Learning With Bayesian Networks

Markus Kalisch
ETH Zürich
Inference in BNs - Review

P(Burglary|JohnCalls=TRUE, MaryCalls=TRUE)

- **Exact Inference:**
  - \( P(b|j,m) = c \sum_e \sum_a P(b)P(e)P(a|b,e)P(j|a)P(m|a) \)
  - Deal with sums in a clever way: Variable elimination, message passing
  - Singly connected: linear in space/time
  - Multiply connected: exponential in space/time (worst case)

- **Approximate Inference:**
  - Direct sampling
  - Likelihood weighting
  - MCMC methods

Markus Kalisch, ETH Zürich
Learning BNs - Overview

• Brief summary of Heckerman Tutorial
• Recent provably correct Search Methods:
  – Greedy Equivalence Search (GES)
  – PC-algorithm
• Discussion
Abstract and Introduction

Graphical Modeling offers:

• Easy handling of missing data
• Easy modeling of causal relationships
• Easy combination of prior information and data
• Easy to avoid overfitting

Markus Kalisch, ETH Zürich
Bayesian Approach

- Degree of belief
- Rules of probability are a good tool to deal with beliefs
- Probability assessment: Precision & Accuracy
- Running Example: Multinomial Sampling with Dirichlet Prior

Markus Kalisch, ETH Zürich
Bayesian Networks (BN)

Define a BN by
- a network structure
- local probability distributions

To *learn* a BN, we have to
- choose the variables of the model
- choose the structure of the model
- assess local probability distributions
Inference

We have seen up to now:

• Book by Russell / Norvig:
  – exact inference
  – variable elimination
  – approximate methods

• Talk by Prof. Loeliger:
  – factor graphs / belief propagation / message passing

• Probabilistic inference in BN is NP-hard:
  Approximations or special-case-solutions are needed
Learning Parameters (structure given)

- Prof. Loeliger: Trainable parameters can be added to the factor graph and therefore be inferred
- Complete Data
  - reduce to one-variable case
- Incomplete Data (missing at random)
  - formula for posterior grows exponential in number of incomplete cases
  - Gibbs-Sampling
  - Gaussian Approximation; get MAP by gradient based optimization or EM-algorithm
Learning Parameters AND structure

- Can learn structure only up to likelihood equivalence
- Averaging over all structures is infeasible: Space of DAGs and of equivalence classes grows super-exponentially in the number of nodes.
Model Selection

- Don't average over all structures, but select a good one (Model Selection)
- A good scoring criterion is the log posterior probability:
  \[ \log(P(D,S)) = \log(P(S)) + \log(P(D|S)) \]
  Priors: Dirichlet for Parameters / Uniform for structure
- Complete cases: Compute this exactly
- Incomplete cases: Gaussian Approximation and further simplification lead to BIC
  \[ \log(P(D|S)) = \log(P(D|ML-Par,S)) - d/2 \times \log(N) \]
  This is usually used in practice.
Search Methods

• Learning BNs on discrete nodes (3 or more parents) is NP-hard (Heckerman 2004)
• There are provably (asymptotically) correct search methods:
  – Search and Score methods: Greedy Equivalence Search (GES; Chickering 2002)
  – Constrained based methods: PC-algorithm (Spirtes et al. 2000)
GES – The Idea

- Restrict the search space to equivalence classes
- Score: BIC
  “separable search criterion” => fast
- Greedy Search for “best” equivalence class
- In theory (asymptotic): Correct equivalence class is found
GES – The Algorithm

GES is a two-stage greedy algorithm

• Initialize with equivalence class $E$ containing the empty DAG

• Stage 1: Repeatedly replace $E$ with the member of $E^+(E)$ that has the highest score, until no such replacement increases the score

• Stage 2: Repeatedly replace $E$ with the member of $E^-(E)$ that has the highest score, until no such replacement increases the score

Markus Kalisch, ETH Zürich
PC – The idea

• Start: Complete, undirected graph
• Recursive conditional independence tests for deleting edges
• Afterwards: Add arrowheads
• In theory (asymptotic): Correct equivalence class is found

Markus Kalisch, ETH Zürich
PC – The Algorithm

Form complete, undirected graph $G$

$l = -1$

repeat

\[ l = l + 1 \]

repeat

select ordered pair of adjacent nodes $A, B$ in $G$

select neighborhood $N$ of $A$ with size $l$ (if possible)

delete edge $A, B$ in $G$ if $A, B$ are cond. indep. given $N$

until all ordered pairs have been tested

until all neighborhoods are of size smaller than $l$

Add arrowheads by applying a couple of simple rules
Example

Conditional Independencies:

- l=0: none
- l=1: \[ A \perp C \mid B \]
  \[ A \perp D \mid B \]
  \[ C \perp D \mid B \]

\[ \text{PC-algorithm: correct skeleton} \]
Sample Version of PC-algorithm

- Real World: Cond. Indep. Relations not known
- Instead: Use statistical test for Conditional independence
- Theory: Using statistical test instead of true conditional independence relations is often ok
Comparing PC and GES

For $p = 10$, $n = 50$, $E(N) = 0.9$, 50 replicates:

<table>
<thead>
<tr>
<th>Method</th>
<th>ave[TPR]</th>
<th>ave[FPR]</th>
<th>ave[TDR]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>0.57 (0.06)</td>
<td>0.02 (0.01)</td>
<td>0.91 (0.05)</td>
</tr>
<tr>
<td>GES</td>
<td>0.85 (0.05)</td>
<td>0.13 (0.04)</td>
<td>0.71 (0.07)</td>
</tr>
</tbody>
</table>

The PC-algorithm

- finds less edges
- finds true edges with higher reliability
- is fast for sparse graphs  
  (e.g. $p=100,n=1000,E[N]=3$: $T = 13$ sec)
Learning Causal Relationships

• Causal Markov Condition:
  Let $C$ be a causal graph for $X$
  then
  $C$ is also a Bayesian-network structure for the pdf of $X$

• Use this to infer causal relationships
Conclusion

• Using a BN: Inference (NP-Hard)
  – exact inference, variable elimination, message passing (factor graphs)
  – approximate methods

• Learn BN:
  – Parameters:
    Exact, Factor Graphs
    Monte Carlo, Gauss
  – Structure: GES, PC-algorithm; NP-Hard