Chapter 18
Machine Learning

There are at least two ways for us to learn: *Learning by being told,* and *learning by discovery.*

In the first case, the learner is explicitly told what is to be learned, e.g., students take classes. The key is in an intelligent *interactive* process.

In the second case, the learner discovers, by herself, new information from unstructured observations, planing, or experimenting in the environment. For example, this is what you are expected to do in doing senior projects.

In between...

We can also learn by examples: look at a few examples, then try to get something general out of them. For example, given such examples as

$$1 = 1, 1 + 3 = 4, 1 + 3 + 5 = 9, ...,$$

a learner might be able to observe something general via partial induction, i.e.,

$$\sum_{i=1}^{n} (2i - 1) = n^2.$$

Then, s/he might proceed to prove it using mathematical induction.

Inductive learning

This approach of learning by examples is also called inductive learning, and is one of the most investigated approaches in AI.

Following this approach, a machine can learn how to predict weather, based on the historic data (examples) that we have already collected for the second week of November. For example, "This is in the second week of November, thus, the average temperature is between $50^\circ F$ and $60^\circ F$"

By exactly the same token, a machine can also learn how to diagnose a patient, or plant disease, in exactly the same way.

Characterize the problem

Let U be a universal set of objects that a learner may observe, with $|U|$, the size of U, typically unknown before the process starts. A concept C is defined as a subset of U.

The problem of *learning* C is to learn how to recognize objects in C, in the sense that *given any object in* U, *can we "correctly" say it is in* C *or not.*

Once C is learned, the system is able to, for any object in U, that we might see *in the future,* recognize whether it is in C.

Concept examples

In the weather example, U is the collection of all the average weekly temperatures for the second week of November, and C is the recorded range of the temperatures for the second week of November. Given any year, can we tell its temperature range?

The concept of poisonous mushroom is the set of all poisonous mushrooms out of the collection of all the mushrooms. Given any mushroom, can we tell if it is poisonous?

For the concept of an arch in a block world, U is the set of all structures made of blocks. Arch is the subset of U containing all the arch-like stuff. Given any structure, can we tell if it is an arch?

More examples of concepts

For the concept of multiplication, U is the set of triplets of numbers, the concept of multiplication is a collection of triplets $< a, b, c >$ such that

$$c = a * b.$$

Given any triplet $< a, b, c >$, can we tell c is derived following the multiplication rule.

For the concept of a database D, U could be the collection of patient descriptions in terms of several key features, D is the collection of those descriptions of patients who suffer from a certain disease. Given any description of a patient, can we tell if he suffers from that particular disease?

If we answer "yes" to any of these questions, we have *learned* about that concept.

Examples and hypotheses

When we want to predict what the weather will look like for the middle of this November, we can look back at all the data we have collected, and give out educated guess that it is between $50^\circ F$ and $60^\circ F$"

This is what we call an hypothesis, since no body knows for sure. On the other hand, this is a very likely one, if out of all the data collected, the average temperature of 75% of the second week of November fall into this range.

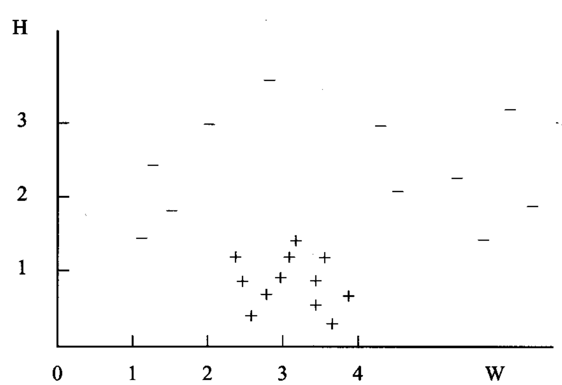
We would say that this educated guess is *similar to* 75% of the examples that we have seen, and *dissimilar to* only 25% of the examples.

Another example

Assume that we have collected a bunch of mushrooms for each of which, we have also got an expert opinion.

Each of the mushroom is described with its *height and width:* and is also categorized into one of two classes: *edible and poisonous,* represented with ‘+’ (edible) or ‘-’.

Below shows the features of the mushrooms we have collected.



Assume we just got another one, with its height being 1, and width being 3. Looking at the picture, we might say it is edible, based on the examples.

We can certainly be surprised since this classification of the newly found to be edible is still an *hypothesis.* On the other hand, this hypothesis is a very likely one, since its features are similar to those of known examples, and dissimilar to all the poisonous.

A general assumption in machine learning is that similar objects belong to the same class.

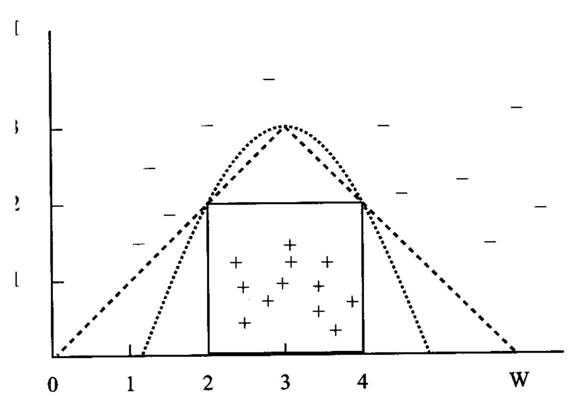
Hypothesis language

The result of learning could be a *concept description*, or a *classifier*, that will classify new objects. They can be described in various languages. For example, we can use *if-then* rules:

1) If $2 < W < 4$ and $H < 2$, then edible else poisonous.

2) If $H > W$ then poisonous, else if $H > 6 - W$ then poisonous else edible.

3) If $H < 3 - (w - 3)^2$, then edible, else poisonous.



Accuracy

The only thing we have for sure is S, the collection of already classified examples. If H does classify examples in S correctly, then it is likely that H may classify unknown objects correctly as well.

Thus, we can apply various hypotheses to reclassify S, and choose the one(s) that correctly reclassify everything in S, called a *consistent* hypothesis.

But, a more important criterion is its predictive accuracy. Those two measurement often don't agree with each other, particularly, when working with noisy data.

Examples

1. Based on the following examples

$$S = \{1 = 1, 1 + 3 = 4, 1 + 3 + 5 = 9\}$$

we made the following hypothesis

$$\sum_{i=1}^{n} 2i - 1 = n^2.$$

This not only correctly classify all pairs $< \sum_{i=1}^{n} 2i - 1, m >, n \in [1, 3]$, but will also correctly predict for all the cases, $n > 3$.

2. Let $e_i, i \geq 1$, be the i^{th} prime number, the following statement only holds for $n \in [1, 4]$, since $e_5 = 1807$ is not prime.

$$e_1 = 2,$$
$$e_n = e_1 \cdot \cdots \cdot e_{n-1} + 1.$$

3. *The performance data shown represents past performance, which is not a guarantee of future results.*

Object description

In a relational description, an object is described in terms of its components and their relations. For example, an arch can be described as a structure consisting of two posts and a lintel. Each of them is a block. Both posts support the lintel, and they don't touch each other.

As a special kind of relational description, an attribute-valued description of a particular arch could be its *length* is 9 meters, *height* is 7 meters, and its *color* is yellow.

Thus, we can use if-then rules, attribute values, predicate logic, etc., to describe objects and concepts.

The scenario

Let's assume that we want to learn some undefined target concept C.

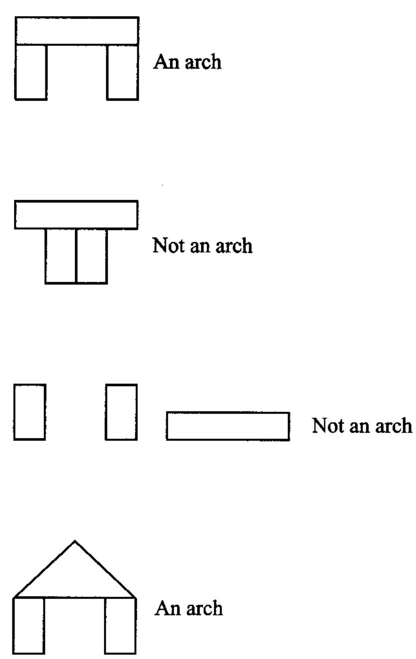
We can use a certain description language L to specify a hypothesis about C. The only source of information for learning about C is a collection of classified examples S, represented as a pair (Object, Class).

Finally, our goal is to find a formula H in L so that H corresponds as well as possible to C.

Learning relational description

Let's check out the learning problem of characterizing an arch in the block world.

Below are four examples: two positive, and two negative.



What should we get?

After going through the four examples one by one, we should learn that

1) An arch consists of three parts: post1, post2, and an lintel.

2) Both post1 and post2 are rectangles; while the lintel can be a more general figure, e.g., a kind of polygon, based on example 1 and 4.

3) post1 and post2 must not touch, based on example 2.

4) Both post1 and post2 must support the lintel.

What has happened?

Regarding to the learning process, it proceeds through a sequence of hypotheses, H_1, H_2, etc. about the concept being learned.

Each of them is an approximation to the target concept, and is the result of all the examples seen so far. After yet another example is studied, the current hypothesis is updated to reflect the new information.

Such an hypothesis keeps on refinement until all the known examples are exhausted, when we have an hypothesis that is consistent with all the observed examples.

Procedurally speaking...

To learn a new concept C from a given sequence of examples, $E_1, E_2, \ldots, E_n$, where E_1 must be a positive example, we do the following:

1. Adopt E_1 as the initial hypothesis, H_1, about the concept C.

2. For each $E_i, i \in [2, n]$,

2.1. Match the current H_{i-1} with E_i; let the result be some description D of the difference between them.

2.2. Act on H_{i-1} according to D and the nature of E_i to obtain a refined hypothesis H_i.

The final result, H_n, represents the system's understanding of the concept C based on the examples, E_1 through E_n.

Two general rules

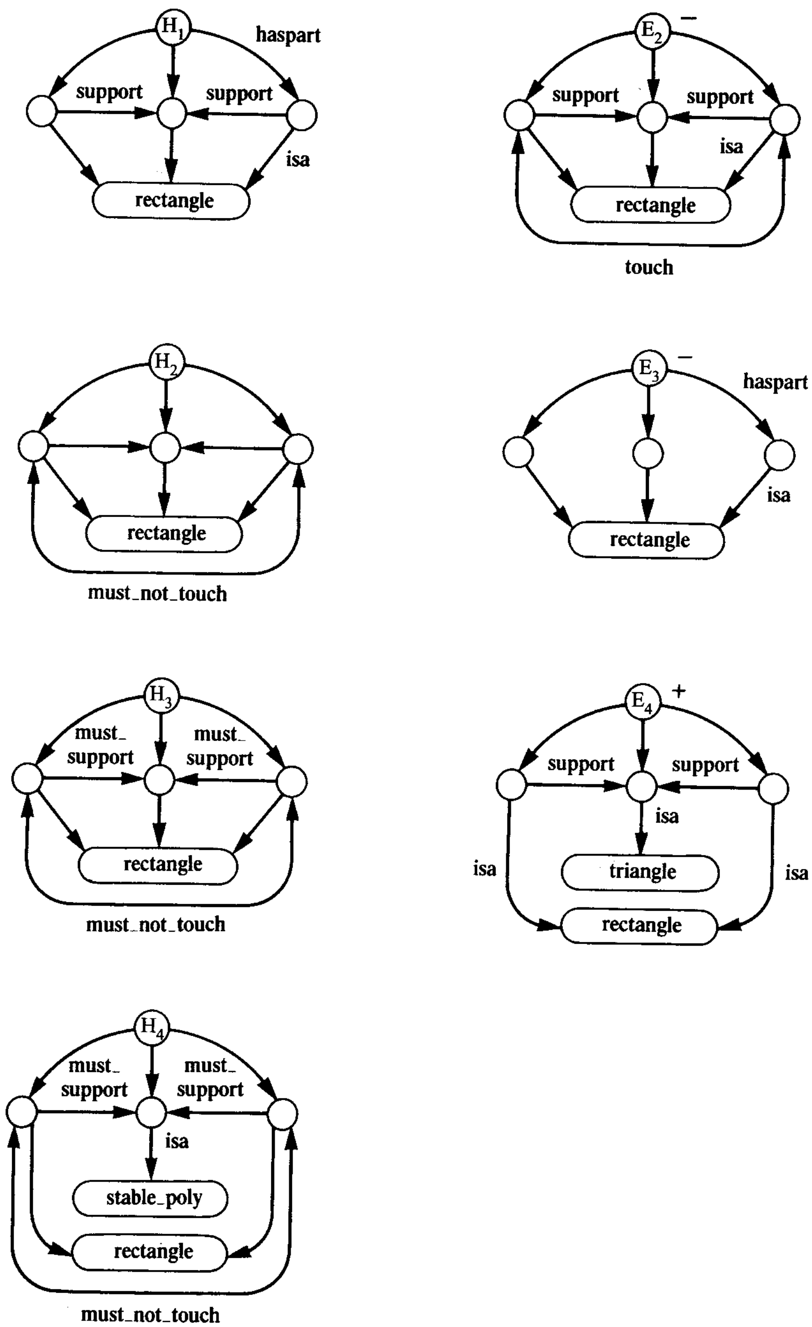
When updating hypotheses, the following rules are useful:

1. If the example is negative and it contains a relation R which is not present in the current hypothesis H, then forbid R in H by add `must_not_R` in H.

2. If the example is negative and it does not contain a relation R which is present in the current hypothesis H, then require R in the new hypothesis by adding `must_R` in H.

We now have an hypothesis development process.

An example



H1
haspart
support
support
isa
rectangle
E2
–
support
support
isa
rectangle
touch
H2
rectangle
must_not_touch
E3
–
haspart
isa
rectangle
H3
must_
support
must_
support
rectangle
must_not_touch
E4
+
support
support
isa
isa
triangle
isa
rectangle
H4
must_
support
must_
support
isa
stable_poly
rectangle
must_not_touch