What are contentful mental states? Dretske's theory of mental content viewed in the light of robot learning and planning algorithms

Paul Cohen
Department of Computer Science, Lederle GRC, University of Massachusetts, Amherst MA 01003
Phone: 413 545 3638; fax: 413 545 1249

Mary Litch
Department of Philosophy
University of Alabama

Abstract: One concern of philosophy of mind is how sensorimotor agents such as human infants can develop contentful mental states. This paper discusses Fred Dretske's theory of mental content in the context of results from our work with mobile robots. We argue that Dretske's theory, while attractive in many ways, relies on a distinction between kinds of representations that cannot be practically maintained when the subject of one's study is robotic agents. In addition, Dretske fails to distinguish classes of representations that carry different kinds of mental content. We conclude with directions for a theory of mental content that maintains the strengths of Dretske's theory.

Content areas: Philosophical foundations

Tracking Number: A260

What are contentful mental states? Dretske's theory of mental content viewed in the light of robot learning and planning algorithms

Tracking Number: A260

Abstract

One concern of philosophy of mind is how sensorimotor agents such as human infants can develop contentful mental states. This paper discusses Fred Dretske's theory of mental content in the context of results from our work with mobile robots. We argue that Dretske's theory, while attractive in many ways, relies on a distinction between kinds of representations that cannot be practically maintained when the subject of one's study is robotic agents. In addition, Dretske fails to distinguish classes of representations that carry different kinds of mental content. We conclude with directions for a theory of mental content that maintains the strengths of Dretske's theory.

Introduction

An empirical philosophy of mind might tackle the question, "How do sensorimotor agents develop contentful mental states," by building a sensorimotor agent and, constrained by certain ground rules, try to have it develop contentful mental states. Our sensorimotor agent is a Pioneer 1 mobile robot, which roams around our lab and records its experiences through roughly forty sensors, controlled by a remote computer via radio modem. The ground rules for the project are designed to counteract the tendency in Good Old Fashioned AI to build systems that do exactly what *we* want them to do: First, while prior or innate structure is necessary, it should be minimized, and the robot should learn most of what it knows. Only when learning proves intractable will we consider adding some prior structure to facilitate the learning. Second, the learning should be unsupervised. Specifically, our learning algorithms either require no training signal, or the signal is endogenous to the robot (e.g., a pain sensor). We will not argue that what the robot learns is completely independent of us, but we do strive to have the robot learn what the environment affords, rather than what we want it to learn.

Constrained by these ground rules, the robot has learned quite a lot. We will argue in this paper that the robot's mental states are representational and contentful. This conclusion presents some difficulties for an account of content ascription due to the philosopher Fred Dretske (1991). One difficulty is that an essential distinction in Dretske's theory between two kinds of representation is not practical; the authorship of mental states, on which Dretske's distinction depends, is ambiguous in learning robots, as Dennett has noted (1992). Second, and more importantly, the class of representations that figures in Dretske's theory of content ascription includes many subclasses, some more "contentful" than others.

The Robot and Some of What It Learns

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 has learned numerous contingencies [references removed for blind review], including dependencies between its actions, the world state, and changes in the world state. More accurately, several algorithms have learned these contingencies by processing data gathered by the robot as it roams around our laboratory. In this section we will focus on one learning method, *clustering by dynamics*, and a primitive ontology of actions that it learned without supervision.

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. Given a little practice, one can see that this short time series represents (in a sense we will explain later) 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.

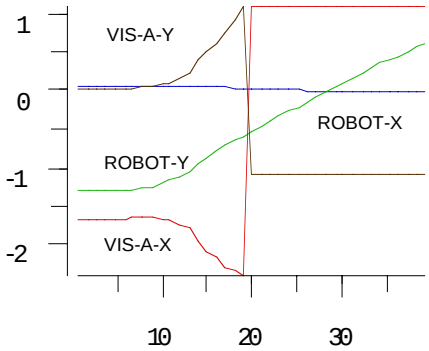

Figure 1. A time series of four sensors that represents the robot moving past an object on its left.

Every time series that corresponds to moving past an object has qualitatively the same structure as the one in Figure 1, namely, ROBOT-Y increases; VIS-A-Y increases to a maximum then takes a constant value; and VIS-A-X either increases or decreases to a maximum or minimum depending on whether the object is on the robot's left or right, then takes a constant value. ROBOT-X might change or not, depending on whether the robot changes its heading or not.

It follows that if we had a statistical technique to group the robot's experiences by the characteristic patterns in time series, then this technique would in effect learn a taxonomy of the robot's experiences. *Clustering by dynamics* is such a technique. The version we describe here was developed by [reference removed for blind review], similar methods are described in [reference removed for blind review]. First, one divides a long time series 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 can be inserted by humans or by a simple algorithm that looks for simultaneous changes in multiple state variables. Obviously we prefer the latter technique (and apply it in [references removed for blind review]) because it minimizes human involvement in the learning process; however, for the experiment described here, episode boundaries were marked by us. We did *not* label the episodes in any way. Second, a dynamic time warping algorithm compares every pair of episodes and returns a number that represents the degree of similarity of the time series in the pair. Dynamic time warping is a technique for "morphing" one multivariate time series into another by stretching and compressing the horizontal (temporal) axis of one series relative to the other (Sankoff and Kruskal, 1983). 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 amount of stretching and compressing is thus a proxy for the similarity of two series. Third, having found this similarity number for the series that correspond to every pair of episodes, it is straightforward to cluster episodes by their similarity. Agglomerative clustering is a method to group episodes by similarity such that the within-cluster similarity among episodes is high and the between-cluster similarity is low. Fourth, another algorithm finds the "central member" of each cluster, which we call the cluster *prototype* following Rosch (1975).

In a recent experiment, this procedure produced prototypes corresponding to passing an object on the left, passing an object on the right, driving toward an object, bumping into an object, and backing away from an object [reference removed for blind review].

We claim that these prototypes were learned largely without supervision and constitute a primitive ontology of activities – the robot learned some of the things it can do. 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. Thus we claim that the robot's ontology of activities is its own.

Recently we have been trying to "close the loop" and have the robot learn enough about the preconditions and effects of its actions that it can plan to accomplish a goal. For instance, suppose the robot wants to drive the state of the bump sensor from low to high (i.e., it wants to bump into something); what should it do?[1] [Reference removed for blind review] discusses how the robot learns models of single actions from its prototypes. But planning is more than just executing single actions. Planning means reasoning about the effects of sequences of actions to achieve a goal state. To plan, the robot needs to transform prototypes into planning operators that specify the preconditions and effects of actions. Actually, prototypes already specify the effects of actions because they are multivariate time series, and the effects of actions are just

[1] The robot's "wants" are implemented by a trivial algorithm that selects a sensor and tries to change the sensor's current value. Obviously, most human wants are more sophisticated, yet we think our simple algorithm is a primitive model of exploratory motor behavior in infants.

the values of state variables over time. The tricky thing is to learn preconditions. First, each episode is labeled with the cluster to which it belongs. Next, the first 1000 msec. time series of each state variable in each episode is replaced by its mean value. These are the "initial conditions," the average values of state variables at the beginning (i.e., the first 1000 msec.) of each episode. Initial conditions are not the same as preconditions. To qualify as a precondition in an episode, an initial condition must at least make good predictions about how the episode will unfold. That is, an initial condition cannot be a precondition for a prototype if it is uncorrelated with that prototype across episodes. We have a batch of episodes, and each is labeled with its cluster membership, and each has a list of initial conditions, so it is a simple matter to run these data through a decision tree induction algorithm to find those initial conditions that best predict the cluster membership of the episodes. Since each cluster is represented by exactly one prototype, these predictive initial conditions are interpreted as preconditions for the prototypes.

Representational States in the Robot

We claim that our robot possesses contentful mental states. More precisely, we claim that, after the learning process, our robot possesses perceptually-based beliefs (a sub-type of mental states). This section is concerned with arguing for that thesis. Recall that the last stage of the learning process involves transforming the prototypes into planning operators that specify the preconditions and effects of actions. A single set of preconditions (one set for each prototype) is the vector of average sensor values that accurately predicts the future series of sensor values when a particular operator is applied. After learning, the robot will perform the operation specified by a prototype whenever both (i) its most recent time series of sensor state values matches the set of preconditions for that prototype, and (ii) it currently has a want that is satisfied by an effect of the operator. We shall henceforth use the term "preconditions satisfier" (abbreviated "PS") to refer to the data structure encoding the time series of sensor state values, *when that time series matches any of the sets of learned preconditions*. (This term is applicable to the sensor state data structure only when a match occurs.)

Note that a preconditions satisfier has several interesting properties. First, PSs are caused by things going on in the environment external to the robot. (This causal relation is indirect and is mediated by the analog to digital converter associated with each sensor.) Second, a PS, when instantiated, causes the robot to act in a way that is appropriate, given the robot's other mental states (in particular, given the robot's wants). ("Appropriate" here means "tends to bring about satisfaction of a want".) A third property of PSs to note is that they are doubly the result of learning. We (i.e., the designers of the robot's controller) do not stipulate which sensor time series states are the preconditions for actions, nor do we preordain which set of preconditions will ultimately be associated with which action. (Indeed, the action types are themselves the result of learning.) Both the actual preconditions and the causal role played by the PSs are determined during the learning process. Notice that the above-mentioned properties are nothing other than the properties associated with perceptually-based beliefs in general: (a) they are caused by something external to the individual, (b) they cause the individual to act in appropriate ways, given the individual's other mental states, and (c) they are the result of learning in the individual's past. (Note that properties (a) and (b) are just the functionalist interpretation of perceptually-based beliefs.) Therefore, we feel justified in saying that PSs are perceptually-based beliefs. Some philosophers of mind (Fodor and LePore, 1992) have added an additional condition: the web of beliefs and desires must attain some critical level of complexity; thus, punctate minds (e.g., minds containing only one belief) are impossible. We reject this complexity condition. Our reason is simple: the model of a cognitive agent that we have uppermost is not an adult human (for whom the complexity condition is appropriate), but an infant. We are trying to understand how mental content can be bootstrapped, given a small primitive set of wants and action types. A major goal of our project is to show how this bootstrapping is possible with limited innate structure. Thus, our rejection of the complexity condition is justified.

Some may object to our argument that PSs are perceptually-based beliefs as follows: we set the terms (by providing the definition of "perceptually-based belief"), so it should come as no surprise that PSs are perceptually-based beliefs. A more legitimate approach would be for us to have used some independent theory of mental states and to have argued that, *according to that analysis*, PSs are mental states. So, let's begin again.

We adopt the analysis provided by Dretske in *Explaining Behavior*. In that work, he provides and motivates a taxonomy of representational states and argues that mental states are one subclass within that taxonomy (a subclass which he names "Type III learned representational states"). The taxonomy is part of a larger project that includes an analysis of mental content in naturalistic terms and a defense of the view that mental content has an explanatory role to play in the behavior of humans and other minded individuals. Space considerations prevent us from a review of the wider project – all we shall focus on here is the claim that PSs are Type III learned representational states (henceforth, "Type III states"). Dretske sets out very clear criteria for being a Type III state. (We first give the criteria in Dretske's terminology, then unpack them in subsequent discussion.) In order to fit the bill for Type III status, PSs must: *indicate* some external condition, have the *function* of indicating that condition, and have this function assigned as the result of a *learning* process. According to Dretske, one physical state *indicates* some external condition when the first state is a state in a larger system,

and the larger system goes into that state when and only when the external condition happens. So, the physical states of a standard thermometer (in particular, the level of mercury) indicate the ambient temperature. Likewise, the states of the data structure encoding the time series of sensor state values indicate certain states of affairs involving the position of the robot relative to objects in the world. Indicators can acquire the *function* of indicating in one of two ways according to Dretske, either by having an outside agent stipulate what an indicator indicates, or as a result of learning. If the function of an indicator is assigned by an exogenous agent, the indicator is not a Type III representation – Dretske calls it a representation of Type II – and it doesn't qualify as a contentful mental state. Only learned indicator functions – Type III representations – qualify.

We shall argue that PSs acquire their indicator functions through learning; although, as we discuss in the next section, there are problems in applying the learning criterion from Drestke's theory to our robot. PSs in the post-learning robot cause the robot to take specific actions to satisfy its wants. PSs have been given this control duty because of what they indicate about the state of the world (namely, that the state of the world is such that execution of these actions is likely to bring about the desired result). The specific control duty assigned to a PS is determined by a learning process. – namely, by the learning algorithm that runs on top of the controller. Thus, PSs are Type III states. Attaching the above argument to Dretske's overall theory, we reach our ultimate conclusion: PSs are mental states.

Difficulties with Dretske

In the above argument, we showed that PSs are Type III states; however, there were a couple of places where the fit wasn't exact (i.e., where the distinctions made within Dretske's theory didn't exactly match the distinctions we want to make in describing the actual robot). The first point of mismatch involves Dretske's understanding of the sort of learning necessary to achieve Type III status. We are certainly not the first to question this aspect of Dretske's theory (Dennett, 1991 and 1992 and Davidson, 1987); however, our take on the issue differs from that taken by others and is based, not on "philosophical" concerns, but on concerns relating to the application of the theory to a concrete system.

We have tried to avoid "reverse engineering" the robot or the robot's environment so as to coerce the end product we are looking for. (The charge of "reverse engineering" arises in both the traditional AI and connectionist approach; the "tweaking" of network parameters that goes on in connectionist research is not qualitatively difference from the "reverse engineering" in traditional AI learning systems.) Even though we have minimized task-specific innate structure, we cannot avoid playing a significant role in the design of the robot's control algorithms and learning algorithms. That said, does our role invalidate the learning achieved by the robot, such that it is not really a Type III representational system? If so, we must ask which if any choices we are allowed to make and still claim Type III status for the robot's representations. If the answer is "none," then according to Dretske's theory our robot will never have contentful mental states; but if the answer is "some," then we face the impossible task of teasing apart the design choices that do and do not prevent our robot's mental states from being contentful.

To illustrate, although we didn't mention this earlier, the prototypes for moving toward and past objects were learned from trials that we set up. We varied the placement of objects relative to the robot and instructed the robot to move. Even so, we had no idea whether the robot's learning algorithm would produce prototypes, we did not anticipate the prototypes it produced, and we were pleasantly surprised that the prototypes made sense to us. So who is the author of the prototypes? Dretske might take a hard line and say that the robot did not develop its prototypes all by itself, so they are not Type III representations, but then he would also have to rule out concepts in a curriculum learned by human schoolchildren. In fact, curricular concepts are even more suspect than our robot's prototypes, because teachers intend their students to believe something while our placement of objects around the robot was not intended to "teach" the robot any particular prototype. On the other hand, if our contribution to what the robot learned does not disqualify its prototypes as Type III representations, then what other help are we allowed to provide? Suppose we classify prototypes as good and bad, and make the robot learn our classification rule. The fact that a learning algorithm does the work of inserting the rule into the robot's knowledge hardly qualifies the rule as a Type III representation, because the rule is entirely conventional and could just as well have been inserted by a programmer – it is *our* rule. We must also consider "in-between" cases like those in reinforcement learning, where a system learns *our* rule by seeking to maximize a reinforcement signal that may well be endogenous; for example, when I reward a child for saying "please," she undoubtedly learns, mediated by her unique reinforcement function, but she is learning *my* rule. In short, the fact that a system learns *p* does not mean that *p* is not conventional, nor does the fact that someone helps a system to learn *p* mean *p* is conventional. Dretske wishes to distinguish types of representations based on whether they are learned, but as he points out himself, learning is not well-defined. Dennett (1992) accuses Dretske of wanting "do it yourself understanding," in which the authorship of concepts belongs entirely to the individual. Among our robots, at least, there are no such individuals and they have no such concepts. The distinction between Type II representations, in which the functions of indicators are assigned, and Type III representations, in which the functions are learned, is not practical.

Even if it were a practical distinction, we find that not all kinds of Type III representations are equally contentful. At a minimum, contentful means "being about something," so when you think of a good bottle of wine, your mental state has content by merit of "being about" the wine. We have learned that the robot's mental states can be in several different relations to the world. Consider again the sensory prototype illustrated in Figure 1. This prototype is "about" passing an object on the left, as it is evoked when the robot starts to pass an object on the left and it makes accurate predictions about the robot's sensory experience during this activity. The prototype in Figure 1 *represents* the robot passing an object on the left (and as a learned prototype, it qualifies as a Type III representation in Dretske's taxonomy) but what does it represent, exactly, what is its *content*? Suppose we told the robot, "Turn to the object on your left," would it understand? No, because although the prototype represents passing an object on the left, it does not *denote* the object, the robot, or the spatial relationships between them. The object, the robot and the spatial relationship between them are not part of the content of the robot's prototype. Prototypes represent the *sensory* experience of activities, they do not denote the roles of participants in the activities. If our robot had prototypes that denote the roles of participants in its activities, these prototypes would be more contentful than the robot's sensory prototypes, though both would be equally Type III representations.

Toward an Explanation of Mental Content

Perhaps there is a better way to explain the content of mental states than Dretske's theory. The problem of naturalizing content – explaining how mental states come to be about something – is not completely solved unless one can explain how we come to have representations that denote the roles of participants in activities, not just simple sensory prototypes. Dretske's taxonomy does not differentiate these types of representation, so his theory cannot naturalize content. We agree with Dretske that learning is an important component of a theory of content, but we disagree with Dretske's use of learning as a criterion for whether mental states are genuinely contentful. The content of a state – what it is about what reasoning it supports – is orthogonal to whether the state is learned. But Dretske is forced to make learning a criterion for genuine content to *divorce* mental states from any possible causes of those states exogenous to the learner. After all, if mental states are caused by something else, then Dretske hasn't explained them until he has explained their cause. Learning serves Dretske as an insulator of mental states from other causes. Yet the fact remains that states can be contentful even if they aren't learned – the only problem is that their content isn't explained. Rather than making learning a criterion for *whether* states have content, we think it should be part of the explanation of *how* states have content. In particular, a theory of the content of mental states would explain how humans make the transition from sensorimotor representations very much like our robot's sensory prototypes, to representations of the roles of participants in activities – representations that support classification of things by their roles, and thus the development of ontologies, and support mental activities that depend on representations of roles, such as language.

References

Davidson, D. 1987. Knowing One's Own Mind. *Proceedings and Addresses of the American Philosophical Association*, pp. 441-458.

Dennett, D. 1991. Ways of Establishing Harmony in B. McLaughlin's (editor*) Dretske and His Critics*, Basil Blackwell. Cambridge, MA. 1991.

Dennett, D. 1992. Do-It-Yourself Understanding. Reprinted in D. Dennett's *Brainchildren*. MIT Press. Cambridge, MA. 1998.

Dretske, F. 1988. *Explaining Behavior*. MIT Press. Cambridge, MA.

Fodor, J. and LePore, E. (editors) 1992. *Holism: A Shopper's Guide*. Basil Blackwell. Cambridge, MA.

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

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