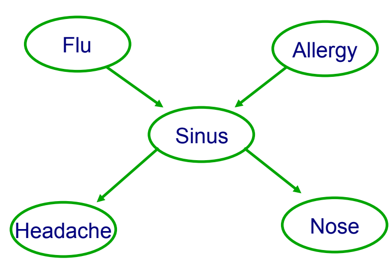
Bayesian Networks – (Structure) Learning

Machine Learning – CSE546
Carlos Guestrin
University of Washington

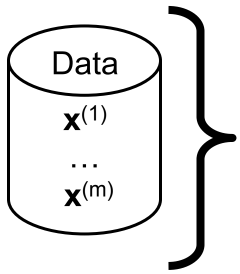
November 25, 2013

Review

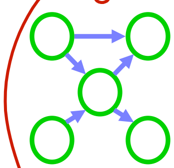
- Bayesian Networks
  - Compact representation for probability distributions
  - Exponential reduction in number of parameters
- Fast probabilistic inference
  - As shown in demo examples
  - Compute P(X|e)
- Today
  - Learn BN structure



Learning Bayes nets



G



structure

+

θ_G

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

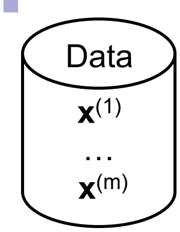
parameters

max likelihood approach

structure, & params $P(D \mid G, \theta_G)$

Learning the CPTs

$|Y|$ is # of assignments,
e.g. $Pa_{X_i} = \{F, A, H\}$, binary
$|Y| = |F,A,H| = 2^3$, $\widehat{Count}(Pa_{X_i} = u)$



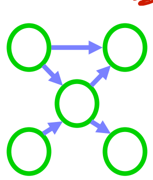
For each discrete variable X_i

$$\hat{P}(S=t \mid A=t) \overset{MLE}{=} \frac{Count(S=t, A=t)}{Count(A=t)}$$

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

Small (huge) subtlety: $Count(Pa_{X_i} = u) = 0$?

given G

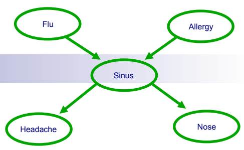


Smoothing / AKA regularization / AKA Bayesian learning

e.g. $\widehat{Count}(Y=y) = Count(Y=y) + \alpha \frac{1}{|Y|}$, for $\alpha > 0$, usually $\alpha \approx 1$

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

Information-theoretic interpretation of maximum likelihood 1



n variables, m data points

- Given structure, log likelihood of data:

max G, params, structure

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) \overset{iid}{=} \log \prod_{j=1}^{m} P(x_1^{(j)}, \dots, x_n^{(j)} \mid \theta_G, G)$$

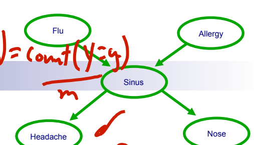
$$= \log \prod_{j=1}^{m} \prod_{i=1}^{n} P(x_i^{(j)} \mid Pa_{x_i,G} = u_i^{(j)}, \theta_G)$$

$$= \sum_{j=1}^{m} \sum_{i=1}^{n} \log P(x_i^{(j)} \mid Pa_{x_i,G} = u_i^{(j)}, \theta_G) \leftarrow \max_G$$

$$x^{(j)} \leftarrow (\underbrace{F=t, A=f}_{u_i^{(j)}}, \underbrace{S=t}_{x_i^{(j)}}, \dots)$$

Information-theoretic interpretation of maximum likelihood 2

$P(y) = \frac{Count(Y=y)}{m}$



- Given structure, log likelihood of data:

flip

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

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} \log P(x_i^{(j)} \mid Pa_{x_i,G} = u_i^{(j)})$$

$$= \sum_{i=1}^{n} \sum_{x_i} \sum_{u_i} Count(X_i = x_i, Pa_{X_i,G} = u_i) \log P(x_i \mid Pa_{x_i,G} = u_i)$$

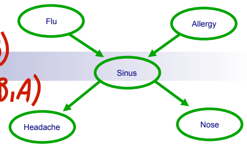
$$= m \sum_{i=1}^{n} \underbrace{\sum_{x_i} \sum_{u_i} \hat{P}(x_i \mid Pa_{x_i,G} = u_i) \log \hat{P}(x_i \mid Pa_{x_i,G} = u_i)}_{-H(X_i \mid Pa_{x_i,G})}$$

e.g. $\sum_{j=1}^{m} \log P(h^{(j)} \mid s^{(j)})$

$= Count(H=t, S=t) \cdot \log P(H=t \mid S=t)$
$+ Count(H=f, S=f) \cdot \log P(H=t \mid S=f)$
$+ Count(H=f, S=t) \cdot \log P(H=f \mid S=t)$
$+ Count(H=f, S=f) \cdot \log P(H=f \mid S=f)$

Information-theoretic interpretation of maximum likelihood 3

$\hat I(A,B) = \hat I(B,A)$



- Given structure, log likelihood of data:

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

$$\max_G = -m \sum_{i=1}^n \hat H(X_i \mid Pa_{X_i,G}) \equiv \min_G m \sum_{i=1}^n \hat H(X_i \mid Pa_{X_i,G})$$

$$\equiv \max_G m \sum_{i=1}^n \underbrace{\hat I(X_i, Pa_{X_i,G})}_{} - \underbrace{m \sum_{i=1}^n \hat H(X_i)}_{\text{Constant wrt } G}$$

Mutual Information

information theoretic measure of dependency

$H(A|B)$

$= -\sum_a \sum_b p(a,b) \log p(a|b)$

How uncertain X_i is given parents $\Rightarrow$ minimize this over G

Mutual information

$I(A,B)$

$= H(A) - H(A|B)$

Decomposable score

OK: $A \to B$, $A \to C$, $B \to C$

NOT: $A \to B$, $C \to A$, $B \to C$

- Log data likelihood

$$\max_G \log \hat P(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \underbrace{\hat I(X_i, \mathbf{Pa}_{X_i,\mathcal{G}})}_{\text{families}} - \overbrace{m \sum_i \hat H(X_i)}^{\text{constant}}$$

- Decomposable score:
  - Decomposes over families in BN (node and its parents)
  - Will lead to significant computational efficiency!!!
  - $\max_G$ Score($G : D$) = $\sum_{i=1}^n$ FamScore($X_i|\mathbf{Pa}_{X_i} : D$)

e.g. $= \sum_{i=1}^n \hat I(X_i, Pa_{X_i,G})$

also get this decomposition for other losses

How many trees are there?

Nonetheless – Efficient optimal algorithm finds best tree

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

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

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

HMM is a tree

$B \leftarrow A \to C$, $A \to D$ ← Naive Bayes is a tree

How many trees are there?

For n variables: $O(n \log n)$?

exhaustive search is impossible

$I(A,\emptyset) = 0$

Scoring a tree 1: equivalent trees

$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = m \max_{\mathcal{G}} \sum_i \hat{I}(X_i, \mathbf{Pa}_{X_i,\mathcal{G}}) - m \sum_i \hat{H}(X_i) = \max_{\mathcal{G}} \sum_{i=1} \hat{I}(X_i, \mathrm{Pa}_{X_i,\mathcal{G}})$$

$A \to B \to C$

$0 + I(B,A) + I(C,B)$ same score

$A \perp C \mid B$

$A \leftarrow B \leftarrow C$

$I(A,B) + I(B,C)$

$A \perp C \mid B$

represent same dist.

$A \leftarrow B \to C$

$A \perp C \mid B$

$I(A,B) + I(C,B)$

same score

Every graph with same independence assumptions has same score

$A \to B \leftarrow C$ $\quad A \perp C \mid \emptyset$

not same score

$I(B, AC)$ not a tree

Scoring a tree 2: similar trees

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

$A \to B \to C$ Score $I(A,B) + I(B,C)$

$B \leftarrow A \to C$ $I(A,B) + I(A,C)$

$\max_G$ Score of tree $\equiv$ Score(T) $= \sum_{(i,j) \in T} I(X_i, X_j) =$ Sum over edges of score of edge

Chow-Liu tree learning algorithm 1

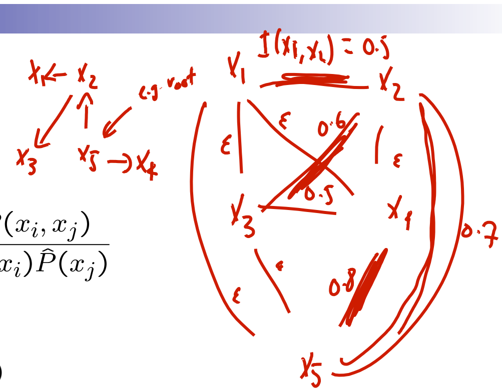
- For each pair of variables X_i, X_j
  - Compute empirical distribution:

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

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



⊳ Maximum Spanning tree: Find tree with maximum weight on edges } ← complexity about $O(E \log E)$

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 root give same score

MST: A — B — C

A → B → C   A ← B → C   A ← B ← C

Structure learning for general graphs

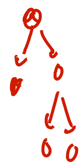
- In a tree, a node only has one parent
- **Theorem**:
  - The problem of learning a BN structure with at most *d* parents is NP-hard for any (fixed) *d>1*

for d>1

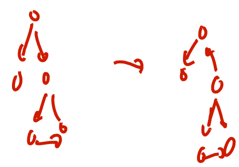
- Most structure learning approaches use heuristics
  - (Quickly) Describe the two simplest heuristic

Learn BN structure using local search

Starting from Chow-Liu tree



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



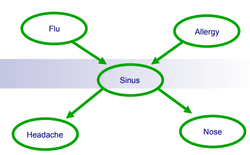
Score using BIC

penalize for dense graphs

Push away from fully connected graph

Converge to local optima

Learn Graphical Model Structure using LASSO



- Graph structure is about selecting parents:

$P(X_i \mid Pa_{X_i,G})$ ← "logistic regression"

$Pa_{X_i,G} \subset \{X_1, \ldots X_{i-1}, X_{i+1}, \ldots X_n\}$

- If no independence assumptions, then CPTs depend on all parents:

$P(H \mid F A S N)$

- With independence assumptions, depend on key variables:

$P(H \mid F A S N) = P(H \mid S)$ ← sparse conditional model where other dependencies are zero

- One approach for structure learning, sparse logistic regression!

LR for each variable: $P(X_1 \mid X_2 \ldots X_n)$ ← Sparse LR

⇓

add edges from all non-zero parents to X_i

caveat: this approach not appropriate for BNs, but used in other graphical models, like Markov Networks, undirected

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

Learning Theory

Machine Learning – CSE546
Carlos Guestrin
University of Washington

October 27, 2013

What now…

- We have explored **many** ways of learning from data
- But…
  - How good is our classifier, really?
  - How much data do I need to make it "good enough"?

A simple setting…

- Classification
  - N data points iid
  - **Finite** number of possible hypothesis (e.g., dec. trees of depth d)
- A learner finds a hypothesis *h* that is **consistent** with training data
  - Gets zero error in training – $\text{error}_{\text{train}}(h) = 0$
- What is the probability that *h* has more than ε true error?
  - $\text{error}_{\text{true}}(h) \geq \varepsilon$ For some $\varepsilon > 0$

How likely is a bad hypothesis to get *N* data points right?

- Hypothesis *h* that is **consistent** with training data → got *N* i.i.d. points right
  - h "bad" if it gets all this data right, but has high true error

$\varepsilon > 0$

- Prob. *h* with $\text{error}_{\text{true}}(h) \geq \varepsilon$ gets one data point right

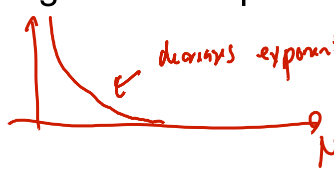
less than $1-\varepsilon$

if error $\varepsilon = 0.25$
75% points are correct $= 1-\varepsilon$

- Prob. *h* with $\text{error}_{\text{true}}(h) \geq \varepsilon$ gets *N* data points right

less than $(1-\varepsilon)^N$

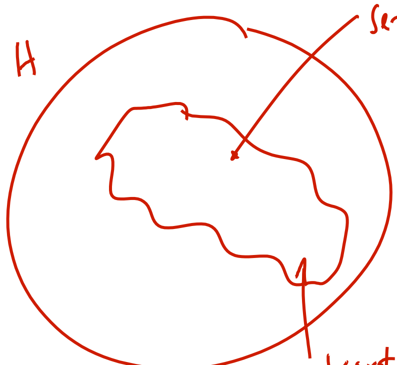
Prob bad h wins



decays exponentially in N

But there are many possible hypothesis that are consistent with training data

H



Set of hypothesis consistent with training data, don't want to make assumptions about which one was picked

worst-case bound

How likely is learner to pick a bad hypothesis

- Prob. h with $\text{error}_{\text{true}}(h) \geq \varepsilon$ gets N data points right

less than $(1-\varepsilon)^N$

$h_1, \dots h_k$

- There are k hypothesis consistent with data
  - How likely is learner to pick a bad one?

some bad, some good

⇒ deal with worst case

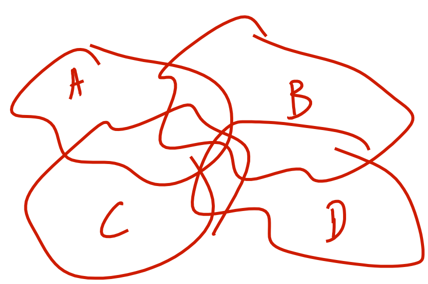
$P(\exists h$ consistent with train data, $\text{error}_{\text{true}}(h) \geq \varepsilon)$

$= P(\text{error}_{\text{true}}(h_1) \geq \varepsilon$ OR $\text{error}_{\text{true}}(h_2) \geq \varepsilon$ OR … OR $\text{error}_{\text{true}}(h_k) \geq \varepsilon)$

Bound?

Union bound

- P(A or B or C or D or …) $\leq P(A) + P(B) + P(C) + P(D) \dots$



How likely is learner to pick a bad hypothesis

- Prob. a particular h with $error_{true}(h) \geq \varepsilon$ gets N data points right less than $(1-\varepsilon)^N$
- There are k hypothesis consistent with data
  - How likely is it that learner will pick a bad one out of these k choices?

$P(\exists h$ consistent with train data, $error_{true}(h) \geq \varepsilon) \leq k(1-\varepsilon)^N$

$\leq |H| (1-\varepsilon)^N$

what's k?

$K \leq |H|$

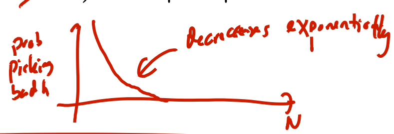
total # hypothesis

crazy loose

Generalization error in finite hypothesis spaces [Haussler '88]

- ***Theorem***: Hypothesis space H finite, dataset D with N i.i.d. samples, $0 < \varepsilon < 1$: for any learned hypothesis h that is consistent on the training data:

$$P(error_{true}(h) \geq \epsilon) \leq |H| e^{-N\epsilon}$$



$\leq |H| (1-\varepsilon)^N \leq |H| (e^{-\varepsilon})^N = |H| e^{-\varepsilon N}$

for $0 \leq \varepsilon \leq 1$

$1-\varepsilon \leq e^{-\varepsilon}$