General probabilistic inference

- **Query:** $P(X | e) \frac{P(X,e)}{P(e)}$
- **Using Bayes rule:** 
  
  $P(X | e) \propto P(X,e)$
- **Normalization:** 
  
  $P(A \mid H=t, N=t) = \frac{0.2}{0.2 + 0.15} = ...$

Given a Bayesian network with nodes for Flu, Allergy, Sinus, Headache, and Nose, the inference process involves calculating the probability of certain events given evidence. The query focuses on the conditional probability of a particular symptom given a set of evidence, and the calculation uses Bayes' rule along with normalization.
Marginalization

\[
P(F, S, N) = P(F) \cdot P(S|F) \cdot P(N|S, F)
\]

\[
P(F | N = t) \propto P(F, S|N = t) = P(F = t, S = t, N = t) + P(F = t, S = f, N = t)
\]

\[
= \sum_s P(F = t, s, N = t)
\]

\[
= \sum_s P(F = t) \cdot P(s | F = t) \cdot P(N = t | s)
\]

\[
P(F = t) \sum_s P(s | F = t) P(N = t | s)
\]

Probabilistic inference example

\[
P(F | N = t) \propto P(F, N = t)
\]

\[
P(F, A, S, H, N) = P(F) \cdot P(A) \cdot P(S|F) \cdot P(H|S) \cdot P(N|S, F)
\]

\[
P(F, N = t) = \sum_{A, S, H} P(F, A, S, H, N = t)
\]

\[
\sum_{x_1, x_2, \ldots, x_5} = 2^5 = 32 \text{ terms}
\]

Inference seems exponential in number of variables!
Actually, inference in graphical models is NP-hard 😞
Fast probabilistic inference example – Variable elimination

(Potential for) Exponential reduction in computation!

Understanding variable elimination – Exploiting distributivity

\[ a(s+c) = ab + ac \]
Understanding variable elimination –
Order can make a HUGE difference

\[
P(F, N=\ell) = \sum_{a, s, h} P(F) \cdot P(a) \cdot P(s|F, a) \cdot P(h|a) \cdot P(N=\ell|s, a, h)
\]

Order matters:

\[
P(F, a, h) \cdot \sum_{\ell} P(s|F, a) \cdot P(h|a) \cdot P(N=\ell|s, a, h)
\]

Another example:

\[
P(Y, N=\ell) = \sum_{s, h, c} P(s) \cdot P(h|s) \cdot P(N=\ell|s, h) \cdot P(Y|h, c)
\]

"lose" independence: a priori \( s \| Y \| H \)

But after eliminating \( H \):

\[
P(Y, N=\ell) = \sum_{s, c} P(s) \cdot P(N=\ell|s) \cdot \sum_{c} P(c|N=\ell, s) g_2(s, Y)
\]
Variable elimination algorithm

- Given a BN and a query \( P(X|e) \) / \( P(X,e) \)
- Instantiate evidence \( e \) in every!
- Choose an ordering on variables, e.g., \( X_1, \ldots, X_n \)
- For \( i = 1 \) to \( n \), if \( X_i \not\in \{X,e\} \), eliminate \( X_i \):
  - Collect factors \( f_1, \ldots, f_k \) that include \( X_i \)
  - Generate a new factor by eliminating \( X_i \) from these factors

\[
g = \sum_{X_i} \prod_{j=1}^{k} f_j
\]
- Variable \( X_i \) has been eliminated!
- Normalize \( P(X,e) \) to obtain \( P(X|e) \)

IMPORTANT!!!

Complexity of variable elimination – (Poly)-tree graphs

- (no loops in graph)
  - Inference in linear time
- Variable elimination order:
  - Start from “leaves” up –
    - find topological order, eliminate variables in reverse order
  - Eliminate \( X_i \):
    - \( \sum_{a} P(a).P(c|a) = g_i(c) \)
    - never generate any factor that is larger than an original CPT
    - \( \ell \) running time linear in number of nodes \( \ell \)

Linear in number of variables!!! (versus exponential)
Complexity of variable elimination –
Graphs with loops

\[ g = \sum_{i} \prod_{j=1}^{k} f_{ij} \]

how many vars appear in \( g \):

\[ |g(x_1, x_2, \ldots, x_m)\] \( \leq 2^m \) terms

A binary

Exponential in number of variables in largest factor generated

Complexity of variable elimination –
Tree-width

Moralize graph:
Connect parents into a clique and remove edge directions

Tree-width: \( \max\{d_1, d_2, \ldots, d_n\} \) (not neighbors)

\( \min \) number nodes I remove to separate

Complexity of VE elimination:
(“Only”) exponential in tree-width
Tree-width is maximum node cut +1
Example: Large tree-width with small number of parents

- Compact representation → Easy inference

Choosing an elimination order

- Choosing best order is NP-complete
  - Reduction from MAX-Clique
- Many good heuristics (some with guarantees)
- Ultimately, can’t beat NP-hardness of inference
  - Even optimal order can lead to exponential variable elimination computation
- In practice
  - Variable elimination often very effective
  - Many (many many) approximate inference approaches available when variable elimination too expensive
Announcements

- HW4 out later today
  - will come out in two installments, no interest
  - no hidden fees

- Project milestone
  - Next Monday (11/12 in class)

HMMs

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University
November 5th, 2007
Adventures of our BN hero

- Compact representation for probability distributions
- Fast inference
- Fast learning

But... Who are the most popular kids?

1. Naïve Bayes
2 and 3. Hidden Markov models (HMMs) Kalman Filters

Handwriting recognition

Character recognition, e.g., kernel SVMs
Example of a hidden Markov model (HMM)

Understanding the HMM Semantics
HMMs semantics: Details

Just 3 distributions:

\[
P(X_1)
\]

\[
P(X_i \mid X_{i-1})
\]

\[
P(O_i \mid X_i)
\]
Learning HMMs from fully observable data is easy

\[ P(X_1) \]
\[ P(O_i \mid X_i) \]
\[ P(X_i \mid X_{i-1}) \]

Learn 3 distributions:

Possible inference tasks in an HMM

Marginal probability of a hidden variable:

Viterbi decoding – most likely trajectory for hidden vars:
Using variable elimination to compute $P(X_i|o_1:n)$

$X_1 = \{a, \ldots, z\}$
$O_1 = b$
$X_2 = \{a, \ldots, z\}$
$O_2 = c$
$X_3 = \{a, \ldots, z\}$
$O_3 = d$
$X_4 = \{a, \ldots, z\}$
$O_4 = e$
$X_5 = \{a, \ldots, z\}$
$O_5 = f$

Variable elimination order?

Example:

What if I want to compute $P(X_i|o_1:n)$ for each $i$?

Variable elimination for each $i$?

Variable elimination for each $i$, what's the complexity?
Reusing computation

Compute:

\[ P(X_i \mid o_{1..n}) \]

The forwards-backwards algorithm

- **Initialization:** \( \alpha_1(X_1) = P(X_1)P(o_1 \mid X_1) \)
- For \( i = 2 \) to \( n \)
  - Generate a forwards factor by eliminating \( X_{i-1} \)
  \[
  \alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i)P(X_i \mid X_{i-1} = x_{i-1})\alpha_{i-1}(x_{i-1})
  \]
- **Initialization:** \( \beta_n(X_n) = 1 \)
- For \( i = n-1 \) to \( 1 \)
  - Generate a backwards factor by eliminating \( X_{i+1} \)
  \[
  \beta_i(X_i) = \sum_{x_{i+1}} P(o_{i+1} \mid x_{i+1})P(x_{i+1} \mid X_i)\beta_{i+1}(x_{i+1})
  \]
- \( 8 \) \( i \), probability is:
  \[
  P(X_i \mid o_{1..n}) \propto \alpha_i(X_i)\beta_i(X_i)
  \]
What you’ll implement 1: multiplication

\[ \alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i) P(X_i \mid X_{i-1} = x_{i-1}) \alpha_{i-1}(x_{i-1}) \]

What you’ll implement 2: marginalization

\[ \alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i) P(X_i \mid X_{i-1} = x_{i-1}) \alpha_{i-1}(x_{i-1}) \]
Higher-order HMMs

Add dependencies further back in time!
better representation, harder to learn

What you need to know

- Hidden Markov models (HMMs)
  - Very useful, very powerful!
  - Speech, OCR, …
  - Parameter sharing, only learn 3 distributions
  - Trick reduces inference from $O(n^2)$ to $O(n)$
  - Special case of BN