Structure Learning \textit{finally}
(The Good), The Bad, The Ugly

Inference

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University
October 13\textsuperscript{th}, 2008

Decomposable score

- Log data likelihood
\[ \log \hat{P}(D \mid \theta, \mathcal{G}) = m \sum_i \hat{I}(X_i, \text{Pa}_{X_i}) - m \sum_i \hat{H}(X_i) \]

- Decomposable score:
  - Decomposes over families in BN (node and its parents)
  - Will lead to significant computational efficiency!!!
  - Score($G : D$) = $\sum_i \text{FamScore}(X_i \mid \text{Pa}_{X_i} : D)$

For MLE
\[ \text{FamScore}(x_i \mid \text{Pa}_{X_i} : D) = m \hat{I}(x_i; \text{Pa}_{X_i}) - m \hat{H}(X_i) \]
Structure learning for general graphs

- In a tree, a node only has one parent

**Theorem:**
- The problem of learning a BN structure with at most \(d\) parents is NP-hard for any (fixed) \(d \geq 2\)

- Most structure learning approaches use heuristics
  - Exploit score decomposition
  - (Quickly) Describe two heuristics that exploit decomposition in different ways

Understanding score decomposition

- \(\text{Score}(G : D)\)
- \(\text{Score}(C : D) + \text{Score}(D : D) + \text{Score}(I : D) + \ldots\)
- Only change: \(\text{SF}(I|C : D)\)

"Optimal" algorithm:
- each node picks set of parents that maximize score function
- but would get a BN with cycles
Fixed variable order 1

- Pick a variable order
  - e.g., $X_1, \ldots, X_n$
- $X_i$ can only pick parents in
  - Any subset
  - Acyclicity guaranteed!
- Total score = sum score of each node

Optimal BN with $d$ parents, consistent with order.

Fixed variable order 2

- Fix max number of parents to $k$
- For each $i$ in order
  - Pick $\text{Pa}_X \subseteq \{X_1, \ldots, X_i\}$
    - Exhaustively search through all possible subsets
    - $\text{Pa}_X$ is maximum $U \subseteq \{X_1, \ldots, X_i\}$ $\text{FamScore}(X|U : D)$
- Optimal BN for each order!!!
- Greedy search through space of orders:
  - E.g., try switching pairs of variables in order
  - If neighboring vars in order are switched, only need to recompute score for this pair
    - $O(n)$ speed up per iteration
Learn BN structure using local search

Starting from Chow-Liu tree

Local search, possible moves:
- Add edge
- Delete edge
- Invert edge

Only if acyclic!!!

Select using favorite score

BIC

Exploit score decomposition in local search

- Add edge and delete edge:
  - Only rescore one family!

- Reverse edge
  - Rescore only two families
Some experiments

Order search versus graph search

Order search advantages
- For fixed order, optimal BN – more “global” optimization
- Space of orders much smaller than space of graphs

Graph search advantages
- Not restricted to k parents
  - Especially if exploiting CPD structure, such as CSI
- Cheaper per iteration
- Finer moves within a graph
Bayesian model averaging

- So far, we have selected a single structure
- But, if you are really Bayesian, must average over structures
  - Similar to averaging over parameters
    \[ \log P(D | G) = \log \int_{\theta_G} P(D | G, \theta_G) P(\theta_G | G) d\theta_G \]
- Inference for structure averaging is very hard!!!
  - Clever tricks in reading

What you need to know about learning BN structures

- Decomposable scores
  - Data likelihood
  - Information theoretic interpretation
  - Bayesian
  - BIC approximation
- Priors
  - Structure and parameter assumptions
  - BDe if and only if score equivalence
- Best tree (Chow-Liu)
- Best TAN
- Nearly best k-treewidth (in O(N^{k+1}))
- Search techniques
  - Search through orders
  - Search through structures
- Bayesian model averaging
Inference in graphical models:
Typical queries 1

- Conditional probabilities
  - Distribution of some var(s). given evidence
    \[ P(A = t | H = t) \]
    \[ P(A = t | H = t) \times P(A = t, H = t) \]
    \[ P(A = t, H = t) = \sum_{s} \sum_{f} P(A = t, H = t, s, f) \]

Inference in graphical models:
Typical queries 2 – Maximization

- Most probable explanation (MPE)
  - Most likely assignment to all hidden vars given evidence
    \[ \max_{f, s, h, n} p(F = f, A = s, S = s, N = n | H = t) \]

- Maximum a posteriori (MAP)
  - Most likely assignment to some var(s) given evidence
    \[ \max_{a} P(A = a | H = t) \]
    \[ = \max_{a} \sum_{s} \sum_{f} P(A = a, s, f, n | H = t) \]
Are MPE and MAP Consistent?

- **Most probable explanation (MPE)**
  - Most likely assignment to all hidden vars given evidence
  
  \[ \text{MPE: } S = +, N = t \]

- **Maximum a posteriori (MAP)**
  - Most likely assignment to some var(s) given evidence
  
  \[ \max_s P(s = s) \Rightarrow \text{MAP}(s) = S = f \]

---

C++ Library

- Now available, join:

- The library implements the following functionality:
  - random variables, random processes, and linear algebra
  - factorized distributions, such as Gaussians, multinomial distributions, and mixtures
  - graph structures and basic graph algorithms
  - graphical models, including Bayesian networks, Markov networks, and junction trees
  - basic static and dynamic inference algorithms
  - parameter learning for Gaussian distributions, Chow Liu

- Fairly advanced C++ (not for everyone 😊)
Complexity of conditional probability queries 1

- How hard is it to compute $P(X|E=e)$?

Reduction – 3-SAT

$$\bigwedge \left( \overline{X_1} \lor X_2 \lor X_3 \right) \land \left( \overline{X_2} \lor X_3 \lor X_4 \right) \land \ldots$$

$E = \emptyset$

Complexity of conditional probability queries 2

- How hard is it to compute $P(X|E=e)$?

- At least NP-hard, but even harder!

$$2^n \text{ assignments, each has probability } \frac{1}{2^n}$$

$$P(Y = e) = \frac{\# \text{ sat assignments}}{2^n}$$
Inference is \#P-complete, hopeless?

- Exploit structure!
- Inference is hard in general, but easy for many (real-world relevant) BN structures

Complexity for other inference questions

- Probabilistic inference
  - general graphs: \#P-complete
  - poly-trees and low tree-width: polynomial

- Approximate probabilistic inference
  - Absolute error: \(|P(x) - \hat{P}(x)| \leq \varepsilon \) \(\in\) NP-hard for any \(\varepsilon \leq 0.5\)
  - Relative error: \(1 - \varepsilon \leq \frac{\hat{P}(x)}{P(x)} \leq 1 + \varepsilon \) \(\in\) NP-hard for any \(\varepsilon > 0\)

- Most probable explanation (MPE)
  - general graphs: NP-complete
  - poly-trees and low tree-width: polynomial

- Maximum a posteriori (MAP)
  - general graphs: NP\(^{\#}\)-complete
  - poly-trees and low tree-width: NP-hard
Inference in BNs hopeless?

- In general, yes!
  - Even approximate!

- In practice
  - Exploit structure
  - Many effective approximation algorithms (some with guarantees)

- For now, we’ll talk about exact inference
  - Approximate inference later this semester

General probabilistic inference

- Query: \( P(X \mid e) \)

- Using def. of cond. prob.:
  \[
  P(X \mid e) = \frac{P(X, e)}{P(e)}
  \]

- Normalization:
  \[
  P(X \mid e) \propto P(X, e)
  \]

\[
P(A=t \mid H=t) = \frac{\Pr(A=t, H=t)}{\Pr(H=t)} = \frac{0.2}{0.1} = 2\]

\[
\Pr(A=t \mid H=t) = \frac{2}{3}
\]
Marginalization

\[ p(\text{F}=t, \text{N}=t) = \sum_j p(\text{F}=t, \text{S}=j, \text{N}=t) \]
\[ = p(\text{F}=t, \text{S}=t, \text{N}=t) + p(\text{F}=t, \text{S}=f, \text{N}=t) \]

Probabilistic inference example

\[ p(A | \text{N}=t) \propto p(A, \text{N}=t) \]
\[ = \sum_f \sum_j p(A, \text{F}=f, \text{S}=j, \text{N}=t) \]
\[ = \sum_f \sum_j p(A) p(\text{F}=f, \text{A}) p(\text{S}=j | \text{F}, \text{A}) p(\text{N}=t | j) \sum_h p(\text{H}=h) \]
\[ = \sum_f \sum_j p(A) p(\text{F}=f) p(\text{S}=j | \text{F}) p(\text{N}=t | j) \sum_h p(\text{H}=h) \]

Inference seems exponential in number of variables!
Fast probabilistic inference example – Variable elimination

(Potential for) Exponential reduction in computation!

Understanding variable elimination – Exploiting distributivity
Understanding variable elimination – Order can make a HUGE difference

Intermediate results are probability distributions
Understanding variable elimination – Another example

Pruning irrelevant variables

Prune all non-ancestors of query variables
More generally: Prune all nodes not on active trail between evidence and query vars