# CMSC 471: Probability, and Reasoning and Learning with Uncertainty

Frank Ferraro – ferraro@umbc.edu

Some slides courtesy Tim Finin

# Topics

- Review probability theory
- Bayesian inference
  - From the joint distribution
  - Using independence/factoring
  - From sources of evidence
- Representation and Learning
  - Bayes nets (a type probabilistic graphical models)
  - MLE (maximum likelihood estimation)
  - Naïve Bayes algorithm for inference and classification tasks

# Many Sources of Uncertainty

- Uncertain **inputs --** missing and/or noisy data
- Uncertain **knowledge**
  - Multiple causes lead to multiple effects
  - Incomplete enumeration of conditions or effects
  - Incomplete knowledge of causality in the domain
  - Probabilistic/stochastic effects
- Uncertain **outputs**
  - Abduction and induction are inherently uncertain
  - Default reasoning, even deductive, is uncertain
  - Incomplete deductive inference may be uncertain

▸ Probabilistic reasoning only gives probabilistic results

# Decision making with uncertainty

**Rational** behavior: for each possible action:

- Identify possible outcomes and for each
  - Compute **probability** of outcome
  - Compute **utility** of outcome
- Compute probability-weighted **(expected) utility** over possible outcomes
- Select action with the highest expected utility (principle of **Maximum Expected Utility**)

# Consider


- Your house has an alarm system
- It should go off if a burglar breaks into the house
- It can go off if there is an earthquake
- How can we predict what's happened if the alarm goes off?
  - Someone has broken in!
  - It's a minor earthquake

# Probability theory 101

- **Random variables**
  - Domain
- **Atomic event**: complete specification of state
- **Prior probability**: degree of belief without any other evidence or info
- **Joint probability**: matrix of combined probabilities of set of variables

- Alarm, Burglary, Earthquake
  - Boolean (like these), discrete, continuous
- Alarm=T∧Burglary=T∧Earthquake=F
  alarm ∧ burglary ∧ ¬earthquake
- P(Burglary) = 0.1
  P(Alarm) = 0.1
  P(earthquake) = 0.000003
- P(Alarm, Burglary) =

| | **alarm** | **¬alarm** |
|---|---|---|
| **burglary** | .09 | .01 |
| **¬burglary** | .1 | .8 |

# Probability theory 101

| | alarm | ¬alarm |
|---|---|---|
| burglary | .09 | .01 |
| ¬burglary | .1 | .8 |

- **Conditional probability**: prob. of effect given causes
- **Computing conditional probs**:
  - $P(a \mid b) = P(a \wedge b) / P(b)$
  - $P(b)$: **normalizing** constant
- **Product rule**:
  - $P(a \wedge b) = P(a \mid b) * P(b)$
- **Marginalizing**:
  - $P(B) = \Sigma_a P(B, a)$
  - $P(B) = \Sigma_a P(B \mid a) P(a)$ (**conditioning**)

- P(burglary | alarm) = .47
  P(alarm | burglary) = .9
- P(burglary | alarm) =
  P(burglary ∧ alarm) / P(alarm)
  = .09/.19 = .47
- P(burglary ∧ alarm) =
  P(burglary | alarm) * P(alarm)
  = .47 * .19 = .09
- P(alarm) =
  P(alarm ∧ burglary) +
  P(alarm ∧ ¬burglary)
  = .09+.1 = .19

# Example: Inference from the joint

| | alarm | | ¬alarm | |
|---|---|---|---|---|
| | **earthquake** | **¬earthquake** | **earthquake** | **¬earthquake** |
| **burglary** | .01 | .08 | .001 | .009 |
| **¬burglary** | .01 | .09 | .01 | .79 |

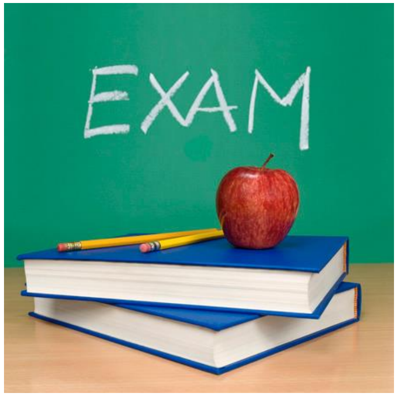
P(burglary | alarm) = α P(burglary, alarm)
= α [P(burglary, alarm, earthquake) + P(burglary, alarm, ¬earthquake)
= α [ (.01, .01) + (.08, .09) ]
= α [ (.09, .1) ]

Since P(burglary | alarm) + P(¬burglary | alarm) = 1, α = 1/(.09+.1) = 5.26
(i.e., P(alarm) = 1/α = .19 – **quizlet**: how can you verify this?)

P(burglary | alarm) = .09 * 5.26 = .474

P(¬burglary | alarm) = .1 * 5.26 = .526

# Consider




- A student has to take an exam
- She might be smart
- She might have studied
- She may be prepared for the exam
- How are these related?

# Exercise:
# Inference from the joint

| p(smart ∧ study ∧ prep) | smart | | ¬smart | |
|---|---|---|---|---|
| | **study** | **¬study** | **study** | **¬study** |
| **prepared** | .432 | .16 | .084 | .008 |
| **¬prepared** | .048 | .16 | .036 | .072 |

## Queries:

- **What is the prior probability of *smart*?**
- What is the prior probability of *study*?
- What is the conditional probability of *prepared*, given *study* and *smart*?

# Exercise:
# Inference from the joint

| p(smart ∧ study ∧ prep) | smart | | ¬smart | |
|---|---|---|---|---|
| | study | ¬study | study | ¬study |
| **prepared** | .432 | .16 | .084 | .008 |
| **¬prepared** | .048 | .16 | .036 | .072 |

## Queries:

- **What is the prior probability of *smart*?**
- What is the prior probability of *study*?
- What is the conditional probability of *prepared*, given *study* and *smart*?

p(smart) = .432 + .16 + .048 + .16 = **0.8**

# Exercise:
# Inference from the joint

| p(smart ∧ study ∧ prep) | smart | | ¬smart | |
|---|---|---|---|---|
| | **study** | **¬study** | **study** | **¬study** |
| **prepared** | .432 | .16 | .084 | .008 |
| **¬prepared** | .048 | .16 | .036 | .072 |

## Queries:

- What is the prior probability of *smart*?
- **What is the prior probability of *study*?**
- What is the conditional probability of *prepared*, given *study* and *smart*?

# Exercise:
# Inference from the joint

| p(smart ∧ study ∧ prep) | smart | | ¬smart | |
|---|---|---|---|---|
| | study | ¬study | study | ¬study |
| prepared | .432 | .16 | .084 | .008 |
| ¬prepared | .048 | .16 | .036 | .072 |

## Queries:

- What is the prior probability of *smart*?
- **What is the prior probability of *study*?**
- What is the conditional probability of *prepared*, given *study* and *smart*?

p(study) = .432 + .048 + .084 + .036 = **0.6**

# Exercise:
# Inference from the joint

| p(smart ∧ study ∧ prep) | smart | | ¬smart | |
|---|---|---|---|---|
| | **study** | **¬study** | **study** | **¬study** |
| **prepared** | .432 | .16 | .084 | .008 |
| **¬prepared** | .048 | .16 | .036 | .072 |

## Queries:

- What is the prior probability of *smart*?
- What is the prior probability of *study*?
- **What is the conditional probability of *prepared*, given *study* and *smart*?**

# Exercise:
# Inference from the joint

| p(smart ∧ study ∧ prep) | smart | | ¬smart | |
|---|---|---|---|---|
| | **study** | **¬study** | **study** | **¬study** |
| **prepared** | .432 | .16 | .084 | .008 |
| **¬prepared** | .048 | .16 | .036 | .072 |

## Queries:

- What is the prior probability of *smart*?
- What is the prior probability of *study*?
- **What is the conditional probability of *prepared*, given *study* and *smart*?**

p(prepared|smart,study)= p(prepared,smart,study)/p(smart, study)
= .432 / (.432 + .048)
= **0.9**

# Independence

- When variables don't affect each others' probabil-ities, they are **independent;** we can easily compute their joint & conditional probability:

Independent(A, B) → $P(A \wedge B) = P(A) * P(B)$ or $P(A|B) = P(A)$

- {moonPhase, lightLevel} *might* be independent of {burglary, alarm, earthquake}
  - Maybe not: burglars may be more active during a new moon because darkness hides their activity
  - But if we know light level, moon phase doesn't affect whether we are burglarized
  - If burglarized, light level doesn't affect if alarm goes off
- Need a more complex notion of independence and methods for reasoning about the relationships

# Exercise: Independence

| p(smart ∧ study ∧ prep) | smart | | ¬smart | |
|---|---|---|---|---|
| | **study** | **¬study** | **study** | **¬study** |
| **prepared** | .432 | .16 | .084 | .008 |
| **¬prepared** | .048 | .16 | .036 | .072 |

**Queries:**

- **Q1: Is *smart* independent of *study*?**
- **Q2: Is *prepared* independent of *study*?**

**How can we tell?**

# Exercise: Independence

| p(smart ∧ study ∧ prep) | smart | | ¬smart | |
|---|---|---|---|---|
| | study | ¬study | study | ¬study |
| prepared | .432 | .16 | .084 | .008 |
| ¬prepared | .048 | .16 | .036 | .072 |

**Q1: Is *smart* independent of *study*?**

- You might have some intuitive beliefs based on your experience
- You can also check the data

Which way to answer this is better?

# Exercise: Independence

| p(smart ∧ study ∧ prep) | smart | | ¬smart | |
|---|---|---|---|---|
| | study | ¬study | study | ¬study |
| prepared | .432 | .16 | .084 | .008 |
| ¬prepared | .048 | .16 | .036 | .072 |

**Q1: Is *smart* independent of *study*?**

Q1 true iff p(smart|study) == p(smart)

p(smart|study) = p(smart,study)/p(study)
= (.432 + .048) / .6 = 0.8
0.8 == 0.8, so smart is independent of study

# Exercise: Independence

| p(smart ∧ study ∧ prep) | smart | | ¬smart | |
|---|---|---|---|---|
| | study | ¬study | study | ¬study |
| prepared | .432 | .16 | .084 | .008 |
| ¬prepared | .048 | .16 | .036 | .072 |

**Q2: Is *prepared* independent of *study*?**

- What is prepared?
- Q2 true iff

# Exercise: Independence

| p(smart ∧ study ∧ prep) | smart | | ¬smart | |
|---|---|---|---|---|
| | study | ¬study | study | ¬study |
| prepared | .432 | .16 | .084 | .008 |
| ¬prepared | .048 | .16 | .036 | .072 |

**Q2: Is *prepared* independent of *study*?**

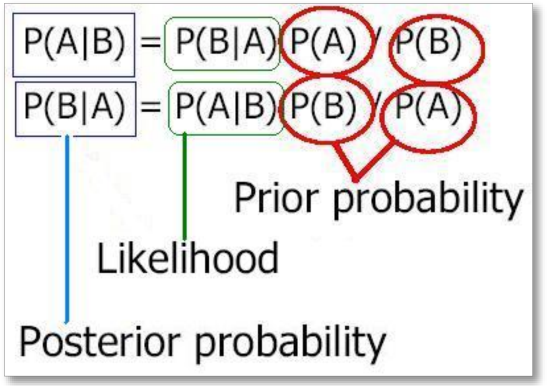
Q2 true iff p(prepared|study) == p(prepared)

p(prepared|study) = p(prepared,study)/p(study)

= (.432 + .084) / .6 = .86

0.86 ≠ 0.8, so prepared not independent of study

# Bayes' rule




Derived from the product rule:

– P(A, B) = P(A|B) * P(B) *# from definition of conditional probability*

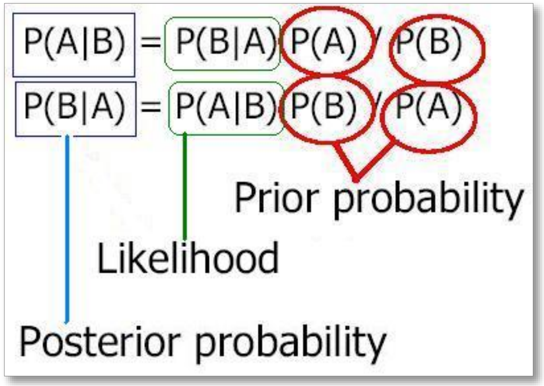
– P(B, A) = P(B|A) * P(A) *# from definition of conditional probability*

– P(A, B) = P(B, A) *# since order is not important*

So…

$$\mathbf{P(A|B) = \frac{P(B|A) * P(A)}{P(B)}}$$

# Useful for diagnosis!




- *C is a cause, E is an effect*:
  - P(C|E) = P(E|C) * P(C) / P(E)
- **Useful for diagnosis**:
  - E are (observed) effects and C are (hidden) causes,
  - Often have model for how causes lead to effects P(E|C)
  - May also have info (based on experience) on frequency of causes (P(C))
  - Which allows us to reason abductively from effects to causes (P(C|E))

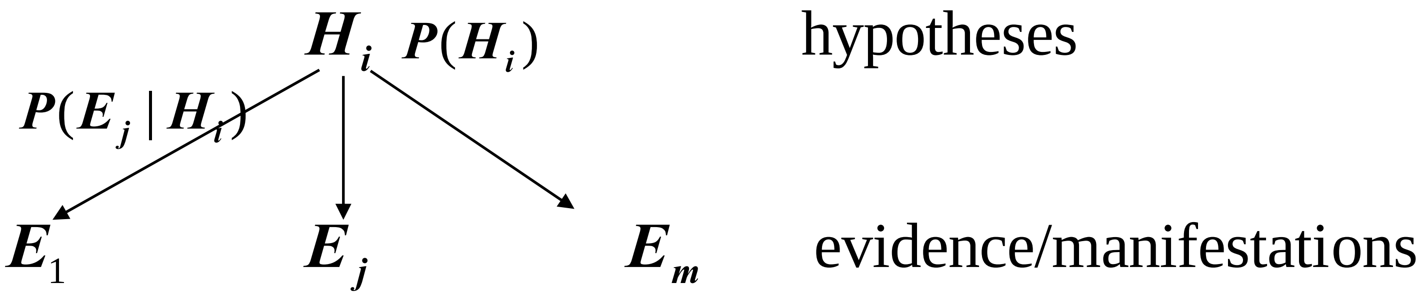
# Ex: meningitis and stiff neck

- Meningitis (M) can cause stiff neck (S), though there are other causes too
- Use S as a diagnostic symptom and estimate **p(M|S)**
- Studies can estimate p(M), p(S) & p(S|M), e.g. p(M)=0.7, p(S)=0.01, p(M)=0.00002
- Harder to directly gather data on p(M|S)
- Applying Bayes' Rule:

  p(M|S) = p(S|M) * p(M) / p(S) = 0.0014

# Reasoning from evidence to a cause

- In the setting of diagnostic/evidential reasoning




  – Know prior probability of hypothesis $P(H_i)$

  conditional probability $P(E_j \mid H_i)$

  – Want to compute the *posterior probability* $P(H_i \mid E_j)$

- Bayes' s theorem:

$$P(H_i \mid E_j) = P(H_i) * P(E_j \mid H_i) / P(E_j)$$

# Simple Bayesian diagnostic reasoning

- **Naive Bayes classifier**
- Knowledge base:
  - Evidence / manifestations: $E_1, \dots E_m$
  - Hypotheses / disorders: $H_1, \dots H_n$

    Note: $E_j$ and $H_i$ are **binary**; hypotheses are **mutually exclusive** (non-overlapping) and **exhaustive** (cover all possible cases)
  - Conditional probabilities: $P(E_j \mid H_i)$, $i = 1, \dots n$; $j = 1, \dots m$
- Cases (evidence for a particular instance): $E_1, \dots, E_l$
- Goal: Find the hypothesis $H_i$ with highest posterior
  - $\text{Max}_i \ P(H_i \mid E_1, \dots, E_l)$

# Simple Bayesian diagnostic reasoning

- Bayes' rule:

  $P(H_i \mid E_1 \ldots E_m) = P(E_1 \ldots E_m \mid H_i)\, P(H_i) \,/\, P(E_1 \ldots E_m)$

- Assume each evidence $E_i$ is conditionally independent of the others, *given* a hypothesis $H_i$, then:

  $P(E_1 \ldots E_m \mid H_i) = \prod^{m}_{j=1} P(E_j \mid H_i)$

- If only care about relative probabilities for $H_i$, then:

  $P(H_i \mid E_1 \ldots E_m) = \alpha\, P(H_i) \prod^{m}_{j=1} P(E_j \mid H_i)$

# Naïve Bayes

- Use Bayesian modeling
- Make the simplest possible independence assumption:
  - Each attribute is independent of the values of the other attributes, given the class variable
  - In our restaurant domain: Cuisine is independent of Patrons, *given* a decision to stay (or not)

# Bayesian Formulation

- $p(C \mid F_1, ..., F_n) = p(C)\, p(F_1, ..., F_n \mid C) / P(F_1, ..., F_n)$
  $= \alpha\, p(C)\, p(F_1, ..., F_n \mid C)$
- Assume each feature $F_i$ is *conditionally independent* of others given the class C. Then:
  $p(C \mid F_1, ..., F_n) = \alpha\, p(C)\, \Pi_i\, p(F_i \mid C)$
- Estimate each of these conditional probabilities from the observed **counts** in the training data:
  $p(F_i \mid C) = N(F_i \wedge C) / N(C)$
  - One subtlety of using the algorithm in practice: when your estimated probabilities are zero, ugly things happen
  - Fix: Add one to every count (aka Laplace smoothing—they have a different name for *everything*!)

# Naive Bayes: Example

$p(\text{Wait} \mid \text{Cuisine, Patrons, Rainy?}) =$

$= \alpha \bullet p(\text{Wait}) \bullet p(\text{Cuisine}|\text{Wait}) \bullet p(\text{Patrons}|\text{Wait}) \bullet p(\text{Rainy?}|\text{Wait})$

$$= \frac{p(\text{Wait}) \bullet p(\text{Cuisine}|\text{Wait}) \bullet p(\text{Patrons}|\text{Wait}) \bullet p(\text{Rainy?}|\text{Wait})}{p(\text{Cuisine}) \bullet p(\text{Patrons}) \bullet p(\text{Rainy?})}$$

We can estimate all of the parameters (p(F) and p(C) just by counting from the training examples

# Naive Bayes: Analysis

- Naive Bayes is amazingly easy to implement (once you understand the math behind it)
- Naive Bayes can outperform many much more complex algorithms—it's a baseline that should be tried or used for comparison
- Naive Bayes can't capture interdependencies between variables (obviously)—for that, we need Bayes nets!

# Bag of Words Classifier

$\gamma($

| | |
|---|---|
| seen | 2 |
| sweet | 1 |
| whimsical | 1 |
| recommend | 1 |
| happy | 1 |
| ... | ... |

$) = c$

*classifier*

*classifier*

# Naïve Bayes (NB) Classifier

$$\text{argmax}_Y \, p(X \mid Y) * p(Y)$$

label

text

*Start with Bayes Rule*

# Naïve Bayes (NB) Classifier

label

each word

$$\text{argmax}_Y \prod_t p(X_t|Y) * p(Y)$$

Iterate through possible vocab words

*Adopt naïve bag of words representation $X_t$*

*Assume position doesn't matter*

# Learning for a Naïve Bayes Classifier

Assuming V vocab types $w_1, \ldots, w_V$ and L classes $u_1, \ldots, u_L$(and appropriate corpora)

# Learning for a Naïve Bayes Classifier

Assuming V vocab types $w_1, \ldots, w_V$ and L classes $u_1, \ldots, u_L$(and appropriate corpora)

Q: What parameters (values/weights) must be learned?

# Learning for a Naïve Bayes Classifier

Assuming V vocab types $w_1, \ldots, w_V$ and L classes $u_1, \ldots, u_L$(and appropriate corpora)

Q: What parameters (values/weights) must be learned?

A: $p(w_v|u_l), p(u_l)$

# Learning for a Naïve Bayes Classifier

Assuming V vocab types $w_1, \ldots, w_V$ and L classes $u_1, \ldots, u_L$ (and appropriate corpora)

Q: What parameters (values/weights) must be learned?

A: $p(w_v | u_l), p(u_l)$

Q: How many parameters must be learned?

# Learning for a Naïve Bayes Classifier

Assuming V vocab types $w_1, \dots, w_V$ and L classes $u_1, \dots, u_L$(and appropriate corpora)

Q: What parameters (values/weights) must be learned?

A: $p(w_v | u_l), p(u_l)$

Q: How many parameters must be learned?

A: $LK + L$

# Learning for a Naïve Bayes Classifier

Assuming V vocab types $w_1, \ldots, w_V$ and L classes $u_1, \ldots, u_L$(and appropriate corpora)

Q: What parameters (values/weights) must be learned?

A: $p(w_v | u_l), p(u_l)$

Q: How many parameters must be learned?

A: $LK + L$

Q: What distributions need to sum to 1?

# Learning for a Naïve Bayes Classifier

Assuming V vocab types $w_1, \ldots, w_V$ and L classes $u_1, \ldots, u_L$ (and appropriate corpora)

| | |
|---|---|
| Q: What parameters (values/weights) must be learned? | A: $p(w_v \mid u_l), p(u_l)$ |
| Q: How many parameters must be learned? | A: $LK + L$ |
| Q: What distributions need to sum to 1? | A: Each $p(\cdot \mid u_l)$, and the prior |

# Multinomial Naïve Bayes: Learning

From training corpus, extract *Vocabulary*

**Calculate $P(c_j)$ terms**

For each $c_j$ in $C$ do

*$docs_j$* = all docs with class $=c_j$

$$p(c_j) = \frac{|docs_j|}{\#\,docs}$$

**Calculate $P(w_k \mid c_j)$ terms**

$Text_j$ = single doc containing all $docs_j$

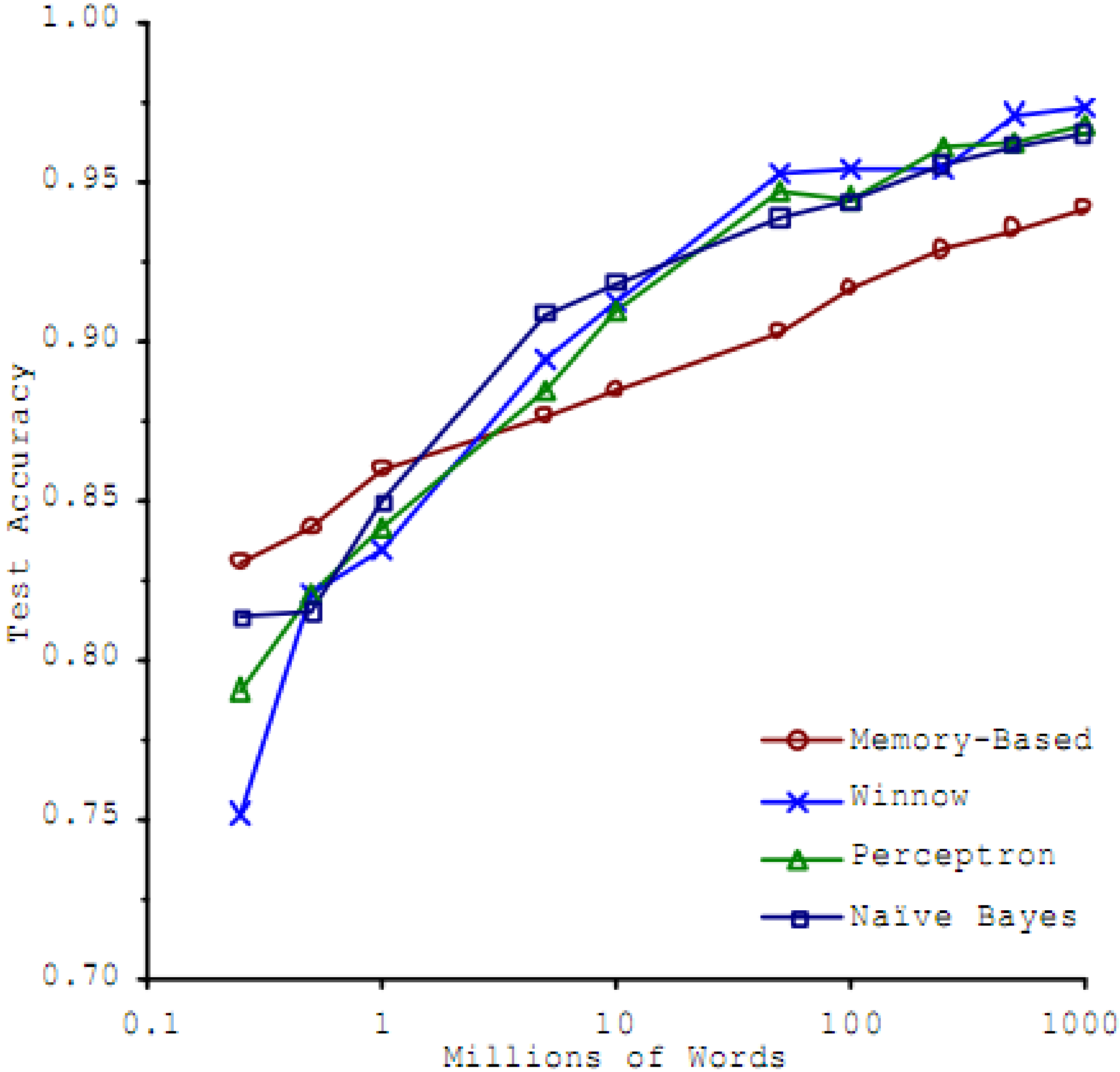
For each word $w_k$ in *Vocabulary*

$n_k$ = # of occurrences of $w_k$ in $Text_j$

$$p(w_k \mid c_j) = \text{class (unigram) LM} \propto \text{count(word } w_k \text{ in doc labeled with } c_j)$$

# Naive Bayes: Analysis

- Naive Bayes is amazingly easy to implement (once you understand the math behind it)
- Naive Bayes can outperform many much more complex algorithms—it's a baseline that should be tried or used for comparison
- Naive Bayes can't capture interdependencies between variables (obviously)—for that, we need Bayes nets!




Brill and Banko (2001)

With enough data, the classifier may not matter

# Naive Bayes: Analysis

- Naive Bayes is amazingly easy to implement (once you understand the math behind it)
- Naive Bayes can outperform many much more complex algorithms—it's a baseline that should be tried or used for comparison
- Naive Bayes can't capture interdependencies between variables (obviously)—for that, we need Bayes nets!

# Limitations

- Can't easily handle **multi-fault situations** or cases where intermediate (hidden) causes exist:
  - Disease D causes syndrome S, which causes correlated manifestations $M_1$ and $M_2$
- Consider composite hypothesis $H_1 \wedge H_2$, where $H_1$ & $H_2$ independent. What's relative posterior?

  $P(H_1 \wedge H_2 \mid E_1, \ldots, E_l) = \alpha\, P(E_1, \ldots, E_l \mid H_1 \wedge H_2)\, P(H_1 \wedge H_2)$

  $= \alpha\, P(E_1, \ldots, E_l \mid H_1 \wedge H_2)\, P(H_1)\, P(H_2)$

  $= \alpha \prod^{l}_{j=1} P(E_j \mid H_1 \wedge H_2)\, P(H_1)\, P(H_2)$
- How do we compute $P(E_j \mid H_1 \wedge H_2)$ ?

# Summary

- Probability a rigorous formalism for uncertain knowledge
- Joint probability distribution specifies probability of every atomic event
- Answer queries by summing over atomic events
- Must reduce joint size for non-trivial domains
- Bayes rule: compute from known conditional probabilities, usually in causal direction
- Independence & conditional independence provide tools
- Next: Bayesian belief networks

# Overview

- Bayesian Belief Networks (BBNs) can reason with networks of propositions and associated probabilities
- Useful for many AI problems
  - Diagnosis
  - Expert systems
  - Planning
  - Learning

# Probabilistic Graphical Models

A graph G that represents a probability distribution over random variables $X_1, \ldots, X_N$

# Probabilistic Graphical Models

A graph G that represents a probability distribution over random variables $X_1, \ldots, X_N$

Graph G = (vertices V, edges E)

Distribution $p(X_1, \ldots, X_N)$

# Probabilistic Graphical Models

A graph G that represents a probability distribution over random variables $X_1, \ldots, X_N$

Graph G = (vertices V, edges E)

Distribution $p(X_1, \ldots, X_N)$

Vertices $\leftrightarrow$ random variables

Edges show dependencies among random variables

# Probabilistic Graphical Models

A graph G that represents a probability distribution over random variables $X_1, \ldots, X_N$

Graph G = (vertices V, edges E)
Distribution $p(X_1, \ldots, X_N)$

Vertices $\leftrightarrow$ random variables
Edges show dependencies among random variables

Two main flavors: *directed* graphical models and *undirected* graphical models (come talk to me)

# Directed Graphical Models

A *directed* (acyclic) graph G=(V,E) that represents a probability distribution over random variables $X_1, \ldots, X_N$

Joint probability factorizes into factors of $X_i$ conditioned on the parents of $X_i$

# Directed Graphical Models

A *directed* (acyclic) graph G=(V,E) that represents a probability distribution over random variables $X_1, \ldots, X_N$

Joint probability factorizes into factors of $X_i$ conditioned on the parents of $X_i$

Benefit: the independence properties are *transparent*

# Directed Graphical Models

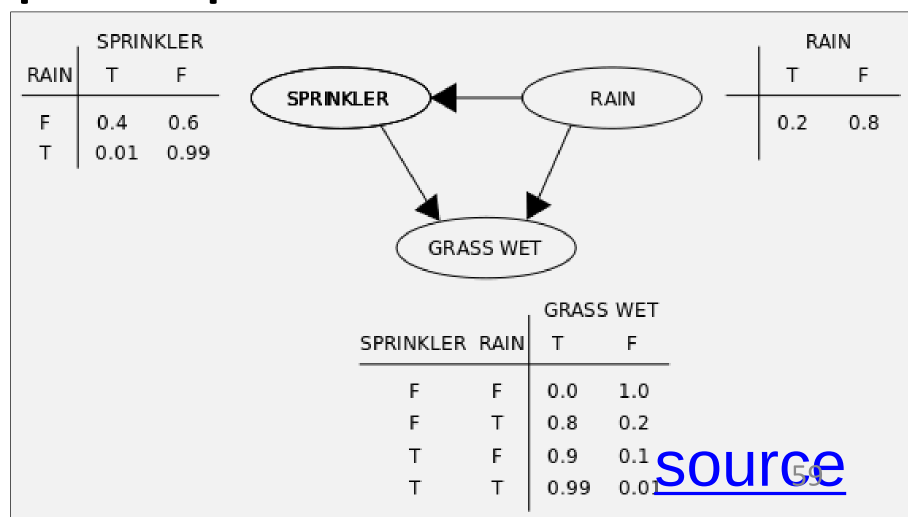
A *directed* (acyclic) graph G=(V,E) that represents a probability distribution over random variables $X_1, \dots, X_N$

Joint probability factorizes into factors of $X_i$ conditioned on the parents of $X_i$

A graph/joint distribution that follows this is a **Bayesian network**

# BBN Definition

- AKA Bayesian Network, Bayes Net
- A graphical model (as a DAG) of probabilistic relationships among a set of random variables
- Nodes are variables, links represent direct influence of one variable on another
- Nodes have associated prior probabilities or Conditional Proability Tables (CPTs)




source

# Why? Three (Four) kinds of reasoning

BBNs support three main kinds of reasoning:

- **Predicting** conditions given predispositions
- **Diagnosing** conditions given symptoms (and predisposing)
- **Explaining** a condition by one or more predispositions

To which we can add a fourth:

- **Deciding** on an action based on probabilities of the conditions

# Recall Bayes Rule

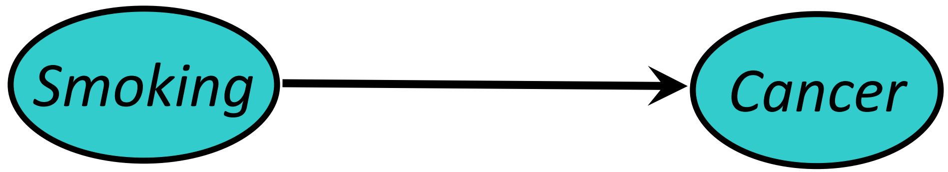
$$P(H,E) = P(H \mid E)P(E) = P(E \mid H)P(H)$$

$$\boxed{P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}}$$

Note symmetry: can compute probability of a ***hypothesis given its evidence*** as well as probability of ***evidence given hypothesis***

# Simple Bayesian Network

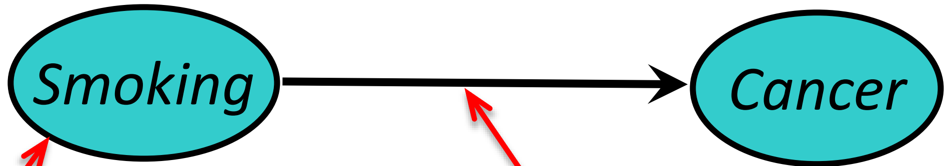
$S \in \{no, light, heavy\}$




$C \in \{none, benign, malignant\}$

# Simple Bayesian Network

$S \in \{no, light, heavy\}$



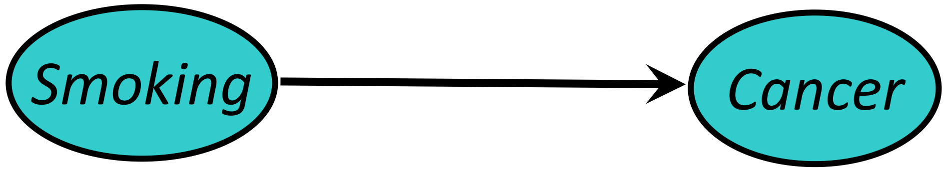

$C \in \{none, benign, malignant\}$

Nodes represent variables

Links represent *"causal"* relations

# Simple Bayesian Network

$S \in \{no, light, heavy\}$




$C \in \{none, benign, malignant\}$

**Prior probability of S**

| | |
|---|---|
| *P( S=no)* | 0.80 |
| *P( S=light)* | 0.15 |
| *P( S=heavy)* | 0.05 |

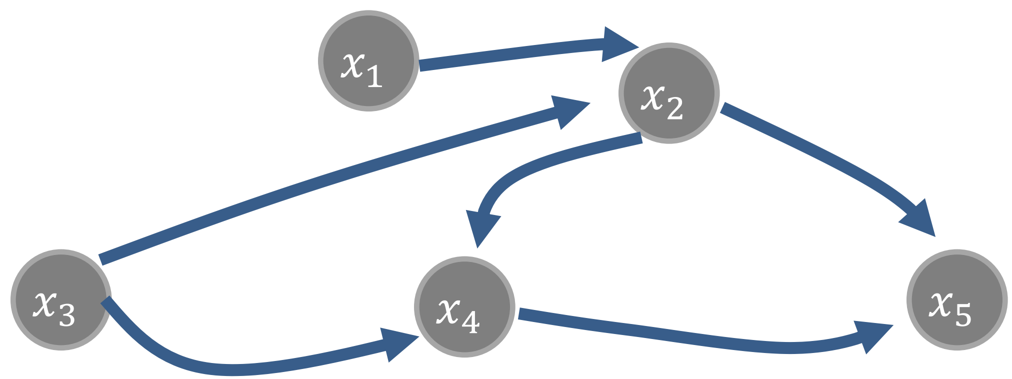
Nodes with no in-links have prior probabilities

**Conditional distribution of S and C**

| *Smoking=* | *no* | *light* | *heavy* |
|---|---|---|---|
| *C=none* | 0.96 | 0.88 | 0.60 |
| *C=benign* | 0.03 | 0.08 | 0.25 |
| *C=malignant* | 0.01 | 0.04 | 0.15 |

Nodes with in-links have joint probability distributions

# Bayesian Networks: Directed Acyclic Graphs

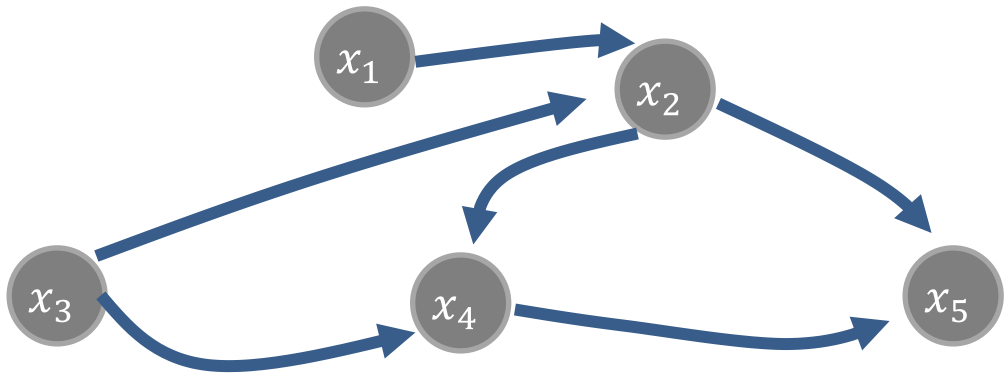



$$p(x_1, x_2, x_3, \ldots, x_N) = \prod_i p(x_i \mid \pi(x_i))$$

topological sort

"parents of"

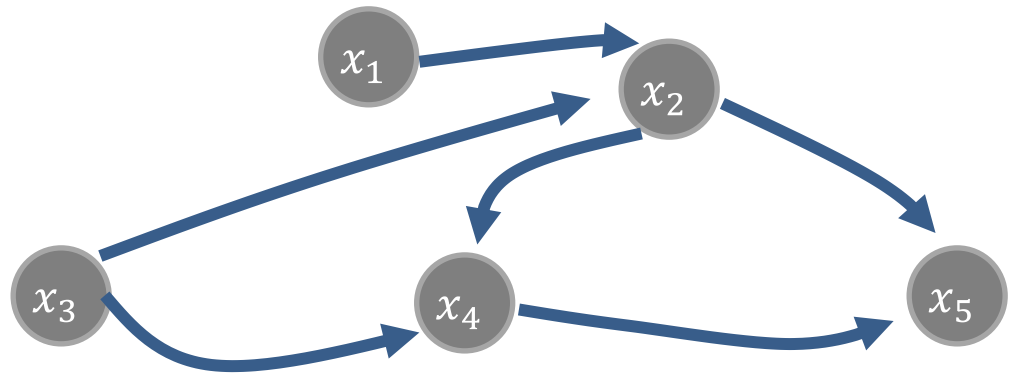
# Bayesian Networks: Directed Acyclic Graphs




$$p(x_1, x_2, x_3, \dots, x_N) = \prod_i p(x_i \mid \pi(x_i))$$

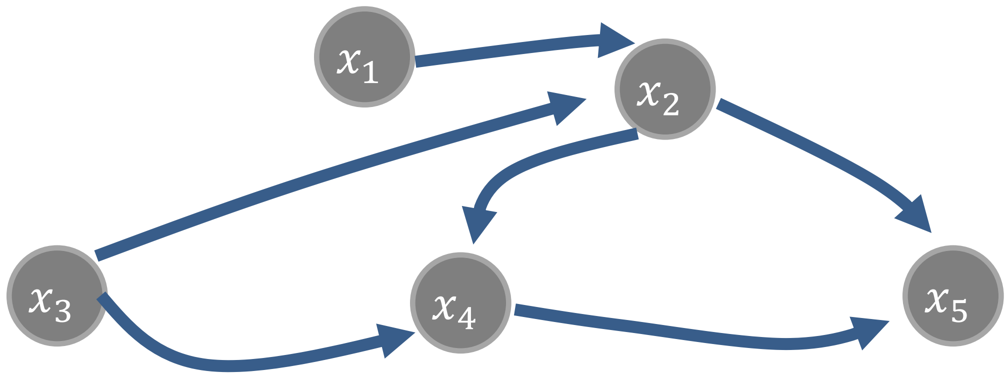
$$p(x_1, x_2, x_3, x_4, x_5) = ???$$

# Bayesian Networks: Directed Acyclic Graphs




$$p(x_1, x_2, x_3, x_4, x_5) =$$
$$p(x_1)p(x_3)p(x_2|x_1, x_3)p(x_4|x_2, x_3)p(x_5|x_2, x_4)$$

# Bayesian Networks: Directed Acyclic Graphs

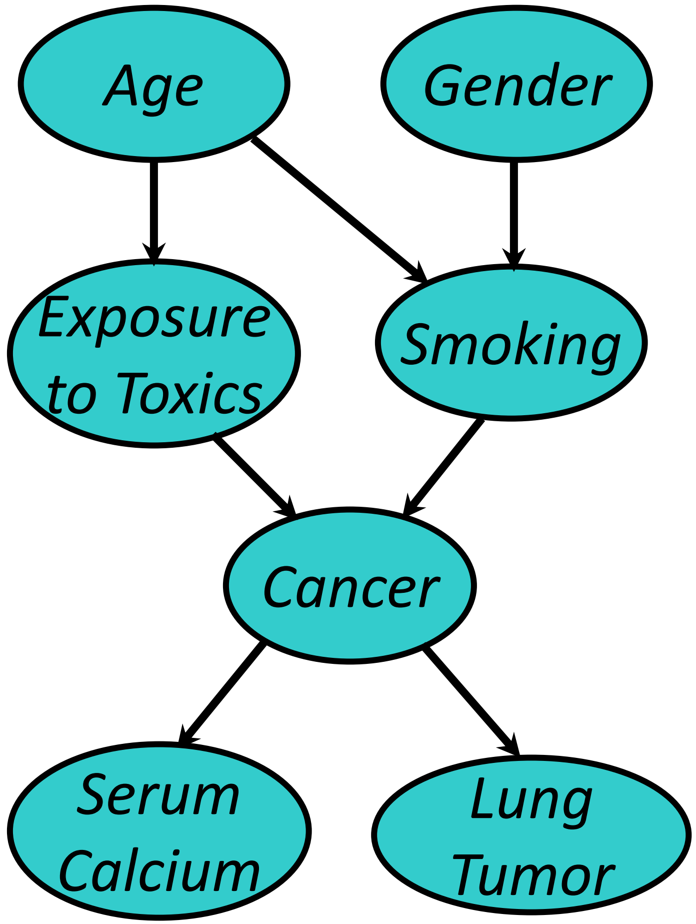



$$p(x_1, x_2, x_3, \dots, x_N) = \prod_i p(x_i \mid \pi(x_i))$$

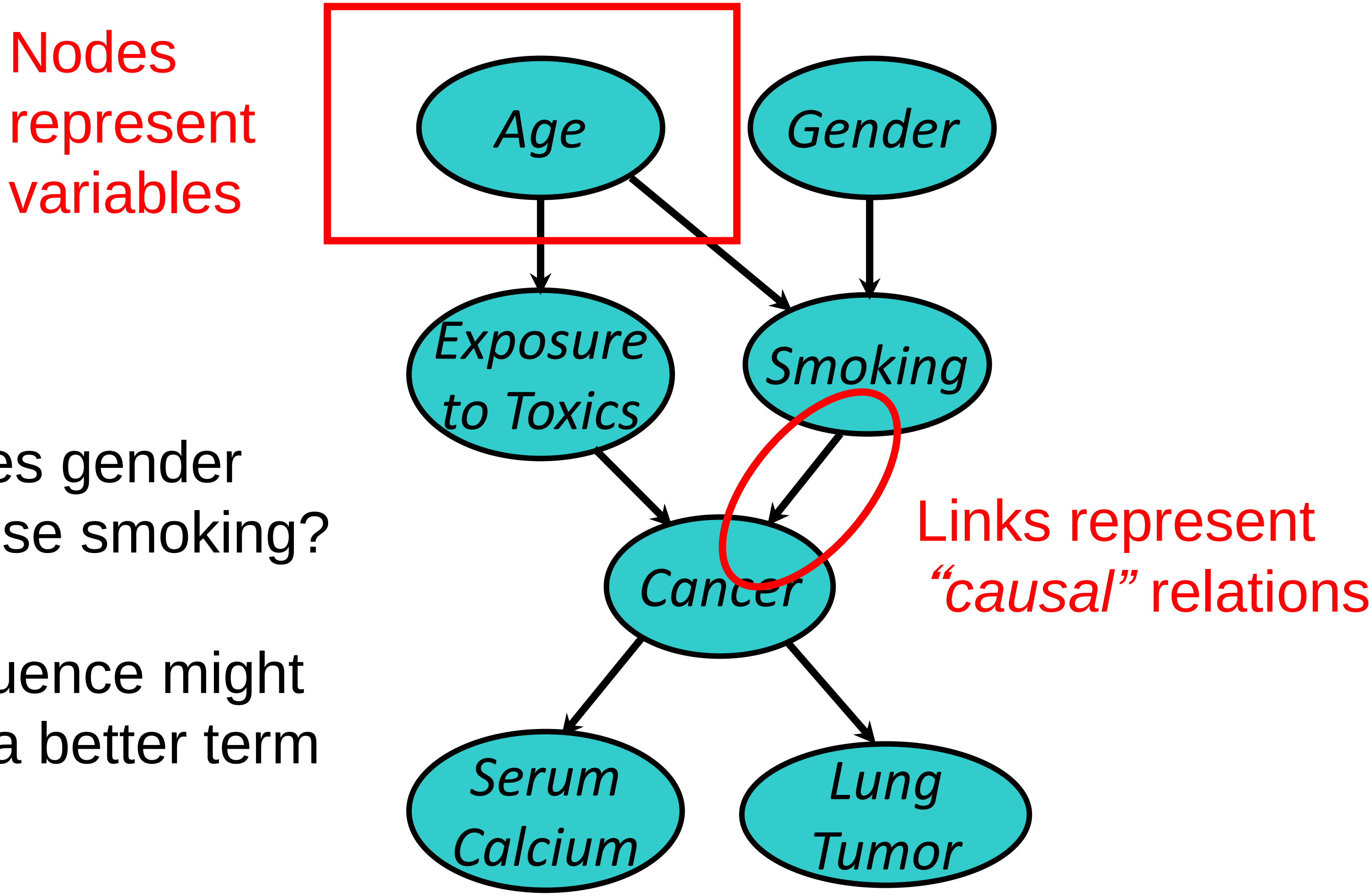
exact inference in general DAGs is NP-hard

inference in trees can be exact

# More Complex Bayesian Network

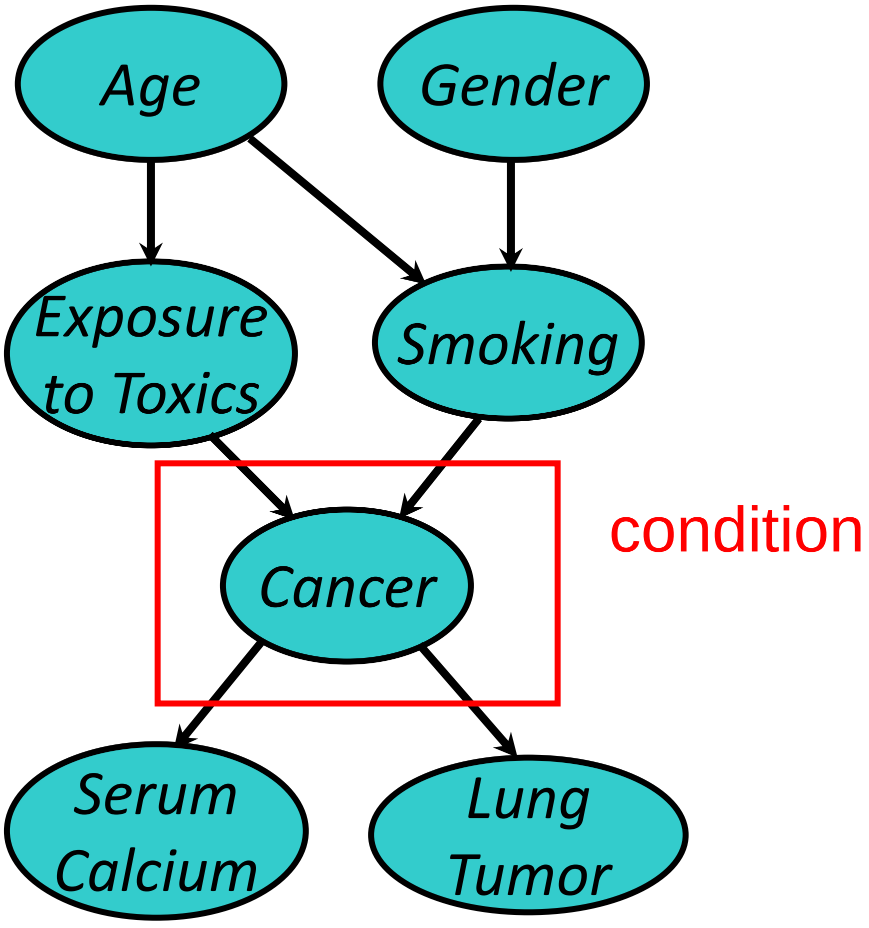


# More Complex Bayesian Network

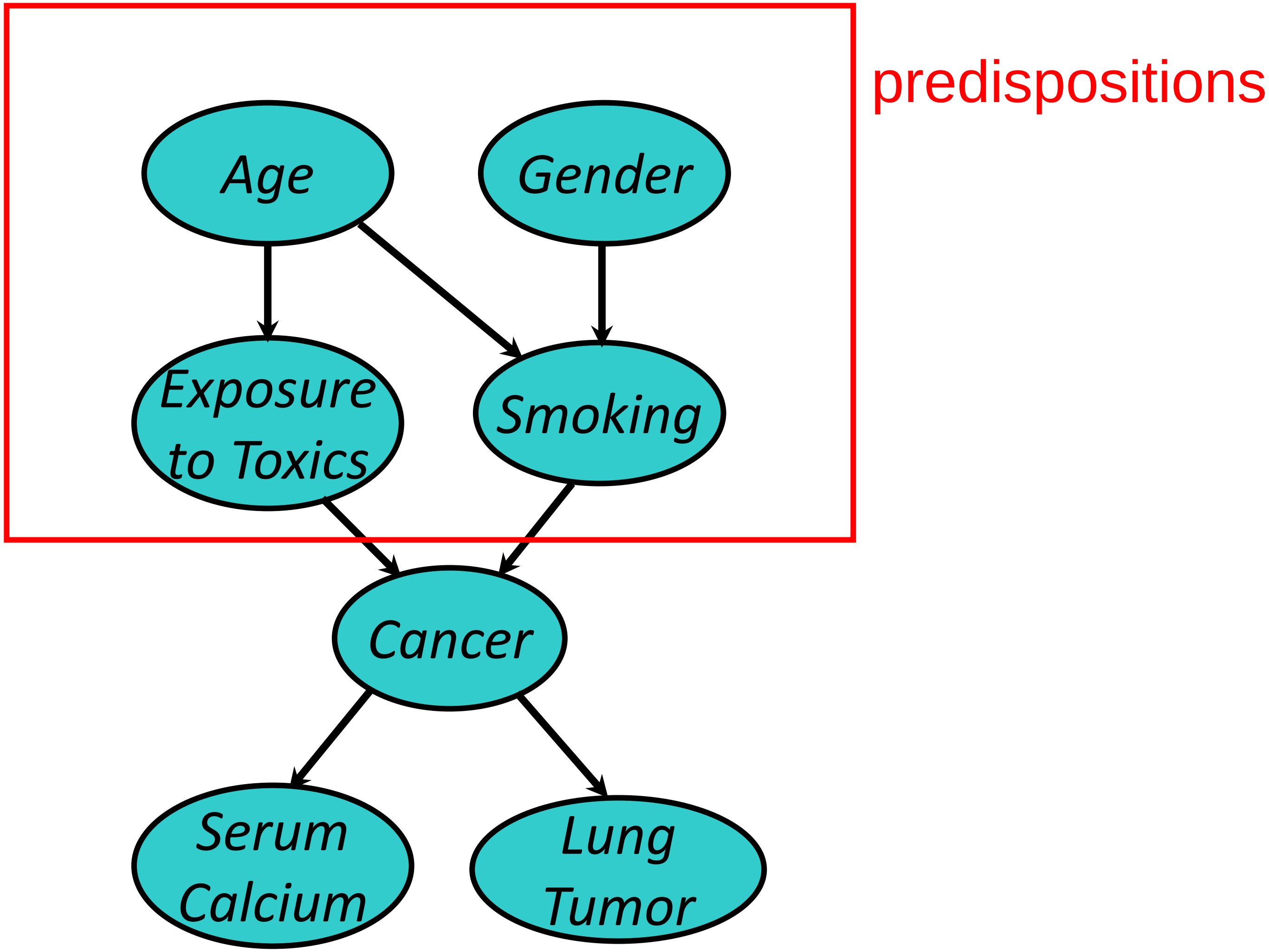



- Does gender cause smoking?
- Influence might be a better term

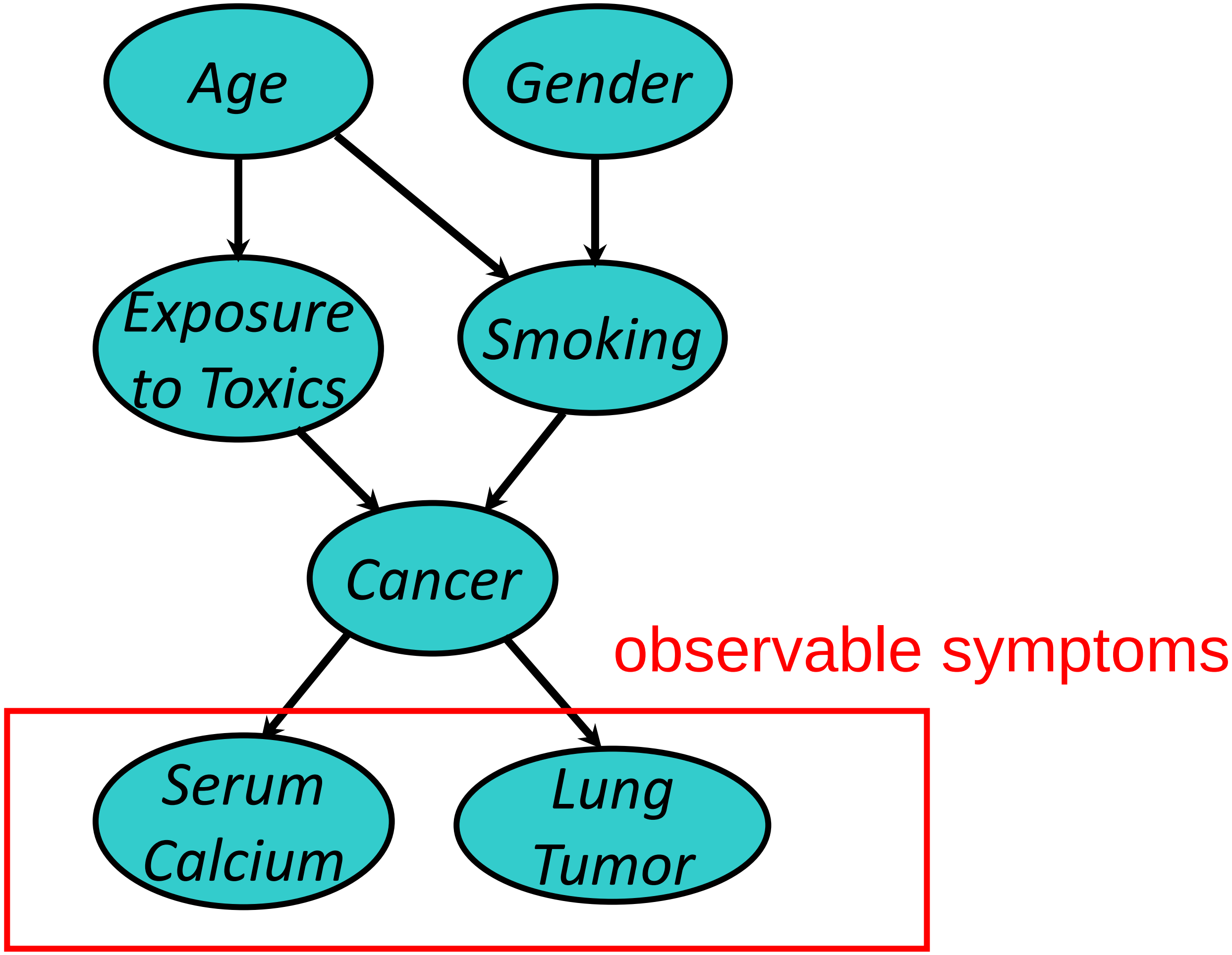
# More Complex Bayesian Network



# More Complex Bayesian Network

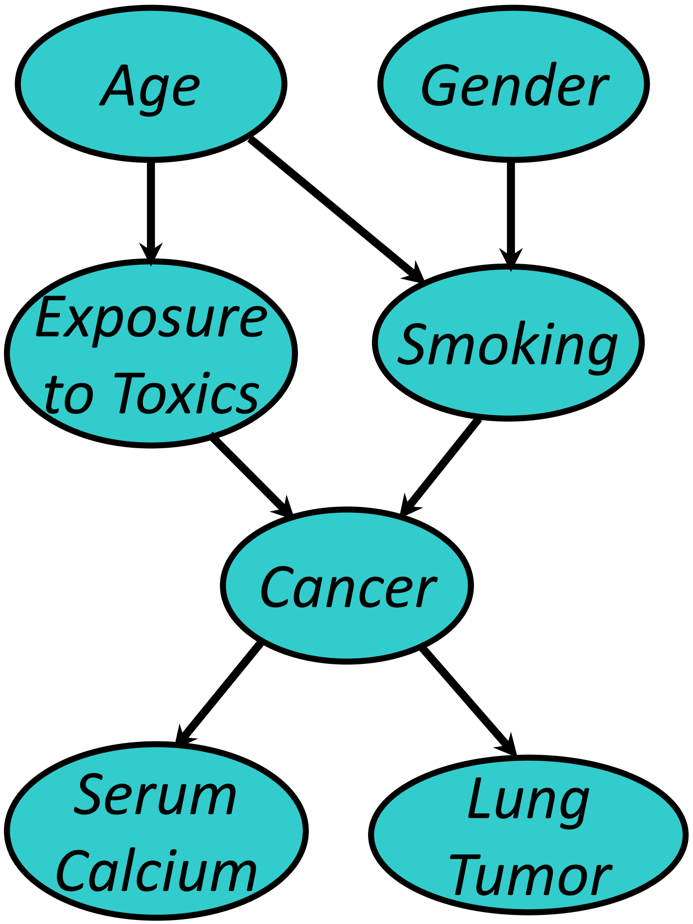


# More Complex Bayesian Network



# More Complex Bayesian Network

Can we predict likelihood of **lung tumor** given values of other 6 variables?




- Model has 7 variables
- Complete joint probability distribution will have 7 dimensions!
- Too much data required ☹
- BBN simplifies: a node has a CPT with data on itself & parents in graph

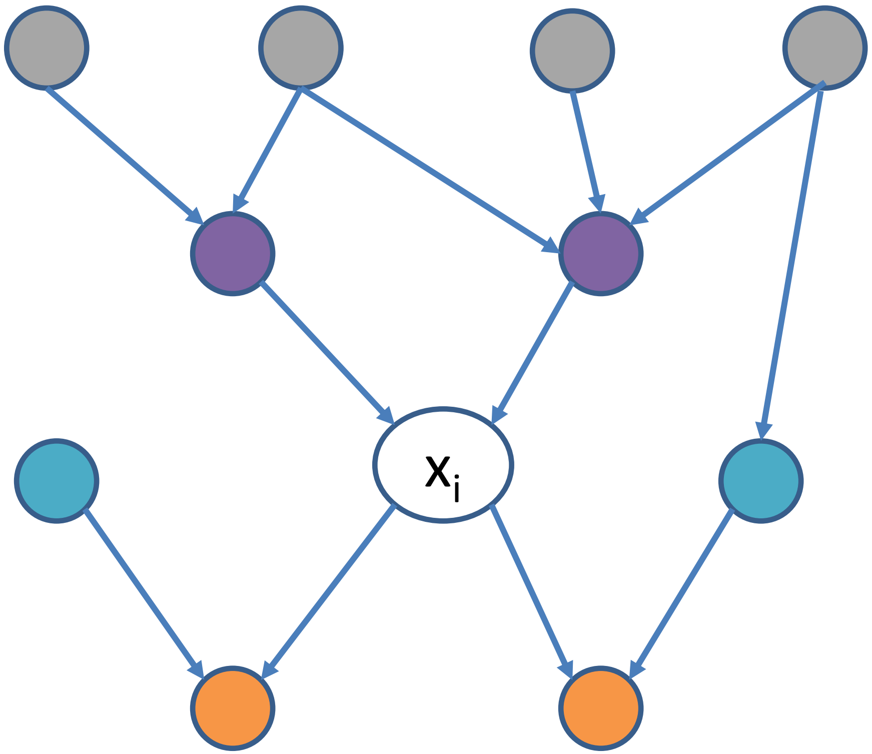
# Independence & Conditional Independence in BBNs

Read these independence relationships right from the graph!

There are two common concepts that can help:

1. Markov blanket
2. D-separation (not covering)

# Markov Blanket

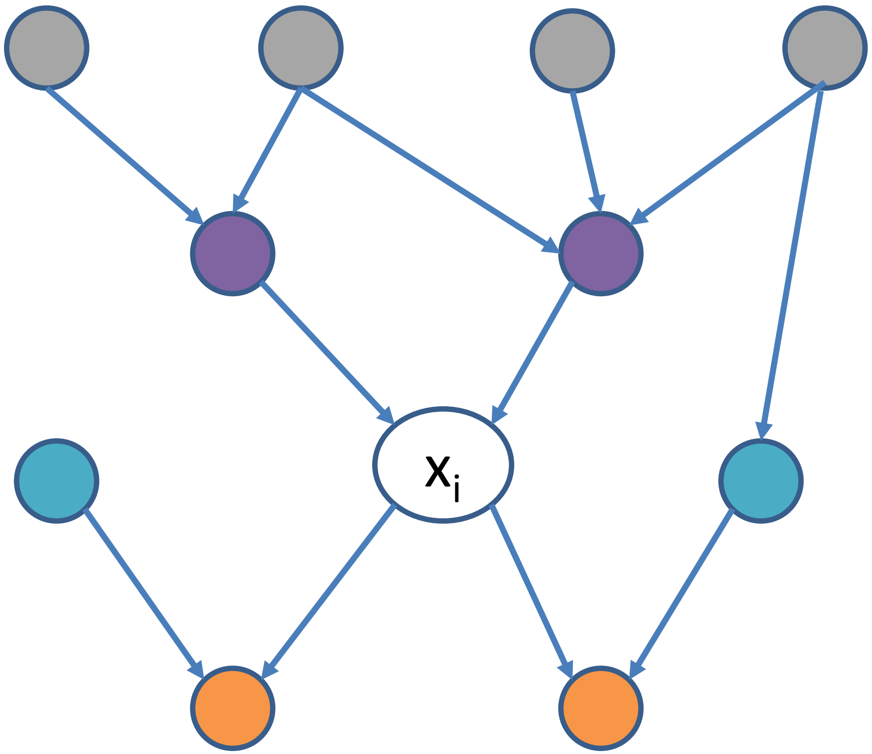



Markov blanket of a node x is its parents, children, and children's parents

The **Markov Blanket** of a node $x_i$ the set of nodes needed to form the complete conditional for a variable $x_i$

*(in this example, shading does not show observed/latent)*

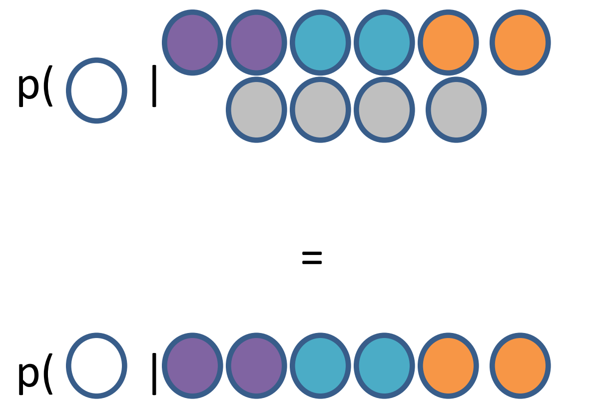
# Markov Blanket




Markov blanket of a node x is its parents, children, and children's parents

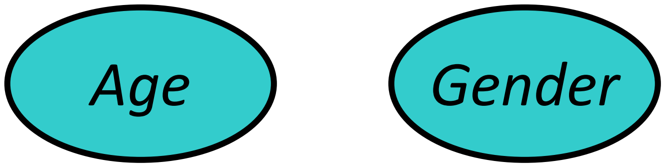
*(in this example, shading does not show observed/latent)*

The **Markov Blanket** of a node $x_i$ the set of nodes needed to form the complete conditional for a variable $x_i$




Given its Markov blanket, a node is conditionally independent of all other nodes in the BN

# Independence




*Age* and *Gender* are independent.

$$P(A,G) = P(G) * P(A)$$

There is no path between them in the graph

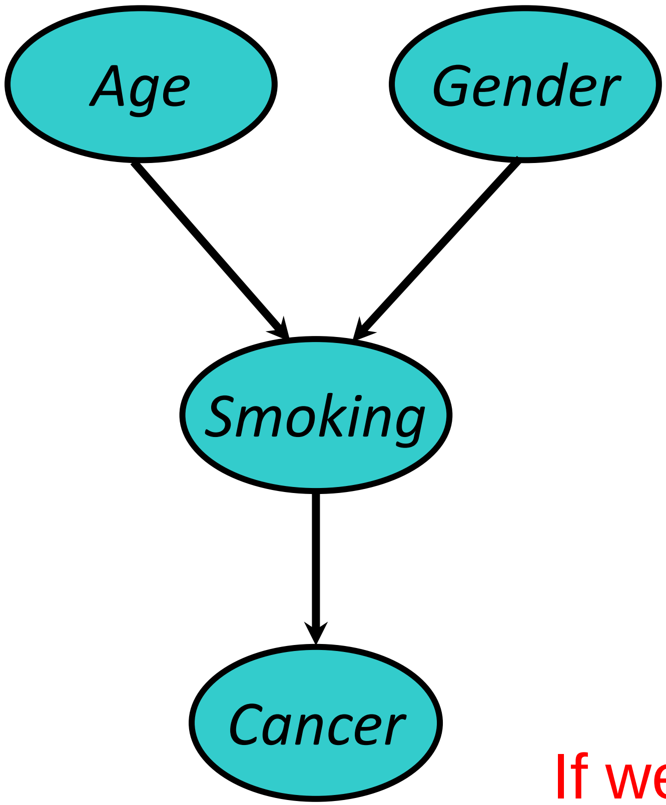
$$P(A \mid G) = P(A)$$

$$P(G \mid A) = P(G)$$

$$P(A,G) = P(G|A)\, P(A) = P(G)P(A)$$

$$P(A,G) = P(A|G)\, P(G) = P(A)P(G)$$

# Conditional Independence

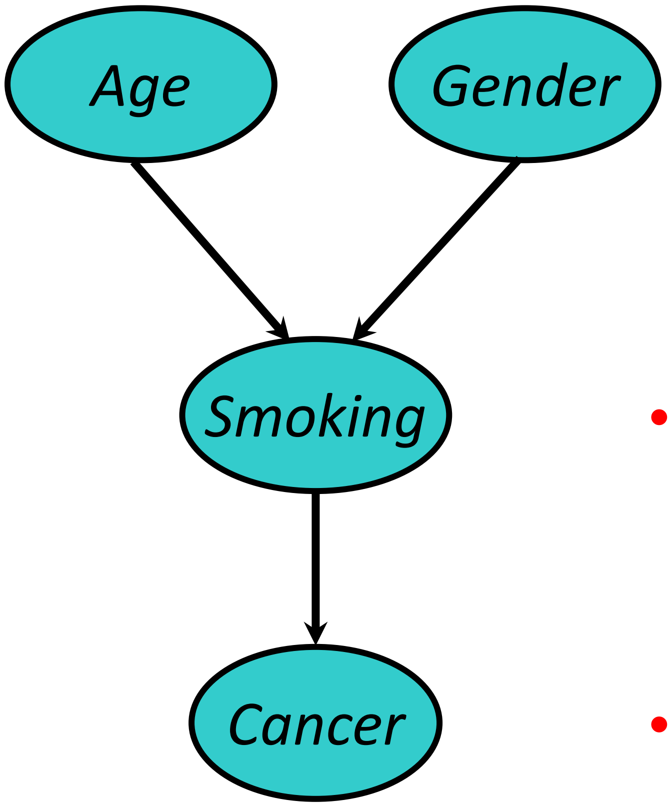



*Cancer* is independent of *Age* and *Gender* given *Smoking*

$$P(C \mid A,G,S) = P(C \mid S)$$

If we know value of smoking, no need to know values of age or gender

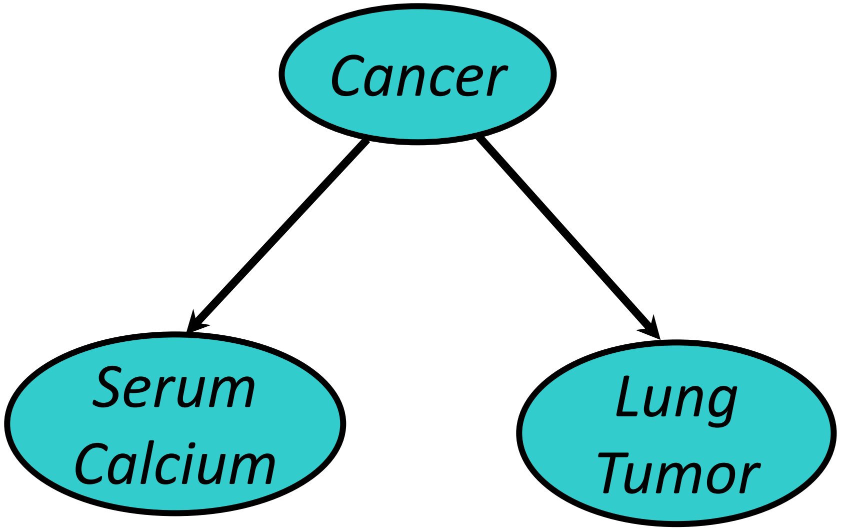
# Conditional Independence




*Cancer* is independent of *Age* and *Gender* given *Smoking*

- Instead of one big CPT with 4 variables, we have two smaller CPTs with 3 and 2 variables
- If all variables binary: 12 models ($2^3 + 2^2$) rather than 16 ($2^4$)

# Conditional Independence: Naïve Bayes



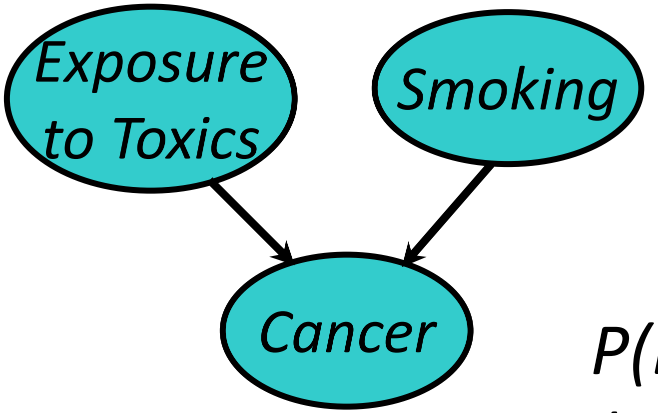

*Serum Calcium* and *Lung Tumor* are dependent

*Serum Calcium* is independent of *Lung Tumor*, given *Cancer*

$$P(L \mid SC, C) = P(L \mid C)$$
$$P(SC \mid L, C) = P(SC \mid C)$$

[Naïve Bayes](#) assumption: evidence (e.g., symptoms) independent given disease; easy to combine evidence

# Explaining Away



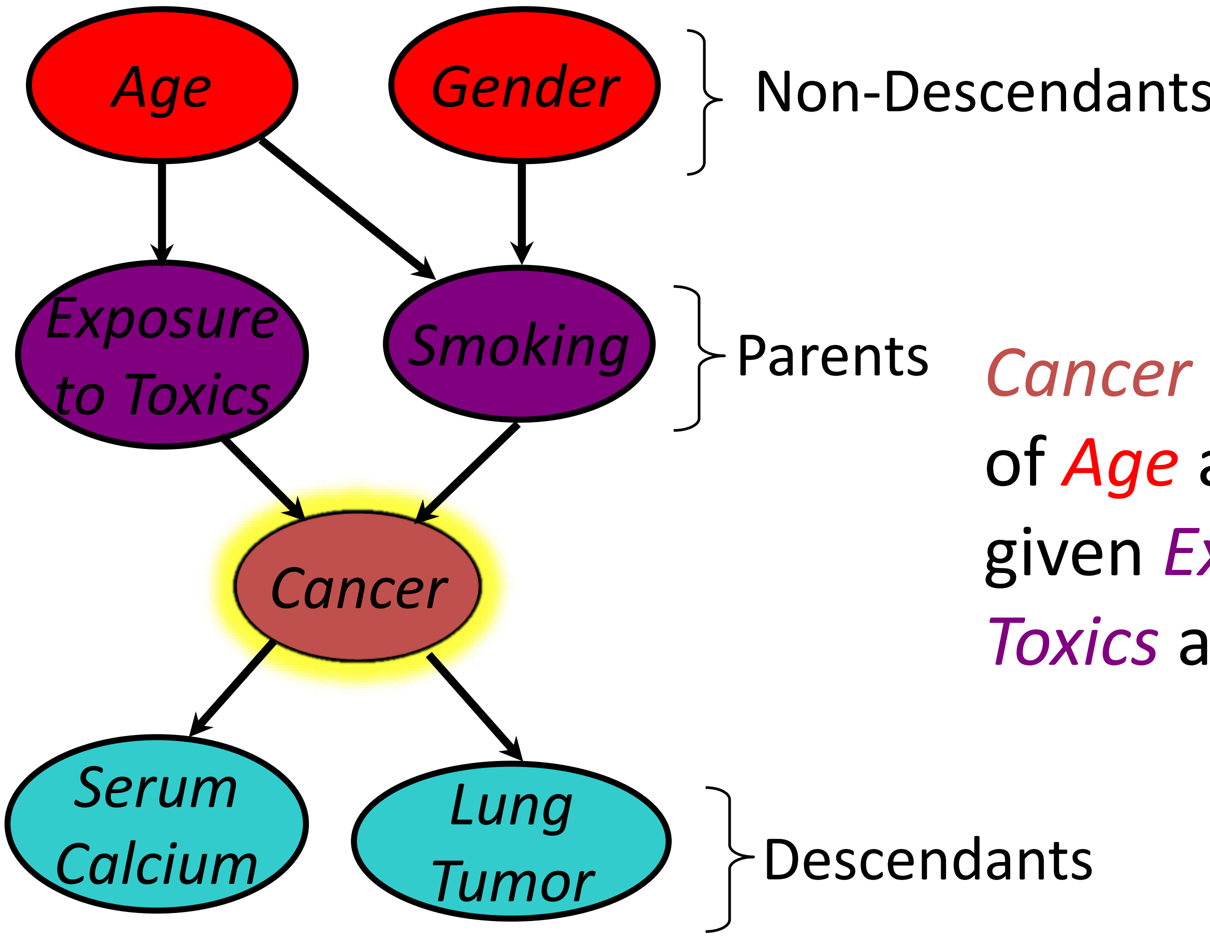

*Exposure to Toxics* and *Smoking* are independent

*Exposure to Toxics* is **dependent** on *Smoking*, given *Cancer*

*P(E=heavy | C=malignant) > P(E=heavy | C=malignant, S=heavy)*

- *Explaining away:* reasoning pattern where confirmation of one cause reduces need to invoke alternatives
- Essence of Occam's Razor (prefer hypothesis with fewest assumptions)
- Relies on independence of causes

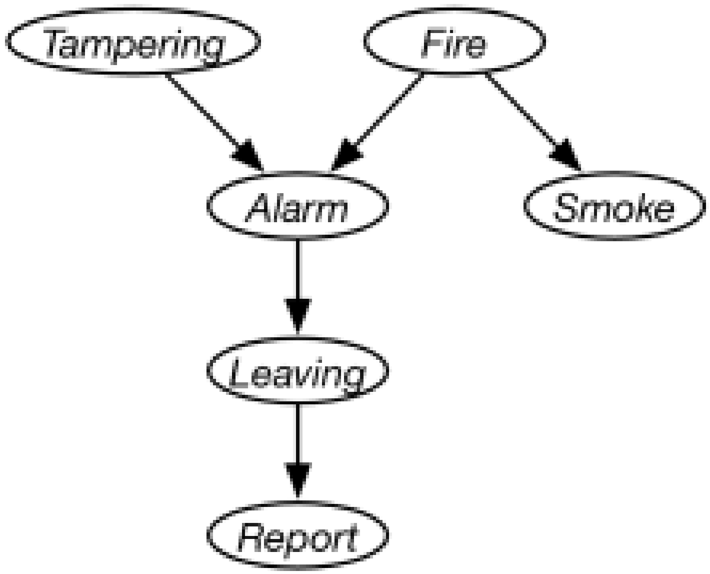
# Conditional Independence




*Cancer* is independent of *Age* and *Gender* given *Exposure to Toxics* and *Smoking*.

# Example from the Book: 8.15

http://artint.info/2e/html/ArtInt2e.Ch8.S3.SS2.html




Some questions:

1. What's the joint factorization? That is, simplify the joint distribution

   p(F, T, A, S, L, R)

2. Are A & S independent?
3. Are there any nodes that make A & S conditionally independent?
4. How many different conditional distributions do we need?

Advanced topic

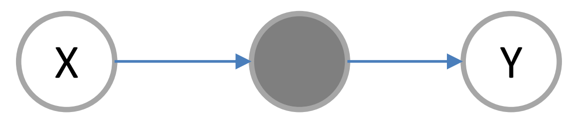
# D-Separation: Testing for Conditional Independence

Variables X & Y are conditionally independent given Z if all (undirected) paths from (any variable in) X to (any variable in) Y are d-separated by Z

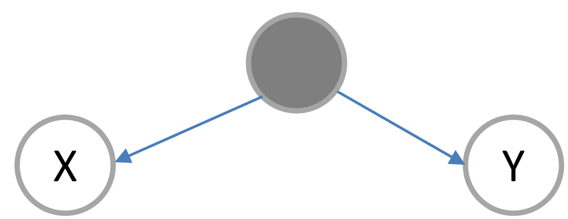
## d-separation

X & Y are d-separated if for **all** paths P, one of the following is true:

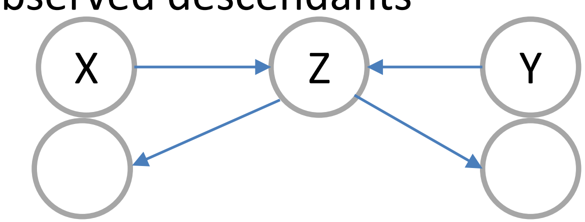
P has a chain with an observed middle node




P has a fork with an observed parent node




P includes a “v-structure” or “collider” with all unobserved descendants



Advanced topic

# D-Separation: Testing for Conditional Independence

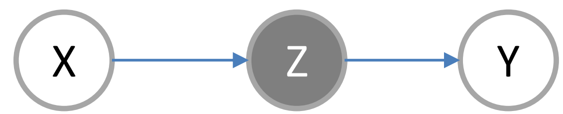
Variables X & Y are conditionally independent given Z if all (undirected) paths from (any variable in) X to (any variable in) Y are d-separated by Z

## d-separation

X & Y are d-separated if for **all** paths P, one of the following is true:

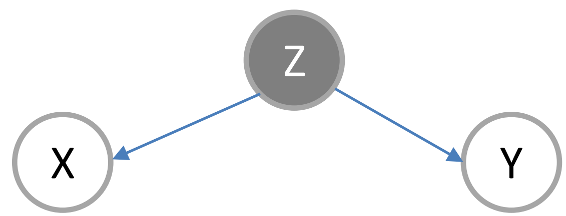
P has a chain with an observed middle node

*observing Z blocks the path from X to Y*



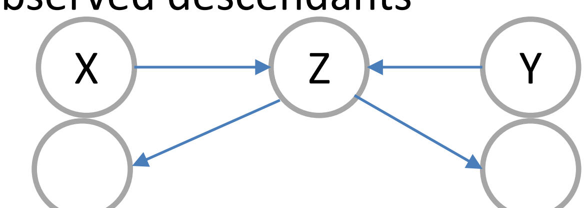

P has a fork with an observed parent node

*observing Z blocks the path from X to Y*




P includes a “v-structure” or “collider” with all unobserved descendants

*not observing Z blocks the path from X to Y*



Advanced topic

# D-Separation: Testing for Conditional Independence

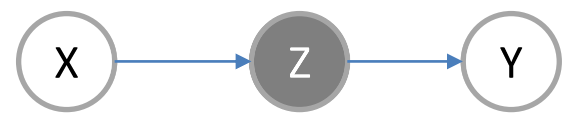
## d-separation

Variables X & Y are conditionally independent given Z if all (undirected) paths from (any variable in) X to (any variable in) Y are d-separated by Z

X & Y are d-separated if for **all** paths P, one of the following is true:

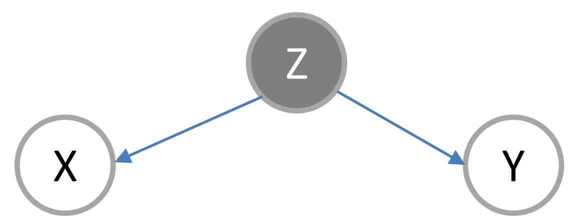
P has a chain with an observed middle node

*observing Z blocks the path from X to Y*




P has a fork with an observed parent node

*observing Z blocks the path from X to Y*

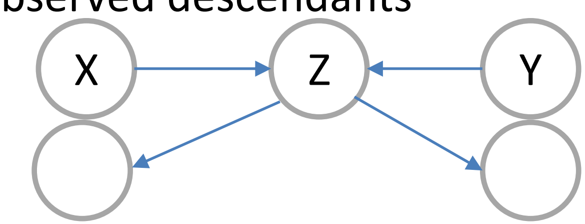



P includes a "v-structure" or "collider" with all unobserved descendants

*not observing Z blocks the path from X to Y*

$$p(x, y, z) = p(x)p(y)p(z|x, y)$$

$$p(x, y) = \sum_z p(x)p(y)p(z|x, y) = p(x)p(y)$$



# Probabilistic Graphical Models

A graph G that represents a probability distribution over random variables $X_1, \dots, X_N$

Graph G = (vertices V, edges E)
Distribution $p(X_1, \dots, X_N)$

Vertices $\leftrightarrow$ random variables
Edges show dependencies among random variables

Two main flavors: *directed* graphical models and *undirected* graphical models (come talk to me)

*Advanced topics*

Advanced topic

# Maxent Models Make a Reappearance

- **features** $f(x, y)$ between x and y that are meaningful;
- **weights** $\theta$ (one per feature) to say how important each feature is; and
- a way to **form probabilities** from $f$ and $\theta$

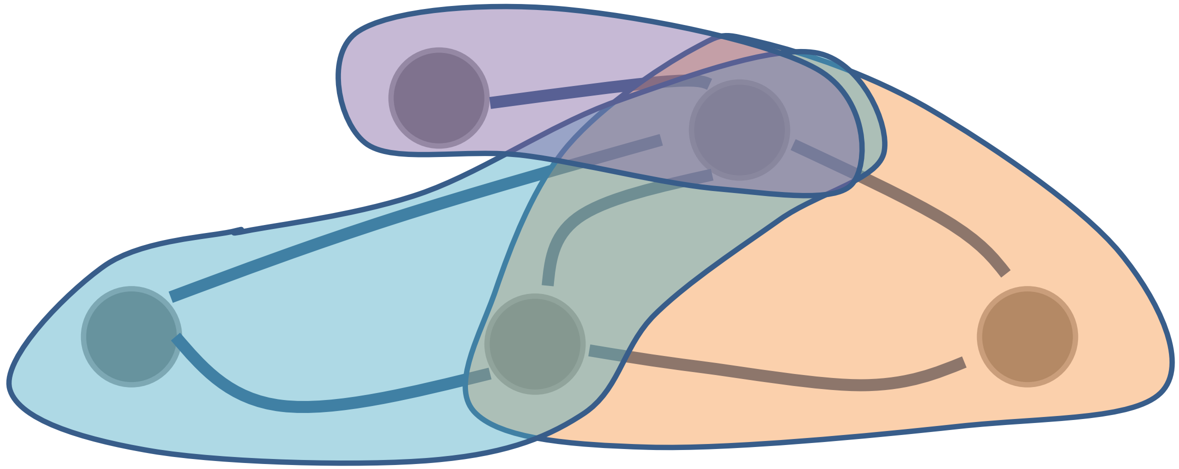
$$p(y \mid x) \propto \exp(\theta^T f(x, y))$$

Advanced topic

# Markov Random Fields: Undirected Graphs

**clique**: subset of nodes, where nodes are pairwise connected

**maximal clique**: a clique that cannot add a node and remain a clique



$$p(x_1, x_2, x_3, \dots, x_N) = \frac{1}{Z} \prod_{C} \exp(-E_C(x_c))$$

global normalization

maximal cliques

Energy function (reweighted features)

variables part of the clique C

# BBN Construction

The **knowledge acquisition** process for a BBN involves three steps

**KA1**: Choosing appropriate variables

**KA2**: Deciding on the network structure

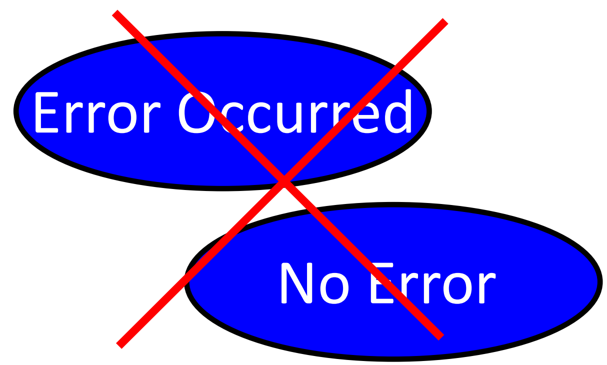
**KA3**: Obtaining data for the conditional probability tables

# KA1: Choosing variables

- Variable values: integers, reals or enumerations
- Variable should have collectively *exhaustive*, *mutually exclusive* values

$$x_1 \vee x_2 \vee x_3 \vee x_4$$

$$\neg(x_i \wedge x_j) \quad i \neq j$$



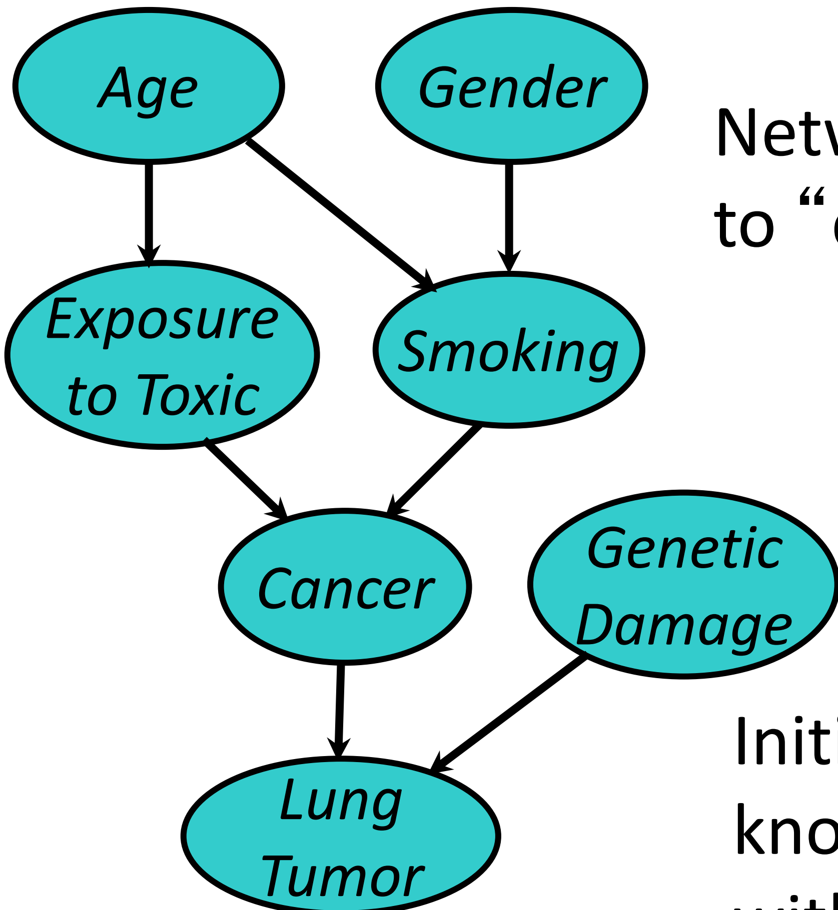

- They should be values, not probabilities


# Heuristic: Knowable in Principle

Example of good variables

- Weather: {Sunny, Cloudy, Rain, Snow}
- Gasoline: Cents per gallon {0,1,2…}
- Temperature: { $\geq 100^\circ$ F , $< 100^\circ$ F}
- User needs help on Excel Charts: {Yes, No}
- User's personality: {dominant, submissive}

# KA2: Structuring




Network structure corresponding to "causality" is usually good.

Initially this uses the designer's knowledge but can be checked with data

# KA3: The Numbers

- For each variable we have a table of probability of its value for values of its **parents**
- For variables w/o parents, we have **prior probabilities**

$$S \in \{no, light, heavy\}$$
$$C \in \{none, benign, malignant\}$$


| smoking priors | |
|---|---|
| no | 0.80 |
| light | 0.15 |
| heavy | 0.05 |

| | smoking | | |
|---|---|---|---|
| **cancer** | **no** | **light** | **heavy** |
| none | 0.96 | 0.88 | 0.60 |
| benign | 0.03 | 0.08 | 0.25 |
| malignant | 0.01 | 0.04 | 0.15 |

# Three (Four) kinds of reasoning

BBNs support three main kinds of reasoning:

- **Predicting** conditions given predispositions
- **Diagnosing** conditions given symptoms (and predisposing)
- **Explaining** a condition by one or more predispositions

To which we can add a fourth:

- **Deciding** on an action based on probabilities of the conditions

# Fundamental Inference & Learning Question

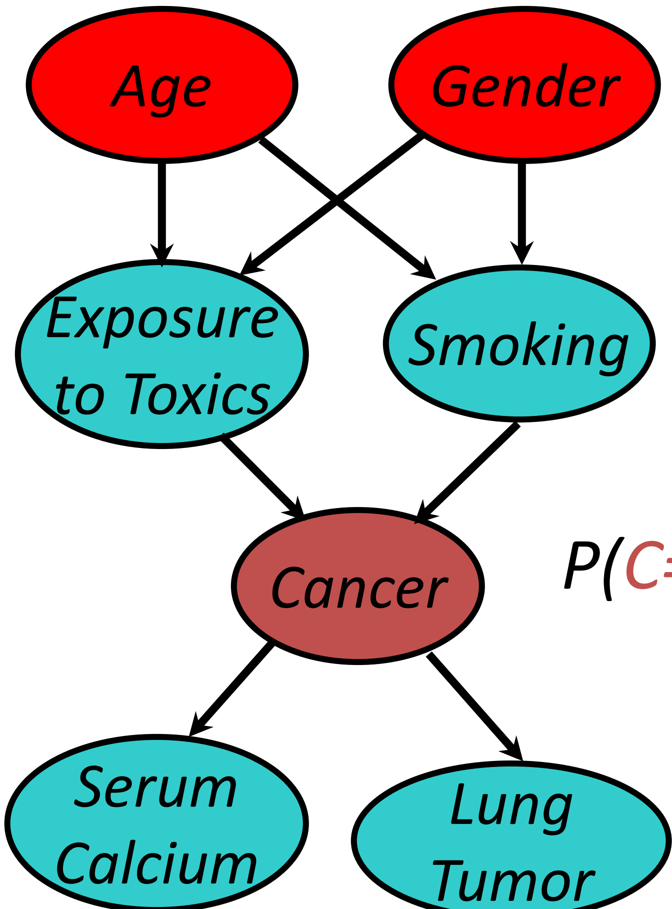
- Compute posterior probability of a node given some other nodes

$$p(Q|x_1, \ldots, x_j)$$

- Some techniques
  - MLE (maximum likelihood estimation)/MAP (maximum a posteriori) [covered 2nd]
  - Variable Elimination [covered 1st]
  - (Loopy) Belief Propagation ((Loopy) BP)
  - Monte Carlo
  - Variational methods
  - …

*Advanced topics*

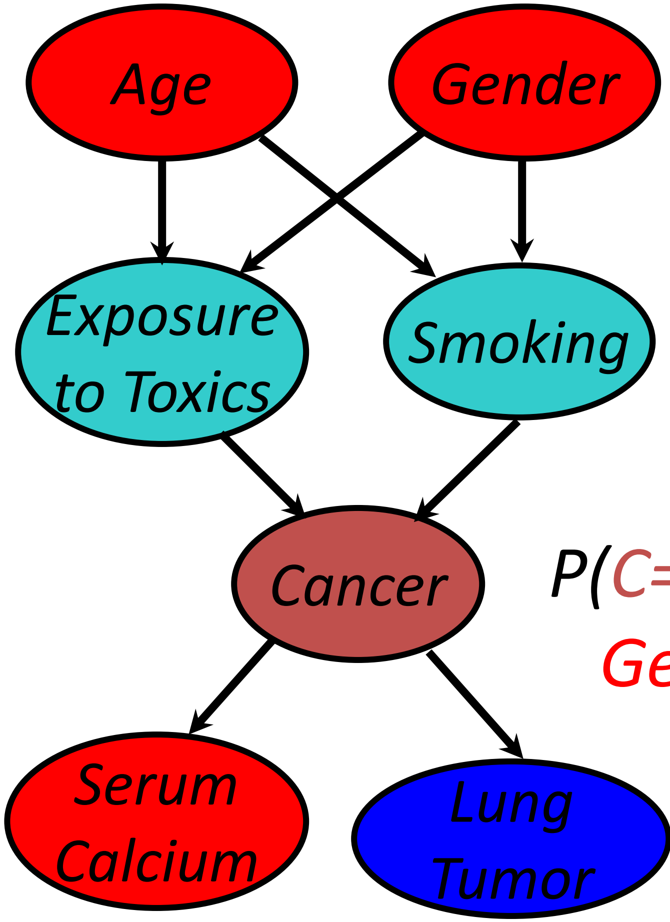
# Predictive Inference




How likely are elderly males to get malignant cancer?

*P(C=malignant | Age>60, Gender=male)*

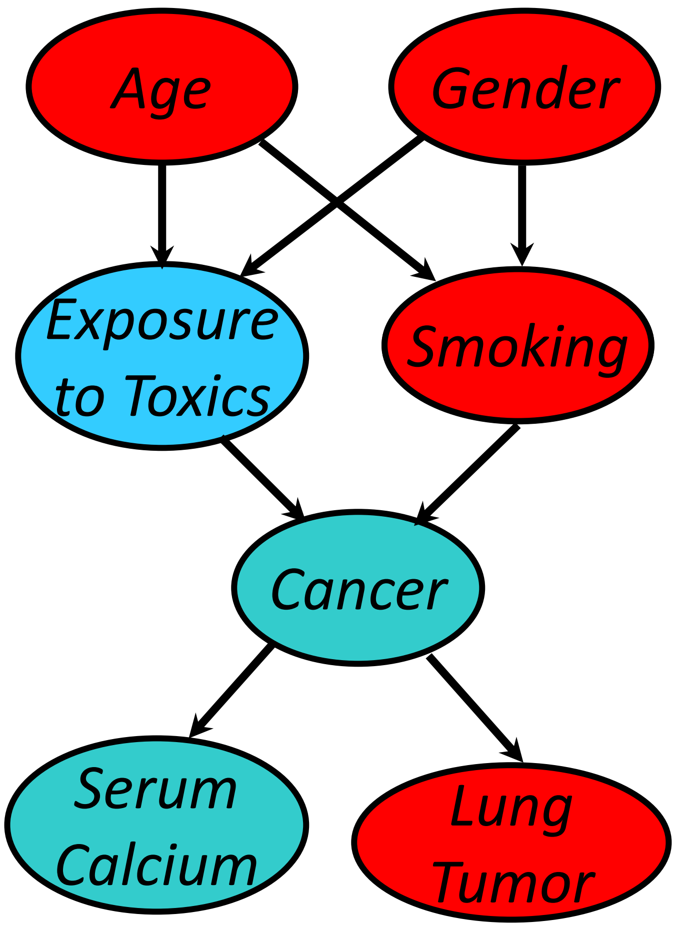
# Predictive and diagnostic combined




How likely is an elderly male patient with high Serum Calcium to have malignant cancer?

*P(C=malignant | Age>60, Gender= male, Serum Calcium = high)*

# Explaining away




- If we see a lung tumor, the probability of heavy smoking and of exposure to toxics both go up

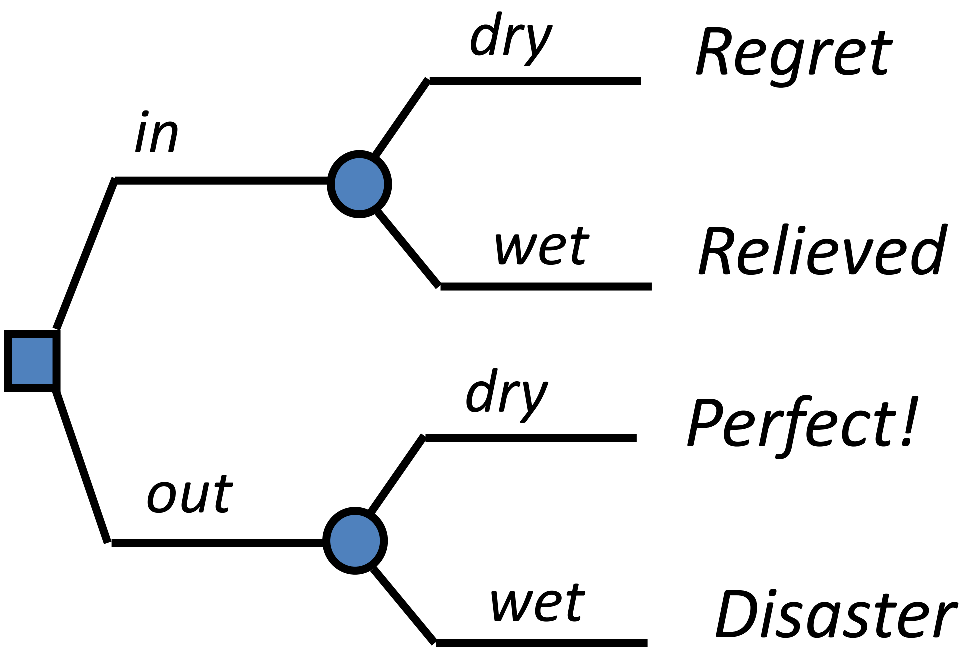
- If we then observe heavy smoking, the probability of exposure to toxics goes back down

# Decision making

- A decision is a medical domain might be a choice of treatment (e.g., radiation or chemotherapy)
- Decisions should be made to maximize expected utility
- View decision making in terms of
  - Beliefs/Uncertainties
  - Alternatives/Decisions
  - Objectives/Utilities

# Decision Problem

Should I have my party inside or outside?



# Decision Making with BBNs

- Today's weather forecast might be either sunny, cloudy or rainy
- Should you take an umbrella when you leave?
- Your decision depends only on the forecast
  - The forecast "depends on" the actual weather
- Your satisfaction depends on your decision and the weather
  - Assign a utility to each of four situations: (rain|no rain) x (umbrella, no umbrella)

# Decision Making with BBNs

- Extend BBN framework to include two new kinds of nodes: **decision** and **utility**
- **Decision** node computes the expected utility of a decision given its parent(s) (e.g., forecast) and a valuation
- **Utility** node computes utility value given its parents, e.g. a decision and weather
  - Assign utility to each situations: (rain|no rain) x (umbrella, no umbrella)
  - Utility value assigned to each is probably subjective

# Fundamental Inference & Learning Question

- Compute posterior probability of a node given some other nodes

$$p(Q|x_1, \dots, x_j)$$

- Some techniques
  - MLE (maximum likelihood estimation)/MAP (maximum a posteriori) [covered 2nd]
  - Variable Elimination [covered 1st]
  - (Loopy) Belief Propagation ((Loopy) BP)
  - Monte Carlo
  - Variational methods
  - …

*Advanced topics*

# Variable Elimination

- Inference: Compute posterior probability of a node given some other nodes
$$p(Q|x_1, \ldots, x_j)$$
- Variable elimination: An algorithm for exact inference
  – Uses dynamic programming
  – Not necessarily polynomial time!

# Variable Elimination (High-level)

Goal: $p(Q|x_1, \ldots, x_j)$

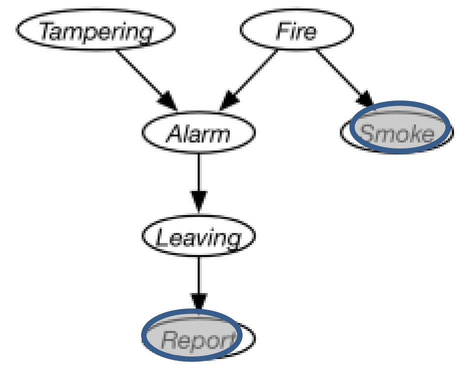
(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

# Variable Elimination: Example

(The word “factor” is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

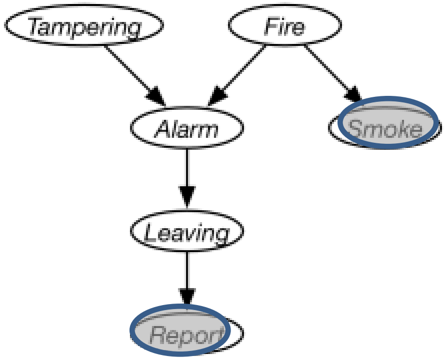



Goal: P(Tampering | Smoke=true ∧ Report=true)

# Variable Elimination: Example

(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.



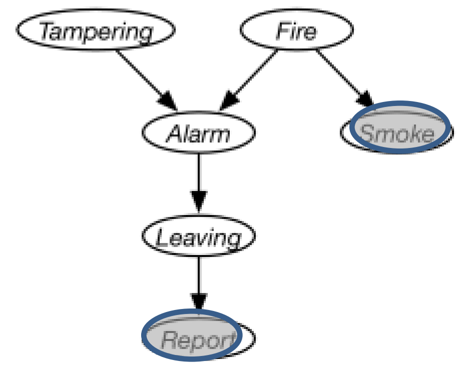

Goal: P(Tampering | Smoke=true ∧ Report=true)

| $Conditional Probability$ | $Factor$ |
|---|---|
| $P(Tampering)$ | $f_0(Tampering)$ |
| $P(Fire)$ | $f_1(Fire)$ |
| $P(Alarm \mid Tampering, Fire)$ | $f_2(Tampering, Fire, Alarm)$ |
| $P(Smoke = yes \mid Fire)$ | $f_3(Fire)$ |
| $P(Leaving \mid Alarm)$ | $f_4(Alarm, Leaving)$ |
| $P(Report = yes \mid Leaving)$ | $f_5(Leaving)$ |

# Variable Elimination: Example

(The word “factor” is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.




Goal: P(Tampering | Smoke=true ∧ Report=true)

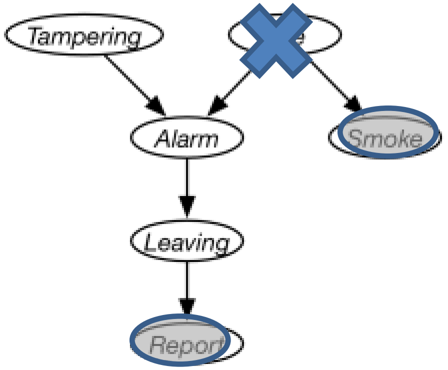
Task: Eliminate Fire

| $Conditional Probability$ | $Factor$ |
|---|---|
| $P(Tampering)$ | $f_0(Tampering)$ |
| $P(Fire)$ | $f_1(Fire)$ |
| $P(Alarm \mid Tampering, Fire)$ | $f_2(Tampering, Fire, Alarm)$ |
| $P(Smoke = yes \mid Fire)$ | $f_3(Fire)$ |
| $P(Leaving \mid Alarm)$ | $f_4(Alarm, Leaving)$ |
| $P(Report = yes \mid Leaving)$ | $f_5(Leaving)$ |

# Variable Elimination: Example

(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.




Goal: P(Tampering | Smoke=true ∧ Report=true)

f1(Fire)
f2(Tampering, Fire, Alarm)
f3(Fire)

f6(Tampering, Alarm) =

$$= \sum_u f_1(\text{Fire} = u) f_2(T, F = u, A) f_3(F = u)$$

$$= \sum_u p(\text{Fire} = u) p(A \mid T, F = u) p(S = y \mid F = u)$$

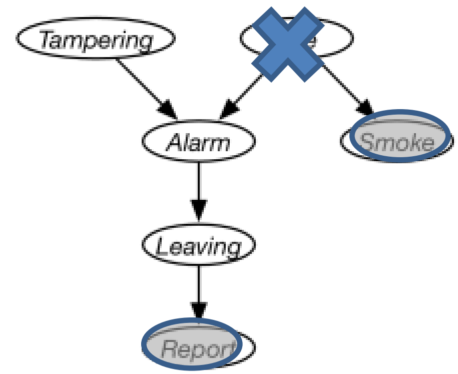
| $Conditional Probability$ | $Factor$ |
|---|---|
| $P(Tampering)$ | $f_0(Tampering)$ |
| $P(Fire)$ | $f_1(Fire)$ |
| $P(Alarm \mid Tampering, Fire)$ | $f_2(Tampering, Fire, Alarm)$ |
| $P(Smoke = yes \mid Fire)$ | $f_3(Fire)$ |
| $P(Leaving \mid Alarm)$ | $f_4(Alarm, Leaving)$ |
| $P(Report = yes \mid Leaving)$ | $f_5(Leaving)$ |

# Variable Elimination: Example

(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

| *ConditionalProbability* | *Factor* |
|---|---|
| $P(Tampering)$ | $f_0(Tampering)$ |
| $P(Fire)$ | $f_1(Fire)$ |
| $P(Alarm \mid Tampering, Fire)$ | $f_2(Tampering, Fire, Alarm)$ |
| $P(Smoke = yes \mid Fire)$ | $f_3(Fire)$ |
| $P(Leaving \mid Alarm)$ | $f_4(Alarm, Leaving)$ |
| $P(Report = yes \mid Leaving)$ | $f_5(Leaving)$ |




Goal: P(Tampering | Smoke=true ∧ Report=true)

f6(Tampering, Alarm) =

$$= \sum_u p(\text{Fire} = u)p(A \mid T, F = u)p(S = y \mid F = u)$$

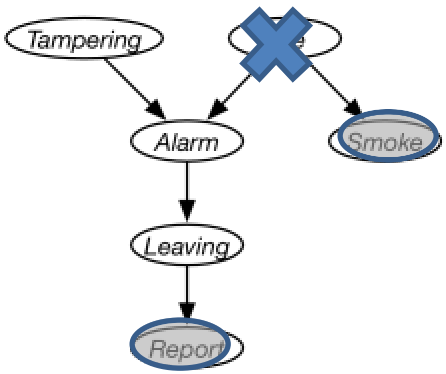
$$= p(\text{Fire} = y)p(A \mid T, F = y)p(S = y \mid F = y) + p(\text{Fire} = n)p(A \mid T, F = n)p(S = y \mid F = n)$$

# Variable Elimination: Example

(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

| $Conditional Probability$ | $Factor$ |
|---|---|
| $P(Tampering)$ | $f_0(Tampering)$ |
| $P(Fire)$ | $f_1(Fire)$ |
| $P(Alarm \mid Tampering, Fire)$ | $f_2(Tampering, Fire, Alarm)$ |
| $P(Smoke = yes \mid Fire)$ | $f_3(Fire)$ |
| $P(Leaving \mid Alarm)$ | $f_4(Alarm, Leaving)$ |
| $P(Report = yes \mid Leaving)$ | $f_5(Leaving)$ |




Goal: P(Tampering | Smoke=true ∧ Report=true)

f6(Tampering, Alarm) =

$$= \sum_u p(\text{Fire} = u) p(A \mid T, F = u) p(S = y \mid F = u)$$

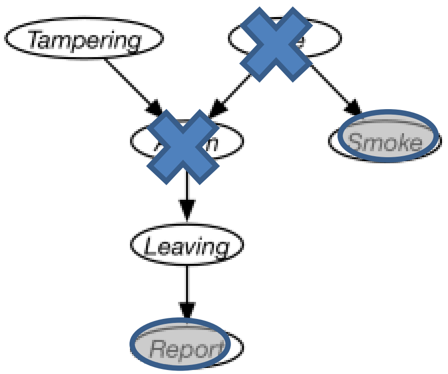
| Tamp. | Alarm | f6 |
|---|---|---|
| Yes | Yes | $p(\text{Fire} = y) p(A = y \mid T = y, F = y) p(S = y \mid F = y) +$ $p(\text{Fire} = n) p(A = y \mid T = y, F = n) p(S = y \mid F = n)$ |
| Yes | No | … |
| No | No | … |
| No | Yes | … |

# Variable Elimination: Example

(The word “factor” is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

| $Conditional Probability$ | $Factor$ |
|---|---|
| $P(Tampering)$ | $f_0(Tampering)$ |
| $P(Fire)$ | $f_1(Fire)$ |
| $P(Alarm \mid Tampering, Fire)$ | $f_2(Tampering, Fire, Alarm)$ |
| $P(Smoke = yes \mid Fire)$ | $f_3(Fire)$ |
| $P(Leaving \mid Alarm)$ | $f_4(Alarm, Leaving)$ |
| $P(Report = yes \mid Leaving)$ | $f_5(Leaving)$ |




Goal: P(Tampering | Smoke=true ∧ Report=true)

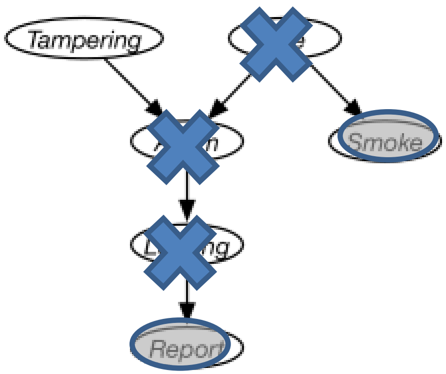
Task: Eliminate Alarm

# Variable Elimination: Example

(The word “factor” is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

| $ConditionalProbability$ | $Factor$ |
|---|---|
| $P(Tampering)$ | $f_0(Tampering)$ |
| $P(Fire)$ | $f_1(Fire)$ |
| $P(Alarm \mid Tampering, Fire)$ | $f_2(Tampering, Fire, Alarm)$ |
| $P(Smoke = yes \mid Fire)$ | $f_3(Fire)$ |
| $P(Leaving \mid Alarm)$ | $f_4(Alarm, Leaving)$ |
| $P(Report = yes \mid Leaving)$ | $f_5(Leaving)$ |




Goal: P(Tampering | Smoke=true ∧ Report=true)

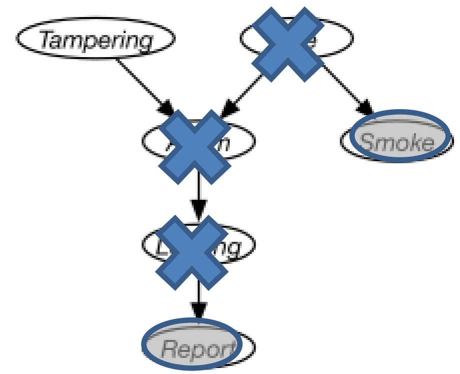
…other computations not shown---see the book…

# Variable Elimination: Example

(The word "factor" is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.

| $Conditional Probability$ | $Factor$ |
|---|---|
| $P(Tampering)$ | $f_0(Tampering)$ |
| $P(Fire)$ | $f_1(Fire)$ |
| $P(Alarm \mid Tampering, Fire)$ | $f_2(Tampering, Fire, Alarm)$ |
| $P(Smoke = yes \mid Fire)$ | $f_3(Fire)$ |
| $P(Leaving \mid Alarm)$ | $f_4(Alarm, Leaving)$ |
| $P(Report = yes \mid Leaving)$ | $f_5(Leaving)$ |




Goal: P(Tampering | Smoke=true ∧ Report=true)

Task: Normalize in order to compute p(Tampering)

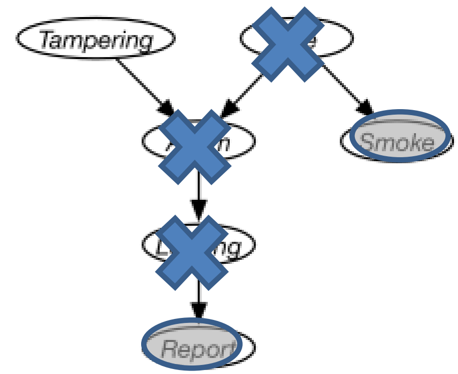
We'll have a single factor f9(Tampering):

$$p(T = u) = \frac{f_9(T = u)}{\sum_v f_9(T = v)}$$

# Variable Elimination: Example

(The word “factor” is used for each CPT.)

1. Pick one of the non-conditioned, MB variables
2. Eliminate this variable by marginalizing (summing) it out from all factors (CPTs) that contain it
3. Go back to 1 until no (MB) variables remain
4. Multiply the remaining factors and normalize.




Goal: P(Tampering | Smoke=true ∧ Report=true)

Task: Normalize in order to compute p(Tampering)

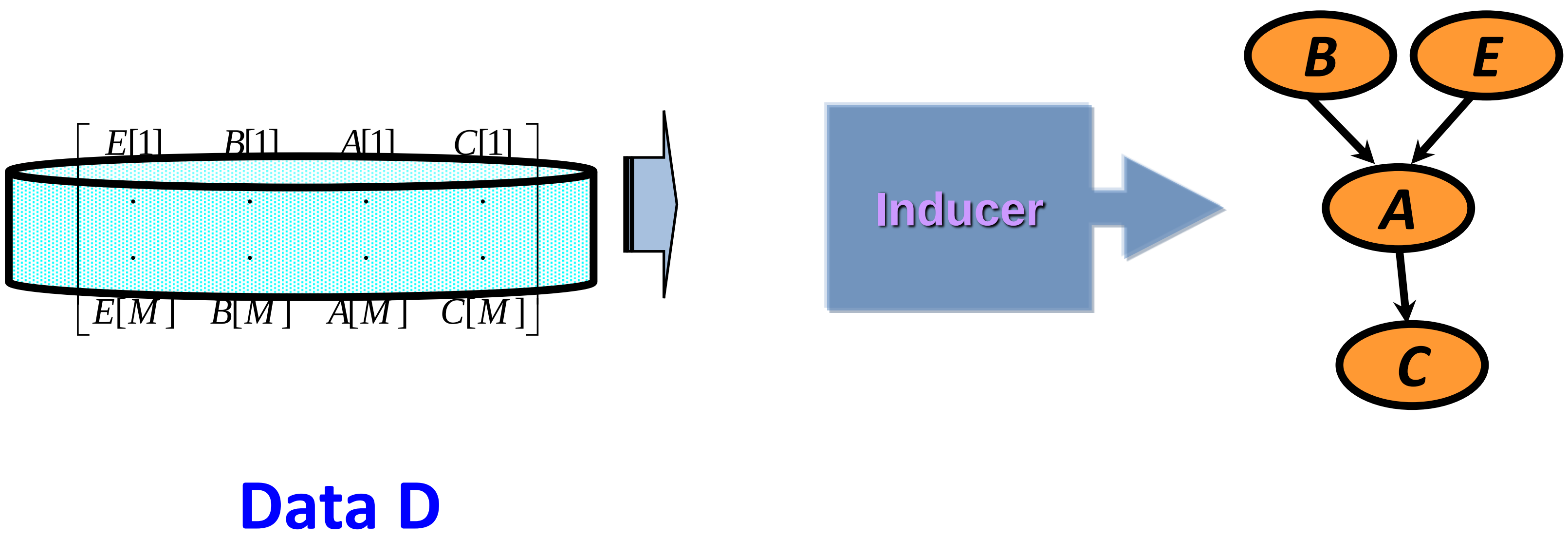
We’ll have a single factor f9(Tampering):

$$p(T=y) = \frac{f_9(T=y)}{f_9(T=y) + f_9(T=n)}$$

| $Conditional Probability$ | $Factor$ |
|---|---|
| $P(Tampering)$ | $f_0(Tampering)$ |
| $P(Fire)$ | $f_1(Fire)$ |
| $P(Alarm \mid Tampering, Fire)$ | $f_2(Tampering, Fire, Alarm)$ |
| $P(Smoke = yes \mid Fire)$ | $f_3(Fire)$ |
| $P(Leaving \mid Alarm)$ | $f_4(Alarm, Leaving)$ |
| $P(Report = yes \mid Leaving)$ | $f_5(Leaving)$ |

# Learning Bayesian networks

- Given training set $\boldsymbol{D} = \{\boldsymbol{x}[1], \ldots, \boldsymbol{x}[M]\}$
- Find graph that best matches ***D***
  - model selection
  - parameter estimation



# Learning Bayesian Networks

- Describe a BN by specifying its (1) structure and (2) conditional probability tables (CPTs)
- Both can be learned from data, but
  – learning structure much harder than learning parameters
  – learning when some nodes are hidden, or with missing data harder still
- Four cases:

| *Structure* | *Observability* | *Method* |
| --- | --- | --- |
| Known | Full | Maximum Likelihood Estimation |
| Known | Partial | EM (or gradient ascent) |
| Unknown | Full | Search through model space |
| Unknown | Partial | EM + search through model space |

# Variations on a theme

- **Known structure, fully observable**: only need to do parameter estimation
- **Unknown structure, fully observable:** do heuristic search through structure space, then parameter estimation
- **Known structure, missing values:** use expectation maximization (EM) to estimate parameters
- **Known structure, hidden variables:** apply adaptive probabilistic network (APN) techniques
- **Unknown structure, hidden variables:** too hard to solve!

# Fundamental Inference Question

- Compute posterior probability of a node given some other nodes

$$p(Q|x_1, \dots, x_j)$$

- Some techniques
  - MLE (maximum likelihood estimation)/MAP (maximum a posteriori) [covered 2nd]
  - Variable Elimination [covered 1st]
  - (Loopy) Belief Propagation ((Loopy) BP)
  - Monte Carlo
  - Variational methods
  - …

*Advanced topics*

# Parameter estimation

- Assume known structure
- Goal: estimate BN parameters $\theta$
  - entries in local probability models, P(X | Parents(X))
- A parameterization $\theta$ is good if it is likely to generate the observed data:

$$L(\theta : D) = P(D \mid \theta) = \prod_m P(x[m] \mid \theta)$$

i.i.d. samples

- Maximum Likelihood Estimation (MLE) Principle: Choose $\theta^*$ so as to maximize *L*

# Parameter estimation II

- The likelihood **decomposes** according to the structure of the network
  → we get a separate estimation task for each parameter
- The MLE (maximum likelihood estimate) solution for **discrete** data & RV values:
  - for each value *x* of a node *X*
  - and each instantiation ***u*** of *Parents(X)*

$$\theta^*_{x|\mathbf{u}} = \frac{N(x,\mathbf{u})}{N(\mathbf{u})}$$

sufficient statistics

  - Just need to collect the counts for every combination of parents and children observed in the data
  - MLE is equivalent to an assumption of a uniform prior over parameter values

# Learning: Maximum Likelihood Estimation (MLE)

Core concept in intro statistics:

- Observe some data $\mathcal{X}$
- Compute some distribution $g(\mathcal{X})$ to {predict, explain, generate} $\mathcal{X}$
- Assume $g$ is controlled by parameters $\phi$, i.e., $g_\phi(\mathcal{X})$
  - Sometimes written $g(\mathcal{X}; \phi)$
- Learning appropriate value(s) of $\phi$ allows you to **GENERALIZE** about $\mathcal{X}$

# Learning: Maximum Likelihood Estimation (MLE)

Central to machine learning:

- Observe some data $(\mathcal{X}, \mathcal{Y})$
- Compute some function $f(\mathcal{X})$ to {predict, explain, generate} $\mathcal{Y}$
- Assume $f$ is controlled by parameters $\theta$, i.e., $f_\theta(\mathcal{X})$
  - Sometimes written $f(\mathcal{X}; \theta)$

# Learning Parameters for the Die Model

$$p(w_1, w_2, \dots, w_N) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

maximize (log-) likelihood to learn the probability parameters

Q: Why is maximizing log-likelihood a reasonable thing to do?

# Learning Parameters for the Die Model

$$p(w_1, w_2, \dots, w_N) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

maximize (log-) likelihood to learn the probability parameters

Q: Why is maximizing log-likelihood a reasonable thing to do?

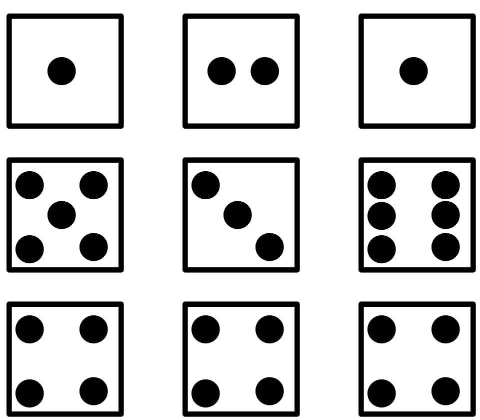
A: Develop a good model for what we observe

# Learning Parameters for the Die Model: Maximum Likelihood (Intuition)

$$p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

maximize (log-) likelihood to learn the probability parameters

If you observe these 9 rolls…



…what are "reasonable" estimates for p(w)?

p(1) = ? p(2) = ?

p(3) = ? p(4) = ?

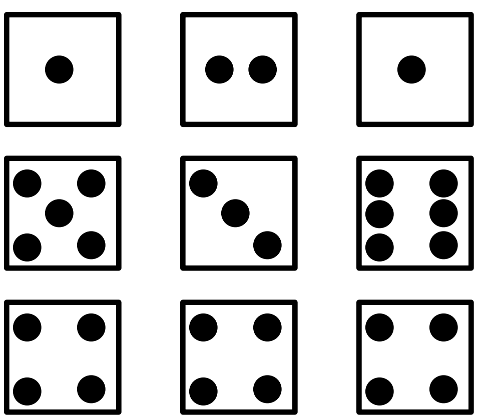
p(5) = ? p(6) = ?

# Learning Parameters for the Die Model: Maximum Likelihood (Intuition)

$$p(w_1, w_2, \dots, w_N) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

maximize (log-) likelihood to learn the probability parameters

If you observe these 9 rolls…



…what are "reasonable" estimates for p(w)?

p(1) = 2/9 p(2) = 1/9

p(3) = 1/9 p(4) = 3/9

p(5) = 1/9 p(6) = 1/9

maximum likelihood estimates

# Learning: Maximum Likelihood Estimation (MLE)

Core concept in intro statistics:

- Observe some data $\mathcal{X}$
- Compute some distribution $g(\mathcal{X})$ to {predict, explain, generate} $\mathcal{X}$
- Assume $g$ is controlled by parameters $\phi$, i.e., $g_\phi(\mathcal{X})$
  - Sometimes written $g(\mathcal{X}; \phi)$
- Learning appropriate value(s) of $\phi$ allows you to **GENERALIZE** about $\mathcal{X}$

*How do we "learn appropriate value(s) of $\phi$?"*

Many different options: a common one is **maximum likelihood estimation (MLE)**

- Find values $\phi$ s.t. $g_\phi(\mathcal{X} = \{x_1, \ldots, x_N\})$ is maximized
- Independence assumptions are very useful here!
- Logarithms are also useful!

# Learning: Maximum Likelihood Estimation (MLE)

Core concept in intro statistics:

- Observe some data $\mathcal{X}$
- Compute some distribution $g(\mathcal{X})$ to {predict, explain, generate} $\mathcal{X}$
- Assume $g$ is controlled by parameters $\phi$, i.e., $g_\phi(\mathcal{X})$
  - Sometimes written $g(\mathcal{X}; \phi)$
- MLE: Find values $\phi$ s.t. $g_\phi(\mathcal{X} = \{x_1, \ldots, x_N\})$ is maximized

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn $\phi$ such that $g$ correctly models, as accurately as possible, the amount of snow likely

Advanced topic

# Learning: Maximum Likelihood Estimation (MLE)

Core concept in intro statistics:

- Observe some data $\mathcal{X}$
- Compute some distribution $g(\mathcal{X})$ to {predict, explain, generate} $\mathcal{X}$
- Assume $g$ is controlled by parameters $\phi$, i.e., $g_\phi(\mathcal{X})$
  - Sometimes written $g(\mathcal{X}; \phi)$
- MLE: Find values $\phi$ s.t. $g_\phi(\mathcal{X} = \{x_1, \dots, x_N\})$ is maximized

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn $\phi$ such that $g$ correctly models, as accurately as possible, the amount of snow likely
- Assumption: each $x_i$ is independent from all others

$$\max_{\phi} \sum_{i=1}^{N} \log g_\phi(x_i)$$

Advanced topic

# MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn $\phi$ such that $g$ correctly models, as accurately as possible, the amount of snow likely
- Assumption: each $x_i$ is independent from all others

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

Q: Why is taking logarithms okay?

Q: What other assumptions, or decisions, do we need to make?

Advanced topic

# MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn $\phi$ such that $g$ correctly models, as accurately as possible, the amount of snow likely
- Assumption: each $x_i$ is independent from all others, but all from $g$

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

Q: Why is taking logarithms okay?

Q: What other assumptions, or decisions, do we need to make?

$x_i$ is positive, real-valued. What's a faithful probability distribution for $x_i$?

- Normal? ✗
- Gamma? ✓
- Exponential? ✓
- Bernoulli? ✗
- Poisson? ✗

Advanced topic

# MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn $\phi$ such that $g$ correctly models, as accurately as possible, the amount of snow likely
- Assumption: each $x_i$ is independent from all others, but all from $g$

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

Q: Why is taking logarithms okay?

Q: What other assumptions, or decisions, do we need to make?

$x_i$ is positive, real-valued. What's a faithful probability distribution for $x_i$?

- Normal? ✗
- Gamma? ✓ $p(X = x) = \frac{x^{k-1} \exp(\frac{-k}{\theta})}{\theta^k \Gamma(k)}$
- Exponential? ✓
- Bernoulli? ✗
- Poisson? ✗

Advanced topic

# MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn $\phi$ such that $g$ correctly models, as accurately as possible, the amount of snow likely
- Assumption: each $x_i$ is independent from all others, but all from $g$

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

Q: Why is taking logarithms okay?

Q: What other assumptions, or decisions, do we need to make?

$x_i$ is positive, real-valued. What's a faithful/nice-to-compute-and-good-enough probability distribution for $x_i$?

- Normal? ✗ ✓
- Gamma? ✓ ?
- Exponential? ✓ ?
- Bernoulli? ✗ ✗
- Poisson? ✗ ✗

$p(X = x) = \frac{1}{\sqrt{2\pi\sigma}} \exp(\frac{-(x-\mu)^2}{2\sigma^2})$

Advanced topic

# MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn $\phi$ such that $g$ correctly models, as accurately as possible, the amount of snow likely
- Assumption: each $x_i$ is independent from all others, but all from $g$

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

$$x_i \sim \mathrm{Normal}(\mu, \sigma^2)$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \log \mathrm{Normal}_{\mu,\sigma^2}(x_i) =$$

# MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn $\phi$ such that $g$ correctly models, as accurately as possible, the amount of snow likely
- Assumption: each $x_i$ is independent from all others, but all from $g$

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

$$x_i \sim \text{Normal}(\mu, \sigma^2)$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \log \text{Normal}_{\mu,\sigma^2}(x_i) =$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \left[\frac{-(x_i - \mu)^2}{\sigma^2}\right] - N \log \sigma = F$$

Advanced topic

# MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn $\phi$ such that $g$ correctly models, as accurately as possible, the amount of snow likely
- Assumption: each $x_i$ is independent from all others, but all from $g$

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

$$x_i \sim \mathrm{Normal}(\mu, \sigma^2)$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \log \mathrm{Normal}_{\mu,\sigma^2}(x_i) =$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \left[\frac{-(x_i-\mu)^2}{\sigma^2}\right] - N \log \sigma = F$$

Q: How do we find $\mu, \sigma^2$?

Advanced topic

# MLE Snowfall Example

Example: How much does it snow?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- Goal: learn $\phi$ such that $g$ correctly models, as accurately as possible, the amount of snow likely
- Assumption: each $x_i$ is independent from all others, but all from $g$

$$\max_{\phi} \sum_{i=1}^{N} \log g_{\phi}(x_i)$$

$$x_i \sim \text{Normal}(\mu, \sigma^2)$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \log \text{Normal}_{\mu,\sigma^2}(x_i) =$$

$$\max_{(\mu,\sigma^2)} \sum_{i=1}^{N} \left[\frac{-(x_i-\mu)^2}{\sigma^2}\right] - N \log \sigma = F$$

Q: How do we find $\mu, \sigma^2$?

A: Differentiate and find that

$$\hat{\mu} = \frac{\sum_i x_i}{N}$$

$$\sigma^2 = \frac{\sum_i (x_i - \hat{\mu})^2}{N}$$

# Learning: Maximum Likelihood Estimation (MLE)

Central to machine learning:

- Observe some data $(\mathcal{X}, \mathcal{Y})$
- Compute some function $f(\mathcal{X})$ to {predict, explain, generate} $\mathcal{Y}$
- Assume $f$ is controlled by parameters $\theta$, i.e., $f_\theta(\mathcal{X})$
  - Sometimes written $f(\mathcal{X}; \theta)$

# Learning: Maximum Likelihood Estimation (MLE)

Central to machine learning:

- Observe some data $(\mathcal{X}, \mathcal{Y})$
- Compute some function $f(\mathcal{X})$ to {predict, explain, generate} $\mathcal{Y}$
- Assume $f$ is controlled by parameters $\theta$, i.e., $f_\theta(\mathcal{X})$
  - Sometimes written $f(\mathcal{X}; \theta)$
- Parameters are learned to $\underbrace{\text{minimize error (loss) } \ell}_{\text{Advanced topic}}$

# Learning: Maximum Likelihood Estimation (MLE)

Example: Can I sleep in the next time it snows/is school canceled?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- $\mathcal{Y} = \{y_1, y_2, \dots, y_N\}$ are closure results from the previous N storms
- Goal: learn $\theta$ such that $f$ correctly predicts, as accurately as possible, if UMBC will close in the next storm:
  - $y^*_{n+1}$ from $x_{n+1}$

- If we assume the output of $f$ is a *probability distribution* on $\mathcal{Y}|\mathcal{X}$…
  - $f(\mathcal{X}) \rightarrow \{p(\text{yes}|\mathcal{X}), p(\text{no}|\mathcal{X})\}$
- Then re: $\theta$, {predicting, explaining, generating} $\mathcal{Y}$ means… *what*?

# Learning: Maximum Likelihood Estimation (MLE)

Example: Can I sleep in the next time it snows/is school canceled?

- $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ are snowfall values from the previous N storms
- $\mathcal{Y} = \{y_1, y_2, \ldots, y_N\}$ are closure results from the previous N storms
- Goal: learn $\theta$ such that $f$ correctly predicts, as accurately as possible, if UMBC will close in the next storm:
  - $y^*_{n+1}$ from $x_{n+1}$

- If we assume the output of $f$ is a *probability distribution* on $\mathcal{Y}|\mathcal{X}$…
- Then re: $\theta$, {predicting, explaining, generating} $\mathcal{Y}$ means… *what*?

# Learning: Maximum Likelihood Estimation (MLE)

Example: Can I sleep in the next time it snows/is school canceled?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- $\mathcal{Y} = \{y_1, y_2, \dots, y_N\}$ are closure results from the previous N storms
- Goal: learn $\theta$ such that $f$ correctly predicts, as accurately as possible, if UMBC will close in the next storm:
  - $y^*_{n+1}$ from $x_{n+1}$

- If we assume the output of $f$ is a *probability distribution* on $\mathcal{Y}|\mathcal{X}$...
- Then re: $\theta$, {predicting, explaining, generating} $\mathcal{Y}$ means finding a value for $\theta$ that maximizes the probability of $\mathcal{Y}$ given $\mathcal{X}$

# Learning: Maximum Likelihood Estimation (MLE)

Example: Can I sleep in the next time it snows/is school canceled?

- $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ are snowfall values from the previous N storms
- $\mathcal{Y} = \{y_1, y_2, \dots, y_N\}$ are closure results from the previous N storms
- Goal: learn $\theta$ such that $f$ correctly predicts, as accurately as possible, if UMBC will close in the next storm:
  - $y^*_{n+1}$ from $x_{n+1}$

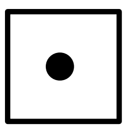
- If we assume the output of $f$ is a *probability distribution* on $\mathcal{Y}|\mathcal{X}$…
- Then re: $\theta$, {predicting, explaining, generating} $\mathcal{Y}$ means finding a value for $\theta$ that maximizes the probability of $\mathcal{Y}$ given $\mathcal{X}$, according to $f$
- To model $\mathcal{X}$: learn a distribution $g$, on $\mathcal{X}$

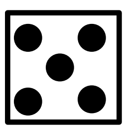
# *Extended examples of MLE*

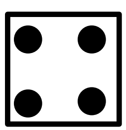
Advanced topic

# Learning Parameters for the Die Model: Maximum Likelihood (Math)

$$p(\overbrace{w_1, w_2, \ldots, w_N}^{\text{N different (independent) rolls}}) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

$w_1 = 1$ 

$w_2 = 5$ 

$w_3 = 4$ 

$\ldots$

**Generative Story**

for roll $i = 1$ to $N$:

$\quad w_i \sim \mathrm{Cat}(\theta)$

**Maximize Log-likelihood**

$$\mathcal{L}(\theta) = \sum_i \log p_\theta(w_i)$$

$$= \sum_i \log \theta_{w_i}$$

Advanced topic

# Learning Parameters for the Die Model: Maximum Likelihood (Math)

$$p(\overbrace{w_1, w_2, \ldots, w_N}^{\text{N different (independent) rolls}}) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

**Maximize Log-likelihood (with distribution constraints)**

$$\mathcal{L}(\theta) = \sum_i \log \theta_{w_i} \ \text{s.t.} \sum_{k=1}^{6} \theta_k = 1$$

(we can include the inequality constraints $0 \leq \theta_k$, but it complicates the problem and, *right now*, is not needed)

solve using Lagrange multipliers

# Learning Parameters for the Die Model: Maximum Likelihood (Math)

Advanced topic

$$p(\overbrace{w_1, w_2, \ldots, w_N}^{\text{N different (independent) rolls}}) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

**Maximize Log-likelihood (with distribution constraints)**

$$\mathcal{F}(\theta) = \sum_i \log \theta_{w_i} - \lambda\left(\sum_{k=1}^{6} \theta_k - 1\right)$$

(we can include the inequality constraints $0 \leq \theta_k$, but it complicates the problem and, *right now*, is not needed)

$$\frac{\partial \mathcal{F}(\theta)}{\partial \theta_k} = \sum_{i:w_i=k} \frac{1}{\theta_{w_i}} - \lambda \qquad \frac{\partial \mathcal{F}(\theta)}{\partial \lambda} = -\sum_{k=1}^{6} \theta_k + 1$$

Advanced topic

# Learning Parameters for the Die Model: Maximum Likelihood (Math)

$$p(\overbrace{w_1, w_2, \ldots, w_N}^{\text{N different (independent) rolls}}) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

**Maximize Log-likelihood (with distribution constraints)**

$$\mathcal{F}(\theta) = \sum_i \log \theta_{w_i} - \lambda\left(\sum_{k=1}^{6} \theta_k - 1\right)$$

(we can include the inequality constraints $0 \leq \theta_k$, but it complicates the problem and, *right now*, is not needed)

$$\theta_k = \frac{\sum_{i:w_i=k} 1}{\lambda}$$

$$\text{optimal } \lambda \text{ when } \sum_{k=1}^{6} \theta_k = 1$$

Advanced topic

# Learning Parameters for the Die Model: Maximum Likelihood (Math)

$$p(\overbrace{w_1, w_2, \dots, w_N}^{\text{N different (independent) rolls}}) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

**Maximize Log-likelihood (with distribution constraints)**

$$\mathcal{F}(\theta) = \sum_i \log \theta_{w_i} - \lambda\left(\sum_{k=1}^{6} \theta_k - 1\right)$$

(we can include the inequality constraints $0 \le \theta_k$, but it complicates the problem and, *right now*, is not needed)

$$\theta_k = \frac{\sum_{i:w_i=k} 1}{\sum_k \sum_{i:w_i=k} 1} = \frac{N_k}{N}$$

$$\text{optimal } \lambda \text{ when } \sum_{k=1}^{6} \theta_k = 1$$

# Example: Conditionally Rolling a Die

$$p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

*add complexity to better explain what we see*

$$p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = p(z_1)p(w_1|z_1)\cdots p(z_N)p(w_N|z_N)$$
$$= \prod_i p(w_i|z_i)\,p(z_i)$$

# Example: Conditionally Rolling a Die

$$p(w_1, w_2, \dots, w_N) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

*add complexity to better explain what we see*

$$p(z_1, w_1, z_2, w_2, \dots, z_N, w_N) = p(z_1)p(w_1|z_1)\cdots p(z_N)p(w_N|z_N)$$

$$= \prod_i p(w_i|z_i)\, p(z_i)$$

First flip a coin…

$z_1 = T$


$z_2 = H$

…

# Example: Conditionally Rolling a Die

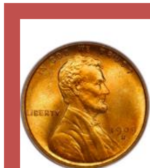
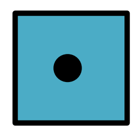
$$p(w_1, w_2, \dots, w_N) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

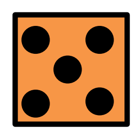
$\Big\downarrow$ *add complexity to better explain what we see*

$$p(z_1, w_1, z_2, w_2, \dots, z_N, w_N) = p(z_1)p(w_1|z_1)\cdots p(z_N)p(w_N|z_N)$$
$$= \prod_i p(w_i|z_i)\, p(z_i)$$

First flip a coin…

…then roll a different die depending on the coin flip

 $z_1 = T$ $\quad w_1 = 1$ 

$z_2 = H$ $\quad w_2 = 5$ 

…

# Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

$$p(w_1, w_2, \ldots, w_N) = p(w_1)p(w_2)\cdots p(w_N) = \prod_i p(w_i)$$

*add complexity to better explain what we see*

$$p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = p(z_1)p(w_1|z_1)\cdots p(z_N)p(w_N|z_N)$$
$$= \prod_i p(w_i|z_i)\, p(z_i)$$

If you observe the $z_i$ values, this is easy!

Advanced topic

# Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

$$p(z_1, w_1, z_2, w_2, \dots, z_N, w_N) = \prod_i p(w_i | z_i)\, p(z_i)$$

If you observe the $z_i$ values, this is easy!

**First: Write the Generative Story**

$\lambda =$ distribution over coin (z)

$\gamma^{(H)} =$ distribution for die when coin comes up heads

$\gamma^{(T)} =$ distribution for die when coin comes up tails

for item $i = 1$ to $N$:

$\quad z_i \sim \text{Bernoulli}(\lambda)$

$\quad w_i \sim \text{Cat}\left(\gamma^{(z_i)}\right)$

Advanced topic

# Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

$$p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = \prod_i p(w_i | z_i)\, p(z_i)$$

If you observe the $z_i$ values, this is easy!

**First: Write the Generative Story**

$\lambda =$ distribution over coin (z)
$\gamma^{(H)} =$ distribution for H die
$\gamma^{(T)} =$ distribution for T die
for item $i = 1$ to $N$:
  $z_i \sim \text{Bernoulli}(\lambda)$
  $w_i \sim \text{Cat}\left(\gamma^{(z_i)}\right)$

**Second: Generative Story → Objective**

$$\mathcal{F}(\theta) = \sum_i^n \left(\log \lambda_{z_i} + \log \gamma_{w_i}^{(z_i)}\right)$$

$-$ Lagrange multiplier constraints

Advanced topic

# Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

$$p(z_1, w_1, z_2, w_2, \ldots, z_N, w_N) = \prod_i p(w_i | z_i)\, p(z_i)$$

If you observe the $z_i$ values, this is easy!

**First: Write the Generative Story**

$\lambda = \text{distribution over coin (z)}$

$\gamma^{(H)} = \text{distribution for H die}$

$\gamma^{(T)} = \text{distribution for T die}$

for item $i = 1$ to $N$:

$\quad z_i \sim \text{Bernoulli}(\lambda)$

$\quad w_i \sim \text{Cat}\left(\gamma^{(z_i)}\right)$

**Second: Generative Story → Objective**

$$\mathcal{F}(\theta) = \sum_i^n (\log \lambda_{z_i} + \log \gamma_{w_i}^{(z_i)})$$

$$-\eta\left(\sum_{k=1}^{2} \lambda_k - 1\right) - \sum_{k=1}^{2} \delta_k \left(\sum_{j=1}^{6} \gamma_j^{(k)} - 1\right)$$

Advanced topic

# Learning in Conditional Die Roll Model: Maximize (Log-)Likelihood

$$p(z_1, w_1, z_2, w_2, \dots, z_N, w_N) = \prod_i p(w_i | z_i)\, p(z_i)$$

If you observe the $z_i$ values, this is easy!

But if you don't observe the $z_i$ values, this is not easy!

**First: Write the Generative Story**

$\lambda$ = distribution over coin (z)

$\gamma^{(H)}$ = distribution for H die

$\gamma^{(T)}$ = distribution for T die

for item $i = 1$ to $N$:

$z_i \sim \text{Bernoulli}(\lambda)$

$w_i \sim \text{Cat}\left(\gamma^{(z_i)}\right)$

**Second: Generative Story → Objective**

$$\mathcal{F}(\theta) = \sum_i^n \left(\log \lambda_{z_i} + \log \gamma_{w_i}^{(z_i)}\right)$$

$$-\eta\left(\sum_{k=1}^{2} \lambda_k - 1\right) - \sum_{k=1}^{2} \delta_k \left(\sum_{j=1}^{6} \gamma_j^{(k)} - 1\right)$$

# Model selection

**Goal:** Select the best network structure, given the data

**Input:**

- Training data
- Scoring function

**Output:**

- A network that maximizes the score

# Structure selection: Scoring

- Bayesian: prior over parameters and structure
  - get balance between model complexity and fit to data as a byproduct

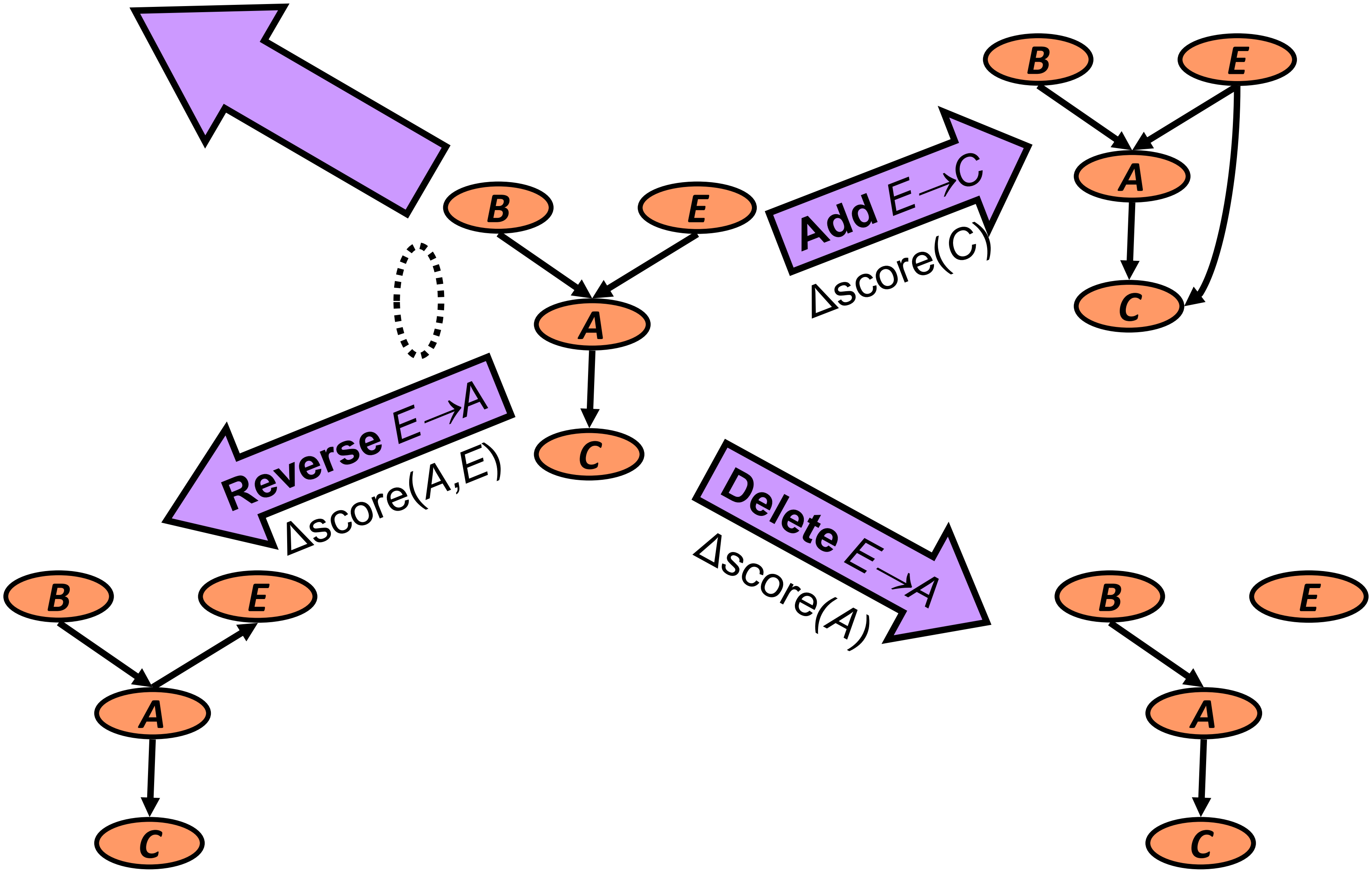
***Marginal likelihood*** ***Prior***

- Score (G:D) = log P(G|D) $\alpha$ log [P(D|G) P(G)]
- Marginal likelihood just comes from our parameter estimates
- Prior on structure can be any measure we want; typically a function of the network complexity

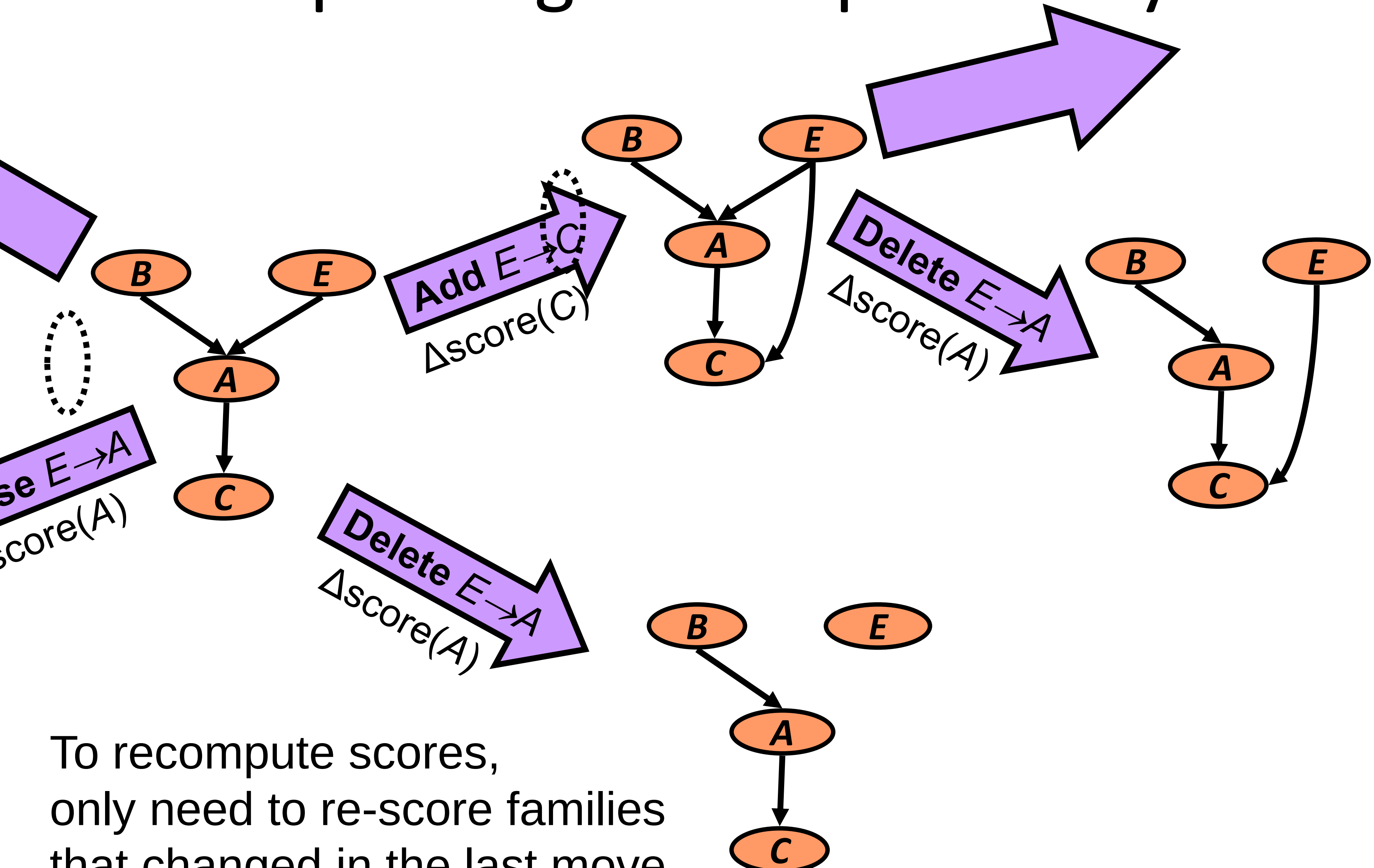
**Same key property: Decomposability**

$$\text{Score(structure)} = \textstyle\sum_i \text{Score(family of } X_i)$$

# Heuristic search



# Exploiting decomposability




To recompute scores,
only need to re-score families
that changed in the last move