Part-1: Bayesian Learning

Intro:

- Bayesian learning is a probabilistic approach that applies Bayes' Theorem to update the probability estimate for a hypothesis, as more evidence or information becomes available.
- Bayesian Learning is suitable for problems where there is plenty of information/training/prior data to go with.
- That is, the hypothesis for a data is determined and updated every time the new data is added into the considered dataset.
- Terminologies:
- **Hypothesis Space (H)**: A set of all possible hypotheses that can explain the observed data.
- **Prior Probability (P(H))**: The initial probability assigned to a hypothesis before observing any data. This represents prior knowledge or belief about the hypothesis.
- **Likelihood (P(D|H))**: The probability of observing the data D given that the hypothesis H is true. This measures how well the hypothesis explains the data.
- **Posterior Probability (P(H|D))**: The updated probability of the hypothesis after considering the observed data. This is the core of Bayesian updating.
- Bayesian Learning Process:

1. Define the Prior: Start with an initial belief about the hypotheses, represented by the prior distribution P(H)P(H)P(H).
2. Collect Data: Gather observed data DDD.
3. Compute the Likelihood: Evaluate how likely the observed data is under each hypothesis using P(D∣H)P(D|H)P(D∣H).
4. Update Beliefs: Apply Bayes' Theorem to update the prior beliefs and obtain the posterior distribution P(H∣D)P(H|D)P(H∣D).

Bayes' Theorem:

- It is used to find the probability of occurrence of an event, based on some pre-provided evidence.
- Its is stated as, for a given dataset(evidence) (D), the Probability that (H) is the correct hypothesis for (D) is:

$$P(H|D) = \frac{P(D|H) \cdot P(H)}{P(D)}$$ (known as posterior probability)

Where,

P(D | H) - Probability that (D) exists for given Hypothesis (H) (current probability)
P(H) - Probability of Hypothesis (H)
P(D) - Probability of Dataset(D)

- P(D | H) is termed as a likelihood.
- P(H) and P(D) are termed as Dataset(D).

Example

Consider a simple binary classification problem: classifying emails as spam or not spam. Let $H = \{\text{spam}, \text{not spam}\}$ and let D represent the presence of certain keywords in an email.

1. **Prior Probability**: Suppose we believe that 20% of emails are spam, so $P(\text{spam}) = 0.2$ and $P(\text{not spam}) = 0.8$.
2. **Likelihood**: Assume we observe an email containing the word "discount". From past data, we know $P(\text{discount}|\text{spam}) = 0.7$ and $P(\text{discount}|\text{not spam}) = 0.1$.
3. **Evidence**: The marginal likelihood $P(\text{discount})$ is calculated using the law of total probability:

$$P(\text{discount}) = P(\text{discount}|\text{spam}) \cdot P(\text{spam}) + P(\text{discount}|\text{not spam}) \cdot P(\text{not span}$$

$$P(\text{discount}) = (0.7 \times 0.2) + (0.1 \times 0.8) = 0.14 + 0.08 = 0.22$$

4. **Posterior Probability**: Using Bayes' Theorem, we update our belief about the email being spam given the presence of the word "discount":

$$P(\text{spam}|\text{discount}) = \frac{P(\text{discount}|\text{spam}) \cdot P(\text{spam})}{P(\text{discount})} = \frac{0.7 \times 0.2}{0.22} \approx 0.636$$

Similarly, for not spam:

$$P(\text{not spam}|\text{discount}) = \frac{P(\text{discount}|\text{not spam}) \cdot P(\text{not spam})}{P(\text{discount})} = \frac{0.1 \times 0.8}{0.22} \approx 0.36$$

Thus, after observing the keyword "discount," the posterior probability that the email is spam is approximately 63.6%, which updates our prior ↓ ief.

- From the set of posterior probabilities for each hypothesis, we can determine the most probable hypothesis by using h_{map}.
- h_{map} (Maximum-A-Posteriori Hypothesis) is a Hypothesis that maximizes the posterior probability using bayes' theorem.
- It is given by:

$$h_{MAP} = \arg\max_{h \in H} P(H|D)$$

Substituting Bayes' Theorem into this equation, we get:

$$h_{MAP} = \arg\max_{h \in H} \frac{P(D|H) \cdot P(H)}{P(D)}$$

Since $P(D)$ is constant for all hypotheses in the hypothesis space H, it can be ignored in the optimization, leading to:

$$h_{MAP} = \arg\max_{h \in H} P(D|H) \cdot P(H)$$

Bayes' Theorem and Concept Learning:

Maximum Likelihood and Least Squared Error Hypothesis:

- Maximum Likelihood Estimation (MLE) is a method used in statistics and machine learning to estimate the parameters of a machine learning model.
- The principle behind MLE is to find the parameter values that maximize the likelihood(probability) of the observed data.
- Since, the h_{map} focuses on likelihood and prior probabilities, but the h_{ML} focuses only on likelihood probability.
- It is given by:

$$h_{ML} = \arg\max_{h \in H} P(D|H)$$

- The above relation is also called continuous-valued target function (ML Hypothesis), because the set to target-data-values(D) makes it continuous.
- But the ML models like Linear Regression, Non-linear Regression, and Curve Fitting cannot/unable to learn directly from this above relation.
- Hence we derive the Maximum Likelihood Hypothesis relation to get the LSE relation.
- The Least Squared Error (LSE) hypothesis is a method used in regression analysis to find the best-fitting line to a set of observed data points.
- The principle behind LSE is to minimize the sum of squared differences between observed values and predicted values.
- It is given by:

$$h_{LSE} = \arg\min_{h \in H} \sum_i (y_i - h(x_i))^2$$

- By deriving the maximum likelihood hypothesis with a normalization distribution function, we get the minimized/least squared error hypothesis.
- **Derivation:** (Notes)

---

2. Locally Weighted Regression:

- Regression is a statistical approach used to analyze the relationship between a dependent variable (target variable) and one or more independent variables (predictor variables). The objective is to determine the most suitable function that characterizes the connection between these variables.
- When similar data points in a dataset can be separated with a single straight line, then it is known as linear regression, and the data is known as linearly separable data.
- Hence, Locally Weighted Regression is a technique to separate the non-linearly separable data.
- In this method, weights are assigned to each data point, by using the below relation known as Kernel Smoothing:

$$D = ae^{\frac{-\|X - X_0\|}{2c^2}}$$

Where, X = each training input (X1, X2, …… Xn)
X0 = value we are predicting

- The weight to be assigned for a data point is inversely proportional to the difference b/w expected value and predicted value. That is, less the error value, more the weight will be.

$$\text{Weight} \propto \frac{1}{|\text{Expected Value} - \text{Predicted Value}|}$$

- We have to construct a weight matrix for each input value (X).

Drawbacks:
- Computation cost is high.
- Memory requirements are high.

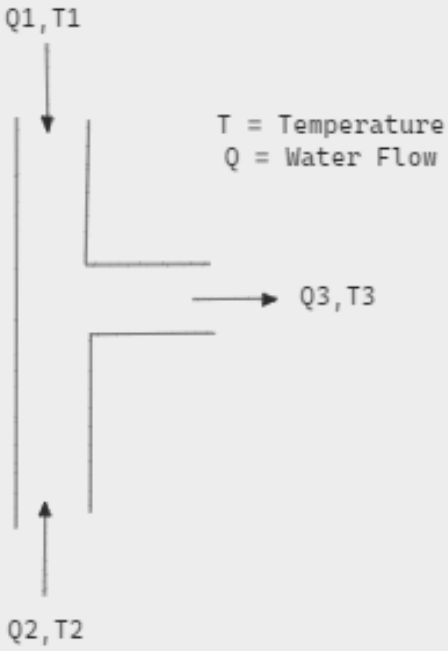
4. Case Based Reasoning:

- Case-Based Reasoning (CBR) is a problem-solving approach in artificial intelligence and cognitive science where new problems are solved by referencing and adapting solutions from previously encountered, similar cases.
- In this method, everything is assumed/considered as a separate case.
- In this method, the instances are represented as the symbols.
- CBR is similar to Instance based learning. That is, the new instances are classified by analyzing the existing instances directly. Similarly, CBR takes previously solved cases, while solving a new case (classifying).
- CBR uses CADET (Case-Based-Desgned-Tool) system which provides 75 predefined case designs.
- Working Procedure of CBR:

1. Case Retrieval: When a new problem arises, the system retrieves cases from its database of previously solved problems that are similar to the current problem. This involves finding cases that have similar features or are in similar contexts.
2. Case Adaptation: Once similar cases are retrieved, the system adapts the solutions from these cases to fit the new problem. This may involve modifying the solution to account for differences between the old and new cases.
3. Solution Application: The adapted solution is then applied to the current problem. This step involves executing the solution and potentially testing it to ensure it works in the new context.

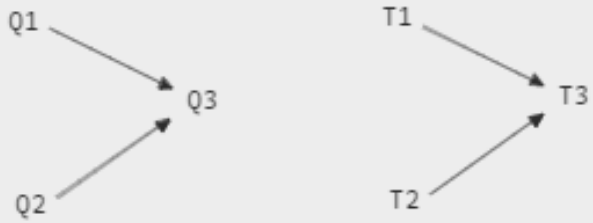
4. Case Storage: After solving the problem, the system stores the new case and its solution in the database. This allows the system to build up a repository of cases over time, improving its ability to solve future problems.

Ex: Consider a case of T-Junction Water Pipe

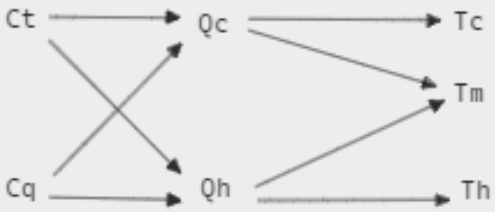


The output water flow(Q) and Temperature(T) are:

Q3 = Q1 + Q2
T3 = T1 + T2



Lets say we have to build a new system, where we can control the direction of waterflow and temperature, then the above design can modified as:



Ct = Control of Temperature
Cq = Control of Waterflow
Qc = Waterflow for cold water
Qh = Waterflow for hot water
Tc = Temp. of cold water
Tm = Temp. of medium water
Th = Temp. of hot water

Remarks on lazy and eager learning:

Lazy Learning: Prediction is made directly by analyzing the pre existing instances. That is, without building a model.
- Computation time and Memory requirement is more, because all the existing instances must be checked and new instances must be stored so that it is used for next new instance classification.
Ex: KNN, etc
Eager Learning: Prediction is made by a trained model with previously provided instances. That is, by building a model.
- Computation time and Memory requirements are relatively less.
Ex: Decision Tree Learning, Naive Bayes Classifier, etc