Scaling Laws for Learning High-Dimensional Markov Forest Distributions

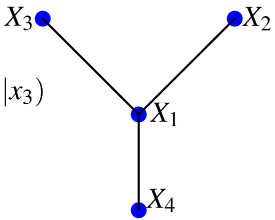
Vincent Tan†, Animashree Anandkumar‡, Alan S. Willsky†

† Stochastic Systems Group,
Laboratory for Information and Decision Systems,
Massachusetts Institute of Technology

‡ Center for Pervasive Communications and Computing,
Electrical Engineering and Computer Science,
University of California, Irvine.

Allerton Conference (Sep 29, 2010)

- Learning tree-structured graphical models given i.i.d. samples is well-known.

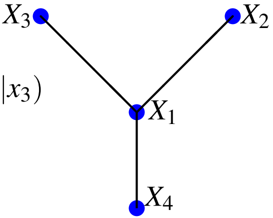


$$P(\mathbf{x}) = P_1(x_1)P(x_2|x_1)P(x_3|x_2)P(x_4|x_3)$$

$$\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{i.i.d.}{\sim} P$$

Motivation

- Learning tree-structured graphical models given i.i.d. samples is well-known.



$$P(\mathbf{x}) = P_1(x_1)P(x_2|x_1)P(x_3|x_2)P(x_4|x_3)$$

$$\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{i.i.d.}{\sim} P$$

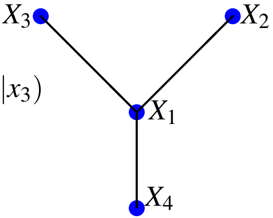
- The Chow-Liu algorithm (1968) provides an efficient implementation of maximum-likelihood estimation.

Motivation

- Learning tree-structured graphical models given i.i.d. samples is well-known.

$$P(\mathbf{x}) = P_1(x_1)P(x_2|x_1)P(x_3|x_2)P(x_4|x_3)$$

$$\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{i.i.d.}{\sim} P$$



- The Chow-Liu algorithm (1968) provides an efficient implementation of maximum-likelihood estimation.

- What if we want a larger class of acyclic models?

Motivation: Prevent Overfitting

- High-dimensional setting.
- If the number of samples n is significantly fewer than the number of dimensions d, i.e.,

$$n \ll d$$

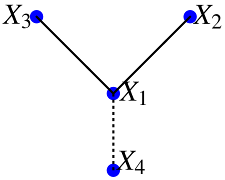
learning forest-structured distributions may reduce overfitting [Liu, Lafferty and Wasserman, 2010].

Motivation: Prevent Overfitting

- High-dimensional setting.
- If the number of samples n is significantly fewer than the number of dimensions d, i.e.,

$$n \ll d$$

  learning forest-structured distributions may reduce overfitting [Liu, Lafferty and Wasserman, 2010].
- Strategy: Remove "weak" edges to prevent overfitting.



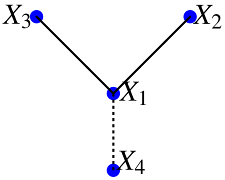
Motivation: Prevent Overfitting

- High-dimensional setting.
- If the number of samples n is significantly fewer than the number of dimensions d, i.e.,

$$n \ll d$$

  learning forest-structured distributions may reduce overfitting [Liu, Lafferty and Wasserman, 2010].

- Strategy: Remove “weak” edges to prevent overfitting.



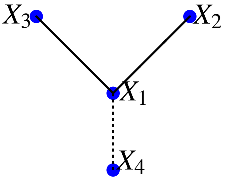
$\Rightarrow$ Reduce Num Params $\Rightarrow$

Motivation: Prevent Overfitting

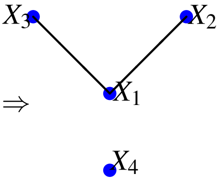
- High-dimensional setting.
- If the number of samples n is significantly fewer than the number of dimensions d, i.e.,

$$n \ll d$$

  learning forest-structured distributions may reduce overfitting [Liu, Lafferty and Wasserman, 2010].
- Strategy: Remove “weak” edges to prevent overfitting.



$\Rightarrow$ Reduce Num Params $\Rightarrow$



Natural Questions

- For a fixed model $P \in \mathcal{P}(\mathcal{X}^d)$, are there any simple modifications to Chow-Liu to learn forests consistently?

Natural Questions

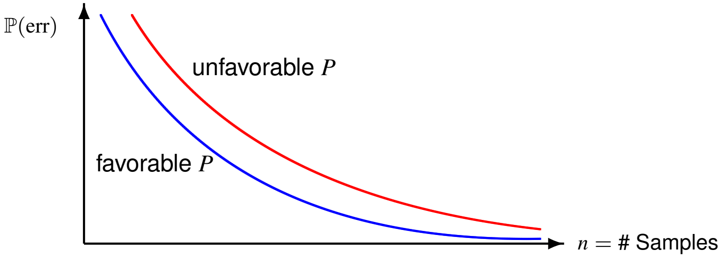
- For a fixed model $P \in \mathcal{P}(\mathcal{X}^d)$, are there any simple modifications to Chow-Liu to learn forests consistently?
- What are the rates of convergence for a particular P?

Natural Questions

- For a fixed model $P \in \mathcal{P}(\mathcal{X}^d)$, are there any simple modifications to Chow-Liu to learn forests consistently?
- What are the rates of convergence for a particular P?

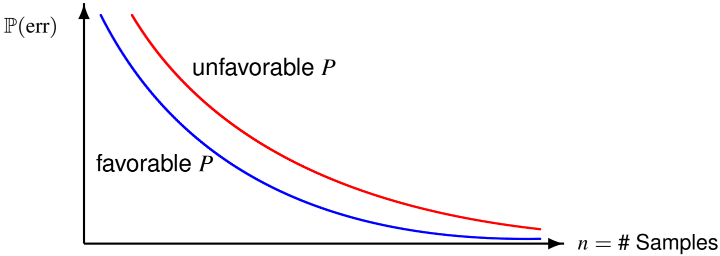
Natural Questions

- For a fixed model $P \in \mathcal{P}(\mathcal{X}^d)$, are there any simple modifications to Chow-Liu to learn forests consistently?
- What are the rates of convergence for a particular P?



Natural Questions

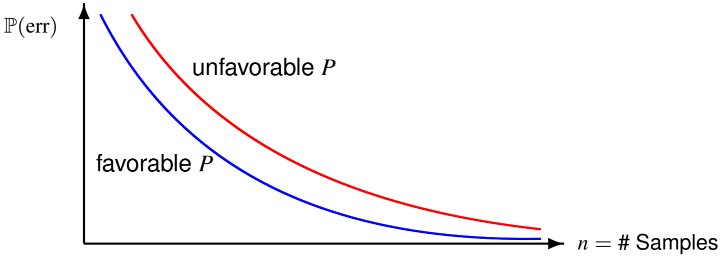
- For a fixed model $P \in \mathcal{P}(\mathcal{X}^d)$, are there any simple modifications to Chow-Liu to learn forests consistently?
- What are the rates of convergence for a particular P?



- How can following parameters scale with one another in the high-dimensional setting?
  1. Number of samples n

Natural Questions

- For a fixed model $P \in \mathcal{P}(\mathcal{X}^d)$, are there any simple modifications to Chow-Liu to learn forests consistently?
- What are the rates of convergence for a particular P?

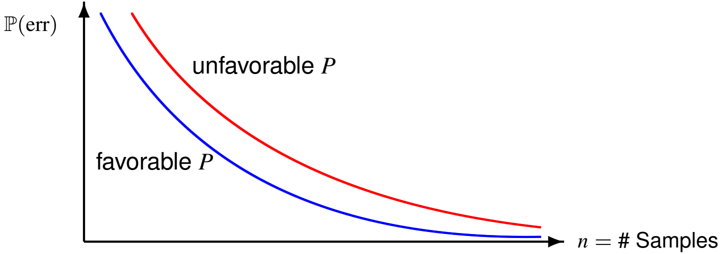


- How can following parameters scale with one another in the high-dimensional setting?

1. Number of samples n
2. Number of variables d

Natural Questions

- For a fixed model $P \in \mathcal{P}(\mathcal{X}^d)$, are there any simple modifications to Chow-Liu to learn forests consistently?
- What are the rates of convergence for a particular P?



- How can following parameters scale with one another in the high-dimensional setting?
  1. Number of samples n
  2. Number of variables d
  3. Number of edges $k \leq d - 1$

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 $P \in \mathcal{P}(\mathcal{X}^d)$.

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) for consistent recovery in high-dimensions. Roughly speaking,

$$n > C_1 \log^{1+\delta}(d-k), \qquad \forall \delta > 0$$

is achievable.

Problem Setup

- Let $\mathcal{X}$ be a finite set and let $\mathcal{P}(\mathcal{X}^d)$ be the probability simplex over $\mathcal{X}^d$.
- We say that $P \in \mathcal{P}(\mathcal{X}^d)$ is a forest-structured model if it factorizes as

$$P(\mathbf{x}) = \prod_{i \in V} P(x_i) \prod_{(i,j) \in E_P} \frac{P(x_i, x_j)}{P(x_i)P(x_j)}$$

where $V = [1 : d]$ and $E_P \subset \binom{V}{2}$ and note $|E_P| \le d - 1$.

Problem Setup

- Let $\mathcal{X}$ be a finite set and let $\mathcal{P}(\mathcal{X}^d)$ be the probability simplex over $\mathcal{X}^d$.

- We say that $P \in \mathcal{P}(\mathcal{X}^d)$ is a forest-structured model if it factorizes as

$$P(\mathbf{x}) = \prod_{i \in V} P(x_i) \prod_{(i,j) \in E_P} \frac{P(x_i, x_j)}{P(x_i) P(x_j)}$$

  where $V = [1 : d]$ and $E_P \subset \binom{V}{2}$ and note $|E_P| \leq d - 1$.

- Given n i.i.d. samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ drawn from P, a forest-structured model with edge set E_P.

Problem Setup

- Let $\mathcal{X}$ be a finite set and let $\mathcal{P}(\mathcal{X}^d)$ be the probability simplex over $\mathcal{X}^d$.

- We say that $P \in \mathcal{P}(\mathcal{X}^d)$ is a forest-structured model if it factorizes as

$$P(\mathbf{x}) = \prod_{i \in V} P(x_i) \prod_{(i,j) \in E_P} \frac{P(x_i, x_j)}{P(x_i)P(x_j)}$$

  where $V = [1 : d]$ and $E_P \subset \binom{V}{2}$ and note $|E_P| \leq d - 1$.

- Given n i.i.d. samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ drawn from P, a forest-structured model with edge set E_P.

- Output an estimate of the structure $\widehat{E}$.

Main Difficulty

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

Main Difficulty

- Unknown minimum mutual information $I_{\min}$ in the forest model.
- Markov order estimation.

Main Difficulty

- Unknown minimum mutual information $I_{\min}$ in the forest model.
- Markov order estimation.
- If known, can easily use a threshold, i.e,

$$\text{if} \qquad \widehat{I}(X_i; X_j) < I_{\min}, \qquad \text{remove} \quad (i,j)$$

Main Difficulty

- Unknown minimum mutual information $I_{\min}$ in the forest model.
- Markov order estimation.
- If known, can easily use a threshold, i.e,

$$\text{if} \qquad \widehat{I}(X_i;X_j) < I_{\min}, \qquad \text{remove} \quad (i,j)$$

- How to deal with classic tradeoff between over- and underestimation errors?

The CLThres Algorithm

- Compute the set of empirical mutual information $\widehat{I}(X_i; X_j)$ for all $(i,j) \in V \times V$.

The CLThres Algorithm

- Compute the set of empirical mutual information $\widehat{I}(X_i; X_j)$ for all $(i,j) \in V \times V$.
- Max-weight spanning tree

$$\widehat{E}_{d-1} := \operatorname*{argmax}_{E:\mathrm{Tree}} \sum_{(i,j)\in E} \widehat{I}(X_i; X_j)$$

The CLThres Algorithm

- Compute the set of empirical mutual information $\widehat{I}(X_i; X_j)$ for all $(i,j) \in V \times V$.
- Max-weight spanning tree

$$\widehat{E}_{d-1} := \operatorname*{argmax}_{E:\text{Tree}} \sum_{(i,j)\in E} \widehat{I}(X_i; X_j)$$

- Estimate number of edges given threshold ϵ_n

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

The CLThres Algorithm

- Compute the set of empirical mutual information $\widehat{I}(X_i; X_j)$ for all $(i,j) \in V \times V$.
- Max-weight spanning tree

$$\widehat{E}_{d-1} := \underset{E:\text{Tree}}{\text{argmax}} \sum_{(i,j)\in E} \widehat{I}(X_i; X_j)$$

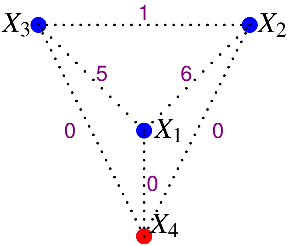
- Estimate number of edges given threshold ϵ_n

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

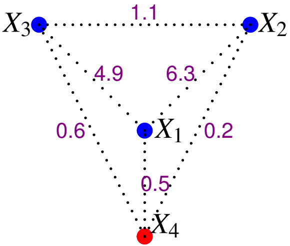
- Output the forest with the top $\widehat{k}_n$ edges.
- Computational Complexity = $O((n + \log d)d^2)$.

The CLThres Algorithm with $\epsilon_n = 1$

The CLThres Algorithm with $\epsilon_n = 1$

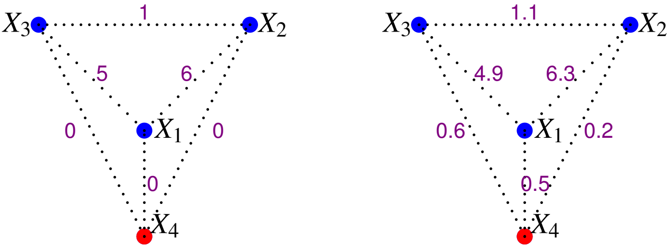


True MI $I(X_i; X_j)$

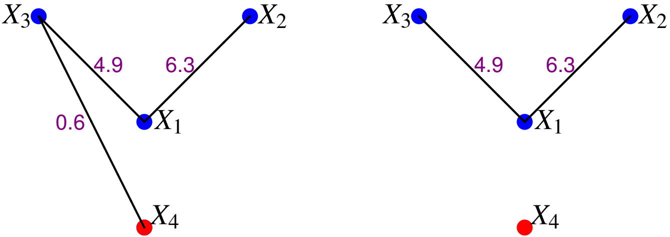


Empirical MI $\widehat{I}(X_i; X_j)$

The CLThres Algorithm with $\epsilon_n = 1$



True MI $I(X_i; X_j)$ Empirical MI $\widehat{I}(X_i; X_j)$



Max-weight spanning tree $\widehat{E}_{d-1}$ Thresholded Forest $\widehat{E}_{\widehat{k}_n}$

A Convergence Result for CLThres

We first assume that $P \in \mathcal{P}(\mathcal{X}^d)$ is a fixed distribution, i.e., d does not grow with n.

A Convergence Result for CLThres

We first assume that $P \in \mathcal{P}(\mathcal{X}^d)$ is a fixed distribution, i.e., d does not grow with n.

Theorem ("Moderate Deviations")

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

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

A Convergence Result for CLThres

We first assume that $P \in \mathcal{P}(\mathcal{X}^d)$ is a fixed distribution, i.e., d does not grow with n.

Theorem ("Moderate Deviations")

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

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

Then

$$\limsup_{n\to\infty} \frac{1}{n\epsilon_n} \log \mathbb{P}(\widehat{E}_{\widehat{k}_n} \neq E_P) \leq -1$$

We first assume that $P \in \mathcal{P}(\mathcal{X}^d)$ is a fixed distribution, i.e., d does not grow with n.

Theorem ("Moderate Deviations")

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

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

Then

$$\limsup_{n\to\infty} \frac{1}{n\epsilon_n} \log \mathbb{P}(\widehat{E}_{\widehat{k}_n} \neq E_P) \leq -1$$

Roughly speaking, $\mathbb{P}(\widehat{E}_{\widehat{k}_n} \neq E_P) \approx \exp(-n\epsilon_n)$

Also have a "liminf" lower bound.

Remarks: A Convergence Result for CLThres

- The Chow-Liu phase is consistent with exponential rate of convergence [Tan, Anandkumar, Tong and Willsky 2009].

Remarks: A Convergence Result for CLThres

- The Chow-Liu phase is consistent with exponential rate of convergence [Tan, Anandkumar, Tong and Willsky 2009].
- 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 [Tan, Anandkumar, Tong and Willsky 2009].
- The sequence can be taken to be $\epsilon_n := n^{-\beta}$ for $\beta \in (0, 1)$.
- For all n sufficiently large,

$$\epsilon_n < I_{\min}$$

  implies no underestimation asymptotically.

Remarks: A Convergence Result for CLThres

- The Chow-Liu phase is consistent with exponential rate of convergence [Tan, Anandkumar, Tong and Willsky 2009].
- The sequence can be taken to be $\epsilon_n := n^{-\beta}$ for $\beta \in (0, 1)$.
- For all n sufficiently large,

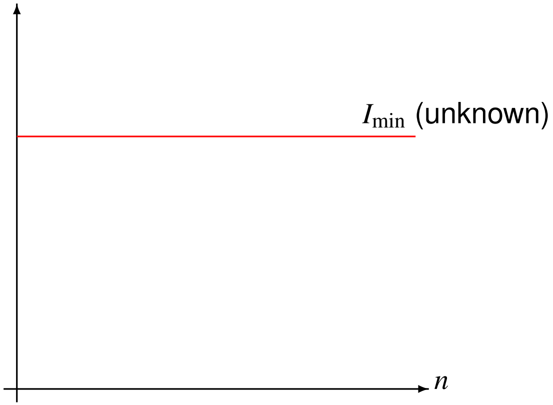
$$\epsilon_n < I_{\min}$$

  implies no underestimation asymptotically.
- Note that for two independent random variables X_i and X_j with product pmf $Q_i \times Q_j$,

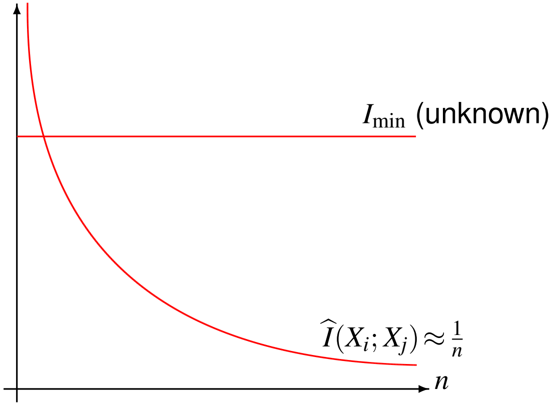
$$\mathrm{std}(\widehat{I}(X_i; X_j)) = \Theta(1/n)$$

  Since the sequence $\epsilon_n = \omega(\log n/n)$ decays slower than $\mathrm{std}(\widehat{I}(X_i; X_j))$, no overestimation asymptotically.

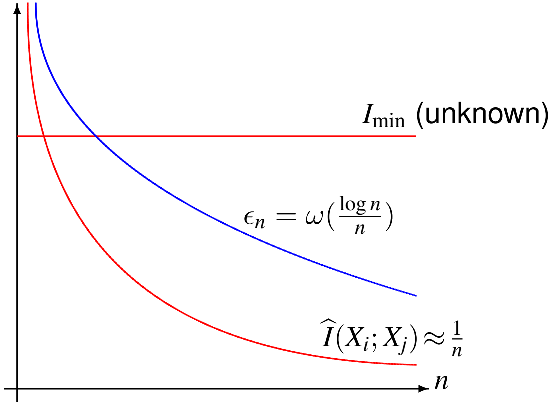
Pruning Away Weak Empirical Mutual Informations



Pruning Away Weak Empirical Mutual Informations



Pruning Away Weak Empirical Mutual Informations



Asymptotically, ϵ_n will be smaller than $I_{\min}$ and larger than $\widehat{I}(X_i; X_j)$ with high probability.

Proof Idea

Based fully on the method of types [Csiszár and Körner].

Proof Idea

Based fully on the method of types [Csiszár and Körner].

- Estimate Chow-Liu learning error.

Proof Idea

Based fully on the method of types [Csiszár and Körner].

- Estimate Chow-Liu learning error.
- Estimate underestimation error.

$$\mathbb{P}(\widehat{k}_n < k) \doteq \exp(-nL_P).$$

Proof Idea

Based fully on the method of types [Csiszár and Körner].

- Estimate Chow-Liu learning error.
- Estimate underestimation error.

$$\mathbb{P}(\widehat{k}_n < k) \doteq \exp(-nL_P).$$

- Estimate overestimation error:

  This can be shown to decay subexponentially but faster than any polynomial:

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

  Upper bound has no dependence on P.

Proof Idea

Based fully on the method of types [Csiszár and Körner].

- Estimate Chow-Liu learning error.
- Estimate underestimation error.

$$\mathbb{P}(\widehat{k}_n < k) \doteq \exp(-nL_P).$$

- Estimate overestimation error:

  This can be shown to decay subexponentially but faster than any polynomial:

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

  Upper bound has no dependence on P.

  Additional Technique: Ideas from Euclidean Information Theory [Borade and Zheng 2008].

High-Dimensional Learning

High-Dimensional Learning

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

High-Dimensional Learning

- Consider a sequence of structure learning problems indexed by number of samples n.
- For each particular problem, we have data $\mathbf{x}^n = \{\mathbf{x}_i\}_{i=1}^n$.

High-Dimensional Learning

- Consider a sequence of structure learning problems indexed by number of samples n.
- For each particular problem, we have data $\mathbf{x}^n = \{\mathbf{x}_i\}_{i=1}^n$.
- Each sample $\mathbf{x}_i \in \mathcal{X}^d$ is drawn independently from a forest-structured model with d nodes and k edges.

High-Dimensional Learning

- Consider a sequence of structure learning problems indexed by number of samples n.
- For each particular problem, we have data $\mathbf{x}^n = \{\mathbf{x}_i\}_{i=1}^n$.
- Each sample $\mathbf{x}_i \in \mathcal{X}^d$ is drawn independently from a forest-structured model with d nodes and k edges.
- Sequence of tuples $\{(n, d_n, k_n)\}_{n=1}^{\infty}$.

High-Dimensional Learning

- Consider a sequence of structure learning problems indexed by number of samples n.
- For each particular problem, we have data $\mathbf{x}^n = \{\mathbf{x}_i\}_{i=1}^n$.
- Each sample $\mathbf{x}_i \in \mathcal{X}^d$ 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

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

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

An Achievable Scaling Law for CLThres

Theorem ("Achievability")

Assume (A1) and (A2). Fix $\delta > 0$. Then if

$$n > \max \left\{ C_1 \log d, C_2 \log k, \right.$$

An Achievable Scaling Law for CLThres

Theorem ("Achievability")

Assume (A1) and (A2). Fix $\delta > 0$. Then if

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

An Achievable Scaling Law for CLThres

Theorem ("Achievability")

Assume (A1) and (A2). Fix $\delta > 0$. Then 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

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

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

$$\begin{aligned} d & \quad \text{subexponential} \\ k & \quad \text{subexponential} \\ d - k & \quad \text{subexponential} \end{aligned}$$

structure recovery is asymptotically possible.

Remarks on the Achievable Scaling Law for CLThres

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

$$\begin{aligned} d \quad & \text{subexponential} \\ k \quad & \text{subexponential} \\ d - k \quad & \text{subexponential} \end{aligned}$$

  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

$$\begin{aligned} d &\quad \text{subexponential} \\ k &\quad \text{subexponential} \\ d-k &\quad \text{subexponential} \end{aligned}$$

  structure recovery is asymptotically possible.

- d can grow much faster than n.

- Close to the strong converse lower bound.

Remarks on the Achievable Scaling Law for CLThres

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

$$\begin{aligned} d &\quad \text{subexponential} \\ k &\quad \text{subexponential} \\ d - k &\quad \text{subexponential} \end{aligned}$$

  structure recovery is asymptotically possible.

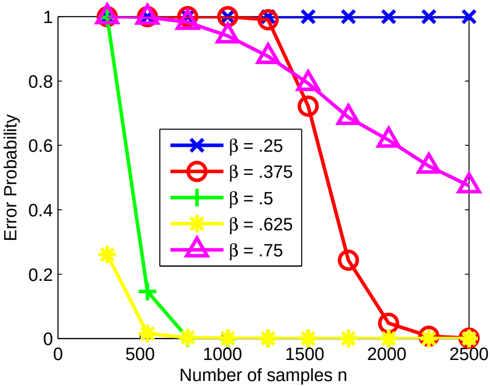
- d can grow much faster than n.

- Close to the strong converse lower bound.

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

Finite Number of Samples?

There exists a tradeoff between under- and overestimation in the finite-sample case:



Design of

$$\epsilon_n := n^{-\beta}$$

takes into account the tradeoff.

But asymptotically, overestimation error dominates.

Concluding Remarks

- Proposed a simple extension of Chow-Liu's max-weight spanning tree algorithm to learn forests consistently.

Concluding Remarks

- Proposed a simple extension of Chow-Liu's max-weight spanning tree algorithm to learn forests consistently.
- Derived precise error rates in the form of a "moderate deviations" result.

Concluding Remarks

- Proposed a simple extension of Chow-Liu's max-weight spanning tree algorithm to learn forests consistently.
- Derived precise error rates in the form of a "moderate deviations" result.
- Derived scaling laws on (n, d, k) for structural consistency in high dimensions.

Concluding Remarks

- Proposed a simple extension of Chow-Liu's max-weight spanning tree algorithm to learn forests consistently.
- Derived precise error rates in the form of a "moderate deviations" result.
- Derived scaling laws on (n, d, k) for structural consistency in high dimensions.

Extensions:

Concluding Remarks

- Proposed a simple extension of Chow-Liu's max-weight spanning tree algorithm to learn forests consistently.
- Derived precise error rates in the form of a "moderate deviations" result.
- Derived scaling laws on (n, d, k) for structural consistency in high dimensions.

Extensions:

- Risk consistency has also been analyzed. See manuscript on arXiv.

Concluding Remarks

- Proposed a simple extension of Chow-Liu's max-weight spanning tree algorithm to learn forests consistently.
- Derived precise error rates in the form of a "moderate deviations" result.
- Derived scaling laws on (n, d, k) for structural consistency in high dimensions.

Extensions:

- Risk consistency has also been analyzed. See manuscript on arXiv.
- Need to find the right balance between over- and underestimation for the finite sample case.