Expressivity and learning geometric concepts across levels of expressivity

Sophia Ray Searcy

The University of Louisville

Expressivity and learning geometric concepts across levels of expressivity

Introduction

This proposal introduces a new term to the concept learning literature: expressivity. When applied to models of concept learning, expressivity is a measure of the concepts that can be represented by each model. Some models are able to accurately express concepts that other models simply cannot, and a formal notion of expressivity permits comparisons along this line. Expressivity can also be applied to individual concepts; we may speak of the expressivity required in order to accurately represent a specific concept (e.g. the concept ‘all X are Y’ requires a model that can at least represent quantities). Expressivity has been used informally to argue against certain models of concept learning like the classical model and the perceptron. I provide a more formal analysis of the expressivity of several concept learning models and related experiments.

This discussion of expressivity then leads to a gap in the concept learning literature that concerns the remainder of the proposal. The idea is that some common concepts have structure at multiple levels of expressivity. A square can be defined at the object-level by defining a rule that determines which objects are included in the concept: a square is an object has four sides, opposing sides are parallel, all sides are the same size, all interior angles are 90°. However, a square may also be defined in terms of its relationship to other geometric concepts: all squares are rectangles, all squares are rhombuses, all squares are parallelograms. These define the concept at different levels of expressivity; they are two different *kinds* of structure. For instance, the object-level definition is applied to individual objects in isolation whereas the relational-level definition applies to objects but refers to other concepts as well.

Here, I argue that one critically important difference is that of expressivity: the representational-level concepts require a more expressive representation than the object-level concepts. This is reflected in the fact that the object-level concepts have a limited domain covering observed examples while relational-level concepts, such as “all squares are also parallelograms" necessarily have an unlimited domain. In a technical sense, this means that the relational-level definition requires quantifiers (.e.g "for all X, Y is True"); in other words, it cannot be expressed in finite terms at the Boolean level. Studies of concept learning have explored the learning of concepts that are analogous to each level individually but it seems that no prior study has investigated simultaneous learning at both levels.

I present experiments that investigate how learning at the object-level might be used to promote learning at the relational-level, i.e. how to set up learning so that subjects learn at multiple levels of expressivity simultaneously. The first experiment made use of a rational teaching model to select examples to teach one of four geometric concepts. In this case there appears to be no learning at the relational-level and no improvement over a condition that provides no teaching information whatsoever. The second experiment added a sequential training phase at the object-level for training and found limited improvements in performance at the relational-level. Finally, I propose an experiment that will provide insight into the factors that promote learning at multiple levels of expressivity. Mathy and Feldman (2009), Eaves and Shafto (2014) have both found that learning is promoted by ordering examples to respect the underlying representation. Experiment 3 then tests a training process where subjects might learn object-level concepts directly through the examples while the presentation order promotes learning at the relational-level.

Expressivity

Concepts are critical parts of thought; the building blocks of our most important decisions. People spend a great deal of time and effort learning concepts throughout life. Teachers are responsible for communicating concepts to students. A longstanding project of psychologists has been to understand the process whereby people learn concepts. To be of value, this understanding should help people learn and teach the kinds of concepts people actually use. But the concepts that people use are often different in several ways from those that are the subject of concept learning research.

Explanations of concept learning take the form of a model, a formal description of the process of acquiring and using a concept. These models include a representational system, which is what defines the candidate concepts that can be learned.[1] There are several key questions in the concept learning literature, a subset of which is included in table 1. Probably the most prominent of these questions is that of representation: which representation best captures how concepts are used? Are concepts made up of a set of necessary and sufficient features, rules (e.g. Goodman, Tenenbaum, Feldman, & Griffiths, 2008), prototypes (e.g. Rosch, 1973), exemplars (e.g. Medin & Shoben, 1988), or networks of weighted connections (e.g. Love, Medin, & Gureckis, 2004)?

A critical difference between representational systems is the set of candidate concepts that each can generate. If the representational system used by a model is incapable of generating concepts of the form "_____ or _____" then the model cannot explain how someone might learn that an eligible presidential candidate is "a natural born citizen, *or* a citizen of the United States, at the time of the adoption of this Constitution" (emphasis mine). Here I focus on the question of expressivity, which concerns the kinds of concepts to which the models of concept learning can be applied. A common way to construct concepts is to use Boolean logic, where features are either True or False and concepts categorize examples made up of these features. In such a world, the number and combination of possible examples and concepts is fixed. If there are few enough features, learning can be accomplished by searching the space of concepts exhaustively. These properties have made this a convenient and fruitful test bed for models of concept learning (e.g. Bruner, Goodnow, & Austin, 1956; Rosch & Mervis, 1975; Smith & Medin, 1981; Feldman, 2000).

However, some concepts cannot be expressed in such a world. The concept of the

[1] In general, there is a clear relationship between a model and the representational system used by the model, and indeed when discussing models I will generally separate them according to the representational systems used. In these cases, when I refer to the expressivity of a model, it is specifically the expressivity of the representational system used by the model.

number “three”, for instance, seems to require examples to be understood as sets of objects rather than the individualized objects of Boolean logic. The system of concepts we think of as “arithmetic” seems to require a representation of sets of sets see, Searcy and Shafto (2016). This then raises the question of how the understanding of the learning of certain concepts should be informed by models that *cannot be applied* to those concepts. In other words, how can a science of the learning of algebra or physics be informed by a model that itself could never, even in principle, learn algebra or physics?

The very first step is to clarify the expressivity of many of the most used models of concept learning. The difficulty here is that different models have made use of different representational systems so that the relationship between the sets of concepts that each is able to express is not immediately clear. I begin this step in the Modeling section below. A second step is to consider how the existing empirical evidence in the concept learning literature informs questions of expressivity. Finally, I will conclude by discussing the problems that remain as well as the prospects indicated by this work.

Preliminaries

What is expressivity?

Expressivity refers to the ability of a model to learn, use, or otherwise include a concept. The expressivity of a model can be likened to the set of concepts that the model can (accurately) express. For representation languages, expressivity is simply the set of all sentences (concepts) in the language.

The classical model of concepts provides a quick illustration of the importance of expressivity. For instance, in its simplest version, the Classical model stipulates that each concept is represented by a set of features that are each necessary and jointly sufficient. A well-worn example is that of the concept “bachelor” defined as an unmarried male. Both unmarried and male are necessary features and, if both are present, together sufficient to consider someone a bachelor. There are many concepts which cannot be expressed within the Classical model, though. Consider a strike in baseball: a pitch is considered a strike if EITHER the batter fails to swing at a ball pitched into the strike zone, OR the batter swings at any pitch and misses, OR the batter hits the ball into foul territory and there is no more than 1 strike in the existing count. Because the description includes “ORs”—often called disjunctions in logic—no single feature is necessary in order for a pitch to be a strike. The batter might swing at the ball or not, might strike the ball or not, the ball might pass through the strike zone or not and still be considered a strike. Additionally, no set of features are jointly sufficient (i.e. their presence would guarantee the application of the concept). Thus the Classical model can be said to be incapable of expressing the concept of strike in baseball (in fact, any concept that requires a disjunction is inexpressible in the Classical model).

In general, more expressive models include all concepts of less expressive models, meaning that the corresponding set of concepts for less expressive models is typically a proper subset of the set of concepts corresponding to more expressive models. Expressivity can thus be thought of in terms of a hierarchy of expression, with the least expressive models at one end and the most expressive at the other. The major milestones in expressivity are presented in table 2. At the level of Boolean logic, models can express concepts that apply to examples made up of Boolean features (i.e. the feature may take on the value of either True or False). Much of the models of concept learning until recently have been limited to this level. First-order logic includes quantifiers over objects in sets which permit the expression of quantity concepts as well as comparisons. The next level, second-order logic, includes higher-order quantifiers such as quantifiers over sets or concepts. These are necessary for expressing concepts like the natural numbers or arithmetic. Finally, at the most expressive end of this list are universal representations (Turing, 1937; Abelson, Sussman, & Sussman, 1996) that are capable of expressing any computable function.

Notation

There are a few points of notation that should be cleared up before use.

A representation language is a representational system that can be defined in some formal language. A common means of defining a formal language is using a context-free grammar (CFG), a grammar defined by production rules that govern the abstract structure of the sentences in a language originally developed by Chomsky (1956). A CFG is a system for generating formulas, or 'sentences' that compose a language. A CFG consists of: 'variables' which are symbols that *do not* appear in the produced sentences and one of which is the start symbol, 'terminals' which are the terminal symbols that *do* appear in the sentences, and the production rules that link variables to terminals. Variables are often capitalized and terminals are often lowercase to distinguish between them. An example of a small CFG is,

$$\begin{aligned} S_0 &\to S_0 \wedge F \mid F \\ F &\to f_0 \mid f_1 \mid \cdots \mid f_n \ , \end{aligned} \tag{1}$$

where n is the total number of possible features. To see how this grammar produces the sentence $c = f_1 \wedge f_2$, begin with the start symbol S_0, then use the first production $S_0 \to S_0 \wedge F$ (the vertical lines separate the possible productions for a given variable). The second S_0 can then be removed using the second production $S_0 \to F$ to give us $F \wedge F$. Finally, we can pick productions from the abbreviated bottom row to give $f_1 \wedge f_2$. The initial "$c =$" is omitted because that is the same for every representation language used here.

The grammar in eq. (1) corresponds to a common interpretation of the classical model of concepts: that a concept is represented by the properties (or features) necessary and jointly sufficient for its application. In Boolean logic, this description is captured by using the AND operation ($\wedge$) which returns True if both inputs are True and False otherwise. If f_1 and f_2 are each necessary, and together sufficient, for the application concept c, we can represent the concept with $c = f_1 \wedge f_2$. When this relationship is extended to larger groups of features, the necessary and jointly sufficient properties still hold.

Many models of concept learning use a representational system that can be equivalently defined in a representation language. Because representation languages provide a useful tool for discussing the expressivity of a representation, my approach will be to present such models using a representation language where possible.

The representation languages discussed here include a few symbols and operations which may not be familiar to the reader. I use $\{0, 1\}$ for the Boolean labels corresponding to False and True respectively. 'Boolean logic' is a representation language in which all the variables and operations may only output one of the Boolean labels. First-order Logic (FOL) is like Boolean Logic with a few additions: Quantifiers are operations that deal with number in a set of objects. The existential operator—"$\exists$" in a formula like $\exists o \in \mathcal{O}\ p(o)$, read "there exists an o in $\mathcal{O}$ such that $p(o)$"—applies an expression to every item in a set and returns 1 if the expression is true for one or more item. The universal quantifier—"$\forall$" in a formula like $\forall o \in \mathcal{O}\ p(o)$, read "for all o in $\mathcal{O}$ such that $p(o)$"—operates similarly by applying an expression to every item in a set but instead requires the expression to be true for *all* items.

Modeling

When surveying the modeling literature I follow a common trajectory (e.g. Smith & Medin, 1981; Laurence & Margolis, 1999; Murphy, 2002) by beginning with the classical model of concepts, followed by probabilistic/prototype models, and exemplar models. This provides a convenient narrative for introducing rule-based models that have found success in recent years. The process of assessing the expressivity of different models requires precise specification of a representational system (ideally a representation language). This means that in some cases, where the description of a model is incomplete or unclear about representational details, assumptions will have to be made. The analysis that follows is thus only intended to pertain to the published descriptions of each model together with the clearly indicated and hopefully reasonable assumptions made.

Classical model

The classical model proposes that each concept is made up of a list of necessary and jointly sufficient features. For example, the concept "bachelor" might be defined using a classical model consisting of the features "unmarried" and "man". Any example that satisfied both of those features would then be categorized as a "bachelor" and any example that fails either or both would not be. Using a logical formula, this looks like $c_{bachelor} = f_{unmarried} \wedge f_{man}$. The representational system inherent in the classical model is simply a series of conjunctions (described formally in eq. (1)).

Because the classical model has one means, conjunction, for combining features in order to represent a concept, the expressivity of this model is limited. An illustration should suffice to explain the problems this creates. Imagine a list of features: $\{f_{color}, f_{shape}\}$. We might assume that a classical model can combine such features with their possible values to create Boolean features, such as $(f_{color} = v_{blue})$. Consider, then, the number of concepts in this simple world. If we restrict the universe to objects with these two features, and restrict each feature to three possible values, then there are $3^2 = 9$ different examples and $2^9 = 512$ different concepts in this universe. How many of those concepts can be represented using a classical model? Just $4^2 = 16$ because there are two features each of which can take on three values or be omitted. In such a world, the classical model has coverage of about 3% of the possible concepts. The model has a concept for "red objects," "blue and square objects," and "circle objects" but no concept for "objects that are not blue"[2] nor "either red or not square but not both." And the proportion of the total concepts that can be represented quickly decreases as either the number of features or the

[2] The problem here is that no necessary feature exists. The use of negation, discussed further in the Expressivity remedies section, would allow the model to construct a new feature "NOT(blue)" that would then satisfy this concept.

number of values for each feature increases. To put it plainly, in any realistic context with a large number of complex features, the share of possible concepts that can be represented by the classical model diminishes to nothing.

Expressivity remedies. One means to address the lack of expressivity is the inclusion of negation, the operation that simply flips 0s to 1s and vice-versa. Such a model might add a third rule to the classical model's grammar,

$$\begin{aligned} S_0 &\to S_0 \wedge N \mid N \\ N &\to \overline{F} \mid F \\ F &\to f_0 \mid f_1 \mid \cdots \mid f_n \ . \end{aligned} \tag{2}$$

This allows the model to represent concepts like "objects that are not blue." With negation, the model can represent 9% of the concepts in the universe of two features with three values described above. This isn't a very large improvement, and the proportion still declines to zero in the limit, but it sets up a promising possibility that will have to wait for the next fix.

The other improvement is the inclusion of compositionality. Composition roughly corresponds to building novel concepts out of existing ones. Composition is thought to be an important part of concept representation (Goodman et al., 2008) but it is omitted from many models for a variety of reasons, chief among them that allowing composition makes the set of sentences that can be produced by the representation language infinite and thus impossible to completely enumerate. The following grammar adds a list of m concepts to the set of terminals that can take the place of features,

$$\begin{aligned} S_0 &\to S_0 \wedge N \mid N \\ N &\to \overline{F} \mid F \\ F &\to C \mid f_0 \mid f_1 \mid \cdots \mid f_n \\ C &\to c_0 \mid c_1 \mid \cdots \mid c_m \ . \end{aligned} \tag{3}$$

In essence, this change allows any concept to take the place of a feature in another concept.

If compositionality is added to the classical model alone (without negation) it provides no effect, as composing a series of conjunctions with another series of conjunctions results in nothing more than another series of conjunctions. However, when combined with negation we arrive at a representation that has full coverage of the Boolean concepts. To see why, consider the concept c_1, “objects that are not blue and not square.” The Classical model can learn this by inserting the negated features into a single conjunction.

Only when the Classical model is modified to include both negation and compositionality can it represent c_2, “objects that are not (blue and square).” The difference is subtle, but because “not” operates outside of the conjunction in c_2, any object that is “not blue” is in the concept regardless of the shape. The c_2 concept has the form of the NAND operation, an extremely important operation in digital logic design because of its ability to represent all logical functions and convenient physical implementation in transistors (Roth, 2013). Because NAND can be combined to make any logical function and the modified Classical model includes composition, this means that such a model would be capable of representing all Boolean concept.

The problems with the Classical model and the two-part solution proposed illustrate a few things about the expressivity of models of representation. The improvements that would enable expressive representation here are orthogonal to other purposes of the representational model. If nothing else, the use of necessary and sufficient features is a critical part of the Classical model, but the remedies necessary to allow expressivity completely remove this—a Classical model that can freely use negation and composition can express concepts with no necessary or sufficient features whatsoever. In fact, such a modification allows the expression of all possible Boolean concepts.

Prototype models

The representational system used by the Classical model is a simple list of features. The next model, called the prototype or probabilistic model, includes the addition of a weight for each feature. When determining if an example is in a prototype concept, the model adds together the weights for each matching feature. If the sum of the weights exceeds some threshold (possibly zero) then the example is in the concept.

Where the various models in this class differ is in how that representational system is *used*. For instance, Collins and Loftus (1975) developed the Spreading Activation model which uses the prototype representation of weighted features. When categorizing an example, the spreading activation model compares the example with the concept along random features, adding the weight for each feature where the concept and example match and subtracting the weight for mismatches. Whenever the combined weights exceed a positive threshold, the example is determined to be a match; whenever they become more negative than a separate negative threshold, the example is determined not to be in the concept.

Another notable model in this category is the Perceptron, originated by Rosenblatt (1958). While more modern connectionist models surely do not belong in a discussion of prototype models, the Perceptron, despite being the precursor to such connectionist models, certainly does. The representation used by the Perceptron model is two variables, an n-length vector of weights w and a bias b, where n is the maximal number of features. This representation is used for classification of an example x by calculating $H(w \cdot x + b)$, where $H()$ is the Heaviside function that outputs 1 if the input is positive and 0 otherwise.

In spite of its relationship to other models, the Perceptron functions in a fundamentally similar way to the prototype model. A Perceptron works by adding together the weights for every true feature (i.e. where $x_j = 1$), just as prototype models add together weights for every feature that matches between the candidate and the concept. Rather than a list of both features and weights, the Perceptron only represents a list of weights, but because the weight for any excluded feature can be set to 0, these are equivalent representations.

The similarity between prototype models and the Perceptron means that many analyses of the Perceptron model can be expected to generalize to the remainder of models in this class. Importantly for our purposes, understanding the Perceptron as a “linear classifier” clarifies the limits of the model’s expressivity. The Perceptron can only learn linearly separable categories and is incapable of learning some simple logical functions like XOR (the logical function for ‘either a or b but not both’) (Minsky & Papert, 1969). This will be true of all models that use a weighted list of features for representation. While Smith and Medin (1981) argue that prototype concepts are capable of learning disjunctive concepts (e.g. a concept ‘furniture’ which is a disjunction of, among other things, ‘chair’ and ‘rug’), this is only true of disjunctive concepts that are linearly separable in terms of the input features.

Like the Classical model, this limited expressivity can be solved in several ways, but each of those ways would fundamentally alter the representation of the models. For instance, composite features can be made by using operations to combine features. Highly correlated features might be learned in conjunctions, so that a ‘flies and has feathers’ feature might be learned instead of learning ‘flies’ or ‘has feathers’ individually. Alternatively, complex combinations of features can be used that would permit a concept that was not previously linearly separable to be linearly separable in a feature space that uses the new features. The danger here is that by allowing additional assumptions to do this much work, the model might no longer be recognizable as the prototype model.

Exemplar models

Exemplar models add another layer of complexity to concept representation. This class of models uses a system based on the prototypes from before (i.e. a list of weights and features) and combines these *exemplars* into a set or list. This approach is closely associated with the work of Rosch and collaborators (Mervis, 1980; Rosch & Mervis, 1975), though Rosch has stated explicitly that this work was never intended to produce a formal model of representation (Rosch, 1999). In spite of this, Smith and Medin (1981) developed a representational model that captures some of the notions implicit in the prototype literature, called the Best Examples model and additionally presented a somewhat different alternative called the Context model.

The Best Examples model categorizes an example as a member of a target concept, C, by comparing the example to objects in the concept and a contrasting concept, $\overline{C}$, which might take the form of the union of examples in various non-C concepts. Given a target example, x, that is to be categorized, the model draws objects, o_i, from the set of all objects (both C and $\overline{C}$) according to how similar each object is to the target using a similarity metric discussed below. The model keeps track of the number of objects that come from C and $\overline{C}$ until one of them reaches a specified number of objects.

Similarity is the main distinction between the Context model and Best Examples as the Context model uses a multiplicative similarity metric. The Best Examples model uses an additive similarity metric, meaning that the chance of of drawing an object o_i given an example, x, is proportional to the sum of the weights of the matching features. Under the Best Examples model, consider one object o_9 matches example x on 9 out of 10 features and another o_10 that matches on all 10 features. If all features are weighted similarly, this mismatch reduces similarity by 10%.

The Context model, on the other hand, uses a multiplicative similarity measure. In place of weights the Context model has a cost $c_f < 1$ for each feature. Similarity between o_i and x is then determined by multiplying the cost of each mismatched feature. Consider how the above example with o_9 and o_10 changes when a multiplicative measure is used. Assuming that the cost for each feature is $c_f = 0.1$ then a single mismatched feature reduces similarity by 90%.

For expressivity, the important point is that the Context model is able to express all

Boolean concepts and this is likely true of the Best Examples model as well. This is somewhat trivial to demonstrate for the context model. The cost for each feature can be set to $c_f = 0$ so that the only objects with a nonzero chance of being drawn are those that exactly match the target example. Any concept can then be constructed by adding the entire positive extension as the exemplars of C and the entire negative extension as the exemplars of $\overline{C}$. Using this, any concept that can be extensionally defined (thus, any finite Boolean concept) can be represented using the full set of exemplars in the extension. While this demonstration doesn't translate perfectly to the Best Examples model, it suggests that such expressivity likely also belongs to the model.

Rule-based models with limited expressivity

Rule-based models typically utilize some form of the representation length approach (Feldman, 2000; Kemp, 2012; Goodwin & Johnson-Laird, 2011) In this approach, a model consists of: i) a representation language, ii) a method for computing the length of formulas in the language, and iii) a method for scoring the length of the shortest consistent formula given a concept. The language is often generated by a formal grammar. The length function typically counts the number of operations in each formula and assigns a (not necessarily uniform) cost for each of them. The minimization process is often exhaustive for most limited representation languages (in other words, if one can generate a set of all consistent formulas, that set is guaranteed to include the minimal formula). But this approach is generally intractable so estimation techniques are used in many cases.

The rule-plus-exception model (RULEX) of Nosofsky, Palmeri, and McKinley (1994) was one of the earliest concept learning models that explicitly built complex concepts (i.e. concepts that include multiple operations) out of rules. In RULEX, concepts are built as a disjunction of simple rules and one or more exceptions. This is implemented as a decision tree equivalent to the following grammar,

$$\begin{aligned} S_0 &\to (RE_n)\, E_p \\ R &\to F \mid R \wedge F \\ E_n &\to \wedge\overline{(R)}E_n \mid \epsilon \\ E_p &\to \vee(R)E_p \mid \epsilon\ . \end{aligned} \tag{4}$$

This grammar uses the F production from eq. (1) to generate features. Presenting the RULEX model as a grammar makes it somewhat easier to see its relationship to the Classical model and exemplar models. Like the Classical model in eq. (1), the rule production, R, generates simple conjunctive rules. And like exemplar models, RULEX incorporates observed positive and negative examples through the exception productions, E_n and E_p. These two parts function in a model that probabilistically adds complexity (e.g. either more features to the rule or new exceptions) at each step.

Together, these components have two effects that are of interest. First, the addition of an unbounded number of exceptions means that RULEX is capable of expressing all Boolean concepts. To show this, consider that all Boolean concepts include one set of negative examples and one set of positive ones. One can see that RULEX can represent any concept by starting with an empty R production and adding all positive examples as positive exceptions and negative examples as negative exceptions. This would then accurately represent the concept and the R production could be used to simplify the representation, so that fewer exceptions are needed. The second effect is that the probabilistic construction of formulas places a soft limit on representation length. Each exception and rule is added with a certain probability < 1 so that the more complex the concept, the less likely it is to be produced. The RULEX model demonstrates that rule-based systems can achieve a balance that includes both an expressive model and a plausible processing mechanism.

Feldman (2000) designed a model that explains the difficulty of learning a large number of concepts based on two effects: Boolean complexity and parity. For each concept, Boolean complexity is derived from the representation length approach for the language of unstructured Boolean formulas,

$$
\begin{aligned}
&S_0 \to (S_0) \wedge (S_0)\\
&S_0 \to (S_0) \vee (S_0)\\
&S_0 \to \overline{(S_0)}\\
&S_0 \to F \ .
\end{aligned}
$$

The minimal formula was found using a heuristic method (finding the absolute minimal Boolean formula is an NP-Hard problem, meaning that it quickly becomes infeasible to compute for large inputs). Parity is determined by the fraction of examples that are positive or "in the concept". So concepts with *up* parity have more negative examples than positive, like $f \equiv a \wedge b$, and concepts with *down* parity have more negative examples than positive, like $f \equiv a \vee b$.

Unlike the RULEX model, Boolean Complexity uses rules that need not have any correspondence to the examples observed. Boolean Complexity additionally uses a different notion of complexity—rather than probabilistically generating formulas for a given concept, it finds the minimal ones—that has a similar effect. Complex concepts can be represented while still being less likely (here more complex).

Goodwin and Johnson-Laird (2011) designed a model based on mental models (Johnson-Laird, 1983), an important theory of concepts that had previously not been applied to concept learning for logical concepts. Goodwin and Johnson-Laird use a representation length approach with the following representation language,

$$
\begin{aligned}
&S_0 \to S_0 \otimes S_0 \mid S_1\\
&S_1 \to S_1 \wedge S_1\\
&S_1 \to F \mid \overline{F} \ ,
\end{aligned}
$$

where $f_0 \otimes f_1$ is the XOR operation. The exclusive-or operations separate the "mental models", so that the concept $f \equiv (f_0 \wedge f_1) \otimes \overline{f_1}$ would contain two models, $f_0 \wedge f_1$ and $\overline{f_1}$. The number of models in the minimal formula for each concept (and equivalently the number of $\otimes$ operations plus 1) is used to set the complexity of each formula.

The Rational Rules model of Goodman et al. (2008) uses disjunctive normal form (DNF) grammar and applies this grammar with a creative measure of complexity and minimization strategy. The grammar is as follows:

$$\begin{aligned} &S_0 \rightarrow S_0 \vee S_0 \mid (S_1) \\ &S_1 \rightarrow S_1 \wedge S_1 \\ &S_1 \rightarrow F \mid \overline{F}\ . \end{aligned} \tag{5}$$

The complexity is calculated as the probability of the formula given the set of labeled examples. The precise details of the probability calculation are not important for current purposes but the gist of it is. The probability depends on how often each variable (i.e. each line in eq. (5)) is used for each production (i.e. the options separated by |). So a formula that reuses many f_1 terminals but few others will be more probable than a formula with the same number of total productions spread out more evenly.

Models with greater expressivity

The rule-based models discussed above (with the possible exception of the Classical model) are each capable of expressing all Boolean concepts. However, as discussed previously, Boolean logic is at the bottom of the hierarchy of expressivity for logical languages (see table 2). I now turn my attention to models designed explicitly to apply to concepts that require more expressive representations.

Kemp (2012) developed a rule-based model that accounted for concepts that can only be expressed in first-order logic by incorporating quantification into DNF at different levels. Object quantification is perhaps more common in the concept learning literature. It allows for statements that apply a certain sub-formula to all objects in a set, such as ‘for all objects, the color is blue.’ Feature quantification, on the other hand, allows statements of the form ‘there exists a feature with value 1’ which are then applied to objects. Kemp additionally designed a set of models to permit a comparison between these two quantification types.

Piantadosi, Tenenbaum, and Goodman (2016) used a lambda calculus-based representation system to implement several different representational languages including an unstructured Boolean logic language (Feldman, 2000), DNF (Goodman et al., 2008), and several others. Instead of finding the minimal formula consistent with each concept, they used a probabilistic sampling method similar to Goodman et al. (2008) to generate formulas for a set of observed examples. The full setup allowed for the comparison of several candidate representation languages by substituting in different grammars. Piantadosi et al. then used a second set of representation languages that included several different methods for accounting for more expressive representations. Each of these were implemented by adding a new set of productions to the grammar generating the language and, as such, these methods could be mixed and matched. In addition to quantification, which we have already discussed, this set includes second-order predicates, relational predicates, a one-or-fewer predicate, and small cardinalities (e.g. ‘there exists exactly one/two/three objects such that...’).

Empirical work

Shepard, Hovland, and Jenkins (1961) performed a set of classic experiments that test the difficulty of learning six artificial concepts—meaning the concepts were defined entirely in the lab. I will refer to these with the notation SHJ 1 - SHJ 6.) Using such concepts allowed the researchers to precisely measure the relationship between inputs (the explicitly defined concepts themselves) and behavioral outputs (learning rates over a number of trials) in a way that is not possible using pre-existing concepts. The experiments were able to control for factors such as the ratio of positive examples to negative examples and the set of relevant features. These factors are difficult to measure, much less control for, in concepts not artificially defined. The main finding is that the six concepts follow a consistent ordering in learning difficulty across different tasks. This ordering has since been replicated several times (e.g. Feldman, 2000; Kemp & Regier, 2012; Crump, McDonnell, & Gureckis, 2013).

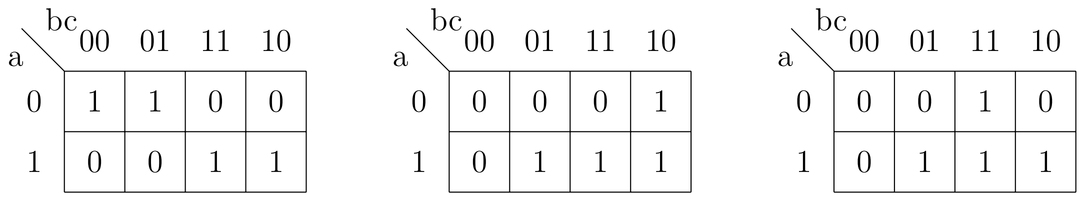
These results provide indirect insight into questions of expressivity as well. As stated before, there are several concepts that simply cannot be represented by the Classical model and prototype model. Of the six used by Shepard et al. (1961), only one of them can be accurately expressed by either the Classical or prototype models while all were learned by participants above chance. However, this is not clear evidence of failure for less expressive models. There remain other explanations. One possibility is that when participants are asked to learn a concept for which their model cannot express an accurate description, they instead learn the most accurate description that can be expressed by the model. Shepard et al. did not test this possibility; indeed no model's predictions were compared to the results. Yet this set of experiments does provide a template for more direct tests of models of concept learning (and hence their expressivity) by systematically measuring learning outcomes on artificial concepts.

Feldman (2000) does just that by expanding the analysis to 76 distinct concepts across six *families*[3]. Feldman showed that for this set of concepts, two factors are good predictors of the difficulty of learning. First is Boolean Complexity, which is the length of the shortest formula in Boolean logic that is consistent with the concept. Some concepts are very compressible in Boolean logic and thus have short formulas, such as $c = a$, while others are much less compressible, such as $c = (a \wedge b) \vee (a \wedge c) \vee (b \wedge c)$. The second factor

[3] A family was defined as a group of concepts over the same number of features and that have the same number of positive examples. For example, the original six concepts in Shepard et al. (1961) compose a single family of concepts over three features with four positive examples.

is parity—concepts with more positive examples than negative examples have a “down” parity while concepts with more negative than positive have an “up” parity. Subjects tended to learn “up” concepts much more easily than down and this factor was largely orthogonal to Boolean Complexity.

Two later analyses (Goodwin & Johnson-Laird, 2011; Vigo, 2009; Feldman, 2006) have fit newer models, each based on the idea of logical complexity (or, equivalently, compressibility), to the same data, explaining up to 57% of the variance. This idea, that measures of learning are predicted by logical complexity of a concept, has become increasingly important as the concepts under study have required more expressive models. Feldman (2000) suggests that the limited scope of concepts under study made it appear that other factors orthogonal to complexity are the primary factors that determine concept learning difficulty. Complexity was suggested informally by some (e.g. Shepard et al., 1961) and used indirectly in at least one analysis (Neisser & Weene, 1962), but it wasn’t until the turn of the century that results for an expanded set of concepts supported complexity-based models. Together, the empirical results from Feldman (2000) and the later analyses provide strong evidence in favor of the use of more expressive models of learning as well as studying concepts that require more expressive models (only six of the 76 concepts could be expressed by the Classical and prototype models).



a \ bc	00	01	11	10
0	1	1	0	0
1	0	0	1	1

(a) SHJ 2: $c = (a \wedge b) \vee (\overline{a} \wedge \overline{b})$

a \ bc	00	01	11	10
0	0	0	0	1
1	0	1	1	1

(b) SHJ 3: $c = (a \wedge b) \vee (a \wedge c) \vee (b \wedge \overline{c})$

a \ bc	00	01	11	10
0	0	0	1	0
1	0	1	1	1

(c) SHJ 4: $c = (a \wedge b) \vee (a \wedge c) \vee (b \wedge c)$

Figure 1. Three of the six concepts used by Shepard, Hovland, and Jenkins (1961). These cannot be represented accurately by the Classical or prototype models.

In one experiment, Kemp (2012) tests several variations on the classic Shepard et al. (1961) paradigm. Kemp and Regier showed that subtle differences in how features are represented, such as whether each feature is additive (i.e. either on or off) or substitutive (i.e. either A or B or C), has an effect on the number of possible concepts and the complexity of each. He also proposed that in some of these cases the ability to quantify over features or objects would be a sensible method for representing these concepts.

Indeed, Kemp found that including quantification resulted in better predictions of learning difficulty in these concept spaces but only for object quantification. One interesting detail is that none of the concepts used required quantification in order to represent them, i.e. all could be accurately expressed by Boolean logic. So, even in cases where concepts can be represented without quantification, it seems that a RL using quantification provides a better prediction of how subjects learn and use concepts.

Piantadosi et al. (2016) designed experiments that tested several concepts that, in terms of the intension[4], require a more expressive language to represent. The examples in these experiments were individual objects in a set of objects and the use of sets allowed intensions that include quantification, such as 'one of the largest'. In one experiment, the authors took the best-performing RL without quantification and added the several quantification methods discussed previously. They found that models performed much better when quantification methods were added, even accounting for the number of free parameters. The quantification methods that performed best were those conventional to FOL with the addition of a one-or-fewer quantifier.

One finding from this series of experiments is that several concepts that could not be represented in Boolean logic could be learned very accurately, indeed more accurately than most Boolean-compatible concepts in the experiment. For example, the concept "one of the smallest" (true of any object in the set whose size was the smallest of the set) cannot be

[4] An intension is a description of a concept, such as a formula or rule. This is as opposed to an extension which defines a concept according to input and output.

expressed using Boolean logic but was learned more successfully than most concepts that could be expressed using Boolean logic. A Boolean representational system can cheat, in a sense, by learning True and False for a given set of examples in such a concept, but the resulting formula will not always generalize. Piantadosi et al. also were careful to make strategies like this less useful by setting up the experiment so that each example was a new inductive inference conditioned on the previous examples.

Discussion

Expressivity is an important consideration in the evaluation of models of concept learning. Many models that have been used to explain concept learning lack the machinery necessary for application to concepts routinely taught to children, including many topics in early math. It therefore remains uncertain how the existing concept models come to bear on such concepts that require more expressive models.

The first step is to assess and document the level of expressivity of major concept models. The Modeling section of this paper has begun that work. Important early models, like the Classical model, prototype model, and Perceptron, were shown to be incapable of expressing even the full space Boolean logic concepts. Many modern models, including exemplar models and several rule-based models are capable of expressing Boolean concepts but are no more expressive than that. There are also a few rule-based models that manage to express first-order logic concepts and even beyond.

Remarkably, Piantadosi et al. (2016) have developed a framework for testing against empirical data any representational system that can be defined using a representation language. This, or something like it, seems to be an ideal test bed for evaluating models of concept learning going forward. Current open questions include: in which cases people's learning is best explained by more expressive models as opposed to less expressive ones, and the ways in which each increase in expressivity is accomplished. Currently these questions are treated as modeling decisions but Piantadosi et al. have shown how decisions like these can be informed by the data instead.

One notable omission from this assessment of concept models is the group of connectionist models more complex than the Perceptron. These were omitted for practical purposes, as it seems particularly difficult to assess the expressivity of connectionist models. It is possible in theory to construct a universal computer out of neural networks (Siegelmann & Sontag, 1991) and the connectionist models that have been applied to concept learning studies (Kruschke, 1992; Love et al., 2004) are certainly capable of expressing the class of Boolean concepts, but where exactly these fall is less clear. This remains a topic for further study.

A second step is to evaluate the implications of the existing empirical work for the expressivity of models. This can mean several things. In some cases, experiments involving concepts that require more expressive models, like those from Shepard et al. (1961) and Feldman (2000) have been key factors in establishing effects, like logical complexity, that were not apparent when only concepts from less expressive models were used. In other cases, expressivity is tested more explicitly. Evidence from Kemp (2012) and Piantadosi et al. (2016) can inform the particular method of achieving certain kinds of expressivity: object quantification is more likely than feature quantification and people might be inclined to use a one-or-fewer quantifier in addition to the conventional quantifiers of FOL.

This work points to promising areas of future study. Until concepts that require even more expressive models (e.g. second-order logic or higher) are tested, it remains possible that there are important learning effects that are more difficult to discern using only less expressive concepts. In addition, it is worth further investigating the specific mechanisms people use to represent concepts that require more expressive representations. What factors lead people to learn using less or more expressive representations? How do people make the switch, say, from Boolean logic to first-order logic (or whatever equivalently expressive representations) and vice-versa? Are there ways to strategically encourage learning at one level of expressivity over others?

Then there are practical matters of constructing models with greater expressivity. The tractability of computational models, especially social learning, is of concern (Beal & Roberts, 2009; van Rooij et al., 2011; Searcy & Shafto, 2016) even for Boolean logic concepts. Generally speaking, as the expressivity of a model increases, so does the the space of possible concepts, and this space is one of the inputs to learning models. This problem is compounded in models of social learning, where one agent must predict the learning outcome (and the computations involved) of another. There are multiple ways to approach this problem and all are worth exploring. Probabilistic sampling methods (Goodman et al., 2008; Piantadosi et al., 2016) allow a model to make inferences about a potentially infinite space of concepts without exploring the entire space. Another plausible approach specific to social situations is that learners might use inferences about cooperation to bypass steps that would otherwise be computationally expensive (Searcy & Shafto, 2016).

Expressivity is not just a matter to its own. In order to better understand how people learn concepts, we should seek to better understand matters of expressivity in the models we use to explain learning. Expressivity is one of the many guideposts informing decisions about how models should be constructed, which concepts should be tested, and which questions should be investigated. Considering expressivity when designing models and experiments will help ensure that we build explanations that apply to the concepts we most care about.

Connection to experiments

For the remainder of this paper, I wish to argue for empirical work that might begin to answer the questions raised by the previous discussion of expressivity. Expressivity is an important aspect of concept learning models as well as the concepts used in experiments throughout cognitive science for several reasons. Most directly, the expressivity of a model is what describes the kinds of concepts the model can even speak of.

The experiments that follow consider expressivity in a novel way. This involves considering real concepts that have multiple definitions at different levels of expressivity. For the experiments, geometric concepts for quadrilaterals are used: 'square', 'rectangle', 'rhombus', 'parallelogram'. These shape concepts have structure at the object-level, where they can be defined in terms of the Boolean features f_e for 'all sides equal', f_r for 'all right angles', and f_p for 'all sides parallel' (see table 3). Each shape concept also has a relational structure that defines it's relationship to other shape concepts. For present purposes, I focus on the proper subset relation corresponding to the natural language description "all As are also Bs" (see fig. 2).

This relationship is of particular interest because of the perhaps surprising ways these levels of concept structure interact. Many models of concept learning assume that when learners are learning about a concept, they construct a formula or definition that can used to determine membership for the concept. Under such a model, any learner that builds the object-level concept definition for each of the four quadrilateral concepts would then be able to *deduce* the relational-level structure in fig. 2. However, one problem is that the relational structure is incompletely determined by any finite domain of examples, including those observed by subjects in the study that follows. Even if a subject sees 4 unique objects labelled 'square' and each such object is also labelled 'parallelogram', this is only evidence that some squares are also parallelograms.

Nothing that the subjects see precludes the possibility of a 5th such shape being labelled 'square' but not 'parallelogram'. The difference is that the object-level tests used in this and other concept-learning studies have a limited domain (i.e. those examples that are either observed in the training phase or queried in the test phases). Even experimental measures that use generalization at the object-level necessarily have a finite domain because only a finite number of examples can be given to a subject. This difference is helpfully understood in terms of expressivity: The object-level concepts are concepts with expressivity equivalent to Boolean Logic. Concepts at this level of expressivity can each be exhaustively defined through their extensions. The relational-level structure allows for concepts that use quantification, and the questions selected make use of universal quantification, requiring the concept definition to apply all examples in a given set, even an infinite set.

These experiments address the question of how it might be possible to teach these concepts in a way that respects the fact that each has a structure at two levels. Teaching at the object-level is done straightforwardly, using an iterated test with feedback design. The process for learning simultaneously at the object- and relational-level is less clear, however. While several studies have taught relational concepts directly (e.g. Kemp, Goodman, & Tenenbaum, 2008; Mathy & Feldman, 2009; Gentner & Markman, 1997) little is known about how learning might differ for concepts with structure at multiple levels of expressivity.

In order to accomplish this, I draw from research on how representation and the presentation-order of examples interact to affect teaching. Mathy and Feldman (2009) showed that a presentation order that respects the representation of concepts facilitates learning. Their experiment included concepts made up of a disjunction of clusters. They found that when examples were grouped by cluster and the largest cluster was taught first, participants learned concepts more quickly than when example order was randomized. Eaves and Shafto (2014) investigated a related process but for concepts with several different relational structures. They found that for three broad classes of relational concepts, participants learned the best when examples were presented in the order that ruled out simple alternative guesses about the target concept and minimized the number of symbols participants needed to remember. Experiment 3 makes use of this finding that examples ordered according to the underlying representation promote learning of that very representation. To do so, the first condition orders queries according to the concept label, such that first four queries are the same (e.g. “is this a square”) as each of the different shape examples are paired with the query. The second condition orders queries according to the examples, so the example that is strictly a rectangle would be paired with all four shape concepts before proceeding to the next example.

The experiments that follow seek to answer several questions about the relationship between these two levels of concept structure. Experiment 1 establishes a baseline for the change in relational-level accuracy when little or no learning occurs at the object-level. Experiment 2 is designed to measure the change in relational-level accuracy while ensuring that subjects experience measurable learning at the object-level under a generic or randomized condition. Finally, Experiment 3 repeats Experiment 2 but using two methods for ordering training examples that respect the structure of the concepts at the relational-level. Together, these experiments will allow the comparison of three distinct cases, one where little learning has occurred at either level, a second where learning has occurred primarily at the object-level, and a third where learning has occurred at the object-level in a way intended to promote learning at the relational-level as well.

One hypothesis, that learners transfer learning across levels, such as by deduction, leads to the prediction that learning at both object- and relational-levels will be similar between Experiments 2 and 3. If, however, the two levels operate distinctly, the Experiment 3 should lead to greater learning at the relational-level compared to Experiment 2, despite subjects seeing identical examples in both cases.

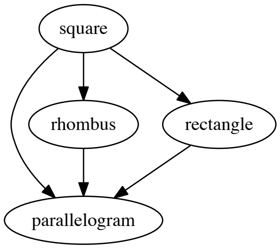


Figure 2. A directed graph representing the relational structure of the geometric concepts used. Each arrow from A to B represents the relation “all As are also Bs”.

Experiment 1

Experiment 1 consists of two parts: The pre- and posttests measure accuracy at the relational-level while the training phase provides object-level information in a batch learning presentation like those used in many concept learning studies (e.g. Feldman, 2000; Shafto & Goodman, 2008). By presenting information at the object-level and measuring learning at the relational-level, the experiment measures learning transfer across levels. Previous work demonstrating object-level learning through a similar training phase leads to the hypothesis that the teaching condition should lead to greater relational-level improvement than the baseline condition.

A technical note: Experiment 1 was conducted prior to the development of this hypothesis. All statistical tests and analyses are post-hoc.

Method

Participants. 213 total subjects participated in the experiment: 37 subjects in the baseline condition, 66 subjects in the rhombus x teaching condition, 63 subjects in the parallelogram x teaching condition, 48 subjects in the rectangle x teaching condition. All subjects were recruited through Amazon Mechanical Turk where the experiment was performed as an 'external question'. All subjects who completed the experiment were paid $0.25 and none who completed were excluded from the analysis.

Design. The primary design consisted of a teaching condition and a baseline condition. The teaching condition varied by target shape so within the teaching condition were three shape conditions: rhombus, rectangle, and parallelogram. The baseline condition did not vary by shape and thus only consisted of a single condition.

The dependent measure in each condition was the difference between the score on the posttest and pretest.

Procedure. The experiment was advertised as "Short and fun! A 2-3 minute game about geometry." Participants were asked to provide informed consent before proceeding to the pretest phase. The pretest and posttest phases consisted of the 12 nontrivial relational questions, e.g. “Are all rectangles also parallelograms: Yes, No” but excluding “Are all rectangles also rectangles: Yes, No”. The questions were randomized across participants but the same order was used for pre and posttest.

In the baseline condition, subjects are presented with a grid of 12 examples of quadrilaterals. Three are are strictly parallelograms, i.e. a parallelogram but not a rectangle, rhombus, or square. And three each are strictly rectangles, strictly rhombuses, and squares. Subjects were told to “Look at these examples of quadrilaterals” after which they proceeded to the posttest.

In the teaching condition, the same grid of quadrilaterals was presented but with three of the shapes highlighted in green. Subjects were asked to “click on the three shapes with boxes around them to learn whether each of them was a *parallelogram* or not” for each target shape. The positive examples were then highlighted in blue and negative examples were highlighted in red.

The three examples were chosen according to a rational model of teaching the target shape. The model selects the minimal set of examples to teach the target concept and adds additional examples if that number is less than three so that the conditions are roughly matched. The examples for each shape are as follows. Rectangle condition: rectangle (positive), rhombus (negative), parallelogram (negative). Rhombus condition: square (positive), rectangle (negative), parallelogram (negative). Parallelogram condition: square (positive), rectangle (positive), parallelogram (positive).

Results and discussion. The difference between the posttest and pretest scores was used as the dependent measure (see fig. 4), called relational-level improvement. This was calculated by giving each True or False question a score of 1 (improvement), 0 (no change), or -1 (decline). To begin, the two conditions are combined in order to measure the presence of any improvement at the relational-level from training at the object-level. Relational improvement for the aggregated group does differ from zero: (M = 0.0244, SD =

0.4624), t(2579) = 2.6822, p = 0.0074. This indicates that there is some improvement at the relational-level across conditions. However, relational-level improvement does not significantly differ between the teaching (M = 0.0243, SD = 0.4696) and baseline (M = 0.0248, SD = 0.4269) conditions using Welch's t-test: t(2578) = -0.0190, p = 0.9849. This indicates that while there is some relational-level improvement between pre- and post-tests, this is not likely due to the batch training phase.

Responses to individual questions were compared to chance to ensure that subjects were responding discerningly. For the individual questions, the absolute proportion correct (see fig. 3) was different from chance (0.5) for all questions except 'Are all squares also rhombuses?', (M = 0.4605, SD = 0.4990), t(429) = -1.6429, p = 0.1011. Additionally, the relational-level improvement for each question was compared to zero to determine which ones might account for the overall improvement. This relational-level improvement was different from zero for three questions: 'Are all squares also rectangles?', (M = 0.0698, SD = 0.4428), t(214) = 2.3104, p = 0.0218; 'Are all squares also rhombuses?', (M = 0.1395, SD = 0.5107), t(214) = 4.0062, p = 0.0001; 'Are all parallelograms also rhombuses?', (M = 0.0930, SD = 0.4840), t(214) = 2.8179, p = 0.0053.

Experiment 2

The findings of Experiment 1 suggest that teaching an object-level concept using a batch format results in little improvement in subjects' understanding of relational concepts and a similar level of improvement compared to a baseline condition. Experiment 1 provides little insight into the object-level knowledge subjects learned from the teaching phase. As such, several explanations are consistent with the experiment 1 results including two of particular interest:

1. Subjects are not learning from the teaching procedure in experiment 1.

2. Subjects are learning at the object-level but are not applying that knowledge to the relational-level.

Experiment 2 was designed to rule out one or both of these explanations. In it, the teaching phase was replaced with a learning phase like that of Shepard et al. (1961) and replicated by several others including Crump et al. (2013), who also used Amazon Mechanical Turk. This procure has subjects learn concepts by categorizing individual examples and receiving feedback. Thus, the categorization attempts provide a measure learning at the object-level. This can then be compared to learning at the relational-level. The expectation is that, for subjects who demonstrate learning at the object-level, learning at the relational-level will be different from that of the baseline condition in experiment 1.

Method

Participants. There were 39 total subjects who were recruited as in Experiment 1 except that due to this experiments length, each was paid $1.00.

Design. There was a single condition in this experiment. Subjects were separated by whether or not they reached the learning criterion (discussed in the Procedure) providing two groups.

Procedure. The procedure was as in Experiment 1 except that a learning phase was used instead of a teaching phase. In each learning trial, the 16 possible example–concept label pairs were presented to the learner in randomized order. For example, a subject would be presented with an example that was strictly a parallelogram (i.e. a parallelogram but no other shape concept) along with the question "Is this a rhombus?". Subjects then answered yes or no and were given feedback. The 16 shape-label combinations were randomly ordered for each trial. The learning criterion was two consecutive trials with no error. For subjects that did not reach criterion, the experiment was cut short at 16 trials.

Results and discussion

Thirteen subjects reached the learning criterion of scoring perfectly on two consecutive trials before the 15 trial learning phase terminated.

Ordinary least square regression is used to evaluate the extent of learning at the object-level in the learning phase (see fig. 6). Overall, the number of blocks to go (i.e. the number of blocks until the subject reaches criterion) predicted trial accuracy: b = 0.0306, t(8343) = 48.169, $p < 0.0001$. This was the case for the group that reached criterion: b = 0.0225, t(1479) = 13.442, $p < 0.0001$; and the group that did not b = 0.0310, t(6863) = 44.693, $p < 0.0001$. This indicates that subjects' scores on object-level questions improved throughout the training phase, whether or not the subject would eventually reach criterion in time.

Just as in Experiment 1, the relational-level improvement was measured by the difference between the relational pretest and posttest. Subjects were divided into groups according to whether or not they reached criterion (two consecutive trials with 100% accuracy). The group that did reach criterion (M = 0.0962, SD = 0.4217) had greater object-level improvement than subjects in experiment 1 (M = 0.0244, SD = 0.4624) using Welch's t-test: t(2734) = 2.0516, p = 0.0417, d=0.1559. So, those that reached criterion at the object-level in Experiment 2 did show greater relational-level improvement than subjects in Experiment 1.

Does this greater relational-level improvement in those that reached criterion indicate a transfer of learning across levels? Even though both the combined group (those who reached criterion and those who did not) showed significant improvement over the training phase, the relational-level improvement of the combined group (M = 0.0513, SD = 0.5210) did not differ from that of the Experiment 1 baseline: t(3046) = 1.0434, p = 0.2972, d=0.0569. The difference in object-level improvement between the group that did not reach criterion (M = 0.0288, SD = 0.5635), and the group that did trends but is non-significant: t(466) = -1.4491, p = 0.1481, d=0.1293.

In order to more directly test whether object-level learning is the source of relational-level improvement, a measure of object-level improvement was added. Object-level improvement is calculated by the difference between performance on the subject's final training block and and initial training block. If subjects are learning relational knowledge from the object-level training, then the relational-level improvement should be predicted by object-level improvement. However, improvement at the relational-level does not appear to be correlated with improvement at the object-level, r(37)=-0.05, p=0.75 (see fig. 5).

Experiment 2 provides evidence that when learning is measured at the object-level, those that reach criterion at the object-level might be able to transfer that learning to the relational-level. However, though those that did not reach criterion also demonstrated significant learning at the object-level, and this group did not seem to transfer object-level improvement to the relational-level. Further, there seems to be no correlation between object-level improvement and relational-level improvement. This then raises the question of how it might be possible to promote learning across levels of expressivity, the topic of Experiment 3.

Proposed Experiments 3

The following experiment is proposed to test a method for promoting the transfer of learning from the object-level to the relational-level. There are three conditions, each of which make use of a different presentation order in the training phase of the experiment. The random condition will be just as in Experiment 2, with examples presented in the training phase in random order. The remaining two conditions use the presentation order to emphasize the relational structure of the concepts (Mathy & Feldman, 2009; Eaves & Shafto, 2014). The last two conditions order the example presentation by one such argument to the relation (see table 4). In the label-order condition, examples are ordered such that the labels in the query are grouped together and the order of the labels is randomized. In the shape-order condition, examples are ordered such that the shapes in the query are grouped together and the order of the shapes is randomized.

The two main hypotheses for Experiment 3 are: First, the ordered conditions should both promote learning at the relational-level. Relational-level improvement will be lower in the random order condition than both the label-order and shape-order conditions. Second, the improvement at the relational-level should be predicted by object-level improvement in both of the ordered conditions but not in the random condition. There should be a significant correlation between object-level improvement and relational improvement for both ordered conditions.

Power analysis

The effect size between the three groups can be conservatively estimated from those of Experiment 2. For the comparison of the combined (reached criterion and did not) group object-level improvement to zero, Cohen's $d = 0.1559$, and for the comparison of the object-level improvement between criterion group and the non-criterion group, Cohen's $d = 0.1293$. A conservative estimate for the effect size between conditions is then $d = 0.1$. With power $1 - \beta = 0.8$ and $\alpha = 0.05$, the experiment would require $n = 787$ observations or, with 12 observations per subject, approximately 66 subjects per condition. For the correlational test, at the same power and Type I error rate, $n = 66$ would be a sufficient sample size for a correlational test with $r > 0.30$.

Discussion

Several researches have commented on the importance of using concept learning models and studies to better account for the full complexity of the concepts people learn both inside and outside of the lab as well as the representations people use to learn them (Kemp, 2012; Feldman, 2000; Piantadosi et al., 2016). This proposal introduces the idea of expressivity for concept learning models and concepts. I used this idea to organize several important studies and models of concept learning according the kinds of concepts they are able to express in principle. This analyses has yielded some surprising conclusions, such as that in some cases, using a more expressive model than necessary (i.e. using a first-order logic model on Boolean logic concepts) better explains subjects' learning.

Expressivity is also important for the topic of the experiments in this proposal. Some important concepts, such as the geometric concepts learned in school and used in these experiments, have structures at different levels of expressivity. One aspect of these geometric concepts can be defined completely in terms of Boolean logic while another requires a more expressive model. The experiments investigate learning across these two levels. Experiment 1 found that, using a common paradigm for learning at the object-level does not seem to lead to learning at the relational-level. Experiment 2 added a more involved paradigm for object-level learning as well as a measure of objet-level improvement. In Experiment 2, subjects who reached criterion did show greater relational-level improvement than in Experiment 1. However relational-level learning does not seem to be predicted by object-level learning, suggesting that transfer of learning might not be the best explanation in this case.

The first two experiments show that two common paradigms for learning concepts at the object-level provide little or no learning for related concepts at the relational-level. I have proposed Experiment 3 in order to test a promising method for promoting transfer across levels of expressivity. Previous work has shown that ordering examples of a concept promotes learning (Mathy & Feldman, 2009) including for relational concepts (Eaves & Shafto, 2014). I introduce two conditions that order examples according to the relational-level concepts in a training phase for object-level concepts.

The results of Experiment 3 should be relevant to the study and practice of concept learning and teaching. The results of the first two experiments indicate the value of understanding concepts like the geometric ones used as having distinct structures that might need to be taught and learned separately. If the ordered conditions of Experiment 3 do not promote transfer of learning across levels, then this conclusion will be strengthened further. However, if the ordered conditions of Experiment 3 do promote transfer of learning across levels, this will indicate an important method for teaching efficiently: teaching according to the full complex structure of the concepts to be taught.

Acronyms

CFG context-free grammar

DNF disjunctive normal form

FOL first-order Logic

RL representation language

RULEX rule-plus-exception model

References

Abelson, H., Sussman, G. J., & Sussman, J. (1996). *Structure and interpretation of computer programs*. Justin Kelly.

Beal, J. & Roberts, J. (2009). Enhancing Methodological Rigor for Computational Cognitive Science: Complexity Analysis. In *Proceedings of the 31th annual conference of the cognitive science society* (pp. 99–104).

Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). *A study of thinking*. Transaction Books.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on information theory*, *2*(3), 113–124.

Collins, A. M. & Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological review*, *82*(6), 407.

Crump, M. J., McDonnell, J. V., & Gureckis, T. M. (2013). Evaluating amazon's mechanical turk as a tool for experimental behavioral research. *PloS one*, *8*(3), e57410.

Eaves, B. S. & Shafto, P. (2014). Order effects in learning relational structures. In *Proceedings of the 36th annual conference of the cognitive science society*.

Feldman, J. (2000). Minimization of Boolean complexity in human concept learning. *Nature*, *407*(6804), 630–633.

Feldman, J. (2006). An algebra of human concept learning. *Journal of mathematical psychology*, *50*(4), 339–368.

Gentner, D. & Markman, A. B. (1997). Structure mapping in analogy and similarity. *American psychologist*, *52*(1), 45.

Goodman, N. D., Tenenbaum, J. B., Feldman, J., & Griffiths, T. L. (2008). A Rational Analysis of Rule-Based Concept Learning. *Cognitive Science*, *32*(1), 108–154.

Goodwin, G. P. & Johnson-Laird, P. N. (2011). Mental models of Boolean concepts. *Cognitive psychology*, *63*(1), 34–59.

Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Harvard University Press.

Kemp, C. (2012). Exploring the conceptual universe. *Psychological review*, *119*(4), 685–722.

Kemp, C., Goodman, N., & Tenenbaum, J. B. (2008). Learning and using relational theories. In *Advances in neural information processing systems* (Vol. 20, pp. 753–760).

Kemp, C. & Regier, T. (2012). Kinship categories across languages reflect general communicative principles. *Science*, *336*(6084), 1049–1054.

Kruschke, J. K. (1992). ALCOVE: an exemplar-based connectionist model of category learning. *Psychological review*, *99*(1), 22.

Laurence, S. & Margolis, E. (1999). Concepts and cognitive science. *Concepts: core readings*, 3–81.

Love, B. C., Medin, D. L., & Gureckis, T. M. (2004). SUSTAIN: a network model of category learning. *Psychological review*, *111*(2), 309.

Mathy, F. & Feldman, J. (2009). A rule-based presentation order facilitates category learning. *Psychonomic bulletin & review*, *16*(6), 1050–1057.

Medin, D. L. & Shoben, E. J. (1988). Context and structure in conceptual combination. *Cognitive Psychology*, *20*(2), 158–190.

Mervis, C. B. (1980). Category structure and the development of categorization. *Theoretical issues in reading comprehension*, 279–307.

Minsky, M. & Papert, S. (1969). Perceptrons.

Murphy, G. L. (2002). *The big book of concepts*. MIT press.

Neisser, U. & Weene, P. (1962). Hierarchies in concept attainment. *Journal of Experimental Psychology*, *64*(6), 640.

Nosofsky, R. M., Palmeri, T. J., & McKinley, S. C. (1994). Rule-plus-exception model of classification learning. *Psychological review*, *101*(1), 53–79.

Piantadosi, S. T., Tenenbaum, J. B., & Goodman, N. D. (2016). The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological Review*.

Rosch, E. H. (1973). On the internal structure of perceptual and semantic categories.

Rosch, E. H. (1999). Principles of categorization. *Concepts: core readings*, 189–206.

Rosch, E. H. & Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, *7*(4), 573–605.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Roth, J. C. (2013). *Fundamentals of logic design*. Cengage Learning.

Searcy, S. R. & Shafto, P. (2016). Cooperative inference: A theoretically and practically feasible model of teaching by example. *Psychological Review*.

Shafto, P. & Goodman, N. D. (2008). Teaching games: Statistical sampling assumptions for learning in pedagogical situations. In *Proceedings of the thirtieth annual conference of the cognitive science society* (pp. 1632–1637).

Shepard, R. N., Hovland, C. I., & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs: General and Applied*, *75*(13), 1–42.

Siegelmann, H. T. & Sontag, E. D. (1991). Turing computability with neural nets. *Applied Mathematics Letters*, *4*(6), 77–80.

Smith, E. E. & Medin, D. L. (1981). *Categories and concepts*. Harvard University Press Cambridge, MA.

Turing, A. M. (1937). Computability and λ-definability. *The Journal of Symbolic Logic*, *2*(04), 153–163.

van Rooij, I., Kwisthout, J., Blokpoel, M., Szymanik, J., Wareham, T., & Toni, I. (2011). Intentional communication: Computationally easy or difficult? *Frontiers in Human Neuroscience*, *5*(52), 1–18.

Vigo, R. (2009). Categorical invariance and structural complexity in human concept learning. *Journal of Mathematical Psychology*, *53*(4), 203–221.

Representation (general)	What is the basic structure of the representation? (rules, exemplars, prototypes, possibly even associations)
Representation (specific)	Given each representation type, there are many different ways to instantiate the representation. What kind of features are used? What operations?
Expressivity	How expressive are different models? What expressivity do certain concepts require? How does expressivity affect learning?
Complexity	Given the representation, what determines the complexity (i.e. difficulty of processing and using) of the individual concepts in the representation?
Composition	How are existing concepts used in the service of learning new concepts?
Minimization	When learning/using a concept, does one find the absolute minimum version, an approximate minimum, or any at all?
Determinism	Many of the questions above can be answered either deterministically (i.e. given the same input, the response should always be the same) or not (e.g. probabilistically)
Aggregation	Related to the question of determinism is the idea that some important evidence of different representations might be lost when responses from many are aggregated.
Mixture	Related to the minimization and composition questions is the question of whether individuals hold a single definitive version of each concept or some composite or mixture.

Table 1

Key questions in literature

A (certainly incomplete) list of the significant questions to be addressed by the study of representation and concept learning. These questions are all at least partially orthogonal, meaning that answering one might provide some information for answering others but should not be expected to completely answer any others. Additionally, the relationship between these questions

Level	Description	Examples of concepts	Models
Boolean Logic and predicate calculus	Each feature and operator take as input only the Boolean labels or expressions that evaluate to them	AND, OR, XOR, NAND, the SHJ set of concepts (Shepard, Hovland, & Jenkins, 1961)	Exemplar models (Rosch, 1973; Smith & Medin, 1981) and many rule-based models (Nosofsky, Palmeri, & McKinley, 1994; Feldman, 2000; Goodwin & Johnson-Laird, 2011)
First-order logic	As Boolean logic with the addition of quantifiers over elements	Quantities and comparisons	Kemp (2012)
Second-order logic	As First-order logic with the addition of quantifiers over relations, functions, and/or sets	Natural number line and arithmetic	Piantadosi, Tenenbaum, and Goodman (2016)
Higher-order logics, lambda-calculus, and others	Universal models of computation that can describe any computable function	Arbitrary computer programs and concepts	

Table 2

Levels of expression

Expressivity can be organized into a hierarchy. An example of this is presented in the above table with the less expressive representation languages (RLs) on top. Each row corresponds to a class of RLs that can express any concept expressible by those above.

Shape concept	Object-level definition
Square	$f_r \wedge f_e \wedge f_q$
Rectangle	$f_r \wedge f_q$
Rhombus	$f_e \wedge f_q$
Parallelogram	f_q

Table 3

Object-level definitions for geometric concepts

The four geometric concepts are defined at the object-level using the Boolean formulas above.

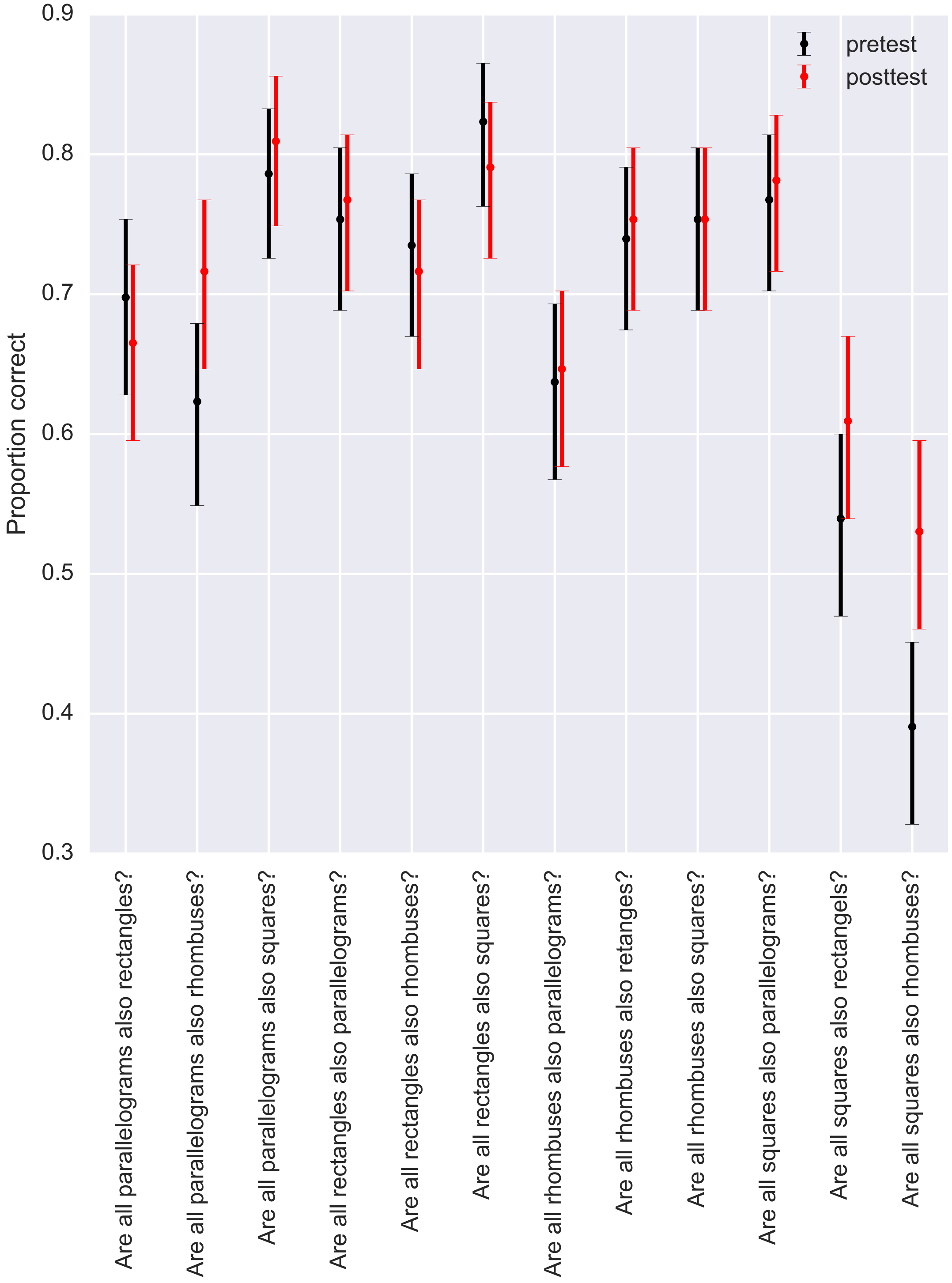


Figure 3. Mean accuracy for pre- and posttest aggregated across teaching and baseline conditions by question. Bootstrapped standard error is shown.

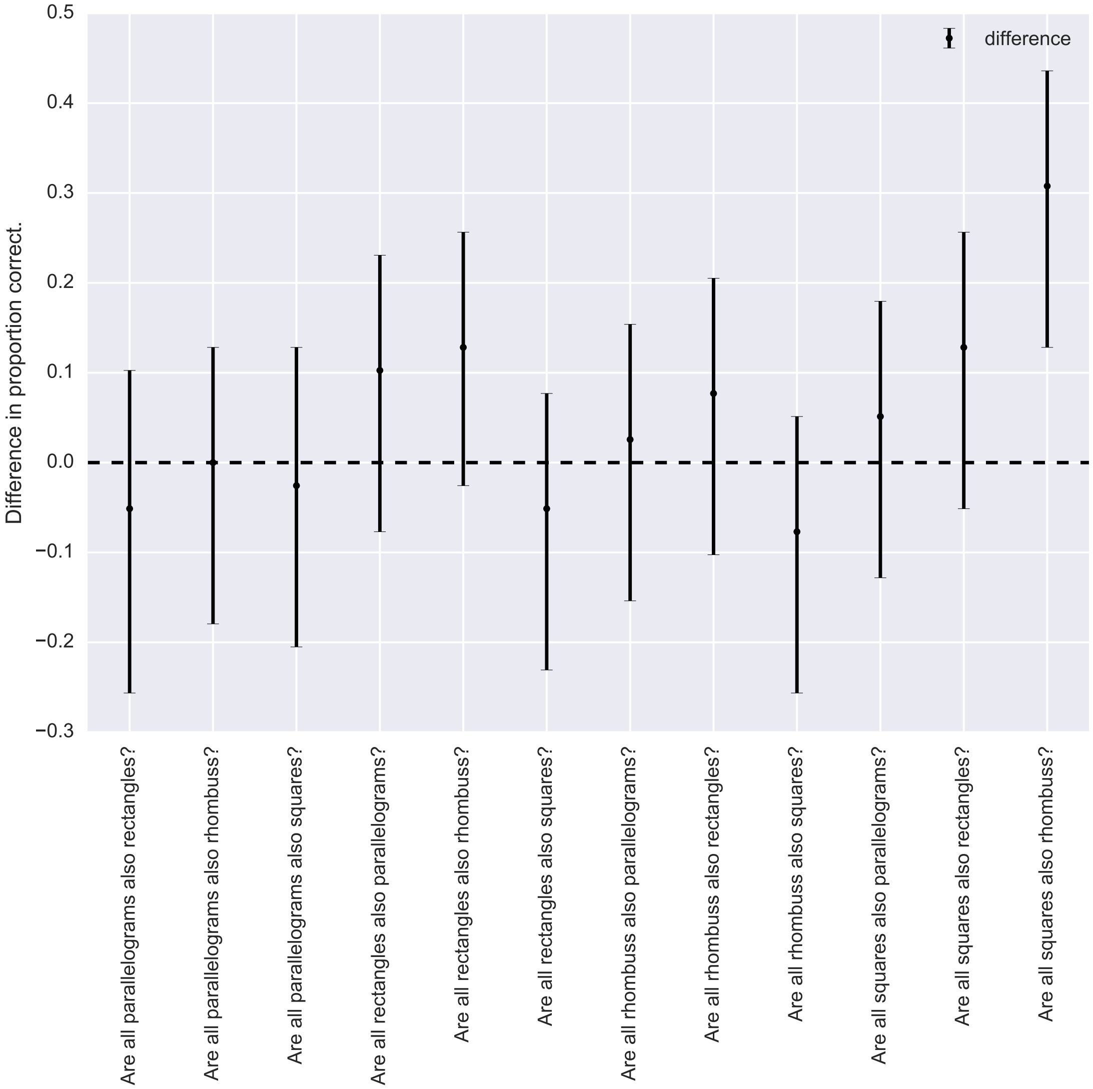


Figure 4. Difference between pre- and posttest aggregated across conditions by question.

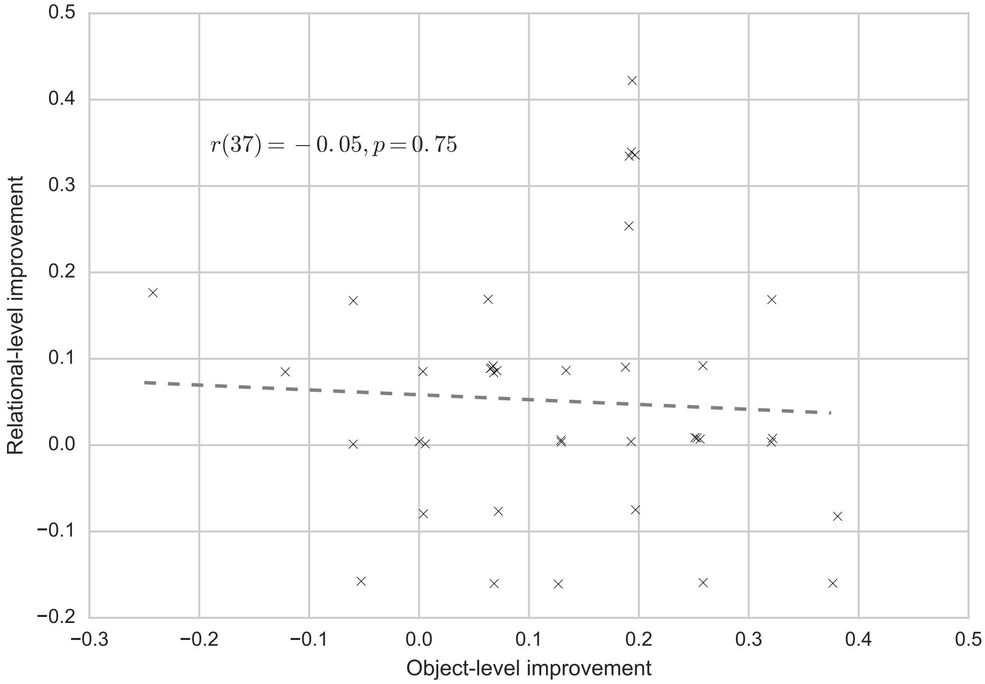


Figure 5. Subjects' improvement at the relational-level does not appear to be correlated with improvement at the object-level. (Points are plotted with noise to more clearly indicate when multiple subjects have the same value.)

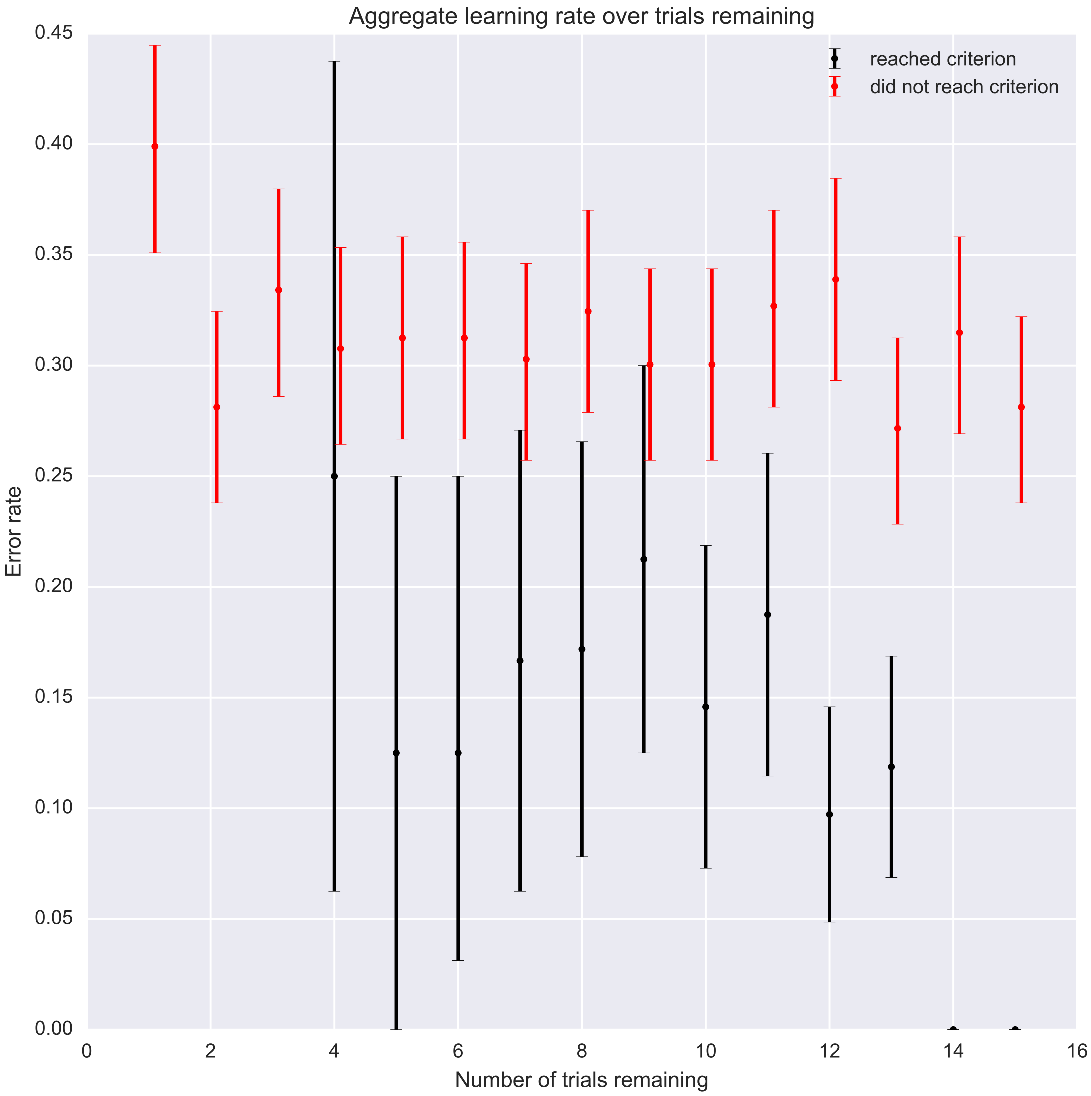


Figure 6. The average error rate in each trial as a function of the number of trials remaining for that participant. Subjects are grouped according to whether or not they reached criterion.

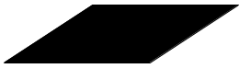
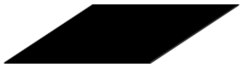
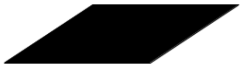
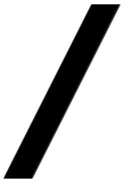
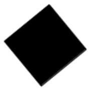
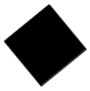
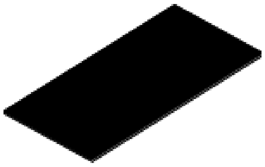
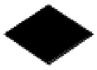
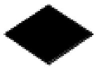
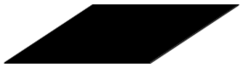
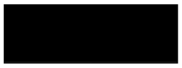
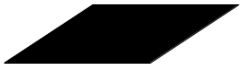
Shape-order	Label-order	Random-order
		
Is this a rhombus?	Is this a rhombus?	Is this a rhombus?
		
Is this a rectangle?	Is this a rhombus?	Is this a rectangle?
		
Is this a square?	Is this a rhombus?	Is this a square?
		
Is this a parallelogram?	Is this a rhombus?	Is this a parallelogram?
. . .	. . .	. . .

Table 4

Experiment 3 order-based conditions

Shown are examples of the three conditions planned for Experiment 3. The shape-order condition groups object-level questions by the shape presented, label-order condition groups by the label queried, and the random condition randomizes all questions.