Learning the topology of a Bayesian Network from a database of cases using the K2 algorithm

Maria Camila Paris Diaz, Gabriele Poccianti, Laura Schulze

July 18, 2024

Outline

1. The K2 algorithm
    - Overview
    - Scoring function
    - Node ordering
2. Test datasets
3. Analysis of the Heart Disease Dataset

The K2 algorithm

Bayesian Network Learning

Bayesian Network learning: [1]

- **Structure** learning vs. **Parameter** learning
- **constraint**-based, **score**-based or **hybrid** methods
- **heuristic** vs. **exact**

K2 algorithm [2]

- introduced 1992 by Gregory F. Cooper and Edward Herskovits
- score-based, heuristic method
- assumptions:
  - known node ordering
  - discrete variables, no missing values
  - equal priors for all possible networks

[1] M. Scutari and J.-B. Denis. ***Bayesian Networks: With Examples in R (2nd ed.)*** Chapman and Hall/CRC, 2021. DOI: 10.1201/9780429347436

[2] Gregory F. Cooper and Edward Herskovits. **"A Bayesian method for the induction of probabilistic networks from data".** In: *Machine Learning* 9.4 (Oct. 1992), pp. 309–347. ISSN: 1573-0565

K2 Algorithm

Algorithm 1 K2 Algorithm.

procedure K2;
{**Input**: A set of nodes n, an ordering of the nodes, an upper bound u on the number of parents a node may have, and a database D containing m cases.}
{**Output**: For each node, a printout of the parents of the node.}
for $i := 1$ **to** n **do**
  $\pi_i := \emptyset$;
  $P_{\text{old}} := g(i, \pi_i)$; {scoring function}
  OKToProceed := **true**;
  while OKToProceed and $|\pi_i| < u$ **do**
    Let z be the node in Pred(x_i) - π_i that maximizes $g(i, \pi_i \cup \{z\})$;
    $P_{\text{new}} := (i, \pi_i \cup \{z\})$;
    if $P_{\text{new}} > P_{\text{old}}$ **then**
      $P_{\text{old}} := P_{\text{new}}$;
      $\pi_i := \pi_i \cup z$
    else
      OKToProceed := **false**;
    end if
  end while
  write('Node:', x_i, 'Parents of this node', π_i)
end for
end K2;

Scoring function $g(i, \pi_i)$

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!$$

- π_i: set of parents of node x_i
- q_i: # of all possible instantiations of the parents of x_i in database D
- r_i: # of all possible values of the attribute x_i
- α_{ijk}: # of cases in D in which the attribute x_i is instantiated with its kth value, and the parents of x_i in π_i are instantiated with the jth possible instantiation
- $N_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$. # of instances in the database in which the parents of x_i in π_i are instantiated with the jth possible instantiation

Scoring function $g(i, \pi_i)$

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!$$

- π_i: set of parents of node x_i
- q_i: # of all possible instantiations of the parents of x_i in database D
- r_i: # of all possible values of the attribute x_i
- α_{ijk}: # of cases in D in which the attribute x_i is instantiated with its kth value, and the parents of x_i in π_i are instantiated with the jth possible instantiation
- $N_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$. # of instances in the database in which the parents of x_i in π_i are instantiated with the jth possible instantiation

Scoring function $g(i, \pi_i)$

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!$$

- π_i: set of parents of node x_i
- q_i: # of all possible instantiations of the parents of x_i in database D
- r_i: # of all possible values of the attribute x_i
- α_{ijk}: # of cases in D in which the attribute x_i is instantiated with its kth value, and the parents of x_i in π_i are instantiated with the jth possible instantiation
- $N_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$. # of instances in the database in which the parents of x_i in π_i are instantiated with the jth possible instantiation

Scoring function $g(i, \pi_i)$

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!$$

- π_i: set of parents of node x_i
- q_i: # of all possible instantiations of the parents of x_i in database D
- r_i: # of all possible values of the attribute x_i
- α_{ijk}: # of cases in D in which the attribute x_i is instantiated with its kth value, and the parents of x_i in π_i are instantiated with the jth possible instantiation
- $N_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$. # of instances in the database in which the parents of x_i in π_i are instantiated with the jth possible instantiation

Scoring function $g(i, \pi_i)$

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!$$

- π_i: set of parents of node x_i
- q_i: # of all possible instantiations of the parents of x_i in database D
- r_i: # of all possible values of the attribute x_i
- α_{ijk}: # of cases in D in which the attribute x_i is instantiated with its kth value, and the parents of x_i in π_i are instantiated with the jth possible instantiation
- $N_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$. # of instances in the database in which the parents of x_i in π_i are instantiated with the jth possible instantiation

Logarithmic scoring function $\log(g(i, \pi_i))$

- $g(i, \pi_i)$ may require computation of large factorials
- idea: use **logarithmic version** instead

$$log(g(i, \pi_i)) = \log\left(\prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!\right) \tag{1}$$

(2)

Logarithmic scoring function $\log(g(i, \pi_i))$

- $g(i, \pi_i)$ may require computation of large factorials
- idea: use **logarithmic version** instead

$$log(g(i, \pi_i)) = \log\left(\prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!\right) \tag{1}$$

$$log(g(i, \pi_i)) = \sum_{j=1}^{q_i} \left[\sum_{l=1}^{r_i - 1} \log l - \sum_{m=1}^{N_{ij} + r_i - 1} \log m + \sum_{k=1}^{r_i} \sum_{n=1}^{\alpha_{ijk}} \log n\right] \tag{2}$$

Node ordering

Example: Assume node ordering **A, B, C, D, E, F, G**

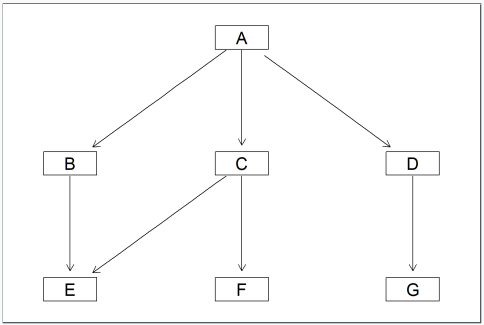


Fig. 1: Allowed topology.

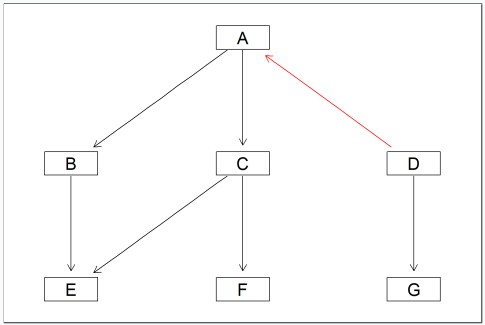


Fig. 2: Forbidden topology.

Node ordering

- Node ordering in K2 limits number of possible topologies
- given an ordering of n nodes, number of possible network structures is $2^{\binom{n}{2}} = 2^{\frac{n(n-1)}{2}}$
- "wrong" order $\Rightarrow$ algorithm learns wrong structure
- **How to find a suitable node ordering?**

Brute Force

- try every possible node ordering
- there are $n!$ unique node orderings for n variables
- not feasible to test all for medium or large networks!

Random search

- try fixed number of randomly sampled node orderings
- choose network with best overall K2 score

Test datasets

RUIZ, ASIA and CHILD dataset

- **RUIZ Dataset**
  - 3 nodes, 2 arcs
  - dataset of 10 records
- **ASIA Dataset**
  - 8 nodes, 8 arcs
  - dataset of 5000 records
- **CHILD Dataset**
  - 20 nodes, 25 arcs
  - dataset of 5000 records

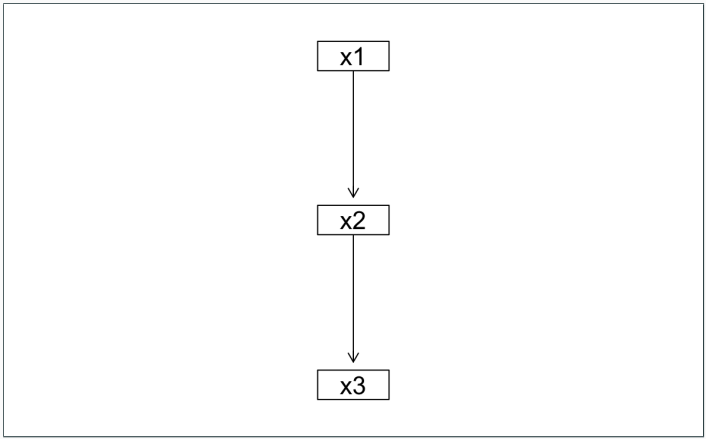


Fig. 5: RUIZ Network.

RUIZ, ASIA and CHILD dataset

- **RUIZ Dataset**
  - 3 nodes, 2 arcs
  - dataset of 10 records
- **ASIA Dataset**
  - 8 nodes, 8 arcs
  - dataset of 5000 records
- **CHILD Dataset**
  - 20 nodes, 25 arcs
  - dataset of 5000 records

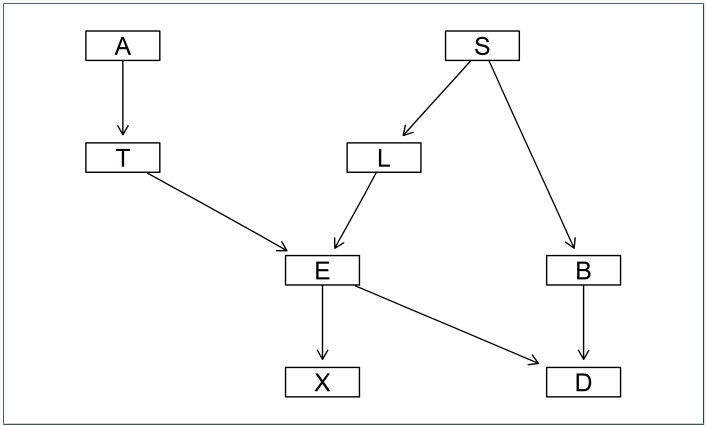


Fig. 6: ASIA Network.

RUIZ, ASIA and CHILD dataset

- **RUIZ Dataset**
  - 3 nodes, 2 arcs
  - dataset of 10 records
- **ASIA Dataset**
  - 8 nodes, 8 arcs
  - dataset of 5000 records
- **CHILD Dataset**
  - 20 nodes, 25 arcs
  - dataset of 5000 records

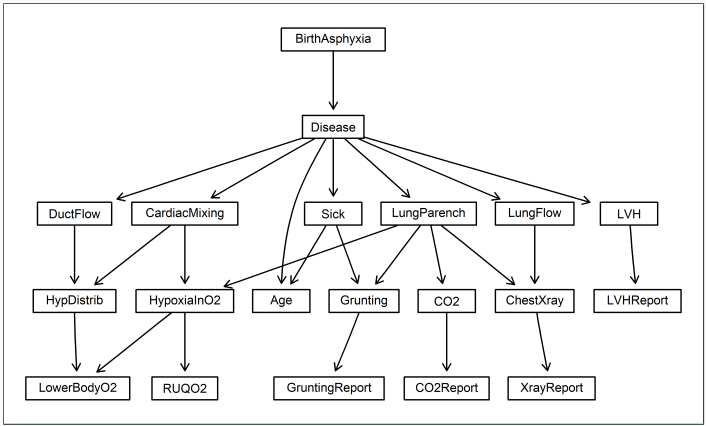


Fig. 7: CHILD Network.

RUIZ, ASIA and CHILD dataset

- **RUIZ Dataset**
  - 3 nodes, 2 arcs
  - dataset of 10 records
- **ASIA Dataset**
  - 8 nodes, 8 arcs
  - dataset of 5000 records
- **CHILD Dataset**
  - 20 nodes, 25 arcs
  - dataset of 5000 records

Strategy

For each test dataset:

1. Test self-implemented K2 algorithm with fixed number of random node orderings (up to 300)
2. Run Tabu search with K2 score from `bnlearn` library
3. Compare to each other and to ground truth

Tabu search

Tabu search

- Heuristic greedy algorithm
- can implement blacklist/ whitelist
- Explore search space starting from a network structure
- for each iteration:
  1. add, delete, reverse one arc (without generating a loop)
  2. Search for local optimum solution $\mathbf{G}^*$ that isn't in the **tabu table**
  3. update tabu table

Results: RUIZ dataset

Ground truth node order: [x1] → [x2] → [x3]

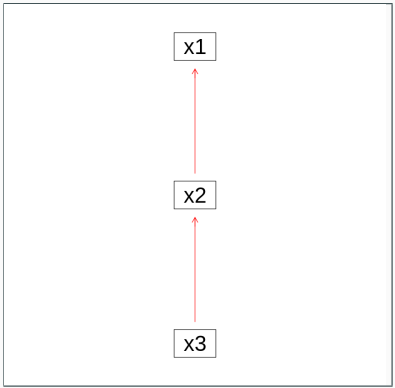


Fig. 8: K2 result vs. ground truth. Found node order: [x3] → [x2] → [x1].

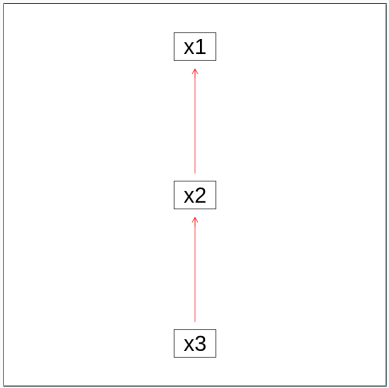


Fig. 9: Tabu result vs. ground truth. Found node order: [x3] → [x2] → [x1].

Results: ASIA dataset

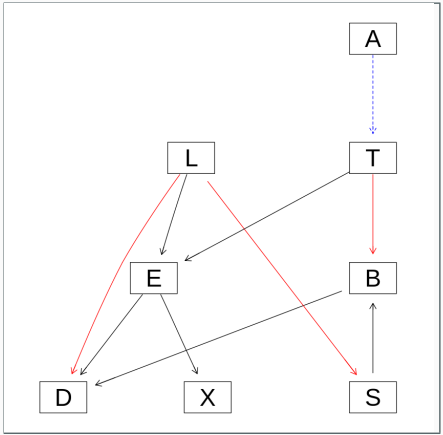


Fig. 10: K2 result vs. ground truth.

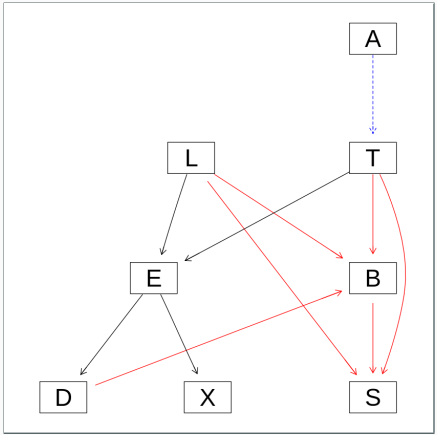


Fig. 11: Tabu result vs. ground truth.

Results: CHILD dataset

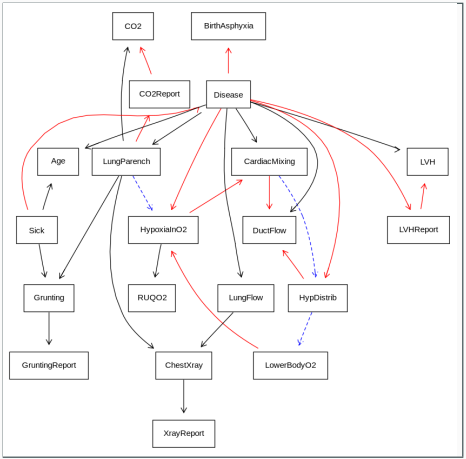


Fig. 12: K2 result vs. ground truth.

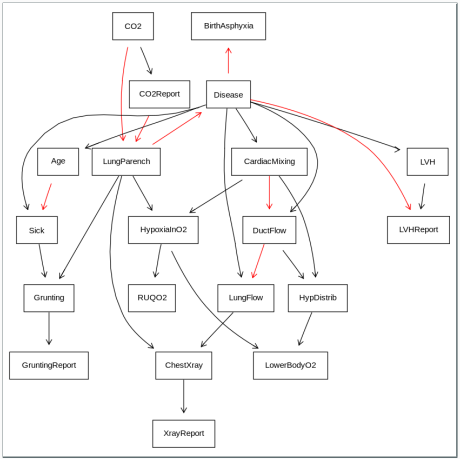


Fig. 13: Tabu result vs. ground truth.

General results: test datasets

Results

- Implementation captures general structure (even for medium-large networks)
- Similar results for tabu and K2
- false positives more frequent than false negatives

Possible improvements

- "control" arcs using blacklist/ whitelist
- Compare other algorithms (e.g. Hill climbing)
- Vary max number of parents
- Large networks $\rightarrow$ search not representative $\rightarrow$ improved ordering search algorithm needed

Analysis of the Heart Disease Dataset

UCI Heart Disease Dataset [3]

- data collected by
  - Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
  - University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
  - University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
  - V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.
- contains attributes that are considered relevant for predicting heart disease in patients
- predicted attribute **num**:
  - 0 for absence,
  - integer between 1 and 4 for presence of heart disease

[3] Andras Janosi et al. ***Heart Disease.*** UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C52P4X. 1988

Heart Disease Data Set: Attributes

Attribute	Description
age	Age of the patient in years
sex	Male/Female
cp	Chest pain type (typical angina, atypical angina, non-anginal, asymptomatic)
trestbps	Resting blood pressure (in mm Hg on admission to the hospital)
chol	Serum cholesterol in mg/dl
fbs	If fasting blood sugar > 120 mg/dl
restecg	Resting electrocardiographic results (normal, stt abnormality, lv hypertrophy)
thalch	Maximum heart rate achieved
exang	Exercise-induced angina (True/False)
oldpeak	ST depression induced by exercise relative to rest
slope	The slope of the peak exercise ST segment
ca	Number of major vessels (0-3) colored by fluoroscopy
thal	(normal; fixed defect; reversible defect)
num	The predicted attribute

Heart Disease Dataset: Approach

Goal: model as Bayesian network & try to predict presence of heart disease

Strategy

1. learn structure through self-implemented K2 algorithm
2. test 1000 random node orderings, select best network
3. compare to `bnlearn` tabu search with K2 score
4. test a revised tabu search using blacklist

Heart Disease Data Set: Data Preparation

1. Discretise continuous variables
   - concerns age, trestbps, chol, thalch, oldpeak
   - here: divide each into 3 equal-width intervals
   - convert to 3-level factors accordingly

2. Handle missing data
   - 617 out of 920 records have missing values
   - strategy: treat missing values as separate factor level in the K2 algorithm

Heart Disease Data Set Results: K2 Algorithm

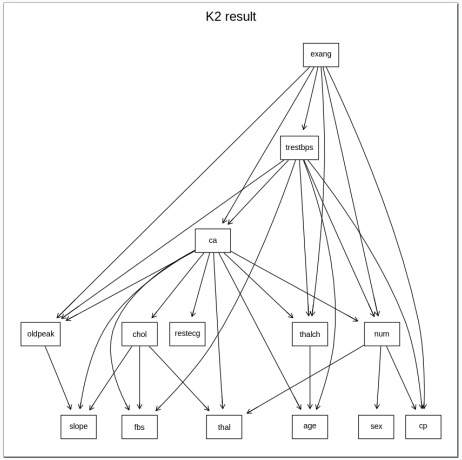


Fig. 14: K2 Result

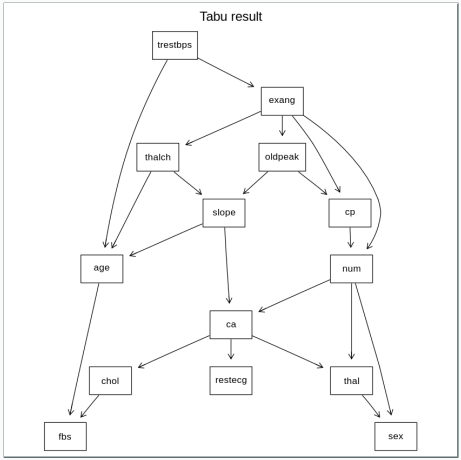


Fig. 15: Tabu Result

Heart Disease Data Set Comparison: K2 vs. Tabu

Fig. 16: Comparison: K2 vs. Tabu

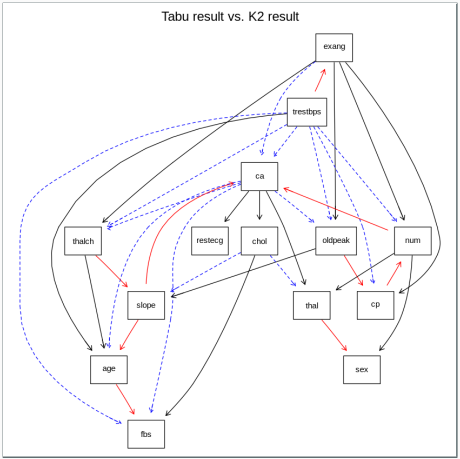
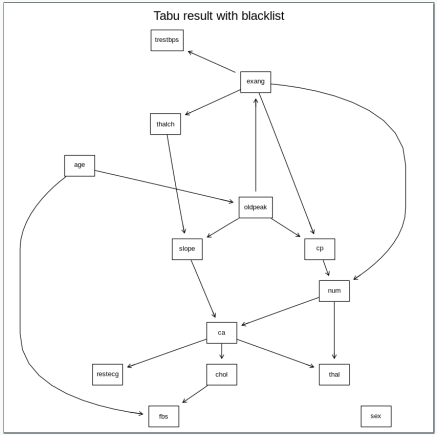
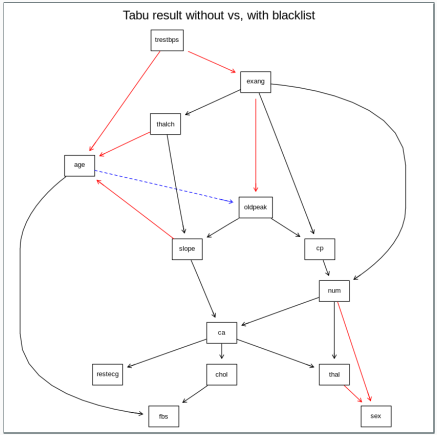


Fig. 17: Detailed Comparison: K2 vs. Tabu

Heart Disease Data Set Node Orders and Blacklist

A custom node order and blacklist were created to improve the learning process. The blacklist was used to prevent certain nodes (e.g., age and sex) from being parents of each other.





Predictions using the Learned Bayesian Network

- The learned Bayesian network was used to make predictions based on specific evidence. The probability of heart disease given certain evidence was calculated using the cpquery function.
- Predictions were generated for multiple evidence scenarios. The standard deviation of the predictions was calculated to estimate the error.

Example Prediction

Given the evidence:

- Age: 44.3-60.7 years
- Sex: Male
- Chest Pain: Typical angina
- Resting Blood Pressure: 133-200 mm Hg
- Cholesterol: 201-402 mg/dl
- Maximum Heart Rate: 107-155 bpm
- ST Depression: 0.333-3.27

The probability of no heart disease (`num` = 0) was calculated to be approximately 0.4659396.

Evidence 1: Male, Typical Angina

Given the evidence:

- Age: 44.3-60.7 years
- Sex: Male
- Chest Pain: Typical angina
- Resting Blood Pressure: 133-200 mm Hg
- Cholesterol: 201-402 mg/dl
- Maximum Heart Rate: 107-155 bpm
- ST Depression: 0.333-3.27

- Average prediction: 0.453
- Standard deviation: 0.00631

Evidence 2: Female, Typical Angina

Given the evidence:

- Age: 44.3-60.7 years
- Sex: Female
- Chest Pain: Typical angina
- Resting Blood Pressure: 133-200 mm Hg
- Cholesterol: 201-402 mg/dl
- Maximum Heart Rate: 107-155 bpm
- ST Depression: 0.333-3.27

- Average prediction: 0.812
- Standard deviation: 0.00369

Evidence 3: Male, Asymptomatic

Given the evidence:

- Age: 44.3-60.7 years
- Sex: Male
- Chest Pain: Asymptomatic
- Resting Blood Pressure: 133-200 mm Hg
- Cholesterol: 201-402 mg/dl
- Maximum Heart Rate: 107-155 bpm
- ST Depression: 0.333-3.27

- Average prediction: 0.103
- Standard deviation: 0.00236

Thank you for your attention!

Gracias por su atención!

Grazie per la vostra attenzione!

Vielen Dank für die Aufmerksamkeit!

Appendix: Notes on Derivation of the K2 score (1)

Working assumptions:

- **Assumption 1:** the database variables are discrete
- **Assumption 2:** Cases occur independently, given a belief-network model
- **Assumption 3:** There are no cases that have variables with missing values
- **Assumption 4:** The conditional probabilities are equally probable given the structure

Given assumptions 1 through 4 it is possible to derive

$$P(B_S, D) = P(B_S) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}! \tag{3}$$

where B_S is a particular topology of the Network and D is the given data set

Appendix: Notes on Derivation of the K2 score (2)

A recursive function for determining the number of possible belief-network structures with n nodes is:

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i) \tag{4}$$

The growth is exponential with n. Additional assumptions must be considered:

- **Assumption 5:** attributes are ordered, where, if attribute $\mathbf{x_i}$ precedes $\mathbf{x_j}$ in the order, structures with an arc from $\mathbf{x_j}$ to $\mathbf{x_i}$ (so $\mathbf{x_i}$ being a parent of $\mathbf{x_j}$) are not allowed
- **Assumption 6:** a node can have at most ***u*** parents
- **Assumption 7:** $P(B_S)$ is uniform

There remain $\mathbf{2^{\frac{n(n-1)}{2}}}$ possible belief-network structures The most probable network can be found as:

$$\prod_{i=1}^{n} \underset{\pi_i}{\operatorname{argmax}} \left[\prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}! \right] \tag{5}$$

Appendix: Calculation of logarithmic scoring function $\log(g(i, \pi_i))$

$$\log\left(\prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!\right) \tag{6}$$

- $g(i, \pi_i)$ may require computation of large factorials
- idea: use **logarithmic version** instead

Appendix: Calculation of logarithmic scoring function $\log(g(i, \pi_i))$

- $g(i, \pi_i)$ may require computation of large factorials
- idea: use **logarithmic version** instead

$$\log\left(\prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!\right) \tag{6}$$

$$= \sum_{j=1}^{q_i} \log\left(\frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!\right) \tag{7}$$

Appendix: Calculation of logarithmic scoring function $\log(g(i, \pi_i))$

- $g(i, \pi_i)$ may require computation of large factorials
- idea: use **logarithmic version** instead

$$\log\left(\prod_{j=1}^{q_i} \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!\right) \tag{6}$$

$$= \sum_{j=1}^{q_i} \log\left(\frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!\right) \tag{7}$$

$$= \sum_{j=1}^{q_i} \left[\log((r_i-1)!) - \log((N_{ij}+r_i-1)!) + \log\left(\prod_{k=1}^{r_i} \alpha_{ijk}!\right)\right] \tag{8}$$

Appendix: Calculation of logarithmic scoring function $\log(g(i, \pi_i))$

- $g(i, \pi_i)$ may require computation of large factorials
- idea: use **logarithmic version** instead

$$\log\left(\prod_{j=1}^{q_i} \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!\right) \tag{6}$$

$$= \sum_{j=1}^{q_i} \log\left(\frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} \alpha_{ijk}!\right) \tag{7}$$

$$= \sum_{j=1}^{q_i} \left[\log((r_i-1)!) - \log((N_{ij}+r_i-1)!) + \log\left(\prod_{k=1}^{r_i} \alpha_{ijk}!\right)\right] \tag{8}$$

$$= \sum_{j=1}^{q_i} \left[\sum_{l=1}^{r_i-1} \log l - \sum_{m=1}^{N_{ij}+r_i-1} \log m + \sum_{k=1}^{r_i} \sum_{n=1}^{\alpha_{ijk}} \log n\right] \tag{9}$$

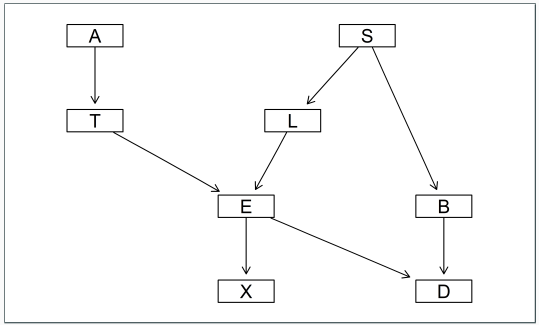
Appendix: Calculation of logarithmic scoring function $\log\left(g(i,\pi_i)\right)$

- $g(i,\pi_i)$ may require computation of large factorials
- idea: use **logarithmic version** instead

$$log(g(i,\pi_i)) = \sum_{j=1}^{q_i}\left[\sum_{l=1}^{r_i-1}\log l - \sum_{m=1}^{N_{ij}+r_i-1}\log m + \sum_{k=1}^{r_i}\sum_{n=1}^{\alpha_{ijk}}\log n\right] \tag{10}$$

Appendix: ASIA Dataset Attributes

The ASIA dataset contains the following (2-level factor) attributes:



- **D**: dyspnoea (shortness of breath) [yes/no]
- **T**: tuberculosis [yes/no]
- **L**: lung cancer [yes/no]
- **B**: bronchitis [yes/no]
- **A**: visit to Asia [yes/no]
- **S**: smoking [yes/no]
- **X**: positive chest X-ray [yes/no]
- **E**: either tuberculosis or lung cancer/ bronchitis [yes/no]

Appendix: CHILD Dataset Attributes

The CHILD dataset contains the following attributes:

- **BirthAsphyxia**: [no, yes]
- **Disease**: [Fallot, Lung, PAIVS, PFC, TAPVD, TGA]
- **Sick**: [no, yes]
- **DuctFlow**: [Lt to Rt, None, Rt to Lt]
- **CardiacMixing**: [Complete/Mild, None, Transp.]
- **LungParench**: [Abnormal, Congested, Normal]
- **LungFlow**: [High, Low, Normal]
- **LVH**: [no, yes]
- **Age**: [0-3 days, 11-30 days, 4-10 days]
- **Grunting**: [no, yes]
- **HypDistrib**: [Equal, Unequal]
- **HypoxiaInO2** [Mild, Moderate, Severe]
- **CO2** [High, Low, Normal]
- **ChestXray** [Asy/Patch, Grd Glass, Normal, Oligaemic, Plethoric']
- **LVHreport** [no, yes]
- **GruntingReport**[no, yes]
- **LowerBodyO2** [<5, 12+, 5-12]
- **RUQO2** [<5, 12+, 5-12]
- **CO2Report** [<7.5, >=7.5]
- **XrayReport** [Asy/Patchy, Grd Glass, Normal, Oligaemic, Plethoric']

Appendix: Handling missing data

- Our project: mostly filtered datasets without missing data
- real-world applications: missing data is common!

Possible strategies

1. Consider all possible cases for each missing value
2. Define missing values as a new factor level
3. Compute estimates of missing values