© September 2022 | IJIRT | Volume 9 Issue 4 | ISSN: 2349-6002

Concept Learning and the General-to-Specific Ordering of Hypotheses in Machine Learning

Dr. Kothuri Parashu Ramulu[1], Dr. Jogannagari Malla Reddy[2]
[1]*Associate Professor in CSE, Indur Institute of Engineering & Technology*
[2]*Professor in CSE, Mahaveer Institute of Science & Technology*

***Abstract*—The problem of inducing general functions from specific training examples is central to learning. This paper considers concept learning: acquiring the definition of a general category given a sample of positive and negative training examples of the category. Concept learning can be formulated as a problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples. In many cases this search can be efficiently organized by taking advantage of a naturally occurring structure over the hypothesis space-a generalto-specific ordering of hypotheses. This chapter presents several learning algorithms and considers situations under which they converge to the correct hypothesis. We also examine the nature of inductive learning and the justification by which any program may successfully generalize beyond the observed training data**

Index Terms— General-to specific, concept Learning, FIND-S, Boolean-valued Function

I. INTRODUCTION

Much of learning involves acquiring general concepts from specific training examples. People, for example, continually learn general concepts or categories such as "bird," "car," "situations in which I should study more in order to pass the exam," etc. Each such concept can be viewed as describing some subset of objects or events defined over a larger set (e.g., the subset of animals that constitute birds).

Alternatively, each concept can be thought of as a boolean-valued function defined over this larger set (e.g., a function defined over all animals, whose value is true for birds and false for other animals). In this paper I consider the problem of automatically inferring the general definition of some concept, given examples labeled as members or non members of the concept. This task is commonly referred to as concept learning, or approximating a Boolean-valued function from examples.

II. CONCEPT LEARNING TASK

To ground our discussion of concept learning, consider the example task of learning the target concept "days on which my friend Amar enjoys his favorite water sport." Table 1.1 describes a set of example days, each represented by a set of attributes. The attribute EnjoySport indicates whether or not Amar enjoys his favorite water sport on this day. The task is to learn to predict the value of EnjoySport for an arbitrary day, based on the values of its other attributes. What hypothesis representation shall we provide to the learner in this case? Let us begin by considering a simple representation in which each hypothesis consists of a conjunction of constraints on the instance attributes. In particular, let each hypothesis be a vector of six constraints, specifying the values of the six attributes Sky, AirTemp, Humidity, Wind, Water, and Forecast. For each attribute, the hypothesis will either

- indicate by a "?' that any value is acceptable for this attribute,
- specify a single required value (e.g., Warm) for the attribute, or
- indicate by a "0" that no value is acceptable.

If some instance x satisfies all the constraints of hypothesis h, then h classifies x as a positive example (h(x) = 1). To illustrate, the hypothesis that Amar enjoys his favorite sport only on cold days with high humidity (independent of the values of the other attributes) is represented by the expression

(?, Cold, High, ?, ?, ?)

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

TABLE 2.1

Positive and negative training examples for the target concept EnjoySport.

The most general hypothesis that every day is a positive example is represented by (?, ?, ?, ?, ?, ?) and the most specific possible hypothesis that no day is a positive example-is represented by (0,0,0,0,0,0)

Notation: Throughout this paper, we employ the following terminology when discussing concept learning problems. The set of items over which the concept is defined is called the set of instances, which we denote by X. In the current example, X is the set of all possible days, each represented by the attributes Sky, AirTemp, Humidity, Wind, Water, and Forecast. The concept or function to be learned is called the target concept, which we denote by c. In general, c can be any booleanvalued function defined over the instances X; that is, c : X-> {O, 1). In the current example, the target concept corresponds to the value of the attribute EnjoySport (i.e., c(x) = 1 if EnjoySport = Yes, and c(x) = 0 if EnjoySport = No).

- Given:
  - Instances X: Possible days, each described by the attributes
    - Sky (with possible values Sunny, Cloudy, and Rainy),
    - AirTemp (with values Warm and Cold),
    - Humidity (with values Normal and High),
    - Wind (with values Strong and Weak),
    - Water (with values Warm and Cool), and
    - Forecast (with values Same and Change).
  - Hypotheses H: Each hypothesis is described by a conjunction of constraints on the attributes Sky, AirTemp, Humidity, Wind, Water, and Forecast. The constraints may be "?" (any value is acceptable), "0 (no value is acceptable), or a specific value.
  - Target concept c: EnjoySport : X -> (0,l)
  - Training examples D: Positive and negative examples of the target function (see Table 1.1).
- Determine:
  - A hypothesis h in H such that h(x) = c(x) for all x in X.

TABLE 2.2

The EnjoySport concept learning task

When learning the target concept, the learner is presented a set of training examples, each consisting of an instance x from X, along with its target concept value c(x) (e.g., the training examples in Table 2.1). Instances for which c(x) = 1 are called positive examples, or members of the target concept. Instances for which C(X) = 0 are called negative examples, or non members of the target concept. We will often write the ordered pair (x, c(x)) to describe the training example consisting of the instance x and its target concept value c(x). We use the symbol D to denote the set of available training examples. Given a set of training examples of the target concept c, the problem faced by the learner is to hypothesize, or estimate, c. We use the symbol H to denote the set of all possible hypotheses that the learner may consider regarding the identity of the target concept. Usually H is determined by the human designer's choice of hypothesis representation. In general, each hypothesis h in H represents a boolean-valued function defined over X; that is, h : X -> {O, 1). The goal of the learner is to find a hypothesis h such that h(x) = c(x) for all x in X.

III CONCEPT LEARNING AS SEARCH

Concept learning can be viewed as the task of searching through a large space of hypotheses implicitly defined by the hypothesis representation. The goal of this search is to find the hypothesis that best fits the training examples. It is important to note that by selecting a hypothesis representation, the designer of the learning algorithm implicitly defines the space of all hypotheses that the program can ever represent and therefore can ever learn. Consider, for example, the instances X and hypotheses H in the EnjoySport learning task. Given that the attribute Sky has three possible values, and that AirTemp, Humidity, Wind, Water, and Forecast each have two possible values, the instance space X contains exactly 3 .2 2 .2 2 .2 = 96 distinct instances. A similar calculation shows that there are 5.4-4 -4 -4.4 = 5 120 syntactically distinct hypotheses within H. Notice, however, that every hypothesis containing one or more "IZI" symbols represents the empty set of instances; that is, it classifies every instance as negative. Therefore, the number of semantically distinct hypotheses is only 1 + (4.3.3.3.3.3) = 973. Our EnjoySport example is a very simple learning task, with a relatively small, finite hypothesis space. Most practical learning tasks involve much larger, sometimes infinite, hypothesis spaces. If we view learning as a search problem, then it is natural that our study of learning algorithms will exa~the different strategies for searching the hypothesis space. We will be particula ly interested in algorithms capable of efficiently searching very large or infinite hypothesis spaces, to find the hypotheses that best fit the training data.

General-to-Specific Ordering of Hypotheses

Many algorithms for concept learning organize the search through the hypothesis space by relying on a very useful structure that exists for any concept learning problem: a general-to-specific ordering of hypotheses. By taking advantage of this naturally occurring structure over the hypothesis space, we can design learning algorithms that exhaustively search even infinite hypothesis spaces without explicitly enumerating every hypothesis. To illustrate the general-to-specific ordering, consider the two hypotheses

hi = (Sunny, ?, ?, Strong, ?, ?)

h2 = (Sunny, ?, ?, ?, ?, ?)

Now consider the sets of instances that are classified positive by h_l and by h_2. Because h2 imposes fewer constraints on the instance, it classifies more instances as positive. In fact, any instance classified positive by hl will also be classified positive by h2. Therefore, we say that h2 is more general than hl. This intuitive "more general than" relationship between hypotheses can be defined more precisely as follows. First, for any instance x in X and hypothesis h in H, we say that x satisfies h if and only if h(x) = 1.We now define the more_general_than_or_equal_to relation in terms of the sets of instances that satisfy the two hypotheses: Given hypotheses hj and hk, hj is more-general-than-or-- equalto hk if and only if any instance that satisfies hk also satisfies hj.

Definition: Let hj and hk be boolean-valued functions defined over X. Then hj is more-general-than-or-equal-to hk (written hj $>=_g$ hk) if and only if

$$(\forall x \in X)[(h_k(x) = 1) \rightarrow (h_j(x) = 1)]$$

We will also find it useful to consider cases where one hypothesis is strictly more general than the other. Therefore, we will say that h_j is (strictly) more-general-than hk (written hj $>_g$ hk) if and only if (hj p, hk) A (hk 2, hi). Finally, we will sometimes find the inverse useful and will say that hj is more-special-than hk when hk is more-general-than hj.

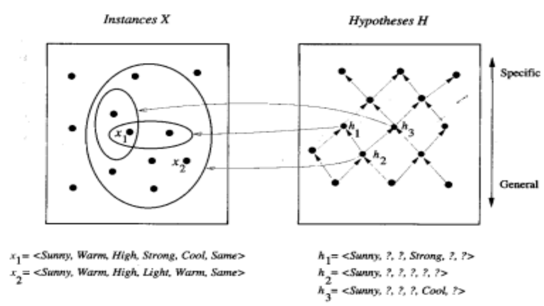

FIGURE 2.1

Instances, hypotheses, and the more-general-than relation. The box on the left represents the set X of all instances, the box on the right the set H of all hypotheses. Each hypothesis corresponds to some subset of X-the subset of instances that it classifies positive. The arrows connecting hypotheses represent the more-general-than relation, with the arrow pointing toward the less general hypothesis. Note the subset of instances characterized by h2 subsumes the subset characterized by h1, hence h2 is more-general-than h1.

To illustrate these definitions, consider the three hypotheses hl, h2, and h3 from our Enjoysport example, shown in Figure 2.1. How are these three hypotheses related by the p, relation? As noted earlier, hypothesis h2 is more general than hl because every instance that satisfies hl also satisfies h2. Similarly, h2 is more general than h3. Note that neither hl nor h3 is more general than the other; although the instances satisfied by these two hypotheses intersect, neither set subsumes the other. Notice also that the p, and >, relations are defined independent of the target concept. They depend only on which instances satisfy the two hypotheses and not on the classification of those instances according to the target concept. Formally, the p, relation defines a partial order over the hypothesis space H (the relation is reflexive, antisymmetric, and transitive). Informally, when we say the structure is a partial (as opposed to total) order, we mean there may be pairs of hypotheses such as hl and h3, such that hl 2, h3 and h3 2, hl. The pg relation is important because it provides a useful structure over the hypothesis space H for any concept learning problem

IV SUMMARY

Concept learning can be cast as a problem of searching through a large, predefined space of potential hypothesis. The general-to-specific partial ordering of hypothesis, which can be defined for any concept learning problems, a useful structure for organizing the search through the hypothesis space. Further the study can be done on the FIND-S algorithm.

ACKNOWLEDGMENT

We are thankful to our friends, faculty members and management for their help in writing this paper.

REFERENCE

[1] Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1957). A study of thinking. New York: John Wiey & Sons.
[2] Buchanan, B. G. (1974). Scientific theory formation by computer. In J. C. Simon (Ed.), Computer Oriented Learning Processes. Leyden: Noordhoff.
[3] Gunter, C. A., Ngair, T., Panangaden, P., & Subramanian, D. (1991). The common order-theoretic structure of version spaces and ATMS's. Proceedings of the National Conference on Artijicial Intelligence (pp. 500-505). Anaheim.
[4] Haussler, D. (1988). Quantifying inductive bias: A1 learning algorithms and Valiant's learning framework. Artijicial Intelligence, 36, 177-221.
[5] Hayes-Roth, F. (1974). Schematic classification problems and their solution. Pattern Recognition, 6, 105-113.
[6] Hirsh, H. (1990). Incremental version space merging: A general framework for concept learning. Boston: Kluwer.
[7] Hirsh, H. (1991). Theoretical underpinnings of version spaces. Proceedings of the 12th IJCAI (pp. 665-670). Sydney.