Learning with Graphical Models

STAT 37710 / CMSC 35300

Rebecca Willett and Yuxin Chen

Example: medical diagnosis

- One variable for each symptom
  - e.g. “fever”, “cough”, “fast breathing”, “nausea”, ...
- One variable for each disease
  - e.g. “pneumonia”, “flu”, “common cold”, “bronchitis”, ...
- Diagnosis is performed by inference in the model:

$$p(\text{pneumonia} = 1 \mid \text{cough} = 1, \text{fever} = 1, \text{nausea} = 0)$$

fever	cough	f. breathing	nausea	pneumonia	flu	common cold	brochitis	...
1	1	0	0	1	1	0	1	
1	0	1	1	1	0	0	0	
0	0	1	1	0	0	0	1	
0	1	0	0	0	1	1	0	
1	0	0	1	0	0	0	0	
...								

Structure through independence

- If $X_1, \ldots, X_n$ are independent, then

$$p(x_1, \ldots, x_n) = p(x_1) \cdot \cdots \cdot p(x_n)$$

- 2^n entries can be described by just n numbers (if $X_i \in \{0, 1\}$)!
- However, this is not a very useful model: observing a variable X_i cannot influence our predictions of X_j
- If $X_1, \ldots, X_n$ are *conditionally independent* given Y, i.e. $X_i \perp X_{-i} \mid Y$, then

$$p(y, x_1, \ldots, x_n) = p(y) \prod_{i=1}^{n} p(x_i \mid y)$$

- This is a simple yet powerful model!

More generally: graphical models

Two types of graphical models:

- Directed graphs (aka **Bayesian Networks** (BNs))
- Undirected graphs (aka Markov Random Fields (MRFs))

Principled and general methods for

- Probabilistic inference
- **Learning**

Why should we care?

- supervised learning (e.g. GBC), unsupervised learning (e.g. latent variable models), reinforcement learning (e.g., learning markov decision processes) ..

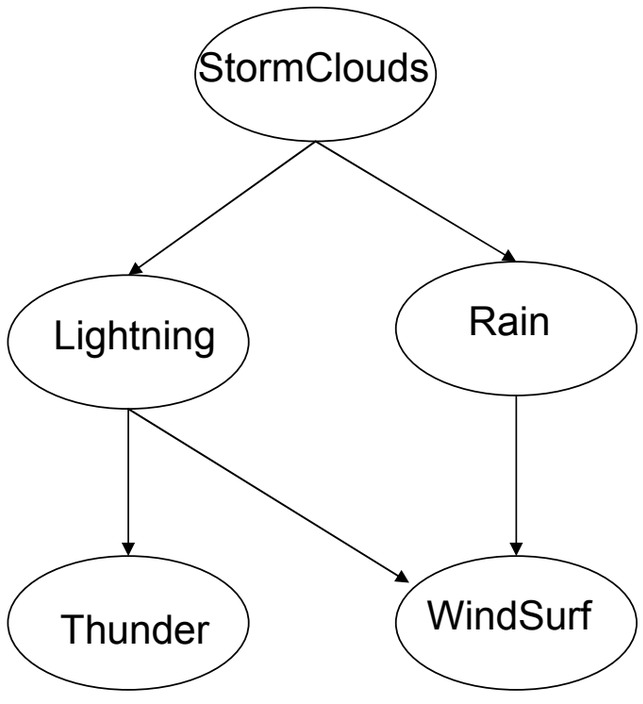
Bayesian network

Notation Pa_i: random variables, X_i's parents; pa_i: realization

Definition: Bayesian network

A Bayesian network is specified by a directed acyclic graph (DAG) $G = (V, E)$ with

- One node $i \in V$ for each random variable X_i
- One conditional probability distribution (CPD) per node, $p(x_i \mid \text{pa}_i)$, specifying the variable's probability conditioned on its parents' values



Parents	P(W|Pa)	P(¬W|Pa)
L, R	0	1.0
L, ¬R	0	1.0
¬L, R	0.2	0.8
¬L, ¬R	0.9	0.1

WindSurf

Bayesian network

Notation Pa_i: random variables, X_i's parents; pa_i: realization

Definition: Bayesian network

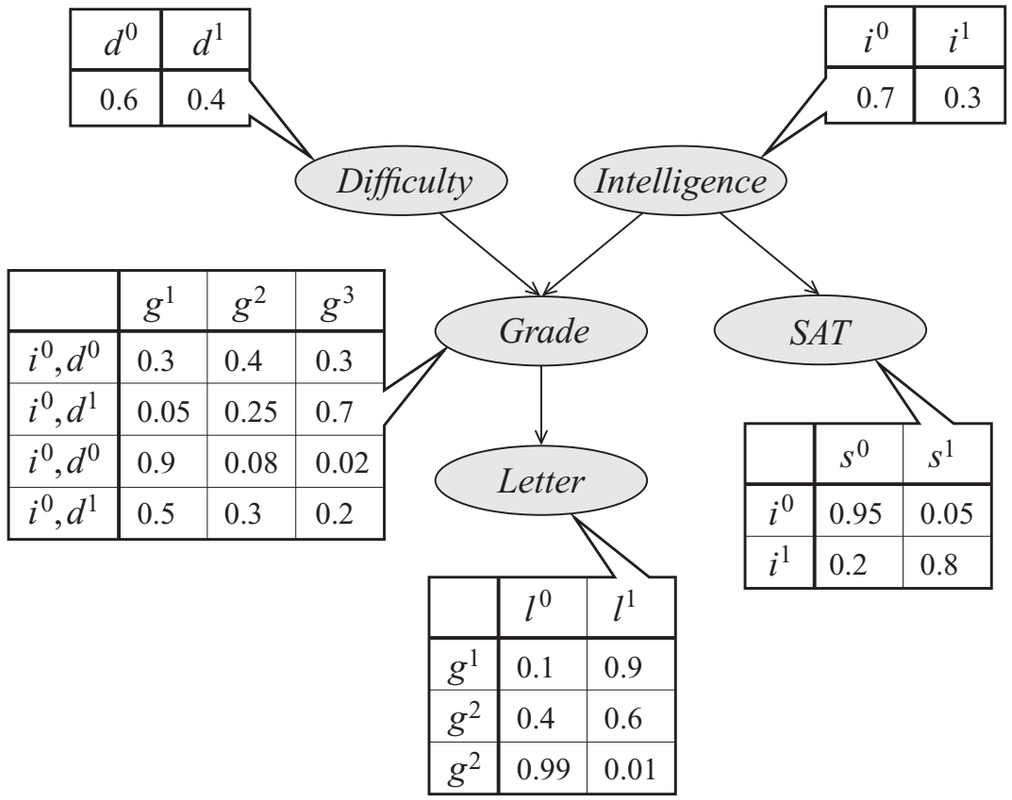
A Bayesian network is specified by a directed acyclic graph (DAG) $G = (V, E)$ with

- One node $i \in V$ for each random variable X_i
- One conditional probability distribution (CPD) per node, $p(x_i \mid \mathsf{pa}_i)$, specifying the variable's probability conditioned on its parents' values

Corresponds 1-1 with a particular factorization of the joint distribution:

$$P(x_1, \ldots, x_n) = \prod_{i \in V} P(x_i \mid \mathsf{pa}_i)$$

BN structure implies conditional independencies



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^0, d^0	0.9	0.08	0.02
i^0, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^2	0.99	0.01

- The joint distribution for above BN factors as

$$p(d, i, g, s, l) = p(d)p(i)p(g \mid i, d)p(s \mid i)p(l \mid g)$$

- Note that by the chain rule, any distribution can be written as

$$p(d, i, g, s, l) = p(d)p(i \mid d)p(g \mid i, d)p(s \mid i, d, g)p(l \mid g, d, i, g, s)$$

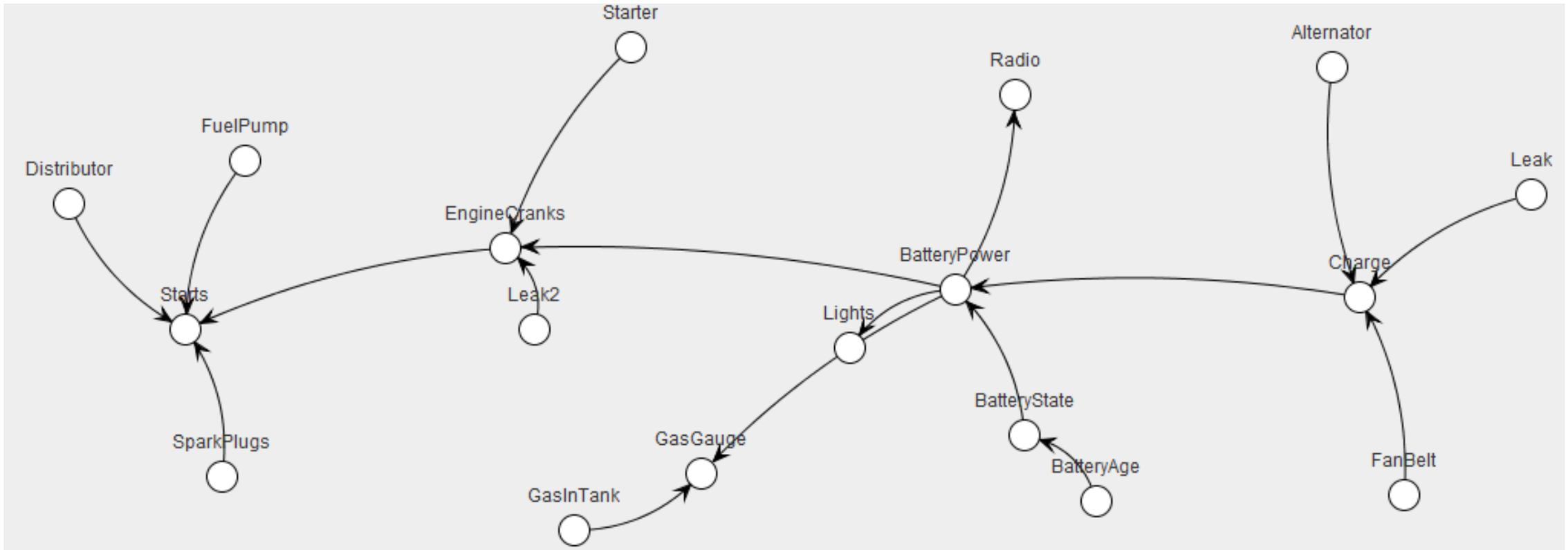
- Thus, we are assuming the following additional independencies

$$D \perp I \qquad S \perp \{D, G\} \mid I \qquad L \perp \{I, D, S\} \mid G \qquad \ldots$$

BN: another example

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \mathsf{pa}_i)$$

Will my car start this morning?

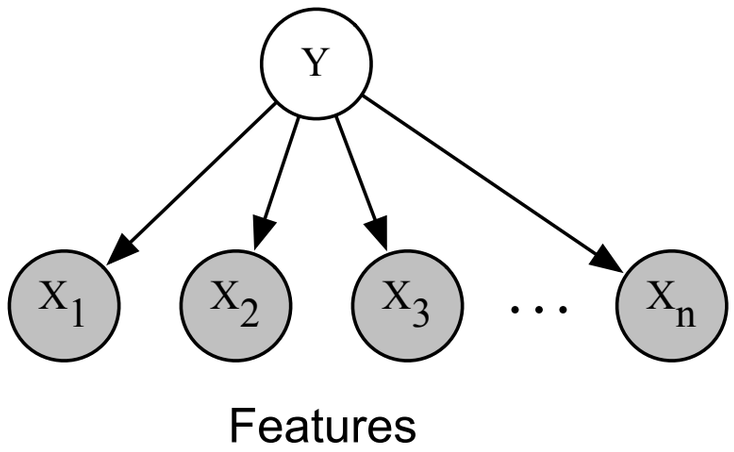


Heckerman et al., Decision-theoretic Troubleshooting, 1995

Bayesian networks are generative models

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \mathsf{pa}_i)$$

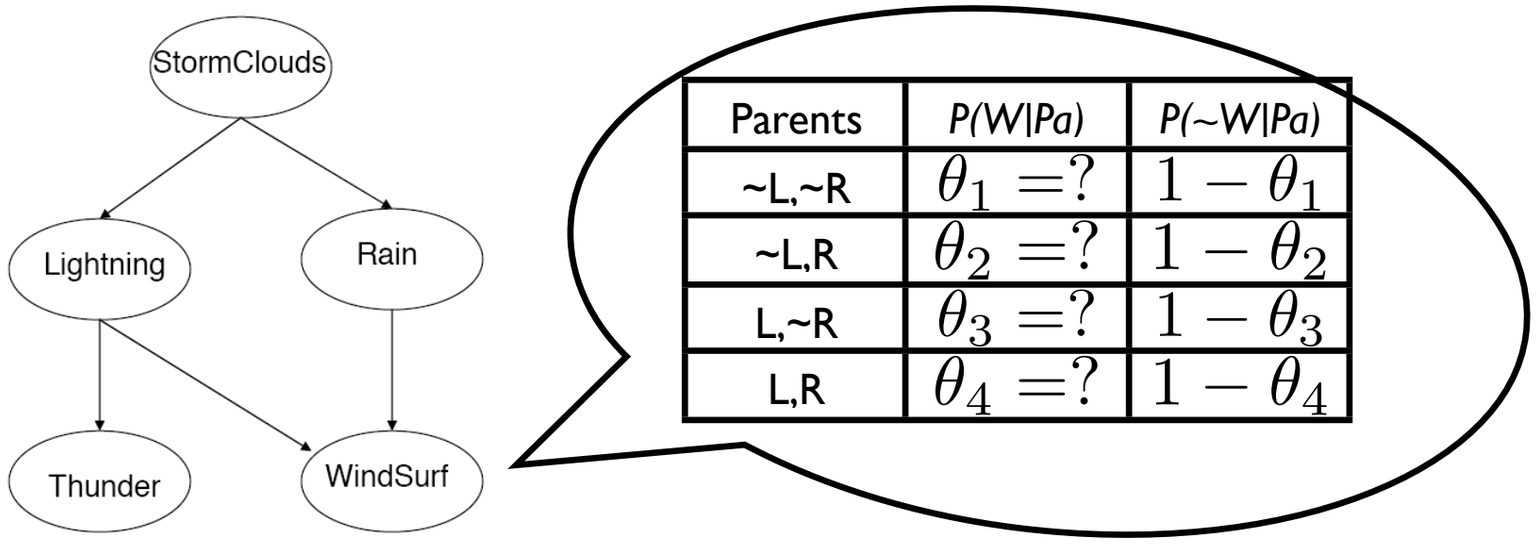
naive Bayes



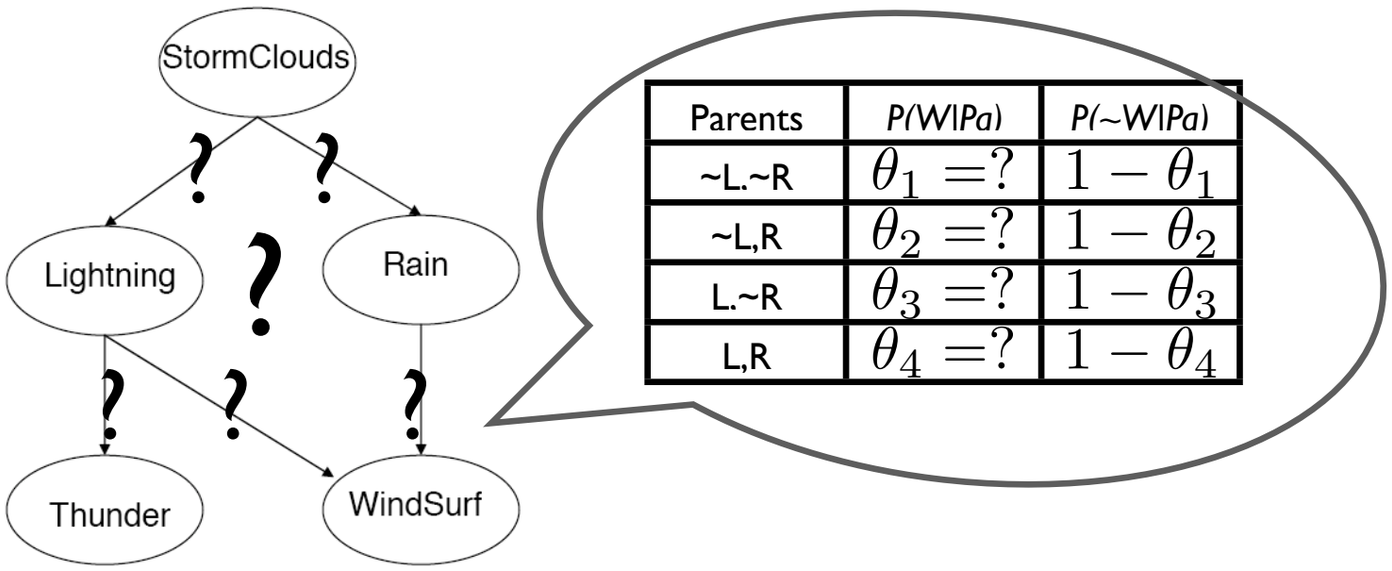
Can interpret Bayesian network as a generative process, where variables are sampled in topological order.

Learning BN from data

- Parameter learning/estimation: infer θ from data, given G



- Structure learning: inferring G and θ from data



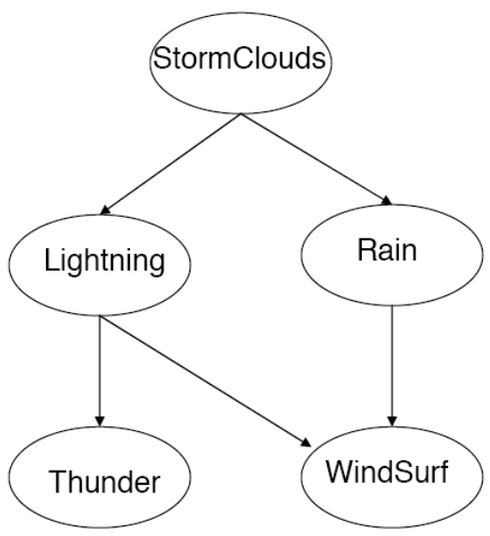
Parameter learning: estimating CPDs

Given

- a DAG G over n variables $X_1, \ldots, X_n$
- M i.i.d. samples/records of data $D = \{x^1, \ldots, x^M\}$
- each record $x^m = (x_1^m, \ldots, x_n^m), \quad m \in \{1, \ldots, M\}.$

Goal

- estimate θ from D

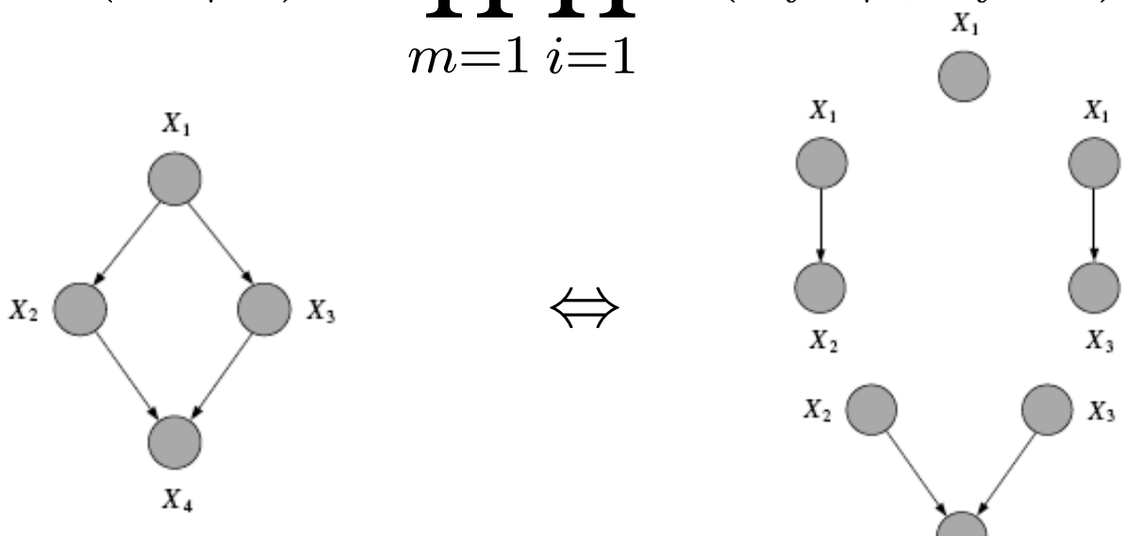


Parents	*P(W|Pa)*	*P(~W|Pa)*
~L,~R	$\theta_1 = ?$	$1 - \theta_1$
~L,R	$\theta_2 = ?$	$1 - \theta_2$
L,~R	$\theta_3 = ?$	$1 - \theta_3$
L,R	$\theta_4 = ?$	$1 - \theta_4$

MLE for BN parameters from complete data

- Likelihood

$$P(D \mid \theta) = \prod_{m=1}^{M} \prod_{i=1}^{n} P(x_i^m \mid \mathsf{pa}_i^m, \theta)$$



$$P(x \mid \theta) = P(x_1 \mid \theta_1) P(x_2 \mid x_1, \theta_2) P(x_3 \mid x_1, \theta_3) P(x_4 \mid x_2, x_3, \theta_4)$$

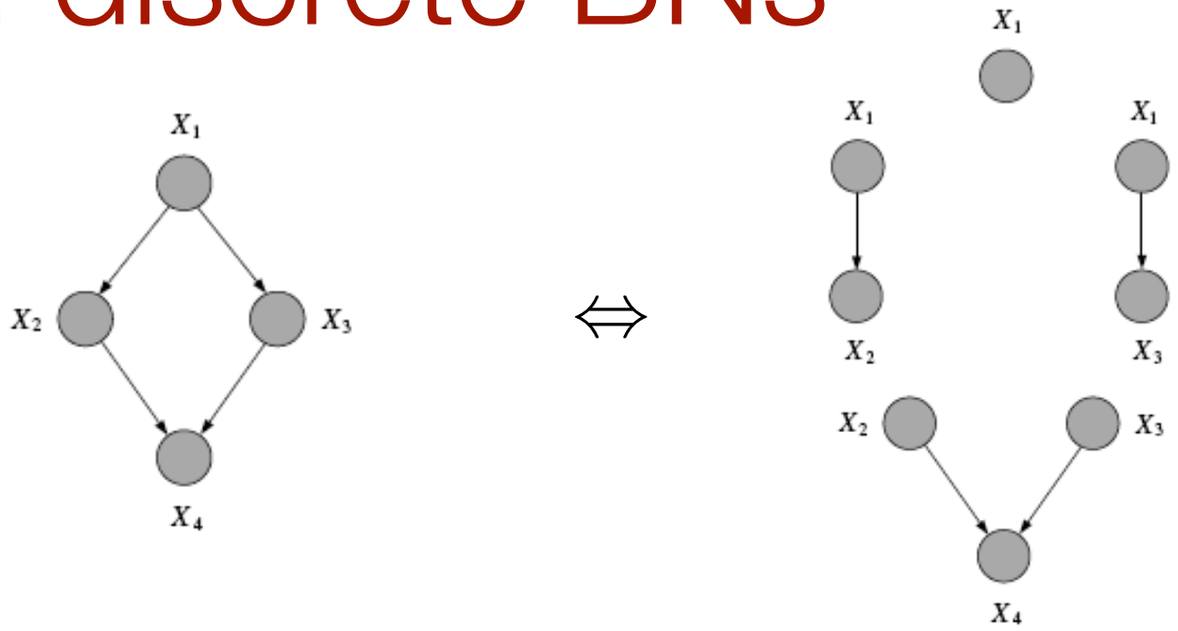
- Log likelihood

$$L(\theta; D) = \log P(D \mid \theta) = \sum_{m=1}^{M} \sum_{i=1}^{n} \log P(x_i^m \mid \mathsf{pa}_i^m, \theta)$$

- For each variable X_i estimate

$$\widehat{\theta}_{X_i|\mathsf{Pa}_i} \in \arg\max_{\theta_{X_i|\mathsf{Pa}_i}} \sum_{m=1}^{M} \log P(x_i^m \mid \mathsf{pa}_i^m, \theta_{X_i|\mathsf{Pa}_i})$$

Example: discrete BNs



$$P(x \mid \theta) = P(x_1 \mid \theta_1)P(x_2 \mid x_1, \theta_2)P(x_3 \mid x_1, \theta_3)P(x_4 \mid x_2, x_3, \theta_4)$$

- For discrete variables, each CPD can be represented as a table

$$\theta_{X_i|\mathsf{Pa}_i} := P(X_i \mid \mathsf{Pa}_i)$$

- Log likelihood

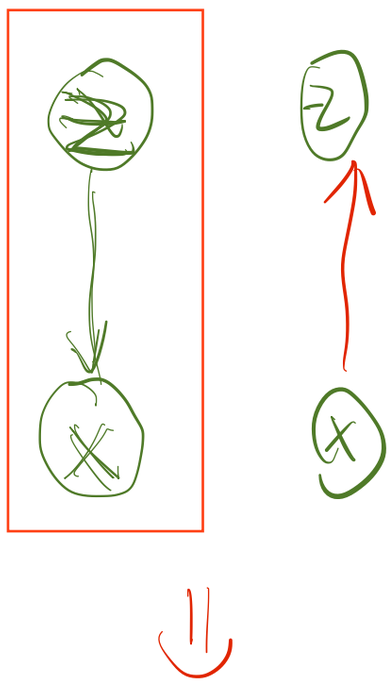
$$L(\theta; D) = \log \prod_{m=1}^{M} \prod_{i=1}^{n} P(x_i^m \mid \mathsf{pa}_i^m)$$

- For each variable X_i estimate

$$\widehat{\theta}_{X_i|\mathsf{Pa}_i} \in \arg\max_{\theta_{X_i|\mathsf{Pa}_i}} \sum_{m=1}^{M} \log P(x_i^m \mid \mathsf{pa}_i^m, \theta_{X_i|\mathsf{Pa}_i}) = \frac{\mathsf{Count}(X_i, \mathsf{Pa}_i)}{\mathsf{Count}(\mathsf{Pa}_i)}$$

Estimate θ from partially observed data

- Let $\mathbf{X}$ be (all) observed variables
- Let $\mathbf{Z}$ be (all) un-observed variables
- can't calculate MLE



$$\widehat{\theta} \in \arg\max_{\theta} \log P(\mathbf{X}, \mathbf{Z} \mid \theta)$$

P(X, Z)

- EM algorithm (c.f. lecture on *Gaussian Mixture Model*)

$$\widehat{\theta} \in \arg\max_{\theta} \mathbb{E}_{\mathbf{Z}|\mathbf{X},\theta}[\log P(\mathbf{X}, \mathbf{Z} \mid \theta)]$$

E step : compute expected complete-data log likelihood

$Q(\theta; \hat{\theta}^{(t-1)}) = \mathbb{E}$

Learning Bayesian network structure

We have discussed how to learned parameters for given a Bayesian network struture.

Where does the structure come from?

Score-based structure learning

- Define scoring function $S(G; D)$, which quantifies, for each structure G, the fit to the data D
- Search over Bayes net structures G

$$G^* \in \arg\max_G S(G; D)$$

- Key question: how should we score a Bayes Net?

MLE score for BN structure

For fixed structure G, already know how to calculate MLE for parameters:

$$\widehat{\theta}_G \in \arg\max_{\theta} \log P(D \mid \theta, G)$$

Can use maximum likelihood of data to score a Bayes Net

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

This scoring function has the following form:

$$\log \widehat{P}(D \mid \widehat{\theta}_G, G) = M \sum_{i=1}^{n} \widehat{\mathbb{I}}(X_i; \mathsf{Pa}_i) + \text{const.}$$

where $n := \#\text{samples}$, $M := \#\text{variables}$, and $\widehat{\mathbb{I}}(\cdot;\cdot)$ is the *empirical mutual information.*

Information-theoretic interpretation of S

Given structure, max log likelihood of data

$$\log \widehat{P}(D \mid \widehat{\theta}_G, G) = \sum_{m=1}^{M} \sum_{i=1}^{n} \log \widehat{P}(X_i = x_i^m \mid \mathsf{Pa}_i = \mathsf{pa}_i^m)$$

$$= \sum_{i=1}^{n} \sum_{x_i, \mathsf{pa}_i} \mathsf{Count}(X_i = x_i, \mathsf{Pa}_i = \mathsf{pa}_i) \log \widehat{P}(x_i \mid \mathsf{pa}_i)$$

$$\overset{\mathsf{MLE}}{=} M \sum_{i=1}^{n} \underbrace{\sum_{x_i, \mathsf{pa}_i} \widehat{P}(X_i = x_i, \mathsf{Pa}_i = \mathsf{pa}_i) \log \widehat{P}(x_i \mid \mathsf{pa}_i)}_{-\widehat{\mathbb{H}}(X_i \mid \mathsf{Pa}_i)}$$

$$= -M \sum_{i=1}^{n} \widehat{\mathbb{H}}(X_i \mid \mathsf{Pa}_i)$$

$$= M \sum_{i=1}^{n} \widehat{\mathbb{I}}(X_i; \mathsf{Pa}_i) \quad \underbrace{-M \sum_{i=1}^{n} \mathbb{H}(X_i)}_{\mathsf{const} \leftarrow \text{independent of } G}$$

Mutual information

Definition: Mutual Information

Measure of dependence between random variables

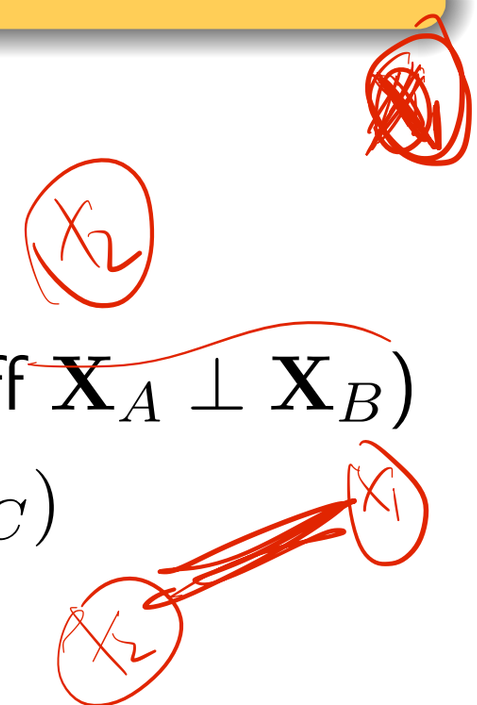
$$\mathbb{I}(X_i; X_j) = \sum_{x_i, x_j} P(x_i, x_j) \log \frac{P(x_i, x_j)}{P(x_i)P(x_j)}$$

For set of variables

$$\mathbb{I}(\mathbf{X}_A; \mathbf{X}_B) = \sum_{\mathbf{x}_A; \mathbf{x}_B} P(\mathbf{x}_A; \mathbf{x}_B) \log \frac{P(\mathbf{x}_A; \mathbf{x}_B)}{P(\mathbf{x}_A)P(\mathbf{x}_B)}$$

Properties of mutual information

- Symmetry: $\mathbb{I}(\mathbf{X}_A; \mathbf{X}_B) = \mathbb{I}(\mathbf{X}_B; \mathbf{X}_A)$
- Non-negativity: $\mathbb{I}(\mathbf{X}_A; \mathbf{X}_B) \geq 0$ (equality holds iff $\mathbf{X}_A \perp \mathbf{X}_B$)
- Monotonicity: $\forall B \subseteq C : \mathbb{I}(\mathbf{X}_A; \mathbf{X}_B) \leq \mathbb{I}(\mathbf{X}_A; \mathbf{X}_C)$



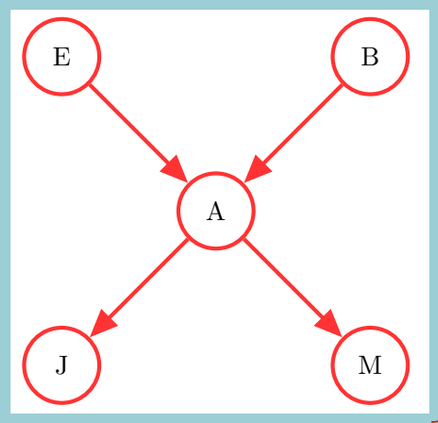
Empirical mutual information

Measure of dependence between empirical distributions

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

For discrete random variables, $\widehat{P}(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{M}$

Example: Scoring a Bayes net



E	B	A	M	J
0	0	0	0	0
1	0	1	1	0
1	0	1	1	1
0	1	1	0	1
0	1	0	0	1
...	...	...	...	...

Courtesy [Krause, Probablistic AI]

$$S(G; D) = \sum_{i=1}^{n} \widehat{\mathbb{I}}(X_i, \mathsf{Pa}_i) = \widehat{\mathbb{I}}(E; \emptyset) + \widehat{\mathbb{I}}(B; \emptyset) + \widehat{\mathbb{I}}(A; B, E) + \widehat{\mathbb{I}}(J; A) + \widehat{\mathbb{I}}(M; A)$$

Maximizing the MLE score

MLE of a Bayes Net (structure):

$$\widehat{G} = \arg\max_G S(G; D) = \arg\max_G \sum_{i=1}^{n} \widehat{\mathbb{I}}(X_i; \mathsf{Pa}_i)$$

What's the optimal solution?

Regularizing a BN

- Optimal solution for MLE is always the fully connected graph
  - Non-compact representation; overfitting
- Solution: prefer "simpler" models by adding a prior:

$$P(D \mid G) = \int_{\Theta_G} P(D \mid \theta_G, G) P(\theta_G \mid G) d\theta_G$$

- Bayesian Information Criterion (BIC) score:

$$S_{\text{BIC}}(G, D) = \sum_{i=1}^{n} \widehat{\mathbb{I}}(X_i; \text{Pa}_i) - \frac{\log M}{2M} |G|$$

where $|G|$ is the #parameters of G, n is number of variables, and M is number of training examples.

BIC is consistent

BIC will identify correct structure as $M \to \infty$.

(Additional reading: Ch 4.4.1, Bishop, 2006)

Finding the best structure

Finding the BN that maximizes the BIC score is NP hard

How many graphs?

Use local search

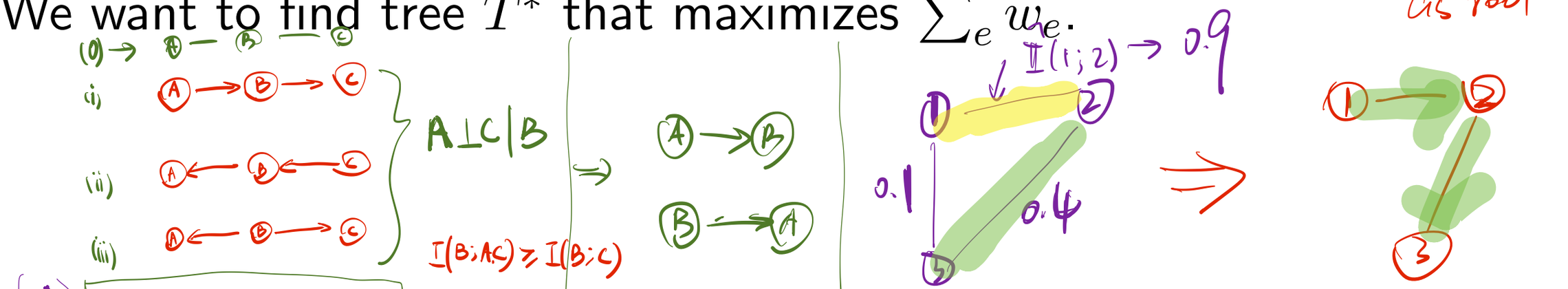
- Start with no edges
- Add/remove edges to increase score while preserving acyclicity
- Generally may get stuck in local optima

Tree-shaped Bayes Net: finding the optimal tree

Given

- graph $G = (V, E)$
- nonnegative weights $w_e := \widehat{\mathbb{I}}(X_i, X_j)$ for each edge $e = (X_i, X_j)$.

We want to find tree T^* that maximizes $\sum_e w_e$.



- Maximum (weight) spanning tree! (e.g. Kruskal's algorithm)
  - Greedily add edges, while making sure it's a tree at every step.
  - Can solve optimally in time $O(|E| \log |E|)$.
- Pick any variable as root, and fix edge directions via breadth first search. Why?

Algorithm: Chow-Liu tree learning 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 empirical mutual information

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

- Define complete graph with weight of edge $w_e := \widehat{\mathbb{I}}(X_i, X_j)$
- Find maximum spanning tree (undirected tree)
- Pick any variable as root and orient the edges away using breadth-first search

Extension: Tree-augmented Naive Bayes

- Naive Bayes model overcounts, because correlcation between features not considered (c.f. Generative Classification lecture)
- Same as Chow-Liu on features, but with $\widehat{\mathbb{I}}(X_i, X_j \mid Y)$ instead of $\widehat{\mathbb{I}}(X_i, X_j)$:

$$\widehat{\mathbb{I}}(X_i; X_j \mid Y) = \sum_{y, x_i, x_j} \widehat{P}(y, x_i, x_j) \log \frac{\widehat{P}(x_i, x_j \mid y)}{\widehat{P}(x_i \mid y)\widehat{P}(x_j \mid y)}$$

- Then learn $P(X_i \mid \mathsf{Pa}_i, Y)$ as before

Summary

Graphical model and Bayesian networks

- Structure through independence

Learning BN parameters

- MLE/MAP learns parameters
- For compete discrete data, compute MLE by counting.
  Optionally, use pseudo-counts (Beta prior) to avoid overfitting
- For partially observed data, use EM

Learning BN structure

- Score = likelihood for best choice of parameters
- Search for structure by maximizing score function
- Best tree (Chow-Liu)

Reading materials & acknowledgement

- D. Koller: Graphical Models in a Nutshell, 2007
  `https://ai.stanford.edu/~koller/Papers/Koller+al:SRL07.pdf`
- Ch 4.4.1 (BIC), Ch 8.1, Ch 8.4.8: C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006
- Materials from C. Guestrin as part of Machine Learning course (CMU/UW).