# Learning Concepts by Interaction

Paul R. Cohen

Department of Computer Science
University of Massachusetts, Amherst, MA 01003

cohen@cs.umass.edu

## Abstract

This paper presents a theory of how robots may learn concepts by interacting with their environment in an unsupervised way. First, categories of activities are learned, then abstractions over those categories result in concepts. The meanings of concepts are discussed. Robotic systems that learn categories of activities and concepts are presented.

## Introduction

If machines could acquire conceptual knowledge with the same facility as humans, then AI would be much better off. There's no denying the dream of a machine that knows roughly what we know, organized roughly as we organize it, with roughly the same values and motives as we have. It makes sense, then, to ask how this knowledge is acquired by humans and how might it be acquired by machines.

I focus on the origins of conceptual knowledge, the earliest distinctions and classes, the first efforts to carve the world at its joints. One reason is just the desire to get to the bottom of, or in this case the beginning of, anything. Another is that the origin of concepts is hotly debated: Some people think that neonates are born with moderately sophisticated conceptual systems (e.g., 4,5,15), others dispute this and seek an empiricist, or non-nativist, account of development (e.g., 10,11). I think one has to take a minimalist stance and avoid innate knowledge in one's explanations of the acquisition of later knowledge. This reluctance comes, in part, from years of slogging in AI, where we must always provide a lot of knowledge for our systems to do relatively little with. I want to focus on the first concepts because unless I do, I will have to provide them by hand, which is difficult, and also makes me very uncertain about the explanatory power of what follows.

The claim of this paper is that *concepts can be learned without supervision by abstracting over representations of activities*. Piaget [12] insisted that concepts must arise out of activities because simple action schemas develop before conceptual thought, but the mechanisms he proposed to explain concept acquisition were vague by the standards of artificial intelligence. I will present methods for learning *clusters* of activities, implemented in robots and software agents, and argue that these representations identify categories and, with further processing, concepts.

## Why Should Agents Learn Concepts?

Imagine yourself to be a mobile robot wandering around the lab. What concepts do you need? Some would suggest that you need no concepts at all, only behaviors. These recognize aspects of your internal state and the world state, and generate actions in response; for example, when you detect an obstacle in your path, you change direction or stop. As described, the concept "obstacle" doesn't exist for you, the robot. It exists for the person who wrote a function `obstacle-in-path`, not for you. You do not know that obstacles may sometimes be pushed aside, although you often push them aside; you do not know that obstacles impede paths, although obstacles impede your paths; you do not even know what a path is, although you trace one whenever you move. You have no concepts, just a bunch of behaviors. And why *should* you know anything, if these routines work? Why do you need concepts like "obstacle" and "impede" and "path" at all?

The function `obstacle-in-path` was written by a person who thought about all the situations you might encounter and realized that some of them involve an "obstacle" in your "path." Because this person conceptualized your experiences, you have an appropriate response to a common situation. Your behavior is organized around your programmer's conceptual structure. This structure serves you well. This is what concepts are for: They give you interpretations of your sensors, a basis for judgments that situations are similar, and the distinctions on which you decide what to do. Concepts are necessary even when they are implicit. If `obstacle-in-path` produces intelligent behavior, the intelligence must be attributed in no small part to the programmer who dreamed up the concepts "obstacle" and "path." I want robots to dream up such concepts for themselves.

## From Activities to Prototypes

Here is the problem to be solved: An agent such as a robot is "born" with a small set of physical activities but no conceptual system. As it interacts with its environment, it forms categories, and by induction over category instances, concepts. How does this work? What innate structures are required? This section presents some implemented methods that solve parts of the problem.

## How the Pioneer 1 Robot Learns Prototypes

The Pioneer 1 robot has two independent drive wheels, a trailing caster, a two degree of freedom gripper, and roughly forty sensors including five forward-pointing sonars, two side-pointing sonars, a rudimentary vision system, bump and stall sensors, and sensors that report the state of the gripper. The robot is controlled by a remote computer, connected by radio modem.

The robot's state is polled every 100msec., so a vector of 40 sensed values is collected ten times each second. These vectors are ordered by time to yield a multivariate time series. Figure 1 shows four seconds of data from just four of the Pioneer's forty sensed values. With a little practice, one can see that this short time series represents the robot moving past an object. Prior to moving, the robot establishes a coordinate frame with an x axis perpendicular to its heading and a y axis parallel to its heading. As it begins to move, the robot measures its location in this coordinate frame. Note that the ROBOT-X line is almost constant. This means that the robot did not change its location on a line perpendicular to its heading, that is, it did not change its heading, during its move. In contrast, the ROBOT-Y line increases, indicating that the robot does increase its distance along a line parallel to its original heading. Note especially the VIS-A-X and VIS-A-Y lines, which represent the horizontal and vertical locations, respectively, of the centroid of a patch of light on the robot's "retina," a CCD camera. VIS-A-X decreases, meaning that the object drifts to the left on the retina, while VIS-A-Y increases, meaning the object moves toward the top of the retina. Simultaneously, both series jump to constant values. These values are returned by the vision system when nothing is in the field of view. In sum, the four-variable time series in Figure 1 represents the robot moving in a straight line past an object on its left, which is visible for roughly 1.8 seconds and then disappears from the visual field.

Every time series that corresponds to moving past an object has qualitatively the same structure as the one in Figure 1. It follows that if we had a statistical technique to group the robot's experiences by the characteristic patterns in multivariate time series (where the variables represent sensor readings), then this technique would in effect learn a taxonomy of the robot's experiences. We have developed several such techniques, called, collectively, *clustering by dynamics* [1,2]. One method, developed by Tim Oates in our laboratory, is based on dynamic time warping:

1. A long multivariate time series is divided into segments, each of which represents an *episode* such as moving toward an object, avoiding an object, crashing into an object, and so on. (Episode boundaries are inserted by humans, but in other work we use a simple algorithm that looks for simultaneous changes in multiple state variables.) We did *not* label the episodes in any way.
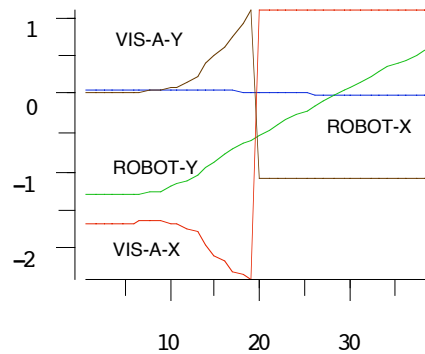


**Figure 1.** As the robot moves, an object approaches the periphery of its field of view then passes out of sight.

2. A *dynamic time warping* algorithm compares every pair of episodes and returns a number that represents the degree of similarity between the time series in the pair. Dynamic time warping "morphs" one multivariate time series into another by stretching and compressing the horizontal (temporal) axis of one series relative to the other [14]. If two multivariate series are very similar, relatively little stretching and compressing is required to warp one series into the other. A number that indicates the remaining mismatch after the optimal warping is thus a proxy for the similarity of two series.

3. Having found this similarity number for every pair of episodes, it is straightforward to cluster episodes by similarity. An agglomerative algorithm merges clusters until a simple stopping criterion is met.

4. Another algorithm finds the "central member" of each cluster, called the cluster *prototype* after Rosch [13].

In a recent experiment involving over 150 episodes, we first clustered the episodes by hand and then ran the algorithm. Then, for all pairs of episodes, we measured the number of pairs that were classified as belonging to the same (or different) clusters by the human and by the algorithm. Remarkably, the concordance between these classifications exceeded 90% [1].

These clusters were learned largely without supervision and constitute a primitive ontology of activities – the robot learned some of the activities it can do in its environment. What supervision or help did we provide? We wrote the programs that controlled the robot and made it do things. We divided the time series into episodes (although this can be done automatically). We limited the number of variables that the dynamic time warping code had to deal with, as it cannot efficiently handle multivariate series of forty state variables. We did not label the episodes to tell the learning algorithm which clusters of activities it should consider. In fact, the only guidance we provided to the formation of clusters was a threshold statistic for adding an episode to a cluster. To reiterate, we did not anticipate, hope for, or otherwise coerce the algorithms to learn

*particular* clusters and prototypes. The robot's ontology of activities is its own.

## Prototypes from Delay Coordinate Embeddings

To cluster episodes, one needs to judge their similarity. This function was accomplished by dynamic time warping in the previous study. Another way to do it is with delay coordinate embedding [2]. Let an episode $e_i = X_{i,t-\tau}, \mathrm{K}, X_{i,t-1}, X_{i,t}$ be a time series of sensor $X$ for $\tau + 1$ time steps ending with an event at time $t$. Clearly, $e_i$ can be represented by a single point $p_i$ in a space of $\tau + 1$-dimensions, as each of $X_{i,t-\tau}, \mathrm{K}, X_{i,t-1}, X_{i,t}$ identifies the location of $p_i$ in one dimension. This transformation of a $\tau + 1$-step time series to a point in $\tau + 1$-dimensional space is called a delay coordinate embedding. The Euclidian distance between two points $p_i$ and $p_j$ is a measure of similarity of episodes $e_i$ and $e_j$.

In one experiment [2] we implemented two simulated agents, each of which adopts one of nine behaviors, including crash, avoid, kiss, and so on. For instance, A might try to *avoid* B while B tries to *crash* into A. We ran several episodes of pairs of behaviors. Each episode was represented by a time series of the distance $d$ between the agents. An episode ended when the agents touched each other or the distance between the agents exceeded a threshold. Delay coordinate embeddings were constructed for each episode, the Euclidean distance $p_i - p_j$ was calculated for each pair of episodes $e_i, e_j$, and these distances were used as a similarity metric for agglomerative clustering of episodes.

With very little training, this procedure came up with six clusters. The prototypes of these clusters, obtained by averaging the time series within each cluster, are shown in Figure 2. Prototypes 1, 2 and 3 involve one agent eluding the other; the difference between the clusters is how close the agents get to each other before moving apart. Prototype 4 represents interactions in which the agents repeatedly draw together and move apart over time, but never touch – a kind of perpetual chase. Prototype 5 is a case in which the agents close on each other so quickly that inertia carries them past each other – they overshoot one another – then they change direction and eventually touch. Prototype 6 is simple case in which the agents rapidly draw together and touch.

Our clustering system was not instructed to cluster episodes by their outcomes (e.g., touch, escape, etc.); indeed, the clusters and their prototypes were constructed entirely without supervision. But time series are so redundant that clustering by dynamics often produces prototypes that have qualitatively different outcomes. As in the previous study, we claim that the prototypes learned by delay coordinate embedding constitute a primitive ontology

of activities. Because the learning is unsupervised, the ontology is not ours, but is acquired through experience. In fact, we were quite surprised to see the "overshoot contact" prototype. It is an emergent interaction of the agents that we had never noticed.
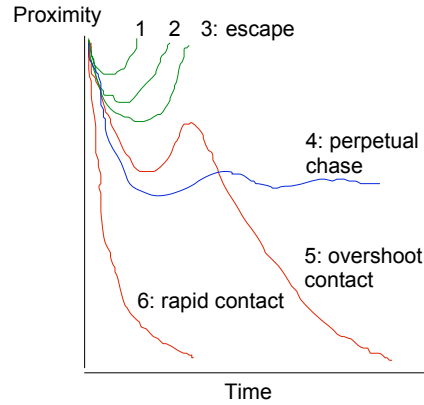


**Figure 2.** Six prototypes for agent interactions, learned without supervision, and plotted as time series of the proximity between the agents.

## From Prototypes to Concepts

Each of the robot's activities has roles. The robot is an *actor*, it approaches an *object*, it pushes the object to a *destination*, past another object on its *left*. Roles are predicates that describe the relationships between the robot's actions and objects in its environment. Familiar roles such as actor, action, object, subject, and instrument, may be augmented with more specific roles such as "the object on my left." General or specific, the idea is the same: A scene is described and presumably represented in terms of the roles of its participants – the causal relations that hold among the participants' actions.

Roles are the key to having our robot learn an ontology through interaction with its environment. Just as we may define "chair" in terms of interaction – a chair is something we sit in – so might our robot ontology include objects that fit between the grippers, objects that can be lifted, objects that do not move, objects that are so large that they are not entirely visible when the robot touches them, and so on. Classes of objects follow directly from roles; indeed, for each role played by an object (e.g., being picked up by the robot) there is a class of objects that have played or could play that role. The same point holds for actions which play roles in activities. If the robot could perceive the roles played by objects and actions, it could develop a conceptual system, an ontology, an organization of objects and actions into classes defined by experience.

Unfortunately, prototypes are monolithic representations of activities, not denoting, compositional representations. In

Figure 1, for example, we see an instance of the robot's prototype *pass-an-object-on-the-left*, but the figure does not individuate an object, or spatial relationships, or the activity of passing. It is just four time series, four lists of numbers. Suppose I removed the labels on the time series in the figure, and the explanations of the series, would you be able to tell that Figure 1 represents a robot passing an object? Of course not: The lines in Figure 1 do not identify a robot or an object as participants in a prototype. The lines might *be* a prototype or an instance of one, but they do not identify participants. For prototypes to become concepts, they have to denote things and relationships explicitly.

To this end we have developed another approach to clustering by dynamics that works not with sensor values but with the output of a primitive perceptual system. This work, described in a companion paper to this one, takes time series of propositions and clusters them. For example, the first ten time steps of an experience are shown below:

```
((STOP R) (IS-RED A))
((STOP R) (IS-RED A))
((APPROACH A R) (IS-RED A))
((STOP R) (IS-RED A))
((RECEDE A R) (STOP R) (IS-RED A))
((IS-RED A))
((MOVING-FORWARD R) (IS-RED A))
((MOVING-FORWARD R) (IS-RED A))
((MOVING-FORWARD R) (IS-RED A))
((MOVING-FORWARD R) (IS-RED A))
```

These propositions are produced automatically by the robot's perceptual system, and they describe the robot approaching a red object, stopping, receding from the object, and then moving forward but not approaching the object (i.e., moving obliquely to it).

Our current work involves extracting concept definitions from prototypes of experiences. Much of the meaning of a token, such as a word, or a sonar reading, is in the roles played by the denotation of the token. For example, the red object denoted by the token A, above, plays the role of something that the robot can approach. This seems trivial and obvious, but it is only a simple example of a general and quite powerful way of learning definitions of objects. The idea we are currently implementing is to first collect objects that play roles in activities, and then inductively extract the features of the objects that cause it to play the roles it has in given activities.

## Conclusion

The claim of this paper is that agents should grow their own ontologies, unsupervised by us, by interacting with their environments. Success would also address a longstanding problem in human development, namely, how conceptual systems arise from exercising motor schemas. I

have argued for the primacy of activities: Concepts may represent activities themselves or participants in the activities.

## References

[1] Oates, T. Schmill, M. and Cohen, P. 1999. Identifying Qualitatively Different Experiences: Experiments with a Mobile Robot. To appear at The Sixteenth International Joint Conference on Artificial Intelligence.

[2] Rosenstein, M and Cohen P.. 1999. Continuous Categories for a Mobile Robot. To appear at The Sixteenth National Conference on Artificial Intelligence.

[3] Schmill, Matthew D., Michael T. Rosenstein, Paul R. Cohen, and Paul Utgoff. 1998. Learning what is relevant to the effects of actions for a mobile robot. In Proceedings of the Second International Conference on Autonomous Agents. Pp. 247-253

[4] R.Baillargeon. How do infants learn about the physical world? Current Directions in Psychological Science, 3:133--140, 1994.

[5] S.Carey and R. Gelman. The epigenesis of mind: Essays on biology and cognition. Hillsdale NJ: Erlbaum, 1991.

[6] Gary L. Drescher. Made-Up Minds. MIT Press, 1991.

[7] Mark Johnson. The Body in the Mind. University of Chicago Press, 1987.

[8] George Lakoff. Women, Fire, and Dangerous Things. University of Chicago Press, 1984.

[9] Lenat, D. and Feigenbaum, E. A. On the thresholds of knowledge. Proceedings of IJCAI -87. pp. 1173 – 1182.

[10] Jean M. Mandler. How to build a baby: On the development of an accessible representational system. Cognitive Development, 3:113--136, 1988.

[11] Jean M. Mandler. How to build a baby: II. conceptual primitives. Psychological Review, 99(4):587--604, 1992.

[12] Jean Piaget. The Origins of Intelligence in Childhood. International Universities Press, 1952.

[13] E. Rosch and C. B. Mervis. Family resemblances: Studies in the internal structure of categories. Cognitive Psychology, 7:573--605, 1975.

[14] David Sankoff and Joseph B. Kruskal (Eds.) Time Warps, String Edits, and Macromolecules: Theory and Practice of Sequence Comparisons. Addison-Wesley. Reading, MA. 1983

[15] E. S. Spelke, K. Breilinger, J. Macomber, and K. Jacobson. Origins of knowledge.. Psychological Review, 99:605--632, 1992.