Bayesian Networks

George Konidaris
gdk@cs.duke.edu

DUKE COMPUTER SCIENCE

Spring 2015

Recall

Joint distributions:
- $P(X_1, \ldots X_n)$.
- *All you (statistically) need to know* about $X_1 \ldots X_n$.
- From it you can infer $P(X_1)$, $P(X_1 \mid Xs)$, etc.

Raining	Cold	Prob.
True	True	0.3
True	False	0.1
False	True	0.4
False	False	0.2

Joint Distributions Are Useful

Classification
- $P(X_1 \mid X_2 \ldots X_n)$

Co-occurrence
- $P(X_a, X_b)$

Rare event detection
- $P(X_1, \ldots Xn)$

Modeling Joint Distributions

Gets large fast
- 2^n entries for *n* binary RVs.

Independence!
- A bit too strong.
- Rarely holds.

Conditional independence.
- Good compromise.

Conditional Independence

A and B are **conditionally independence given C** if:

- P(A | B, C) = P(A | C)
- P(A, B | C) = P(A | C) P(B | C)

(recall independence: P(A, B) = P(A)P(B))

This means that, ***if we know C***, we can treat A and B as independent.

A and B might not be independent otherwise!

Example

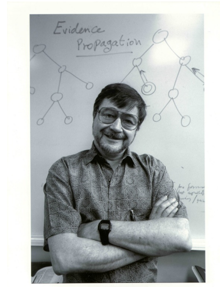
Consider 3 RVs:

- Temperature
- Humidity
- Season

Temperature and humidity are not independent.

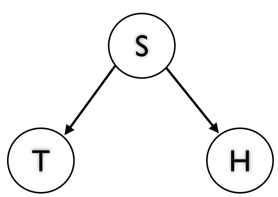
But, they might be, given the season: *the season explains both*, and they become independent of each other.

Bayes Nets



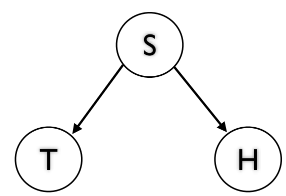
A particular type of graphical model:

- A directed, acyclic graph.
- A node for each RV.



Given parents, each RV independent of non-descendants.

Bayes Net



Probability decomposes:

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

So for each node, store conditional probability table (CPT):

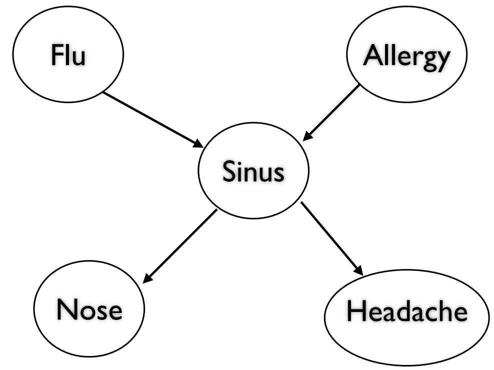
$$P(x_i|\text{parents}(x_i))$$

Example

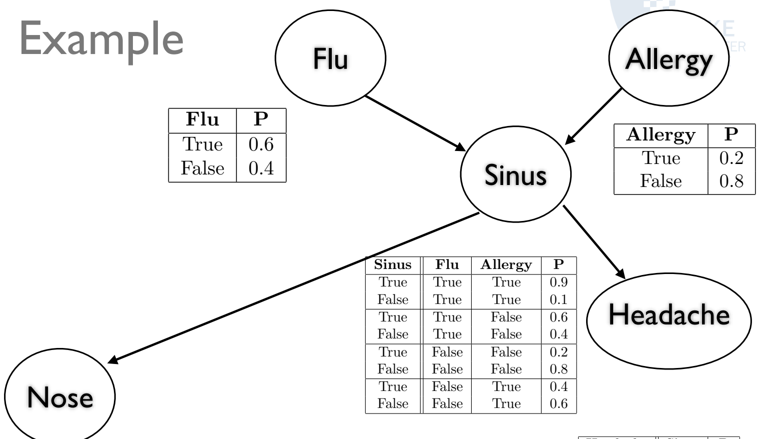
Suppose we know:

- The flu causes sinus inflammation.
- Allergies cause sinus inflammation.
- Sinus inflammation causes a runny nose.
- Sinus inflammation causes headaches.

Example



Example



Flu	P
True	0.6
False	0.4

Allergy	P
True	0.2
False	0.8

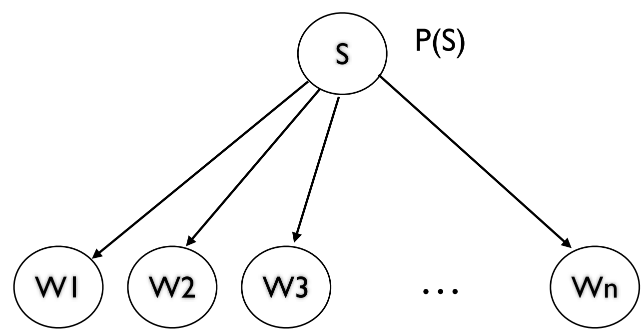
Sinus	Flu	Allergy	P
True	True	True	0.9
False	True	True	0.1
True	True	False	0.6
False	True	False	0.4
True	False	False	0.2
False	False	False	0.8
True	False	True	0.4
False	False	True	0.6

Nose	Sinus	P
True	True	0.8
False	True	0.2
True	False	0.3
False	False	0.7

Headache	Sinus	P
True	True	0.6
False	True	0.4
True	False	0.5
False	False	0.5

joint: 32 (31) entries

Naive Bayes



(spam filter!)

Uses

Things you can do with a Bayes Net:

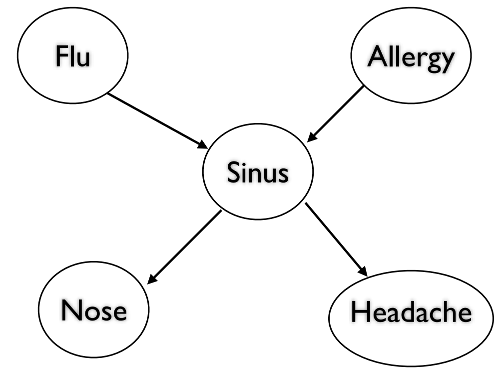
- Inference: given some variables, posterior?
  - (**might be intractable**: NP-hard)
- Learning (fill in CPTs)
- Structure Learning (fill in edges)

Generally:

- Often few parents.
- Inference cost often reasonable.
- Can include domain knowledge.

Inference

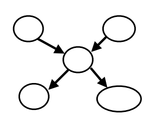
What is:
P(f | h)?



Inference

$$P(f|h) = \frac{P(f,h)}{P(h)} = \frac{\sum_{SAN} P(f,h,S,A,N)}{\sum_{SANF} P(h,S,A,N,F)}$$

We know from definition of Bayes net:



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

$$P(h) = \sum_{SANF} P(h|S)P(N|S)P(S|A,F)P(F)P(A)$$

Variable Elimination

So we have:

$$P(h) = \sum_{SANF} P(h|S)P(N|S)P(S|A,F)P(F)P(A)$$

… we can *eliminate variables* one at a time:
(distributive law)

$$P(h) = \sum_{SN} P(h|S)P(N|S) \sum_{AF} P(S|A,F)P(F)P(A)$$

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

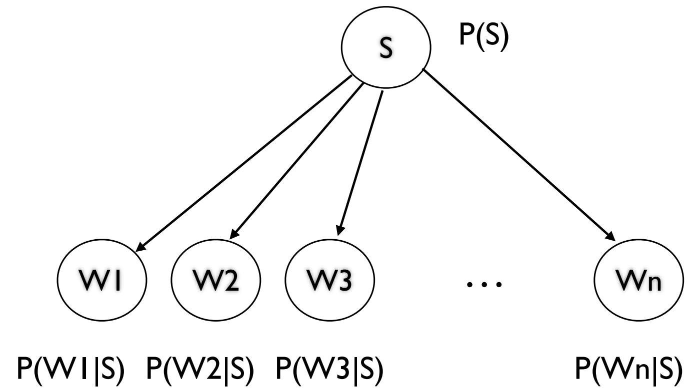
Variable Elimination

Generically:
- Query about X_i and X_j.
- Write out $P(X_1 \dots X_n)$ in terms of $P(X_i \mid \text{parents}(X_i))$
- Sum out all variables except X_i and X_j
- Answer query using joint distribution $P(X_i, X_j)$

Good news:
- Potentially exponential reduction in computation.
- Polynomial for trees.

Bad news:
- Picking variables in optimal order NP-Hard.
- For some networks, no elimination.

Spam Filter (Naive Bayes)



Want $P(S \mid W_1 \dots W_n)$

Naive Bayes

$$P(S|W_1,\dots,W_n) = \frac{P(W_1,\dots,W_n|S)P(S)}{P(W_1,\dots,W_n)}$$

given

$$P(W_1,\dots,W_n|S) = \prod_i P(W_i|S)$$

(from the Bayes Net)

Bayes Nets

Potentially very compressed *but exact.*
- Requires careful construction!

VS

Approximate representation.
- Hope you're not too wrong!

Many, many applications in all areas.