FADML Scribe

P Prabhanjan (24BM6JP40)
11th March 2025

BAYESIAN BELIEF NETWORK

The Naïve Bayes Classifier makes significant use of the assumption that the values of attributes are **Conditionally Independent** for a given target value, reducing the complexity of learning the target function. However, this assumption is overly restrictive.

A Bayesian belief network describes the probability distribution governing a set of variables by specifying a set of Conditional Independent assumptions and a set of conditional probabilities. In Contrast to the naïve Bayes Classifier, Bayesian belief networks allow stating Conditional Independence assumptions that apply to subsets of the variables.

Conditional Independence

Let X, Y and Z be three discrete-valued random variables. X is conditionally independent of Y given Z if the probability distribution of X is independent of the value of Y given a value for Z:

$$P(X = x_i \mid Y = y_i, Z = z_i) = P(X = x_i \mid Z = z_i) \;\forall\, x_i, y_i, z_i$$

$$or\; P(X \mid Y, Z) = P(X \mid Z)$$

This definition of conditional independence can be extended to sets of variables as well:

$$P(X_1 \dots X_l \mid Y_1 \dots Y_m, Z_1 \dots Z_n) = P(X_1 \dots X_l \mid Z_1 \dots Z_n)$$

Say A_1 is conditionally independent of instance attribute A_2 given the target value V, then:

$$P(A_1, A_2 \mid V) = P(A_1 \mid A_2, V)\, P(A_2 \mid V) = P(A_1 \mid V)\, P(A_2 \mid V)$$

Representation

The Bayesian network represents the joint probability distribution by specifying a set of conditional independent assumptions (represented by a directed acyclic graph) and sets of local conditional probabilities. A node in the Bayesian network represents each variable in the joint space. The network is generally built from domain knowledge.

For each variable, two types of information are specified:
1. The network arcs assert that the variable is conditionally independent of its non-descendants in the network, given its immediate predecessors. We say X is a descendant of Y if there is a directed path from Y to X.

2. A conditional probability table is given for each variable, describing the probability distribution for that variable given the values of its immediate predecessors.

Let us consider a Bayesian Network (B Net) and the Conditional Probability Distribution table (CPDT) of Boolean variables Stormcloud (SC), Depression (D), Lightning (L), Rain (R), Thunder (T), Power cut (PC), Water logging (WL).

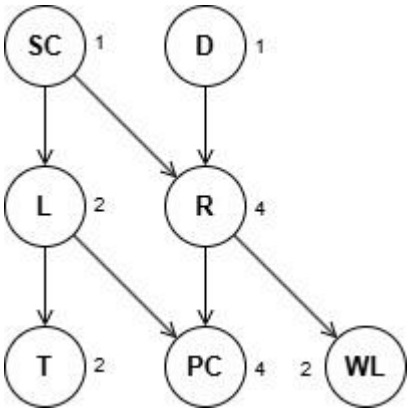


Figure: Bayesian Network

The numbers beside the node represent the total estimates of probabilities required

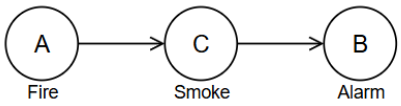
16 Probability estimates are required (a drop from 2^7). For example, Node 'L' requires two estimates:

P (L | SC) and *P (L | ⌐SC)*

D-Separation

D-separation (*d* connotes "directional") is a criterion for deciding, from a given causal graph, whether a set *X* of variables is independent of another set *Y*, given a third set *Z*. The idea is to associate "dependence" with "connectedness" (i.e., the existence of a connecting path) and "independence" with "unconnected-ness" or "separation". Three scenarios of directed graphs are possible:

1. **Chain**:



a) The joint distribution corresponding to this graph:

$$p(a,b,c) = p(a)p(c|a)p(b|c).$$

b) Suppose C is not observed, can test to see if a and b are independent by marginalizing over c to give:

$$p(a,b) = p(a)\sum_{c} p(c|a)p(b|c) = p(a)p(b|a).$$

this in general does not factorize into *p*(*a*)**p*(*b*), hence:

$$a \not\perp b \mid \emptyset$$

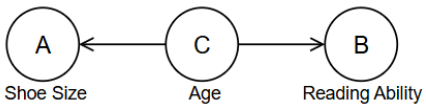
c) If c is observed,

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= \frac{p(a)p(c|a)p(b|c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

We obtain the conditional independence property:

$$a \perp\!\!\!\perp b \mid c.$$

2. Fork



a) The Joint distribution corresponding to this graph,

$$p(a,b,c) = p(a|c)p(b|c)p(c).$$

b) If c is not observed, then we can investigate whether a and b are independent by marginalizing both sides with respect to c to give:

$$p(a,b) = \sum_c p(a|c)p(b|c)p(c).$$

In general, this does not factorize into the product $p(a)p(b)$, and so:

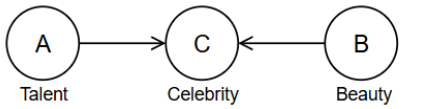
$$a \not\perp\!\!\!\perp b \mid \emptyset$$

c) Suppose c is observed; we can easily write down the conditional distribution of a and b, given c, in the following form, hence the conditional independence.

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

$$a \perp\!\!\!\perp b \mid c.$$

3. Collide



a) The Joint distribution corresponding to this graph,

$$p(a,b,c) = p(a)p(b)p(c|a,b).$$

b) If C is not observed, Marginalizing both sides over *c* we obtain:

$$p(a,b) = p(a)p(b)$$
$$a \perp\!\!\!\perp b \mid \emptyset.$$

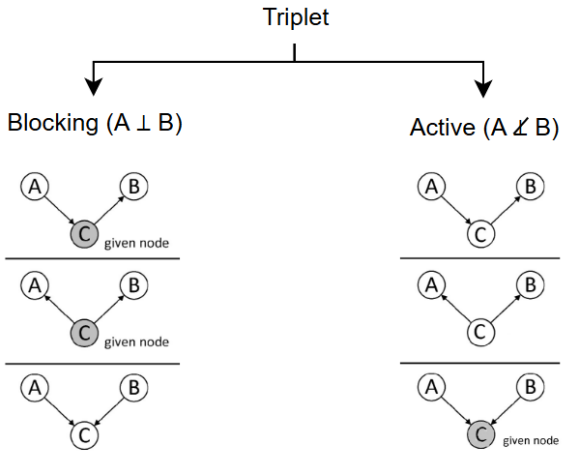
c) If C is observed, the conditional distribution of *A* and *B* is then given by:

$$p(a,b|c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(b)p(c|a,b)}{p(c)}$$

which is not equivalent to product $p(a)p(b)$, so:

$$a \not\perp\!\!\!\perp b \mid c.$$

Summary



EM ALGORITHM

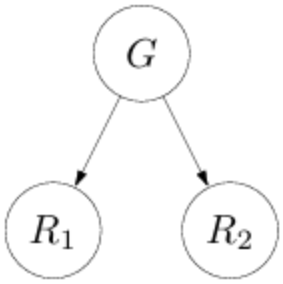
The **Expectation-Maximization (EM) algorithm** is an **iterative method** used to estimate unknown parameters in statistical models. It helps find the best values for unknown parameters, especially when some data is missing or hidden.

It works in two steps:

1. **E-step (Expectation Step):** Estimates missing or hidden values using current parameter estimates.
2. **M-step (Maximization Step):** Updates model parameters to maximize the likelihood based on the estimated values from the E-step.

By iteratively repeating these steps, the EM algorithm seeks to maximize the likelihood of the observed data. It is commonly used for clustering, where latent variables are inferred and has applications in various fields, including machine learning, computer vision, and natural language processing.

Example:
Let us illustrate the EM algorithm with the Movie Rating Model. We'll use:



$$\mathcal{D}_{\text{train}} = \{(?, 2, 2), (?, 1, 2)\}$$

G ∈ {comedy(c), drama(d)} (hidden variable)
R1, R2 ∈ {1,2,3,4,5} (observed variables)

Suppose we have the following dataset where genre G is unobserved:
D_{train} = {(?, 2, 2), (?, 1, 2)} [the values represent (G, R1, R2)]

Initial parameter estimates:
P(G=c) = 0.5, P(G=d) = 0.5
P(R=1|G=c) = 0.4, P(R=2|G=c) = 0.6
P(R=1|G=d) = 0.6, P(R=2|G=d) = 0.4

One Iteration of EM

E-step: For the first example (?, 2, 2), we compute:
P(G=c, R1=2, R2=2) = 0.5 × 0.6 × 0.6 = 0.18
P(G=d, R1=2, R2=2) = 0.5 × 0.4 × 0.4 = 0.08

Normalizing to get P (G|R1=2, R2=2):
P(G=c|R1=2, R2=2) = 0.18/(0.18+0.08) = 0.69
P(G=d|R1=2, R2=2) = 0.08/(0.18+0.08) = 0.31

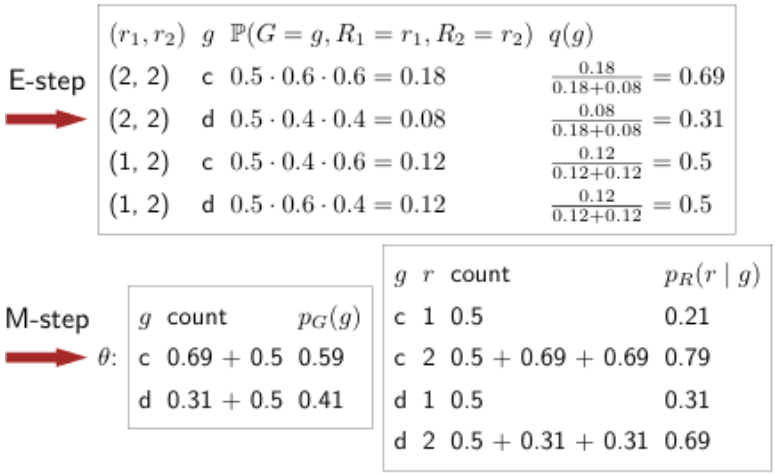
Similarly, for the second example (?, 1, 2):
P(G=c, R1=1, R2=2) = 0.5 × 0.4 × 0.6 = 0.12
P(G=d, R1=1, R2=2) = 0.5 × 0.6 × 0.4 = 0.12

Normalizing:
P(G=c|R1=1, R2=2) = 0.12/(0.12+0.12) = 0.5
P(G=d|R1=1, R2=2) = 0.12/(0.12+0.12) = 0.5

M-step: Update parameters using weighted counts:
P(G=c) = (0.69 + 0.5)/2 = 0.595
P(G=d) = (0.31 + 0.5)/2 = 0.405

For the conditional probabilities:
P(R=1|G=c) = 0.5×1 / (0.5+0.69) = 0.42
P(R=2|G=c) = (0.69×1 + 0.5×1) / (0.69+0.5+0.69+0.5) = 0.58
P(R=1|G=d) = 0.5×1 / (0.5+0.31) = 0.62
P(R=2|G=d) = (0.31×1 + 0.5×1) / (0.31+0.5+0.31+0.5) = 0.38

These updated parameters would then be used in the next iteration. Iterations are done till the parameters converge.



References:

1. Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill Science/Engineering/Math.
2. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer Verlag.
3. *CAUSALITY - discussion*. (n.d.). https://bayes.cs.ucla.edu/BOOK-2K/d-sep.html
4. GeeksforGeeks. (2025, February 4). *ML | ExpectationMaximization Algorithm*. GeeksforGeeks. https://www.geeksforgeeks.org/ml-expectation-maximization-algorithm/
5. *CS221: Artificial Intelligence: Principles and Techniques*. (n.d.). https://stanford-cs221.github.io/autumn2022/