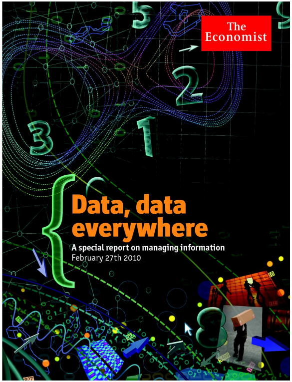
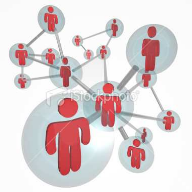
Beyond Sparse Graphical Models: Incorporating Mixtures and Residuals

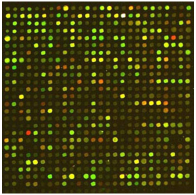
Anima Anandkumar

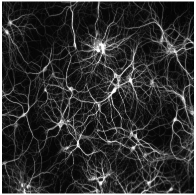
U.C. Irvine

Data Deluge and Data Desert





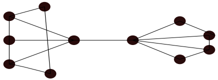




- Current technologies unable to handle data deluge.
- Current algorithms unable to handle data desert.

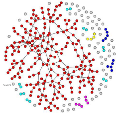
High-dimensional data: Many variables, few samples

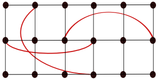
Graph-Based Models for High-Dimensional Data

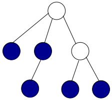


- Qualitative: Graph structure(s).
- Quantitative: Interaction strengths.

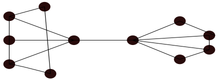
Parsimonious representation via sparse graphs







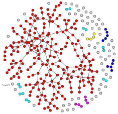
Graph-Based Models for High-Dimensional Data



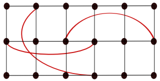
- Qualitative: Graph structure(s).
- Quantitative: Interaction strengths.

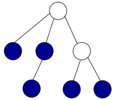
Parsimonious representation via sparse graphs

Steps

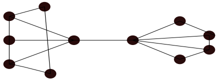


- Estimate structure(s) and parameters from samples.
- Employ model to predict future behavior.





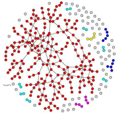
Graph-Based Models for High-Dimensional Data



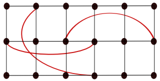
- Qualitative: Graph structure(s).
- Quantitative: Interaction strengths.

Parsimonious representation via sparse graphs

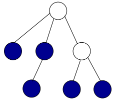
Steps



- Estimate structure(s) and parameters from samples.
- Employ model to predict future behavior.

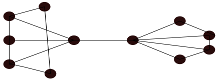


Challenges



- Computational complexity: Large no. of variables.
- Sample complexity: Fewer observations.
- Latent or Hidden Variables: Unobserved influences.
- Parsimony vs. Faithful representation

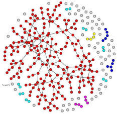
Graph-Based Models for High-Dimensional Data



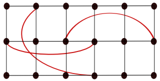
- Qualitative: Graph structure(s).
- Quantitative: Interaction strengths.

Parsimonious representation via sparse graphs

Steps

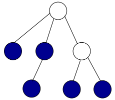


- Estimate structure(s) and parameters from samples.
- Employ model to predict future behavior.



Challenges

- Computational complexity: Large no. of variables.
- Sample complexity: Fewer observations.
- Latent or Hidden Variables: Unobserved influences.
- Parsimony vs. Faithful representation



Goals

Tractable models, Novel algorithms, Provable guarantees, Applications.

Examples of Graph-based Representations

Motivating Example: Topic Modeling

Data: word counts in documents.

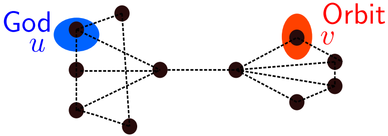
Graph: Topic-word relationships.

Examples of Graph-based Representations

Motivating Example: Topic Modeling

Data: word counts in documents. Graph: Topic-word relationships.

Independence models



Marginal Independence

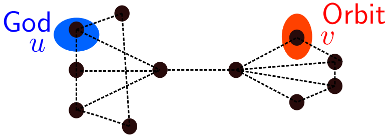
$X_u \perp\!\!\!\perp X_v$

Examples of Graph-based Representations

Motivating Example: Topic Modeling

Data: word counts in documents. Graph: Topic-word relationships.

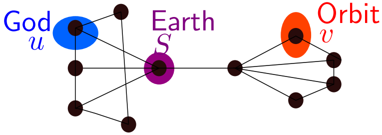
Independence models



Marginal Independence

$X_u \perp\!\!\!\perp X_v$

Markov/graphical models



Conditional Independence

$X_u \perp\!\!\!\perp X_v | X_S$

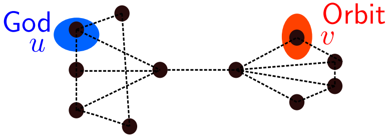
Examples of Graph-based Representations

Motivating Example: Topic Modeling

Data: word counts in documents.

Graph: Topic-word relationships.

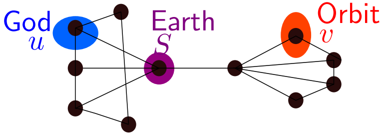
Independence models



Marginal Independence

$X_u \perp\!\!\!\perp X_v$

Markov/graphical models



Conditional Independence

$X_u \perp\!\!\!\perp X_v | X_S$

Shortcoming

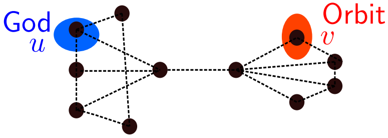
A single independence/Markov graph may not capture all the relationships

Examples of Graph-based Representations

Motivating Example: Topic Modeling

Data: word counts in documents. Graph: Topic-word relationships.

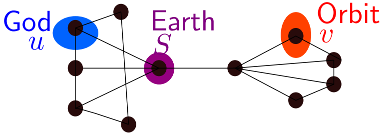
Independence models



Marginal Independence

$X_u \perp\!\!\!\perp X_v$

Markov/graphical models



Conditional Independence

$X_u \perp\!\!\!\perp X_v | X_S$

Shortcoming

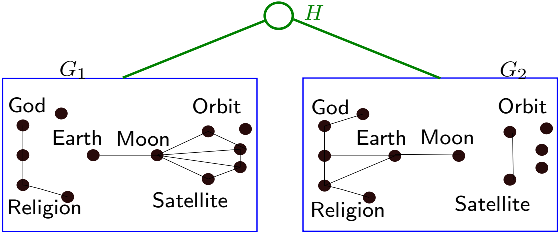
A single independence/Markov graph may not capture all the relationships

Solution: High-dimensional modeling via multiple graphs

High-dimensional Modeling via Multiple Graphs

Graphical Model Mixtures

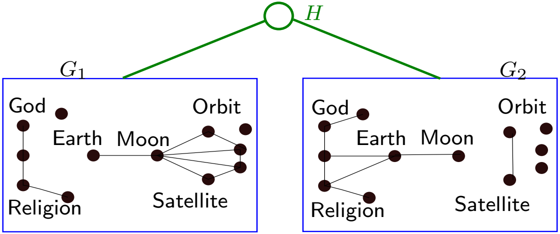
- Multiple graphs: context specific dependencies
- Hidden context
- Learning guarantees



High-dimensional Modeling via Multiple Graphs

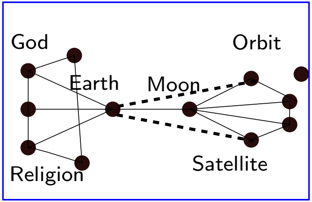
Graphical Model Mixtures

- Multiple graphs: context specific dependencies
- Hidden context
- Learning guarantees



Markov+Independence Models

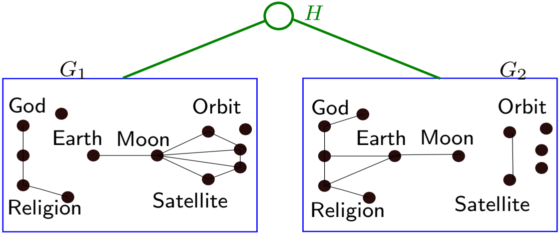
- Multiple graphs: different statistical relationships
- Markov and Independence Graphs
- Efficient decomposition



High-dimensional Modeling via Multiple Graphs

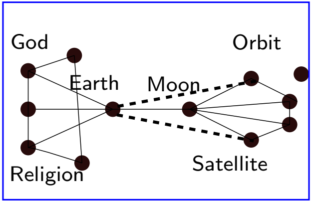
Graphical Model Mixtures

- Multiple graphs: context specific dependencies
- Hidden context
- Learning guarantees



Markov+Independence Models

- Multiple graphs: different statistical relationships
- Markov and Independence Graphs
- Efficient decomposition



Novel Approaches Beyond Sparse Graphical Modeling

State of Art Approaches

Learning Sparse Graphical Models

- Combinatorial: Bresler, Mossel & Sly. **A***, Tan & Willsky.
- Convex: Meinshausen & Bühlmann. Ravikumar, Wainwright & Lafferty.

Learning with Latent Variables

- Trees: Erdös, et. al., Daskalakis, Mossel & Roch. Choi, Tan, **A*** & Willsky.
- Loopy models: Chandrasekaran, Parrilo & Willsky. **A*** & Valluvan.

---

*Special EECS Seminar, March 12, 2012.

State of Art Approaches

Learning Sparse Graphical Models

- Combinatorial: Bresler, Mossel & Sly. **A***, Tan & Willsky.
- Convex: Meinshausen & Bühlmann. Ravikumar, Wainwright & Lafferty.

Learning with Latent Variables

- Trees: Erdös, et. al., Daskalakis, Mossel & Roch. Choi, Tan, **A*** & Willsky.
- Loopy models: Chandrasekaran, Parrilo & Willsky. **A*** & Valluvan.

Learning Mixture Models

- Gaussian Mixtures: Dasgupta. Kannan et al. Chaudhuri et al.
  - Separation condition for mixture components
- Method of Moments: Prony, Belkin & Sinha. Moitra & Valiant.
  - Comp. & sample complexities exponential in no. of components
- Latent Class Models: Chang. Hsu, Kakade & Zhang. Mossel & Roch.
  - Mixtures of discrete product distributions.

---

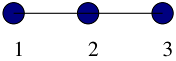
*Special EECS Seminar, March 12, 2012.

Outline

Warm-up: Learning Tree Models

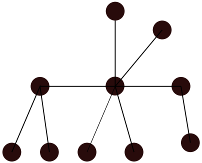
Data processing inequality for Markov chains

$$I(X_1; X_3) \leq I(X_1; X_2), I(X_2; X_3).$$



Tree Structure Estimation (Chow and Liu '68)

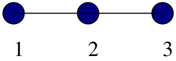
- MLE: Max-weight tree with estimated mutual information weights



Warm-up: Learning Tree Models

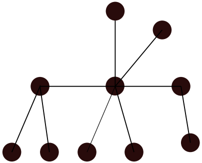
Data processing inequality for Markov chains

$$I(X_1; X_3) \le I(X_1; X_2), I(X_2; X_3).$$



Tree Structure Estimation (Chow and Liu '68)

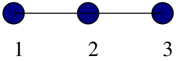
- MLE: Max-weight tree with estimated mutual information weights
- Pairwise statistics suffice



Warm-up: Learning Tree Models

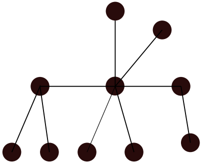
Data processing inequality for Markov chains

$$I(X_1; X_3) \leq I(X_1; X_2), I(X_2; X_3).$$



Tree Structure Estimation (Chow and Liu '68)

- MLE: Max-weight tree with estimated mutual information weights
- Pairwise statistics suffice
- n samples and p nodes

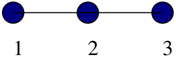


Sample complexity: $\dfrac{\log p}{n} = O(1).$

Warm-up: Learning Tree Models

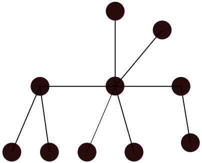
Data processing inequality for Markov chains

$$I(X_1; X_3) \leq I(X_1; X_2), I(X_2; X_3).$$



Tree Structure Estimation (Chow and Liu '68)

- MLE: Max-weight tree with estimated mutual information weights
- Pairwise statistics suffice
- n samples and p nodes



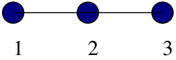
Sample complexity: $\frac{\log p}{n} = O(1)$.

- Efficient inference using belief propagation

Warm-up: Learning Tree Models

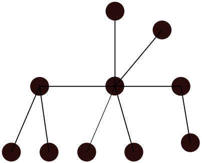
Data processing inequality for Markov chains

$$I(X_1;X_3) \le I(X_1;X_2), I(X_2;X_3).$$



Tree Structure Estimation (Chow and Liu '68)

- MLE: Max-weight tree with estimated mutual information weights
- Pairwise statistics suffice
- n samples and p nodes



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

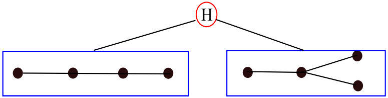
- Efficient inference using belief propagation

What other models are tractable for learning and inference?

Beyond Trees: Tree Mixture Models

Tree Mixture Model

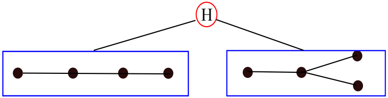
- Each component is a tree model
- Class variable is latent or hidden



Beyond Trees: Tree Mixture Models

Tree Mixture Model

- Each component is a tree model
- Class variable is latent or hidden



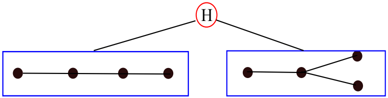
Why use tree mixtures?

- Efficient Inference: BP on component trees and combining them.
- Similarly marginalization and sampling also efficient.

Beyond Trees: Tree Mixture Models

Tree Mixture Model

- Each component is a tree model
- Class variable is latent or hidden



Why use tree mixtures?

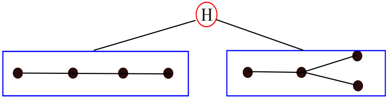
- Efficient Inference: BP on component trees and combining them.
- Similarly marginalization and sampling also efficient.

Learning Tree Mixtures?

Beyond Trees: Tree Mixture Models

Tree Mixture Model

- Each component is a tree model
- Class variable is latent or hidden



Why use tree mixtures?

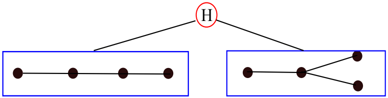
- Efficient Inference: BP on component trees and combining them.
- Similarly marginalization and sampling also efficient.

Learning Tree Mixtures? Alternatives to EM (Meila & Jordan)?

Beyond Trees: Tree Mixture Models

Tree Mixture Model

- Each component is a tree model
- Class variable is latent or hidden



Why use tree mixtures?

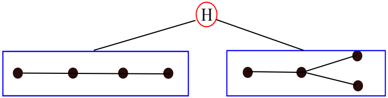
- Efficient Inference: BP on component trees and combining them.
- Similarly marginalization and sampling also efficient.

Learning Tree Mixtures? Alternatives to EM (Meila & Jordan)?

Our approach

- Approximating graphical model mixtures with a tree mixture model
- Efficient algorithms with guarantees to learn best approximation

Beyond Trees: Tree Mixture Models



Tree Mixture Model

- Each component is a tree model
- Class variable is latent or hidden

Why use tree mixtures?

- Efficient Inference: BP on component trees and combining them.
- Similarly marginalization and sampling also efficient.

Learning Tree Mixtures? Alternatives to EM (Meila & Jordan)?

Our approach

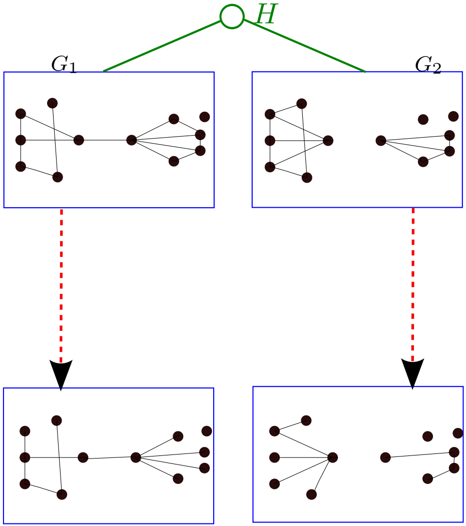
- Approximating graphical model mixtures with a tree mixture model
- Efficient algorithms with guarantees to learn best approximation

Novel approach to learning tree mixture approximations

Mixtures of Graphical Models: Our Approach

Our Approach

- Consider data from graphical model mixture
- Output tree mixture: best tree approx. of each component

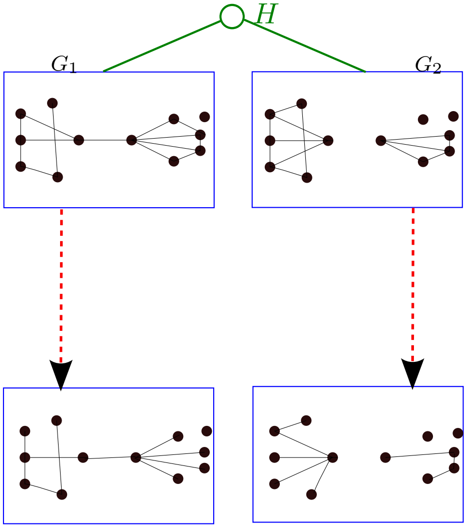


Mixtures of Graphical Models: Our Approach

Our Approach

- Consider data from graphical model mixture
- Output tree mixture: best tree approx. of each component

Steps



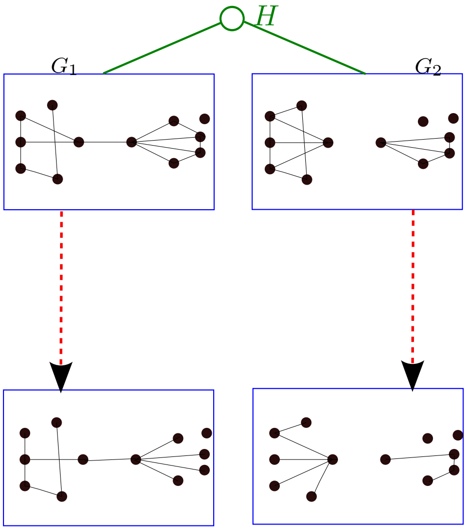
Mixtures of Graphical Models: Our Approach

Our Approach

- Consider data from graphical model mixture
- Output tree mixture: best tree approx. of each component

Steps

- Estimation of union graph



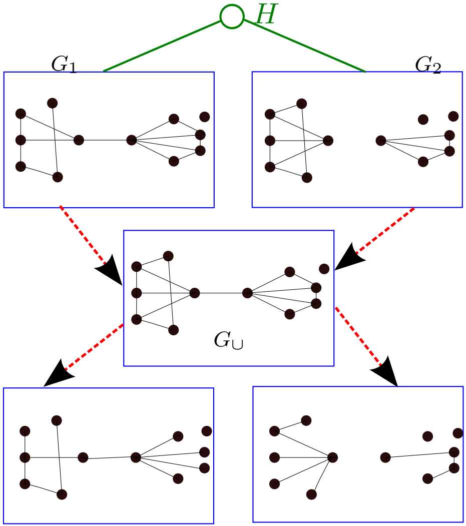
Mixtures of Graphical Models: Our Approach

Our Approach

- Consider data from graphical model mixture
- Output tree mixture: best tree approx. of each component

Steps

- Estimation of union graph



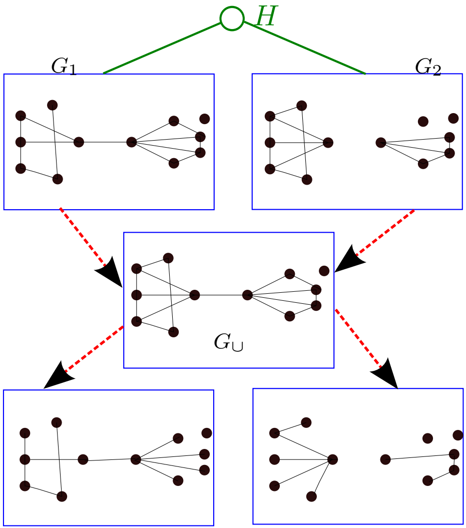
Mixtures of Graphical Models: Our Approach

Our Approach

- Consider data from graphical model mixture
- Output tree mixture: best tree approx. of each component

Steps

- Estimation of union graph
- Estimation of pairwise moments in each component



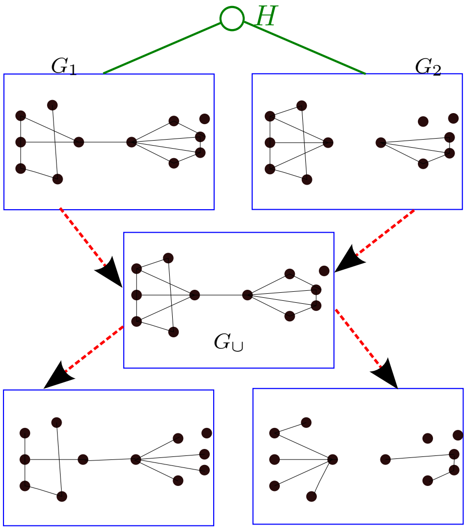
Mixtures of Graphical Models: Our Approach

Our Approach

- Consider data from graphical model mixture
- Output tree mixture: best tree approx. of each component

Steps

- Estimation of union graph
- Estimation of pairwise moments in each component
- Tree approximation of each component via Chow-Liu algorithm.



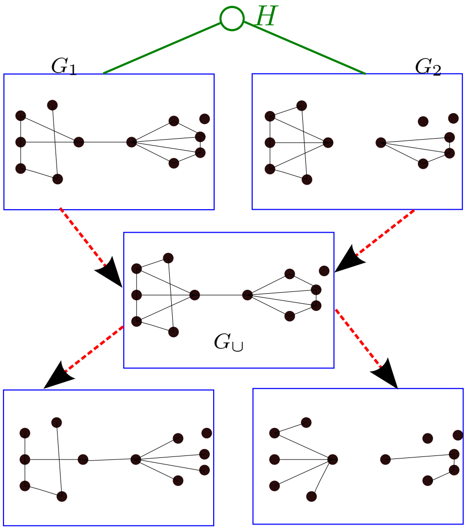
Mixtures of Graphical Models: Our Approach

Our Approach

- Consider data from graphical model mixture
- Output tree mixture: best tree approx. of each component

Steps

- Estimation of union graph
- Estimation of pairwise moments in each component
- Tree approximation of each component via Chow-Liu algorithm.



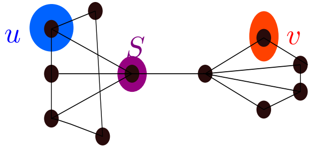
Outline

Sparse Graphical Model Selection: Intuitions

- First consider a graphical model with no latent variables

Markov Property of Graphical Models

$$\boxed{X_u \perp\!\!\!\perp X_v | \mathbf{X}_S \iff I(X_u; X_v | \mathbf{X}_S) = 0}$$

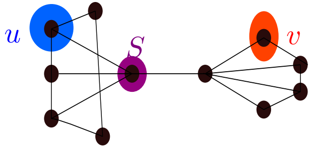


Sparse Graphical Model Selection: Intuitions

- First consider a graphical model with no latent variables

Markov Property of Graphical Models

$$\boxed{X_u \perp\!\!\!\perp X_v | \mathbf{X}_S \iff I(X_u; X_v | \mathbf{X}_S) = 0}$$



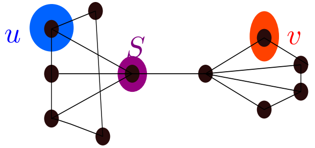
Alternative Test for Conditional Independence?

Sparse Graphical Model Selection: Intuitions

- First consider a graphical model with no latent variables

Markov Property of Graphical Models

$$\boxed{X_u \perp\!\!\!\perp X_v | \mathbf{X}_S \iff I(X_u; X_v | \mathbf{X}_S) = 0}$$



Alternative Test for Conditional Independence?

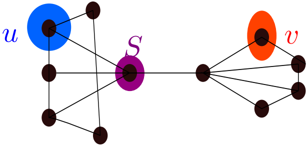
$$P(X_u = i, X_v = j | \mathbf{X}_S = k) = \boxed{P(X_u = i | \mathbf{X}_S = k)}\ \boxed{P(X_v = j | \mathbf{X}_S = k)}$$

Sparse Graphical Model Selection: Intuitions

- First consider a graphical model with no latent variables

Markov Property of Graphical Models

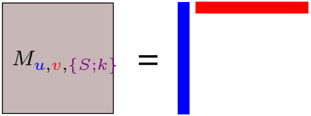
$$\boxed{X_u \perp\!\!\!\perp X_v | \mathbf{X}_S \iff I(X_u; X_v | \mathbf{X}_S) = 0}$$



Alternative Test for Conditional Independence?

$$P(X_u = i, X_v = j | \mathbf{X}_S = k) = \boxed{P(X_u = i | \mathbf{X}_S = k)}\ \boxed{P(X_v = j | \mathbf{X}_S = k)}$$

$$M_{u,v,\{S;k\}} := [P(X_u = i, X_v = j, \mathbf{X}_S = k)]_{i,j}.$$

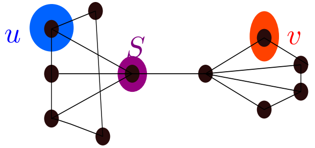


Sparse Graphical Model Selection: Intuitions

- First consider a graphical model with no latent variables

Markov Property of Graphical Models

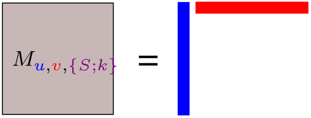
$$\boxed{X_u \perp\!\!\!\perp X_v | \mathbf{X}_S \iff \mathrm{Rank}(M_{u,v,\{S;k\}}) = 1}$$



Alternative Test for Conditional Independence?

$$P(X_u = i, X_v = j | \mathbf{X}_S = k) = \boxed{P(X_u = i | \mathbf{X}_S = k)}\ \boxed{P(X_v = j | \mathbf{X}_S = k)}$$

$$M_{u,v,\{S;k\}} := [P(X_u = i, X_v = j, \mathbf{X}_S = k)]_{i,j}.$$

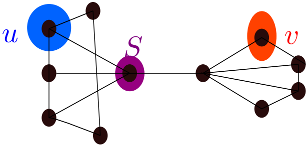


Sparse Graphical Model Selection: Intuitions

- First consider a graphical model with no latent variables

Markov Property of Graphical Models

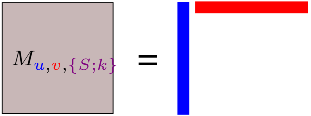
$$\boxed{X_u \perp\!\!\!\perp X_v | \mathbf{X}_S \iff \mathrm{Rank}(M_{u,v,\{S;k\}}) = 1}$$



Alternative Test for Conditional Independence?

$$P(X_u = i, X_v = j | \mathbf{X}_S = k) = \boxed{P(X_u = i | \mathbf{X}_S = k)} \; \boxed{P(X_v = j | \mathbf{X}_S = k)}$$

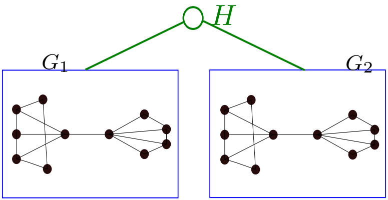
$$M_{u,v,\{S;k\}} := [P(X_u = i, X_v = j, \mathbf{X}_S = k)]_{i,j}.$$



Rank Test on Pairwise Probability Matrices

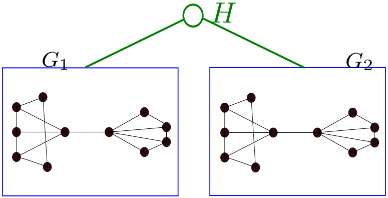
Extending Rank Tests to Mixtures

- Dimension of latent H is r and each observed variable is $d > r$.



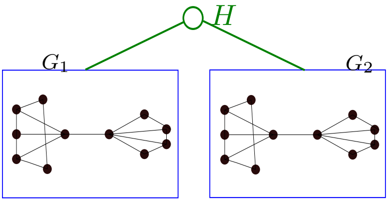
Extending Rank Tests to Mixtures

- Dimension of latent H is r and each observed variable is $d > r$.
- First assume Markov graph is the same for all components.

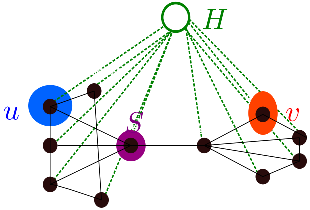


Extending Rank Tests to Mixtures

- Dimension of latent H is r and each observed variable is $d > r$.
- First assume Markov graph is the same for all components.

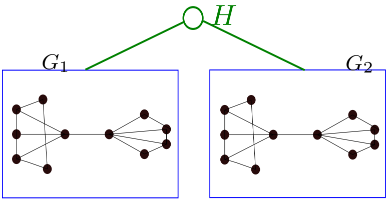


$\boxed{X_u \not\perp X_v | \mathbf{X}_S}$

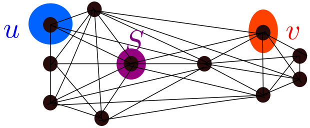


Extending Rank Tests to Mixtures

- Dimension of latent H is r and each observed variable is $d > r$.
- First assume Markov graph is the same for all components.

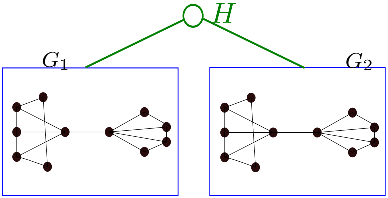


$$\boxed{X_u \not\perp X_v | \mathbf{X}_S}$$

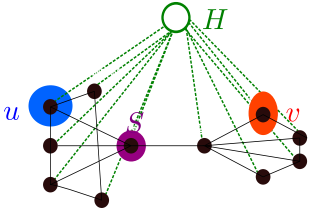


Extending Rank Tests to Mixtures

- Dimension of latent H is r and each observed variable is $d > r$.
- First assume Markov graph is the same for all components.

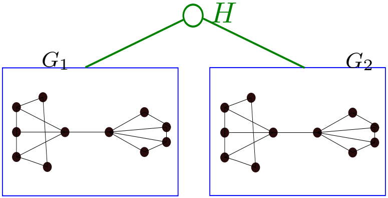


$$\boxed{X_u \not\perp X_v | \mathbf{X}_S}$$

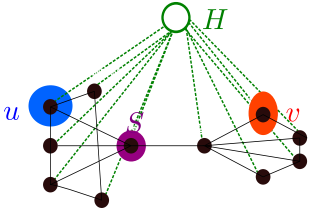


Extending Rank Tests to Mixtures

- Dimension of latent H is r and each observed variable is $d > r$.
- First assume Markov graph is the same for all components.

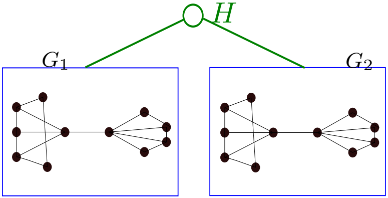


$$\boxed{X_u \perp\!\!\!\perp X_v | \mathbf{X}_S, H}$$

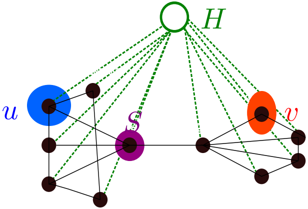


Extending Rank Tests to Mixtures

- Dimension of latent H is r and each observed variable is $d > r$.
- First assume Markov graph is the same for all components.



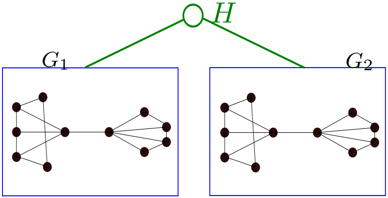
$$\boxed{X_u \perp\!\!\!\perp X_v | \mathbf{X}_S, H}$$



$$P(X_u, X_v | \mathbf{X}_S) = \sum_{h=1}^{r} \boxed{P(X_u | \mathbf{X}_S, \mathbf{H} = \mathbf{h})} \; \boxed{P(\mathbf{H} = \mathbf{h} | \mathbf{X}_S)} \; \boxed{P(X_v | \mathbf{X}_S, \mathbf{H} = \mathbf{h})}$$

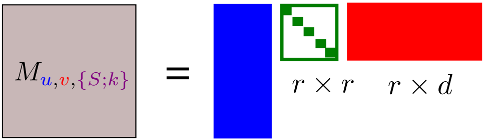
Extending Rank Tests to Mixtures

- Dimension of latent H is r and each observed variable is $d > r$.
- First assume Markov graph is the same for all components.

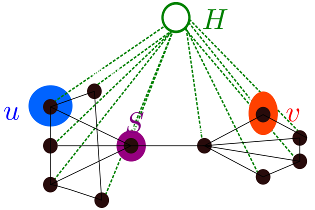


$$\boxed{X_u \perp\!\!\!\perp X_v | \mathbf{X}_S, H}$$

$$M_{u,v,\{S;k\}} := [P(X_u = i, X_v = j, \mathbf{X}_S = k)]_{i,j}.$$



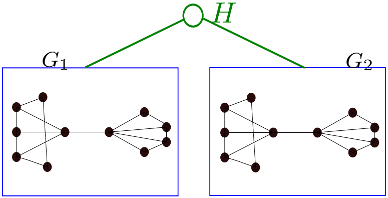
$d \times d$ $\quad$ $d \times r$



$$P(X_u, X_v | \mathbf{X}_S) = \sum_{h=1}^{r} \boxed{P(X_u | \mathbf{X}_S, \mathbf{H} = \mathbf{h})} \; \boxed{P(\mathbf{H} = \mathbf{h} | \mathbf{X}_S)} \; \boxed{P(X_v | \mathbf{X}_S, \mathbf{H} = \mathbf{h})}$$

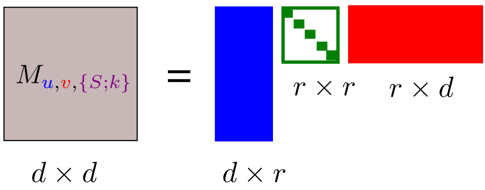
Extending Rank Tests to Mixtures

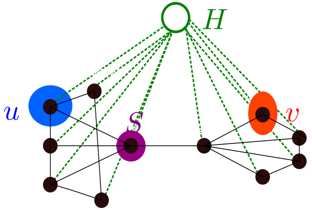
- Dimension of latent H is r and each observed variable is $d > r$.
- First assume Markov graph is the same for all components.



$$\boxed{X_u \perp\!\!\!\perp X_v | \mathbf{X}_S, H} \iff \boxed{\mathrm{Rank}(M_{u,v,\{S;k\}}) = r}$$

$$M_{u,v,\{S;k\}} := [P(X_u = i, X_v = j, \mathbf{X}_S = k)]_{i,j}.$$

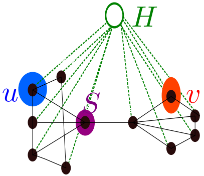




$$P(X_u, X_v | \mathbf{X}_S) = \sum_{h=1}^{r} \boxed{P(X_u | \mathbf{X}_S, \mathbf{H} = \mathbf{h})} \boxed{P(\mathbf{H} = \mathbf{h} | \mathbf{X}_S)} \boxed{P(X_v | \mathbf{X}_S, \mathbf{H} = \mathbf{h})}$$

Rank Test for Mixtures

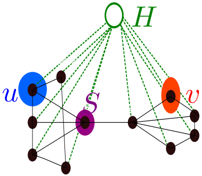
- Dim(H) is r and each observed variable is $d > r$.
- $G_\cup$: union of Markov graphs of components.
- η: Bound on separators btw. node pairs in $G_\cup$.
- $M_{u,v,\{S;k\}} := [P(X_u = i, X_v = j, \mathbf{X}_S = k)]_{i,j}$



Declare (u, v) as edge if $\min_{\substack{S \subset V \setminus \{u,v\} \\ |S| \leq \eta}} \max_{k \in \mathcal{X}^{|S|}} \mathrm{Rank}(M_{u,v,\{S;k\}}; \xi_{n,p}) > r.$

Rank Test for Mixtures

- Dim(H) is r and each observed variable is $d > r$.
- $G_\cup$: union of Markov graphs of components.
- η: Bound on separators btw. node pairs in $G_\cup$.
- $M_{u,v,\{S;k\}} := [P(X_u = i, X_v = j, \mathbf{X}_S = k)]_{i,j}$

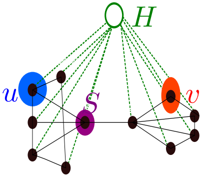


Declare (u, v) as edge if $\min_{\substack{S \subset V \setminus \{u,v\} \\ |S| \leq \eta}} \max_{k \in \mathcal{X}^{|S|}} \text{Rank}(M_{u,v,\{S;k\}}; \xi_{n,p}) > r.$

Small $\eta \Rightarrow$ computationally efficient, uses only low order statistics.

Rank Test for Mixtures

- $\mathrm{Dim}(H)$ is r and each observed variable is $d > r$.
- $G_\cup$: union of Markov graphs of components.
- η: Bound on separators btw. node pairs in $G_\cup$.
- $M_{u,v,\{S;k\}} := [P(X_u = i, X_v = j, \mathbf{X}_S = k)]_{i,j}$



Declare (u, v) as edge if $\min\limits_{\substack{S \subset V \setminus \{u,v\} \\ |S| \le \eta}} \max\limits_{k \in \mathcal{X}^{|S|}} \mathrm{Rank}(M_{u,v,\{S;k\}}; \xi_{n,p}) > r.$

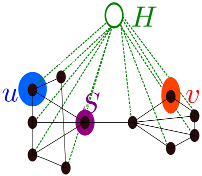
Small $\eta \Rightarrow$ computationally efficient, uses only low order statistics.

Examples of graphs $G_\cup$ with small η

- Mixture of product distributions: $G_\cup$ is trivial and $\eta = 0$.
- Mixture on same tree: $G_\cup$ is a tree and $\eta = 1$.
- Mixture on arbitrary trees: $G_\cup$ is union of r trees and $\eta = r$.

Rank Test for Mixtures

- $\mathrm{Dim}(H)$ is r and each observed variable is $d > r$.
- $G_\cup$: union of Markov graphs of components.
- η: Bound on separators btw. node pairs in $G_\cup$.
- $M_{u,v,\{S;k\}} := [P(X_u = i, X_v = j, \mathbf{X}_S = k)]_{i,j}$



$$\boxed{\text{Declare } (u,v) \text{ as edge if } \min_{\substack{S \subset V \setminus \{u,v\} \\ |S| \leq \eta}} \max_{k \in \mathcal{X}^{|S|}} \mathrm{Rank}(M_{u,v,\{S;k\}}; \xi_{n,p}) > r.}$$

Small $\eta \Rightarrow$ computationally efficient, uses only low order statistics.

Examples of graphs $G_\cup$ with small η

- Mixture of product distributions: $G_\cup$ is trivial and $\eta = 0$.
- Mixture on same tree: $G_\cup$ is a tree and $\eta = 1$.
- Mixture on arbitrary trees: $G_\cup$ is union of r trees and $\eta = r$.

Guarantees on Rank Test

Theorem (**A.** , Hsu, Kakade '12)

Rank test recovers graph structure $G_\cup$ correctly w.h.p on p nodes under n samples when

$$\boxed{\frac{\rho_{\min}^{-2}\log p}{n} = O(1).}$$

- $\rho_{\min}$: Min. $(r+1)^{\text{th}}$ singular value between neighbors.

Guarantees on Rank Test

Theorem (**A.** , Hsu, Kakade '12)

Rank test recovers graph structure $G_\cup$ correctly w.h.p on p nodes under n samples when

$$\boxed{\frac{\rho_{\min}^{-2}\log p}{n} = O(1).}$$

- $\rho_{\min}$: Min. $(r+1)^{\text{th}}$ singular value between neighbors.

Efficient Test with Low Sample and Computational Requirements.

Guarantees on Rank Test

Theorem (**A.** , Hsu, Kakade '12)

Rank test recovers graph structure $G_\cup$ correctly w.h.p on p nodes under n samples when

$$\boxed{\frac{\rho_{\min}^{-2}\log p}{n} = O(1).}$$

- $\rho_{\min}$: Min. $(r+1)^{\text{th}}$ singular value between neighbors.

Efficient Test with Low Sample and Computational Requirements.

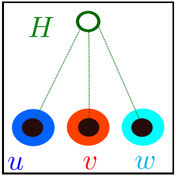
Recall examples of graphs $G_\cup$ with small η

- Mixture of product distributions: $G_\cup$ is trivial and $\eta = 0$.
- Mixture on same tree: $G_\cup$ is a tree and $\eta = 1$.
- Mixture on arbitrary trees: $G_\cup$ is union of r trees and $\eta = r$.

Outline

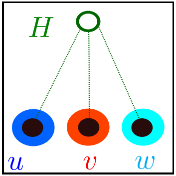
Mixture of Product Distributions: Intuitions

- $X_u \perp\!\!\!\perp X_v \perp\!\!\!\perp X_w | H$
- Union graph of mixture components is trivial
- Transition matrices full rank (non-singular)
- Consider special case when $d = r$



Mixture of Product Distributions: Intuitions

- $X_u \perp\!\!\!\perp X_v \perp\!\!\!\perp X_w | H$
- Union graph of mixture components is trivial
- Transition matrices full rank (non-singular)
- Consider special case when $d = r$

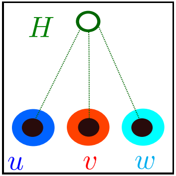


Parameter Estimation: Generalized Eigenvalue Problem

- Studied by (Chang), (Mossel & Roch), (Hsu, Kakade & Zhang)

Mixture of Product Distributions: Intuitions

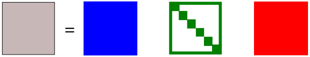
- $X_u \perp\!\!\!\perp X_v \perp\!\!\!\perp X_w | H$
- Union graph of mixture components is trivial
- Transition matrices full rank (non-singular)
- Consider special case when $d = r$



Parameter Estimation: Generalized Eigenvalue Problem

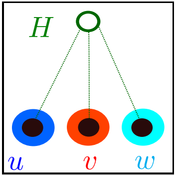
- Studied by (Chang), (Mossel & Roch), (Hsu, Kakade & Zhang)

$$M_{u,v} := [P(X_u = i, X_v = j)]_{i,j}$$



Mixture of Product Distributions: Intuitions

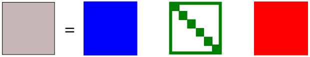
- $X_u \perp\!\!\!\perp X_v \perp\!\!\!\perp X_w | H$
- Union graph of mixture components is trivial
- Transition matrices full rank (non-singular)
- Consider special case when $d = r$



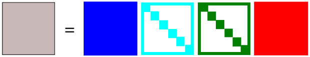
Parameter Estimation: Generalized Eigenvalue Problem

- Studied by (Chang), (Mossel & Roch), (Hsu, Kakade & Zhang)

$$M_{u,v} := [P(X_u = i, X_v = j)]_{i,j}$$

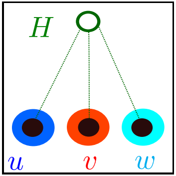


$$M_{u,v,\{w;k\}} := [P(X_u = i, X_v = j, X_w = k)]_{i,j}$$



Mixture of Product Distributions: Intuitions

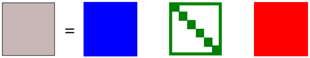
- $X_u \perp\!\!\!\perp X_v \perp\!\!\!\perp X_w | H$
- Union graph of mixture components is trivial
- Transition matrices full rank (non-singular)
- Consider special case when $d = r$



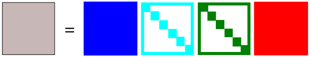
Parameter Estimation: Generalized Eigenvalue Problem

- Studied by (Chang), (Mossel & Roch), (Hsu, Kakade & Zhang)

$$M_{u,v} := [P(X_u = i, X_v = j)]_{i,j}$$



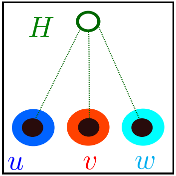
$$M_{u,v,\{w;k\}} := [P(X_u = i, X_v = j, X_w = k)]_{i,j}$$



$$M_{u,v,\{w;k\}} M_{u,v}^{-1} = V \underbrace{\mathrm{Diag}(P(X_w = k|H))} V^{-1}$$

Mixture of Product Distributions: Intuitions

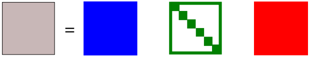
- $X_u \perp\!\!\!\perp X_v \perp\!\!\!\perp X_w | H$
- Union graph of mixture components is trivial
- Transition matrices full rank (non-singular)
- Consider special case when $d = r$



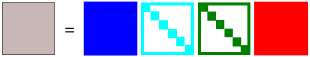
Parameter Estimation: Generalized Eigenvalue Problem

- Studied by (Chang), (Mossel & Roch), (Hsu, Kakade & Zhang)

$$M_{u,v} := [P(X_u = i, X_v = j)]_{i,j}$$



$$M_{u,v,\{w;k\}} := [P(X_u = i, X_v = j, X_w = k)]_{i,j}$$



$$M_{u,v,\{w;k\}} M_{u,v}^{-1} = V \underbrace{\mathsf{Diag}(P(X_w = k|H))}_{} V^{-1}$$

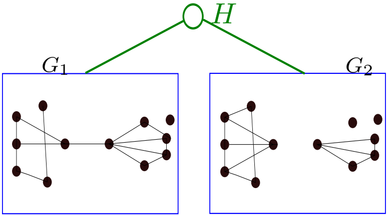
Efficient estimation of non-singular product mixtures

Learning Graphical Model Mixtures

Adapt Eigenvalue Method for Graphical Model Mixtures?

Challenges

- Not a mixture of product distributions: Eigenvalue method not valid.



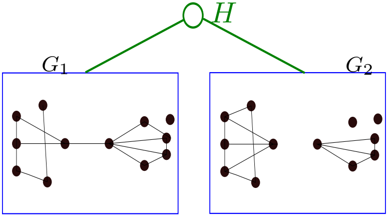
Solutions

Learning Graphical Model Mixtures

Adapt Eigenvalue Method for Graphical Model Mixtures?

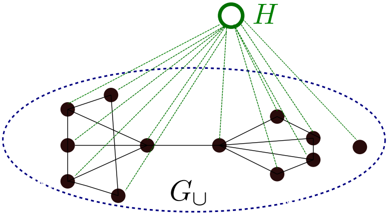
Challenges

- Not a mixture of product distributions: Eigenvalue method not valid.



Solutions

- $G_\cup$: union graph learnt from rank test

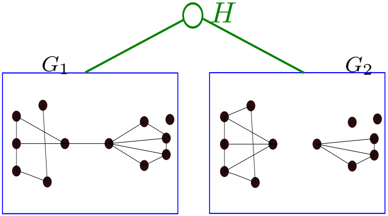


Learning Graphical Model Mixtures

Adapt Eigenvalue Method for Graphical Model Mixtures?

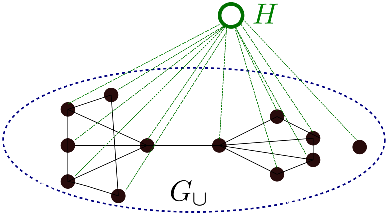
Challenges

- Not a mixture of product distributions: Eigenvalue method not valid.



Solutions

- $G_\cup$: union graph learnt from rank test
- Use separators on union graph $G_\cup$:

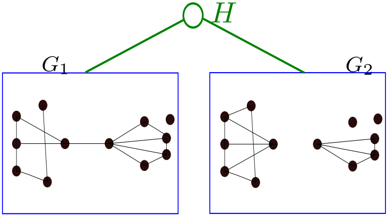


Learning Graphical Model Mixtures

Adapt Eigenvalue Method for Graphical Model Mixtures?

Challenges

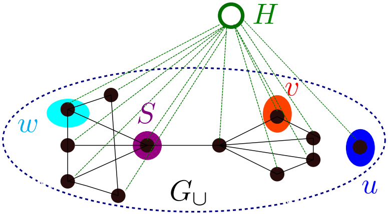
- Not a mixture of product distributions: Eigenvalue method not valid.



Solutions

- $G_\cup$: union graph learnt from rank test
- Use separators on union graph $G_\cup$:

$$\boxed{X_u \perp\!\!\!\perp X_v \perp\!\!\!\perp X_w \,|X_S, H}$$

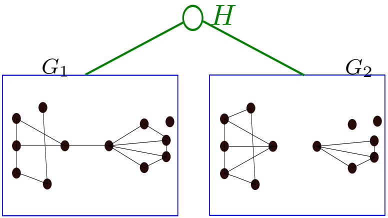


Learning Graphical Model Mixtures

Adapt Eigenvalue Method for Graphical Model Mixtures?

Challenges

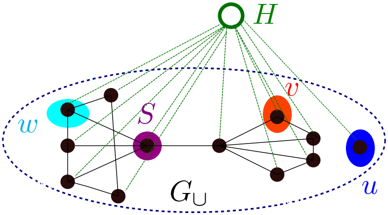
- Not a mixture of product distributions: Eigenvalue method not valid.
- Need to learn higher order moments of mixture components.



Solutions

- $G_\cup$: union graph learnt from rank test
- Use separators on union graph $G_\cup$:

$$\boxed{X_u \perp\!\!\!\perp X_v \perp\!\!\!\perp X_w \,|X_S, H}$$

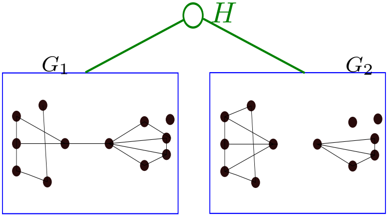


Learning Graphical Model Mixtures

Adapt Eigenvalue Method for Graphical Model Mixtures?

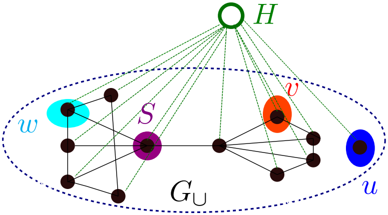
Challenges

- Not a mixture of product distributions: Eigenvalue method not valid.
- Need to learn higher order moments of mixture components.



Solutions

- $G_\cup$: union graph learnt from rank test
- Use separators on union graph $G_\cup$:
  $\boxed{X_u \perp\!\!\!\perp X_v \perp\!\!\!\perp | X_S, H}$
- Learn tree mixture approximation: estimate pairwise mixture moments

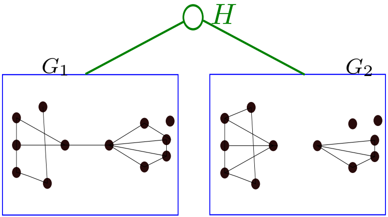


Learning Graphical Model Mixtures

Adapt Eigenvalue Method for Graphical Model Mixtures?

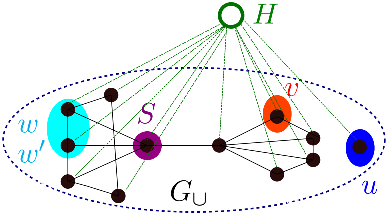
Challenges

- Not a mixture of product distributions: Eigenvalue method not valid.
- Need to learn higher order moments of mixture components.



Solutions

- $G_\cup$: union graph learnt from rank test
- Use separators on union graph $G_\cup$:
  $\boxed{X_u \perp\!\!\!\perp X_v \perp\!\!\!\perp X_{w,w'} | X_S, H}$
- Learn tree mixture approximation: estimate pairwise mixture moments

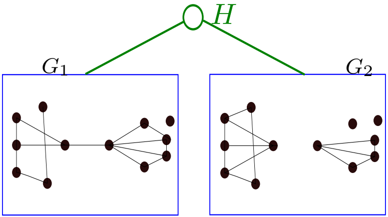


Learning Graphical Model Mixtures

Adapt Eigenvalue Method for Graphical Model Mixtures?

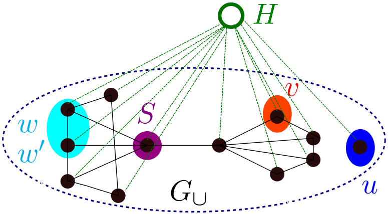
Challenges

- Not a mixture of product distributions: Eigenvalue method not valid.
- Need to learn higher order moments of mixture components.



Solutions

- $G_\cup$: union graph learnt from rank test
- Use separators on union graph $G_\cup$: $\boxed{X_u \perp\!\!\!\perp X_v \perp\!\!\!\perp X_{w,w'} | X_S, H}$
- Learn tree mixture approximation: estimate pairwise mixture moments



Efficient Estimation of Tree Mixture Approximations

Guarantees for Learning Graphical Model Mixtures

Steps Involved in Tree Mixture Approximation

- Rank tests for structure estimation of union graph $G_\cup$
- Eigenvalue decomposition for estimation of pairwise moments of mixture components
- Chow-Liu algorithm to estimate mixture component trees

Computationally Efficient Algorithm for Learning Graphical Model Mixtures

Guarantees for Learning Graphical Model Mixtures

Steps Involved in Tree Mixture Approximation

- Rank tests for structure estimation of union graph $G_\cup$
- Eigenvalue decomposition for estimation of pairwise moments of mixture components
- Chow-Liu algorithm to estimate mixture component trees

Computationally Efficient Algorithm for Learning Graphical Model Mixtures

Theorem (A. , Hsu, Kakade '12)

The above method recovers correct tree mixture approximation correctly w.h.p on p nodes of r component mixture under n samples when

$$\boxed{n = \mathrm{poly}(p, r).}$$

Guarantees for Learning Graphical Model Mixtures

Steps Involved in Tree Mixture Approximation

- Rank tests for structure estimation of union graph $G_\cup$
- Eigenvalue decomposition for estimation of pairwise moments of mixture components
- Chow-Liu algorithm to estimate mixture component trees

Computationally Efficient Algorithm for Learning Graphical Model Mixtures

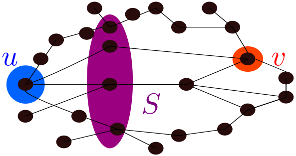
Theorem (A. , Hsu, Kakade '12)

The above method recovers correct tree mixture approximation correctly w.h.p on p nodes of r component mixture under n samples when

$$\boxed{n = \mathrm{poly}(p, r).}$$

Efficient Learning of Multiple Graphs and Models in High Dimensions

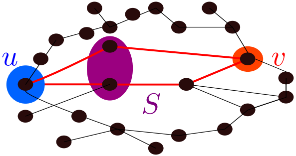
Extensions and Connections



Exact vs. Local Separators in Union Graph

η is bound on local separators of $G_\cup$ and mixture components satisfy correlation decay.

Extensions and Connections



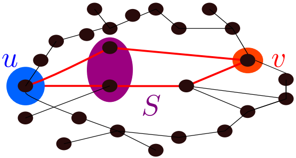
Exact vs. Local Separators in Union Graph

η is bound on local separators of $G_\cup$ and mixture components satisfy correlation decay.

Extensions and Connections

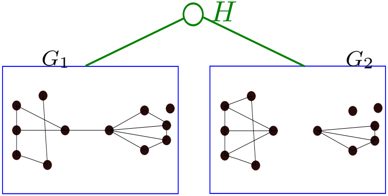
Exact vs. Local Separators in Union Graph

η is bound on local separators of $G_\cup$ and mixture components satisfy correlation decay.

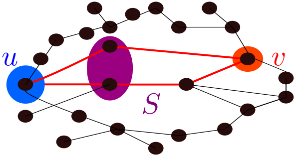


Estimation of component graphs

- Estimate joint moments in nbd. of union graph for each component.
- Neighborhood selection for each mixture component.
- Efficient for low degree union graphs.



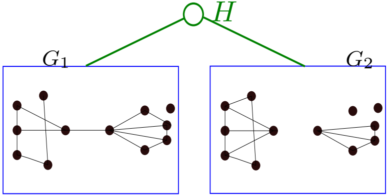
Extensions and Connections



Exact vs. Local Separators in Union Graph

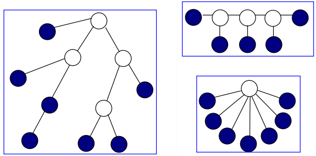
η is bound on local separators of $G_\cup$ and mixture components satisfy correlation decay.

Estimation of component graphs



- Estimate joint moments in nbd. of union graph for each component.
- Neighborhood selection for each mixture component.
- Efficient for low degree union graphs.

Estimation in other models



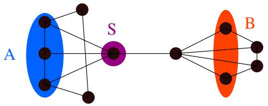
- HMM, latent trees and general multiview mixtures
- Improvement for product mixtures

Outline

Beyond Graphical Modeling: Incorporating Residuals

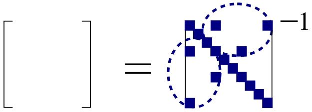
Recall notion of graphical models...

- Conditional Independence: $\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_S$
- $P(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} \Psi_{i,j}(x_i, x_j)\right].$



Gaussian Graphical Models

- $f(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} J^*_{i,j} x_i, x_j\right].$

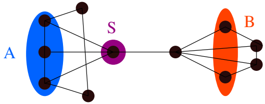


- Covariance: $\Sigma^* = J^{*-1}$.

Beyond Graphical Modeling: Incorporating Residuals

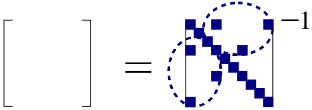
Recall notion of graphical models...

- Conditional Independence: $\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_S$
- $P(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} \Psi_{i,j}(x_i, x_j)\right]$.



Gaussian Graphical Models

- $f(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} J^*_{i,j} x_i, x_j\right]$.
- Covariance: $\Sigma^* = J^{*-1}$.

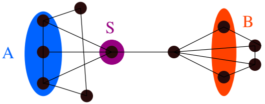


Incorporating Residuals: $\boxed{\Sigma^* = {J^*_M}^{-1} + \Sigma^*_R.}$

- Generalize independence and graphical modeling

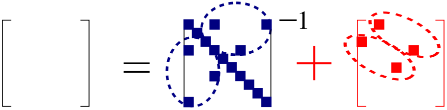
Beyond Graphical Modeling: Incorporating Residuals

Recall notion of graphical models...



- Conditional Independence: $\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_S$
- $P(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} \Psi_{i,j}(x_i, x_j)\right].$

Gaussian Graphical Models



- $f(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} J^*_{i,j} x_i, x_j\right].$
- Covariance: $\Sigma^* = J^{*-1}$.

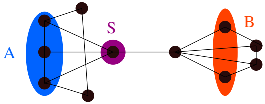
Incorporating Residuals: $\boxed{\Sigma^* = {J^*_M}^{-1} + \Sigma^*_R.}$

- Generalize independence and graphical modeling

Beyond Graphical Modeling: Incorporating Residuals

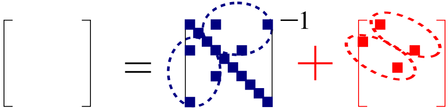
Recall notion of graphical models...

- Conditional Independence: $\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_S$
- $P(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} \Psi_{i,j}(x_i, x_j)\right].$



Gaussian Graphical Models

- $f(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} J^*_{i,j} x_i, x_j\right].$



- Covariance: $\Sigma^* = J^{*-1}$.

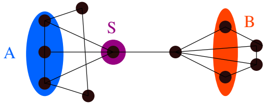
Incorporating Residuals: $\boxed{\Sigma^* = {J^*_M}^{-1} + \Sigma^*_R.}$

- Generalize independence and graphical modeling
- Decompose and simultaneously enforce sparsity in J_M and Σ_R

Beyond Graphical Modeling: Incorporating Residuals

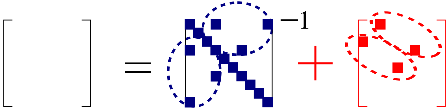
Recall notion of graphical models...

- Conditional Independence: $\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_S$
- $P(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} \Psi_{i,j}(x_i, x_j)\right].$



Gaussian Graphical Models

- $f(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} J^*_{i,j} x_i, x_j\right].$
- Covariance: $\Sigma^* = {J^*}^{-1}$.



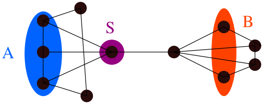
Incorporating Residuals: $\boxed{\Sigma^* = {J^*_M}^{-1} + \Sigma^*_R.}$

- Generalize independence and graphical modeling
- Decompose and simultaneously enforce sparsity in J_M and Σ_R
- Convex programs? Regularizers? High-dimensional guarantees?

Beyond Graphical Modeling: Incorporating Residuals

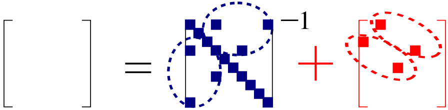
Recall notion of graphical models...

- Conditional Independence: $\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_S$
- $P(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} \Psi_{i,j}(x_i, x_j)\right]$.



Gaussian Graphical Models

- $f(\mathbf{x}) \propto \exp\left[\sum_{(i,j)\in G} J^*_{i,j} x_i, x_j\right]$.



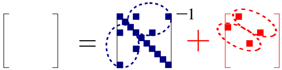
- Covariance: $\Sigma^* = J^{*-1}$.

Incorporating Residuals: $\boxed{\Sigma^* = {J^*_M}^{-1} + \Sigma^*_R.}$

- Generalize independence and graphical modeling
- Decompose and simultaneously enforce sparsity in J_M and Σ_R
- Convex programs? Regularizers? High-dimensional guarantees?

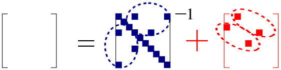
Algorithm for Covariance Decomposition

$$\boxed{\Sigma^* = {J_M^*}^{-1} + \Sigma_R^*.}$$



Algorithm for Covariance Decomposition

$$\boxed{\Sigma^* = {J_M^*}^{-1} + \Sigma_R^*.}$$

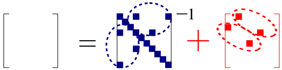


ℓ_1 penalized MLE for Graphical Models (Ravikumar et. al. ‘08)

- $\widehat{\Sigma}^n$: sample covariance using n i.i.d. samples

Algorithm for Covariance Decomposition

$$\boxed{\Sigma^* = {J_M^*}^{-1} + \Sigma_R^*.}$$



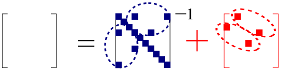
ℓ_1 penalized MLE for Graphical Models (Ravikumar et. al. '08)

- $\widehat{\Sigma}^n$: sample covariance using n i.i.d. samples

$$\widehat{J}_M := \underset{J_M \succ 0}{\operatorname{argmin}} \langle \widehat{\Sigma}^n, J_M \rangle - \log\det J_M + \gamma \|J_M\|_{1,\text{off}}$$

Algorithm for Covariance Decomposition

$$\boxed{\Sigma^* = {J^*_M}^{-1} + \Sigma^*_R.}$$



ℓ_1 penalized MLE for Graphical Models (Ravikumar et. al. '08)

- $\widehat{\Sigma}^n$: sample covariance using n i.i.d. samples

$$\widehat{J}_M := \underset{J_M \succ 0}{\operatorname{argmin}} \langle \widehat{\Sigma}^n, J_M \rangle - \log\det J_M + \gamma \|J_M\|_{1,\text{off}}$$

Max-entropy Formulation for Graphical Models

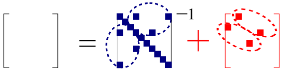
- Lagrangian dual of ℓ_1-penalized MLE

$$\widehat{\Sigma}_M := \underset{\Sigma_M \succ 0}{\operatorname{argmax}} \; \log\det \Sigma_M$$

$$\text{s.t.} \quad \|\widehat{\Sigma}^n - \Sigma_M \|_{\infty,\text{off}} \leq \gamma, \quad \big(\Sigma_M\big)_d = \big(\widehat{\Sigma}^n\big)_d$$

Algorithm for Covariance Decomposition

$$\boxed{\Sigma^* = {J_M^*}^{-1} + \Sigma_R^*.}$$



ℓ_1 penalized MLE for Graphical Models (Ravikumar et. al. '08)

- $\widehat{\Sigma}^n$: sample covariance using n i.i.d. samples

$$\widehat{J}_M := \underset{J_M \succ 0}{\operatorname{argmin}} \langle \widehat{\Sigma}^n, J_M \rangle - \log\det J_M + \gamma \|J_M\|_{1,\text{off}}$$

Max-entropy Formulation for Graphical Models

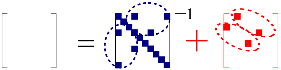
- Lagrangian dual of ℓ_1-penalized MLE

$$\widehat{\Sigma}_M := \underset{\Sigma_M \succ 0}{\operatorname{argmax}} \ \log\det \Sigma_M$$

$$\text{s. t.} \quad \|\widehat{\Sigma}^n - \Sigma_M - \Sigma_R\|_{\infty,\text{off}} \leq \gamma, \quad \big(\Sigma_M\big)_d = \big(\widehat{\Sigma}^n\big)_d$$

Algorithm for Covariance Decomposition

$$\boxed{\Sigma^* = {J_M^*}^{-1} + \Sigma_R^*.}$$



ℓ_1 penalized MLE for Graphical Models (Ravikumar et. al. '08)

- $\widehat{\Sigma}^n$: sample covariance using n i.i.d. samples

$$\widehat{J}_M := \underset{J_M \succ 0}{\operatorname{argmin}} \langle \widehat{\Sigma}^n, J_M \rangle - \log\det J_M + \gamma \|J_M\|_{1,\text{off}}$$

Max-entropy Formulation for Graphical Models

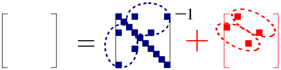
- Lagrangian dual of ℓ_1-penalized MLE

$$(\widehat{\Sigma}_M, \widehat{\Sigma}_R) := \underset{\Sigma_M \succ 0, \Sigma_R}{\operatorname{argmax}} \ \log\det \Sigma_M$$

$$\text{s.t.} \quad \|\widehat{\Sigma}^n - \Sigma_M - \Sigma_R\|_{\infty,\text{off}} \le \gamma, \quad \left(\Sigma_M\right)_d = \left(\widehat{\Sigma}^n\right)_d, \left(\Sigma_R\right)_d = 0.$$

Algorithm for Covariance Decomposition

$$\boxed{\Sigma^* = {J_M^*}^{-1} + \Sigma_R^*.}$$



ℓ_1 penalized MLE for Graphical Models (Ravikumar et. al. '08)

- $\widehat{\Sigma}^n$: sample covariance using n i.i.d. samples

$$\widehat{J}_M := \underset{J_M \succ 0}{\operatorname{argmin}} \langle \widehat{\Sigma}^n, J_M \rangle - \log\det J_M + \gamma \|J_M\|_{1,\text{off}}$$

$$\text{s.t. } \|J_M\|_{\infty,\text{off}} \leq \lambda,$$

Max-entropy Formulation for Graphical Models

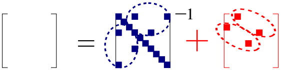
- Lagrangian dual of ℓ_1-penalized MLE

$$(\widehat{\Sigma}_M, \widehat{\Sigma}_R) := \underset{\Sigma_M \succ 0, \Sigma_R}{\operatorname{argmax}} \ \log\det \Sigma_M - \lambda \|\Sigma_R\|_{1,\text{off}}$$

$$\text{s.t. } \quad \|\widehat{\Sigma}^n - \Sigma_M - \Sigma_R\|_{\infty,\text{off}} \leq \gamma, \quad \left(\Sigma_M\right)_d = \left(\widehat{\Sigma}^n\right)_d, \left(\Sigma_R\right)_d = 0.$$

Algorithm for Covariance Decomposition

$$\boxed{\Sigma^* = {J_M^*}^{-1} + \Sigma_R^*.}$$



ℓ_1 penalized MLE for Graphical Models (Ravikumar et. al. '08)

- $\widehat{\Sigma}^n$: sample covariance using n i.i.d. samples

$$\widehat{J}_M := \underset{J_M \succ 0}{\operatorname{argmin}} \langle \widehat{\Sigma}^n, J_M \rangle - \log\det J_M + \gamma \|J_M\|_{1,\text{off}}$$

$$\text{s.t. } \|J_M\|_{\infty,\text{off}} \leq \lambda,$$

Max-entropy Formulation for Graphical Models (Janzamin, A. '12)

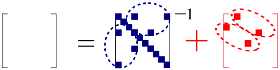
- Lagrangian dual of ℓ_1-penalized MLE

$$(\widehat{\Sigma}_M, \widehat{\Sigma}_R) := \underset{\Sigma_M \succ 0, \Sigma_R}{\operatorname{argmax}} \ \log\det \Sigma_M - \lambda \|\Sigma_R\|_{1,\text{off}}$$

$$\text{s.t. } \quad \|\widehat{\Sigma}^n - \Sigma_M - \Sigma_R\|_{\infty,\text{off}} \leq \gamma, \quad \left(\Sigma_M\right)_d = \left(\widehat{\Sigma}^n\right)_d, \left(\Sigma_R\right)_d = 0.$$

Algorithm for Covariance Decomposition

$$\boxed{\Sigma^* = {J_M^*}^{-1} + \Sigma_R^*.}$$



ℓ_1 penalized MLE for Graphical Models (Ravikumar et. al. '08)

- $\widehat{\Sigma}^n$: sample covariance using n i.i.d. samples

$$\widehat{J}_M := \underset{J_M \succ 0}{\operatorname{argmin}} \langle \widehat{\Sigma}^n, J_M \rangle - \log\det J_M + \gamma \|J_M\|_{1,\text{off}}$$

$$\text{s.t. } \|J_M\|_{\infty,\text{off}} \leq \lambda,$$

Max-entropy Formulation for Graphical Models (Janzamin, A. '12)

- Lagrangian dual of ℓ_1-penalized MLE

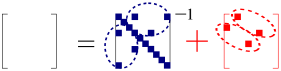
$$(\widehat{\Sigma}_M, \widehat{\Sigma}_R) := \underset{\Sigma_M \succ 0, \Sigma_R}{\operatorname{argmax}} \ \log\det \Sigma_M - \lambda \|\Sigma_R\|_{1,\text{off}}$$

$$\text{s.t.} \quad \|\widehat{\Sigma}^n - \Sigma_M - \Sigma_R\|_{\infty,\text{off}} \leq \gamma, \quad \left(\Sigma_M\right)_d = \left(\widehat{\Sigma}^n\right)_d, \left(\Sigma_R\right)_d = 0.$$

Efficient Method for Covariance Decomposition and Estimation

Guarantees for High-Dimensional Estimation

$$\boxed{\Sigma^* = {J_M^*}^{-1} + \Sigma_R^*.}$$



Theorem (Janzamin and A. '12)

When the number of samples n, number of nodes p and maximum degree Δ in the Markov graph (corresponding to J_M^*) satisfy

$$\boxed{\frac{\Delta^2 \log p}{n} = O(1),}$$

- $(\widehat{J}_M, \widehat{\Sigma}_R)$ are sparsistent and sign consistent
- satisfy norm guarantees

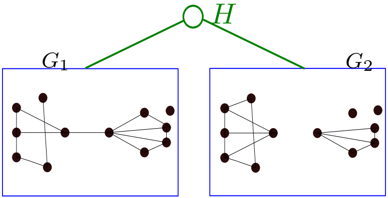
$$\boxed{\|\widehat{J}_M - J_M^*\|_\infty, \|\widehat{\Sigma}_R - \Sigma_R^*\|_\infty = O\left(\sqrt{\frac{\log p}{n}}\right).}$$

Guarantee Sparsistency and Efficient Estimation in Both Domains

Outline

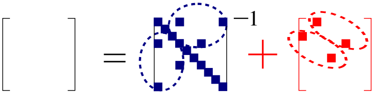
Summary and Outlook

Learning Graphical Model Mixtures



- Tree mixture approximations
- Combinatorial search + spectral decomposition
- Computational and sample guarantees

Markov/Independence Decomposition



- Efficient convex program for decomposition
- Similar requirements as graphical model selection

Outlook

- Converse results for learning graphical mixtures
- Mixed variables, latent models etc.