Finding Good Policies for Large Domains

Kurt Steinkraus & Leslie Pack Kaelbling

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

http://www.ai.mit.edu

The Problem: Suppose an agent has a probabilistic model for how the world works, it knows that it has certain actions that have certain effects, and it has certain goals. How should the agent act so as to best achieve its goals?

For example, consider an office helper robot. Perhaps the robot knows how to make and deliver coffee, collect printouts, take out the trash, distribute packages, etc. If the robot is in the middle of delivering packages and Bob asks it for some coffee, how should the robot respond? Should it finish delivering its packages first, or will that make Bob cranky when he doesn't get his early-morning coffee fix right away? Should the robot pick up some coffee right now and drop it off at Bob's office when it delivers packages near there, or will the coffee be too cold by then? Should the robot leave the packages in the kitchen while it makes and delivers coffee to Bob, or is the risk that they will get stolen too high?

Motivation: There are many different ways of precisely describing the world and how it behaves, to allow a computer to calculate the probable course of events or perhaps a good course of action to take. These domain modelling techniques include Markov Decision Processes (MDPs), Bayes nets, and influence diagrams. Using such representations is desirable because they model the world probabilistically, since the world is an uncertain place after all, and because they can incorporate such things as partial observability, actions, and utility functions.

Dynamic Bayes nets with actions and utilities provide an appealing, intuitive way for a domain expert to model an environment for, say, a mobile robot. The big downside to using these models is that, so far, all algorithms developed to deal with these models have turned out to be intractable on any examples of significant size. Bayes nets (and dynamic Bayes nets, influence diagrams, etc.) are a step up from MDPs because the structure and the compactness of representation generally allow larger problems to be reasoned through. Even so, inference on Bayes nets with a hundred nodes, or optimal policy generation with a few tens of nodes, is generally not doable.

Previous Work: One associated problem is how to do inference and belief propagation in dynamic Bayes nets. Several approximate extensions of the standard exact Bayesian inference techniques (clique tree propagation and variable elimination) have been proposed; see Minka's thesis [2] for an overview. A survey of techniques to extract optimal policies from PoMDPs is given by Murphy [3]. Boutilier [1] and others have experimented with different ways of applying PoMDP techniques to factored representations of PoMDPs.

Approach: The first order of business is to come up with a sample domain that is both large and realistic. It needs to be large in order to simulate correctly the kinds of issues arising in a task with multiple possibilities for reward and many interacting parts. The sample domain also needs to be realistic, however, to ensure that it contains the same sort of regularities that a real-world task would. We have created a domain that models the world of a robotic office assistant. It is a large partially observable dynamic Bayes net that has approximately 2^{50} states, 20 actions, and 20 separate aspects to the utility function. Currently, no method exists for tractably finding a good policy in such a domain, let alone an optimal one.

Our approach to using such large dynamic Bayes nets is to take advantage of the structure not only the connections but also the structure hidden in the conditional probability tables for each node. This structure has previously been used to speed up inference [4]. We hope to be able to do the same sort of thing, not for inference and expectation monitoring, but for policy generation in dynamic Bayes nets with decision and utility nodes. By paying careful attention to the structure in the conditional probability tables, we can set a particular goal and pay attention just to the part of the dynamic Bayes net that is likely to impact this current goal.

Impact: Given a large domain modelled in the language of dynamic Bayes nets with actions and utilities, our research will allow it to be tractably queried for useful information. For instance, we will be able to be able to

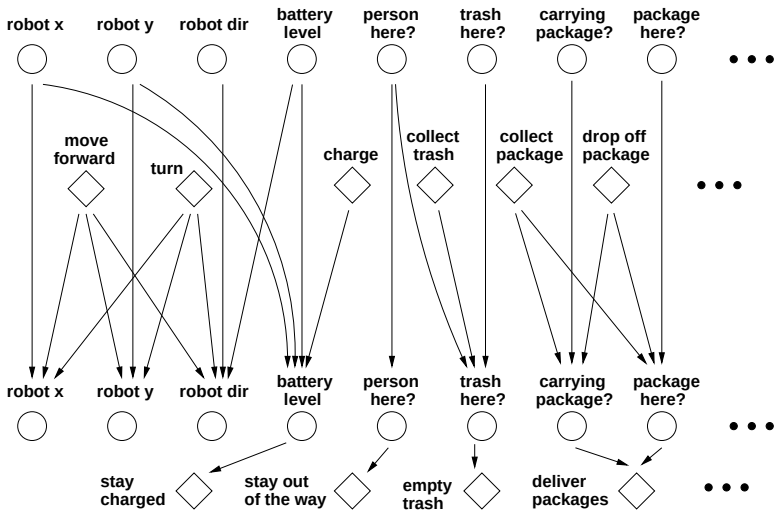

Figure 1: A domain for an office robot

create a policy that successfully attains some goal involving a small part of the Bayes net. This involves figuring out which parts of the domain are irrelevant, how they are irrelevant, and ignoring them, something that current policy generation techniques cannot do. We also hope to be able to generate a policy for the entire space, and have the policy be good but not necessarily optimal (since optimality is intractable).

Research Support: This research is supported in part by NASA, under award # NCC2-1237.

References:

[1] C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy construction. In *Extending Theories of Action: Formal Theory & Practical Applications: Papers from the 1995 AAAI Spring Symposium*, pages 33–38. AAAI Press, Menlo Park, California, 1995.

[2] Thomas P. Minka. *A family of algorithms for approximate Bayesian inference.* PhD thesis, Massachusetts Institute of Technology, 2001.

[3] Kevin P. Murphy. A survey of pomdp solution techniques.

[4] David Poole. Probabilistic partial evaluation: Exploiting rule structure in probabilistic inference. In *IJCAI*, pages 1284–1291, 1997.