Weekly report of lessons

Name: Mayank Kumar
Roll No: 19CS30029
The week: 10-08-2021 to 15-08-2021

The topics covered:

Introduction - Paradigms of Computing, Underlying Assumptions, Various Contexts of Machine Learning, Definition of Learning Problem, Learning and Computing, Different Types of Learning
Concept Learning - Aim of learning task, Inductive Learning Hypothesis, General and Specific Ordering of Hypothesis, Find-S Algorithm

Summary topic wise:

Introduction

Paradigms of Computing:
Computing by algorithms- Knowledge-driven paradigm; has well-defined algorithms to transform input into output.
Computing by learning- Data-driven paradigm; extracts algorithm from examples and applies it to unobserved input.

Assumptions: Existence of process to generate data, Existence of certain patterns in data, Possibility of approximate construction of model.

Contexts:
Data Mining is extracting valuable information from large databases.
Intelligent System has ability to make sophisticated decisions and learn algorithms in dynamic environments.

Learning Problem consists of Task T, Experience E, and Performance P

Learning and Computing involve building model with described parameters, learning parameters, and optimizing performance using examples or past experiences.

Types of learning:
Supervised Learning: uses labeled datasets to learn input-output mapping.
Unsupervised Learning: finds hidden patterns or clusters in input from unlabeled datasets.
Reinforcement Learning: learns a sequence of actions and is based on rewarding desired behaviors and punishing undesired ones.

Concept Learning

The aim is to learn a target function through its partial description from positive and negative examples.

Inductive learning hypothesis: Any hypothesis h approximating target function over a large dataset will also approximate the function well over unobserved examples.

Each attribute can have - Specific Value, Don't Care Value (?) and No Value Allowed(Φ)

General and Specific Hypothesis:

If any instance x satisfies all constraints of hypothesis h, x is positive example, (h(x) = 1).
Most general hypothesis: (?, ?, ?, ?, ?, ?)
Most specific hypothesis: (Φ, Φ, Φ, Φ, Φ, Φ)

For boolean-valued functions h_j and h_k defined over X, h_j is more-general-than-or-equal-to h_k if and only if for all x in X, ($h_k(x) = 1$) implies ($h_j(x) = 1$).
If an instance satisfies a hypothesis, then it will also satisfy its general hypotheses.

Find-S Algorithm:
Finds a maximally specific hypothesis
Initialize with most specific hypothesis. For each positive example, go to its general hypothesis if necessary (don't change anything for negative examples).
It is assumed that training examples are error-free and some hypothesis exists for them.

Concepts challenging to comprehend:
In second class, I was puzzled on how to count the number of hypotheses (to be precise, how the number was 2^64 in class), but I understood it upon putting a thought for it.

Interesting and exciting concepts:
It was exciting to note that in the example, 'Sky' is an attribute while <Sky=Sunny> is a variable. The Find-S algorithm appeared to be interesting as well, it was totally new to me.

Concepts not understood:
None. Since the introductory classes were very basic, there was no difficulty in understanding a concept.

Any novel idea of yours out of the lessons:
Find-S algorithm is based on assumptions that some hypothesis exists, training examples are error-free and negative examples won't have any impact. Exhaustive checking can lead to exponential runtime, which costs huge time. So, there is need for algorithm which works accurately with inconsistent inputs, considers negative examples and runs faster.

In the example to predict if one will play sport on a day, the output was binary (Yes or No). I think that output can be any real no. between 0 and 1 which would tell the probability for certain values of attributes.
