Large-Deviations and Applications for Learning Tree-Structured Graphical Models

Vincent Tan

Stochastic Systems Group,
Lab of Information and Decision Systems,
Massachusetts Institute of Technology

The following is joint work with:

- Alan Willsky (MIT)
- Lang Tong (Cornell)
- Animashree Anandkumar (UC Irvine)
- John Fisher (MIT)
- Sujay Sanghavi (UT Austin)
- Matt Johnson (MIT)
1 Motivation, Background and Main Contributions
Outline

1. Motivation, Background and Main Contributions

2. Learning Discrete Trees Models: Error Exponent Analysis
1. Motivation, Background and Main Contributions
2. Learning Discrete Trees Models: Error Exponent Analysis
3. Learning Gaussian Trees Models: Extremal Structures
1 Motivation, Background and Main Contributions
2 Learning Discrete Trees Models: Error Exponent Analysis
3 Learning Gaussian Trees Models: Extremal Structures
4 Learning High-Dimensional Forest-Structured Models
1. Motivation, Background and Main Contributions

2. Learning Discrete Trees Models: Error Exponent Analysis

3. Learning Gaussian Trees Models: Extremal Structures

4. Learning High-Dimensional Forest-Structured Models

5. Related Topics and Conclusion
1. Motivation, Background and Main Contributions
2. Learning Discrete Trees Models: Error Exponent Analysis
3. Learning Gaussian Trees Models: Extremal Structures
4. Learning High-Dimensional Forest-Structured Models
5. Related Topics and Conclusion
Motivation: A Real-Life Example

- Manchester Asthma and Allergy Study (MAAS)
- More than $n \approx 1000$ children
- Number of variables $d \approx 10^6$
  - Environmental, Physiological and Genetic (SNP)
Motivation: A Real-Life Example

- Manchester Asthma and Allergy Study (MAAS)
- More than $n \approx 1000$ children
- Number of variables $d \approx 10^6$
  - Environmental, Physiological and Genetic (SNP)

www.maas.org.uk
Motivation: Modeling Large Datasets I

- How do we model such data to make useful inferences?

Motivation: Modeling Large Datasets I

- How do we model such data to make useful inferences?
- Model the relationships between variables by a sparse graph
- Reduce the number of interdependencies between the variables

Motivation: Modeling Large Datasets

- How do we model such data to make useful inferences?
- Model the relationships between variables by a sparse graph
- Reduce the number of interdependencies between the variables

Reduce the dimensionality of the covariates (features) for predicting a variable for interest (e.g., asthma)

Information-theoretic limits†?


Reduce the dimensionality of the covariates (features) for predicting a variable for interest (e.g., asthma)

Information-theoretic limits†?

Learning graphical models tailored specifically for hypothesis testing

Can we learn better models in the finite-sample setting‡?


Graphical Models: Introduction

- **Graph structure** $G = (V, E)$ represents a multivariate distribution of a random vector $\mathbf{X} = (X_1, \ldots, X_d)$ indexed by $V = \{1, \ldots, d\}$

- Node $i \in V$ corresponds to random variable $X_i$

- Edge set $E$ corresponds to conditional independencies
Graphical Models: Introduction

- **Graph structure** $G = (V, E)$ represents a multivariate distribution of a random vector $X = (X_1, \ldots, X_d)$ indexed by $V = \{1, \ldots, d\}$
- Node $i \in V$ corresponds to random variable $X_i$
- Edge set $E$ corresponds to conditional independencies

\[ X_i \perp \perp X_{V \setminus \{\text{nbd}(i) \cup i\}} \mid X_{\text{nbd}(i)} \]
\[ X_A \perp \perp X_B \mid X_S \]
Hammersley-Clifford Theorem (1971)

Let $P$ be the joint pmf of graphical model Markov on $G = (V, E)$:

$$P(x) = \frac{1}{Z} \exp \left[ \sum_{c \in C} \Psi_c(x_c) \right]$$
Hammersley-Clifford Theorem (1971)

Let $P$ be the joint pmf of graphical model Markov on $G = (V, E)$:

$$P(x) = \frac{1}{Z} \exp \left[ \sum_{c \in C} \Psi_c (x_c) \right]$$
Hammersley-Clifford Theorem (1971)
Let $P$ be the joint pmf of graphical model Markov on $G = (V, E)$:

$$P(x) = \frac{1}{Z} \exp \left[ \sum_{c \in C} \Psi_c(x_c) \right]$$

where $C$ is the set of maximal cliques.
Tree-structured Graphical Models

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

\[ = P_1(x_1) \frac{P_{1,2}(x_1, x_2)}{P_1(x_1)} \frac{P_{1,3}(x_1, x_3)}{P_1(x_1)} \frac{P_{1,4}(x_1, x_4)}{P_1(x_1)} \]
Tree-structured Graphical Models: Tractable Learning and Inference

- Maximum-Likelihood learning of tree structure is tractable
  - **Chow-Liu Algorithm** (1968)

Mathematical expression:

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

\[
= P_1(x_1) \frac{P_{1,2}(x_1, x_2)}{P_1(x_1)} \frac{P_{1,3}(x_1, x_3)}{P_1(x_1)} \frac{P_{1,4}(x_1, x_4)}{P_1(x_1)}
\]
Tree-Structured Graphical Models

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

\[
= P_1(x_1) \frac{P_{1,2}(x_1, x_2)}{P_1(x_1)} \frac{P_{1,3}(x_1, x_3)}{P_1(x_1)} \frac{P_{1,4}(x_1, x_4)}{P_1(x_1)}
\]

Tree-structured Graphical Models: Tractable Learning and Inference

- Maximum-Likelihood learning of tree structure is tractable
  - **Chow-Liu Algorithm** (1968)

- Inference on Trees is tractable
  - **Sum-Product Algorithm**
Tree-Structured Graphical Models: Tractable Learning and Inference

- Maximum-Likelihood learning of tree structure is tractable
  - Chow-Liu Algorithm (1968)

- Inference on Trees is tractable
  - Sum-Product Algorithm

Which other classes of graphical models are tractable for learning?
Main Contributions in Thesis: I

Error Exponent Analysis of Tree Structure Learning (Ch. 3 and 4)
Main Contributions in Thesis: I

Error Exponent Analysis of Tree Structure Learning (Ch. 3 and 4)

High-Dimensional Structure Learning for Forest Models (Ch. 5)
Learning Graphical Models for Hypothesis Testing (Ch. 6)

- Devised algorithms for learning trees for hypothesis testing
Learning Graphical Models for Hypothesis Testing (Ch. 6)

- Devised algorithms for learning trees for hypothesis testing

Information-Theoretic Limits for Salient Subset Recovery (Ch. 7)

- Devised necessary and sufficient conditions for estimating of salient set of features
Main Contributions in Thesis: II

Learning Graphical Models for Hypothesis Testing (Ch. 6)
- Devised algorithms for learning trees for hypothesis testing

Information-Theoretic Limits for Salient Subset Recovery (Ch. 7)
- Devised necessary and sufficient conditions for estimating of salient set of features

We will focus on Chapters 3 - 5 here. See thesis for Chapters 6 and 7.
Outline

1. Motivation, Background and Main Contributions

2. Learning Discrete Trees Models: Error Exponent Analysis

3. Learning Gaussian Trees Models: Extremal Structures

4. Learning High-Dimensional Forest-Structured Models

5. Related Topics and Conclusion
ML learning of tree structure given i.i.d. $\mathcal{X}^d$-valued samples

When does the error probability decay exponentially?

What is the exact rate of decay of the probability of error?

How does the error exponent depend on the parameters and structure of the true distribution?
ML learning of tree structure given i.i.d. $\mathcal{X}^d$-valued samples

When does the error probability decay exponentially?
What is the exact rate of decay of the probability of error?
How does the error exponent depend on the parameters and structure of the true distribution?
ML learning of **tree structure** given i.i.d. $\mathcal{X}^d$-valued samples

$P^n(\text{err}) \doteq \exp(-n \text{ Rate})$

When does the error probability decay exponentially?
What is the exact rate of decay of the probability of error?
How does the error exponent depend on the parameters and structure of the true distribution?
Motivation

ML learning of tree structure given i.i.d. \( \mathcal{X}^d \)-valued samples

\[
P^n(\text{err}) = \exp(-n \text{ Rate})
\]

- When does the error probability decay exponentially?
ML learning of tree structure given i.i.d. $\mathcal{X}^d$-valued samples

$P^n(\text{err}) \overset{\text{def}}{=} \exp(-n \text{ Rate})$

- When does the error probability decay exponentially?
- What is the exact rate of decay of the probability of error?
Motivation

ML learning of tree structure given i.i.d. $\mathcal{X}^d$-valued samples

When does the error probability decay exponentially?
What is the exact rate of decay of the probability of error?
How does the error exponent depend on the parameters and structure of the true distribution?
Main Contributions

- Discrete case:
  - Provide the exact rate of decay for a given $P$
  - Rate of decay $\approx$ SNR for learning
Main Contributions

- **Discrete case:**
  - Provide the exact rate of decay for a given $P$
  - Rate of decay $\approx$ SNR for learning

- **Gaussian case:**
  - Extremal structures: Star (worst) and chain (best) for learning
Related Work in Structure Learning

- **ML for trees:** Max-weight spanning tree with mutual information edge weights (Chow & Liu 1968)

- **Causal dependence trees:** directed mutual information (Quinn, Coleman & Kiyavash 2010)

- **Convex relaxation methods:** $\ell_1$ regularization
  - Gaussian graphical models (Meinshausen and Buehlmann 2006)
  - Logistic regression for Ising models (Ravikumar et al. 2010)

- Learning **thin junction trees** through conditional mutual information tests (Chechetka et al. 2007)

- **Conditional independence tests** for bounded degree graphs (Bresler et al. 2008)
Related Work in Structure Learning

- **ML for trees**: Max-weight spanning tree with mutual information edge weights (Chow & Liu 1968)

- **Causal dependence trees**: directed mutual information (Quinn, Coleman & Kiyavash 2010)

- **Convex relaxation methods**: $\ell_1$ regularization
  - Gaussian graphical models (Meinshausen and Buehlmann 2006)
  - Logistic regression for Ising models (Ravikumar et al. 2010)

- **Learning thin junction trees** through conditional mutual information tests (Chechetka et al. 2007)

- **Conditional independence tests** for bounded degree graphs (Bresler et al. 2008)

We obtain and analyze error exponents for the ML learning of trees (and extensions to forests)
Samples $x^n = \{x_1, \ldots, x_n\}$ drawn i.i.d. from $P \in \mathcal{P}(\mathcal{X}^d)$, $\mathcal{X}$ is finite
Samples $x^n = \{x_1, \ldots, x_n\}$ drawn i.i.d. from $P \in \mathcal{P}(\mathcal{X}^d)$, with $\mathcal{X}$ finite

- Solve the ML problem given the data $x^n$

$$P_{ML} \triangleq \arg\max_{Q \in \text{Trees}} \frac{1}{n} \sum_{k=1}^{n} \log Q(x_k)$$
ML Learning of Trees (Chow-Liu) I

Samples $x^n = \{x_1, \ldots, x_n\}$ drawn i.i.d. from $P \in \mathcal{P}(\mathcal{X}^d)$, $\mathcal{X}$ is finite

- Solve the ML problem given the data $x^n$

$$P_{ML} \triangleq \operatorname*{argmax}_{Q \in \text{Trees}} \frac{1}{n} \sum_{k=1}^{n} \log Q(x_k)$$

- Denote $\hat{P}(a) = \hat{P}(a; x^n)$ as the empirical distribution of $x^n$
Samples $x^n = \{x_1, \ldots, x_n\}$ drawn i.i.d. from $P \in \mathcal{P}(X^d)$, $X$ is finite

- Solve the ML problem given the data $x^n$

$$P_{ML} \triangleq \arg\max_{Q \in \text{Trees}} \frac{1}{n} \sum_{k=1}^{n} \log Q(x_k)$$

- Denote $\hat{P}(a) = \hat{P}(a; x^n)$ as the empirical distribution of $x^n$

- Reduces to a max-weight spanning tree problem (Chow-Liu 1968)

$$E_{ML} = \arg\max_{E_Q \in \text{Trees}} \sum_{e \in E_Q} I(\hat{P}_e)$$

- $\hat{P}_e$ is the marginal of the empirical on $e = (i,j)$

- $I(\hat{P}_e)$ is the mutual information of the empirical $\hat{P}_e$
True MI \{ I(P_e) \} 

Max-weight spanning tree \[ \mathcal{E} \]

Empirical MI \{ I(\hat{P}_e) \} from \[ \mathbb{E} \]

\[ \mathcal{E}_{\text{ML}} \neq \mathcal{E}_{P} \]
ML Learning of Trees (Chow-Liu) II

True MI \( \{I(P_e)\} \)

Max-weight spanning tree \( E_P \)

\[ \begin{align*}
X_3 & \quad 1 \quad X_2 \\
X_1 & \quad 5 \quad 6 \\
X_4 & \quad 3 \quad 2 \\
X_4 & \quad 4 \\
X_2 & \quad 5 \quad 6 \\
X_1 & \quad 4 \\
X_4 &
\end{align*} \]
True MI $\{I(P_e)\}$

Max-weight spanning tree $E_P$
ML Learning of Trees (Chow-Liu) II

True MI \( \{I(P_\epsilon)\} \)

Empirical MI \( \{I(\hat{P}_\epsilon)\} \) from \( x^n \)

Max-weight spanning tree \( E_P \)

Max-weight spanning tree \( E_{ML} \neq E_P \)
Define $P_{ML}$ to be ML tree-structured distribution with edge set $E_{ML}$ and the error event is $\{E_{ML} \neq E_{P}\}$
Define $P_{\text{ML}}$ to be ML tree-structured distribution with edge set $E_{\text{ML}}$ and the error event is $\{E_{\text{ML}} \neq E_P\}$.
Problem Statement

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

- Find the error exponent $K_P$:

$$K_P \triangleq \lim_{n \to \infty} -\frac{1}{n} \log P^n (E_{ML} \neq E_P)$$
Problem Statement

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

- Find the error exponent $K_P$:

$$K_P \triangleq \lim_{n \to \infty} -\frac{1}{n} \log P^n (E_{\text{ML}} \neq E_P) \quad P^n (E_{\text{ML}} \neq E_P) \doteq \exp(-nK_P)$$
Define $P_{\text{ML}}$ to be ML tree-structured distribution with edge set $E_{\text{ML}}$ and the error event is $\{E_{\text{ML}} \neq E_{P}\}$.

Find the error exponent $K_P$:

$$K_P \triangleq \lim_{n \to \infty} -\frac{1}{n} \log P^n (E_{\text{ML}} \neq E_{P}) \quad P^n (E_{\text{ML}} \neq E_{P}) \doteq \exp(-nK_P)$$

Naïvely, what could we do to compute $K_P$?
Problem Statement

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

- Find the error exponent $K_P$:

$$K_P \triangleq \lim_{n \to \infty} -\frac{1}{n} \log P^n (E_{ML} \neq E_P) \quad P^n (E_{ML} \neq E_P) \approx \exp(-nK_P)$$

- Naively, what could we do to compute $K_P$? I-projections onto all trees?
The Crossover Rate I

Correct Structure

<table>
<thead>
<tr>
<th>True MI $I(P_e)$</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emp MI $\hat{I}(P_e)$</td>
<td>6.2</td>
<td>5.6</td>
<td>4.5</td>
<td>3.2</td>
<td>2.2</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Vincent Tan (MIT)
### Correct Structure

<table>
<thead>
<tr>
<th>True MI $I(P_e)$</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emp MI $I(\hat{P}_e)$</td>
<td>6.2</td>
<td>5.6</td>
<td>4.5</td>
<td>2.8</td>
<td>2.2</td>
<td>1.1</td>
</tr>
</tbody>
</table>

### Incorrect Structure!

<table>
<thead>
<tr>
<th>True MI $I(P_e)$</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emp MI $I(\hat{P}_e)$</td>
<td>6.3</td>
<td>4.9</td>
<td>3.5</td>
<td>3.6</td>
<td>2.2</td>
<td>1.1</td>
</tr>
</tbody>
</table>
The Crossover Rate I

Correct Structure

<table>
<thead>
<tr>
<th>True MI $I(P_e)$</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emp MI $\hat{I}(P_e)$</td>
<td>6.2</td>
<td>5.6</td>
<td>4.5</td>
<td>2.8</td>
<td>2.2</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Incorrect Structure!

<table>
<thead>
<tr>
<th>True MI $I(P_e)$</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emp MI $\hat{I}(P_e)$</td>
<td>6.3</td>
<td>4.9</td>
<td>3.5</td>
<td>3.6</td>
<td>2.2</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Structure Unaffected

<table>
<thead>
<tr>
<th>True MI $I(P_e)$</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emp MI $\hat{I}(P_e)$</td>
<td>5.5</td>
<td>5.6</td>
<td>4.5</td>
<td>3.0</td>
<td>2.2</td>
<td>1.1</td>
</tr>
</tbody>
</table>
Given two node pairs \( e, e' \in (V^2) \) with joint distribution \( P_e, P_{e'} \in P(X^4) \), s.t.
\[
I(P_e) > I(P_{e'}). 
\]
Consider the crossover event of the empirical MI:
\[
\{ I(\hat{P}_e) \leq I(\hat{P}_{e'}) \} 
\]
\[
\text{Def: Crossover Rate} \quad J_e, e' \equiv \lim_{n \to \infty} -\frac{1}{n} \log P_n( I(\hat{P}_e) \leq I(\hat{P}_{e'}) ) 
\]
Given two node pairs $e, e' \in \binom{V}{2}$ with joint distribution $P_{e,e'} \in \mathcal{P}(\mathcal{X}^4)$, s.t.

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

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

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

Consider the \textit{crossover event} of the empirical MI

$$\{I(\hat{P}_e) \leq I(\hat{P}_{e'})\}$$
The Crossover Rate I

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

\[
I(P_e) > I(P_{e'}). 
\]

Consider the crossover event of the empirical MI

\[
\{ I(\hat{P}_e) \leq I(\hat{P}_{e'}) \} 
\]

Def: Crossover Rate

\[
J_{e,e'} \triangleq \lim_{n \to \infty} - \frac{1}{n} \log P^n \left( I(\hat{P}_e) \leq I(\hat{P}_{e'}) \right) 
\]
Proposition

The crossover rate for empirical mutual informations is

\[ J_e, e' = \min_{Q \in P(X^d)} \{ D(Q \| P_e, e') : I(Q_e) = I(Q'_e) \} \]

\[ \{ I(\hat{P}_e) \leq I(\hat{P}_e') \} \]
Proposition

The crossover rate for empirical mutual informations is

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

The crossover rate for empirical mutual informations is

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

where \( e \) and \( e' \) are empirical distributions.
Proposition

The crossover rate for empirical mutual informations is

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

\[ \mathcal{P}(\mathcal{X}^4) \quad \bullet \quad P_{e,e'} \]
Proposition

The crossover rate for empirical mutual informations is

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

\[ \mathcal{P}(X^4) \]

\[ P_{e,e'} \]
Proposition

The crossover rate for empirical mutual informations is

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

The **crossover rate** for empirical mutual informations is

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

- I-projection (Csiszár)
- Sanov’s Theorem
- Exact but not intuitive
- Non-Convex
How to calculate the error exponent $K_P$ with the crossover rates $J_{e,e'}$?

Easy only in some very special cases

"Star" graph with $I(Q_a) > I(Q_b) > 0$

There is a unique crossover rate

The unique crossover rate is $K_P = \min_{R \in P(X)} \left\{ D(R|Q_a,b) : I(R_e) = I(R_{e'}) \right\}$
How to calculate the error exponent $K_P$ with the crossover rates $J_{e,e'}$?

Easy only in some very special cases

- “Star” graph with $I(Q_a) > I(Q_b) > 0$
- There is a unique crossover rate
- The unique crossover rate is the error exponent
How to calculate the error exponent $K_P$ with the crossover rates $J_{e,e'}$?

Easy only in some very special cases

- “Star” graph with $I(Q_a) > I(Q_b) > 0$
- There is a unique crossover rate
- The unique crossover rate is the error exponent
How to calculate the error exponent $K_P$ with the crossover rates $J_{e,e'}$?

Easy only in some very special cases

- “Star” graph with $I(Q_a) > I(Q_b) > 0$
- There is a unique crossover rate
- The unique crossover rate is the error exponent
How to calculate the error exponent $K_P$ with the crossover rates $J_{e,e'}$?

Easy only in some very special cases

- "Star" graph with $I(Q_a) > I(Q_b) > 0$
- There is a unique crossover rate
- The unique crossover rate is the error exponent

$$K_P = \min_{R \in \mathcal{P}(X^4)} \left\{ D(R \| Q_{a,b}) : I(R_e) = I(R_{e'}) \right\}$$
A large deviation is done in the least unlikely of all unlikely ways.

“Large deviations” by F. Den Hollander
A large deviation is done in the least unlikely of all unlikely ways.

“Large deviations” by F. Den Hollander

\[ T_P \in \mathcal{T} \]
A large deviation is done in the least unlikely of all unlikely ways.

– “Large deviations” by F. Den Hollander
A large deviation is done in the least unlikely of all unlikely ways.

– “Large deviations” by F. Den Hollander
A large deviation is done in the least unlikely of all unlikely ways.

– “Large deviations” by F. Den Hollander
A large deviation is done in the least unlikely of all unlikely ways.

– “Large deviations” by F. Den Hollander

\[ e' \notin E_P \]

\[ \text{Path}(e'; E_P) \]

\[ T'_P \neq T_P \]

\[ \text{dominates} \]
A large deviation is done in the least unlikely of all unlikely ways.

- “Large deviations” by F. Den Hollander

Theorem (Error Exponent)

\[ K_P = \min_{e' \not\in E_P} \left( \min_{e \in \text{Path}(e'; E_P)} J_{e,e'} \right) \]
\[ P^n (E_{ML} \neq E_P) = \exp \left[ -n \min_{e' \notin E_P} \left( \min_{e \in \text{Path}(e'; E_P)} J_{e,e'} \right) \right] \]
\( P^n (E_{ML} \neq E_P) \doteq \exp \left[ -n \min_{e' \notin E_P} \left( \min_{e \in \text{Path}(e'; E_P)} J_{e,e'} \right) \right] \)

We have a finite-sample result too! See thesis
\[ P^n (E_{ML} \neq E_P) \overset{\cdot}{=} \exp \left[ -n \min_{e' \notin E_P} \left( \min_{e \in \text{Path}(e';E_P)} J_{e,e'} \right) \right] \]

We have a \textit{finite-sample} result too! See thesis

**Proposition**

\textit{The following statements are equivalent:}

(a) \textit{The error probability decays exponentially, i.e., } K_P > 0

(b) \textit{T_P is a connected tree, i.e., not a proper forest}
\[ P^n (E_{ML} \neq E_P) = \exp \left[ -n \min_{e' \notin E_P} \min_{e \in \text{Path}(e'; E_P)} J_{e,e'} \right] \]

We have a finite-sample result too! See thesis

**Proposition**

*The following statements are equivalent:*

**(a)** *The error probability decays exponentially, i.e., \( K_P > 0 \)*

**(b)** *\( T_P \) is a connected tree, i.e., not a proper forest***
$$P^n (E_{ML} \neq E_P) \doteq \exp \left[ -n \min_{e' \notin E_P} \left( \min_{e \in \text{Path}(e';E_P)} J_{e,e'} \right) \right]$$

We have a finite-sample result too! See thesis.

**Proposition**

The following statements are equivalent:

(a) The error probability **decays exponentially**, i.e., $K_P > 0$

(b) $T_P$ is a **connected** tree, i.e., not a proper forest
Error Exponent for Structure Learning III

\[ P^n \left( E_{ML} \neq E_P \right) = \exp \left[ -n \min_{e' \notin E_P} \left( \min_{e \in \text{Path}(e' ; E_P)} J_{e,e'} \right) \right] \]

We have a finite-sample result too! See thesis

**Proposition**

The following statements are equivalent:

(a) The error probability **decays exponentially**, i.e., \( K_P > 0 \)

(b) \( T_P \) is a **connected** tree, i.e., not a proper forest
Def: Very-noisy learning condition on $P_{e,e'}$

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

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

$$I(P_e) \approx I(P_{e'})$$
Approximating The Crossover Rate I

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

\[ P_e \approx P_{e'} \]

\[ I(P_e) \approx I(P_{e'}) \]
Def: *Very-noisy* learning condition on $P_{e,e'}$

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

$$I(P_e) \approx I(P_{e'})$$
Def: Very-noisy learning condition on $P_{e,e'}$

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

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

Euclidean Information Theory [Borade & Zheng '08]:

$$P \approx Q \implies D(P \parallel Q) \approx \frac{1}{2} \sum_a \frac{(P(a) - Q(a))^2}{P(a)}$$
Def: Very-noisy learning condition on $P_{e,e'}$

$P_e \approx P_{e'}$

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

Euclidean Information Theory [Borade & Zheng ’08]:

$P \approx Q \Rightarrow D(P \parallel Q) \approx \frac{1}{2} \sum_a \frac{(P(a) - Q(a))^2}{P(a)}$

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

$S_e(X_i; X_j) \triangleq \log \frac{P_{i,j}(X_i, X_j)}{P_i(X_i)P_j(X_j)}$, \quad \mathbb{E}[S_e] = I(P_e)$. 
Convexifying the optimization problem by linearizing constraints
Convexifying the optimization problem by linearizing constraints

\[ D(Q^*_e, e' \| P_e, e') \]

\[ \{ I(Q_e) = I(Q_{e'}) \} \]
Convexifying the optimization problem by linearizing constraints

\[ D(Q^*_{e,e'} \| P_{e,e'}) \]

\[ \{ I(Q_e) = I(Q_{e'}) \} \]

\[ \frac{1}{2} \| Q^*_{e,e'} - P_{e,e'} \|_{P_{e,e'}}^2 \]
Convexifying the optimization problem by linearizing constraints

**Theorem (Euclidean Approximation of Crossover Rate)**

\[
\tilde{J}_{e,e'} = \frac{(I(P_{e'}) - I(P_e))^2}{2 \text{ Var}(S_{e'} - S_e)}
\]
Convexifying the optimization problem by linearizing constraints

Theorem (Euclidean Approximation of Crossover Rate)

\[ \tilde{J}_{e,e'} = \frac{(I(P_{e'}) - I(P_e))^2}{2 \text{Var}(S_{e'} - S_e)} = \frac{(E[S_{e'} - S_e])^2}{2 \text{Var}(S_{e'} - S_e)} = \frac{1}{2} \text{SNR} \]
The Crossover Rate

How good is the approximation? We consider a binary model

![Graph showing the crossover rate comparison between true and approximate rates.](image)
Remarks for Learning Discrete Trees

- Characterized precisely the error exponent for structure learning

\[ P^n (E_{ML} \neq E_P) \doteq \exp(-nK_P) \]

---

Remarks for Learning Discrete Trees

- Characterized precisely the error exponent for structure learning

\[ P^n (E_{\text{ML}} \neq E_P) = \exp(-nK_P) \]

- Analysis tools include the method of types (large-deviations) and simple properties of trees

---

Remarks for Learning Discrete Trees

- Characterized precisely the error exponent for structure learning
  \[ P^n (E_{ML} \neq E_P) = \exp(-nK_P) \]

- Analysis tools include the method of types (large-deviations) and simple properties of trees

- Analyzed the very-noisy learning regime (Euclidean Information Theory) where learning is error-prone

- Extensions to learning the tree projection for non-trees have also been studied.

---

Outline

1. Motivation, Background and Main Contributions
2. Learning Discrete Trees Models: Error Exponent Analysis
3. Learning Gaussian Trees Models: Extremal Structures
4. Learning High-Dimensional Forest-Structured Models
5. Related Topics and Conclusion
Setup

- **Jointly Gaussian** distribution in very-noisy learning regime

\[ p(x) \propto \exp \left( -\frac{1}{2} x^T \Sigma^{-1} x \right), \quad x \in \mathbb{R}^d. \]

- Zero-mean, unit variances
Setup

- **Jointly Gaussian** distribution in very-noisy learning regime

\[ p(x) \propto \exp \left( -\frac{1}{2} x^T \Sigma^{-1} x \right), \quad x \in \mathbb{R}^d. \]

- Zero-mean, unit variances

- Keep **correlations coefficients** on edges fixed – specifies the Gaussian graphical model by **Markovianity**

\[ \rho_i \text{ is the correlation coefficient on edge } e_i \text{ for } i = 1, \ldots, d - 1 \]
Setup

- **Jointly Gaussian** distribution in very-noisy learning regime
  \[ p(x) \propto \exp \left( -\frac{1}{2} x^T \Sigma^{-1} x \right), \quad x \in \mathbb{R}^d. \]

- Zero-mean, unit variances

- Keep **correlations coefficients** on edges fixed – specifies the Gaussian graphical model by Markovianity

  \[ \rho_i \text{ is the correlation coefficient on edge } e_i \text{ for } i = 1, \ldots, d - 1 \]
Setup

- **Jointly Gaussian** distribution in very-noisy learning regime

\[ p(x) \propto \exp \left( -\frac{1}{2} x^T \Sigma^{-1} x \right), \quad x \in \mathbb{R}^d. \]

- Zero-mean, unit variances

- **Keep correlations coefficients** on edges fixed – specifies the Gaussian graphical model by **Markovianity**

  \( \rho_i \) is the correlation coefficient on edge \( e_i \) for \( i = 1, \ldots, d - 1 \)

- Compare the error exponent associated to different structures
The Gaussian Case: Extremal Tree Structures

Theorem (Extremal Structures)

Under the very-noisy assumption,
- **Star graphs are hardest to learn (smallest approx error exponent)**

Vincent Tan (MIT)
Large-Deviations for Learning Trees
Thesis Defense 32 / 52
Theorem (Extremal Structures)

Under the very-noisy assumption,

- *Star graphs are hardest to learn* (smallest approx error exponent)
- *Markov chains are easiest to learn* (largest approx error exponent)
The Gaussian Case: Extremal Tree Structures

**Theorem (Extremal Structures)**

Under the *very-noisy* assumption,

- **Star graphs are hardest to learn (smallest approx error exponent)**
- **Markov chains are easiest to learn (largest approx error exponent)**

\[ \rho_4 \rho_3 \rho_2 \rho_1 \]

\[ \rho_{\pi(1)} \rho_{\pi(2)} \rho_{\pi(3)} \rho_{\pi(4)} \]

\( \pi \): Permutation
The Gaussian Case: Extremal Tree Structures

Theorem (Extremal Structures)

Under the very-noisy assumption,

- Star graphs are hardest to learn (smallest approx error exponent)
- Markov chains are easiest to learn (largest approx error exponent)

\[ P_n(\text{err}) = \# \text{ Samples} \]

\[ n = \# \text{ Samples} \]

\( \pi \): Permutation
The Gaussian Case: Extremal Tree Structures

Theorem (Extremal Structures)

Under the very-noisy assumption,

- **Star graphs are hardest to learn** (smallest approx error exponent)
- **Markov chains are easiest to learn** (largest approx error exponent)

\[
\pi: \text{Permutation} \\
\rho_{\pi(1)} \quad \rho_{\pi(2)} \quad \rho_{\pi(3)} \quad \rho_{\pi(4)} \\
\text{Star} \\
\rho_1 \quad \rho_2 \quad \rho_3 \quad \rho_4
\]

\[
P^n(\text{err}) \\
n = \# \text{Samples}
\]

\[
\text{Chain}
\]
The Gaussian Case: Extremal Tree Structures

Theorem (Extremal Structures)

Under the very-noisy assumption,

- **Star graphs are hardest to learn** \((\text{smallest approx error exponent})\)
- **Markov chains are easiest to learn** \((\text{largest approx error exponent})\)

\[ P^n(\text{err}) \]

\(n = \#\text{ Samples} \)

\(\pi: \text{Permutation} \)

\( \rho_{\pi(1)} \rho_{\pi(2)} \rho_{\pi(3)} \rho_{\pi(4)} \)

Star

Chain
The Gaussian Case: Extremal Tree Structures

Theorem (Extremal Structures)

Under the very-noisy assumption,

- **Star graphs are hardest to learn** (smallest approx error exponent)
- **Markov chains are easiest to learn** (largest approx error exponent)

\[ P^n(\text{err}) \]

\[ n = \# \text{ Samples} \]

\( \rho \): Standard deviation

\( \rho_1, \rho_2, \rho_3, \rho_4 \)

\( \rho_{\pi(1)}, \rho_{\pi(2)}, \rho_{\pi(3)}, \rho_{\pi(4)} \)

\( \pi \): Permutation

\[ n = \# \text{ Samples} \]
Numerical Simulations

Chain, Star and Hybrid for $d = 10$

$\rho_i = 0.1 \times i \quad i \in [1 : 9]$
Numerical Simulations

Chain, Star and Hybrid for $d = 10$

\[ \rho_i = 0.1 \times i \quad i \in [1:9] \]

\[ P(\text{error}) \quad - \frac{1}{n} \log P(\text{error}) \]
Numerical Simulations

Chain, Star and Hybrid for $d = 10$

$\rho_i = 0.1 \times i \quad i \in [1 : 9]$

\[ \mathbb{P}(\text{error}) \]

\[ -\frac{1}{n} \log \mathbb{P}(\text{error}) \]
Numerical Simulations

Chain, Star and Hybrid for \( d = 10 \)

\[ \rho_i = 0.1 \times i \quad i \in [1 : 9] \]

\[
\mathbb{P}(\text{error})
\]

\[
- \frac{1}{n} \log \mathbb{P}(\text{error})
\]

Simulated Prob of Error

Simulated Error Exponent

Vincent Tan (MIT)
Proof Idea and Intuition

- Correlation decay
Proof Idea and Intuition

- Correlation decay

![Diagram showing correlation decay](image-url)
Proof Idea and Intuition

- Correlation decay
Proof Idea and Intuition

- Correlation decay
Proof Idea and Intuition

- Correlation decay

\[ e' \notin E_P \]

\[ O(d^2) \]

\[ O(d) \]
Proof Idea and Intuition

- Correlation decay

Number of distance-two node pairs in:

- Star is $O(d^2)$
- Markov chain is $O(d)$
Gaussianity allows us to perform further analysis to find the extremal structures for learning.
Gaussianity allows us to perform further analysis to find the extremal structures for learning.

Allows to derive a data-processing inequality for crossover rates.

---

Concluding Remarks for Learning Gaussian Trees

- Gaussianity allows us to perform further analysis to find the extremal structures for learning.
- Allows to derive a data-processing inequality for crossover rates.
- Universal result – not (strongly) dependent on choice of correlations.

$$\rho = \{\rho_1, \ldots, \rho_{d-1}\}$$

Motivation: Prevent Overfitting

- Chow-Liu algorithm tells us how to learn trees
- Suppose we are in the high-dimensional setting where
  \[ n \ll d \]
  learning forest-structured graphical models may reduce overfitting vis-à-vis trees [Liu, Lafferty and Wasserman, 2010]
Motivation: Prevent Overfitting

- Chow-Liu algorithm tells us how to learn trees
- Suppose we are in the high-dimensional setting where
  \[ n \ll d \]
  learning forest-structured graphical models may reduce overfitting vis-à-vis trees [Liu, Lafferty and Wasserman, 2010]
- Extend Liu et al.'s work for discrete models and improve convergence results
Motivation: Prevent Overfitting

- Chow-Liu algorithm tells us how to learn trees
- Suppose we are in the high-dimensional setting where
  \[ \text{Samples } n \ll \text{Variables } d \]

learning forest-structured graphical models may reduce overfitting vis-à-vis trees [Liu, Lafferty and Wasserman, 2010]

- Extend Liu et al.’s work for discrete models and improve convergence results

- Strategy: Remove “weak” edges
Motivation: Prevent Overfitting

- Chow-Liu algorithm tells us how to learn trees
- Suppose we are in the high-dimensional setting where
  \[ \text{Samples } n \ll \text{Variables } d \]
  learning forest-structured graphical models may reduce overfitting vis-à-vis trees [Liu, Lafferty and Wasserman, 2010]
- Extend Liu et al.'s work for discrete models and improve convergence results
- Strategy: Remove “weak” edges

\[ X_3 \quad X_2 \quad X_1 \quad X_4 \]
⇒ Reduce Num Params ⇒
Motivation: Prevent Overfitting

- Chow-Liu algorithm tells us how to learn trees
- Suppose we are in the high-dimensional setting where $n \ll d$
  - Samples $n \ll \text{Variables } d$
  - learning forest-structured graphical models may reduce overfitting vis-à-vis trees [Liu, Lafferty and Wasserman, 2010]
- Extend Liu et al.’s work for discrete models and improve convergence results
- Strategy: Remove “weak” edges

\[ \begin{align*}
X_3 & \quad X_2 \\
X_1 & \quad X_4 \\
X_1 & \quad X_4 \\
\end{align*} \]
⇒ Reduce Num Params ⇒

Vincent Tan (MIT)
Main Contributions

- Propose **CLThres**, a thresholding algorithm, for consistently learning forest-structured models
Main Contributions

- Propose **CLThres**, a thresholding algorithm, for consistently learning forest-structured models.
- Prove convergence rates ("moderate deviations") for a fixed discrete graphical model $P \in \mathcal{P}(\mathcal{X}^d)$. 

Vincent Tan (MIT) 
Large-Deviations for Learning Trees 
Thesis Defense 38 / 52
Main Contributions

- Propose **CLThres**, a thresholding algorithm, for consistently learning forest-structured models.
- Prove convergence rates ("moderate deviations") for a fixed discrete graphical model $P \in \mathcal{P}(\mathcal{X}^d)$.
- Prove achievable scaling laws on $(n, d, k)$ ($k$ is the num edges) for consistent recovery in high-dimensions. Roughly speaking,

$$n \gtrsim \log^{1+\delta}(d - k)$$

is achievable.
Main Difficulty

- Unknown \textit{minimum mutual information} $I_{\text{min}}$ in the forest model

Markov order estimation [Merhav, Gutman, Ziv 1989]

If known, can easily use a threshold, i.e.,

\[ I(\hat{P}_{i,j}) < I_{\text{min}}, \]

remove $(i, j)$
Main Difficulty

- Unknown minimum mutual information $I_{\text{min}}$ in the forest model
- Markov order estimation [Merhav, Gutman, Ziv 1989]
Main Difficulty

- Unknown **minimum mutual information** $I_{\text{min}}$ in the forest model
- Markov order estimation [Merhav, Gutman, Ziv 1989]
- If known, can easily use a **threshold**, i.e,

  $$\text{if } I(\hat{P}_{i,j}) < I_{\text{min}}, \text{ remove } (i,j)$$
Main Difficulty

- Unknown **minimum mutual information** $I_{\text{min}}$ in the forest model
- Markov order estimation [Merhav, Gutman, Ziv 1989]
- If known, can easily use a **threshold**; i.e,
  
  \[
  \text{if } I(\hat{P}_{i,j}) < I_{\text{min}}, \text{ remove } (i,j)
  \]

- How to deal with classic tradeoff between **over-** and **underestimation** errors?
The CLThres Algorithm

- Compute the set of empirical mutual information $I(\hat{P}_{i,j})$ for all $(i,j) \in V \times V$
The CLThres Algorithm

- Compute the set of empirical mutual information $I(\hat{P}_{i,j})$ for all $(i,j) \in V \times V$

- Max-weight spanning tree

$$\hat{E}_{d-1} := \arg\max_{E: \text{Tree}} \sum_{(i,j) \in E} I(\hat{P}_{i,j})$$
The CLThres Algorithm

- Compute the set of empirical mutual information $I(\hat{P}_{i,j})$ for all $(i,j) \in V \times V$

- Max-weight spanning tree

$$\hat{E}_{d-1} := \arg\max_{E: \text{Tree}} \sum_{(i,j) \in E} I(\hat{P}_{i,j})$$

- Estimate number of edges given threshold $\epsilon_n$

$$\hat{k}_n := \left| \left\{ (i,j) \in \hat{E}_{d-1} : I(\hat{P}_{i,j}) \geq \epsilon_n \right\} \right|$$
The CLThres Algorithm

- Compute the set of empirical mutual information $I(\hat{P}_{i,j})$ for all $(i,j) \in V \times V$

- Max-weight spanning tree

$$\hat{E}_{d-1} := \text{argmax}_{E: \text{Tree}} \sum_{(i,j) \in E} I(\hat{P}_{i,j})$$

- Estimate number of edges given threshold $\epsilon_n$

$$\hat{k}_n := \left| \left\{ (i,j) \in \hat{E}_{d-1} : I(\hat{P}_{i,j}) \geq \epsilon_n \right\} \right|$$

- Output the forest with the top $\hat{k}_n$ edges

- Computational Complexity = $O((n + \log d)d^2)$
A Convergence Result for CLThres

Assume that $P \in \mathcal{P}(\mathcal{X}^d)$ is a fixed forest-structured graphical model. $d$ does not grow with $n$. 
A Convergence Result for CLThres

Assume that $P \in \mathcal{P}(\mathcal{X}^d)$ is a \textbf{fixed forest-structured} graphical model

$d$ does not grow with $n$

**Theorem ("Moderate Deviations")**

Assume that the sequence $\{\epsilon_n\}_{n=1}^\infty$ satisfies

$$\lim_{n \to \infty} \epsilon_n = 0, \quad \lim_{n \to \infty} \frac{n\epsilon_n}{\log n} = \infty, \quad (\epsilon_n := n^{-1/2} \text{ works})$$
A Convergence Result for CLThres

Assume that $P \in \mathcal{P}(\mathcal{X}^d)$ is a fixed forest-structured graphical model. $d$ does not grow with $n$.

**Theorem (“Moderate Deviations”)**

Assume that the sequence $\{\epsilon_n\}_{n=1}^{\infty}$ satisfies

$$\lim_{n \to \infty} \epsilon_n = 0, \quad \lim_{n \to \infty} \frac{n \epsilon_n}{\log n} = \infty, \quad (\epsilon_n := n^{-1/2} \text{ works})$$

Then

$$\limsup_{n \to \infty} \frac{1}{n \epsilon_n} \log P(\hat{E}_{kn} \neq E_P) \leq -1, \quad \Rightarrow$$
Assume that $P \in \mathcal{P}(\mathcal{X}^d)$ is a fixed forest-structured graphical model and $d$ does not grow with $n$.

**Theorem ("Moderate Deviations")**

Assume that the sequence $\{\epsilon_n\}_{n=1}^{\infty}$ satisfies

$$\lim_{n \to \infty} \epsilon_n = 0, \quad \lim_{n \to \infty} \frac{n \epsilon_n}{\log n} = \infty,$$

$(\epsilon_n := n^{-1/2} \text{ works})$

Then

$$\limsup_{n \to \infty} \frac{1}{n \epsilon_n} \log \mathbb{P}(\hat{E}_{kn} \neq E_P) \leq -1, \quad \Rightarrow \quad \mathbb{P}(\hat{E}_{kn} \neq E_P) \approx \exp(-n \epsilon_n)$$
A Convergence Result for CLThres

Assume that $P \in \mathcal{P}(\mathcal{X}^d)$ is a **fixed forest-structured** graphical model.

d does not grow with $n$

**Theorem (“Moderate Deviations”)**

Assume that the sequence $\{\epsilon_n\}_{n=1}^{\infty}$ satisfies

\[
\lim_{n \to \infty} \epsilon_n = 0, \quad \lim_{n \to \infty} \frac{n\epsilon_n}{\log n} = \infty, \quad (\epsilon_n := n^{-1/2} \text{ works})
\]

Then

\[
\limsup_{n \to \infty} \frac{1}{n\epsilon_n} \log \mathbb{P}(\hat{E}_{kn} \neq E_P) \leq -1, \quad \Rightarrow \quad \mathbb{P}(\hat{E}_{kn} \neq E_P) \approx \exp(-n\epsilon_n)
\]

Also have a “liminf” lower bound
Remarks: A Convergence Result for CLThres

- The Chow-Liu phase is consistent with exponential rate of convergence
The Chow-Liu phase is consistent with \textit{exponential rate of convergence}

The sequence can be taken to be $\epsilon_n := n^{-\beta}$ for $\beta \in (0, 1)$
Remarks: A Convergence Result for CLThres

- The Chow-Liu phase is consistent with exponential rate of convergence.
- The sequence can be taken to be \( \epsilon_n := n^{-\beta} \) for \( \beta \in (0, 1) \).
- For all \( n \) sufficiently large,
  \[
  \epsilon_n < I_{\text{min}}
  \]
  implies no underestimation asymptotically.
The Chow-Liu phase is consistent with exponential rate of convergence.

The sequence can be taken to be $\epsilon_n := n^{-\beta}$ for $\beta \in (0, 1)$.

For all $n$ sufficiently large, $\epsilon_n < I_{\text{min}}$ implies no underestimation asymptotically.

Note that for two independent random variables $X_i$ and $X_j$ with product pmf $Q_i \times Q_j$, 

$$\text{std}(I(\hat{P}_{i,j})) = \Theta(1/n)$$

Since the sequence $\epsilon_n = \omega(\log n/n)$ decays slower than $\text{std}(I(\hat{P}_{i,j}))$, no overestimation asymptotically.
Asymptotically, $\epsilon_n$ will be smaller than $I_{\text{min}}$ and larger than $I(\hat{P}_{i,j})$ with high probability.
Pruning Away Weak Empirical Mutual Informations

Asymptotically, $\epsilon_n$ will be smaller than $I_{\text{min}}$ and larger than $I(\hat{P}_{i,j})$ with high probability.

$\begin{align*}
I(\hat{P}_{i,j}) &\approx \frac{1}{n} \\
I_{\text{min}} &\text{ (unknown)}
\end{align*}$
Asymptotically, \( \epsilon_n \) will be smaller than \( I_{\min} \) and larger than \( I(\widehat{P}_{i,j}) \) with high probability.
Based fully on the method of types
Proof Idea

Based fully on the *method of types*

- Estimate **Chow-Liu** learning error
Proof Idea

Based fully on the method of types

- Estimate Chow-Liu learning error
- Estimate underestimation error

\[ \mathbb{P}(\hat{k}_n < k) \approx \exp(-nL_P) \]
Proof Idea

Based fully on the method of types

- Estimate Chow-Liu learning error
- Estimate underestimation error

$$\mathbb{P}(\hat{k}_n < k) \approx \exp(-nL_P)$$

- Estimate overestimation error

Decays subexponentially but faster than any polynomial:

$$\mathbb{P}(\hat{k}_n > k) \approx \exp(-n\epsilon_n)$$

Upper bound has no dependence on $P$ (there exists a duality gap).
Proof Idea

Based fully on the method of types

- Estimate Chow-Liu learning error
- Estimate underestimation error

\[ P(\hat{k}_n < k) \doteq \exp(-nL_P) \]

- Estimate overestimation error

Decays subexponentially but faster than any polynomial:

\[ P(\hat{k}_n > k) \approx \exp(-n\epsilon_n) \]

Upper bound has no dependence on \( P \) (there exists a duality gap)

Additional Technique: Euclidean Information Theory
Consider a sequence of structure learning problems indexed by number of samples $n$. 
Consider a sequence of structure learning problems indexed by number of samples $n$.

For each particular problem, we have data $x^n = \{x_i\}_{i=1}^n$. 

Assumptions:

(A1) $I_{inf} := \inf_{d \in \mathbb{N}} \min_{(i,j) \in E} P_{i,j}(P_i, j) > 0$

(A2) $\kappa := \inf_{d \in \mathbb{N}} \min_{(x_i, x_j) \in X_2} P_{i,j}(x_i, x_j) > 0$
Consider a sequence of structure learning problems indexed by number of samples $n$

For each particular problem, we have data $x^n = \{x_i\}_{i=1}^n$

Each sample $x_i \in \mathcal{X}^d$ is drawn independently from a forest-structured model with $d$ nodes and $k$ edges
Consider a sequence of structure learning problems indexed by number of samples $n$.

For each particular problem, we have data $x^n = \{x_i\}_{i=1}^n$.

Each sample $x_i \in \mathcal{X}^d$ is drawn independently from a forest-structured model with $d$ nodes and $k$ edges.

Sequence of tuples $\{(n, d_n, k_n)\}_{n=1}^\infty$. 

Assumptions:

(A1) $I_{\inf} := \inf_{d \in \mathbb{N}} \min_{(i, j) \in E} P_{i,j} > 0$.

(A2) $\kappa := \inf_{d \in \mathbb{N}} \min_{(x_i, x_j) \in \mathcal{X}^2} P_{i,j}(x_i, x_j) > 0$. 

Consider a sequence of structure learning problems indexed by number of samples $n$

For each particular problem, we have data $x^n = \{x_i\}_{i=1}^n$

Each sample $x_i \in \mathcal{X}^d$ is drawn independently from a forest-structured model with $d$ nodes and $k$ edges

Sequence of tuples $\{(n, d_n, k_n)\}_{n=1}^\infty$

**Assumptions**

(A1) $I_{\text{inf}} := \inf_{d \in \mathbb{N}} \min_{(i,j) \in E_P} I(P_{i,j}) > 0$

(A2) $\kappa := \inf_{d \in \mathbb{N}} \min_{(x_i,x_j) \in \mathcal{X}^2} P_{i,j}(x_i,x_j) > 0$
Theorem (Sufficient Conditions)

Assume (A1) and (A2). Fix \( \delta > 0 \). There exists constants \( C_1, C_2 > 0 \) such that if

\[
n > \max \left\{ C_1 \log d, C_2 \log k, \right\}
\]

the error probability of structure learning \( P(error) \to 0 \) as \( (n, d_n, k_n) \to \infty \).
Theorem (Sufficient Conditions)

Assume (A1) and (A2). Fix $\delta > 0$. There exists constants $C_1, C_2 > 0$ such that if

$$n > \max \left\{ C_1 \log d, C_2 \log k, (2 \log (d - k))^{1+\delta} \right\}$$

the error probability of structure learning $P_{\text{error}} \to 0$ as $(n, d, k) \to \infty$. 
Theorem (Sufficient Conditions)

Assume (A1) and (A2). Fix $\delta > 0$. There exists constants $C_1, C_2 > 0$ such that if

$$n > \max \left\{ C_1 \log d, C_2 \log k, (2 \log(d - k))^{1+\delta} \right\}$$

the error probability of structure learning

$$\mathbb{P}(\text{error}) \to 0$$

as $(n, d_n, k_n) \to \infty$
Remarks on the Achievable Scaling Law for CLThres

- If the model parameters \((n, d, k)\) grow with \(n\) but if

\[
\begin{align*}
  d & \quad \text{subexponential} \\
  k & \quad \text{subexponential} \\
  d - k & \quad \text{subexponential}
\end{align*}
\]

structure recovery is \textit{asymptotically possible}
Remarks on the Achievable Scaling Law for CLThres

- If the model parameters \((n, d, k)\) grow with \(n\) but if

\[
\begin{align*}
    d & \text{ subexponential} \\
    k & \text{ subexponential} \\
    d - k & \text{ subexponential}
\end{align*}
\]

structure recovery is asymptotically possible

- \(d\) can grow much faster than \(n\)
Remarks on the Achievable Scaling Law for CLThres

- If the model parameters \((n, d, k)\) grow with \(n\) but if

\[
\begin{align*}
  d & \text{ subexponential} \\
  k & \text{ subexponential} \\
  d - k & \text{ subexponential}
\end{align*}
\]

structure recovery is asymptotically possible

- \(d\) can grow much faster than \(n\)

- Proof uses:
  1. Previous fixed \(d\) result
  2. Exponents in the limsup upper bound do not vanish with increasing problem size as \((n, d_n, k_n) \to \infty\)
Proposition (A Necessary Condition)

Assume forests with $d$ nodes are chosen uniformly at random. Fix $\eta > 0$. Then if

$$n < \frac{(1 - \eta) \log d}{\log |\mathcal{X}|}$$

the error probability of structure learning

$$\mathbb{P}(\text{error}) \to 1$$

as $(n, d_n) \to \infty$ (independent of $k_n$)
A Simple Strong Converse Result

Proposition (A Necessary Condition)

Assume forests with $d$ nodes are chosen uniformly at random. Fix $\eta > 0$. Then if

$$n < \frac{(1 - \eta) \log d}{\log |\mathcal{X}|}$$

the error probability of structure learning

$$\mathbb{P}(\text{error}) \to 1$$

as $\lim_{n,d_n \to \infty}$ (independent of $k_n$)

- $\Omega(\log d)$ is necessary for successful recovery
- This lower bound is independent of parameters
- The dependence on num of edges $k_n$ can be made more explicit
- Close to the sufficient condition
Concluding Remarks for Learning Forests

- Proposed a simple extension of Chow-Liu’s MWST algorithm to learn forests **consistently**

---

**VYFT**, A. Anandkumar and A. S. Willsky “Learning High-Dimensional Markov Forest Distributions: Analysis of Error Rates”, Allerton 10, Submitted to JMLR.
Concluding Remarks for Learning Forests

- Proposed a simple extension of Chow-Liu’s MWST algorithm to learn forests **consistently**
- Error rates in the form of a “**moderate deviations**” result

**VYFT**, A. Anandkumar and A. S. Willsky “Learning High-Dimensional Markov Forest Distributions: Analysis of Error Rates”, Allerton 10, Submitted to JMLR.

Vincent Tan (MIT)  Large-Deviations for Learning Trees  Thesis Defense
Concluding Remarks for Learning Forests

- Proposed a simple extension of Chow-Liu’s MWST algorithm to learn forests **consistently**
- Error rates in the form of a “**moderate deviations**” result
- Scaling laws on $(n, d, k)$ for **structural consistency** in high dimensions

**VYFT**, A. Anandkumar and A. S. Willsky “Learning High-Dimensional Markov Forest Distributions: Analysis of Error Rates”, Allerton 10, Submitted to JMLR.
Concluding Remarks for Learning Forests

- Proposed a simple extension of Chow-Liu’s MWST algorithm to learn forests consistently
- Error rates in the form of a “moderate deviations” result
- Scaling laws on \((n, d, k)\) for structural consistency in high dimensions

Extensions:

---

**VYFT**, A. Anandkumar and A. S. Willsky “Learning High-Dimensional Markov Forest Distributions: Analysis of Error Rates”, Allerton 10, Submitted to JMLR.
Concluding Remarks for Learning Forests

- Proposed a simple extension of Chow-Liu’s MWST algorithm to learn forests **consistently**
- Error rates in the form of a “**moderate deviations**” result
- Scaling laws on \((n, d, k)\) for **structural consistency** in high dimensions

**Extensions:**

- **Risk consistency** has also been analyzed (See thesis for details)

\[
R(P^*) = O_p \left( \frac{d \log d}{n^{1-\gamma}} \right)
\]

**References:**

**VYFT**, A. Anandkumar and A. S. Willsky “Learning High-Dimensional Markov Forest Distributions: Analysis of Error Rates”, Allerton 10, Submitted to JMLR.
Concluding Remarks for Learning Forests

- Proposed a simple extension of Chow-Liu’s MWST algorithm to learn forests consistently
- Error rates in the form of a “moderate deviations” result
- Scaling laws on \((n, d, k)\) for structural consistency in high dimensions

Extensions:

- Risk consistency has also been analyzed (See thesis for details)

\[
R(P^*) = O_p \left( \frac{d \log d}{n^{1-\gamma}} \right)
\]

- Need to find the right balance between over- and underestimation for the finite sample case

---

**VYFT**, A. Anandkumar and A. S. Willsky “Learning High-Dimensional Markov Forest Distributions: Analysis of Error Rates”, Allerton 10, Submitted to JMLR.
Outline

1. Motivation, Background and Main Contributions
2. Learning Discrete Trees Models: Error Exponent Analysis
3. Learning Gaussian Trees Models: Extremal Structures
4. Learning High-Dimensional Forest-Structured Models
5. Related Topics and Conclusion
Techniques extend to learning other classes of graphical models.
Techniques extend to learning other classes of graphical models.
Techniques extend to learning other classes of graphical models

- Learn latent trees, where only a subset of nodes are observed.
- If the original graph is drawn from the Erdős-Rényi ensemble $G(n, c/n)$, we can also provide guarantees for structure learning.
- Utilize the fact that the model is locally tree-like.
Conclusions

- Graphical models provide a powerful and parsimonious representation of high-dimensional data

- (Ch. 3) Provided large-deviation analysis of ML learning of tree-structured distributions
  - (Ch. 4) Identified extremal structures for tree-structured Gaussian graphical models
Conclusions

- Graphical models provide a powerful and parsimonious representation of high-dimensional data

- (Ch. 3) Provided large-deviation analysis of ML learning of tree-structured distributions
  - (Ch. 4) Identified extremal structures for tree-structured Gaussian graphical models

- (Ch. 5) Extended analysis to forest-structured graphical models
  - Derived scaling laws on num variables, num edges and num samples for consistent learning in high-dimensions
Conclusions

- Graphical models provide a powerful and parsimonious representation of high-dimensional data

(Ch. 3) Provided large-deviation analysis of ML learning of tree-structured distributions

  (Ch. 4) Identified extremal structures for tree-structured Gaussian graphical models

(Ch. 5) Extended analysis to forest-structured graphical models

  Derived scaling laws on num variables, num edges and num samples for consistent learning in high-dimensions

(Ch. 6) Also proposed algorithms for learning tree models for hypothesis testing