Hypothesis space

Hendrik Blockeel
Katholieke Universiteit Leuven, Belgium
Leiden Institute of Advanced Computer Science, The Netherlands

Synonyms

Model space.

Definition

The **hypothesis space** used by a machine learning system is the set of all hypotheses that might possibly be returned by it. It is typically defined by a hypothesis language, possibly in conjunction with a language bias.

Motivation, background

Many machine learning algorithms rely on some kind of search procedure: given a set of observations and a space of all possible hypotheses that might be considered (the "hypothesis space"), they look in this space for those hypotheses that best fit the data (or are optimal with respect to some other quality criterion).

To describe the context of a learning system in more detail, we introduce the following terminology. The key terms have separate entries in this encyclopaedia, and we refer to those entries for more detailed definitions.

A learner takes observations as inputs. The observation language is the language used to describe these observations.

The hypotheses that a learner may produce, will be formulated in a language that is called the hypothesis language. The **hypothesis space** is the set of hypotheses that can be described using this hypothesis language.

Often, a learner has an implicit, built-in, hypothesis language, but in addition the set of hypotheses that can be produced can be restricted further by the user by specifying a language bias. This language bias defines a subset of the hypothesis language, and correspondingly a subset of the hypothesis space. A separate language, called the bias specification language, is used to define this language bias. Note that while elements of a hypothesis language refer to a single hypothesis, elements of a bias specification language refer to sets of hypotheses, so these languages are typically quite different. Bias specification languages have been studied in detail in the field of inductive logic programming.

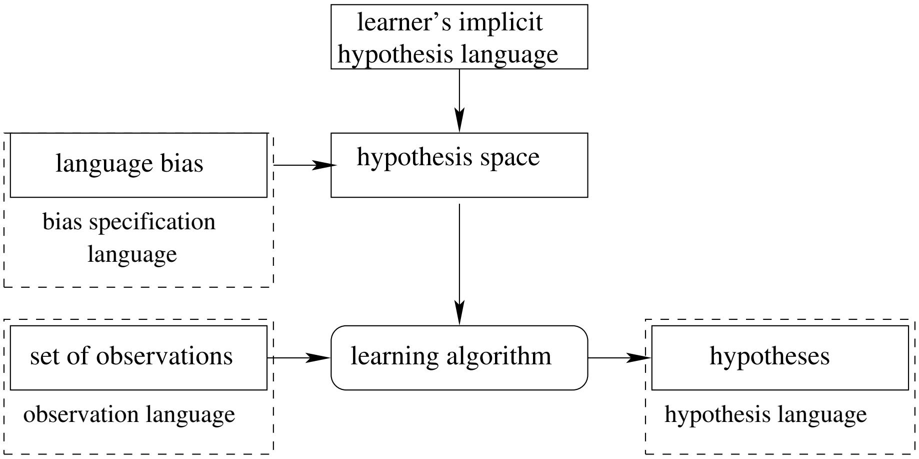

Figure 1: Structure of learning systems that derive one or more hypotheses from a set of observations.

The terms "hypothesis language" and "hypothesis space" are sometimes used in the broad sense (the language that the learner is inherently restricted to, e.g., Horn clauses), and sometimes in a more narrow sense, referring to the smaller language or space defined by the language bias.

The structure of a learner, in terms of the above terminology, is summarized in Figure 1.

Theory

For a given learning problem, let us denote with $\mathcal{O}$ the set of all possible observations (sometimes also called the instance space), and with $\mathcal{H}$ the hypothesis space, i.e., the set of all possible hypotheses that might be learned. Let 2^X denote the power set of a set X. Most learners can then be described abstractly as a function $T : 2^{\mathcal{O}} \rightarrow \mathcal{H}$, which takes as input a set of observations (also called the training set) $S \subseteq \mathcal{O}$, and produces as output a hypothesis $h \in \mathcal{H}$.

In practice, the observations and hypotheses are represented by elements of the observation language $\mathcal{L}_O$ and the hypothesis language $\mathcal{L}_H$, respectively. The connection between language elements and what they represent is defined by functions $\mathcal{I}_O : \mathcal{L}_O \rightarrow \mathcal{O}$ (for observations) and $\mathcal{I}_H : \mathcal{L}_H \rightarrow \mathcal{H}$ (for hypotheses). This mapping is often, but not always, bijective. When it is not bijective, different representations for the same hypothesis may exist, possibly leading to redundancy in the learning process.

We will use the symbol $\mathcal{I}$ as a shorthand for $\mathcal{I}_O$ or $\mathcal{I}_H$. We also define the application of $\mathcal{I}$ to any set S as follows: $\mathcal{I}(S) = \{\mathcal{I}(x) | x \in S\}$, and to any function f as follows: $\mathcal{I}(f) = g \Leftrightarrow \forall x : g(\mathcal{I}(x)) = \mathcal{I}(f(x))$.

Thus, a machine learning system really implements a function $T' : 2^{\mathcal{L}_O} \rightarrow$

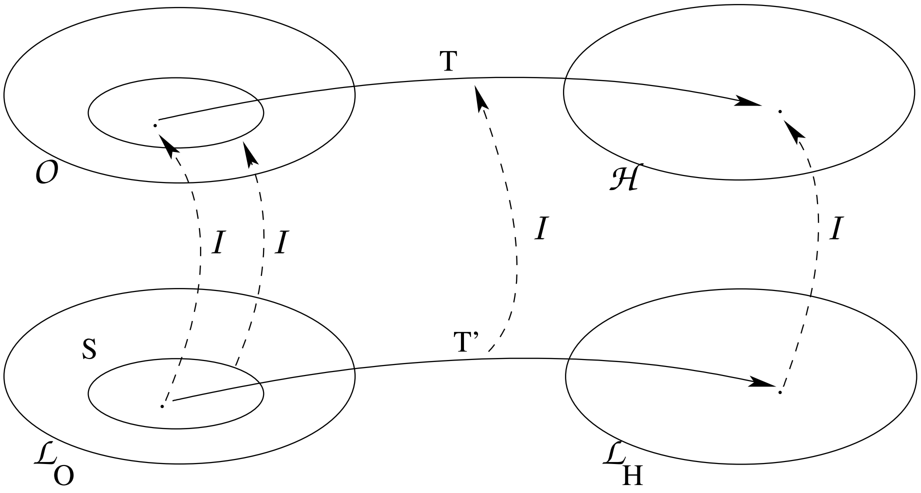

Figure 2: Illustration of the interpretation function $\mathcal{I}$ mapping $\mathcal{L}_O$, $\mathcal{L}_H$ and T' onto $\mathcal{O}$, $\mathcal{H}$ and T.

$\mathcal{L}_H$, rather than a function $T : 2^{\mathcal{O}} \rightarrow \mathcal{H}$. The connection between T' and T is straightforward: for any $S \subseteq \mathcal{L}_O$ and $h \in \mathcal{L}_H$, $T'(S) = h$ if and only if $T(\mathcal{I}(S)) = \mathcal{I}(h)$; in other words: $T = \mathcal{I}(T')$.

Figure 2 summarizes these languages and spaces and the connections between them. We further illustrate them with a few examples.

Example 1 *In supervised learning, the observations are usually pairs* (x, y) *with* $x \in X$ *an instance and* $y \in Y$ *its label, and the hypotheses are functions mapping* X *onto* Y*. Thus* $\mathcal{O} = X \times Y$ *and* $\mathcal{H} \subseteq Y^X$*, with* Y^X *the set of all functions from* X *to* Y*.* $\mathcal{L}_O$ *is typically chosen such that* $\mathcal{I}(\mathcal{L}_O) = \mathcal{O}$*, i.e., each possible observation can be represented in* $\mathcal{L}_O$*. In contrast to this, in many cases* $\mathcal{I}(\mathcal{L}_H)$ *will be a strict subset of* Y^X*, i.e.,* $\mathcal{I}(\mathcal{L}_H) \subset Y^X$*. For instance,* $\mathcal{L}_H$ *may contain representations of all polynomial functions from* X *to* Y *if* $X = \mathbf{R}^n$ *and* $Y = \mathbf{R}$ *(with* $\mathbf{R}$ *the set of real numbers), or may be able to represent all conjunctive concepts over* X *when* $X = \mathbf{B}^n$ *and* $Y = \mathbf{B}$ *(with* $\mathbf{B}$ *the set of booleans).*

When $I(\mathcal{L}_H) \subset Y^X$, the learner cannot learn every imaginable function. Thus, $\mathcal{L}_H$ reflects an inductive bias that the learner has, called its *language bias*. We can distinguish an implicit language bias, inherent to the learning system, and corresponding to the hypothesis language (space) in the broad sense, and an explicit language bias formulated by the user, corresponding to the hypothesis language (space) in the narrow sense.

Example 2 *Decision tree learners and rule set learners use a different language for representing the functions they learn (call these languages* $\mathcal{L}_{DT}$ *and* $\mathcal{L}_{RS}$*, respectively), but their language bias is essentially the same: for instance, if*

$X = \mathbf{B}^n$ *and* $Y = \mathbf{B}$*,* $\mathcal{I}(\mathcal{L}_{DT}) = \mathcal{I}(\mathcal{L}_{RS}) = Y^X$*: both trees and rule sets can represent any boolean function from* $\mathbf{B}^n$ *to* $\mathbf{B}$*.*

In practice a decision tree learner may employ constraints on the trees that it learns, for instance, it might be restricted to learning trees where each leaf contains at least two training set instances. In this case, the actual hypothesis language used by the tree is a subset of the language of all decision trees.

Generally, if the hypothesis language in the broad sense is $\mathcal{L}_H$ and the hypothesis language in the narrow sense is $\mathcal{L}'_H$, then we have $\mathcal{L}'_H \subseteq \mathcal{L}_H$ and the corresponding spaces fulfill (in the case of supervised learning)

$$\mathcal{I}(\mathcal{L}'_H) \subseteq \mathcal{I}(\mathcal{L}_H) \subseteq Y^X.$$

Clearly, the choice of $\mathcal{L}_O$ and $\mathcal{L}_H$ determines the kind of patterns or hypotheses that can be expressed. See the entries on observation language and hypothesis language for more details on this.

Recommended Reading

The term “hypothesis space” is ubiquitous in the machine learning literature, but few articles discuss the concept itself. In inductive logic programming, a significant body of work exists on how to define a language bias (and thus a hypothesis space), and on how to automatically weaken the bias (enlarge the hypothesis space) when a given bias turns out to be too strong. The expressiveness of particular types of learners (e.g., classes of neural networks) has been studied, and this relates directly to the hypothesis space they use. We refer to the respective entries in this volume for more information on these topics.

References

[1] L. De Raedt. *Interactive Theory Revision: an Inductive Logic Programming Approach.* Academic Press, 1992.

[2] C. Nédellec, H. Adé, F. Bergadano, and B. Tausend. Declarative bias in ILP. In L. De Raedt, editor, *Advances in Inductive Logic Programming*, volume 32 of *Frontiers in Artificial Intelligence and Applications*, pages 82–103. IOS Press, 1996.

See also:

bias specification language, hypothesis language, inductive logic programming, observation language