Open-World Learning for Radically Autonomous Agents

Pat Langley

Information Technology and Systems Division
Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311

Abstract

In this paper, I pose a new research challenge – to develop intelligent agents that exhibit radical autonomy by responding to sudden, long-term changes in their environments. I illustrate this idea with examples, identify abilities that support it, and argue that, although each ability has been studied in isolation, they have not been combined into integrated systems. In addition, I propose a framework for characterizing environments in which goal-directed physical agents operate, along with specifying the ways in which those environments can change over time. In closing, I outline some approaches to the empirical study of such open-world learning.

1 Background and Motivation

Our society is increasingly reliant on autonomous physical agents, from self-driving vehicles to planetary robots to military drones. These entities are capable of a broad range of behaviors, from low-level perception and action to high-level reasoning and planning. Despite many differences, these autonomous systems share a dependence on accurate models of the environment in which they operate. The AI community has championed two paradigms for generating such expertise – manual creation of knowledge bases (e.g., Waterman, 1986; Giarratano and Riley, 2018) and learning from large training sets (e.g., Clark and Storkey, 2015). Both approaches have been successful for certain classes of problems, but neither handles an inevitable hurdle for any fielded system: encountering situations where available expertise is incomplete or incorrect. The solution is to develop radically autonomous systems that can revise their expertise in such novel cases. In this paper, I describe this challenge of *open-world learning* more fully and outline ways to address it.

Consider an unmanned underwater vehicle designed for near-surface use in a coastal area. The autonomous agent that controls the platform has reliable expertise for missions in this setting, but suppose it enters a new region with different physical characteristics. Here, it may encounter new types of rock formations, unfamiliar kelp that can foul its propellers, and aggressive predators that may attack it. The vehicle may enter an area where visibility is low and where sonar reflections are distorted or where strong currents threaten to drag it off course. When it approaches a

volcanic vent, the agent may find unexpected upward currents and new chemical reactions that degrade its batteries. A fully autonomous system would realize these fall outside its expertise and adapt rapidly enough to prevent disaster and continue its mission. Such agents must respond to environments that change in ways unforseen by their designers.

In the next section, I define the task of open-world learning for radical autonomy and analyze some component abilities that would support it. After this, I propose a framework for characterizing both environments and transformations of them that can make agent expertise inadequate. Next, I discuss experimental studies of open-world learning, including promising testbeds, relevant variables, and candidate hypotheses. The analysis draws ideas from the description of DARPA's *SAIL-ON* program (Senator, 2019), but it approaches the challenge from a complementary perspective.

2 Elements of Open-World Learning

The first step in any computational research effort is to define the problem in terms of inputs and outputs. We can specify the task of open-world learning as:

- *Given:* An agent architecture that uses expertise to operate in a class of environmental situations;
- *Given:* Expertise that supports acceptable agent performance in this class of environments;
- *Given:* Limited experience with an environment in which sudden, unannounced changes degrade performance;
- *Find:* When environmental changes occurred and what revised expertise will support acceptable performance.

This formulation is abstract so it can apply to a wide range of agents and situations; it does not commit to any particular performance task, but does assume that this involves achieving goals or carrying out missions. Neither does the statement indicate whether initial expertise is handcrafted or learned, but does require that updates be more rapid than environmental change, and thus rely on limited observations.

If we assume that the environmental changes are reasonably stable, then they require some form of long-term learning rather than short-term adaptation. Because only some aspects of the environment will change and because expertise is modular, induction from scratch would be inefficient and would slow the process unacceptably. Moreover, changes

may occur repeatedly and relearning from the ground up each time is undesirable. Clearly, open-world learning for autonomous agents requires a more nuanced response.

This problem statement suggests a number of component abilities that appear necessary for such radically autonomous systems (Senator, 2019). These include:

- A *performance element* that uses known expertise to pursue tasks and achieve goals (e.g., finding and collecting underwater objects with desired features) through mechanisms for perception, inference, planning, and control;

- A *monitoring element* that compares observations with expectations to detect anomalies in both the environment (e.g., sonar unreliable, unfamiliar predators) and in the agent's own behavior (e.g., vehicle veering to right);

- A *diagnostic element* that localizes problems in expertise, generates hypotheses about the causes (e.g., nonreflective surfaces, cross currents, misaligned propeller), evaluates alternative candidates, and selects among them; and

- A *repair element* that revises those facets of expertise deemed to be responsible for the performance problems and corrects them (e.g., updated sonar equations, current-sensitive controller, new propeller model).

These components must be embedded in an architecture that specifies how they interact (Langley et al., 2009). The architecture must also interface with the environment and with the agent's expertise, both of which are subject to change.

Technically, one could approach open-world settings with reinforcement learning methods, which have produced impressive results in game playing (e.g., Clark and Storkey, 2015). However, their need for millions of runs is impractical for physical settings where simulations are unavailable. The problem statement specified *limited* experience with the altered environment and *rapid* repair of expertise. As Senator (2019) notes, a variety of other paradigms are relevant:

- *Model-based diagnosis / repair* (Hamscher et al., 1992), although it focuses on specific cases, not generic models;

- Integrated *generation, execution, and monitoring* of plans (Langley et al., 2017) to achieve an agent's objectives;

- *Metacognition* (Cox, 2007), which inspects traces of basic cognition to find incorrect or incomplete knowledge;

- *Problem reformulation* (Riddle et al., 2013), which alters state and operator representations to make tasks tractable;

- *Change detection* in mining data streams (Pears et al., 2014), which typically addresses classification tasks;

- *Theory revision* (Bay et al., 2002), which uses training cases to improve predictive accuracy of existing models;

- *Scientific discovery* (Džeroski et al., 2007), which finds laws and models to describe or explain observations; and

- *Transfer of learned expertise* (Senator, 2011) to new situations when informed about a change.

Despite their relevance, none of these approaches are enough by themselves to support open-world learning as defined. However, some combination of these techniques, organized into an agent architecture, holds promise for an integrated solution, and more researchers should pursue this objective.

3 Characterizing Environments

Any theory of environmental change depends on some way to characterize possible environments. I will not propose any specific formalism here, but I will suggest some broad outlines. These draw on four widely accepted ideas: environments involve a space-time matrix, structures are situated in this matrix, processes change these structures over time, and the environment obeys certain constraints. This description is distinct from the primary agent's world model, which may use the same format or adopt an entirely different notation. However, the analysis assumes that an environment includes the behavior of nonprimary agents which populate it.

A *spatio-temporal matrix* provides the physical setting in which entities reside, events transpire, and agents operate. We can describe this matrix with two elements:

- *Spatial extent* indicates the number of dimensions and their range of values. An underwater vehicle would operate in three dimensions, whereas an autonomous race car might operate on a bounded two-dimensional track.

- *Spatial-temporal fields* describe attributes that vary over location and time. Their values may be constant, piecewise constant, or a continuous function of location and time. Classic examples are salinity, viscosity, and illumination in sea water, as well as traction and wind on roads.

Structures comprise the entities that populate the world and their associated characteristics. The framework assumes that an environment incorporates two varieties:

- *Physical structures* describe both particular objects and the categories into which they fall. Categories specify attributes, ranges and distributions on their values, and equations or formulae that relate them. Some attribute values are associated with a category (e.g., specific heat), others are constant for a given object (e.g., mass), and still others change over time (e.g., temperature). Composite entities describe constituent elements and configurations.

- *Mental structures* describe the contents of agents' memories. These include perceptions of objects in the external world and inferred beliefs about them (including unobserved entities) and knowledge used to generate them. Perceptions and beliefs need not take the same form as external structures. Agents may also encode beliefs about relationships to others (e.g., enemy or commander).

Processes describe behavior in the environment over time. Differential equations are one formalism for such behavior, but one can also organize them into sets of changes to entities' attributes that co-occur (Arvay and Langley 2016). Three types of processes arise in agent-oriented situations:

- *Physical processes* specify behavior that is independent of any agent activities. They include various forms of motion (e.g., falling), energy transfer (e.g., temperature exchange), and chemical reactions. Physical processes can occur at different rates that depend on fixed parameters and the values of entity or spatial field attributes.

- *Control processes* describe the effects an agent's actions have on external objects' attributes, including its own body. They include activities like turning wheels, applying brakes, and lifting objects. Control processes specify

how derivatives depend on entities' attributes (e.g., agent strength, object mass) and field variables (e.g., road friction, ocean currents), but occur only when agents invoke them (Fox and Long, 2006). In settings with multiple agents, each may have different associated processes.[1]

- *Perceptual processes* translate information about objects in the environment into an agent's internal perceptions. They are influenced by factors like object distance, size, and transparency, occlusion by other objects, and field variables like illumination. Agent characteristics like acuity also influence results. Perceptual models may incorporate physical theories of light and sound transmission.

Constraints specify what situations or events can occur in the environment, as well as structural or numeric conditions on them. These divide into two broad categories:

- *Physical constraints* describe situations that may or may not occur in the physical environment. These might specify that two solid objects cannot occupy the same location, that a given reaction can only occur above a certain temperature, or that a lift action only has an effect if the agent's strength exceeds an object's weight. These correspond to the environment's physical laws.

- *Mental constraints* describe preferences on nonprimary agents' behavior. These include symbolic goals (e.g., reaching a target location or driving a specified route) and associated value functions, as well as conditions under which they are relevant. They may also include social norms (Malle et al., 2015), which specify prescribed or proscribed situations and activities (e.g., staying under the speed limit or halting at red lights). These are soft constraints in that agents can decide to ignore them when they conflict with higher-priority criteria.

Taken together, the spatio-temporal setting, structures, processes, and constraints can describe both single-agent settings and multi-agent scenarios involving social interactions. They provide the ability to specify alternative environments, which is a necessary foundation for open-world learning.

4 Characterizing Environmental Changes

Progress on open-world learning also requires a theory of *novelty* (Senator, 2019). This will aid in the development of methods for revising expertise in response to different types of change and will suggest hypotheses about such methods' effectiveness. If we view learning as search through a space of models (Simon and Lea, 1974), then we need operators that generate new candidates from existing ones. We can state these as transformations on elements in an environment description. Let us consider examples of such transitions.

Spatio-temporal transformations can increase or decrease the spatial extent, giving larger or smaller volume. They can also shift from a Euclidean (flat) environment to one that is non-Euclidean (e.g., sphere or cylinder). Other changes can alter the character of spatial fields by adding or removing attributes like salinity or viscosity from locations. Still others might alter the distribution of values for a field attribute, say

from a setting with constant viscosity to one that varies with location. Similarly, a transition might increase the speed of water or air currents or even reverse their flows entirely.

Structural transformations can add or remove object categories (e.g., vehicle types or species), introduce new attributes (e.g., texture), and shift the distribution of values for categories' attributes. Other transitions might alter the possible types of composite objects (e.g., chemical compounds) or their parameters (e.g., bond strengths). Changes to agents' mental structures can affect their perceptions (e.g., which objects are detected) and beliefs (e.g., who is an ally).

Process transformations can introduce new physical, control, or perceptual mechanisms or eliminate existing ones (e.g., adding hail storms, allowing jumping actions, creating blind spots). Alternatively, they might alter parameters in an existing process (e.g., turning speed) or the associated derivatives (e.g., rate expressions in chemical reactions). Changes to control and perceptual processes can alter the abilities of both the primary agent and others that interact with it. Since processes refer to entities and field attributes, shifts in the latter can indirectly influence dynamic behavior.

Constraint transformations can change the natural laws that govern structures (possible categories or configurations) and processes that affect them. The latter may revise the types of entities to which a process applies (e.g., which categories are digestable) and the conditions under which they are active (e.g., temperature limits). They can also shift agents' goals and values (e.g., the priority of collecting samples) or, in social settings, revise the norms that influence group behavior (e.g., changing the rules in a race). Constraint transformations can alter the situations that an agent encounters and the ways that they evolve over time.

A full theory of environmental novelty would include a formalism for stating such transitions, which we can then use to specify trajectories through the space of possible worlds.

5 Experiments with Open-World Learning

Machine learning has been an empirical discipline for over three decades (Kibler and Langley, 1989), but open-world learning raises special challenges for researchers and we should consider the form experiments might take in this area.

Domains and testbeds. The objective is to develop radically autonomous agents that pursue goal-oriented activities in physical environments. Thus, natural domains for research include underwater exploration, aerial reconnaissance, self-driving vehicles, and planetary rovers. Open-world learning is important for such systems because they will encounter situations where their expertise is incomplete or incorrect. However, real-world environments offer limited opportunities to inject novelty, which makes high-fidelity simulators more suitable for controlled experiments (Langley et al., 1981; Choi et al., 2007). These should support complex scenarios with many objects of distinct categories, field attributes whose values vary with location, and processes that produce dynamic behavior. Most important, the simulation engine should accept configuration files that specify types of objects, distributions for their attributes, spatial functions

---

[1] Each agent may also include mental processes that control its decision making, but I will not address this factor here.

for field variables, and functions, parameters, and conditions for processes. These will let researchers introduce environmental novelty in straightforward and systematic ways.

Dependent measures. Scientific experiments examine one or more dependent variables that reflect behaviors of interest. For AI systems, these variables involve some aspect of system *performance*, such as diagnostic accuracy or number of goals achieved. Similar metrics appear in most papers on batch induction, which report behavior after learning has ended, but studies of online induction report *learning curves*, borrowed from psychology, that plot performance against number of training cases. Experiments on transfer may report pairs of learning curves: one in the presence of prior training and another without it. We can combine these two ideas for open-world settings to produce *novelty response curves*. These involve a single plot that maps performance over time and marks time steps on which novelty is introduced. Performance should degrade after the environment changes and recover as the learner detects and adapts to the new situation. This will produce a scalloped curve that alternates between increases and decreases in scores.

Ideally, experiments should not measure success in responding to novelty based on a single extended run, but instead average over tens of trials to ensure reliable results. However, if we introduce novelty at random intervals, as seems desirable to avoid agent anticipation, then the times at which it occurs will not be aligned across runs. One response is to shrink or expand each segment to enforce alignment, but this would lose information about how rapidly the agent adapts to changes. Another approach is to measure both the time needed to detect each environmental change and the rate of performance improvement after detection. This would evaluate different elements of the architecture separately and support credit assignment (Senator, 2019).

Independent variables. Scientific experiments vary one or more controllable factors to determine their effects on dependent measures. Traditional studies of computational induction alter some aspect of the learning method (e.g., modules or parameters) or features of the domain (e.g., noise level). These remain relevant to experiments on open-world learning, but other factors are more intriguing. The most obvious concerns the *type* of change introduced into the environment. The Proposers' Day presentation for DARPA's SAIL-ON program (Senator, 2019) identifies nine varieties of novelty relating to object classes, object attributes, representations, relations, interactions, capabilities, environment, goals, and context. Each level corresponds directly to some facet of the current framework,[2] so a simulation engine that covers similar environments should support all forms of SAIL-ON novelty. Two other independent variables also seem highly relevant to open-world learning: the frequency of novelty introduction and number of changes in each case. The latter is analogous to distance between source and target in studies of transfer (Könik et al., 2009; Senator, 2011).

---

[2]Object classes / attributes $\Rightarrow$ physical structures, representations $\Rightarrow$ mental structures, interactions $\Rightarrow$ physical processes, capabilities $\Rightarrow$ control processes, environment $\Rightarrow$ spatial setting, relations $\Rightarrow$ physical constraints, goals / context $\Rightarrow$ agent constraints

Hypotheses and experimental designs. Good AI experiments are designed to answer scientific questions, with the least informative being whether an entire system outperforms another one on test problems. More interesting issues concern which aspects of the system or domain exert the most influence on behavior. For open-world learning, we might examine what types of change are most difficult to handle and how different elements of the agent architecture respond to them. The SAIL-ON program description offers a tentative ranking on nine types of novelty, suggesting hypotheses like:

- *Agents will have less difficulty recovering from changes to categories and attributes (structures) than to changes involving interaction and capabilities (processes).*

Before we can test such claims, we must make them operational by specifying explicit performance tasks, such as collecting desired objects while maintaining a given power level. We must also introduce changes thoughtfully. For instance, altering the landscape's color may have no impact on object collection, and changing the densities of objects may indirectly affect the actions that manipulate them.

Other issues, such as whether changes are global or local, may also be important. Consider a second hypothesis:

- *Global changes will degrade performance more than local ones, but agents will recover from them more rapidly.*

Because global shifts (e.g., in ocean viscosity) impact many behaviors, they will affect performance broadly but also generate more training data to support detection and repair. We should also examine how other factors influence open-world learning. For example, increasing perceptual noise may slow an agent's ability to detect that its world has altered, but have no effect on diagnosis or repair. Researchers interested in this arena should consider carefully what questions they want to answer before designing their experimental studies.

6 Closing Remarks

This paper defined the new problem of open-world learning, which is essential for autonomous systems to overcome outmoded expertise in changing environments. The analysis identified four elements – performance, monitoring, diagnosis, and repair – that an agent architecture needs for timely change detection and model revision. Although a number of existing AI paradigms offer relevant technologies, we saw that none of them, by themselves, will suffice. In addition, the common view of learning as search suggests that we adopt a representation for states (models of the environment) and operators (changes to these models).

This led to a framework that describes states in terms of spatial settings, structures, processes, and constraints, and that characterizes operators in terms of transformations on these elements. Finally, the paper discussed empirical studies in this area, including dependent measures, independent variables, and hypotheses that experiments might test. Open-world learning offers new challenges for the AI community but also holds great promise. I encourage researchers to develop systems that can identify when their expertise is inadequate, identify the flaws responsible, and overcome them rapidly and effectively to support truly radical autonomy.

References

Aha, D. A.; Cox, M. T.; and Muñoz-Avila, H. eds. 2013. *Goal Reasoning: Papers from the ACS Workshop.* Baltimore, MD.

Arvay, A.; and Langley, P. 2016. Heuristic Adaptation of Quantitative Process Models. *Advances in Cognitive Systems* 4: 207–226.

Bay, S. D.; Shapiro, D. G.; and Langley, P. 2002. Revising Engineering Models: Combining Computational Discovery with Knowledge. In *Proceedings of the Thirteenth European Conference on Machine Learning*, 10–22. Helsinki, Finland.

Choi, D.; Morgan, M.; Park, C.; and Langley, P. 2007. A testbed for evaluation of architectures for physical agents. In *Proceedings of the AAAI-2007 Workshop on Evaluating Architectures for Intelligence*, 19–22. Vancouver, BC: AAAI Press.

Clark, C.; and Storkey, A. 2015. Training Deep Convolutional Neural Networks to Play Go. In *Proceedings of the Thirty-Second International Conference on Machine Learning*, 1766–1774. Lille, France.

Cox, M. T. 2007. Perpetual Self-Aware Cognitive Agents. *AI Magazine* 28: 32–45.

Džeroski, S.; Langley, P.; and Todorovski, L. 2007. Computational Discovery of Scientific Knowledge. In S. Džeroski and L. Todorovski, eds., *Computational Discovery of Communicable Scientific Knowledge*. Berlin: Springer.

Forbus, K. D. 1984. Qualitative Process Theory. *Artificial Intelligence* 24: 85–168.

Fox, M.; and Long, D. 2006. Modelling mixed discrete-continuous domains for planning. *Journal of Artificial Intelligence Research* 27: 235—297.

Giarratano, J. C.; and Riley, G. D. 2018. *Expert Systems: Principles and Programming* (4th ed.). Pacific Grove, CA: Brooks-Cole Publishing.

Hamscher, W.; Console, L.; and de Kleer, J. 1992. *Readings in Model-Based Diagnosis*. San Francisco, CA: Morgan Kaufmann

Kibler, D.; and Langley, P. 1988. Machine Learning as an Experimental Science. In *Proceedings of the Third European Working Session on Learning*, 81–92. Glasgow, Scotland: Pittman.

Könik, T.; Choi, D.; Shapiro, D.; Park, C.; Nejati, N.; Langley, P.; and Stracuzzi, D. 2007. Structural Transfer of Cognitive Skills. In *Proceedings of the Eighth International Conference on Cognitive Modeling*. Ann Arbor, MI.

Langley, P.; Choi, D.; Barley, M.; Meadows, B.; and Katz, E. P. 2017. Generating, Executing, and Monitoring Plans with Goal-Based Utilities in Continuous Domains. In *Proceedings of the Fifth Annual Conference on Cognitive Systems*. Troy, NY.

Langley, P.; Laird, J. E.; and Rogers, S. 2009. Cognitive Architectures: Research Issues and Challenges. *Cognitive Systems Research* 10: 141–160.

Langley, P.; Nicholas, D.; Klahr, D.; and Hood, G. 1981. A Simulated World for Modeling Learning and Development. In *Proceedings of the Third Annual Conference of the Cognitive Science Society*, 274–276. Berkeley, CA.

Malle, B. F.; Scheutz, M.; and Austerweil, J. L. 2015. Networks of Social and Moral Norms in Human and Robot Agents. In *Proceedings of the International Conference on Robot Ethics*, 3–17. Lisbon, Portugal.

Pears, R.; Sakthithasan, S.; and Koh, Y. S. 2014. Detecting Concept Change in Dynamic Data Streams: A Sequential Approach Based on Reservoir Sampling. *Machine Learning* 97: 259–293.

Riddle, P. J.; Barley, M. W.; and Franco, S. M. 2013. Problem Reformulation as Meta-Level Search. In *Poster Collection of the Second Annual Conference on Advances in Cognitive Systems*, 199–216. Baltimore, MD.

Senator, T. E. 2011. Transfer Learning: Progress and Potential. *AI Magazine* 32: 84–86.

Senator, T. E. 2019. Science of AI and Learning for Open-world Novelty (SAIL-ON). Presented at the Proposers' Day Meeting. DARPA: Arlington, VA. *https://www.darpa.mil/attachments/SAIL-ON Proposers Day_Distro A_no notes.pdf*

Simon, H. A.; and Lea, G. 1974. Problem Solving and Rule Induction: A Unified View. In L. W. Gregg, ed., *Knowledge and Cognition*. Hillsdale, NJ: Lawrence Erlbaum.

Waterman, D. A. 1986. *A Guide to Expert Systems*. Reading, MA: Addison-Wesley.