Probabilistic Graphical Models

Lecture 4 – Learning Bayesian Networks

CS/CNS/EE 155

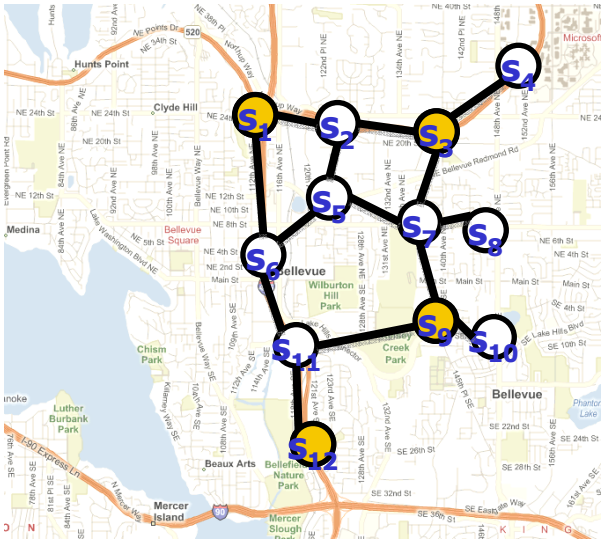
Andreas Krause

Announcements

- Another TA: Hongchao Zhou
- Please fill out the questionnaire about recitations
- Homework 1 out. Due in class Wed Oct 21
- Project proposals due Monday Oct 19

Representing the world using BNs

represent

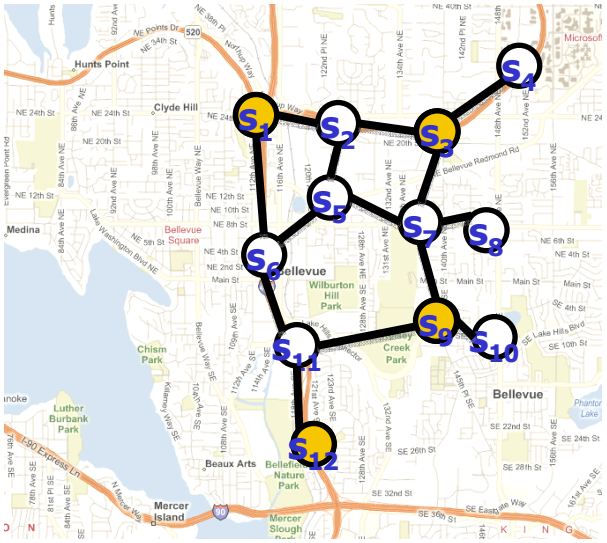


True distribution P'
with cond. ind. I(P')

Bayes net (G,P)
with I(P)

- Want to make sure that I(P) $\subseteq$ I(P')
- Need to understand CI properties of BN (G,P)

Factorization Theorem



$I_{loc}(G) \subseteq I(P)$

G is an **I-map** of P
(independence map)

True distribution P
can be represented exactly as
Bayesian network (G,P)

$$P(X_1, ..., X_n) = \prod_i P(X_i \mid \mathbf{Pa}_{X_i})$$

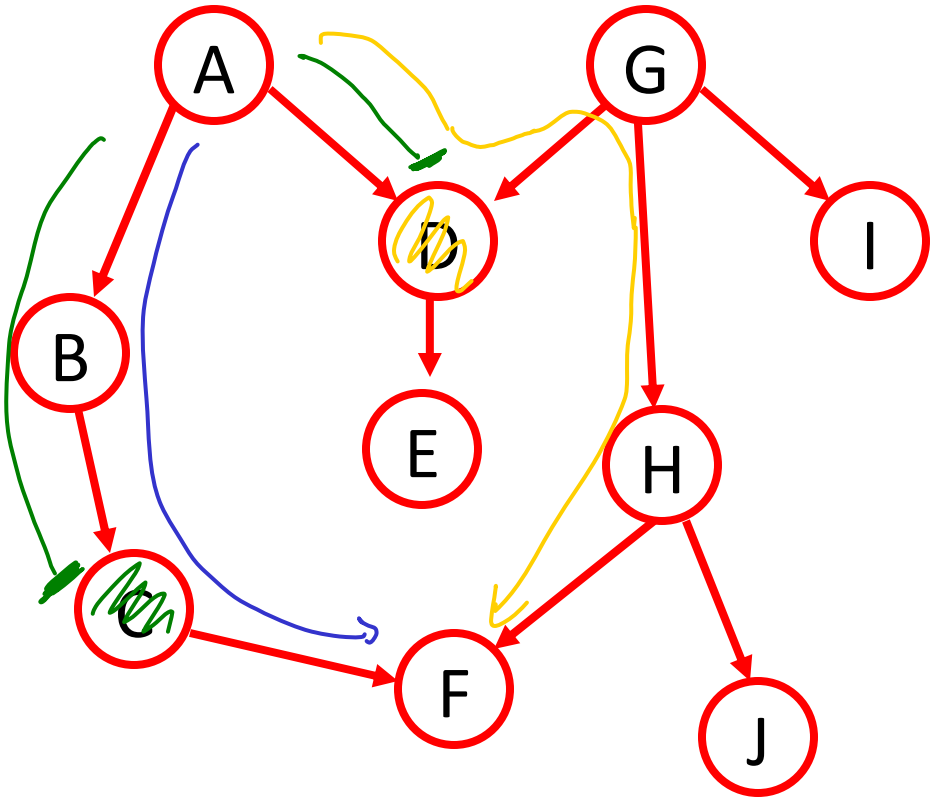
Additional conditional independencies

- BN specifies joint distribution through conditional parameterization that satisfies Local Markov Property
  $I_{loc}(G) = \{(X_i \perp \text{Nondescendants}_{X_i} \mid Pa_{X_i})\}$
- But we also talked about additional properties of CI
  - Weak Union, Intersection, Contraction, …
- Which additional CI does a particular BN specify?
  - All CI that can be derived through algebraic operations

➔ proving CI is very cumbersome!!

Is there an easy way to find all independences of a BN just by looking at its graph??

Examples



A ⊥ F ✓

A ⊥ F | C ✓

A ⊥ F | C, D ×

Active trails

- An undirected path in BN structure G is called **active trail** for observed variables $\mathbf{O} \subseteq \{X_1,\ldots,X_n\}$, if for every consecutive triple of vars X,Y,Z on the path
  - X → Y → Z and Y is unobserved ($Y \notin \mathbf{O}$)
  - X ← Y ← Z and Y is unobserved ($Y \notin \mathbf{O}$)
  - X ← Y → Z and Y is unobserved ($Y \notin \mathbf{O}$)
  - X → Y ← Z and Y **or any of Y's descendants** is observed

- Any variables X_i and X_j for which $\nexists$ active trail for observations **O** are called d-separated by **O**
  We write **d-sep(X_i;X_j | O)**
- Sets **A** and **B** are d-separated given **O** if d-sep(X,Y |**O**) for all X∈**A**, Y∈**B**. Write **d-sep(A; B | O)**

Soundness of d-separation

- Have seen: P factorizes according to G $\Leftrightarrow$ $I_{loc}(G) \subseteq I(P)$
- Define $I(G) = \{(X \perp Y \mid Z): \text{d-sep}_G(X;Y \mid Z)\}$

- **Theorem**: Soundness of d-separation

  P factorizes over G $\rightarrow$ $I(G) \subseteq I(P)$

- Hence, d-separation captures only true independences

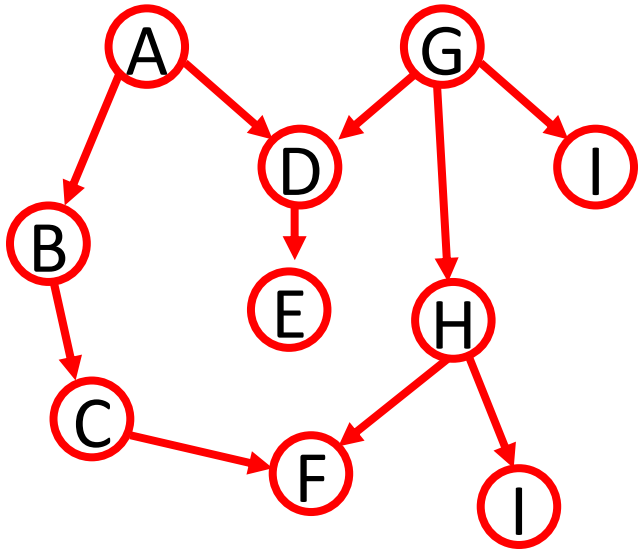
- How about $I(G) = I(P)$?

Completeness of d-separation

- Theorem: For "almost all" distributions P that factorize over G it holds that **I(G) = I(P)**
  - "almost all": except for a set of distributions with measure 0, assuming only that no finite set of distributions has measure > 0

Algorithm for d-separation

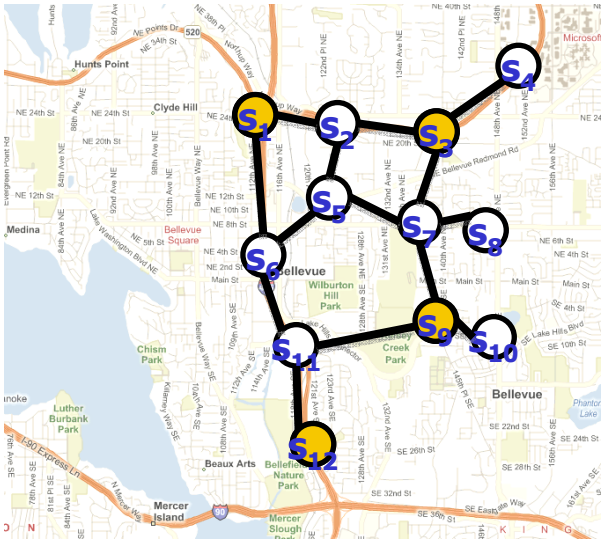
- How can we check if X ⊥ Y | **Z**?
  - Idea: Check every possible path connecting X and Y and verify conditions
  - Exponentially many paths!!! ☹



- Linear time algorithm:
  Find all nodes reachable from X
  - 1. Mark **Z** and its ancestors
  - 2. Do breadth-first search starting from X; stop if path is blocked
  - Have to be careful with implementation details (see reading)

Representing the world using BNs

represent



True distribution P'
with cond. ind. I(P')

Bayes net (G,P)
with I(P)

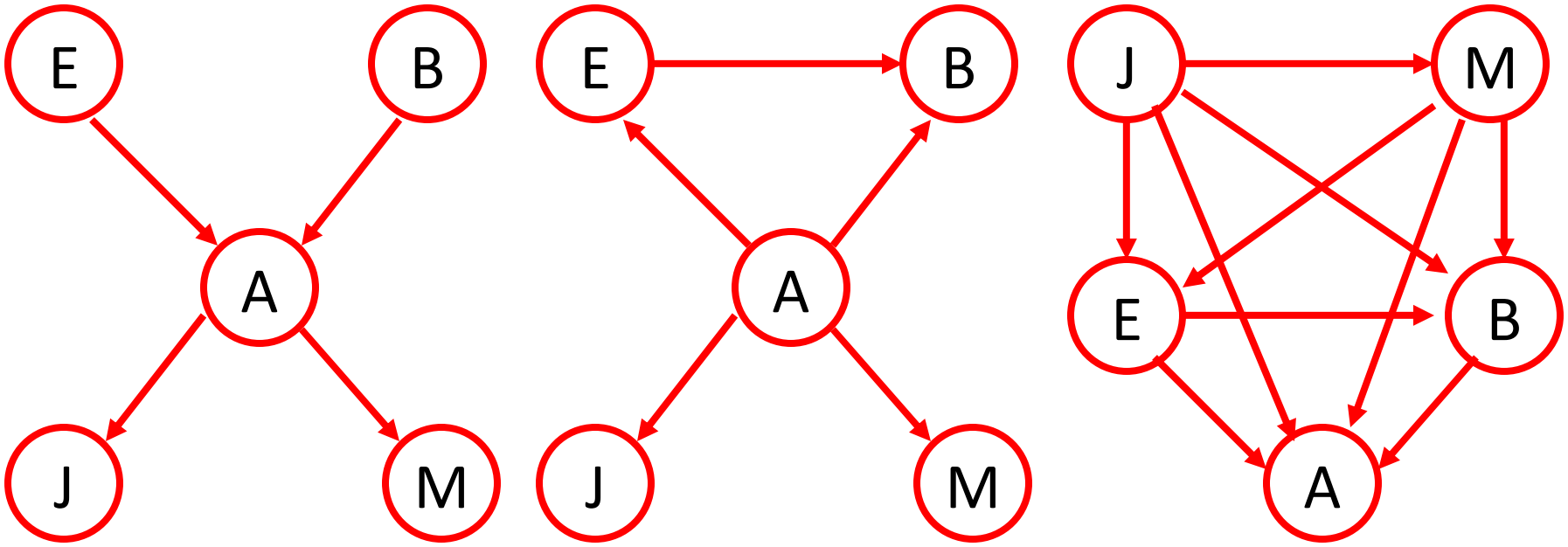
- Want to make sure that I(P) $\subseteq$ I(P')
- Ideally: I(P) = I(P')
- Want BN that **exactly** captures independencies in P'!

Minimal I-map

- Graph G is called minimal I-map if it's an I-map, and if any edge is deleted → no longer I-map.

Uniqueness of Minimal I-maps

- Is the minimal I-Map unique?



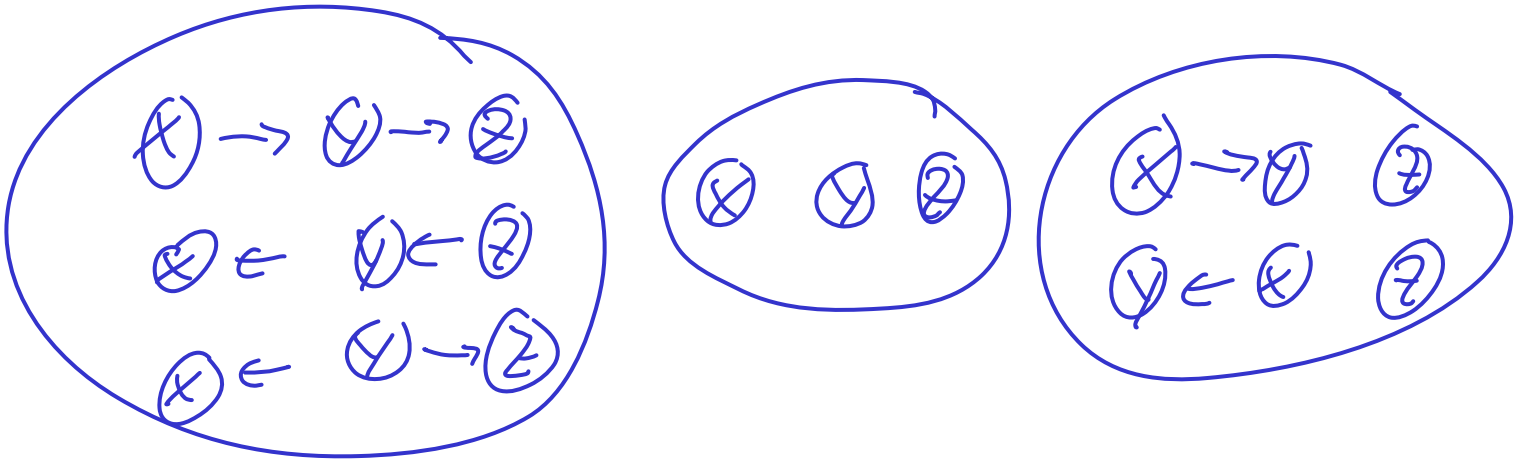
Perfect maps

- Minimal I-maps are easy to find, but can contain many unnecessary dependencies.

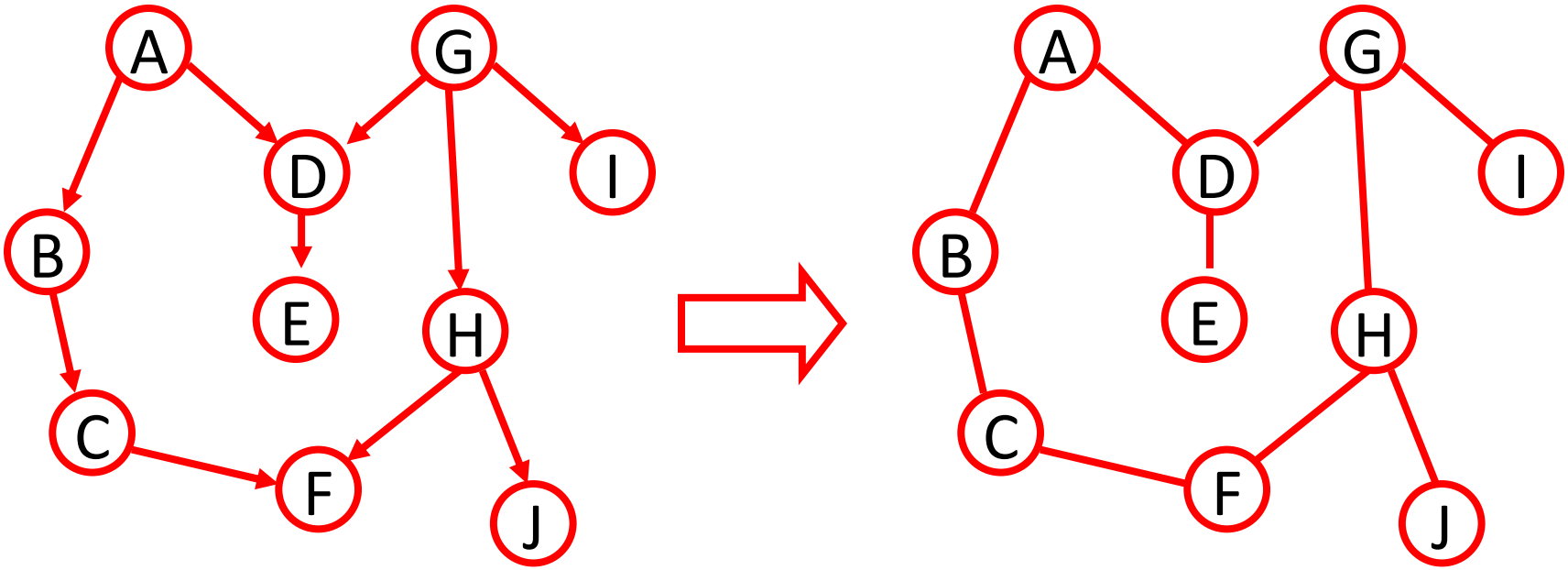
- A BN structure G is called **P-map** (perfect map) for distribution P if **I(G) = I(P)**
- Does every distribution P have a P-map?

I-Equivalence

- Two graphs G, G' are called I-equivalent if I(G) = I(G')
- I-equivalence partitions graphs into equivalence classes



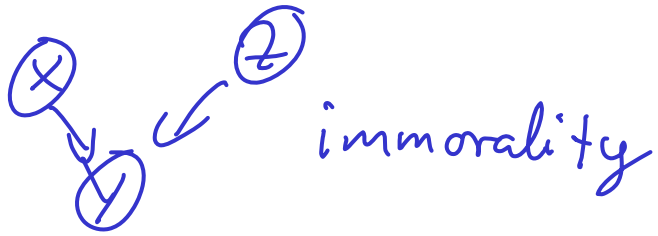
Skeletons of BNs



- I-equivalent BNs must have same skeleton

Immoralities and I-equivalence

- A V-structure X → Y ← Z is called **immoral** if there is no edge between X and Z ("unmarried parents")



- **Theorem**: I(G) = I(G') ⇔ G and G' have the same skeleton and the same immoralities.

Today: Learning BN from data

- Want P-map if one exists
- Need to find
  - Skeleton
  - Immoralities

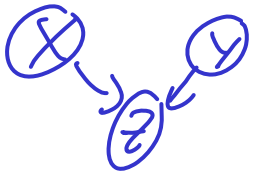
Identifying the skeleton

- When is there an edge between X and Y?

If $\neg(X \perp Y)$? X

: $\neg(X \perp Y)$ but no edge

If $\neg(X \perp Y \mid \text{everything else})$



- When is there no edge between X and Y?

$\Leftrightarrow \exists U \subset \{X_1 \dots X_n\} \setminus \{X, Y\} : X \perp Y \mid U$

(Local Markov assumption)

Algorithm for identifying the skeleton

Set G to be complete graph

For every pair $X, Y \in G$

For every $U \subset \{X_1 \ldots X_n\} \setminus \{X, Y\}$

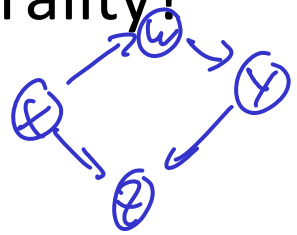
If $X \perp Y \mid U$ then remove edge between X, Y

Not necessarily practical ...

Identifying immoralities

- When is X – Z – Y an immorality?

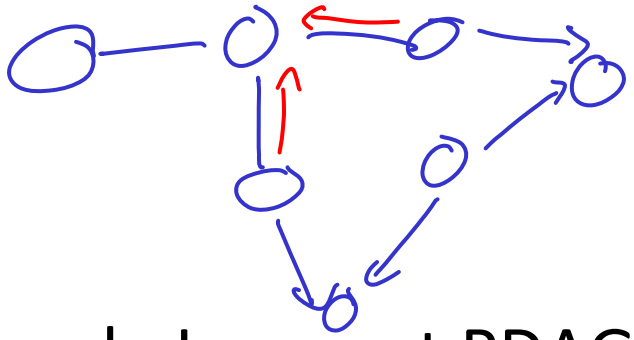
$\neg (X \perp Y \mid Z)$ ✗



- Immoral ⇔ for all **U**, $Z \in \mathbf{U}$: $\neg (X \perp Y \mid \mathbf{U})$

From skeleton & immoralities to BN Structures

- Represent I-equivalence class as partially-directed acyclic graph (PDAG)



- How do I convert PDAG into BN?

Have to be careful when orienting edges to avoid cycles

Poly time algo in reading

Testing independence

- So far, assumed that we know I(P'), i.e., all independencies associated with true dist. P'
- Often, access to P' only through sample data (e.g., sensor measurements, etc.)

- Given vars X, Y, **Z**, want to test whether X ⊥ Y | **Z**

$X \perp Y \mid Z \iff I(X;Y \mid Z) = 0$

$$\sum_{z} \sum_{xy} p(x,y,z) \log \frac{p(x,y \mid z)}{p(x \mid z)\, p(y \mid z)}$$

Estimate $p(x,y,z)$ from data $\Rightarrow \hat{p}(x,y,z)$

Compute $\hat{I}(X;Y \mid Z) = \sum_{xyz} \hat{p}(x,y,z) \log \frac{\hat{p}(x,y \mid z)}{\hat{p}(x \mid z)\, \hat{p}(y \mid z)}$

Test whether $\hat{I}(X;Y \mid Z) < \varepsilon$

Next topic: Learning BN from Data

- Two main parts:
  - Learning structure (conditional independencies)
  - Learning parameters (CPDs)

Parameter learning

- Suppose X is Bernoulli distribution (coin flip) with unknown parameter P(X=H) =θ.
- Given training data D = {$x^{(1)}$,…,$x^{(m)}$} (e.g., H H T H H H T T H T H H H..) how do we estimate θ?

$$P(D \mid \theta) = \theta^{n_H} \cdot (1-\theta)^{n_T}$$

n_H : # of H in D

n_T : — " — T —

$$\hat{\theta} = \arg\max_\theta P(D \mid \theta) = \arg\max_\theta \log P(D \mid \theta)$$
$$= \arg\max_\theta n_H \log\theta + n_T \log(1-\theta)$$

want $\frac{d \log P(D \mid \theta)}{d\theta} \stackrel{!}{=} 0$.

Maximum Likelihood Estimation

- **Given**: data set D
- **Hypothesis**: data generated i.i.d. from binomial distribution with P(X = H) = θ
- Optimize for θ which makes D most likely:

Solving the optimization problem

$$\log P(D|\theta) = n_H \log\theta + n_T \log(1-\theta)$$

$$\frac{d}{d\theta}\left(\log P(D|\theta)\right) = \frac{n_H}{\theta} - \frac{n_T}{1-\theta} \stackrel{!}{=} 0$$

$$n_H(1-\theta) = n_T\theta$$

$$n_H = \theta(n_T + n_H)$$

$$\Rightarrow \hat{\theta} = \frac{n_H}{n_T + n_H} = \frac{n_H}{n}$$

$$= \frac{\text{Count}(X=H)}{n}$$

Learning general BNs

	Known structure	Unknown structure
Fully observable	Easy :)	hard 2.
Missing data	hard 3. (EM)	very hard (lol)

Estimating CPDs

- Given data D = $\{(x_1,y_1),\ldots,(x_n,y_n)\}$ of samples from X,Y, want to estimate P(X | Y)

$$P(X = x \mid Y = y) = \theta_{x|y}$$

$$\hat{\theta}_{x|y} \overset{MLE}{=} \frac{Count(x,y)}{Count(y)}$$

$$\hat{P}(x,y) = \frac{Count(x,y)}{m}$$

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

$$\hat{\theta}_{x|y} = \frac{\hat{P}(x,y)}{\hat{P}(y)}$$

MLE for Bayes nets

$$\log P(D \mid \theta) = \log \prod_l \prod_i P(x_i^{(l)} \mid pa_i^{(l)}; \theta)$$

$$= \sum_l \sum_i \log P(x_i^{(l)} \mid pa_i^{(l)}; \theta_{x_i \mid pa_i})$$

Parameters independent

$$\frac{\partial}{\partial \theta_{x_i \mid pa_i}} \log P(D \mid \theta) = \sum_j \sum_l \frac{\partial}{\partial \theta_{x_i \mid pa_i}} \log P(x_j^{(l)} \mid pa_j^{(l)}; \theta_{x_j \mid pa_j})$$

$$= \sum_l \frac{\partial}{\partial \theta_{x_i \mid pa_i}} \log P(x_i^{(l)} \mid pa_i, \theta_{x_i \mid pa_i}) \stackrel{!}{=} 0$$

Problem breaks down into independent subproblems

Learn every CPD independent of others

Algorithm for BN MLE

Given BN structure G

For each variable X_i

learn $\hat{\theta}_{x_i|pa_i} = \frac{Count(x_i, pa_i)}{Count(pa_i)}$

$\Rightarrow$ globally maximum likelihood estimate for fixed structure G

Learning general BNs

	Known structure	Unknown structure
Fully observable	Easy! ☺ Get CPDs by counting (for MLE)	**???**
Missing data	Hard (EM)	Very hard (later)

Structure learning

- Two main classes of approaches:

- Constraint based
  - Search for P-map (if one exists):
  - Identify PDAG
  - Turn PDAG into BN (using algorithm in reading)
  - **Key problem**: Perform independence tests
- Optimization based ← coming up!
  - Define scoring function (e.g., likelihood of data)
  - Think about structure as parameters
  - More common; can solve simple cases exactly

MLE for structure learning

- For fixed structure, can compute likelihood of data

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) = \sum_{\ell} \sum_{i} \log P(X_i = x_i^{(\ell)} \mid \mathbf{Pa}_i = \mathbf{pa}_i^{(\ell)}, \theta_{\mathcal{G}}, \mathcal{G})$$

$$\overset{MLE}{=} \sum_i \sum_{x_i} \sum_{pa_i} \text{Count}(x_i, pa_i) \log \frac{\hat{P}(x_i, pa_i)}{\hat{P}(pa_i)}$$

$$= m \sum_i \sum_{x_i} \sum_{pa_i} \hat{P}(x_i, pa_i) \log \frac{\hat{P}(x_i, pa_i)\hat{P}(x_i)}{\hat{P}(pa_i)\hat{P}(x_i)}$$

$$= m \sum_i \sum_{x_i} \sum_{pa_i} \hat{P}(x_i, pa_i) \log \frac{\hat{P}(x_i, pa_i)}{\hat{P}(x_i)\hat{P}(pa_i)} + m \sum_i \sum_{x_i} \sum_{pa_i} \hat{P}(x_i, pa_i) \log \hat{P}(x_i)$$

$$= m \sum_i \hat{I}(X_i; Pa_i) - m \sum_i \hat{H}(X_i)$$

Decomposable score

- Log-data likelihood

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

independent of graph structure!

- MLE score decomposes over families of the BN (nodes + parents)
- Score(G ; D) = $\sum_i$ FamScore(X_i | Pa_i; D)
- Can exploit for computational efficiency!

Finding the optimal MLE structure

- Log-likelihood score:

$$\text{Score}(\mathcal{G};\mathcal{D}) = \sum_i \widehat{I}(X_i, \mathbf{Pa}_i)$$

- Want G^* = argmax_G Score(G ; D)
- Lemma: $G \subseteq G'$ ➔ Score(G; D) $\leq$ Score(G'; D)

Complete graph maximizes log data likelihood!

"Information never hurts"

RV X, $A \subset B$

$H(X|A) \geq H(X|B)$

$I(X;A) = H(X) - H(X|A)$

$\Rightarrow I(X;B) \geq I(X;A)$

Finding the optimal MLE structure

- Optimal solution for MLE is always the fully connected graph!!! ☹
  - ➔ Non-compact representation; Overfitting!!

- Solutions:
  - Priors over parameters / structures (later)
  - Constraint optimization (e.g., bound #parents)

Constraint optimization of BN structures

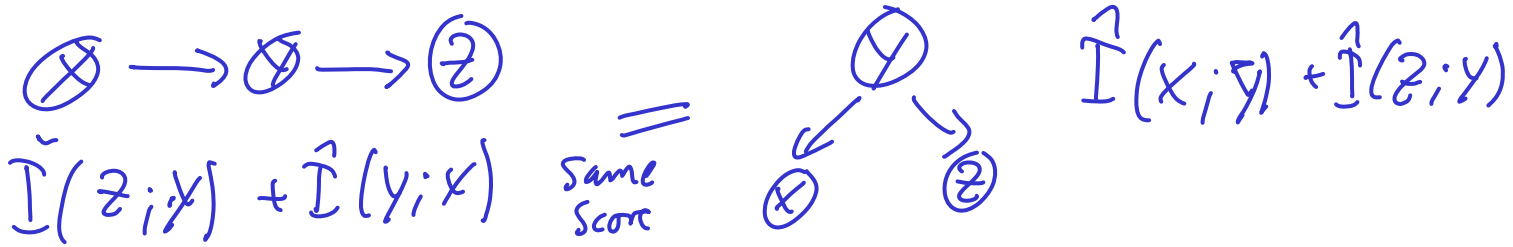
- **Theorem**: for any fixed $d \geq 2$, finding the optimal BN (w.r.t. MLE score) is NP-hard
- What about d=1??
- Want to find optimal tree!

Finding the optimal tree BN

- Scoring function

$$\text{Score}(\mathcal{G};\mathcal{D}) = \sum_i \hat{I}(X_i, \mathbf{Pa}_i)$$

- Scoring a tree



b/c: $I(X,Y) = I(Y,X)$

Same skeleton ⇒ same score

Finding the optimal tree skeleton

- Can reduce to following problem:
- Given graph G = (V,E), and nonnegative weights w_e for each edge $e=(X_i,X_j)$
  - In our case: $w_e = \hat{I}(X_i,X_j)$
- Want to find tree $T \subseteq E$ that maximizes $\sum_{e \in T} w_e$
- Maximum spanning tree problem!
- Can solve in time $O(|E| \log |E|)$!

Chow-Liu algorithm

- For each pair X_i, X_j of variables compute

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

- 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 complete graph with weight of edge (X_i,X_i) given by the mutual information
- Find maximum spanning tree ➔ skeleton
- Orient the skeleton using breadth-first search

Generalizing Chow-Liu

- Tree-augmented Naïve Bayes Model [Friedman '97]
- If evidence variables are correlated, Naïve Bayes models can be overconfident
- **Key idea**: Learn optimal tree for conditional distribution $P(X_1,\ldots,X_n \mid Y)$
- Can do optimally using Chow-Liu (homework! ☺)

Tasks

- Subscribe to Mailing list
  https://utils.its.caltech.edu/mailman/listinfo/cs155
- Select recitation times
- Read Koller & Friedman Chapter 17.1-17.3, 18.1-2, 18.4.1
- Form groups and think about class projects. If you have difficulty finding a group, email Pete Trautman