FADML Scribe

Piyus Pramanik (24BM6JP39)

11th March, 2025 (1st Half)

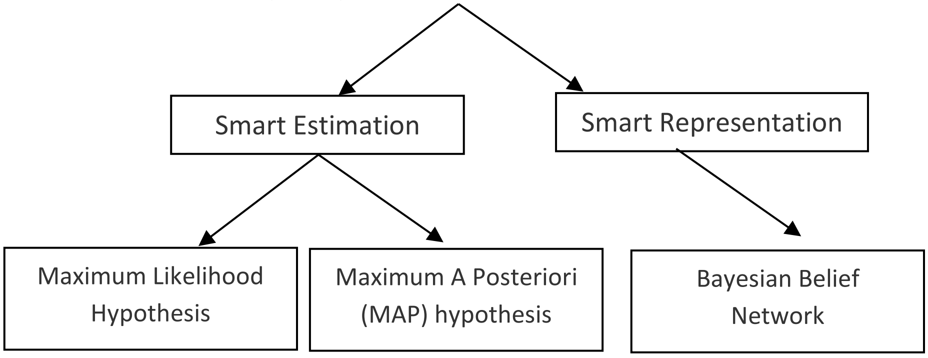
1. BAYESIAN LEARNING: (RECAP)

Y = Target Variable
X_i = Attributes (i =1 - n)

Considering: 1. Number of target variables = 1
2. Number of attributes = n

Concerns :

1. Determine **P(y | x_1,x_2,………x_n)**
2. From Joint Probability Distribution Table (JPDT) of the attributes
3. Issues : Data Sparsity => Solution


Smart Estimation
Smart Representation
Maximum Likelihood Hypothesis
Maximum A Posteriori (MAP) hypothesis
Bayesian Belief Network

Given training data D, we are interested in finding the most probable hypothesis h that hold good

H : Hypothesis space (set of hypothesis)
h ∈ H
D : Available training data

P(h) : Initial probability that hypothesis h holds true
P(D) : Probability of D given no knowledge about hypothesis h
P(D|h) : Probability of observing D given hypothesis (h) holds true

P(h | D): Probability that hypothesis h holds true given D is observed

Posterior Probability -> It reflects the confidence that h holds after we have seen the training data D.

From Bayes Theorem: $$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Maximum A Posteriori (MAP) hypothesis:

During learning, the algorithm considers some set of candidate hypotheses H and is interested in finding the most probable hypothesis h ∈ H, given the observed data D (or at least one of the maximally probable if there are several).

Any such maximally probable hypothesis is called a maximum a posteriori (MAP) hypothesis (h_{MAP})

More precisely, we will say that h_{MAP} is a MAP hypothesis provided:

$$\begin{aligned} h_{MAP} &\equiv \underset{h\in H}{\operatorname{argmax}}\, P(h|D) \\ &= \underset{h\in H}{\operatorname{argmax}}\, \frac{P(D|h)\,P(h)}{P(D)} \\ &= \underset{h\in H}{\operatorname{argmax}}\, P(D|h)\,P(h) \end{aligned}$$

Since P(D) is a constant independent of h.

Maximum Likelihood (ML) hypothesis:

In some cases, we will assume that every hypothesis in H is equally probable a priori (P(hi) = P(h;) for all hi and h; in H). In this case we can further simplify Equation (6.2) and need only consider the term P(D1h) to find the most probable hypothesis. P(D l h) is often called the likelihood of the data D given h.

Any hypothesis that maximizes P(D l h) is called a maximum likelihood (ML) hypothesis, h_{ML}.
More precisely, we will say that h_{ML} is a ML hypothesis provided:

$$h_{ML} \equiv \underset{h\in H}{\operatorname{argmax}}\, P(D|h)$$

2. BAYESIAN CLASSIFIER:

Consider a hypothesis space (H) containing three hypotheses, h_l, h_2, and h_3. Suppose that the posterior probabilities of these hypotheses given the training data(D) are .4, .3, and .3 respectively.

$$P\,(h_1 \mid D) = 0.4$$
$$P\,(h_2 \mid D) = 0.3$$
$$P\,(h_3 \mid D) = 0.3$$

Thus, h_1 is the MAP hypothesis.

Suppose a new instance x is encountered, such that:

$$h_1(x) = (+)\text{ve}$$
$$h_2(x) = (-)\text{ve}$$
$$h_3(x) = (-)\text{ve}$$

Taking all hypotheses into account,

$$P(+) = 0.4$$
$$P(-) = 0.6$$

Hence the most probable classification of the given instance(x) is Negative (with probability = 0.6) , which is different from the classification generated by the MAP Hypothesis.

In general, the most probable classification of the new instance is obtained by combining the predictions of all hypotheses, weighted by their posterior prob- abilities.

If the possible classification of the new example can take on any value v_j from some set V, then the probability $P(v_j \mid D)$ that the correct classification for the new instance is v_j, is just

$$P(v_j|D) = \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

The optimal classification of the new instance is the value v_j, for which $P(v_j \mid D)$ is maximum.
Bayes optimal classification:

$$\underset{v_j \in V}{\operatorname{argmax}} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

3. NAÏVE BAYES CLASSIFIER:

It applies to learning tasks where each instance x is described by a conjunction of attribute values and where the target function f(x) can take on any value from some finite set V.

A set of training examples of the target function is provided,
and a new instance is presented, described by the tuple of attribute values (a_l, a_2.. .a_n).

We need to predict the target value, or classification, for this new instance.

The Bayesian approach: **Assign the most probable target value, V_{MAP}, given the attribute values (**a_l, a_2.. .a_n**) that describe the instance**

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j|a_1, a_2 \ldots a_n)$$

Using Bayes Theorem:

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2 \ldots a_n|v_j)P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

Assumption in Naïve Bayes: The attribute values are **conditionally independent** given the target value.

Consider 2 attributes X_1, X_2 and a target variable Y

1. **$P(X_1 \mid X_2, Y) = P(X_1 \mid Y)$**
2. **$P(X_1, X_2 \mid Y) = P(X_1 \mid Y) * P(X_2 \mid Y)$**

In other words, given the target value of the instance, the probability of observing the conjunction (a_l, a_2.. .a_n) is just the product of the probabilities for the individual attributes:

$$P(a_1, a_2 \ldots a_n|v_j) = \prod_i P(a_i|v_j).$$

Now :

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(v_j)\prod_i P(a_i|v_j)}{\sum_{\text{all j}} P(v_j)\prod_i P(a_i|v_j)}$$

V_{NB} : The target value output by the Naive Bayes classifier

Thus, the naive Bayes learning method involves a learning step in which the various P(v_j) and P($a_i|v_j$) terms are estimated, based on their frequencies over the training data. The set of these estimates corresponds to the learned hypothesis. This hypothesis is then used to classify each new instance by applying the above rule.

In log space Naïve Bayes has a linear classification boundary.

4. GAUSSIAN NAÏVE BAYES CLASSIFIER:

Gaussian Naive Bayes is a type of Naive Bayes method **working on continuous attributes and the data features that follows Gaussian distribution throughout the dataset.** Before diving deep into this topic we must gain a basic understanding of principles on which Gaussian Naive Bayes work. Here are some terminologies that can help us gain knowledge for further study.

Mathematics Behind Gaussian Naive Bayes:

Gaussian Naive Bayes assumes that the likelihood (P(x_i| y)) follows the Gaussian Distribution for each x_i within y_k. Therefore,

$$P(x_i \mid y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Where:

- X_i is the feature value,
- µ is the mean of the feature values for a given class y
- σ is the standard deviation of the feature values for that class,
- π is a constant (approximately 3.14159),
- e is the base of the natural logarithm.

To classify each new data point x the algorithm finds out the maximum value of the posterior probability of each class and assigns the data point to that class.

Consider the following example of a dataset having 2 continuous attribute and a single target variable:
Attribute 1: Height of the student (H)
Attribute 2: FADML marks of the student(F)
Target Variable: Basket Ball Player (BB) / Non-Basket ball player(BB')

Height of the student	FADML Marks of the student	Is a basket ball player
H	**F**	**BB / BB'**

P(H | BB) -> N($\mu_{H|BB}$, $\sigma^2_{H|BB}$)
P(H | BB') -> N($\mu_{H|BB'}$, $\sigma^2_{H|BB'}$)
P(F | BB) -> N($\mu_{F|BB}$, $\sigma^2_{F|BB}$)
P(F | BB') -> N($\mu_{F|BB'}$, $\sigma^2_{F|BB'}$)

Now :

$$P(BB \mid H,F) = \frac{P(H,F \mid BB) * P(BB)}{P(H,F)} \rightarrow P(BB \mid H,F) = \frac{P(H,F \mid BB) * P(BB)}{\{P(H,F \mid BB) * P(BB)\} + \{P(H,F \mid BB') * P(BB')\}}$$

Since given the target variable (BB) H and F are conditionally independent

Hence we can write

$$\rightarrow P(BB \mid H,F) = \frac{P(H \mid BB)*P(F \mid BB) * P(BB)}{\{P(H \mid BB)*P(F \mid BB) * P(BB)\} + \{P(H \mid BB')*P(F \mid BB')* P(BB')\}}$$

For a given training data P(H,F) and P(BB) is const

$$\rightarrow P(BB \mid H,F) = P(H \mid BB)*P(F \mid BB) * Const$$

Similarly

$$P(BB' \mid H,F) = \frac{P(H,F \mid BB') * P(BB')}{P(H,F)} \rightarrow P(BB' \mid H,F) = \frac{P(H,F \mid BB') * P(BB')}{\{P(H,F \mid BB) * P(BB)\} + \{P(H,F \mid BB') * P(BB')\}}$$

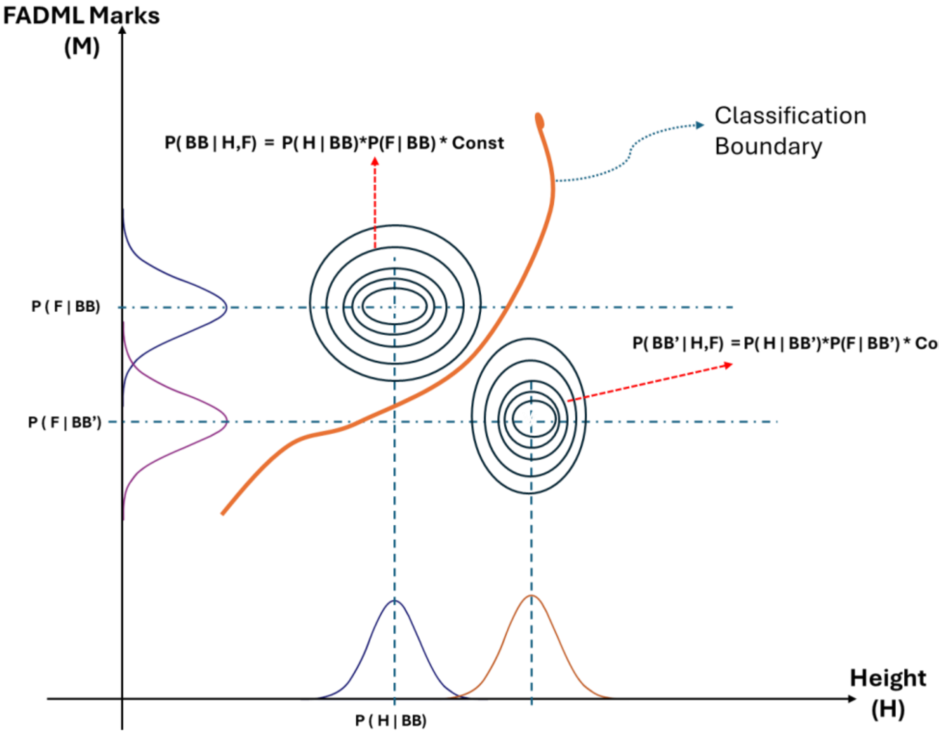
$$\rightarrow P(BB' \mid H,F) = \frac{P(H \mid BB')*P(F \mid BB') * P(BB')}{\{P(H \mid BB')*P(F \mid BB') * P(BB')\} + \{P(H \mid BB) *P(F \mid BB)* P(BB)\}}$$

$$\rightarrow P(BB' \mid H,F) = P(H \mid BB')*P(F \mid BB') * Const2$$

From the available training data we can calculate all the components of the RHS of the above equation, thus the probability of a particular outcome { Prob(Basket Ball Player | height and Marks) / Prob(Not a Basket Ball Player | Height and Marks) } can be determined.

DETERMINING THE CLASSIFICATION BOUNDARY FOR THIS EXAMPLE:

Classification Boundary between “Basketball Player” and “Non-Basketball Player" given the Height and FADML marks of the student.


FADML Marks
(M)
P(BB | H,F) = P(H | BB)*P(F | BB) * Const
Classification Boundary
P (F | BB)
P (F | BB’)
P(BB’ | H,F) = P(H | BB’)*P(F | BB’) * Const2
Height
(H)
P (H | BB)

5. BAYESIAN BELIEF NETWORK:

The naive Bayes classifier makes significant use of the assumption that the values of the attributes ($a_1 . . .a_n$) are conditionally independent given the target value y. This assumption dramatically reduces the complexity of learning the target function. When it is met, the naive Bayes classifier outputs the optimal Bayes classification. However, in many cases this conditional independence assumption is clearly overly restrictive.

In contrast to the naive Bayes classifier, which assumes that all the variables are **conditionally independent** given the value of the target variable, Bayesian belief networks **allow conditional independence assumptions that apply to subsets of the variables.** Thus, Bayesian belief networks provide an intermediate approach that is less constraining than the global assumption of conditional independence made by the naive Bayes classifier, but more tractable than avoiding conditional independence assumptions altogether.

In general, a Bayesian belief network describes the probability distribution over a set of variables.

5a: CONCEPT OF CONDITIONAL INDEPENDENCE

Let X, Y, and Z be three discrete-valued random variables. We say that X is conditionally independent of Y given Z if the probability distribution governing X is independent of the value of Y given a value for 2; that is, if

$$(\forall x_i, y_j, z_k)\ P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

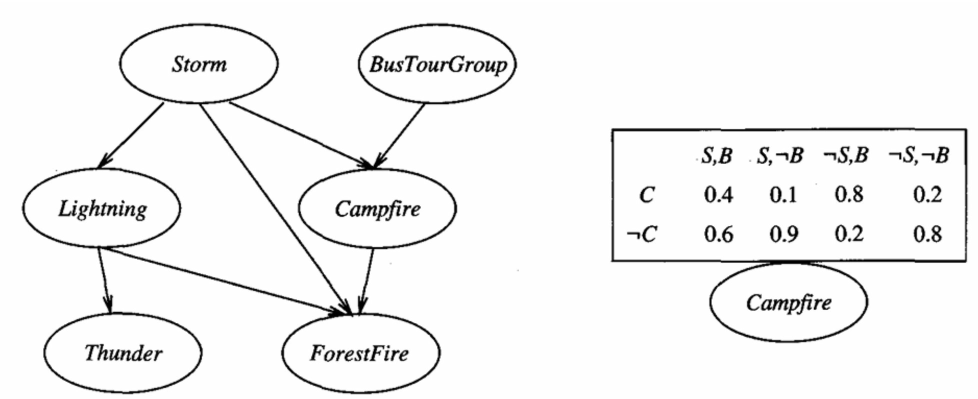
for **x_i ∈ V(X)**, **y_j ∈ V(Y),** and **z_k ∈ V(Z)**.
where V(x) = Set of all possible values of x, and respectively same for V(y) and V(z).

We commonly write the above expression in abbreviated form as **P (X | Y, Z) = P (X | Z)**.
This definition of conditional independence can be extended to sets of variables as well. We say that the set of variables (X_1 . . . X_l) is conditionally independent of the set of variables (Y_l . . . Y_m) given the set of variables (Z_1 . . . Z_n), if

$$\mathbf{P\ (X_1 \ldots X_l \mid (Y_1 \ldots Y_m),\ (Z_1 \ldots Z_n)\ ) = P\ (X_1 \ldots X_l \mid Z_1 \ldots Z_n)}$$

5b: Bayesian Belief Network Representation:

It represents the joint probability distribution for a set of variables.



	S,B	S,¬B	¬S,B	¬S,¬B
C	0.4	0.1	0.8	0.2
¬C	0.6	0.9	0.2	0.8

For example, the Bayesian network above represents the joint probability distribution over the Boolean variables **Storm, Lightning, Thunder, ForestFire, Campjre, and BusTourGroup**. In general, a

Bayesian network represents the joint probability distribution by specifying a set of conditional independence assumptions (represented by a directed acyclic graph), together with sets of local conditional probabilities.

Each variable in the joint space is represented by a node in the Bayesian network. For each variable two types of information are specified.

- First, the network arcs represent the assertion that the variable is conditionally independent of its nondescendants in the network, given its immediate predecessors in the network. (We say X is a descendant of Y if there is a directed path from Y to X).
- Second, a conditional probability table is given for each variable, describing the probability distribution for that variable given the values of its immediate predecessors.

The joint probability for any de- sired assignment of values (y_l, . . ., y_n) to the tuple of network variables (Y_l . . . Y_n) can be computed by the formula

$$P(y_1, \ldots, y_n) = \prod_{i=1}^{n} P(y_i | Parents(Y_i))$$

Parents (Yi) denotes the set of immediate predecessors of Yi in the net- work.

Note the values of **P (y_i | Parents (y_i))** are precisely the values stored in the conditional probability table associated with node y_i.

5c: Illustration of the above Bayesian network:

Consider the node Campfire. The network nodes and arcs represent the assertion that Campfire is conditionally independent of its nondescendants Lightning and Thunder, given its immediate parents Storm and BusTourGroup.

This means that once we know the value of the variables **Storm** and **BusTourGroup**, the variables **Lightning** and **Thunder** provide no additional information about **Campfire**.

The right side of the figure shows the conditional probability table associated with the variable Campfire. The top left entry in this table, for example, expresses the assertion that

P (Campfire = True | Storm = True, BusTourGroup = True) = 0.4

Note this table provides only the conditional probability of CampFire given its parent variables Storm and BusTourGroup.

The set of local conditional probability tables for all the variables, together with the set of conditional independence assumptions described by the network, describe the full joint probability distribution for the network.

5d: Inference:

We might wish to use a Bayesian network to infer the value of some target variable (e.g., ForestFire) given the observed values of the other variables. Given that we are dealing with random variables it will not generally be correct to assign the target variable a single determined value.

What we really wish to infer is the probability distribution for the target variable, which specifies the probability that it will take on each of its possible values given the observed values of the other variables. This inference step can be straightforward if values for all of the other variables in the network are known exactly.

In the more general case we want to find the probability distribution for some variable (e.g., ForestFire) given observed values for only a subset of the other variables (e.g., Thunder and BusTourGroup may be the only observed values available).

5e: Steps for computing the probability of Target variable given the values of other variables

Step 1: Define the Joint Probability Distribution

A Bayesian network represents the joint probability distribution as a product of conditional probabilities:

P(Storm, BusTourGroup, Lightning, Campfire, Thunder, ForestFire) =
P(Storm) * P(BusTourGroup) * P(Lightning | Storm) * P(Campfire | Storm, BusTourGroup) * P(Thunder | Lightning) * P(ForestFire | Lightning, Campfire, Storm)

Step 2: Apply Bayes' Theorem

The goal is to compute:
P (ForestFire | Storm, BusTourGroup, Lightning, Campfire, Thunder)

Using Bayes' rule,

$$P(ForestFire \mid \text{evidence}) = \frac{P(ForestFire, \text{evidence})}{P(\text{evidence})}$$

evidence={Storm, BusTourGroup, Lightning, Campfire, Thunder}.

Step 3: Expand Using Conditional Probabilities

From the structure of the Bayesian network, express the probability of all variables:

= > **P(ForestFire, evidence)**

= **P(Storm, BusTourGroup, Lightning, Campfire, Thunder, ForestFire)**

= P(Storm) * P(BusTourGroup) * P(Lightning | Storm) * P(Campfire | Storm, BusTourGroup) * P(Thunder | Lightning) * P(ForestFire | Lightning, Campfire, Storm)

The denominator, P(evidence), is computed as:

$$P(\text{evidence}) = \sum_{ForestFire} P(ForestFire, \text{evidence})$$

which requires summing over all possible values of **ForestFire (True/False)**.

Step 4: Normalize to Get Conditional Probability

Once both the numerator and denominator are computed, the final probability is:

$$P(ForestFire \mid \text{evidence}) = \frac{P(ForestFire, \text{evidence})}{P(\text{evidence})}$$

This provides the required probability of ForestFire given values of all other variables.