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

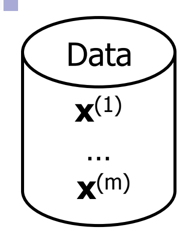
Param. Learning (MLE)

Structure Learning
The Good

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

October 1st, 2008

Learning the CPTs

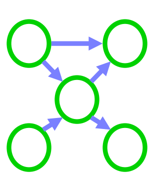


For each discrete variable X_i $Pa_{X_i} = U$

$P(X_i \mid Pa_{X_i}) = P(X_i \mid U)$

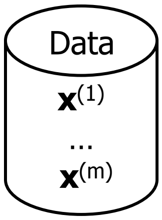
$\hat{P}_{MLE}(X_i = x_i \mid U = u) = \frac{Count(X_i = x_i, U = u)}{Count(U = u)}$

Why??



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

Learning the CPTs

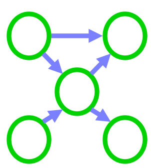


For each discrete variable X_i

MLE: $P(X_i = x_i \mid X_j = x_j) = \dfrac{\text{Count}(X_i = x_i, X_j = x_j)}{\text{Count}(X_j = x_j)}$

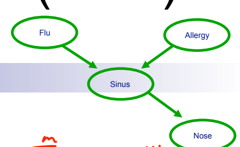
WHY??????????

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



Maximum likelihood estimation (MLE) of BN parameters – example

$\log a \cdot b = \log a + \log b$



- Given structure, log likelihood of data:

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) = \log \prod_{j=1}^{m} P(x^{(j)} \mid \theta_G, G) = \sum_{j=1}^{m} \log P(x^{(j)} \mid \theta_G, G)$$

for the example

$$\sum_{j=1}^{m} \log P(f^{(j)}, a^{(j)}, s^{(j)}, n^{(j)} \mid \theta_G, G) = \sum_{j=1}^{m} \log P(f^{(j)} \mid \theta_G, G) \cdot P(a^{(j)} \mid \theta_G, G)\, P(s^{(j)} \mid a^{(j)}, f^{(j)}, \theta_G, G)\, P(n^{(j)} \mid s^{(j)}, \theta_G, G)$$

$$= \sum_{j=1}^{m} \left[\log P(f^{(j)} \mid \theta_G, G) + \log P(a^{(j)} \mid \theta_G, G) + \log P(s^{(j)} \mid a^{(j)}, f^{(j)}, \theta_G, G) + \log P(n^{(j)} \mid s^{(j)}, \theta_G, G) \right]$$

$$= \sum_{j=1}^{m} \log P(f^{(j)} \mid \theta_F, G) + \sum_{j=1}^{m} \log P(a^{(j)} \mid \theta_A, G) + \sum_{j=1}^{m} \log P(s^{(j)} \mid a^{(j)}, f^{(j)}, \theta_{S|FA}, G) + \sum_{j=1}^{m} \log P(n^{(j)} \mid s^{(j)}, \theta_{N|S}, G)$$

Broke up problem into independent subproblems: one for each CPT

Maximum likelihood estimation (MLE) of BN parameters – General case

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

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G})$$

Taking derivatives of MLE of BN parameters – General case

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

General MLE for a CPT

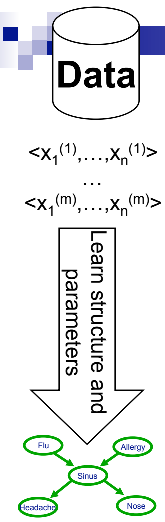
- Take a CPT: P(X|**U**)
- Log likelihood term for this CPT
- Parameter $\theta_{X=x|\mathbf{U}=\mathbf{u}}$:

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

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.

Remember: Obtaining a P-map?

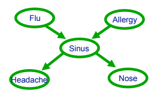
- Given the independence assertions that are true for *P*
  - Obtain skeleton
  - Obtain immoralities
- From skeleton and immoralities, obtain every (and any) BN structure from the equivalence class

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

Score-based approach

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

Possible structures

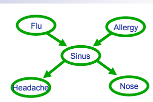


Learn parameters

Score structure

Information-theoretic interpretation of maximum likelihood

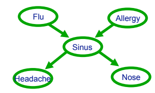
- Given structure, log likelihood of data:



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

Information-theoretic interpretation of maximum likelihood 2

- Given structure, log likelihood of data:



$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \sum_{x_i, \mathbf{Pa}_{x_i,\mathcal{G}}} \hat{P}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) \log \hat{P}(x_i \mid \mathbf{Pa}_{x_i,\mathcal{G}})$$

Decomposable score

- Log data likelihood

$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \hat{I}(X_i, \mathbf{Pa}_{X_i}) - m \sum_i \hat{H}(X_i)$$

- Decomposable score:
  - Decomposes over families in BN (node and its parents)
  - Will lead to significant computational efficiency!!!
  - Score($G : D$) = $\sum_i$ FamScore($X_i|\mathbf{Pa}_{X_i} : D$)

Announcements

- Recitation tomorrow
  - Don't miss it!
- HW2
  - Out today
  - Due in 2 weeks
- Projects!!! ☺
  - Proposals due Oct. 8^{th} 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**

Scoring a tree 1: I-equivalent trees

- $$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \hat{I}(X_i, \mathbf{Pa}_{X_i}) - m \sum_i \hat{H}(X_i)$$

Scoring a tree 2: similar trees

- $\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \hat{I}(X_i, \mathbf{Pa}_{X_i}) - m \sum_i \hat{H}(X_i)$

Chow-Liu tree learning algorithm 1

- For each pair of variables X_i,X_j
  - Compute empirical distribution:
    $$\hat{P}(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{m}$$
  - Compute mutual information:
    $$\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \hat{P}(x_i, x_j) \log \frac{\hat{P}(x_i, x_j)}{\hat{P}(x_i)\hat{P}(x_j)}$$
- Define a graph
  - Nodes $X_1,\ldots,X_n$
  - Edge (i,j) gets weight $\hat{I}(X_i, X_j)$

Chow-Liu tree learning algorithm 2

$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = M \sum_i \hat{I}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) - M \sum_i \hat{H}(X_i)$$

- 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:

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

Can we extend Chow-Liu 2

- (Approximately learning) models with tree-width up to *k*
  - [Chechetka & Guestrin ’07]
  - But, $O(n^{2k+6})$

What you need to know about learning BN structures so far

- Decomposable scores
  - Maximum likelihood
  - Information theoretic interpretation
- Best tree (Chow-Liu)
- Best TAN
- Nearly best k-treewidth (in $O(N^{2k+6})$)