Graphical Models for Integrated Intelligent Robot Architectures

Paul S. Rosenbloom

Department of Computer Science & Institute for Creative Technologies
University of Southern California
12015 Waterfront Drive, Playa Vista, CA 90094
rosenbloom@usc.edu

Abstract

The theoretically elegant yet broadly functional capability of graphical models shows intriguing potential to span in a uniform manner perception, cognition and action; and thus to ultimately yield simpler yet more powerful integrated architectures for intelligent robots and other comparable systems. This position paper explores this potential, with initial support from an effort underway to develop a *graphical architecture* that is based on factor graphs (with piecewise continuous functions).

Robots require a close coupling of (multiple forms of) perception and action. Intelligent robots go beyond this to require a further coupling with cognition. From the perspective of robotics – with its focus on behavior in the world – the construction of intelligent robots generally emphasizes a tightly integrated perceptuomotor system that is then loosely connected to some limited form of cognitive system (such as a planner); as for example in (Bonasso et al. 1997). From the perspective of cognitive architectures – with their focus on integrated embodiments of hypotheses concerning the fixed structure underlying intelligent behavior – the construction of intelligent robots generally emphasizes a highly functional cognitive system that is then loosely connected to limited perceptual and motor modules; as for example in (Laird and Rosenbloom 1990). Neither perspective typically strives for a deep integration across the signal-to-symbol divide that separates the perceptuomotor and cognitive systems, nor even to do full justice to what is on the other side.

Other approaches are possible though. One such is a form of *graphical architecture* that leverages the broadly functional yet theoretically elegant construct of graphical models (Koller and Friedman 2009) to support, among other things, a uniform approach to signal and symbol

processing. At their core, graphical models concern efficient computation over complex multivariate functions by decomposing them into the product of simpler subfunctions. Such models have over the years produced state-of-the-art capabilities across many aspects of perception and robotics under a variety of names, such as factor graphs, Markov and conditional random fields, Bayesian and Markov networks, hidden Markov models, Kalman filters, and the Viterbi algorithm. Less commonly recognized is that they can also yield state-of-the-art symbol processing in areas such as constraint satisfaction (Dechter 2003) and rule match (Rosenbloom 2011a); and are even compatible with full first-order logic (Domingos and Lowd 2009). The key question this raises though is whether all of the requisite capabilities can be combined into a unified architecture that is based on a single form of graphical model capable of exhibiting a superset of the requisite functionality.

The position staked out here is that graphical models do provide a potential basis for the development of integrated intelligent robot architectures that uniformly span the necessary capabilities across perception, cognition and action. Over the past few years I have been exploring a particular graphical architecture that leverages factor graphs (Kschischang, Frey and Loeliger 2001) for simpler yet more general combinations of capabilities resembling those found in existing cognitive architectures, while also incorporating additional capabilities that are beyond them (Rosenbloom 2011a, 2001b). Although factor graphs were originally developed in coding theory, they are the most general form of graphical model so far developed, and show promise for broad applicability across AI and robotics. Many of the mechanisms in existing cognitive architectures, while not normally cast as factor graphs, are in fact amenable to reconceptualization in this form; as are many forms of processing central to robotics.

This graphical architecture is still in a relatively early stage of development, but it has already been shown to

produce forms of many of the major capabilities found in state-of-the-art cognitive architectures, such as Soar (Laird 2012) and ACT-R (Anderson 2009). This has included both procedural (rule) and declarative (semantic and episodic) memories, plus a constraint memory that combines procedural and declarative aspects (Rosenbloom 2010). Beginnings of mental imagery have also been demonstrated (Rosenbloom 2011c), along with preference-based decision making and the ability to reflect upon impasses in decision making (Rosenbloom 2011d). Simple forms of statistical language processing have also been produced – including word sense disambiguation and question answering – as have the beginnings of learning.

Even more interesting from the perspective of intelligent robotics is that forms of perception (based on a CRF) localization (based on the relevant portion of SLAM), and probabilistic decision making (based on POMDPs) have also been demonstrated; and, in fact, integrated together into a single factor graph that performs a simple form of navigation in a virtual corridor (Chen et al. 2011). Although it still has far to go, this work represents an important step in developing the perception and robotics capabilities that will be needed to ground the architecture in the world.

To support the initial plausibility of this paper's position, the next two sections will explain in a bit more detail how the existing graphical architecture works, along with the structure of the combined factor graph built for perception, localization and decision making in virtual corridor navigation. The last section then wraps up with some final discussion.

The Graphical Architecture

The graphical architecture is based on *piecewise continuous functions*, to provide a broad-spectrum representational primitive, plus *factor graphs* that are defined over these primitives to provide more complex representations and the processing needed over them. Together these two techniques enable signals and symbols, along with uncertainty about them, to be represented and processed in a uniform manner.

Piecewise continuous functions partition a multidimensional space into regions, and then specify a continuous function – which is limited to linear in the current architecture – over each region (Figure 1). Such a representation can represent arbitrary continuous functions as closely as desired, albeit at the cost of more regions. It can also represent discrete functions by introducing region boundaries at unit intervals and limiting the functions on these regions to constant values; and represent symbols by further limiting the constant values to 0/1 (for *false*/*true*) and adding a symbol table.
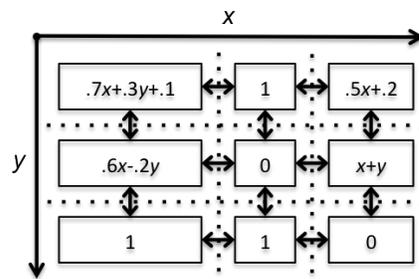
Figure 1: 2D piecewise continuous function as an array of linear regions.

Factor graphs are a flavor of undirected graphical model that replaces the potential functions from Markov networks (aka Markov random fields) with factor nodes in the graph itself. Figure 2, for example, shows a simple factor graph for a polynomial function, with three variable nodes and two factor nodes.
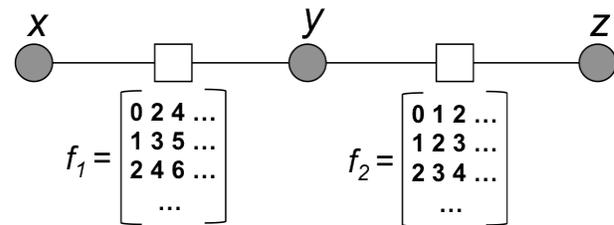
Figure 2: Factor graph for $f(x,y,z) = y^2+yz+2yx+2xz = (2x+y)(y+z) = f_i(x,y)f_2(y,z)$

Factor graphs can be solved via a range of approaches – including message passing, sampling, particle filters, and variational methods – to yield either marginals on variables or a MAP estimate. Some of the methods are exact, at least for some graphs, while others are approximate. The graphical architecture is based on a variant of the summary product algorithm, a message passing approach in which a message in either direction along a link expresses a function over the variable(s) on the link. At a variable node, messages are combined via pointwise product, a computation that is similar to an inner product except that there is no final summation and the input and output functions thus have the same rank. At a factor node, a similar product is computed, but the node's function is also included in the product, and then all variables not to be included in the outgoing message are summarized out. Summarization may occur via integration – since these are continuous functions – to yield marginals, or via maximization to yield MAP. Figure 3, for example, shows how the algorithm generates a marginal for the variable y if evidence is provided for variables x (=5) and z (=3).

In the architecture, factor graphs are specified in a high-level language of *conditionals*, with each conditional consisting of some combination of *conditions*, *actions*, *condacts*, and a *function*. Conditions and actions are much like the same structures in rule-based systems; conditions match to patterns of elements in working memory and pass matches forward, while actions take information about
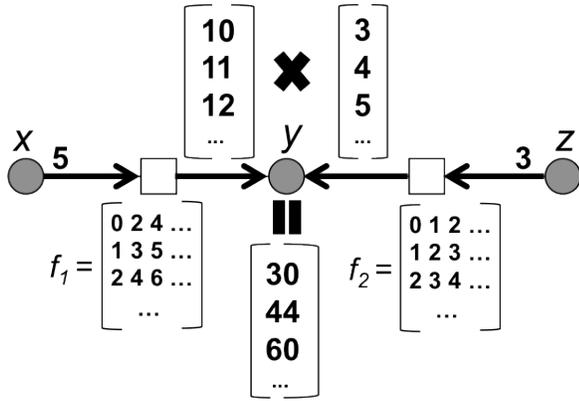
Figure 3: Computation via the summary product algorithm of the marginal on y from evidence on x and z.

matches and convert them into changes in working memory. Figure 4, for example, shows a heuristic rule from the Eight Puzzle. Condacts, a neologism for *cond*itions and *act*ions, meld these two functionalities. When combined with Boolean functions over their variables, condacts enable the bidirectional processing that is key to constraint satisfaction. With numerical functions, this same bidirectional activity is central to signal processing and probabilistic reasoning. Figure 5, for example, shows a conditional that defines the conditional probability of an object's weight given its category.

```
CONDITIONAL GoalReject
 Conditions: (Operator id:o state:s x:x y:y)
             (Goal state:s x:x y:y tile:t)
             (Board state:s x:x y:y tile:t)
 Actions: (Selected - state:s operator:o)
```

Figure 4: Eight Puzzle heuristic that rejects from consideration operators that move tiles out of place.

```
CONDITIONAL ConceptWeight
 Conditions: (Object state:s object:o)
 Condacts: (Concept object:o concept:c)
           (Weight object:o weight:w)
```

$w \backslash c$	Walker	Table	…
[1,10>	.01w	.001w	…
[10,20>	.2–.01w	"	…
[20,50>	0	.025– .00025w	…
[50,100>	…	…	…

Figure 5: Specification of the conditional probability of an object's weight given its category.

The architecture's central processing cycle consists of starting with a pool of evidence stored in working-memory factor nodes, passing messages until quiescence, and then deciding what changes to make in the working-memory factor nodes. Each such cycle can include parallel waves of rule firings, access to declarative knowledge, perception,

and simple forms of reasoning.

Virtual Corridor Navigation

The virtual corridor navigation task concerns moving from an initial location to a goal location in a one dimensional virtual corridor, given: a utility function on locations; noisy perception that is limited to detecting rectangles, circles and colors in the current location; and a fallible ability to move to adjacent locations (Figure 6). The walls at the end of the corridor appear as rectangles, while the three doors appear as colored rectangles with circles (doorknobs).
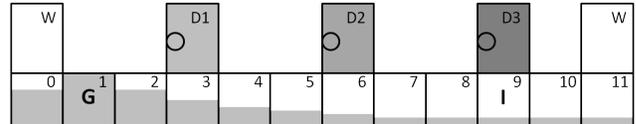
Figure 6: Task environment with two walls, three doors, an initial location (I) and a goal location (G). The relative values of the utility function can be seen in the square shading.

Processing during each decision starts with perception via a conditional random field (CRF). Training was performed outside of the architecture, since the appropriate architectural learning mechanisms have not yet been developed, but the results were then added as conditionals to the architecture's knowledge base, yielding a graph like that in Figure 7 for three time steps. For each such time step, sensations (S) from three (virtual) sensors arrive at the bottom and are passed through perceptual factor nodes (P) to yield distributions over the object seen (O). Object-transition factor nodes (OT) further refine these distributions by imposing probabilistic constraints on what can be seen on adjacent time steps.
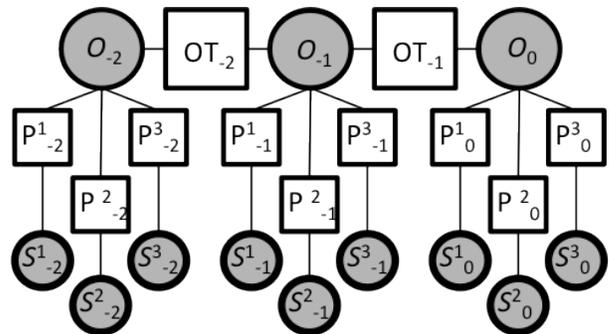
Figure 7: Factor graph for conditional random field (CRF).

The object distributions (O) feed into the localization network (Figure 8), whose task is to determine distributions over the locations (X), via map factor nodes (M) that relate objects to locations, a prior distribution on the initial location (Pr), and location-transition factor nodes (XT) that probabilistically constrain the locations for adjacent time steps. The latter also takes into consideration
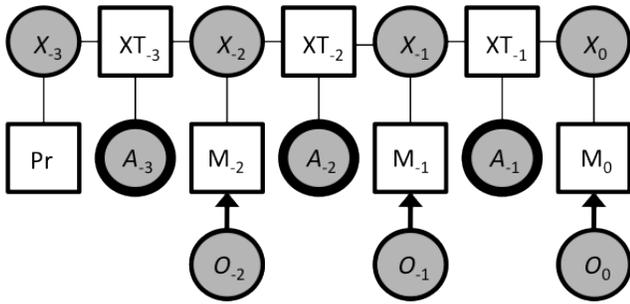
Figure 8: Factor graph for localization via part of simultaneous localization and mapping (SLAM).

evidence concerning what action has been performed at each time step (A). The distribution for the current location (X_0) then feeds into a partially observable Markov decision problem (POMDP), as shown in Figure 9, which projects forward over possible future actions (A) and locations (X), while consulting the utility factor functions (U) for the values of the locations. By the time quiescence is reached, a distribution is generated over the current action (A_0) from which the best choice can then be selected.
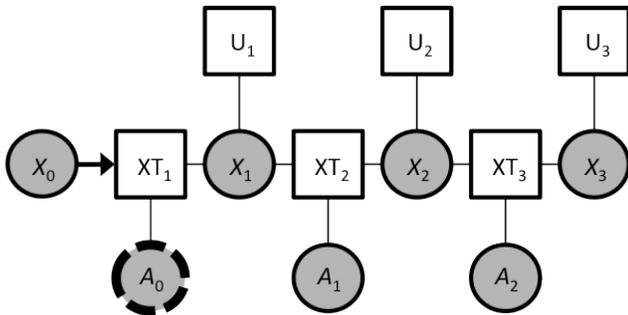
Figure 9: Factor graph for the partially observable Markov decision problem (POMDP).

Although three distinct factor graphs have been shown for this task in Figures 7-9, they are in fact all integrated into a single graph within the architecture, and all processed together within a single central cycle. They are shown separately here for simplicity, and because of the limitations of these narrow article columns.

Final Discussion

Both the graphical architecture and the virtual navigation task obviously have many limitations at present. The architecture is still, for example, missing part or all of several important capabilities, such as learning and motor control. Work has only begun on learning, but an implementation of episodic learning is in hand, and promising approaches are in sight for other varieties. While nothing has yet been done on motor control, others

have investigated both optimal control and motor control via graphical models (Kappen, Gomez and Opper 2009; Toussaint and Goerick 2010), so there is at least a plausibility case for this ultimately fitting into the architecture as well.

The task environment in the previous section was quite limited as well, being for example discrete and spanning only one dimension. However, work on mental imagery in the architecture has demonstrated facility with continuous 2D environments, including

Figure 10. The Eight Puzzle Board.

for example a continuous representation of the 2D Eight Puzzle board (Figure 10). As shown in Figure 11, this board can be represented via nine continuous 2D planes, one for each tile and the empty cell. The region of each plane within which the corresponding tile sits is assigned a value of 1, while the rest of the plane is set to 0. The figure shows the location of the blank (right middle) in the first (0th) plane and tile 1 (top right) in the next (1st) plane.
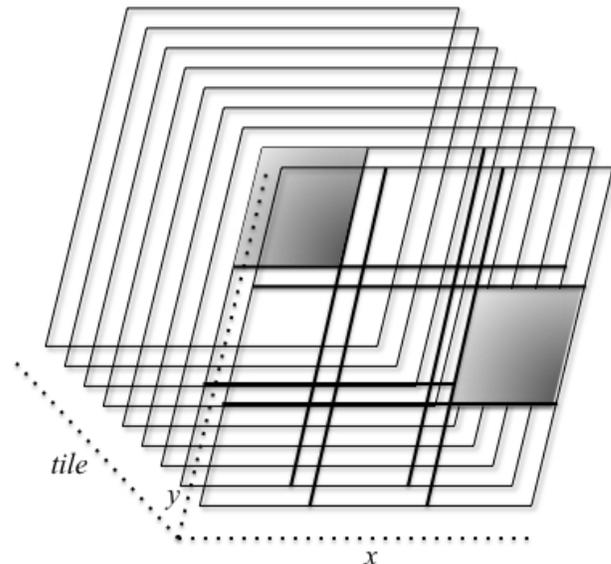
Figure 11. Partial visualization of the hybrid (discrete tiles and continuous x and y dimensions) representation of the Eight Puzzle board.

There are of course many other limitations in the work to date, including the fact that this is all virtual, rather than occurring within a physical robot interacting with the real world. But the main point of presenting this example task, and for the description of the graphical architecture, has been to help establish the initial plausibility of the position

that graphical models offer an intriguing basis for the development of integrated architectures for intelligent robots. Hopefully it has been successful at this.

The position advocated here is closest in spirit to a recent effort to leverage graphical models in integrating together "motor control, planning, grasping and high-level reasoning" to yield a "coherent, control, trajectory optimization, and action planning architecture" (Toussaint, Plath, Lang and Jetchev 2010). Toussaint's work, which comes out of robotics, appears to have independently reached the same conclusion that has been arrived at here via a path from cognitive architectures.

The idea of a graphical architecture also shares features with general languages that are being developed for statistical relational AI, such as Alchemy (Domingos and Lowd 2009), Blog (Milch et al. 2007) and Church (Goodman et al. 2008), but the former focuses on full, integrated architectures rather than on just languages or toolkits. As such, it provides a potential path towards a reintegration of AI for intelligent robots, virtual humans, and other forms of artificially intelligent systems that must perceive and behave in complex and uncertain environments.

Acknowledgements

References

Anderson, J. R. 2009. *How Can the Human Mind Occur in the Physical Universe?* Oxford: Oxford University Press.

Bonasso, R. P., Firby, R. J., Gat, E., Kortenkamp, D., Miller, D. P., and Slack, M. G. 1997. Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental & Theoretical Artificial Intelligence* 9: 237-256.

Chen, J., Demski, A., Han, T., Morency, L-P., Pynadath, P., Rafidi, N., and Rosenbloom, P. S. 2011. Fusing symbolic and decision-theoretic problem solving + perception in a graphical cognitive architecture. *Proceedings of the 2nd International Conference on Biologically Inspired Cognitive Architectures*.

Dechter, R. 2003. *Constraint Processing*. San Francisco, CA: Morgan Kaufmann.

Domingos, P. and Lowd, D. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool.

Goodman, N. D., Mansinghka, V. K., Roy, D., Bonawitz, K., and Tenenbaum, J. B. 2008. Church: a language for generative models. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*.

Kappen, B., Gomez, V., and Opper, M. 2009. Optimal control as a graphical model inference problem. *Computing Research Repository*.

Koller, D. and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: MIT Press.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47: 498-519.

Laird, J. E. (2012). *The Soar Cognitive Architecture*. Cambridge, MA: MIT Press. In press.

Laird, J. E. and Rosenbloom, P. S. 1990. Integrating execution, planning, and learning in Soar for external environments. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, 1022-1029. Cambridge, MA: MIT Press.

Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D. L., and Kolobov, A. 2007. BLOG: Probabilistic models with unknown objects. In L. Getoor and B. Taskar, (Eds.) *Introduction to Statistical Relational Learning*. Cambridge, MA: MIT Press.

Rosenbloom, P. S. 2010. Combining procedural and declarative knowledge in a graphical architecture. In *Proceedings of the 10th International Conference on Cognitive Modeling*.

Rosenbloom, P. S. 2011a. Rethinking cognitive architecture via graphical models. *Cognitive Systems Research* 12: 198-209.

Rosenbloom, P. S. 2011b. Bridging dichotomies in cognitive architectures for virtual humans. In *Proceedings of the AAAI Fall Symposium on Advances in Cognitive Systems*. In press.

Rosenbloom, P. S. 2011c. Mental imagery in a graphical cognitive architecture. In *Proceedings of the 2nd International Conference on Biologically Inspired Cognitive Architectures*. In press.

Rosenbloom, P. S. 2011d. From memory to problem solving: Mechanism reuse in a graphical cognitive architecture. In *Proceedings of the 4th Conference on Artificial General Intelligence*. In press.

Toussaint, M. and Goerick, C. 2010. A Bayesian view on motor control and planning. In O. Sigaud and J. Peters (Eds.) *From motor to interaction learning in robots*. Studies in Computational Intelligence, Vol. 264, Springer.

Toussaint, M., Plath, N., Lang, T., and Jetchev, N. 2010. Integrated motor control, planning, grasping and high-level reasoning in a blocks world using probabilistic inference. In *Proceedings of the IEEE International Conference on Robotics and Automation* (ICRA 2010), 399-405.