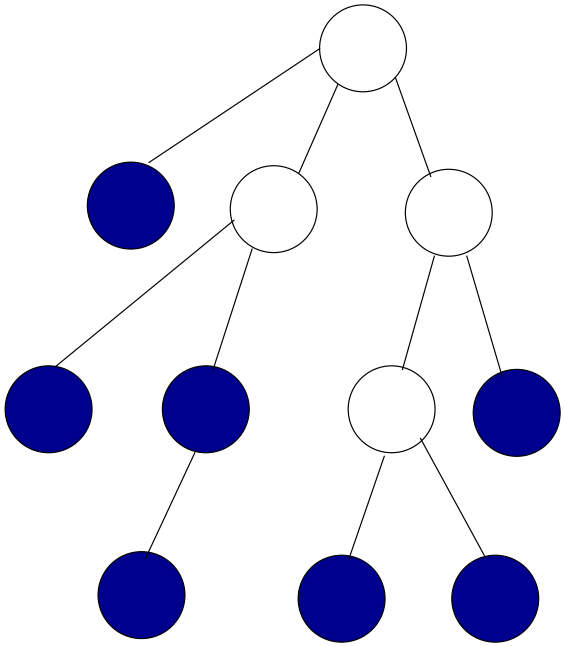


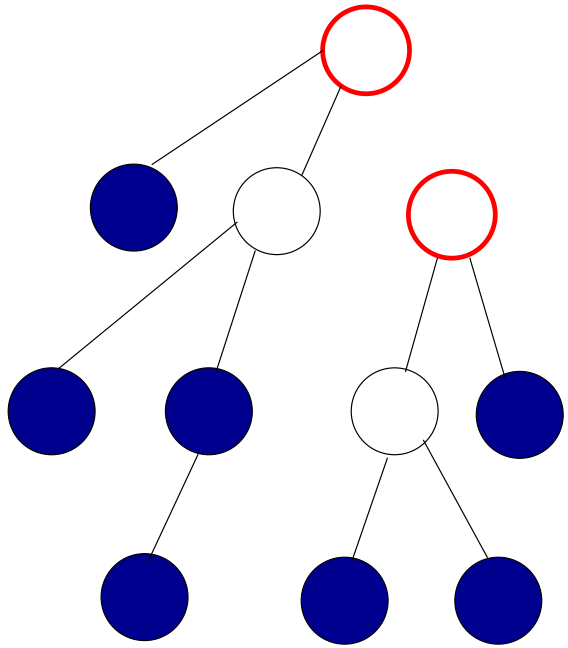
Recursive Grouping: Example and Guarantees



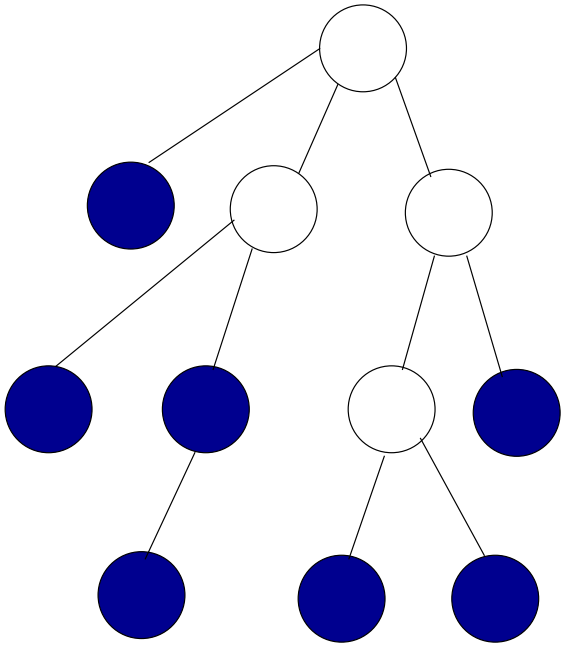


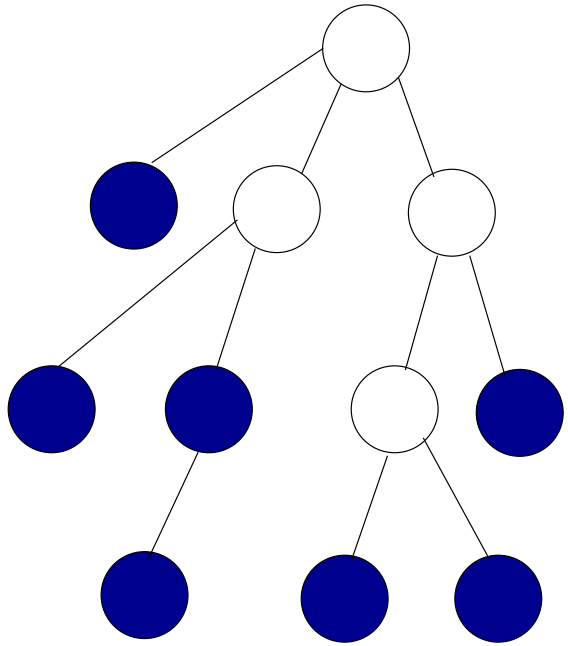
Recursive Grouping: Example and Guarantees



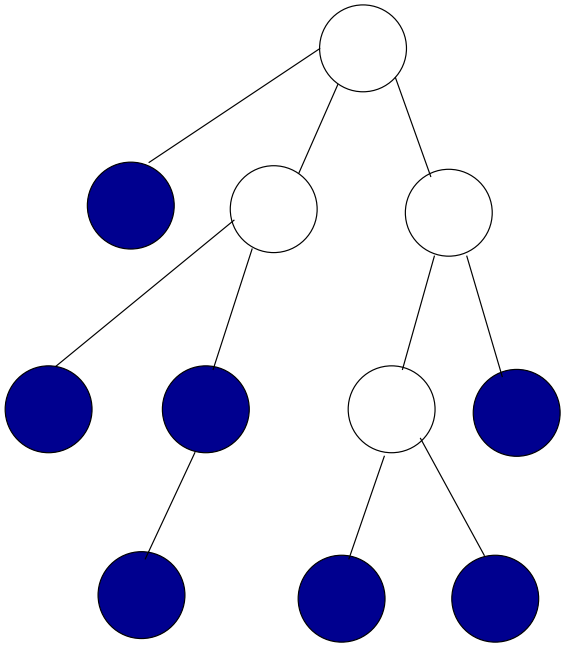


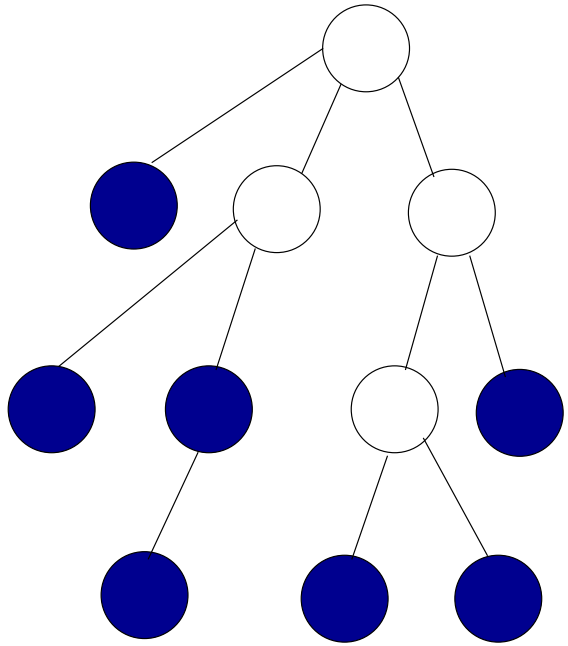
Recursive Grouping: Example and Guarantees





Recursive Grouping: Example and Guarantees





Guarantees

- Structural and estimation consistency for all minimal latent trees
- Sample complexity of $O(\log m)$ for m observed nodes when effective depth is fixed
- Computational complexity of $O(\mathrm{diam}(\hat{T}_p)m^3)$.

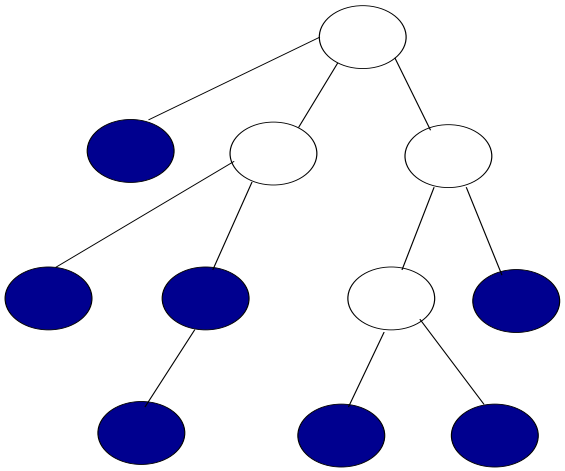
Overview of Chow-Liu Based Grouping

Shortcomings of Recursive Grouping

- Computationally intensive: check all observed node pairs as siblings
- Sibling test: local test. Error prone

Pre-processing to improve efficiency and accuracy

Build a Chow-Liu tree, rule out many pairs of observed nodes as siblings



Reconstruct Latent Tree by Transforming Chow-Liu Tree

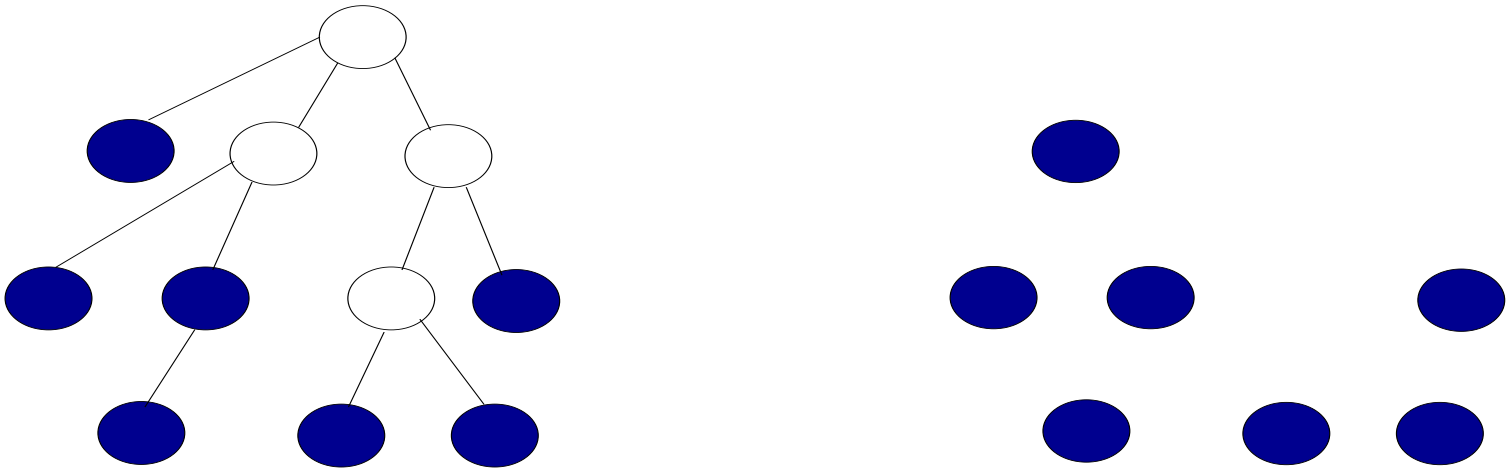
Overview of Chow-Liu Based Grouping

Shortcomings of Recursive Grouping

- Computationally intensive: check all observed node pairs as siblings
- Sibling test: local test. Error prone

Pre-processing to improve efficiency and accuracy

Build a Chow-Liu tree, rule out many pairs of observed nodes as siblings



Reconstruct Latent Tree by Transforming Chow-Liu Tree

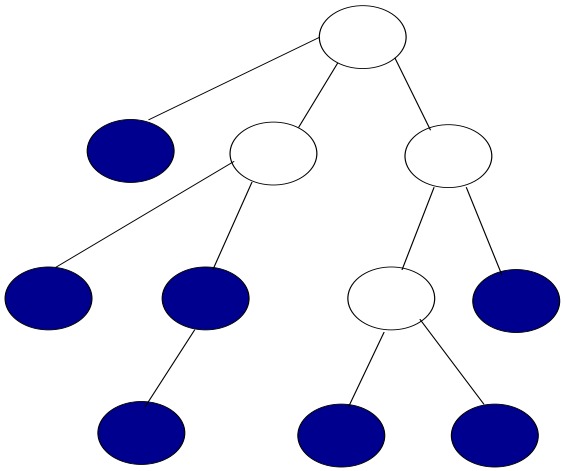
Overview of Chow-Liu Based Grouping

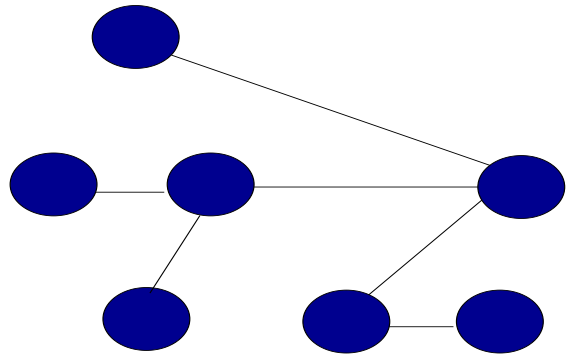
Shortcomings of Recursive Grouping

- Computationally intensive: check all observed node pairs as siblings
- Sibling test: local test. Error prone

Pre-processing to improve efficiency and accuracy

Build a Chow-Liu tree, rule out many pairs of observed nodes as siblings





Reconstruct Latent Tree by Transforming Chow-Liu Tree

Chow-Liu Tree on Observed Nodes

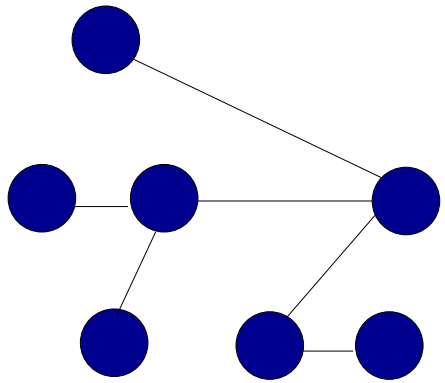
Chow-Liu tree: ML tree over observed nodes V

- $\hat{p}_{\mathrm{CL}}$: Tree distribution closest (in KL-divergence) to the empirical distribution

$$\hat{p}_{\mathrm{CL}} := \operatorname*{argmin}_{\nu \in \mathsf{Tree}} D(\widehat{\mu} \,||\, \nu).$$

- Chow-Liu algorithm: $\widehat{T}_{\mathrm{CL}} = \operatorname*{argmax}_{T=(V,E)\in\mathcal{T}} \sum_{e\in E} I(\hat{\mu}_e)$
- Chow-Liu tree in terms of distance estimates

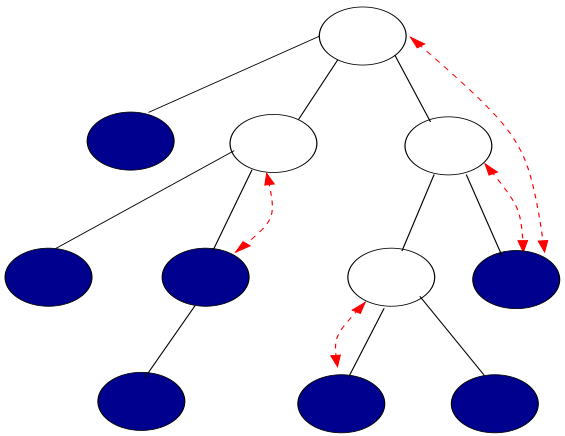
$$\widehat{T}_{\mathrm{CL}} = \mathrm{MST}(V; \widehat{\mathbf{d}}) := \operatorname*{argmin}_{T=(V,E)\in\mathcal{T}} \sum_{e\in E} \hat{d}_e.$$

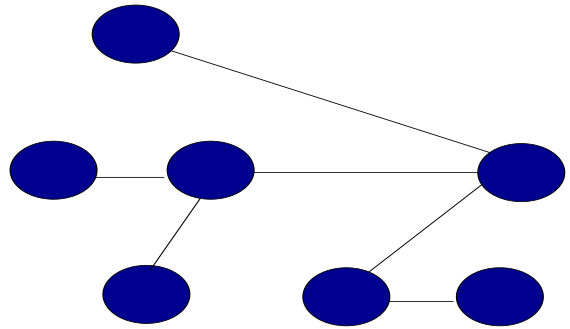


Relating Chow-Liu Tree with Latent Tree

Surrogate $\mathrm{Sg}(i)$ for node i: visible node with strongest correlation

$$\mathrm{Sg}(i; T_p, V) := \operatorname*{argmin}_{j \in V} d_{i,j}$$





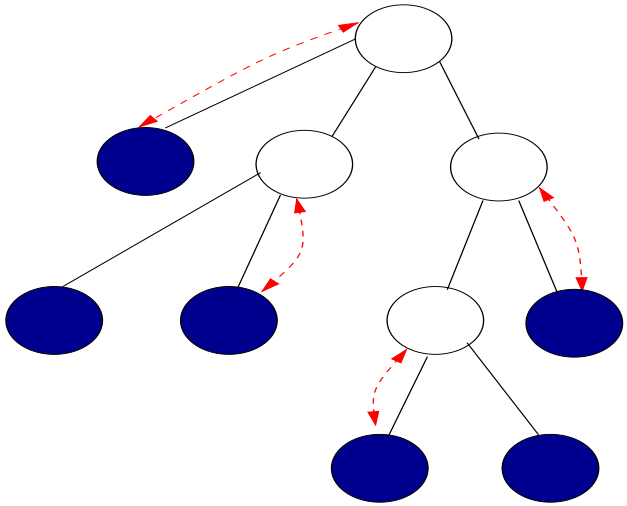
Properties of Chow-Liu Tree and Surrogacy

Neighborhood Preservation: for $i, j \in W$ with $\mathrm{Sg}(i) \neq \mathrm{Sg}(j)$,

$$(i,j) \in E_p \Rightarrow (\mathrm{Sg}(i), \mathrm{Sg}(j)) \in \mathrm{MST}(V; \mathbf{d}).$$

Blind Transformation of Chow-Liu Tree

In Chow-Liu tree, replace each internal node i with a hidden node h and place i as a child of h.

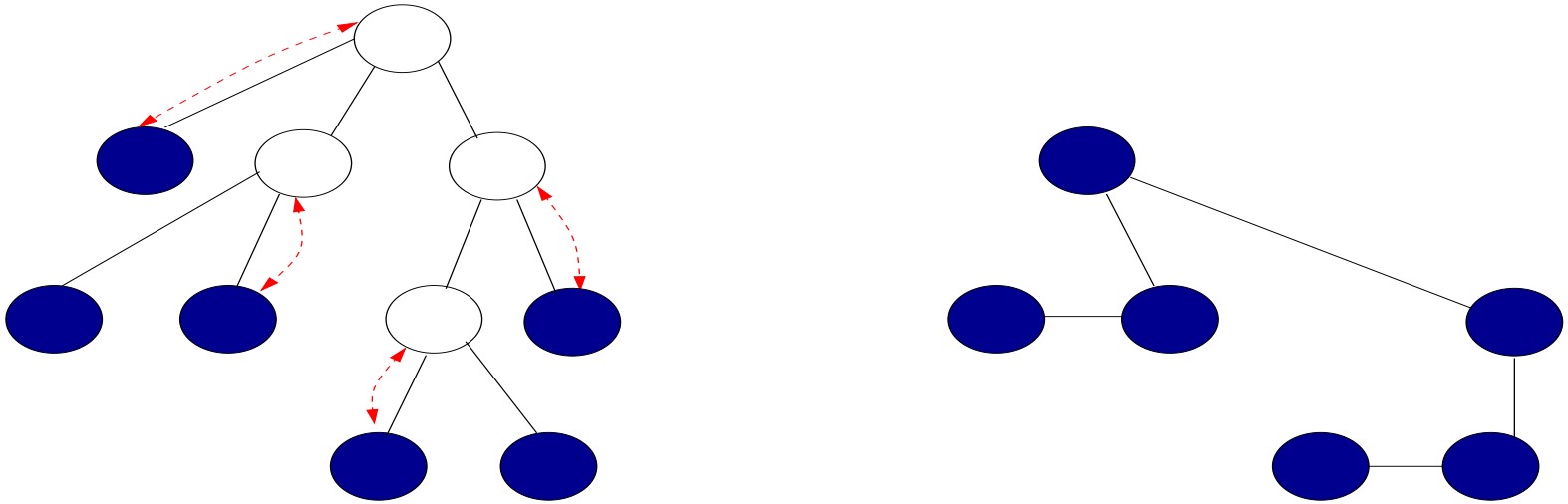


Guarantees

- Consistent: if only leaves observed and each hidden node has strongest correlation with one of its children
- Computational complexity $O(m^2 \log m)$ for m observed nodes

Blind Transformation of Chow-Liu Tree

In Chow-Liu tree, replace each internal node i with a hidden node h and place i as a child of h.

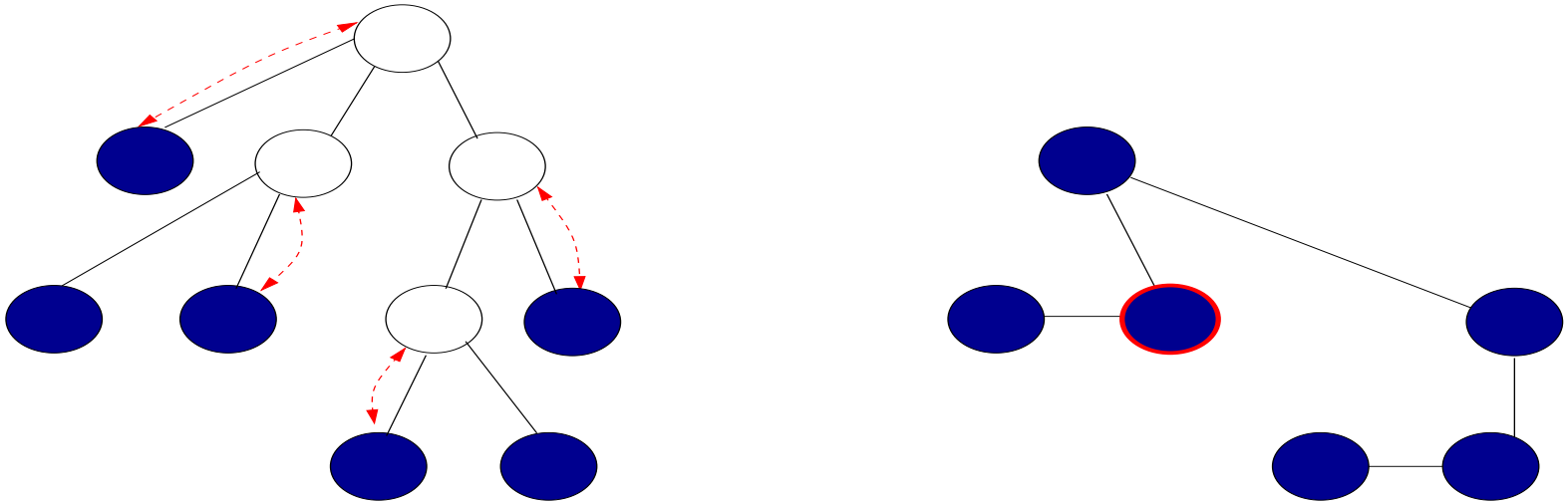


Guarantees

- Consistent: if only leaves observed and each hidden node has strongest correlation with one of its children
- Computational complexity $O(m^2 \log m)$ for m observed nodes

Blind Transformation of Chow-Liu Tree

In Chow-Liu tree, replace each internal node i with a hidden node h and place i as a child of h.

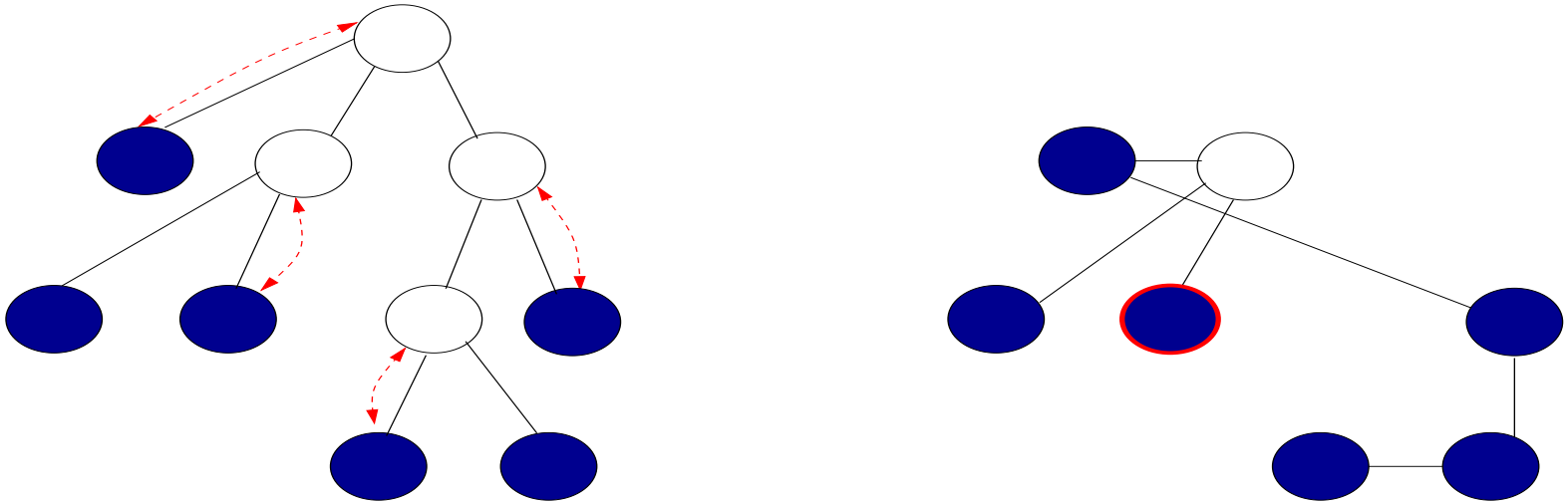


Guarantees

- Consistent: if only leaves observed and each hidden node has strongest correlation with one of its children
- Computational complexity $O(m^2 \log m)$ for m observed nodes

Blind Transformation of Chow-Liu Tree

In Chow-Liu tree, replace each internal node i with a hidden node h and place i as a child of h.

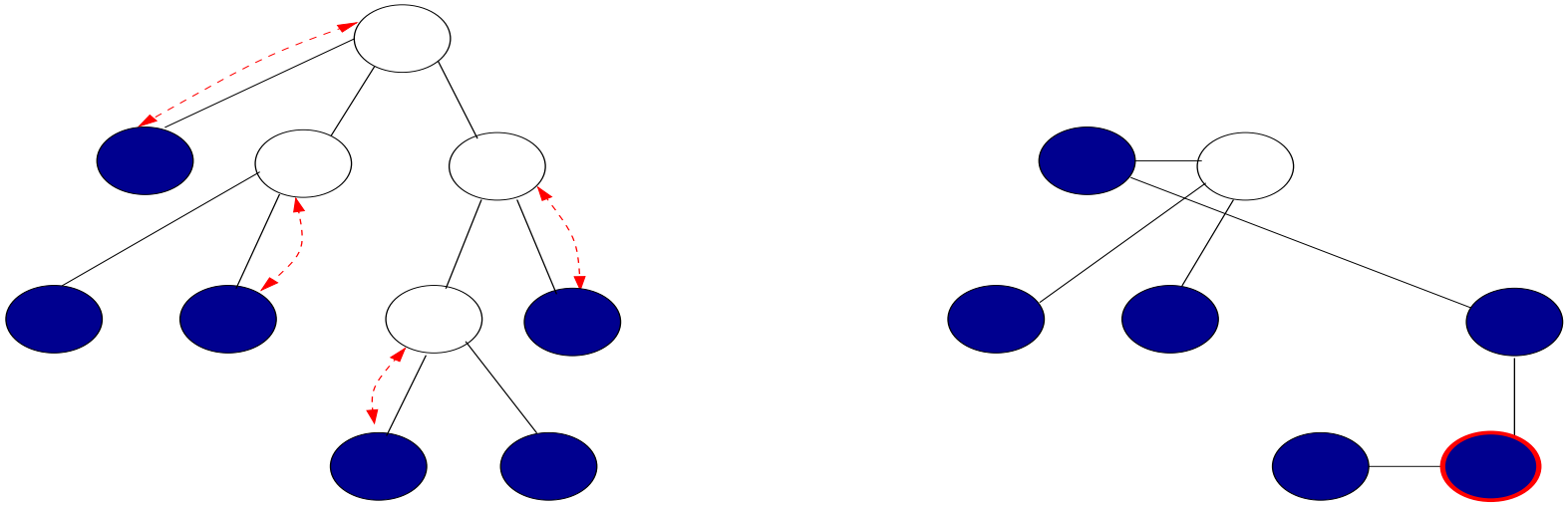


Guarantees

- Consistent: if only leaves observed and each hidden node has strongest correlation with one of its children
- Computational complexity $O(m^2 \log m)$ for m observed nodes

Blind Transformation of Chow-Liu Tree

In Chow-Liu tree, replace each internal node i with a hidden node h and place i as a child of h.

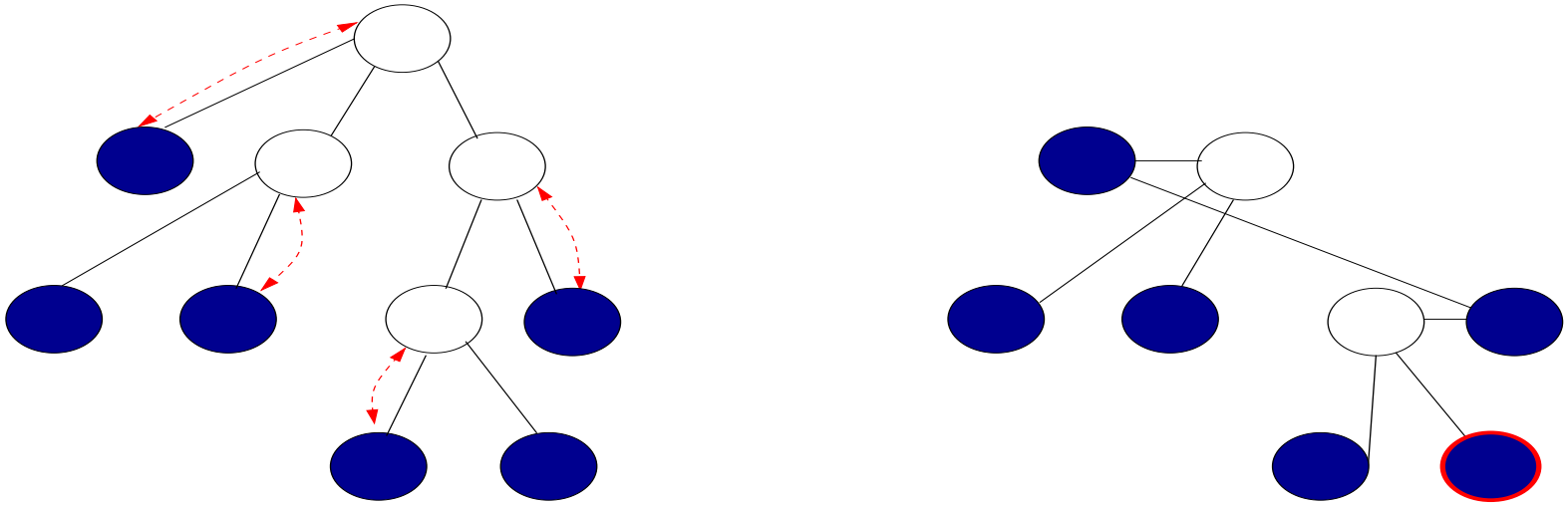


Guarantees

- Consistent: if only leaves observed and each hidden node has strongest correlation with one of its children
- Computational complexity $O(m^2 \log m)$ for m observed nodes

Blind Transformation of Chow-Liu Tree

In Chow-Liu tree, replace each internal node i with a hidden node h and place i as a child of h.

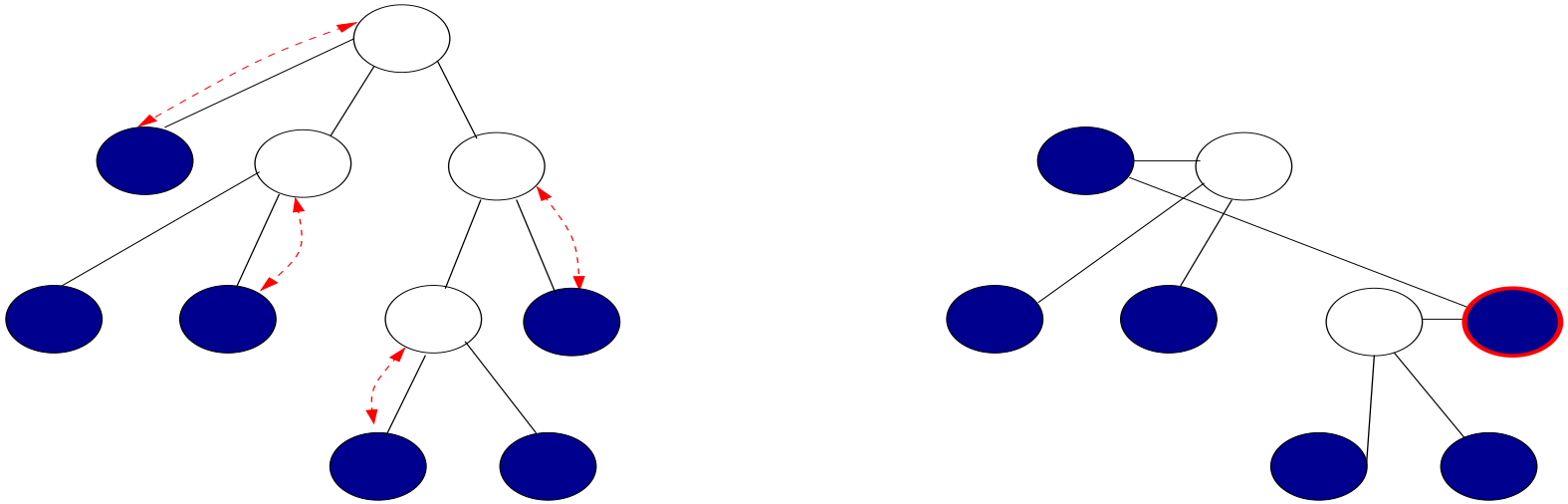


Guarantees

- Consistent: if only leaves observed and each hidden node has strongest correlation with one of its children
- Computational complexity $O(m^2 \log m)$ for m observed nodes

Blind Transformation of Chow-Liu Tree

In Chow-Liu tree, replace each internal node i with a hidden node h and place i as a child of h.

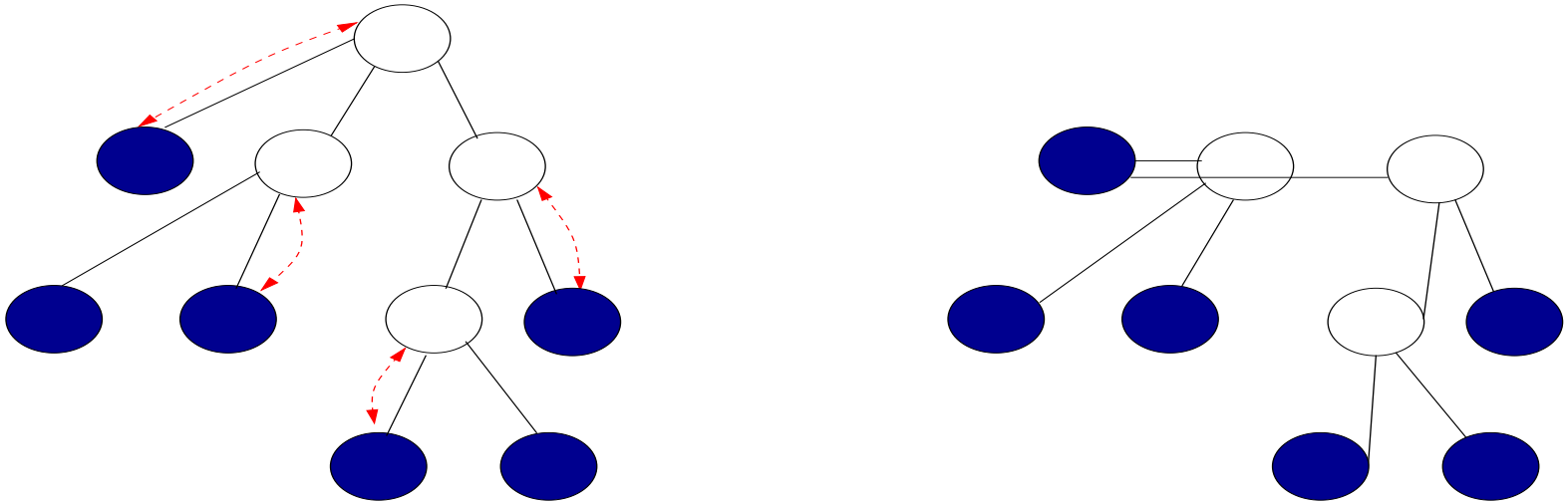


Guarantees

- Consistent: if only leaves observed and each hidden node has strongest correlation with one of its children
- Computational complexity $O(m^2 \log m)$ for m observed nodes

Blind Transformation of Chow-Liu Tree

In Chow-Liu tree, replace each internal node i with a hidden node h and place i as a child of h.

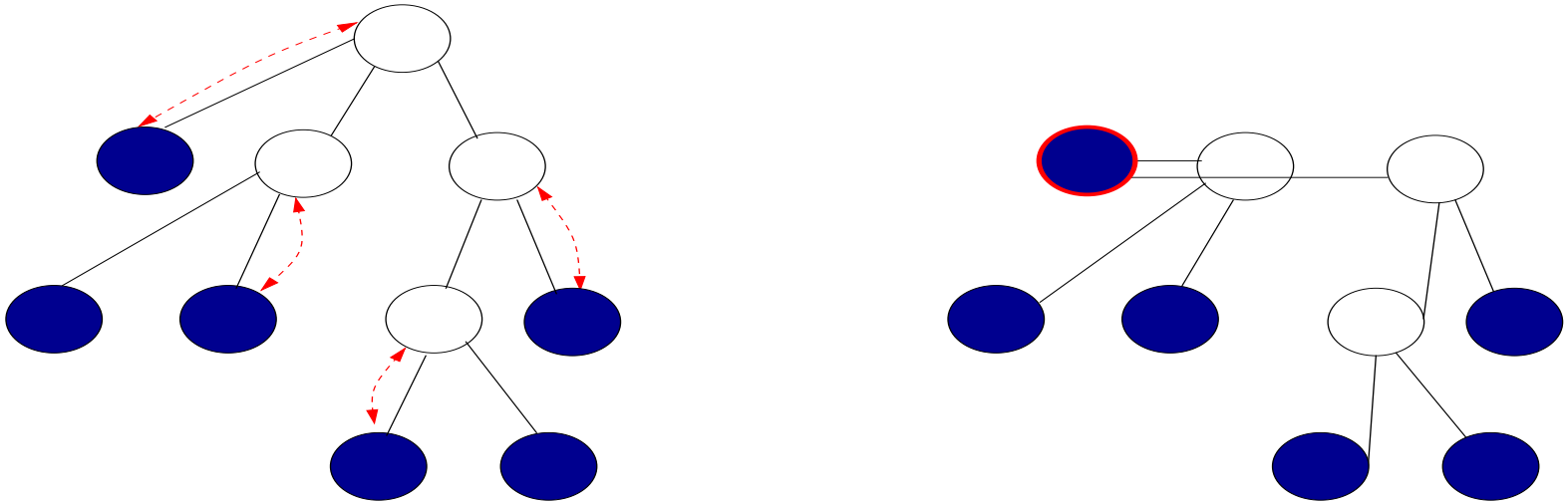


Guarantees

- Consistent: if only leaves observed and each hidden node has strongest correlation with one of its children
- Computational complexity $O(m^2 \log m)$ for m observed nodes

Blind Transformation of Chow-Liu Tree

In Chow-Liu tree, replace each internal node i with a hidden node h and place i as a child of h.

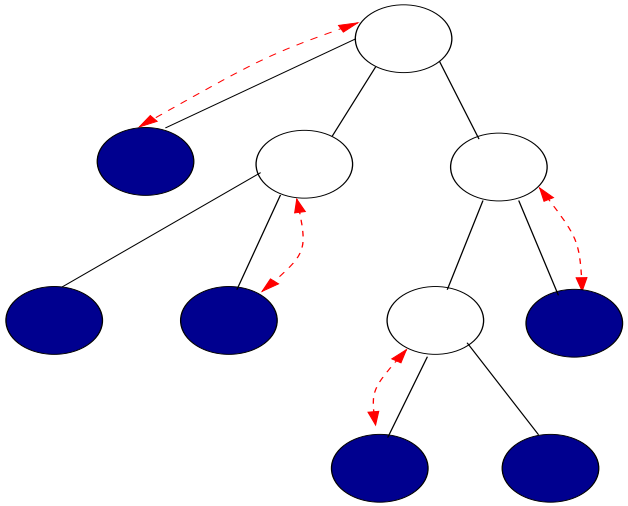


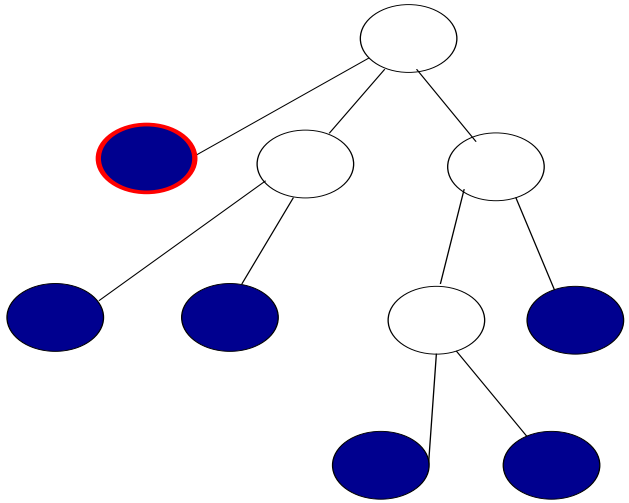
Guarantees

- Consistent: if only leaves observed and each hidden node has strongest correlation with one of its children
- Computational complexity $O(m^2 \log m)$ for m observed nodes

Blind Transformation of Chow-Liu Tree

In Chow-Liu tree, replace each internal node i with a hidden node h and place i as a child of h.



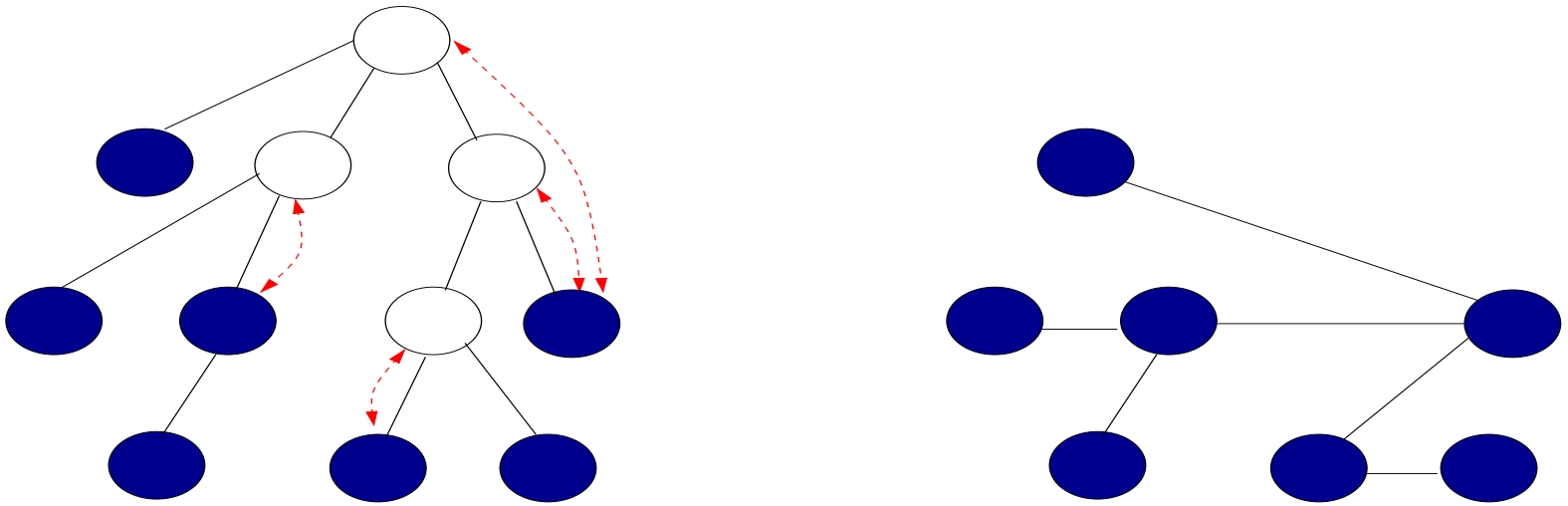


Guarantees

- Consistent: if only leaves observed and each hidden node has strongest correlation with one of its children
- Computational complexity $O(m^2 \log m)$ for m observed nodes

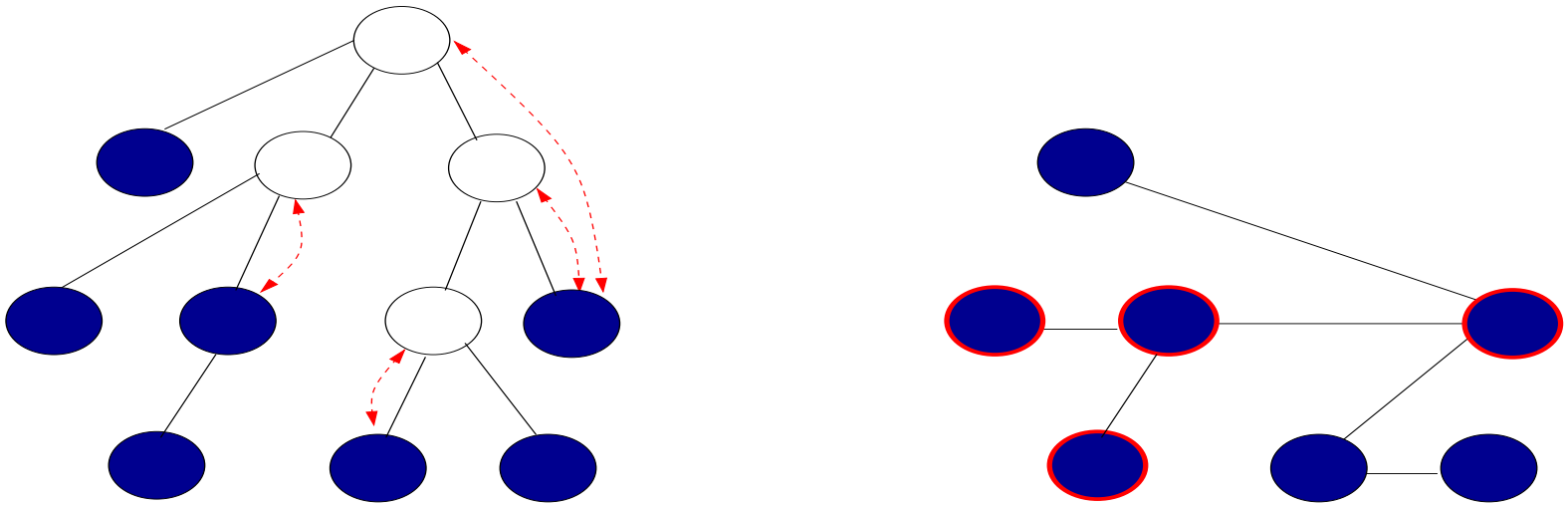
Chow-Liu Grouping for General Latent Trees

- Initialize tree estimate as Chow-Liu tree: $\widehat{T} \leftarrow \widehat{T}_{\mathrm{CL}}$
- Pick an internal node i in Chow-Liu tree $\widehat{T}_{\mathrm{CL}}$ not visited before, Recursive grouping over closed neighborhood $\mathrm{nbd}[i; \widehat{T}]$
- In $\widehat{T}$, replace subtree over $\mathrm{nbd}[i; \widehat{T}]$ with output of recursive grouping



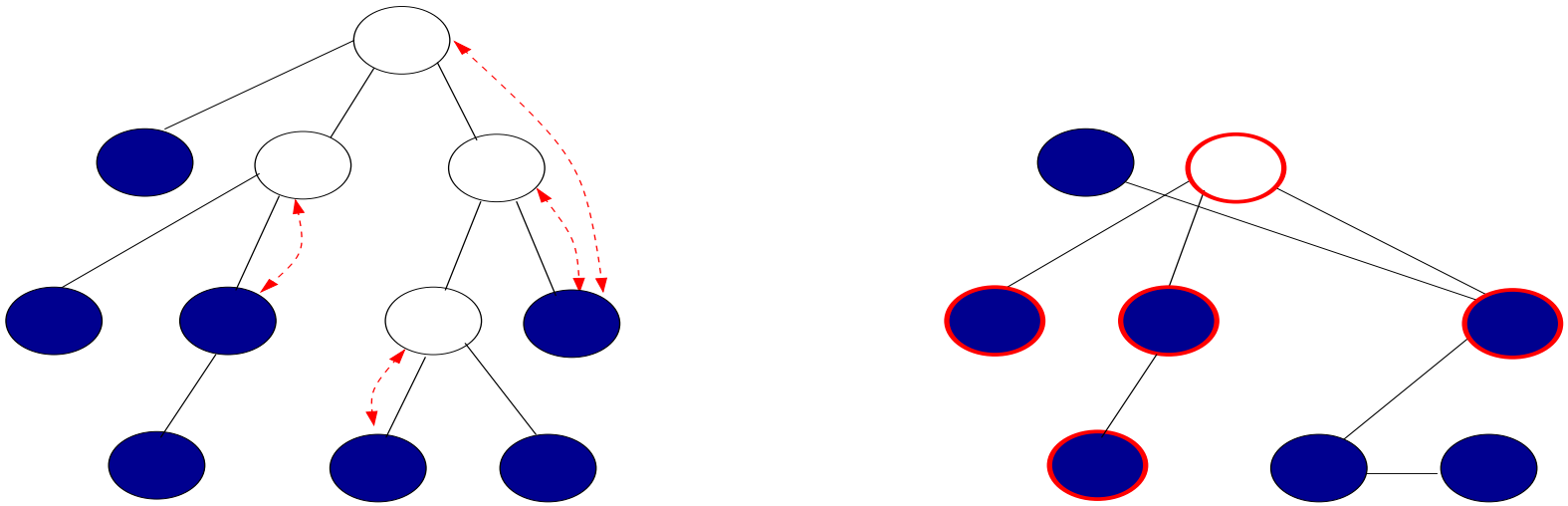
Chow-Liu Grouping for General Latent Trees

- Initialize tree estimate as Chow-Liu tree: $\widehat{T} \leftarrow \widehat{T}_{\mathrm{CL}}$
- Pick an internal node i in Chow-Liu tree $\widehat{T}_{\mathrm{CL}}$ not visited before, Recursive grouping over closed neighborhood $\mathrm{nbd}[i; \widehat{T}]$
- In $\widehat{T}$, replace subtree over $\mathrm{nbd}[i; \widehat{T}]$ with output of recursive grouping



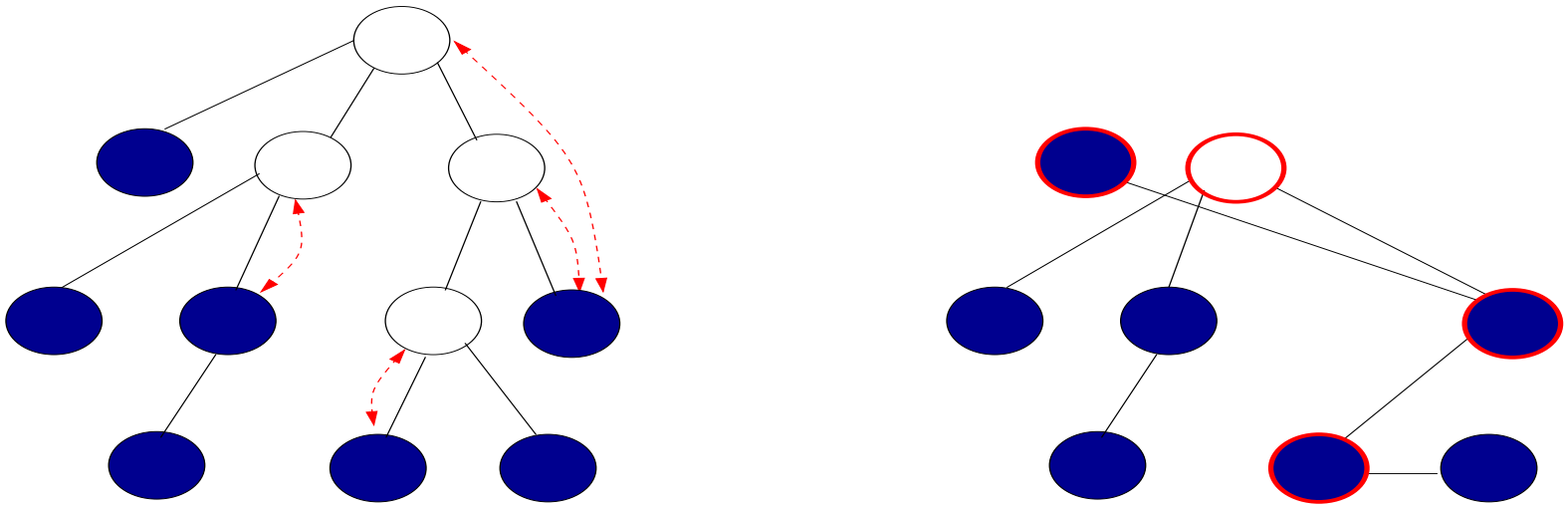
Chow-Liu Grouping for General Latent Trees

- Initialize tree estimate as Chow-Liu tree: $\widehat{T} \leftarrow \widehat{T}_{\mathrm{CL}}$
- Pick an internal node i in Chow-Liu tree $\widehat{T}_{\mathrm{CL}}$ not visited before, Recursive grouping over closed neighborhood $\mathrm{nbd}[i; \widehat{T}]$
- In $\widehat{T}$, replace subtree over $\mathrm{nbd}[i; \widehat{T}]$ with output of recursive grouping



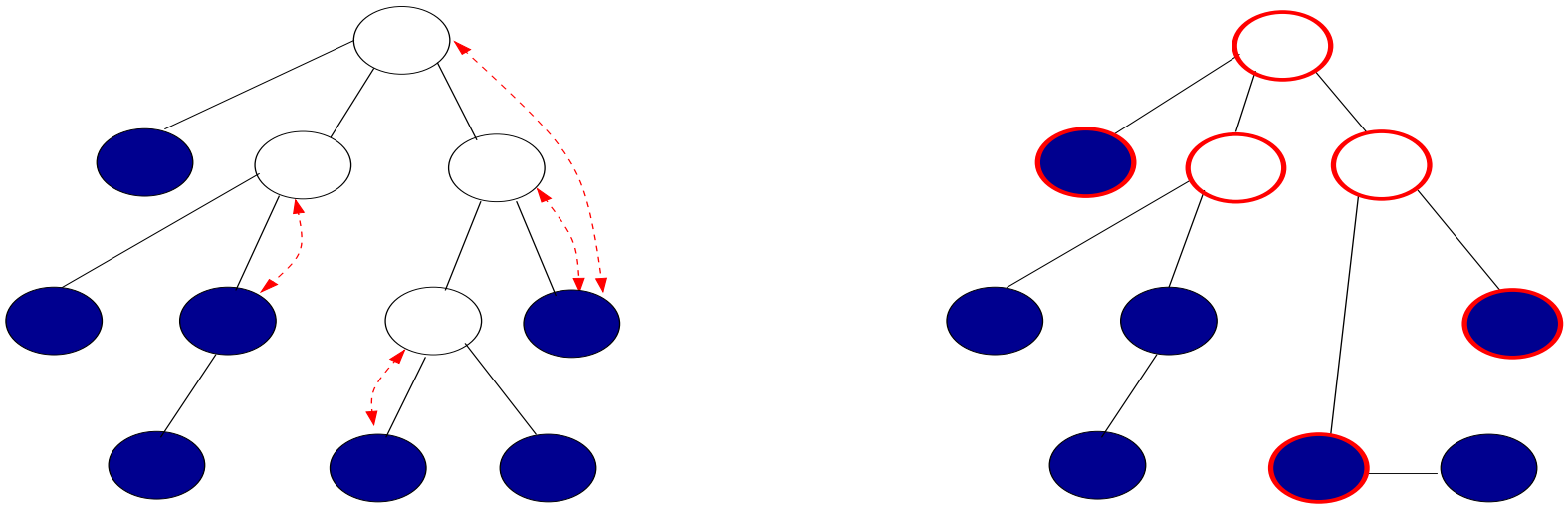
Chow-Liu Grouping for General Latent Trees

- Initialize tree estimate as Chow-Liu tree: $\widehat{T} \leftarrow \widehat{T}_{\mathrm{CL}}$
- Pick an internal node i in Chow-Liu tree $\widehat{T}_{\mathrm{CL}}$ not visited before, Recursive grouping over closed neighborhood $\mathrm{nbd}[i; \widehat{T}]$
- In $\widehat{T}$, replace subtree over $\mathrm{nbd}[i; \widehat{T}]$ with output of recursive grouping



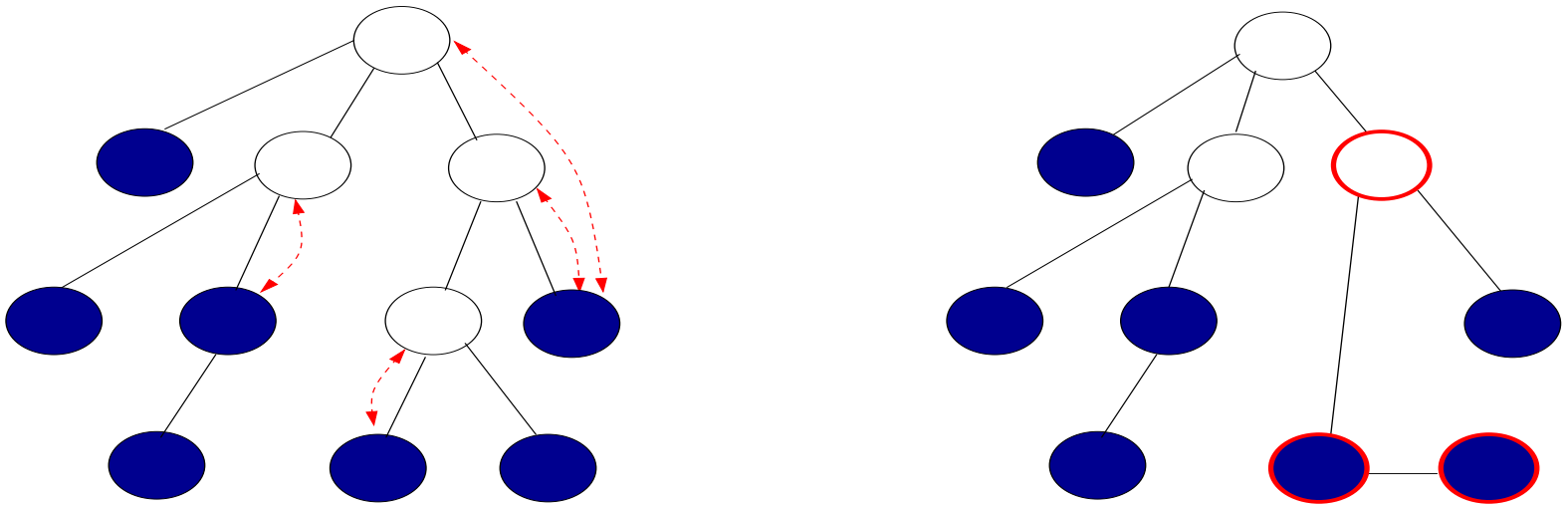
Chow-Liu Grouping for General Latent Trees

- Initialize tree estimate as Chow-Liu tree: $\widehat{T} \leftarrow \widehat{T}_{\mathrm{CL}}$
- Pick an internal node i in Chow-Liu tree $\widehat{T}_{\mathrm{CL}}$ not visited before, Recursive grouping over closed neighborhood $\mathrm{nbd}[i; \widehat{T}]$
- In $\widehat{T}$, replace subtree over $\mathrm{nbd}[i; \widehat{T}]$ with output of recursive grouping



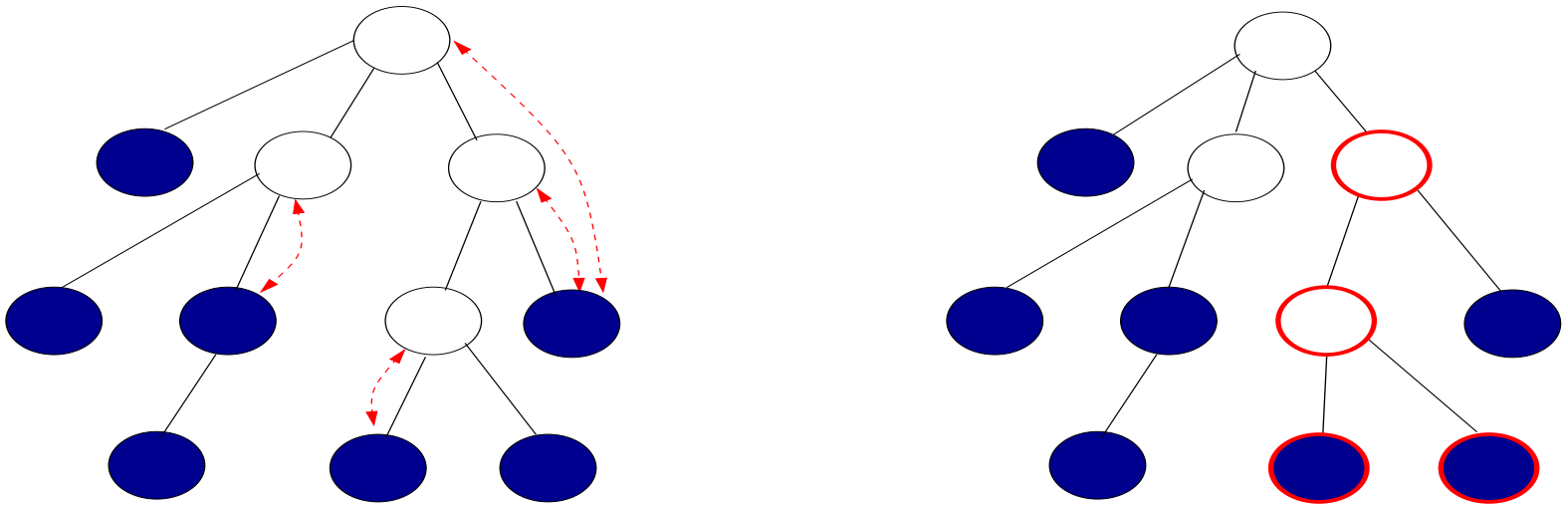
Chow-Liu Grouping for General Latent Trees

- Initialize tree estimate as Chow-Liu tree: $\widehat{T} \leftarrow \widehat{T}_{\mathrm{CL}}$
- Pick an internal node i in Chow-Liu tree $\widehat{T}_{\mathrm{CL}}$ not visited before, Recursive grouping over closed neighborhood $\mathrm{nbd}[i; \widehat{T}]$
- In $\widehat{T}$, replace subtree over $\mathrm{nbd}[i; \widehat{T}]$ with output of recursive grouping



Chow-Liu Grouping for General Latent Trees

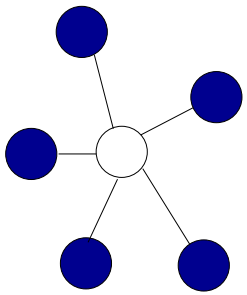
- Initialize tree estimate as Chow-Liu tree: $\widehat{T} \leftarrow \widehat{T}_{\mathrm{CL}}$
- Pick an internal node i in Chow-Liu tree $\widehat{T}_{\mathrm{CL}}$ not visited before, Recursive grouping over closed neighborhood $\mathrm{nbd}[i; \widehat{T}]$
- In $\widehat{T}$, replace subtree over $\mathrm{nbd}[i; \widehat{T}]$ with output of recursive grouping



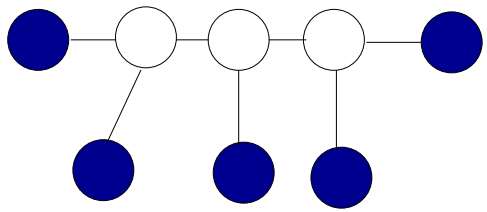
Guarantees for Chow-Liu Grouping

- Structural and estimation consistency for all minimal latent trees
- Sample complexity of $O(\log m)$ for m observed nodes when effective depth is constant
- Computational complexity of $O(m^2 \log m + (\text{No. of internal nodes in CL-tree}) \times (\text{Max. Deg})^3$.

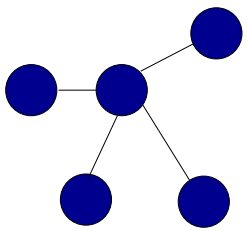
Star: Latent Tree



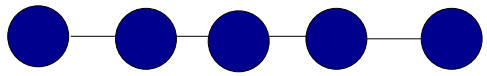
HMM: Latent Tree



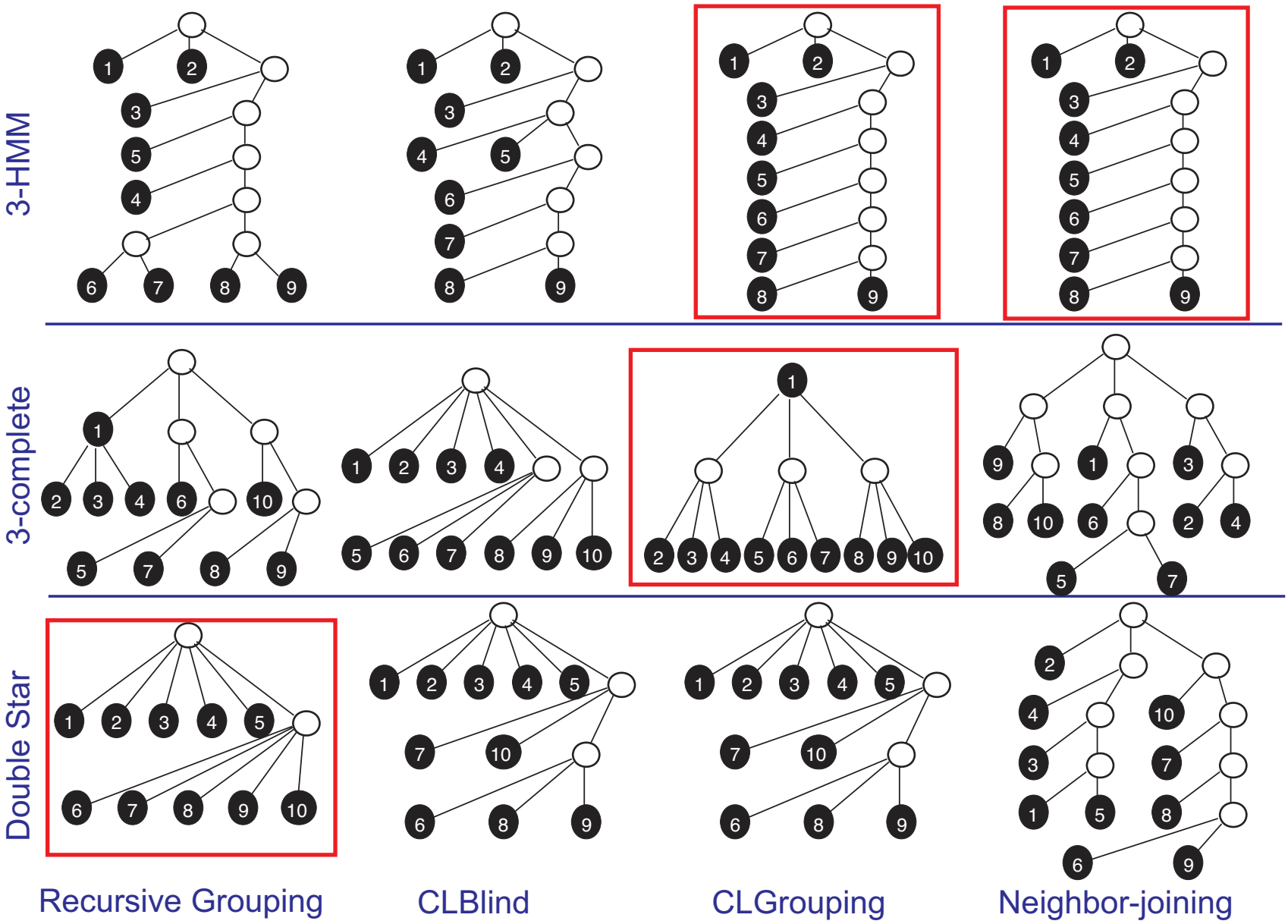
Chow-Liu Tree



Chow-Liu Tree

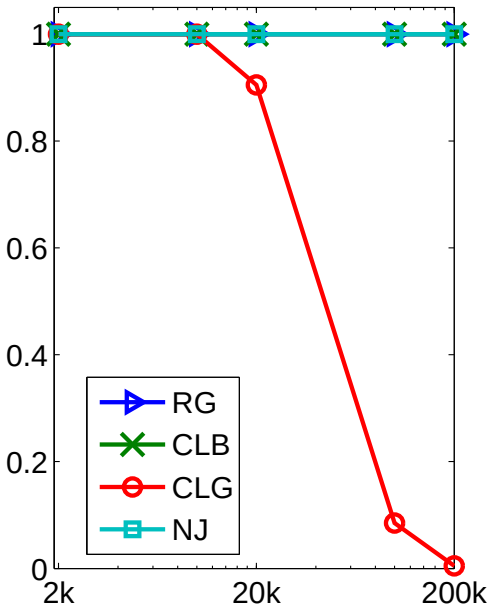


Example Results on Structure Recovery

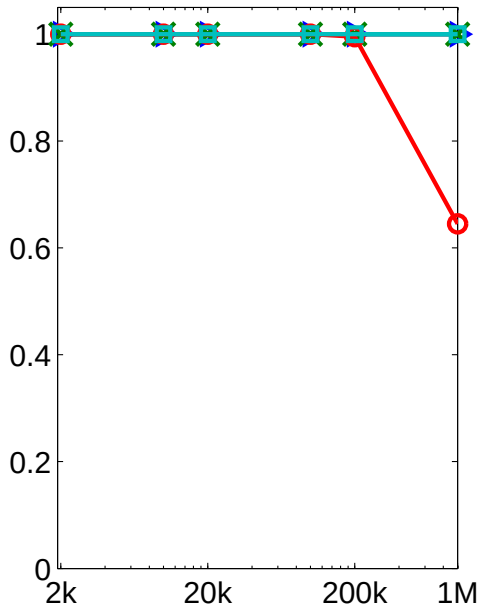


Results: Structure $0-1$ Recovery Error

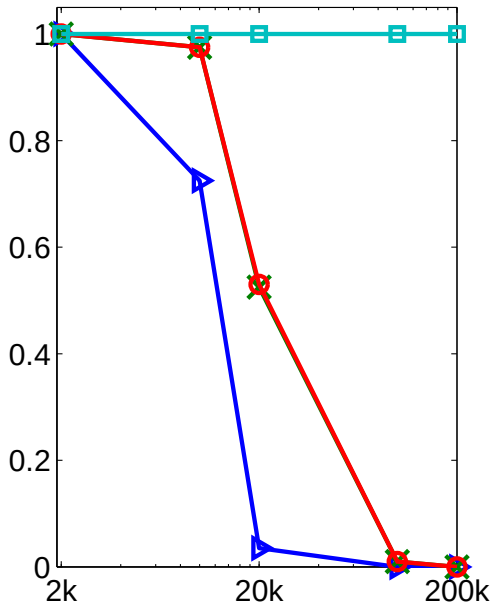
Structure Recovery Error vs. No. of Samples



3-HMM



3-complete



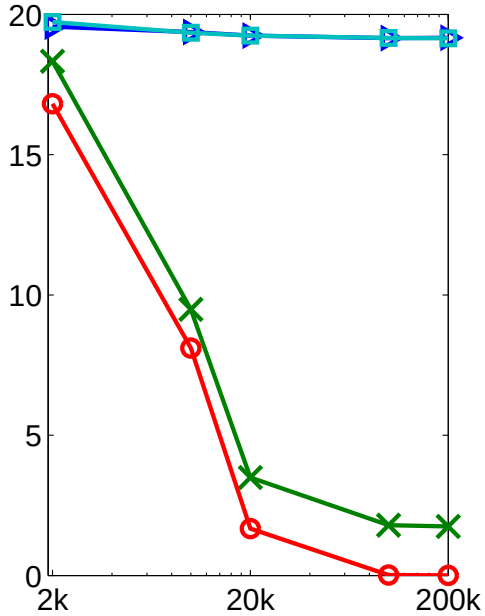
Double Star

RG: Recursive Grouping, CLB: Chow-Liu Blind,
CLG: Chow-Liu Grouping, NJ: Neighbor Joining

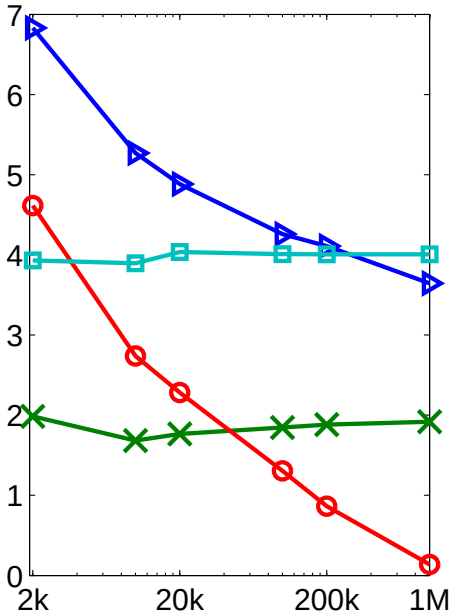
81 Observed Nodes, 200 runs, Gaussian model $\rho_{i,j} \sim \text{Unif}[0.2, 0.8]$ for $(i,j) \in E$.

Results: Avg. Distance Error in Structure Recovery

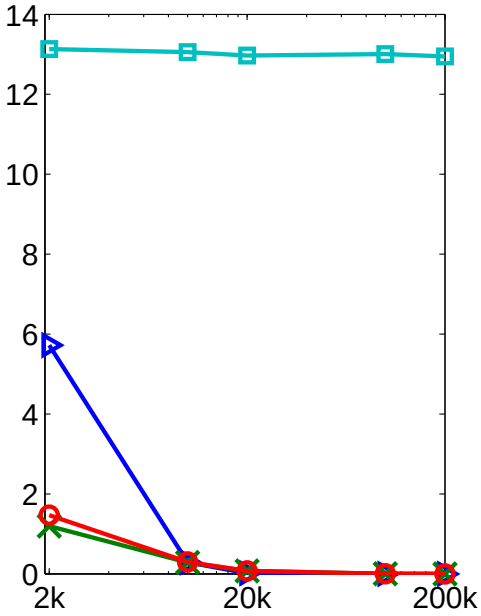
Average Distance Error vs. No. of Samples



3-HMM



3-complete



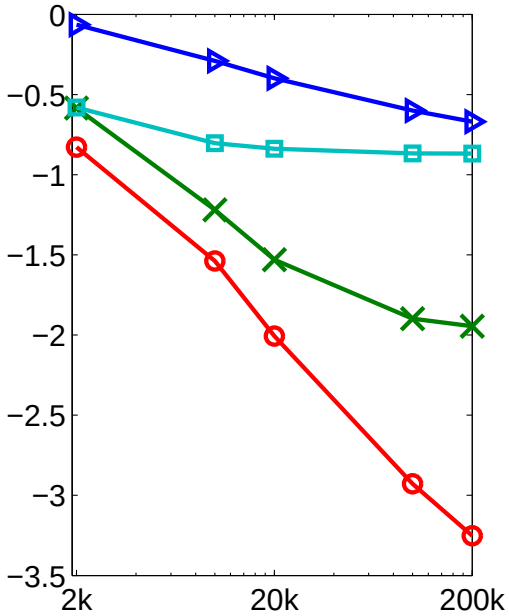
Double Star

RG: Recursive Grouping, CLB: Chow-Liu Blind,
CLG: Chow-Liu Grouping, NJ: Neighbor Joining

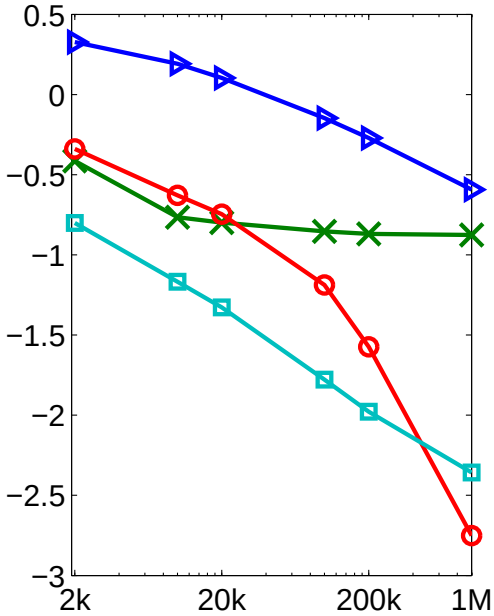
81 Observed Nodes, 200 runs, Gaussian model $\rho_{i,j} \sim \text{Unif}[0.2, 0.8]$ for $(i,j) \in E$.

Results: KL-Divergence Fitting

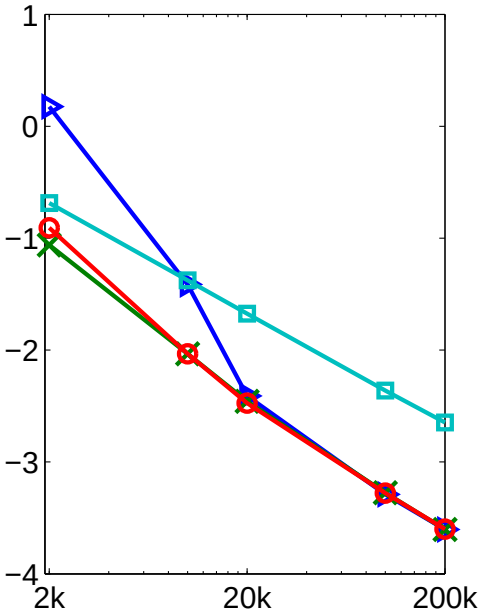
KL-Divergence vs. No. of Samples



3-HMM



3-complete



Double Star

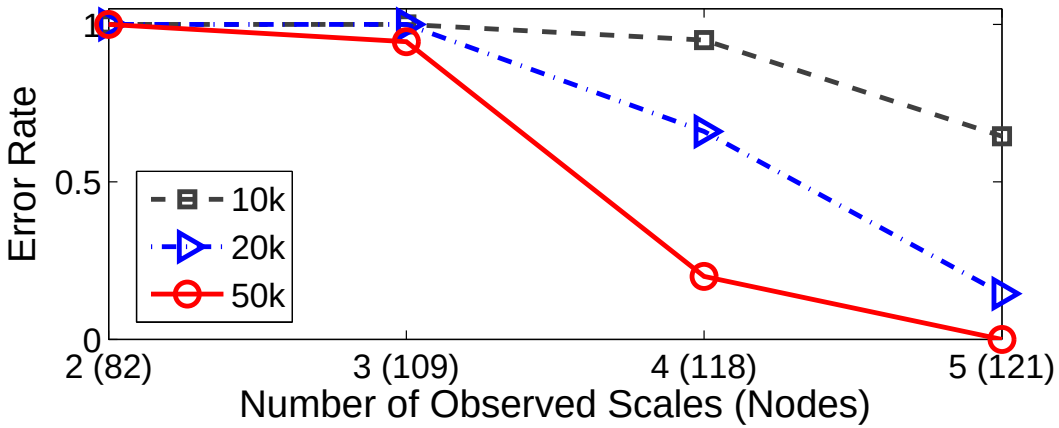
RG: Recursive Grouping, CLB: Chow-Liu Blind,
CLG: Chow-Liu Grouping, NJ: Neighbor Joining

81 Observed Nodes, 200 runs, Gaussian model $\rho_{i,j} \sim \text{Unif}[0.2, 0.8]$ for $(i,j) \in E$.

Results: Running Times of Algorithms

	RG	CLB	CLG	NJ
3-HMM	1.03	0.05	0.08	0.02
3-complete	1.02	0.06	0.08	0.02
Double star	0.37	0.23	0.33	0.03

Structure Error vs. No. of Observed Nodes for CL Grouping



3-complete tree

Outline

Conclusion

Error Exponent for Learning Tree Models

- Completely characterized the error exponent for learning in discrete and Gaussian graphical models.
- Used Euclidean Information Theory to simplify exponent for analysis in the very noisy learning regime.
- Extremal structures for learning: star (hardest) and chain (easiest).

Learning of Latent Tree Models

- Proposed two novel algorithms under unified approach for Gaussian and discrete latent tree models
- Consistency, computational and sample complexities

  Structural and estimation consistency for any minimal latent tree
  Sample complexity of $O(\log m)$ for m observed nodes for fixed depth
  Low computational complexity

---

www.mit.edu/∼animakum