20 Learning by Managing Multiple Models

In this chapter, you learn about an elegant way to use positive and negative examples to create a *version space*, which in turn helps you to zero in on what it takes to be a member of a class.

The *version-space procedure* can learn quickly as long as there is a stream of noise-free positive and negative examples, there is a fixed number of attributes, and a class-characterizing model can be expressed as a combination of values for those attributes.

By way of illustration, you learn how to extract knowledge from a database that describes the recent habits of a student who sometimes exhibits a mood-depressing allergic reaction.

Once you have finished this chapter, you will see that an ability to handle multiple alternative theories eliminates the need for the negative examples to be near-miss negative examples.

THE VERSION-SPACE METHOD

In Chapter 16, one evolving model was modified each time a new example became available. To prevent mistaken modifications, each new example was required to be only a little different from the current model, thus preventing multiple interpretations and uncertainty about how to modify the model.

In this section, you learn about another way to deal with examples. You prevent mistaken modifications by following each interpretation as long as it remains viable.

Version Space Consists of Overly General and Overly Specific Models

A **version space** is a representation that enables you to keep track of all the useful information supplied by a sequence of learning examples, without remembering any of the examples. A version-space description consists of two complementary trees, one containing nodes connected to overly general models and the other containing nodes connected to overly specific models. Each link between nodes in a version space represents either a specialization or a generalization operation between the models connected to the nodes:

A **version space** is a representation

In which

▷ There is a specialization tree and a generalization tree.

▷ Each node is connected to a model.

▷ One node in the generalization tree is connected to a model that matches everything.

▷ One node in the specialization tree is connected to a model that matches only one thing.

▷ Links between nodes denote generalization and specialization relations between their models.

With writers that

▷ Connect a node with a model

With readers that

▷ Produce a node's model

The key idea in version-space learning is that specialization of the general models and generalization of the specific models ultimately leads to just one, guaranteed-correct model that matches all observed positive examples and does not match any negative examples. You are about to see how the necessary specialization and generalization operations are done, first abstractly, and then in the context of the allergy example.

In figure 20.1(a), you see just two nodes. One represents the most general possible model—the one that matches every possible object. The other represents a very specific model that matches only the description of the first positive example seen.

In figure 20.1(b), you see that the most general model has been specialized such that the new models derived from that most general model all fail to match a new negative example. Each of the new models is to be a small perturbation of the old model from which it was derived.

Also in figure 20.1(b), you see that the most specific model has been generalized such that the new models derived from it are all general enough

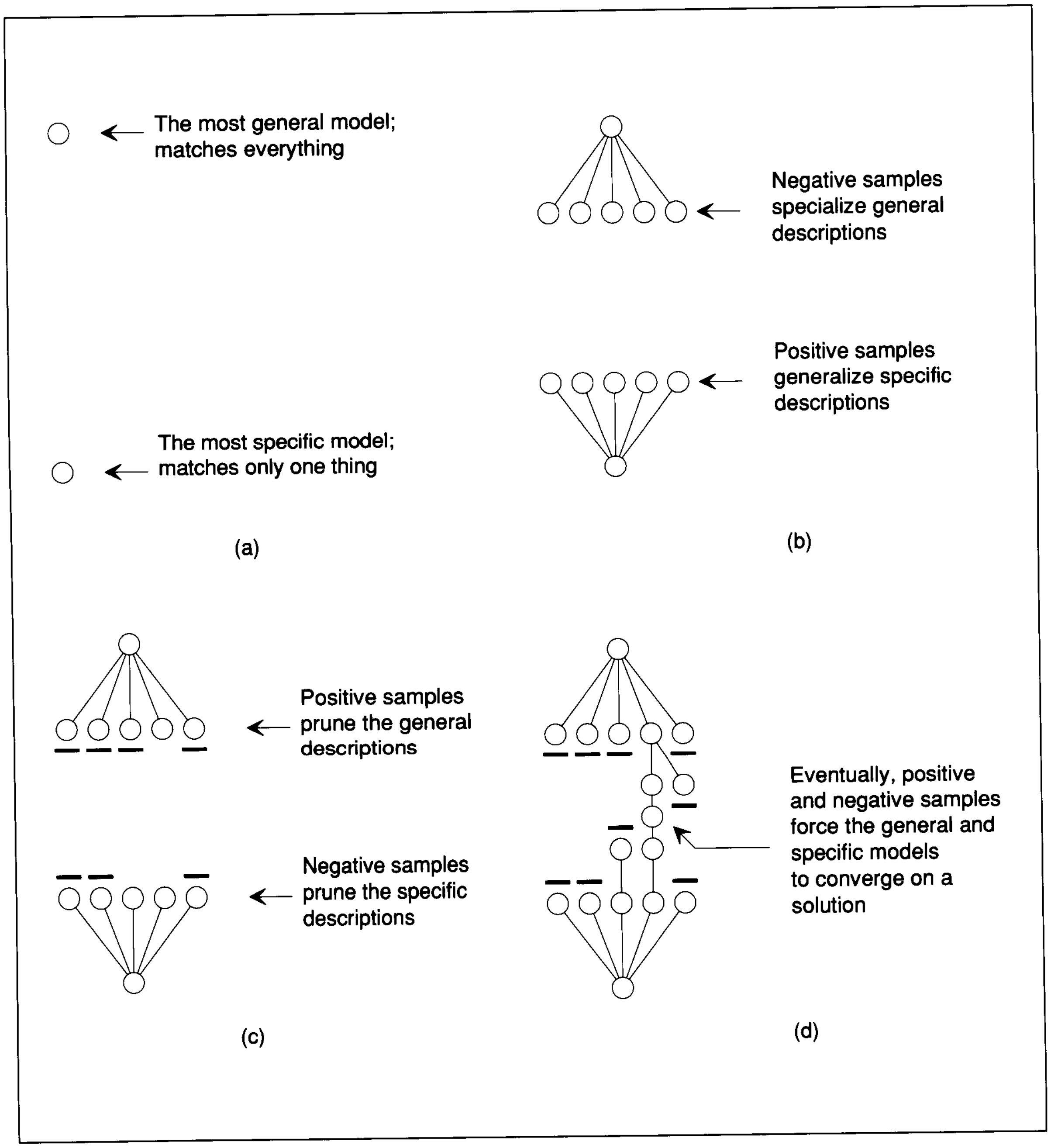

The most general model; matches everything
The most specific model; matches only one thing
(a)
Negative samples specialize general descriptions
Positive samples generalize specific descriptions
(b)
Positive samples prune the general descriptions
Negative samples prune the specific descriptions
(c)
Eventually, positive and negative samples force the general and specific models to converge on a solution
(d)

Figure 20.1 Learning in a version space. Positive examples generalize specific models and prune general models; negative examples specialize general models and prune specific models.

to match a new positive example. Again, each of the new models is to be a small, unique perturbation.

Thus, general models become more specific to prevent match, and specific models become more general to ensure match.

As figure 20.1(c) indicates, however, the trees emerging from the most general model and a specific model do not expand forever. Instead, many branches are pruned away. To see why, consider what happens when you see a negative example. Whenever a specific model matches a negative example, that specific model must be eliminated. The reason is that specific models change only when positive examples are encountered, and positive examples can only make specific models become more general. There is nothing that makes specific models more specific. Hence, if a specific model ever matches a negative example, any other specific model derived from it will match that negative example too.

Symmetrically, each time a positive example is used to generalize the specific models, you need to eliminate those general models that fail to match the positive example. The reason is that general models can only become more specific. Hence, if a general model ever fails to match a positive example, any other general model derived from it will fail to match too.

Eventually, the positive and negative examples may be such that only one general model and one identical specific model survive. When that happens, as shown in figure 20.1(d), no further examples are needed because that sole survivor is sure to be the correct model, assuming that a correct model exists.

Only two details remain. One has to do with what sort of perturbations are allowed to the general and specific models. As emphasized in figure 20.2(a), each time a general model is specialized, that specialization must be a generalization of one of the specific models. Otherwise, no combination of changes could bring the specialization to converge with anything that emerges from the generalization of specific models.

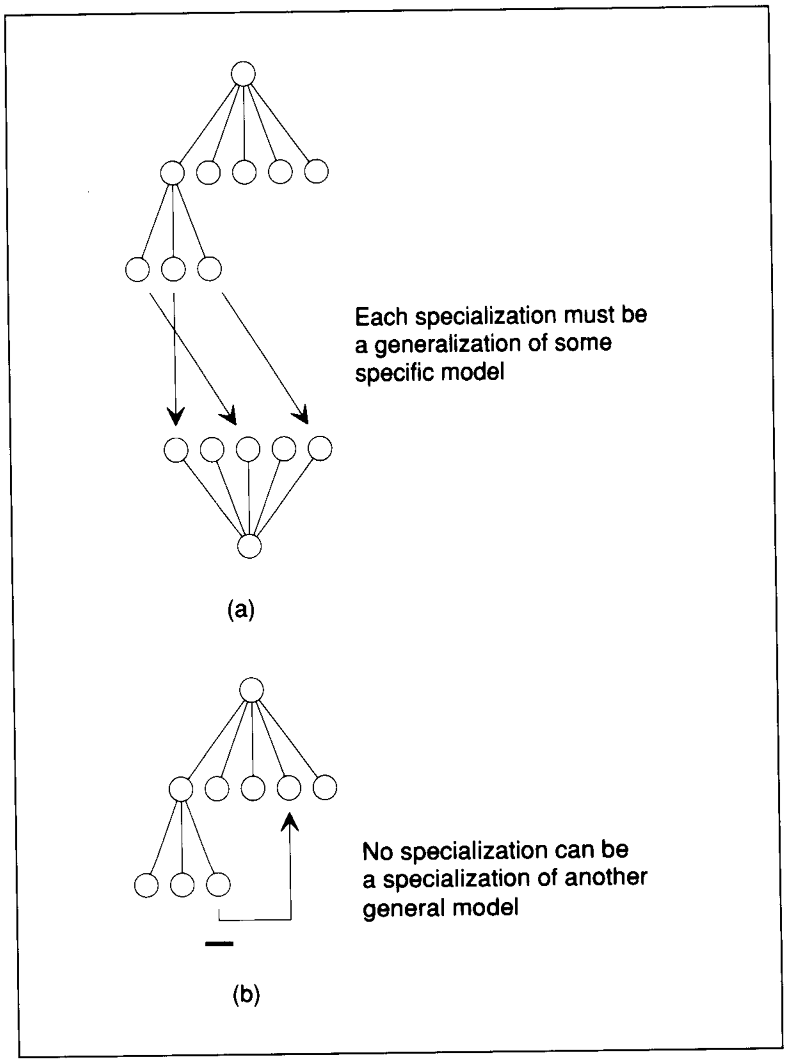
Similarly, each time a specific model is generalized, that generalization must be a specialization of one of the general models. Otherwise, no combination of changes could bring the new generalization to converge with anything that emerges from the specialization of general models.

The other detail is that, each time a general model is specialized, that specialization must not be a specialization of another general model, as suggested in figure 20.2(b). Otherwise, you would be retaining a needlessly specific model in the set of general models.

Generalization and Specialization Leads to Version-Space Convergence

Having learned about version space in abstract terms, let us consider a concrete example. Suppose you are a doctor treating a patient who occa-

Figure 20.2 Each specialization of a general model must be such that it is a generalization of some specific model. Also, each specialization must not be a specialization of some more general generalization.


Each specialization must be a generalization of some specific model
(a)
No specialization can be a specialization of another general model
(b)

sionally suffers an allergic reaction. Your intuition tells you that the allergic condition is a direct result of a certain combination of the place where your patient eats, the time of day, the day of the week, and the amount that your patient spends on food. It might be, for example, that eating breakfast on Fridays is the source of your patient's allergy, or perhaps eating something expensive at Sam's is.

One completely specific model in this food world is a list of values for situation-characterizing attributes such as place, meal, day, and cost. One example of such a model is [Sam's, Dinner, Thursday, Expensive]. Such a model, with every attribute value specified, can match only one restaurant–meal–day–cost combination. In contrast, the most general descriptive model does not specify any attribute value. One notation for this general model is a list of question marks, [?, ?, ?, ?], with each question mark signaling that any attribute value in the corresponding position is allowed.

The model [Sam's, ?, ?, Expensive] matches any situation in which your patient eats expensive food at Sam's on any day at any meal. Models such as this one, with mixed attribute values and question marks, lie between those that are completely specific and the completely general model.

Next suppose, so as to keep the example simple, that the only way to generalize a model is to replace one of the attribute values with a question mark. Thus, one generalization of [Sam's, ?, ?, Expensive] is [Sam's, ?, ?, ?]. Similarly, suppose that the only way to specialize a model is to replace a question mark with an attribute value. Thus, one specialization of [Sam's, ?, ?, Expensive] is [Sam's, Dinner, ?, Expensive].

Note that, if your patient eats three meals per day, seven days per week, some cheap and some expensive, at one of three places, you would have to ask your patient to experiment with 3 × 7 × 2 × 3 = 126 combinations if each combination could, in principle, produce the allergic reaction. But if you happen to believe, somehow, that there is a particular combination of characteristics that produces the allergy in your patient, you can do much better.

Suppose that the examples, in the order analyzed, are as shown in the following table:

Number	Restaurant	Meal	Day	Cost	Reaction
1	Sam's	breakfast	Friday	cheap	yes
2	Lobdell	lunch	Friday	expensive	no
3	Sam's	lunch	Saturday	cheap	yes
4	Sarah's	breakfast	Sunday	cheap	no
5	Sam's	breakfast	Sunday	expensive	no

The first example, a positive one, enables the birth of a version space—the one shown in figure 20.3. It consists of the most general possible model, [? ? ? ?], along with a completely specific model, [Sam's Breakfast Friday Cheap], constructed by straightforward treatment of the positive example itself as a model.

The second example, this time a negative one, forces specialization of the most general model, as shown in figure 20.4. Note that each specialization involves a minimal change to the most general model, [? ? ? ?]. You construct each one by replacing a question mark in the most general model by the corresponding part of the most specific model. This method ensures that each new specialization is a generalization of the most specific model. That, in turn, ensures that specializations of the new general models can eventually converge with generalizations of the existing specific model.

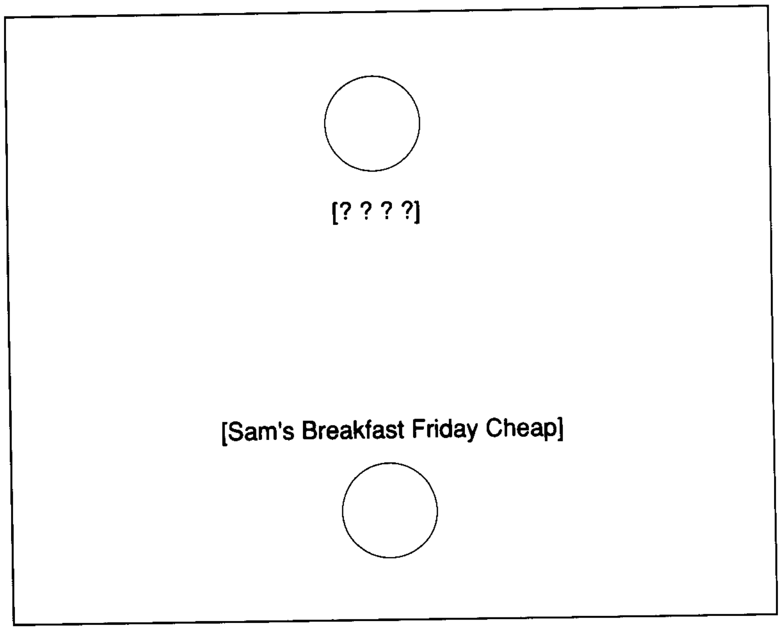


Figure 20.3 A version space is born. It consists of one positive example, together with the most general possible model—the one that matches everything.

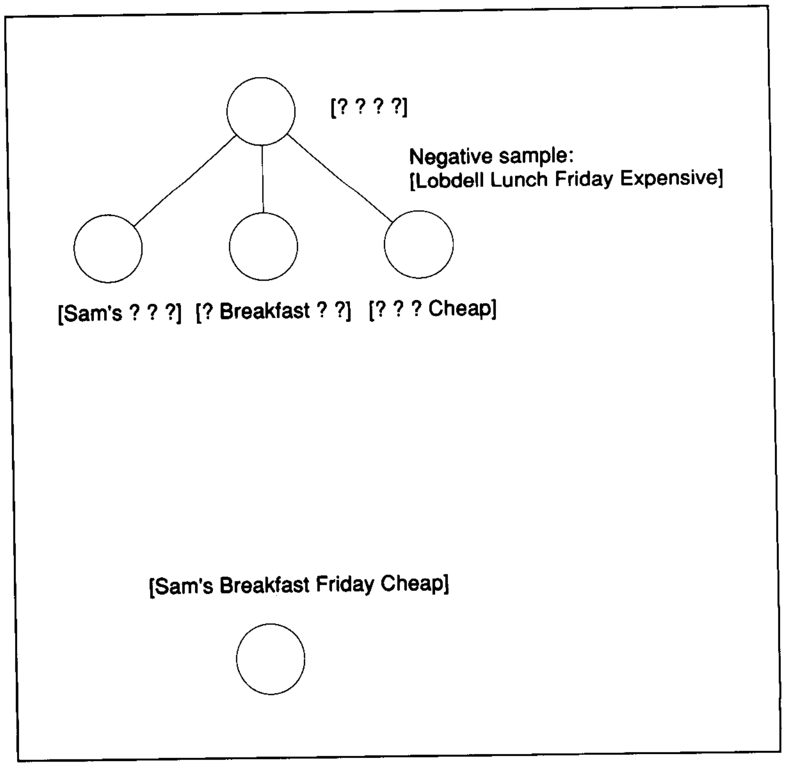


Figure 20.4 A negative example forces specialization of the most general possible model. Each specialization must be a generalization of the single specific model.

Note that there is one exception. The specialization [? ? Friday ?] does not appear in the version space because it matches the negative example, [Lobdell Lunch Friday Expensive]. After all, the point of specializing the most general description is to prevent a match with the negative example.

Without the requirement that specializations must be generalizations of the specific model, many more specializations would be possible. Given three restaurants, three meals, seven days, and either cheap or expensive meals, there must be $3+3+7+2=15$ different ways to specialize [? ? ? ?] by replacing one question mark. Twelve of these 15 possible specializations fail to be generalizations of the specific model.

The third example—another positive one—forces generalization of the most specific model, as shown in figure 20.5. You construct the new model by comparing the existing most specific model with the positive example. In each place where the most specific model differs from the positive example, the existing item in the most specific model is replaced by a question mark. Thus, [Sam's Breakfast Friday Cheap] is compared with [Sam's Lunch Saturday Cheap], producing [Sam's ? ? Cheap]. Thus, the previous most specific model is replaced by a single new generalization that is the same except where the new example positively forces a change.

Note also that [Sam's Lunch Saturday Cheap] cannot possibly match either the general model [? Breakfast ? ?] or any other model that is a specialization of [? Breakfast ? ?]. Accordingly, [? Breakfast ? ?] cannot be on the road toward the ultimate solution, and it is pruned away.

The fourth example—this time another negative one, shown in figure 20.6—forces specialization of any general model that matches it. Because [Sam's ? ? ?] does not match [Sarah's Breakfast Sunday Cheap], however, there is no need to specialize it. Because [? ? ? Cheap] does match [Sarah's Breakfast Sunday Cheap], it requires specialization. As before, the specialization of [? ? ? Cheap] must take the direction of a generalization of a specific model, of which there is only one, [Sam's ? ? Cheap]. Plainly, the only way to specialize [? ? ? Cheap] in the direction of [Sam's ? ? Cheap] is to specialize [? ? ? Cheap] to [Sam's ? ? Cheap], which makes the new general model identical to the only specific model.

Note, however, that this new specialization, [Sam's ? ? Cheap], is also a specialization of another surviving general model—namely, [Sam's ? ? ?]. Accordingly, it is pruned off, because only those specializations that are absolutely forced by the data are to be retained. That way, when the general models and the specific models converge, you can be sure that the positive and negative examples allow no other model.

At this point, only one general model, [Sam's ? ? ?], and one specific model, [Sam's ? ? Cheap], remain. The next example, a negative one, brings them together, as shown in figure 20.7, for [Sam's Breakfast Sunday Expensive] forces a specialization of [Sam's ? ? ?], and the only way to specialize [Sam's ? ? ?] in the direction of [Sam's ? ? Cheap] is to con-

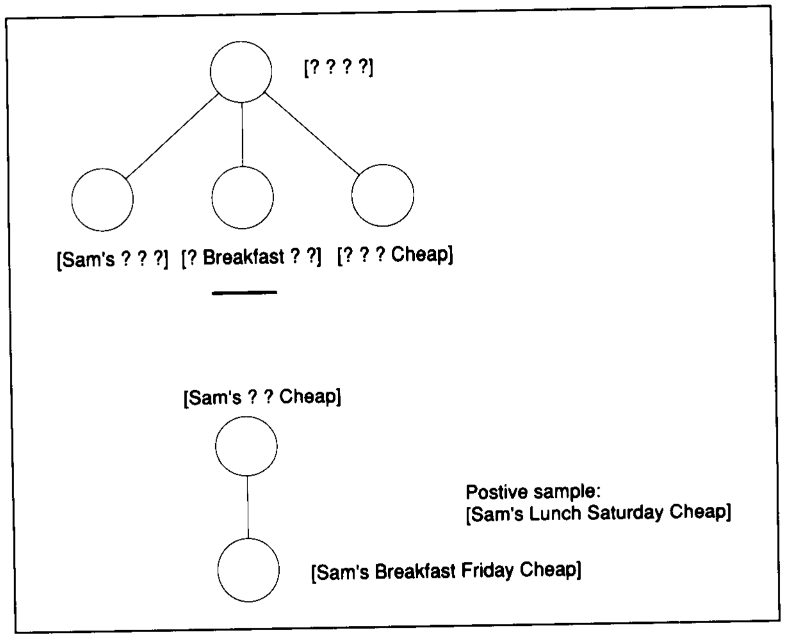


Figure 20.5 A positive example forces generalization of the single specific model. The new, more general specific model is general enough to match both of the positive examples analyzed so far. Also, one general model is pruned.

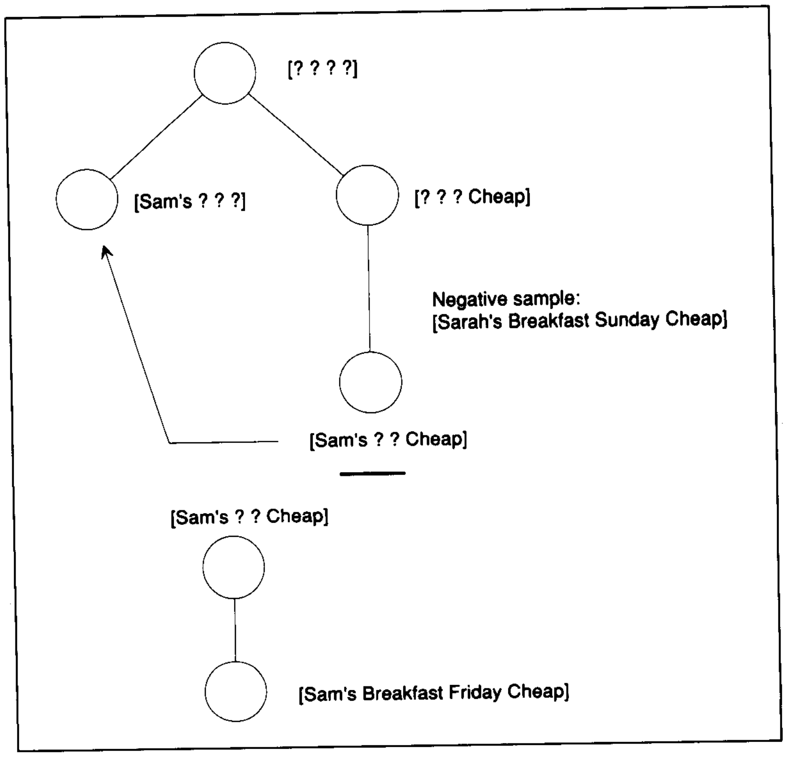


Figure 20.6 Another negative example forces specialization of one of the general models. That new, more specific general model is quickly pruned away, however, for it is a specialization of another general model.

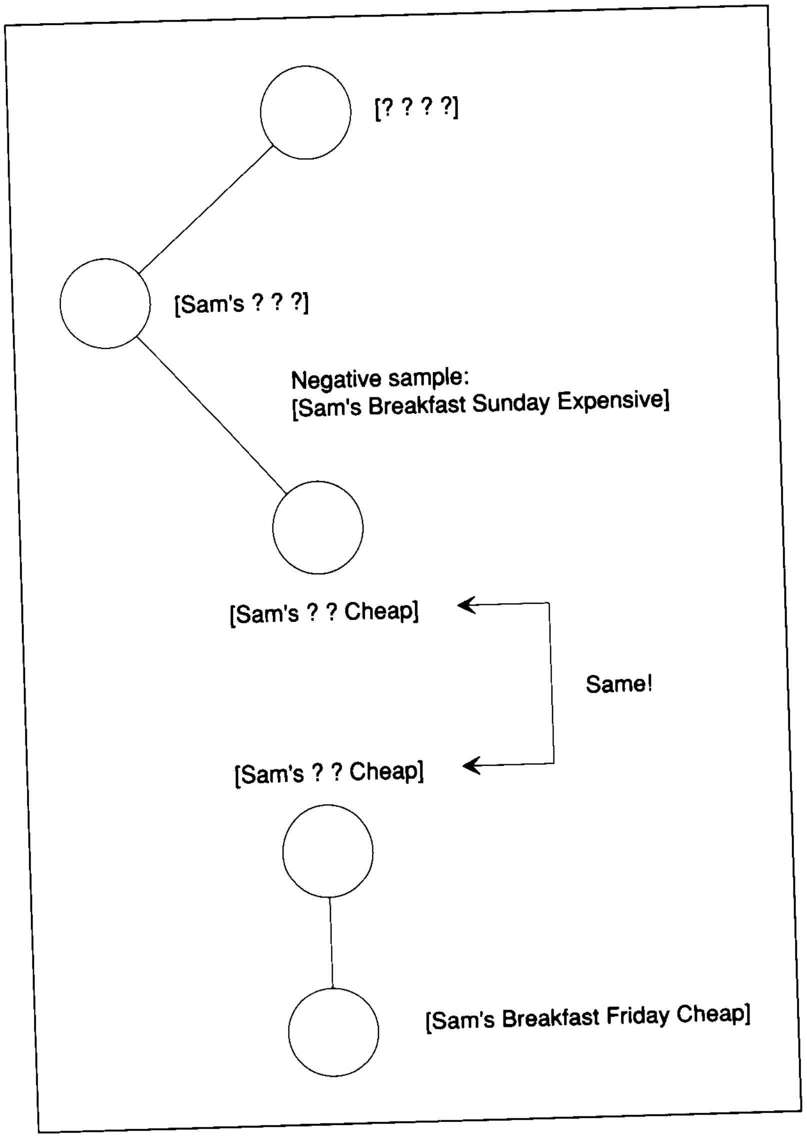


Figure 20.7 Finally, another negative example forces specialization of the only remaining general model. That new, more specific general model is the same as the only specific model. Hence, the general set and the specific set have converged, producing the only possible model that handles the examples properly.

struct a new general model that is the same as the existing specific model. Accordingly, learning concludes. Cheap food at Sam's produces the allergy.

VERSION-SPACE CHARACTERISTICS

In this section, you learn that the version-space procedure handles positive and negative examples symmetrically. You also learn that it can do a lot of good even before it converges on a single description.

The Version-Space Procedure Handles Positive and Negative Examples Symmetrically

By way of summary, here is the version-space procedure. Note that positive and negative examples are handled symmetrically.

To respond to positive and negative examples using a version space,

- ▷ If the example is positive,
  - ▷ Generalize all specific models to match the positive example, but ensure the following,
    - ▷ The new specific models involve minimal changes.
    - ▷ Each new specific model is a specialization of some general model.
    - ▷ No new specific model is a generalization of some other specific model.
  - ▷ Prune away all general models that fail to match the positive example
- ▷ If the example is negative,
  - ▷ Specialize all general models to prevent match with the negative example, but ensure the following,
    - ▷ The new general models involve minimal changes.
    - ▷ Each new general model is a generalization of some specific model.
    - ▷ No new general model is a specialization of some other general model.
  - ▷ Prune away all specific models that match the negative example.

Evidently, the procedure allows multiple specializations as well as multiple generalizations. There is never more than one specialization in the allergy example, because the method used to generalize specializations never produces more than one generalization for each positive example seen. Other generalization methods do not have this characteristic.

The Version-Space Procedure Enables Early Recognition

Once one or more negative examples have been analyzed, you can say with certainty that some examples cannot possibly be positive examples. Just after the first positive and negative examples have been analyzed in the allergy example, there are only three general models. If a combination of attribute values does not match any of them, then it must be a negative example, for none of the three general models can ever become more general.

Thus, you can be sure that [Lobdell Dinner Wednesday Expensive] cannot possibly be a positive example, once you have analyzed [Sam's Breakfast Friday Cheap] and [Lobdell Lunch Friday Expensive], because [Lobdell Dinner Wednesday Expensive] does not match [Sam's ? ? ?], [? Breakfast ? ?], or [? ? ? Cheap].

SUMMARY

- One way to learn is to maintain a set of overly general models and a set of overly specific models in a version space. Then, you can use positive and negative examples to move the two sets together until they converge on one just-right model.
- The version-space procedure enables early recognition before the overly general set and the overly specific set converge.
- The version-space procedure handles positive and negative examples symmetrically. Positive examples generalize the overly specific set and trim down the overly general set. Negative examples specialize the overly general set and trim down the overly specific set.

BACKGROUND

The discussion in this chapter is based on the work of Tom M. Mitchell [1982].