Why Joint Distributions are Important

• Joint distribution gives \( P(X_1, ..., X_n) \)

• Classification/Diagnosis
  – Suppose \( X_1 = \text{disease} \)
  – \( X_2, ..., X_n = \text{symptoms} \)
  – \( P(X_1 | X_2, ..., X_n) \)

• Co-occurrence
  – Suppose \( X_3 = \text{lung cancer} \)
  – \( X_5 = \text{smoking} \)
  – (Precursor to determining causality)

• Rare event Detection
  – Suppose \( X_1, ..., X_n = \text{parameters of a credit card transaction} \)
  – 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
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

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!
Nota>on

Reminder

• $P(A|B)$ is a conditional prob. distribution
  – It is a function!
  – $P(A=\text{true}|B=\text{true})$, $P(A=\text{true}|B=\text{false})$, $P(A=\text{false}|B=\text{true})$, $P(A=\text{false}|B=\text{true})$

• $P(A|b)$ 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)
• Given parents, each variable is independent of non-descendants
• Joint probability decomposes:
  \[ P(x_1 \ldots x_n) = \prod_i P(x_i | \text{parents}(x_i)) \]
• 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 in Microsoft office and Windows
• Used by robots to identify meteorites to study
• Study the human genome: Alex Hartemink et al.
• 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. $4n+2(2n+1)$

2 Ways To Think About BNs

- A (potentially) compressed but exact representation of a probability distribution
- Requires carefully constructing BN so that it respects conditional independence requirements of BN definition
- An approximate representation of a distribution
- Requires designing a BN structure, estimating CPTs, hoping you get answers that aren’t too wrong
- Reality is usually in the middle

Atomic Event Probabilities

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

Atomic event probs guaranteed correct true if we construct net incrementally, so that for each new variable added, we connect all influencing variables as parents (prove it by induction)

(Non)Uniqueness of Bayes Nets

- You can always construct a valid Bayes net by inserting variables one at a time
- 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:
  - Add C, A, then B:
Doing Things the Hard Way

\[ P(f|h) = \frac{P(fh)}{P(h)} = \sum_{SANF} P(fhSANF) = \sum_{SANF} P(FAP(S|Af)P(h|S)P(N|S) \]  

defn. of conditional probability

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

Working Smarter

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

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  
Analogous to Gaussian elimination but exponentially more expensive

Naïve Bayes Efficiency

Given a set of words, we want to know which is larger: \( P(s|w_1...w_n) \) or \( P(\neg s|w_1...w_n) \).

Use Bayes Rule:  
\[ P(S|W_1...W_n) = \frac{P(W_1...W_n|S)P(S)}{P(W_1...W_n)} \]
Naïve Bayes Efficiency II
(Using Bayes Net Structure explicitly)

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

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:
\[
P(W_1 \ldots W_n | S) = \prod_{i=1}^{n} P(W_i | S)
\]

Naïve Bayes Efficiency
(in terms of variable elimination)

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

\[
P(s | w_1 \ldots w_n) = \frac{P(sw_1 \ldots sw_n)}{P(w_1 \ldots w_n)}
\]

\[
= P(s | w_i) \prod_{i=1}^{n} P(w_i | s)
\]

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

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
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

\[
P(c) = 0.5, \quad P(r \mid c) = 0.8, \quad P(r \mid \overline{c}) = 0.2
\]

\[
P(s \mid c) = 0.1, \quad P(s \mid \overline{c}) = 0.5
\]

\[
P(w \mid sr) = 0.99, \quad P(w \mid s\overline{r}) = 0.9
\]

\[
P(w \mid s\overline{r}) = 0.9, \quad P(w \mid \overline{s}r) = 0.0
\]

\[
P(r \mid c) = 0.8, \quad P(r \mid \overline{c}) = 0.2
\]

\[
P(s \mid c) = 0.1, \quad P(s \mid \overline{c}) = 0.5
\]

\[
P(w \mid sr) = 0.99, \quad P(w \mid s\overline{r}) = 0.9
\]

\[
P(w \mid s\overline{r}) = 0.9, \quad P(w \mid \overline{s}r) = 0.0
\]

\[
P(r \mid c) = 0.8, \quad P(r \mid \overline{c}) = 0.2
\]
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 (Sprinkler,Rain)

$P(sr) = 0.09$
$P(sr) = 0.21$
$P(sr) = 0.41$
$P(sr) = 0.29$

$P(w) = \sum_{SR} P(w \mid RS)P(RS)$

$$= 0.09*0.99 + 0.21*0.9 + 0.41*0.9 + 0.29*0$$

$$= 0.6471$$

Eliminating Cloudy

$P(C)=0.5$

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

$P(sr) = 0.5 * 0.1 * 0.2 + 0.5 * 0.5 * 0.8 = 0.21$

$P(sr) = 0.5 * 0.9 * 0.8 + 0.5 * 0.5 * 0.2 = 0.41$

$P(sr) = 0.5 * 0.9 * 0.2 + 0.5 * 0.5 * 0.8 = 0.29$

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

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

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

Dealing With Evidence

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

$$P(R \mid W) = \alpha P(RW)$$

$$= \alpha \sum_{CS} P(CSRW)$$

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

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

Is there a more clever way to deal with $w$?

Only keep the relevant parts.
Cookbooking Bayes Net Queries

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 😊

Cookbooking Bayes Net Queries Using Variable Elimination

1. Given: Probability query involving a subset of the variables, e.g., $X_j$ and $X_k$
2. Write out $P(X_1...X_n)$ in terms of $P(X_i|\text{parents}(X_i))$
3. Sum out all variables except $X_j$ and $X_k$ to get joint distribution over $X_j$ and $X_k$
4. Answer query using joint distribution over $X_j$ and $X_k$

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
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

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

• More clever approaches to sampling are possible

Maximizing

• Setting of variables with the highest probability?
• Can distribute max, just like product:

\[
\sum_S P(H|S) \sum_{AF} P(S|AF) P(F) P(A) \sum_N P(N|S)
\]

\[
\max_{S,A,F,N} P(H|S) \max_{AF} P(S|AF) P(F) P(A) \max_N P(N|S)
\]

• Note: This only gives us the probability of the highest prob assignment; need work backwards to recover the actual assignment
• (Similar to recovering path in search)

Summary of Algorithms for BNs

• Enumeration (consider all atomic events)
  – Exponential
  – Yuck!
• Variable elimination
  – Can be dramatically more efficient than enumeration
  – No guarantee of polynomial time
• Sampling
  – Easy to implement
  – May converge slowly in practice
• Can modify algorithms to compute max prob. assignments
Other Algorithms

- **Exact:**
  - Can “compile” a non-tree BN into an equivalent “factor graph” by clustering variables
  - No free lunch: Factors may be very large

- **Approximate:**
  - Can split up large clusters, or remove arcs
  - “Loopy” methods that treat graphs like trees
  - Variational approximations

Bayes Net Summary

- Bayes net = data structure for joint distribution
- Can give exponential reduction in storage
- Variation 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