Readings:
K&F: 17.3, 17.4, 17.5.1, 8.1, 12.1

Structure Learning
(The Good), The Bad, The Ugly

A little inference too…

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University
October 8th, 2008

Decomposable score

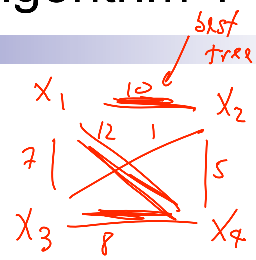
- Log data likelihood

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

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

Chow-Liu tree learning algorithm 1



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

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

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

max trees $\uparrow$ score(tree) $= \sum_{ij} I(X_i, X_j) = \sum_{ij} w_{ij}$

find maximum spanning tree → best tree BN

Maximum likelihood score overfits!

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

- Information never hurts:

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

$H(A \mid B) \leq H(A \mid C)$ $\quad C \subseteq B$

the more parents the higher $I(X_i, Pa_{X_i})$

- Adding a parent always increases score!!!

MLE ⇒ Complete Graph

Bayesian score

- Prior distributions:
  - Over structures
  - Over parameters of a structure
- Posterior over structures given data:

$P(G|D) = \frac{P(D|G)\,P(G)}{P(D)}$

$= \frac{\int_{\theta_G} P(D|G,\theta_G)\,P(\theta_G|G)\,P(G)\,d\theta_G}{P(D)}$

notes: LD $P(D|G,\theta_G)$

prior over graphs, e.g. $P(G) \propto e^{-c\,|\text{number of edges}|}$

prior over CPT parameters

$$\log P(\mathcal{G} \mid D) \propto \log P(\mathcal{G}) + \log \int_{\theta_\mathcal{G}} P(D \mid \mathcal{G}, \theta_\mathcal{G}) P(\theta_\mathcal{G}|\mathcal{G}) d\theta_\mathcal{G}$$

posteriors; prior over graphs; + constant ← log P(D)

Bayesian learning for multinomial

- What if you have a k sided coin???
- Likelihood function if **multinomial**:
  - $P(D|\theta_1,\dots,\theta_k) = \theta_1^{m_1}\,\theta_2^{m_2}\cdots\theta_k^{m_k}$
  - $\sum \theta_i = 1,\ \theta_i \geq 0$
- **Conjugate** prior for multinomial is **Dirichlet**:
  - $\theta \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_k) \sim \prod_i \theta_i^{\alpha_i - 1}$

  $\alpha_i \geq 0$

m_i ← # observations of class value or side i

- **Observe** *m* data points, m_i from assignment i, **posterior**:

$P(\theta_1 \dots \theta_k | m_1 \dots m_k) \propto P(m_1 \dots m_k|\theta_1 \dots \theta_k)\,P(\theta)$

$= \text{Dirichlet}(\alpha_1+m_1, \alpha_2+m_2, \dots, \alpha_k+m_k)$

- **Prediction**:

$E[\theta_i] = \frac{m_i + \alpha_i}{\sum_j (m_j + \alpha_j)}$

Global parameter independence, d-separation and local prediction

- Independencies in **meta BN**:

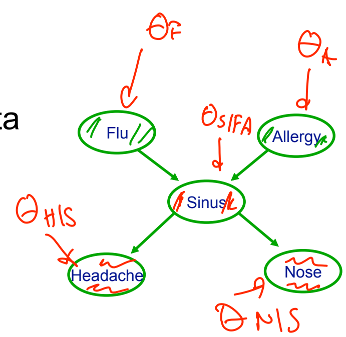
add prior vars to the BN

$P(\theta) = P(\theta_F)\,P(\theta_A)\,P(\theta_{S|FA})\,P(\theta_{N|S})\,P(\theta_{H|S})$

- **Proposition**: For fully observable data *D*, if prior satisfies global parameter independence, then

$$P(\theta \mid \mathcal{D}) = \prod_i P(\theta_{X_i|\mathbf{Pa}_{X_i}} \mid \mathcal{D})$$

params indep. given data



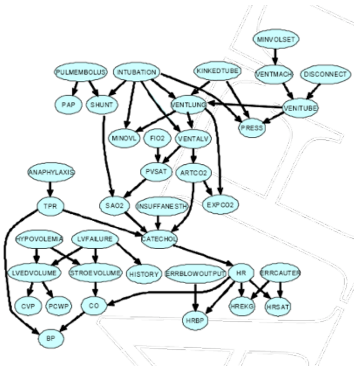
Priors for BN CPTs

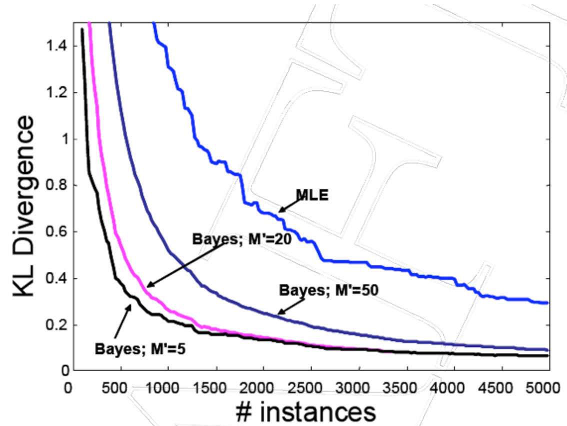
(more when we talk about structure learning)

- Consider each CPT: P(X|**U**=**u**)
- Conjugate prior:
  - Dirichlet($\alpha_{X=1|\mathbf{U}=\mathbf{u}},\ldots,\alpha_{X=k|\mathbf{U}=\mathbf{u}}$) $\equiv$ Dirichlet(Count'(X=1, U=u), ..., Count'(X=k, U=u))
- More intuitive:
  - "prior data set" *D'* with m' equivalent sample size
  - "prior counts": Count'(X=x, U=u) or $m' \cdot P'(X=x, U=u)$
  - prediction:

$$E[\theta_{X=x|U=u}] = \frac{\text{Count}(X=x, U=u) + \text{Count}'(X=x, U=u)}{\text{Count}(U=u) + \text{Count}'(U=u)}$$

An example





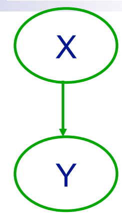
What you need to know about parameter learning

- Bayesian parameter learning:
  - motivation for Bayesian approach
  - Bayesian prediction
  - conjugate priors, equivalent sample size
  - Bayesian learning ⇒ smoothing
- Bayesian learning for BN parameters
  - Global parameter independence
  - Decomposition of prediction according to CPTs
  - Decomposition within a CPT

Bayesian score and model complexity

$$\log P(D \mid \mathcal{G}) = \log \int_{\theta_{\mathcal{G}}} P(D \mid \mathcal{G}, \theta_{\mathcal{G}}) P(\theta_{\mathcal{G}} | \mathcal{G}) d\theta_{\mathcal{G}}$$

True model:



P(Y=t|X=t) = 0.5 + α
P(Y=t|X=f) = 0.5 - α

- Structure 1: X and Y independent
  - Score doesn't depend on alpha
- Structure 2: X $\rightarrow$ Y
  - Data points split between P(Y=t|X=t) and P(Y=t|X=f)
  - For fixed M, only worth it for large α
    - Because posterior over parameter will be more diffuse with less data

Bayesian, a decomposable score

$$\log P(D \mid \mathcal{G}) = \log \int_{\theta_{\mathcal{G}}} P(D \mid \mathcal{G}, \theta_{\mathcal{G}}) P(\theta_{\mathcal{G}} | \mathcal{G}) d\theta_{\mathcal{G}}$$

- As with last lecture, assume:
  - Parameter independence
- Also, prior satisfies **parameter modularity**:
  - If X_i has same parents in *G* and *G'*, then parameters have same prior
- Finally, structure prior P(*G*) satisfies **structure modularity**
  - Product of terms over families
  - E.g., P(*G*) $\propto$ $c^{|G|}$
- Bayesian score decomposes along families!

BIC approximation of Bayesian score

- Bayesian has difficult integrals
- For Dirichlet prior, can use simple Bayes information criterion (BIC) approximation
  - In the limit, we can forget prior!
  - **Theorem**: for Dirichlet prior, and a BN with Dim(*G*) independent parameters, as m→∞:

$$\log P(D \mid \mathcal{G}) = \log P(D \mid \mathcal{G}, \theta_{\mathcal{G}}) - \frac{\log m}{2}\text{Dim}(\mathcal{G}) + O(1)$$

BIC approximation, a decomposable score

- BIC: $\text{Score}_{\text{BIC}}(\mathcal{G} : D) = \log P(D \mid \mathcal{G}, \theta_{\mathcal{G}}) - \frac{\log m}{2}\text{Dim}(\mathcal{G})$

- Using information theoretic formulation:

$$\text{Score}_{\text{BIC}}(\mathcal{G} : D) = m \sum_i \hat{I}(X_i, \mathbf{Pa}_{X_i,\mathcal{G}}) - m \sum_i \hat{H}(X_i) - \frac{\log m}{2} \sum_i \text{Dim}(P(X_i \mid \mathbf{Pa}_{X_i,\mathcal{G}}))$$

Consistency of BIC and Bayesian scores

Consistency is limiting behavior, says nothing about finite sample size!!!

- A scoring function is **consistent** if, for true model G^*, as $m \rightarrow \infty$, with probability 1
  - G^* maximizes the score
  - All structures **not I-equivalent** to G^* have strictly lower score
- **Theorem**: BIC score is consistent
- **Corollary**: the Bayesian score is consistent
- What about maximum likelihood score?

Priors for general graphs

- For finite datasets, prior is important!
- Prior over structure satisfying prior modularity
- What about prior over parameters, how do we represent it?
  - *K2 prior*: fix an α, $P(\theta_{X_i|\mathbf{Pa}_{X_i}})$ = Dirichlet($\alpha, \ldots, \alpha$)
  - K2 is "inconsistent"

BDe prior

- Remember that Dirichlet parameters analogous to "fictitious samples"
- Pick a fictitious sample size m'
- For each possible family, define a prior distribution $P(X_i, \mathbf{Pa}_{X_i})$
  - Represent with a BN
  - Usually independent (product of marginals)
- **BDe prior**:
- Has "consistency property":

Score equivalence

- If *G* and *G'* are **I**-equivalent then they have same score
- **Theorem 1**: Maximum likelihood score and BIC score satisfy score equivalence
- **Theorem 2**:
  - If P(*G*) assigns same prior to **I**-equivalent structures (e.g., edge counting)
  - and parameter prior is dirichlet
  - then **Bayesian score satisfies score equivalence** *if and only if* prior over parameters represented as a **BDe prior**!!!!!!

Chow-Liu for Bayesian score

- Edge weight $w_{X_j \to X_i}$ is advantage of adding X_j as parent for X_i

- Now have a directed graph, need directed spanning forest
  - Note that adding an edge can hurt Bayesian score – choose forest not tree
  - Maximum spanning forest algorithm works

Structure learning for general graphs

- In a tree, a node only has one parent

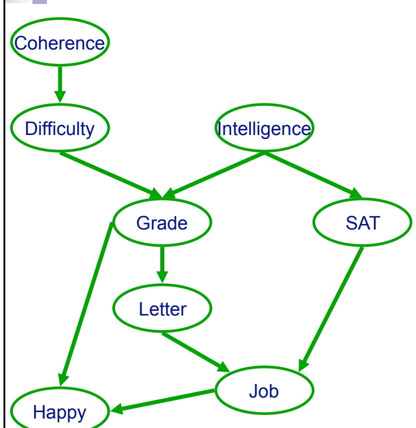
- **Theorem**:
  - The problem of learning a BN structure with at most *d* parents is NP-hard for any (fixed) $d \geq 2$

- Most structure learning approaches use heuristics
  - Exploit score decomposition
  - (Quickly) Describe two heuristics that exploit decomposition in different ways

Announcements

- **Recitation tomorrow**
  - **Don't miss it!!! ☺**

Understanding score decomposition



Fixed variable order 1

- Pick a variable order
  - e.g., $X_1,\ldots,X_n$
- X_i can only pick parents in $\{X_1,\ldots,X_{i-1}\}$
  - Any subset
  - Acyclicity guaranteed!
- Total score = sum score of each node

Fixed variable order 2

- Fix max number of parents to k
- For each *i* in order
  - Pick $\mathbf{Pa}_{X_i} \subseteq \{X_1,\ldots,X_{i-1}\}$
    - Exhaustively search through all possible subsets
    - $\mathbf{Pa}_{X_i}$ is maximum $\mathbf{U} \subseteq \{X_1,\ldots,X_{i-1}\}$ FamScore($X_i | \mathbf{U}$: *D*)
- Optimal BN for each order!!!
- Greedy search through space of orders:
  - E.g., try switching pairs of variables in order
  - If neighboring vars in order are switched, only need to recompute score for this pair
    - O(n) speed up per iteration

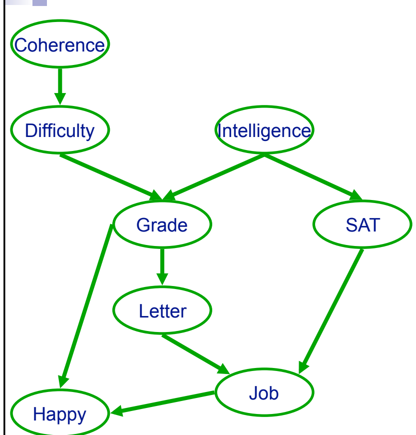
Learn BN structure using local search

Starting from Chow-Liu tree

Local search,
possible moves:
Only if acyclic!!!
- Add edge
- Delete edge
- Invert edge

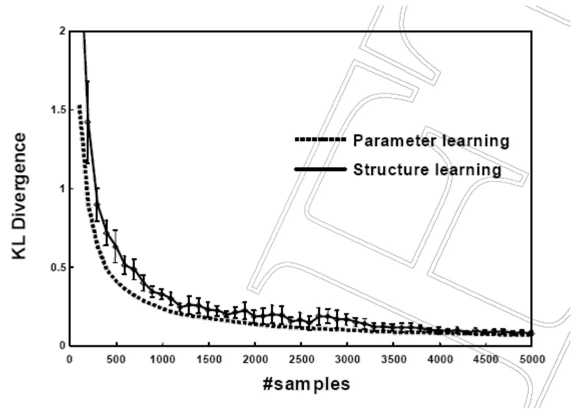
Select using favorite score

Exploit score decomposition in local search



- Add edge and delete edge:
  - Only rescore one family!
- Reverse edge
  - Rescore only two families

Some experiments



Alarm network

Order search versus graph search

- Order search advantages
  - For fixed order, optimal BN – more "global" optimization
  - Space of orders much smaller than space of graphs
- Graph search advantages
  - Not restricted to k parents
    - Especially if exploiting CPD structure, such as CSI
  - Cheaper per iteration
  - Finer moves within a graph

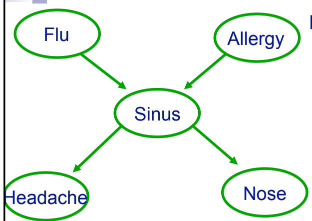
Bayesian model averaging

- So far, we have selected a single structure
- But, if you are really Bayesian, must average over structures
  - Similar to averaging over parameters
    $$\log P(D \mid \mathcal{G}) = \log \int_{\theta_{\mathcal{G}}} P(D \mid \mathcal{G}, \theta_{\mathcal{G}}) P(\theta_{\mathcal{G}} | \mathcal{G}) d\theta_{\mathcal{G}}$$
- Inference for structure averaging is very hard!!!
  - Clever tricks in reading

What you need to know about learning BN structures

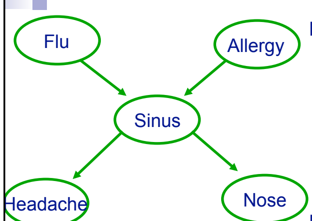
- Decomposable scores
  - Data likelihood
  - Information theoretic interpretation
  - Bayesian
  - BIC approximation
- Priors
  - Structure and parameter assumptions
  - BDe if and only if score equivalence
- Best tree (Chow-Liu)
- Best TAN
- Nearly best k-treewidth (in $O(N^{k+1})$)
- Search techniques
  - Search through orders
  - Search through structures
- Bayesian model averaging

Inference in graphical models: Typical queries 1



- Conditional probabilities
  - Distribution of some var(s). given evidence

Inference in graphical models: Typical queries 2 – Maximization



- Most probable explanation (MPE)
  - Most likely assignment to all hidden vars given evidence
- Maximum a posteriori (MAP)
  - Most likely assignment to some var(s) given evidence

Are MPE and MAP Consistent?

P(S=t)=0.4
P(S=f)=0.6

P(N|S)

- Most probable explanation (MPE)
  - Most likely assignment to all hidden vars given evidence

- Maximum a posteriori (MAP)
  - Most likely assignment to some var(s) given evidence

Complexity of conditional probability queries 1

- How hard is it to compute P(X|**E**=**e**)?

Reduction – 3-SAT

$$(\overline{X}_1 \vee X_2 \vee X_3) \wedge (\overline{X}_2 \vee X_3 \vee X_4) \wedge \ldots$$

Complexity of conditional probability queries 2

- How hard is it to compute P(X|**E**=**e**)?
  - At least NP-hard, but even harder!

Inference is #P-complete, hopeless?

- Exploit structure!
- Inference is hard in general, but easy for many (real-world relevant) BN structures

Complexity for other inference questions

- Probabilistic inference
  - general graphs:
  - poly-trees and low tree-width:
- Approximate probabilistic inference
  - Absolute error:
  - Relative error:
- Most probable explanation (MPE)
  - general graphs:
  - poly-trees and low tree-width:
- Maximum a posteriori (MAP)
  - general graphs:
  - poly-trees and low tree-width:

Inference in BNs hopeless?

- In general, yes!
  - Even approximate!
- In practice
  - Exploit structure
  - Many effective approximation algorithms (some with guarantees)
- For now, we'll talk about exact inference
  - Approximate inference later this semester