

求最小期望权值强循环规划解

李洋¹ 文中华^{2,3} 伍小辉¹ 劳佳琪¹

(湘潭大学信息工程学院 湘潭 411105)¹ (湖南工程学院计算机与通信学院 湘潭 411104)²

(湘潭大学智能计算与信息处理教育部重点实验室 湘潭 411105)³

摘要 现实世界中,动作的执行通常都要耗费一定的代价,且由于外界环境的干扰,动作执行后的结果具有不确定性。针对这一问题,对不确定状态转移系统的动作赋予权值,使用概率分布表示状态转换的随机性,提出了强循环规划解的期望权值,并且设计了求最小期望权值强循环规划解的方法。该方法的主要思想是使用深度优先搜索求出规划问题的所有强循环规划解,再将强循环规划解分别转换成以状态到目标状态的期望权值为变元的线性方程组,最后使用高斯消元法解方程组,从而找出最小期望权值强循环规划解。

关键词 不确定规划,概率分布,最小期望权值强循环规划解,深度优先搜索,高斯消元法

中图分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.4.044

Solving Strong Cyclic Planning with Minimal Expectation Weight

LI Yang¹ WEN Zhong-hua^{2,3} WU Xiao-hui¹ LAO Jia-qi¹

(College of Information Engineering, Xiangtan University, Xiangtan 411105, China)¹

(Department of Computer & Communication, Hunan Institute of Engineering, Xiangtan 411104, China)²

(Key Laboratory of Intelligent Computing & Information Processing, Ministry of Education, Xiangtan University, Xiangtan 411105, China)³

Abstract In real world, the action's execution often takes a cost. Due to the interference of the external environment, the result of an action's execution is uncertain. To solve this problem, we added weight to the action in system, used a probability distribution to show the stochastic state transition. A method of solving strong cyclic planning with minimal expectation weight was designed based on the proposed concept of expectation weight for strong cyclic planning. Mainly, this method applies depth-first search for getting all strong cyclic plannings. Then, it uses Gaussian elimination to solve the problem after converting the plannings into linear equations with variables in expectation weight.

Keywords Uncertainty planning, Probability distribution, Strong cyclic planning with minimal expectation weight, Depth-first search, Gaussian elimination

1 引言

智能规划是人工智能研究中的一个重要领域,具有较高的理论和应用价值。不确定规划^[1]是智能规划下的一个热点问题,包含许多例如涉及可达性目标的^[3]、概率分布的^[6,7]、限于部分可观察的^[2,9]规划问题。在对可达性目标作规划时,使用基于模型检测的方法可以求解规划问题的强、弱、强循环规划解^[4,5,10,12]。

Cimatti 等人首次在文献^[3]中提出强循环规划的概念,在文献^[4]中给出求强循环规划解的方法。之后,又有学者在文献^[8]提出了新的求强循环规划解的算法。但现实生活中动作的执行通常需要耗费不同的代价,针对这一问题,本文对不确定状态转移系统的每个动作赋予不同的权值来表示动作执行时耗费的代价值。又由于现实中外界因素的干扰,动作

的执行带有不确定性,实际执行后到达的状态与预期到达的状态可能不同,为此,本文使用概率分布来描述状态转换的随机性。在执行强循环规划解时,因为强循环规划解中可能存在循环结构,每次从初始状态到达目标状态所执行过的动作个数是随机的,对应地所有动作的权值总和呈概率分布,则动作权值总和的期望值代表强循环规划解实际执行时耗费的平均代价值。文献^[12]中提出了最小权值强规划解,但为了处理以耗费代价值最小为优化目标的强循环规划问题,简单地累加规划解中动作的权值以求最小权值强循环规划解的意义不大,而本文提出的求动作权值总和的期望值最小的强循环规划解即最小期望权值强循环规划解具有十分重要的意义。

本文首先删除不确定状态转移系统中对构成强循环规划解没有影响的状态动作序偶,减小问题规模,并根据删除结果判断问题是否有强循环规划解。在强循环规划解存在的情况

到稿日期:2014-05-03 返修日期:2014-07-29 本文受国家自然科学基金(61272295,61105039,61202398),湘潭大学智能计算与信息处理教育部重点实验室,湖南省重点学科建设项目(0812)资助。

李洋(1991-),男,硕士生,主要研究方向为智能规划,E-mail:1311658830@qq.com;文中华(1966-),男,博士,教授,博士生导师,主要研究方向为智能规划、图论及算法;伍小辉(1988-),男,硕士生,主要研究方向为智能规划;劳佳琪(1990-),男,硕士生,主要研究方向为智能规划。

下,使用深度优先搜索技术从初始状态开始扩展,求出所有的强循环规划解。接着,以规划解中每个状态到目标状态的期望权值为变元,将每个状态动作序偶蕴含的状态转移关系转换成各状态期望权值间的关系,构造出线性方程组,然后利用高斯消元法求解线性方程组,得出每组强循环规划解的期望权值,最后找出最小期望权值强循环规划解。

2 相关定义

定义 1 一个带权值、状态转换呈概率分布的规划领域 $D=(S,A,R,W,P)$ 是一个五元组,其中, S 是一个有限状态集; A 是一个带权值的有限动作集; $R:S \times A \rightarrow 2^S$ 是状态转移函数,在状态 s 下执行动作 a 可能得到的结果状态的集合就是 $R(s,a)$,如果 $|R(s,a)| \geq 1$,则称动作 a 在状态 s 下是可执行的; W 是动作的权值集合,动作 a 的权值为 w_a ; P 是一个概率分布, $P(s,a,s')$ 表示在状态 s 下执行动作 a 之后得到的结果状态是 s' 的概率。对确定动作 a , $P(s,a,s')=1$ 。

$A(s)=\{a \in A \mid \exists s' \in S. P(s,a,s') \neq 0\}$ 为在状态 s 下的可执行动作集,若 $A(s)$ 非空,则称状态 s 是可扩展的。称 (s,a) 为状态动作序偶,称 $\pi=\{(s,a) \mid s \in S, a \in A(s)\}$ 为方针。

设 π 是一个方针,则对于任意状态 s ,至多有一个动作 a 使得 $(s,a) \in \pi$,方针中的状态集即为 $S_\pi=\{s \mid (s,a) \in \pi\}$ 。

定义 2 设 $D=(S,A,R,W,P)$ 是一个带权值的、状态转换呈概率分布的规划领域,一个对于 D 的规划问题 I 是一个三元组 (D,S_0,S_g) ,其中, $S_0 \subseteq S$ 是初始状态集合; $S_g \subseteq S$ 是目标状态集合。

定义 3 设 π 是规划领域 $D=(S,A,R,W,P)$ 中的一个方针, $I=(D,S_0,S_g)$ 是 D 上的一个规划问题,从初始状态集 S_0 所导出的 π 的执行结构为 $K=(Q,T)$,其中, $Q \subseteq S$ 和 $T \subseteq S \times S$ 是满足以下条件的最小集合:

- (1) 若 $s \in S_0$,那么 $s \in Q$;
- (2) 若 $s \in Q$ 且存在某个 $(s,a) \in \pi, s' \in R(s,a)$,那么 $s' \in Q$ 且 $(s,s') \in T$ 。

状态 $s \in Q$ 是 K 的一个终止状态当且仅当不存在 $s' \in Q$ 使得 $(s,s') \in T$ 。在后面的内容中一律用 $S_{terminal(K)}$ 来表示执行结构 K 的终止状态集。

π 是 I 的强循环规划解当且仅当 $S_{terminal(K)} \subseteq S_g$, 并且 $(\forall s \in Q, \exists s' \in S_{terminal(K)}) s'$ 是 s 可达的。

定义 4 设 π 是规划问题 $I=(D,S_0,S_g)$ 的一个方针,对 $(s,a) \in \pi$,若 a 是不确定动作即 $|R(s,a)| > 1$,则称状态 s 为方针 π 中的汇合状态,称 $s' \in R(s,a)$ 为 s 下的分支状态, a 为状态 s 下的汇合动作。

定义 5 设 $D=(S,A,R,W,P)$ 是一个规划领域, $I=(D,S_0,S_g)$ 是一个规划问题,状态和动作的交替序列 $L=s_{i0} a_{j1} s_{i1} a_{j2} \dots a_{jk} s_{ik}$ 称为状态 s_{i0} 到状态 s_{ik} 的一条路径,其中, $s_{ir} \in R(s_{ir-1}, a_{jr})$ 。若 L 中不存在相同的动作,则称 L 为无环路径。

路径 L 出现的概率 $P_L=P(s_{i0}, a_{j1}, s_{i1}) \times P(s_{i1}, a_{j2}, s_{i2}) \times \dots \times P(s_{ik-1}, a_{jk}, s_{ik})$ 。

$SL=s_{i0} s_{i1} \dots s_{ik}$ 称为路径的状态序列。若 $\exists a_{jk+1} s_{ik+1}$, 满足 $s_{ik+1} \in R(s_{ik}, a_{jk+1})$, 有:

- ① 若 $s_{ik+1} \in \{s_{i0}, s_{i1}, \dots, s_{ik}\}$, 记为 $s_{ik+1} \text{ In } SL$;
- ② 若 $s_{ik+1} \in \{s_{i0}, s_{i1}, \dots, s_{ik}\}$, 且 $s_{ik+1} \in \{s_{i0}, s_{i1}, \dots, s_{ir}\}$, 记

为 $s_{ik+1} \text{ Front}(s_{ir}, SL)$;

③ 若 $s_{ik+1} \in \{s_{i0}, s_{i1}, \dots, s_{ik}\}$, 且 $s_{ik+1} \in \{s_{ir+1}, s_{ir+2}, \dots, s_{ik}\}$, 记为 $s_{ik+1} \text{ Back}(s_{ir}, SL)$ 。

若 $s_{ik+1} \in S_g$ 或者 $s_{ik+1} \text{ In } SL$, 则称状态序列 $SL'=s_{i0} s_{i1} \dots s_{ik} s_{ik+1}$ 扩展结束,也称状态 s_{i0} 扩展结束。

定义 6 设 $D=(S,A,R,W,P)$ 是一个规划领域, $L=s_{i0} a_{j1} s_{i1} a_{j2} \dots a_{jk} s_{ik}$ 是状态 s_{i0} 到 s_{ik} 的一条路径,则称路径 L 的权值为 L 中所有动作权值的总和,即 $W_L=w_{a_{j1}}+w_{a_{j2}}+\dots+w_{a_{jk}}$ 。设 L_1, L_2, \dots, L_n 为 s_{i0} 到 s_{ik} 的所有路径, $P_{L_1}, P_{L_2}, \dots, P_{L_n}$ 为各路径出现的概率, $W_{L_1}, W_{L_2}, \dots, W_{L_n}$ 为各路径的权值,则 s_{i0} 到 s_{ik} 的期望权值为 $E_{s_{i0} s_{ik}}=W_{L_1} * P_{L_1} + W_{L_2} * P_{L_2} + \dots + W_{L_n} * P_{L_n}$ 。

定义 7 设 π 是规划问题 $I=(D,S_0,S_g)$ 的一个强循环规划解,则 π 的期望权值为 s 到 s' 的期望权值 $E_{ss'}$, 简记为 E_s , 其中 $s, s' \in S_\pi$ 且 $s \in S_0, s' \in S_g$ 。若 π 的期望权值在 I 的所有强循环规划解中最小,则称 π 为 I 的最小期望权值强循环规划解。

3 问题求解

设 $D=(S,A,R,W,P)$ 是一个带权值的、状态转换呈概率分布的不确定状态转移系统, $I=(D,S_0,S_g)$ 是 D 上的一个规划问题,规划问题 I 是在 D 上求从 S_0 出发到 S_g 的最小期望权值强循环规划解。

3.1 删除不可能构成强循环规划解的状态动作序偶

实际上,有很多状态动作序偶是不可能构成强循环规划解的,若在搜索强循环规划解之前将这些状态动作序偶删除,可以减小问题规模,提高求规划解的效率。算法 1 以全方针 $AllPol=\{(s,a) \mid a \in A(s)\}$ 为初始方针,通过反复调用 while 循环体中的子程序 DeleteUnconnected 和 DeleteOut 来修剪全方针,直到其不变为止。再根据结果判断强循环规划解是否存在,若存在,则返回 success,可继续搜索强循环规划解。

算法 1

1. DeleteStateActionPair(S_0, S_g)
2. $\pi' \leftarrow \emptyset; \pi \leftarrow AllPol$
3. while $\pi \neq \pi'$ do
4. $\pi' \leftarrow \pi$
5. $\pi \leftarrow DeleteUnconnected(DeleteOut(\pi, S_g), S_g)$
6. if $S_0 \subseteq (S_g \cup S_\pi)$ then
7. return (success)
8. else return (failure)

算法 2 和算法 3 分别解释了子程序 DeleteOut 和 DeleteUnconnected, DeleteOut 负责删除跳出当前状态集 $S_g \cup S_\pi$ 的状态动作序偶, DeleteUnconnected 用于删去到不了目标的边。

其中, $ComputeOut(\pi, S)=\{(s,a) \in \pi \mid R(s,a) \not\subseteq S \text{ 且 } R(s,a) \neq S\}$, $WeakPreImg(S)=\{(s,a) \mid R(s,a) \cap S \neq \emptyset\}$ 。

算法 2

1. DeleteOut(π, S_g)
2. $\pi' \leftarrow \pi - ComputeOut(\pi, S_g \cup S_\pi)$
3. return(π')

算法 3

1. DeleteUnconnected(π, S_g)

2. $\pi' \leftarrow \emptyset$
3. while $\pi'' \neq \pi'$ do
4. $\pi'' \leftarrow \pi'$
5. $\pi' \leftarrow \pi \cap \text{WeakPreImg}(S_g \cup S_{\pi'})$
6. return (π')

3.2 深度优先搜索所有的强循环规划解

当规划问题存在强循环规划解时,本文使用深度优先搜索技术,从初始状态开始,对每个状态选择一个可执行的动作进行扩展,到达新的状态后继续扩展,直到搜索得到一个强循环规划解。当一个状态所有的可执行动作都被搜索执行过时,搜索将回溯到执行某动作后到达此状态的上层状态。反复执行此过程,求出所有的强循环规划解。

算法 4

1. StrongCyclicPlan($s_p, \&\pi, SL, s_j, a_j, s_{jx}$)
2. $SL \leftarrow SL \cup s_p$
3. foreach $a \in A(s_p)$
4. $s_q[1 \dots n] = R(s_p, a)$
5. if $|R(s_p, a)| = 1$ //动作 a 是确定动作
6. CertainAction($s_p, s_q, \pi, SL, s_j, a_j, s_{jx}$)
7. else UncertainAction($s_p, s_q, \pi, SL, s_j, a_j, s_{jx}$)
8. return (all_pi)

算法 4 是深度优先搜索求强循环规划解的主程序。 s_p 表示待扩展状态, s_j 是状态序列 SL 中次序最后的汇合状态, a_j 和 s_{jx} 分别表示 s_j 下的汇合动作和分支状态。当在 SL 中状态 s_p 之前无汇合状态时, s_j, a_j, s_{jx} 的值都为 Null。子程序 CertainAction 和 UncertainAction 分别对确定动作和不确定动作进行处理。由于可能存在多个强循环规划解,最后返回方针 π 的集合 all_pi 。

在搜索的过程中,某些动作对当前搜索得到强循环规划解没有帮助,本次应该放弃选择此类动作,即对状态 $s, a \in A(s)$:

I. 若 a 是确定动作,那么

①当 $s_j = \text{Null}$ 时,若 $s_q \text{ In } SL$,本次不选择 a 对 s 扩展。

例 1 如图 1(a)所示,对 s_2 扩展时, $L = s_1 a_1 s_2, SL = s_1 s_2, s_j = \text{Null}$,若选择 a_2 ,有 $s_1 \text{ In } SL, \pi = \{(s_1, a_1), (s_2, a_2)\}$,非强循环规划解,故本次不选择 a_2 。

②当 $s_j \neq \text{Null}$ 时,若 $s_q \text{ Back}(s_j, SL)$,本次不选择 a 对 s 扩展。

例 2 如图 1(a)所示,对 s_5 扩展时, $L = s_1 a_1 s_2 a_3 s_4 a_5 s_5, SL = s_1 s_2 s_4 s_5, s_j = s_2$,若选择 a_6 ,有 $s_4 \text{ Back}(s_2, SL), \pi = \{(s_1, a_1), (s_2, a_3), (s_4, a_5), (s_5, a_6)\}$ 非强循环规划解,故本次不选择 a_6 。

II. 若 a 是不确定动作,那么

①当 $s_j = \text{Null}$ 时,若 $\forall s_q \in R(s, a)$,有 $s_q \text{ In } SL$,本次不选择 a 对 s 扩展。

例 3 如图 1(b)所示,对 s_3 扩展时, $L = s_1 a_1 s_2 a_2 s_3, SL = s_1 s_2 s_3, s_j = \text{Null}$,若选择 a_3 ,有 $s_1 \text{ In } SL, s_2 \text{ In } SL, \pi = \{(s_1, a_1), (s_2, a_2), (s_3, a_3)\}$,非强循环规划解,故本次不选择 a_3 。

②当 $s_j \neq \text{Null}$ 时,若 $\forall s_q \in R(s, a)$,有 $s_q \text{ Back}(s_j, SL)$,本次不选择 a 对 s 扩展。

例 4 如图 1(b)所示,对 s_7 扩展时, $L = s_1 a_1 s_2 a_2 s_3 a_4 s_5 a_5 s_6 a_7 s_7, SL = s_1 s_2 s_3 s_5 s_6 s_7, s_j = s_3$,若选择 a_8 ,有 $s_5 \text{ Back}(s_3,$

$SL), s_6 \text{ Back}(s_3, SL), \pi = \{(s_1, a_1), (s_2, a_2), (s_3, a_4), (s_5, a_5), (s_6, a_7), (s_7, a_8)\}$,非强循环规划解,故本次不选择 a_8 。

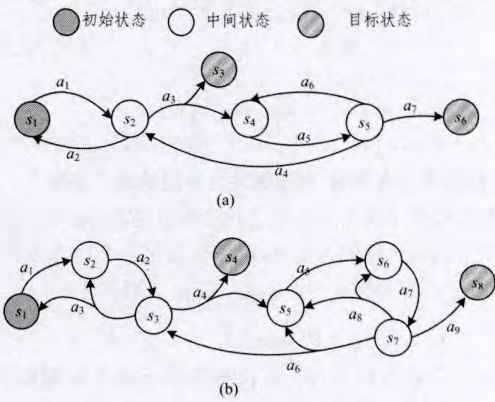


图 1 两个不确定的状态转移系统

算法 5 和 6 分别解释了 CertainAction 和 UncertainAction,两者都采用递归结构。算法 5 中第 2-3 行判断动作对本次搜索强循环规划解是否无用,若是,则不选择。第 5-8 行根据不同的扩展结束情况分别处理,当 $s_j = \text{Null}$ 时,若 $s_q \in S_g$,扩展结束,此时得到强循环规划解,更新 all_pi ;当 $s_j \neq \text{Null}$ 时,若 $s_q \in S_g$ 或者 $s_q \text{ Front}(s_j, SL)$,扩展结束,将 π 并入 $part_pi(a_j, s_{jx})$ 。 $part_pi(a_j, s_{jx})$ 记录汇合动作 a_j 下分支状态 s_{jx} 扩展结束时的方针集合。

算法 5

1. CertainAction($s_p, s_q, \pi, SL, s_j, a_j, s_{jx}$)
2. if ($s_j = \text{Null}$ and $s_q \text{ In } SL$) or $s_q \text{ Back}(s_j, SL)$
3. continue //本次不选择此动作
4. $\pi \leftarrow \pi \cup (s_p, a)$
5. if $s_j = \text{Null}$ and $s_q \in S_g$ //扩展结束
6. $all_pi \leftarrow all_pi \cup \{\pi\}$ //更新 all_pi
7. else if $s_j \neq \text{Null}$ and ($s_q \in S_g$ or $s_q \text{ Front}(s_j, SL)$)
8. $part_pi(a_j, s_{jx}) \leftarrow part_pi(a_j, s_{jx}) \cup \{\pi\}$
9. $\pi \leftarrow \pi - (s_p, a)$
10. continue
11. StrongCyclicPlan($s_q, \pi, SL, s_j, a_j, s_{jx}$)
12. $\pi \leftarrow \pi - (s_p, a)$

算法 6 解释了子程序 UncertainAction,第 6-9 行对此汇合动作下的每个分支状态 $s_q[i]$ 进行判断,若扩展结束,将 π 并入 $part_pi(a, s_q[i])$,否则继续递归调用 StrongCyclicPlan 对 $s_q[i]$ 进行扩展。第 10-13 行是算法回溯到汇合动作时,首先合并其每个分支状态扩展结束时得到的方针集合。若此时 $s_j = \text{Null}$,将合并结果并入强循环规划解集合,否则,状态 s_p 位于扩展 s_{jx} 的路径中,将合并结果更新到 $part_pi(a_j, s_{jx})$ 中。其中, $Merge(A, B) = \{A_i \cup B_j \mid A_i \subseteq A, B_j \subseteq B, i = 1, 2, \dots, m, j = 1, 2, \dots, n\}$ 。

算法 6

1. UncertainAction($s_p, s_q, \pi, SL, s_j, a_j, s_{jx}$)
2. if $s_j = \text{Null}$ and $s_q[1 \dots n] \text{ In } SL$ or $s_q[1 \dots n] \text{ Back}(s_j, SL)$
3. continue
4. $\pi \leftarrow \pi \cup (s_p, a)$
5. $\pi'(a, s_q[1 \dots n]) \leftarrow \pi$
6. for $i = 1$ to n
7. if $s_q[i] \in S_g$ or $s_q[i] \text{ In } SL$

```

8.     part_π(a, sq[i]) ← part_π(a, sq[i]) ∪ {π}
9.     else StrongCyclicPlan(sq[i], π'(a, sq[i]), SL, sp, a, sq[i])
10.    M ← Merge(part_π(a, sq[1..n]))
11.    if sj = Null // 回溯
12.        all_π ← all_π ∪ M // 合并规划解
13.    else part_π(aj, sjx) ← part_π(aj, sjx) ∪ M
14.    π ← π'(a, sq[1]) ∪ ... ∪ π'(a, sq[n]) - (sp, a)

```

3.3 构造线性方程组, 求出线性方程组的增广矩阵

状态动作序偶包含状态之间的转移关系, 如果以状态到目标状态的期望权值为变元, 可以构造出一个线性方程。即对 $(s, a) \in \pi$, 若 $R(s, a) = \{s_{e1}, s_{e2}, \dots, s_{ex}\}$, 则有:

$$E_s = w_a + \sum_{i=1}^x E_{s_{ei}} * P(s, a, s_{ei}) \quad (1)$$

其中, w_a 是动作权值, E_s 是状态到目标状态的期望权值, $P(s, a, s_{ei})$ 是状态转移的概率。

求出所有的强循环规划解后, 对每个规划解可以构造生成一个由多个式(1)组成的线性方程组。算法 7 将规划解转换成线性方程组并求得其增广矩阵, 其中第 5 行是生成形如式(1)的线性方程, 第 6 行将线性方程并入线性方程组, 第 7 行的 AugMatrix 函数求出线性方程组的增广矩阵。

算法 7

```

1. PlanToAugMatrix(all_π)
2.  foreach πi ∈ all_π
3.     foreach (sij, aij) ∈ πi
4.         R(sij, aij) = {se1, se2, ..., sex}
5.         Esij = waij + Ese1 * P(sij, aij, se1) + ... + Esex * P(sij,
            aij, sex)          (**)
6.         Xπi ← Xπi ∪ (**)
7.         Mπi ← AugMatrix(Xπi)
8.     return Mπ1, Mπ2, ..., Mπk

```

3.4 高斯消元法解线性方程组, 求出最小期望权值的强循环规划解

高斯消元法的基本思想是用初等行变换把增广矩阵转换为行阶梯矩阵, 然后回代求出线性方程组的解。由于本文中待求解的线性方程组由现实的强循环规划解转换而来, 因此每组线性方程组有且仅有唯一解。对增广矩阵 M_{π_i} , 算法从左至右, 每作出以下 3 步, 跳到下一列和下一行:

- ① 找出每列的绝对值最大的一个非 0 的数, 将第一行的值与该行交换, 使得第一行拥有这一列的最大值;
- ② 将第一行的数除以该数, 使该行第一个数成为 1;
- ③ 对每一行减去第一行乘以每一行的第一个数, 使得每一行的第一个数变为 0。

所有步骤完成后, 增广矩阵 M_{π_i} 会变成一个行阶梯矩阵 M_{π_i}' , 即:

$$M_{\pi_i} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \Rightarrow M_{\pi_i}' = \begin{pmatrix} b_1 & \dots & \dots & \dots & \dots \\ 0 & b_2 & \dots & \dots & \dots \\ 0 & 0 & \ddots & \dots & \dots \\ 0 & 0 & \dots & b_m & \dots \end{pmatrix}$$

再用代入法就可以求解该方程组。得出所有规划解对应方程组的解后, 可以找出期望权值最小的强循环规划解。

4 算法实例说明及分析

例 设 $D = (S, A, R, W, P)$ 是一个带权值的、状态转换

呈概率分布的不确定状态转移系统。如图 2 所示, $I = (D, S_0, S_g)$ 是 D 上的一个规划问题, 其中, $S_0 = \{s_1\}$ 是初始状态集合, $S_g = \{s_8\}$ 是目标状态集合, 规划问题 I 是在 D 上求从 S_0 出发到 S_g 的最小期望权值强循环规划解。

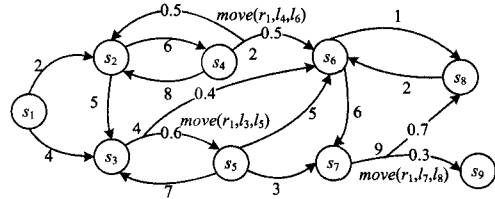


图 2 带权值的、状态转换呈概率分布的不确定状态转移系统

首先调用算法 DeleteStateActionPair, 会删去状态 s_7, s_9 以及动作 $move(r_1, l_7, l_8), move(r_1, l_5, l_7), move(r_1, l_6, l_7)$, 接着深度优先搜索得到规划问题的所有强循环规划解, $\pi_1 = \{(s_1, move(r_1, l_1, l_2)), (s_2, move(r_1, l_2, l_4)), (s_4, move(r_1, l_4, l_6)), (s_6, move(r_1, l_6, l_8))\}$, $\pi_2 = \{(s_1, move(r_1, l_1, l_2)), (s_2, move(r_1, l_2, l_3)), (s_3, move(r_1, l_3, l_5)), (s_5, move(r_1, l_5, l_6)), (s_8, move(r_1, l_6, l_8))\}$, $\pi_3 = \{(s_1, move(r_1, l_1, l_3)), (s_3, move(r_1, l_3, l_5)), (s_5, move(r_1, l_5, l_6)), (s_6, move(r_1, l_6, l_8))\}$ 。再将每个强循环规划解转换成线性方程组, 如对 π_1 构造的方程组及方程组的增广矩阵为:

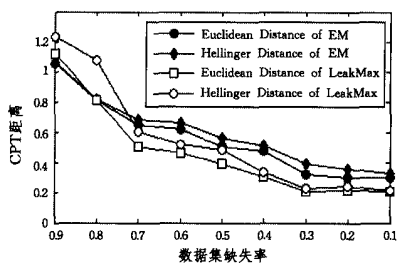
$$\begin{cases} E_{s_1} = E_{s_2} + 2 \\ E_{s_2} = E_{s_4} + 6 \\ E_{s_4} = 0.5E_{s_2} + 0.5E_{s_6} + 2 \\ E_{s_6} = E_{s_8} + 1 \\ E_{s_8} = 0 \end{cases} \Rightarrow \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 2 \\ 0 & 1 & -1 & 0 & 0 & 6 \\ 0 & -5 & 10 & -5 & 0 & 20 \\ 0 & 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

最后利用高斯消元法, 可解得 π_1 的期望权值为 $E_{s_1} = 19$; 同理, 可求得 π_2 的期望权值为 $E_{s_1}' = 15$, π_3 的期望权值为 $E_{s_1}'' = 12$ 。因此, 规划问题所求的最小期望权值的强循环规划解为 $\pi_3 = \{(s_1, move(r_1, l_1, l_3)), (s_3, move(r_1, l_3, l_5)), (s_5, move(r_1, l_5, l_6)), (s_6, move(r_1, l_6, l_8))\}$ 。

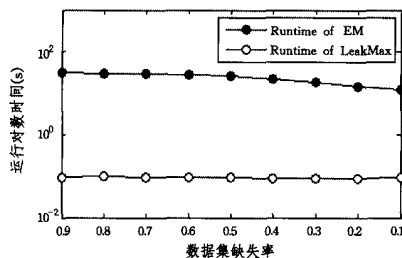
结束语 本文在基于模型检测的方法下使用概率分布描述状态转换的优先关系, 提出了强循环规划解的期望权值这一概念, 设计了求解最小期望权值的强循环规划解的算法。算法实例证明了算法的正确性。本文的进一步工作有: 将求解最小期望权值强循环规划解的思想应用到其他规划领域, 如多 Agent 规划。

参考文献

- [1] Kuter U, Nau D, Reisner, et al. Using Classical Planners to Solve Nondeterministic Planning Problems[C]//Proc. of the 18th Int Conf on Automated Planning and Scheduling. Menlo Park, CA: AAAI press, 2008
- [2] Bertoli P, Cimatti A, Roveri M, et al. Strong planning under partial observability[J]. Artificial Intelligence, 2006, 170 (4/5): 337-384
- [3] Cimatti A, Roveri M, Traverso P. Automatic OBDD-based generation of universal plans in nondeterministic domains[C]//Proc. of the 15th National Conf on Artificial Intelligence. Menlo Park, CA: AAAI Press, 1998: 875-881



(a)学习精度比较



(b)学习效率比较

图4 不完整样本集下参数学习性能的比较

由图4(a)可见,在样本集的数据缺失率达到了0.9和0.8时,本文所提算法的学习精度略低于EM算法,但随着数据缺失率逐渐减少,其学习精度逐步优于EM算法,这也与图3(a)所示的统计结果一致。图4(b)显示出,在样本中数据随机缺失时,EM算法的学习效率远低于本文所提算法,这是由于EM算法通过一定的迭代次数来保证学习精度,且数据缺失越多,其计算收敛速度越慢^[1]。在本实验中,EM算法的迭代次数设为10,其与本文所提算法的运行时间最大差值出现在数据缺失率为0.9处,达到29.95s。综合上述对比实验分析可知,在样本信息不充分的小数据集条件下的CPT参数学习中,本文所提算法能有效提高参数学习的精度与效率。

结束语 贝叶斯网络利用条件独立关系,将联合概率分布分解为局部条件概率分布的乘积,降低了系统概率模型的复杂度。对于复杂多态系统,当变量的数量很多或变量的值域范围较大时,局部条件概率表的规模和复杂度将按指数级增长,这使得条件概率参数学习变得困难,从而影响了整个网络模型的实用性。本文利用因果机制独立假设,将条件概率分布分解成以单一原因的独立影响概率分布为因子的连乘形式,使条件概率表的规模表现为父节点数和状态数的线性形式,降低了条件概率分布的复杂度。在模型不精确和样本信息不充分情况下,利用Leaky Noisy-MAX模型,提出了一种条件概率表参数学习方法,其能有效降低参数学习的复杂度,

提高参数学习性能,增强了模型的实用性。

参考文献

- [1] 张连文,郭海鹏. 贝叶斯网引论[M]. 北京:科学出版社,2006
- [2] Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference[M]. San Francisco, CA: Morgan Kaufmann Publishers Inc., 1988:383-408
- [3] 王伟华,周经伦,何祖玉,等. 基于贝叶斯网络的复杂系统故障诊断[J]. 计算机集成制造系统,2004,10(2):230-234
- [4] 康长青,方磊,华丽,等. 基于贝叶斯 Noisy Or Gate 网络的多传感器目标分类识别[J]. 计算机测量与控制,2011,19(6):1387-1389
- [5] 柴慧敏,王宝树. 用于态势估计的一种构造贝叶斯网络参数的方法[J]. 计算机科学,2006,33(9):140-142
- [6] Heckerman D. Causal Independence for Knowledge Acquisition and Inference[C]//Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence (UAI'93). San Mateo: Morgan Kaufmann Publishers Inc., 1993:122-127
- [7] Zhang N L, Poole D. Exploiting Causal Independence in Bayesian Network Inference[J]. Journal of Artificial Intelligence Research, 1996, 5: 301-328
- [8] Díez F J, Galán S F. Efficient Computation for the Noisy-Max[J]. International Journal of Intelligent Systems, 2004, 18(2): 165-177
- [9] Díez F J, Druzdzel M J. Canonical Probabilistic Models for Knowledge Engineering[R]. Technical Report CISIAD-06-01. UNED Madrid, 2007
- [10] Spirtes P, Glymour C N, Scheines R. Causation, Prediction, and Search[M]. Cambridge: The MIT Press, 2000
- [11] Zagorecki A, Druzdzel M J. Knowledge Engineering for Bayesian Networks: How Common Are Noisy-MAX Distributions in Practice? [J]. IEEE Transactions on Systems, Man, and Cybernetics; Systems, 2013, 43(1): 186-195
- [12] Shi D. Extending Noisy-Max Gates to Bidirectional Models[J]. Journal of Information & Computational Science, 2013, 10(13): 4085-4096
- [13] Zagorecki A, Voortman M, Druzdzel M J. Decomposing Local Probability Distributions in Bayesian Networks for Improved Inference and Parameter Learning[C]//Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference (FLAIRS-2006). Menlo Park, CA: AAAI Press, 2006:860-865

(上接第220页)

- [4] Cimatti A, Pistore M, Roveri M, et al. Weak, strong, and strong cyclic planning via symbolic model checking[J]. Artificial Intelligence, 2003, 147(1/2): 35-84
- [5] 文中华,黄巍,刘任任,等. 模型检测规划中的状态分层方法[J]. 软件学报, 2009, 20(4): 858-869
- [6] Oh J, Meneguzzi F, Sycara K. Probabilistic plan recognition for intelligent information agents[C]//Proc. ICAART. 2011
- [7] Martín Y E, Moreno M D R, Smith D E. Probabilistic Plan Graph Heuristic for Probabilistic Planning[C]//Proc. of the Twenty-Fifth AAAI Conference on Artificial Intelligence. 2011: 1774-1745

- [8] Fu J, Ng V, Bastani F B, et al. Simple and fast strong cyclic planning for fully-observable nondeterministic planning problems [C]//Proc. of the 22nd Int Joint Conf on Artificial Intelligence. Menlo Park, CA: AAAI Press, 2011:1949-1954
- [9] 周俊萍,殷明浩,谷文祥,等. 部分可观察强规划中约减观察变量的研究[J]. 软件学报, 2009, 20(2): 290-304
- [10] 唐杰,文中华,汪泉,等. 不确定可逆规划的强循环规划解[J]. 计算机研究与发展, 2013, 50(9): 1970-1980
- [11] 刘铭,徐杨,陈峥,等. 基于 Multi-agent 系统的多飞行器协同路径规划方法的研究[J]. 计算机科学, 2012, 39(1): 219-233
- [12] 陈建林,文中华,马丽丽,等. 一种求解最小权值强规划的方法[J]. 计算机工程, 2011, 37(17): 167-171