Curiosity

Time is Money: No More Thumb Twiddling

Jordi Bieger, Icelandic Institute for Intelligent Machines, jordi@iiim.is

Many people believe that curiosity is a major indicator of a person's intelligence. While scientific evidence to support that claim is scarce, it makes intuitive sense that the drive to learn and explore the unknown would expand one's knowledge and by extension the ability to achieve goals with insufficient resources and slightly less insufficient knowledge. Of course, this is not just the case for humans and other animals, but for any generally intelligent system.

It has been mentioned in our lectures that when the current implementation of AERA is given the opportunity to reflect on its recent training session in the absence of new input, it will just sit there and twiddle its thumbs. This is clearly inefficient, because as they say: time is money. To say time is one of the limited resources that any intelligent system must effectively manage would perhaps be less eloquent, but not less true. Other resources that the system must manage are energy and useful input data, since it can only observe (and pay attention to) a small fraction of the world.

In Schmidhuber's formal theory of curiosity he considers two aspects of an intelligent system: A) a system that learns to better represent the observations it has had in the past (a predictor or compressor) and B) a reinforcement learning system that selects the actions that the whole system takes. Curiosity can then be implemented by using the predictor's improvements as rewards for the reinforcement learner. This will cause the reinforcement learner to seek out situations where much can be learned by the predictor. These situations are neither too simple nor too chaotic, because in either case not much can be learned. Schmidhuber uses this theory to explain art, music, humor, science, fun and active learning.

Schmidhuber's theory focuses on getting better at prediction. When implemented in any intelligent system, it will steer that system towards utilizing the scarce resource of "input data" in a way that helps it optimize its predictions. In AERA, every model is essentially a prediction: if the antecedent happens, expect the consequent within a certain timeframe with a certain certainty. Since AERA is a self-improving system, getting better at prediction means getting better at everything, including managing time and energy. Furthermore, targeted exploration can help eliminate bad models and strengthen good models to the point where they might be compiled. Both effects help minimize time and energy (working memory) usage.

As Oudeyer and Kaplan point out, the way in which the learning progress is measured matters a great deal. Simply looking at the system's total prediction error over time won't lead to good results, because it can easily be "gamed". In their Intelligent Adaptive Curiosity model they suggest keeping track of improvement in several situations which they determine with classifiers. In AERA this idea is taken to the extreme, because the learning improvement can be measured on a per-model basis.

Unfortunately it will not always be possible for the system to indulge its curiosity and *explore*. When the system is fully operational its resources may be required to perform the task at hand as best as it can: it should *exploit* its current knowledge and abilities. To deal with this famous exploration/exploitation tradeoff (amongst other things) Steunebrink et al. suggest a framework of Work-Play-Dream (WPD). When working, the system should serve the user as best as it can (exploitation). During playtime it can indulge its curiosity by seeking out novel situations and knowledge. Since exploration necessarily means not taking the action that is thought best at the current moment, some amount of supervision may be necessary to keep it from harming itself or guide it otherwise. While dreaming the system can perform time-intensive operations such as compression (in AERA), it can process information it didn't get the chance for during work or play or it can start making plans for "tomorrow".

While it is interesting to consider how curiosity might help make an architecture like AERA more intelligent, it is also worth asking how other systems deal with these issues. NARS doesn't seem to explicitly implement curiosity, but we have all seen that it will keep making new inferences even in the absence of input; there is no thumb twiddling. What mechanism allows this behavior in NARS that is currently missing from AERA? And what about OpenCog?