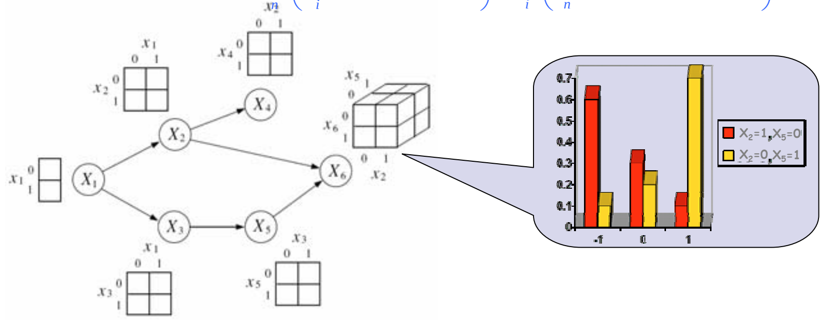
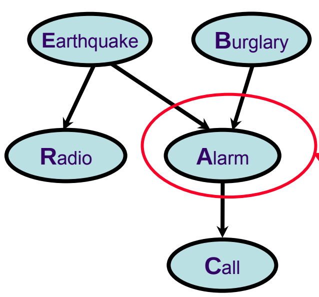
MLE for general BNs

- If we assume the parameters for each CPD are globally independent, and all nodes are fully observed, then the log-likelihood function decomposes into a sum of local terms, one per node:

$$\ell(\theta; D) = \log p(D \mid \theta) = \log \prod_n \left(\prod_i p(x_{n,i} \mid \mathbf{x}_{n,\pi_i}, \theta_i) \right) = \sum_i \left(\sum_n \log p(x_{n,i} \mid \mathbf{x}_{n,\pi_i}, \theta_i) \right)$$



How to define parameter prior?



Factorization: $p(\mathbf{X} = \mathbf{x}) = \prod_{i=1}^{M} p(x_i \mid \mathbf{x}_{\pi_i})$

Local Distributions
defined by, e.g., multinomial parameters:

$$p(x_i^k \mid \mathbf{x}_{\pi_i}^j) = \theta_{x_i^k \mid \mathbf{x}_{\pi_i}^j}$$

$$p(\theta \mid G)\,?$$

Global & Local Parameter Independence

- Global Parameter Independence

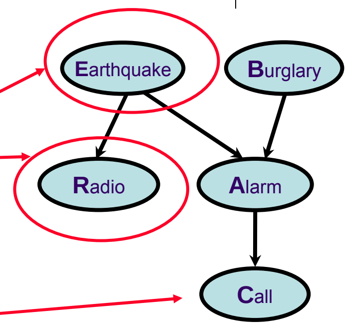
  For <u>every</u> DAG model:

  $$p(\theta \mid G) = \prod_{i=1}^{M} p(\theta_i \mid G)$$

- Local Parameter Independence

  For <u>every</u> node:

  $$p(\theta_i \mid G) = \prod_{j=1}^{q_i} p(\theta_{x_i^k \mid \mathbf{x}_{\pi_i}^j} \mid G)$$



- **The Bayesian posterior**

$$P(\theta \mid D, G) \propto P(D \mid \theta) P(\theta \mid G)$$
$$= \prod_{i,j} p(x_i \mid \mathbf{x}_{\pi_i}^j, \theta_{i,j}) P(\theta_{i,j} \mid G)$$

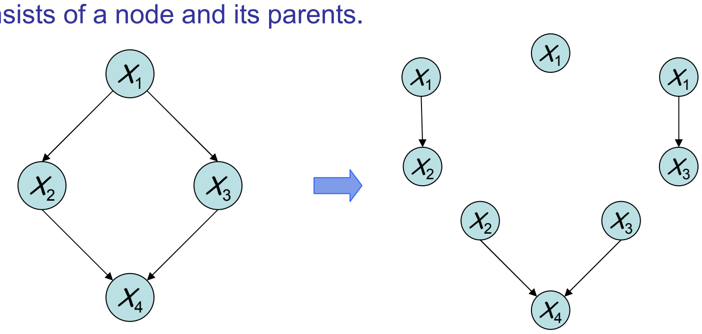
Eric Xing

Example: decomposable likelihood of a directed model

- Consider the distribution defined by the directed acyclic GM:

$$p(x \mid \theta) = p(x_1 \mid \theta_1) p(x_2 \mid x_1, \theta_1) p(x_3 \mid x_1, \theta_3) p(x_4 \mid x_2, x_3, \theta_1)$$

- This is exactly like learning four separate small BNs, each of which consists of a node and its parents.

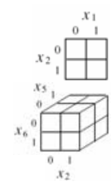


Eric Xing

MLE for BNs with tabular CPDs

- Assume each CPD is represented as a table (multinomial) where

$$\theta_{ijk} \stackrel{\text{def}}{=} p(X_i = j \mid X_{\pi_i} = k)$$



  - Note that in case of multiple parents, $\mathbf{X}_{\pi_i}$ will have a composite state, and the CPD will be a high-dimensional table
  - The sufficient statistics are counts of family configurations

$$n_{ijk} \stackrel{\text{def}}{=} \sum_n x_{n,i}^j x_{n,\pi_i}^k$$

- The log-likelihood is

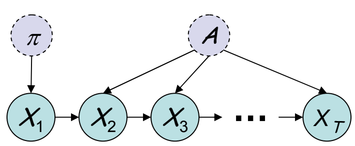
$$\ell(\theta; D) = \log \prod_{i,j,k} \theta_{ijk}^{n_{ijk}} = \sum_{i,j,k} n_{ijk} \log \theta_{ijk}$$

- Using a Lagrange multiplier to enforce $\sum_j \theta_{ijk} = 1$, we get:

$$\theta_{ijk}^{ML} = \frac{n_{ijk}}{\sum_{i,j',k} n_{ij'k}}$$

Eric Xing

Parameter sharing



- Consider a time-invariant (stationary) 1st-order Markov model
  - Initial state probability vector: $\pi_k \stackrel{\text{def}}{=} p(X_1^k = 1)$
  - State transition probability matrix: $A_{ij} \stackrel{\text{def}}{=} p(X_t^j = 1 \mid X_{t-1}^i = 1)$
- The joint: $p(X_{1:T} \mid \theta) = p(x_1 \mid \pi) \prod_{t=2}^{T} \prod_{t=2} p(X_t \mid X_{t-1})$
- The log-likelihood: $\ell(\theta; D) = \sum_n \log p(x_{n,1} \mid \pi) + \sum_n \sum_{t=2}^{T} \log p(x_{n,t} \mid x_{n,t-1}, A)$
- Again, we optimize each parameter separately
  - π is a multinomial frequency vector, and we've seen it before
  - What about A?

Eric Xing

Learning a Markov chain transition matrix

- A is a stochastic matrix: $\sum_j A_{ij} = 1$
- Each row of A is multinomial distribution.
- So **MLE** of A_{ij} is the fraction of transitions from i to j

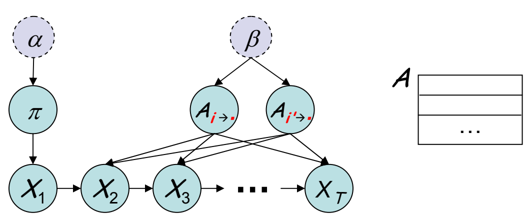
$$A_{ij}^{ML} = \frac{\#(i \to j)}{\#(i \to \bullet)} = \frac{\sum_n \sum_{t=2}^{T} x_{n,t-1}^{i} x_{n,t}^{j}}{\sum_n \sum_{t=2}^{T} x_{n,t-1}^{i}}$$

- Application:
  - if the states X_t represent words, this is called a *bigram language model*
- Sparse data problem:
  - If $i \to j$ did not occur in data, we will have A_{ij} =0, then any futher sequence with word pair $i \to j$ will have zero probability.
  - A standard hack: *backoff smoothing* or *deleted interpolation*

$$\tilde{A}_{i\to\bullet} = \lambda \eta_t + (1-\lambda) A_{i\to\bullet}^{ML}$$

Bayesian language model

- Global and local parameter independence



  - The posterior of $A_{i\to\bullet}$ and $A_{i'\to\bullet}$ is factorized despite v-structure on X_t, because X_{t-1} acts like a multiplexer
  - Assign a Dirichlet prior β_i to each row of the transition matrix:

$$A_{ij}^{Bayes} \overset{\text{def}}{=} p(j \mid i, D, \beta_i) = \frac{\#(i \to j) + \beta_{i,k}}{\#(i \to \bullet) + |\beta_i|} = \lambda_i \beta'_{i,k} + (1-\lambda_i) A_{ij}^{ML}, \text{ where } \lambda_i = \frac{|\beta_i|}{|\beta_i| + \#(i \to \bullet)}$$

  - We could consider more realistic priors, e.g., mixtures of Dirichlets to account for types of words (adjectives, verbs, etc.)

Example: HMM: two scenarios

- **Supervised learning**: estimation when the "right answer" is known
  - **Examples:**

    GIVEN: a genomic region x = $x_1 \ldots x_{1,000,000}$ where we have good (experimental) annotations of the CpG islands

    GIVEN: the casino player allows us to observe him one evening, as he changes dice and produces 10,000 rolls

- **Unsupervised learning**: estimation when the "right answer" is unknown
  - **Examples:**

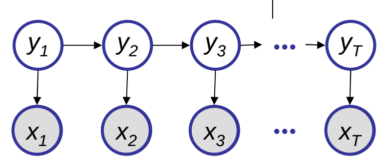
    GIVEN: the porcupine genome; we don't know how frequent are the CpG islands there, neither do we know their composition

    GIVEN: 10,000 rolls of the casino player, but we don't see when he changes dice

- **QUESTION:** Update the parameters θ of the model to maximize $P(x|\theta)$ --- Maximal likelihood (ML) estimation

Eric Xing

Recall definition of HMM



- Transition probabilities between any two states

$$p(y_t^j = 1 \mid y_{t-1}^i = 1) = a_{i,j},$$

or $$p(y_t \mid y_{t-1}^i = 1) \sim \mathrm{Multinomial}\left(a_{i,1}, a_{i,2}, \ldots, a_{i,M}\right), \forall i \in \mathbb{I}.$$

- Start probabilities

$$p(y_1) \sim \mathrm{Multinomial}\left(\pi_1, \pi_2, \ldots, \pi_M\right).$$

- Emission probabilities associated with each state

$$p(x_t \mid y_t^i = 1) \sim \mathrm{Multinomial}\left(b_{i,1}, b_{i,2}, \ldots, b_{i,K}\right), \forall i \in \mathbb{I}.$$

or in general: $$p(x_t \mid y_t^i = 1) \sim \mathrm{f}\left(\cdot \mid \theta_i\right), \forall i \in \mathbb{I}.$$

Eric Xing

Supervised ML estimation

- Given $x = x_1 \ldots x_N$ for which the true state path $y = y_1 \ldots y_N$ is known,
  - **Define:**

    A_{ij} = # times state transition $i \to j$ occurs in $\mathbf{y}$

    B_{ik} = # times state i in $\mathbf{y}$ emits k in $\mathbf{x}$

  - **We can show that the maximum likelihood parameters θ are:**

$$a_{ij}^{ML} = \frac{\#(i \to j)}{\#(i \to \bullet)} = \frac{\sum_n \sum_{t=2}^T y_{n,t-1}^i y_{n,t}^j}{\sum_n \sum_{t=2}^T y_{n,t-1}^i} = \frac{A_{ij}}{\sum_{j'} A_{ij'}}$$

$$b_{ik}^{ML} = \frac{\#(i \to k)}{\#(i \to \bullet)} = \frac{\sum_n \sum_{t=1}^T y_{n,t}^i x_{n,t}^k}{\sum_n \sum_{t=1}^T y_{n,t}^i} = \frac{B_{ik}}{\sum_{k'} B_{ik'}}$$

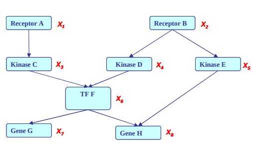
  - **What if x is continuous? We can treat $\{(x_{n,t}, y_{n,t}) : t = 1:T, n = 1:N\}$ as $N \times T$ observations of, e.g., a Gaussian, and apply learning rules for Gaussian …**

School of Computer Science
Carnegie Mellon

Learning BN Structure

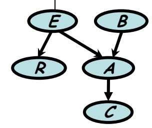
Probabilistic Graphical Models (10-708)

Lecture 10, Oct 17, 2007

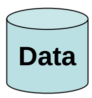


Eric Xing

Reading: KF-Chap. 16



ML Structural Learning for completely observed GMs

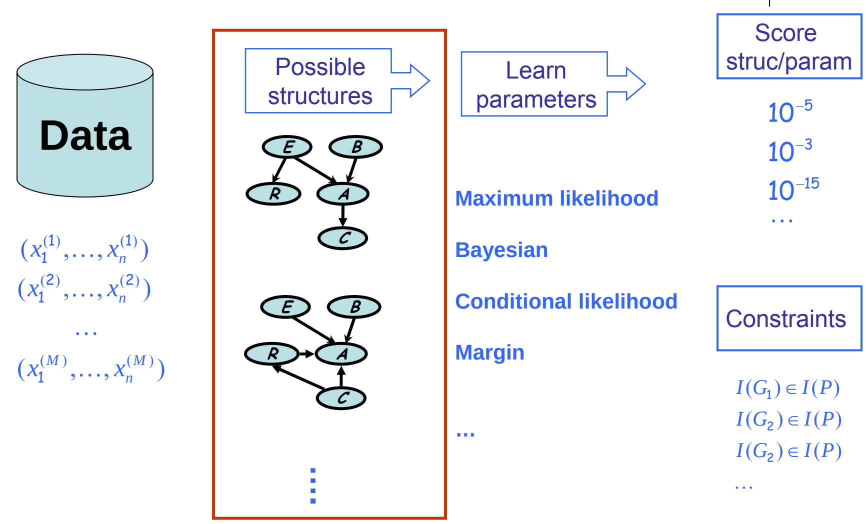


$$(x_1^{(1)},\ldots,x_n^{(1)})$$
$$(x_1^{(2)},\ldots,x_n^{(2)})$$
$$\ldots$$
$$(x_1^{(M)},\ldots,x_n^{(M)})$$

Where are we now on the map?

- Graphical models
  - Bayesian networks
  - Undirected models
  - Conditional independence statements + factorization law of joint dist.
- Exact inference in GMs
  - Variable elimination <=> Graph elimination
  - Sum-product on tree, factor tree, clique tree
  - Very fast for models with low tree-width
- Learning GMs
  - Given structure, estimate parameters
    - Maximum likelihood estimation (just counts for BNs)
    - Bayesian learning
    - MAP for Bayesian learning
- What about learning structure?

Learning the structure of a BN



Eric Xing

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.

Eric Xing

Recall P-Map

- **Defn (3.4.3):** We say that a graph object G is a *perfect map (P-map)* for a set of independencies I if we have that I(G) = I. We say that G is a perfect map for P if I(G) = I(P).
  - Not all P has a perfect map as DAG!
  - The P-map of a distribution *is* unique up to I-equivalence between networks. That is, a distribution P can have many P-maps, but all of them are I-equivalent.
  - The P-DAG algorithm
- Constraint-based approach:
  - Key question: Independence test

Eric Xing

Constraint-based approach: Independence tests

- Statistically difficult task!
- Intuitive approach:
  - Mutual information

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

  - Mutual information and independence:
    - X_i and X_j are independent if and only if $I(X_i,X_j)=0$
- Conditional mutual information:

Eric Xing

Empirical independence tests

- Using the data *D*
  - Empirical distribution:
    $$\hat{P}(x_i, x_j) = \frac{\mathrm{count}(x_i, x_j)}{M}$$
  - Mutual information:
    $$\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \log \hat{P}(x_i, x_j) \frac{\hat{P}(x_i, x_j)}{\hat{P}(x_i)\hat{P}(x_j)}$$
  - Similarly for conditional MI
- More generally, use learning PDAG algorithm:
  - When algorithm asks: (X⊥Y|**U**)?
  - Must check if statistically-significant
  - Choosing *t*
  - See reading…

Eric Xing

Score-based approach:

- Desirable properties of a scoring function
  - **Consistency**: i.e., if the data is generated by G^*, then G and all I-equivalent models maximize the score.
  - Decomposability:
    $$\mathrm{Score}(G \mid D) = \sum_i \mathrm{FamScore}(D(X_i \mid X_{\pi_i})$$
    which makes it cheap to compare score of G and G' if they only differ in a small number of families.
- Bayesian score (evidence), likelihood, and penalized likelihood (BIC) are all decomposable and consistent.

Eric Xing

Maximizing the score

- Consider the family of DAGs G_d with maximum fan-in (number of parents) equal to d.
- **Thm**: It is NP-hard to find

$$G^* = \arg\max_{G \in G_d} \text{Score}(G \mid D)$$

  for any d ≥ 2.

- In general, we need to use heuristic local search
  - For d ≤ 1 (i.e., trees), we can solve the problem in $O(n^2)$ time using max spanning tree (forthcoming)
  - If we know the ordering of the nodes, we can solve the problem in $O(d\binom{n}{d})$ time

Eric Xing

Information Theoretic Interpretation of ML

$$\begin{aligned}
\ell(\theta_G, G; D) &= \log p(D \mid \theta_G, G) \\
&= \log \prod_n \left(\prod_i p(x_{n,i} \mid \mathbf{x}_{n,\pi_i(G)}, \theta_{i|\pi_i(G)}) \right) \\
&= \sum_i \left(\sum_n \log p(x_{n,i} \mid \mathbf{x}_{n,\pi_i(G)}, \theta_{i|\pi_i(G)}) \right) \\
&= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \frac{count(x_i, \mathbf{x}_{\pi_i(G)})}{M} \log p(x_i \mid \mathbf{x}_{\pi_i(G)}, \theta_{i|\pi_i(G)}) \right) \\
&= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \hat{p}(x_i, \mathbf{x}_{\pi_i(G)}) \log p(x_i \mid \mathbf{x}_{\pi_i(G)}, \theta_{i|\pi_i(G)}) \right)
\end{aligned}$$

From sum over data points to sum over count of variable states

Eric Xing

Information Theoretic Interpretation of ML (con'd)

$$\begin{aligned}
\ell(\theta_G, G; D) &= \log \hat{p}(D \mid \theta_G, G) \\
&= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \hat{p}(x_i, \mathbf{x}_{\pi_i(G)}) \log \hat{p}(x_i \mid \mathbf{x}_{\pi_i(G)}, \theta_{i|\pi_i(G)}) \right) \\
&= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \hat{p}(x_i, \mathbf{x}_{\pi_i(G)}) \log \frac{\hat{p}(x_i, \mathbf{x}_{\pi_i(G)}, \theta_{i|\pi_i(G)})}{\hat{p}(\mathbf{x}_{\pi_i(G)})} \frac{\hat{p}(x_i)}{\hat{p}(x_i)} \right) \\
&= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \hat{p}(x_i, \mathbf{x}_{\pi_i(G)}) \log \frac{\hat{p}(x_i, \mathbf{x}_{\pi_i(G)}, \theta_{i|\pi_i(G)})}{\hat{p}(\mathbf{x}_{\pi_i(G)}) \hat{p}(x_i)} \right) - M \sum_i \left(\sum_{x_i} \hat{p}(x_i) \log p(x_i) \right) \\
&= M \sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)}) - M \sum_i \hat{H}(x_i)
\end{aligned}$$

Decomposable score and a function of the graph structure

Decomposable Score

- Log data likelihood

$$\begin{aligned}
\ell(\theta_G, G; D) &= \log \hat{p}(D \mid \theta_G, G) \\
&= M \sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)}) - M \sum_i \hat{H}(x_i)
\end{aligned}$$

- Decomposable score:
  - Decomposes over families in BN (node and its parents)
  - Will lead to significant computational efficiency!!!
  - The score function:

$$\mathrm{Score}(G \mid D) = \sum_i \mathrm{FamScore}(D(X_i \mid X_{\pi_i})$$

- Search space:

Structural Search

- How many graphs over n nodes? $O(2^{n^2})$
- How many trees over n nodes? $O(2^{n \log n})$
- But it turns out that we can find exact solution of an optimal tree (under MLE)!
  - Trick: in a tree each node has only one parent!
  - Chow-liu algorithm

Scoring a tree 1: equivalent trees

Scoring a tree 2: similar trees

Eric Xing

Chow-Liu tree learning algorithm

- Objection function:

$$\ell(\theta_G, G; D) = \log \hat{p}(D \mid \theta_G, G) = M\sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)}) - M\sum_i \hat{H}(x_i) \quad \Rightarrow \quad \boxed{C(G) = M\sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)})}$$

- Chow-Liu:
  - For each pair of variable x_i and x_j
    - Compute empirical distribution: $\hat{p}(X_i, X_j) = \frac{count(x_i, x_j)}{M}$
    - 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 with node $x_1, \ldots, x_n$
    - Edge (I,j) gets weight $\hat{I}(X_i, X_j)$

Eric Xing

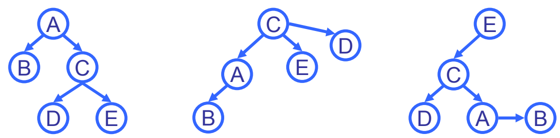
Chow-Liu algorithm (con'd)

- Objection function:

$$\begin{aligned}\ell(\theta_G, G; D) &= \log \hat{p}(D \mid \theta_G, G) \\ &= M\sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)}) - M\sum_i \hat{H}(x_i)\end{aligned} \quad \Rightarrow \quad \boxed{C(G) = M\sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)})}$$

- Chow-Liu:

  Optimal tree BN
  - Compute maximum weight spanning tree
  - Direction in BN: pick any node as root, do breadth-first-search to define directions
  - I-equivalence:



$$C(G) = I(A,B) + I(A,C) + I(C,D) + I(C,E)$$

Extensions of Chow-Liu

- Tree augmented naïve Bayes(TAN) [Friedman et al. '97]
  - Naïve Bayes model overcounts, because correlation between features not considered
  - Tree-augmented feature list

- Same as Chow-Liu, but score edges w

$$\hat{p}(X_i, X_j \mid C) = \frac{count(x_i, x_j \mid C)}{M}$$

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

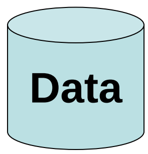
Structure Learning for general graphs

- Theorem:
  - The problem of learning a BN structure with at most *d* parents is NP-hard for any (fixed) *d*≥2
- Most structure learning approaches use heuristics
  - Exploit score decomposition
  - Two heuristics that exploit decomposition in different ways
    - Greedy search through space of node-orders
    - Local search of graph structures

Known order (K2 algorithm)

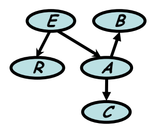
- Suppose we a total ordering of the nodes $X_1 \prec X_2 \prec \cdots \prec X_n$ and want to find a DAG consistent with this with maximum score.
  - The choice of parents for X_i, from $\mathrm{Pa}_i\{X_1, \ldots, X_{i-1}\}$, is independent of the choice for X_j: since we obey the ordering, we cannot create a cycle.
  - Hence we can pick the best set of parents for each node independently.
  - For X_i, we need to search all $\binom{i-1}{d}$ subsets of size up to d for the set which maximizes FamScore.
  - We can use greedy techniques for this, c.f., learning a decision tree.
- What if order isn't known
  - Search in the space of orderings, then conditioned on , pick best graph using K2
  - Search in the space of DAGs.

Learn BN structure using local search



$$(x_1^{(1)},\ldots,x_n^{(1)})$$
$$(x_1^{(2)},\ldots,x_n^{(2)})$$
$$\ldots$$
$$(x_1^{(M)},\ldots,x_n^{(M)})$$

Starting from Chow-Liu tree → **Local search** → **Select using favorite score**

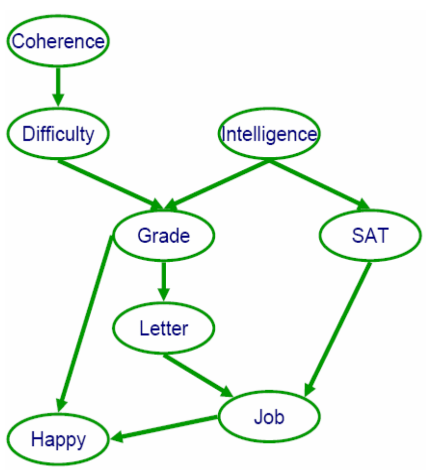


Possible moves:
Only if acyclic!!!

- **Add edge**
- **Delete edge**
- **Invert edge**

10^{-5}
10^{-3}
10^{-15}
$\cdots$

Exploit score decomposition in local search



- Add edge and delete edge
  - Only rescore one family
- Reverse edge
  - Rescore only two families
- Simplest search algorithm: greedy hill climbing.

Local maxima

- Greedy hill climbing will stop when it reaches a local maximum or a plateau (a set of neighboring networks that have the same score).
- Unfortunately, plateaus are common, since equivalence classes form contiguous regions of search space (thm 14.4.4), and such classes can be exponentially large.
- Solutions:
  - Random restarts
  - TABU search (prevent the algorithm from undoing an operator applied in the last L steps, thereby forcing it to explore new terrain).
  - Data perturbation (dynamic local search): reweight the data and take step.
  - Simulated annealing: if $\delta(o) > 0$, take move, else accept with probability $e^{\delta(o)/t}$, where t is the temperature. Slow!

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

Identifiability

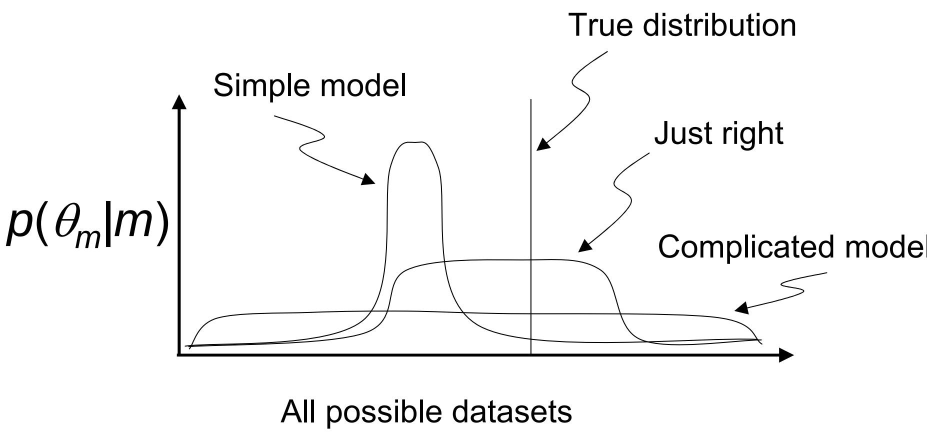
- DAGs are I-equivalent if they encode the same set of conditional independencies
  - e.g., X → Y → Z and X ← Y ← Z are indistinguishable given just observational data.
- However, X → Y ← Z has a v-structure, which has a unique statistical signature. Hence some arc directions can be inferred from passive observation.
- The set of I-equivalent DAGs can be represented by a PDAG (partially directed acyclic graph).
- Distinguishing between members of an equivalence class requires interventions/ experiments.

ML score overfits!

$$\ell(\theta_G, G; D) = \log \hat{p}(D \mid \theta_G, G)$$
$$= M \sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)}) - M \sum_i \hat{H}(x_i)$$

- Information never hurts
- Adding a parent always increases your score!

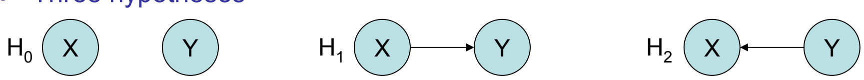
Occam's Razor



Eric Xing

Model selection

- Three hypotheses



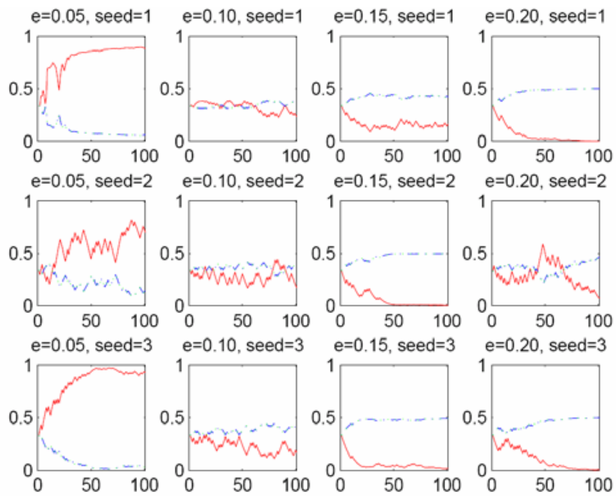
  - $P(X=1)=0.5$ and $P(Y=1|X=0)=0.5-\varepsilon,\ P(Y=1|X=1)=0.5+\varepsilon$
  - As we increase , we increase the dependence of Y on X
  - X ← Y and X → Y are I-equivalent (have the same likelihood)
- Suppose we use a uniform Dirichlet prior for each node in each graph, with equivalent pseudo-counts (K2-prior):

$$P(\theta_X | H_1) = Dir(\alpha, \alpha) \qquad P(\theta_{X|Y=i} | H_2) = Dir(\alpha, \alpha)$$

  - In H_1, the equivalent sample size for X is 2, but in H_2 it is 4 (since two conditioning contexts). Hence the posterior probabilities are different.
- Under which H the P(H|D) is higher?

Eric Xing

Model selection (model posterior)



red = H0 (independence), blue/green = H1/H2 (dependence)

Bayesian model selection

- Why is $P(H_0|D)$ higher when then dependence on X and Y is weak (small)?
  - It is not because the prior P(Hi) explicitly favors simpler models (although this is possible).
  - It because the evidence $P(D)=\int dw P(D|w)P(w)$ automatically penalizes complex models.
- "Occam's razor" says "If two models are equally predictive, prefer the simpler one".
  - This is an automatic consequence of using Bayesian model selection.
  - Maximum likelihood would always pick the most complex model, since it has more parameters, and hence can fit the training data better.
- Good test for a learning algorithm: feed it random noise, see if it "discovers" structure!

Global & Local Parameter Independence

- Global Parameter Independence

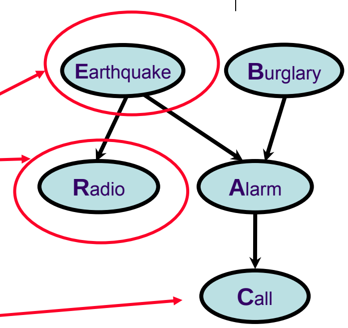
  For <u>every</u> DAG model**:**

  $$p(\theta \mid G) = \prod_{i=1}^{M} p(\theta_i \mid G)$$

- Local Parameter Independence

  For <u>every</u> node**:**

  $$p(\theta_i \mid G) = \prod_{j=1}^{q_i} p(\theta_{x_i^k \mid \mathbf{x}_{\pi_i}^j} \mid G)$$



- **The Bayesian score**

$$\begin{aligned}
\log P(G|D) &= \log P(G) + \log \int_\theta P(D \mid \theta) P(\theta \mid G) d\theta + C \\
&= \log P(G) + \sum_{i,j} \int_{\theta_{i,j}} p(x_i \mid \mathbf{x}_{\pi_i}^j, \theta_{i,j}) P(\theta_{i,j} \mid G) d\theta_{i,j} + C \\
&= \log P(G) + C + \sum_i score(x_i, \mathbf{x}_{\pi_i})
\end{aligned}$$

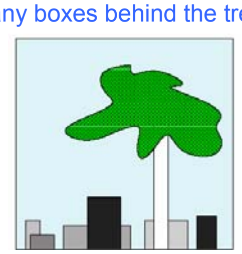
Eric Xing

Selection criteria

- BIC (Bayesian Information Criterion):

  $$\log P(D) \approx \log P(D \mid \hat{\theta}_{ML}) - \frac{d}{2} \log N$$

  - Quiz: How many boxes behind the tree?



- Other criteria:
  - AIC (Akaike Information Criterion):
  - Minimum description length

Eric Xing

Consistency of BIC and Bayesian scores

- A scoring function is **consistent** if, for true model *G**, as $m \to \infty$, 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?

Eric Xing

Choice of Priors

- 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, $P(\theta_i|\mathbf{Pa}_{X_i}) = \text{Dirichlet}(\alpha, \ldots, \alpha)$
    - K2 is "inconsistent"

Eric Xing

BDe prior

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

Eric Xing