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

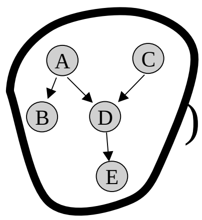
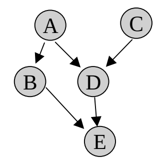
Scoring Metrics for complete data

- Likelihood: $I(D:G) = \log L[G:D] = M \sum_i \left(I(X_i : Pa_i) - H(X_i) \right)$
- MDL: $MDL(D:G) = I(G:D) - \frac{d}{2}\log M - DL(G) \approx I(G:D) - \frac{d}{2}\log M$
- BDe: $\log P\{D \mid G\} = \sum_{i=1}^{N} \log\left(\prod_{pa_i^G} \frac{\Gamma(\alpha(pa_i^G))}{\Gamma(\alpha(pa_i^G) + N(pa_i^G))} \prod_{x_i} \frac{\Gamma(\alpha(x_i, pa_i^G) + N(x_i, pa_i^G))}{\Gamma(\alpha(x_i, pa_i^G))} \right)$

Basic Bayes Net Model Selection

The extremely basic idea:

Given two nets, prefer the one with the higher score.

Score() > Score()

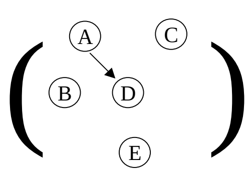
We know how to score multinomials…

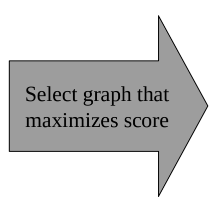
How do we select the proper model?

Structure Optimization (Complete Data)

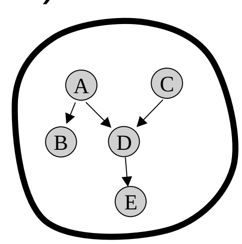
A	B	C	D	E
1	2	1	0	1
1	1	0	1	1
0	1	1	1	1
1	1	1	1	2

+





Candidate Generation



+

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.

Trees

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

Score

$$\begin{aligned} Score(G:D) &= \sum_i Score(X_i : Pa_i) \\ &= \sum_{i,p(i)>0} Score(X_i : X_{p(i)}) + \sum_{i,p(i)=0} Score(X_i) \\ &= \sum_{i,p(i)>0} \left(Score(X_i : X_{p(i)}) - Score(X_i)\right) + \sum_i Score(X_i) \end{aligned}$$

Trees

Algorithm

Construct graph with vertices 1, …, N

Set $w(i \rightarrow j)$ to $Score(X_j \mid X_i) - Score(X_i)$

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.

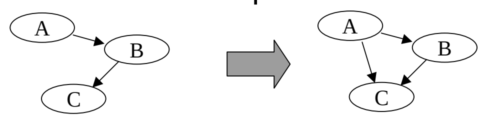
THM: Finding the maximal scoring DAG with at most k parents for each variable is NP-hard for k>1.

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

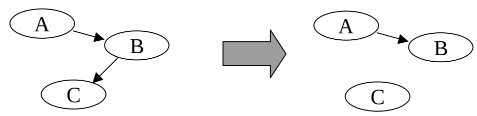
Search over DAGs

Define operations to transform an input DAG

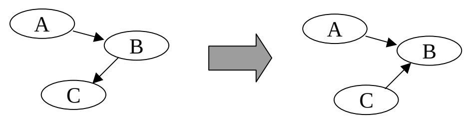
- Add an arc



- Remove an arc



- Reverse an arc



Search

Difficult optimization problem

- Local maxima, plateaus

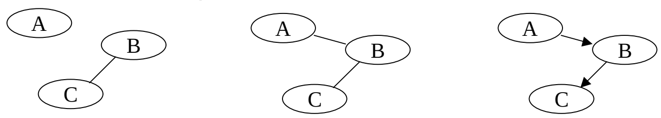
Approaches

- 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.



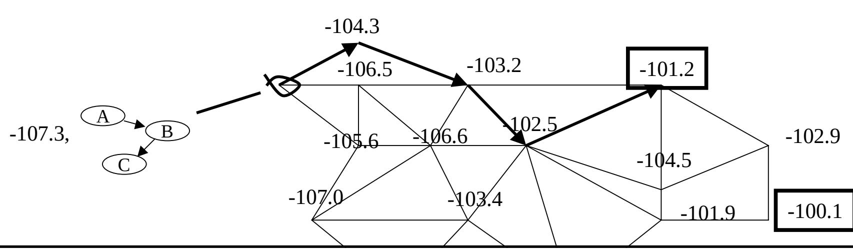
Benefits

- 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

$$P\{x[M+1] \mid D\} = \sum_G P\{x[M+1] \mid D, G\} P\{G \mid D\}$$

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

$$\frac{P\{G_1 \mid D\}}{P\{G_2 \mid D\}} = \frac{P\{G_1\}P\{D \mid G_1\}P\{D\}}{P\{G_2\}P\{D \mid G_2\}P\{D\}} = \frac{P\{G_1\}P\{D \mid G_1\}}{P\{G_2\}P\{D \mid G_2\}}$$

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