High-Dimensional Latent Graphical Models: Girth-Constrained Graph Families

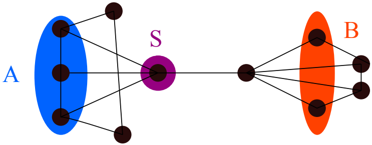
Anima Anandkumar

U.C. Irvine

Graphical Models: Definition

Conditional Independence

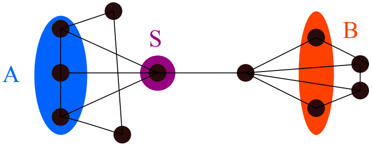
$\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_S$



Graphical Models: Definition

Conditional Independence

$\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_S$



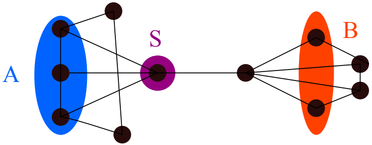
Factorization

$$P(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} \Psi_{i,j}(x_i, x_j)\right].$$

Graphical Models: Definition

Conditional Independence

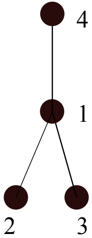
$\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_S$



Factorization

$$P(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} \Psi_{i,j}(x_i, x_j)\right].$$

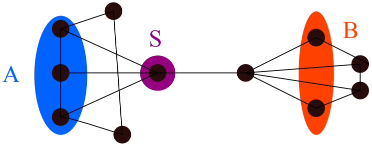
Tree-Structured Graphical Models



Graphical Models: Definition

Conditional Independence

$\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_S$



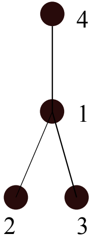
Factorization

$$P(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} \Psi_{i,j}(x_i, x_j)\right].$$

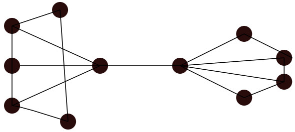
Tree-Structured Graphical Models

$$P(\mathbf{x}) = \prod_{i\in V} P_i(x_i) \prod_{(i,j)\in E} \frac{P_{i,j}(x_i, x_j)}{P_i(x_i)P_j(x_j)}$$

$$= P_1(x_1)P_{2|1}(x_2|x_1)P_{3|1}(x_3|x_1)P_{4|1}(x_4|x_1).$$



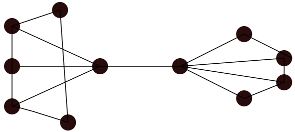
Structure Learning in Latent Graphical Models



- Graphical model on m nodes
- n i.i.d. samples from multivariate distribution of p observed nodes
- Estimate $\widehat{G}^n$ over m nodes

Structural Consistency: $\lim_{n\to\infty} P\left[\widehat{G}^n \neq G\right] = 0.$

Structure Learning in Latent Graphical Models



- Graphical model on m nodes
- n i.i.d. samples from multivariate distribution of p observed nodes
- Estimate $\widehat{G}^n$ over m nodes

Structural Consistency: $\lim_{n\to\infty} P\left[\widehat{G}^n \neq G\right] = 0.$

Challenge: High Dimensionality ("Data-Poor" Regime)

- Large m, small n regime $(m \gg n)$
- Sample Complexity: Required # of samples to achieve consistency
- Large m, small p regime $(m \gg p)$
- Manifest Complexity: Fraction of observed nodes (p/m) required

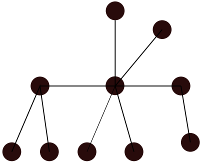
Challenge: Computational Complexity

Goal: Address above challenges and provide provable guarantees

Tree Graphical Models: Tractable Learning

Maximum Likelihood Learning in Fully Observed Setting $(m = p)$

- Proposed by Chow and Liu (68)
- Max. weight spanning tree

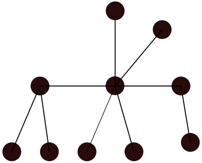


$$\hat{T}_{\text{ML}} = \arg\max_{T} \sum_{k=1}^{n} \log P(\mathbf{x}_V).$$

Tree Graphical Models: Tractable Learning

Maximum Likelihood Learning in Fully Observed Setting $(m = p)$

- Proposed by Chow and Liu (68)
- Max. weight spanning tree

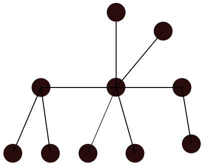


$$\hat{T}_{\mathrm{ML}} = \arg\max_{T} \sum_{k=1}^{n} \log P(\mathbf{x}_V).$$

$$\hat{T}_{\mathrm{ML}} = \arg\max_{T} \sum_{(i,j)\in T} \widehat{I}^{n}(X_i; X_j).$$

Tree Graphical Models: Tractable Learning

Maximum Likelihood Learning in Fully Observed Setting $(m = p)$



- Proposed by Chow and Liu (68)
- Max. weight spanning tree

$$\hat{T}_{\mathrm{ML}} = \arg\max_{T} \sum_{k=1}^{n} \log P(\mathbf{x}_V).$$

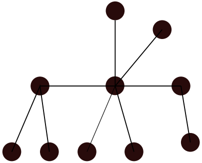
$$\hat{T}_{\mathrm{ML}} = \arg\max_{T} \sum_{(i,j)\in T} \widehat{I}^n(X_i; X_j).$$

- Pairwise statistics suffice for ML

Tree Graphical Models: Tractable Learning

Maximum Likelihood Learning in Fully Observed Setting $(m = p)$

- Proposed by Chow and Liu (68)
- Max. weight spanning tree



$$\hat{T}_{\mathrm{ML}} = \arg\max_{T} \sum_{k=1}^{n} \log P(\mathbf{x}_V).$$

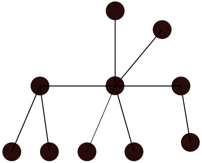
$$\hat{T}_{\mathrm{ML}} = \arg\max_{T} \sum_{(i,j)\in T} \widehat{I}^n(X_i; X_j).$$

- Pairwise statistics suffice for ML
- n samples and $m = p$ nodes: Sample complexity: $\dfrac{\log p}{n} = O(1)$.

Tree Graphical Models: Tractable Learning

Maximum Likelihood Learning in Fully Observed Setting $(m = p)$

- Proposed by Chow and Liu (68)
- Max. weight spanning tree



$$\hat{T}_{\mathrm{ML}} = \arg\max_{T} \sum_{k=1}^{n} \log P(\mathbf{x}_V).$$

$$\hat{T}_{\mathrm{ML}} = \arg\max_{T} \sum_{(i,j)\in T} \widehat{I}^n(X_i; X_j).$$

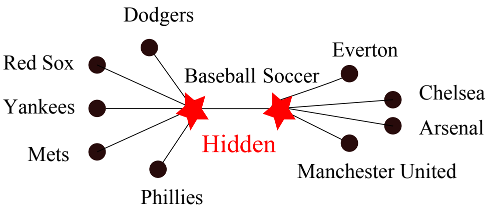
- Pairwise statistics suffice for ML
- n samples and $m = p$ nodes: Sample complexity: $\dfrac{\log p}{n} = O(1)$.

What other classes of graphical models are tractable for learning?

Learning Latent Tree Models

Latent Trees

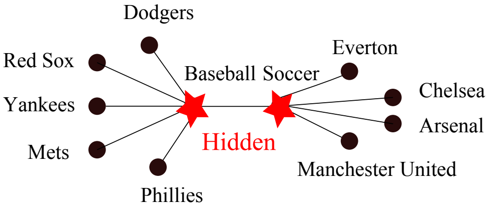
- Richer model class
- Detect hidden influences
- Hierarchical Representation



Learning Latent Tree Models

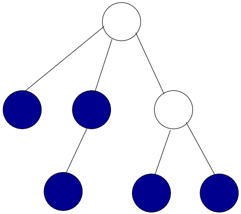
Latent Trees

- Richer model class
- Detect hidden influences
- Hierarchical Representation



Setup

- n samples at observed nodes
- No knowledge about hidden nodes

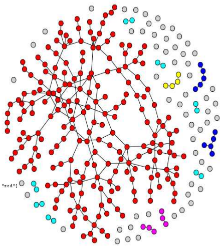


Learning Latent Graphical Models Beyond Trees

Challenges

- Presence of cycles
  - Computing partition function is $\#P$-complete: maximum likelihood is not tractable

$$P(\mathbf{x}) = \frac{1}{Z} \exp \left[\sum_{(i,j) \in G} \Psi_{i,j}(x_i, x_j) \right].$$



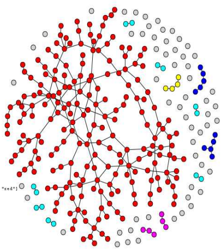
Learning Latent Graphical Models Beyond Trees

Challenges

- Presence of cycles
  - Computing partition function is $\#P$-complete: maximum likelihood is not tractable

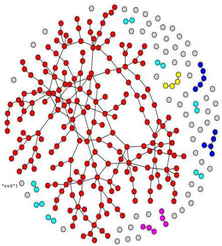
$$P(\mathbf{x}) = \frac{1}{Z} \exp\left[\sum_{(i,j)\in G} \Psi_{i,j}(x_i, x_j)\right].$$

- Presence of hidden variables
  - Model may not be identifiable



Learning Latent Graphical Models Beyond Trees

Challenges



- Presence of cycles
  - Computing partition function is $\#P$-complete: maximum likelihood is not tractable

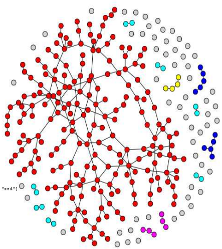
$$P(\mathbf{x}) = \frac{1}{Z} \exp\left[\sum_{(i,j)\in G} \Psi_{i,j}(x_i, x_j) \right].$$

- Presence of hidden variables
  - Model may not be identifiable

Can we provide learning guarantees under above conditions?

Learning Latent Graphical Models Beyond Trees

Challenges



- Presence of cycles
  - Computing partition function is $\#P$-complete: maximum likelihood is not tractable

$$P(\mathbf{x}) = \frac{1}{Z} \exp\left[\sum_{(i,j)\in G} \Psi_{i,j}(x_i, x_j)\right].$$

- Presence of hidden variables
  - Model may not be identifiable

Can we provide learning guarantees under above conditions?

In this talk: Latent Models on Large Girth Graphs

- Girth is the length of the shortest cycle
- Relaxation of the tree model
- Relevant for real datasets, e.g., topic-word models

Summary of Results: Structure Estimation in Latent Graphical Models

- Characterize conditions for (asymptotic) identifiability
- Propose efficient algorithm for structure estimation
- Establish sample complexity for structure consistency under PAC model under transparent sufficient conditions
- Obtain necessary conditions for structure estimation
- Experimental validation using newsgroups dataset

A. Anandkumar, "Learning High-Dimensional Latent Graphical Models: Girth-Constrained Graph Families," available on webpage.

Related Work in Structure Learning

Structure Learning For Fully Observed Models

- Chow and Liu (68)
- Bresler, Mossel and Sly (09)
- Ravikumar, Wainwright and Lafferty (10)
- A. Anandkumar, Tan, Willsky, (11) ...

Learning with Hidden Variables

- Daskalakis, Mossel and Roch (06)
- Choi, Tan, A. Anandkumar, Willsky, (10)
- Chandrasekaran, Parrilo and Willsky (10), ...

Approaches Employed

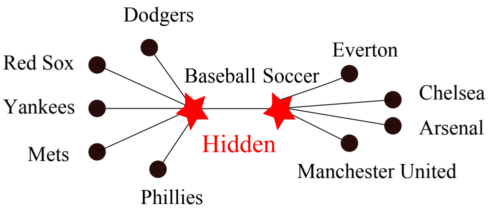
- EM/Search approaches
- Combinatorial/Greedy approach
- Convex relaxation, ...

Outline

Learning Latent Tree Models

Latent Trees

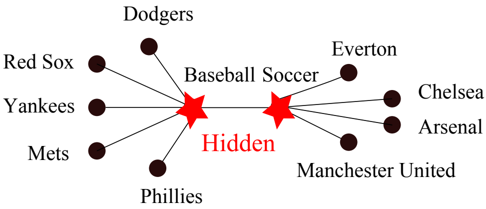
- Richer model class
- Detect hidden influences
- Hierarchical Representation



Learning Latent Tree Models

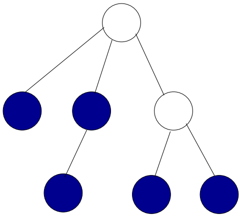
Latent Trees

- Richer model class
- Detect hidden influences
- Hierarchical Representation



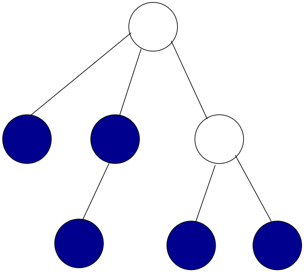
Setup

- n samples at observed nodes
- No knowledge about hidden nodes



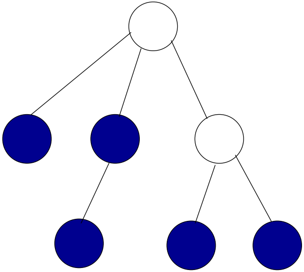
Information Distances $[d_{i,j}]$ on Tree Models

Learning latent tree using pairwise statistics



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

Learning latent tree using pairwise statistics



Discrete Model

$d_{ij} := -\log|\,\mathrm{Det}(P_{i,j})|$.

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

Learning latent tree using pairwise statistics

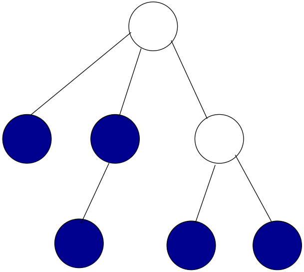
Discrete Model

$d_{ij} := -\log|\operatorname{Det}(P_{i,j})|.$

$[d_{i,j}]$ is a tree metric

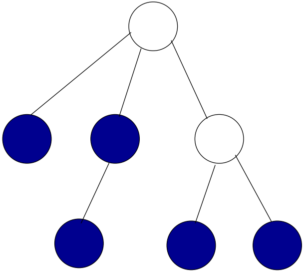
$$d_{k,l} = \sum_{(i,j)\in \mathrm{Path}(k,l;E)} d_{i,j}.$$

Follows from Markov property.



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

Learning latent tree using pairwise statistics



Discrete Model

$d_{ij} := -\log|\operatorname{Det}(P_{i,j})|$.

$[d_{i,j}]$ is a tree metric

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

Follows from Markov property.

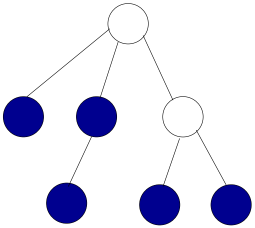
Learning latent tree using $[\hat{d}_{i,j}]$

Sibling Test and Recursive Grouping

Sibling Test (Choi, Tan, A., Willsky)

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

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

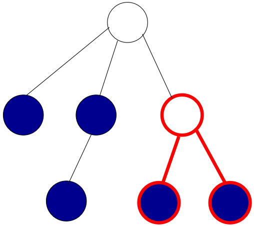


Sibling Test and Recursive Grouping

Sibling Test (Choi, Tan, A., Willsky)

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

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

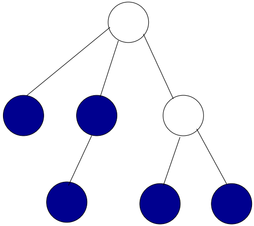


Sibling Test and Recursive Grouping

Sibling Test (Choi, Tan, A., Willsky)

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

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



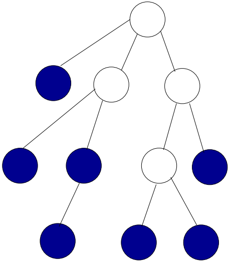
Recursive Grouping Algorithm (Choi, Tan, A., Willsky)

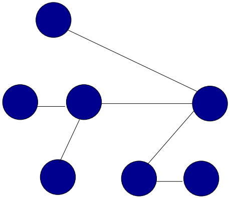
- Sibling test and remove leaves
- Build tree from bottom up

Chow-Liu Based Grouping Algorithm

Optimal Tree on Observed Nodes (Chow and Liu '68)

$$\boxed{\widehat{T}_{\mathrm{ML}} = \operatorname*{argmin}_{T} \sum_{(i,j)\in T} \hat{d}_{i,j}.}$$



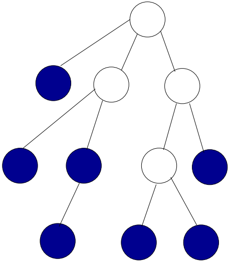


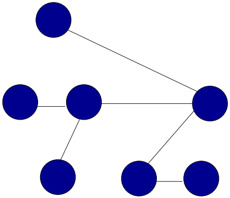
Chow-Liu Based Grouping Algorithm

Optimal Tree on Observed Nodes (Chow and Liu '68)

$$\boxed{\widehat{T}_{\mathrm{ML}} = \operatorname*{argmin}_{T} \sum_{(i,j)\in T} \hat{d}_{i,j}.}$$

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)



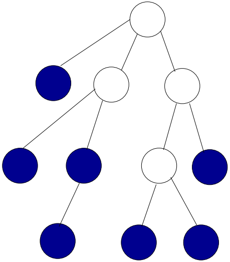


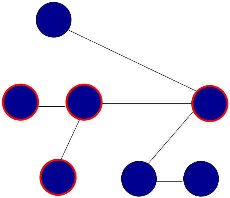
Chow-Liu Based Grouping Algorithm

Optimal Tree on Observed Nodes (Chow and Liu '68)

$$\boxed{\widehat{T}_{\mathrm{ML}} = \operatorname*{argmin}_{T} \sum_{(i,j)\in T} \hat{d}_{i,j}.}$$

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)



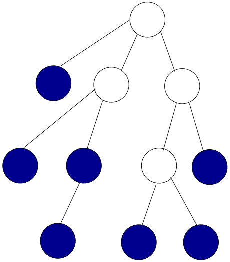


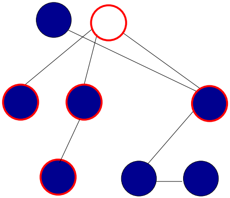
Chow-Liu Based Grouping Algorithm

Optimal Tree on Observed Nodes (Chow and Liu '68)

$$\boxed{\widehat{T}_{\mathrm{ML}} = \operatorname*{argmin}_{T} \sum_{(i,j)\in T} \hat{d}_{i,j}.}$$

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)



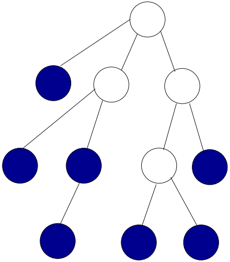


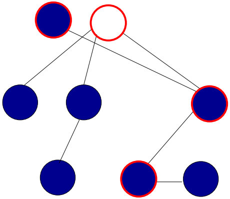
Chow-Liu Based Grouping Algorithm

Optimal Tree on Observed Nodes (Chow and Liu '68)

$$\boxed{\widehat{T}_{\mathrm{ML}} = \operatorname*{argmin}_{T} \sum_{(i,j)\in T} \hat{d}_{i,j}.}$$

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)



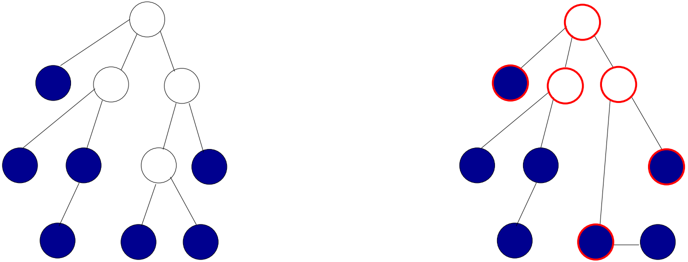


Chow-Liu Based Grouping Algorithm

Optimal Tree on Observed Nodes (Chow and Liu '68)

$$\boxed{\widehat{T}_{\mathrm{ML}} = \operatorname*{argmin}_{T} \sum_{(i,j)\in T} \hat{d}_{i,j}.}$$

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)

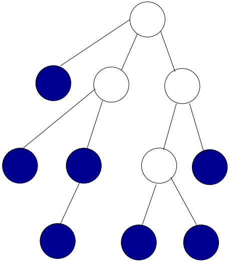


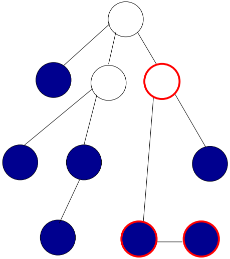
Chow-Liu Based Grouping Algorithm

Optimal Tree on Observed Nodes (Chow and Liu '68)

$$\boxed{\widehat{T}_{\mathrm{ML}} = \operatorname*{argmin}_{T} \sum_{(i,j)\in T} \hat{d}_{i,j}.}$$

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)



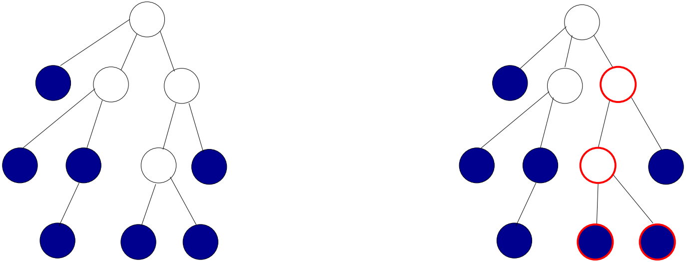


Chow-Liu Based Grouping Algorithm

Optimal Tree on Observed Nodes (Chow and Liu '68)

$$\boxed{\widehat{T}_{\mathrm{ML}} = \operatorname*{argmin}_{T} \sum_{(i,j)\in T} \hat{d}_{i,j}.}$$

Chow-Liu Based Grouping (Choi, Tan, A., Willsky '11)



Guarantees for Chow-Liu Grouping

Theorem

Structural consistency for any minimal latent tree with p nodes, n samples and constant effective depth, satisfying

$$\frac{\log p}{n} = O(1).$$

M.J. Choi, V. Tan, **A. Anandkumar** & A.S. Willsky, "Learning Latent Tree Graphical Models," *J. of Machine Learning Research*, May 2011.

Guarantees for Chow-Liu Grouping

Theorem

Structural consistency for any minimal latent tree with p nodes, n samples and constant effective depth, satisfying

$$\frac{\log p}{n} = O(1).$$

Efficient Implementation of BIC Rule

$$\mathsf{BIC}(\hat{T}) := \log P(\mathbf{x}^n; \hat{T}) - C|H(\hat{T})| \log n.$$

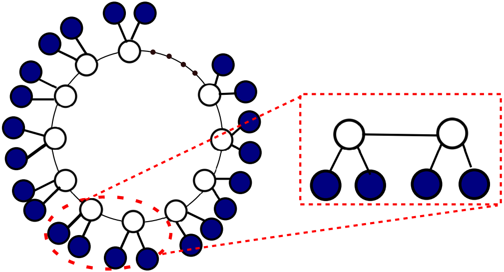
M.J. Choi, V. Tan, **A. Anandkumar** & A.S. Willsky, "Learning Latent Tree Graphical Models," *J. of Machine Learning Research*, May 2011.

Outline

Learning Models on Large Girth Graphs: Intuitions

Large Girth Graphs are Locally Tree Like

Local statistics converge to a tree limit under correlation decay regime as number of nodes $p \to \infty$



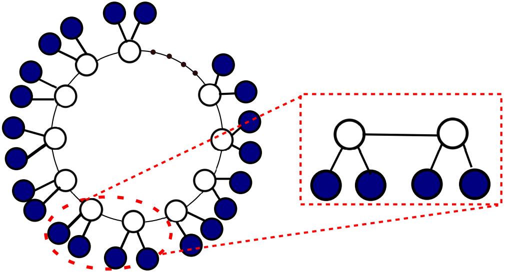
Learning Models on Large Girth Graphs: Intuitions

Large Girth Graphs are Locally Tree Like

Local statistics converge to a tree limit under correlation decay regime as number of nodes $p \to \infty$

Information distance

$d_{ij} := -\log |\operatorname{Det}(P_{i,j})|.$



Learning Models on Large Girth Graphs: Intuitions

Large Girth Graphs are Locally Tree Like

Local statistics converge to a tree limit under correlation decay regime as number of nodes $p \to \infty$

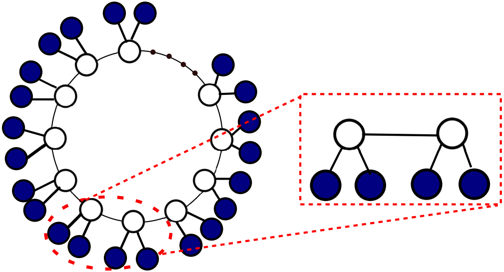
Information distance

$d_{ij} := -\log|\operatorname{Det}(P_{i,j})|.$

Under correlation decay

Local additivity:

$$d_{k,l} \approx \sum_{(i,j)\in \mathrm{Path}(k,l;E)} d_{i,j}.$$



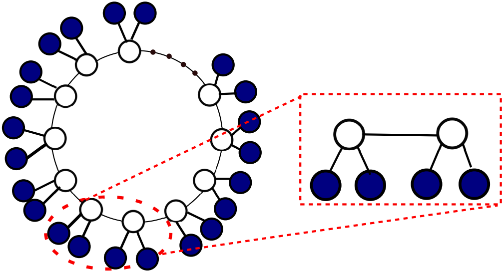
Learning Models on Large Girth Graphs: Intuitions

Large Girth Graphs are Locally Tree Like

Local statistics converge to a tree limit under correlation decay regime as number of nodes $p \to \infty$

Information distance

$d_{ij} := -\log |\operatorname{Det}(P_{i,j})|.$



Under correlation decay

Local additivity:

$$d_{k,l} \approx \sum_{(i,j)\in \mathrm{Path}(k,l;E)} d_{i,j}.$$

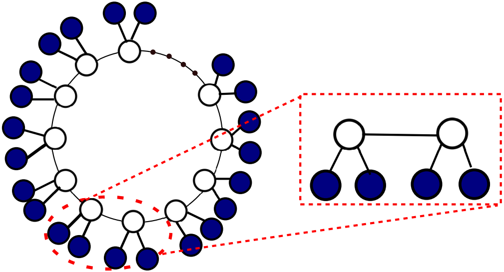
Possible algorithmic approach

- Build local latent trees by considering local groups of variables
- Non-trivial issue: merging of different local latent trees

Learning Models on Large Girth Graphs: Intuitions

Large Girth Graphs are Locally Tree Like

Local statistics converge to a tree limit under correlation decay regime as number of nodes $p \to \infty$



Information distance

$$d_{ij} := -\log |\operatorname{Det}(P_{i,j})|.$$

Under correlation decay

Local additivity:

$$d_{k,l} \approx \sum_{(i,j) \in \mathrm{Path}(k,l;E)} d_{i,j}.$$

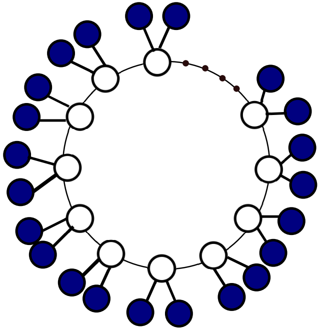
Possible algorithmic approach

- Build local latent trees by considering local groups of variables
- Non-trivial issue: merging of different local latent trees

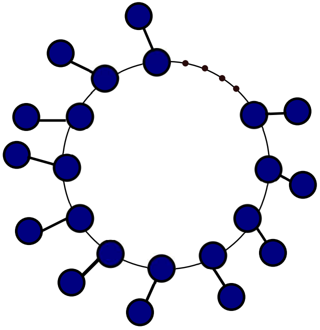
Solution: Adapt Chow-Liu grouping method for learning loopy graphs

Overview of Local Chow-Liu Grouping

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods



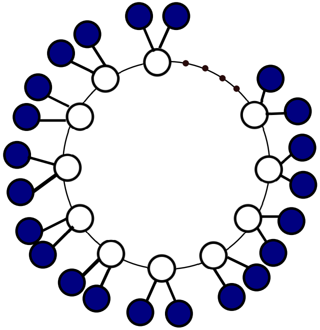
Original Graph



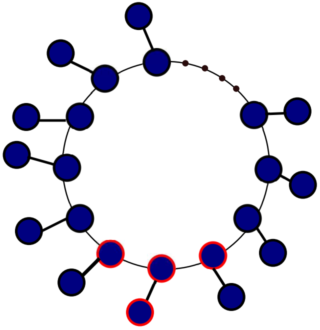
Local CL Grouping

Overview of Local Chow-Liu Grouping

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods



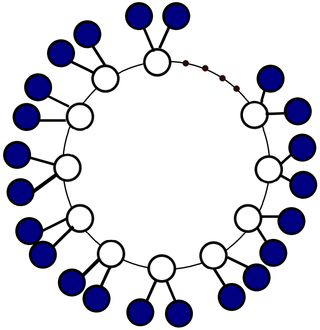
Original Graph



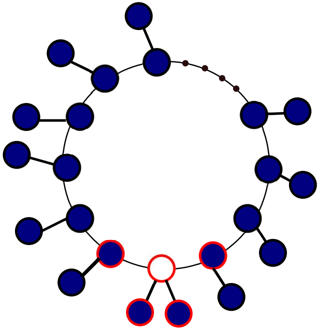
Local CL Grouping

Overview of Local Chow-Liu Grouping

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods



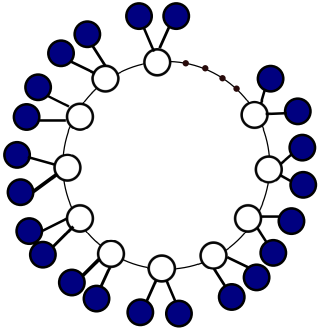
Original Graph



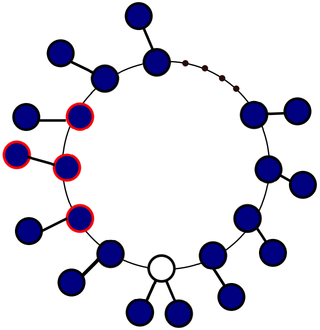
Local CL Grouping

Overview of Local Chow-Liu Grouping

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods



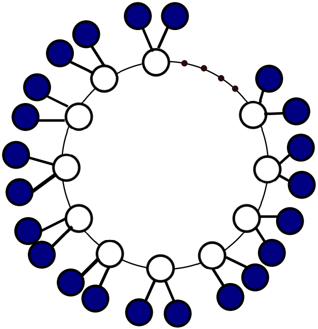
Original Graph



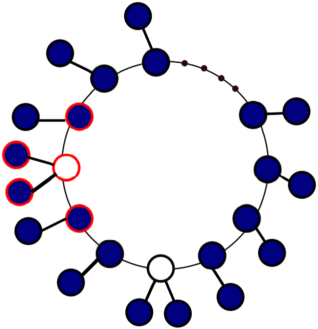
Local CL Grouping

Overview of Local Chow-Liu Grouping

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods



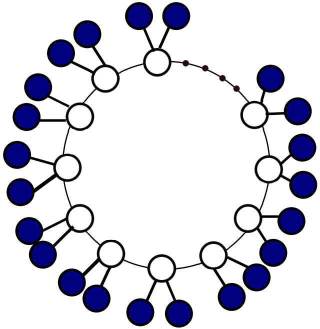
Original Graph



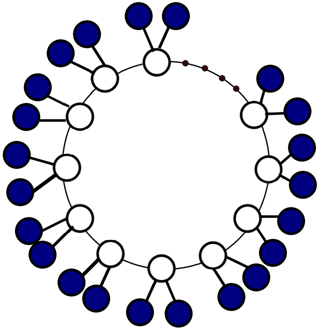
Local CL Grouping

Overview of Local Chow-Liu Grouping

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods



Original Graph



Local CL Grouping

Conditions for Learning Latent Graphical Models

- Underlying graph has girth at least g and maximum degree $\Delta_{\max}$
- Depth δ: worst-case distance between hidden and observed nodes
- Ising model with edge potentials $\{\theta_{i,j}\}$ and node potentials ϕ

$$P(\mathbf{x}) \propto \exp\left[\frac{1}{2}\mathbf{x}^T \boldsymbol{\theta}_G \mathbf{x} + \boldsymbol{\phi}^T \mathbf{x}\right], \quad \mathbf{x} \in \{-1, 1\}^p.$$

- Minimum edge potential $\theta_{\min}$ and maximum edge potential $\theta_{\max}$
- Correlation decay regime: the maximum edge potential satisfies

$$\theta_{\max} < (\Delta_{\max})^{-1}$$

- Parameter η: depends on min. and max. node and edge potentials
  - $\eta = 1$ for homogeneous models

Guarantees for Structure Learning

- Ising model on m nodes, with p nodes observed
- No. of samples n such that

$$\frac{n}{\theta_{\min}^{-2\delta\eta(\eta+1)-2} \log p} = O(1)$$

Theorem

Local Chow-Liu Grouping is structurally consistent

$$\lim_{m,p,n\to\infty} P\left[\widehat{G}_m^n \neq G_m\right] = 0.$$

Guarantees for Structure Learning

- Ising model on m nodes, with p nodes observed
- No. of samples n such that

$$\frac{n}{\theta_{\min}^{-2\delta\eta(\eta+1)-2} \log p} = O(1)$$

Theorem

Local Chow-Liu Grouping is structurally consistent

$$\lim_{m,p,n\to\infty} P\left[\widehat{G}_m^n \neq G_m\right] = 0.$$

Best-case sample complexity

Homogeneous models: $\eta = 1$, Potentials: $\theta_{\min} = \theta_{\max} = \Theta\left(\Delta_{\max}^{-1}\right)$

Sample Complexity scales as $n = \Omega\left(\Delta_{\max}^{4\delta+2} \log p\right)$

Necessary Conditions for Structure Learning

- Graph has minimum degree $\Delta_{\min}$
- Fraction of observed nodes ρ
- Number of observed nodes is p
- Number of samples is n

Theorem

For any deterministic estimator $\widehat{G}_m^n$, for structure consistency, it is necessary that

$$n = \Omega\left(\frac{\Delta_{\min}}{\rho} \log p\right)$$

Necessary Conditions for Structure Learning

- Graph has minimum degree $\Delta_{\min}$
- Fraction of observed nodes ρ
- Number of observed nodes is p
- Number of samples is n

Theorem

For any deterministic estimator $\widehat{G}_m^n$, for structure consistency, it is necessary that

$$n = \Omega\left(\frac{\Delta_{\min}}{\rho}\log p\right)$$

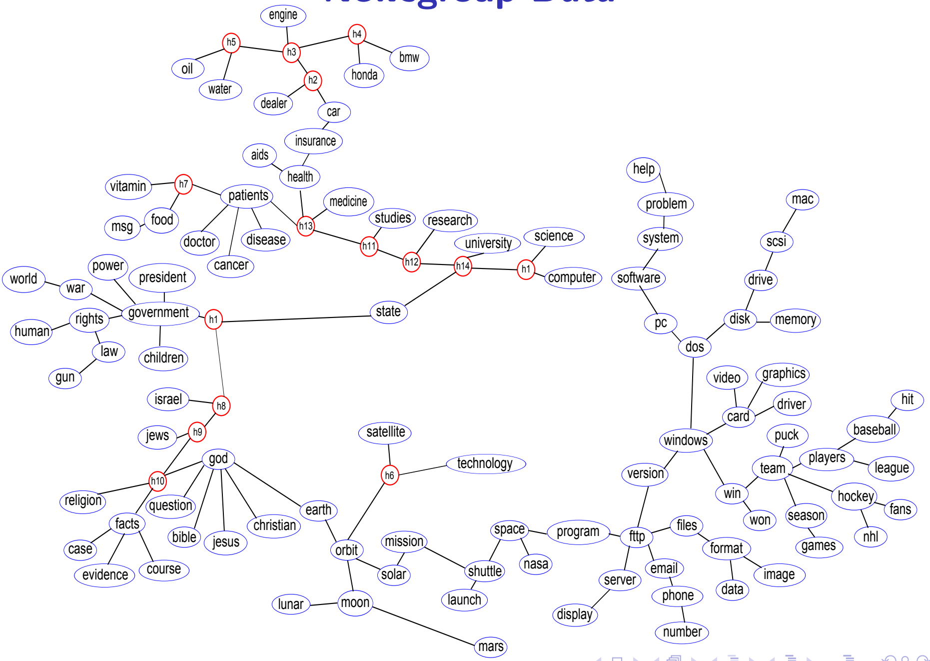
Compare with Local Chow-Liu Grouping

- Homogeneous models $\eta = 1$, Potentials: $\theta_{\min} = \theta_{\max} = \Theta(\Delta_{\max}^{-1})$
- Observed nodes are uniformly sampled

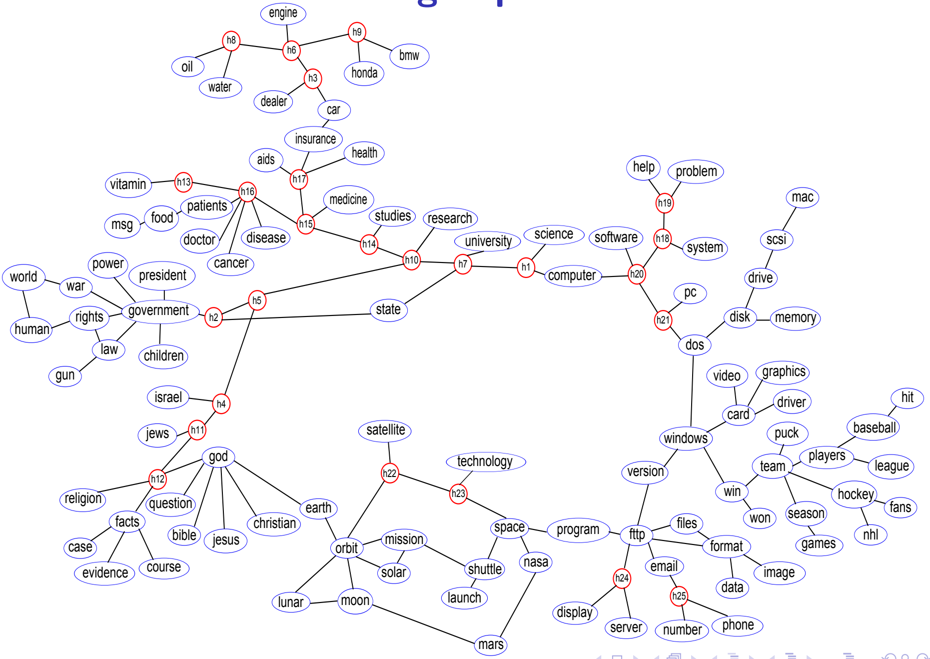
Sample Complexity scales as $n = \Omega\left(\Delta_{\max}^2 \rho^{-4} (\log p)^5\right)$

Outline

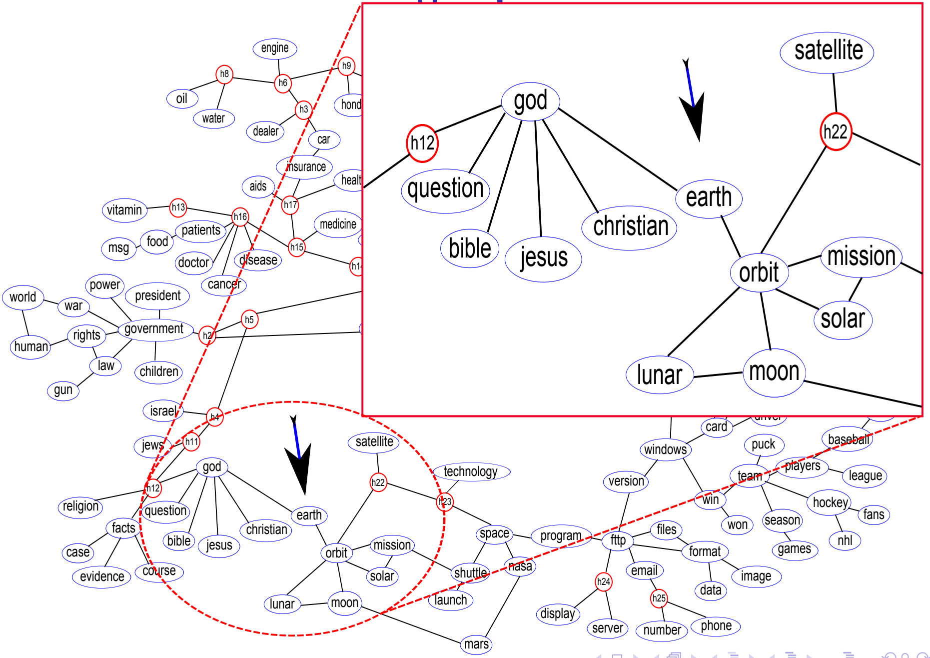
Newsgroup Data



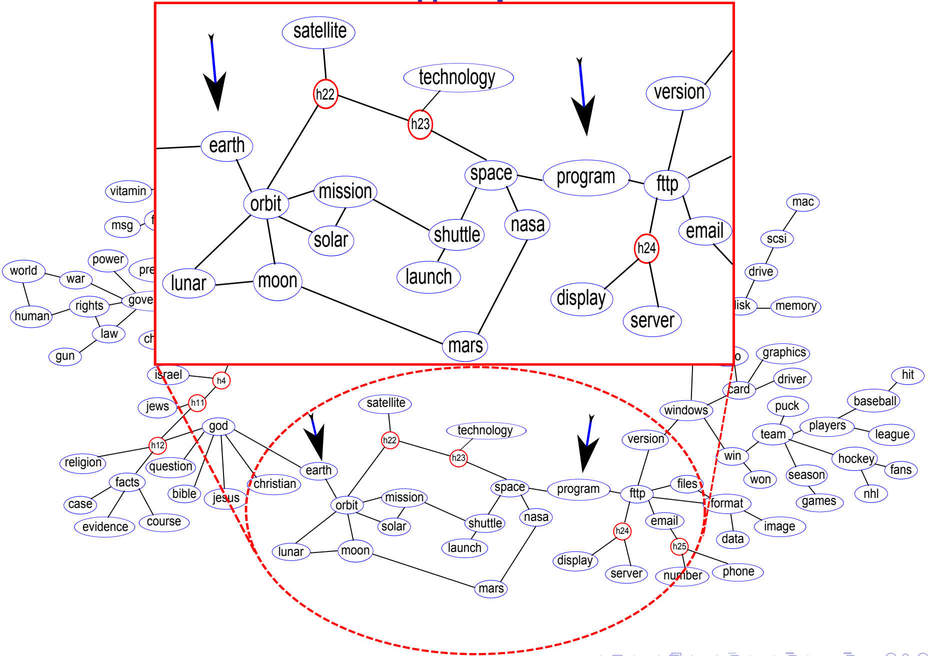
Newsgroup Data



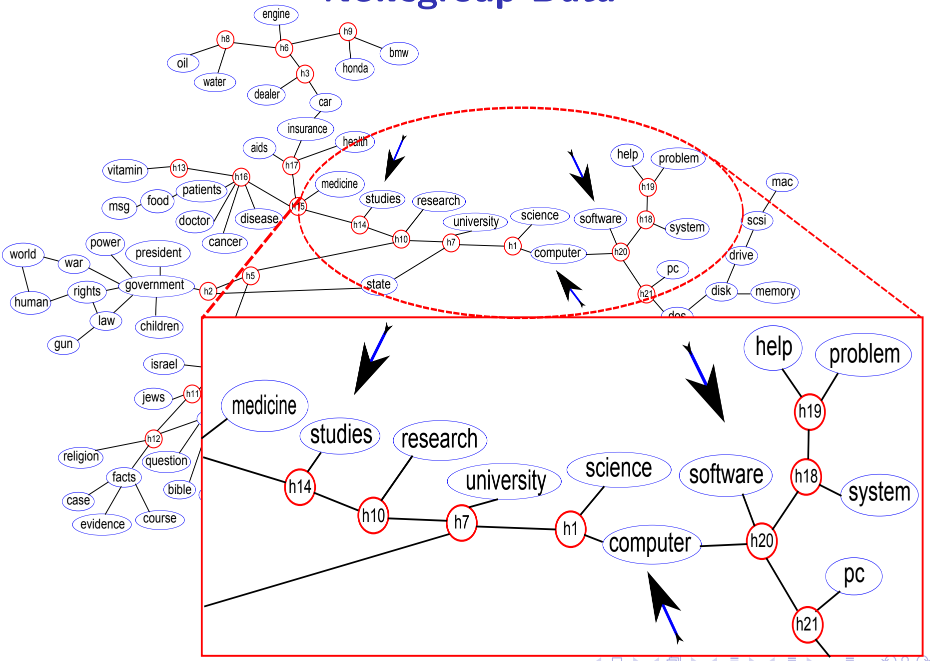
Newsgroup Data



Newsgroup Data



Newsgroup Data



Results on Data sets

20 Newsgroups with 100 words

- 16,242 binary samples of 100 words
- Latent tree learned using regularized Local CL Grouping.

Table: Performance of Local Chow-Liu Grouping on Test Data

Threshold	# Hidden nodes	# Edges	Log-Likelihood	BIC
r=9 (loopy)	25	133	-91973	-93134
r=13 (tree)	15	114	-103772	-104802

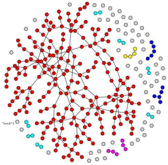
Datasets and code will be soon available

- http://newport.eecs.uci.edu/anandkumar
- Prepared by R. Valluvan, UCI.

Outline

Conclusion

Summary



- Efficient algorithm for learning latent graphical models
- First method to handle discrete latent models on loopy graphs
- Transparent assumptions and efficient sample complexity

Outlook

- Is learning beyond correlation decay regime hard?
- Relaxing the girth constraint
- Methods combining combinatorial and convex approaches

---

http://newport.eecs.uci.edu/anandkumar