My Plan

5-9 April: Bayesian Classifiers, Local Structure, Trip
12 April: Learning Time Series (DBNs)
14 April: Gaussian/Discrete networks
16 April: Normal Wishart
19 April: Summary
21 + 23 April: Student presentations

Last Time

Bayesian prediction with incomplete data.
MAP approximation
MCMC

Scoring Metrics for complete data

Likelihood:  \[ I(D : G) = \log L[G : D] = M \sum_i (I(X_i : Pa_i) - H(X_i)) \]

MDL:  \[ MDL(D : G) = I(G : D) - \frac{d}{2} \log M - DL(G) = I(G : D) - \frac{d}{2} \log M \]

BDe:  \[ \log P[D|G] = \sum_{\mathcal{X}} \log \left( \prod_{a_i} \frac{\Gamma(\alpha(p_a^c))}{\Gamma(\alpha(p_a^c) + N(p_a^c))} \prod_{a_i} \frac{\Gamma(\alpha(x_i, pa_a^c))}{\Gamma(\alpha(x_i, pa_a^c) + N(x_i, pa_a^c))} \right) \]
Basic Bayes Net Model Selection

The extremely basic idea:
Given two nets, prefer the one with the higher score.

Score( ) > Score( )

We know how to score multinomials...
How do we select the proper model?

Structure Optimization

(Candidate Generation)

Select graph that maximizes score

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

P(A)
P(B|A)
P(C)
P(D|A,C)
P(D|A)
P(E|D)
Trees

Tree:
At most one parent per variable.

O(N) Solution

Later: In classification, tree-augmented classifiers provides an intermediate step between naïve bayes classifiers and general network structures.

Define $p(i)$ to be the parent of $X_i$. $p(i) = 0$ if $X_i$ has no parent.

Score

$\text{Score}(G : D) = \sum_i \text{Score}(X_i : Pa_i)$

$= \sum_{i, p(i) \neq 0} \text{Score}(X_i : X_{p(i)}) + \sum_{i, p(i) = 0} \text{Score}(X_i)$

$= \sum_{i, p(i) \neq 0} (\text{Score}(X_i : X_{p(i)}) - \text{Score}(X_i)) + \sum_i \text{Score}(X_i)$
Trees

Algorithm

Construct graph with vertices 1, ..., N
Set $w(i \rightarrow j)$ to $Score(X_j | X_i) - Score(X_i)$
Find tree (or forest) with maximal weight
variant on Kruskal’s algorithm

When the score is likelihood, then this is known as a Chow & Liu method.

General DAGs

Finding DAG of maximum score is much harder.

THM: Finding the maximal scoring DAG with at most k parents for each variable is NP-hard for k>1.

Implication: Greedy (O(n)) approaches are no longer guaranteed to work.
Search over DAGs

Define operations to transform an input DAG

- Add an arc
- Remove an arc
- Reverse an arc

Search

Difficult optimization problem
- Local maxima, plateaus

Approaches
- Greedy hill climbing
- Greedy with Tabu search
- Simulated annealing
Search over I-Equivalence Classes

Idea:
Search space of I-equivalence classes
Each I-equivalence class is represented by a PDAG: graph skeleton + V-structures
Add arcs to 'complete' PDAG in order to score.

Benefits
Fewer local maxima and plateaus
Fewer PDAGs than DAGs

Summary: Search over structure

Pieces of the solution:
Structure generation
DAGs
PDAGs
Structure scoring
Easy closed form solution (need conjugate prior for BDe).
Search algorithm (not discussed)
Structure generation algorithm defines a search space.
Model Averaging

Recall, Bayesian inference started with

\[ P\{\mathbf{y}[M+1]|D\} = \sum_G P\{\mathbf{y}[M+1]|D,G\} P\{G|D\} \]

True Bayes solution:
average over all possible graphs.

We focussed on finding the best scoring model:
implicit assumption: Best model dominates weighted sum.

Problems:
Over commitment to a single structure?

---

Model Averaging

Full Averaging
Sum over all structures (intractable)

Approximate Averaging
Find K highest scoring structures
Approximate overall prediction as a weighted average of the individual predictions
Relative weight of each structure is determined by the Bayes Factor

\[
\frac{P\{G_1|D\}}{P\{G_2|D\}} = \frac{P\{G_1\}P\{D|G_1\}P\{D\}}{P\{G_2\}P\{D|G_2\}P\{D\}} = \frac{P\{G_1\}P\{D|G_1\}}{P\{G_2\}P\{D|G_2\}}
\]

In limit:
Equiprobable dist'n over structures with independence relations that are “closest” to the underlying distribution.
Structure Optimization: Conclusion

Multiple scores: Likelihood, MLE, BDe
  Equivalent for large data sets.
Select structure that optimizes score
  Tree: Simple O(n) search algorithm.
  General BNs: NP-Hard
    Proposal: local search based on modifying candidates
    Lots of plateaus/local maxima.
Bayesian model averaging