CS839: Probabilistic Graphical Models

Lecture 9: Structure Learning

Theo Rekatsinas
Learning so far

\[ \ell(\theta; D) = \log p(D | \theta) = \log \prod_x \left( \prod_{i} \sum_{\pi} \log p(x_{n,i} | \pi, \theta_i) \right) = \sum \left( \sum_{n} \log p(x_{n,i} | x_{n,\pi_i}, \theta_i) \right) \]

- We assumed that the structure of G is specified and fixed.
- Given access to a set of samples how do we learn what independencies hold between the corresponding random variables?
Learning in Graphical Models

• Goal: Given a set of independent samples (assignments to random variables), find the best network

(B,E,A,C,R) = (T,F,F,T,F)  
(B,E,A,C,R) = (T,F,T,T,F)  
...  
(B,E,A,C,R) = (F,T,T,T,F)

Structure learning

Parameter learning

| E | B | P(A | E,B) |
|---|---|-----------|
| e | b | 0.9 0.1   |
| e | b | 0.2 0.8   |
| e | b | 0.9 0.1   |
| e | b | 0.01 0.99 |
Why aim for accurate structure?

- Cannot be compensated for by fitting parameters
- Wrong assumptions about domain structure

True Structure

- Increases the number of parameters to be estimated
- Wrong assumptions about domain structure
Different approaches

• There are many heuristics but they come with no “guarantees”

• In the context of structure learning we want a “guarantee” that a certain algorithm will maximize a certain objective.

• We will see two classes of algorithms for guaranteed structure learning (but they apply to certain families of graphs)
  • For tree BNs the Chow-Liu algorithm
  • For pairwise MRFs: Covariance selection, neighborhood-selection
Score-based Learning

• Define a scoring function that evaluates how well a structure matches the data

• Search for a structure that maximizes the score
Information Theoretic Interpretation

\[
\ell(\theta_G, G; D) = \log p(D \mid \theta_G, G)
\]

\[
= \log \prod_i \left( \prod_n p(x_{n,i} \mid x_{n,\pi_i(G)}, \theta_{\mid \pi_i(G)}) \right)
\]

\[
= \sum_i \left( \sum_n \log p(x_{n,i} \mid x_{n,\pi_i(G)}, \theta_{\mid \pi_i(G)}) \right)
\]

\[
= M \sum_i \left( \sum_{x_{i,\pi_i(G)}} \frac{\text{count}(x_i, x_{\pi_i(G)})}{M} \log p(x_i \mid x_{\pi_i(G)}, \theta_{\mid \pi_i(G)}) \right)
\]

\[
= M \sum_i \left( \sum_{x_{i,\pi_i(G)}} \hat{p}(x_i, x_{\pi_i(G)}) \log p(x_i \mid x_{\pi_i(G)}, \theta_{\mid \pi_i(G)}) \right)
\]

• Sum over data points and sum over count of variable states given the candidate graph structure
Information Theoretic Interpretation

\[ \ell(\theta_G, G; D) = \log \hat{p}(D | \theta_G, G) \]

\[ = M \sum_i \left( \sum_{x_i, x_{\pi_i(G)}} \hat{p}(x_i, x_{\pi_i(G)}) \log \frac{\hat{p}(x_i, x_{\pi_i(G)}, \theta_{\pi_i(G)})}{\hat{p}(x_{\pi_i(G)})} \hat{p}(x_i) \right) \]

\[ = M \sum_i \left( \sum_{x_i, x_{\pi_i(G)}} \hat{p}(x_i, x_{\pi_i(G)}) \log \frac{\hat{p}(x_i, x_{\pi_i(G)}, \theta_{\pi_i(G)})}{\hat{p}(x_{\pi_i(G)})} \hat{p}(x_i) \right) - M \sum_i \left( \sum_{x_i} \hat{p}(x_i) \log \hat{p}(x_i) \right) \]

\[ = M \sum_i \hat{I}(x_i, x_{\pi_i(G)}) - M \sum_i \hat{H}(x_i) \]

- We get a decomposable score and a function of the graph structure
Information Theoretic Interpretation

\[
\ell(\theta_G, G; D) = \log \hat{p}(D \mid \theta_G, G)
\]

\[
= M \sum_i \left( \sum_{x_i, x_{\pi_i}(G)} \hat{p}(x_i, x_{\pi_i}(G)) \log \frac{\hat{p}(x_i, x_{\pi_i}(G), \theta_{\pi_i}(G))}{\hat{p}(x_{\pi_i}(G))} \hat{p}(x_i) \right)
\]

\[
= M \sum_i \left( \sum_{x_i, x_{\pi_i}(G)} \hat{p}(x_i, x_{\pi_i}(G)) \log \frac{\hat{p}(x_i, x_{\pi_i}(G), \theta_{\pi_i}(G))}{\hat{p}(x_{\pi_i}(G))} \hat{p}(x_i) \right) - M \sum_i \left( \sum_{x_i} \hat{p}(x_i) \log \hat{p}(x_i) \right)
\]

\[
= M \sum_i \hat{I}(x_i, x_{\pi_i}(G)) - M \sum_i \hat{H}(x_i)
\]

• Larger dependence of Xi on the parents defined by G => higher score
• Problems?
Information Theoretic Interpretation

\[ \ell(\theta_G, G; D) = \log \hat{p}(D | \theta_G, G) \]

\[ = M \sum_i \left( \sum_{x_i, x_{\pi_i(G)}} \hat{p}(x_i, x_{\pi_i(G)}) \log \frac{\hat{p}(x_i | x_{\pi_i(G)}, \theta_{\pi_i(G)})}{\hat{p}(x_{\pi_i(G)})} \hat{p}(x_i) \right) \]

\[ = M \sum_i \left( \sum_{x_i, x_{\pi_i(G)}} \hat{p}(x_i, x_{\pi_i(G)}) \log \frac{\hat{p}(x_i, x_{\pi_i(G)}, \theta_{\pi_i(G)})}{\hat{p}(x_{\pi_i(G)})} \hat{p}(x_i) \right) \]

\[ = M \sum_i \left( \sum_{x_i, x_{\pi_i(G)}} \hat{p}(x_i, x_{\pi_i(G)}) \log \frac{\hat{p}(x_i, x_{\pi_i(G)}, \theta_{\pi_i(G)})}{\hat{p}(x_{\pi_i(G)})} \hat{p}(x_i) \right) - M \sum_i \left( \sum_{x_i} \hat{p}(x_i) \log \hat{p}(x_i) \right) \]

\[ = M \sum_i \hat{I}(x_i, x_{\pi_i(G)}) - M \sum_i \hat{H}(x_i) \]

- Larger dependence of Xi on the parents defined by G => higher score
- Adding edges always helps! \( I(X;Y) \leq I(X; \{Y,Z\} \)
- Max score when we have a fully connected graph. We are overfitting to our data!
Bayesian Score

• We will follow a Bayesian approach to attack overfitting
  • We need to place a distribution over our “uncertain” elements
  • These are both the graph G and the parameters $\theta$ of the network

\[
P(G \mid D) = \frac{P(D \mid G)P(G)}{P(D)}
\]

• Bayesian score for G

\[
Score_B(G : D) = \log P(D \mid G) + \log P(G)
\]
Likelihood of data given G

• Bayesian score for G

\[ \text{Score}_B(G : D) = \log P(D | G) + \log P(G) \]

• Marginal Likelihood

\[ P(D|G) = \int_{\theta_G} P(D|G, \theta_G)P(\theta_G|G)d\theta_G \]

• Similar to max-likelihood score but with the key difference that max-likelihood finds the maximum of likelihood while here we compute an average of terms over the parameter space
Marginal Likelihood: Bayes Nets

• We assume a Dirichlet prior over our parameters
• The marginal likelihood has the form

\[
P(D|G) = \prod_i \prod_{pa_i^G} \frac{\Gamma(\alpha_{pa_i^G})}{\Gamma(\alpha_{pa_i^G} + M[pa_i^G])} \prod_{x_i} \frac{\Gamma(\alpha_{x_i,pa_i^G} + M[x_i,pa_i^G])}{\Gamma(\alpha_{x_i,pa_i^G})}
\]

• M are the counts from the data
• a are the hyperparameters

• Key property: Decomposability of the score
Bayesian Score: Asymptotic Behavior

• For $M \to \infty$, a network $G$ with Dirichlet priors satisfies
  \[ \log P(D \mid G) = l(\hat{\theta}_G : D) - \frac{\log M}{2} \text{Dim}(G) + O(1) \]
  \[ \text{Dim}(G): \text{number of independent parameters in } G \]

• Bayesian score is consistent: observed data eventually overrides prior
• An approximation of the Bayesian score is the BIC score
  \[ \text{Score}_{\text{BIC}}(G : D) = l(\hat{\theta}_G : D) - \frac{\log M}{2} \text{Dim}(G) \]

• We have a tradeoff between fit to data and complexity
  • MI grows linearly with $M$ while complexity grows logarithmically with $M$
Structure learning as an optimization problem

• Input:
  • Training data D
  • Scoring function (with or without priors)
  • Set of possible structures (search space)

• Output:
  • A network that maximizes the score

• Key property:
  • Decomposability: the score decomposes as a sum over terms

\[ \text{Score}(G : D) = \sum_{i} \text{Score}(X_i \mid Pa_i^G : D) \]
Tree BNs

• We will assume at most one parent per variable

• Why trees?
  • Sparse parameterization: We avoid overfitting
  • We can solve the optimization problem with a greedy algorithm (with guarantees)
Chow-Liu tree learning algorithm

• Objective function:

\[
\mathcal{L}(\theta_G, G; D) = \log \hat{p}(D | \theta_G, G) = M \sum_i \hat{I}(x_i, x_{\pi_i(G)}) - M \sum_i \hat{H}(x_i) \Rightarrow C(G) = M \sum_i \hat{I}(x_i, x_{\pi_i(G)})
\]

• Chow-Liu:
  • For each pair of variables \(x_i\) and \(x_j\)
    • Compute empirical distribution
  • Compute mutual information
  • Define a graph with nodes \(x_1, \ldots, x_n\)
    • Edge \((i,j)\) gets weight \(\hat{I}(X_i, X_j)\)

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

\[
\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \hat{p}(x_i, x_j) \log \frac{\hat{p}(x_i, x_j)}{\hat{p}(x_i) \hat{p}(x_j)}
\]
Chow-Liu tree learning algorithm

• Objective function:

\[ \ell(\theta, G; D) = \log \hat{p}(D | \theta, G) = M \sum_i \hat{I}(x_i, x_{\pi_i(G)}) - M \sum_i \hat{H}(x_i) \]

\[ C(G) = M \sum_i \hat{I}(x_i, x_{\pi_i(G)}) \]

• Chow-Liu:
  • Optimal tree BN
    • Compute maximum weight spanning tree
    • Direction in BN: pick any node as root, do BFS to define directions

\[ C(G) = I(A, B) + I(A, C) + I(C, D) + I(C, E) \]
Structure learning for general BNs

• The problem of learning a BN structure with at most $d$ parents is $\text{NP}$-hard for any (fixed) $d \geq 2$

• Most structure learning approaches use heuristics
  • Exploit score decomposition
  • Greedy search through space of node-orders
  • Local search of graph structures
Learning undirected Graphical Models
Gaussian Graphical Models

• Multivariate Gaussian density:

\[
p(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\}
\]

• Let \( \mu = 0 \) \( Q = \Sigma^{-1} \)

\[
p(x_1, x_2, \ldots, x_p \mid \mu = 0, Q) = \frac{|Q|^{1/2}}{(2\pi)^{n/2}} \exp\left\{ -\frac{1}{2} \sum_i q_{ii} (x_i)^2 - \sum_{i<j} q_{ij} x_i x_j \right\}
\]

• We can view this as a continuous MRF with potentials defined on every node and edge

\[
P(x_d \mid \Theta) = \exp\left( \sum_{i \in V} \theta_{ii}^t x_{d,i} + \sum_{(i,j) \in E} \theta_{ij} x_{d,i} x_{d,j} - A(\Theta) \right)
\]
The covariance and the precision matrices

• Covariance matrix \( \Sigma \)

\[
\Sigma_{i,j} = 0 \implies X_i \perp X_j \quad \text{or} \quad p(X_i, X_j) = p(X_i)p(X_j)
\]

• What is the graphical model interpretation?

• Precision matrix \( Q = \Sigma^{-1} \)

\[
Q_{i,j} = 0 \implies X_i \perp X_j|X_{-ij} \quad \text{or} \quad p(X_i, X_j|X_{-ij}) = p(X_i|X_{-ij})p(X_j|X_{-ij})
\]

• What is the graphical model interpretation?
The covariance and the precision matrices

- **Covariance matrix** $\Sigma$

  $$\Sigma_{i,j} = 0 \quad \Rightarrow \quad X_i \perp X_j \quad \text{or} \quad p(X_i, X_j) = p(X_i)p(X_j)$$

  - What is the GM interpretation? Marginally independent/Correlation graph

- **Precision matrix** $Q = \Sigma^{-1}$

  $$Q_{i,j} = 0 \quad \Rightarrow \quad X_i \perp X_j | X_{-ij} \quad \text{or} \quad p(X_i, X_j | X_{-ij}) = p(X_i | X_{-ij})p(X_j | X_{-ij})$$

  - What is the GM interpretation? Conditionally independent/Markov graph

- Why important?
Precision vs. Covariance

\[
\Sigma^{-1} = \begin{pmatrix}
1 & 6 & 0 & 0 & 0 \\
6 & 2 & 7 & 0 & 0 \\
0 & 7 & 3 & 8 & 0 \\
0 & 0 & 8 & 4 & 9 \\
0 & 0 & 0 & 9 & 5 \\
\end{pmatrix}
\quad \Sigma = \begin{pmatrix}
0.10 & 0.15 & -0.13 & -0.08 & 0.15 \\
0.15 & -0.03 & 0.02 & 0.01 & -0.03 \\
-0.13 & 0.02 & 0.10 & 0.07 & -0.12 \\
-0.08 & 0.01 & 0.07 & -0.04 & 0.07 \\
0.15 & -0.03 & -0.12 & 0.07 & 0.08 \\
\end{pmatrix}
\]

\[
\Sigma_{15}^{-1} = 0 \iff X_1 \perp X_5 | X_{nbs(1)} \text{ or } nbs(5)
\]

\[
\not\iff
\]

\[
X_1 \perp X_5 \iff \Sigma_{15} = 0
\]
Example

$$Q = \begin{pmatrix}
* & * & * & * & * & 0 \\
* & * & * & * & 0 & 0 \\
* & * & 0 & * & 0 & 0 \\
* & * & 0 & 0 & * & 0 \\
0 & 0 & 0 & 0 & 0 & * \\
\end{pmatrix}$$

- How to estimate this MRF?
- What if \( p \) (number of dimensions) \( >> \) \( n \) (number of data points)?
Example

\[ Q = \begin{pmatrix} * & * & * & * & * & 0 \\ * & * & * & * & * & 0 \\ * & * & * & 0 & 0 & 0 \\ * & * & 0 & * & 0 & 0 \\ * & * & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * \end{pmatrix} \]

- How to estimate this MRF?
  - Say we can estimate a sample covariance. Can we get Q?
- What if \( p \) (number of dimensions) >> \( n \) (number of data points)?
Example

\[
Q = \begin{pmatrix}
* & * & * & * & * & 0 \\
* & * & * & * & * & 0 \\
* & * & * & 0 & 0 & 0 \\
* & * & 0 & * & 0 & 0 \\
* & * & 0 & 0 & * & 0 \\
0 & 0 & 0 & 0 & 0 & * \\
\end{pmatrix}
\]

- How to estimate this MRF?
  - Say we can estimate a sample covariance. Can we get $Q$?
- What if $p$ (number of dimensions) $\gg n$ (number of data points)?
  - Let’s look into sparsity: $s = o(n)$
Recall Lasso

\[ \hat{\theta}_i = \arg \min_{\theta_i} l(\theta_i) + \lambda_1 \| \theta_i \|_1 \]

where \( l(\theta_i) = \log P(y_i|x_i, \theta_i) \).
Recall Lasso

\[ \hat{\theta}_i = \arg \min_{\theta_i} l(\theta_i) + \lambda_1 \| \theta_i \|_1 \]

where \( l(\theta_i) = \log P(y_i | x_i, \theta_i) \).

\[ P(x[1], \ldots, x[p]) \sim N(0, Q) \]

\[ P(x[i] | x[-i]) \sim N(q * x[-i]/c, \sigma) \]

Given this we can estimate \( Q \) column-by-column using a slightly different estimator. We estimate \( p \) vectors.

How can we estimate \( N(q * x[-i]/c, \sigma) \)? Linear regression corresponds to MLE of this model.
Recall Lasso

\[ \hat{\theta}_i = \arg\min_{\theta_i} l(\theta_i) + \lambda_1 || \theta_i ||_1 \]

where \( l(\theta_i) = \log P(y_i | x_i, \theta_i). \)

Given this we can estimate Q column-by-column using a slightly different estimator. We estimate p vectors.

How can we estimate \( N( q^*x[-i]/c, \sigma) \)? Linear regression corresponds to MLE of this model.
Graph Regression

Pick one node
Perform regression and keep non-zero coefficients: That is our neighborhood of our node. We do not care about the magnitude of $\theta$ here. Why?
Graph Regression
Graph Regression

It can be shown that:
Given iid samples, and under several conditions the recovered structure is “sparsistent”
even when \( p >> n \)
Learning Ising Model

• Assuming the nodes are discrete, and edges are weighted then for a sample we have

\[ P(x_d | \Theta) = \exp \left( \sum_{i \in V} \theta_{ii} x_{d,i} + \sum_{(i,j) \in E} \theta_{ij} x_{d,i} x_{d,j} - A(\Theta) \right) \]
Learning Ising Model

• Assuming the nodes are discrete, and edges are weighted then for a sample we have

\[
P(x_d|\Theta) = \exp \left( \sum_{i \in V} \theta_{ii}^t x_{d,i} + \sum_{(i,j) \in E} \theta_{ij} x_{d,i} x_{d,j} - A(\Theta) \right)
\]

• It can be shown following the same logic that we can use L1 regularized logistic regression to obtain a sparse estimate of the neighborhood of each variable in the discrete case.
Learning Ising Model

• Assuming the nodes are discrete, and edges are weighted then for a sample we have

\[ P(x_d|\Theta) = \exp \left( \sum_{i \in V} \theta_{ii} x_{d,i} + \sum_{(i,j) \in E} \theta_{ij} x_{d,i} x_{d,j} - A(\Theta) \right) \]

• It can be shown following the same logic that we can use L1 regularized logistic regression to obtain a sparse estimate of the neighborhood of each variable in the discrete case.
  • This is an approximation to the likelihood of the original problem.
  • We can be wrong when estimating the value of \( \theta \) but not the structure.

• What if I care about the value of \( \theta \)?
Learning Ising Model

• Assuming the nodes are discrete, and edges are weighted then for a sample we have

\[ P(x_d|\Theta) = \exp \left( \sum_{i \in V} \theta_{ii}^t x_{d,i} + \sum_{(i,j) \in E} \theta_{ij} x_{d,i} x_{d,j} - A(\Theta) \right) \]

• It can be shown following the same logic that we can use L1 regularized logistic regression to obtain a sparse estimate of the neighborhood of each variable in the discrete case.
  • This is an approximation to the likelihood of the original problem.
  • We can be wrong when estimating the value of \( \theta \) but not the structure.

• What if I care about the value of \( \theta \)? We can now estimate parameters with known graph structure.
Consistency

• **Theorem**: for the graphical regression algorithm, under certain conditions

\[ \mathbb{P} \left[ \hat{G}(\lambda_n) \neq G \right] = \mathcal{O} \left( \exp \left( -C n^c \right) \right) \rightarrow 0 \]

• Here the regularizer is not actually used to introduce an “artificial” sparsity bias, but it is a devise to ensure consistency under finite data and high dimensionality.
Summary

• There are many heuristics but they come with no “guarantees”

• We saw two classes of algorithms for guaranteed structure learning (but they apply to certain families of graphs)
  • For tree BNs the Chow-Liu algorithm
  • For pairwise MRFs: Covariance selection, neighborhood-selection