Decomposable score

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

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

**For MLE:**
\[ \text{Fam Score}(X_i | \text{Pa}_{X_i}; \mathcal{D}) = m \hat{I}(X_i; \text{Pa}_{X_i}) - m \hat{H}(X_i) \]
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

- Difficulty, Coherence, Intelligence
- Grade, SAT
- Letter, Happy, Job
Fixed variable order 1

- Pick a variable order
  - e.g., $X_1, \ldots, X_n$
- $X_i$ can only pick parents in
  - $\{X_1, \ldots, X_{i-1}\}$
  - Any subset
  - Acyclicity guaranteed!
- Total score = sum score of each node

Fixed variable order 2

- Fix max number of parents to k
- For each $i$ in order
  - Pick $P_{a_X} \subseteq \{X_1, \ldots, X_{i-1}\}$
    - Exhaustively search through all possible subsets
    - $P_{a_X}$ is maximum $U \subseteq \{X_1, \ldots, X_{i-1}\}$ FamScore($X_i|U : D$)
- 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

Select using favorite score

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
    \[ \log P(D \mid G) = \log \int_{\theta_G} P(D \mid G, \theta_G) P(\theta_G \mid G) \, d\theta_G \]
- 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

Inference in graphical models:
Typical queries 2 – Maximization

- Most probable explanation (MPE)
  - Most likely assignment to all hidden vars given evidence

- Maximum a posteriori (MAP)
  - Most likely assignment to some var(s) given evidence
Are MPE and MAP Consistent?

- **Most probable explanation (MPE)**
  - Most likely assignment to all hidden vars given evidence

- **Maximum a posteriori (MAP)**
  - Most likely assignment to some var(s) given evidence

---

C++ Library

- Now available, join:
  - [http://groups.google.com/group/10708-f08-code/](http://groups.google.com/group/10708-f08-code/)

- 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

\[(\overline{X}_1 \lor X_2 \lor X_3) \land (\overline{X}_2 \lor X_3 \lor X_4) \land \ldots \]

Complexity of conditional probability queries 2

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

- At least NP-hard, but even harder!
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:
  - poly-trees and low tree-width:

- Approximate probabilistic inference
  - Absolute error:
  - Relative error:

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

- Maximum a posteriori (MAP)
  - general graphs:
  - poly-trees and low tree-width:
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:
  \[
  P(X \mid e) \propto P(X, e)
  \]
Marginalization

Probabilistic inference example

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

Flu \rightarrow Allergy
\rightarrow Sinus
\rightarrow Headache \rightarrow Nose=t

Intermediate results

Understanding variable elimination –
Intermediate results

Flu \rightarrow Allergy
\rightarrow Sinus
\rightarrow Headache \rightarrow Nose=t

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