Artificial Intelligence 212 (2014) 80–103

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

# Action-model acquisition for planning via transfer learning

Hankz Hankui Zhuo [a], Qiang Yang [b,*]

[a] School of Advanced Computing, Sun Yat-sen University, Guangzhou, China
[b] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clearwater Bay, Kowloon, Hong Kong

ARTICLE INFO

*Article history:*
Received 4 January 2013
Received in revised form 9 March 2014
Accepted 19 March 2014
Available online 25 March 2014

*Keywords:*
AI planning
Action model learning
Transfer learning
Markov logic networks
Web search

ABSTRACT

Applying learning techniques to acquire action models is an area of intense research interest. Most previous work in this area has assumed that there is a significant amount of training data available in a planning domain of interest. However, it is often difficult to acquire sufficient training data to ensure the learnt action models are of high quality. In this paper, we seek to explore a novel algorithm framework, called TRAMP, to learn action models with limited training data in a target domain, via transferring as much of the available information from other domains (called source domains) as possible to help the learning task, assuming action models in source domains can be transferred to the target domain. TRAMP transfers knowledge from source domains by first building structure mappings between source and target domains, and then exploiting extra knowledge from Web search to bridge and transfer knowledge from sources. Specifically, TRAMP first encodes training data with a set of propositions, and formulates the transferred knowledge as a set of weighted formulas. After that it learns action models for the target domain to best explain the set of propositions and the transferred knowledge. We empirically evaluate TRAMP in different settings to see their advantages and disadvantages in six planning domains, including four International Planning Competition (IPC) domains and two synthetic domains.



## 1. Introduction

AI planning techniques [21] often require a given set of action models as input. Creating action models, however, is a difficult task that costs much manual effort. The problem of action-model acquisition has drawn lots of interest from researchers in the past. For instance, McCluskey et al. [5,41] designed a system to interact with a human expert to generate action models. Amir [1] introduced a tractable and exact technique for learning action models known as Simultaneous Learning and Filtering, where the state observations were needed for learning. Yang et al. [67] presented a framework for automatically discovering STRIPS [20] action models from a set of successfully observed plans; Zhuo et al. [71] proposed to learn method preconditions and action models simultaneously for HTN planning, assuming that a set of hierarchical structures was given; Xu and Laird [65] proposed to simultaneously learn discrete and continuous action model in Soar cognitive architecture; just to name a few. Despite the success of the previous systems, they are all based on the assumption that there are enough training examples for learning high-quality action models. However, in many real-world applications, collecting a large amount of training examples for learning high-quality action models is often both difficult and costly. For example, in military operation, it is both difficult and expensive to collect a large number of plan traces (i.e., training data),

* Corresponding author.
*E-mail addresses:* zhuohank@mail.sysu.edu.cn (H.H. Zhuo), qyang@cse.ust.hk (Q. Yang).

http://dx.doi.org/10.1016/j.artint.2014.03.004
0004-3702/

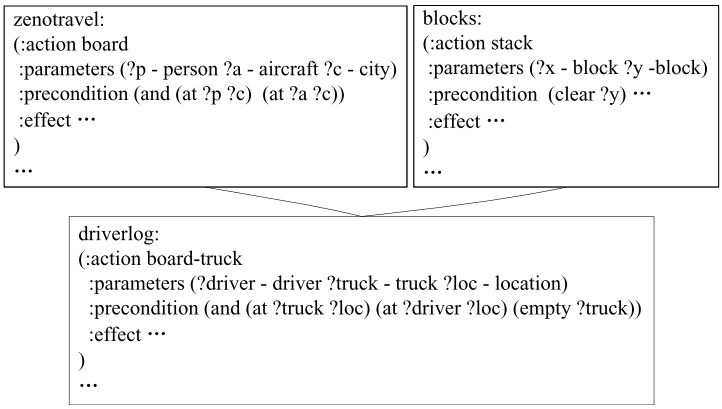


**Fig. 1.** Transfer knowledge from *board* and *stack* for learning *board-truck*.

since there is only a small amount of plan examples that can be collected in the real domain (e.g., a plan example may correspond to a war that does not happen frequently), and plan examples themselves may take lots of resources, such as ammunition. It is a difficult learning task when there is only a limited plan traces available [47].

By observation, we find that there is some knowledge available in other domains, which could be helpful for the learning. One motivating example for this problem can be illustrated in the domains *blocks*,[1] *zenotravel*[2] and *driverlog*[2], as shown in Fig. 1. Suppose that we wish to learn the logical models of the action "board-truck" in *driverlog* domain, which means that the driver "?driver" enters a truck "?truck" in a location "?loc". To execute this action, the current location of "?driver" and "?truck" should be in "?loc", indicating that the preconditions of "(at ?truck ?loc)" and "(at ?driver ?loc)" should be satisfied. This logical relationship is similar to the action of "board" from the *zenotravel* domain, which shows that if a person "?p" wants to enter an aircraft "?a" in a city "?c", the preconditions of "(at ?p ?c)" and "(at ?a ?c)" should be satisfied. Similarly, in the action "board-truck", if "?driver" wants to enter "?truck", "?truck" should be empty, which means the precondition of "(empty ?truck)" should be also satisfied. This relationship is similar to the action of "stack" from the domain *blocks*, which requires that the block "?y" should be clear (i.e. the precondition "(clear ?y)" is satisfied) if the block "?x" needs to be stacked on "?y". Thus, to learn the model of the action "board-truck", it is likely to be helpful to "borrow" the knowledge from the domains *zenotravel* and *blocks*.

Other motivation examples can also be found from many real world applications, such as *Coal mining* and *Urban search and rescue*. In the Coal mining domain, the goal of coal mining is to obtain coal from the ground. Coal mining has had a lot of developments over the recent years, from the early days of men tunneling, digging and manually extracting the coal on carts to large open cut and long wall mines. Mining at this scale requires the use of draglines, trucks, conveyor, jacks and shearers. Urban search and rescue (i.e., USAR) involves the location, extrication, and initial medical stabilization of victims trapped in confined spaces due to natural disasters, structural collapse, transportation accidents, and collapsed trenches. USAR services can be faced with complex rescue operations within hazardous environment. These two domains share some common knowledge that can be used to help each other. For example, in Fig. 2(a), the action *scoop-up* in the Coal mining domain, i.e., a robot scoops up coal using a coal shovel, is similar to the action *take-up* in the USAR domain (as is shown in Fig. 2(b)), i.e., a robot takes up a victim from hazardous environment. They both suggest that a robot is changing the location of some object (coal or victim). It would be greatly beneficial if we can transfer the knowledge from one domain where we have lots of knowledge on actions to another, where we have less domain knowledge, sine collecting a large number of plan traces from these domains is costly.

We propose a novel transfer learning algorithm framework to leverage knowledge from source domains, called TRAMP, which stands for **TR**ansfer learning **A**ction **M**odels for **P**lanning (the early version is presented in [70,73]). TRAMP seeks to learn action models in the target domain by transferring knowledge from source domains, assuming action models in source domains are already created by experts and can be transferred to the target domain. To transfer knowledge from source domains, TRAMP exploits two approaches to bridging source and target domains. The first approach is to autonomously map predicates and actions in source domains to the target domain via testing whether the mapped predicates and actions can best explain the plan traces in the target domain (the original idea was presented by our previous work [70]). Previous work [19] has demonstrated the success of structure-mapping in both theoretical and experimental aspects. The second approach is to bridge source and target domains via searching keywords of predicates and actions from the Web and calculating similarities of Web pages [10] (the original idea was presented by our previous work [73]). TRAMP is built based on Markov Logic Network (MLN) [53]. Specifically, TRAMP takes as inputs a set of action models from the source domains and a set of plan traces from the *target* domain, and outputs action models for the *target* domain (in this paper we focus

[1] http://www.cs.toronto.edu/aips2000/.
[2] http://planning.cis.strath.ac.uk/competition/.