MAC-5789 Laboratório de IA

Bayesian Networks

Denis Deratani Mauá
denis.maua@usp.br
Room C-8

June 10, 2015

Outline

Last class:
- Informal Introduction
- Probabilistic Reasoning
- Representing Independences
- Bayesian Networks
- Assignment: First Part

Today:
- Structural Learning
- Assignment: Second Part

Structural Learning

Bayesian Network

- A DAG G over a set of variables $X_1, \dots, X_n$
- Markov assumption: $\mathbb{I}(X_i, \mathsf{ND}(X_i) | \mathsf{Pa}(X_i))$
- Probability constraints: $\mathbb{P}(X_i = k | \mathsf{Pa}(X_i) = j) = \theta_{ijk}$

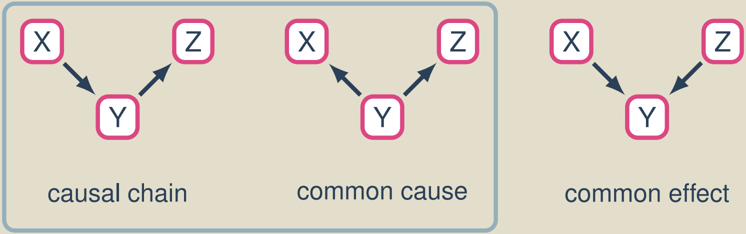
Joint Distribution

There is a unique probability function consistent with a BN:

$$\mathbb{P}(X_1, \dots, X_n) = \prod_{i=1}^{n} \mathbb{P}(X_i | \mathsf{Pa}(X_i)) = \prod_{i=1}^{n} \theta_{ijk}$$

Markov Equivalence

If two DAGs encode the same independences, then they represent the same joint probability distribution; Markov equivalent DAGs cannot be distinguished by data alone



Bayesian Network Structure Learning

- Constraint-based approaches
- Score-based approaches

Learning as optimization

Given dataset D, select G that maximizes score function $s(G, D)$

- Score $s(G, D)$ is usually a mix of data fitness F and model complexity P:

$$s(G, D) = F(G) + \psi(N) \cdot P(G)$$

  with $\psi(N) \geq 0$ is a function of data size $|D| = N$

- We usually ommit depenced on D: $s(G)$

Data Log-Likelihood

Probability that data has been generated by given DAG with MLE parameter estimates:

$$\mathsf{LL}(G) = \max_{\Theta} \log \mathbb{P}(D|G, \Theta) = \max_{\Theta} \log \prod_{i=1}^{n} \theta_{ijk}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{|\Omega_i|} \sum_{j=1}^{|\Omega_p|} N_{ijk} \log \frac{N_{ijk}}{N_{ij}} \, ,$$

where

$$N_{ijk} = \text{nr. of observations of } X_i = x_i^{(k)} \text{ and } \mathsf{Pa}(X_i) = \pi_i^{(j)}$$

and $N_{ij} = \sum_k N_{ijk}$

Structure Scoring

Data Log-Likelihood

$$\mathsf{LL}(G) = \sum_{i=1}^{n} \sum_{k} \sum_{j} N_{ijk} \log \frac{N_{ijk}}{N_{ij}}$$

Theorem

If G is obtained from G' by the inclusion of an arc then

$$LL(G) \geq LL(G')$$

- Every complete DAG maximizes LL (poor DAG scorer)
- We can limit the number of parents; then we learn classes of maximal DAGs (an arc cannot be inserted without violating in-degree upperbound): these are locally dense

- Variables $X_1, \ldots, X_n$ taking values in $\Omega_1, \ldots, \Omega_n$

Penalized Data Log-Likelihood

Penalize models by number of parameters:

$$s(G) = \mathsf{LL}(G) - \psi(N) \cdot \mathsf{size}(G)$$

where

$$\mathsf{size}(G) = \sum_{i=1}^{n} (|\Omega_i| - 1) \prod_{X_j \in \mathsf{Pa}(X_i)} |\Omega_j|$$

- log-likelihood decreases linearly with dataset size

- Variables $X_1, \ldots, X_n$ taking values in $\Omega_1, \ldots, \Omega_n$

Bayesian Information Criteria

Penalize models by number of parameters:

$$\mathsf{BIC}(G) = \mathsf{LL}(G) - \frac{\log(N)}{2} \cdot \mathsf{size}(G)$$

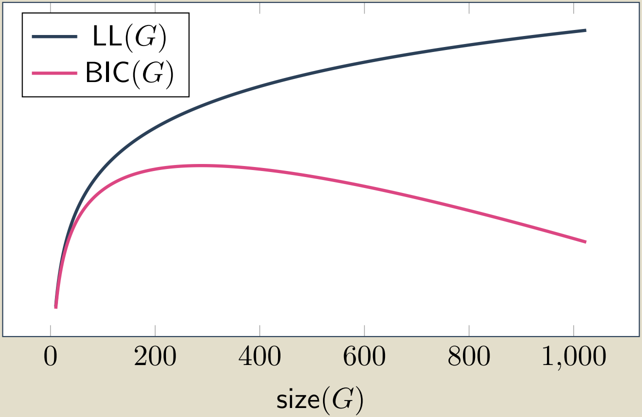
where

$$\mathsf{size}(G) = \sum_{i=1}^{n} (|\Omega_i| - 1) \prod_{X_j \in \mathsf{Pa}(X_i)} |\Omega_j|$$

- relevance of model complexity penalty grows logarithmically with dataset size

Bayesian Information Criteria

$$\mathsf{BIC}(G) = \mathsf{LL}(G) - \frac{\log(N)}{2} \cdot \mathsf{size}(G)$$



Conditional Entropy

$$H(X_i|\mathsf{Pa}(X_i)) = -\sum_k \sum_j \frac{N_{ijk}}{N} \log \frac{N_{ijk}}{N_{ij}}$$

Data Log-Likelihood

$$\begin{aligned}
\mathsf{LL}(G) &= \sum_{i=1}^{n} \sum_k \sum_j N_{ijk} \log \frac{N_{ijk}}{N_{ij}} \\
&= -N \sum_{i=1}^{n} \left[-\sum_k \sum_j \frac{N_{ijk}}{N} \log \frac{N_{ijk}}{N_{ij}} \right] \\
&= -N \sum_{i=1}^{n} H(X_i|\mathsf{Pa}(X_i))
\end{aligned}$$

- Variables $X_1, \ldots, X_n$ taking values in $\Omega_1, \ldots, \Omega_n$

Bayesian Information Criteria

$$\begin{aligned}
\mathsf{BIC}(G) &= \mathsf{LL}(G) - \frac{\log(N)}{2} \cdot \mathsf{size}(G) \\
&= \sum_{i=1}^{n} \left[-N \cdot H(X_i|\mathsf{Pa}(X_i)) - \frac{\log(N)}{2} \cdot \mathsf{size}(X_i, \mathsf{Pa}(X_i)) \right] \\
&= \sum_{i=1}^{n} s_i(X_i, \mathsf{Pa}(X_i))
\end{aligned}$$

where

$$\mathsf{size}(X_i, \mathsf{Pa}(X_i)) = (|\Omega_i| - 1) \prod_{X_j \in \mathsf{Pa}(X_i)} |\Omega_j|$$

Learning as optimization

Select G that maximizes decomposable score function

$$\sum_i s_i(X_i, \mathsf{Pa}(X_i))$$

- BIC:

$$s_i(X_i, P) = -N \cdot H(X_i|P) - \frac{\log(N)}{2} \cdot \mathsf{size}(X_i, P)$$

$$H(X_i|P) = -\sum_{i=1}^{|\Omega_i|} \sum_{j=1}^{|\Omega_P|} \frac{N_{ijk}}{N} \log \frac{N_{ijk}}{N_{ij}}$$

$$\mathsf{size}(X_i, P) = (|\Omega_i| - 1) \prod_{X_j \in P} |\Omega_j|$$

Order-Based Learning

Assume topological variable order is known: $X_1, \ldots, X_n$

$$\max_G \sum_{i=1}^{n} s_i(X_i, \mathsf{Pa}(X_i)) = \sum_{i=1}^{n} \underbrace{\max_{P \subseteq \{X_j < X_i\}} s_i(X_i, P)}_{2^{i-1} \text{ choices}}$$

- Fix maximum in-degree d (say $d = 4$):

$$\max_{P \subseteq \{X_j < X_i\} : |P| \leq d} s_i(X_i, P) \qquad 2^d \text{ choices}$$

- Perform greedy search:
  1. Start with set P
  2. Select $X_k \in \{X_j < X_i, X_j \notin P\}$ that maximizes $s_i(X_i, P \cup X_k)$
  3. If $s_i(X_i, P \cup X_k) > s_i(X_i, P)$ add X_k to P and go to step 2

There are too many orderings to consider: $n!$

Uniform sampling of orderings

For $k = 1, \ldots, M$

1. Sample order $X_1, \ldots, X_n$ uniformly at random
2. Optimize $s_i(X_i, \mathsf{Pa}(X_i))$ independently

Sampling orders (Fisher-Yates shuffle)

Start with arbitrary list $L = [X_1, \ldots, X_n]$ and repeat for $i = n - 1, \ldots, 1$:

1. Sample random integer j in $[0, i]$
2. Swap $L[i]$ and $L[j]$

Local Search in the Space of Orderings

Start with arbitrary ordering $L = \text{perm}([X_1, \ldots, X_n])$

1. For $i = 1, \ldots, n-1$ compute

$$\delta(i) = \left(\max_G s(G) \text{ s.t. } L_i\right) - \left(\max_G s(G) \text{ s.t. } L\right)$$

   where $L_i = [L[1], \ldots, L[i+1], L[i], \ldots, L[n]]$
2. Select i that maximizes $\delta(i)$
3. If $\delta(i) > 0$ set $L \leftarrow L_i$ and go to 1

$$\delta(i) = s_{L_i}(X_i, \mathsf{Pa}(X_i)) + s_{L_i}(X_{i+1}, \mathsf{Pa}(X_{i+1})) \\ - s_L(X_i, \mathsf{Pa}(X_i)) - s_L(X_{i+1}, \mathsf{Pa}(X_{i+1}))$$

Generalization ability:

Estimate performance on "unseen" data

Test Log-Likelihood

$$\mathsf{TLL}(G) = \sum_{i=1}^{n} \sum_{k=1}^{|\Omega_i|} \sum_{j=1}^{|\Omega_P|} N'_{ijk} \log \frac{N_{ijk}}{N_{ij}}$$

where N_{ijk} is the count on training data and N'_{ijk} is the count on test data

Assignment

Simulated Data

1. Read Teissyer and Koller 2005's article
2. Generate training and test data sets using your handcrafted network (about 10k samples in each set)
3. Implement a Order-Based Structure Learning Procedure using BIC score
   - Random sampling approach
   - Greedy Search (use multiple re-starts)
4. Compare performance of top 100 networks learned on training data on test data (compute statistics, draw histograms, box plots, etc)

The outcome of this part should indicate that your implementation is correct

Real Data

1. Learn models using random sampling and greedy approaches on the original training data set
2. Compare top 100 learned models on test data along with your handcrafted network
3. Write (decent) report (overview, problem description, methodology, experimental results, discussion, conclusion)
4. Upload pdf before Jun, 24

You should answer questions such as which approach is better (under which criteria), are there enough data (compare with your results in first part), what was too burdensome, what can be done to possibly improve results, etc

Suggested Reading

- Marc Teissyer & Daphne Koller, 2005. Order-Based Structure Learning. UAI 2005.
- Eugene Charniak, 1991. Bayesian networks without tears. AI Magazine, 12:4, pp. 50-63.