Latent Variable Modeling: Tensor and Graphical Approaches

Anima Anandkumar

U.C. Irvine
Latent Variable Modeling

Goal: Discover hidden effects from observed measurements

Example: document modeling


Learning latent variable models: efficient methods and guarantees
Challenges and Approaches

Challenges: High-Dimensional Regime

- Sample and Computational complexities
- Identifiability: when can hidden variables be discovered?
Challenges and Approaches

Challenges: High-Dimensional Regime

- Sample and Computational complexities
- Identifiability: when can hidden variables be discovered?

Our Approach: Two Perspectives
Challenges and Approaches

Challenges: High-Dimensional Regime
- Sample and Computational complexities
- Identifiability: when can hidden variables be discovered?

Our Approach: Two Perspectives

Method of Moments
- Hidden choice variable and observed samples
- Inverse moment method: solve equations relating hidden variable to observed moments
- Low order tensor form and efficient decomposition methods
Challenges and Approaches

Challenges: High-Dimensional Regime

- Sample and Computational complexities
- Identifiability: when can hidden variables be discovered?

Our Approach: Two Perspectives

Method of Moments

- Hidden choice variable and observed samples
- Inverse moment method: solve equations relating hidden variable to observed moments
- Low order tensor form and efficient decomposition methods

Graphical Modeling

- Qualitative: graph structure. Quantitative: interaction strengths.
- Markov relationships: graphs with long cycles and hidden variables.
- Greedy graph estimation method: efficient tradeoffs.
## Results from Two Approaches

### Learning Mixture Models through Tensor Decomposition

<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>bush</td>
<td>company</td>
<td>show</td>
</tr>
<tr>
<td>president</td>
<td>percent</td>
<td>book</td>
</tr>
<tr>
<td>government</td>
<td>million</td>
<td>women</td>
</tr>
<tr>
<td>official</td>
<td>companies</td>
<td>family</td>
</tr>
<tr>
<td>campaign</td>
<td>market</td>
<td>film</td>
</tr>
<tr>
<td>political</td>
<td>business</td>
<td>school</td>
</tr>
<tr>
<td>law</td>
<td>stock</td>
<td>look</td>
</tr>
<tr>
<td>leader</td>
<td>billion</td>
<td>home</td>
</tr>
<tr>
<td>george_bush</td>
<td>money</td>
<td>children</td>
</tr>
<tr>
<td>al_gore</td>
<td>cost</td>
<td>friend</td>
</tr>
</tbody>
</table>

- Top 10 words for three topics from NYTimes data set.
Results from Two Approaches

Learning Mixture Models through Tensor Decomposition

<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>bush</td>
<td>company</td>
<td>show</td>
</tr>
<tr>
<td>president</td>
<td>percent</td>
<td>book</td>
</tr>
<tr>
<td>government</td>
<td>million</td>
<td>women</td>
</tr>
<tr>
<td>official</td>
<td>companies</td>
<td>family</td>
</tr>
<tr>
<td>campaign</td>
<td>market</td>
<td>film</td>
</tr>
<tr>
<td>political</td>
<td>business</td>
<td>school</td>
</tr>
<tr>
<td>law</td>
<td>stock</td>
<td>look</td>
</tr>
<tr>
<td>leader</td>
<td>billion</td>
<td>home</td>
</tr>
<tr>
<td>george_bush</td>
<td>money</td>
<td>children</td>
</tr>
<tr>
<td>al_gore</td>
<td>cost</td>
<td>friend</td>
</tr>
</tbody>
</table>

Top 10 words for three topics from NYTimes data set.

Graph Estimation Through Greedy Methods

Graph: Topic-Word Relationships.
Other Motivating Applications

Social Network Modeling

- Community detection: Discovering hidden communities
- Dynamic network modeling: Predicting vertex co-presence
Other Motivating Applications

Social Network Modeling
- Community detection: Discovering hidden communities
- Dynamic network modeling: Predicting vertex co-presence

Bio-Informatics
- Modeling gene associations
- Hidden variables may be regulators that control groups of functionally similar genes
Other Motivating Applications

Social Network Modeling
- Community detection: Discovering hidden communities
- Dynamic network modeling: Predicting vertex co-presence

Bio-Informatics
- Modeling gene associations
- Hidden variables may be regulators that control groups of functionally similar genes

Computer Vision, Phylogenetics, Financial Modeling
Outline

1. Introduction

2. Inverse Moment Methods
   - Moment Tensor Form
   - Tensor Decomposition Methods

   - Latent Tree Models
   - Loopy Latent Models

4. Experiments and Applications

5. Conclusion
Warmup: Exchangeable Single Topic Models

Exchangeability
- Order of words does not matter
- Sufficient statistics: word counts
- DeFinetti’s theorem: latent variable

Exchangeable Topic Models
- \(l\) words in a document \(x_1, \ldots, x_l\).
- Document: topic mixture (draw of \(h\)).
- Word \(x_i\) generated from topic \(y_i\).
- Exchangeability: \(x_1 \perp \!\!\!\perp x_2 \perp \!\!\!\perp \ldots | h\)
- \(\Phi(i, j) := \mathbb{P}[x_m = i | y_m = j]\).
Warmup: Exchangeable Single Topic Models

Exchangeability

- Order of words does not matter
- Sufficient statistics: word counts
- DeFinetti’s theorem: latent variable

Exchangeable Topic Models

- $l$ words in a document $x_1, \ldots, x_l$.
- Document: topic mixture (draw of $h$).
- Word $x_i$ generated from topic $y_i$.
- Exchangeability: $x_1 \perp \perp x_2 \perp \perp \ldots | h$
- $\Phi(i, j) := \mathbb{P}[x_m = i | y_m = j]$. 

\[
\begin{aligned}
\Phi(i, j) &:= \mathbb{P}[x_m = i | y_m = j]. 
\end{aligned}
\]
Warmup: Exchangeable Single Topic Models

Exchangeability
- Order of words does not matter
- Sufficient statistics: word counts
- DeFinetti’s theorem: latent variable

Exchangeable Topic Models
- \( l \) words in a document \( x_1, \ldots, x_l \).
- Document: topic mixture (draw of \( h \)).
- Word \( x_i \) generated from topic \( y_i \).
- Exchangeability: \( x_1 \perp \perp x_2 \perp \perp \ldots | h \)
- \( \Phi(i, j) := \mathbb{P}[x_m = i | y_m = j] \).
Warmup: Exchangeable Single Topic Models

Exchangeability
- Order of words does not matter
- Sufficient statistics: word counts
- DeFinetti’s theorem: latent variable

Exchangeable Topic Models
- $l$ words in a document $x_1, \ldots, x_l$.
- Document: topic mixture (draw of $h$).
- Word $x_i$ generated from topic $y_i$.
- Exchangeability: $x_1 \perp \perp x_2 \perp \perp \ldots | h$
- $\Phi(i, j) := \mathbb{P}[x_m = i | y_m = j]$. 

Single topic model
- Each document has only one hidden topic: $y_i = h$.
- $h$ is a discrete variable and let $\lambda_i := \mathbb{P}[h = i]$. 

\[
\begin{array}{c}
x_1 \Phi \Phi \Phi \Phi \Phi \\
y_1 \Phi \Phi \Phi \Phi \Phi \\
\end{array}
\]
Form of Observed Moments

\[ \vec{\lambda} := [\mathbb{P}[h = i]]_i. \]

\[ \Phi(i, j) := \mathbb{P}[x_m = i | h = j]. \]

Learning: Loading matrix \( \Phi \) and Vector \( \vec{\lambda} \)
Form of Observed Moments

- $\vec{\lambda} := [\mathbb{P}[h = i]]_i$.
- $\Phi(i, j) := \mathbb{P}[x_m = i | h = j]$.

**Learning:** Loading matrix $\Phi$ and Vector $\vec{\lambda}$

**Pairwise Probability Matrix $M_2$**

$$M_2(a, b) := \mathbb{P}(x_1 = a, x_2 = b) = \sum_r \lambda_r \Phi(a, r) \Phi(b, r)$$
Form of Observed Moments

\[ \lambda := [P[h = i]]_i. \]
\[ \Phi(i, j) := P[x_m = i | h = j]. \]

Learning: Loading matrix $\Phi$ and Vector $\lambda$

Pairwise Probability Matrix $M_2$

\[ M_2(a, b) := P(x_1 = a, x_2 = b) = \sum_r \lambda_r \Phi(a, r) \Phi(b, r) \]

Triples Probability Tensor $M_3$

\[ M_3(a, b, c) := P(x_1 = a, x_2 = b, x_3 = c) = \sum_r \lambda_r \Phi(a, r) \Phi(b, r) \Phi(c, r) \]
Form of Observed Moments

- \( \vec{\lambda} := [\mathbb{P}[h = i]]_i \).

- \( \Phi(i, j) := \mathbb{P}[x_m = i | h = j] \).

Learning: Loading matrix \( \Phi \) and Vector \( \vec{\lambda} \)

Pairwise Probability Matrix \( M_2 \)

\[ M_2(a, b) := \mathbb{P}(x_1 = a, x_2 = b) = \sum_r \lambda_r \Phi(a, r) \Phi(b, r) \]

Triples Probability Tensor \( M_3 \)

\[ M_3(a, b, c) := \mathbb{P}(x_1 = a, x_2 = b, x_3 = c) = \sum_r \lambda_r \Phi(a, r) \Phi(b, r) \Phi(c, r) \]

Matrix and Tensor Forms: \( \phi_r := r^{th} \) column of \( \Phi \).

\[
M_2 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r. \\
M_3 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r \otimes \phi_r
\]
Tensor Basics: Multilinear Transformations

- For a tensor $M_3$, define (for matrices $V_i$ of appropriate dimensions)

$$[M_3(V_1, V_2, V_3)]_{i_1,i_2,i_3} := \sum_{j_1,j_2,j_3} (M_3)_{j_1,j_2,j_3} \prod_{m \in [3]} V_1(j_m, i_m)$$

- For a matrix $M_2$

$$M(V_1, V_2) := V_1^\top M_2 V_2.$$ 

$$M_3 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r \otimes \phi_r$$

$$M_3(W, W, W) = \sum_{r \in [k]} \lambda_r (W^\top \phi_r)^\otimes 3$$

$$M_3(I, v, v) = \sum_{r \in [k]} \lambda_r \langle v, \phi_r \rangle^2 \phi_r.$$ 

$$M_3(I, I, v) = \sum_{r \in [k]} \lambda_r \langle v, \phi_r \rangle \phi_r \phi_r^\top.$$
Inverse Moment Methods for Learning

$$M_2 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r, \quad M_3 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r \otimes \phi_r$$

Identifiability Using 2nd and 3rd Order Moments

Matrix $\Phi$ has linearly independent columns and $\bar{\lambda} > 0$. 
Inverse Moment Methods for Learning

\[
M_2 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r, \quad M_3 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r \otimes \phi_r
\]

Identifiability Using 2\textsuperscript{nd} and 3\textsuperscript{rd} Order Moments

Matrix Φ has linearly independent columns and \( \vec{\lambda} > 0 \).

Special Case: Orthogonality

- If Φ is an orthogonal matrix \( M_3(I, \phi_r, \phi_r) = \lambda_r \phi_r \).
- Loading vectors \( \{\phi_r\} \) are eigenvectors of the tensor \( M_3 \).
Inverse Moment Methods for Learning

\[ M_2 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r, \quad M_3 = \sum_{r=1}^{k} \lambda_r \phi_r \otimes \phi_r \otimes \phi_r \]

Identifiability Using 2\textsuperscript{nd} and 3\textsuperscript{rd} Order Moments

Matrix \( \Phi \) has linearly independent columns and \( \vec{\lambda} > 0 \).

Special Case: Orthogonality

- If \( \Phi \) is an orthogonal matrix, \( M_3(I, \phi_r, \phi_r) = \lambda_r \phi_r \).
- Loading vectors \( \{\phi_r\} \) are eigenvectors of the tensor \( M_3 \).

How to obtain an orthogonal tensor form?
Orthogonal Tensor Decomposition

\[ M_2 = \sum_{r \in [k]} \lambda_r \phi_r \otimes \phi_r, \quad M_3 = \sum_{r \in [k]} \lambda_r \phi_r \otimes \phi_r \otimes \phi_r \]

- Define \( W = UD^{-1} \), where \( M_2 = UDU^\top \).
- Let \( \tilde{\phi}_i := \sqrt{\lambda_i} W^\top \phi_i \). They are orthonormal.

\[ M_2(W,W) = \sum_{i \in [k]} W^\top (\sqrt{\lambda_i} \phi_i)(\sqrt{\lambda_i} \phi_i)^\top W = \sum_{i \in [k]} \tilde{\phi}_i \tilde{\phi}_i^\top = I, \]

- Now define \( \tilde{M}_3 \), so that

\[ \tilde{M}_3 = M_3(W,W,W) = \sum_{i \in [k]} \lambda_i (W^\top \phi_i) \otimes^3 = \sum_{i \in [k]} \frac{1}{\sqrt{\lambda_i}} \tilde{\phi}_i \otimes^3. \]

Learning: Tensor Decomposition of \( \tilde{M}_3 \)
Consider orthogonal symmetric tensor $T = \sum_i w_i \mu_i \otimes^3$

\[
T = \sum_{i=1}^{k} w_i \mu_i \otimes^3. \quad T(I, \mu_i, \mu_i) = w_i \mu_i
\]
Orthogonal Tensor Eigen Analysis

Consider orthogonal symmetric tensor $T = \sum_i w_i \mu_i \otimes^3$

\[
T = \sum_{i=1}^{k} w_i \mu_i \otimes^3. \quad T(I, \mu_i, \mu_i) = w_i \mu_i
\]

Obtaining eigenvectors through power iterations

\[
u \mapsto \frac{T(I, u, u)}{\|T(I, u, u)\|}
\]
Orthogonal Tensor Eigen Analysis

Consider orthogonal symmetric tensor $T = \sum_i w_i \mu_i \otimes 3$

$$T = \sum_{i=1}^{k} w_i \mu_i \otimes 3. \quad T(I, \mu_i, \mu_i) = w_i \mu_i$$

Obtaining eigenvectors through power iterations

$$u \to \frac{T(I, u, u)}{\|T(I, u, u)\|}$$

Challenges and Solution

- **Challenge:** Other eigenvectors present
- **Solution:** Only stable vectors are basis vectors $\{\mu_i\}$
Orthogonal Tensor Eigen Analysis

Consider orthogonal symmetric tensor $T = \sum_i w_i \mu_i \otimes^3$

$T = \sum_{i=1}^{k} w_i \mu_i \otimes^3$. $T(I, \mu_i, \mu_i) = w_i \mu_i$

Obtaining eigenvectors through power iterations

$u \mapsto \frac{T(I, u, u)}{\|T(I, u, u)\|}$

Challenges and Solution

- Challenge: Other eigenvectors present
  Solution: Only stable vectors are basis vectors $\{\mu_i\}$

- Challenge: empirical moments
  Solution: robust tensor decomposition methods
Optimization Viewpoint for Tensor Eigen Analysis

Consider Norm Optimization Problem for Tensor $T$

\[
\max_u T(u, u, u) \quad s.t. \ u^\top u = I
\]

- Constrained stationary fixed points $T(I, u, u) = \lambda u$ and $u^\top u = I$.
- $u$ is a local isolated maximizer if $w^\top (T(I, I, u) - \lambda I) w < 0$ for all $w$ such that $w^\top w = I$ and $w$ is orthogonal to $u$.

Review for Symmetric Matrices $M = \sum_i w_i \mu_i \otimes^2$

- Constrained stationary points are the eigenvectors
- Only top eigenvector is a maximizer and stable under power iterations

Orthogonal Symmetric Tensors $T = \sum_i w_i \mu_i \otimes^3$

- Stationary points are the eigenvectors (up to scaling)
- All basis vectors $\{\mu_i\}$ are local maximizers and stable under power iterations
Tensor Decomposition: Perturbation Analysis

- Observed tensor $\tilde{T} = T + E$, where $T = \sum_{i \in K} w_i \mu_i^3$ is orthogonal tensor and perturbation $E$, and $\|E\| \leq \epsilon$.

- Recall power iterations $u \mapsto \frac{\tilde{T}(I, u, u)}{\|\tilde{T}(I, u, u)\|}$.
Tensor Decomposition: Perturbation Analysis

- Observed tensor $\tilde{T} = T + E$, where $T = \sum_{i \in k} w_i \mu_i^3$ is orthogonal tensor and perturbation $E$, and $\|E\| \leq \epsilon$.

- Recall power iterations $u \mapsto \frac{\tilde{T}(I, u, u)}{\|\tilde{T}(I, u, u)\|}$

- "Good" initialization vector $\langle u^{(0)}, \mu_i \rangle = \Omega \left( \frac{\epsilon}{w_{\min}} \right)$
Tensor Decomposition: Perturbation Analysis

- Observed tensor $\tilde{T} = T + E$, where $T = \sum_{i \in k} w_i \mu_i \otimes 3$ is orthogonal tensor and perturbation $E$, and $\|E\| \leq \epsilon$.

- Recall power iterations $u \mapsto \tilde{T}(I, u, u) / \|\tilde{T}(I, u, u)\|

- “Good” initialization vector $\langle u^{(0)}, \mu_i \rangle = \Omega \left( \frac{\epsilon}{w_{\text{min}}} \right)

Perturbation Analysis

After $N$ iterations, eigen pair $(w_i, \mu_i)$ is estimated up to $O(\epsilon)$ error, where

$$N = O \left( \log k + \log \log \frac{w_{\text{max}}}{\epsilon} \right).$$

---

Robust Tensor Power Method

\[ \tilde{T} = \sum_i w_i \mu_i^{\otimes 3} + E \]

Basic Algorithm

- Pick random initialization vectors

- Run power iterations
  \[ u \leftarrow \frac{\tilde{T}(I, u, u)}{\|\tilde{T}(I, u, u)\|} \]

- Go with the winner, deflate and repeat
Robust Tensor Power Method

\[ \tilde{T} = \sum_i w_i \mu_i^\otimes 3 + E \]

Basic Algorithm

- Pick random initialization vectors
- Run power iterations
  \[ u \mapsto \frac{\tilde{T}(I, u, u)}{\|\tilde{T}(I, u, u)\|} \]
- Go with the winner, deflate and repeat

Further Improvements

- Initialization: Use long document vectors for initialization
- Stabilization:
  \[ u^{(t)} \mapsto \alpha \frac{\tilde{T}(I, u^{(t-1)}, u^{(t-1)})}{\|\tilde{T}(I, u^{(t-1)}, u^{(t-1)})\|} + (1 - \alpha) u^{(t-1)} \]

Efficient Learning Through Tensor Power Iterations
Extensions...

Latent Dirichlet Allocation

- Each document a topic mixture rather than a single topic
- Modified second and third order moments reduce to symmetric tensor.
Extensions…

Latent Dirichlet Allocation
- Each document a topic mixture rather than a single topic
- Modified second and third order moments reduce to symmetric tensor.

Spherical Gaussian Mixtures, Hidden Markov Models, Independent Component Analysis (ICA) …
Latent Dirichlet Allocation
- Each document a topic mixture rather than a single topic
- Modified second and third order moments reduce to symmetric tensor.

Spherical Gaussian Mixtures, Hidden Markov Models, Independent Component Analysis (ICA) …

Community Modeling and Detection in Social Networks
- Mixed membership model (Airoldi et. al): overlapping communities
- Edge counts and 3-star counts: tensor decomposition

---

A. Anandkumar, R. Ge, D. Hsu, S. Kakade, ” Learning Mixed Membership Block Models.”
## Preliminary Experiments

### Top 10 words for 5 topics (NYTimes data)

<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
<th>Topic 4</th>
<th>Topic 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>bush</td>
<td>company</td>
<td>show</td>
<td>official</td>
<td>team</td>
</tr>
<tr>
<td>president</td>
<td>percent</td>
<td>book</td>
<td>water</td>
<td>game</td>
</tr>
<tr>
<td>government</td>
<td>million</td>
<td>women</td>
<td>attack</td>
<td>season</td>
</tr>
<tr>
<td>official</td>
<td>companies</td>
<td>family</td>
<td>u_s</td>
<td>player</td>
</tr>
<tr>
<td>campaign</td>
<td>market</td>
<td>film</td>
<td>food</td>
<td>play</td>
</tr>
<tr>
<td>political</td>
<td>business</td>
<td>school</td>
<td>united_states</td>
<td>games</td>
</tr>
<tr>
<td>law</td>
<td>stock</td>
<td>look</td>
<td>afghanistan</td>
<td>point</td>
</tr>
<tr>
<td>leader</td>
<td>billion</td>
<td>home</td>
<td>taliban</td>
<td>run</td>
</tr>
<tr>
<td>george_bush</td>
<td>money</td>
<td>children</td>
<td>air</td>
<td>win</td>
</tr>
<tr>
<td>al_gore</td>
<td>cost</td>
<td>friend</td>
<td>military</td>
<td>won</td>
</tr>
</tbody>
</table>
Outline

1. Introduction

2. Inverse Moment Methods
   - Moment Tensor Form
   - Tensor Decomposition Methods

   - Latent Tree Models
   - Loopy Latent Models

4. Experiments and Applications

5. Conclusion
Hierarchical Latent Variable Models

So far...

Latent Tree

Loopy Model

Graph Estimation with Latent Variables

- # and location of hidden variables unknown
- Estimate graph over all variables
- Trees and girth-constrained graphs
Learning Latent Tree Models

\[ h_1 \]
\[ h_2 \]
\[ h_3 \]
Information Distances $\{d_{ij}\}$

- Gaussian: $d_{ij} := -\log |\rho_{ij}|$.
- Discrete: $d_{ij} := -\log |\text{Det}(P_{i,j})|$.
Learning Latent Tree Models

Information Distances \( \{d_{ij}\} \)

- **Gaussian:** \( d_{ij} := - \log |\rho_{ij}| \).
- **Discrete:** \( d_{ij} := - \log |\text{Det}(P_{i,j})| \).

\( [d_{i,j}] \) is an additive tree metric:

\[
d_{k,l} = \sum_{(i,j) \in \text{Path}(k,l;E)} d_{i,j}.
\]
Learning Latent Tree Models

Information Distances \( \{d_{ij}\} \)

- Gaussian: \( d_{ij} := -\log |\rho_{ij}|. \)
- Discrete: \( d_{ij} := -\log |\text{Det}(P_{i,j})|. \)

\([d_{i,j}]\) is an additive tree metric:

\[
d_{k,l} = \sum_{(i,j) \in \text{Path}(k,l;E)} d_{i,j}.
\]

Learning latent tree using \( [\hat{d}_{i,j}] \)
Siblings Test Based on Information Distances

Exact Statistics: Distances \([d_{i,j}]\)

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

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

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

Use only short distances: \(d_{i,k}, d_{j,k} < \tau\), Relax equality relationships
Siblings Test Based on Information Distances

Exact Statistics: Distances \([d_{i,j}]\)

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

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

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

Use only short distances: \(d_{i,k}, d_{j,k} < \tau\), Relax equality relationships
Siblings Test Based on Information Distances

Exact Statistics: Distances \([d_{i,j}]\)

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

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

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

Use only short distances: \(d_{i,k}, d_{j,k} < \tau\), Relax equality relationships
Siblings Test Based on Information Distances

Exact Statistics: Distances \([d_{i,j}]\)

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

- \(-d_{i,j} < \Phi_{ijk} = \Phi_{ijk'} < d_{i,j} \ \forall \ k, k' \neq i, j, \iff i, j \) leaves with common parent

- \(\Phi_{ijk} = d_{i,j}, \ \forall \ k \neq i, j, \iff i \) is a leaf and \(j \) is its parent.

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

Use only short distances: \(d_{i,k}, d_{j,k} < \tau\), Relax equality relationships
Siblings Test Based on Information Distances

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

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

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

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

Use only short distances: $d_{i,k}, d_{j,k} < \tau$, Relax equality relationships
Siblings Test Based on Information Distances

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

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

- $-d_{i,j} < \Phi_{ijk} = \Phi_{ijk} ' < d_{i,j} \quad \forall \, k, k' \neq i, j, \quad \iff \quad i, j \text{ leaves with common parent}$

- $\Phi_{ijk} = d_{i,j}, \quad \forall \, k \neq i, j, \quad \iff \quad i \text{ is a leaf and } j \text{ is its parent}.$

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

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

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

- Sibling test and remove leaves
- Build tree from bottom up
Recursive Grouping Algorithm (Choi, Tan, A., Willsky)

- Sibling test and remove leaves
- Build tree from bottom up
Recursive Grouping Algorithm (Choi, Tan, A., Willsky)

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

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

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

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

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

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

- Sibling test and remove leaves
- Build tree from bottom up
Chow-Liu Based Grouping Algorithm

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$. 
Chow-Liu Based Grouping Algorithm

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

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

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

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

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

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

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

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

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

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

Efficient Initial Tree on Observed Nodes (MST)
Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

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

Efficient Initial Tree on Observed Nodes (MST)

Minimum spanning tree using edge weights $[\hat{d}_{i,j}]$.

Chow-Liu Based Grouping (Choi, Tan, A., Willsky ‘11)
Proof Ideas

Relating Chow-Liu Tree with Latent Tree

- Surrogate $Sg(i)$ for node $i$: observed node with strongest correlation
  \[ Sg(i) := \arg\min_{j \in V} d_{i,j} \]

- Neighborhood preservation
  \[ (i, j) \in T \Rightarrow (Sg(i), Sg(j)) \in T_{ML}. \]

Chow-Liu grouping reverses edge contractions

Proof by induction
Loopy Graphical Models with Latent Nodes

Motivation: Topic Models
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.

Overview of Proposed Method
- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods
Motivation: Topic Models

- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.

Overview of Proposed Method

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

- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.

Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods
Motivation: Topic Models
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.

Overview of Proposed Method
- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods
Loopy Graphical Models with Latent Nodes

Motivation: Topic Models

- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.

Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods
Loopy Graphical Models with Latent Nodes

Motivation: Topic Models
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.

Overview of Proposed Method
- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods
Motivation: Topic Models
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.

Overview of Proposed Method
- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods
Loopy Graphical Models with Latent Nodes

Motivation: Topic Models
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.

Overview of Proposed Method
- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods
Loopy Graphical Models with Latent Nodes

Motivation: Topic Models

- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.

Overview of Proposed Method

- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods
Motivation: Topic Models
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.

Overview of Proposed Method
- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods
Loopy Graphical Models with Latent Nodes

Motivation: Topic Models
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.

Overview of Proposed Method
- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods
Motivation: Topic Models
- Common words among topics.
- Latent or hidden nodes.
- Typically long cycles: Locally tree-like.

Overview of Proposed Method
- Consider local neighborhoods for building local MST
- Merge the MSTs to obtain a loopy graph
- Run latent tree routine on different local neighborhoods
Guarantees for Latent Structure Learning

- Ising model with minimum edge potential $J_{\min}$.

\[
p(x) \propto \exp \left[ \sum_{(i,j) \in G} J_{i,j} x_i x_j + \sum_{i \in V} h_i x_i \right]
\]

- Depth $\delta$: worst-case distance between hidden and observed nodes.
- Parameter $\beta$: depends on min. and max. node and edge potentials
  - $\beta = 1$ for homogeneous models.
Guarantees for Latent Structure Learning

- Ising model with minimum edge potential $J_{\text{min}}$.

$$p(x) \propto \exp \left[ \sum_{(i,j) \in G} J_{i,j} x_i x_j + \sum_{i \in V} h_i x_i \right]$$

- Depth $\delta$: worst-case distance between hidden and observed nodes.
- Parameter $\beta$: depends on min. and max. node and edge potentials
  - $\beta = 1$ for homogeneous models.

---

**Theorem (A., Valluvan ‘12)**

Proposed method correctly recovers graph structure w.h.p. on $p$ observed nodes and $n$ samples when

$$\frac{J_{\text{min}}^{-2\delta\beta(\beta+1)-2} \log p}{n} = O(1).$$

---

Insights and Implications

Tradeoff between depth $\delta$ and girth $g$

Roughly require: $\delta < g/4$.

Tradeoff between max. edge strength $J_{\text{max}}$ and degree $\Delta$

Require $J_{\text{max}} < \text{atanh}(\Delta^{-1})$. 
Insights and Implications

Tradeoff between depth $\delta$ and girth $g$

Roughly require: $\delta < g/4$.

Tradeoff between max. edge strength $J_{\text{max}}$ and degree $\Delta$

Require $J_{\text{max}} < \tanh(\Delta^{-1})$.

Sample complexity for uniform node sampling

Given $\rho$ fraction of nodes as observed nodes,

$$n = \Omega \left( \Delta^2 \rho^{-4} (\log p)^5 \right).$$

Necessary conditions for structure recovery

For any deterministic algorithm, the number of samples $n$ needs to be

$$n = \Omega \left( \frac{\Delta_{\text{min}}}{\rho} \log p \right).$$
Insights and Implications

Tradeoff between depth $\delta$ and girth $g$

Roughly require: $\delta < g/4$.

Tradeoff between max. edge strength $J_{\text{max}}$ and degree $\Delta$

 Require $J_{\text{max}} < \text{atanh}(\Delta^{-1})$.

Sample complexity for uniform node sampling

Given $\rho$ fraction of nodes as observed nodes, 

$$n = \Omega \left( \Delta^2 \rho^{-4} (\log p)^5 \right).$$

Necessary conditions for structure recovery

For any deterministic algorithm, the number of samples $n$ needs to be 

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

Efficient Method for Learning Loopy Latent Models
Outline

1. Introduction

2. Inverse Moment Methods
   - Moment Tensor Form
   - Tensor Decomposition Methods

   - Latent Tree Models
   - Loopy Latent Models

4. Experiments and Applications

5. Conclusion
Discovering Word Relationships
Discovering Word Relationships
Discovering Word Relationships
Discovering Word Relationships
Discovering Word Relationships
Discovering Word Relationships
Dynamic Network Modeling

- Observations: series of graph $G_t = (V_t, E_t)$ and covariates
- Modeling vertex participation through latent graphical model
- Logistic regression for edge prediction given vertices
- Data: windsurfer interaction on a beach
- Improvement over baseline: 164% for vertices and 45% for edges.

Modeling Hazard-related Tweets

In collaboration with Furong Huang and Carter Butts at UCI
Modeling Gene Associations

- Observed: gene expression levels
- Relationships between genes, e.g. genes that encode ribosomal subunits group together
- Hidden nodes: regulators that control groups of functionally similar genes, e.g. transcription factors

In collaboration with Anthony Gitter (Microsoft) and Ernest Fraenkel (MIT)
Outline

1. Introduction

2. Inverse Moment Methods
   - Moment Tensor Form
   - Tensor Decomposition Methods

   - Latent Tree Models
   - Loopy Latent Models

4. Experiments and Applications

5. Conclusion
Summary on Learning Latent Variable Models

Tensor Methods
- Tensor forms for a range of models
- Efficient decomposition methods
- Perturbation analysis

Graph Estimation
- Latent modeling via graphical approaches
- Efficient methods for graph estimation
- Guarantees on sample and computational complexities
The Big Picture

High-dimensional Latent Variable Modeling

- Method of moments
- Algorithms and complexity
- Statistical physics
- Tensor analysis
- Information theory
- Graphical Models

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