

Learning Object Permanence by Ontological Leaps

Dean Foster (foster@diskworld.wharton.upenn.edu)

Statistics Department
University of Pennsylvania
Philadelphia, PA 19104

Lyle Ungar (ungar@cis.upenn.edu)

Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104

Abstract

One of the key questions of both cognition and machine learning is how children learn (and how computers might learn) to recognize different objects and to know that they continue to exist, change, and move, even when not seen. Cognitive scientists are divided on the question of how much of the representational structure for object permanence is innate. We argue, by example, that a quite simple representation and learning scheme is sufficient to learn object permanence. Key to this learning is making what we term an *ontological leap*: the automatic creation of new concepts from combinations of properties of existing ones. Unlike many systems which create new features from combinations or existing ones, the concepts we learn are treated on a par with innate concepts, allowing them to enter into relationships and to have properties. This allows, among other things, the leap from raw pixels to persistent physical objects.

Motivation

Children of a young age are not surprised when an object reappears after passing behind another; they recognize that objects continue to exist even when not seen. They do, however, exhibit surprise when one object passes behind another and two objects appear on the other side (Baillargeon et al. 1999). This behavior suggests that they have an underlying representation of objects, which includes some representation of the objects images and their velocities. Object permanence is ubiquitous: a tree in the summer and tree in the winter are perceived to be the same tree, even though they look very different. Similarly, if less dramatic, an object may look quite different when viewed from different angles or under different illuminations. A related issue is the correlation of different senses. A child (or, for that matter a dog) knows that the image, the sound, and the smell of her father all refer to the same person. In the language of machine learning, there is an unobservable or latent variable, which correlates with all these features. An unanswered question is which of these latent variables are innate and which are learned.

One of the key questions of both cognition and machine learning is how children learn (and how

computers might learn) to recognize different objects and to know that they continue to exist even as they move and change, or even are fully occluded by other objects. The standard approach to machine vision is to build by hand a representation for objects and features by which these objects can be recognized, and then to learn parameters in these models; I.e., to learn very little. From an engineering perspective, this makes sense – if it is slower to learn something than to program it, then there is no point in learning it. From a science perspective, however, the more abstract issue of learnability is important. Cognitive scientists are divided on the question of how much of the representational structure for object permanence is innate (Baillargeon et al. 1999). We argue, by example, that a quite simple representation and learning scheme is sufficient to learn object permanence.¹ The same scheme solves a host of other learning problems.

There has been extensive argument as to what type and degree of cognition must be present in a child for her to exhibit competence at recognizing object permanence. The goal of this paper is to present a relatively simple knowledge representation and learning scheme which allows a computer to learn object permanence and, more generally, to efficiently learn complex features of the world, for example, absolute location of objects based on viewer-centric perception.

We start with a set of primitive *concepts*, such as pixels, *relationships* between them, such as being to the left of or occurring before and *properties*, such as color or location. These can be thought of as forming a semantic net. New concepts are derived by any classical concept formation method, such as learning rules or prototypes for class memberships (Angluin 1988; Rosch 1978). In the implementation described below, we define new concepts by clustering existing ones. E.g., a cluster of pixels which have a similar colors might be a concept. We then make an *ontological leap* from a cluster of pixels or times with similar properties to a new concept which can

¹Of course, this does not prove that object permanence *is* learned, merely that learning it is easier than many people believe.

in turn have its own properties (e.g. expected color at some future time). New concepts are added to the the semantic net, and have the same ontological status as the original ones. In particular, they have properties whose values can be forecast. Learning is now a search in the space of concept definitions for ones whose properties help us to make accurate forecasts. This search space is, of course, intractably large, but local search (e.g., spreading activation) gives excellent results.

Problem specification

Let us formulate a simple problem to illustrate our approach. Consider a world consisting of moving objects such as blue balloons that rise and of red bricks that fall. Given a sequence of images of one or more objects moving around in that space, one should be able to predict images at future times by estimating where each of the objects will be based on their current position and velocity. To do this, however, requires first recognizing that there *are* objects, secondly that they have location and velocity, and finally that these objects persist over time, even if not fully visible. One might try to make predictions directly from the raw pixels, but in a world with noisy perception, much greater accuracy can be obtained by averaging over all the pixels in each object, and in a world with occlusion, objects must be modeled as existing even when not seen. We wish to do this without building in representations of objects, or of properties such as velocity.

Our method can be summarized in general terms as follows:

- Start with a semantic network of *concepts*. These may be *primitive* (innate) (e.g., pixels, times, outputs of filters) or *derived* (learned) (e.g., edges, objects, trajectories). The concepts are linked by *relations*, which map from one concept to another (e.g, $x = \text{left-of}(y)$ or $t_1 = \text{previous}(t_2)$) and they have *properties*, which map to real values. (e.g., $\text{darkness}(x) = 0.7$, $\text{size}(x) = 2.4$, $x\text{-location}(y) = 9.1$).
- Build models to forecast future values of each concept using values of properties of its neighbors (other concepts connected by relations), including neighbors of neighbors, recursively to the extent that can be computationally afforded. Select models based on accuracy of prediction and use of minimal numbers of features. We use linear regression for the modeling, and stepwise regression for feature selection, but any modeling form could be used.
- Cluster concepts based on their having correlated properties or having correlated errors in their forecasts.

- Reify the clusters into new concepts, compute further properties of them (e.g., the average, variance, min and max of each of the properties of the items in each cluster), and add these derived concepts to the semantic network.
- Repeat

The middle three steps, which form the core of the method, correspond to feature selection (as part of the model building and fitting), feature creation (or concept learning) and feature reification (adding features to the semantic network).

The first question is how to learn that a set of pixels refer to an object. We assume, for now, the knowledge of spatial relations (*left-of*, *right-of*, *above* and *below*) which return the appropriate pixels. In the brain, relative locations appear to be innately coded (given by local receptive fields). They could, however, easily be learned by noting which pixels are most correlated with with other pixels.² We then try to predict the color of each pixel at time $t + 1$ as a function of all known properties at time t . For simplicity and speed, we use stepwise linear regression. Linear regression is particularly attractive because new variables can be cheaply added to a regression using a “sweep” method, and because there are well-established criteria for variable selection. For nonlinear relationships, we extend our search over quadratic or cubic terms in the primary variables. However, any function approximation method, for example, a neural network, could be used. We claim that humans and other animals notice correlations, not that they compute t-statistics.

So far, nothing has driven the creation of new concepts. To do that, we need to consider the effect of noise. If perceptions are noisy (perhaps due to sporadic partial occlusion), then the color of a pixel is best estimated by averaging the color estimates of all the pixels in the object (for objects of a single color). These new concepts provide new variables to use in making predictions. Search over the properties of such concepts will find that for many pixels, the best prediction is a function of the same combination of variables: the colors of the pixels of the object. Similarly, if perceptions are noisy, then the velocity of an edge (for rigid motion) is best estimated by averaging the velocity estimates of all the edge pixels of the object.

The Ontological Leap

A key step in learning object permanence is the jump from original features to new ones which extend beyond being merely combinations of the original features. At first sight, this may even seem impossible: how can one create new features which are not just combinations of previously known ones?

²Since there are n pixels, this requires computing n^2 correlations, which is not excessive.

Our basic strategy is two-fold. First, we cluster concepts or properties by their value (e.g. being the same color), their conditional value (being the same color, controlling for other variables such as it being day or night) or by their prediction error (having the same error in the predicted color). We then create new variables which are sets of lower level concepts (e.g., a set of pixels). These concept-sets are first class objects, and will have their own properties (e.g., average color). Given a number of pixels of unknown physical location and their values over time, one can, for example, learn which ones are neighbors by seeing which ones are most highly correlated. If there are patterns in the world such as vertical stripes or balloons that rise, one can learn concepts of above/below by looking at the correlations with current or previous neighbors. Some pixels will be more highly correlated in certain circumstances (e.g. a zebra or a balloon present). Edges have distinctive patterns of correlation and of errors forecast. Further, the same method that aggregates pixels into edges can be used to aggregate these edges into parts of objects, say, leg, head, arm, etc. Repeating the same algorithm yet again allows aggregating these body parts into whole objects. The learning procedure is very general.

Concept learning

Many different methods could be used to learn new concepts. Most represent concepts as some function of a set of properties (Smith Medin, 1981), but the representation form and is widely variable. Our approach is agnostic as to how concepts are found and represented. The novelty in our method is the ontological leap, not the methods used to derive new concepts. We do assume that the world has structure in it, and that patterns (clusters) in percepts really do exist. This is certainly true, otherwise we could never understand the world, learn, or even exist (Richards and Bobick, 1988). One can search for these concepts in many ways. Logical or rule-based methods have been widely proposed for concept learning (Angluin 1988), and applied to problems such as recognizing conceptual groupings in scene analysis (Scholl 2001). In vision, an enormous variety of filters combined and tuned in various ways have been successfully used (see e.g., Jaehne 1993), as have projection methods such as PCA (Black et al. 1997). Alternatively, prototype-based concept definitions have been argued for by psychologists (Rosch 1978) as being the “correct” model of how people represent concepts.

In the machine learning literature, these methods generally go under the heading of *unsupervised learning*, as they derive classification rules without having any data with examples of objects (and their properties) labeled with their “true” class membership. Any such unsupervised learning method could be incorporated into our scheme. We describe two

clustering methods that we have found particularly attractive for learning object persistence: clustering by similarity of property values and clustering by similarity of error in predicting property values.

One way of finding new concepts/properties is to cluster concepts by common value. A set of pixels which have the same value at the same time might have other things in common, like being part of the same brick. A group of times where many pixels have similar values might also form a concept, such as “night”. Our basic operation is to cluster lower level concepts of the same type (e.g. pixel) which are (recursively) neighbors of each other and have a property with similar values. (E.g., pixels which are physical neighbors of each other and have a similar redness or, at a higher level, objects which are of people and have a common velocity or orientation.)³

If a pixel is an element of a derived concept, such as a cluster of pixels representing a brick, the pixel inherits all the properties of the brick. Thus, the pixel might, in addition to its local properties like *redness* also have properties such as the average redness of the brick or the velocity of the brick. Since the new brick concept has the full status of a node in the semantic net, it has properties of its own. These properties can, in turn, be forecast. This is important because it is often the case that the forecast of a variable is much more useful than the variable itself. For example, the average of a set of human forecasts is typically more accurate than the individual forecasts and, more relevantly, one can often learn a forecasting model which predicts more accurately than the forecaster whose predictions were used as a training set.

The cluster of pixels we have called a brick does not yet capture object permanence; it is a cluster at a single time. However, further clustering the pixel-clusters across time will create a new concept *pixel-cluster-over-time*, which is now very close to our notion of a brick. This new concept could finally have complex properties, such as velocity at each time, that we seek. Yet further search for concepts which are useful for making predictions will reveal that a *pixel-cluster-over-time* may have a property *occluded* upon which the pixels colors strongly depend. Thus, an object can be come to be represented as existing, and having properties (such as estimated velocity), even when it is not currently seen.

An alternate way of forming new concepts is to cluster pixels by the errors made in forecasting their values. Consider the case where we are using the simple forecasting model that the redness of a pixel at at time is equal to its redness at the preceding time. In this case, the errors for a falling brick will

³One can also cluster by common conditional value; rather than grouping pixels based on their color, one might group pixels based on their color controlling for the estimated local illumination or presence of an occluding object.

be the row of pixels directly above the brick and the bottom row of pixels in the brick. A new concept denoting all the pixels which share those errors describes something similar to the edges of the brick. Looking only at pixels that are physically close to each other gives concepts such as “top edge”.

Detailed Example

As a concrete illustration, we have implemented a system which models a sequence of images (pixels and their colors) of red bricks falling at the rate of one pixel per time step across a white background. At each time step, random noise is added to the colors of both the brick and background pixels. Our goal is to learn objects, but we do so implicitly by selecting new concepts which help us predict future colors of the pixels. Let each pixel at each location i, j and time t be a primitive concept, $pixel_{i,j,t}$. Pixels are related in time by *next* and *previous* relations (e.g., $next(pixel_{1,1,1}) = pixel_{1,1,2}$) and in space by *left-of*, *right-of*, *above* and *below* (e.g., $right-of(pixel_{1,1,1}) = pixel_{1,2,1}$). Each pixel also has a set of real-valued properties $xloc$, $yloc$, $time$, and $redness$.

These properties might be represented in a table, and will constitute the initial set of features used to try to predict the future *redness* of each pixel.

pixel	xloc	yloc	time	redness
$pixel_{1,1,1}$	1.0	1.0	1.0	0.2
$pixel_{1,2,1}$	1.0	2.0	1.0	0.8
$pixel_{2,1,1}$	2.0	1.0	1.0	0.9
$pixel_{2,2,1}$	2.0	2.0	1.0	0.6
...				
$pixel_{1,1,2}$	1.0	1.0	2.0	0.1
$pixel_{1,2,2}$	1.0	1.0	2.0	0.2
$pixel_{2,1,2}$	2.0	1.0	1.0	0.8
$pixel_{2,2,2}$	2.0	2.0	1.0	0.8
...				

We start by searching for a model which will forecast the redness of the pixels at the second time step, given values at the first time step.⁴ Given only the primitive properties, a simple forecast rule for the *redness* of some pixel x is $redness(previous(x))$. More accurate for falling bricks would be $redness(previous(north(x)))$ or some linear combination of the colors of items above and to either side of x .

If we search for derived concepts, looking for a set of similar pixels at the initial time, we will find a set of three pixels with similar redness. (They

⁴The above implementation does not have any explicit representation of time, only the *next* and *previous* operators. We could equally well have made *time* a primitive concept, on a par with *pixel*, and used properties such as $redness(pixel_{1,1}, time_1) = 0.2$, and $next(time_1) = time_2$.

also have similar locations, but for this tiny example, all the pixels have similar locations.) Call this concept $object_1$. It has a set of elements, $pixel_{1,2,1}$, $pixel_{2,1,1}$, $pixel_{2,2,1}$, and a number of properties automatically derived by averaging the properties of the pixel which constitute it. Similar clustering on pixels at time two gives a second object. The objects include properties given in the following table.

object	xloc	yloc	time	redness
$object_1$	1.66	1.66	1.0	0.76
$object_2$	2.0	1.5	2.0	0.80

Since we can now use features of the concepts which contain each pixel x when forecasting its redness, we now find a better model is $redness(x) = redness(object(x))$, where the automatically generated relation $object(x)$ returns the object of which x is an element.

Further search discovers more complex concepts. Objects which are close in location, time, and redness will be clustered into a new concept, which we can call *lasting-object*. Again, properties such as average locations, time, and redness are computed. Since the color of the brick is constant over time, a better forecast can now be made using $redness(x) = redness(lasting-object(object(x)))$, which averages the color over the entire time the object is observed.

Discussion

The toy example above gives a picture of how object permanence might emerge. In trying to forecast pixel values one step in the future, one learns physical neighbors. (In animals, this learning probably occurred as part of evolution, not during development of each individual.) Given physical neighbors, new concepts such as edges and objects, which are composed of changing sets of pixels over time, can be learned, and they provide major increase in predictive power. These new concepts are treated on a par with percepts: they can enter into relationships, can have properties associated with them, and can be elements of yet more complex concepts. To predict a pixel color of an object which is sporadically occluded, one needs to have a representation of the occluded object as existing and possessing a velocity even when it was not visible. Such representations can be learned by ontological leaps – creating new concepts which have properties derived the properties of the primitive concepts from which they are derived.

We have only sketched out the set of concepts which would be learned in the above example. Velocity arises naturally as the difference between the centers of objects: $xloc(next(object)) - xloc(object)$. Occlusion will be learned by looking at the errors in predicting the color of pixels which are believed to belong to an object. Grouping based on similar-

ity of prediction error and proximity (predicting the wrong color for pixels which are close to one another) will give rise to a new concept *occlusion*. Since each such relationship between a base concept (e.g. *pixel*) and a derived concept (e.g. *occlusion*) automatically generates a property of each *pixel* which is 1 if it is an element of *occlusion* and 0 if it is not, the redness of a pixel x can now be learned to be:

$$\text{redness}(x) = \text{redness}(\text{occlusion}(x)) * \text{occluded}(x) + \text{redness}(\text{object}(x)) * (1 - \text{occluded}(x))$$

The above equation, like all we have shown, can be learned by stepwise regression, this time including quadratic interaction terms.

In future work, we plan to run learn to predict pixels multiple time steps in the future, by training on a images containing a mixture of visible and occluded objects. We expect that we will be able to replicate the lack of surprise babies show when an object passes behind an occlusion and then reappears, by forecasting that behavior, and to replicate the surprise that babies show when one object passes behind an occlusion and two objects appear, but showing that that contradicts our forecasts.⁵

Thus, it appears that one need not have a specific innate concept of object permanence. It suffices to have the more general notion of neighboring percepts and an ability to create new features which are first class objects (i.e., are indistinguishable from the raw percepts). We cannot, of course, prove whether human infants learn object permanence, or whether it has been “learned” by evolution; we only show that it is learnable with general purpose machinery.

Similar methods could be used to learn a vast array of different features and relationships. We briefly mention a few which are relevant to vision. The ontological leap approach could also be extended to 3-D objects, which change in apparent size and shape as they move, to objects with changing illumination, to moving perceivers, to non-ridged objects (people walking) and to objects that really do change in size or shape (balloons being blown up or children growing up).

Many of these problems can be approached similarly to recognizing that a tree in the winter is the same object as that the in the summer, even though it appears to be very different. Each successive image of the tree is similar in location, time, and appearance. These can be grouped by either by single link clustering through time, or by first forming concepts such as “tree-in-winter” and “tree-in-summer”, and then clustering them. For changes in illumination, one could form a concept which extends over time, consisting of multiple single-time images, similarly to the summer/winter tree. For

⁵We have used purely deterministic models so far; to really capture surprise, one would want a probabilistic model which forecasts the probabilities of different events. This fits cleanly into the approach we have described.

rapid local changes in illumination, one could also use the equivalent of velocity: the change in the lightness.

We have sketched out a general approach to learning concepts such as object permanence. Our method can be viewed as a concrete computational approach to solving the symbol grounding problem (Harnad 1990) of relating continuous (real-valued) percepts to discrete derived concepts. Our method contains three steps: (1) feature selection (as part of the model building and fitting), (2) feature creation (concept formation) and (3) concept reification (adding concepts to the semantic network). The feature selection conforms to the requirement that symbol grounding be driven by the effectiveness of the new symbols (concepts) in modeling the world. This is related to Sun’s (1999) suggestion that symbols be grounded in interaction between the agent and the world. Key to our method is the ontological leap of integrating new concepts into a network of relationships so that we can compute and use additional properties of the new concepts. This allows us to learn objects which persist over time and have properties such as velocity and color, even at times when they are not observed.

References

- Angluin, D. Queries and concept learning, *Machine Learning*, 2(4):319–342, 1988.
- Baillargeon, R. Children’s expectations about hidden objects: A reply to three challenges and L. B. Smith. Do infants possess innate knowledge structures? with Peer Commentaries *Developmental Science* 2:2 115–163. 1999.
- Black, M. et al. Learning parameterized models of image motion. *Proc IEEE CVPR* 561–67. 1997.
- Harnad, S. (1990). The symbol grounding problem. *Physica D* 42, 335–346.
- Jaehne, B. *Digital Image Processing*. Springer-Verlag. Berlin. 1993.
- Richards, W. and Bobick, A. Playing twenty questions with nature. in Z. Pylyshyn (ed.) *Computational processes in Human Vision*. Ablex, Norwood, N.J. 1988.
- Roth, D., M-H. Yang and N. Ahuja. Learning to Recognize Objects *CVPR ’00*, June 2000.
- Rosch, Eleanor. *Cognition and Categorization*. Erlbaum, Hillsdale, New Jersey. 1978.
- Scholl, B., Pylyshyn, Z. and Feldman, J. What is a visual object? Evidence from target merging in multiple-object tracking. *Cognition*, 80(1-2), 159–177. 2001.
- Smith, E. and Medin, D. *Categories and concepts*. Harvard University Press, Cambridge. 1981.
- Sun, R. Symbol grounding: a new look at an old idea. *Philosophical Psychology*, 13, 149–172. 2000.
- Wildes, R.P. and J.R. Bergen. Qualitative Spatiotemporal Analysis using an oriented energy rep-

resentation. in *Proc. European Conference on Computer Vision*, 768-784, 2000.