Graphical Models

Aarti Singh & Barnabas Poczos
Slides Courtesy: Carlos Guestrin

Machine Learning 10-701/15-781
Mar 25, 2014
Topics in Graphical Models

• Representation
  – Which joint probability distributions does a graphical model represent?

• Inference
  – How to answer questions about the joint probability distribution?
    • Marginal distribution of a node variable
    • Most likely assignment of node variables

• Learning
  – How to learn the parameters and structure of a graphical model?
Inference

• Possible queries:
  1) Marginal distribution e.g. $P(S)$
     Posterior distribution e.g. $P(F|H=1)$

  2) Most likely assignment of nodes
     $\arg \max_{f,a,s,n} P(F=f,A=a,S=s,N=n | H=1)$
Inference

• Possible queries:
  1) Marginal distribution e.g. $P(S)$
  Posterior distribution e.g. $P(F|H=1)$

$$P(F|H=1) = \frac{P(F, H=1)}{P(H=1)}$$

$$= \frac{P(F, H=1)}{\sum_{f} P(F=f, H=1)}$$

$\propto P(F, H=1)$

will focus on computing this, posterior will follow with only constant times more effort
Marginalization

Need to marginalize over other vars

\[ P(S) = \sum_{f,a,n,h} P(f,a,S,n,h) \]

\[ P(F,H=1) = \sum_{a,s,n} P(F,a,s,n,H=1) \]

To marginalize out \( n \) binary variables, need to sum over \( 2^n \) terms

Inference seems exponential in number of variables!

Actually, inference in graphical models is NP-hard 😞
Bayesian Networks Example

- 18 binary attributes

- Inference
  - $P(\text{BatteryAge} | \text{Starts}=f)$

- need to sum over $2^{16}$ terms!

- Not impressed?
  - HailFinder BN – more than $3^{54} = 58149737003040059690390169$ terms
Fast Probabilistic Inference

\[ P(F, H=1) = \sum_{a,s,n} P(F,a,s,n,H=1) \]

\[ = \sum_{a,s,n} P(F)P(a)P(s|F,a)P(n|s)P(H=1|s) \]

\[ = P(F) \sum_{a} P(a) \sum_{p} P(s|F,a)P(H=1|s) \sum_{n} P(n|s) \]

Push sums in as far as possible

Distributive property: \[ x_1z + x_2z = z(x_1 + x_2) \]

2 multiply \quad 1 multiply
Fast Probabilistic Inference

\[ P(F, H=1) = \sum_{a,s,n} P(F, a, s, n, H=1) \]
\[ = \sum_{a,s,n} P(F) P(a) P(s \mid F, a) P(n \mid s) P(H=1 \mid s) \]
\[ = P(F) \sum_a P(a) \sum_s P(s \mid F, a) P(H=1 \mid s) \sum_n P(n \mid s) \]
\[ = P(F) \sum_a P(a) g_1(F, a) \]
\[ = P(F) g_2(F) \]

(Potential for) exponential reduction in computation!

Flu

Allergy

Sinus

Headache

Nose

\(2^n \text{ vs. } n \ 2^k \text{ multiplies}\)
\(k - \text{scope of (number of variables in) largest factor}\)
Fast Probabilistic Inference – Variable Elimination

\[ P(F, H=1) = \sum_{a,s,n} P(F)P(a)P(s|F,a)P(n|s)P(H=1|s) \]

\[ = P(F) \sum_a P(a) \sum_s P(s|F,a)P(H=1|s) \sum_n P(n|s) \]

(Potential for) exponential reduction in computation!
Variable Elimination – Order can make a HUGE difference

\[
P(F, H=1) = \sum_{a,s,n} P(F)P(a)P(s|F,a)P(n|s)P(H=1|s)
\]

\[
= P(F) \sum_a P(a) \sum_s P(s|F,a)P(H=1|s) \sum_n P(n|s)
\]

\[
P(F, H=1) = P(F) \sum_a P(a) \sum_n \sum_s P(s|F,a)P(n|s)P(H=1|s)
\]

(Potential for) exponential reduction in computation!
Variable Elimination – Order can make a HUGE difference

\[ P(X_1) = \sum_{Y,X_2,\ldots,X_n} P(Y)P(X_1|Y) \prod_{i=2}^{n} P(X_i|Y) \]

\[ = \sum_{Y,X_3,\ldots,X_n} P(Y)P(X_1|Y) \prod_{i=3}^{n} P(X_i|Y) \sum_{X_2} P(X_2|Y) \]

\[ = \sum_{X_2,\ldots,X_n} \sum_{Y} P(Y)P(X_1|Y) \prod_{i=2}^{n} P(X_i|Y) \]

1 - scope of largest factor

\[ g(Y) \]

n - scope of largest factor

\[ g(X_1,X_2,\ldots,X_n) \]
Variable Elimination Algorithm

- Given BN – DAG and CPTs (initial factors – p(x_i|pa_i) for i=1,..,n)
- Given Query P(X|e) ≡ P(X,e)   X – set of variables e - evidence
- Instantiate evidence e  e.g. set H=1
- Choose an ordering on the variables e.g., X(1), ..., X(n)
- For i = 1 to n, If X(i) ∉{X,e} (i.e. need to marginalize it out)
  - Collect factors g_1,...,g_k that include X(i)
  - Generate a new factor by eliminating X(i) from these factors
    \[ g = \sum_{X_i} \prod_{j=1}^{k} g_j \]
  - Variable X(i) has been eliminated!
  - Remove g_1,...,g_k from set of factors but add g
- Normalize P(X,e) to obtain P(X|e)
Variable elimination order:
- Consider undirected version (ignore edge directions)
- Start from “leaves” up
- find topological order
- eliminate variables in that order

Does not create any factors bigger than original CPTs

For polynomials, inference is linear in # variables (vs. exponential in general)!
Complexity for graphs with loops

- Loop – undirected cycle

Linear in # variables but exponential in size of largest factor generated!

Moralize graph (connect parents into a clique & drop direction of all edges)

When you eliminate a variable, add edges between its neighbors
Complexity for graphs with loops

- Loop – undirected cycle

Var eliminated
S
B
D
C
T
O

Factor generated
$g_1(C,B)$
$g_2(C,O,D)$
$g_3(C,O)$
$g_4(T,O)$
$g_5(O)$
$g_6(X)$

Linear in # variables but exponential in size of largest factor generated ~ tree-width (max clique size-1) in resulting graph!
Example: Large tree-width with small number of parents

At most 2 parents per node, but tree width is \(O(\sqrt{n})\)

Compact representation \(\not\Rightarrow\) 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
Inference

- Possible queries:
  2) Most likely assignment of nodes
     \[ \text{arg max } P(F=f,A=a,S=s,N=n|H=1) \]

Use Distributive property:
\[ \max(x_1 z, x_2 z) = z \max(x_1, x_2) \]

2 multiply 1 mulitply
Topics in Graphical Models

• Representation
  – Which joint probability distributions does a graphical model represent?

• Inference
  – How to answer questions about the joint probability distribution?
    • Marginal distribution of a node variable
    • Most likely assignment of node variables

• Learning
  – How to learn the parameters and structure of a graphical model?
Given set of $m$ independent samples (assignments of random variables),

find the best (most likely?) Bayes Net (graph Structure + CPTs)
Learning the CPTs (given structure)

For each discrete variable $X_k$

Compute MLE or MAP estimates for

$$p(x_k | \text{pa}_k)$$

Recall

**MLE:** $P(X_i = x_i \mid X_j = x_j) = \frac{\text{Count}(X_i = x_i, X_j = x_j)}{\text{Count}(X_j = x_j)}$

**MAP:** Add pseudocounts
MLEs decouple for each CPT in Bayes Nets

• Given structure, log likelihood of data

\[
\log P(D \mid \theta \mathcal{G}, \mathcal{G})
\]

\[
= \log \prod_{j=1}^{m} P(f^{(j)}) P(a^{(j)}) P(s^{(j)} \mid f^{(j)}, a^{(j)}) P(h^{(j)} \mid s^{(j)}) P(n^{(j)} \mid s^{(j)})
\]

\[
= \sum_{j=1}^{m} \left[ \log P(f^{(j)}) + \log P(a^{(j)}) + \log P(s^{(j)} \mid f^{(j)}, a^{(j)}) + \log P(h^{(j)} \mid s^{(j)}) + \log P(n^{(j)} \mid s^{(j)}) \right]
\]

\[
= \sum_{j=1}^{m} \log P(f^{(j)}) + \sum_{j=1}^{m} \log P(a^{(j)}) + \sum_{j=1}^{m} \log P(s^{(j)} \mid f^{(j)}, a^{(j)}) + \sum_{j=1}^{m} \log P(h^{(j)} \mid s^{(j)}) + \sum_{j=1}^{m} \log P(n^{(j)} \mid s^{(j)})
\]

Depends only on \( \theta_F \), \( \theta_A \), \( \theta_{S \mid F, A} \), \( \theta_{H \mid S} \), and \( \theta_{N \mid S} \).

Can computer MLEs of each parameter independently!
Information theoretic interpretation of MLE

\[
\log P(D \mid \theta_G, G) = \sum_{j=1}^{m} \sum_{i=1}^{n} \log P \left( X_i = x_i^{(j)} \mid \text{Pa}_{X_i} = x_{\text{Pa}X_i}^{(j)} \right)
\]

\[
= \sum_{i=1}^{n} \sum_{x_i} \sum_{x_{\text{Pa}X_i}} \text{count}(X_i = x_i, \text{Pa}_{X_i} = x_{\text{Pa}X_i}) \log P \left( X_i = x_i \mid \text{Pa}_{X_i} = x_{\text{Pa}X_i} \right)
\]

Plugging in MLE estimates: ML score

\[
\log \hat{P}(D \mid \hat{\theta}_G, G) = \sum_{j=1}^{m} \sum_{i=1}^{n} \log \hat{P} \left( x_i^{(j)} \mid x_{\text{Pa}X_i}^{(j)} \right)
\]

\[
= m \sum_{i=1}^{n} \sum_{x_i} \sum_{x_{\text{Pa}X_i}} \hat{P}(x_i, x_{\text{Pa}X_i}) \log \hat{P} \left( x_i \mid x_{\text{Pa}X_i} \right)
\]

Reminds of entropy
Information theoretic interpretation of MLE

\[
\log \hat{P}(\mathcal{D} \mid \hat{\theta}_G, \mathcal{G}) = m \sum_{i=1}^{n} \sum_{x_i} \sum_{x_{\text{Pa}_X_i}} \hat{P}(x_i, x_{\text{Pa}_X_i}) \log \hat{P}(x_i \mid x_{\text{Pa}_X_i}) 
\]

\[
= -m \sum_{i=1}^{n} \hat{H}(X_i \mid \text{Pa}_{X_i}) 
\]

\[
= m \sum_{i=1}^{n} [\hat{I}(X_i, \text{Pa}_{X_i}) - \hat{H}(X_i)]
\]

ML score for graph structure $\mathcal{G}$

\[
\arg \max_{\mathcal{G}} \log \hat{P}(\mathcal{D} \mid \hat{\theta}_G, \mathcal{G}) = \arg \max_{\mathcal{G}} \sum_{i=1}^{n} \hat{I}(X_i, \text{Pa}_{X_i})
\]

Doesn’t depend on graph structure $\mathcal{G}$
ML – Decomposable Score

• Log data likelihood

\[ \log \hat{P}(D \mid \hat{\theta}_g, \mathcal{G}) = m \sum_{i=1}^{n} [\hat{I}(X_i, \text{Pa}_{X_i}) - \hat{H}(X_i)] \]

• Decomposable score:
  – Decomposes over families in BN (node and its parents)
  – Will lead to significant computational efficiency!!!
  – Score\((G : D) = \sum_i \text{FamScore}(X_i \mid \text{Pa}_{X_i} : D)\)
How many trees are there?

- Trees – every node has at most one parent
- $n^{n-2}$ possible trees (Cayley’s Theorem)

Nonetheless – Efficient optimal algorithm finds best tree!
Scoring a tree

\[
\arg \max_{\mathcal{G}} \log \hat{P}(\mathcal{D} | \hat{\theta}_\mathcal{G}, \mathcal{G}) = \arg \max_{\mathcal{G}} \sum_{i=1}^{n} \hat{I}(X_i, \text{Pa}_{X_i})
\]

Equivalent Trees (same score): \(I(A,B) + I(B,C)\)

Score provides indication of structure:

- I(A,B) + I(B,C)
- I(A,B) + I(A,C)
**Chow-Liu algorithm**

- For each pair of variables $X_i, X_j$
  - Compute empirical distribution: 
    $$\hat{P}(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{m}$$
  - Compute mutual information:
    $$\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \hat{P}(x_i, x_j) \log \frac{\hat{P}(x_i, x_j)}{\hat{P}(x_i)\hat{P}(x_j)}$$

- Define a graph
  - Nodes $X_1, \ldots, X_n$
  - Edge $(i,j)$ gets weight $\hat{I}(X_i, X_j)$

- Optimal tree BN
  - Compute maximum weight spanning tree (e.g. Prim’s, Kruskal’s algorithm $O(n\log n)$)
  - Directions in BN: pick any node as root, breadth-first-search defines directions
Chow-Liu algorithm example
Scoring general graphical models

• Graph that maximizes ML score -> complete graph!
• Information never hurts
  \[ H(A|B) \geq H(A|B,C) \]

• Adding a parent always increases ML score
  \[ I(A,B,C) \geq I(A,B) \]

• The more edges, the fewer independence assumptions, the higher the likelihood of the data, but will overfit...

• Why does ML for trees work?
  Restricted model space – tree graph
Regularizing

• Model selection
  – Use MDL (Minimum description length) score
  – BIC score (Bayesian Information criterion)

• Still NP –hard

**Theorem**: The problem of learning a BN structure with at most $d$ parents is **NP-hard for any (fixed) $d > 1$** (Note: tree $d=1$)

• Mostly heuristic (exploit score decomposition)
• Chow-Liu: provides best tree approximation to any distribution.
• Start with Chow-Liu tree. Add, delete, invert edges. Evaluate BIC score
What you should know

• Learning BNs
  – Maximum likelihood or MAP learns parameters
  – ML score
    • Decomposable score
    • Information theoretic interpretation (Mutual information)
  – Best tree (Chow-Liu)
  – Other BNs, usually local search with BIC score