Introduction to Machine Learning

Machine Learning

- Machine learning is an application of artificial intelligence that involves algorithms and data that automatically analyze and make decision by itself without human intervention.

Training

- In most learning problems, the task is to learn to classify inputs according to a finite (or sometimes infinite) set of classifications.
- A learning system is provided with a set of training data, which have been classified by hand. The system then attempts to learn from these training data how to classify the same data (usually a relatively easy task) and also how to classify new data that it has not seen.

Rote Learning

- The simplest way for a computer to learn from experience is simply to learn by rote.
- Training involves storing each piece of training data and its classification.
- A new item of data is classified by looking to see if it is stored in memory.
- If it is, then the classification that was stored with that item is returned. Otherwise, the method fails.

Learning Concepts

- Concept learning involves determining a mapping from a set of input variables to a Boolean value.
- These methods are known as inductive-learning methods.
- These methods are based on the principle that if a function is found that correctly maps a large set of training data to classifications, then it will also correctly map unseen data.

Example

- The learning task is to determine whether driving in a particular manner in particular road conditions is safe or not.

Attribute	Possible values
Speed	slow, medium, fast
Weather	wind, rain, snow, sun
Distance from car in front	10ft, 20ft, 30ft, 40ft, 50ft, 60ft
Units of alcohol driver has drunk	0, 1, 2, 3, 4, 5
Time of day	morning, afternoon, evening, night
Temperature	cold, warm, hot

- A hypothesis is a vector of values for these attributes. A possible hypothesis is,
  - h1 = <slow, wind, 30ft, 0, evening, cold>

- We also want to represent in a hypothesis that we do not care what value an attribute takes. This is represented by "?", as in the following hypothesis:

  h2 = <fast, rain, 10ft, 2, ?, ?>

- Negative training example
- In other cases, we need to represent a hypothesis that no value of a particular attribute will provide a positive example. We write this as "∅", as in the following hypothesis:
  - h3 = <fast, rain, 10ft, 2, ∅, ∅>

General-to-Specific Ordering

- hg = <?, ?, ?, ?, ?, ?>
- hs = <∅, ∅, ∅, ∅, ∅, ∅>

- hg is the hypothesis that it is safe to drive regardless of the conditions—this is the **most general hypothesis.**
- hs is the **most specific hypothesis**, which states that it is never safe to drive, under any circumstances.

A Simple Learning Algorithm

- The algorithm uses the general-to-specific ordering of hypotheses to search the hypothesis space for a suitable hypothesis.
- The method is as follows:
  - Start with the most specific hypothesis.
  - For each positive training example, determine whether each attribute in the example is matched by the current hypothesis.
    - If it is not, replace the attributes in the hypothesis with the next more general value that does match.

    Example:

    <slow, wind, 30ft, 0, evening, cold>

    <slow, rain, 20ft, 0, evening, warm>

    <slow, snow, 30ft, 0, afternoon, cold>

Version Spaces

- Given a set of training examples (positive and negative), the set of hypotheses that correctly map each of the training examples to its classification is called the version space.

Candidate Elimination

- To find a set of hypothesis that is consistent with all the training example.
- The method operates as follows:
  - Two sets are maintained of hypotheses, hs and hg.
  - hs is initialized as {<∅, ∅, ∅, ∅, ∅, ∅, ∅, ∅>} and

    hg is initialized as {<?, ?, ?, ?, ?, ?, ?, ?>}.
  - When a positive training example is encountered,
    - it is compared with the hypotheses contained in hg. If any of these hypotheses does not match the training example, it is removed from hg.
    - The positive training data are then compared with the hypotheses contained in hs. If one of these hypotheses does not match the training data, it is replaced by the set of slightly more general hypotheses that are consistent with the data, and such that there is at least one hypothesis in hg that is more general.
  - This method is applied in reverse for negative training data.

Initialize the generic and specific boundary

For each training example d, do:

If d is *positive* example

Remove from G any hypothesis h inconsistent with d

For each hypothesis s in S not consistent with d:

- Remove s from S
- Add to S all minimal generalizations of s consistent with d

If d is *negative* example

Remove from S any hypothesis h inconsistent with d

For each hypothesis g in G not consistent with d:

- Remove g from G
- Add to G all minimal specializations of g consistent with d

Inductive Bias

- Inductive bias refers to the restrictions that are imposed by the assumptions made in the learning method.
- The inductive bias of the candidate elimination algorithm is that it is only able to classify a new piece of data if all the hypotheses contained within its version space give the data the same classification. Hence, the inductive bias does impose a limitation on the learning method.

Decision-Tree Induction

- Solution of ID3 algorithm.
- Refer the class notes for solved problem on decision tree.

The Problem of Overfitting

- Overfitting usually occurs when there is noise in the training data, or when the training data do not adequately represent the entire space of possible data.
- If the training data do not adequately and accurately represent the entire data set, the decision tree that is learned from it may not match unseen data.

Reinforcement Learning

- A system that uses reinforcement learning is given a positive reinforcement when it performs correctly and a negative reinforcement when it performs incorrectly.
- The information that is provided to the learning system when it performs its task correctly does not tell it why or how it performed it correctly, simply that it did.