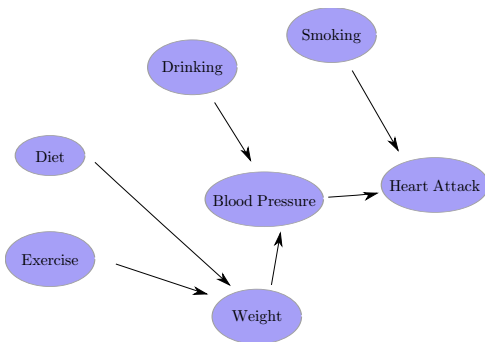


Bayesian networks

- It is a formalism used to represent dependence/independence relationships among attributes as a DAG
- Each node stores the **joint probability distribution** of the variable and its parents in the graph
- From this and using the topology of the network we can infer the probability distribution of the values of any attribute given the values of any subset of attributes in the network

Notes

Bayesian networks



Notes

What information they represent?

- The structure of a bayesian network represents:
 - The *joint probability distribution* of the data
 - How to factorize the joint probability distribution into independent components to reduce the cost of probability estimation
- From the bayesian network we can obtain
 - The probability of the values of an attribute given a subset of attributes
 - The set of attributes that have direct influence over an attribute

Notes

The Machine Learning perspective

Bayesian Networks

- Allow to study **attribute relationships**
- Allow to discover the patterns of the relationships among attributes
- Allow to **infer** the values of an unknown attribute using their dependencies
- Can be used to discover the attributes relevance in classification tasks
- Can be used as a **classifier** (generalized version of naive bayes)

Notes

Learning of Bayesian networks

- It is difficult to build a Bayesian Network (expert knowledge)
- The topology and the probabilities distribution can be learned from a dataset
- We want to learn the best network topology (DAG) that fits the data

Notes

Learning of Bayesian networks

- We can define the problem as a search problem
 - Search space: All possible DAGs that can be defined with the variables

$$f(n) = \sum_{i=1}^n (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i) \text{ with } f(0) = 1 \text{ and } f(1) = 1$$

$$f(14) \sim 1,8 \times 10^{31}$$

- Search operators: Possible modifications to a DAG
- Heuristic function: We want to obtain the most simple BN that explains the probability distribution of the data

Notes

Bayesian networks evaluation

Three components:

- **A priori information:** Prior information that bias the search to specific types of networks, (for example: Prior probabilities, partial order among nodes, ...)
- **Information from the dataset:** Joint probability estimation for a given topology, adequacy of the estimation to the actual data
- **Network complexity:** Value that allows to bias the search toward simpler networks, penalties over the number of connections or estimation parameters

Notes

Quality functions

- There are multiple goal functions that can be used to evaluate bayesian networks using different criteria:
 - Bayesian estimation (like naive bayes/EM algorithm)
 - Minimum Description Length (DAG that best compresses the dataset)
 - Information theory (like decision trees)
- The estimation methods differ on the types of the attributes: Bayesian networks with discrete attributes (multinomial) or continuous (multinormal)

Notes

Quality functions (multinomial networks)

- Given a dataset, this represents multiple instantiations of a random variable $X = (X_1, X_2, \dots, X_n)$
- Each random variable X_i has a multinomial distribution that has r_i values

Variable	Values
X_1	{a,b,c}
X_2	{0,1}
X_3	{a,b,c,d}
...	
X_n	{a,b}

Notes

Quality functions (multinomial networks)

- Π_i is the set of parents of the variable X_i (variables that have influence over X_i), for example:

$$\Pi_1 = \{X_2, X_3\}$$

- We define s_i as the number of possible combinations of the values of the variables Π_i (r_j number of values of variable i)

$$s_i = \prod_{X_j \in \Pi_i} r_j$$

for example:

$$s_1 = r_2 \times r_3 = 2 \times 4 = 8$$

Notes

Quality functions (multinomial networks)

- We define N_{ijk} as the number of cases in the dataset where the variable $X_i = j$ (j is the j -th value of the variable) and the parents of X_i have the values of the k -th combination of Π_i
- We define $N_{ik} = \sum_{j=0}^{r_i} N_{ijk}$ as the number of cases in the dataset where the parents of X_i have the values of the k -th combination of Π_i independently of the value of X_i
- We can estimate the similarity of this two probability distributions as:

$$\sum_{j=1}^{r_i} \sum_{k=1}^{s_i} N_{ijk} \log \frac{N_{ijk}}{N_{ik}}$$

Notes

Quality functions (multinomial networks)

A	B	C	f(A,B)	f(A,B,C)
0	0	0	0.5	0.4
0	0	1	0.5	0.1
0	1	0	0.2	0
0	1	1	0.2	0.2
1	0	0	0.1	0.05
1	0	1	0.1	0.05
1	1	0	0.2	0.15
1	1	1	0.2	0.05

$$A, B \rightarrow C? \Rightarrow \sum_{j=1}^{r_C} \sum_{k=1}^{s_C} N_{Cjk} \log \frac{N_{Cjk}}{N_{Ck}}$$

Notes

Quality functions (multinomial networks)

- Summing this values for all the variables of the network gives an estimation about how well the topology of the network fits the joint probability distribution of the data

$$Q_l(B) = \sum_{i=1}^n \left(\sum_{j=1}^{r_i} \sum_{k=1}^{s_j} N_{ijk} \log \frac{N_{ijk}}{N_{ik}} \right)$$

Notes

Quality functions (multinomial networks)

- The apriori probability of the network $p(D)$
 - This probability can be described explicitly as probability over the network topologies
 - Bias information can be introduced in the search (order of exploration of the variables, constraints about possible parents of a variable, maximum number of parents, ...)

Notes

Quality functions (multinomial networks)

- The penalty for the complexity of the network (the simpler the better)
 - This penalty is usually a function of the number of parameters to estimate the probabilities and the size of the dataset
 - There are different criterion
 - Maximum likelihood information criteria:** No penalization for network complexity
 - Akaike information criteria:** The penalty is the number of parameters to estimate the network probabilities
 - Minimum Description length criteria:** The penalty is the product of the number of parameters and the logarithm of the size of the dataset

Notes

Algorithm K-2

- Hill climbing search
- The initial state is an empty network
- We assume that the precedence order of the nodes are known
- We evaluate all possible parents for a node (Only those that precede it in the order)
- We define $q_i(\Pi_i)$ as the evaluation function restricted to the nodes that are parents of the variable X_i
- We add all nodes that improve this function

Notes

Algorithm K-2

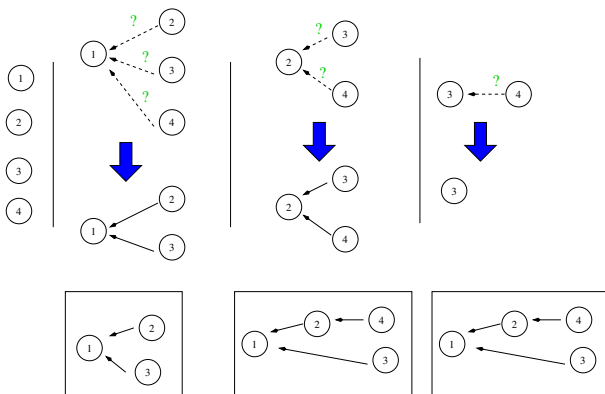
Algorithm: K-2

Order the variables

for $i=1$ **to** n **do** $\Pi_i = \emptyset$ **for** $i=1$ **to** n **do** **repeat** Select $Y \in \{X_1, \dots, X_{i-1}\} \setminus \Pi_i$ that maximizes $g = q_i(\Pi_i \cup \{Y\})$ $\delta = g - q_i(\Pi_i)$ **if** $\delta > 0$ **then** $\Pi_i = \Pi_i \cup \{Y\}$ **end** **until** $\delta \leq 0$ **o** $\Pi_i = \{X_1, \dots, X_{i-1}\}$ **end**

Notes

Algorithm K-2



Notes
