Georgia Institute of Technology

CSIP Center for Signal & Image Processing

ECE 8803

Structure Learning in Graphical Models

Module 11:
Bayesian and Markov Random Fields:

1. Unknown Structure &
2. Fully Observed Variables

Faramarz Fekri

Center for Signal and Information Processing

Overview

- Learning Structure in Graphical Models (BN & MRF):
  - Fully Observed and Unknown Structure:

  – Structure Learning: High-Level Methods
  – Likelihood Score for BN
  – Bayesian Score for BN
  – Learning Tree Structures: Chow-Liu algorithm

  – Structure Learning in MRF: Gaussian Graphical Models
  – Graphical Lasso

Chapter 18 & 20 in Koller, Chapter 26 in K. Murphy

Why Structure Learning?

- Structure Discovery: the underlying dependency structure of the domain (i.e., a perfect map G^* for distribution P^*)
  - Not just statistical correlations between individual variables, but also detect direct vs. indirect correlations
  - However: at best, we can recover the structure up to the I-equivalence class to G^*.
- Density estimation
  - Estimate a statistical model of the underlying distribution P^* so that we can do inference and prediction for new instances (i.e., G to be I-map of P^*)
    - Need infinite data
  - Instead, at best, we want a simple G (i.e. P) that is a good approximate to P^*

Structure Learning: High-Level Methods

- Constraint based methods (Single Model learning)
  - Obtain a network that best explains dependencies in data
  - Limitation:
    - sensitive to errors in single dependencies
    - Not applicable to large networks
- Score based methods (Single Model learning) (focus of our course)
  - View learning as a model selection problem
    - Define a scoring function (via Likelihood or Bayesian based score) as to how well the model fits the data
    - Search for a high-scoring network structure
    - Limitation: super-exponential search space
- Bayesian model averaging (ensemble learning: mostly for structure discovery)
  - Average predictions across all structure candidates
  - Limitation: very complex and the gain over single model is subtle.

Optimality?

- By optimality we mean that the employed algorithms guarantee to return a structure that maximizes the objectives (e.g., Likelihood)
- Note that many heuristics used to be popular (structured EM, Module network, greedy structural search(add/delete edge)), but they provide no guarantee on optimality.
- Our course will consider two classes of methods with guaranteed structure learning, but only applicable to certain families of graphs:
  - Trees: The Chow-Liu algorithm
  - Pairwise MRFs: The neighborhood-selection method (covariance selection)

Complexity of Structural Search

- How many graphs over *n* nodes?
  - There are (counting isomorphic graphs different)

$$2^{\binom{n}{2}} \Rightarrow O(2^{n^2})$$

  - How many DAG of N nodes? $\prod_{n=1}^{N-1} 2^n = 2^{N(N-1)/2}$

- How many trees over *n* nodes? $O(2^{nlogn})$
  - significantly less than $O(2^{n^2})$

- We can find exact solution of an optimal tree (under MLE) via Chow-Liu algorithm!
  - Noting that MLE score (i.e., loss function) decomposable (will see this) to edge-related elements
  - Noting that in a tree, each node has only one parent!

Information Theory Quantities

- Mutual information

$$I(X_i, X_j) = \sum_{x_i, x_j} P(x_i, x_j) \log \frac{P(x_i, x_j}{P(x_i) P(x_j)}$$

- A "distance" away from independence
  - X_i and X_j are independent if and only if $I(X_i, X_j) = 0$

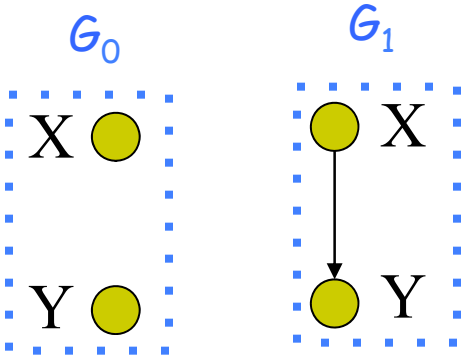
- Given M iid data point $D = \{x^l\}$, $\hat{P}(x_i, x_j) = \frac{\#(x_i, x_j)}{M}$
  - Empirical Mutual Information:

$$\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \hat{p}(x_i, x_j) \log \frac{\hat{p}(x_i, x_j)}{\hat{p}(x_i)\, \hat{p}(x_j)}$$

Likelihood Score for BN

- Goal: find (G,Θ) that maximize the likelihood of data (M observed data).
  - $Score_L(G:D) = \log P(D \mid G, \Theta'_G)$ where Θ'_G is MLE for G
  - Find $G^* = \arg\max_G Score_L(G:D)$

Consider two-node scenario:



$$Score_L(G_0 : D) = \sum_m \log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]}$$

$$Score_L(G_1 : D) = \sum_m \log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]|x[m]}$$

$$Score_L(G_1 : D) - Score_L(G_0 : D) = \sum_m \log \hat{\theta}_{y[m]|x[m]} - \log \hat{\theta}_{y[m]}$$

$$= \sum_{x,y} M[x,y] \log \hat{\theta}_{y|x} - \sum_y M[y] \log \hat{\theta}_y = M \sum_{x,y} \hat{P}(x,y) \log \hat{P}(y \mid x) - M \sum_y \hat{P}(y) \log \hat{P}(y)$$

$$= M \sum_{x,y} \hat{P}(x,y) \log \hat{P}(y \mid x) - M \sum_{x,y} \hat{P}(x,y) \log \hat{P}(y) = M \cdot \mathbf{I}_{\hat{P}}(X,Y)$$

By definition of mutual information between X and Y

Likelihood Score for BN: General Decomposition (I)

$$Score_L(G:D) = \ell(\theta_G, G; D) = \log p(D \mid \theta_G, G)$$

General BN graph

$$= \log \prod_n \left(\prod_i p(x_{n,i} \mid \mathbf{x}_{n,\pi_i(G)}, \theta_{i|\pi_i(G)}) \right) = \sum_i \left(\sum_n \log p(x_{n,i} \mid \mathbf{x}_{n,\pi_i(G)}, \theta_{i|\pi_i(G)}) \right)$$

Parents of node i in BN

$$= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \frac{count(x_i, \mathbf{x}_{\pi_i(G)})}{M} \log p(x_i \mid \mathbf{x}_{\pi_i(G)}, \theta_{i|\pi_i(G)}) \right)$$

$$= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \hat{p}(x_i, \mathbf{x}_{\pi_i(G)}) \log p(x_i \mid \mathbf{x}_{\pi_i(G)}, \theta_{i|\pi_i(G)}) \right)$$

Sum over BN nodes

Sum over variable configurations

$$= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \hat{p}(x_i, \mathbf{x}_{\pi_i(G)}) \log \frac{\hat{p}(x_i, \mathbf{x}_{\pi_i(G)}, \theta_{i|\pi_i(G)})}{\hat{p}(\mathbf{x}_{\pi_i(G)})} \frac{\hat{p}(x_i)}{\hat{p}(x_i)} \right)$$

Likelihood Score for BN: General Decomposition (II)

$$M\sum_i\left(\sum_{x_i,\mathbf{x}_{\pi_i(G)}}\hat{p}(x_i,\mathbf{x}_{\pi_i(G)})\log\frac{\hat{p}(x_i,\mathbf{x}_{\pi_i(G)},\theta_{i|\pi_i(G)})}{\hat{p}(\mathbf{x}_{\pi_i(G)})}\frac{\hat{p}(x_i)}{\hat{p}(x_i)}\right)$$

$$=M\sum_i\left(\sum_{x_i,\mathbf{x}_{\pi_i(G)}}\hat{p}(x_i,\mathbf{x}_{\pi_i(G)})\log\frac{\hat{p}(x_i,\mathbf{x}_{\pi_i(G)},\theta_{i|\pi_i(G)})}{\hat{p}(\mathbf{x}_{\pi_i(G)})\hat{p}(x_i)}\right)+M\sum_i\left(\sum_{x_i}\hat{p}(x_i)\log\hat{p}(x_i)\right)$$

$$=M\sum_i\hat{I}(x_i,\mathbf{x}_{\pi_i(G)})-M\sum_i\hat{H}(x_i)$$

Sum over BN nodes

Information-theoretic interpretation:

- High mutual information implies stronger dependency. Stronger dependency implies stronger preference for the model where X_i and it parents depend on each other.

Likelihood Score for BN: General Decomposition (III)

- Likelihood score decomposes as:

$$Score_L(G:D) = M\sum_{i=1}^{n}\mathbf{I}_{\hat{P}}(X_i, Pa_{X_i}^G) - M\sum_{i=1}^{n}\mathbf{H}_{\hat{P}}(X_i)$$

  - Second term (entropy term) does not depend on network structure and thus is irrelevant for selecting between two structures
  - Score is larger as mutual information (i.e., strength of dependence) between connected variables increases

- With further manipulation (via chain rule):

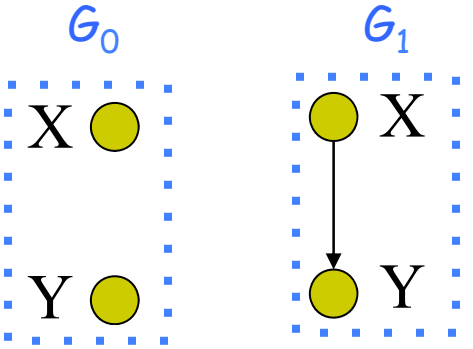
$$Score_L(G:D) = \mathbf{H}_{\hat{P}}(X_1,...,X_n) - \sum_{i=1}^{n}\mathbf{I}_{\hat{P}}(X_i, \{X_1,...X_{i-1}\} - Pa_{X_i}^G \mid Pa_{X_i}^G)$$

independent of network structure

measuring the extent of the independence of X from its predecessors given its parents

Likelihood Score for BN: Over Fitting

$$Score_L(G_1 : D) - Score_L(G_0 : D) = M \cdot \mathbf{I}_{\hat{P}}(X, Y) \geq 0$$



$\Rightarrow$ $Score_L(G_1:D) \geq Score_L(G_0:D)$

- In general: Property of mutual information:

$$I(\boldsymbol{X}; \boldsymbol{Y} \cup \boldsymbol{Z}) \geq I(\boldsymbol{X}; \boldsymbol{Y})$$

- Adding edge always increases Likelihood score
- Maximum score attained by fully connected network which tends to overfitting data (i.e., fit the noise in the data)

- Potential remedies for overfitting:
  - Restricting the hypotheses space (e.g., restrict # of parents or # of parameters)
  - Minimum description length (MDL) criteria to penalize complexity
    - Prefer models that compactly describes the training data
  - Bayesian methods (typically done using MCMC, very slow)
    - Average over all possible parameter values via prior knowledge

Bayesian Score for BN

- Place a distribution over variables that have uncertainty (i.e., random).
- Uncertainty in our problem: (G, Θ_G)

$$P(G \mid D) = \frac{P(D \mid G)P(G)}{P(D)}$$

- Clearly, P(D) (marginal probability of data) does not depend on the structure.

Bayesian Score: $Score_B(G:D) = \log P(D \mid G) + \log P(G)$

Marginal Likelihood

Prior over structures

Likelihood score

$$P(D \mid G) = \int_{\theta_G} P(D \mid G, \theta_G) P(\theta_G \mid G) d\theta_G$$

- Similar to likelihood score, but with the key difference that ML finds maximum of likelihood but Bayesian computes average of the terms over parameter space
- We do not have time to explore details of Bayesian score in our course.

BIC approximation of Bayesian Score

- Bayesian has difficult integrals
- For Dirichlet prior, can use simple Bayes information criterion (BIC) approximation:
  - In the limit, we can forget prior!

Theorem: for Dirichlet prior, and a BN with Dim(G) independent parameters, as M$\rightarrow\infty$:

$$\log P(D \mid \mathcal{G}) = \log P(D \mid \mathcal{G}, \theta_{\mathcal{G}}) - \frac{\log M}{2}\text{Dim}(\mathcal{G}) + O(1)$$

$$\text{Score}_{\text{BIC}}(\mathcal{G} : D) = M \sum_i \hat{I}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) - M \sum_i \hat{H}(X_i) - \frac{\log M}{2} \sum_i \text{Dim}(P(X_i \mid \mathbf{Pa}_{x_i,\mathcal{G}}))$$

Likelihood score:
Likes fully connected graph

Regularization penalty:
Likes simple graph

Properties of Network Scoring Methods for BN

- Decomposability
  - Likelihood, Bayesian, etc,, have the form

$$Score(G:D) = \sum_i Score(X_i \mid Pa_i^G : D)$$

- All I-equivalent graphs are score-equivalent:

G I-equivalent to *G'* $\Longrightarrow$ *Score(G) = Score(G')*

Learning Tree Structures: Chow-Liu algorithm (I)

- Likelihood score: $\ell(\theta_G, G; D) = M\sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)}) - M\sum_i \hat{H}(x_i)$

- Find a tree T to maximize the Objective function:

$$T^* = argmax_T \, M \sum_{(i,j)\in T} \hat{I}(x_i, x_j)$$

- Define a tree with nodes $x_1,\ldots,x_n$
- Edge (i,j) gets score/weight $\hat{I}(X_i, X_j)$

- For each pair of variables x_i and x_j

  - Compute empirical joint distribution:

$$\hat{p}(X_i, X_j) = \frac{count(x_i, x_j)}{M}$$

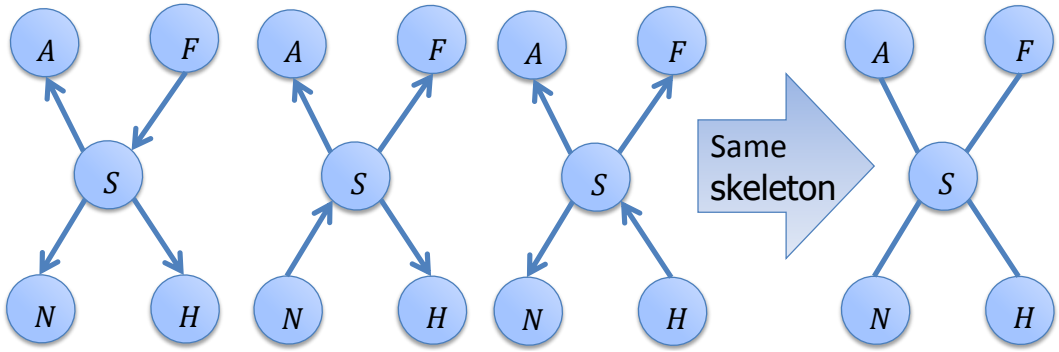
  - Compute weight (score) of each edge:

$$\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \hat{p}(x_i, x_j) \log \frac{\hat{p}(x_i, x_j)}{\hat{p}(x_i)\hat{p}(x_j)}$$

Learning Tree Structures: Chow-Liu algorithm (II)

- Structure learning problem: Find a tree structure with maximum sum of weights.
- Solve an undirected weighted spanning tree (forest) problem and determine directions of edges afterwards.
  - This can be done using standard algorithms in low-order polynomial time by building a tree in a greedy fashion (e.g. Kruskal's maximum weight spanning tree algorithm)
- Direction in BN: pick any node as root, do breadth-first-search to define directions (within I-equivalence graph):

These graphs have the same total likelihood score because of symmetric property of mutual information

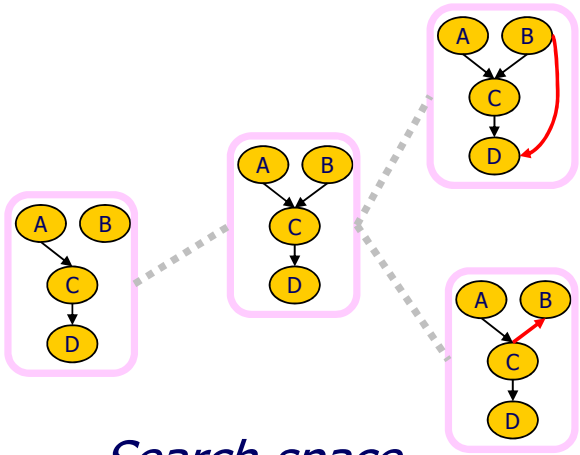


Structure Learning for general graphs: Don't Push your Luck!

- Allowing two parents (or more), greedy algorithm is no longer guaranteed to find the optimal network.

- Theorem:
  Finding maximal scoring network structure, with at most k parents for each variable, is NP-hard for k>1.

- In fact, no efficient algorithm exists
  - Heuristic Search
    - Greedy hill-climbing
    - Best first search
    - Simulated Annealing



Search space

Next Topic

- So far, we studied structural learning for Bayesian networks in fully observed setting. Next we study:

ML Structural Learning via Neighborhood Selection for Fully observed Markov Random Fields (MRF)

MRF is desirable in many applications, even when the original setting is BN:

MN: Gaussian Graphical Models

- We will study the MRF structure learning only under Multivariate Gaussian setting:

Data: $(x_1^{(1)},\ldots,x_n^{(1)})$ $(x_1^{(2)},\ldots,x_n^{(2)})$ … $(x_1^{(M)},\ldots,x_n^{(M)})$

- A Gaussian distribution can be represented by a fully connected graph with pairwise edge potentials over continuous variable nodes

- The overall exponential form is:

$$P(X_1,\ldots,X_n)=\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp\left\{-\tfrac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

$$\propto \exp\left(-\sum_{ij\in E}(X_i-\mu_i)\Sigma_{ij}^{-1}(X_j-\mu_j)\right)$$

- Using: $\mu=0 \quad Q=\Sigma^{-1}$ ⟹

Precision (information) Matrix: $Q=[q_{ij}]$

$$P(X_1,\ldots,X_n)\propto \exp\left\{-\tfrac{1}{2}\sum_i q_{ii}(x_i)^2-\sum_{i<j}q_{ij}x_ix_j\right\}$$

- MRF with edge and node potentials, (known as Gaussian graphical models-GGM)

Sparse Precision vs. Sparse Covariance Matrices

Assume Multivariate Gaussian:

$$\Sigma^{-1} =$$

1	6	0	0	0
6	2	7	0	0
0	7	3	8	0
0	0	8	4	9
0	0	0	9	5

$$\Sigma =$$

0.10	0.15	-0.13	-0.08	0.15
0.15	-0.03	0.02	0.01	-0.03
-0.13	0.02	0.10	0.07	-0.12
-0.08	0.01	0.07	-0.04	0.07
0.15	-0.03	-0.12	0.07	0.08

$\Sigma^{-1}_{15} = 0$ No edge potential, no link

$$\exp\left\{-\tfrac{1}{2}\sum_i q_{ii}(x_i)^2 - \sum_{i<j} q_{ij}x_i x_j\right\}$$

$X_1 \perp X_5 \,|TheRest$

~~$X_1 \perp X_5 \Leftrightarrow \Sigma_{15} = 0$~~

$\Sigma_{i,j} = 0 \quad \Rightarrow \quad X_i \perp X_j$

Independency under rest of nodes: Precision matrix naturally useful for sparsification of MRF

Useful for studying correlation graph (pairwise Independency not Conditioned on anything)

Recap from Lasso

- Given vector y and matrix A, find sparse vector x: $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$.

$$\Longrightarrow \qquad \min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1, \qquad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}.$$

$\|\mathrm{x}\|_1$ denotes the sum of absolute values of the elements of vector x.

This problem can be solved by

$$\text{Lasso:} \quad \hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2M} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_1, \ \lambda > 0.$$

Structure learning for Gaussian GM

- Structure: zero patterns in the inverse covariance (precision) matrix
- Key idea:
  - Maximize the log-likelihood of the data (fit data)
  - Place sparsity in the inverse covariance matrix (regularization) to penalize complexity of graph

- Log-likelihood of data:

$$l(D,\Sigma) \propto \log \prod_i |\Sigma^{-1}|^{\frac{1}{2}} \exp\left(-\frac{X_i^\top \Sigma^{-1} X_i}{2}\right)$$

$$\propto \sum_i \log|\Sigma^{-1}| - X_i^\top \Sigma^{-1} X_i$$

$$\propto \sum_i \log|\Sigma^{-1}| - tr(\Sigma^{-1} X_i X_i^\top)$$

$$\propto n \log|\Sigma^{-1}| - tr(\Sigma^{-1} S)$$

See pages 99-100 from K. Murphy

Graphical Lasso

- Maximize Gaussian log-likelihood with l_1 regularization on the inverse covariance matrix

$max_\Theta \, n \log|\Theta| - tr\,\Theta\, S \; - \lambda\|\Theta\|_1$

$\Theta = \Sigma^{-1}$ is positive semidefinite (Additional constraint)

$\|\Theta\|_1$ denotes the sum of absolute values of the elements of the matrix

- This is a convex optimization problem and can be solved by various optimization algorithms, e.g.,
  - Coordinate descent: Graphical lasso (http://www-stat.stanford.edu/~tibs/ftp/graph.pdf)
  - Interior point methods (Yuan & Lin 2007)

See also K.Murphy page 940 for more refs on Algorithms to solve above constraint Lasso optimization.