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

Param. Learning (MLE)

Structure Learning for BN

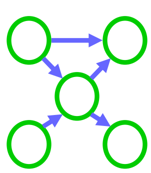
The Good

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

October 1st, 2008

Learning the CPTs

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



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

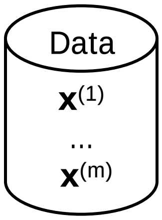
$P(X_i \mid Pa_{X_i}) = P(X_i \mid U)$

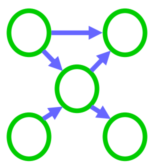
$$\hat{P}_{MLE}(X_i = x_i \mid U = u) = \frac{\text{Count}(X_i = x_i, U = u)}{\text{Count}(U = u)}$$

Why??

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

Learning the CPTs





For each discrete variable X_i

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

$\boxed{\textbf{WHY??????????}}$

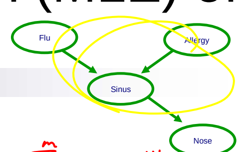
if only one var

then take derivative, set to ϕ

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

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

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

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

Broke up problem into independent subproblems : one for each CPT

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

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

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

$$= \max_{\theta_{X_1|Pa_{X_1}}} \max_{\theta_{X_2|Pa_{X_2}}} \ldots \max_{\theta_{X_n|Pa_{X_n}}} \sum_{i=1}^{n} \left[\sum_{j=1}^{m} \log P(x_i^{(j)} \mid x^{(j)}[Pa_{X_i}], \theta_{X_i|Pa_{X_i}}) \right]$$

$$= \sum_{i=1}^{n} \Bigg[\underbrace{\max_{\theta_{X_i|Pa_{X_i}}} \sum_{j=1}^{m} \log P(x_i^{(j)} \mid x^{(j)}[Pa_{X_i}], \theta_{X_i|Pa_{X_i}})}_{\text{indep. Max prob.}} \Bigg]$$

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

Taking derivatives of MLE of BN parameters – General case

param sharing decomposition a little diff. e.g. HMM

$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow \ldots$ same CPT

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

$$\frac{\partial \log P(D \mid \theta_G, G)}{\partial \theta_{X_7|Pa_{X_7}}} = \frac{\partial}{\partial \theta_{X_7|Pa_{X_7}}} \left[\sum_{j=1}^{m} \sum_{i=1}^{n} \log P(X_i = x_i^{(j)} \mid Pa_{X_i} = x^{(j)}[Pa_{X_i}], \theta_{X_i|Pa_{X_i}}) \right]$$

$$= \underbrace{\sum_{j=1}^{m} \frac{\partial}{\partial \theta_{X_7|Pa_{X_7}}} \log P(X_7 = x_7^{(j)} \mid Pa_{X_7} = x^{(j)}[Pa_{X_7}], \theta_{X_7|Pa_{X_7}}) = 0}$$

Same as usual $\Longrightarrow$ $\hat{P}(X_7 = x_7 \mid Pa_{X_7} = u) \stackrel{MLE}{=} \dfrac{\text{Count}(X_7 = x_7, U = u)}{\text{Count}(U = u)}$

General MLE for a CPT

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

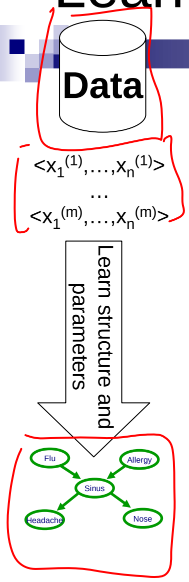
- Parameter $\theta_{X=x|\mathbf{U}=\mathbf{u}}$:

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



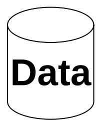
- **Constraint-based approach**
  - BN encodes conditional independencies
  - Test conditional independencies in data
  - Find an I-map
- **Score-based approach**
  - Finding a structure and parameters is a density estimation task
  - Evaluate model as we evaluated parameters
    - Maximum likelihood
    - 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

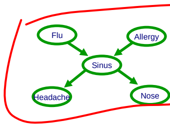


<$x_1^{(1)},\ldots,x_n^{(1)}$>

…

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

Possible structures



Learn parameters

Score structure

may be $\log P(D \mid \theta, G)$

others

✓ −50

F ↓ A → H → N ↓ S

−125

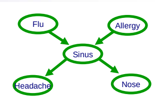
F A

S H N

−250

Information-theoretic interpretation of maximum likelihood

- Given structure, log likelihood of data:



fixed

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

$$= \sum_{i=1}^{n} \left[\sum_{j=1}^{m} \log P\left(X_i = x_i^{(j)} \mid Pa_{X_i} = x^{(j)}[Pa_{X_i}]\right)_\theta \right]$$

$$= \sum_{i=1}^{n} \left[\sum_{x_i} \sum_{u \in Val(Pa_{X_i})} Count(X_i = x_i, Pa_{X_i} = u) \log P(X_i = x_i \mid Pa_{X_i} = u)_\theta \right]$$

$$= m \sum_{i=1}^{n} \left[\sum_{x_i} \sum_{u \in Val(Pa_{X_i})} \hat{P}(x_i, Pa_{X_i} = u) \log P(X_i = x_i \mid Pa_{X_i} = u)_\theta \right]$$

if MLE $\theta \leftarrow \hat{P}(X_i = x_i \mid Pa_{X_i} = u)$

$X_i = x_i,\ Pa_{X_i} = u$

$Count(X_i = x_i, Pa_{X_i} = u)$

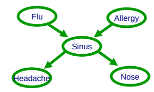
number of times

MLE:

$\hat{P}(X_i = x_i, Pa_{X_i} = u) = \frac{Count(x_i, u)}{m}$

Information-theoretic interpretation of maximum likelihood 2

- Given structure, log likelihood of data:



$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) \overset{MLE}{=} m \sum_i \underbrace{\sum_{x_i, \mathbf{Pa}_{x_i,\mathcal{G}}} \hat{P}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) \log \hat{P}(x_i \mid \mathbf{Pa}_{x_i,\mathcal{G}})}_{-H(X_i \mid Pa_{X_i})}$$

$$= -m \sum_i \hat{H}(X_i \mid Pa_{X_i})$$

$$= m \sum_i \hat{I}(X_i, Pa_{X_i}) - \underbrace{m \sum_i \hat{H}(X_i)}_{\text{doesn't depend on structure}}$$

$H(A \mid B) = -\sum_a \sum_b P(a,b) \log P(a \mid b)$

$H(A \mid B) = H(A)$ $\quad A \perp B$

$H(A \mid B) = 0$ $\quad$ A B perfectly correlated

$H(A \mid B) \geq 0$

$I(A,B) = H(A) - H(A \mid B)$

$I(X_i, Pa_{X_i}) = H(X_i) - H(X_i \mid Pa_{X_i})$

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)

for MLE $\quad \text{FamScore}(X_i \mid Pa_{X_i} : D) = m \hat{I}(X_i, Pa_{X_i}) - m \hat{H}(X_i)$

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!!!

D it's good related to research, but must be new

D it must have something to do w: Graphical models

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?

parents = 1

- **Nonetheless – Efficient optimal algorithm finds best tree**

$A \to B$, $B \to C$, $B \to D$

$A \to C \to D \to B$

$\vdots$

$$2^{\Theta(n \log n)}$$

Scoring a tree 1: I-equivalent trees

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

$A \to B \to C$

Score: $m[I(B,A) + I(C,B) - H(A) - H(B) - H(C) + \cancel{I(A,\emptyset)}]$ ($I(A,\emptyset) = 0$)

$A \leftarrow B \to C$

$m[I(A,B) + I(C,B) - H(B) - H(A) - H(C)]$

$A \leftarrow B \leftarrow C$

$m[I(A,B) + I(B,C) - H(A) - H(B) - H(C)]$

I-equivalent

same score!! :)

$A \to B \leftarrow C$

not a tree!! :(

SAME Skeleton ⇒ Same Score!! :) (only for trees)

Scoring a tree 2: similar trees

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

G_1

$A \to B \to C$

$m[I(A,B) + \underline{I(B,C)} - H(A) - H(B) - H(C)]$

G_2

$B \to A \to C$

$m[I(B,A) + \underline{I(A,C)} - H(A) - H(B) - H(C)]$

Score(G_1) - Score(G_2)

$= I(B,C) - I(A,C)$

edges skeleton same:

$A - B$

$\Rightarrow$ score $\leftarrow I(A,B)$

different

G_1	G_2
$B - C$	$A - C$
$I(B,C)$	$I(A,C)$

$$\text{Score(tree)} = m \sum_{\substack{ij \\ \text{in skeleton}}} I(X_i, X_j) \; \underbrace{- \; m \sum_i H(X_i)}_{\text{constant ignore}}$$

Chow-Liu tree learning algorithm 1

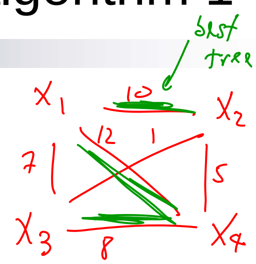
- For each pair of variables X_i, X_j
  - Compute empirical distribution:

$$\hat{P}(x_i, x_j) \overset{\text{MLE}}{=} \frac{\text{Count}(x_i, x_j)}{m}$$

  - 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 $\hat{I}(X_i, X_j)$ w_{ij}



max trees ↑ score(tree)

$= \sum_{ij} I(X_i, X_j)$

$= \sum_{ij} w_{ij}$

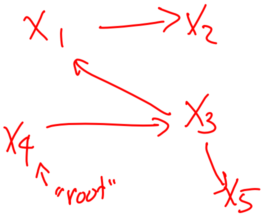
find maximum spanning tree → best tree BN

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

all trees (BNs) with 1 parent & same skeleton have no immoralities
⇒ I equivalent
⇒ pick any

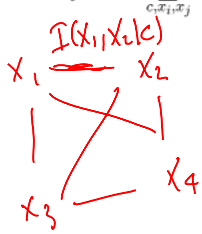


using Chow-Liu
OPTIMAL tree BN

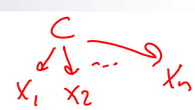
Can we extend Chow-Liu 1

- 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:

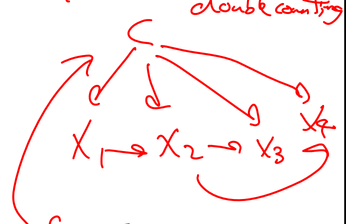
$$\hat{I}(X_i, X_j \mid C) = \sum_{c, x_i, x_j} \hat{P}(c, x_i, x_j) \log \frac{\hat{P}(x_i, x_j \mid c)}{\hat{P}(x_i \mid c)\hat{P}(x_j \mid c)}$$



maximum spanning tree, ⇒ optimal TAN



TAN can reduce double counting



SCORE:
$\cancel{I}(C,\emptyset) + I(C,X_1) + I(X_2,CX_1) + I(X_3,CX_2) + I(X_4,CX_2)$
+ entropy