Imperial College
London

NOTES

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

493 Data Analysis and Probabilistic Inference

Author:
(CID:)

Date: February 3, 2017

1 Bayes Theorem and Bayesian Inference

1. Bayes Theorem:

$$P(D \cap S) = P(D|S)P(S) = P(S|D)P(D)$$
$$P(D|S) = \alpha \times P(D) \times P(S|D)$$

2. Law of Total Probability:

$$P(F) = \sum_{i=1}^{n} P(F \cap E_i) = \sum_{i=1}^{n} P(F|E_i)P(E_i)$$

3. Conditional Independence: Two events D and S are conditionally independent given G if $P(G) \neq 0$ and one of the following holds:
   (a) $P(D|S \cap G) = P(D|G)$ and $P(D|G) \neq 0, P(S|G) \neq 0$
   (b) $P(D|G) = 0$ or $P(S|G) = 0$
   (c) $P(D \cap S|G) = P(D|G)P(S|G)$

4. Bayesian Inference: Given a set of competing hypothesis which explain a data set, for each hypothesis:
   (a) Convert the prior and likelihood information in the data into probabilities and take their product
   (b) Normalize the result to get the posterior probabilities of each hypothesis given the evidence
   (c) Select the most probably hypothesis

2 Simple Bayesian Networks

1. We have to assume the Causal Markov Condition has the following difficulties inherent in large instances

   (a) The joint probabilities are hard to estimates

   (b) Even if the joint probabilities can be obtained, there are too many number of instances

2. **Causal Markov Condition:** Suppose we have a joint probability distribution P of the random variables in some set $\mathcal{V}$ and a DAG $\mathbb{G} = (\mathcal{V}, E)$. We say that $(\mathbb{G}, P)$ satisfies the Markov condition if for each variable $X \in \mathcal{V}$, $\{X\}$ is conditional independent of **the set of all its nondescendents given the set of all its parents**. Let ND_X be the non-descendents and PA_X be the parents of X.

$$I_P(\{X\}, ND_X | PA_X)$$

3. Possible violations of the Causal Markov Condition:

   (a) Hidden cause: X and Y is said to have a common cause if there exists some variable that has causal paths into both of them. If we fail to model this common cause, in short (exists a hidden cause), the Markov condition would be violated as it assumes independence.

   (b) Selection bias: It is similar to hidden cause. The variables we observe shows independence when actually because of our sampling error.

   (c) Feedback loops: Causal relationships need to be only one way. The child node under no circumstances should influence the parent node.

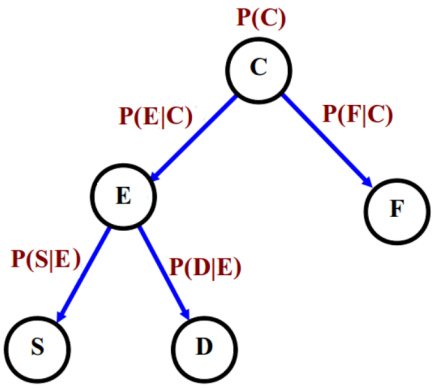


Figure 1: Example of naive bayes network. Given the parent C, the node E and F meet the causal markov condition, i.e. they are conditionally independent.

4. Each arc in a simple network is represented by a link matrix (conditional probability matrix)

$$\boldsymbol{P}(\boldsymbol{D}|\boldsymbol{C}) = \Big[P(d_i|c_j)\Big] = \begin{bmatrix} P(d_1|c_1) & P(d_1|c_2) \\ P(d_2|c_1) & P(d_2|c_2) \\ P(d_3|c_1) & P(d_3|c_2) \\ P(d_4|c_1) & P(d_4|c_2) \end{bmatrix}$$

5. The root nodes of a network do not have any parents. They have a vector giving the prior probabilities

$$\boldsymbol{P}(\boldsymbol{C}) = [P(c_i)] = \Big[P(c_1) \quad P(c_2)\Big]$$

6. **Instantiation** means setting the value of a node.

7. **Bayesian Classifiers** using the above network.

   (a) We cannot compute E as it is a latent variable that we compute and we do not have measurements. However, we can computed the likelihood of E.

$$\begin{aligned} P(E|S \cap D) &= \alpha P(E)P(S|E)P(D|E) \qquad &&= \alpha P(E)L(E|S \cap D) \\ L(E|S \cap D) &= (S|E)P(D|E) \end{aligned}$$

   (b) Then we look at the root note C. Given $F = f_5$:

$$P(C|E \cap F) = \alpha P(C)P(E|C)P(F|C)$$

$$P(e|c_k) = \sum_{i=1}^{3} P(e_i|c_k)L(e_i)$$

$$P'(c_k) = P(c_k|e \cap f_5) = \alpha P(c_k)\left(\sum_{i=1}^{3} P(e_i|c_k)L(e_i)\right)P(f_5|c_k)$$

   (c) We see the the evidence for C comes from
      i. Evidence coming from E and its sub-tree
      ii. Evidence from everywhere else.

$$\begin{aligned} P_E(C) &= \alpha P(C)P(F|C) \\ \boldsymbol{P}(\boldsymbol{E}) &= \boldsymbol{P}(\boldsymbol{E}|\boldsymbol{C})\boldsymbol{P}_{\boldsymbol{E}}(\boldsymbol{C}) \end{aligned}$$

   (d) Suppose if we have the instantiations $S = s_3$ and $D = d_2$

$$P'(e_i) = \alpha P(e_i)P(s_3|e_i)P(d_2|e_i)$$

3 Evidence and Message Passing: Pearl's Algorithm

1. New concepts to deal with complex networks with intermediate nodes:
   - **Evidence** is the information that we have at a node -this may be gathered through instantiation (exact value or virtual evidence), or inferred from passing messages. Evidence is unnormalized probabilities so the absolute values are meaningless, but they are useful for making comparisons.
   - **Messages** is the information (evidence) passed between nodes to provide evidence to another node.
2. **Theorem:** Let $(\mathbb{G}, P)$ be a Bayesian network whose DAG is a tree, where $\mathbb{G} = (V, E)$, and a be a set of values of a subset $A \subset V$.
   (a) λ **messages**: For each child Y of X, $\forall x \in X$
   $$\lambda_Y(x) = \sum_y P(y|x)\lambda(y)$$
   (b) λ **values**:
      i. If $X \in A$ and X is instantiated to $\hat{x}$
      $$\lambda(\hat{x}) = 1, \text{ for } x = \hat{x}$$
      $$\lambda(x) = 0, \text{ for } x \neq \hat{x}$$
      ii. if $X \notin A$ and X is a leaf, $\forall x \in X$,
      $$\lambda(x) = 1$$
      iii. If $X \notin A$ and X is not a leaf, $\forall x \in X$
      $$\lambda(x) = \prod_{U \in CH_X} \lambda_U(x),$$
      where CH_X denotes the set of the children of X.
   (c) π **messages**: If Z is the parent of X, then $\forall z \in Z$
   $$\pi_X(z) = \pi(z) \prod_{U \in CH_Z - \{X\}} \lambda_U(z)$$
   (d) π **values**:
      i. If $X \in A$ and X is instantiated to $\hat{x}$:
      $$\pi(\hat{x}) = 1, \text{ for } x = \hat{x}$$
      $$\pi(x) = 0, \text{ for } x \neq \hat{x}$$
      ii. If $X \notin A$ and X is the root, $\forall x \in X$
      $$\pi(X) = P(x)$$

iii. If $X \notin A$, X is not the root and Z is the parent of X, $\forall x \; in X$

$$\pi(x) = \sum_z P(x|z)\pi_X(z)$$

(e) Given the definitions, for each variable X, we have for all values of x,

$$P(x|a) = \alpha\lambda(x)\pi(x)$$

3. The π values are basically evidence from the parents and it generalises the concept of prior. The λ values are basically evidence from the descendents and it generalises the concept of likelihood probability.

4. Mnemonic: **pi** (π), **p**rior, and "**p**arent" all start with letter "p"; and **l**ambda (λ), **l**ikelihood, and "**l**ad" all start with "l". Further, prior comes *before* so from parents.

5. If we use virtual evidence at the leaf nodes, we can use the conditioning equation (above b(iii))

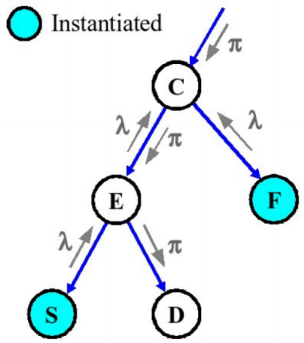


Figure 2: Upon instantiation of a node, we can propagate the λ and π messages

6. Important equations:

$$\boldsymbol{\lambda_S(E) = \lambda(S)P(S|E)} \qquad \boldsymbol{\pi(E) = P(E|C)\pi_E(C)}$$

$$\lambda(e_i) = \prod_{CH_E}\left[\sum_j \lambda(s_j)P(s_j|e_i)\right] \qquad \pi(e_i) = \sum_j\left[P(e_i|c_j)\pi(c_j)\prod_{k\setminus E}\lambda_k(c_j)\right]$$

7. Notation

   (a) $\boldsymbol{\lambda_F(c)}$ means λ evidence from node F to node C.

   (b) $\boldsymbol{\pi_F(c)}$ means π from all nodes besides F for node C. (This can alternatively viewed as the information from C to F)

4 Probability Propagation: Single Connected Networks

1. A DAG is singly connected if there is at most one path between any two nodes. In a singly-connected network, each node can have more than 1 parents.

2. The main properties of these networks are:

   (a) The parents of a node are always independent given their common child (i.e. they don't have a common parent). This lets us calculate their joint probability as the product of their marginals.

   (b) When updating evidence of a node, the belief propagated through the net to update all nodes is guaranteed to reach a steady state.

3. An example of a link matrix for a node with multiple parents looks as follows:

$$\boldsymbol{P(e|w,c)} = \begin{bmatrix} P(e_1|w_1,c_1) & P(e_1|w_1,c_2) & P(e_1|w_2,c_1) & P(e_1|w_2,c_2) \\ (e_2|w_1,c_1) & P(e_2|w_1,c_2) & P(e_2|w_2,c_1) & P(e_2|w_2,c_2) \\ P(e_3|w_1,c_1) & P(e_3|w_1,c_2) & P(e_3|w_2,c_1) & P(e_3|w_2,c_2) \end{bmatrix}$$

4. To calculate the π evidence of a node with 2 parents (assuming independence between the parents):

$$\pi(\boldsymbol{E}) = \boldsymbol{P(e|w,c)\pi_e(w,c)} = \boldsymbol{P(e|w,c)\pi_e(w)\pi_e(c)}$$

5. To calculate the λ evidence of a node with 2 parents, we have to calculate one λ message for each of the parents. If c has parents a and b, then the evidence from c to a is as follows:

$$\lambda_c(a_i) = \sum_{j=1}^{n} \pi_c(b_j) \sum_{k=1}^{m} P(c_k|a_i,b)\lambda(c_k)$$

   where n is the number of values that b takes, and m is the number of values c takes.

6. **Theorem:** Let $(\mathbb{G}, P)$ be a Bayesian network that is singly-connected, where $\mathbb{G} = (V, E)$, and a be a set of values of a subset $A \subset V$.

   (a) λ **messages**: For each child Y of X, $\forall x \in X$

$$\lambda_Y(x) = \sum_{y} \left[\sum_{w_1,\ldots,w_k} \left(P(y|x,w_i,\ldots,w_k) \prod_{i=1}^{k} \pi_Y(w_i) \right) \right] \lambda(y)$$

   where $\{W_i\}_{i=1}^{k}$ are the other parents of Y.

   (b) λ **values**:

      i. If $X \in A$ and X is instantiated to $\hat{x}$

$$\lambda(\hat{x}) = 1, \text{ for } x = \hat{x}$$
$$\lambda(x) = 0, \text{ for } x \neq \hat{x}$$

ii. if $X \notin A$ and X is a leaf, $\forall x \in X$,

$$\lambda(x) = 1$$

iii. If $X \notin A$ and X is not a leaf, $\forall x \in X$

$$\lambda(x) = \prod_{U \in CH_X} \lambda_U(x),$$

where CH_X denotes the set of the children of X.

(c) π **messages**: If Z is the parent of X, then $\forall z \in Z$

$$\pi_X(z) = \pi(z) \prod_{U \in CH_Z - \{X\}} \lambda_U(z)$$

(d) π **values**:

i. If $X \in A$ and X is instantiated to $\hat{x}$:

$$\pi(\hat{x}) = 1, \text{ for } x = \hat{x}$$
$$\pi(x) = 0, \text{ for } x \neq \hat{x}$$

ii. If $X \notin A$ and X is the root, $\forall x \in X$

$$\pi(X) = P(x)$$

iii. If $X \notin A$, X is not the root and $\{Z_i\}_{i=1}^{j}$ are the parents of X, $\forall x \; in X$

$$\pi(x) = \sum_{z_1,\ldots,z_j} \left(P(x|z_1,\ldots,z_j) \prod_{i=1}^{j} \pi_X(z_i) \right)$$

(e) Given the definitions, for each variable X, we have for all values of x,

$$P(x|a) = \alpha \lambda(x) \pi(x)$$

7. **The Operating Equations for Probability Propagation**

(a) The λ Message

$$\lambda_C(a_i) = \sum_{j=1}^{m} \pi_C(b_j) \sum_{k=1}^{n} P(c_k|a_i \cap b_j) \lambda(c_k)$$

$$\boldsymbol{\lambda_C(A) = \lambda(C) P(C|A)}$$

$$\lambda_C(a_i) = \sum_{j=1}^{m} \pi_C(b_j) \lambda_C(a_i \cap b_j)$$

(b) The π Message: If C is a child of A, the π message from A to C is:

$$\pi_C(a_i) = \begin{cases} 1 & \text{if } A \text{ is instantiated for } a_i \\ 0 & \text{if } A \text{ is instantiated but not for } a_i \\ P'(a_i)/\lambda_C(a_i) & \text{if } A \text{ is not instantiated} \end{cases}$$

(c) The λ Evidence: If C is a node with n children $D_1,\ldots,D_n$, then the λ evidence for C is

$$\lambda(c_k) = \begin{cases} 1 & \text{if } C \text{ is instantiated for } c_k \\ 0 & \text{if } C \text{ is instantiated but not for } c_k \\ \prod_i \lambda_{D_i}(c_k) & \text{if } C \text{ is not instantiated} \end{cases}$$

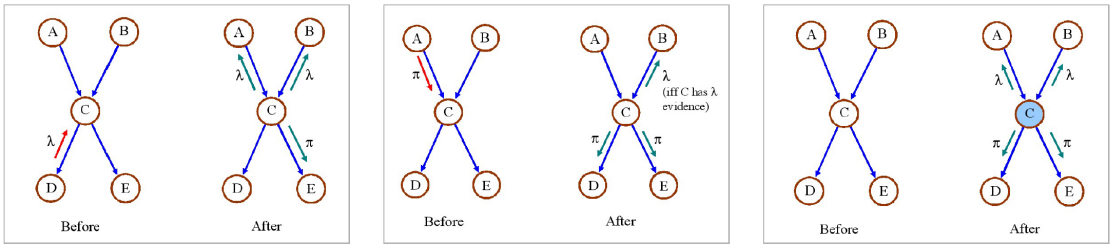
(d) The π Evidence: If C is a child of two parents A and B, the pi evidence for C is:

$$\pi(c_k) = \sum_{i=1}^{l}\sum_{j=1}^{m} P(c_k|a_i \cap b_j)\pi_C(a_i)\pi_C(b_i)$$

$$\boldsymbol{\pi(C) = P(C|A)\pi_C(A)}$$

(e) The Posterior Probabilities: If C is a variable, the posterior probability of C based on the evidence received is

$$P'(c_k) = \alpha\lambda(c_k)\pi(c_k)$$



(a) Upward Propagation (b) Downward propagation (c) Instantiation

Figure 3: Probability propagation

8. Message Passing

(a) **Initialization:** For all the root nodes, the π values are set to the prior probabilities and propagate the π messages using downward propagation

(b) **Upward Propagation (only for uninstantiated nodes):** A node C receives λ from a child.

- Compute the new $\lambda(C)$ and $P'(C)$. Then, post a λ message to all its parents and post a π message to all C's other children.

(c) **Downward Propagation:** If a variable C receives a π message from one parent.
- If C is not instantiated: Compute $\pi(C)$ and $P'(C)$ and post π message to each child.
- If there is evidence in C: Post λ message to the other parents

(d) **Instantiation:** If C is instantiated for state c_k,
- $P'(c_k) = 1$ and $P'(c_j) = 0 \forall j \neq k$. Compute $\lambda(C)$
- Post λ and π message to each parent and each child of C respectively.

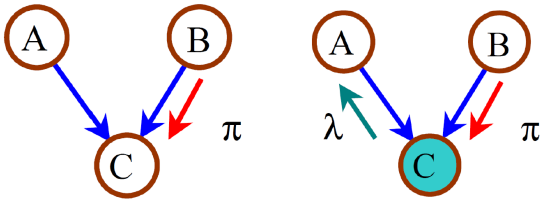


Figure 4: Converging Connections

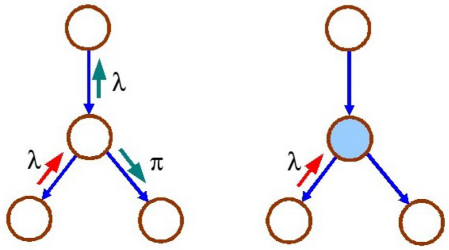


Figure 5: Diverging Connection

9. **Blocked Path (very important to understand this):**
- For a diverging path: when the node is instantiated, it will block the passing message between other nodes.
- For a converging path: it is blocked when there is no λ evidence on the child node, but unblocked when there is λ evidence or when the node is instantiated.

$$\lambda_C(a_i) = \sum_{j=1}^{m} \pi_C(b_j) \sum_{k=1}^{n} P(c_k | a_i \cap b_j) \lambda(c_k)$$

$$\lambda(c_k) = 1, \forall c_k \Rightarrow \lambda_C(a_i) = \sum_{j=1}^{m} \pi_C(b_j)$$

5 Building Networks from Data

1. Building networks from data:
   (a) **Expert knowledge**: We can obtained the network structure would be known or readily obtainable from an expert. This approach is highly subjective and reliant on the expert advice.
   (b) **Spanning tree algorithms**: Main idea is to start with a set of nodes to which we add arcs until a complete network is created.
      i. The nodes that are joined by arcs are depedent and those not are at least conditionally independent. Hence the strategy is to add arcs that connect variables that are most dependent.
      ii. For every pair of variables calculate the dependency. Then join the nodes in dependency order, provided the resulting structure has no loops
2. **Adding Causal Directions:**
   (a) If a node is considered a root node, we will point all arrows from it.
   (b) On the whole, cause is a semantic entity that needs human intervention to determine.
3. **Measures of dependencies**
   - $L1$ **dependency measure**:

$$Dep(A,B) = \sum_{A\times B} \left| P(a_i \cap b_j) - P(a_i)P(b_j) \right|$$
$$Dep(A,B) = \sum_{A\times B} P(a_i \cap b_j) \times \left| P(a_i \cap b_j) - P(a_i)P(b_j) \right|$$

   - $L2$ **dependency measure**:

$$Dep(A,B) = \sum_{A\times B} \Big(P(a_i \cap b_j) - P(a_i)P(b_j) \Big)^2$$
$$Dep(A,B) = \sum_{A\times B} P(a_i \cap b_j) \times \Big(P(a_i \cap b_j) - P(a_i)P(b_j) \Big)^2$$

     As the probabilities becomes small they contribute less to the dependency, and this effect is acceptable since we have little information on rare events. The weighted version of the $L1$ and $L2$ further reduces the dependency for low probability values.
   - **Kullback-Leibler Measure (mutual entropy)**
     – It is zero when two variables are completely independent
     – It is positive and increasing with dependency when applied to probability distributions

  - It is independent of the actual value of probability
  - Can be computed as follows

$$Dep(A,B) = \sum_{A\times B} P(a_i \cap b_j) \log_2 \left[\frac{P(a_i \cap b_j)}{P(a_i)P(b_j)} \right]$$

- **Correlation**
  - Measures only linear dependency whereas mutual entropy can characterise higher order dependencies more accurately.
  - Can be computed as follows:

$$C(A,B) = \frac{\Sigma_{AB}}{\sqrt{\sigma_A \sigma_B}}$$
$$\Sigma_{AB} = \frac{1}{N-1} \sum_{i=1}^{N} (a_i - \bar{a}_i)(b_i - \bar{b}_i)$$
$$\sigma_A = \frac{1}{N-1} \sum_{i=1}^{N} (a_i - \bar{a}_i)^2$$

6 Cause and Independence

1. Independence: In a network, if there is no path between two variables, they are independent.

   - Two variables can be connected by a path in a network and still be independent as the conditional probability can express no dependency.
   - In theory, the absence of an arc is more significant than the presence of an arc. However, in theory we avoid arcs with very low dependency in networks.
   - The spanning tree algorithm can be terminated when the dependecy becomes too low to be significant.

2. (d-separation) Let $\mathbb{G} = (V, E)$ be a DAG, $A \subseteq V$ and X and Y be distinct nodes in $V - A$. X and Y are d-separated by A in $\mathbb{G}$ if every chain between X and Y is blocked by A.

3. Possible configurations of connected triplets:

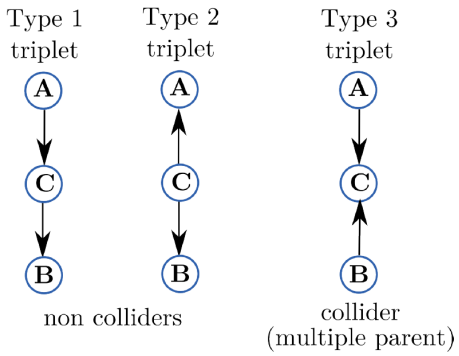


Figure 6: Possible configurations of connected triplets.

(a) **Non-colliders**: If C is instantiated, no messages pass from A to B.

(b) **Colliders:** The nodes A and B are independent if there is no information on C.

- Marginal Independence: We can measure the dependence between A and B using all the data. If this is low, the configuration may be multiple parent.
- Conditional Independence: We can partition data according to states of C and compute the dependency between A and B. If this is high, the configuration may be multiple parent.

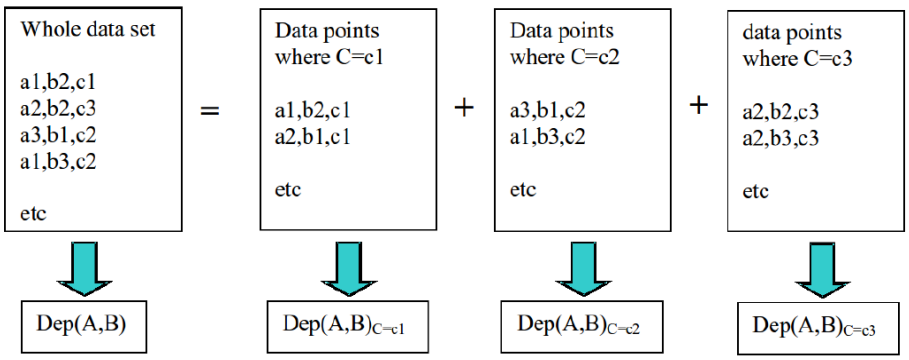


Figure 7: Practical computation.

4. **Determining Causal Directions:**

(a) Compute the maximally weighted spanning tree

(b) For each connected triplet in the spanning tree

- Compute the joint probability of the triplet
- Compute the marginal dependence and condition dependence
- If marginal dependence is low and the conditional dependence is high, put in causal directions corresponding to a collider

(c) Propagate the causal arrows as far as possible

5. Structure and Parameter Learning

   - Bayesian networks
     - Combined both structure and parameter learning.
     - We can express our knowledge by choosing the structure or we can learn it through data.
     - Incorporate known causal relations but have the potential to learn cause from data.
   - Neural networks
     - Offers only parameter learning.
     - It is difficult to embed knowledge or infer structure from a neural network.
     - Most applicable when we have no causal knowledge but we cannot extract causal information from them.
   - Rule based system (logic based)
     - Have just structure but they are not scalable and difficult to optimise.
     - Most applicable when we have causal knowledge as these systems can represent it well

7 Model Accuracy

1. The model may be evaluated using **maximum likelihood estimation**.

$$\begin{aligned} P(Ds|Bn) &= \prod_{data} P(Bn) \\ &= \prod_{i} P(x_i)P(y_i|x_i) \text{ (case of two variables)} \end{aligned}$$

   - Above is the general formulation, but to avoid underflow errors, one may take the log likelihood. If the log we take is base 2, then the log likelihood is known as the information measure since $\log_2$ N bits are required to represent integers up to N.

$$\log(P(Ds|Bn)) = \sum_{data} log(P(Bn))$$

   - Ds is a given dataset, Bn is a our BayesNet model, and $\prod_{data} P(Bn)$ is the joint probability of the model multiplied over all values in the dataset.
   - Larger models get higher scores, so size becomes a confounding variable.

2. **Minimum Description Length Score** (MDLScore, also known as Bayesian Information Criterion) penalises larger models to provide a fair comparison for smaller models. The size of the model is characterized by the number of parameters.

$$MDLScore = \frac{|B_n|}{2}\log_2 N - \log_2 P(D_s|B_n)$$

   - A gentle reminder of principle of parsimony or Occam's Razor: A simple model is preferred to a complex model.
   - The first part of the above equation is the average number of bits required to store the values.
   - **Example**. If a network has a parent and a child, and we have 4 datapoints ($N = 4$, each node takes 2 values), the parent has a prior probability of 2 parameters, and the conditional table has $2 \times 2 = 4$. Therefore, $Size(Bn|Ds) = |Bn|log_2(4)/2 = [(2-1)+(2-1)\times 2]log_2(4)/2 = 3 \times 2/2 = 3$.
   - MDL Score is not an absolute measure. It can only be used to compare two models of the same data set.

3. **Measuring Predictive Accuracy**: Split the data into training data and test data, then use the test data to measure the network accuracy. It can be used to minimize the number of variables in a spanning tree.

   - Build a spanning tree and obtain an ordering of the nodes starting at the root.
   - Remove all the arcs.
   - Add arcs in the order of their dependency.
   - If the predictive accuracy is good enough, stop. Else, go back to 3rd step.

8 Approximate Inference

9 Exact Inference

10 Probability Propagation: Join Trees