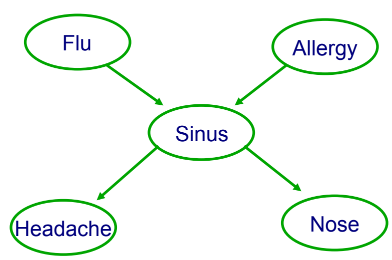
Bayesian Networks – (Structure) Learning

Machine Learning – CSE446
Carlos Guestrin
University of Washington

May 31, 2013

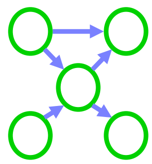
Review

- Bayesian Networks
  - Compact representation for probability distributions
  - Exponential reduction in number of parameters
- Fast probabilistic inference
  - As shown in demo examples
  - Compute P(X|e)
- Today
  - Learn BN structure



Learning Bayes nets

Data
$\mathbf{x}^{(1)}$
…
$\mathbf{x}^{(m)}$



structure

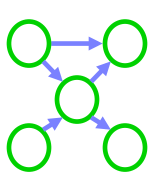
+

CPTs – $P(X_i \mid \mathbf{Pa}_{X_i})$

parameters

Learning the CPTs

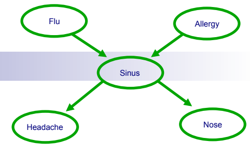
Data
$\mathbf{x}^{(1)}$
…
$\mathbf{x}^{(m)}$



For each discrete variable X_i

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

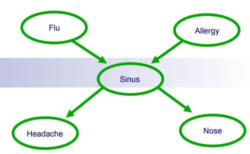
Information-theoretic interpretation of maximum likelihood 1



- Given structure, log likelihood of data:

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

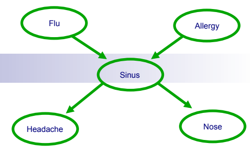
Information-theoretic interpretation of maximum likelihood 2



- 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 3



- 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

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

- 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)

How many trees are there?

Nonetheless – Efficient optimal algorithm finds best tree

Scoring a tree 1: equivalent trees

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

Scoring a tree 2: similar trees

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

Chow-Liu tree learning algorithm 1

- For each pair of variables X_i, X_j
  - Compute empirical distribution:

$$\widehat{P}(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{m}$$

  - Compute mutual information:

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

- Define a graph
  - Nodes $X_1, \ldots, X_n$
  - Edge (i,j) gets weight $\widehat{I}(X_i, X_j)$

Chow-Liu tree learning algorithm 2

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

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

Structure learning for general graphs

- In a tree, a node only has one parent
- **Theorem**:
  - The problem of learning a BN structure with at most *d* parents is NP-hard for any (fixed) *d*>1
- Most structure learning approaches use heuristics
  - (Quickly) Describe the two simplest heuristic

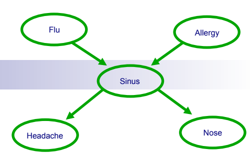
Learn BN structure using local search

Starting from Chow-Liu tree

Local search,
possible moves:
• Add edge
• Delete edge
• Invert edge

Score using BIC

Learn Graphical Model Structure using LASSO



- Graph structure is about selecting parents:
- If no independence assumptions, then CPTs depend on all parents:
- With independence assumptions, depend on key variables:
- One approach for structure learning, sparse logistic regression!

What you need to know about learning BN structures

- Decomposable scores
  - Maximum likelihood
  - Information theoretic interpretation
- Best tree (Chow-Liu)
- Beyond tree-like models is NP-hard
- Use heuristics, such as:
  - Local search
  - LASSO