Machine Learning (CS60050) – Weekly Report

Kaushal Banthia (19CS10039)

Week 3: 25th – 27th August, 2021

Topics Covered:

- Next Training Example and Incomplete Hypothesis Space
- Inductive Bias
- Computational Complexity of the Version Space and the size of S and G
- Probably Approximately Correct (PAC) Learning Model
- Theorem of ε – exhausting the Version Space
- Sample Complexity of infinite H
- Handling Noise in Data
- Triple Trade Off and Model Selection
- Decision Trees
- Decision Tree Construction
- Entropy

Summary (Topic Wise):

- Next Training Example and Incomplete Hypothesis Space

  - If we could choose the next Training Example, then we should choose it such that, it reduces, maximally, the number of hypotheses in the version space. The best training example would be the one which satisfies precisely 50% of the hypotheses, present in the version space.
  - If the target function is not present in H, then the preceding discussed algorithms generally do not work. In that case, we need more expressive hypotheses.
  - If we use full logical representation, then all the instances of the training examples are required to learn the concept. Also, only the training examples can be learnt. There would be no generalization on the new examples.

- Inductive Bias

  - Inductive bias (IB) of learning algorithm L is the minimal set of assertions B used to logically infer the value c(x) of any instance x from B, D, and x for any target concept c and training examples D.
  - For a rote learner, B = {}, and there is no IB.
  - For candidate elimination algorithm, the target concept c can be represented in H
  - For Find-S algorithm, the target concept c can be represented in H (all instances that are not positive are negative).

- Computational Complexity of the Version Space and the size of S and G
  - The S set for conjunctive feature vectors is linear in the number of features and the number of training examples.
  - The G set for conjunctive feature vectors is exponential in the number of training examples.
  - Thus, S and G can both grow exponentially, thus significantly affecting computational complexity.

- Probably Approximately Correct (PAC) Learning Model

  - For a consistent hypothesis,
    - Training Error = 0
    - True Error ≠ 0 $(error_D(h) = P(c(x) \neq h(x))\ ;\ D: Population\ Distribution)$
    - C: Concept class defined over a set of instances X of length n, L: A learner using hypothesis space H.
    - C PAC learnable if: ∀ c ∈ C, distribution D over X, $0 < \varepsilon,\ \delta < ½$
    - Learner L with probability of at least $(1 - \delta)$ outputs a hypothesis h ∈ H, such that $error_D(h) \leq \varepsilon$ in time that is polynomial in $1/\varepsilon$, $1/\delta$, n and size(c).

- Theorem of ε – exhausting the Version Space

  - Given a finite H of a target concept c, and m training samples independently, randomly drawn, forming data D, for any $0 < \varepsilon < 1$, $P(VS_{H,D}\ is\ not\ \varepsilon - exhausted) < |H|.e^{-\varepsilon m}$

- Sample Complexity of infinite H

  - By partitioning into 2 sets, the number of possible dichotomies = $2^{|S|}$
  - Vapnik-Chervonenkis (VC) dimension is the size of the largest finite subset in X shattered by H. $VC(H) \leq log_2|H|$
  - Significance of VC Dimension:
    - Measure of model capacity and complexity
    - A bit of a pessimistic measure

- Handling Noise in Data

  - Three major sources:
    - Imprecision in measurement of features.
    - Error in labeling (Teacher noise).
    - Missing additional attributes in representation (hidden or latent attributes).
  - Noise may not provide a consistent hypothesis.
  - Tolerate training error within a limit to use a simpler model.

- Matching Complexities in Data

  - Lower complex model → Higher training and generalization error → Underfitting.
  - Higher complex model → Low training error, but may have high generalization error → Overfitting.
  - Occam's razor: Given comparable empirical error, a simple (but not too simple) model would generalize better than a complex model. This is because simpler explanations are better and also unnecessary complexities are removed.

- Triple Trade Off and Model Selection

  - In any data driven learning algorithm, there is always a trade-off between:
    - The complexity of the hypothesis.

    - The amount of the training data.
    - The generalization error on new examples.
  - Model choices depends on the number of parameters and degree of the polynomial for the regression.
  - Divide the input into training, testing and validation sets. Increase the model complexity by keeping the training and validation error low, but also keep a check on the generalization error.

- Decision Trees

  - Disjunction of Conjunction of attributes (Union of rectangles, instead of a single rectangle) (Can be represented as a tree).
  - They are used to represent learned target functions.
  - Each internal node tests an attribute. Each branch corresponds to an attribute value. Each leaf node assigns a classification
  - The representation of rules in a Decision Tree is like if-else statements.

- Decision Tree Construction

  Top-Down Construction Algorithm

```
1. Start with an empty Tree
2. Main Loop
3.     Split the best decision attribute (A) for the next node.
4.     Assign A as the decision attribute for the node.
5.     For each value of A, create a new descendant of node.
6.     Sort Training examples to leaf nodes.
7.     if Training examples perfectly classified
8.         STOP
9.     else
10.     Iterate over new leaf nodes.
```

  - Try to form a tree with pure leaves. (Correct classification) (Can be done using a greedy approach).

- Entropy

  - A measure for uncertainty, purity and information content.
  - If S is a sample of training examples and p_+ is the proportion of positive examples in S, while p_- is the proportion of negative examples in S, then, entropy of S is $Entropy(S) = -p_+(log_2 p_+) - p_-(log_2 p_-)$
  - This can be generalized to more than 2 values too.
  - Information Gain is the reduction in entropy after choosing attribute A. It is defined as $Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$

Concepts Challenging to Comprehend: None yet.

Interesting and Exciting Concepts: Probably Approximately Correct (PAC) Learning Model and Decision Trees

Concepts not understood: None yet.

A novel idea: Use of decision trees can be done for continuous values (regression), by maybe treating the numbers as discrete labels.