A Brief Introduction to Bayes Nets

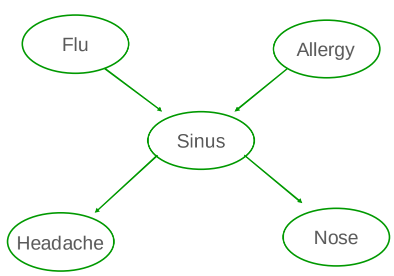
CPS 1/296

Ron Parr

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?

Causal Structure



Knowing sinus separates the variables from each other.

Conditional Independence

- We say that two variables, A and B, are conditionally independent given C if:
  - P(A|BC) = P(A|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

Getting More Formal

- What is a Bayes net?
  - A directed acyclic graph (DAG)
  - Given the parents, each variable is independent of non-descendents
  - Joint probability decomposes:

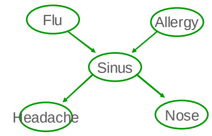
$$P(x_1 \ldots x_n) = \prod_i P(x_i \mid \text{parents}(x_i))$$

  - For each node X_i, store $P(X_i|\text{parents}(X_i))$
  - Represent as table called a CPT

Real Applications of Bayes Nets

- Diagnosis of lymph node disease
- Used in Microsoft office and Windows
  - http://www.research.microsoft.com/research/dtg/
- Used by robots to identify meteorites to study
- Study the human genome:Alex Hartemink et al.
- Many other applications…

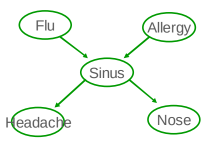
Space Efficiency



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

Atomic Event Probabilities

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



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

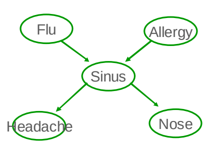
Doing Things the Hard Way

$$P(f \mid h) = \frac{P(fh)}{P(h)} = \frac{\sum_{SAN} P(fhSAN)}{\sum_{SANF} P(hSANF)}$$

defn. of conditional probability   marginalization

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

Working Smarter



$$\begin{aligned}
P(h) &= \sum_{SANF} P(hSANF) \\
&= \sum_{SANF} P(hN \mid SAF) P(SAF) \\
&= \sum_{NS} P(hN \mid S) \sum_{AF} P(SAF) \\
&= \sum_{S} P(h \mid S) \sum_{N} P(N \mid S) \sum_{AF} P(SAF) \\
&= \sum_{S} P(h \mid S) \sum_{N} P(N \mid S) \sum_{AF} P(S \mid AF) P(A) P(F)
\end{aligned}$$

Potential for exponential reduction in computation.

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?

Computational Efficiency

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

The distributive law allows us to decompose the sum.

Potential for an exponential reduction in computation costs.

What Is a Bayes Net, Really?

- A Bayes net is a data structure (with associated algorithms) for fast manipulation of probability distributions
- Bayes nets solve computational problems
- Bayes nets represent; they do not solve

- Q: How often can a bnet solve a computational efficiency problem?

Now the Bad News…

- In full generality: Inference is NP-hard (actually PP complete)
- 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

Checkpoint

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

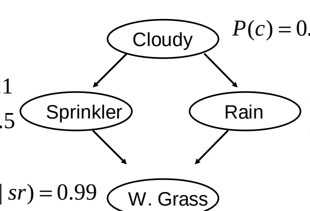
- Hope that worst is case:
  - Avoidable
  - Easily characterized in some way

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

Another Example Network



$P(c) = 0.5$

$P(s|c) = 0.1$
$P(s|\bar{c}) = 0.5$

$P(r|c) = 0.8$
$P(r|\bar{c}) = 0.2$

$P(w|sr) = 0.99$
$P(w|s\bar{r}) = 0.9$
$P(w|\bar{s}r) = 0.9$
$P(w|\bar{s}\bar{r}) = 0.0$

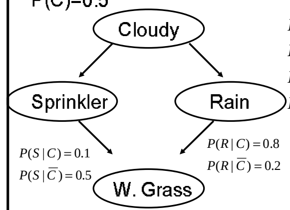
Marginal Probabilities

Suppose we want P(W):

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

Eliminating Cloudy

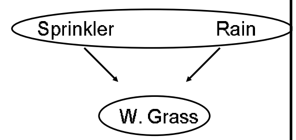
P(C)=0.5



$P(S \mid C) = 0.1$
$P(S \mid \overline{C}) = 0.5$

$P(R \mid C) = 0.8$
$P(R \mid \overline{C}) = 0.2$

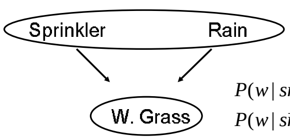
$$P(sr) = 0.5*0.1*0.8 + 0.5*0.5*0.2 = 0.09$$
$$P(s\bar{r}) = 0.5*0.1*0.2 + 0.5*0.5*0.8 = 0.21$$
$$P(\bar{s}r) = 0.5*0.9*0.8 + 0.5*0.5*0.2 = 0.41$$
$$P(\bar{s}\bar{r}) = 0.5*0.9*0.2 + 0.5*0.5*0.8 = 0.29$$



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

Eliminating Sprinkler/Rain

$P(sr) = 0.09$
$P(s\bar{r}) = 0.21$
$P(\bar{s}r) = 0.41$
$P(\bar{s}\bar{r}) = 0.29$



$P(w \mid sr) = 0.99$
$P(w \mid s\bar{r}) = 0.9$
$P(w \mid \bar{s}r) = 0.9$
$P(w \mid \bar{s}\bar{r}) = 0.0$

$$P(w) = \sum_{SR} P(w \mid RS)P(RS)$$
$$= 0.09*0.99 + 0.21*0.9 + 0.41*0.9 + 0.29*0$$
$$= 0.6471$$

Dealing With Evidence

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

$$P(R \mid W) = \alpha P(RW)$$
$$= \alpha \sum_{CS} P(CSRW)$$
$$= \alpha \sum_{CS} P(C)P(S \mid C)P(R \mid C)P(W \mid RS)$$
$$= \alpha \sum_{C} P(R \mid C)P(C) \sum_{S} P(S \mid C)P(W \mid RS)$$

Is there a more clever way to deal with w?

The Variable Elimination Algorithm

```
Elim(bn, query)
If bn.vars = query
  return bn
Else
  x = pick_variable(bn)
  newbn.vars = bn.vars - x
  newbn.vars = newbn.vars - neighbors(x)
  newbn.vars = newbn.vars + newvar
  newbn.vars(newvar).function =
```

$$\sum_{X} \prod_{Y \in X \cup neighbors(X)} \text{bn.vars(Y) .function}$$

```
  return(elim(newbn, query))
```

Efficiency of Variable Elimination

- Exponential in the largest domain size of new variables created
- Equivalently: Exponential in largest function created by pushing in summations

- Linear for trees
- Almost linear for almost trees ☺
- (See examples on board…)

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
  - Note 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
  - Variational methods

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

Bayes Net Summary

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