

基于众包模式的开放式规划问题研究

高 洁^{1,2}, 卓汉逵¹, 刘亚松², 李 磊¹

(1. 中山大学信息科学与技术学院, 广东广州 510275; 2. 吉林大学珠海学院, 广东珠海 519041)

摘 要: 在开放世界中求解智能规划问题往往是比较困难的, 这是由于在开放世界中, 某些对象可能是未知的, 因而在搜索规划解时需要考虑不同的可能性. 一种解决的方法是使用传感器观察未知的对象, 而该方法使用的前提是传感器能够保证获取规划所需的所有信息. 与以往工作不同的是, 本文考虑利用外部人士 (Crowd) 求解规划问题. 假设存在一些外部人士可以为开放世界中某个规划问题提供必要的信息, 然而在实际情况下, 某些外部人士提供的信息可能是具有欺骗性的, 如何使用此类信息求解规划问题是本文关注的重点. 针对此类问题, 本文提出了一个新颖的求解方法, 首先获取一个求解开放世界下的规划问题所需的带有变量的命题公式集合, 然后根据外部人士对命题公式的标注估计出变量所取的值, 从而将开放世界中的规划问题转化为一般的规划问题求解. 最后通过实验验证了该算法的有效性.

关键词: 智能规划; 众包; 开放世界

中图分类号: TP301

文献标识码: A

文章编号: 0372-2112 (2016)08-2025-08

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2016.08.036

Research on Crowdsourced Open Planning

GAO Jie^{1,2}, ZHUO Han-kui¹, LIU Ya-song², LI Lei¹

(1. School of Information Science and Technology, Sun Yat-Sen University, Guangzhou, Guangdong 510275, China;

2. Zhuhai College, Jilin University, Zhuhai, Guangdong 519041, China)

Abstract: Plan synthesis in an open world is challenging, since some objects in an open world might be unknown, then we need to consider various scenarios before planning. One way to solve this problem is to employ sensors to observe unknown objects, assuming the sensors are capable of correctly capturing all information needed for planning. Different with previous work, we turn to the crowd for help before doing planning. We assume there are abundant annotators available to provide information needed before planning, however there is possibly a substantial amount of discrepancy from the crowd in practice. It is thus challenging to solve the planning problem with possibly noisy information provided by the crowd. We propose a novel approach with two phases. We first build a set of propositions with variables, and collect values from crowd for those propositions. We then estimate the actual values of variables and transform the problem in an open world into a normal planning problem and solve it. Finally, we empirically exhibit the effectiveness of our approach.

Key words: automated planning; crowdsource; open world

1 引言

智能规划^[1,2]是人工智能研究领域近年来发展起来的一个热门分支. 目前智能规划研究主要集中在封闭世界假设的前提下进行, 这是由于在开放世界中求解规划问题往往是比较困难的. 在开放世界中, 某些对象可能是未知的, 因而在搜索规划解时需要考虑未知对象的各种可能信息. 一种解决的方法是在规划器上安装传感器, 在搜索规划解的过程中, 利用传感器所获

得的未知对象的相关信息再进行规划^[3,4]. 这种规划方法使用的前提是假设传感器可以感知规划所需的所有信息. 然而在开放世界中, 在规划任务给出之前, 很难确定所需传感器的个数与类型, 因而也无法感知规划所需的所有信息.

受众包模式^[5] (Crowdsource) 的启发, 我们尝试利用众人的智慧来解决规划问题, 假设存在一些外部人士能够在开放世界中为规划问题提供相应的信息. 根据文献[5]中的定义, 众包描述了一种新的工作模式,

即企业或机构利用一个较大的人际网络通过公开召集方式将某项工作分配出去. 该项工作可能是通过集体完成,也可能是由某个人单独完成. 目前已有一些成功的普遍型众包平台问世,例如亚马逊土耳其机器人^①;也有一些专业型的众包平台,例如呼叫中心^②. 众包模式已应用在各种问题中,例如口语翻译^[6],书写辨识^[7],故事生成^[8],以及机器学习中训练数据标示^[9]等方面.

目前在规划问题中使用众包模式还较为少见,但是在规划领域中已产生了一些与众包模式有关的应用,例如众包模式下行程规划问题^[10]. 一个行程由有序的动作序列组成,在整个过程中按照次序依次执行动作. 众人利用他们的知识与经验求解行程规划问题. 与求解众包模式下的行程规划问题不同的是,我们不要要求众人具备求解规划问题的能力,而仅仅要求众人能够提供求解规划问题所需的“资源”,相对而言这是比较容易的任务. 在规划过程中,开发一个可以集合众人智慧的规划系统是一件非常有意义的工作,同时也非常具有挑战性,这是由于我们需要考虑到众人意见的无序性与矛盾性.

在本文中,我们提出了 COP 算法 (Crowdsourced Open Planning),即利用众人的知识解决开放世界中的规划问题. 具体来说,COP 算法分成两个主要步骤,COP 算法首先获得一个求解开放世界中的规划问题所需的带变量的命题公式集合,然后由外部人士对命题公式进行标注,再根据标注值估计变量所取的值,从而将开放世界中的规划问题转化为一般的规划问题求解. 在假设开放世界中的规划问题可解的前提下,COP 算法将重复进行取值步骤,直到求得规划问题的规划解或超过设定的标注公式个数的上限.

2 相关工作

开放世界中的规划问题^[11]是指在初始状态,目标状态等信息不完整情况下的规划问题. 利用封闭世界中的规划器去求解开放世界中的规划问题是相当困难的事,这是由于假设世界是封闭的,或者假设可以获取所有缺失的信息以将开放世界转变为封闭世界,都是错误的.

Babaian 等人^[12]提出了 PSIPLAN 知识表示体系,可以用于有效地在开放世界中表示规划域. 已有的规划系统^[3,13]通过检测动作执行状态并利用传感器的感知能力找到不一致的执行状态,再进行规划修改或再规划来解决开放世界中的规划问题. Nareyek 等人^[14]使用结构性的限制可满足条件来构造规划结构以处理开放式规划问题. Talamadupula^[15]等人表明了团队问题中处理开放世界问题的必要性,并给出了条件目标的定义.

这些系统都依赖于传感器的感知能力,与其不同之处在于,我们旨在利用众人的知识来确定初始状态或目标状态中的不确定信息.

为了获得规划所需的条件,我们采用众包模式求解开放世界中的规划问题. 众包模式首先被用于在数据挖掘时提高训练数据标签的正确性^[16]. Raykar^[17]等人提出了一种模型,其中反映工人准确性的参数依赖于正确答案. 虽然众包模式已被大量应用于不同领域,但是在规划领域中的应用还是非常少见的. 作为第一个探索众包模式在智能规划领域中应用的工作,Talamadupula 等人^[18]提出了一种通用框架,奠定了众包模式规划问题的基础. Manikonda^[19]等人开发了一套旅行规划生成系统 AI-MIX,其思想是使用自动检测的方法提高众包模式所生成的行程规划解的质量. 卓汉逵等人^[20]给出了一种称为 CAMA 的动作模型学习算法,用于利用众包模式获取动作模型的前提条件与动作效果. 本文提出了在开放世界中,当初始状态与目标状态中包含未知信息时,一种基于众包模式的规划问题求解方法.

3 开放式规划问题

一般的规划问题定义为一个三元组 (s_0, g, U) ,这里 s_0 表示初始状态, g 表示目标状态,都由一个命题公式集合组成. U 表示一个 STRIPS 动作模型集,每个动作模型由一个四元组 $(a, \text{PRE}, \text{ADD}, \text{DEL})$ 构成,这里 a 表示一个动作模式,包括动作名称及参数信息(零个或者多个). PRE 是动作 a 的前提条件列表,指明了实施动作 a 之前应满足的条件. ADD 表示增加效果列表,指明了实施动作 a 以后增加的新效果. DEL 表示删除效果列表,指明了实施动作 a 以后删除的效果. 这里所涉及的动作模型为 STRIPS 模型. 规划问题的规划解即为从初始状态到目标状态所需执行的动作序列.

当命题的参数列表中含有变量时. 我们称之为开放式命题. 例如,“on($A, ?x$)”是一个开放式命题,这里 $?x$ 表示命题 on 的参数列表中包含的一个变量. 我们将开放世界中的规划问题称之为开放式规划问题,并将其定义为一个四元组 $(\tilde{s}_0, \tilde{g}, O, U)$,这里 \tilde{s}_0 由一个开放式命题集合组成,称之为开放式初始状态. 同理, \tilde{g} 称为一个开放式目标状态,也由一个开放式命题集合组成. \tilde{s}_0 和 \tilde{g} 所包含的变量集合记为 V . V 中变量可以取到的所有值构成的集合用 O 表示,且假设 O 可以从历史数据中获得. U 表示一个 STRIPS 动作模型集合. 一个开放式规划问题的规划解是由一个动作序列,以及 \tilde{s}_0 和 \tilde{g} 中变量赋值表所构成. 变量赋值表 θ 定义为 $\theta = \{(v, o)$

① www.mturk.com

② www.liveops.com

$\{v \in V \wedge o \in O\}$. 表 1 给出了开放式规划问题的一个例子. 这里前缀“?”表示一个变量. 输入由三部分组成, 即开放式初始状态, 开放式目标状态以及动作模型. 输入由两部分组成, 即一个变量的赋值表和一个规划解. 在输出结果中, “ $?x = C$ ”意味着变量“ x ”被赋值为“ C ”, 同样“ $?y = B$ ”意味着变量“ y ”被赋值为“ B ”.

表 1 一个开放式规划问题的例子

| | |
|-----------------------|---|
| 开放式初始状态 \tilde{s}_0 | $\text{on}(?x, A), \text{clear}(?x), \text{ontable}(A)$ |
| 开放式目标状态 \tilde{g} | $\text{on}(C, ?y), \text{clear}(C), \text{ontable}(?y), \text{clear}(A), \text{ontable}(A), \text{handempty}$ |
| 变量取值集合 O | A, B, C, D |
| 动作模型集合 U | Action schema: pickup(?x) PRE: handempty, clear(?x), ontable(?x) ADD: holding(?x) DEL: handempty, clear(?x), ontable(?x) |
| | Action schema: unstack(?x, ?y) PRE: handempty, clear(?x), on(?x, ?y) ADD: holding(?x), clear(?y) DEL: handempty, clear(?x), on(?x, ?y) |
| | Action schema: putdown(?x) PRE: holding(?x) ADD: handempty, clear(?x), ontable(?x) DEL: holding(?x) |
| 一个规划解 | Action schema: stack(?x, ?y) PRE: holding(?x), clear(?y) ADD: handempty, clear(?x), on(?x, ?y) DEL: holding(?x), clear(?y) |
| | 变量赋值表为($?x = C, ?y = B$) 动作序列为($\text{unstack}(C, A), \text{stack}(C, B)$) |

4 COP 算法步骤

我们在算法 1 中给出了 COP 算法的整体框架, 下面的小节中我们将详细阐述 COP 算法的每个步骤.

4.1 构建带变量的命题公式组

在 COP 算法的第一步中, 我们根据输入的动作模型集合 U , 开放式初始状态信息 \tilde{s}_0 和开放式目标状态信息 \tilde{g} , 构建一个带变量的命题公式集合族. 首先从 \tilde{g} 着手, 搜索动作正效果与 \tilde{g} 所匹配的动作, 然后从 \tilde{g} 中删去该动作的正效果, 再加入该动作的前提条件, 构成一个中间状态. 接下来判断 \tilde{s}_0 是否与中间状态相匹配, 并从可以匹配的中间状态集中选取带有变量的命题. 从带有相同变量的命题中, 任意选取其中一个命题, 构成一个带变量的命题公式集合, 作为该步骤的返回值. 我们将这一步骤用 Buildprop 子算法表示, 具体步骤见算法 2. 算法 3 中的命题匹配算法用于判断两个命题是否匹配. 如果集合 S_1 中的每个命题与集合 S_2 中的某个命题匹配, 则称集合 S_1 可与集合 S_2 相匹配. 例如, 假设开放式初始状态 \tilde{s}_0 由命题

$\text{on}(?x, A), \text{clear}(?x), \text{ontable}(A)$

组成. 一个中间状态集合 G 由命题 $\text{ontable}(?y), \text{ontable}(A), \text{clear}(?y), \text{on}(C, A), \text{clear}(C), \text{handempty}$ 组成. 在匹配过程中, \tilde{s}_0 中的命题 $\text{ontable}(A)$ 与 G 中的同一命题相匹配; \tilde{s}_0 中的命题 $\text{clear}(?x)$ 与命题 $\text{on}(?x, A)$ 与 G 中的命题 $\text{on}(C, A)$ 相匹配. 因此开放式初始条件 \tilde{s}_0 与中间状态集合 G 匹配. 中间状态集合 G 中的命题 $\text{clear}(?y)$ 相匹配; \tilde{s}_0 中的命题 $\text{on}(?x, A)$ 与 G 中的命题 $\text{on}(C, A)$ 相匹配. 因此开放式初始条件 \tilde{s}_0 与中间状态集合 G 匹配.

算法 1 COP 算法框架

输入: 开放式规划问题 $P = \langle \tilde{s}_0, \tilde{g}, O, U \rangle$

输出: 规划解 sol 以及变量的赋值表 θ

- 步骤: 1. 利用 Buildprop 子算法, 从目标状态 \tilde{g} 出发, 获取与初始状态 \tilde{s}_0 相匹配的带变量的命题公式集合族 $T_i = \text{buildprop}(\tilde{s}_0, \tilde{g}, U), i = 1 \dots n$.
2. 根据 T_i , 构建待标注公式集合 Y_i .
3. 由外部人士 (Crowds) 标注公式集合 Y_i , 并将所获取的真假值表, 记为 K_i .
4. 根据 K_i , 计算 Y_i 中各个公式的值
- $$\hat{Z} = \text{EstimateLables}(K_i, Y_i, P).$$
5. 在 \hat{Z} 的基础上, 将规划问题更新为新的规划问题 \bar{P} , 并获取一个变量赋值表 θ .
6. 利用常规规划器求解新的规划问题 \bar{P} .
7. 如果规划问题 \bar{P} 可求解, 则返回一个规划解 sol 以及变量的赋值表 θ .
8. 否则返回第 1 步, 并取下一个带变量的命题公式集合 T_{i+1} , 直到 $i = n + 1$.

算法 2 Buildprop 子算法

输入: 开放式规划问题 $P = \langle \tilde{s}_0, \tilde{g}, U \rangle$

输出: 带变量的命题公式集合族 $T_i, i = 1 \dots n$

- 步骤: 1. 令 $G = \tilde{g}$;
2. 对于 S 中每一个动作 $a \in S$, 如果动作 a 的正效果与中间状态集 G 相匹配, 则更新中间状态集 $G = G - a$ 的正效果 + a 的正效果.
3. 如果初始状态 \tilde{s}_0 与中间状态集 G 相匹配, 则取出中间状态集 G 中带有变量的公式.
4. 在第 3 步所获得的公式中, 如果部分公式含有的变量完全相同, 则任选其中一个公式, 构建带变量的命题公式集合 T_i .
5. 返回第 2 步, 直到无可匹配的中间状态集为止.
6. 输出 $T_i, i = 1, \dots, n$, 这里 n 表示可以匹配的中间状态集的个数.

算法 3 命题匹配算法

输入: 两个命题 $P_1(X_1, \dots, X_n), P_2(Y_1, \dots, Y_n)$

输出: True 或 False

1. $i = 1$;
2. 如果 $P_1 \neq P_2$, 则返回 False. 否则, for $1 \leq i \leq n$,
3. 如果 $\text{is_constant}(X_i) \wedge \text{is_constant}(Y_i) \wedge (X_i \neq Y_i)$, 则返回 False.
4. 如果 $\text{is_variable}(X_i) \wedge \text{is_constant}(Y_i)$, 则返回 True 且 $X_i = Y_i$.
5. 如果 $\text{is_constant}(X_i) \wedge \text{is_variable}(Y_i)$, 则返回 True 且 $Y_i = X_i$.
6. 如果 $\text{is_variable}(X_i) \wedge \text{is_variable}(Y_i)$, 则返回 True 且将变量 Y_i 更名为 X_i . 否则返回假.

4.2 根据 T_i , 构建待标注公式集合 Y_i

在 COP 算法的第二步, 我们需获取一个待标注的命题公式集合. 在第 i 步所获得的带变量的命题公式集合 T_i 中, 将命题公式中的变量带入 O 中可以取到的值, 从而获得完全例化的命题公式以供外部人士标注. 当两个命题公式包含相同变量时, 为达到标注公式个数极小的目的, 我们随机选择其中的一个命题公式进行例化. 最后将获得的所有完全例化的命题公式集合记为 Y_i , 并由外部人士进行标注. 下面, 我们用一个例子从直观上解释一下该步骤的主要思想. 依然用表 1 中的例子来解释. 应用以上步骤以后, 我们获得了一个可以与初始状态相匹配的中间状态, 由命题公式

$$\begin{aligned} &\text{ontable}(Y), \text{ontable}(A), \text{clear}(Y), \\ &\text{on}(C, A), \text{clear}(C), \text{handempty} \end{aligned}$$

组成, 选取其中带有变量的命题公式 $\text{ontable}(?y)$ 与 $\text{clear}(?y)$. 由于这两个公式具有相同的变量, 为达到标注公式个数极小化的目的, 我们随机选择其中一个公式 $\text{ontable}(?y)$ 作为待标注公式. 假设包含所有可能物体的集合 $O = \{A, B, C, D\}$, 除去已在中间状态中出现的物体 A, C 后, 变量 $?y$ 可以取值 B 或 D . 因而得到待标注公式 $\text{ontable}(B)$ 与 $\text{ontable}(D)$.

4.3 由外部人士标注公式集合 Y_i , 并获取真值表 K_i

我们将需要标注的命题公式映射为一个调查表, 并用相同的调查表模板来表示每一个谓词. 例如, 我们将命题公式“ $\text{ontable}(B)$ ”映射为

“Is the block B on the table?”.

一个标注者只可以选择“ Yes ”或“ No ”其中之一作为回复. 我们付给标注者 10 分钱作为对每个命题的报酬. 每个标注者只能标注每个命题公式一次. 每个调查表由二十个标注者分别给出标注. 冗余度用于降低标注者给予错误或恶意回复的可能性. 另外, 我们也可以利用命题之间的互斥约束减少调查表的数量, 例如 $\text{clear}(A)$ 和 $\text{on}(B, A)$ 是彼此互斥的, 我们只需取其中的一个调查表用于标注. 最后获得一个待标注公式的真值表, 记为 K_i .

4.4 计算 Y_i 中各个公式的值

假设存在 R 个标注者与 N 个标注为 1 或 0 的公式, 其真实标注记作 $Z = (z_1, z_2, \dots, z_N)^T$, 这里 $z_i \in \{0, 1\}$. 公式标注方式可以用一个二分图来表示, 这里二分图

的一条边 (i, j) 表示第 i 个公式由第 j 个工人标注, 标注结果可以用矩阵 $Y \in \{0, 1\}^{N \times R}$ 表示. 当给定公式的标注值 Y 时, 求 N 个公式的真实标注 Z 的一个最优估计值 $\hat{Z} = (\hat{z}_1, \hat{z}_2, \dots, \hat{z}_N)^T$, 以最小化平均错误率

$$\frac{1}{N} \sum_{i \in \{1, 2, \dots, N\}} \text{prob}[\hat{z}_i \neq z_i].$$

构造训练数据集 $D = \{x_i, y_i^1, \dots, y_i^R\}_{i=1}^N$, 这里 $x_i \in X$ 表示一个实例 (通常用一个 d 维属性向量表示), 在这里 x_i 表示第 i 个命题公式, y_i^j 表示第 j 个标注者所给出的标注 (0 或 1). 我们分别利用正负例子对外部注解者的可信度建模. 对某一公式而言, 若为真, 则对于第 j 个标注者的正确正比率 (True positive rate) TP^j 定义为标注者标注该公式为 1 的概率, 即

$$TP^j = p(y_i^j = 1 | z_i = 1).$$

另一方面, 若为假, 则对于第 j 个标注者的正确负比率 (True negative rate) TN^j 定义为标注者标注该公式为 0 的概率, 即

$$TN^j = p(y_i^j = 0 | z_i = 0).$$

利用线性判别函数, 正例的概率可以用一个 logistic 函数来表示, 即

$$p(y = 1 | x, \omega) = \sigma(\omega^T x),$$

这里 $x, \omega \in R^d$, 且 $\sigma(z) = \frac{1}{1 + e^{-z}}$.

这里需估计参数 ω 以及正确正比率 $P = \langle TP^1, \dots, TP^R \rangle$ 和正确负比率 $N = \langle TN^1, \dots, TN^R \rangle$. 设 $\theta = \{\omega, P, N\}$, 训练数据集 D 的概率可以定义为

$$p(D | \theta) = \prod_{i=1}^N p(y_i^1, \dots, y_i^R | x_i, \theta).$$

给定真实标注 y_i , 我们假设 $y_i^1, y_i^2, \dots, y_i^R$ 是彼此独立的, 从而可得

$$p(D | \theta) = \prod_{i=1}^N [a_i p_i + b_i (1 - p_i)],$$

这里

$$p_i = \sigma(\omega^T x_i),$$

$$a_i = \prod_{j=1}^R [TP^j]^{y_i^j} [1 - TP^j]^{1 - y_i^j},$$

$$b_i = \prod_{j=1}^R [TN^j]^{1 - y_i^j} [1 - TN^j]^{y_i^j}.$$

通过最大化对数似然函数, 应用 EM 算法^[25] 估计参数 θ , 即

$$\hat{\theta} = \arg \max_{\theta} \{\ln p(D | \theta)\},$$

这里 $\hat{\theta} = \{\hat{\omega}, \hat{P}, \hat{N}\}$. 获得 $\hat{\theta}$ 后, 再根据概率 $p(y_i = 1 | y_i^1, \dots, y_i^R, x_i, \hat{\theta})$ 估计实例 x_i 的标注. 我们将未知的真实标注 z_i 作为缺失数据, 则有

$$\ln p(D, Z | \theta) = \sum_{i=1}^N z_i \ln p_i a_i + (1 - z_i) \ln (1 - p_i) b_i.$$

可以得到期望值如下:

$$E\{\ln p(D, Z|\theta)\} = \sum_{i=1}^N \mu_i \ln p_i a_i + (1 - \mu_i) \ln(1 - p_i) b_i.$$

这里 $\mu_i = p(z_i = 1 | y_i^1, \dots, y_i^R, x_i, \theta)$. 在本文中,我们选择信任某一位标注者超过其他标注者的程度. 达到这一目的的方法之一是对 P 和 N 设置先验概率. 我们使用 β 分布作为 P 和 N 的先验概率,即 $\beta(TP^i | a_1, a_2)$ 和 $\beta(TN^i | b_1, b_2)$. 值得注意的是,我们根据均值 μ 和方差 σ^2 来确定先验概率. β 分布的先验概率的均值与方差可以表示为

$$\mu = a_1 / (a_1 + a_2)$$

$$\sigma^2 = a_1 a_2 / ((a_1 + a_2)^2 (a_1 + a_2 + 1)),$$

进而可得

$$a_1 = (-\mu^3 + \mu^2 - \mu\sigma^2) / \sigma^2$$

且

$$a_2 = a_1(1 - \mu) / \mu.$$

类似地,可得 b_1 与 b_2 的表达式.

在本文中,每个实例实际上是一个公式,并不存在属性向量 x_i ,我们希望由多个注释者提供的标注获取公式真实值的一个估计值. 因此我们用 $p = \text{prob}[z = 1]$ 来估计正类的普遍性,并假设普遍性的 β 先验概率为 $\beta(P | a_1, a_2)$. EM 算法可以简化为以下步骤:

(1) 根据投票原理,设定 $\mu_i = \frac{1}{N} \sum_{j=1}^R y_i^j$.

(2) 给定 μ_i , 正确正比率与正确负比率可以估计为:

$$TP^j = \frac{a_1^j - 1 + \sum_{i=1}^N \mu_i y_i^j}{a_1^j + a_2^j - 2 + \sum_{i=1}^N \mu_i}$$

$$TN^j = \frac{b_1^j - 1 + \sum_{i=1}^N (1 - \mu_i)(1 - y_i^j)}{b_1^j + b_2^j - 2 + \sum_{i=1}^N (1 - \mu_i)}$$

正例的概率值可以估计为: $p = \frac{a_1 - 1 + \sum_{i=1}^N \mu_i}{a_1 + a_2 - 2 + N}$.

(3) 更新 μ_i 为 $\frac{a_i p}{a_i p + b_i(1 - p)}$,

这里 $a_i = \prod_{j=1}^R [TP^j]^{y_i^j} [1 - TP^j]^{1 - y_i^j}$,

且 $b_i = \prod_{j=1}^R [TN^j]^{1 - y_i^j} [1 - TN^j]^{y_i^j}$.

正例的概率值可以估计为: $p = \frac{a_1 - 1 + \sum_{i=1}^N \mu_i}{a_1 + a_2 - 2 + N}$. 第(2)

步和第(3)步迭代进行,直到收敛为止. 一旦 EM 算法停止,我们使用参数 μ_i 并设定一个阈值,用于估计 z_i 的真实标注,即如果 $\mu_i > 0.5$,则 z_i 的值为 1,否则为 0.

4.5 更新并求解规划问题

在算法 1 的第 5 步,我们利用第 4 步中所获得的 \hat{Z} ,将开放式的规划问题 P 更新为一个封闭世界的规划问题 \bar{P} ,即将初始状态完整化. 首先,我们选取 \hat{Z} 中标注为 1 ($z_i = 1$) 的公式,并将其转化为实例化的命题公式,从而将第 1 步所获得的可匹配的中间状态集合 T_i 更新为实例化的中间状态集合 \bar{T}_i . 与此同时,我们通过比较 T_i 与 \bar{T}_i ,可以计算出开放式初始状态 \bar{s}_0 与目标状态 \bar{g} 中的变量赋值 θ . 例如, T_i 中包含命题 $\text{on}(?x A)$ 且 \bar{T}_i 中包含命题 $\text{on}(B, A)$,则有 $?x = B$. 进一步,再利用变量赋值 θ 将开放式初始状态与目标状态实例化,从而将开放式的规划问题 P 更新为一个封闭世界的规划问题 \bar{P} . 利用一般的规划器对 \bar{P} 进行求解. 如果无解,则再回到第 2 步,并选取下一个可与初始状态匹配的中间状态集 T_{i+1} ,重复上面的步骤以求得一个规划解.

5 实验验证

5.1 实验数据与评价标准

在这一节中,我们利用国际规划大赛 (IPC) 中的 STRIPS 规划域 blocks world, depots 及 driverlog 来验证 COP 算法的效果. 我们按照以下两个步骤生成开放式规划问题. 我们首先利用规划问题生成器^①,生成 150 个封闭世界规划问题 \bar{P} . 接下来我们从 \bar{P} 的初始状态中按照比例 α 随机删去一些命题,再从目标状态与剩余的初始状态中按照相同比例 α 随机将部分物体换为变量. 我们将所生成的开放式规划问题记作 P . 在前面的算法中,我们将待标注的命题转化为用自然语言表示的调查表,即注释者可以仅仅根据他们的背景来回答问题,而不需要额外了解 AI 规划背景. 我们使用 20 个模拟注释者来进行仿真实验. 每个仿真注释者 j 对应正确正比率 TP^j 与正确负比率 TN^j ,其由 β 分布 $\beta(TP^j | a_1, a_2)$ 与 $\beta(TN^j | b_1, b_2)$ 生成,这里 a_1, a_2, b_1, b_2 是随机生成的参数. 仿真注释者根据他们的正确正比率 TP^j 与正确负比率 TN^j 来完成调查表.

我们按照以下方式来说明 COP 算法的精确度. 我们使用 FF 规划器来求解所有 150 个封闭世界规划问题 $\{\bar{P}\}$,并将规划解集合记为 SOL. 对于每个问题,我们将

^① 以上三个规划域的规划问题生成器的下载地址为: <http://rakaposhi.eas.asu.edu/planning-class.html/>

FF 规划器所求得解记录下来,即 $|SOL| = 150$. 我们再通过 COP 算法求解对应的开放式规划问题 $\{P\}$,从而得到 150 个解,记为 SOL' . 对于每个问题 \bar{P} 和对应的开放式规划问题 P ,如果它们的解 \bar{L} 与 L 所包含的动作名称相同,且动作所包含的物体(不包含那些由注释者所指定的)也是相同的,则认为这两个是相同的. 例如, $pickup(A), stack(A,B), unstack(C,D)$ 与 $pickup(K), stack(K,B), unstack(C,D)$ 可以被认为相同的解,这里 A, K 是由注释者所指定的. 我们因此定义 COP 的准确率如下:

$$e = \frac{\#IdenticalSolution(SOL, SOL')}{|SOL|}$$

这里 $\#IdenticalSolution(SOL, SOL')$ 表示将 SOL 与 SOL' 相比较,其中相同解的个数.

5.2 实验结果及分析

我们首先估计 COP 算法的准确率. 我们设定物体个数为 10 个,并使得变量个数的比率 α 从 0.1 变化到 0.5,我们运行 COP 算法以测试问题的准确性. 从图 2 中曲线可知,随着变量个数比率 α 的增加,三个测试领域中的求解准确率基本上呈逐渐降低的趋势,这与我们的直觉是一致的. 这是由于变量个数比率 α 越大,开放式规划问题中所含有的不确定信息越多. 另外如图 1 所示,当变量比率低于 0.3 时,求解准确率不低于 70%. 另外,我们希望获取随着物体个数的变化, COP 算法准确率的变化趋势. 设物体个数从 4 个变化到 13 个,随着变量个数比率 α 的变化,计算 COP 算法准确率的平均值,其结果如表 2~表 4 所示. M_{obj} 表示物体的个数. 这里 $M_{obj} \in [4, 7]$ 所对应的列表示当物体个数从 4 个变化到 7 个时,对应变量个数比率 α ,所求得的准确率的平均值,同理对于 $M_{obj} \in [8, 10]$ 与 $M_{obj} \in [11, 13]$ 所对应的列也表示相同的意思. 从以上表格中可知,在每个规划域中,随着未知变量比率的提高(从 0.1 提高到 0.5),COP 算法的精确率相应地减少. 这是因为未知变量数量的增加,使得带有未知变量的命题公式数量增加,这将导致未知信息量增加,从而使得算法 COP 的精确率降低. 这里我们设定外部标注公式总次数最大

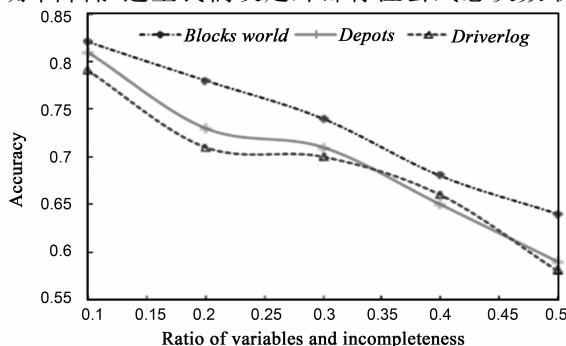


图1 COP算法准确率变化趋势

阈值 $N = 100$.

表 2 在 Blocks World 域中 COP 算法精确率平均值

| a | $M_{obj} \in [4, 7]$ | $M_{obj} \in [8, 10]$ | $M_{obj} \in [11, 13]$ |
|-----|----------------------|-----------------------|------------------------|
| 0.1 | $P = 0.96$ | $P = 0.92$ | $P = 0.84$ |
| 0.2 | $P = 0.92$ | $P = 0.88$ | $P = 0.80$ |
| 0.3 | $P = 0.86$ | $P = 0.82$ | $P = 0.80$ |
| 0.4 | $P = 0.82$ | $P = 0.76$ | $P = 0.72$ |
| 0.5 | $P = 0.80$ | $P = 0.74$ | $P = 0.70$ |

表 3 在 Depots 域中 COP 算法精确率的平均值

| a | $M_{obj} \in [4, 7]$ | $M_{obj} \in [8, 10]$ | $M_{obj} \in [11, 13]$ |
|-----|----------------------|-----------------------|------------------------|
| 0.1 | $P = 0.94$ | $P = 0.90$ | $P = 0.84$ |
| 0.2 | $P = 0.94$ | $P = 0.88$ | $P = 0.82$ |
| 0.3 | $P = 0.90$ | $P = 0.88$ | $P = 0.80$ |
| 0.4 | $P = 0.82$ | $P = 0.74$ | $P = 0.66$ |
| 0.5 | $P = 0.76$ | $P = 0.64$ | $P = 0.60$ |

表 4 在 Driverlog 域中 COP 算法精确率的平均值

| a | $M_{obj} \in [4, 7]$ | $M_{obj} \in [8, 10]$ | $M_{obj} \in [11, 13]$ |
|-----|----------------------|-----------------------|------------------------|
| 0.1 | $P = 0.90$ | $P = 0.82$ | $P = 0.80$ |
| 0.2 | $P = 0.88$ | $P = 0.80$ | $P = 0.80$ |
| 0.3 | $P = 0.82$ | $P = 0.74$ | $P = 0.72$ |
| 0.4 | $P = 0.80$ | $P = 0.72$ | $P = 0.68$ |
| 0.5 | $P = 0.72$ | $P = 0.64$ | $P = 0.58$ |

我们也测试了所需支付注释者劳务费的情况. 假设注释者劳务费仅仅取决于由 COP 算法所产生的待标注命题公式的数量,而不考虑标注不同类型的待标注命题公式的难度. 待标注命题公式的数量取决于物体个数与变量个数比率两个因素. 我们通过改变物体个数,并设定变量个数比率以计算所需标注命题公式的个数,结果如图 2~图 4 所示. 由图 2~图 4 中的图像可知,随着物体个数增多,无论变量个数比率如何,一般来说待标注命题公式的个数也逐渐增多. 这是因为物体个数越多,变量赋值方式越多,从而导致需标注命题个数越多. 同样地,变量个数比率越大,所有可能的变量组合方式也越多,也导致需标注命题个数越多.

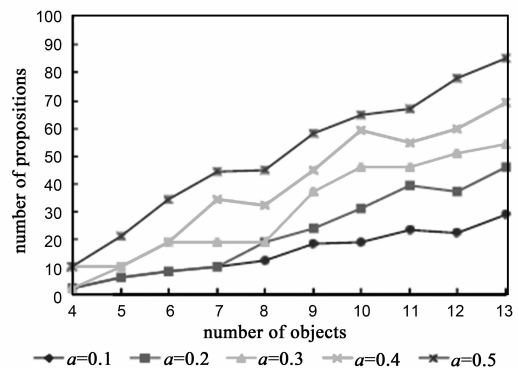


图2 Blocks world规划领域中开放式规划问题所需外部标注的公式总数

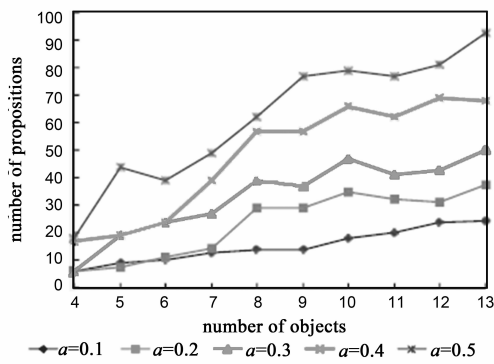


图3 Depots规划领域中开放式规划问题所需外部标注的公式总数

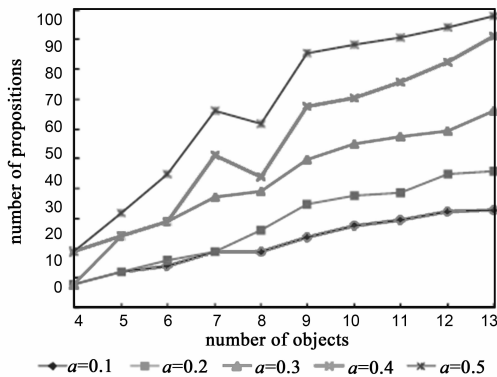


图4 Driverlog规划领域中开放式规划问题所需外部标注的公式总数

6 结论

目前智能规划研究主要集中在封闭世界假设的前提下,对于开放世界中的规划问题鲜有研究.目前开放世界中的规划问题一般采用了传感器获取未知信息或者在给定预先假设的前提下进行求解.在实际问题中,开放世界中的规划问题更加具有一般性,仍然是当前规划研究的难点.本文关注于采用众包模式解决开放世界中的规划问题.我们提出了基于众包模式的开放式规划算法 COP,有效地解决了初始状态与目标状态中带有变量的开放式规划问题,并利用三个改造后的规划域验证了算法的有效性.今后,我们将研究如何改善 COP 算法的性能,以期提高 COP 算法有效比例.

参考文献

- [1] 胡亮,解男男,等.基于智能规划的多步攻击场景识别算法[J].电子学报,2013,41(9):1753-1759.
Hu Liang,Xie Nan-nan,et al. A multi-stage attack scenario recognition algorithm based on intelligent planning[J]. Acta Electronica Sinica,2013,41(9):1753-1759. (in Chinese)
- [2] 王楨珍,武小悦,刘忠.一种基于智能规划的信息安全风

险过程建模方法[J].电子学报,2008,36(S1):76-80,70.

Wang Zhen-zhen, Wu Xiao-yue, Liu Zhong. A planning-based method of risk process modeling for information security[J]. Acta Electronica Sinica,2008,36(S1):76-80,70. (in Chinese)

- [3] Knight R, Rabideau G, et al. Casper: Space exploration through continuous planning [J]. Intelligent Systems, 2001,16(5):70-75.
- [4] Talamadupula K, Benton J, et al. Planning for human-robot teaming in open worlds [J]. ACM Transactions on Intelligent Systems and Technology, 2010,1(2):14-22.
- [5] Howe J. The rise of crowdsourcing [J]. Wired Magazine, 2006,14(6):1-4.
- [6] Liem B. An Iterative dual pathway structure for speech-to-text transcription [A]. Hartman B, Proceedings of the 3rd Workshop on Human Computation [C]. USA: AAAI, 2011. 1123-1131.
- [7] Ouyang T. Bootstrapping personal gesture shortcuts with the wisdom of the crowd and handwriting recognition [A]. Rosemary W, Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems [C]. USA: ACM, 2012. 2895-2904.
- [8] Li B. Story generation with crowdsourced plot graphs [A]. Ferguson G, Proceedings of the 27th AAAI Conference on Artificial Intelligence [C]. USA: AAAI, 2013. 598-604.
- [9] Raykar V C. Ranking annotators for crowdsourced labeling tasks [A]. Taylor J, Neural Information Processing Systems [C]. USA: MIT, 2011. 1809-1817.
- [10] Zhang, Haoqi. Human computation tasks with global constraints [A]. Joseph A. Konstan, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems [C]. USA: ACM, 2012. 217-226.
- [11] Kambhampati S. Model-lite Planning for the Web Age Masses: The challenges of planning with incomplete and evolving domain models [A]. Ferguson G, Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence [C]. USA: AAAI, 2007. 1112-1119.
- [12] Babarian T, Schmolze G. Efficient Open World Reasoning for Planning [J]. Logical Methods in Computer Science, 2006,2:1-39.
- [13] Lemai S, Ingrand F. Interleaving temporal planning and execution in robotics domains [A]. Ferguson G, Proceedings of the Nineteenth AAAI Conference on Artificial Intelligence [C]. USA: AAAI, 2004. 617-622.
- [14] Nareyek A. Open World Planning as SCSP [M]. USA: AAAI Press, 2000. 35-46.
- [15] Talamadupula K, Kambhampati S, Schermerhorn P, Scheutz M. Planning for human-robot teaming in open worlds

- [J]. ACM Transactions on Intelligent Systems and Technology, 2010, 1(2), 14:1 – 24.
- [16] Dawid A P, Skene A M. Maximum likelihood estimation of observer error-rates using the EM algorithm [J]. Applied statistics, 1979:20 – 28.
- [17] Raykar V C, Yu S, Zhao L H, et al. Learning from crowds [J]. The Journal of Machine Learning Research, 2010, 11:1297 – 1322.
- [18] Talamadupula K. Herding the crowd: Automated planning for crowdsourced planning [A]. Kristen Grauman, The First AAI Conference on Human Computation and Crowdsourcing [C]. USA: AAI, 2013. 1121 – 1132.
- [19] Manikonda L. AI-MIX: using automated planning to steer human workers towards better crowdsourced plans [A]. Ferguson G, Proceedings of the Twenty-Eighth AAI Conference on Artificial Intelligence [C]. USA: AAI, 2014. 3004 – 3009.
- [20] Hankz Hankui Zhuo. Crowdsourced action model acquisition for planning [A]. Ferguson G, Proceedings of the Twenty-Eighth AAI Conference on Artificial Intelligence [C]. USA: AAI, 2015.

作者简介



高洁 女, 1978 年 9 月出生, 湖南新化人. 2004 年毕业于加拿大滑铁卢大学数学学院, 其后在吉林大学珠海学院工作, 2008 年到中山大学信息科学与技术学院学习, 获工学博士学位, 目前从事人工智能方面的有关研究.

E-mail: jiecao26@163.com

卓汉遼 男, 1982 年生于广东陆丰. 中山大学超级计算学院副教授, 博士. 研究方向为智能规划、数据挖掘.

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