Envisioning a Robust, Scalable Metacognitive Architecture Built on Dimensionality Reduction

Jason B. Alonso and **Kenneth C. Arnold** and **Catherine Havasi**
MIT Media Laboratory
20 Ames St.
Cambridge, MA 02139
{jalonso,kcarnold,havasi}@media.mit.edu

Abstract

One major challenge of implementing a metacognitive architecture lies in its scalability and flexibility. We postulate that the difference between a reasoner and a metareasoner need not extend beyond what inputs they take, and we envision a network made of many instances of a few types of simple but powerful reasoning units to serve both roles. In this paper, we present a vision and motivation for such a framework with reusable, robust, and scalable components.

This framework, called Scruffy Metacognition, is built on a symbolic representation that lends itself to processing using dimensionality reduction and principal component analysis. We discuss the components of such as system and how they work together for metacognitive reasoning. Additionally, we discuss evaluative tasks for our system focusing on social agent role-playing and object classification.

Introduction

One major challenge of implementing a metacognitive architecture lies in its scalability: most metareasoners seem to be hand-crafted to serve a specific role in a specific implementation or solve a specific problem. Without a low-cost means to deploy arbitrary architectures, the consideration of complex architectures, like Marvin Minsky's Model-6 (Minsky 2006), will be prohibitively expensive.

In this paper, we postulate that the difference between a reasoner and a metareasoner need not extend beyond what inputs they take, and we envision a network made of many instances of a few types of simple but powerful reasoning units to serve both roles. To let these units be reusable in our model, they communicate using very simple symbols, whose specific semantics are generally opaque. Additionally, by choosing iterative machine learning algorithms, these reasoning units act in real-time, allowing the system to adapt to changing circumstances.

If an intelligent metareasoning system is to be built from largely homogeneous components, what function should each component perform? The connectionist answer—switches, or neurons—may be theoretically satisfying to some, but it is practically less than enlightening. We consider instead that one basic process of an intelligent system is to identify useful patterns in its input and its output; a symbol in such a system can represent the presence of one such pattern. Since the system summarizes a large number of inputs and outputs using a smaller number of symbols, we notice that this operation is, in essence, *dimensionality reduction*. We claim that a simple but effective form of planning can be accomplished by treating planning as a pattern completion problem that leverages dimensionality reduction. We also claim that metacognitive functions can be built on these principles.

Vantage Point in AI

We were led to this way of thinking by our experiments with dimensionality reduction on natural-language commonsense knowledge. Our work with common sense computing began in 1999 with the Open Mind Common Sense project (Singh et al. 2002). From there, we have developed a representation of common sense knowledge in the form of a semantic network we call ConceptNet (Havasi, Speer, and Alonso 2007), which we have found to have compelling latent semantics in a dimensionality-reduced form called AnalogySpace (Speer, Havasi, and Lieberman 2008). We have found that AnalogySpace and its underlying methods allow common sense to combine effectively with many textual data sets to reveal connections that only common sense can make (Havasi et al. 2009). These results have demonstrated to us the utility of thinking of a variety of processes as dimensionality reduction, which then informed our approach in this work.

Scruffy Metacognition

Scruffy Metacognition is our ongoing effort to bring narrative understanding, planning, and metacognition into a space that can benefit directly from our common sense corpora and techniques. Our approach to planning might be best described as "narrative completion," wherein an incomplete symbolic narrative is made whole using ideas borrowed from AnalogySpace, a process notably similar to those used in recommender systems. This imprecise and loosely statistical handling of symbolic representations distinguishes our approach from contemporary planning systems much the way that "scruffies" are distinct in the classic "neats vs. scruffies" philosophical dichotomy in artificial intelligence. Thus, we call such a narrative completion unit a "Scruffy Planner." As we will motivate, metacognition can be accomplished by a network of planners operating on the sequences of observations and actions in other planners. Scruffy Planners, other

processing units, and the network of interconnections between them form the building blocks of the Scruffy Metacognition framework.

Dimensionality Reduction

The concept of dimensionality reduction originates in linear algebra: a set of samples with a large number of observable values is approximated by a set of samples with a smaller number of components that can be mapped back to the original set. We employ the concept more broadly: in our usage, dimensionality reduction is the construction of compact representations from redundant unstructured inputs. A useful representation makes important facts explicit while suppressing irrelevant detail (Winston 1979). By seeking to construct a small set of "symbols" that summarize a large amount of observations, a dimensionality reduction process must necessarily find dimensions that are useful in some sense while truncating dimensions that are less relevant.

We have found that AnalogySpace, a dimensionality-reduced form of ConceptNet, gives useful *a posteriori* representations of commonsense concepts (Speer, Havasi, and Lieberman 2008). It enables discovering patterns in and drawing conclusions from semantic data. But observations about the world also tend to be redundant. In one robotics example, the observables "the object-in-hand is soft" and "the object-in-hand is fluffy" are likely very related. Such relationships can also be discovered by dimensionality reduction, and may suggest underlying semantics of the world, as we have seen in AnalogySpace. To process streams of data, such as continuous observations, various incremental analysis techniques have been developed. We have had most success recently with a variant on candid covariance-free incremental principal component analysis (CCIPCA) (Weng, Zhang, and Hwang 2003). It updates the dimensionality-reduced model as new data arrives, either because the model has not yet seen a complete set of examples, or because the state of the world that the data describes is itself changing. In the Scruffy Metacognition framework, we call a streaming dimensionality-reduction element a Reducer, represented in this paper by the symbol in Figure 1.

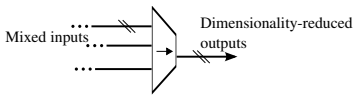

Figure 1: A basic reducer element

Dimensionality Reduction as Planning

In our model of planning, we represent sequences of actions and observations as timelines. These timelines are divided into discrete time windows that we call "timeframes," which can be host to any number of concurrent actions and observations. A timeline tracks the evolution of multiple actions and observations, which are both measured as real-valued weights on symbols in our implementations. We only loosely couple our notion of timeframes to real time measurements, which permits a dynamic variation of their relationship to real time, permitting a metacognitive architecture to "hurry" if necessary. We apply dimensionality reduction over multiple examples of these timelines to accomplish planning.

By using dimensionality-reduced representations of timelines, we hope to extract salient descriptions of the activities underlying each timeline, such that each timeline may be described as a combination of several fundamental activities, which we call "eigentimelines." This design choice serves a dual purpose: we wish to make such summarizations of timelines available for subsequent analysis in other components (metacognition), and we wish to interpolate missing timeline elements (planning) in a manner akin to some methods in recommender systems.

In our implementations, we expect the number of observations tracked by a timeline to be significantly greater than the number of timeframes in the timeline. This construction should lend itself well to dimensionality reduction. Also, though otherwise similar to the well-studied discipline of time series prediction, the shape of the problem (largely multivariate across a small number of time steps) appears to be uncommon for our understanding of time series prediction.

Actions are Observations

In Scruffy Metacognition, actions are treated the same way as observations, at least mathematically. In our ongoing implementations, this means that actions must be represented, like observations, as real-valued weights assigned to symbolic statements, and the inner semantics of those statements are generally opaque to the computation elements. By conflating observations with actions, dimensionality reduction methods can then interpolate actions as well as observations in an incomplete timeline, thus accomplishing planning.

The Scruffy Planner

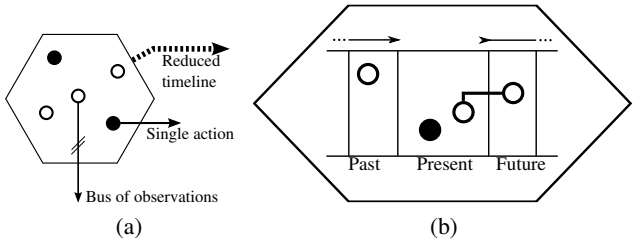

Figure 2: (a) A planner, in abbreviated form, showing observables (input-only, open dots), actions (output, filled dots), component lines, and bus lines. (b) Planner showing its internal timeline, with observables in its past and present, a goal in its future wired to an observable, and an action it can control in its present.

The basic operation of the planner, and the basis for Scruffy Planning altogether, is to take an incomplete timeline and provide its completion. An incomplete timeline may have its future unspecified, and thus the operation would be predictive in nature: the completed timeline will have a forecast of future observations. If an incomplete timeline has its present

unspecified, and its future is populated with a desirable goal state, then the operation will be planning: the completed timeline will have an estimation of necessary actions and observations in the present that would be expected to result in the future occurrence of goal state. An action is thus asserted through the belief that the action must happen in the present in order for the desired future to occur. Feedback on whether an action manages to occur can be considered an observation, and the timeline reflecting the actual consequences can be used as further training data for the planner.

This proposed planning component is the core unit for Scruffy Metacognition, which we illustrate it in Figure 2.

Scruffy Metacognition

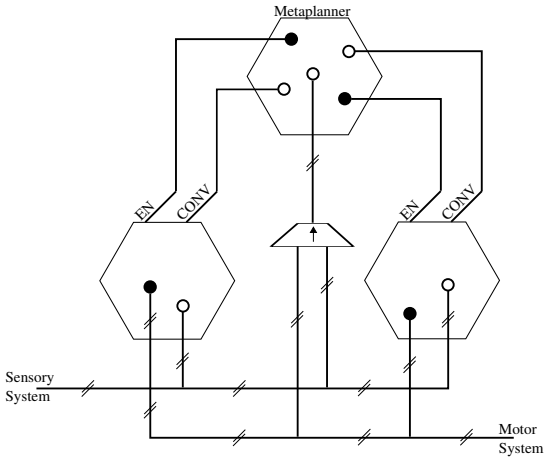

Figure 3: A basic metacognitive architecture with two planners managed by one metaplanner, which monitors their convergence (CONV) and can enable or disable them (EN)

Connecting Components

Though individually interesting, the planner and reducer on their own cannot realize a complex metacognitive architecture. However, components can be connected and chained to produce more complicated networks. The dimensionality-reduced output of one element can very easily be used as a collection of observations for another element.

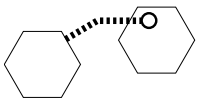

Figure 4: A planner provides its reduced timeline as a bus of observables to another planner—because this is a timeline bus, the receiving planner sees observables for every eigentimeline in each time frame

Given how singular value decomposition over ConceptNet reveals the latent semantics in AnalogySpace, we expect a computation element in Scruffy Metacognition to yield similarly compelling semantics, enabling other elements to reason comfortably over generalities. In Figure 4, we show a planner whose dimensionality reduction output is fed as a bus of observables to another planner. An alternative strategy would be to use a reducer across the inputs and outputs of one or more planners (from the present time frame only), and submit the result for analysis by another planner. This is the strategy used in the basic metacognitive architecture shown in Figure 3.

Repeated levels of dimensionality reduction is not new. In fact, the CCIPCA algorithm that drives the reducer was also used in a cascading network of dimensionality reducers in a computer vision application (Zhang, Weng, and Zhang 2002). As such, that work could largely be reproduced using a Scruffy Metacognition network made entirely of reducers.

Additionally, in situations where a substantial amount of bidirectional communication between planners would be helpful, an easy way to accomplish this is to allow them to see the same observations and each other's actions.

Planners and Goals

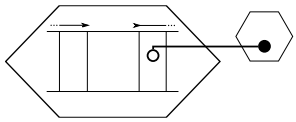

Figure 5: A planner with its goal state controlled as an action of another planner—also an example of an action in one planner controlling an observable in another

For a planner whose function is to give high-level instructions to one or more other planners, a simple and powerful control mechanism is to control the goal state of the subordinate planners by asserting some or all of the observations that populate the desired future in the subordinate planner's incomplete timeline.

One way for a planner to have feedback in a supervisory role over another planner is to take as an observation the actions that the subordinate planner asserts as actions. If the supervising planner has a goal that specifies a desired state for the feedback observable, then the supervisor will aim to induce behavior in the subordinate that causes the feedback observations to assume the desired state.

Evaluation through Examples

Once we have implemented a machine learning algorithm that satisfies our criteria for the Scruffy Planner, we have a handful of straightforward experiments we plan to attempt. First, we would like to model a few human-human interaction corpora that have been collected using crowd-sourcing methods, as this would allow us to look into developing artificial agents modeled on human behavior. Second, we would like to extend the cognitive architecture of our Mobile, Dexterous, Social (MDS) robot (Breazeal et al. 2008) to enable run-time learning of simple tasks.

The Restaurant Game and Mars Explorer

We plan to analyze the data collected through Jeff Orkin's *The Restaurant Game* (Orkin and Roy 2007) and Sonia Cher-

nova's *Mars Escape* (Chernova and Breazeal 2010), the latter system being built on the same framework and principles as the first. These both involve a first-person online game played by pairs of people matched arbitrarily by a hosting server—the players, working in different roles, then work together to accomplish a task, while the system records a replayable log of their activities and changes in the environment. In the case of *The Restaurant Game*, the participants role-play the common narrative of a customer visiting a restaurant and being served by wait staff. In the case of *Mars Escape*, the participants role-play characters of markedly different capabilities (one is a human while the other is a robot) work together to gather scientific specimens from a lab on Mars before time runs out.

Each of these games aim to capture some aspect of either social common sense (as in *The Restaurant Game*) or collaborative activity (as in *Mars Escape*). Each game role's typical activity sequences can be extracted from the game's logs (Orkin and Roy 2007). We hope to use these sequences to train Scruffy Planners to drive artificial agent behaviors that mimic humans playing these role. In *The Restaurant Game*, we expect that multiple planners will be helpful or required in order to master the various stages of the restaurant experience, while metacognitive facilities will be required to manage the transitions between them and to detect and manage failures or exceptions for atypical behavior.

The Sheep Games

It is our belief that a Scruffy Planner could be used in an MDS robot to learn how to play simple games that use many of the basic functions of the robot (voice recognition, object identification and tracking, and rudimentary object manipulation). The successful integration of a metacognitive architecture could be evaluated by its ability to enable the robot to learn to play two different games.

A simple demonstration of the major components of the MDS robot, which has been accomplished, involves two voice recognition commands, object tracking, object grasping and dropping, a bucket, and a stuffed sheep. A human holds out the stuffed sheep and says, "Nexi, come take the sheep." The robot responds by rolling over to the human, reaching up, and taking the sheep from the person's hand. The human then says, "Nexi, put it in the bucket," whereupon the robot rolls over to a bucket, positions the sheep over the bucket, and drops it. If enhanced with Scruffy Planning, the MDS robot should be able to learn to execute many steps without instruction.

If we introduced a second game that involved, say, a pen and a pasture, we could create a game where the MDS robot must sort various animals to their appropriate locations. Learning to play this game uses the capabilities of a Scruffy Planner, and is made easier with metacognitive elements as in Figure 3.

Contributions and Future Steps

In a large sense, the "scruffiness" that we believe is key to our approach is that we do not design compact representations for reasoning *a priori*—instead, we turn to dimensionality reduction to derive compact representations that expose salient semantics. It is our hypothesis this low-effort choice of representation enables the use of the same mechanism for reasoning at multiple levels of perception, cognition, and metacognition.

Perhaps a distinguishing feature of our approach is that we envision the larger framework before focusing on its components. We claim that we are therefore envisioning a framework for metacognition that is scalable, by virtue of using reasoning components that do not need to be deeply tailored to their precise uses and of using a vector representation that, though unorthodox, encapsulates semantics usable to the reasoning components with little or no engineering intervention.

References

Breazeal, C.; Siegel, M.; Berlin, M.; Gray, J.; Grupen, R.; Deegan, P.; Weber, J.; Narendran, K.; and McBean, J. 2008. Mobile, dexterous, social robots for mobile manipulation and human-robot interaction. In *SIGGRAPH '08: ACM SIGGRAPH 2008 new tech demos*. New York, NY, USA: ACM.

Chernova, S., and Breazeal, C. 2010. Learning Temporal Plans from Observation of Human Collaborative Behavior. In *AAAI Spring Symposia, It's All In the Timing: Representing and Reasoning About Time in Interactive Behavior*.

Havasi, C.; Speer, R.; Pustejovsky, J.; and Lieberman, H. 2009. Digital Intuition: Applying Common Sense Using Dimensionality Reduction. *IEEE Intelligent Systems*.

Havasi, C.; Speer, R.; and Alonso, J. 2007. ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge. In *Recent Advances in Natural Language Processing*.

Minsky, M. 2006. *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon & Schuster.

Orkin, J., and Roy, D. 2007. The Restaurant Game: Learning Social Behavior and Language from Thousands of Players Online. *Journal of Game Development* 3(1):39–60.

Singh, P.; Lin, T.; Mueller, E. T.; Lim, G.; Perkins, T.; and Zhu, W. L. 2002. Open Mind Common Sense: Knowledge Acquisition from the General Public. In *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, 1223–1237. London, UK: Springer-Verlag.

Speer, R.; Havasi, C.; and Lieberman, H. 2008. AnalogySpace: Reducing the Dimensionality of Common Sense Knowledge. *Proceedings of AAAI 2008*.

Weng, J.; Zhang, Y.; and Hwang, W.-S. 2003. Candid covariance-free incremental principal component analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25(8):1034–1040.

Winston, P. H. 1979. Learning and reasoning by analogy: The details. Technical Report AIM-520, Massachusetts Institute of Technology.

Zhang, N.; Weng, J.; and Zhang, Z. 2002. A developing sensory mapping for robots. *International Conference on Development and Learning* 13.