Bayesian Networks

CompSci 270
Duke University
Ron Parr

Why Joint Distributions are Important

• Joint distributions gives $P(X_1...X_n)$

• Classification/Diagnosis
  – Suppose $X_1$=disease
  – $X_2...X_n$ = symptoms

• Co-occurrence
  – Suppose $X_3$=lung cancer
  – $X_5$=smoking

• Rare event Detection
  – Suppose $X_1...X_n$ = parameters of a credit card transaction
  – Call card holder if $P(X_1...X_n)$ 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
- Factor graphs
- Belief propagation
- Undirected models
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?)

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!
  – $P(A=\text{true}|B=\text{true}), P(A=\text{true}|B=\text{false}),$
    $P(A=\text{false}|B=\text{true}), P(A=\text{false}|B=\text{false})$
• $P(A|b)$ 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)
• Given parents, each variable is conditionally independent of non-descendants
• Joint probability decomposes:
  \[ P(x_1...x_n) = \prod_i P(x_i \mid \text{parents}(x_i)) \]
• For each node $X_i$, store $P(X_i \mid \text{parents}(X_i))$
• 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...

Space Efficiency

- Entire joint distribution as 32 (31) entries
  - $P(H|S), P(N|S)$ have 4 (2)
  - $P(S|AF)$ has 8 (4)
  - $P(A)$ has 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. $4n+2(2n+1)$

---

How To Build A Bayes Net

- Can always construct a valid Bayes net by inserting variables one at a time
- How to pick edges for new variable?
  - Parents of each new variable are all variables that influence the distribution of the new variable
  - No need (yet) for outgoing edges from new variable
- Can prove by induction that this preserves property that all variables are conditionally independent of non-descendants given parents
(Non)Uniqueness of Bayes Nets

- Order of adding variables can lead to different Bayesian networks for the same distribution
- Suppose A and B are independent, but C is a function of A and B
  - Add A, B, then C: \( \text{parents}(A)=\text{parents}(B)=\emptyset, \text{parents}(C)=(A,B) \)
  - Add C, A, then B: \( \text{parents}(C)=\emptyset, \text{parents}(A)=(C), \text{parents}(B)=(A,C) \)

Suppose A and B are uniform, \( C=(A \lor B) \)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.25</td>
</tr>
</tbody>
</table>

\( P(A)=P(\overline{A})=P(B)=P(\overline{B})=0.5 \)
\( P(C)=0.75, P(\overline{C})=0.25 \)
\( P(ab)=P(a\overline{b})=P(\overline{a}b)=0.25 \)
\( P(c|ab)=P(c|a\overline{b})=P(c|\overline{a}b)=1.0, P(c|\overline{a}\overline{b})=0 \)

(only showing \( c=\text{true case} \))

Add A, then B, then C case
Suppose A and B are uniform, C=(A V B)

\[
\begin{array}{cccc}
A & B & C & P \\
0 & 0 & 0 & 0.25 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0.25 \\
1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0.25 \\
1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0.25 \\
\end{array}
\]

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

Add C, then A, then B case
(only showing b=true case)

Atomic Event Probabilities

\[P(x_1 \ldots x_n) = \prod_{i} P(x_i | \text{parents}(x_i))\]

True for the distribution P(X_1\ldots X_n) and for any individual setting of the variables P(x_1\ldots x_n)
Answering Queries Using Marginalization

\[ P(f \mid h) = \frac{P(fh)}{P(h)} = \sum_{SAN} P(fhSAN) = \sum_{SAN} P(F)P(A)P(S \mid AF)P(h \mid S)P(N \mid S) \]

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

Digression: Notation I

- \( \sum_A P(AB) = P(B) \), a distribution over B
- \( \sum_A P(Ab) = P(b) \), a probability
- \( P(A)P(B) \) = a function (lookup table of size 4) from combinations of assignments to A and B to numbers (if A, an B are independent, the this function is P(AB))
Digression: Notation II

- \[ \sum_{BC} P(A|B)P(B|C)P(C) = P(A), \text{ a distribution over } A \]

- How do we compute this?

- For each value of A we sum over all combinations of B and C

Digression: Notation III

- \[ \sum_{C} P(A|BC)P(B|C)P(C) = \text{ a function of } A \text{ and } B \text{ (table over 4 numbers)} \]

- How do we compute this?

- For each combination of A and B we sum over all values of C
Working Smarter

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

Potential for exponential reduction in computation.

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.
AKA: Variable elimination

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:

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

Naïve Bayes Efficiency II

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:

$$P(W_1 \ldots W_n | S) = \prod_{i=1}^{n} P(W_i | S)$$
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
Reduction

\[(\overline{X_1} \lor X_2 \lor X_3) \land (\overline{X_2} \lor X_3 \lor X_4) \land \ldots\]

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

Cloudy
- $P(c) = 0.5$
- $P(s | c) = 0.1$
- $P(s | \bar{c}) = 0.5$
- $P(w | sr) = 0.99$
- $P(w | s\bar{r}) = 0.9$
- $P(w | \bar{s}r) = 0.9$
- $P(w | \bar{s}\bar{r}) = 0.0$

Sprinkler

Rain

W. Grass

$P(r | c) = 0.8$

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

Suppose we want $P(W)$:

$$P(W) = \sum_{CSR} P(CSRW)$$

$$= \sum_{CSR} P(C)P(S | C)P(R | C)P(W | RS)$$

$$= \sum_{SR} P(W | RS) \sum_{C} P(S | C)P(C)P(R | C)$$

A function of $R$ and $S$

Eliminating Cloudy

$P(C) = 0.5$

$P(sr) = 0.5 \times 0.1 \times 0.8 + 0.5 \times 0.5 \times 0.2 = 0.09$

$P(s\overline{r}) = 0.5 \times 0.1 \times 0.2 + 0.5 \times 0.5 \times 0.8 = 0.21$

$P(\overline{s}r) = 0.5 \times 0.9 \times 0.8 + 0.5 \times 0.5 \times 0.2 = 0.41$

$P(\overline{s}\overline{r}) = 0.5 \times 0.9 \times 0.2 + 0.5 \times 0.5 \times 0.8 = 0.29$
Eliminating Sprinkler/Rain

\[ P(sr) = 0.09 \]
\[ P(s\overline{r}) = 0.21 \]
\[ P(\overline{s}r) = 0.41 \]
\[ P(\overline{s}\overline{r}) = 0.29 \]
\[ P(w | sr) = 0.99 \]
\[ P(w | s\overline{r}) = 0.9 \]
\[ P(w | \overline{s}r) = 0.9 \]
\[ P(w | \overline{s}\overline{r}) = 0.0 \]

\[
P(w) = \sum_{sr} P(w | RS)P(RS)
\]
\[
= 0.09 \times 0.99 + 0.21 \times 0.9 + 0.41 \times 0.9 + 0.29 \times 0
\]
\[
= 0.6471
\]

Dealing With Evidence

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

\[
P(R | W) = \alpha P(RW)
\]
\[
= \alpha \sum_{CS} P(CSRW)
\]
\[
= \alpha \sum_{CS} P(C)P(S | C)P(R | C)P(W | RS)
\]
\[
= \alpha \sum_{C} P(R | C)P(C)\sum_{S} P(S | C)P(W | RS)
\]

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
- 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

Review: Computing Marginal Probabilities

• Suppose you want $P(X_1X_2)$

• Write: $P(X_1X_2) = \sum_{X_3...X_n} P(X_1...X_n)$

• Using Bayes net: 
  $$P(X_1X_2) = \sum_{X_3...X_n} \Pi_i P(X_i | \text{parents}(X_i))$$

• Distribute sum over product
Key Things to Remember

- All variables summed out must be to the right of your summation
- The result of your summation should be a function of the remaining variables

\[
P(W) = \sum_{CSR} P(CSRW) = \sum_{CSR} P(C)P(S|C)P(R|C)P(W|RS) = \sum_{SR} P(W|RS) \sum_{C} P(S|C)P(C)P(R|C)
\]

No terms with C here!  
Result is a function of R and S

Beyond Variable Elimination I

- 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

- Intuition:

\[
\begin{align*}
&\text{C} \\
&\text{S} \quad \text{R} \\
&\text{W} \\
\end{align*}
\]

\[
\begin{align*}
&\text{C} \\
&\text{SR} \\
&\text{W}
\end{align*}
\]
Beyond Variable Elimination II

- Compiling can be expensive if we are forced to make big clusters
- Big clusters correspond to inefficient variable elimination
- (But no guarantee that efficient elimination is possible in general)

- 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 $P(F)$

Flip a coin to decide $A$ based on $P(A)$

Flip a coin to decide $S$ based on $P(S| \overline{F})$
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 baked in)
  – Exploit structure in problem to reduce storage/computation
  – Not always efficient, but inefficient cases are well understood
  – Work and used in practice