Artificial Intelligence 241 (2016) 131–152

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Online belief tracking using regression for contingent planning

Ronen I. Brafman, Guy Shani *

Ben-Gurion University of the Negev, Be'er Sheva, Israel

ARTICLE INFO

Article history:
Received 22 March 2015
Received in revised form 27 July 2016
Accepted 19 August 2016
Available online 7 September 2016

Keywords:
Contingent planning
Partial observability
Non-deterministic planning
Regression
Belief

ABSTRACT

In online contingent planning under partial observability an agent decides at each time step on the next action to execute, given its initial knowledge of the world, the actions executed so far, and the observation made. Such agents require some representation of their belief state to determine which actions are valid, or whether the goal has been achieved. Efficient maintenance of a belief state is, given its potential exponential size, a key research challenge in this area. In this paper we develop the theory of regression as a useful tool for belief-state maintenance. We provide a formal description of regression, discussing various alternatives and optimization techniques, and analyze its space and time complexity. In particular, we show that, with some care, the regressed formula will contain variables relevant to the current query only, rather than all variables in the problem description. Consequently, under suitable assumptions, the complexity of regression queries is at most exponential in its *contextual width*. This parameter is always upper bounded by Bonet and Geffner's *width* parameter, introduced in their state-of-the-art *factored belief tracking (FBT)* method. In addition, we show how to obtain a poly-sized circuit representation for the online regression formula even with non-deterministic actions. We provide an empirical comparison of regression with FBT-based belief maintenance, showing the power of regression for online belief tracking. We also suggest caching techniques for regression, and demonstrate their value in reducing runtime in current benchmarks.

1. Introduction

Agents that act in the real world are often limited to partial knowledge about the world, obtained through their sensors. This is typically known as planning under partial observability. Agents that plan in a partially observable domain, typically maintain some representation of their state of knowledge online. A complete description of the agent's state of knowledge, consisting of the set of possible states of the world (or a probability distribution over possible states, in the probabilistic case), is called the agent's *belief state*. Many planners for partially observable domains search directly in the space of belief states, known as the *belief space*.

Maintaining and updating an explicit belief state can be expensive because the number of possible states of the world can be exponential in the description size of a single state, i.e. the number of state variables. Thus, directly maintaining sets of states becomes unmanageable both space and time-wise as the problem grows. To alleviate this, methods that maintain a more compact, symbolic description of the set of possible states have been developed, such as methods based on BDDs [3], prime-implicates, CNF, and DNF [27]. Unfortunately, symbolic representations also have an exponential worst-case

* Corresponding author.
E-mail addresses: brafman@cs.bgu.ac.il (R.I. Brafman), shanigu@bgu.ac.il (G. Shani).

http://dx.doi.org/10.1016/j.artint.2016.08.005

description, and when not, may be expensive to update. Furthermore, every representation that was suggested thus far, while being very compact for certain benchmark problems, demonstrated the worst-case performance on other benchmarks.

Still, planning algorithms can benefit from an important observation [15] – during planning and plan execution it is sufficient for the agent to answer only two types of queries with respect to a belief state: *has the goal been achieved?* and for each action, *is this action applicable?* That is, whether the action's precondition is satisfied in the belief state.

The regression-based method we study takes a lazy approach to belief maintenance, maintaining only the initial belief state, the set of actions executed, and sensed observations. This approach is similar in spirit to the Situation Calculus [16] where a state is represented in terms of the initial state and sequence of actions. Using this information, one could regress the conditions required currently (e.g., p) towards the initial belief state. If the regressed condition is implied by the initial state then we know that p holds now. Otherwise, there exists some current possible state that does not satisfy p.

In earlier work [8], we implemented this approach within an online contingent planner and showed that, empirically, the regression-based method, coupled with some caching, is highly efficient on current benchmark problems. In this paper we provide detailed description and analysis of regression-based belief-state maintenance, focusing on *online* belief maintenance – which is sometimes called *filtering* [2,25]. In online belief maintenance, the agent, after having performed a sequence of actions and observing some observations, must determine whether the goal or a precondition literal l holds. This is a somewhat simpler task than *offline* belief maintenance, where the agent must consider arbitrary hypothetical sequences, and for each such sequence, not only determine the resulting belief state, but also, determine whether this sequence is possible.

There is a long line of research on regression in the area of KR (e.g., [21,13,24]). Regression is also a key component of the circuit-based filtering approach proposed by Shahaf and Amir [25]. Yet, none of the previous regression-based methods have been shown to be empirically efficient or competitive for modern, state-space planning, whereas the method we describe has been implemented as part of a number of state-of-the-art contingent planners [7,8].

Our first contribution is to extend Rintanen's formalism of regression [22] to handle observations, allowing us to use regression for online belief-state queries in domains with partial observability. Exploiting our focus on online belief maintenance, we obtain a method that is on par with the state of the art, as our empirical evaluation shows. On the theoretical side, we exploit the notion of width [19,5,6] to provide a tighter, width-based bound on the complexity of regression. We show that regression enjoys similar, and potentially better, complexity bounds to those of the state-of-the-art *factored belief tracking (FBT)* method [6]. FBT maintains an explicit model of a factored belief state which utilizes a notion of relevant variables. We show that one can ensure that the regression formula, too, will contain relevant variables only. This occurs naturally when regressing actions, but requires more care for regression of observations. Moreover, the worst-case complexity of regression depends only on the actions that actually appear in the sequence, as opposed to the entire set of actions in the progression-based FBT, potentially leading to a lower width notion, which we call *contextual width*.

Then, we use a well-known technique to compile non-deterministic actions into deterministic actions. Now, with deterministic actions only, one can easily update the initial state formula with the regression of each observation once, ignoring these observations afterward while maintaining completeness. This compilation leads to a surprising result – the existence of a poly-sized circuit representation of the regression formula (assuming polynomial horizon), which typically requires exponential space in other forms. Earlier work considered this to be possible only for domains with deterministic actions.

Next, we empirically evaluate our method, comparing the belief update and the query time of the regression method to approximate FBT, showing regression to be very efficient, scaling up similarly. We also empirically analyze the effect of simple caching – the main optimization to regression we suggest. We conclude with a discussion of related work.

2. Background

We begin by defining the contingent planning model and its specification language, following in most places the definitions of Bonet and Geffner (BG henceforth) [6]. We then review the various concepts of problem width introduced by Bonet and Geffner.

2.1. Model

We focus on contingent planning problems with sensing. A contingent planning problem is a tuple of the form $\langle S, b_I, S_G, A, Tr, \Omega, O\rangle$, where S is a set of states, $b_I \subseteq S$ is the set of possible initial states, also called the *initial belief state*, $S_G \subseteq S$ is the set of goal states, A is a set of action symbols, and Tr is a transition function, such that $Tr(s,a) \subseteq S$ is the set of states that can be reached by applying a in state s, Ω is a set of observation symbols, and $O(a,s') \in \Omega$ is the observation obtained when s' is reached following the application of a. Without loss of generality, we assume an observation follows each action.

In this paper we allow for non-deterministic outcomes of actions, but restrict ourselves to deterministic observations. Semantically, this is not a limiting assumption, as non-deterministic observations can be compiled away using non-deterministic actions; If $s' \in Tr(a,s)$ and $O(a,s') = \{o_1, o_2\}$, copy s' into two new states s'_1, s'_2 such that $s'_1, s'_2 \in Tr(a,s)$. Define $O(a,s'_1) = o_1$ and $O(a,s'_2) = o_2$. Transitions from s_1 and s_2 are identical to those from s.