LEARNING BAYESIAN NETWORKS

Outline

1. Introduction
2. Learning Parameters
3. Learning Structures
4. Summary

Outline

1 Introduction

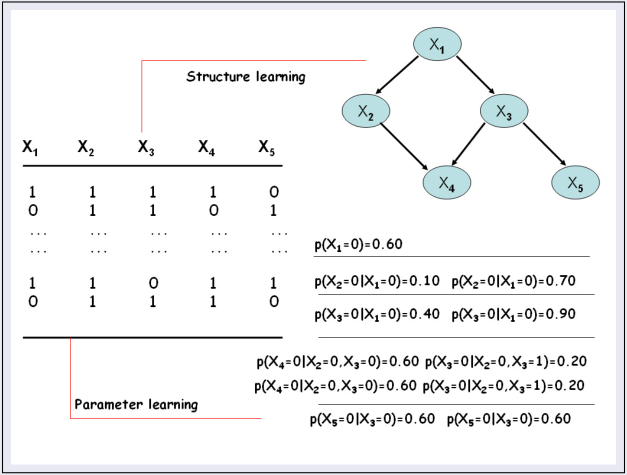
2 Learning Parameters

3 Learning Structures

4 Summary

From data to Bayesian networks

Learning structure and parameters



Discovering associations

The task of learning Bayesian networks from data

- Given a data set of cases $D = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)}\}$ drawn at random from a joint probability distribution $p_0(x_1, ..., x_n)$ over $X_1, ..., X_n$, and possibly some domain expert background knowledge
- The task consists of identifying (learning) a DAG (directed acyclic graph) structure S and a set of corresponding parameters $\mathbf{\Theta}$

Discovering associations

The task of learning Bayesian networks from data

- When discovering associations all the variables have the same treatment
- There is not a target variable, as in supervised classification
- There is not a hidden variable, as in clustering

Outline

1. Introduction
2. Learning Parameters
3. Learning Structures
4. Summary

Maximum likelihood estimation

Parameter space

- Consider a variable X with r possible values: $\{1, 2, ..., r\}$
- We have N observations (cases) of X: $D = \{x_1, .., x_N\}$, that is a sample of size N extracted from X
    - Example: X variable measuring the result obtained after rolling a dice five times. $D = \{1, 6, 4, 3, 1\}$, $r = 6$, and $N = 5$
- We are interested in estimating: $P(X = k)$
- The parametric space is $\mathbf{\Theta} = \{\boldsymbol{\theta} = (\theta_1, ..., \theta_r) | \theta_i \in [0, 1], \sum_{i=1}^{r} \theta_i = 1\}$
- $P(X = k | \theta_1, ..., \theta_r) = \theta_k$

Maximum likelihood estimation

Likelihood function

- $L(D:\theta) = P(D|\theta)$ = $P(X = x_1, ..., X = x_N|\theta)$
- The likelihood function measures how probable is to obtain the dataset of cases for a concrete value of the parameter θ
- Assuming that the cases are independent:

$$P(D|\theta) = \prod_{i=1}^{N} P(X = x_i|\theta) = \prod_{k=1}^{r} \theta_k^{N_k}$$

N_k = number of cases in the dataset for which $X = k$

Likelihood function

Example

	X
1	0
2	0
3	0
4	0
5	0
6	1
7	1
8	1
9	1
10	1

$$\theta = P(X = 1) = \frac{1}{4}$$
$$L(D : \tfrac{1}{4}) = P(D|\tfrac{1}{4})$$
$$= P(X = 0, ..., X = 1|\tfrac{1}{4}) = \tfrac{3}{4}^5 \tfrac{1}{4}^5$$

$$\theta = P(X = 1) = \frac{1}{2}$$
$$L(D : \tfrac{1}{2}) = P(D|\tfrac{1}{2})$$
$$= P(X = 0, ..., X = 1|\tfrac{1}{2}) = \tfrac{1}{2}^5 \tfrac{1}{2}^5$$
$$= \tfrac{1}{2}^{10} > \tfrac{3}{4}^5 \tfrac{1}{4}^5$$

Maximum likelihood estimation

Categorical distribution: relative frequencies

- $\boldsymbol{\theta}^* = (\theta_1^*, \theta_2^*, ..., \theta_r^*) = \arg max_{(\theta_1, \theta_2, ..., \theta_r)} P(D|\boldsymbol{\theta})$
- In a categorical distribution, the maximum likelihood estimator for $P(X = k)$ is:

$$\theta_k^* = \frac{N_k}{N}$$

  i.e., the relative frequency
  - In the previous example, the maximum likelihood estimator of $\theta = P(X = 1)$ is $\theta^* = \frac{5}{10}$

Bayesian estimation

Prior, posterior distributions

- $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_r)$ is assumed to be a random variable
- $f(\boldsymbol{\theta}|S) \sim Dir(a_1, ..., a_r)$ PRIOR distribution
- $\Rightarrow f(\boldsymbol{\theta}|D, S) \propto p(D|S, \boldsymbol{\theta}) f(\boldsymbol{\theta}|S) \sim Dir(a_1 + N_1, ..., a_r + N_r)$ POSTERIOR distribution
- The Bayesian estimation is the posterior mean:

$$\theta_k^* = \frac{N_k + a_k}{N + \sum_{i=1}^r a_i}$$

- $Dir(\theta_1, ..., \theta_r; a_1, ..., a_r) = \frac{\Gamma(\sum_{i=1}^r a_i)}{\prod_{i=1}^r \Gamma(a_i)} \theta_1^{a_1 - 1} ... \theta_r^{a_r - 1}$

Bayesian estimation

Many rules for estimation

- Lindstone rule, with $a_k = \lambda, \forall k$:

$$\theta_k^* = \frac{N_k + \lambda}{N + r\lambda}$$

- Laplace rule with $\lambda = 1$: $\theta_k^* = \frac{N_k+1}{N+r}$
- Jeffreys-Perks rule with $\lambda = 0.5$: $\theta_k^* = \frac{N_k+0.5}{N+\frac{r}{2}}$
- Schurmann-Grassberger rule with $\lambda = \frac{1}{r}$: $\theta_k^* = \frac{N_k+\frac{1}{r}}{N+1}$

Estimation of parameters

Parameters θ_{ijk}

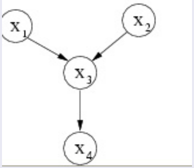
- Bayesian network structure $S = (\boldsymbol{X}, A)$ with $\boldsymbol{X} = (X_1, ..., X_n)$ and A denoting the set of arcs
- Variable X_i has r_i possible values: $x_i^1, \ldots, x_i^{r_i}$
- Local probability distribution $P(x_i \mid \boldsymbol{pa}_i^{j,S}, \boldsymbol{\theta}_i)$:

$$P(x_i^k \mid \boldsymbol{pa}_i^{j,S}, \boldsymbol{\theta}_i) = \theta_{x_i^k \mid \boldsymbol{pa}_i^j} \equiv \theta_{ijk}$$

  - The parameter θ_{ijk} represents the conditional probability of variable X_i being in its k-th value, knowing that the set of its parent variables is in its j-th value
- $\boldsymbol{pa}_i^{1,S}, \ldots, \boldsymbol{pa}_i^{q_i,S}$ denotes the values of $\boldsymbol{Pa}_i^S$, the set of parents of the variable X_i in the structure S
  - The term q_i denotes the number of possible different instances of the parent variables of X_i. Thus, $q_i = \prod_{X_g \in \boldsymbol{Pa}_i} r_g$
- The local parameters for variable X_i are given by $\boldsymbol{\theta}_i = ((\theta_{ijk})_{k=1}^{r_i})_{j=1}^{q_i})$
- Global parameters: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n)$

Maximum likelihood estimation of parameters

Parameters θ_{ijk} example



Local probabilities

$$
\begin{array}{lcll}
\boldsymbol{\theta}_1 & = & (\theta_{1-1}, \theta_{1-2}) & P(x_1^1), P(x_1^2) \\
\boldsymbol{\theta}_2 & = & (\theta_{2-1}, \theta_{2-2}, \theta_{2-3}) & P(x_2^1), P(x_2^2), P(x_2^3) \\
\boldsymbol{\theta}_3 & = & (\theta_{311}, \theta_{321}, \theta_{331}, & P(x_3^1|x_1^1, x_2^1), P(x_3^1|x_1^1, x_2^2), P(x_3^1|x_1^1, x_2^3), \\
 & & \theta_{341}, \theta_{351}, \theta_{361}, & P(x_3^1|x_1^2, x_2^1), P(x_3^1|x_1^2, x_2^2), P(x_3^1|x_1^2, x_2^3), \\
 & & \theta_{312}, \theta_{322}, \theta_{332}, & P(x_3^2|x_1^1, x_2^1), P(x_3^2|x_1^1, x_2^2), P(x_3^2|x_1^1, x_2^3), \\
 & & \theta_{342}, \theta_{352}, \theta_{362}) & P(x_3^2|x_1^2, x_2^1), P(x_3^2|x_1^2, x_2^2), P(x_3^1|x_1^2, x_2^3), \\
\boldsymbol{\theta}_4 & = & (\theta_{411}, \theta_{421}, \theta_{412}, \theta_{422}) & P(x_4^1|x_3^1), P(x_4^1|x_3^2), P(x_4^2|x_3^1), P(x_4^2|x_3^2)
\end{array}
$$

Factorisation of the joint mass probability

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2)P(x_3|x_1, x_2)P(x_4|x_3)$$

Figure: Structure, local probabilities and resulting factorization for a Bayesian network with four variables (X_1, X_3 and X_4 with two possible values, and X_2 with three possible values)

variable X_i	possible values r_i	parent variables $\boldsymbol{Pa}_i$	possible values of the parents q_i
X_1	2	$\emptyset$	0
X_2	3	$\emptyset$	0
X_3	2	$\{X_1, X_2\}$	6
X_4	2	$\{X_3\}$	2

Table: Variables (X_i), number of possible values of variables (r_i), set of variable parents of a variable ($\boldsymbol{Pa}_i$), number of possible instantiations of the parent variables (q_i)

Maximum likelihood estimation of parameters

Global independence of the parameters

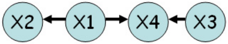
- Assuming global independence of the parameters:

$$L(D:\boldsymbol{\theta}) = \prod_{i=1}^{n} L(D_i:\boldsymbol{\theta}_i)$$

- It is possible to estimate the parameter for each variable X_i independently of the other variables

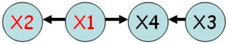
Maximum likelihood estimation of parameters

Global independence: $L(D:\theta) = \prod_{i=1}^{n} L(D_i:\theta_i)$



D

X1	X2	X4	X3
1	0	0	1
0	1	0	1
0	1	1	0
0	1	0	0
1	0	0	1



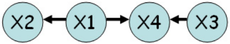
D2

X1	X2	X4	X3
1	0	0	1
0	1	0	1
0	1	1	0
0	1	0	0
1	0	0	1

Figure: Dataset D_2 for estimating the parameters of variable X_2

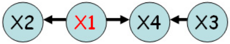
Maximum likelihood estimation of parameters

Global independence: $L(D:\theta) = \prod_{i=1}^{n} L(D_i:\theta_i)$



D

X1	X2	X4	X3
1	0	0	1
0	1	0	1
0	1	1	0
0	1	0	0
1	0	0	1



D1

X1	X2	X4	X3
1	0	0	1
0	1	0	1
0	1	1	0
0	1	0	0
1	0	0	1

Figure: Dataset D_1 for estimating the parameters of variable X_1

Maximum likelihood estimation of parameters

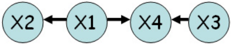
Local independence of the parameters

- Assuming local independence of the parameters:

$$L(D:\theta) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} L(D_{ij}:\theta_{ij})$$

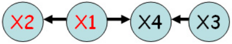
Maximum likelihood estimation of parameters

Local independence: $L(D:\theta) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} L(D_{ij}:\theta_{ij})$



D

X1	X2	X4	X3
1	0	0	1
0	1	0	1
0	1	1	0
0	1	0	0
1	0	0	1



D21=D2|X1=0

X1	X2	X4	X3
1	0	0	1
0	1	0	1
0	1	1	0
0	1	0	0
1	0	0	1

Figure: Dataset D_{21} for estimating the parameters of variable X_2 when $X_1 = 0$

Maximum likelihood estimation of parameters

$L(D:\theta) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}$

- $P(X_i = x_i^k \mid \mathbf{pa}_i^j) = \theta_{ijk}$ with $i = 1, \ldots, n; j = 1, \ldots, q_i$ and $k = 1, \ldots, r_i$
- N_{ij} number of cases in D where the configuration $\boldsymbol{pa}_i^j$ has been observed
- N_{ijk} number of cases in D where simultaneously $X_i = x_i^k$ and $\boldsymbol{Pa}_i = \boldsymbol{pa}_i^j$ has been observed ($N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$)

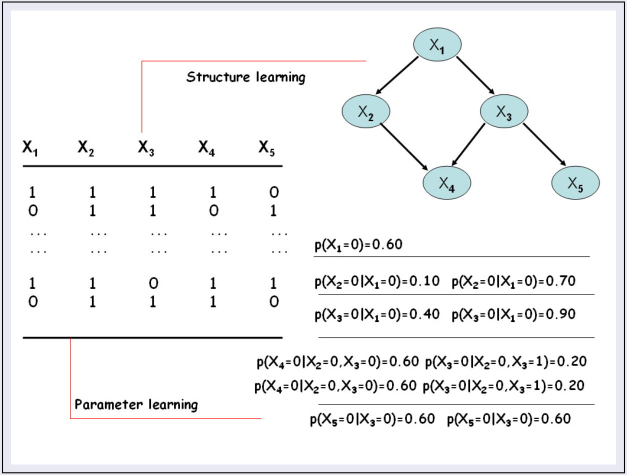
$$L(D:\theta) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}$$

Outline

1. Introduction
2. Learning Parameters
3. Learning Structures
4. Summary

Introduction

Learning structure (DAG) and parameters (conditional tables)



Introduction

Three types of methods

- Based on detecting conditional independencies (Constraint-based methods)
  1. Study the (in)dependence relationships between the variables by means of statistical tests
  2. Try to find the structure(s) that represents the most (or all) of these relationships
- Based on score + search
  - Try to find the structure that best “fit” the data
  - They need:
    1. A score (metric or evaluation function) to measure the fitness of each candidate structure
    2. A search method (heuristic) to explore in an intelligent manner the space of possible solutions
    3. Several types of spaces can be considered
- Hybrid methods
  - Based on a search technique guided by a score and the detection of conditional independencies

Testing conditional independencies

PC algorithm (Spirtes et al. 1993)

- General idea is based on generating a skeleton derived through statistical tests for detecting conditional independencies
- Start from the complete undirected graph
- Recursive conditional independence tests for deleting edges
- The output is a CPDAG where the edges should be transformed into arcs

Testing conditional independencies

Some considerations

- X_i and X_j are independent given $\mathbf{Z}$ iff $2N\,MI(X_i, X_j|\mathbf{Z}) \rightarrow \chi^2_{(r_i-1)(r_j-1)|\mathbf{Z}|}$
- The reliability of the test:
  - Increases with N, the number of cases (it is an asymptotic test)
  - Reduces dramatically with the order of the test (number of variables in $\mathbf{Z}$)

Testing conditional independencies

Completed Partially DAG (CPDAG)

- Using only conditional independence tests: not possible to obtain a unique DAG
- Usually a completed partially DAG (CPDAG) is obtained
- Each CPDAG represents an equivalent class of DAGs
- Two DAGs, S_1 and S_2 are equivalent (or Markov equivalent) if for all $\boldsymbol{W}, \boldsymbol{Y}, \boldsymbol{Z}$

$$I_{S_1}(\boldsymbol{W}, \boldsymbol{Y}|\boldsymbol{Z}) \Longleftrightarrow I_{S_2}(\boldsymbol{W}, \boldsymbol{Y}|\boldsymbol{Z})$$

- Two DAGs, S_1 and S_2 are equivalent iff they have the same edges (no direction) and the same head to head patterns (arcs $X \rightarrow Z$ and $Y \rightarrow Z$ and X and Y are not adjacent)

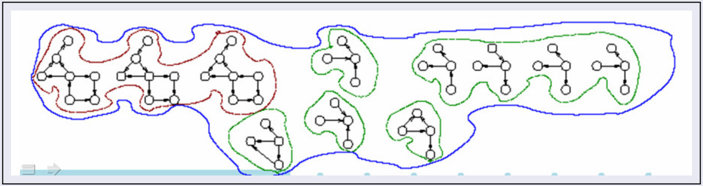
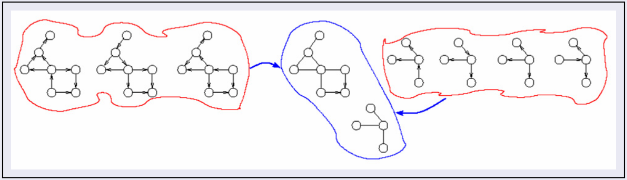
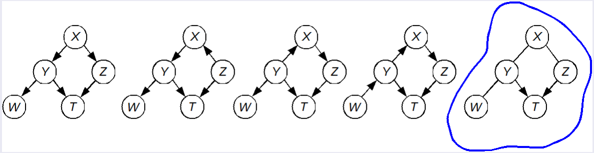


Figure: Equivalent DAGs

Testing conditional independencies

Completed Partially DAG (CPDAG)



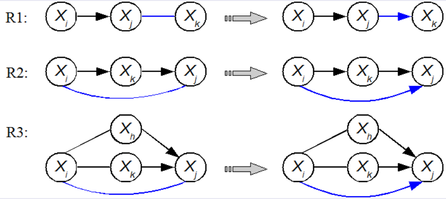


- Arcs in the CPDAG appear in all DAGs of its equivalent class
- Edges in the CPDAG can be orientated in different ways in each DAG of its class (without new head to head patterns or cycles)

Testing conditional independencies

PC algorithm (Spirtes et al. 1993)

Form complete, undirected graph S
$t = -1$
repeat
    $t = t + 1$
    repeat
        select ordered pair of adjacent nodes A, B in S
        select neighborhood C of A of size t (if possible)
        delete edge $A - B$ in S if A and B cond. ind. given C
    until all ordered pairs have been tested
until all neighborhood are of size smaller than t
Transform edges in arcs by applying some simple rules



Testing conditional independencies

PC algorithm (Spirtes et al. 1993). Example with $t = 2$

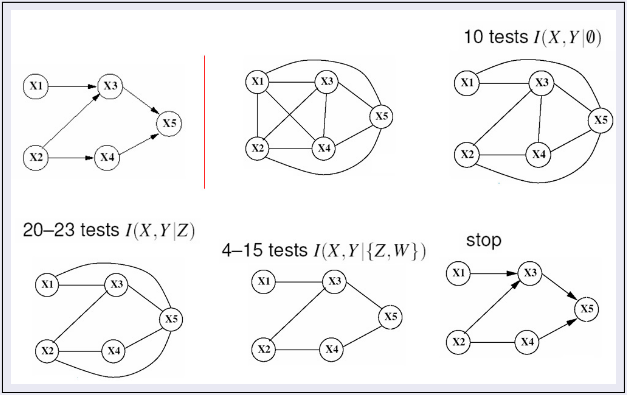


Figure: Example of the PC algorithm with $t = 2$

Score+search approaches

Introduction

- They try to find the structure that best “fit” the data
- They are characterized by:
  - A score (metric or evaluation function) to measure the fitness of each candidate structure
    - Penalized log-likelihood
    - Bayesian metrics
  - A space of structures where the search is carried out
    - Directed acyclic graphs
    - Equivalence classes
    - Order between the variables
  - A search method to explore in an intelligent manner the space of possible solutions
    - Local search
    - Heuristics

Score+search approaches

Score metrics. Penalized log-likelihood

- Log-likelihood of the data: $\log P(D : S, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \log(\theta_{ijk})^{N_{ijk}}$
- $\log P(D : S, \widehat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}}$
  with $\widehat{\theta}_{ijk} = \frac{N_{ijk}}{N_{ij}}$ (maximum likelihood estimate)

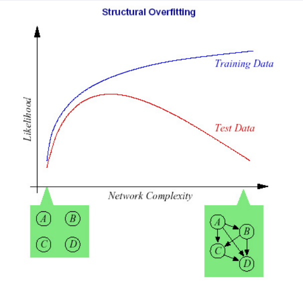


Figure: Likelihood increases monotonically wrt model complexity

Score+search approaches

Score metrics. Penalized log-likelihood

- Avoid overfitting penalizing the complexity of the Bayesian network in the log-likelihood :

$$\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}} - dim(S)pen(N)$$

  - $dim(S) = \sum_{i=1}^{n} q_i(r_i - 1)$ model dimension
  - $pen(N)$ no negative penalization function
    - $pen(N) = 1$: Akaike's information criterion (AIC) (Akaike, 1974)
    - $pen(N) = \frac{1}{2} \log N$: Bayesian information criterion (BIC) (Schwarz, 1978). It is equivalent to the minimum description lenght (MDL) (Lam and Bacchus, 1994) criterion

Score+search approaches

Score metrics. Bayesian model selection

- Try to obtain the structure with maximum a posteriori probability given the data: that is arg $max_S P(S|D)$
- Using Bayes formula:

$$P(S|D) = \frac{P(D|S)P(S)}{P(D)}$$

$$P(S|D) \propto P(D|S)P(S)$$

- $P(D|S)$ is the marginal likelihood of the data
- $P(S)$ denotes the prior distribution over structures
- If $P(S)$ is uniform ($maxP(S|D) \equiv maxP(D|S)$) we try to obtain the structure with maximum marginal likelihood

Score+search approaches

Score metrics. Bayesian model selection. K2 metric

- Accounts for uncertainty also in the parameters:

$$P(D|S) = \int P(D|S,\theta)p(\theta|S)d\theta$$

  - $P(D|S)$ posterior probability of the data given the structure
  - $P(D|S,\theta)$ likelihood of the data given the Bayesian network (structure + parameters)
  - $p(\theta|S)$ prior distribution over the parameters

Score+search approaches

Score metrics. Bayesian model selection. K2 metric

- Assuming that $p(\theta_{ij}|S)$ is uniform, it is possible to obtain a closed formula for $P(D|S)$ (Cooper and Herskovits, 1992)

$$P(D|S) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!$$

  - n: number of variables
  - r_i: number of states X_i can have
  - q_i: number of possible state combinations of $\boldsymbol{Pa}_i$
  - N_{ijk}: number of cases in D where X_i takes its k-th value and the parent set of X_i are on their j-th combination of values
  - N_{ij}: $\sum_{k=1}^{r_i} N_{ijk}$

Score+search approaches

Score metrics. Bayesian model selection. K2 algorithm

- An ordering between the nodes is assumed
- An upper bound is set on the number of parents for any node
- For every node, X_i, K2 searches for the set of parent nodes that maximizes:

$$g(X_i, \boldsymbol{Pa}_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!$$

- K2 assumes initially that a node does not have parents
- At each step K2 incrementally adds the parent whose addition provides the best value for $g(X_i, \boldsymbol{Pa}_i)$
- K2 stops when adding a single parent to any node cannot increase $g(X_i, \boldsymbol{Pa}_i)$
- K2 is a greedy algorithm

Score+search approaches

Score metrics. Bayesian model selection. BDe metric

- Assuming that $p(\boldsymbol{\theta}_{ij}|S)$ follows a Dirichlet distribution, it is possible to obtain a closed formula for $P(D|S)$

$$P(D|S) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

- This is called the Bayesian Dirichlet (BD) score
- α_{ijk} denotes the parameters of the Dirichlet distribution
    - $\alpha_{ijk} = 1$: K2 metric (Cooper and Herskovits, 1992)
    - $\alpha_{ijk} = \alpha P(x_i^k, \mathbf{Pa}_i = \mathbf{pa}_i^j|S)$: likelihood-equivalent Bayesian Dirichlet (BDe) score (Heckerman et al., 1995)
    - $\alpha_{ijk} = \alpha / q_i r_i$: BDeu score (Buntine, 1991)
- Decomposable score = can be expressed as a sum of values that depend on only one node and its parents. All (estimated log-likelihood, AIC, BIC/MDL, BD, K2, BDe and BDeu)
- Score equivalence property = two Markov equivalent graphs score the same. All but K2 and BD are score equivalent

Score+search approaches

Different spaces for search

- Space of directed acyclic graphs

$$d(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} d(n-i); \quad d(0) = 1; \quad d(1) = 1$$

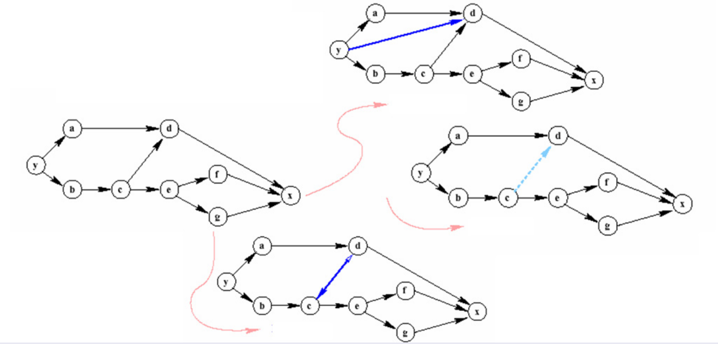
  E.g., $d(10) \simeq 4.2 \times 10^{18}$

- Space of equivalence classes (each class reflects the same set of conditional independencies)
  - Scores: score equivalent (Chickering, 1996)
- Ordering between the variables (Larrañaga et al., 1996, Friedman and Koller, 2002): cardinality of the search space $n!$

Score+search approaches

Search algorithms. Local search. B algorithm (Buntine, 1991)

- Local operators: insert, delete and invert an arc
- Efficient search due to the decomposability of the scores



Score+search approaches

Search algorithms. Metaheuristics and exact methods

- Greedy search (Buntine, 1991; Cooper and Herskovits, 1992), simulated annealing (Heckerman et al., 1995), genetic algorithms (Larrañaga et al., 1996), MCMC methods (Giudici and Green, 1999; Friedman and Koller, 2003; Grzegorczyk and Husmeier, 2008) and estimation of distribution algorithms (Larrañaga et al., 2000; Blanco et al., 2003)
- Exact methods (several dozens of variables only): dynamic programming (Koivisto and Sood, 2004; Silander and Myllymäki, 2006; Malone et al., 2011), branch and bound (de Campos and Ji, 2011), and mathematical programming (Martínez-Rodríguez et al., 2008; Jaakkola et al., 2010)

Outline

1. Introduction
2. Learning Parameters
3. Learning Structures
4. Summary

Learning Bayesian networks

Structure + parameters

- Learning parameters
  - Maximum likelihood estimation
  - Bayesian estimation (Dirichlet distribution)
- Learning structures
  - Detecting conditional independencies (PC algorithm)
  - Score + search (penalized log-likelihood (AIC, BIC, MDL), Bayesian metrics (K2, BD, BDe, BDeu); local, metaheuristics)
  - Hybrid methods