软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(1):51−63 [doi: 10.13328/j.cnki.jos.004417]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

在部分观测环境下的不确定动作模型学习*

饶东宁[1], 蒋志华[2], 姜云飞[3]

[1](广东工业大学 计算机学院,广东 广州 510090)

[2](暨南大学 信息科学与技术学院 计算机科学系,广东 广州 510632)

[3](中山大学 信息科学与技术学院 软件研究所,广东 广州 510275)

通讯作者: 蒋志华, E-mail: tjiangzhh@jnu.edu.cn

摘 要: 近年来,动作模型学习引起了研究人员的极大兴趣.可是,尽管不确定规划已经研究了十几年,动作模型学习的研究仍然集中于经典的确定性动作模型上.提出了在部分观测环境下学习不确定动作模型的算法,该算法可应用于假定人们对转移系统一无所知的情形下进行,输入只有动作-观测序列.在现实世界中,这样的场景很常见.致力于动作是由简单逻辑结构组成的、且观测以一定频率出现的一类问题的研究.学习过程分为 3 个步骤:首先,计算命题在状态中成立的概率;然后,将命题抽取成效果模式,再抽取前提;最后,对效果模式进行聚类以去除冗余.在基准领域上进行的实验结果表明,动作模型学习技术可推广到不确定的部分观测环境中.

关键词: 人工智能;自动规划;动作模型学习;不确定动作;部分观测

中图法分类号: TP181 **文献标识码**: A

中文引用格式: 饶东宁,蒋志华,姜云飞.在部分观测环境下的不确定动作模型学习.软件学报,2014,25(1):51−63. http://www.jos.org.cn/1000-9825/4417.htm

英文引用格式: Rao DN, Jiang ZH, Jiang YF. Learning partially observable non-deterministic action models. Ruan Jian Xue Bao/Journal of Software, 2014,25(1):51−63 (in Chinese). http://www.jos.org.cn/1000-9825/4417.htm

Learning Partially Observable Non-Deterministic Action Models

RAO Dong-Ning[1], JIANG Zhi-Hua[2], JIANG Yun-Fei[3]

[1](School of Computers, Guangdong University of Technology, Guangzhou 510090, China)

[2](Department of Computer Science, School of Information Science and Technology, Ji'nan University, Guangzhou 510632, China)

[3](Software Research Institute, School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, China)

Corresponding author: JIANG Zhi-Hua, E-mail: tjiangzhh@jnu.edu.cn

Abstract: Recently, interests in learning action models have been increasing. Although non-deterministic planning has been developed for several decades, most previous studies in the field of action model learning still focus on classical and deterministic action models. This paper presents an algorithm for identifying non-deterministic actions, including effects and preconditions, in partially observable domains. It can be applied when people know nothing about a transferring system and only the action-observation sequences are given. Such scenarios are common in real-world applications. This work focuses on problems in which actions are composed of simple logical structures and features are observed under some frequency. The learning process is divided into three steps: First, compute the probability of each proposition which holds in a state. Second, extract effect schema from propositions and then extract preconditions. Third, cluster effect schema to remove redundancy. Experimental results on benchmark domains show that action model learning is still useful in non-deterministic and partial observable environments.

Key words: artificial intelligence; automated planning; learning action models; non-deterministic action; partial observability

领域建模是规划研究中的核心问题,而规划领域和规划问题的描述是领域建模中的关键.经过几十年的发

* 基金项目: 国家自然科学基金(61100134, 61003179); 广东省自然科学基金(S2011040001427)
收稿时间: 2012-08-13; 修改时间: 2013-01-25; 定稿时间: 2013-04-09

展,已有很多种规划领域建模语言被提了出来.比如,Stanford Research Institute Problem Solver(STRIPS)[1], Action Description Language(ADL)[2]以及 Planning Domain Definition Language(PDDL)[3].然而,从无至有的撰写领域描述是非常困难和消耗时间的,因此领域模型的获取,特别是自动获取是一个非常值得研究的课题.最近十几年来,利用学习技术来获取领域模型吸引了越来越多人的注意,其中,动作模型学习发展得最快.不过,迄今为止,大多数已有的动作模型学习都是针对经典规划领域,特别是 STRIPS 领域.

近年来,越来越多的实际应用问题被纳入到规划社区的视野中来.比如说,由于完全可观测对环境的要求过于苛刻,研究人员于是引入了部分可观测领域.这类研究主要针对不能观测到完整系统状态的情况.比如在互联网上,我们一次只能看到一部分网页,而不可能看到整个互联网.当规划目标可能发生变化时,动作模型学习显得尤为重要.比如,如果能够通过行为和反馈学习到自身的动作模型,一个自治智能体就可以自动地对自己的目标做出规划调整.现实世界中,应用问题的另一个特点是动作往往带有不确定性,也就是说,它们的效果可能会有多个.不过,学习不确定动作模型,特别是在部分观测环境中学习不确定动作模型是非常困难的,这主要是基于以下两个原因:首先,困难来自于缺少有用的条件独立的结构[4];其二,动作的不确定性使得学习的效果更加难以保证.

不确定规划一直是自动规划领域的研究热点[5−9],在不确定规划中,动作效果是不确定的,或者初始状态是不确定的.这样的问题比经典规划问题更难[5],因为追踪信念状态比追踪状态更难,并且求解动作策略比求解动作序列更难.这些不确定性往往可以通过对环境的观测来识别.出于节约成本的考虑,环境经常假设为部分可观测的,此时状态信息是不完备的,状态变成了信念状态.在部分观测环境下的不确定规划分为两类:随机规划和 POMDP 规划.在随机规划[5−7]中,信念状态是状态的集合,求解方法主要是在信念空间进行搜索;而在 POMDP 规划[8,9]中,信念状态由在状态集合上的概率分布来表示,求解方法主要依赖于 MDP 方法中的值迭代和策略迭代过程.此外,在不确定规划中,动作模型的自动获取也同样受到广泛的关注.但是,已有的研究成果主要集中在部分观测环境下学习确定性的动作模型[4,10−13],或者是完全观测环境下学习概率动作效果[14−18],而对部分观测环境下学习不确定动作模型的研究几乎没有.如何处理部分观测的信息以及如何处理动作的不确定效果对学习精度的影响,成为解决这一难题的关键.

为了尝试解决这一问题,本文提出一种识别不确定动作模型的算法,该算法在部分观测环境中学习不确定动作的前提和多个效果.它使用一个序列化的观测和动作作为输入,然后输出可能带来这些观测结果的动作模型.学习过程分为 3 个步骤:首先,根据最小修改假定和均等机会假定来计算命题在状态中成立的概率;然后,将在相邻状态中概率变化明显的命题抽取成效果模式,根据效果模式来抽取前提;最后,对效果模式进行聚类以去除冗余.其中,用于处理观测信息的两个假定和用于对不确定性结果进行去冗的聚类方法是使得在部分观测下学习不确定动作模型可行的关键方法.在基准领域上进行的实验结果说明,动作模型学习技术在部分观测的不确定规划领域仍然有效.这是首个在部分观测环境中进行的不确定动作模型学习的一种尝试.本文所提出的技术对于概率领域也是可行的.

本文第 1 节给出动作模型学习的文献回顾.在第 2 节中定义学习问题.然后,具体算法在第 3 节中给出.第 4 节的内容是实验部分,包括实验环境、评估方法和实验结果.最后,第 5 节给出本文的总结和对将来工作的展望.

1 动作模型学习

动作模型学习技术已经发展了 20 多年,其中有两大趋势:学习被逐步扩展到部分观测领域;学习的领域越来越具有不确定属性.

1.1 在部分观测环境中学习动作模型

在动作模型学习的研究中,第一个趋势是状态信息越来越不完备,研究逐渐延伸到部分观测领域.这是因为,很多时候完全观测在现实世界中难以保证.于是与相关领域一样[19],研究者们越来越多地开始将部分观测环境问题考虑进来.

在最开始的时候,研究人员把精力集中在完全观测领域.在 1994 年,Sablon 和 Bruynooghe[20]学习了事件演

算.然后在1995年,Wang提出了一种通过观测和实践来学习操作的算法[21].5年后,Balac利用回归树来学习动作模型[22].到此为止,所有的研究都是在完全观测领域中进行的.到21世纪,动作模型学习领域被扩展到了部分观测领域.就在2000年,Schmill使用聚类和决策树推导来学习部分观测环境中的动作模型[12].4年后,在部分观测马尔可夫过程(POMDP)中学习动作模型的方法被Holmes和Isbell提了出来[11].直到最近,不完备信息的说法被正式地加以使用.其中在2007年,Yang[10]提出了在不完备信息条件下从规划例中学习动作模型的方法,主要技术是基于约束的特征选择.随后,Zhou[13]扩展了Yang的做法,在不完备信息条件下学习包含量词和蕴涵关系的复杂动作模型.另外一个例子是Amir在2008年的工作[4],他假设所有的特性都能够被观测,但是被观测得到的频率是指定的,然后在这种环境下学习动作模型,具体方法是使用领域动作来演化信念状态以及使用观测动作来过滤信念状态.不过,上述所有工作都是针对确定性动作模型的.

1.2 支持不确定特性

在动作模型学习的研究中,第二个趋势是越来越多地支持不确定特性.STRIPS有很多不现实的假设,所以很多应用都无法用STRIPS建模.为了移除这些不现实的假设,STRIPS的后继版本逐步增加了很多新特性,其中最重要的就是不确定性.例如,Probabilistic Planning Domain Definition Language(PPDDL)[23]就支持不确定动作以及带有概率的动作效果.为了跟上PDDL演进的步伐,动作模型学习领域的学者们也逐渐开始支持新的特性,比如Rao[24]在2010年提出学习PDDL中的派生谓词规则[3].

不过,在这些新特性中,最早被学习的仍然是不确定性,包括概率特性.在1992年,Chrisman提出了一种抽象得到概率动作模型的方法[14].4年后,Oates和Cohen学习了带有概率效果的动作模型[15].而最近几年,对概率动作模型的学习有复兴的趋势.比如2007年,Pasula学习了随机领域中的符号化动作模型[16].随后,Hajishirzi在假设只有初始状态的分布已知的情况下学习动作模型[17].再比如,在2010年,Rao学习了网络服务的不确定动作模型[18].不过,上述这些研究都是基于完全观测的这一假设的.

2 问题陈述

本文的目标问题是跟踪一个动态系统,然后通过所得到的时间步和部分观测序列来学习不确定动作模型.同时,像一些已有研究一样[4],初始状态是未知的.对这个问题的解,应该是一个动作模型的组合,规划器使用这个动作模型的组合能够给出期望的观测序列.当然,对这个解的计算和验证可以通过递归地在时间步 t 进行演算,然后得到时间步 t+1 的信念状态来进行.这涉及到确定一组可能的动作模型以及这些动作模型可能改变系统状态的方式.这种改变的方式就是转移模型,它决定了系统可能处在哪些状态中.也就是说,任意一个转移模型决定了一系列可能的状态,一个目标问题的解就是一个转移模型及其所关联的可能状态.在本节,我们将形式化地定义本文的目标问题.

定义 1. 一个转移系统是一个四元组$\langle P,S,A,R\rangle$,其中,

- P 是一个有限命题集合;
- $S\subseteq 2^P$ 是一个状态集合;
- A 是一个有限动作集合,$A=\{a|a=\langle \mathrm{pre}(a),\mathrm{add}(a),\mathrm{del}(a)\rangle\}$.其中,$\mathrm{pre}(a),\mathrm{add}(a),\mathrm{del}(a)\subseteq 2^P$,分别表示动作 a 的前提集合、增加效果集合和删除效果集合;
- $R\subseteq S\times A\times S$ 是一个转换关系,序对$\langle s,a,s'\rangle$表示状态 s 经过应用动作 a 可转换到 s',即

$$s'=(s-\mathrm{del}(a))\cup\mathrm{add}(a).$$

在上述定义中,对于给定的 s 和 a,如果 s'是唯一的,则称该转移系统是确定的;反之,如果 s'不唯一,则称该转移系统是不确定的.上述形式化的转移系统最终需要通过规划定义语言来描述,才能使用规划系统进行求解.下面我们通过一个例子来说明一个具体的转移系统是怎样用规划语言来描述的.例1给出了概率规划大赛中一个基准领域的描述.国际规划大赛(Int'l Planning Competition,简称IPC)是智能规划主要的交流平台,自从2004年起,规划大赛分设了专门进行概率规划领域的比赛,即国际概率规划大赛(Int'l Probabilistic Planning Competition,简称IPPC).考虑到本文系统的通用性以及测试领域的代表性,我们采用国际概率规划大赛的基准

领域进行介绍和测试.

例 1:在国际概率规划大赛 2008(IPPC 2008)的基准领域中,有一个 Blocksworld 领域(如图 1 所示),它是使用 PPDDL 定义的概率规划领域.其中,为了定义概率的和决策论的规划领域和问题,PPDDL 增加了对概率效果的支持.PPDDL 中对概率效果的语法为(probabilistic p_1 e_1 ... p_k e_k),意思是效果 e_i 会以概率 p_i 发生,并且 $\sum_{i=1}^{k} p_i = 1$. 例如,动作 pick-up 是不确定动作,它有两组效果,第 1 组效果发生的概率是 3/4,第 2 组效果发生的概率为 1/4.

```
//domain definition
(define (domain blocks-domain)
  (:requirements :probabilistic-effects :equality :typing)
  (:types block)
  (:predicates (holding ?b - block) (emptyhand) (on-table ?b - block) (on ?b1 ?b2 - block) (clear ?b - block))
  (:action pick-up
    :parameters (?b1 ?b2 - block)
    :precondition (and (emptyhand) (clear ?b1) (on ?b1 ?b2))
    :effect
      (probabilistic
        3/4 (and (holding ?b1) (clear ?b2) (not (emptyhand)) (not (on ?b1 ?b2)))
        1/4 (and (clear ?b2) (on-table ?b1) (not (on ?b1 ?b2))))
          )
……
//problem definition
(define (problem bw_5_p01)
  (:domain blocks-domain)
  (:objects b1 b2 b3 b4 b5 - block)
  (:init (emptyhand) (on-table b1) (on-table b2) (on b3 b5) (on b4 b1) (on-table b5) (clear b2) (clear b3) (clear b4))
  (:goal (and (emptyhand) (on b1 b3) (on b2 b4) (on-table b3) (on b4 b1) (on b5 b2) (clear b5)))
  (:goal-reward 1)
  (:metric maximize (reward))
)
```

Fig.1 Blocksworld domain in the IPPC 2008

图 1 IPPC 2008 Blocksworld 领域

接下来,我们定义观测.一个状态 $s \subseteq S$ 是 P 的一个子集,它包括了那些在该状态中为真的命题.一个文字是一个命题 $p \in P$ 或者它的否定 $\neg p$.如果一个文字的真值随着状态的变化而变化,我们称其为流文字(fluent literals),反之,称为非流文字(non-fluent literals).例如,在 Blocksworld 领域中,on(A,B)在当前状态下为真,在应用了动作 pickup(A,B)之后,在下一个状态下为假,因此,on(A,B)是流文字.而 block(A)在任何状态下都为真,因此它不是流文字.而我们需要观测的显然是流文字,在这里,一个观测变量就是一个流文字.在已有的研究中,一个观测可以定义成动作[4],也可以定义成一个观测变量构成的逻辑公式[5].当定义成动作时,观测 o 是一个序对⟨pre(o),obs(o)⟩,其中,pre(o)和 obs(o)都是命题集合,该定义表明:当 $pre(o) \subseteq s$ 成立时,可以观测到 obs(o)中命题的真值.这样定义的好处是把观测动作当成领域动作一样来处理.而当定义成逻辑公式时,观测 o 是观测变量集合 L 的一个合取范式(conjunctive normal form,简称 CNF),它表明:每个合取项在当前状态 s 下的真值为真,其中,合取项是由观测变量组成的逻辑公式.这样定义的好处是便于向状态中添加经过观测而确立的命题.本文采用观测的第 2 种定义,并且为了简化处理,限定每个合取项至多由一个观测变量组成.

定义 2. 给定状态 s,一个观测 o 是在观测变量集合 L(即流文字集合)上受限的合取范式,即 $o=l_1 \wedge l_1 \wedge \ldots \wedge l_k$,其中,$l_i \in L$,$1 \leqslant i \leqslant k$.观测 o 表明,l_i($1 \leqslant i \leqslant k$)在 s 中的真值为真.

最后,我们定义本文的学习问题.在现实世界的某些应用中,一个 agent 有可能对周围的环境是一无所知的,只能通过与环境交互来获取环境的信息.因此,在本文的学习问题中,转移系统的各组成部分可能是完全不知道的,学习目标是通过给定的动作-观测系列来获取转移系统的核心部分——转换关系.此外,若无特别说明,本文中所有的观测都假设为准确的.

定义 3. 一个在部分观测领域学习不确定动作模型的问题是,从一个 t 步动作-观测序列⟨a_i,o_i⟩($1 \leqslant i \leqslant t$)中学习转移系统中转换关系 R 的子集.

为了更好地描述上述学习问题的输入,例2给出了IPPC 2008 Blocksworld领域的一个实例(instance)产生的⟨动作,观测⟩序列片段.

例 2:求解例 1 中问题的⟨动作,观测⟩序列的片段如图 2 所示.其中,第 1 行表示动作 pick-up(b4,b1),第 2 行~第 11 行表示执行完该动作所做的观测,该观测表明哪些文字在应用动作 pick-up(b4,b1)之后的状态里为真,例如 on-table(b1),clear(b1),holding(b4)的真值为真等等.

```
……
<action><name>pick-up</name><term>b4</term><term>b1</term></action>
<state><atom><predicate>on-table</predicate><term>b1</term></atom>
<atom><predicate>on-table</predicate><term>b2</term></atom>
<atom><predicate>on</predicate><term>b3</term><term>b5</term></atom>
<atom><predicate>on-table</predicate><term>b5</term></atom>
<atom><predicate>clear</predicate><term>b2</term></atom>
<atom><predicate>clear</predicate><term>b3</term></atom>
<atom><predicate>clear</predicate><term>b4</term></atom>
<atom><predicate>clear</predicate><term>b1</term></atom>
<atom><predicate>holding</predicate><term>b4</term></atom>
<fluent><function>reward</function><value>0</value></fluent></state>
……
```

Fig.2 A series of actions and observations for Example 1

图 2 例 1 中问题产生的⟨动作,观测⟩序列

3 学习不确定动作模型

为了解决本文的学习问题,我们首先要从动作-观测序列中建立状态序列,并且计算每个命题在其中成立的概率;然后,我们会先学习每个动作的前提,再学习每个动作的多个效果.得到的状态序列以及学习到的动作即可构成转换关系中的序对$\langle s,a,s'\rangle$.

3.1 建立状态序列

动作的前提和效果反映的是在相邻状态中变化的命题.因此,对于本文学习问题中的唯一输入——动作-观测序列,我们首先需要重建状态序列.同时,由于初始状态未知,而且命题的真值由于观测的不完备性可能在状态中不可知,所以对于所有的$p\in P$,我们需要判断它在状态中成立的概率.为了进行这样的判断,我们需要做一些假定,这些假定用于处理观测中的信息,以便状态的重建.

假定 1(最少修改假设). 给定一个文字 l 和一个 t 步动作-观测序列$\langle a_i,o_i\rangle(1\leqslant i\leqslant t)$,并且假设在动作 a_i 之前的状态是 s_{i-1}.如果 $l\in o_i$ 并且对$\forall j(0<j<i)$有 $l\notin o_j$ 成立,则有 $l\notin s_j$ 成立.另外,如果 $l\in o_i$,并且对$\forall j(i<j\leqslant t)$有 $l\notin o_j$ 成立,则有 $l\in s_j$ 成立.

做出假设 1 的原因是动作效果中的文字相对于所有文字的全集来说是非常少的,而且初始状态也没有给定.换言之,一个文字在第 1 次被观测到之前被改变的概率是非常小的.类似地,一个文字在最后一次被观测到之后被修改的概率也是非常小的.实际上,如果不做出这样的假设,那么相关的信息就会变得非常不可知,我们的问题也就演变成了状态信息不完备的情况.

假定 2(均等机会假设). 给定一个文字 l 和一个 t 步动作-观测序列$\langle a_i,o_i\rangle(1\leqslant i\leqslant t)$,并且假设在动作 a_i 之前的状态是 s_{i-1}.如果对于 $1\leqslant j<i\leqslant t$,有 $l\in o_i$ 且$\neg l\in o_j$,并且对$\forall k(j<k<i)$有 $l\notin o_k$ 且$\neg l\notin o_k$,则 $l\in s_k$ 的概率是

$$\sum_{m=0}^{k-j-1}\left(1-\frac{1}{i-j}\right)^m\frac{1}{i-j}.$$

做出假设 2 的原因是:在建立状态序列的时候,动作的效果是未知的,因此无法判定具体是在哪一步产生相应的变化.那么,我们只能假定每个动作都有相同的机会去改变一个文字.如假定 2 所述,当 $j<i$ 且 $l\in o_i$ 和$\neg l\in o_j$时,$a_{j+1},\ldots,a_i$ 均有 $\frac{1}{i-j}$ 的概率使得 l 为真,那么 $l\in s_k(j<k<i)$的概率为

$$\frac{1}{i-j}+\left(1-\frac{1}{i-j}\right)\frac{1}{i-j}+\left(1-\frac{1}{i-j}\right)^2\frac{1}{i-j}+\ldots+\left(1-\frac{1}{i-j}\right)^{k-j-1}\frac{1}{i-j}=\sum_{m=0}^{k-j-1}\left(1-\frac{1}{i-j}\right)^m\frac{1}{i-j}.$$

下面的算法 1 给出了建立状态序列的过程.在这个算法中,*prob* 是二维数组,每个元素 *prob*[*i*][*j*]表示命题集 *P* 中第 *j* 个流文字 p_j 在 s_i 中成立的概率,其中,$1\leqslant j\leqslant n$,$n=|P|$,$1\leqslant i\leqslant t$.

算法 1. 状态建立算法.

输入:*t* 步动作-观测序列$\langle a_i,o_i\rangle(1\leqslant i\leqslant t)$;

输出:命题集合 *P*,状态序列 $S=\langle s_0,\ldots,s_t\rangle$,动作集合 *A* 和概率数组 *prob*[1...*t*][1...*n*].

1. for every $\langle a_i,o_i\rangle$,$1\leqslant i\leqslant t$
2. for every $p\in o_i$ or $\neg p\in o_i$
3. $P=P\cup\{p\}$;
4. $S=S\cup\{o_i\}$;
5. $A=A\cup\{a_i\}$;
6. for every o_i, $1\leqslant i\leqslant t$
7. for every $p\in P$
8. if $p\in o_i$
9. for $j=i-1$ to 1
10. if $p\in o_j$
11. break;
12. else if $\neg p\in o_j$
13. for $k=j+1$ to $i-1$
14. $prob[k][k']=\sum_{m=0}^{k-j-1}\left(1-\frac{1}{i-j}\right)^m\frac{1}{i-j}$, assume that p is the k' element in P;
15. for $k=i$ to t
16. $prob[k][k']=1$;
17. break;
18. for $k=i$ to t
19. $prob[k][k']=1$;
20. else if $\neg p\in o_i$
21. for $j=i-1$ to 1
22. if $p\in o_j$
23. for $k=j+1$ to $i-1$
24. $prob[k][k']=1-\sum_{m=0}^{k-j-1}\left(1-\frac{1}{i-j}\right)^m\frac{1}{i-j}$, assume that p is the k' element in P
25. for $k=i$ to t
26. $prob[k][k']=0$;
27. break;
28. else if $\neg p\in o_j$
29. break;
30. for $k=i$ to t
31. $prob[k][k']=0$;
32. return *P*, *S*, *A*, *prob*;

关于算法 1 的解释如下:在算法 1 中,第 1 行~第 5 行重建了 *P*,*S*,*A*,其中,*P* 的元素来自于观测中出现的命题,

由于观测的不完备性,S 是信念状态集合,A 由动作-观测序列中的动作构成.第 6 行~第 31 行用来计算各命题在状态中出现的概率,其中,第 15 行、第 16 行、第 18 行、第 19 行、第 25 行、第 26 行、第 30 行和第 31 行的依据是假设 1,而第 13 行、第 14 行、第 23 行和第 24 行的依据是假设 2.对于每个在观测中出现的文字,分为正文字和负文字两种情况来讨论:如果是正文字(第 8 行),并且在之前的观测中出现其负文字(第 12 行),则依据假设 2,这两个观测之间的动作有均等机会使得该文字对应的命题为真(即动作的增加效果),因而计算其在两个观测之间的状态中出现的概率(第 13 行、第 14 行);负文字的情况与正文字的情况类似,区别只是在于如果在之前的观测中出现其正文字(第 22 行),那么这两个观测之间的动作有均等机会使得该文字对应的命题为假(即动作的删除效果),因此,该命题在状态中成立的概率为 1 减去其不成立的概率(第 24 行).

算法 1 的时间复杂性是 $O(n\times t^3)$,其中,$n=|P|$,P 是命题集合,t 是动作-观测序列的步数.分析如下:在算法 1 中,第 1 行~第 5 行的时间复杂性是 $O(n\times t)$.第 6 行循环的次数是 $O(t)$,而第 7 行循环的次数是 $O(n)$.接着,算法的第 8 行~第 19 行处理观测中的正文字.当观测中出现正文字时,即第 8 行的条件成立,第 14 行和第 16 行的计算在最坏情况下均执行 $O(t^2)$次.而第 18 行的循环在最坏情况下执行 $O(t)$次,第 18 行与第 9 行是顺序进行的.因此算法的第 8 行~第 19 行的时间复杂性是 $O(n\times t^3)$.不难看到,算法的第 8 行~第 19 行和第 20 行~第 31 行是两个对称结构,分别处理观测中的正文字和负文字,因而这两部分的时间复杂性是相同的.于是,整个算法的时间复杂性是 $O(n\times t^3)$.实际上,在大多数情况下,算法 1 执行得很快,因为在一般情况下,我们有 $n>>t$.

3.2 学习动作前提

根据动作模型的定义,前提是在状态中应用动作的充要条件,即动作在状态中可应用,当且仅当该动作的前提在状态中是成立的.因此,从概率统计的观点来看,如果文字 l 是动作前提的一部分,那么在动作执行前的所有状态中该文字成立的概率要么是 1,要么接近 1.此外,由于 PDDL 语言所描述的动作模型是基于一阶谓词逻辑的,因此我们需要在抽取动作前提之前将状态之间的变化进行模式化.算法 2 实现了上述思想.

算法 2. 前提抽取算法.

输入:t 步动作-观测序列$\langle a_i,o_i\rangle(1\leqslant i\leqslant t)$,命题集合 P,动作集合 A,概率数组 $prob[1...t][1...n]$,阈值Δ;

输出:动作 a_i 的前提集合 $precondition_i$ 以及效果集合 $effect_i(1\leqslant i\leqslant|A|)$.

```
1.  for every action a_i∈A, 1≤i≤|A|
2.      prec=∅;
3.      for every a_j in ⟨a_j,o_j⟩, 1≤j≤t
4.          if a_i=a_j
5.              iniPrec←literals which appear in both of s_{i−1} and s_{j−1} with the probability 1;
6.              probDiff=prob[j−1]−prob[j];
7.              for k=1 to n
8.                  iniEffect =∅
9.                  if |probDiff[k]|>Δ
10.                     schema p_k as action effect effSch, assume that p_k is the k proposition in P;
11.                     iniEffect←effSch;
12.             effect_i←iniEffect
13.             remove literals which have no variable correlations with iniEffect from iniPrec
14.             schema iniPrec as action precondition precSch according to iniEffect;
15.             prec←precSch;
16.     precondition_i←the intersection among subset elements in prec;
17.  return precondition_i and effect_i, 1≤i≤|A|;
```

关于算法 2 的解释如下:首先,在算法 2 的第 4 行是对动作名进行比较,而不是对整个动作实例进行比较.这是因为动作模型是基于一阶逻辑的,因此只要动作实例的名字相同,就都属于同一动作模型,就可以用来提取前

提和效果;其次,在算法 2 中,*probDiff* 是矢量,表示在动作执行前后的两个状态中所有命题的概率差.在第 9 行中,当命题在前后两个状态中成立的概率差超过给定的阈值时,就认为该命题被动作改变,成为动作的效果.接着,*effSch* 表示模式化的效果,*precSch* 表示模式化的前提,而模式化指的是将对象常量转化为参数变量.例如,将命题 on(*A*,*B*)模式化的结果为 on(*x*,*y*).在第 14 行中,根据模式化的效果来模式化前提,可以减少无关对象的干扰而方便模式的匹配,也为后面提取效果做好准备.所涉及到的步骤包括:1) 先求出 s_{i-1} 和 s_{j-1} 中相同的且出现概率为 1 的文字,作为动作 a_i 的初始前提集合 *iniPrec*(第 5 行);2) 根据变量关联来筛选 *iniPrec* 中的前提,即与效果 *effSch* 有变量关联的前提留下,没有变量关联的前提被丢弃(第 13 行);3) 根据模式化的效果来模式化前提,即在效果中已有的变量绑定在前提中延续,而在效果中未出现但在前提中出现的新对象被绑定新变量(第 14 行);最后,在第 16 行中,*prec* 是模式化的前提集合,其元素的交集为动作 a_i 的前提 $precondition_i$,而求元素的交集实质上是进行模式匹配的过程.算法 2 除输出动作前提外,还输出包含冗余的效果集合,为下一步提取动作效果做好准备.

算法 2 的时间复杂性为 $O(m\times t\times n^2)$,其中,$m=|A|$,$n=|P|$,A 是动作集合,P 是命题集合,t 为动作-观测序列步长.分析如下:算法 2 的主体由 3 重循环组成:第 1 重循环是在第 1 行,共循环 $O(m)$次;第 2 重循环是在第 3 行,其循环次数为 $O(t)$;而第 3 重循环是在第 7 行,其循环次数为 $O(n)$.在第 2 重循环内部:第 5 行求两个状态的交集,时间复杂性为 $O(n)$;第 6 行求矢量差,时间复杂性也为 $O(n)$.在第 3 重循环内部:第 10 行将单个命题模式化成谓词可以看成是在常数时间内完成,这是因为一般情况下谓词的参数个数都是个位数;第 13 行过滤前提,由于 *iniEffect* 和 *iniPrec* 的大小均不超过 $O(n)$,因此时间复杂性为 $O(n^2)$.同理,第 14 行模式化前提的时间复杂性为 $O(n^2)$;最后,第 16 行中求子集族中元素的交集,时间复杂性为 $O(n^2)$.所以,整个算法 2 的时间复杂性为 $O(m\times t\times n^2)$.此外,A 中的动作均来自于动作-观测序列,因此我们有 $m\leqslant t$,算法 2 的时间复杂性也简略地表示为 $O(t^2\times n^2)$.

不过,在算法 2 中,阈值的选择需要一个标准.在本文中,我们采用一种拟合实验数据的方式.具体地说,我们希望学习所得的前提中文字的数目与原有标准领域定义的动作前提中文字数目基本吻合.为此,我们列举领域中动作的前提数,然后用它的平均值作为该领域期望的动作模型前提文字数.实际上,对于所有的 IPPC 2008 领域我们都进行了统计.作为例子,表 1 展示了 Blocksworld 领域的情况,其前提中文字个数的平均数为 3.在实验中,我们会调整阈值,使得输出的动作前提的文字个数与这个期望值相近.

Table 1 Number of literals in preconditions (IPPC 2008 Blocksworld domain)

表 1 动作前提中的文字个数(IPPC 2008 Blocksworld 领域)

Domain name	Action name	Number of literals in precondition
Blocks-Domain	pick-up	3
	pick-up-from-table	3
	put-on-block	4
	put-down	2
	pick-tower	4
	put-tower-on-block	4
	put-tower-down	2

3.3 学习动作效果

在算法 2 中,在模式化前提之前我们实际上已经得到了所有效果的模式.不过,这些模式有很多是相同的.进一步地,由于部分观测的环境,很多形式上差别不大的模式实际上来自于相同的动作效果.例如,两组效果模式{clear(*x*),holding(*y*),not on(*y*,*x*)}和{clear(*y*),holding(*x*),not on(*x*,*y*)}实际上表示相同的含义.因此,从算法 2 中得到的同一动作的效果模式的数量会远远大于所需要的动作效果个数.这个事实要求我们对相同或者相似的效果进行合并.这个问题实际上是相关研究中实体相似度判断的问题[25].因此,我们借用相似度判断的方式来对算法 2 中得到的效果进行聚类.其中的关键在于如何判断两个效果模式的相似程度,我们采用已有定义[25]来判断.

定义 4(实体相似度). 如果两个实体 A 和 B 都是谓词集合,并且这两个集合的秩分别为$|A|$和$|B|$,那么这两个实体的相似度为 $ES(A,B)=\dfrac{|A\cap B|}{|A\cup B|}$.

如果两个实体 A 和 B 的相似度足够大,那么它们就可以被视为是同一个实体.当然,在进行比较时会进行一

些简单的变量替换的过程,可能会因此带来误差.整个算法见算法 3.

算法 3. 效果聚类算法.

输入:效果集合 $effect_j(1\leqslant j\leqslant|A|)$,阈值$\Delta$;

输出:效果集合 $effect_j(1\leqslant j\leqslant|A|)$.

1. for every action $effect_j$, $1\leqslant j\leqslant|A|$
2. for every $effSch_i\in effect_j$, $1\leqslant i\leqslant|effect_j|$
3. for every $effSch_k\in effect_j$, $i<k\leqslant|effect_j|$
4. if ($ES(effSch_i,effSch_k)>\Delta$)
5. $effSch_i=effSch_i\cap effSch_k$;
6. increase the number of samples in $effSch_i$;
7. $effect_j=effect_j-\{effSch_k\}$;
8. return $effect_j$, $1\leqslant j\leqslant|A|$;

关于算法 3 的解释如下:算法 3 对每个动作的效果集合进行聚类.这些效果来自于动作-观测序列中被动作明显改变的命题.只要与给定的动作名相同,每个动作实例就产生一组效果.这些效果可能包含冗余,因此需要进行聚类.第 4 行计算效果模式的相似度,达到一定的阈值就认为它们是一样的.第 6 行记录一类效果模式的样例数,以便观察冗余程度.然而在具体的实现过程中,为了保证效果模式的有效性,我们在开始算法 3 之前还有一个效果模式的筛选过程,即一个效果模式必须覆盖一定数量的样例之后才被保留,这样保留下来的效果模式才比较有代表性.我们称这个样例的数量为最小覆盖样例数.

算法 3 的时间复杂性为 $O(t^2\times n^2\times m)$,其中,$m=|A|$,$n=|P|$,A 是动作集合,P 是命题集合,t 为动作-观测序列步长.分析如下:从算法 2 可知,每个动作最多有 t 个效果模式,即$|effect_j|\leqslant t$,这是因为在算法 2 的第 3 行的循环中每次最多产生一个效果模式.在算法 3 中,第 1 行~第 3 行构成一个 3 重循环:第 1 重循环共循环 m 次,第 2 重和第 3 重循环均循环 t 次.在第 3 重循环内部,第 4 行计算实体相似度所花费的时间为 $O(n^2)$,因为每个效果模式的大小不超过 $O(n)$.因此,算法 3 的时间复杂性为 $O(t^2\times n^2\times m)$.同上,A 中的动作均来自于动作-观测序列,因此有 $m\leqslant t$,算法 3 的时间复杂性也简略地表示为 $O(t^3\times n^2)$.

像第 3.2 节对前提的提取一样,阈值的选择在算法 3 中依旧需要一个准则.本文中,我们继续采用数据拟合的方式,即要求学习到的动作效果个数与原有领域描述中动作的效果个数基本一致.为此,我们统计了每个领域的动作效果数.表 2 中就以 IPPC 2008 的 Blocksworld 领域为例展示了动作效果的数目,其平均值为 2.我们在实验中就以这个平均值为目标调校阈值.

Table 2 Number of effects for actions (IPPC 2008 Blocksworld domains)

表 2 动作效果的个数(IPPC 2008 Blocksworld 领域)

Domain name	Action name	Number of literals in precondition
Blocksworld	pick-up	2
	pick-up-from-table	2
	put-on-block	2
	put-down	1
	pick-tower	2
	put-tower-on-block	2
	put-tower-down	1

4 实 验

本节将评估本文系统的有效性.为此,我们首先介绍 IPPC 的比赛环境和我们的实验设置;随后,在给出我们的实验结果之前定义错误率,以评估学习的效果;最后给出实验结果.

4.1 实验设置

IPPC 2008 共设有 Fully Observable Probabilistic(FOP),Non Observable Non-Deterministic(NOND)和 Fully

Observable Non-Deterministic(FOND)等几个部分.它们都采用了评估服务器-规划器客户机的模式.测试中的每一轮分 4 步,循环进行:第 1 步是服务器产生初始状态并发给规划器,第 2 步是规划器在线产生规划并返回一个动作,第 3 步是服务器执行动作并评估效果,最后一步是服务器后发送新的状态给规划器.

我们采用 FOP 领域作为基准测试领域,这主要是为了证明我们的方法可以扩展到概率环境.在上述第 3 步中产生的内容将作为动作序列.为了模拟部分观测环境,我们随机地选择一定比例的流文字标记为已知,作为观测的结果,即设置一个观测率λ.这些观测加入到规划解中,构成动作-观测序列.然后,采用 5-交叉验证的测试方法来学习动作模型,即 4/5 的训练例用来学习动作模型,而剩下的 1/5 用来进行评估.具体地说,我们采用 MDPSim (http://code.google.com/p/mdpsim/),IPPC 2008 的官方软件来产生训练例.所有的实验都是在 PC 工作站上进行的,其中,CPU 是 2.4 GHZ Celeron,内存为 2 GB.MDPSim 软件是 C 编写的,我们在 Ubuntu Server 10.04.3 LTS 上运行它.我们的系统是采用 Java 来编写的,在 Windows 2008 Server 上加以运行.对于 IPPC 2008 FOP 的所有领域,我们采用 10 个问题,每个问题 10 次测试,每次测试取 100 步的方法.即总共产生 10 000 个训练例,其中,8 000 个用于训练,2 000 个用于评估.在实际开发过程中,出于编码上的方便,我们将算法 2 和算法 3 加以混编.具体地,我们会首先学习邻接状态间的变化,然后在聚类效果的同时建立动作的模式,而前提的抽取被放在最后.

此外,这里补充关于训练例数量的说明.在部分观测环境下的实验需要体现统计特性,因此往往需要比较大量的数据来说明问题.在前人的一些工作中,例如 Yang 等人[10]的工作,采用 100~200 个规划解作为训练数据集,但是每个规划解并不限制规划步骤的长度.只是对于规划比赛中的基准问题,规划解长度一般小于 100,因此按照规划步来计算训练例总数在 10 000 左右.而在另一些前人的工作中,例如 Amir 等人[4]的工作,则是直接以规划步来作为训练例,数目在 5 000 个左右.因此,本文采用 10 000 个规划步作为训练例集合,是参考了同类相关工作的执行标准.

4.2 错误率定义

参考已有的研究[19],我们说一个动作的前提不能被动作的前驱状态满足时就产生了一个错误,这时,学习到的动作前提没有对训练例进行正确的解释.同样地,如果一个动作的效果在其后继状态中不能被满足,也称出现了一个错误.但是对于概率效果而言,更重要的是用概率分布的差异来表示错误率.

定义 5. 前提错误率 $error_{pre}$ 为 m/n,其中,m 是错误解释的训练例的数量,而 n 是总的训练例的数量.

定义 6. 概率效果错误率 $error_{effs}$ 为概率分布的距离.如果原有动作描述中的分布是 $P(x)$,而学习得到的分布为 $Q(x)$,那么其距离为 $D_{KL}(P\|Q)=\int_{-\infty}^{\infty}p(x)\ln\frac{p(x)}{q(x)}\mathrm{d}x$.

不过,考虑到 IPPC 2008 中仅采用了 Bernoulli 分布,我们在测试中用 Bernoulli 分布的参数的差作为效果错误率的简单估计.

定义 7. 动作模型错误率: $error_{AS}=\frac{error_{prec}+error_{effs}}{2}$.

类似地,本文中领域错误率定义为其所包含的动作模型的错误率的平均值.

4.3 实验结果

由于没有其他的研究工作进行部分观测环境下的不确定动作模型的学习,因此本文的实验结果展示的是本系统在不同规划基准领域中学习错误率的比较.相对于其他相关方法,本文方法的先进性仅在理论上作初步的讨论.与以往工作不同,本文是在部分观测环境下学习不确定动作模型,其中,用于处理观测信息的两个假定和用于对不确定性结果进行去冗的聚类方法成为使这一目标可行的关键方法.以往工作,或者在部分观测环境下学习确定性的动作模型,或者在完全观测环境下学习概率动作效果,在部分观测下学习确定性动作模型的典型方法有 CBFS(constraints-based formulas selecting)[10]和 SLAF(simultaneous learning and filtering)[4].这两种方法都难以用于我们的学习目标.具体原因在于动作的不确定性使得 CBSF 中的约束具有柔性,因此难以直接使用.而 SLAF 尽管声称可扩展到不确定领域,但其在不确定环境中的推理非常复杂.另一方面,以往不确定动作模型的学习或者是有监督的[26],或者假设环境是完全可观测的[16].这些对于我们的学习目标都不现实.正如前面

一节所讨论的学习样例的数量,我们需要大量的数据来进行学习,如果要求有监督,那么工作量就太大了.同样,完全可观测环境也大量增加了实际成本开销.更进一步地,在动作模型的描述语言方面,本文方法也具有优越性.以往工作要么基于 STRIPS,要么基于各自的描述形式,而本文方法是基于在概率规划比赛中使用的标准描述语言 PPDDL.

图 3 给出了在观测率λ=0.9 时本文系统在 IPPC 2008(FOP)领域的学习错误率.其中,效果最好的来自于 Blocksworld 领域,其错误率大概为 0.231.初步的分析是,不同领域的学习错误率之间的差异主要来自于动作模型本身复杂性的影响.至于更深层的原因需要更进一步的领域分析.由于学习到的动作模型具有不确定性,因此学习的错误率要比类似情况下的确定性动作模型学习[4]的错误率高一些.图 4 给出了对应的学习时间.

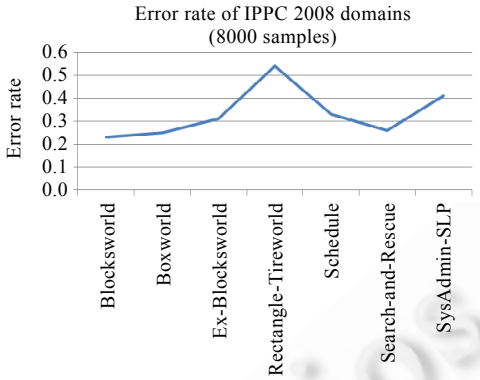

Fig.3 Error rate of IPPC 2008 (FOP) domains

图 3 IPPC 2008 (FOP)领域的学习错误率

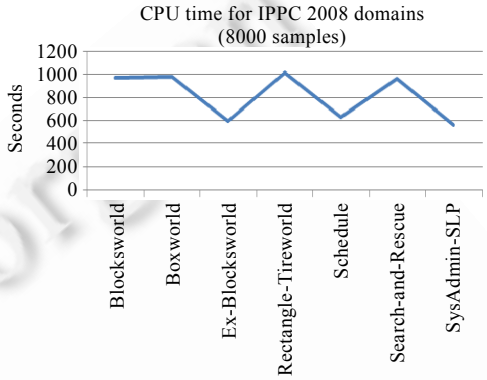

Fig.4 CPU time for IPPC 2008 (FOP) domains

图 4 IPPC 2008 (FOP)领域的学习时间

由于篇幅所限,本文不一一列举出所有的学习结果,仅用一个例子来加以说明.

例 3:当观测率为 0.95 时,本文系统在 Blocksworld 领域学习到的动作 pick-up 的模型如下:

```
(:action pick-up
    :parameters (?x ?y)
    :precondition (and (empty-hand) (clear ?x) (on ?x ?y))
    :effect
        (probabilistic
            0.63 (and (not (empty-hand)) (clear ?y) (not (on ?x ?y)))
            0.37 (and (clear ?y) (on-table ?x)(not (on ?x ?y))))
)
```

与例 1 所示的标准模型对比可以看到,动作的前提都学到了,动作的效果也学到了两组,不过,其中有文字丢失.在第 1 组效果中没有学习到文字(holding ?x).此外,经过统计得到的效果概率分布也与标准模型有稍微差别.尽管这一学习结果与标准模型不完全相同,但是可以看到,它们的相似度是比较高的,效果的组数和大部分的文字是完全相同的.

此外,观测率和最小覆盖样例数的设置对学习结果也有影响.在实验中可以发现,当观测率<0.8 时,学习效果普遍很差,大多数领域动作模型的错误率非常高;而当观测率>0.95 时,大多数领域能够学习到基本正确的动作模型.我们认为,这些范围的观测率反映的是比较极端的情况,即不可观测和完全可观测两种情形.而当观测率约为 0.9 时,不同领域的错误率反映的是正常的领域差异(如图 3 所示),即由领域本身的复杂性所主导的.另一方面,最小覆盖样例数用来筛选初步得到的动作效果模式,在实验中可以发现:当最小覆盖样例数<4 时,筛选之后仍然得到大量效果;而当逐步提高最小覆盖样例数时,筛选结果会有所改善.但是最小覆盖样例数也不能太大,因为太多的效果模式被筛除会导致最后的动作效果很少,从而导致很高的错误率.在本文的实验中我们发现,最

小覆盖样例数选择在 8 左右是比较适宜的.

5 结束语

本文给出了一种在部分观测环境下学习不确定动作模型的方法.学习过程分为 3 个步骤:首先,根据最小修改假定和均等机会假定来计算命题在状态中成立的概率;然后,将在相邻状态中概率变化明显的命题抽取成效果模式,根据效果模式来抽取前提;最后,对效果模式进行聚类以去除冗余.其中,用于处理观测信息的两个假定和用于对不确定性结果进行去冗的聚类方法是使得在部分观测下学习不确定动作模型可行的关键方法.学习过程可以在假定对转移系统一无所知的情形下进行,输入只有动作-观测序列.本文的工作将不确定动作模型学习技术引入到部分观测环境.在基准规划领域上的实验结果表明,在部分观测环境下学习不确定动作模型是可行的.同时,本文的实验特意选取了概率领域进行测试,证明了本文方法对于概率规划领域也是可行的.

但是,本文方法仍然存在一定的局限性,将在未来的工作中加以改进:首先,在本文中,观测被假定为准确无误的,然而在很多现实环境中观测是会出错的,因此,可进一步研究在有噪音的观测环境下如何正确地学习不确定的动作模型;其次,实验发现,无论参数如何调整,有些规划领域(例如 rectangle-tireworld)的学习错误率总是很高,这可能是领域的特殊性所决定的.因此,需要进一步分析领域特性与学习效果之间的关系,即,研究哪些领域是比较难学习的;然后,算法参数的调整是采用数据拟合的方式进行的,我们未来的工作期望能够做到自动进行参数调整.其中,参数之间的关联关系可能会是一个非常有价值的研究内容;最后,我们的最终目标是能够在真实世界的具体应用领域中利用动作模型学习技术帮助形成规划领域描述,以推动智能规划技术的应用和发展.

致谢 感谢吴康恒博士和卓汉逵博士参与本文的讨论.

References:

[1] Fikes RE, Nilsson NJ. STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence, 1971,2(3-4):189−208. [doi: 10.1016/0004-3702(71)90010-5]

[2] Gelfond M, Lifschitz V. Action languages. Electronic Articles in Computer and Information Science, 1998,3(16):193−210.

[3] Fox M, Long D. PDDL2.1: An extension to PDDL for expressing temporal planning domains. Journal of Artificial Intelligence Research (JAIR), 2003,20:61−124. http://jair.org/papers/paper1129.html

[4] Amir E, Chang A. Learning partially observable deterministic action models. Journal of Artificial Intelligence Research (JAIR), 2008,33:349−402. http://jair.org/papers/paper2575.html

[5] Hoffmann J, Brafman RI. Contingent planning via heuristic forward search with implicit belief states. In: Proc. of the ICAPS. 2005. 71−80. http://www.aaai.org/Library/ICAPS/2005/icaps05-008.php

[6] Bryce D, Kambhampati S, Smith DE. Planning graph heuristics for belief space search. Journal of Artificial Intelligence Research (JAIR), 2006,26:35−99. http://jair.org/papers/paper1869.html

[7] Cimatti A, Roveri M, Bertoli P. Conformant planning via symbolic model checking and heuristic search. Artificial Intelligence, 2004,159(1-2):127−206. [doi: 10.1016/j.artint.2004.05.003]

[8] Kaelbling LP, Littman ML, Cassandra AR. Planning and acting in partially observable stochastic domains. Artificial Intelligence, 1998,101(1-2):99−134. [doi: 10.1016/S0004-3702(98)00023-X]

[9] Bonet B, Geffner H. Planning under partial observability by classical replanning: Theory and experiments. In: Proc. of the 22nd Int'l Joint Conf. on Artificial Intelligence. 2011. 1936−1941. http://ijcai.org/papers11/Papers/IJCAI11-324.pdf

[10] Yang Q, Wu KH, Jiang YF. Learning action models from plan examples using weighted MAX-SAT. Artificial Intelligence, 2007, 71(2-3):107−143.

[11] Schmill MD, Oates T, Cohen PR. Learning planning operators in real-world, partially observable environments. In: Chien S, Kambhampati S, Knoblock CA, eds. Proc. of the 5th Int'l Conf. on Artificial Intelligence Planning Systems (AIPS 2000). AAAI Press, 2000. 246−253. http://www.aaai.org/Library/AIPS/2000/aips00-026.php

[12] Holmes MP, Jr. Isbell CL. Schema learning: Experience-Based construction of predictive action models. In: Proc. of the 17th Advances in Neural Information Processing Systems (NIPS 2004). MIT Press, 2004. 585−592. http://books.nips.cc/papers/files/nips17/NIPS2004_0629.pdf

[13] Zhuo HH, Yang Q, Hu DH, Li L. Learning complex action models with quantifiers and logical implications. Artificial Intelligence, 2010,174(18):1540−1569. [doi: 10.1016/j.artint.2010.09.007]

[14] Chrisman L. Abstract probabilistic modeling of action. In: Proc. of the 1st Int'l Conf. on Artificial Intelligence Planning Systems (AIPS'92). AAAI Press, 1992. 28−36. http://dl.acm.org/citation.cfm?id=139492.139496

[15] Oates T, Cohen PR. Searching for planning operators with context-dependent and probabilistic effects. In: Clancey WJ, Weld DS, eds. Proc. of the 13th National Conf. on Artificial Intelligence (AAAI'96). AAAI Press, 1996. 865−868. http://www.aaai.org/Library/AAAI/1996/aaai96-128.php

[16] Pasula HM, Zettlemoyer LS, Kaelbling LP. Learning symbolic models of stochastic domains. Journal of Artificial Intelligence Research (JAIR), 2007,29:309−352. http://jair.org/papers/paper2113.html

[17] Hajishirzi H, Amir E. Reasoning about deterministic action sequences with probabilistic priors. In: Lin F, Sattler U, Truszczynski M, eds. Proc. of the 12th Int'l Conf. on the Principles of Knowledge Representation and Reasoning (KR 2010). AAAI Press, 2010. 456−464. http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1406

[18] Rao DN, Jiang ZH, Jiang YF, Wu KH. Learning non-deterministic action models for Web services from WSBPEL programs. Journal of Computer Research and Development, 2010,47(3):445−454 (in Chinese with English abstract).

[19] Rao DN, Jiang ZH, Jiang YF, Zhu HQ. Further research on observation reduction in non-deterministic planning. Ruan Jian Xue Bao/Journal of Software, 2009,20(5):1254−1268 (in Chinese with English abstract). http://www.Jos.org.cn/1000-9825/3453.htm [doi: 10.3724/SP.J.1001.2009.03453]

[20] Sablon G, Bruynooghe M. Using the event calculus to integrate planning and learning in an intelligent autonomous agent. In: Bäckström C, Sandewall E, eds. Proc. of the Workshop on Planning, Current Trends in AI Planning. AAAI Press, 2000. 254−265. http://www.cs.kuleuven.ac.be/cgi-bin-dtai/publ_info.pl?id=18376

[21] Wang XM. Learning by observation and practice: An incremental approach for planning operator acquisition. In: Prieditis A, Russell SJ, eds. Proc. of the 12th Int'l Conf. on Machine Learning (ICML'95). Morgan Kaufmann Publishers, 1995. 549−557. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.36.7719

[22] Balac N, Gaines DM, Fisher D. Learning action models for navigation in noisy environments. In: Proc. of the Int'l Conf. on Machine Learning Workshop on Learning of Spatial Knowledge. Palo Alto, 2000. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.34.909&rep=rep1&type=pdf

[23] Younes HLS, Littman ML. PPDDL1.0: An extension to PDDL for expressing planning domains with probabilistic effects. 2003. http://www.cs.cmu.edu/lorens/papers/ppddl.pdf

[24] Rao DN, Jiang ZH, Jiang YF, Liu Q. Learning first-order rules for derived predicates from plan examples. Chinese Journal of Computers, 2010,33(2):251−266 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2010.00251]

[25] Bhattacharya I, Getoor L. Entity Resolution in Graphs, Chapter in Mining Graph Data. New York: John Wiley & Sons, 2006.

[26] Hajishirzi H, Amir E. Stochastic filtering in a probabilistic action model. In: Proc. of the 22nd AAAI Conf. on Artificial Intelligence. 2007. 999−1006. http://www.aaai.org/Library/AAAI/2007/aaai07-159.php

附中文参考文献:

[18] 饶东宁,蒋志华,姜云飞,吴康恒.从 WSBPEL 程序中学习 Web 服务的不确定动作模型.计算机研究与发展,2010,47(3):445−454.

[19] 饶东宁,蒋志华,姜云飞,朱慧泉.对不确定规划中观测约简的进一步研究.软件学报,2009,20(5):1254−1268. http://www.jos.org.cn/1000-9825/3453.htm [doi: 10.3724/SP.J.1001.2009.03453]

[24] 饶东宁,蒋志华,姜云飞,刘强.从规划解中学习一阶派生谓词规则.计算机学报,2010,33(2):251−266. [doi: 10.3724/SP.J.1016.2010.00251]

饶东宁(1977－),男,广东兴宁人,博士,副教授,CCF 会员,主要研究领域为智能规划,图论.

E-mail: raodn@gdut.edu.cn

姜云飞(1945－),男,教授,博士生导师,主要研究领域为定理机器证明,智能诊断,智能规划.

E-mail: issjyf@mail.sysu.edu.cn

蒋志华(1978－),女,博士,副教授,主要研究领域为智能规划.

E-mail: tjiangzhh@jnu.edu.cn