Hierarchical Reinforcement Learning of Dialogue Policies in a development environment for dialogue systems: REALL-DUDE

Oliver Lemon and Xingkun Liu
School of Informatics
Edinburgh University
olemon,xliu4@inf.ed.ac.uk

Daniel Shapiro and Carl Tollander
CSLI/Applied Reactivity
Stanford University
dgs,carl@appliedreactivity.com

Abstract

We demonstrate the REALL-DUDE system[1], which is a combination of REALL, an environment for Hierarchical Reinforcement Learning, and DUDE, a development environment for "Information State Update" dialogue systems (Lemon and Liu, 2006) which allows non-expert developers to produce complete spoken dialogue systems based only on a Business Process Model (BPM) and SQL database describing their application (e.g. banking, cinema booking, shopping, restaurant information, ...). The combined system allows rapid development and automatic optimization of spoken dialogue systems. Hierarchical Reinforcement Learning (RL) has not been applied to the problem of dialogue management before. It provides a way of dramatically reducing the size of the state space to be considered in RL problems. REALL-DUDE thus allows iterative development of dialogue policies through Hierarchical RL to be combined with a development environment for complete dialogue systems, encompassing parsing, speech recognition, synthesis, and dialogue management.

1 Introduction

It has been shown in previous work (Singh et al., 2002) that dialogue policies obtained by Reinforcement Learning (RL) can improve over hand-coded dialogue managers. However, a key problem in RL applied to dialogue management is the very large policy spaces generated by the dialogue management problem. REALL 's key source of power is its ability to constrain learning with background knowledge, within a principled framework. It has been shown (Shapiro and Langley, 2002) that this approach generates three order of magnitude reductions in problem size, and two order of magnitude improvements in learning rate, relative to the common formulation of RL tasks which offers all feasible options in all possible situations.

We demonstrate a development environment for dialogue systems which allows iterative development and refinement of dialogue policies through Hierarchical RL. We present the concepts behind REALL and DUDE , and show how to use DUDE to generate complete spoken dialogue systems (Lemon and Liu, 2006). We then demonstrate learning experiments that explore dialogue policies in the presence of different reward signals and channel noise characteristics, and show how the learner acquires different optimized policies.

2 REALL – Reactive Planning and Hierarchial RL

REALL is a language for defining extremely reactive agent behavior. It consists of a representation for expressing hierarchical, goal-oriented plans, together with an interpreter for evaluating those plans that operates in a repetitive loop. This iteration supplies reactivity: even if the world changes radically between two execution cycles, REALL will find a goal-relevant action to employ.

REALL is also a learning system. Because its interpreter contains a model-free reinforcement learning algorithm, every REALL agent has the ability to acquire an action policy from delayed reward. Programmers can access this capability

[1]This research is supported by Scottish Enterprise under the Edinburgh-Stanford Link programme.

brandial 2006: Proceedings of the 10th Workshop on the Semantics and Pragmatics of Dialogue, pages 185–186.

by writing plans with disjunctive elements, and by embedding those choice points in hierarchical plans. As a result, REALL offers a means of invoking learning in the context of background knowledge, and this constrains the learning task.

Because REALL is a learning system, it supports a novel development metaphor called programming by reward. Here, the programmer may encode a dialogue strategy with options, and specify reward functions that serve as the targets of optimization. Via a training period, the reward functions select one of the many policies implicitly contained in the REALL plan, and developers can obtain distinct behaviors by making small changes to the reward functions (Shapiro et al., 2001).

REALL learns a policy by finding the best action to take in every state. It learns the value of a given state-action pair by sampling its future trajectory, and it represents this value using a linear function of currently observable features. REALL bootstraps: it updates the estimate for a state-action pair using its current value, the current reward, and the estimate associated with the next state-action pair. Over time, these estimates converge to their appropriate values.

3 The DUDE development environment

The contribution of DUDE (Lemon and Liu, 2006) is to allow non-expert developers to build ISU dialogue systems using only the Business Process Models (BPMs) and databases that they are already familiar with, as shown in figure 1.

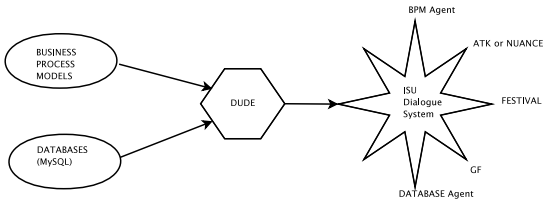

Figure 1: The DUDE development process

The environment includes a development GUI, automatic generation of Grammatical Framework (GF) grammars for robust interpretation of spontaneous speech, and uses application databases to generate lexical entries and grammar rules. The GF grammar is automatically compiled to an ATK or Nuance language model for speech recognition. See (Lemon and Liu, 2006) for details.

The power of REALL-DUDE is to embed Hierarchical Reinforcement policy learning and optimization from REALL within the rich development environment supplied by DUDE .

4 Demonstrating learning

We will present a REALL program, `Slotfiller`, embedded in the DUDE environment, which contains a scaffolding of required dialogue behavior (e.g., confirmations, clarifications, mixed-initiative questions). The demonstration presents a variety of learning experiments that explore these decisions in the presence of different reward signals and channel noise characteristics. We will show how the learner acquires and optimizes distinct dialogue policies in each case.

5 Conclusion

Hierarchical RL has not been applied to the problem of dialogue management before. It provides a principled way of dramatically reducing the size of the state space to be considered in RL of dialogue management. Here we demonstrate a development environment, REALL-DUDE , which combines RL for optimization of dialogue policies with a full development environment for automatic generation of spoken dialogue systems. We will demonstrate how to develop complete spoken dialogue systems using DUDE and then we will demonstrate strategy learning for those systems using REALL, which optimizes policies for different noise and reward conditions in dialogue.

References

Oliver Lemon and Xingkun Liu. 2006. DUDE: a Dialogue and Understanding Development Environment, mapping Business Process Models to Information State Update dialogue systems. In *Proceedings of EACL (demonstration systems)*.

D. Shapiro and P. Langley. 2002. Separating skills from preference: using learning to program by reward. In *Nineteenth International Conference on Machine Learning.*

Dan Shapiro, Pat Langley, and Ross Shachter. 2001. Using background knowledge to speed reinforcement learning. In *Fifth International Conference on Autonomous Agents*.

Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *Journal of Artificial Intelligence Research (JAIR)*.