Weekly Report

Name: Utkarsh Patel
Roll No: 18EC30048
Summary of lectures given in **Week-6** (Oct 7, Oct 8, Oct 9)

Topics Covered

- **Bayes Theorem on Hypothesis Space**
  - **Posterior Probability:** We have for any hypothesis $h \in H$, $P(h|D) = \frac{P(h)P(D|h)}{P(D)}$ via Bayes theorem. Here, $P(h)$ denotes the prior probability of hypothesis h which must be known before observing the training data, $P(D)$ denotes the probability of observing D over instance space X, $P(D|h)$ denotes the likelihood of occurrence of D given hypothesis h holds, and $P(h|D)$ represent the posterior probability of hypothesis h given sample data D.
  - **MAP Hypothesis:** Given a set of candidate hypothesis H, we want to choose the most probable hypothesis $h \in H$ given the observed data D. Such hypothesis is called maximum-a-posterior (MAP) hypothesis.
    $$h_{MAP} \equiv \underset{h \in H}{\operatorname{argmax}} P(h|D) \equiv \underset{h \in H}{\operatorname{argmax}} P(h)P(D|h) \qquad [P(D) \text{ is dropped}]$$
  - **ML Hypothesis:** Some times, the prior probabilities are not known in advance, so all the hypothesis $h \in H$ are given same probability a priori. Hence, in this case we define maximum likelihood (ML) hypothesis.
    $$h_{ML} \equiv \underset{h \in H}{\operatorname{argmax}} P(D|h)$$
- **Bayesian Concept Learning**
  - **Algorithm:** Let X be the instance space, H be the hypothesis space, $c: X \to \{0, 1\}$ be the target concept to be learned and $D \subset (X, c(X))$ be the training data. The algorithm is as follows:
    1. For $\forall h \in H,$ compute posterior probability $P(h|D)$ with Bayes theorem.
    2. Output the hypothesis h_{MAP} with highest posterior probability.
  - **Limitations:** The algorithm is computationally expensive, as it calculates posterior probability of each hypothesis in the hypothesis space. This may become impractical if the hypothesis space is infinite. However, this provides benchmark for other concept learning algorithms.
  - **Assumptions:** The value of $P(h)$ and $P(D|h)$ must be defined under following assumptions:
    1. The training data is error-free.
    2. The target concept c is a member of H.
    3. We have no a priori reason to prefer some hypotheses over others.
  - When no prior probability is known beforehand, we specify the quantities as follows:
    $$P(h) = \frac{1}{|H|}\ ; P(D|h) = \begin{cases} 1, & h(x_i) = c(x_i)\ \forall x_i \in D \\ 0, & otherwise \end{cases}$$
    If so defined, the posterior probability is given as $P(h|D) = \frac{1}{|VS|}$ if $h \in VS$, otherwise 0.
- **Characterising Learning Algorithms under Bayesian Framework**
  - Bayesian analysis can sometimes be used to show that a particular algorithm outputs MAP hypothesis even though it may not explicitly use Bayes theorem or compute probabilities in any form.
    1. Find-S algorithm outputs the maximally specific hypothesis in the version space. This will be a MAP hypothesis under the condition defined in the previous section. Also, let $\mathfrak{H}$ be the distribution of prior probabilities $P(h)$. Under this condition Find-S outputs a MAP hypothesis if $P(h_1) > P(h_2)$, if h_1 is more specific consistent hypothesis as compared to h_2 which is also a consistent hypothesis.
  - The inductive bias in learning algorithms can be depicted through the prior probability and likelihood probability under Bayesian framework.
- **Bayesian Framework in Regression Problems**
  - **Claim 1:** The learning algorithms that minimise the squared error between the predictions and the training data outputs ML hypothesis under Bayesian framework.
  - Consider the following setting: Let X be the instance space, H be the set of hypotheses $h: X \to \mathbb{R}$, let $f: X \to \mathbb{R}$ be the target concept contained in H. Let the training data $D = \langle (x_1, d_1), \dots, (x_m, d_m) \rangle$ be corrupted with random noise drawn from Gaussian distribution. Thus, $d_i = f(x_i) + e_i$, where $e_i \sim N(0, \sigma^2)$. From this, we can infer that $d_i \sim N(h(x_i), \sigma^2)$.
    $$h_{ML} \equiv \underset{h \in H}{\operatorname{argmax}} P(D|h) \equiv \underset{h \in H}{\operatorname{argmax}} \prod\nolimits_{d_i \in D} P(d_i|h) \equiv \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i - f(x_i))^2}$$
    $$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \textstyle\sum_{i=1}^{m} (k - C\left(d_i - f(x_i)\right)^2) = \underset{h \in H}{\operatorname{argmin}} \sum_{i=1}^{m} (d_i - f(x_i))^2, \text{ where } k, C \text{ are positive constants.}$$
    This proves **Claim 1**.
  - Using similar analysis, it can be shown that the learning algorithms considering minimum description length (MDL) outputs MAP hypothesis under Bayesian framework.
- **Bayes Optimal Classifier**
  - Though MAP hypothesis is most probable hypothesis, we can still do better if we perform ensemble learning. Under this framework, we consider all the hypothesis H and possible classes C.
  - The optimal classifier is $\underset{c \in C}{\operatorname{argmax}} \sum_{h \in H} P(c|h)P(h|D)$.
  - This method is computationally intensive. As an alternative, we have *Gibbs* algorithms, which is less optimal. In this algorithm, a hypothesis h is randomly chosen from H as per the posterior probability distribution and this chosen hypothesis is used for making predictions for future instances.
- **Naïve Bayes Classifier**
  - Assume independence among attributes Ai when class is given:
  - $P(A1, A2, \dots, An\ |C) = P(A1|\ Cj)\ P(A2|\ Cj) \dots\ P(An|\ Cj)$
  - Can estimate $P(Ai|\ Cj)$ for all Ai and Cj.
  - New point is classified to Cj if $P(Cj) \prod P(Ai|\ Cj)$ is maximal.
- **Conditional Independence and Bayesian Network**
  - Event A and B are conditionally independent given event C if $P(AB|C) = P(A|C)P(B|C)$.
  - Bayesian Network is a simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions.
  - Using a Bayesian network can save considerable amounts of memory over exhaustive probability tables, if the dependencies in the joint distribution are sparse. For example, a naive way of storing the conditional probabilities of 10 two-valued variables as a table requires storage space for $2^{10} = 1024$ values. If no variable's local distribution depends on more than three parent variables, the Bayesian network representation stores at most $10 \cdot 2^3 = 80$ values.

Regarding Quiz 2

- I was not able to give the Quiz-2 scheduled on Oct 06 due to power cut-off in my region. But, after a day, I gave it a look and problems were quite good.
- No, it was doable under given time constraints.
- Yes, the question regarding *ID3* algorithm improved my understanding of the algorithm.