Learning Bayesian Networks: Shortest Path Perspective

Walter Perez Urcia

Universidade de São Paulo
Instituto de Matemática e Estadística
Departamento de Ciências da Computação

Novembro 2015

Outline

Bayesian Networks

A Bayesian Network consists of

- A DAG G over a set of variables $X_1, \ldots, X_n$
- Markov Property: Given its parents, every variable is conditionally independent from its non-descendant non-parents
- Probability constraints: $\mathbb{P}(X_i = k \mid Pa(X_i) = j) = \theta_{ijk}$

Joint Probability Distribution

There is a unique probability function consistent with a BN:

$$\mathbb{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathbb{P}(X_i \mid Pa(X_i)) = \prod_{i=1}^{n} \theta_{ijk}$$

Car Evaluation Dataset

- Buying price (B): v-high, high, med, low
- Maintain cost (M): v-high, high, med, low
- Doors (D): two, three, four, more
- Persons (P): two, four, more
- Luggage boot (L): small, medium, big
- Safety (S): low, medium, high

Represent:

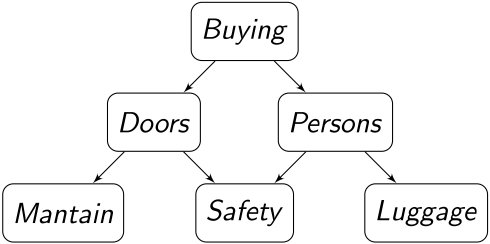
- Half of cars that have four doors have a medium luggage boot
- 15% of cars are low safety, 77% medium safety and 8% high safety

Using a probabilistic model of knowledge to represent all possible relations we have:

$$\mathbb{P}(B, M, D, P, L, S)$$

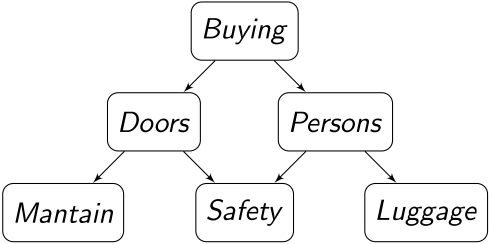
This requires $4 \times 4 \times 4 \times 3 \times 3 \times 3 = 1728$ probabilities hard to estimate, but we can drastically reduce this number by assuming (conditional) independences

For example:



- Doors and Persons are independent given Buying: $\mathbb{P}(D, P \mid B) = \mathbb{P}(D \mid B)\mathbb{P}(P \mid B)$
- Mantain and Safety are independent given Doors: $\mathbb{P}(M, S \mid D) = \mathbb{P}(M \mid D)\mathbb{P}(S \mid D)$

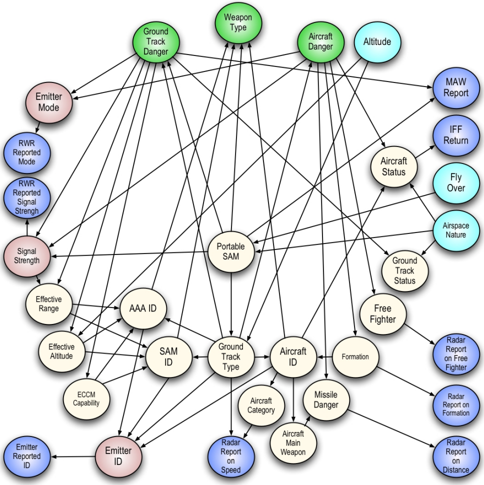
  $\vdots$



$\mathbb{P}(B, M, D, P, L, S) = \mathbb{P}(B)\mathbb{P}(D \mid B)\mathbb{P}(P \mid B)\mathbb{P}(M \mid D)\mathbb{P}(S \mid D, P)\mathbb{P}(L \mid P)$

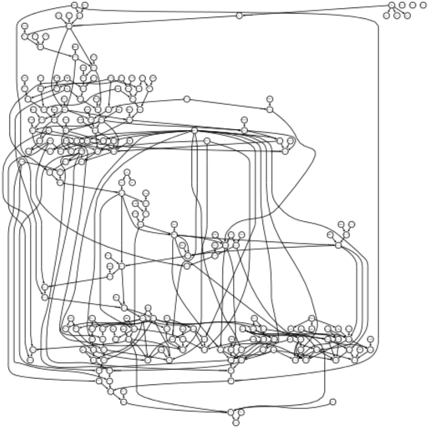
This requires
$4 + (4 \times 4) + (3 \times 4) + (4 \times 4) + (3 \times 4 \times 3) + (3 \times 3) = 93$
probabilities instead of 1728

Consider each variable has k values:
We requires k^{33} probabilities without independences.



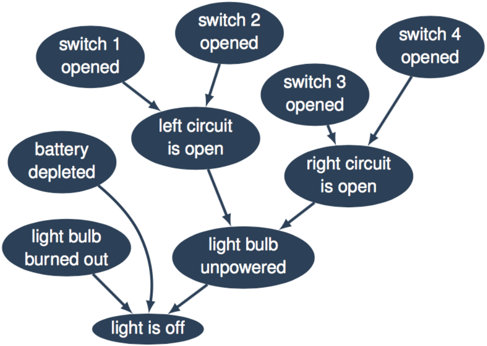
- Elicitation from expert knowledge
- Direct translation
- Learning from data

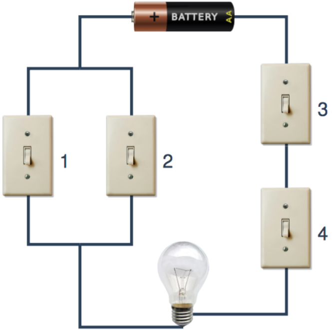
Elicitation



ANDES: Intelligent Tutoring System to teach Newtonian Physics

Direct Translation



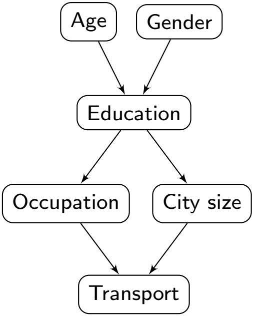


Learning Bayesian networks from data

Learning BN from data

Given a data set infers a Bayesian network structure

Age	Gender	City Size	Education	Occupation	Transport
adult	F	big	high	employee	car
adult	M	small	uni	employee	car
adult	F	big	uni	employee	train
young	M	big	high	self-emp	car
adult	M	big	high	employee	car
⋮	⋮	⋮	⋮	⋮	⋮



Constraint-based approaches

Perform multiple conditional independence hypothesis testing in order to build a DAG

Score-based approaches

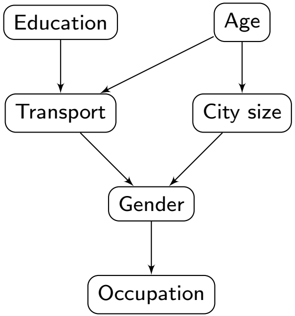
Associate every DAG with a polynomial-time computable score value and search for structure with high score values

Learning as optimization

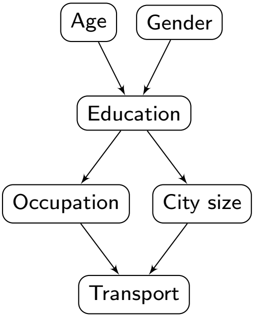
Given dataset D, select G that maximizes decomposable score function:

$$sc(G, D) = LL(D \mid G) + \psi(N) \times |G|$$

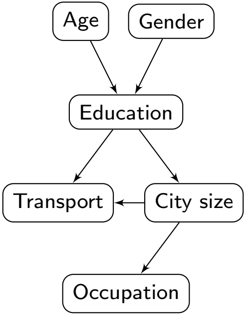
$$sc(G) = \sum_i sc(X_i, Pa(X_i))$$



$$sc(G) = -9508.34$$



$$sc(G) = -6917.23$$



$$sc(G) = -8891.52$$

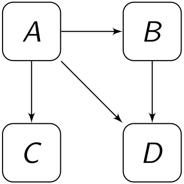
Greedy Search is a popular approach to find an approximate solution

```
GreedySearch( Dataset D , Solution G_0 ) : return a BN G
  G = G_0
  For a number of iterations K
    best_neighbor = find_best_neighbor(G)
    if score(best_neighbor) > score(G) then
      G = best_neighbor
  Return G
```

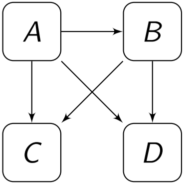
Greedy Search approaches for learning Bayesian networks can be classified as:

- Equivalence-based
- Structure-based
- Order-based

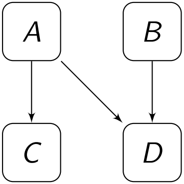
Consider incumbent solution is



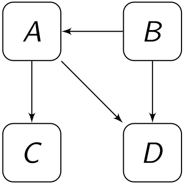
Neighboorhood:



Add an edge



Remove an edge



Revert an edge's direction

Based on the observation that the problem of learning a Bayesian network can be written as

$$G^* = \arg\max_{<} \sum_{i=1}^{n} \max_{P \subseteq \{X_j < X_i\}} sc(X_i, P)$$

An optimal DAG can be found by maximizing the local scores independently given an order of the variables

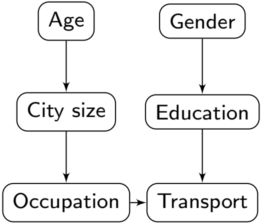
Order-based Greedy Search

```
OrderBasedGreedySearch( Dataset D , Order L_0 ) :
return a BN
  L = L_0
  For a number of iterations K
    current_sol = L
    For each i = 1 to n-1 do
      L_i = swap(L, i, i+1)
      if score(L_i) > score(current_sol)
        current_sol = L_i
    if score(current_sol) > score(L) then
      L = current_sol
  Return network(L)
```

where $swap(L, i, i+1)$ swaps the values $L[i]$ and $L[i+1]$

Consider incumbent solution is

$$[A, G, C, E, O, T]$$



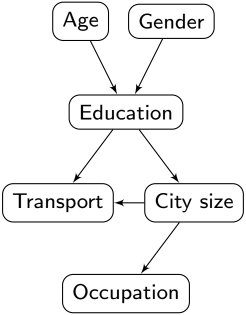
$$sc = -13192.33$$

Neighborhood:

- $[G, A, C, E, O, T]$
  $sc = -10593.82$
- $[A, C, G, E, O, T]$
  $sc = -10891.48$
- $[A, G, E, C, O, T]$
  $sc = -8991.52$
- $[A, G, C, O, E, T]$
  $sc = -9917.23$
- $[A, G, C, E, T, O]$
  $sc = -9158.42$

Now, incumbent solution is

$$[A, G, E, C, O, T]$$

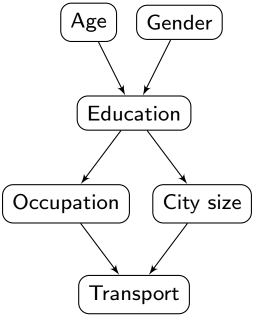


$$sc = -8991.52$$

- $[G, A, E, C, O, T]$
  $sc = -7593.82$
- $[A, E, G, C, O, T]$
  $sc = -8891.48$
- $[A, G, C, E, O, T]$
  $sc = -13192.33$
- $[A, G, E, O, C, T]$
  $sc = -6917.23$
- $[A, G, E, C, T, O]$
  $sc = -6999.99$

Now, incumbent solution is

$$[A, G, E, O, C, T]$$



$$sc = -6917.23$$

- $[G, A, E, O, C, T]$
  $sc = -8593.82$
- $[A, E, G, O, C, T]$
  $sc = -7289.48$
- $[A, G, O, E, C, T]$
  $sc = -9145.13$
- $[A, G, E, C, O, T]$
  $sc = -8991.52$
- $[A, G, E, O, T, C]$
  $sc = -6991.08$

Common approach for initial solutions

- Random generation of a variable order
- Too many possible orders: $n!$
- Slow convergence
- Poor local maxima

- The proposed heuristics lead to better solutions on average, and increase the convergence of the search with only a small overhead
- Larger diferences for datasets with more variables are expected
- Our proposed techniques could return directed acyclic graphs instead of node orderings to be used for Structure- and Equivalence-based search approaches
- Employ the proposed heuristics in branch-and-bound solvers for finding optimal solutions

Thanks!

Learning Bayesian Networks: Shortest Path Perspective

Walter Perez Urcia

Universidade de São Paulo
Instituto de Matemática e Estadística
Departamento de Ciências da Computação

Novembro 2015