Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference

A Framework for the Qualitative Comparison of Diverse Developmental Agents

Lisa B. Soros and Avelino J. Gonzalez
Intelligent Systems Lab
University of Central Florida, USA

Abstract

Developmental agents learn task-nonspecific skills through environmental interactions. The humanlike flexibility of such agents isn't captured by domain-specific performance metrics. We present a novel framework that complements traditional metrics by allowing cross-domain comparison. The framework considers four properties of developmental agents: the degree of human involvement in design, the length of the agent's developmental period, the architectural support for acquiring new behaviors, and the tolerated dimensionality of input. This framework is applied to real-world systems in three case studies. We find that our framework allows cross-domain comparison that would not be contributed by traditional quantitative metrics.

Introduction

Alan Turing envisioned the following direction for AI research: "Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain" (Turing 1950). Turing's vision finds modern realization in *developmental* AI, which emphasizes childlike agents whose cognitive abilities self-organize through autonomous environmental interactions (Weng et al. 2001; Guerin 2011).

Developmental agents are typically measured using domain-specific metrics. While these metrics quantify mastery of isolated tasks, they do not answer questions about agents' developmental potential. Such potential allows humans to become masters of many domains, not just isolated tasks. As a goal of artificial intelligence research is to create agents with humanlike behavior, developmental potential informs us of an agent's potential for general intelligence.

What we need, then, is a framework which allows us to compare agents' prospects for development across dissimilar domains. In this paper, we propose such a framework and show how it can be applied to developmental systems via three subjective case studies.

Copyright © 2013, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Domain-Specific Metrics

The most popular domain-specific metric is *error rate* (Baranes and Oudeyer 2011; Luciw et al. 2011), which measures the difference between an agent's behavior and some optimal behavior. Similar metrics include confusion rate (Boucenna, Gaussier, and Hafemeister 2011), Euclidean distance (Clune and Lipson 2011), and sum of squared errors of prediction (SSE)(Ikle and Goertzel 2011). These metrics all answer the same question: How well is the agent performing some task?

Unfortunately, asking this question limits the possible comparisons we can make; two agents must be designed to do the same task for the error rate comparison to be valid. However, developmental agents are task-nonspecific and warrant a method of comparison with stronger predictive utility. For instance, human development is measured and predicted using a series of milestones. These milestones incorporate both physical and mental growth, such as learning to walk or understanding language (Berk 2012). However, such milestones were derived from observation of complete human lifespans. We can't derive a similar set of developmental milestones for artificial systems because we haven't yet come up with any that have achieved approximately human-level intelligence. What we need to do is ask bigger questions about our systems - questions that allow us to assess not only their current level of intelligence, but also their potential for acquiring complex behaviors over a lifetime.

Our Qualitative Framework

1. **To what extent are humans involved in the design process?** Humans do not always know the optimal way to solve some problem. Recent work supports this position quantitatively: Bongard varied the ratio of human-designed components to evolved components in robotic arms and then evaluating them on multi-objective tasks (Bongard 2010). The results showed that decreasing human involvement had two benefits: it increased performance on expected tasks and also increased robustness when facing unexpected, yet related, ones. These benefits were expressed more strongly as task complexity increased.
2. **How long is the agent's developmental period relative**

to its lifespan? Things change all the time. It is not always easy to predict the ways in which things will change; changes frequently occur before we can formulate plans to deal with them. Robustness in the face of such change requires the capacity to change oneself in response to external conditions. Such change is impossible if an agent's behavior becomes fixed after a limited developmental period. Thus, the length of an agent's developmental period gives insight about the agent's ability to adapt.

3. **To what extent does the system's architecture allow the acquisition of new behaviors?** Adaptation is critical to general intelligence, so we must critically analyze an agent's structure in order to accurately assess its ability to innovate. For instance, an agent that chooses from a fixed set of actions is constrained in the number of new behaviors it could display. A more flexible architecture, such as one that utilizes bootstrapping, would be more effective. This property is difficult to evaluate quantitatively in a comparative setting, but that it is critical to dynamic adaptivity (Raibulet and Masciadri 2009).

4. **What dimensionality of input is tolerated?** It is apparent that complex human behaviors rely on rich internal representations (Oudeyer 2010; Steels 2003). In order for these representations to have depth, they must be derived from detailed sensory data. An artificial developmental agent should similarly exploit highly dimensional sensory data. It is not expected that the agent would retain all of this detail in its mental representations; it is advantageous (and perhaps necessary, for the sake of computational tractability) for the agent to compress perceptual observations. However, assessing the dimensionality of the raw sensory data gives us a feel for the types of representations that the agent can create (and thus, its potential for executing complex behaviors).

Case Studies

Architecture Descriptions

Case Study 1 (CS1) The first system is a curiosity-based reinforcement learning (RL) agent that learns to navigate an initially unknown maze (Luciw et al. 2011). The agent has a noisy birds-eye view of itself and must repeatedly choose to move one square up, down, left, or right until it reaches a marked goal. The system architecture (Fig. 1) has three major components: 1) a manually-parameterized perceptual system that compresses and encodes sensory observations as states; 2) a cognitive system that calculates transitions between perceptual states and potential rewards; and 3) a value system that uses RL methods to formulate a policy based on the expected future discounted rewards.

The first RL method is SARSA (Rummery and Niranjan 1994). SARSA is a state-based RL algorithm for learning a Markov decision process. At each time step, the agent uses its current state to update its Q-value estimate for its available state-action pairs. This update is based on a discount factor (γ), which determines the significance of future rewards, and a manually-specified learning rate (α). The expression for the SARSA update at time t is as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

The agent combines SARSA with LSPI (Lagoudakis and Parr 2003) to construct a Markov decision process from environmental samples. These samples take the form (s, a, r, a'), denoting initial state s, executed action a, resulting reward r, and successive state s'. Using the model built from such samples, the agent iteratively improves its behavioral policy until it exceeds some acceptability threshold.

Case Study 2 (CS2) The second system is a robotic head that develops joint attention and social referencing skills via interactions with a human caregiver (Boucenna, Gaussier, and Hafemeister 2011). The human-robot interactions proceeds in three phases. In the first phase, the robot learns to recognize and imitate the human's facial expressions. It randomly selects and displays a pre-programmed expression, which the caregiver mimics (thus triggering an associative mechanism within the robot's neural network). In the second phase, the robot learns to recognize what the human caregiver is looking at and direct its gaze at the same object. This proceeds in a similar manner as the first phase, with the robot and human taking turns imitating each other. In the final phase, the robot combines its newly-acquired skills to learn to follow the human's gaze, recognize the subject of the caregiver's attention, perceive the caregiver's displayed emotion, and associate that emotion with the object of attention. This process is a form of bootstrapping, in which simple behaviors are used to build more complex ones.

This system is noteworthy because the social referencing skills emerge from a simple sensorimotor architecture. This architecture is largely composed of a neural network, which is further composed of specialized sub-networks.

Case Study 3 (CS3) The third case study examines R-IAC (Baranes and Oudeyer 2011), an intrinsically motivated developmental robot that chooses actions to maximize learning of its sensory and sensorimotor spaces (Fig. 2). That is, it learns the control parameters for its robotic limbs in order to effectively explore its task environment.

The R-IAC algorithm proceeds as follows: first, the Action Selection Machine probabilistically chooses an action to execute based on the current sensory context. The Prediction Machine then calculates the expected consequences of this action. Next, the actual consequences are observed in the environment. The Prediction Machine takes the difference between the expected and actual consequences and thus computes the prediction error. It then updates itself according to the sensorimotor context and the actual, measured consequence. The Split Machine uses the error, the sensorimotor context, and the measured consequence in order to update the internal region tree. Finally, the Prediction Analysis Machine updates its evaluation of the learning progress for the regions covering the sensorimotor context and measured consequence.

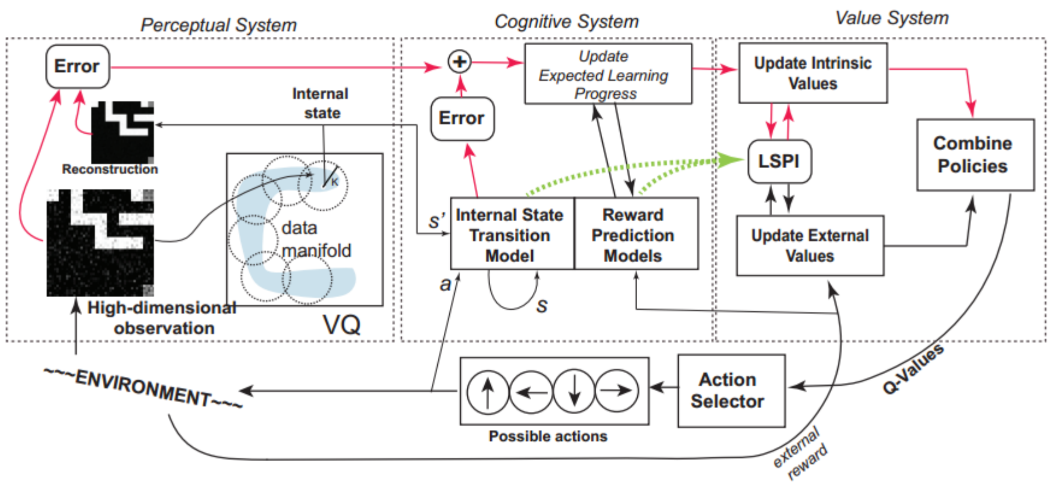

Figure 1: Architecture of Case Study 1, from (Luciw et al. 2011)

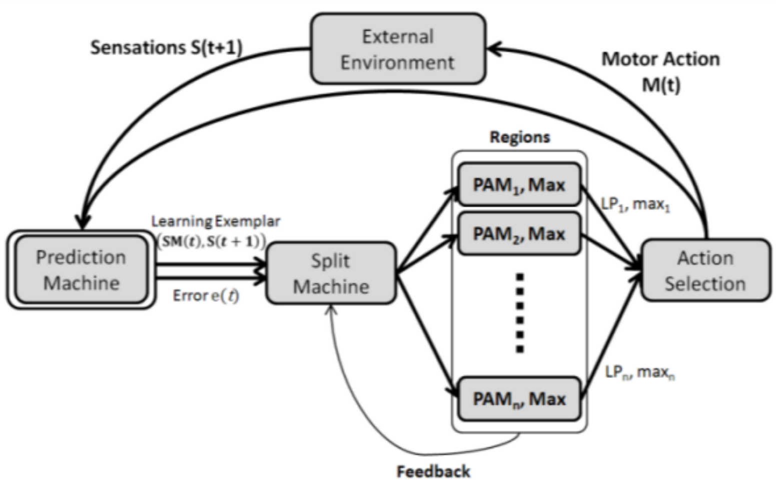

Figure 2: Architecture of Case Study 3, from (Baranes and Oudeyer 2011)

Application of Framework

To what extent are humans involved in the design process? Though CS1's behavioral policies are automatically generated, execution relies on manually-specified parameters, including error averaging rates, initial long-term error, learning rate lower bound, future reward discount factor, and number of steps between LSPI planning phases.

Humans are only really involved in the high-level design of CS2. This is because CS2 relies on neural networks as its base. While humans may have influence over the structure of the networks, they are certainly not determining the fine-grained details such as connection weights.

The design of CS3 allows for different modules to be incorporated (i.e., the original authors are agnostic about the algorithm behind the prediction learning machine). However, the designers manually specify the three strategies that the agent uses for exploration, in addition to the probabilities that the agent will pick each of the strategies.

How long is the agent's developmental period relative to its lifespan? The perceptual, cognitive, and value maps in CS1 update throughout the agent's lifespan. This causes the agent to be continually trying to find a new place to explore and thus satisfy its curiosity drive.

CS2's skills develop via training periods of interaction with a human. There are distinct phases during which the neural network is either learning or not learning. So, though an arbitrarily large number of training/acting phase pairs could occur during the lifetime, we cannot count the non-training time as belonging to a developmental period.

CS3 learns continually as long as it has new sensorimotor space to explore. Given a sufficiently rich environment, the developmental period might extend throughout the agent's lifetime. There may, however, be a limitation imposed on the system by its internal region tree (the explored sensorimotor space's representation) when the dimensionality of the sensorimotor space grows exponentially.

To what extent does the system's architecture allow the acquisition of new behaviors? CS1 is limited; its intrinsic motivation system allows it to self-generate new policies, but, it is incapable of generating new actions. Since the possible actions are manually specified during the design process, a human would have to reprogram the system for it to acquire new behaviors.

CS2 learns new behaviors via its interactions with its caregiver using a vision-based mechanism. However, this mechanism could easily extend beyond vision as it is implemented using a generic neural network. Additionally, the system's bootstrapping mechanism allows it to not only learn new simple behaviors, but to combine its existing behaviors into new, more complex ones.

The algorithm underlying CS3 was intentionally written to generalize. In fact, the reason that the agent learns the skills that it does in the original experiment is because of the particular sensory capabilities of the simulated robot. Thus, the generality of the architecture of the developmental system itself lends itself to the acquisition of new behaviors - however, it must be incorporated into an agent with adequately informative sensors.

What dimensionality of input is tolerated? CS1 tolerates high-dimensional visual data and handles noise well when creating internal representations. Though the described implementation was only simulated, the system's focus on sensory data compression would allow performance to scale well with increased dimensionality and noise.

CS2 also tolerates fairly high-dimensional visual input; its robotic eyes perform online facial recognition and can follow a human's gaze. However, the agent doesn't learn to recognize faces on its own; the skill is pre-programmed. Thus, the system's ability to tolerate one type of raw input does not necessarily extend to other dimensions.

Like CS1 and CS2, CS3 operates on real-world visual data. However, its sensory/state channels are designed to accommodate arbitrary types of input (the creators cite torque motor values and touch sensor values as examples). They can also interpret feedback from higher-level cognitive mechanisms.

Discussion

- Both CS1 and CS3 rely on human-specified parameters (with CS1 doing so to a greater extent than CS3). CS3 is better than both of its competitors in this regard because the relevant parameters are learned over time.
- CS2 has the smallest development-to-lifespan ratio. CS1 and CS3 were approximately equal in this regard, with CS1 showing slightly greater ability for maintaining the developmental period over an arbitrarily long lifespan.
- CS1 in its current format does not support novel behaviors. CS2 and CS3 rely on fairly general algorithms, and show more potential for development in this regard.
- Of the three systems, CS3 leads the pack in terms of displayed potential for tolerating input with arbitrarily high dimensions. CS1 comes in second place, followed by CS2.

No system is comprehensively best; each has strengths and weaknesses. Future work will consider more case studies and investigate deeper architectural features.

References

Baranes, A., and Oudeyer, P.-Y. 2011. The interaction of maturational constraints and intrinsic motivations in active motor development. In *Proceedings of the 2011 IEEE Conference on Development and Learning (ICDL)*, volume 2.

Berk, L. E. 2012. *Infants and Children: Prenatal through middle childhood*. Pearson.

Bongard, J. 2010. The utility of evolving simulated robot morphology increases with task complexity for object manipulation. *Artificial Life* XI.

Boucenna, S.; Gaussier, P.; and Hafemeister, L. 2011. Development of joint attention and social referencing. In *Proceedings of the 2011 IEEE Conference on Development and Learning (ICDL)*, volume 2.

Clune, J., and Lipson, H. 2011. Evolving three-dimensional objects with a generative encoding inspired by developmental biology. *ACM SIGEVOlution* 5(4):2–12.

Guerin, F. 2011. Learning like a baby: A survey of artificial intelligence approaches. *The Knowledge Engineering Review* 26(2):209–236.

Ikle, M., and Goertzel, B. 2011. Nonlinear-dynamical attention allocation via information geometry. In *Artificial General Intelligence*, volume 6830/2011 of *Lecture Notes in Computer Science*. 62–71.

Lagoudakis, M. G., and Parr, R. 2003. Least-squares policy iteration. *Journal of Machine Learning Research* 4:1107–1149.

Luciw, M.; Graziano, V.; Ring, M.; and Schmidhuber, J. 2011. Artificial curiosity with planning for autonomous perceptual and cognitive development. In *Proceedings of the 2011 IEEE Conference on Development and Learning (ICDL)*, volume 2.

Oudeyer, P.-Y. 2010. On the impact of robotics in behavioral and cognitive sciences: from insect navigation to human cognitive development. *IEEE Transactions on Autonomous Mental Development* 2(1):2–16.

Raibulet, C., and Masciadri, L. 2009. Evaluation of dynamic adaptivity through metrics: an achievable target? In *Proceedings of the 4th International Conference on Software Engineering Advances*, 341–344. IEEE.

Rummery, G. A., and Niranjan, M. 1994. On-line q-learning using connectionist systems. Technical report, Cambridge University.

Steels, L. 2003. Intelligence with representation. *Philosophical Transactions of the Royal Society A* 361(1811):2381–2395.

Turing, A. 1950. Computing machinery and intelligence. *Mind* LIX(236):433–460.

Weng, J.; McClelland, J.; Pentland, A.; Sporns, O.; Stockman, I.; Sur, M.; and Thelen, E. 2001. Autonomous mental development by robots and animals. *Science* 291(5504):599–600.