Chapter 3

Representational Specificity and Concept Learning

Joel D. Martin
Dorrit Billman

1. Introduction

When a young child is fed by an adult, she might at first assume that all generally similar creatures, i.e., all `adults`, would treat her equally well. With more experience, she might come to expect food from a smaller group of adults, such as all `familiar adults`, and then from a yet more specific class of creatures, such as `mother, father, and grandmother (MFG)`. On the other hand, the child could initially expect feeding from a very specific set, `mother only`, and with experience generalize to `female relatives (MG)` and then to larger classes, such as `MFG`.

Clearly, both approaches, starting with general classes or starting with specific classes, can produce appropriate results. Both converge toward an optimal level of generality. For this child, `MFG` might be the most appropriate class for several properties such as `feeds`. Even though the two approaches attain the same result, one might be more efficient. For instance, if one approach generally required fewer examples before converging on the correct class, we would use that method in machine learning algorithms and would expect to find it used in natural systems.

One may suppose, though, that neither approach should consistently outperform the other. Rather, depending on the learning situation, sometimes one and sometimes the other would learn most quickly or most cheaply. For example, when regularities in the input occur be-

tween small numbers of properties so that the predictive structure is rule-like, a system that begins by considering general classes (i.e., a general-to-specific method) might learn faster or require fewer resources during learning. On the other hand, a system that begins with specific classes (i.e., a specific-to-general method) might learn relatively slowly. When the regularities occur among many features so that the predictive structure is more idiosyncratic, the specific-to-general system should learn more quickly.

This chapter reports our development and evaluation of these hypotheses about search direction for learning. Surprisingly, we find a different pattern of results than expected. Before we present those results, we first define general and specific representational units, as well as general-to-specific and specific-to-general learning methods. We then outline specific algorithms that embody these methods and report the behavior of the algorithms in systematically varied domains. Finally, we consider the implications of our results for the interaction between domain and method.

2. General and Specific Representational Units

Traditionally, general *representational units*[1] that cover many instances have been contrasted with specific representational units that cover few instances. As in our simple example, a young child may expect the property of `feeds` from a very general class, `adults`, or from a specific class, `mother`. With experience, the child may specialize representational units that are too general, or may generalize units that are too specific.

For many years, both psychologists and computer scientists have employed this distinction (e.g., Medin & Schaffer, 1978; Mitchell, 1982). Some researchers believe that learning methods which rely on specific representational units have the advantage because they preserve much valuable, context-specific information. In contrast, others argue for learning methods that rely on general representational units because they yield better applicability and are less sensitive to errors.

1. A representational unit is a cohesive component of a system's representation of its world. For example, an instance is a representational unit in a case-based approach, and a rule is a representational unit in a rule-based approach.

Psychologists have traditionally tried to determine how humans form generalizations. Recently, though, several researchers have proposed that learning methods which acquire only instances provide equally good or better models of human behavior (e.g., Medin & Schaffer, 1978) than do learning methods which form explicit generalizations. However, it is rarely clear when a representational unit is general or specific. For instance, a representational unit can combine several instances but still not lose any information about idiosyncrasies, yielding a unit that is general in one sense but that has specific content (Barsalou, 1989). Even when instances and rules can be distinguished, the corresponding learning methods can produce equivalent results. The performance algorithm that operates on learned knowledge can simply carry out the additional work necessary to dynamically extract rules from cases or to reconstruct cases from appropriately stored rules.

The same issues arise in the field of machine learning, in which some argue for the use of specific cases over general rules, because cases have important contextual richness that rules lack. However, in practice, cases and rules are difficult to distinguish. Some 'cases' are abstracted and become far more general than some rules. Also, 'rules' are often used to represent individual instances, particularly exceptions. Furthermore, even when the specificity of representational units can be easily distinguished, performance algorithms can carry out the additional processing to produce equivalent behavior. Put simply, the specificity of representational units is ill defined and seems to be a poor determiner of learning behavior.

Moreover, many learning systems represent knowledge at different levels of specificity at different stages, making this a continuum rather than a dichotomy. Nevertheless, these systems typically move in a single direction as they gain experience, working either from specific to general or from general to specific. If one holds the performance element constant, the effects of such a control strategy on behavior depends on one's definition of specificity. Thus, to study these effects, we first extract some precise definitions from the intuitive meanings of specificity.

2.1 Types of Specificity

Three contrasting senses of specificity can be identified, each of which provides a different formalization of the notion that a specific representational unit is somehow 'closer' to the input instances.

2.1.1 Amount of Evidence

Our first sense defines specificity as the amount of evidence that supports a representational unit, that is, the number of observed instances that have contributed to the unit's description. This measure increases even if the same instance is observed multiple times. By this definition, a representational unit is specific if it combines information from one or very few observed instances, and is general if it combines information from many observed instances. For example, a category of tailless cats may combine information from only one or two observed instances and therefore be quite specific. Alternatively, a category of house cats may combine many observed instances and hence be quite general. Several existing systems adopt one or the other extreme on this dimension. Trace and instance models in cognitive psychology (e.g., Estes, 1986; Hintzman, 1986) assume that each representational unit combines one or few instances. The same is true for low-level or leaf nodes in decision tree methods such as ID3 (Quinlan, 1986) and Cobweb (Fisher, 1987). In contrast, prototype models (e.g., Neumann, 1974) integrate across many instances, as do high-level nodes in ID3 and Cobweb.

In general, this definition of specificity may seem most natural for characterizing probabilistic rather than logical representational units, because the former is more concerned with amount of evidence. However, the distinction is still useful for logical units. Suppose a system forms logical representational units by intersecting the attribute values of multiple instances. Early in learning, the system may not want to intersect many instances for fear of overgeneralizing, but as it becomes more confident, it intersects more and more instances. Hence it begins with low-evidence representational units and moves toward higher-evidence ones. Analogous learning methods exist that move in the opposite direction.

2.1.2 Number of Attributes

Our second sense defines specificity as the number of attributes specified in the representational unit. If an instance of a particular disease has ten attributes, a maximally specific representational unit would maintain and use information about all ten, while a general representational unit would include information about only one or a few attributes. Different numbers of attributes are appropriate for different

situations. For example, if most furry animals give live birth, that information can be stored in a representational unit that specifies only the attributes `body-covering` and `birth-method`. On the other hand, the learner sometimes needs to use several attributes. If there is an exception to the previous relationship involving furry animals with webbed feet and a fleshy bill that lay eggs, one must keep information about `body-covering`, `birth-method`, `foot-type`, and `mouth-type`.

Specificity as amount of evidence (definition 1) is conceptually independent from specificity as number of attributes (definition 2). A representational unit that combines information about many instances can maintain either many or few attributes, and the same is true for units that combine few instances.

Many methods maintain information about all attributes, presumably because the additional contextual information is expected to lead to better performance. Classic examples from psychology include most prototype models (e.g., Neumann, 1974) and the Medin and Schaffer model (1978). From machine learning, Fisher's Cobweb (1987) maintains and uses information about all attributes in all representational units.[2] At the other extreme, there are also several methods whose representational units maintain information about only a few attributes. Bruner, Goodnow, and Austin (1956) suggested such methods in their classic work on concept attainment. In addition, most early work in machine learning of concepts proposed methods that formed highly general representational units with values of only one or a few attributes. More recently, Schlimmer and Granger (1987) proposed a method that begins with highly general representational units and that can specialize or generalize the units as necessary.

This definition of specificity is most useful for logical rather than probabilistic representational units, because removing attributes is the only way the former can reduce the impact of some attributes on performance. However, this distinction is still useful for probabilistic units when the learner decides to generalize and not represent some of the possible variation. For example, all hamsters have fur, all are small, and many live outside, although some live inside. A learner may decide that where a hamster lives is not sufficiently related to its being a hamster and may drop that variation from the representation, possibly

2. Although the category utility function at the heart of Cobweb always combines information about all attributes, not all are equally relevant during performance.

allowing some other representational unit, such as one about pets, to make predictions about where a particular hamster lives.

2.1.3 Extensions of the Unit

These two definitions are related to another common interpretation of the general versus specific dichotomy. Under this third interpretation, a general representational unit 'matches' many distinct instances chosen from the theoretical space of all possible instances (see Mitchell, 1982, for an example of this definition of specificity). By 'matches', we mean that the representational unit is somehow activated by an instance. On the other hand, specific representational units match relatively few distinct instances. For example, in the set of possible instances, there may be millions of distinguishable objects that bark, but by logical necessity only a proper subset of these possible objects bark and are black. A representational unit that matches all members of the larger set of barking creatures is more general, by this definition, than one that only matches the members of the smaller set.

This third definition refers to the extension — the instances in the domain. In contrast, the first two definitions focus on what the learner represents. Because we wanted to separate characteristics of the learning method from characteristics of the domain, we chose to use only the first two definitions to guide algorithm development.

2.2 Distinct Definitions

Although related, our three definitions of specificity do differ. They are related both because all three are variants of the intuitive meaning and because in some situations all three definitions agree as to which units are specific. For instance, if we choose a representational unit that matches a large number of instances (general by definition 3), that representational unit would likely summarize a great deal of evidence (general by definition 1). Moreover, with a large extension, the system can maintain fewer attributes (general by definition 2), because larger classes, like mammals, often have fewer distinguishing attributes than do smaller classes like platypuses.

Nevertheless, in some circumstances, the three definitions diverge. To begin with, a representational unit that matches many instances would not be based on much evidence if the members of that class were rarely

encountered. For example, there may be more distinct insects than there are humans in the space of possible instances. However, as humans, we are more likely to encounter and notice other humans. Therefore, we would collect more evidence about the smaller of the two classes simply because we more frequently encounter members of that class. In other words, a representational unit that matches many instances (general by definition 3) does not always produce a corresponding representational unit based on a lot of evidence (general by definition 1).

Furthermore, a representational unit that maintains few attributes does not necessarily match more possible instances than does a unit with many attributes. In particular, these definitions of specificity will differ when the matcher permits partial or best matches. In this case, a representational unit can match an instance even when some attribute values in the instance contradict the representational unit. For example, suppose we have three specific representational units (many attributes), one each for the categories of `poodle`, `siamese`, and `finch`. Of these, the `siamese` category would be the best match for all cat instances, because it shares more attribute values with them than do the other two descriptions. Moreover, when we encounter our first lion, our matcher would again choose the `siamese` category as the best match, even though the lion is bigger and meaner than we expect. Alternatively, we might only have general representational units (few attributes) for the more general categories of `dog`, `cat`, and `lion`. Of these, the `cat` category will be the best match for all cat instances, but will not be the best match for lions. The more specific category, `siamese`, matches more instances than the more general one, `cat`, because the former does not have any strong competitors. Therefore, a representational unit like `cat`, with few attributes (general by definition 2), will not always match more instances (general by definition 3) than will a unit like `siamese` with many attributes.

Finally, a representational unit with few attributes does not necessarily have more evidence supporting it, and vice versa. Consider a system that builds a hierarchy of categories. Concepts higher in the tree, such as `animal`, summarize more evidence than those lower in the tree, such as `dog`. Unfortunately, this tells us nothing about how many attributes must be stored with each category. For instance, suppose we have a hierarchy of mammals and fish that organizes dogs, cats, sharks, and guppies. Some set of features determines whether a creature is a mammal or a fish, and some other set of features determines whether a

particular mammal is a cat or a dog. Neither set of features is necessarily larger than the other. As a simple example, we may know about the features `birth-method`, `body-covering`, and `produced-sound`. The first two attributes distinguish between the high-evidence categories fish and mammal, and only the last one distinguishes between cat and dog. Therefore, a representational unit that is general by definition 1 (amount of evidence) is not always general by definition 2 (number of attributes) and vice versa. It is important to note that although the above examples distinguished the different definitions for logical rather than probabilistic representational units, similar arguments apply for both types of formalism.

2.3 Changing the Specificity of Representational Units

During learning, some representational units are overly general and some are overly specific. The learner must exploit its experience to move toward optimally general representations. In this chapter we are addressing two simple approaches: general-to-specific learning and specific-to-general learning.

The first approach begins with general representational units by one of the two definitions and specializes the general units as necessary. According to the 'amount of evidence' definition, a learning method can specialize by splitting high-evidence units into multiple low-evidence units. Each of the resulting specialized units would then contain a proper subset of the evidence available to the original representational unit. According to the 'number of attributes' definition, a learning method can specialize by adding attributes to a representational unit. In contrast, the specific-to-general approach begins with specific representational units and generalizes those units as necessary. A system can achieve this either by combining low-evidence units (generalizing by definition 1) or by reducing the number of attributes (generalizing by definition 2).

By varying the direction of learning, generalization versus specialization, and by varying the definition of specificity, we generated four different learning methods. Table 1 organizes these four possibilities: amount of evidence plus specific to general; amount of evidence plus general to specific; number of attributes plus specific to general; and number of attributes plus general to specific. We expect differences between the specific-to-general and general-to-specific methods. For exam-

Table 1. The four learning methods.

	Amount of evidence Instance partition	Number of attributes Pattern composition
Specific to general	Early predictions use single instances As necessary, use units with many instances	Early predictions use units with many attributes As necessary, use units with few attributes
General to specific	Early predictions use units with many instances As necessary, use units with fewer instances	Early predictions use units with few attributes As necessary, use units with many attributes

ple, we expect the former to be superior for domains with predominately idiosyncratic relations and the latter to be superior for more abstract relations. However, there is reason to suspect that specific-to-general methods will be generally superior (Fisher & Chan, 1990). Furthermore, because the two definitions of specificity stress different aspects of learning, we expect differences between the methods that control amount of evidence and those that control number of attributes.

3. The Learning Methods

The four methods we generated are all incremental, unsupervised learners that accept instances as lists of attribute values but that, as mentioned, vary in their representational bias. They share many subroutines and they have a common foundation. We begin with an overview of all methods, and then follow with a detailed discussion of each in turn.

3.1 Overview of the Methods

As shown in Table 1, two of the four methods allow explicit variation of the amount of evidence in a representational unit. These two methods are based on a general technique that we call *instance partitioning*. This learning technique recursively partitions the input set into disjoint

categories of similar instances. The other two learning methods in Table 1 allow explicit variation of the number of attributes in a representational unit, based on a technique that we call *pattern composition.* This learning method maintains multiple partitions of simplified categories called *patterns* and combines the information from several patterns.

3.1.1 Instance Partition Methods

Under the first definition of specificity, a representational unit is specific if it has relatively little evidence to support it. The amount of evidence contained in an individual representational unit can be explicitly controlled by organizing representational units as nodes in a tree. In this tree, lower-level nodes have relatively little evidence and higher-level nodes combine the evidence of their children. For example, the category `mammal` would combine the evidence — the number of supporting instances — from `dog`, `cat`, `horse`, etc. A general-to-specific method initially uses higher-level nodes to make prediction decisions and gradually move down the tree. A specific-to-general method begins near the leaves of the tree and gradually moves up to higher-level nodes.[3]

An *instance partition* (IP) method builds a tree of categories by recursively partitioning instances into disjoint sets, and classifies an instance by sorting along a single path in the tree (Figure 1). For example, a basset hound would be classified as a mammal as opposed to a fish, then further classified as a dog as opposed to a cat, and finally classified as a basset hound. Instance partition methods have shown considerable promise (Fisher, 1987; Bobick, 1987; Anderson & Matessa, this volume) both in machine learning and in modeling human categorization. Borrowing from that success, we used two IP algorithms that are similar to Fisher's (1987) Cobweb and Bobick's categorization algorithm (1987).

3.1.2 Pattern Composition Methods

The second definition of specificity states that a representational unit is more specific if it maintains and uses a greater number of attributes. This quantity can be controlled by storing and using multiple patterns

3. None of the methods described in this chapter increases the generality or specificity of the system as a whole. Rather, they vary the specificity of the representational units that are actually used during performance. For the current study, we chose to ignore the possible confound of relearning information.

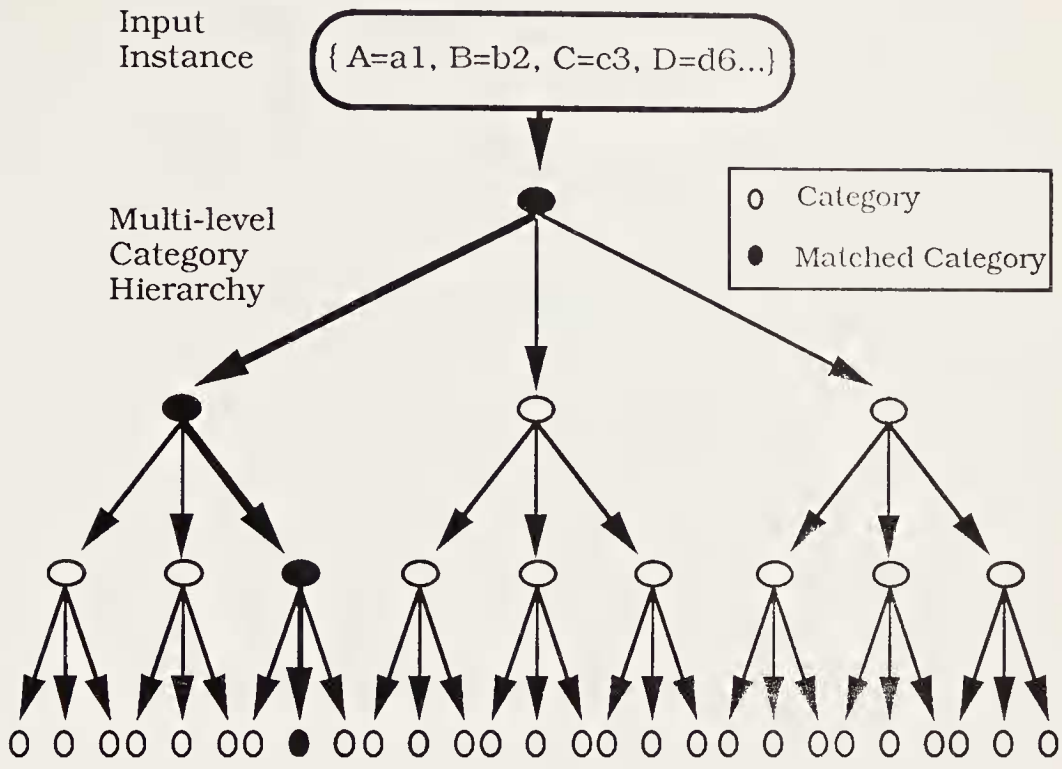

Figure 1. The IP processing architecture, which classifies an instance into exactly one category at each level of the category hierarchy.

that vary the number of attributes used during prediction. A simple type of pattern is a collection of attribute values that tend to cooccur, such as `color = white`, `fat-ratio = high`, and `habitat = cold`. This pattern specifies three attributes, `color`, `fat-ratio`, and `habitat`, a more general pattern might specify only two, and a more specific pattern might specify four or more. During prediction, if known attribute values of a new instance match one part of a pattern, then the learner could predict that the instance would also have the rest of the pattern's attribute values. Under this view, a specific-to-general method begins by matching patterns with many attributes and gradually reduces the number of attributes through learning. A general-to-specific method begins with detailed matches and shifts toward more general cues.

A *pattern composition* (PC) method constructs two levels of structure. First, the patterns are organized into several disjoint sets, and then activated patterns are combined by the second level (Figure 2). For instance, at the pattern level, the pattern `color = white`, `fat-ratio = high`, and `habitat = cold` would contrast with `color = bright`,

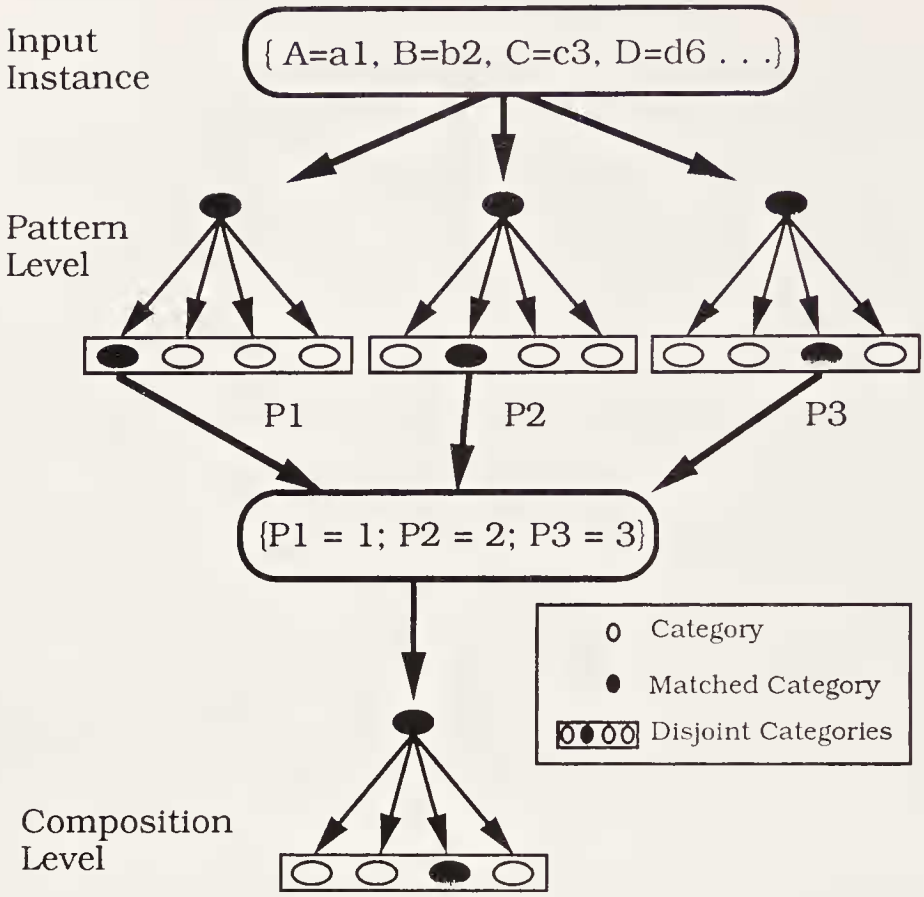

Figure 2. The PC processing architecture, which classifies an instance to one pattern from each disjoint set at the pattern level, and then combines the matched patterns by choosing one composition pattern.

`fat-ratio = low`, and `habitat = temperate`, as well as other possibilities.[4] Significantly, PC methods create several of these disjoint sets of patterns, each set being roughly orthogonal and concentrating on different attributes. For instance, one set of patterns might link particular adaptations (`fat`) to different habitats (`far-north`) and another orthogonal set might relate aspects of an animal's relationship to humans (`pet`, `pack-animal`, etc.).

After some patterns have been activated, predictions about attribute values are made by combining the predictions of those patterns. Making predictions about attribute values for reindeer might combine a pattern describing animals living in the `far-north` and another orthogonal pattern describing `pack-animals`.

4. In general, PC methods could recursively divide a pattern into mutually exclusive subpatterns, but this study omitted the capability so that only the IP methods could explicitly vary the amount of evidence of representational units.