gilleskratzer.netlify.com

University of Zurich[UZH]

GILLES KRATZER, APPLIED STATISTICS GROUP, UZH

DATALAB BROWN BAG SEMINAR, ZHAW 30.05.2018

BAYESIAN NETWORKS LEARNING IN A

gilles.kratzer@math.uzh.ch

OUTLINE

- Motivational examples
- Elements of graph theory/probability theory
- Bayesian Network Learning
  - Constraint-based algorithms
  - Score-and-search
- Causal versus acausal thinking
- Real-data applications using R

MOTIVATIONAL EXAMPLE: CREDIT CARD FRAUD DETECTION PREDICTION

Credit Card Fraud Detection Using Bayesian and Neural Networks

Sam Maes Karl Tuyls Bram Vanschoenwinkel
Bernard Manderick
Vrije Universiteit Brussel - Department of Computer Science
Computational Modeling Lab (COMO)
Pleinlaan 2
B-1050 Brussel, Belgium
{sammaes@,ktuyls@,bvschoen@,bernard@arti.}vub.ac.be

Abstract

This paper discuss
tection by means
of digitalization,
great importance
two machine learn
ing under uncertai

experiment	±10% false pos	±15% false pos
ANN-fig 2(a)	60% true pos	70% true pos
ANN-fig 2(a)	47% true pos	58% true pos
ANN-fig 2(c)	60% true pos	70% true pos
BBN-fig 2(e)	68% true pos	74% true pos
BBN-fig 2(g)	68% true pos	74% true pos

Table 1: This table compares the results achieved with ANN and BBN, for a false positive rate of respectively 10% and 15%.

process of learning,
ble to correctly clas-
seen before as fraud-
ome features of that

s as follows: first we
main of credit card
and 4 we briefly ex-

MOTIVATIONAL EXAMPLE: VETERINARY EPIDEMIOLOGY DATA VISUALISATION

University of Zurich[UZH]

Contents lists available at SciVerse ScienceDirect

Preventive Veterinary Medicine

journal homepage: www.elsevier.com/locate/prevetmed

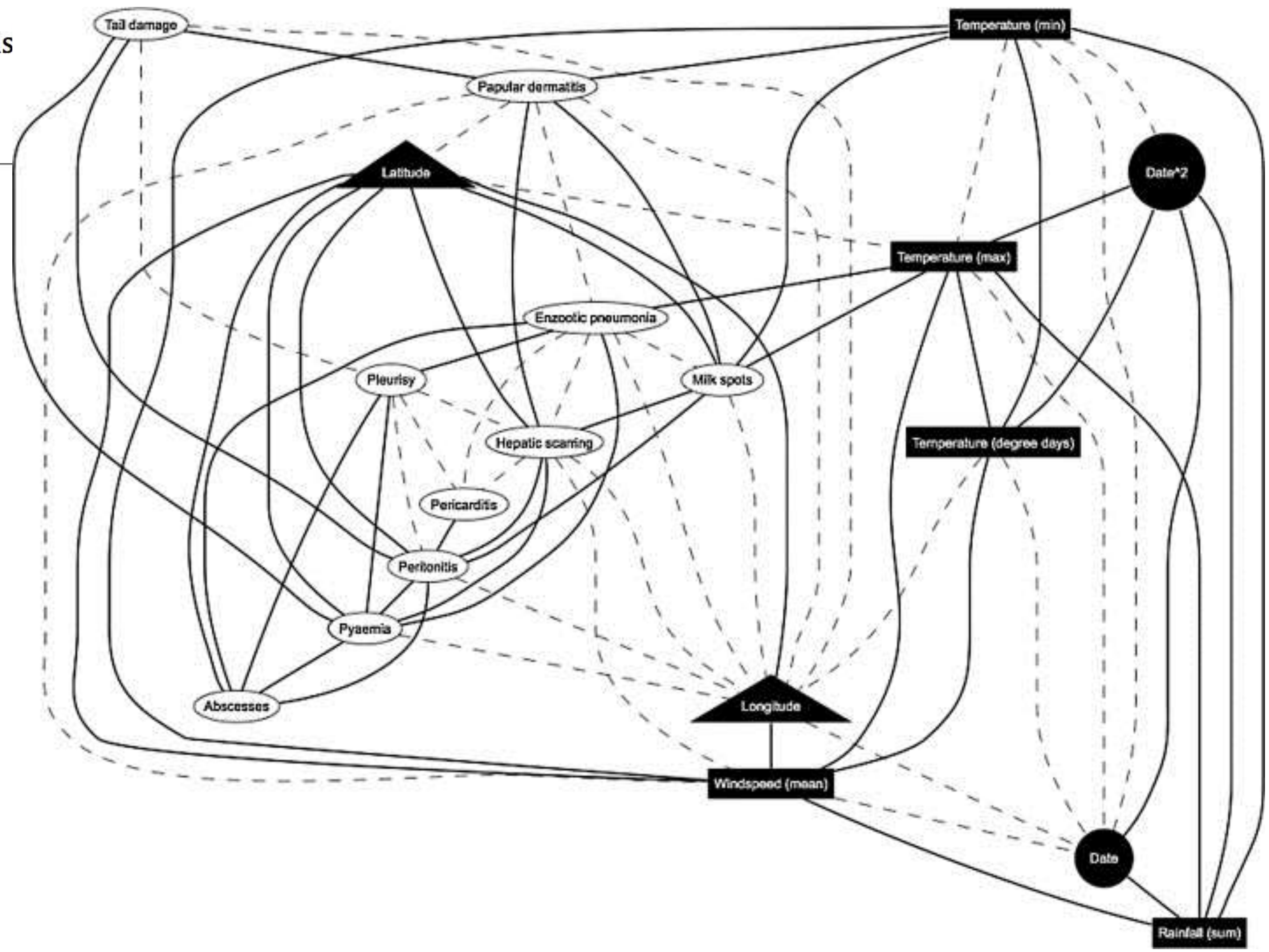
ELSEVIER

Using Bayesian networks to explore the role of weather as a potential determinant of disease in pigs

CrossMark

B.J.J. McCormick[a], M.J. Sanchez-Vazquez[b], F.I. Lewis

[a] *Fogarty International Center, National Institutes of Health, Bethesda, MD 20892, USA*
[b] *OIE Organisation Mondiale de la Santé Animale, 12, rue de Prony, 75017 Paris, France*
[c] *Section of Epidemiology, University of Zurich, Zurich, Switzerland*



MOTIVATIONAL EXAMPLE: SOCIAL SCIENCES DATA INTERPRETATION

Discovering complex interrelationships between socioeconomic status and health in Europe: A case study applying Bayesian Networks

Javier Alvarez-Galvez [a, b, *]

[a] Loyola University Andalusia, Department of International Studies, Campus de Palmas Altas, Faculty of Political Sciences and Law, Seville 41014, Spain

[b] Complutense University of Madrid, Department of Sociology IV (Research Methodology and Communication Theory), Campus de Somosaguas, Faculty of Political

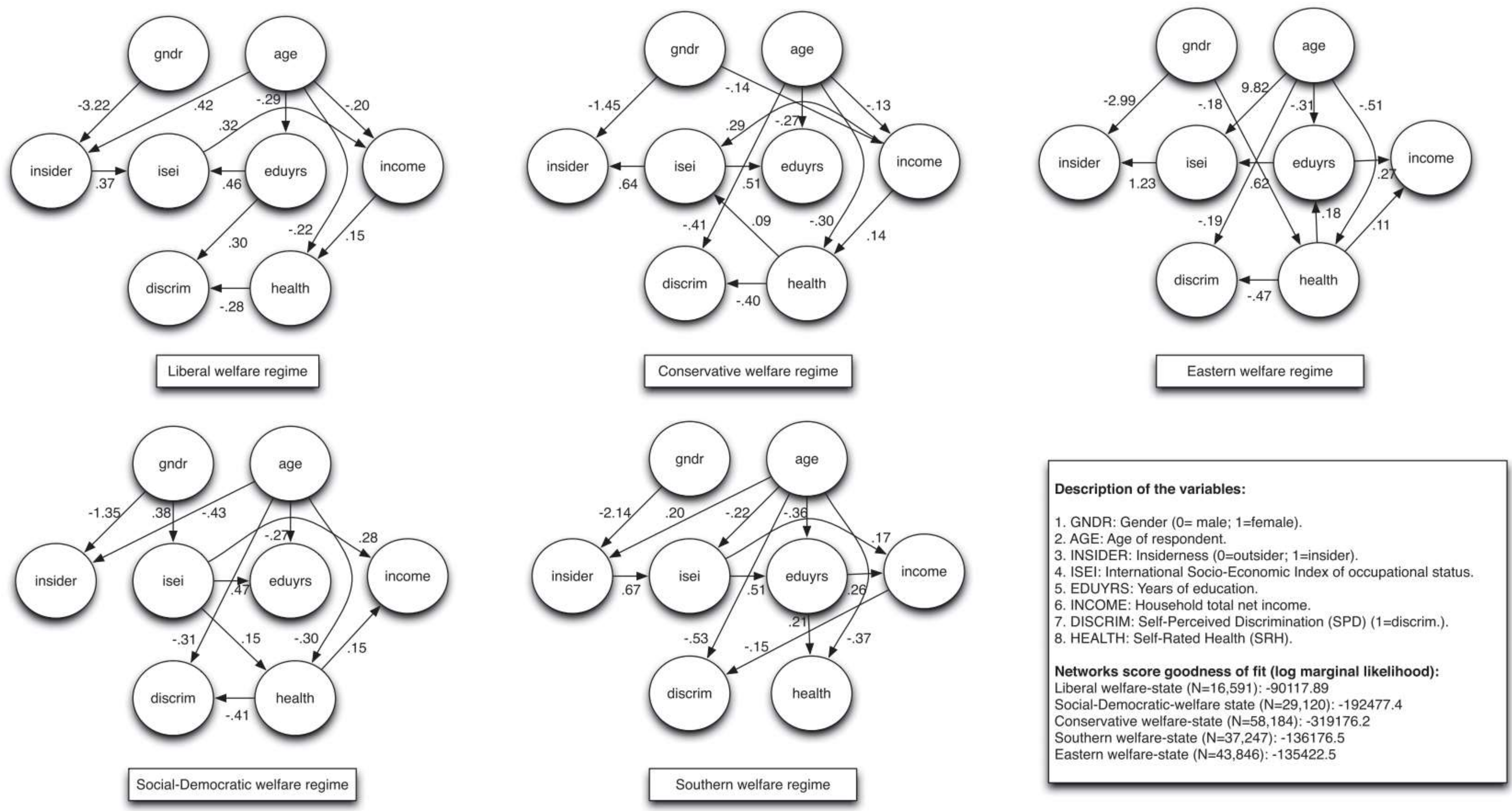


Fig. 1. Bayesian networks describing interrelationships between SES and health in five European welfare states.

BAYESIAN NETWORKS IN THE MACHINE LEARNING WORLD

University of Zurich[UZH]

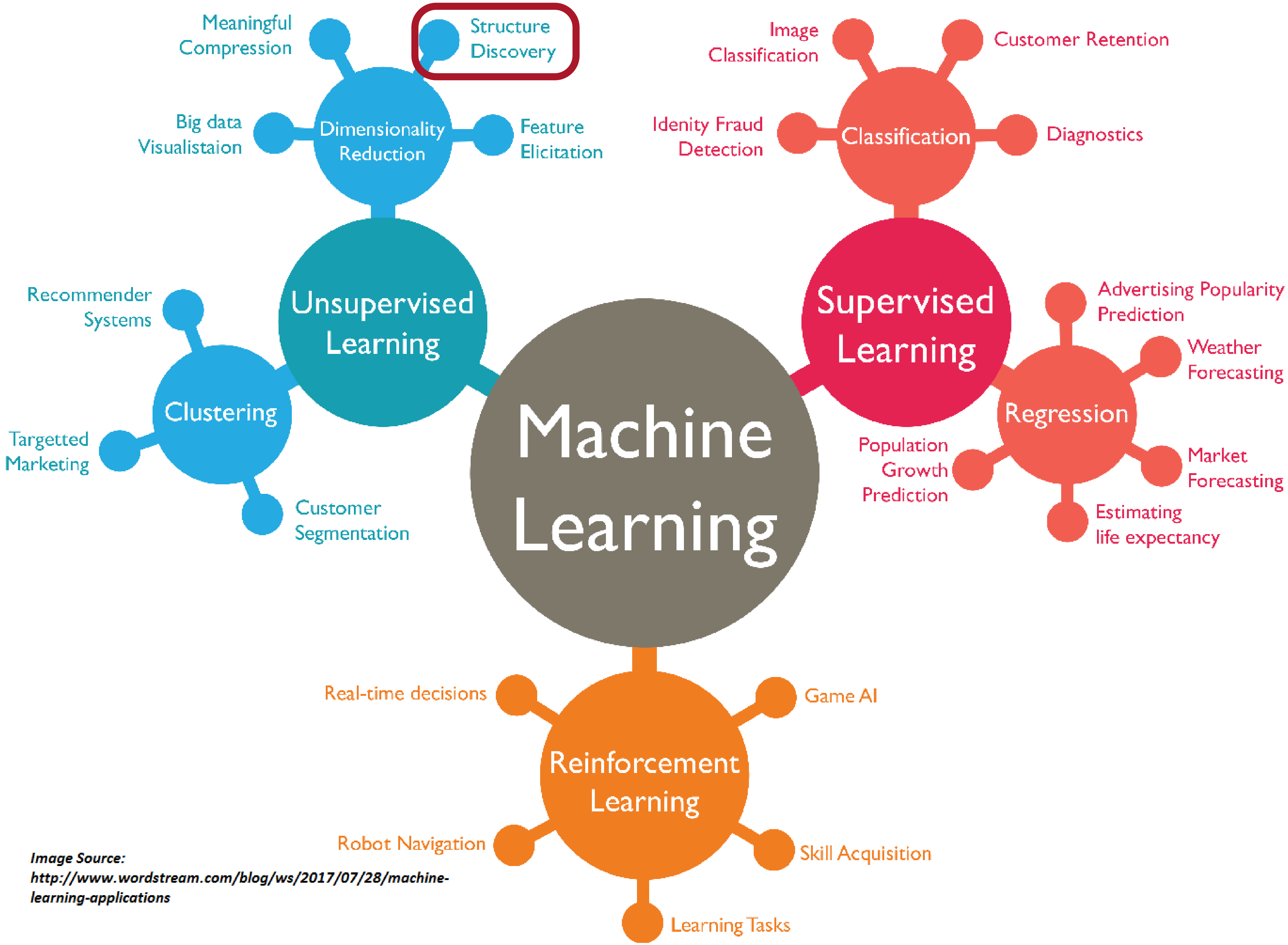


Image Source: http://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications

WHAT IS A BAYESIAN NETWORK?

Bayesian Networks are defined by two elements:

Network structure:

Directed Acyclic Graph (DAG): G = (V, A)

in which each node vi ∈ V corresponds to a random variable Xi

Probability distribution:

Probability distribution X with parameters Θ, which can be factorised into smaller local probability distributions according to the arcs aij ∈ A present in the graph.

A BN encodes the factorisation of the joint distribution

$$P(\mathbf{X}) = \prod_{j=1}^{n} P(X_j \mid \mathbf{Pa}_j, \Theta_j), \text{ where } \mathbf{Pa}_j \text{ is the set of parents of } X_j$$

The conditional probability of A given B is: $$P(A \mid B) = \frac{P(A,B)}{P(B)}$$

Bayes theorem: $$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Let A, B and C non intersecting subsets of nodes in a DAG G

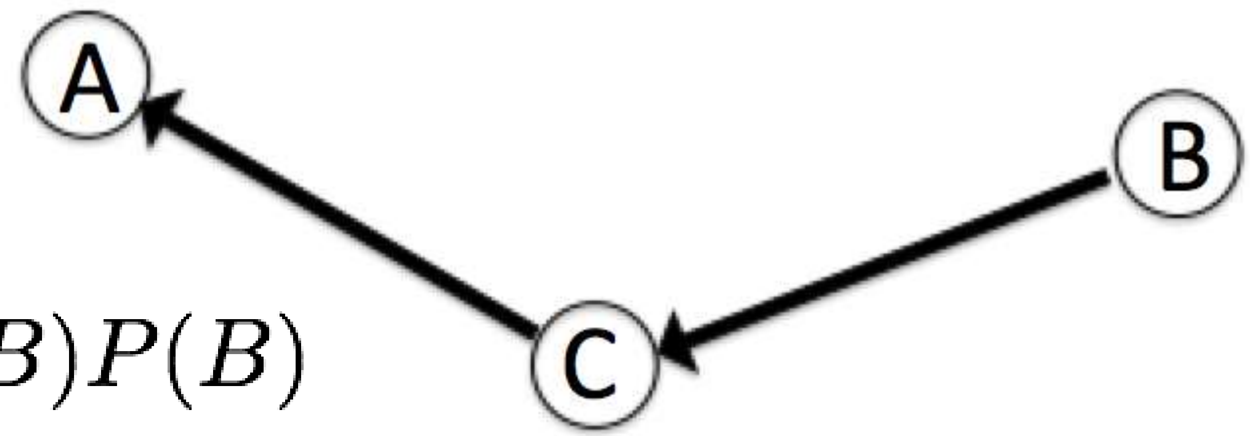
A is conditionally independent of B given C if: $A \perp\!\!\!\perp_P B|C$

$$P(A,B \mid C) = P(A \mid C)P(B \mid C)$$

Let A, B and C non intersecting subsets of nodes in a DAG G

A is conditionally independent of B given C if: $A \perp\!\!\!\perp_P B|C$

$$P(A, B \mid C) = P(A \mid C)P(B \mid C)$$



$$P(A, B, C) = P(A \mid C)P(C \mid B)P(B)$$

$$P(A, B \mid C) = \frac{P(A \mid C)P(C \mid B)P(B)}{P(C)}$$

$$= \frac{P(A \mid C)P(B, C)}{P(C)}$$

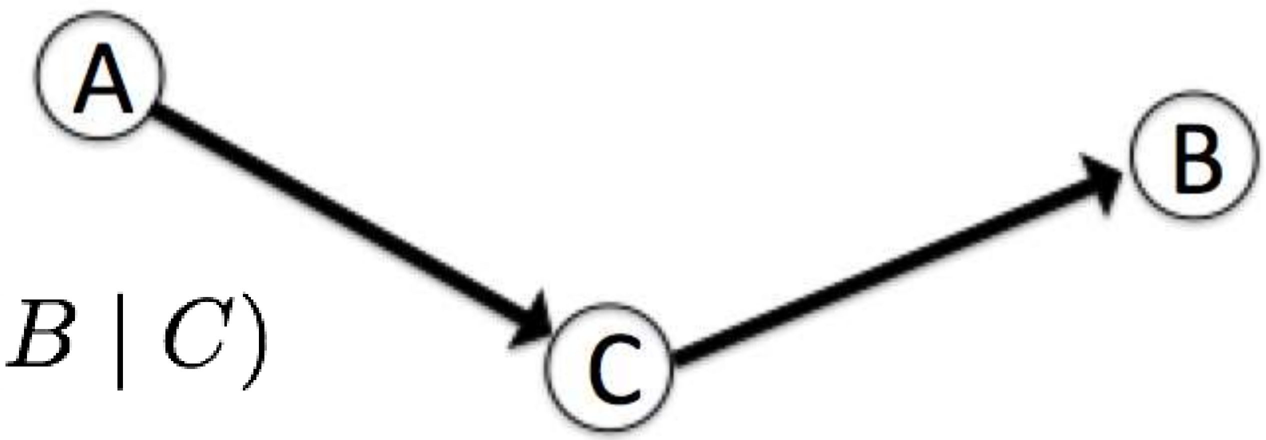
$$= P(A \mid C)P(B \mid C)$$

University of Zurich[UZH]

Let A, B and C non intersecting subsets of nodes in a DAG G

A is conditionally independent of B given C if: $A \perp\!\!\!\perp_P B|C$

$$P(A, B \mid C) = P(A \mid C)P(B \mid C)$$



$$P(A, B, C) = P(A)P(C \mid A)P(B \mid C)$$

$$P(A, B \mid C) = \frac{P(A)P(C \mid A)P(B \mid C)}{P(C)}$$

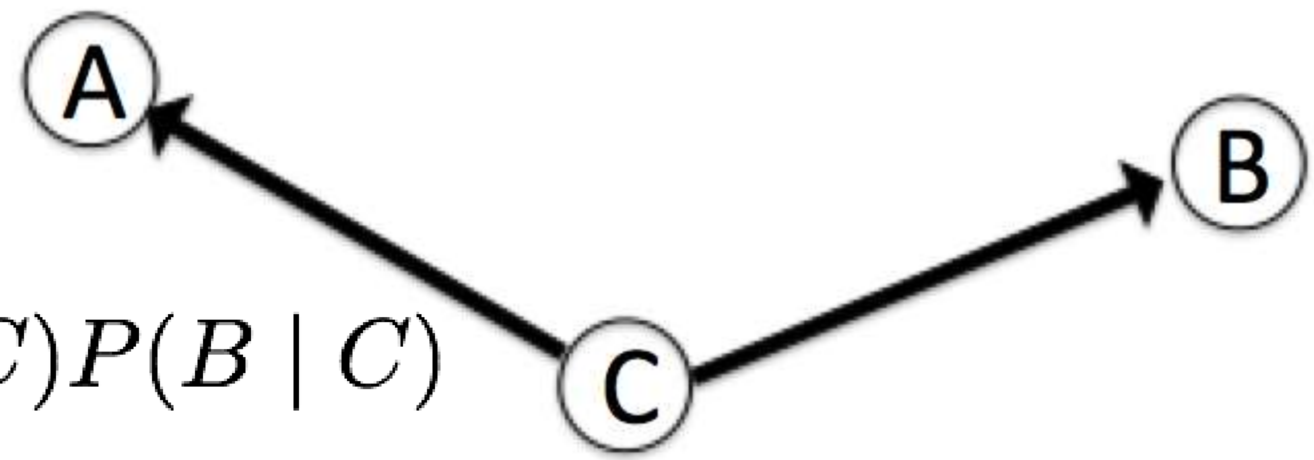
$$= \frac{P(A, C)P(B \mid C)}{P(C)}$$

$$= P(A \mid C)P(B \mid C)$$

Let A, B and C non intersecting subsets of nodes in a DAG G

A is conditionally independent of B given C if: $A \perp\!\!\!\perp_P B|C$

$$P(A, B \mid C) = P(A \mid C)P(B \mid C)$$



$$P(A, B, C) = P(C)P(A \mid C)P(B \mid C)$$

$$P(A, B \mid C) = \frac{P(C)P(A \mid C)P(B \mid C)}{P(C)}$$

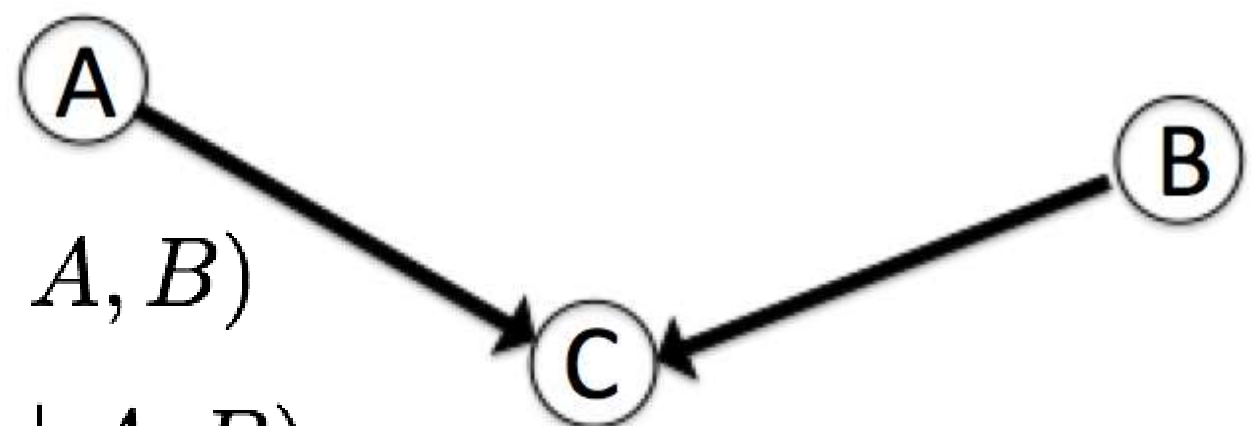
$$= P(A \mid C)P(B \mid C)$$

ELEMENT OF GRAPH THEORY

Let A, B and C non intersecting subsets of nodes in a DAG G

A is conditionally independent of B given C if: $A \perp\!\!\!\perp_P B|C$

$$P(A,B \mid C) = P(A \mid C)P(B \mid C)$$



$$P(A,B,C) = P(A)P(B)P(C \mid A,B)$$

$$P(A,B \mid C) = \frac{P(A)P(B)P(C \mid A,B)}{P(C)}$$

$$= \frac{P(A)P(B)P(A,B,C)}{P(A)P(B)P(C)}$$

$$= P(A,B \mid C)$$

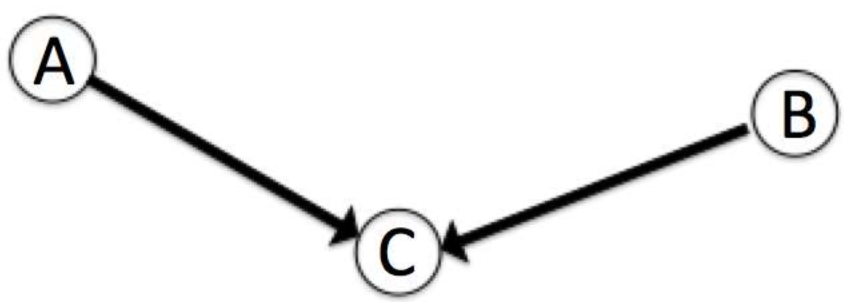
$$A \not\perp\!\!\!\perp_P B|C$$

Let A, B and C non intersecting subsets of nodes in a DAG G

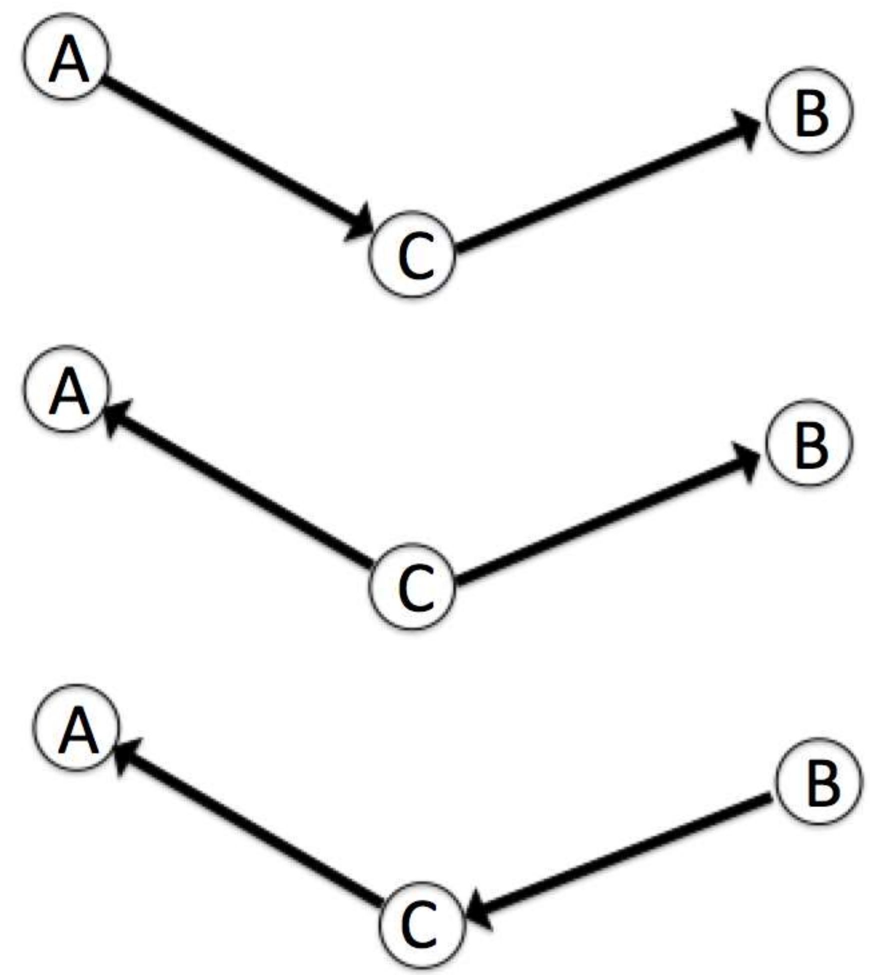
A is conditionally independent of B given C if: $A \perp\!\!\!\perp_P B|C$

$$P(A, B \mid C) = P(A \mid C)P(B \mid C)$$

$A \not\perp\!\!\!\perp_P B|C$

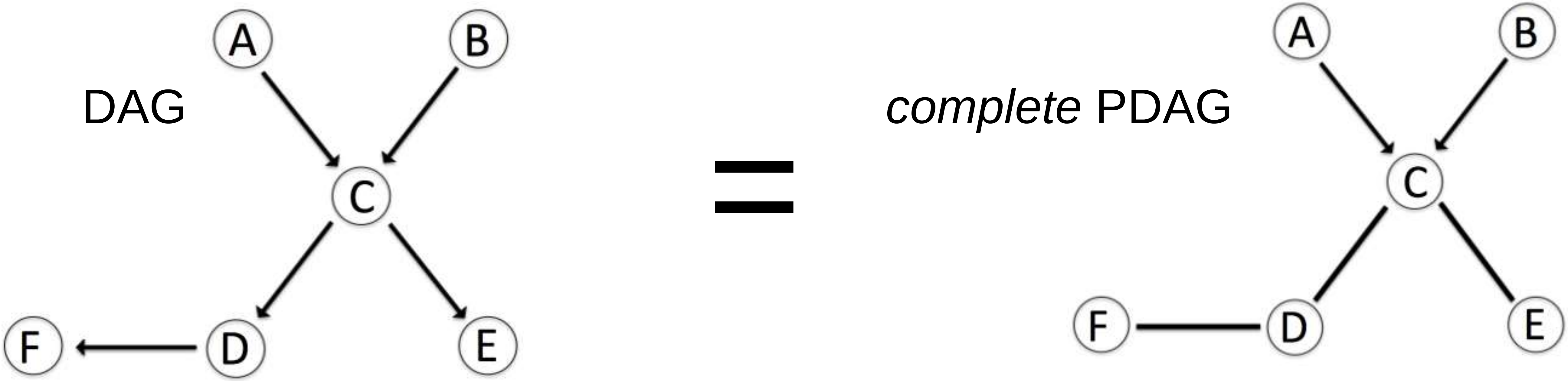


$A \perp\!\!\!\perp_P B|C$



- In a practical perspective, for observational data, if learning algorithms rely on probabilistic learning algorithm. Then one can learn up to the Markov equivalence class.

- Markov equivalence class are the set of DAGs that have the same skeleton and v-structure.



A path from A to B is blocked if it contains a node s.t. either

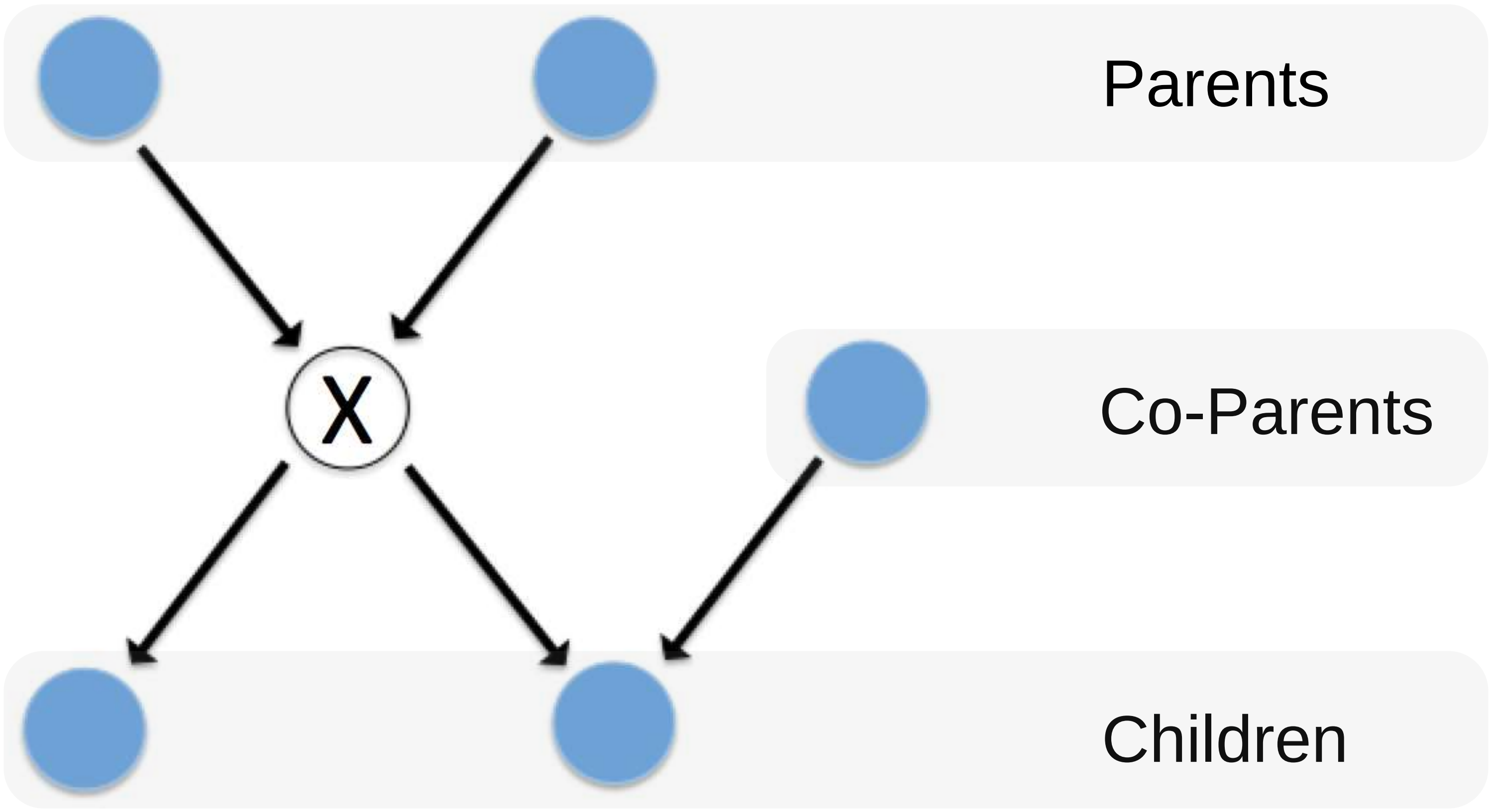
- the arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the set C, or
- the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, are C.

If all paths from A to B are blocked, A is said to be d-separated from B by C.

Theorem (Verma & Pearl, 1988): A is d-separated from B by C if, and only if, the joint distribution over all variables in the graph satisfies:

$$A \perp\!\!\!\perp_G B | C$$

The Markov Blanket of a node is the set of parents, co-parents and children.



$$P(X_k \mid X_n, k \neq n) = P(X_k \mid X_{\mathrm{MB}(k)}), \forall k$$

The Markov Blanket of a node is the set of nodes that shields the index node from the res

$$\mathcal{M} = (\mathcal{S}, \theta_{\mathcal{M}})$$

Model selection

Structure learning

Parameter estimation

Parameter learning

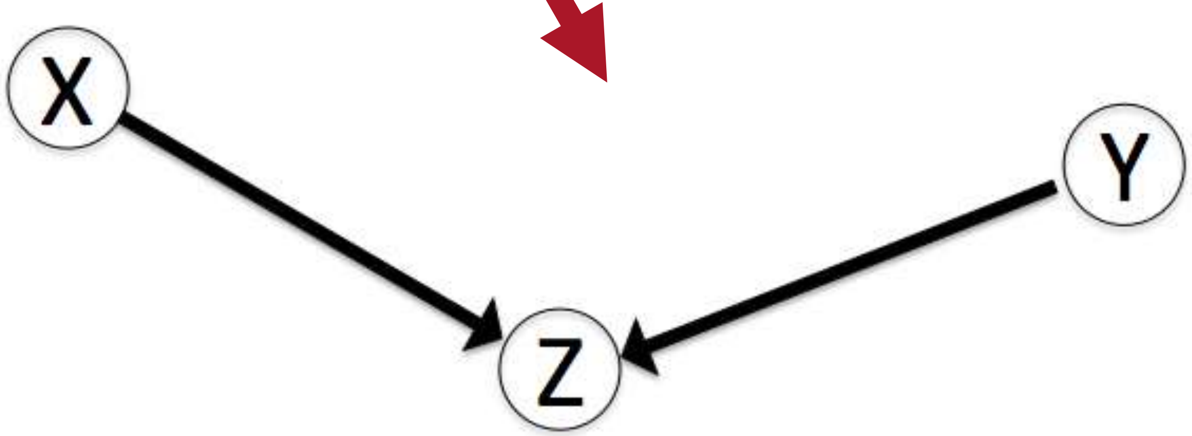
$$P(\mathcal{M}|\mathcal{D}) = \underbrace{P(\theta_{\mathcal{M}}, \mathcal{S}|\mathcal{D})}_{\text{model learning}} = \underbrace{P(\theta_{\mathcal{M}}|\mathcal{S}, \mathcal{D})}_{\text{parameter learning}} \underbrace{P(\mathcal{S}|\mathcal{D})}_{\text{structure learning}}$$

	Fully Observed data	Missing data/hidden variables
Known graph structure	Easy Sample statistics	EM algorithm Gradient ascent Variational inference Doable
Unknown graph structure	Doable **Search-and-score** **PC algorithm**	Hard Structural EM

Constraint based algorithms

$$P_{X \perp\!\!\!\perp Y|Z} < \alpha$$

$$X \perp\!\!\!\perp_s Y|Z = X \perp Y|Z$$



Search-and-score algorithms

Maximum a posteriori score

$$G^* = \underset{G}{\operatorname{argmax}}\, f(\mathcal{D}, G, n, \ldots)$$

Example of scoring functions:

- Bayesian versus ML scores
  - log marginal likelihood
  - Bayesian-Dirichlet (BDeu,BDs,BDe)
  - Bayesian Information Criterion (BIC)

Constraint-based algorithms

- *Inductive Causation* (IC): (Verma and Pearl, 1991)
  - Provides a framework for learning the structure of Bayesian networks using conditional independence tests in three steps
  - A major problem of the IC algorithm is that the first two steps cannot be applied to any real-world problem due to computational complexity …
- *PC*: first practical application of the IC algorithm (Spirtes et al., 2001)
  - backward selection procedure from the saturated graph
- *Grow-Shrink* (GS) (Margaritis, 2003)
  - Simple forward selection MB detection approach
- *Incremental Association* (IAMB): (Tsamardinos et al., 2003)
  - two-phase selection scheme based on a forward selection followed by a backward selection of the MB

- Constraint based methods require a Markov and faithfulness assumption
- Conditional independencies in the distribution exactly equal the ones encoded in the DAG via d-separation

$$A \perp\!\!\!\perp_G B | C \underset{\text{Faithful}}{\overset{\text{Markov}}{\rightleftarrows}} A \perp\!\!\!\perp_P B | C$$

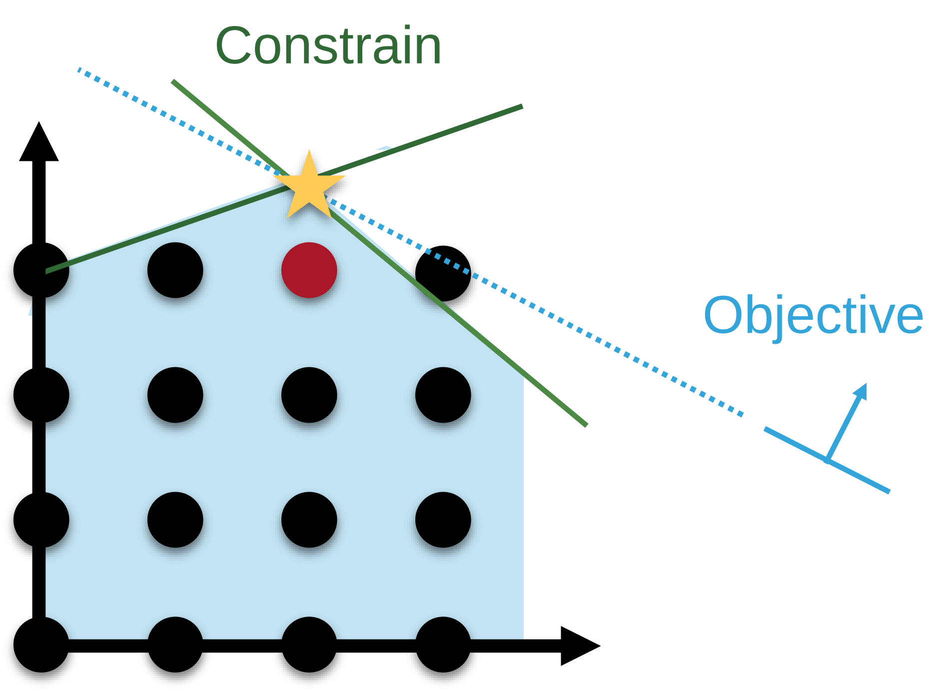
- Causal sufficiency: no unmeasured common causes

In a pratical perspective:

- Testing mixture of data?
- Testing assumptions?

Score-and-search algorithms

- *Heuristic approaches / Greedy search*
  - Hill-climbing (with possibly random restarts/stochastics …)
  - Tabu search (Glover, 1986)
  - Simulated annealing (Kirkpatrick et al, 1983)
  - Plus an entire zoo of methods …
- *Exact search*
  - Exact node ordering (Koivisto et al. , 2004)
  - Learning with cutting planes (Cussens, 2012)



Scores

- Decomposability!
- Discrete BNs:
  - Bayesian-Dirichlet: BDeu (Heckerman et al. ,1995)
- Score equivalence for additive regression framework:
  - Bayesian based scores: not always score equivalent due to the prior!
  - Information theoretic scores: BIC asymptotically score equivalent

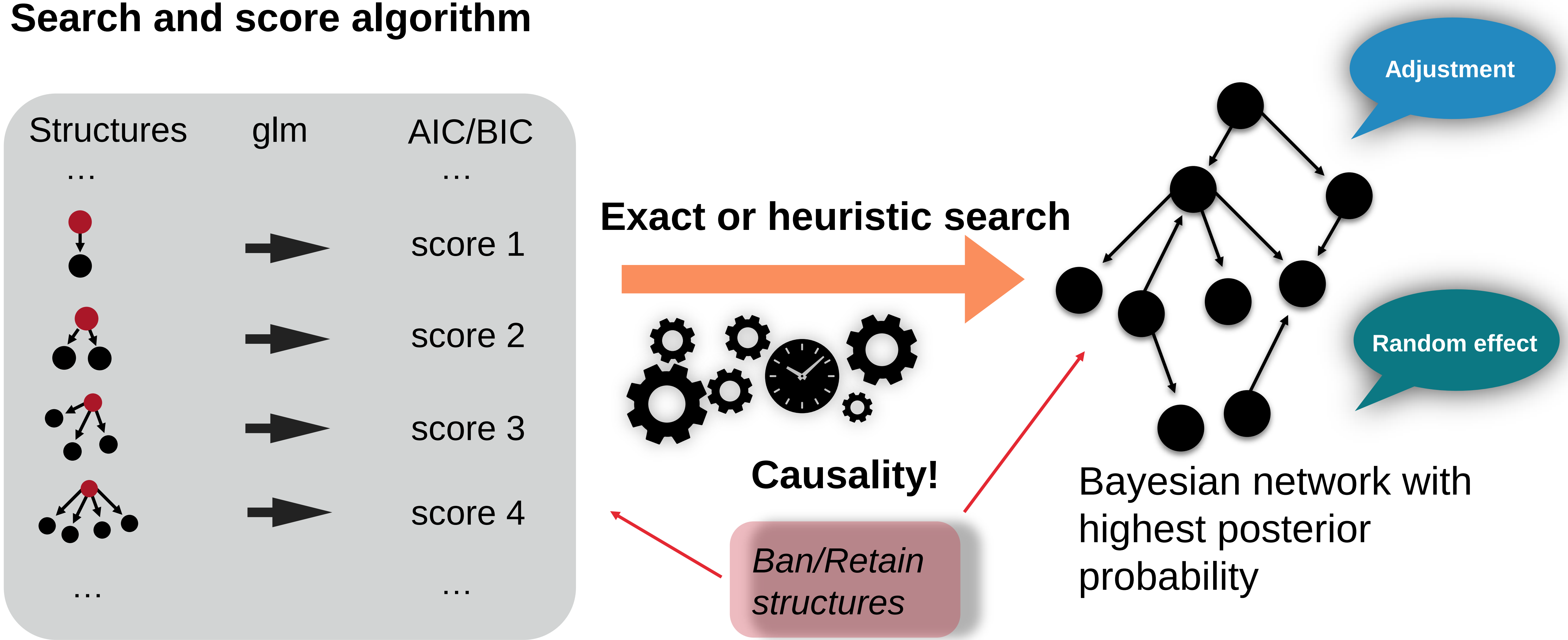
Counter example

- Maximum likelihood estimator … return fully connected BN!

In a practical perspective:

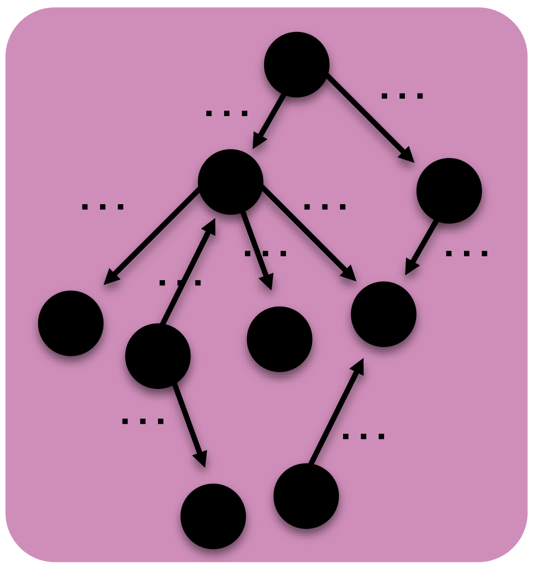
- Scoring mixture of data?
- Score equivalence!

Search and score algorithm



Parameter estimation

- compute marginal posterior density
- regression estimate



Using R

buildscorecache()

mostprobable()

fitabn()

CAUSAL THINKING VERSUS ACAUSAL THINKING

- Strong assumptions … but common in statistics, no?
- "*It seems that if conditional independence judgements are byproducts of stored causal relationships, then tapping and representing those relationships directly would be a more natural and more reliable way of expressing what we know or believe about the world. This is indeed the philosophy behind causal Bayesian networks.*" (Pearl, 2009)
- The do-calculus
  - Interventions
  - In epidemiology: Randomised Controlled Trial
- So … BN is a nice framework to treat causal and causal thinking

Popular R packages (available on CRAN)

bnlearn

- Learning via constraint-based and score-based algorithms (many!)

pcalg

- Robust estimation of CPDAG via the PC-Algorithm

deal

- Learning BNs with mixed (discrete and continuous) variables

catnet

- Discrete BNs using likelihood-based criteria

abn

- Learning BNs with mixed (discrete, continuous, Poisson) variables
- Score based methods: Bayesian and frequentist estimation
- Exact and heuristic search

Disclaimer: I am author and maintainer of the abn R package. I will use it for the example part.

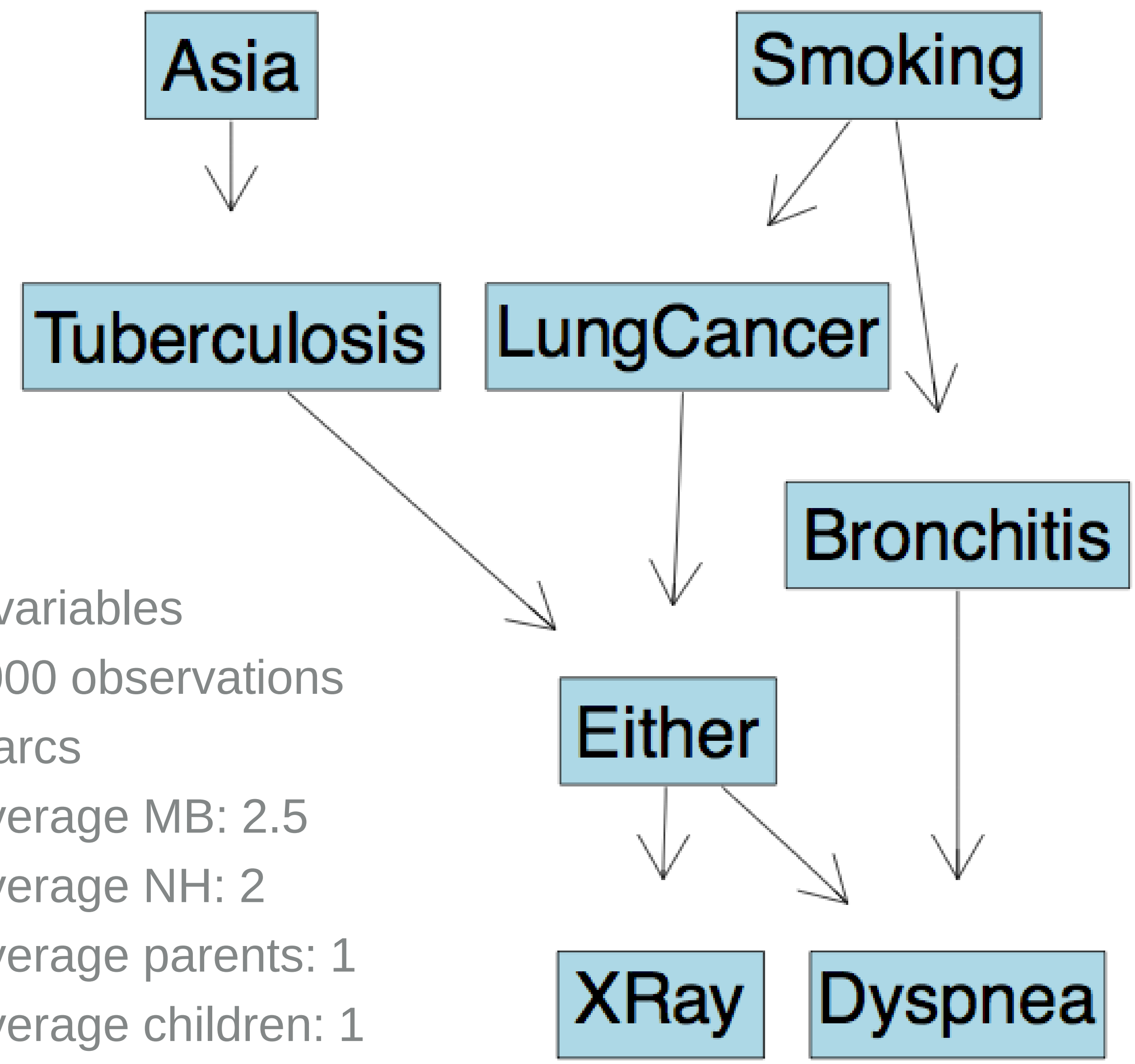
R CODE: EXAMPLE ASIA

Proposed by Lauritzen et al.,1988 and provided by Scutari, 2009

“Shortness-of-breath (dyspnoea) may be due to tuberculosis, lung cancer or bronchitis, or none of them, or more than one of them. A recent visit to Asia increases the chances of tuberculosis, while smoking is known to be a risk factor for both lung cancer and bronchitis. The results of a single chest X-ray do not discriminate between lung cancer and tuberculosis, as neither does the presence or absence of dyspnoea.”

```
##defining distributions
dist = list(Asia = "binomial",
            Smoking = "binomial",
            Tuberculosis = "binomial",
            LungCancer = "binomial",
            Bronchitis = "binomial",
            Either = "binomial",
            XRay = "binomial",
            Dyspnea = "binomial")

#plot BN
plotabn(dag.m = ~Asia|Tuberculosis +
            Tuberculosis|Either +
            Either|XRay:Dyspnea +
            Smoking|Bronchitis:LungCancer +
            LungCancer|Either +
            Bronchitis|Dyspnea,
          data.dists = dist,
          edgedir = "cp",
          fontsize.node = 30,
          edge.arrowwise = 3)
```

8 variables
5000 observations
8 arcs
Average MB: 2.5
Average NH: 2
Average parents: 1
Average children: 1



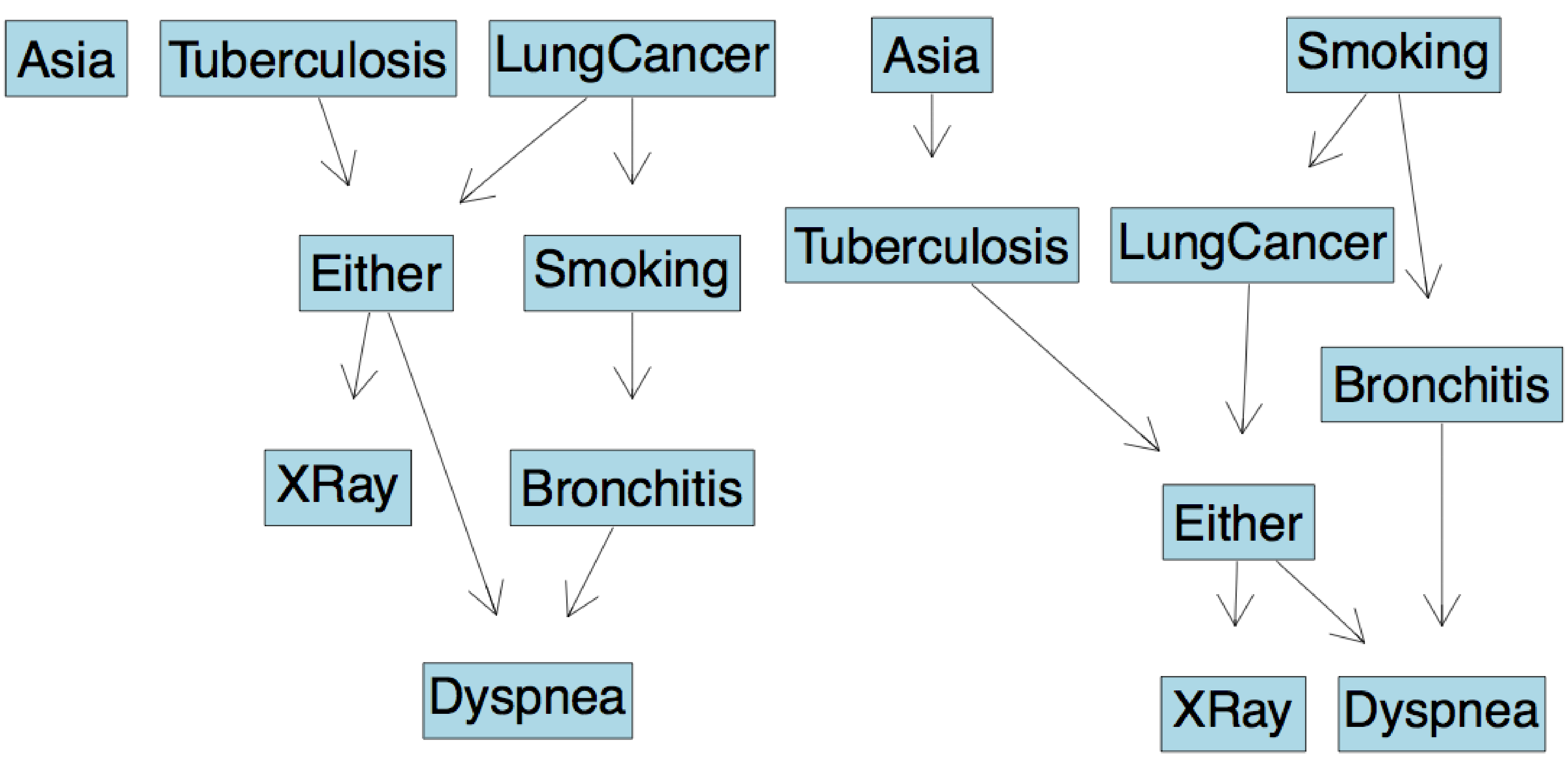
ASIA: SCORE BASED ALGORITHM

```
##===========================
##score based algorithm
##===========================

#loglikelihood score
bsc.compute <- buildscorecache(data.df = asia,
                                data.dists = dist,
                                max.parents = 2)

dag <- mostprobable(score.cache = bsc.compute)
plotabn(dag.m = dag,data.dists = dist, fontsize.node = 30, edge.arro
```

```
> compareDag(ref = t(dag.adj),
+            test = dag)
$TPR
[1] 0.75

$FPR
[1] 0.01785714

$Accuracy
[1] 0.953125

$FDR
[1] 0.2857143

$`G-measure`
[1] 0.8017837

$`F1-score`
[1] 44.8

$PPV
[1] 0.8571429

$FOR
[1] 0.2857143

$`Hamming-distance`
[1] 3
```

Learned

Truth



ASIA: KNOWN NETWORK

University of Zurich[UZH]

```
fitabn(dag.m = ~Asia|Tuberculosis+
        Tuberculosis|Either +
        Either|XRay:Dyspnea +
        Smoking|Bronchitis:LungCancer +
        LungCancer|Either +
        Bronchitis|Dyspnea,data.df = asia,data.dists = dist)$modes
```

```
$Asia
 Asia|(Intercept) Asia|Tuberculosis
        -4.811200          1.765763

$Smoking
Smoking|(Intercept)   Smoking|LungCancer   Smoking|Bronchitis
          -1.027065             2.356988             1.807460

$Tuberculosis
Tuberculosis|(Intercept)       Tuberculosis|Either
               -12.22120                  10.21823

$LungCancer
LungCancer|(Intercept)       LungCancer|Either
             -12.07565                14.18547

$Bronchitis
Bronchitis|(Intercept)      Bronchitis|Dyspnea
             -1.388644                3.200393

$Either
Either|(Intercept)          Either|XRay       Either|Dyspnea
         -8.656348             8.259773             1.538789

$XRay
XRay|(Intercept)
       -2.052496

$Dyspnea
Dyspnea|(Intercept)
         -0.1201444
```

```
fitabn.mle(dag.m = dag.adj,data.df = asia,data.dists = dist)$coef
```

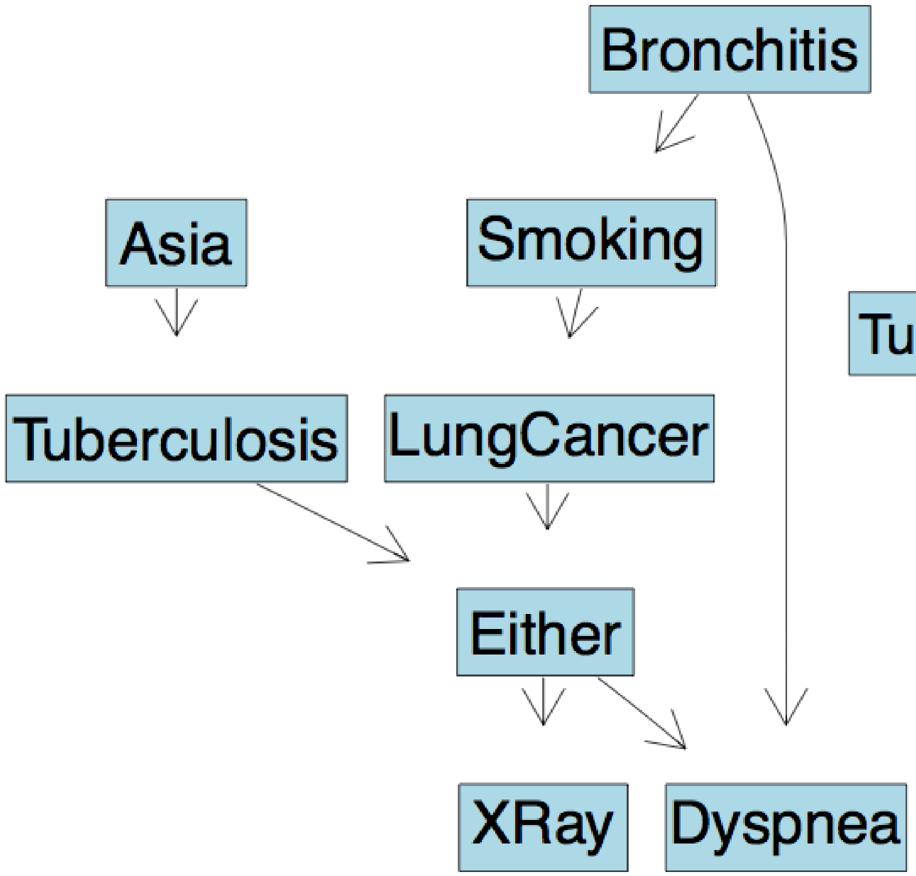
```
$Asia
     Asia|intercept Tuberculosis
[1,]      -4.811371     1.766849

$Smoking
     Smoking|intercept LungCancer Bronchitis
[1,]         -1.027075   2.357079   1.807472

$Tuberculosis
     Tuberculosis|intercept   Either
[1,]              -8.517393 6.516139

$LungCancer
     LungCancer|intercept   Either
[1,]            -8.517393 10.62598

$Bronchitis
     Bronchitis|intercept  Dyspnea
[1,]            -1.388655 3.200415

$Either
     Either|intercept     XRay  Dyspnea
[1,]        -8.665128 8.268402 1.539146

$XRay
     XRay|intercept
[1,]        -2.0525

$Dyspnea
     Dyspnea|intercept
[1,]        -0.1201443
```

ASIA: EXTERNAL KNOWLEDGE

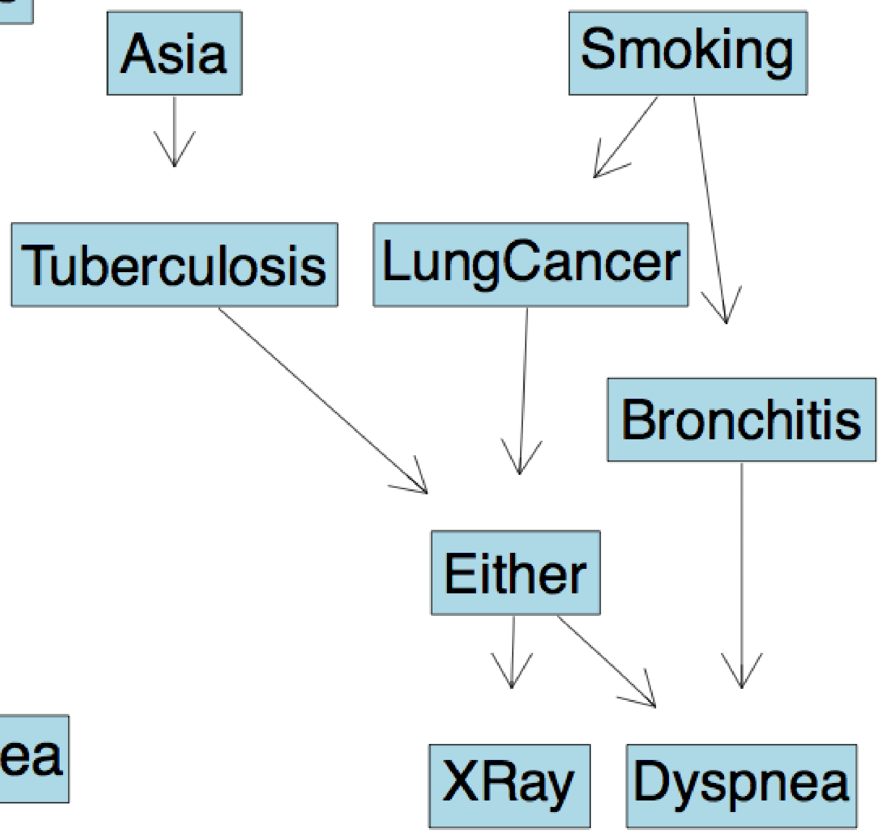
```
##==========================
##external knowledge
##==========================

##recent visit to Asia increases risk of tuberculosis
bsc.compute <- buildscorecache.mle(data.df = asia,
                                   data.dists = dist,
                                   max.parents = 2,
                                   dag.retained = ~Tuberculosis|Asia)

dag <- mostprobable(score.cache = bsc.compute,score = "bic")
plotabn(dag.m = dag,data.dists = dist, fontsize.node = 30, edge.arro
```

```
> compareDag(ref = t(dag.adj),
+            test = (dag))
$TPR
[1] 0.875

$FPR
[1] 0.01785714

$Accuracy
[1] 0.96875

$FDR
[1] 0.125

$`G-measure`
[1] 0.875

$`F1-score`
[1] 56

$PPV
[1] 0.875

$FOR
[1] 0.125

$`Hamming-distance`
[1] 2
```

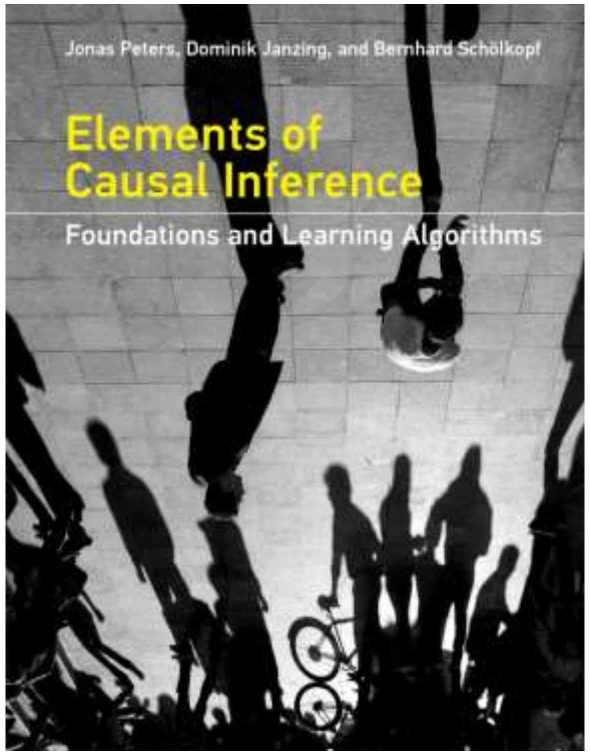
Learned

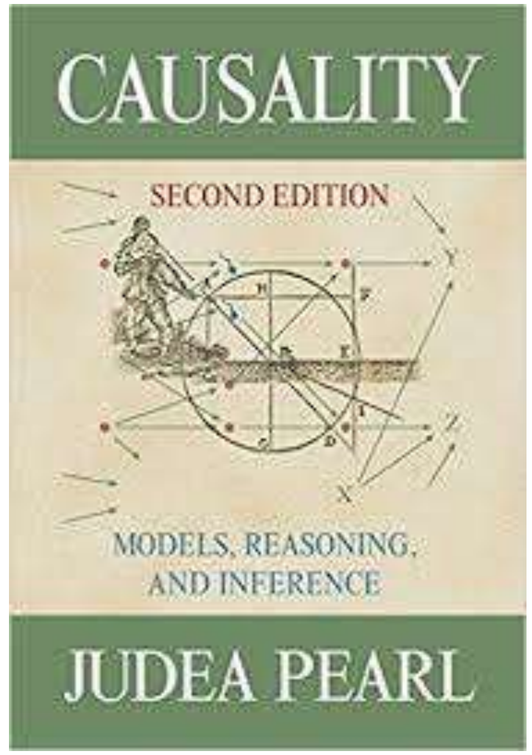


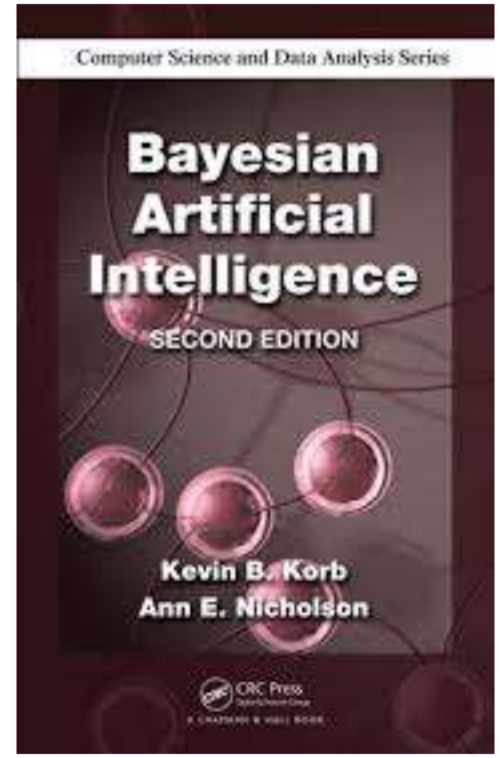
Truth

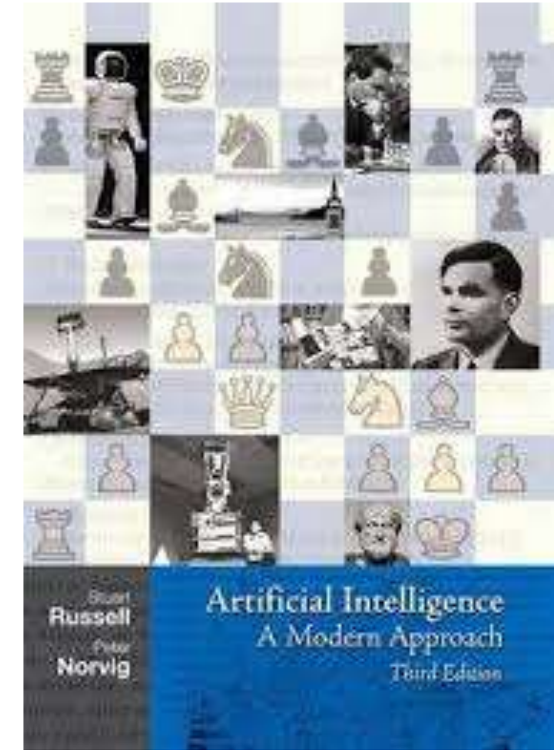


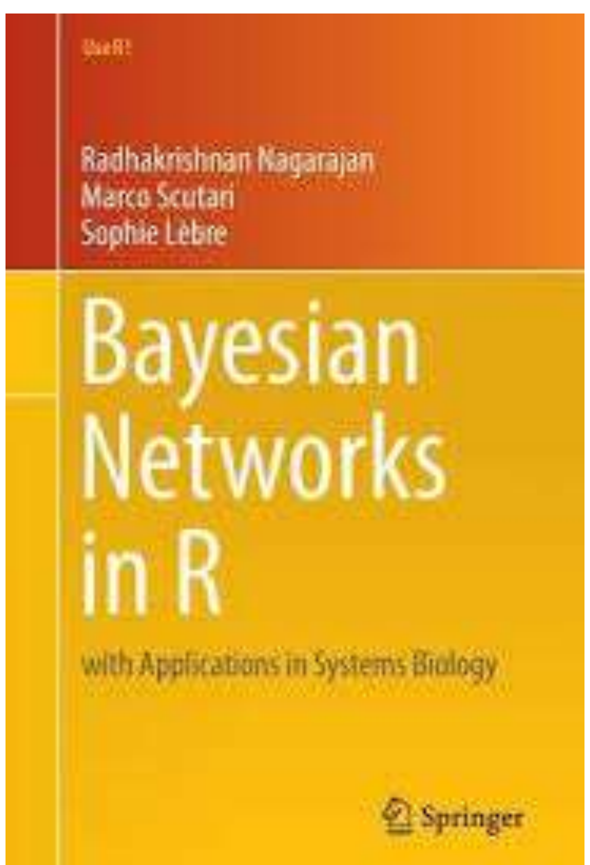
SELECTED BIBLIOGRAPHY

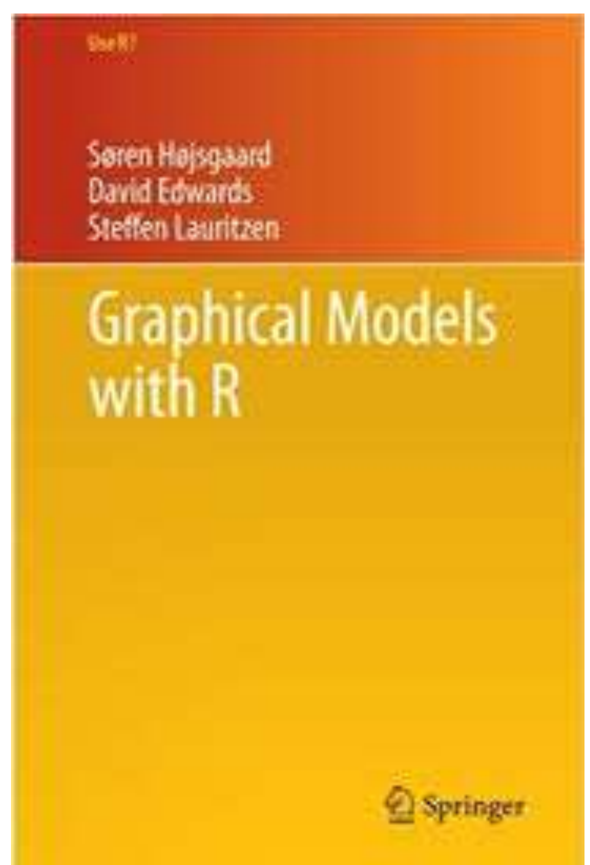


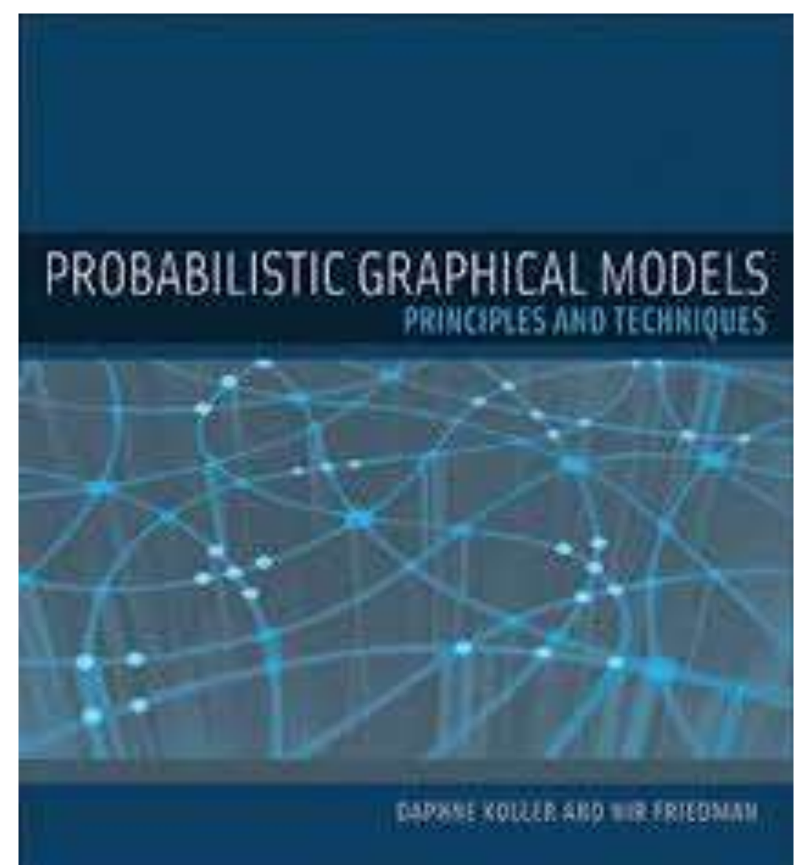












Thank you for your attention

xkcd.com

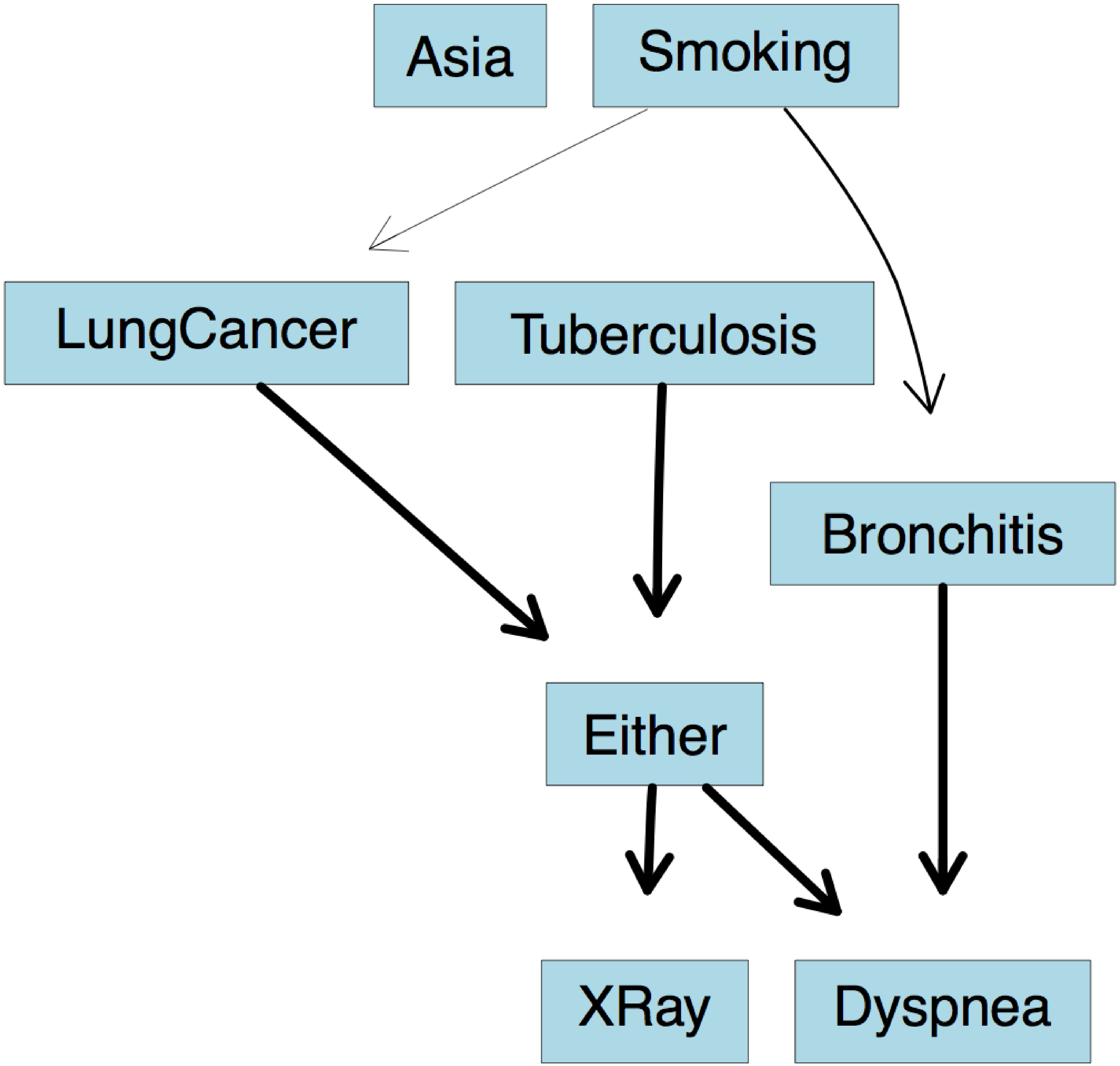
Backup slides

ASIA: BOOTSTRAPPING

University of Zurich[UZH]

```
library(doParallel)
library(foreach)

cl <- makeCluster(2)
registerDoParallel(cl)

set.seed(1120)
nboot <- 200
nvars <- dim(asia)[2]
nobs <- dim(asia)[1]
bootstrap.dag <- array(data = NA,dim = c(nvars, nvars, nboot))

start_time <- Sys.time()
bootstrap.dag <- foreach(i = 1:nboot,.packages = c("mlabn", "abn")) %dopar% {
  mycache.computed.mle <- buildscorecache.mle(data.df = asia[sample(x = 1:nobs,size = 0.8*nobs,replace = FALSE),],
                                                                  data.dists = dist,
                                                                  max.parents = 2,
                                                                  dry.run = FALSE,
                                                                  maxit = 1000,
                                                                  tol = 1e-11)

  dag <- mostprobable(score.cache = mycache.computed.mle, score = "bic")}
compute_time <- Sys.time()-start_time

##analysis
df.boot <- array(data = unlist(bootstrap.dag), dim = c(8, 8, 200))

dag<-apply(df.boot, 1:2, mean)

#dag.mdl<-dag.before

colnames(dag) <- rownames(dag) <- names(dist)

dag.boot.50 <- dag
dag.boot.50[dag>0.5] <- 1
dag.boot.50[dag<=0.5] <- 0

dag[dag<=0.5] <- 0

colnames(dag.boot.50) <- rownames(dag.boot.50) <- names(dist)

plotabn(dag.m = t(dag.boot.50),data.dists = dist,fontsize.node = 30,arc.strength = 10*dag,digit.precision = 2,edge.arrowwise = 3)
```

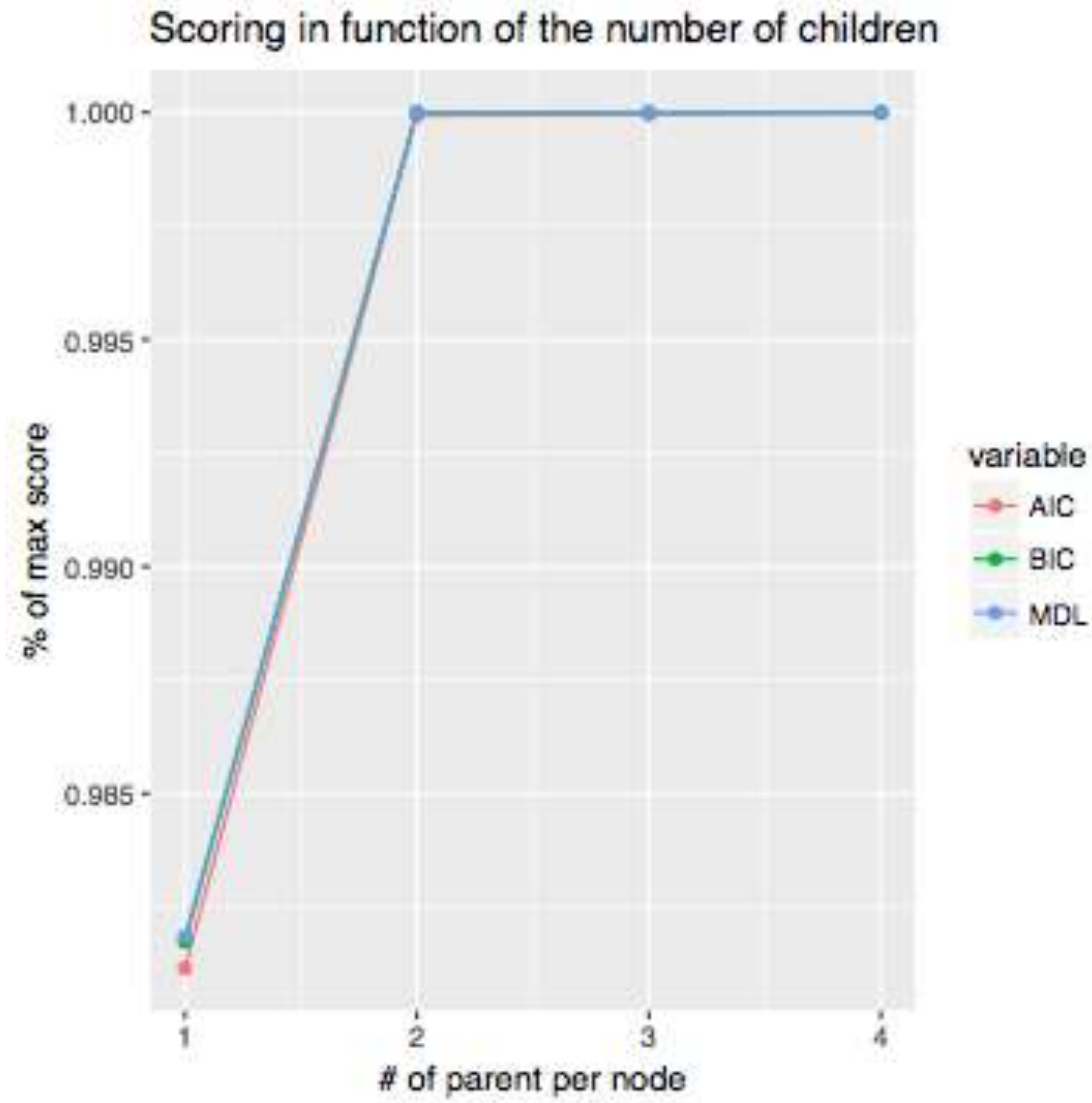
ASIA: BOOTSTRAPPING



ASIA: HOW MANY PARENT ARE NEEDED?

```r
res.mlik <- NULL
res.aic <- NULL
res.bic <- NULL
res.mdl <- NULL

for(i in 1:4){
  mycache.computed.mle <- buildscorecache.mle(data.df = asia,
                                              data.dists = dist,
                                              max.parents = i,
                                              dry.run = FALSE,
                                              maxit = 1000,
                                              tol = 1e-11)

  dag <- mostprobable(score.cache = mycache.computed.mle,score = "aic")
  res.aic <- rbind(res.aic,fitabn.mle(dag.m = dag,data.df = mycache.computed.mle$data.df,data.dists = dist)$aic)
  dag <- mostprobable(score.cache = mycache.computed.mle,score = "bic")
  res.bic <- rbind(res.bic,fitabn.mle(dag.m = dag,data.df = mycache.computed.mle$data.df,data.dists = dist)$bic)
  dag<-mostprobable(score.cache = mycache.computed.mle,score = "mdl")
  res.mdl <- rbind(res.mdl,fitabn.mle(dag.m = dag,data.df = mycache.computed.mle$data.df,data.dists = dist)$mdl)
}

library(ggplot2)
library(reshape)
scoring <- data.frame(AIC = max(-res.aic)/-res.aic, BIC = max(-res.bic)/-res.bic, MDL = max(-res.mdl)/-res.mdl, 1:4)

scoring.long <- melt(scoring, id.vars="X1.4")

ggplot(data = scoring.long, aes(x=X1.4, y=(value), group=variable, color=variable)) +
  geom_line() +
  geom_point() +
  ggtitle("Scoring in function of the number of children", subtitle = NULL) +
  xlab("# of parent per node") +
  ylab("% of max score") +
  scale_x_continuous(breaks=c(1,2,3,4,5,6,7))
```

ASIA: HOW MANY PARENT ARE NEEDED?



ASIA: CONSTRAINT-BASED LEARNING

```
##============================
## constraint-based algorithm
##============================

bn.gs <- gs(asia)
plot(bn.gs)

bn.iamb <- iamb(asia)
plot(bn.iamb)
```