Weekly Report

Name: Utkarsh Patel
Roll No: 18EC30048
Summary of lectures given in **Week-3** (Sept 16, Sept 17 and Sept 18)

Topics Covered

- **Noise in Training data**
  - If the training set is noisy, only a complex model would result in a consistent hypothesis. However, using such complex model on test set is futile as it will have a high generalization error (due to overfitting).
  - **Occam's razor:** Given comparable training error, a simpler (not too simple) model generalizes better as compared to a complex model.
  - **Inductive Bias:** Higher the proportion of training set in instance space, better is the model fitting and lower is the generalization error.
  - It is preferred to divide the instance space into three disjoint subsets: training set, validation set and test set.

- **Decision Trees: A more powerful representation**
  - Until now, we relied upon the concept of version space, which uses conjunction of attributes as a form of hypothesis, for classification problems. However, this type of representation is not very powerful. For example, it fails for the following training instances:
  $$\langle Sunny, High, Normal, Strong, Cool, Same\rangle +$$
  $$\langle Cloudy, High, Normal, Strong, Cool, Same\rangle +$$
  $$\langle Rainy, High, Normal, Strong, Cool, Same\rangle -$$
  - Using disjunction of conjunction of attributes as the form of hypothesis overcomes this limitation. For the above example, our hypothesis will have the form $(Sky = Sunny) \vee (Sky = Cloudy)$. This form can be represented using a tree, hence the nomenclature. Following this type of structure, the size of the hypothesis space increases drastically and it could be guaranteed that the target function exists in this hypothesis space.
  - **Internal Structure:** Each internal node of a decision tree tests the instance for an attribute. The branches coming out of this node correspond to the various values of the given attribute. Each leaf node assigns a label to that instance.
  - **Limitation:** Instances must have a finite set of attributes, and each such attribute must take values from a finite set. Decision trees are usually preferred when target function is discrete-valued, but in some cases, it can also approximate continuous functions.

- **Constructing Decision Trees**
  - The main concern while constructing a decision tree is its depth, which is analogous to its complexity.
  - Given a training set, finding the perfect decision tree is a **NP-hard** problem. However, we can use some heuristics to develop a greedy algorithm for constructing close-to-perfect decision trees.
  - Constructing a decision tree is associated with preference bias, i.e. not all search strategies are exploited.
  - Our approach would be such that after choosing any attribute for present node, the impurity of the samples will be reduced by the maximum amount and we will stop if all the groups of training instances obtained from previous step become pure.

- **Quantifying *impurity* of groups of instances**
  - We use the concept of entropy to quantify impurity of any group of instances.
  - Let G be any group of instances containing n_+ positive instances and n_- negative instances. Then, probability that any randomly chosen instance will be positive is $p_+ = \frac{n_+}{n_+ + n_-}$ and the probability it will be negative is $p_- = \frac{n_-}{n_+ + n_-}$, with $p_+ + p_- = 1$. The entropy of group G is then defined as:
  $$S(G) = -p_+ log_2(p_+) - p_- log_2(p_-)$$
  - Let A be any attribute which takes, for the sake of simplicity, only binary values. Then, using A, we could split G into at most two subgroups, say G_1 and G_2. We define entropy of G_1 and G_2 using the previous definition. For any split, we can define a quantity, let's call it *gain*, to represent how effective was the split in reducing the *impurity* of the group. We can define it as follows (considering the same example):
  $$Gain(A, G) = S(G) - \left(\frac{|G_1|}{|G|}\right) \cdot S(G_1) - \left(\frac{|G_2|}{|G|}\right) \cdot S(G_2)$$
  It can be proved that $Gain(A, G)$ is always positive and is bounded by unity. In general,
  $$Gain(A, G) = S(G) - \sum_i \left(\frac{|G_i|}{|G|}\right) \cdot S(G_i)$$
  Therefore, if we have a set of attributes $\{A_1, A_2, \ldots, A_t\}$, so for any particular split, we choose the attribute such that we achieve the maximum gain after splitting.

Difficulty of Quiz: Easy and contained no creative question
Did the quiz questions enhance your understanding of the topics covered? No, it was just evaluating us on the fundamentals.