Param. Learning (MLE)

Structure Learning $f_0 \rightarrow BN$

The Good

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University
October 1st, 2008

Readings:
K&F: 16.1, 16.2, 17.1, 17.2, 17.3.1, 17.4.1

Learning the CPTs

For each discrete variable $X_i$

$P(x_i) = \frac{\text{Count}(X_i = x_i, U = u)}{\text{Count}(U = u)}$

Why??

$P(x_i | P_a(x_i)) = P(x_i | U)$

$\hat{P}_{\text{MLE}}(X_i | U) = \frac{\text{Count}(X_i = x_i, U = u)}{\text{Count}(U = u)}$

MI F:

$P(X_i = x_i | X_j = x_j) = \frac{\text{Count}(X_i = x_i, X_j = x_j)}{\text{Count}(X_j = x_j)}$
Learning the CPTs

For each discrete variable $X_i$

$$ P(X_i = x_i | X_j = x_j) = \frac{\text{Count}(X_i = x_i, X_j = x_j)}{\text{Count}(X_j = x_j)} $$

WHY???????????

if only one var
then take derivative, set to 0
all is good

Maximum likelihood estimation (MLE) of BN parameters – example

Given structure, log likelihood of data:

$$ \log P(D | \Theta_G, G) = \sum_{i \in \text{PC}} \sum_{j} \log P(x^{(i)} | \Theta_{G}, G) - \sum_{i \in \text{PC}} \sum_{j} \log P(x^{(i)} | \Theta_{G}, G) $$

Break up into independent subproblems: one for each CPT
Maximum likelihood estimation (MLE) of BN parameters – General case

- Data: $x^{(1)}, \ldots, x^{(m)}$
- Restriction: $x^{(j)}[\text{Pa}_{X_i}] \rightarrow \text{assignment to } \text{Pa}_{X_i} \text{ in } x^{(j)}$
- Given structure, log likelihood of data:

$$
\log P(\mathcal{D} \mid \theta_G, G) = \sum_{j=1}^{m} \left( \sum_{i=1}^{n} \log \frac{\sum_{x_i^{(j)} \in \text{Pa}_{X_i}} \log P(x^{(i)} \mid x^{(j)}[\text{Pa}_{X_i}], \theta_{X_i}[\text{Pa}_{X_i}])}{\sum_{x_i^{(j)} \in \text{Pa}_{X_i}} \log P(x^{(i)} \mid x^{(j)}[\text{Pa}_{X_i}], \theta_{X_i}[\text{Pa}_{X_i}])} \right)
$$

Sol. MLE: $\hat{P}(X_i = x_i \mid U = u) = \frac{\text{count}(x_i = x_i, U = u)}{\text{count}(U = u)}$

Taking derivatives of MLE of BN parameters – General case

$$
\frac{\partial \log P(\mathcal{D} \mid \theta_G, G)}{\partial \theta_{X_i}[\text{Pa}_{X_i}]} = \frac{\partial}{\partial \theta_{X_i}[\text{Pa}_{X_i}]} \left[ \sum_{j=1}^{m} \frac{1}{\text{count}(U = u)} \log \frac{\sum_{x_i^{(j)} \in \text{Pa}_{X_i}} \log P(x^{(i)} \mid x^{(j)}[\text{Pa}_{X_i}], \theta_{X_i}[\text{Pa}_{X_i}])}{\sum_{x_i^{(j)} \in \text{Pa}_{X_i}} \log P(x^{(i)} \mid x^{(j)}[\text{Pa}_{X_i}], \theta_{X_i}[\text{Pa}_{X_i}])} \right]
$$

Same as usual $\Rightarrow \hat{P}(X_i = x_i \mid \text{Pa}_{X_i} = u) = \frac{\text{count}(x_i = x_i, \text{Pa}_{X_i} = u)}{\text{count}(\text{Pa}_{X_i} = u)}$
General MLE for a CPT

- Take a CPT: $P(X | U)$
- Log likelihood term for this CPT

- Parameter $\theta_{X=x | U=u}$:

\[
\text{MLE: } P(X = x \mid U = u) = \theta_{X=x | U=u} = \frac{\text{Count}(X = x, U = u)}{\text{Count}(U = u)}
\]

Where are we with learning BNs?

- Given structure, estimate parameters
  - Maximum likelihood estimation
  - Later Bayesian learning
- What about learning structure?
Learning the structure of a BN

- Constraint-based approach
  - BN encodes conditional independencies
  - Test conditional independencies in data
  - Find an I-map

- Score-based approach
  - Finding a structure and parameters is a density estimation task
  - Evaluate model as we evaluated parameters
    - Maximum likelihood
    - Bayesian
    - etc.

Data

<\(x_1^{(1)}, \ldots, x_n^{(1)}\)>
...
<\(x_1^{(m)}, \ldots, x_n^{(m)}\)>

Learn structure and parameters

Remember: Obtaining a P-map?

- Given the independence assertions that are true for \(P\)
  - Obtain skeleton
  - Obtain immoralities
- From skeleton and immoralities, obtain every (and any) BN structure from the equivalence class

- Constraint-based approach:
  - Use Learn PDAG algorithm
  - Key question: Independence test
Score-based approach

Possible structures

Data

\[ <x_1^{(1)}, \ldots, x_n^{(1)}>, \ldots, <x_1^{(m)}, \ldots, x_n^{(m)}> \]

Score structure

\[ \log P(D | q, G) \]

Learn parameters

-50

-125

-250

Information-theoretic interpretation of maximum likelihood

- Given structure, log likelihood of data:

\[
\log P(D | q, G) = \sum_{i=1}^{n} \log \left[ \sum_{x_{-i}^{(i)}} P(x_{i}^{(i)} | P_{a_{x_{i}^{(i)}}} = x_{-i}^{(i)} | P_{a_{x_{i}^{(i)}}}) \right]
\]

\[
= \sum_{i=1}^{n} \left[ \sum_{x_{i}^{(i)}} \log P(x_{i}^{(i)} | P_{a_{x_{i}^{(i)}}} = x_{-i}^{(i)} | P_{a_{x_{i}^{(i)}}}) \right]
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} \left[ \sum_{x_{i}^{(i)}} \log P(x_{i}^{(i)} | P_{a_{x_{i}^{(i)}}} = x_{-i}^{(i)} | P_{a_{x_{i}^{(i)}}}) \right]
\]

\[
= \frac{1}{n} \sum_{x_{i}^{(i)}} \log P(x_{i}^{(i)} | P_{a_{x_{i}^{(i)}}} = x_{-i}^{(i)} | P_{a_{x_{i}^{(i)}}})
\]

\[
\text{MLE:} \quad \hat{P}(x_{i}^{(i)} | P_{a_{x_{i}^{(i)}}} = x_{-i}^{(i)} | P_{a_{x_{i}^{(i)}}}) = \frac{\text{count}(x_{i}^{(i)}, P_{a_{x_{i}^{(i)}}}): \text{number of times}}{m}
\]

\[
\text{if MLE:} \quad \hat{P}(x_{i}^{(i)} | P_{a_{x_{i}^{(i)}}} = x_{-i}^{(i)} | P_{a_{x_{i}^{(i)}}}) = \frac{\text{count}(x_{i}^{(i)}, P_{a_{x_{i}^{(i)}}})}{m}
\]

\[
= \frac{1}{n} \sum_{x_{i}^{(i)}} \log P(x_{i}^{(i)} | P_{a_{x_{i}^{(i)}}} = x_{-i}^{(i)} | P_{a_{x_{i}^{(i)}}})
\]

\[
= \frac{1}{n} \left[ \sum_{x_{i}^{(i)}} \log P(x_{i}^{(i)} | P_{a_{x_{i}^{(i)}}} = x_{-i}^{(i)} | P_{a_{x_{i}^{(i)}}}) \right]
\]

\[
= \frac{1}{n} \left[ \sum_{x_{i}^{(i)}} \log P(x_{i}^{(i)} | P_{a_{x_{i}^{(i)}}} = x_{-i}^{(i)} | P_{a_{x_{i}^{(i)}}}) \right]
\]

\[
= \frac{1}{n} \left[ \sum_{x_{i}^{(i)}} \log P(x_{i}^{(i)} | P_{a_{x_{i}^{(i)}}} = x_{-i}^{(i)} | P_{a_{x_{i}^{(i)}}}) \right]
\]
Information-theoretic interpretation of maximum likelihood 2

Given structure, log likelihood of data:

\[
\log P(D | \theta, G) = \frac{m}{n} \sum_{i=1}^{n} \sum_{x_i} P(x_i, \text{Pa}_{x_i}, D) \log P(x_i | \text{Pa}_{x_i}, D)
\]

\[
H(X_i | \text{Pa}_{x_i}) = -\sum_{x_i} P(x_i) \log P(x_i)
\]

Decomposable score

Decomposes over families in BN (node and its parents)

Will lead to significant computational efficiency!!!

Score(G : D) = \sum_{i} \text{FamScore}(X_i | \text{Pa}_{x_i} : D)

\[
\text{FamScore}(X_i | \text{Pa}_{x_i} : D) = m \hat{I}(X_i ; \text{Pa}_{x_i}) - m \hat{H}(X_i)
\]
Announcements

- Recitation tomorrow
  - Don’t miss it!

- HW2
  - Out today
  - Due in 2 weeks

- Projects!!!
  - Proposals due Oct. 8th in class
  - Individually or groups of two
  - Details on course website
  - Project suggestions will be up soon!!!

BN code release!!!!

- Pre-release of a C++ library for probabilistic inference and learning

  - Features:
    - basic datastructures (random variables, processes, linear algebra)
    - distributions (Gaussian, multinomial, ...)
    - basic graph structures (directed, undirected)
    - graphical models (Bayesian network, MRF, junction trees)
    - inference algorithms (variable elimination, loopy belief propagation, filtering)
  - Limited amount of learning (IPF, Chow Liu, order-based search)

- Supported platforms:
  - Linux (tested on Ubuntu 8.04)
  - MacOS X (tested on 10.4/10.5)
  - limited Windows support

- Will be made available to the class early next week.
How many trees are there? Nonetheless – Efficient optimal algorithm finds best tree

\[ \Theta(n \log n) \]

Scoring a tree 1: I-equivalent trees

\[
\log \hat{P}(D | \theta, G) = m \sum \hat{f}(X_i, Pa_{X_i}) - m \sum \hat{H}(X_i)
\]

\[
\begin{align*}
A \rightarrow B \rightarrow C \\
\text{Score: } m[I(B,A) + I(C,B) - H(A) - H(B) - H(C) + I(A, \emptyset)]
\end{align*}
\]

\[ A \leftarrow B \rightarrow C \]

\[
\begin{align*}
m[I(A,B) + I(C,B) - H(B) - H(A) - H(C)]
\end{align*}
\]

\[ A \leftarrow B \leftrightarrow C \]

\[
\begin{align*}
m[I(A,B) + I(B,C) - H(A) - H(B) - H(C)]
\end{align*}
\]

SAME SKELETON \( \Rightarrow \) SAME Score \( \uparrow \) (only for trees)
Scoring a tree 2: similar trees

\[
\log \hat{P}(D | \theta, G) = m \sum_i \hat{I}(X_i, \text{Pa}_{X_i}) - m \sum_i \hat{H}(X_i)
\]

\[G_1\]

\[
A \rightarrow B \rightarrow C
\]

\[
m \left( I(A, B) + I(B, C) - H(A) - H(B) - H(C) \right)
\]

\[
G_2
\]

\[
B \rightarrow A \rightarrow C
\]

\[
m \left( I(B, A) + I(B, C) - H(A) - H(B) - H(C) \right)
\]

\[
\text{Score}(G) = m \sum_{i \in \text{skeleton}} I(x_i | x_j) - m \sum_i H(x_i)
\]

\begin{align*}
\text{same:} & \quad A - B \quad \Rightarrow \quad \text{score} = I(A, B) \\
\text{different:} & \quad G_1 \quad \text{or} \quad G_2
\end{align*}

Chow-Liu tree learning algorithm 1

- For each pair of variables $X_i, X_j$
  - Compute empirical distribution:
    \[
p(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{m}
\]
  - Compute mutual information:
    \[
    I(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{\hat{p}(x_i, x_j)}{p(x_i)p(x_j)}
    \]
- Define a graph
  - Nodes $X_1, \ldots, X_n$
  - Edge $(i, j)$ gets weight $I(X_i, X_j)$
  - Find maximum spanning tree
  - Choose best tree $\text{BU}$
Chow-Liu tree learning algorithm 2

\[
\log \hat{P}(D \mid \theta, \mathcal{G}) = M \sum_i \hat{I}(x_i, \text{Pa}_{x_i}, \mathcal{G}) - M \sum_i \hat{H}(x_i)
\]

- Optimal tree BN
  - Compute maximum weight spanning tree
  - Directions in BN: pick any node as root, breadth-first-search defines directions

Can we extend Chow-Liu 1

- Tree augmented naïve Bayes (TAN) [Friedman et al. '97]
  - Naïve Bayes model overcounts, because correlation between features not considered
  - Same as Chow-Liu, but score edges with:
  \[
  I(X_i, X_j \mid \mathcal{C}) = \sum_{x_i, x_j} \hat{P}(x_i, x_j \mid \mathcal{C}) \log \frac{\hat{P}(x_i, x_j \mid \mathcal{C})}{\hat{P}(x_i \mid \mathcal{C}) \hat{P}(x_j \mid \mathcal{C})}
  \]