Learning Max-Weight Discriminative Forests

Vincent Tan [1] John W. Fisher III [1,2] Alan S. Willsky [1]

[1]Stochastic Systems Group, LIDS, EECS
Massachusetts Institute of Technology

[2]CSAIL, EECS,
Massachusetts Institute of Technology

ICASSP (April 3, 2008)

Outline

1. Background

Outline

Outline

Outline

1. Background
2. Motivation and Problem Statement
3. Formulation and Optimization of Objective
4. Numerical Results

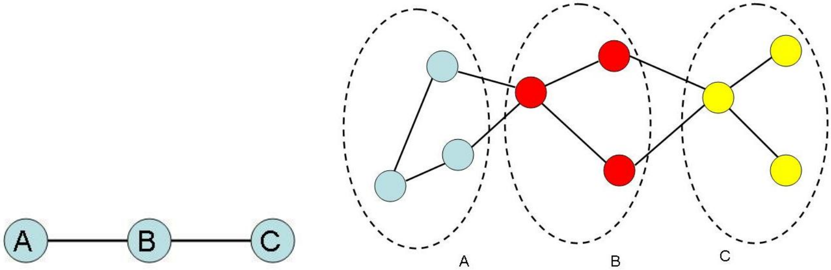
Outline

Graphical Models

Graphical Models

$p(x)$ can be defined on an undirected graph $\mathcal{G}$.

$\mathcal{G} = (V, E)$ encodes conditional independencies.



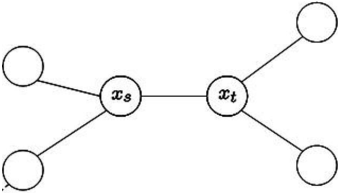
(a) $p(A, C|B) = p(A|B)p(C|B)$

(b) $p(x_A, x_C|x_B) = p(x_A|x_B)p(x_C|x_B)$

Figure: Graphical Models

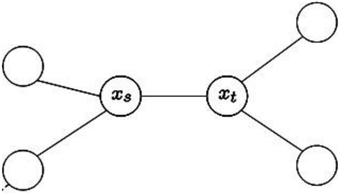
Tree Structured Distributions

- A tree structured distribution $p(x)$ has no loops.



Tree Structured Distributions

- A tree structured distribution $p(x)$ has no loops.



- Trees can be decomposed into node and pairwise terms.

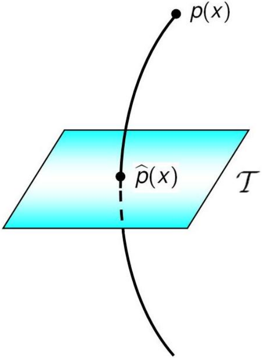
$$p(x) = \prod_{s \in V} p(x_s) \prod_{(s,t) \in E} \frac{p(x_s, x_t)}{p(x_s)p(x_t)} \quad (1)$$

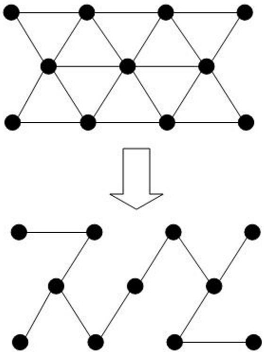
  - Marginal properties on vertex set.
  - Pairwise relationships on edge set.

The Chow-Liu algorithm I

Problem: Fit a tree to a given distribution. [Chow-Liu 1968]

$$\widehat{p}(x) = \operatorname*{argmin}_{\widehat{p} \in \mathcal{T}} D(p(x) \,\|\, \widehat{p}(x)) \tag{2}$$





The Chow-Liu algorithm II

Solution: Max-Weight Spanning Tree (MWST) [Chow-Liu 1968]

The Chow-Liu algorithm II

Solution: Max-Weight Spanning Tree (MWST) [Chow-Liu 1968]

Edge weights = Mutual Information (MI) between variables.

$$I(x_s\,;\,x_t) = \int_{\mathcal{X}^2} p(x_s, x_t) \log \left[\frac{p(x_s, x_t)}{p(x_s)p(x_t)} \right] \, dx_s \, dx_t \quad (3)$$

The Chow-Liu algorithm II

Solution: Max-Weight Spanning Tree (MWST) [Chow-Liu 1968]

Edge weights = Mutual Information (MI) between variables.

$$I(x_s\,;\,x_t) = \int_{\mathcal{X}^2} p(x_s, x_t) \log \left[\frac{p(x_s, x_t)}{p(x_s)p(x_t)} \right] dx_s\, dx_t \tag{3}$$

Proof.

Direct consequence of decomposition of $p(x)$ for tree models. □

The Chow-Liu algorithm II

Solution: Max-Weight Spanning Tree (MWST) [Chow-Liu 1968]

Edge weights = Mutual Information (MI) between variables.

$$I(x_s\,;\,x_t) = \int_{\mathcal{X}^2} p(x_s, x_t) \log\left[\frac{p(x_s, x_t)}{p(x_s)p(x_t)}\right]\, dx_s\, dx_t \tag{3}$$

Proof.

Direct consequence of decomposition of $p(x)$ for tree models. □

Generative learning.

Learning Reduced-Order probability models

Learning Reduced-Order probability models

- Motivation:

  Can sparse graphical models of increasing complexity be learned better if intended purpose is known?

Learning Reduced-Order probability models

- Motivation:

  Can sparse graphical models of increasing complexity be learned better if intended purpose is known?

- Define $\mathcal{T}^{(k)}$ to be the set of trees with no more than $k \leq n-1$ edges.

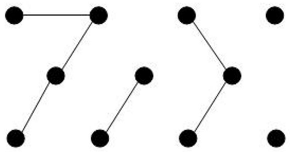


Figure: Tree defined on $n = 11$ nodes with $k = 6$ edges

Learning Reduced-Order probability models

Problem Statement:

Given p, q, sequentially learn lower-order models $\widehat{p}^{(k)}, \widehat{q}^{(k)} \in \mathcal{T}^{(k)}$.

These models are to be used specifically for binary hypothesis testing.

Learning Reduced-Order probability models

Problem Statement:

Given p, q, sequentially learn lower-order models $\widehat{p}^{(k)}, \widehat{q}^{(k)} \in \mathcal{T}^{(k)}$.

These models are to be used specifically for binary hypothesis testing.

Hypotheses H_0 and H_1,

$$H_0 \ : \ x \sim p \quad \text{or} \quad H_1 \ : \ x \sim q \tag{4}$$

Learning Reduced-Order probability models

Problem Statement:

Given p, q, sequentially learn lower-order models $\widehat{p}^{(k)}, \widehat{q}^{(k)} \in \mathcal{T}^{(k)}$.

These models are to be used specifically for binary hypothesis testing.

Hypotheses H_0 and H_1,

$$H_0 \;:\; x \sim p \quad \text{or} \quad H_1 \;:\; x \sim q \tag{4}$$

A Likelihood Ratio Test is used to classify new samples e.g. for x_{test}

$$\frac{\widehat{p}^{(k)}(x_{test})}{\widehat{q}^{(k)}(x_{test})} \underset{\text{declare } H_1}{\overset{\text{declare } H_0}{\gtrless}} 1. \tag{5}$$

The Objective Function

The Objective Function

- We maximize the J-divergence.

$$J(p(x), q(x)) = D(p(x) \,\|\, q(x)) + D(q(x) \,\|\, p(x)). \qquad (6)$$

over all possible structures $E_{\widehat{p}^{(k)}}$ and $E_{\widehat{q}^{(k)}}$.

The Objective Function

- We maximize the J-divergence.

$$J(p(x), q(x)) = D(p(x) \,\|\, q(x)) + D(q(x) \,\|\, p(x)). \qquad (6)$$

over all possible structures $E_{\widehat{p}^{(k)}}$ and $E_{\widehat{q}^{(k)}}$.

- Bounds the $\Pr(\mathrm{err})$ [Basseville 1989].

$$\frac{1}{2}\min(P_0, P_1)e^{-J} \le \Pr(\mathrm{err}) \le \sqrt{P_0 P_1}\left(\frac{J}{4}\right)^{-1/4}, \qquad (7)$$

The Objective Function

- We maximize the J-divergence.

$$J(p(x), q(x)) = D(p(x) \,\|\, q(x)) + D(q(x) \,\|\, p(x)). \tag{6}$$

over all possible structures $E_{\widehat{p}^{(k)}}$ and $E_{\widehat{q}^{(k)}}$.

- Bounds the $\mathrm{Pr}(\mathrm{err})$ [Basseville 1989].

$$\frac{1}{2}\min(P_0, P_1)e^{-J} \leq \mathrm{Pr}(\mathrm{err}) \leq \sqrt{P_0 P_1}\left(\frac{J}{4}\right)^{-1/4}, \tag{7}$$

- Discriminative learning.

The J-divergence

Lemma

The J-divergence of $\widehat{p}$ and $\widehat{q}$ is

$$J(\widehat{p}, \widehat{q}; p, q) = \sum_{s \in V} J(p_s, q_s) + \sum_{(s,t) \in E_{\widehat{p}} \cup E_{\widehat{q}}} w_{st} \quad (8)$$

where w_{st} are multi-valued weights.

The J-divergence

Lemma

The J-divergence of $\widehat{p}$ and $\widehat{q}$ is

$$J(\widehat{p},\widehat{q};p,q) = \sum_{s\in V} J(p_s,q_s) + \sum_{(s,t)\in E_{\widehat{p}}\cup E_{\widehat{q}}} w_{st} \tag{8}$$

where w_{st} are multi-valued weights.

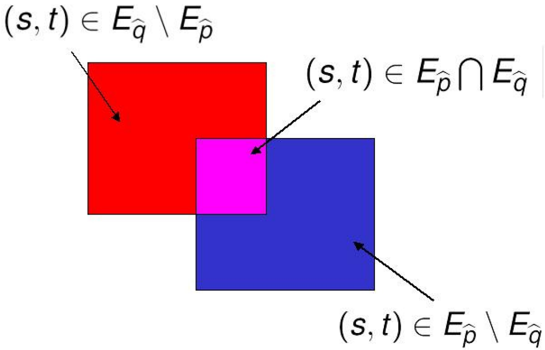
Proof.

Direct consequence of decomposition of $p(x)$ for tree models. □

Multi-valued weights

w_{st} are multi-valued weights.

The expression for w_{st} differs for the three cases



A modified MWST algorithm I

Lemma

$\widehat{p}^{(k)}(x), \widehat{q}^{(k)}(x)$ are optimally chosen via a modified version of the 'k-edge' MWST (Kruskal's) algorithm with edge weights given by w_{st}.

A modified MWST algorithm I

Lemma

$\widehat{p}^{(k)}(x), \widehat{q}^{(k)}(x)$ are optimally chosen via a modified version of the 'k-edge' MWST (Kruskal's) algorithm with edge weights given by w_{st}.

Kruskal's algorithm is of particular interest because:

1. Greedy.
2. Yields a sequence of optimal k-edge optimal forests.

A modified MWST algorithm II

Consider the maximum of the three possible values for w_{st}.

A modified MWST algorithm II

Consider the maximum of the three possible values for w_{st}.

If w_{st} is maximized:

A modified MWST algorithm II

Consider the maximum of the three possible values for w_{st}.

If w_{st} is maximized:

1. $(s,t) \in E_{\widehat{p}} \setminus E_{\widehat{q}}$,
   $\Rightarrow$ Place an edge between s, t for $\widehat{p}$ and not $\widehat{q}$.

A modified MWST algorithm II

Consider the maximum of the three possible values for w_{st}.

If w_{st} is maximized:

1. $(s,t) \in E_{\widehat{p}} \setminus E_{\widehat{q}}$,
   $\Rightarrow$ Place an edge between s,t for $\widehat{p}$ and not $\widehat{q}$.
2. $(s,t) \in E_{\widehat{q}} \setminus E_{\widehat{p}}$,
   $\Rightarrow$ Place an edge between s,t for $\widehat{q}$ and not $\widehat{p}$.

A modified MWST algorithm II

Consider the maximum of the three possible values for w_{st}.

If w_{st} is maximized:

1. $(s,t) \in E_{\widehat{p}} \setminus E_{\widehat{q}}$,
   $\Rightarrow$ Place an edge between s, t for $\widehat{p}$ and not $\widehat{q}$.
2. $(s,t) \in E_{\widehat{q}} \setminus E_{\widehat{p}}$,
   $\Rightarrow$ Place an edge between s, t for $\widehat{q}$ and not $\widehat{p}$.
3. $(s,t) \in E_{\widehat{p}} \bigcap E_{\widehat{q}}$,
   $\Rightarrow$ Place an edge between s, t for both $\widehat{p}$ and $\widehat{q}$.

A modified MWST algorithm II

Consider the maximum of the three possible values for w_{st}.

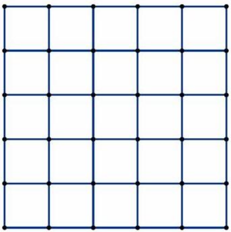
If w_{st} is maximized:

1. $(s,t) \in E_{\widehat{p}} \setminus E_{\widehat{q}}$,
   $\Rightarrow$ Place an edge between s, t for $\widehat{p}$ and not $\widehat{q}$.
2. $(s,t) \in E_{\widehat{q}} \setminus E_{\widehat{p}}$,
   $\Rightarrow$ Place an edge between s, t for $\widehat{q}$ and not $\widehat{p}$.
3. $(s,t) \in E_{\widehat{p}} \bigcap E_{\widehat{q}}$,
   $\Rightarrow$ Place an edge between s, t for both $\widehat{p}$ and $\widehat{q}$.

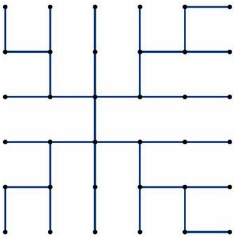
Possibility of early termination.

Example I: Probability Models

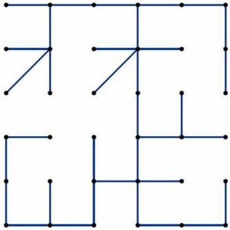
Example I: Probability Models



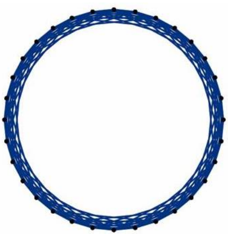
(a) Original Grid p



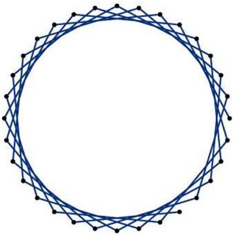
(b) Gen. Grid $\widehat{p}^{(n-1)}$



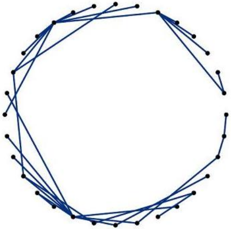
(c) Discri. Grid $\widehat{p}^{(n-1)}$



(d) Original Cycle p

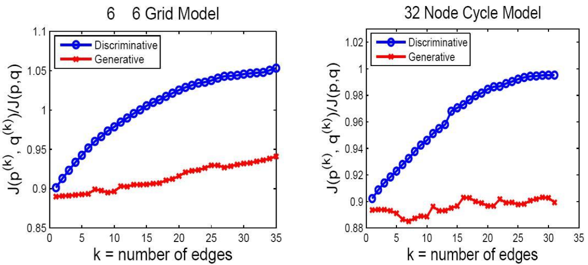


(e) Gen. Cycle $\widehat{p}^{(n-1)}$

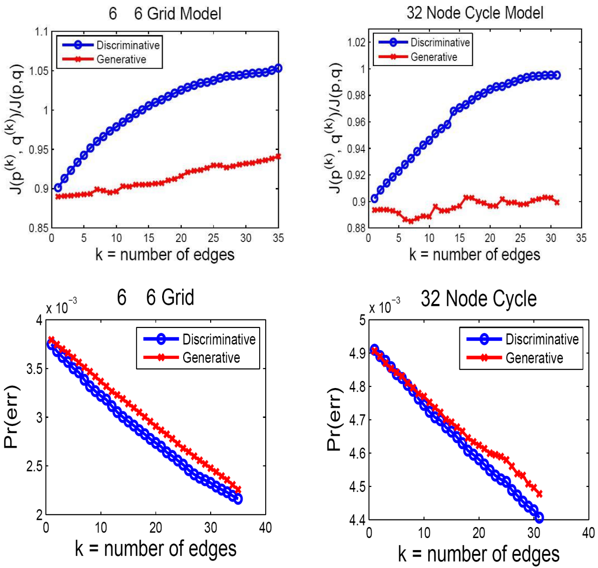


(f) Discri. Cycle $\widehat{p}^{(n-1)}$

J-divergence and Probability of Error



J-divergence and Probability of Error



Example II: Class Conditional Covariance Matrices

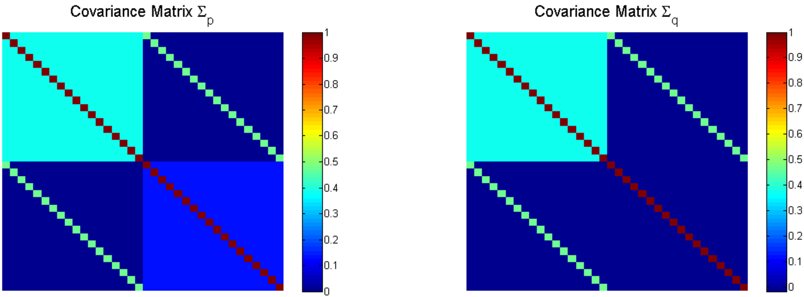


Figure: p, q are zero-mean Gaussian with covariance matrices Σ_p, Σ_q.

Discriminative information comes from the lower-right block.

J-divergence and Probability of Error

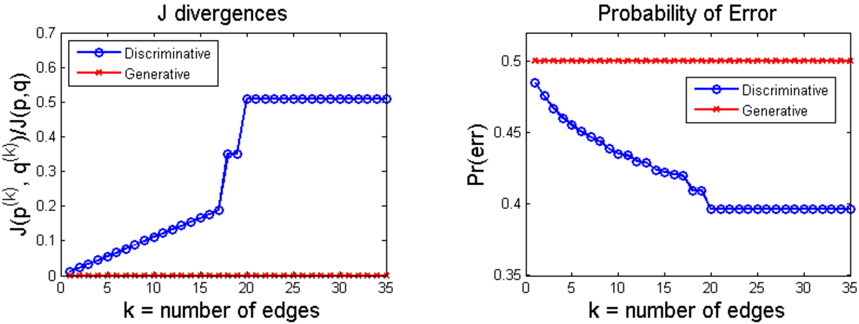


Figure: $J(\widehat{p}^{(k)}(x), \widehat{q}^{(k)}(x))/J(p,q)$ and the $\Pr(err)$ as functions of k.

Graph Structures

How do the structures of $\widehat{p}^{(k)}(x)$ compare under generative and discriminative learning?

Graph Structures

How do the structures of $\widehat{p}^{(k)}(x)$ compare under generative and discriminative learning?

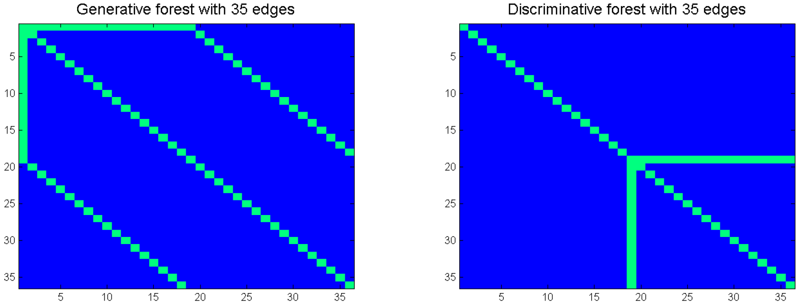


Figure: Structures of $\widehat{p}^{(k)} \in \mathcal{T}^{(k)}$ represented by Adjacency Matrices

Conclusions and Perspectives

Conclusions and Perspectives

- Many machine learning (classification) problems are more effective if discriminative features are identified.

Conclusions and Perspectives

- Many machine learning (classification) problems are more effective if discriminative features are identified.
- Graphical models can be learned better if their intended purpose is known.

Conclusions and Perspectives

- Many machine learning (classification) problems are more effective if discriminative features are identified.
- Graphical models can be learned better if their intended purpose is known.
- We have learned increasingly complex (and nested) models $\widehat{p}^{(k)}, \widehat{q}^{(k)}$ sequentially for hypothesis testing.

Conclusions and Perspectives

- Many machine learning (classification) problems are more effective if discriminative features are identified.
- Graphical models can be learned better if their intended purpose is known.
- We have learned increasingly complex (and nested) models $\widehat{p}^{(k)}, \widehat{q}^{(k)}$ sequentially for hypothesis testing.
- Discriminative learning reduces $\Pr(\mathrm{err})$.