Error Exponents for Composite Hypothesis Testing of Markov Forest Distributions

Vincent Tan, Anima Anandkumar, Alan S. Willsky

Stochastic Systems Group,

Laboratory for Information and Decision Systems,

Massachusetts Institute of Technology

ISIT (Jun 18, 2010)

Motivation

- Continuation of line of work on error exponents for learning tree-structured graphical models:
  - Discrete Case: Tan, Anandkumar, Tong, Willsky, ISIT 2009.

Motivation

- Continuation of line of work on error exponents for learning tree-structured graphical models:
  - Discrete Case: Tan, Anandkumar, Tong, Willsky, ISIT 2009.
  - Gaussian Case: Tan, Anandkumar, Willsky, Trans. SP 2010.

Motivation

- Continuation of line of work on error exponents for learning tree-structured graphical models:
  - Discrete Case: Tan, Anandkumar, Tong, Willsky, ISIT 2009.
  - Gaussian Case: Tan, Anandkumar, Willsky, Trans. SP 2010.
- Instead of learning, we instead focus on hypothesis testing.

Motivation

- Continuation of line of work on error exponents for learning tree-structured graphical models:
  - Discrete Case: Tan, Anandkumar, Tong, Willsky, ISIT 2009.
  - Gaussian Case: Tan, Anandkumar, Willsky, Trans. SP 2010.
- Instead of learning, we instead focus on hypothesis testing.
- Provides intuition for which classes of graphical models are easy for learning in terms of the detection error exponent.

Motivation

- Continuation of line of work on error exponents for learning tree-structured graphical models:
  - Discrete Case: Tan, Anandkumar, Tong, Willsky, ISIT 2009.
  - Gaussian Case: Tan, Anandkumar, Willsky, Trans. SP 2010.
- Instead of learning, we instead focus on hypothesis testing.
- Provides intuition for which classes of graphical models are easy for learning in terms of the detection error exponent.
- Is there a relation between the detection error exponent and the exponent associated to structure learning?

Background on Tree-Structured Graphical Models

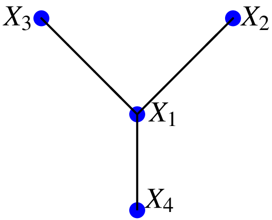
- Graphical model: family of multivariate probability distributions that factorize according to a given graph $G = (V, E)$.
- Vertices in the set $V = \{1, \ldots, d\}$ correspond to variables and edges in $E \subset \binom{V}{2}$ to conditional independences.

Background on Tree-Structured Graphical Models

- Graphical model: family of multivariate probability distributions that factorize according to a given graph $G = (V, E)$.
- Vertices in the set $V = \{1, \dots, d\}$ correspond to variables and edges in $E \subset \binom{V}{2}$ to conditional independences.
- Example for tree-structured $P(\mathbf{x})$ with $d = 4$.

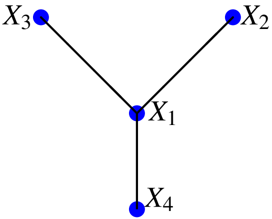
Background on Tree-Structured Graphical Models

- Graphical model: family of multivariate probability distributions that factorize according to a given graph $G = (V, E)$.
- Vertices in the set $V = \{1, \ldots, d\}$ correspond to variables and edges in $E \subset \binom{V}{2}$ to conditional independences.
- Example for tree-structured $P(\mathbf{x})$ with $d = 4$.



Background on Tree-Structured Graphical Models

- Graphical model: family of multivariate probability distributions that factorize according to a given graph $G = (V, E)$.
- Vertices in the set $V = \{1, \ldots, d\}$ correspond to variables and edges in $E \subset \binom{V}{2}$ to conditional independences.
- Example for tree-structured $P(\mathbf{x})$ with $d = 4$.



$$P(x_1, x_2, x_3, x_4) = P_1(x_1) \times \frac{P_{1,2}(x_1, x_2)}{P_1(x_1)} \times \frac{P_{1,3}(x_1, x_3)}{P_1(x_1)} \times \frac{P_{1,4}(x_1, x_4)}{P_1(x_1)}.$$

Learning vs Hypothesis Testing

- Canonical Problem: Given $\mathbf{x}_1, \ldots, \mathbf{x}_n \sim P$, learn structure of P.

Learning vs Hypothesis Testing

- Canonical Problem: Given $\mathbf{x}_1, \ldots, \mathbf{x}_n \sim P$, learn structure of P.
- If P is a tree, can use Chow and Liu (1968) as an efficient implementation of ML.

Learning vs Hypothesis Testing

- Canonical Problem: Given $\mathbf{x}_1, \ldots, \mathbf{x}_n \sim P$, learn structure of P.
- If P is a tree, can use Chow and Liu (1968) as an efficient implementation of ML.
- Denote set of distributions Markov on a tree $T_0 \in \mathcal{T}$ as $\mathcal{D}(T_0)$. Set of distributions Markov on any tree is $\mathcal{D}(\mathcal{T})$.

Learning vs Hypothesis Testing

- Canonical Problem: Given $\mathbf{x}_1, \ldots, \mathbf{x}_n \sim P$, learn structure of P.
- If P is a tree, can use Chow and Liu (1968) as an efficient implementation of ML.
- Denote set of distributions Markov on a tree $T_0 \in \mathcal{T}$ as $\mathcal{D}(T_0)$. Set of distributions Markov on any tree is $\mathcal{D}(\mathcal{T})$.
- Composite hypothesis testing problem considered here:

$$H_0 : \mathbf{x}_1, \ldots, \mathbf{x}_n \sim \Lambda_0 \subset \mathcal{D}(\mathcal{T})$$
$$H_1 : \mathbf{x}_1, \ldots, \mathbf{x}_n \sim \Lambda_1 \subset \mathcal{D}(\mathcal{T})$$

- Λ_i closed and $\Lambda_0 \cap \Lambda_1 = \emptyset$.

Definition of Worst-Case Type-II Error Exponent

- Neyman-Pearson setup. Acceptance regions $(\mathcal{A}_n)$.

Definition of Worst-Case Type-II Error Exponent

- Neyman-Pearson setup. Acceptance regions $(\mathcal{A}_n)$.
- Def: Type-II error exponent for a fixed $Q \in \Lambda_1$ given $(\mathcal{A}_n)$:

$$J(\Lambda_0, Q; \mathcal{A}_n) := \liminf_{n\to\infty} -\frac{1}{n} \log Q^n(\mathcal{A}_n)$$

Definition of Worst-Case Type-II Error Exponent

- Neyman-Pearson setup. Acceptance regions $(\mathcal{A}_n)$.
- Def: Type-II error exponent for a fixed $Q \in \Lambda_1$ given $(\mathcal{A}_n)$:

$$J(\Lambda_0, Q; \mathcal{A}_n) := \liminf_{n\to\infty} -\frac{1}{n} \log Q^n(\mathcal{A}_n)$$

- Def: Optimal Type-II error exponent

$$J^*(\Lambda_0, Q) := \sup_{\mathcal{A}_n : P^n(\mathcal{A}_n) \leq \alpha, \forall P \in \Lambda_0} J(\Lambda_0, Q; \mathcal{A}_n)$$

Definition of Worst-Case Type-II Error Exponent

- Neyman-Pearson setup. Acceptance regions $(\mathcal{A}_n)$.
- Def: Type-II error exponent for a fixed $Q \in \Lambda_1$ given $(\mathcal{A}_n)$:

$$J(\Lambda_0, Q; \mathcal{A}_n) := \liminf_{n\to\infty} -\frac{1}{n} \log Q^n(\mathcal{A}_n)$$

- Def: Optimal Type-II error exponent

$$J^*(\Lambda_0, Q) := \sup_{\mathcal{A}_n : P^n(\mathcal{A}_n) \le \alpha, \forall P \in \Lambda_0} J(\Lambda_0, Q; \mathcal{A}_n)$$

- Def: Worst-Case Optimal Type-II error exponent

$$J^*(\Lambda_0, \Lambda_1) := \inf_{Q \in \Lambda_1} J^*(\Lambda_0, Q)$$

Definition of Worst-Case Type-II Error Exponent

- Neyman-Pearson setup. Acceptance regions $(\mathcal{A}_n)$.
- Def: Type-II error exponent for a fixed $Q \in \Lambda_1$ given $(\mathcal{A}_n)$:

$$J(\Lambda_0, Q; \mathcal{A}_n) := \liminf_{n\to\infty} -\frac{1}{n} \log Q^n(\mathcal{A}_n)$$

- Def: Optimal Type-II error exponent

$$J^*(\Lambda_0, Q) := \sup_{\mathcal{A}_n : P^n(\mathcal{A}_n) \le \alpha, \forall P \in \Lambda_0} J(\Lambda_0, Q; \mathcal{A}_n)$$

- Def: Worst-Case Optimal Type-II error exponent

$$J^*(\Lambda_0, \Lambda_1) := \inf_{Q \in \Lambda_1} J^*(\Lambda_0, Q)$$

- Optimizing distribution Q^* called the least favorable distribution.

Why Difficult?

Why Difficult?

- Many trees: If there are d nodes, there are d^{d-2} trees!
- Searching for the dominant error event may be intractable.

Why Difficult?

- Many trees: If there are d nodes, there are d^{d-2} trees!
- Searching for the dominant error event may be intractable.

Natural Questions:

- Any closed-form expressions for the worst-case error exponent for special Λ_0, Λ_1?
- How does this depend on the true distribution?

Why Difficult?

- Many trees: If there are d nodes, there are d^{d-2} trees!
- Searching for the dominant error event may be intractable.

Natural Questions:

- Any closed-form expressions for the worst-case error exponent for special Λ_0, Λ_1?
- How does this depend on the true distribution?
- Connections to learning?
- Intuition and characterization of the least favorable distribution?

A Simplification

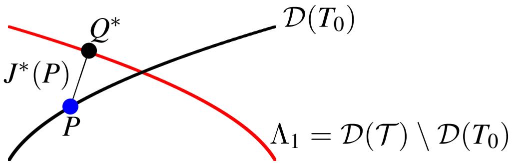
Assume that H_0 is simple and P is Markov on $T_0 = (V, E_0)$.

$$H_0 : \mathbf{x}_1, \ldots, \mathbf{x}_n \sim \{P\}$$
$$H_1 : \mathbf{x}_1, \ldots, \mathbf{x}_n \sim \Lambda_1 = \mathcal{D}(\mathcal{T}) \setminus \mathcal{D}(T_0)$$

A Simplification

Assume that H_0 is simple and P is Markov on $T_0 = (V, E_0)$.

$$H_0 : \mathbf{x}_1, \ldots, \mathbf{x}_n \sim \{P\}$$
$$H_1 : \mathbf{x}_1, \ldots, \mathbf{x}_n \sim \Lambda_1 = \mathcal{D}(\mathcal{T}) \setminus \mathcal{D}(T_0)$$



$J^*(P) := J^*(\{P\}, \mathcal{D}(\mathcal{T}) \setminus \mathcal{D}(T_0))$

Setup for Main Result

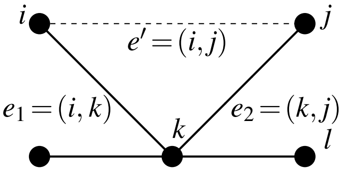
For a non-edge $e' = (i,j)$, let $\mathrm{Path}(e')$ be the unique path joining i and j.

Let $L(i,j)$ be the number of hops between i and j.

Setup for Main Result

For a non-edge $e' = (i,j)$, let $\mathrm{Path}(e')$ be the unique path joining i and j.

Let $L(i,j)$ be the number of hops between i and j.



Setup for Main Result

For a non-edge $e' = (i,j)$, let $\mathrm{Path}(e')$ be the unique path joining i and j.

Let $L(i,j)$ be the number of hops between i and j.

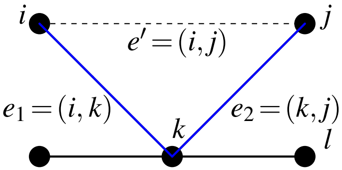


Figure: $\mathrm{Path}(e') = \{(i,k),(k,j)\}$

Setup for Main Result

For a non-edge $e' = (i,j)$, let $\mathrm{Path}(e')$ be the unique path joining i and j.

Let $L(i,j)$ be the number of hops between i and j.

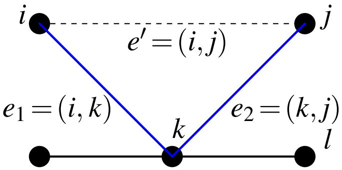


Figure: $\mathrm{Path}(e') = \{(i,k),(k,j)\}$

Mutual information of joint distribution $P_e = P_{i,j}$ denoted as $I(P_e)$.

Main Result

Proposition

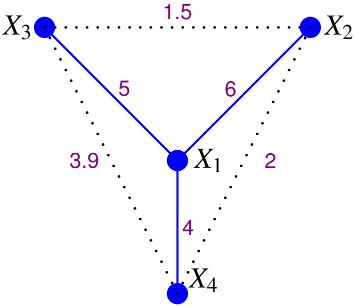
$$J^*(P) = \min_{\substack{e'=(i,j)\notin E_0 \\ L(i,j)=2}} \; \min_{e\in \mathrm{Path}(e')} \{I(P_e) - I(P_{e'})\},$$

Main Result

Proposition

$$J^*(P) = \min_{\substack{e'=(i,j)\notin E_0 \\ L(i,j)=2}} \min_{e\in \mathrm{Path}(e')} \{I(P_e) - I(P_{e'})\},$$

Illustration:

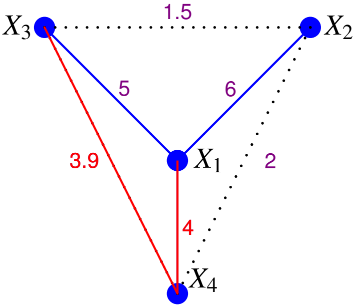


Main Result

Proposition

$$J^*(P) = \min_{\substack{e'=(i,j)\notin E_0 \\ L(i,j)=2}} \; \min_{e\in \mathrm{Path}(e')} \{I(P_e) - I(P_{e'})\},$$

Illustration:

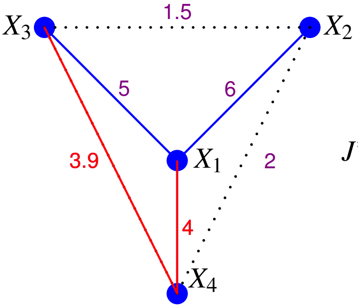


Main Result

Proposition

$$J^*(P) = \min_{\substack{e'=(i,j)\notin E_0 \\ L(i,j)=2}} \min_{e\in \mathrm{Path}(e')} \{I(P_e) - I(P_{e'})\},$$

Illustration:



$$J^*(P) = 4 - 3.9 = 0.1$$

Least Favorable Distribution

The least favorable distribution Q^* is characterized by

$$E_{Q^*} = \operatorname*{argmax}_{E \neq E_0,\, E \text{ acyclic}} \sum_{e \in E} I(P_e)$$

a second-best max-weight spanning tree problem,

Least Favorable Distribution

The least favorable distribution Q^* is characterized by

$$E_{Q^*} = \operatorname*{argmax}_{E \neq E_0,\, E \text{ acyclic}} \sum_{e \in E} I(P_e)$$

a second-best max-weight spanning tree problem, and

$$Q_i^*(x_i) = P_i(x_i), \qquad \forall\, i \in V$$
$$Q_{i,j}^*(x_i, x_j) = P_{i,j}(x_i, x_j), \qquad \forall\, (i,j) \in E_{Q^*}$$

Proof Outline

- Optimization for worst-case exponent is

$$\inf_{Q \in \mathcal{D}(\mathcal{T}) \setminus \mathcal{D}(\{T_0\})} D(Q \,||\, P)$$

Proof Outline

- Optimization for worst-case exponent is

$$\inf_{Q\in\mathcal{D}(\mathcal{T})\setminus\mathcal{D}(\{T_0\})} D(Q\,||\,P) = \min_{T\in\mathcal{T}\setminus\{T_0\}} \left[\inf_{Q\in\mathcal{D}(T)} D(Q\,||\,P)\right]$$

Proof Outline

- Optimization for worst-case exponent is

$$\inf_{Q \in \mathcal{D}(\mathcal{T}) \setminus \mathcal{D}(\{T_0\})} D(Q \,||\, P) = \min_{T \in \mathcal{T} \setminus \{T_0\}} \left[\inf_{Q \in \mathcal{D}(T)} D(Q \,||\, P) \right]$$

- Use tree decomposition (junction tree theorem)

$$Q(\mathbf{x}) = \prod_{i \in V(T)} Q_i(x_i) \prod_{(i,j) \in E(T)} \frac{Q_{i,j}(x_i, x_j)}{Q_i(x_i) Q_j(x_j)}$$

Proof Outline

- Optimization for worst-case exponent is

$$\inf_{Q \in \mathcal{D}(\mathcal{T}) \setminus \mathcal{D}(\{T_0\})} D(Q \,||\, P) = \min_{T \in \mathcal{T} \setminus \{T_0\}} \left[\inf_{Q \in \mathcal{D}(T)} D(Q \,||\, P) \right]$$

- Use tree decomposition (junction tree theorem)

$$Q(\mathbf{x}) = \prod_{i \in V(T)} Q_i(x_i) \prod_{(i,j) \in E(T)} \frac{Q_{i,j}(x_i, x_j)}{Q_i(x_i) Q_j(x_j)}$$

- Emulate Chow and Liu (1968).

Proof Outline

- Optimization for worst-case exponent is

$$\inf_{Q \in \mathcal{D}(\mathcal{T}) \setminus \mathcal{D}(\{T_0\})} D(Q \,||\, P) = \min_{T \in \mathcal{T} \setminus \{T_0\}} \left[\inf_{Q \in \mathcal{D}(T)} D(Q \,||\, P) \right]$$

- Use tree decomposition (junction tree theorem)

$$Q(\mathbf{x}) = \prod_{i \in V(T)} Q_i(x_i) \prod_{(i,j) \in E(T)} \frac{Q_{i,j}(x_i, x_j)}{Q_i(x_i) Q_j(x_j)}$$

- Emulate Chow and Liu (1968).
- Second-best max-weight spanning tree differs from best one by a single edge [Cormen et al. 2003].

Proof Outline

- Optimization for worst-case exponent is

$$\inf_{Q \in \mathcal{D}(\mathcal{T}) \setminus \mathcal{D}(\{T_0\})} D(Q \,||\, P) = \min_{T \in \mathcal{T} \setminus \{T_0\}} \left[\inf_{Q \in \mathcal{D}(T)} D(Q \,||\, P) \right]$$

- Use tree decomposition (junction tree theorem)

$$Q(\mathbf{x}) = \prod_{i \in V(T)} Q_i(x_i) \prod_{(i,j) \in E(T)} \frac{Q_{i,j}(x_i, x_j)}{Q_i(x_i) Q_j(x_j)}$$

- Emulate Chow and Liu (1968).
- Second-best max-weight spanning tree differs from best one by a single edge [Cormen et al. 2003].
- Data processing inequality.

Intuition

$$J^*(P) = \min_{\substack{e'=(i,j)\notin E_0 \\ L(i,j)=2}} \min_{e \in \mathrm{Path}(e')} \{I(P_e) - I(P_{e'})\},$$

- Smaller the difference between MI on true edge and MI on non-edge (along path), smaller the detection error exponent.

Intuition

$$J^*(P) = \min_{\substack{e'=(i,j)\notin E_0 \\ L(i,j)=2}} \min_{e \in \text{Path}(e')} \{I(P_e) - I(P_{e'})\},$$

- Smaller the difference between MI on true edge and MI on non-edge (along path), smaller the detection error exponent.
- Detection error exponent depends only on bottleneck edges.

Comparison to Existing Results

$$J^*(P) = \min_{\substack{e'=(i,j)\notin E_0 \\ L(i,j)=2}} \ \min_{e\in\mathrm{Path}(e')} \{I(P_e) - I(P_{e'})\},$$

- Intuitive in light of the Chow-Liu algorithm for learning trees.

$$\hat{E}_{\mathrm{ML}} := \operatorname*{argmax}_{E\neq E_0,\, E \text{ acyclic}} \sum_{e\in E} I(\hat{\mu}_e)$$

where $\hat{\mu}_e$ is the pairwise type on edge e.

Comparison to Existing Results

$$J^*(P) = \min_{\substack{e'=(i,j)\notin E_0 \\ L(i,j)=2}} \; \min_{e \in \mathrm{Path}(e')} \{I(P_e) - I(P_{e'})\},$$

- Intuitive in light of the Chow-Liu algorithm for learning trees.

$$\hat{E}_{\mathrm{ML}} := \operatorname*{argmax}_{E \neq E_0,\, E \text{ acyclic}} \sum_{e \in E} I(\hat{\mu}_e)$$

  where $\hat{\mu}_e$ is the pairwise type on edge e.

- Learning error exponent in very-noisy regime

$$\widetilde{K}(P) := \min_{e' \notin E_0} \; \min_{e \in \mathrm{Path}(e')} \frac{(I(P_e) - I(P_{e'}))^2}{2\mathrm{Var}(S_e - S_{e'})}$$

Comparison to Existing Results

$$J^*(P) = \min_{\substack{e'=(i,j)\notin E_0 \\ L(i,j)=2}} \min_{e \in \mathrm{Path}(e')} \{I(P_e) - I(P_{e'})\},$$

- Intuitive in light of the Chow-Liu algorithm for learning trees.

$$\hat{E}_{\mathrm{ML}} := \operatorname*{argmax}_{E \neq E_0,\, E \text{ acyclic}} \sum_{e \in E} I(\hat{\mu}_e)$$

  where $\hat{\mu}_e$ is the pairwise type on edge e.

- Learning error exponent in very-noisy regime

$$\widetilde{K}(P) := \min_{e' \notin E_0} \min_{e \in \mathrm{Path}(e')} \frac{(I(P_e) - I(P_{e'}))^2}{2\mathrm{Var}(S_e - S_{e'})}$$

- $J^*(P)$ and $\widetilde{K}(P)$ depend on the difference of mutual informations.

Performing the Hypothesis Test

- Known that the worst-case error exponent is achieved by the Hoeffding Test.
- But hard to implement for tree distributions.

Performing the Hypothesis Test

- Known that the worst-case error exponent is achieved by the Hoeffding Test.
- But hard to implement for tree distributions.
- The generalized likelihood ratio test (GLRT) has acceptance regions

$$\mathcal{A}_n := \left\{ \mathbf{x}^n : \frac{1}{n} \log \frac{\max_{Q \in \Lambda_1} Q^n(\mathbf{x}^n)}{\max_{P \in \Lambda_0} P^n(\mathbf{x}^n)} \geq \gamma \right\}$$

Performing the Hypothesis Test

- Known that the worst-case error exponent is achieved by the Hoeffding Test.
- But hard to implement for tree distributions.
- The generalized likelihood ratio test (GLRT) has acceptance regions

$$\mathcal{A}_n := \left\{ \mathbf{x}^n : \frac{1}{n} \log \frac{\max_{Q \in \Lambda_1} Q^n(\mathbf{x}^n)}{\max_{P \in \Lambda_0} P^n(\mathbf{x}^n)} \geq \gamma \right\}$$

- When the null hypothesis is simple, the GLRT also simplifies.

Performing the Hypothesis Test

- Known that the worst-case error exponent is achieved by the Hoeffding Test.
- But hard to implement for tree distributions.
- The generalized likelihood ratio test (GLRT) has acceptance regions

$$\mathcal{A}_n := \left\{ \mathbf{x}^n : \frac{1}{n} \log \frac{\max_{Q \in \Lambda_1} Q^n(\mathbf{x}^n)}{\max_{P \in \Lambda_0} P^n(\mathbf{x}^n)} \ge \gamma \right\}$$

- When the null hypothesis is simple, the GLRT also simplifies.

$$H_0 : \mathbf{x}_1, \ldots, \mathbf{x}_n \sim \{P\}$$
$$H_1 : \mathbf{x}_1, \ldots, \mathbf{x}_n \sim \Lambda_1 = \mathcal{D}(\mathcal{T}) \setminus \mathcal{D}(T_0)$$

The Generalized Likelihood Ratio Test

- Denote the joint type of $\mathbf{x}^n$ as $\hat{\mu} := \hat{\mu}(\cdot\,;\,\mathbf{x}^n)$.
- Denote the pairwise type on e as $\hat{\mu}_e$.
- True set of edges: E_0.

Proposition

The GLRT simplifies as

$$\mathcal{A}_n = \left\{ \mathbf{x}^n : \sum_{e \in E^*} I(\hat{\mu}_e) - \sum_{e \in E_0} I(\hat{\mu}_e) \geq \gamma \right\}$$

where the "dominating edge set" is

$$E^* = \underset{E \neq E_0,\, E \text{ acyclic}}{\operatorname{argmax}} \sum_{e \in E} I(\hat{\mu}_e)$$

Interpretation and Extensions

- Easy to implement the GLRT for testing between trees.

[1] VTan, A. Anandkumar, A. Willsky "Learning High-Dimensional Markov Forest Distributions: Analysis of Error Rates", Submitted to JMLR, May 2010.

Interpretation and Extensions

- Easy to implement the GLRT for testing between trees.
- Can find the tree structure E^* efficiently once pairwise types $\hat{\mu}_e$ have been computed.

[1] VTan, A. Anandkumar, A. Willsky "Learning High-Dimensional Markov Forest Distributions: Analysis of Error Rates", Submitted to JMLR, May 2010.

Interpretation and Extensions

- Easy to implement the GLRT for testing between trees.
- Can find the tree structure E^* efficiently once pairwise types $\hat{\mu}_e$ have been computed.
- Extensions to forest-structured distributions for error exponent and GLRT are straightforward.

[1] VTan, A. Anandkumar, A. Willsky "Learning High-Dimensional Markov Forest Distributions: Analysis of Error Rates", Submitted to JMLR, May 2010.

Interpretation and Extensions

- Easy to implement the GLRT for testing between trees.
- Can find the tree structure E^* efficiently once pairwise types $\hat{\mu}_e$ have been computed.
- Extensions to forest-structured distributions for error exponent and GLRT are straightforward.
- Recent work on high-dimensional learning of forest-structured distributions. [1]

[1] VTan, A. Anandkumar, A. Willsky "Learning High-Dimensional Markov Forest Distributions: Analysis of Error Rates", Submitted to JMLR, May 2010.

Concluding Remarks

- Analyzed the worst-case type-II error exponent for composite hypothesis testing of Markov forest distributions.
- Close relations to learning.

Concluding Remarks

- Analyzed the worst-case type-II error exponent for composite hypothesis testing of Markov forest distributions.
- Close relations to learning.
- Possible extension 1: Bayesian formulation (Chernoff Information).

Concluding Remarks

- Analyzed the worst-case type-II error exponent for composite hypothesis testing of Markov forest distributions.
- Close relations to learning.
- Possible extension 1: Bayesian formulation (Chernoff Information).
- Possible extension 2: Decomposable graphical models.