CS60050 Machine Learning - Weekly Report

Nisarg Upadhyaya (19CS30031)

Week 2: 19 August

1 Topics covered

- Version Spaces
- Candidate-Elimination Algorithm

2 Summary

While the Find-S algorithm outputs a hypothesis from H that is consistent with the training examples, this may be just one hypothesis out of the many possible consistent ones. The Candidate-Elimination algorithm gives a description of all the possible hypothesis consistent with the training examples without explicitly enumerating the complete hypothesis space H.

2.1 Version Spaces

We denote a hypothesis h is consistent with a set of training examples D in the following manner

$$Consistent(h, D) \equiv h(x) = c(x)\ \forall \langle x, c(x)\rangle \in D$$

A version space with respect to a hypothesis space H and training set D is the set of all those hypothesis in H which are consistent with D. We denote it as

$$VS_{H,D} \equiv \{h \in H | Consistent(h, D)\}$$

2.2 Candidate-Elimination Algorithm

A naive approach to find and represent the version space is to first list all the hypothesis in H. Then eliminate any hypothesis which is inconsistent with the training set D. This is called the List-Then-Eliminate algorithm. The Candidate-Elimination algorithm not only does better in terms of time by not enumerating the complete hypothesis space it even returns the version space in a compact representation using two sets, the general boundary G and the specific boundary S, defined as follows:

1. **General boundary** G is the set of maximally general members of H consistent with D, i.e., there doesn't exist any hypothesis g' consistent with D which is more general than any member of G.
2. **Specific boundary** S is the set of maximally specific members of H consistent with D, i.e., there doesn't exist any hypothesis s' consistent with D which is more specific than any member of S.

The version space then becomes

$$VS_{H,D} = \{h \in H | (\exists s \in S)(\exists g \in G)(g \geq_g h \geq_g s)\}$$

Procedure

Initialize G to the set of maximally general hypotheses in H, $\langle ?, ?, ?, ?, ?, ? \rangle$
Initialize S to the set of maximally specific hypotheses in H, $\langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$
For each training example d

- If d is a positive example
  - Remove from G any hypothesis inconsistent with d
  - For each hypothesis s in S that is not consistent with d
    - Remove s from S
    - Add to S all minimal generalizations h of s such that h is consistent with d, and some member of G is more general than h
    - Remove from S any hypothesis that is more general than another hypothesis in S
- If d is a negative example
  - Remove from S any hypothesis inconsistent with d
  - For each hypothesis g in G that is not consistent with d
    - Remove g from G
    - Add to G all minimal specializations h of g such that h is consistent with d, and some member of S is more specific than h
    - Remove from G any hypothesis that is less general than another hypothesis in G

3 Challenging concepts

The exact procedure of the Candidate-Elimination algorithm is a bit complex.

4 Interesting concepts

The compact representation of the complete version space using only the general and specific boundaries S and G.

5 Concepts not understood

None

6 Ideas

The Candidate-Elimination algorithm returns a description of the version space as boundaries S and G. By looking at these boundaries we can then look for some specific training examples to add to our data set which will help us narrow down our version space even more instead of adding random training data.