Bayesian Networks

CompSci 370
Duke University
Ron Parr

Why Joint Distributions are Important
(Contrast with Pure Prediction)

- Joint distributions gives $P(X_1 \ldots X_n)$
- Classification/Diagnosis
  - Suppose X_1=disease
  - $X_2 \ldots X_n$ = symptoms (naturally handles missing data)
- Co-occurrence
  - Suppose X_3=lung cancer
  - X_5=smoking
- Rare event Detection
  - Suppose $X_1 \ldots X_n$ = parameters of a credit card transaction
  - Call card holder if $P(X_1 \ldots X_n)$ is below threshold?

Modeling Joint Distributions

- To do this correctly, we need a full assignment of probabilities to all atomic events
- Unwieldy in general for discrete variables: n binary variables = 2^n atomic events
- Independence makes this tractable, but too strong (rarely holds)
- Conditional independence is a good compromise: Weaker than independence, but still has great potential to simplify things

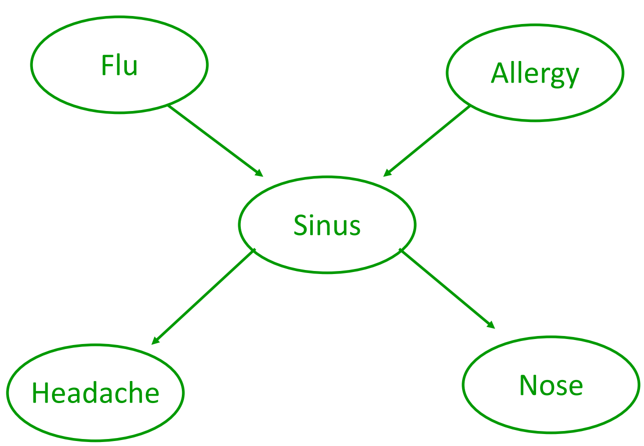
Overview

- Conditional independence
- Bayesian networks
- Variable Elimination
- Sampling

Conditional Independence

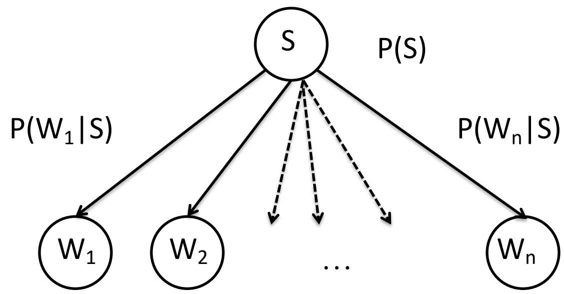
- Suppose we know the following:
  - The flu causes sinus inflammation
  - Allergies cause sinus inflammation
  - Sinus inflammation causes a runny nose
  - Sinus inflammation causes headaches
- How are these connected?

Example 1: Simple graphical structure

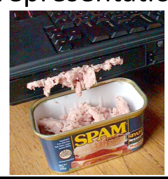


Knowing sinus separates the variables from each other.

Example 2: Naïve Bayes Spam Filter



We will see later why this is a particularly convenient representation.
(Does it make a correct assumption?)



Source: http://www.todayifoundout.com/index.php/2010/09/how-the-word-spam-came-to-mean-junk-message/

Conditional Independence

- We say that two variables, A and B, are conditionally independent given C if:
  - $P(A|BC) = P(A|C)$
  - $P(AB|C) = P(A|C)P(B|C)$
- How does this help?
- We store only a conditional probability table (CPT) of each variable given its parents
- Naïve Bayes (e.g. Spam Assassin) is a special case of this!

Notation Reminder

- P(A|B) is a conditional prob. distribution
  - It is a function!
  - P(A=true|B=true), P(A=true|B=false), P(A=false|B=True), P(A=false|B=true)
- P(A|b) is a probability distribution, function
- P(a|B) is a function, not a distribution
- P(a|b) is a number

What is Bayes Net?

- A directed acyclic graph (DAG)
- Each variable is

conditionally independent of non-descendants, ***given parents***

- Joint probability decomposes:

$$P(x_1..x_n) = \prod_i P(x_i \mid \text{parents}(x_i))$$

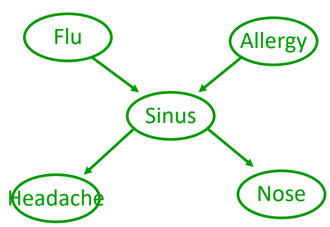
- For each node X_i, store $P(X_i|\text{parents}(X_i))$
- Call this a Conditional Probability Table (CPT)
- CPT size is exponential in number of parents

Real Applications of Bayes Nets

- Diagnosis of lymph node disease
- Used by robots to identify meteorites to study
- Study the human genome: Alex Hartemink et al.
- Used in MS Windows
- Many other applications…

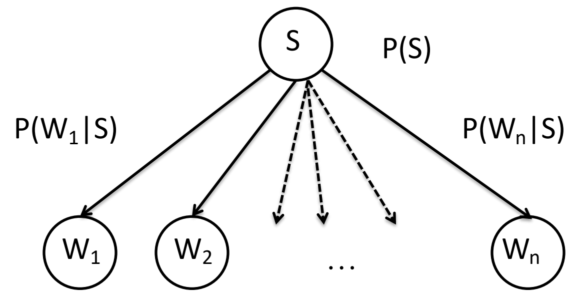
See, e.g., https://www.nature.com/articles/s41598-018-24758-5

Space Efficiency



- Entire joint distribution as 32 (31) entries
  - P(H|S),P(N|S) have 4 (2)
  - P(S|AF) has 8 (4)
  - P(A), P(F) have 2 (1)
  - Total is 20 (10)
- This can require exponentially less space
- Space problem is solved for "most" problems

Naïve Bayes Space Efficiency



Entire Joint distribution has 2^{n+1} (2^{n+1}-1) numbers vs. 4n+2 (2n+1)

(Non)Uniqueness of Bayes Nets I

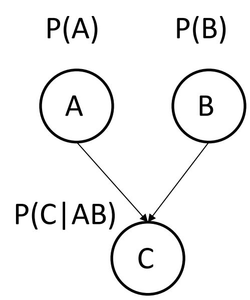
- Suppose you have two variables that are **NOT** independent
- Two possible networks:
  - A is parent of B
  - B is parent of A
- Which is right?
- There is no wrong answer!
- Each network can express arbitrary P(AB)
- Network does **NOT** encode causal or temporal dynamics

(Non)Uniqueness of Bayes Nets II

- Can construct valid Bayes net by adding variables incrementally
- For each new variable, connect all influencing variables as parents – new variables never become parents of existing variables (how does this ensure that all variables are conditionally independent of non-descendents given parents?)
- Different order can lead to different Bayesian networks for the same distribution

Suppose A and B are uniform, C=(A ∨ B)

A	B	C	P
0	0	0	0.25
0	0	1	0
0	1	0	0
0	1	1	0.25
1	0	0	0
1	0	1	0.25
1	1	0	0
1	1	1	0.25



$P(a)=P(\bar{a})=P(b)=P(\bar{b})=0.5$

$P(c)=0.75, P(\bar{c})=0.25$

$P(ab)=P(a\bar{b})=P(\bar{a}b)=P(\bar{a}\bar{b})=0.25$

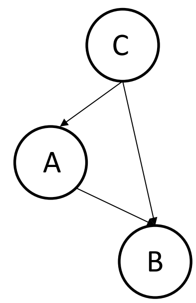
$P(c|ab)=P(c|a\bar{b})=P(c|\bar{a}b)=1.0$, $P(c|\bar{a}\bar{b})=0$

(only showing c=true case)

Add A, then B, then C case

Suppose A and B are uniform, C=(A ∨ B)

A	B	C	P
0	0	0	0.25
0	0	1	0
0	1	0	0
0	1	1	0.25
1	0	0	0
1	0	1	0.25
1	1	0	0
1	1	1	0.25



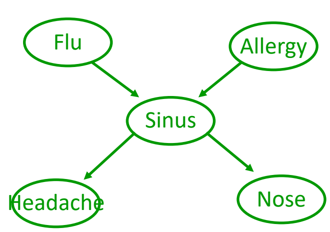
$P(c)=0.75, P(\bar{c})=0.25$
$P(ac)=0.5$, $P(a\bar{c})=0$, $P(\bar{a}c)=P(\overline{ac})=0.25$
$P(a|c)=2/3$, $P(\bar{a}|c)=1/3$, $P(a|\bar{c})= 0$, $P(\bar{a}|\bar{c})=1$
$P(b|ac)=1/2$, $P(b|a\bar{c})=P(b|\overline{ac})=0$, $P(b|\bar{a}c)=1$

(only showing b=true case)

Add C, then A, then B case

Atomic Event Probabilities

$$P(x_1 .. x_n) = \prod_i P(x_i \mid \text{parents}(x_i))$$



Note that this is guaranteed true if we construct net incrementally, so that for each new variable added, we connect all influencing variables as parents (prove it by induction)

Answering Queries Using Marginalization

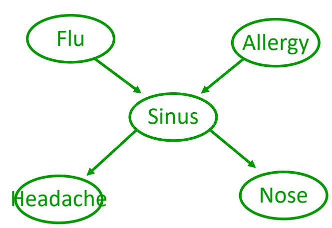
$$P(f \mid h) = \frac{P(fh)}{P(h)} = \frac{\sum_{SAN} P(fhSAN)}{\sum_{SANF} P(hSANF)} = \frac{\sum_{SAN} P(f)P(A)P(S \mid Af)P(h \mid S)P(N \mid S)}{\sum_{SANF} P(F)P(A)P(S \mid AF)P(h \mid S)P(N \mid S)}$$

$$P(hSANF) = \prod_x p(x \mid parents(x))$$
$$= P(h \mid S)P(N \mid S)P(S \mid AF)P(A)P(F)$$

defn. of conditional probability

Doing this naïvely, we need to sum over all atomic events defined over these variables. There are exponentially many of these.

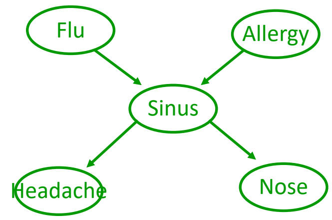
Working Smarter



$$P(h) = \sum_{SANF} P(hSANF)$$
$$= \sum_{SANF} P(h \mid S)P(N \mid S)P(S \mid AF)P(A)P(F)$$
$$= \sum_{NS} P(h \mid S)P(N \mid S) \sum_{AF} P(S \mid AF)P(A)P(F)$$
$$= \sum_{S} P(h \mid S) \sum_{N} P(N \mid S) \sum_{AF} P(S \mid AF)P(A)P(F)$$

Potential for exponential reduction in computation.

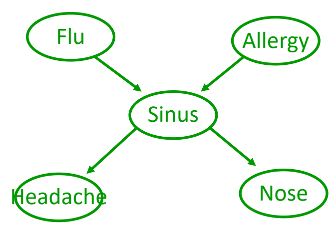
Understanding Notation



$$P(h) = \sum_{SANF} P(hSANF)$$
$$= \sum_{SANF} P(h \mid S)P(N \mid S)P(S \mid AF)P(A)P(F)$$
$$= \sum_{NS} P(h \mid S)P(N \mid S)\sum_{AF} P(S \mid AF)P(A)P(F)$$
$$= \sum_{S} P(h \mid S)\sum_{N} P(N \mid S)\boxed{\sum_{AF} P(S \mid AF)P(A)P(F)}$$

8 combinations 3 binary variables
2 variables summed out
Result is a function of S

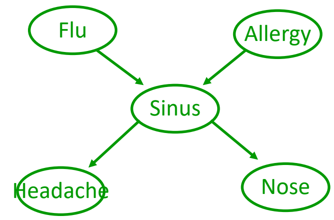
Understanding Notation



$$P(h) = \sum_{SANF} P(hSANF)$$
$$= \sum_{SANF} P(h \mid S)P(N \mid S)P(S \mid AF)P(A)P(F)$$
$$= \sum_{NS} P(h \mid S)P(N \mid S)\sum_{AF} P(S \mid AF)P(A)P(F)$$
$$= \sum_{S} P(h \mid S)\boxed{\sum_{N} P(N \mid S)}\sum_{AF} P(S \mid AF)P(A)P(F)$$

4 combinations 2 binary variables
1 variable summed out
Result is a function of S

Understanding Notation



$$P(h) = \sum_{SANF} P(hSANF)$$

$$= \sum_{SANF} P(h \mid S)P(N \mid S)P(S \mid AF)P(A)P(F)$$

$$= \sum_{NS} P(h \mid S)P(N \mid S) \sum_{AF} P(S \mid AF)P(A)P(F)$$

$$= \boxed{\sum_{S} P(h \mid S)} \sum_{N} P(N \mid S) \sum_{AF} P(S \mid AF)P(A)P(F)$$

2 combinations of 1 binary variable
1 variable summed out
Result is a number

Computational Efficiency

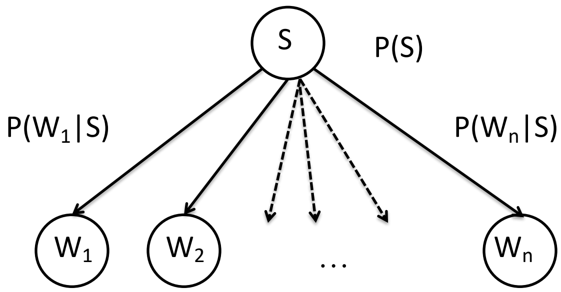
$$\sum_{SANF} P(hSANF) = \sum_{SANF} P(h \mid S)P(N \mid S)P(S \mid AF)P(A)P(F)$$

$$= \sum_{S} P(h \mid S) \sum_{N} P(N \mid S) \sum_{AF} P(S \mid AF)P(A)P(F)$$

The distributive law allows us to decompose the sum.
AKA: Variable elimination, sum-product algorithm, etc.

Potential for an exponential reduction in computation costs.

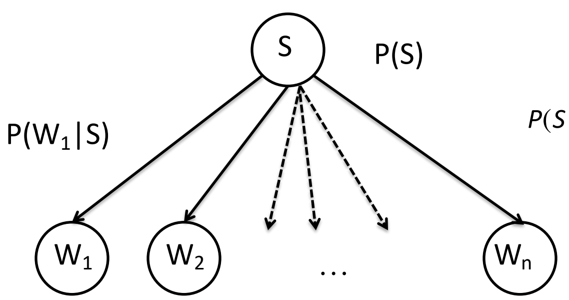
Naïve Bayes Efficiency



Given a set of words, we want to know which is larger: $P(s|W_1\ldots W_n)$ or $P(\neg s|W_1\ldots W_n)$.

Use Bayes Rule: $$P(S \mid W_1 .. W_n) = \frac{P(W_1 .. W_n \mid S)P(S)}{P(W_1 .. W_n)}$$

Naïve Bayes Efficiency II



$$P(S \mid W_1 .. W_n) = \frac{P(W_1 .. W_n \mid S)P(S)}{P(W_1 .. W_n)}$$

Observation 1: We can ignore $P(W_1\ldots W_n)$
Observation 2: P(S) is given
Observation 3: $P(W_1\ldots W_n|S)$ is easy: $$P(W_1 .. W_n \mid S) = \prod_{i=1}^{n} P(W_i \mid S)$$

Note: Can also do variable elimination by summing out leaves first.

Checkpoint

- BNs can give us an exponential reduction in the space required to represent a joint distribution.
- Storage is exponential in largest parent set.
- Claim: Parent sets are often reasonable.
- Claim: Inference cost is often reasonable.
- Question: Can we quantify relationship between structure and inference cost?

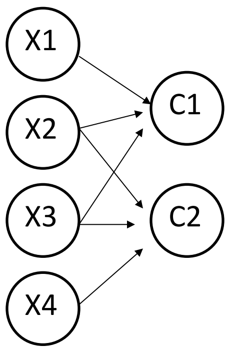
Now the Bad News…

- In full generality: Inference is NP-hard
- Decision problem: Is P(X)>0?
- We reduce from 3SAT
- 3SAT variables map to BN variables
- Clauses become variables with the corresponding SAT variables as parents

(NP-hardness result not covered in lecture but included in slides for those who are interested.)

Reduction

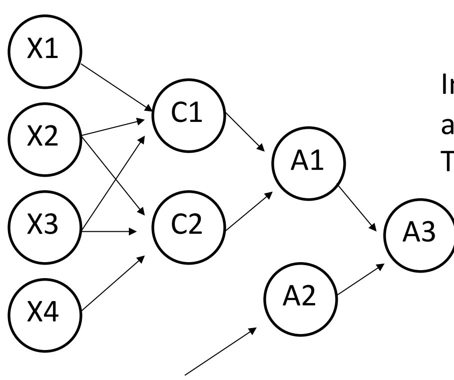
$$(\overline{X}_1 \vee X_2 \vee X_3) \wedge (\overline{X}_2 \vee X_3 \vee X_4) \wedge \ldots$$



Problem: What if we have a large number of clauses? How does this fit into our decision problem framework?

And Trees

We could make a single variable which is the AND of all of our clauses, but this would have CPT that is exponential in the number of clauses.



Implement as a tree of ANDs. This is polynomial.

Checkpoint

- BNs can be very compact
- Worst case: Inference is intractable

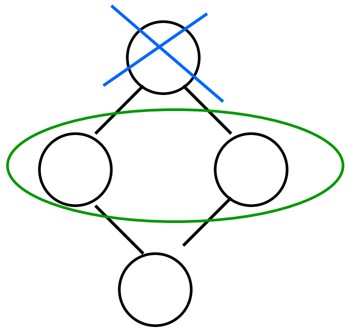
- Hope that worst is case:
  - Avoidable (frequently, but no free lunch)
  - Easily characterized in some way

Clues in the Graphical Structure

- Q: How does graphical structure relate to our ability to push in summations over variables?

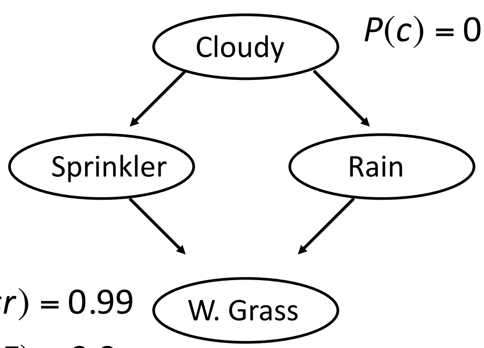
- A:
  - We relate summations to graph operations
  - Summing out a variable =
    - Removing node(s) from DAG
    - Creating new replacement node
  - Relate graph properties to computational efficiency

Variable Elimination as Graph Operations



We can think of summing out a variable as creating a new "super variable" which contains all of that variable's neighbors

Another Example Network



$P(c) = 0.5$

$P(s \mid c) = 0.1$

$P(s \mid \bar{c}) = 0.5$

$P(r \mid c) = 0.8$

$P(r \mid \bar{c}) = 0.2$

$P(w \mid sr) = 0.99$

$P(w \mid s\bar{r}) = 0.9$

$P(w \mid \bar{s}r) = 0.9$

$P(w \mid \bar{s}\bar{r}) = 0.0$

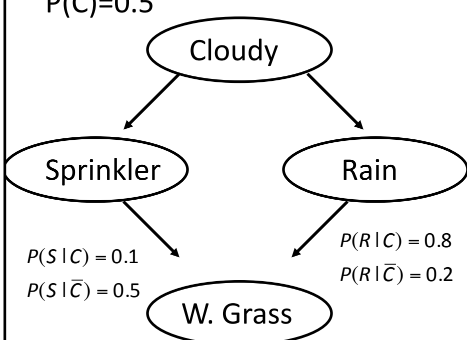
Marginal Probabilities

Suppose we want P(W):

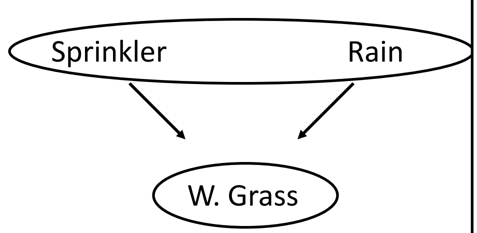
$$
\begin{aligned}
P(W) &= \sum_{CSR} P(CSRW) \\
&= \sum_{CSR} P(C)P(S \mid C)P(R \mid C)P(W \mid RS) \\
&= \sum_{SR} P(W \mid RS) \sum_{C} P(S \mid C)P(C)P(R \mid C)
\end{aligned}
$$

Eliminating Cloudy

P(C)=0.5



$$
\begin{aligned}
P(sr) &= 0.5*0.1*0.8 + 0.5*0.5*0.2 = 0.09 \\
P(s\bar{r}) &= 0.5*0.1*0.2 + 0.5*0.5*0.8 = 0.21 \\
P(\bar{s}r) &= 0.5*0.9*0.8 + 0.5*0.5*0.2 = 0.41 \\
P(\bar{s}\bar{r}) &= 0.5*0.9*0.2 + 0.5*0.5*0.8 = 0.29
\end{aligned}
$$



$$
\begin{aligned}
P(W) &= \sum_{CSR} P(CSRW) \\
&= \sum_{CSR} P(C)P(S \mid C)P(R \mid C)P(W \mid RS) \\
&= \sum_{SR} P(W \mid RS) \sum_{C} P(S \mid C)P(C)P(R \mid C)
\end{aligned}
$$

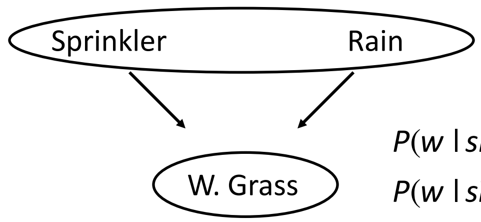
Eliminating Sprinkler/Rain

$P(sr) = 0.09$

$P(s\bar{r}) = 0.21$

$P(\bar{s}r) = 0.41$

$P(\bar{s}\bar{r}) = 0.29$



$P(w \mid sr) = 0.99$

$P(w \mid s\bar{r}) = 0.9$

$P(w \mid \bar{s}r) = 0.9$

$P(w \mid \bar{s}\bar{r}) = 0.0$

$$
\begin{aligned}
P(w) &= \sum\nolimits_{SR} P(w \mid RS)P(RS) \\
&= 0.09 * 0.99 + 0.21 * 0.9 + 0.41 * 0.9 + 0.29 * 0 \\
&= 0.6471
\end{aligned}
$$

Dealing With Evidence

Suppose we have observed that the grass is wet?
What is the probability that it has rained?

$$
\begin{aligned}
P(R \mid W) &= \alpha P(RW) \\
&= \alpha \sum\nolimits_{CS} P(CSRW) \\
&= \alpha \sum\nolimits_{CS} P(C)P(S \mid C)P(R \mid C)P(W \mid RS) \\
&= \alpha \sum\nolimits_{C} P(R \mid C)P(C) \sum_{S} P(S \mid C)P(W \mid RS)
\end{aligned}
$$

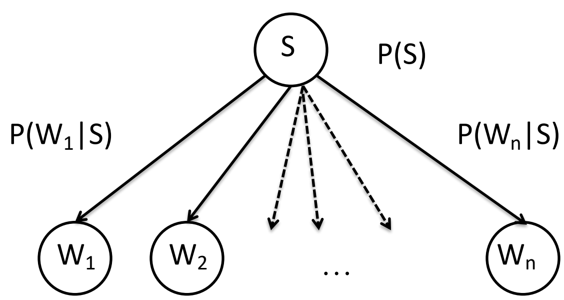
Is there a more clever way to deal with w?
Only keep the relevant parts.

Efficiency of Variable Elimination

- Exponential in the largest domain size of new variables created
- Equivalently: Exponential in largest function created by pushing in summations (sum-product algorithm)

- Linear for trees (DAGs in which undirected structure is a tree)
- Almost linear for almost trees ☺

Naïve Bayes Efficiency



Another way to understand why Naïve Bayes is efficient:
It's a tree!

Facts About Variable Elimination

- Picking variables in optimal order is NP hard
- For some networks, there will be no elimination ordering that results in a poly time solution
  (Must be the case unless P=NP)
- Polynomial for trees
- Need to get a little fancier if there are a large number of query variables or evidence variables

Beyond Variable Elimination

- Variable elimination must be rerun for every new query
- Possible to compile a Bayes net into a new data structure to make repeated queries more efficient
  - Recall that inference in trees is linear
  - Define a "cluster tree" where
    - Clusters = sets of original variables
    - Can infer original probs from cluster probs
- For networks w/o good elimination schemes
  - Sampling (discussed briefly)
  - Cutsets (not covered in this class)
  - Variational methods (not covered in this class)
  - Loopy belief propagation (not covered in this class)

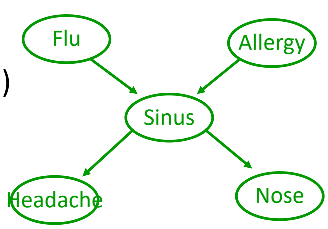
Sampling

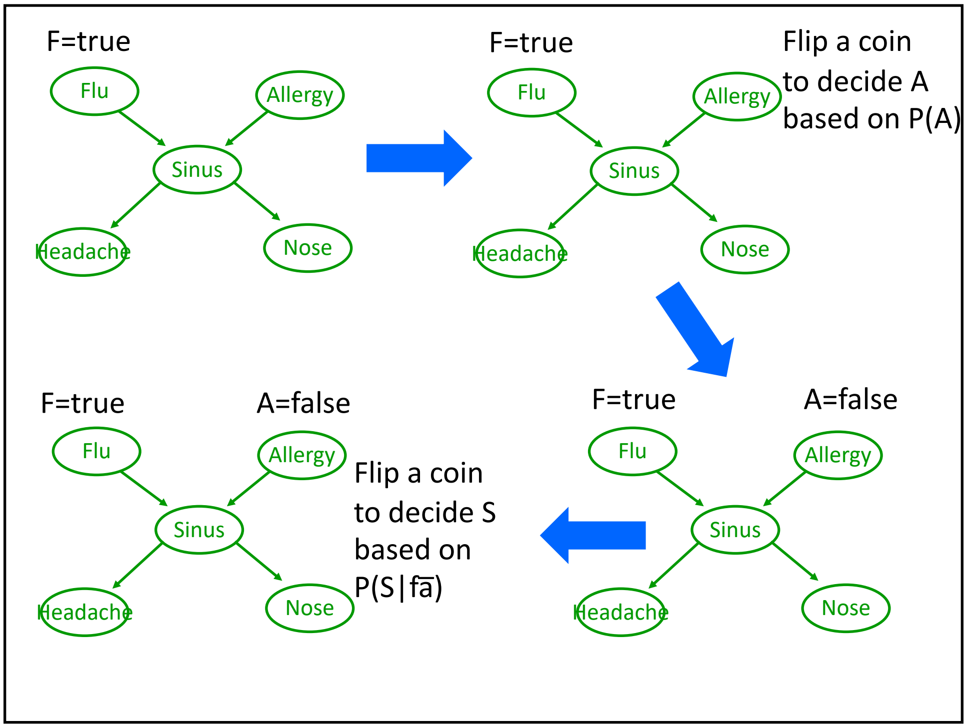
- A Bayes net is an example of a **generative model** of a probability distribution
- Generative models allow one to generate samples from a distribution in a natural way
- Sampling algorithm:
  - While some variables are not sampled
    - Pick variable x with no unsampled parents
    - Assign this variable a value from p(x|parents(x))
  - Do this n times
  - Compute P(a) by counting in what fraction a is true

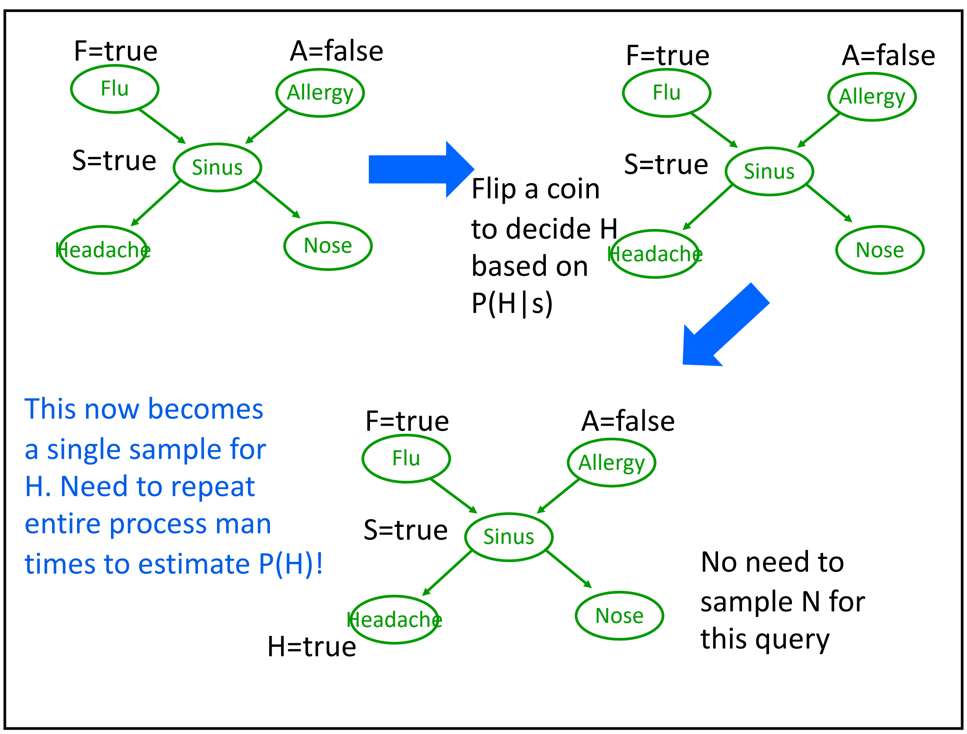
Sampling Example

- Suppose you want to compute P(H)
- Start with the parentless nodes:

Flip a coin to decide F based on P(F)

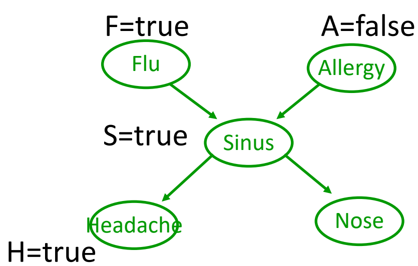



F=true
Flu
Allergy
Sinus
Headache
Nose
F=true
Flu
Allergy
Sinus
Headache
Nose
Flip a coin
to decide A
based on P(A)
F=true
A=false
Flu
Allergy
Sinus
Headache
Nose
Flip a coin
to decide S
based on
P(S|f$\bar{a}$)
F=true
A=false
Flu
Allergy
Sinus
Headache
Nose


F=true
A=false
Flu
Allergy
S=true
Sinus
Headache
Nose
Flip a coin
to decide H
based on
P(H|s)
F=true
A=false
Flu
Allergy
S=true
Sinus
Headache
Nose
This now becomes
a single sample for
H. Need to repeat
entire process man
times to estimate P(H)!
F=true
A=false
Flu
Allergy
S=true
Sinus
Headache
Nose
H=true
No need to
sample N for
this query

Sampling with Observed Evidence

- Suppose you know H=true
- Want to know P(F|h)?
- How can we use sampling?



Count fraction of times F is true/false when H is also true

But what if we flip a coin For H and it turns out false?

Comments on Sampling

- Sampling is the easiest algorithm to implement
- Can compute marginal or conditional distributions by counting
- Not efficient in general

- Problem: How do we handle observed values?
  - Rejection sampling: Quit and start over when mismatches occur
  - Importance sampling: Use a reweighting trick to compensate for mismatches
  - Low probability events are still a problem (low *importance weights* mean that you need many samples to get a good estimate of probability)

- Much more clever approaches to sampling are possible, though mismatch between sampling (proposal) distribution and reality is a constant concern

Bayes Net Summary

- Bayes net = data structure for joint distribution
- Can give exponential reduction in storage
- Variable elimination and variants for tree-ish networks:
  - simple, elegant methods
  - efficient for many networks
- For some networks, must use approximation

- BNs are a major success story for modern AI
  - BNs do the “right” thing (no ugly approximations)
  - Exploit structure in problem to reduce storage/computation
  - Not always efficient, but inefficient cases are well understood
  - Work and used in practice