

基于动态贝叶斯网络的可分解信念状态空间压缩算法

仵 博^{1,2,3}, 吴 敏^{1,2}, 郑红燕³, 冯延蓬³

(1. 中南大学信息科学与工程学院, 湖南 长沙 410083; 2. 先进控制与智能自动化湖南省工程实验室, 湖南 长沙 410083;
3. 深圳职业技术学院教育技术与信息中心, 广东 深圳 518055)

摘 要: 针对部分可观察马尔可夫决策过程 (POMDP) 的信念状态空间规模“维数灾”问题, 根据信念状态变量存在可分解和独立关系的特性, 提出一种基于动态贝叶斯网络 (DBN) 的可分解信念状态空间压缩算法 (factored belief states space compression, FBSSC). 该算法通过构建变量间依赖关系图, 根据独立关系检验去除多余边, 将转移函数联合概率分解成若干个条件概率的乘积, 实现信念状态空间的无损压缩. 对比实验和 RoboCupRescue 仿真结果表明, 本文算法具有较低误差率、较高收敛性和普遍适用性等特性.

关键词: 马尔可夫决策过程; 动态贝叶斯网络; 维数灾; 信念状态空间; 条件独立

中图分类号: TP181

文献标识码: A

文章编号: 1002-0411(2012)-06-0713-07

Factored Belief States Space Compression Algorithm Based on Dynamic Bayesian Network

WU Bo^{1,2,3}, WU Min^{1,2}, ZHENG Hongyan³, FENG Yanpeng³

(1. School of Information Science and Engineering, Central South University, Changsha 410083, China;
2. Hunan Engineering Laboratory for Advanced Control and Intelligent Automation, Changsha 410083, China;
3. Education Technology and Information Center, Shenzhen Polytechnic, Shenzhen 518055, China)

Abstract: For the dimensionality curse problem of belief state space scale of partially observable Markov decision process (POMDP), a factored belief states space compression (FBSSC) algorithm based on dynamic Bayesian network (DBN) is proposed according to the decomposable features and dependent relationship of the belief state variables. Based on the building of the graph of dependent relationship among variables, the algorithm removes the redundant edges by detecting the dependent relationships, and decomposes the joint probability of transition function into the product of several conditional probabilities, which realizes the lossless compression of belief states space. Comparison experiments and RoboCupRescue simulation results show that the algorithm has the characteristics of lower error rate, higher convergence, and general applicability.

Keywords: MDP (Markov decision process); DBN (dynamic Bayesian network); curse of dimensionality; belief states space; conditional independence

1 引言 (Introduction)

在 Agent 系统的实际应用中, Agent 所处的真实环境往往是动态变化的和不确定的, Agent 对环境的感知信息通常是部分的, 这时需要 Agent 能够在动态不确定环境中根据这些部分感知的信息进行决策. 部分可观察马尔可夫决策过程^[1] (partially observable Markov decision process, POMDP) 是 Agent 在动态不确定环境下进行决策的一种理想数学模型. 由于 POMDP 能够更加客观地、准确地描述真实世界, 使它成为研究随机决策过程的重要分支, 因此最近成为计算机、控制和管理等学科研究的热点^[2].

在 POMDP 中, Agent 观察到的信息不完整或者是不可能直接知道自己的当前状态, 它必须利用现有的部分信息、历史动作序列和立即报酬值进行决策, 这种解决方案是非马尔可夫特性的. 为了能够利用 Bellman 理论, POMDP 引入一个信念状态空间, 使其只与当前的信念状态空间有关, 与历史的信念状态空间无关. 当引入信念状态空间后, POMDP 转化为基于信念状态空间的马尔可夫决策过程^[3] (Markov decision process, MDP). 寻求一种最优策略将当前的信念状态映射到 Agent 的行动上, 根据当前的信念状态和动作就可以决定下一个周期的信念状态和动作. 由于求解信念状态空间是一个 NP 难问题, 传统的求解精确值函数和精确策略的算法

难以实现. 为了能够求解信念状态空间, 目前普遍的做法是压缩状态空间冗余信息, 减少计算工作量.

Roy 等人^[4]提出指数族主元分析 (Exponential family Principal Component Analysis, EPCA) 降维算法, 该算法只考虑影响决策的主要特征根, 从而降低信念状态空间. 但是, 该算法使 POMDP 值函数分段线性凸特性丢失, 无法对压缩后的信念状态进行值迭代求解. Li 等人^[5]利用非负矩阵分解 (non-negative matrix factorisation, NMF) 原理^[6], 提出一种正交非负矩阵分解算法 (orthogonal NMF, ONMF), 其主要思想是将原始的大矩阵 $\mathbf{V}$ 近似分解为低秩的 $\mathbf{V} \approx \mathbf{WH}$ 形式, 从而降低维数, ONMF 通过非负分解, 低秩矩阵全部为正值. 该算法具有收敛速度快、存储空间小的特点, 能够将高维的数据进行降维处理. 但是在矩阵维数较大时, ONMF 非常耗时, 已证明非负矩阵分解是个 NP 问题; 另外, 当增加新的训练样本或类别时, ONMF 必须进行重复学习.

在 POMDP 实际世界模型中, 信念状态空间概率分布具有明显的稀疏特性^[7-8], 高维的信念状态空间往往可以由低维的信念状态空间来表示. 这也符合认知心理学的理论, 大脑对外界只会处理其中的“有意义”数据, 而对其它数据“视而不见”. 基于这种思想, 本文提出一种基于动态贝叶斯网络的可分解信念状态空间压缩算法 (factored belief states space compression, FBSSC). 利用 POMDP 的结构特性, 将状态、观察和动作进行可分解表示, 然后利用动态贝叶斯网络 (dynamic Bayesian network, DBN) 的条件独立和上下文独立对其转移函数进行分解压缩, 消除概率为 0 的取值, 降低信念状态空间的稀疏性. 对比实验和机器人救援仿真结果表明, 本文算法具有较低的误差率、较高的收敛性和普遍的适用性等特性.

2 信念状态空间 (Belief states space)

当真实的世界状态无法直接获取时, 智能体的决策将不能够依赖状态来计算, 需要对世界模型观察信息进行建模, 利用概率分布进行决策. 一般情况下, POMDP 可以用一个 6 元组 $\langle S, A, T, R, Z, O \rangle$ 来描述, 其序贯决策示意图如图 1 所示.

- 状态集合 $S = \{s_1, s_2, \dots, s_n\}$, Agent 所有可能的状态.
- 动作集合 $A = \{a_1, a_2, \dots, a_m\}$, Agent 所有可能的动作.
- 观察集合 $Z = \{z_1, z_2, \dots, z_l\}$, Agent 所有可能的观察.

- 状态转移函数 $T(s_i, a, s_j) = P(s_i|a, s_j)$, 当在状态 s_i 下采用动作 a , 转移到状态 s_j 的概率.
- 观察函数 $O(s_i, a_j, z_k) = P(z_k|s_i, a_j)$, 当在状态 s_i 下采用动作 a_j , 观察为 z_k 的概率.
- 报酬函数 $R(s, a) : S \times A \times Z \mapsto R$.

在本文, 除特殊说明外, 上标代表时间, 下标代表集合中的具体实例, 如 s_i 代表状态集合中的第 i 个状态, s^t 代表 t 时刻的状态, $P(s^t = s_i)$ 代表 t 时刻状态为 s_i 时的概率; 没有上下标的表示当前时刻的变量, 上标是 ' 的表示下一时刻的变量, 如 s 表示当前时刻的状态, 而 s' 表示下一时刻的状态.

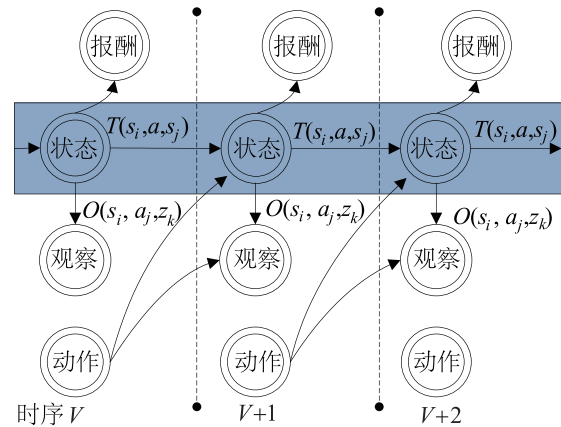


图 1 POMDP 序贯决策示意图

Fig.1 Schematic diagram of POMDP sequential decision-making

根据马尔可夫假设, 下一时刻的状态只与当前的状态有关, 与历史状态无关, 具有与历史无关的特性. 但是在 POMDP 中, 由于状态是未知的, 将来的状态和观察与历史的状态和观察不具有独立性. 图 1 中的灰色部分无法预知, t 时刻的状态可以通过式 (1) 计算:

$$P(s^t|a^t, z^t, a^{t-1}, z^{t-1}, \dots, a^0, z^0, s^0) \quad (1)$$

其中, $P(s^t)$ 表示 t 时刻状态的概率分布, $P(s^0)$ 是初始状态的概率分布. 式 (1) 的计算量会随着时序的增加而成指数规模增加.

为了继续使用马尔可夫特性, 必须对 POMDP 现有模型进行转换. 由于可以使用历史观察和历史动作的概率分布来描述状态空间, 因此状态可以使用历史观察和历史动作的联合概率分布来表示, 并将这一概率分布称为信念状态 (belief state), 整个信念状态的集合称之为信念状态空间 (belief state space). t 时刻的信念状态由式 (2) 来描述, 可由式 (3) 来计算, 其中 η 为归一化因子:

$$b^t = P(s^t|a^t, z^t, a^{t-1}, z^{t-1}, \dots, a^0, z^0, s^0) \quad (2)$$

$$b'(s_i) = P(s_i | z', a', s^{t-1}) \\ = \eta O(s_i, a', z') \sum_{j=1}^{|S|} T(s_j, a', s_i) P(s^{t-1} = s_j) \quad (3)$$

b' 可以用 b'^{-1} 来递归求解, 这样信念状态可以看成与当前时刻有关, 而与历史信念状态无关. 通过引入信念状态空间, POMDP 问题可以看成 belief MDP 问题^[9]. 寻求一种最优策略将当前的信念状态映射到 Agent 的动作上, 根据当前的信念状态和动作就可以决定下一个周期的信念状态和动作. 信念状态转移函数定义如下:

- 信念状态转移函数 $\Gamma(b, a, b') = P(b' | a, b)$.

式 (4) 描述的是在 b_i 状态下采用动作 a 转移到 b_j 的概率, 其中 Z_{b_j} 是在 b_j 状态下的观察集合, $Z_{b_j} \subseteq Z$, $z_k \subseteq Z_{b_j}$:

$$P(b_j | b_i, a) = \sum_{k=1}^{|Z_{b_j}|} P(z_k | b_i, a) \\ = \sum_{k=1}^{|Z_{b_j}|} \sum_{l=1}^{|S|} O(s_l, a, z_k) \sum_{m=1}^{|S|} T(s_m, a, s_l) b_i(s_m) \quad (4)$$

- 信念状态报酬函数 $\rho(b, a) = \sum_{s \in S} b(s) R(s, a)$.

POMDP 决策的策略是将信念状态映射到动作, 同时满足使 Agent 选择的动作能够获得环境报酬的累计值最大, 如式 (5):

$$E \left[\sum_{t=0}^T \gamma^t \rho_t \right], \gamma \in (0, 1] \quad (5)$$

其中 γ 为折扣因子, 其目标是让期望值收敛.

图 2 描述的是引入信念状态后的 POMDP 决策过程. 状态变成信念状态, 转移函数替换为 $\Gamma(b', a, b)$, 状态对观察的影响变为观察对信念状态的影响.

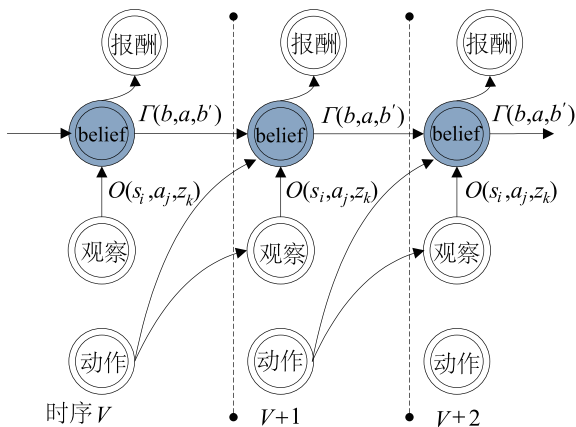


图 2 引入信念状态后的 POMDP 决策过程

Fig.2 The graphical model of POMDP using beliefs

在 MDP 中, 最优策略是从所有的状态中选取一个动作使得期望值最大. 状态空间为 $|S|$. 在 POMDP 中最优策略是从所有的信念状态中选取一个动作使得期望值最大. 信念状态空间为 $R^{|S|-1}$, 其中 R 的大小为 $|S| \cdot |A| \cdot |Z|$, 求解信念状态空间是一个 NP 难问题. 在实际应用中, $|S|$ 往往很大, 造成无法求解. 在 RoboCupRescue 机器人仿真系统^[10]中, 从警察的角度来看, 智能体的动作集合 $A = \{North, South, East, West, Clear\}$, 一个完整的状态由 1500 个随机变量描述, 其中 800 个 Road 变量, 取值为 $\{blocked, cleared\}$, 700 个 Buildings 变量, 取值为 $\{fire, noFire\}$, 智能体位置变量 AgentPosition, 它表示 30 ~ 40 个智能体可能处在 800 个道路上的任意一个位置. 警察智能体的状态空间 $|S| = 2^{800} \times 2^{700} \times 800^{\{30 \sim 40\}}$, 它的信念状态空间更大. 因此, 为求解部分可观察马尔可夫问题, 必须要先求解信念状态空间问题. 目前通用的方法是压缩信念状态空间, 降低求解规模的维数, 达到快速求解的目的.

在 RoboCupRescue 中, Agent 的决策并不是要对所有的道路和建筑等变量都要遍历求解, 而是取主要的、影响 Agent 决策的变量进行求解, 从而达到降低求解规模的目的. 如何选取主要变量, 使得 Agent 能够快速求解又不会影响求解结果的正确性是目前的难点和热点.

3 可分解信念状态空间 (Factored belief states space)

3.1 动态贝叶斯网络及其独立关系

在大多数真实世界模型中, 分析状态变量内部结构可以发现, 这些状态变量可以由一组随机变量集合来表示, 这种内部特性称为可分解特性^[11]. 可分解状态变量由一个随机有限变量集合 $X = \{X_1, \dots, X_n\}$ 来表示, 每一个 X_i 表示状态变量的一个特征, D_{X_i} 表示集合 X 中每个变量的取值集合, 一个状态可以表示成为 $s = \{X_1 = x_1, \dots, X_n = x_n\}$, 其中 $x_i \in D_{X_i}$, 也可以简化表示为 $s = \{x_i\}_{i=1}^n$, 状态变量的空间为 $|S| = \prod_{i=1}^n |D_{X_i}|$. 同理, 观察变量可分解为 $z = \{z_k\}_{k=1}^l$ 来描述, D_{z_k} 表示集合 Z 中每个变量的取值集合. 动作变量可分解为 $a = \{a_j\}_{j=1}^m$ 来描述, D_{a_m} 表示集合 A 中每个变量的取值集合. 状态变量可分解后, 状态转移函数、观察函数以及报酬函数都可以使用 DBN 来压缩. DBN 是一个有向无环图, 节点表示状态、动作和观察变量, 边 (或者弧表示概率依赖关系, 它利用条件独立关系图形化地描述随机决策过程.

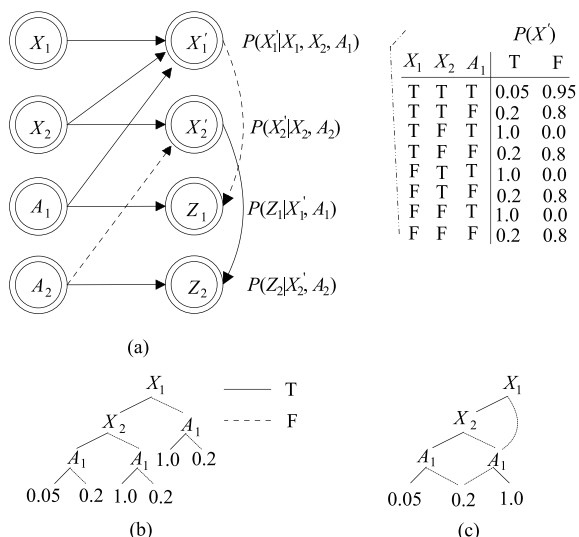


图3 FPOMDP的DBN描述

Fig.3 The description of FPOMDP using DBN

在图3(a)中,假设所有变量都是布尔类型,描述的是在 t 时刻到 $t+1$ 时刻转移函数的DBN表示形式. DBN中每个节点都伴随着一个条件概率表(conditional probability table, CPT)来描述其父亲节点对该节点的影响,条件概率分布为 $P(X'_i | \text{parents}(X'_i))$. CPT中参数的规模是以其父亲节点数为变量的指数形式,但实际的问题中,父亲节点数往往很少. 父亲节点可以是前一个时间片的,也可以是同一时间片的. X'_i 的父亲节点分别为 $\{X_1, X_2, A_1\}$,它们都属于前一个时间片. Z_1 的父亲节点为 $\{A_1, X'_1\}$,其中 A_1 属于前一个时间片, X'_1 属于当前时间片. 跨越2个时间片的边(或者弧)称之为时间弧,同一时间片的边(或者弧)称之为同步弧. 可分解部分可观察马尔可夫(factored POMDP, FPOMDP)具有3个独立特性:条件独立、上下文独立、因果影响独立. 3种独立关系旨在把联合概率分布分解成颗粒更小的因式,从而达到节省存储空间,简化知识获取和领域建模过程,降低推理复杂性的目的.

DBN中每个节点在已知其父亲节点的条件下独立于所有其余的非子孙节点,这种关系称之为条件独立关系. 利用条件独立, DBN把联合概率分解成若干个条件概率的乘积,每个条件概率涉及的变量参数较少,从而达到粒化的目的. 根据贝叶斯原理,转移函数可以被分解为多个小规模条件概率的乘积, $(X_1, \dots, X_n) = \prod_i P(X_i | \text{parents}(X_i))$. 在图3中, $P(X'_1, X'_2 | X_1, X_2, A_1, A_2)$ 被分解为 $P(X'_2 | X_2, A_2)$ 与 $P(X'_1 | X_1, X_2, A_1)$ 的乘积, $P(Z'_1, Z'_2 | X'_1, X'_2, A_1, A_2)$ 被分解为 $P(Z'_1 | X'_1, A_1)$ 与 $P(Z'_2 | X'_2, A_2)$ 的乘积. 在图3(a)

中, $|S||Z| = \prod_{i=1}^2 |D_{X_i}| \prod_{i=1}^2 |D_{Z_i}| = 2^4$, 转移函数为 $2^4 \times 2^4$ 的矩阵,参数为256个. 如果利用条件独立关系,转移函数的矩阵参数仅为40个.

在CPT中,当某些变量取特定值之后,其余变量之间相互独立,这种关系称之为上下文独立关系. DBN中每个节点都带有一张条件概率表,在 $\text{parents}(X_i)$ 的每种取值下给出 X_i 每一种取值的概率,即 $(X_i | \text{parents}(X_i))$. 条件概率表一方面需要的条件概率数目是父亲节点数目的指数幂,另一方面无法捕捉条件概率分布的某些规律. 在图3(a)中,当 $X_1 = T \wedge A_1 = F$ 时,不需要考虑 X_2 的取值, X_1, A_1 与 X_2 独立,这是上下文独立. DBN网状结构可以表达出条件独立,但对上下文独立无能为力. 描述上文独立关系的办法是构造决策树(decision tree, DT). 在图3(b)中,从CPT到DT,参数数量由8个变成6个,因而DT是CPT的一种紧凑表示. 更进一步,对于DT图可以使用代数决策图(algebraic decision diagram, ADD)进行再次压缩. 在图3(b)中,由于DT右子树取值相同,可以将DT图中两个右子树 A_1 进行合并. 又由于右叶子节点取值都为0.2,可以进行再次合并,最终变成图3(c),这样叶子节点就变成3个,大大降低了求解规模.

在 S, A, Z, T, O, R 等变量和转移函数没有可分解表示之前, $t+1$ 时刻变量的父亲节点为 t 时刻所有的变量,假设共有 n 个变量,并且都是布尔型的,那么每做一个动作需要遍历的参数规模为 $O(n^{2^n})$. 如果变量使用可分解表示,假设所有变量的父亲节点数都是 k ,并且 $k \ll n$,那么需要遍历的参数规模为 $O(n^{2^k})$. 如果利用变量的上下文独立关系,使用决策树或者代数决策图来表示,决策树或者代数决策图上的叶子节点数不会超过个 $f(k)$, $f(k)$ 代表变量父亲节点数的多项式函数,那么需要遍历的参数规模为 $O(nf(k))$.

3.2 可分解信念状态空间压缩算法

在POMDP中,信念状态空间的规模直接影响求解的速度和效果. 利用可分解特性同样可以对信念状态空间进行压缩. 为了便于说明问题,假设真实世界模型由4个变量 $\{X_1, X_2, X_3, X_4\}$ 描述,如图4所示. 图4(a)中,4个变量互相独立,每个变量取值的概率用一个列向量表示. 图4(b)中, X_3 与 X_4 相互依赖,它们的联合概率由一个列向量表示,而 X_1, X_2, X_3, X_4 之间相互独立. 在图4(a)和4(b)中, $|b| = |X_1| |X_2| |X_3| |X_4| = 5 \times 6 \times 5 \times 4 = 600$. 然而,当实际数学计算时,概率为0的取值可以忽略不计. 这样,图4(a)中实际的信念状态空间 $|b| = 32$,图

4(b) 中实际的信念状态空间 $|b| = 1 \times 2 \times 5 \times 4 = 40$. 在计算信念状态空间时, 将概率为 0 的变量裁剪掉, 可以减低信念状态空间的规模^[10].

$$b = \begin{pmatrix} X_1 & X_2 & X_3 & X_4 \\ \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 1.0 \\ 0.0 \end{bmatrix} & \begin{bmatrix} 0.2 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.8 \\ 0.0 \end{bmatrix} & \begin{bmatrix} 0.25 \\ 0.05 \\ 0.2 \\ 0.0 \\ 0.0 \\ 0.5 \end{bmatrix} & \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \end{bmatrix} \end{pmatrix} \quad b = \begin{pmatrix} X_1 & X_2 & X_3 & X_4 \\ \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 1.0 \\ 0.0 \end{bmatrix} & \begin{bmatrix} 0.2 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.8 \\ 0.0 \end{bmatrix} & \begin{bmatrix} P(X_3=x_1 \wedge X_4=x_1) \\ P(X_3=x_2 \wedge X_4=x_2) \\ \vdots \\ \vdots \\ \vdots \\ P(X_3=x_5 \wedge X_4=x_5) \end{bmatrix} \end{pmatrix}$$

(a)
(b)

图 4 可分解信念状态空间

Fig.4 Factored belief states space

对于给定的信念状态 b , 可能的状态 $\omega(b)$ 可以使用式 (6) 的一般化描述:

$$\omega(b) = \{\{x_i\}_{i=1}^n | (\forall x_i) P(X_i = x_i | b) \neq 0\} \quad (6)$$

这个函数的返回值为 Agent 所有有意义的状态, 在最坏情况下, $\omega(b) = S$. 但实际情况是 $|\omega(b)| \ll |S|$, 因此可以减少状态数, 从而降低信念状态空间的求解规模. 根据式 (7) 可以修改信念状态报酬函数:

$$\rho(b, a) = \sum_{s \in \omega(b)} b(s) R(s, a) \quad (7)$$

算法 1 可分解信念状态空间压缩算法

输入: 状态 S , 观察 Z , 动作 A

输出: $b_{\text{factored}}(s)$, 是函数 FBSSC(S, A, Z) 的返回值;

```

1: 根据实际问题合理分解  $S = \{X_1, \dots, X_n\}$ ,  $A = \{A_1, \dots, A_m\}$  和  $Z = \{Z_1, \dots, Z_l\}$ ;
2: 利用 DBN 的条件独立关系求解  $P(X_i | \text{parents}(X_i))$ ,  $P(Z_k | \text{parents}(Z_k))$  和  $P(A_j | \text{parents}(A_j))$ , 得到每个节点的 CPT;
3: for ( $i = 1$ ;  $i \leq n$ ;  $i++$ ) do
4:    $\omega(b) = \{\{x_i\}_{i=1}^n | (\forall x_i) P(X_i = x_i | b) \neq 0\}$ ;
5: end for
6: for ( $j = 1$ ;  $j \leq m$ ;  $j++$ ) do
7:    $\alpha(b) = \{\{a_j\}_{j=1}^m | (\forall a_j) P(A_j = a_j | b) \neq 0\}$ ;
8: end for
9: for ( $k = 1$ ;  $k \leq l$ ;  $k++$ ) do
10:   $\beta(b) = \{\{z_k\}_{k=1}^l | (\forall z_k) P(Z_k = z_k | b) \neq 0\}$ ;
11: end for
12:  $\omega'(b) = \{s' | (\forall s \in \omega(b)) T(s, a, s') \neq 0 \wedge O(s', a, z) \neq 0\}$ ;
13:  $O(s, a, z) = \sum_{s' \in \omega'(b)} O(s', a, z) \sum_{s \in \omega(b)} T(s, a, s') b(s)$ ;
14:  $b_{\text{factored}}(s) = \eta O(s, a, z) \sum_{s \in \omega(b)} T(s, a, s) b(s^{t-1})$ ;
15: return  $b_{\text{factored}}(s)$ ;

```

第 4、7 和 10 行分别对状态、动作和观察变量进行清零操作, 第 12 行计算下一时刻的状态, 第 13 行利用 $\omega(b)$ 和 $\omega'(b)$ 来压缩观察函数, 第 14 行

同理, 动作集合和观察集合的压缩描述为式 (8) 和式 (9). 其中, $\alpha(b)$ 是压缩后的动作集合, $\beta(b)$ 是压缩后的观察集合:

$$\alpha(b) = \{\{a_j\}_{j=1}^m | (\forall a_j) P(A_j = a_j | b) \neq 0\} \quad (8)$$

$$\beta(b) = \{\{z_k\}_{k=1}^l | (\forall z_k) P(Z_k = z_k | b) \neq 0\} \quad (9)$$

根据当前的状态 $s \in \omega(b)$, 利用式 (10) 可计算下一时刻的状态 $s' \in \omega'(b)$, $\omega'(b)$ 为下一时刻状态集合:

$$\omega'(b) = \{s' | (\forall s \in \omega(b)) (T(s, a, s') \neq 0) \wedge (O(s', a, z) \neq 0)\} \quad (10)$$

利用 $\omega(b)$ 和 $\omega'(b)$, 观察变量的概率也可以通过式 (11) 来压缩表示:

$$O(z | a, b) = \sum_{s' \in \omega'(b)} O(s', a, z) \sum_{s \in \omega(b)} T(s, a, s') b(s) \quad (11)$$

利用上述结果, 信念状态更新函数的压缩表示如式 (12) 所示, 其中下标 factored 表示经过可分解压缩后的信念状态空间:

$$b'_{\text{factored}}(s') = \eta O(s', a, z) \sum_{s \in \omega(b)} T(s, a, s') b(s) \quad (12)$$

可分解信念状态空间压缩算法描述如下:

算压缩后的信念状态空间. FBSSC 算法是一种无损压缩, 原信念状态空间为 $R^{|S|-1}$, 压缩后的信念状态空间大小为 $(|\omega(b)| \cdot |\alpha(b)| \cdot |\beta(b)|)^{|\omega(b)|-1}$. 在最坏情

况下(所有变量都相互依赖,所有变量的概率都不为0),压缩后的信念状态空间仍然为 $R^{|S|-1}$,但是在实际情况下,使用FBSSC算法可以大大降低原信念状态空间.

4 实验与仿真 (Experiments and simulation)

本文将从3个角度评价FBSSC算法的有效性.第1个是针对Hallway2问题采用不同的降维算法,评价它们的耗时.第2个是针对Tiger-Grid、Hallway、Hallway2和Tag等4个问题,评价不同降维算法的降维效果.第3个是针对RoboCupRescue仿真比赛,进行算法有效性比较.本文的仿真平台为Matlab R2010a,运行环境为32位Windows7, Intel(R) Core(TM) i3 CPU: 3.07 GHz, RAM 4GB.

4.1 Hallway2问题的对比实验

Hallway2描述的是一个机器人在迷宫内的导航问题,是POMDP算法的Benchmark平台.在Hallway2中,共有89个状态,其中22个房间变量,取值范围为4个方向 $\{south,north,west,east\}$,还有一个目标状态;5个动作;17个观察,每个状态下有4个观察变量 $\{osouth,onorth,owest,oeast\}$,每一个观察变量的取值为 $\{wall,nowall\}$,还有一个目标观察.根据图5可知,本文FBSSC压缩算法比较ONMF和EPCA压缩算法具有较高的收敛速度.降到相同的维数,本文算法明显省时.

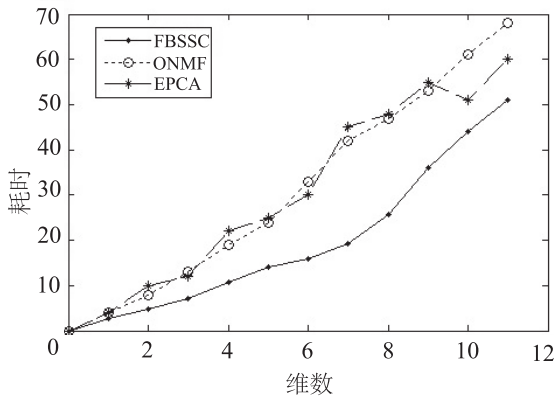


图5 3种算法降维耗时比较

Fig.5 Comparison on time cost among three algorithms

4.2 4种经典问题的对比实验

Tiger-Grid、Hallway、Hallway2和Tag问题都是迷宫问题,它们共同的特点是维度较高、稀疏性低,被普遍应用于检验可扩展的POMDP技术.本文采用Poupart等人^[12]开发的gapMin-2011-06-08软件包进行仿真实验.在每个问题的新旧方法上都进行10次实验,每次实验都会重新设置不同的随机种

子,最终对每个方法上的10次实验取平均值.折扣因子 $\gamma = 0.95$,每次探索在到达目标或超过预设的最大运行时间(1 000 s)时停止.在表1中,n.a.表示不可应用,n.v.表示不可获得.

表1 4类POMDP问题的对比实验
Tab.1 Comparison among four POMDP problems

问题	算法	Goal/(%)	B
Tiger-Grid S = 36 A = 5 O = 17	FBSSC	n.a.	120
	ONMF	n.a.	n.a.
	EPCA	n.a.	470
Hallway S = 61 A = 5 O = 21	FBSSC	100	29
	ONMF	47	n.a.
	EPCA	96	86
Hallway2 S = 89 A = 5 O = 17	FBSSC	99	36
	ONMF	22	n.a.
	EPCA	98	95
Tag S = 870 A = 5 O = 30	FBSSC	85	326
	ONMF	17	n.a.
	EPCA	59	1335

信念状态空间|B|的大小制约着算法收敛性和求解性能,从表1可知,本文算法在目标获得率和信念状态空间压缩上都有较明显的优势,特别是对于较大规模问题,本文算法的压缩效果更为明显.

4.3 RoboCupRescue仿真实验

RoboCupRescue是更为复杂的POMDP问题,其目标是尽可能地减少地震带来的破坏.衡量算法优劣的参数较多,例如营救的市民数量、着火点数量、受损房屋数量和被堵塞道路数量.本文以警察智能体清除道路释放被堵塞智能体数为评价目标,比较采用本文算法与未采用本文算法两种情况下的被堵塞智能体数.从图6可知,在采用本文算法后,被堵塞智能体数将在很短的时间内减少到3个.

5 结论 (Conclusion)

本文FBSSC算法根据信念状态空间稀疏性的特点,利用POMDP的可分解特性,对信念状态空间进行无损压缩.该算法与其他压缩算法相比具有以下特点:

- (1) 对概率为0的变量进行清零,不改变压缩后的信念状态空间对原问题的描述,是无损压缩.
- (2) FBSSC算法操作简单,便于实际应用.

因此, 本文算法可以广泛地应用在实际项目中. 对比实验和仿真实验验证了本文算法的有效性.

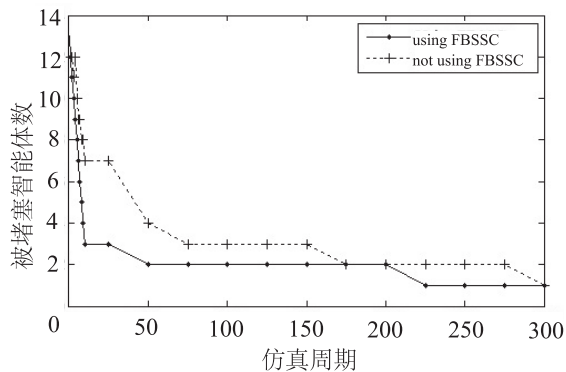


图 6 采用 FBSSC 算法前后被堵塞智能体数的比较

Fig.6 Comparison on numbers of blocked agents before and after using FBSSC

致谢: 感谢 Poupart 等人基于 Matlab 语言开发的 POMDP 软件包 (gapMin-2011-06-08), 在本文的实验中, 作者复用了其中的很多代码.

参考文献 (References)

- [1] Littman M L. A tutorial on partially observable Markov decision processes[J]. Journal of Mathematical Psychology, 2009, 53(3): 119-125.
- [2] Ross S, Pineau J, Chaibdraa B, et al. A Bayesian approach for learning and planning in partially observable Markov decision processes[J]. Journal of Machine Learning Research, 2011, 12(2): 1729-1770.
- [3] Kaelbling L P, Littman M L, Cassandra A R. Planning and acting in partially observable stochastic domains[J]. Artificial Intelligence, 1998, 101(2): 99-134.
- [4] Roy N, Gordon G. Finding approximate POMDP solutions through belief compression[J]. Journal of Artificial Intelligence Research, 2005, 23(9): 1-40.
- [5] Li X, Cheung W K, Liu J M. Improving POMDP tractability via belief compression and clustering[J]. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 2010, 40(1): 125-136.
- [6] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization[J]. Nature, 1999, 401(10): 788-791.
- [7] Sun J G, Crow M, Fyfe C. Extending metric multidimensional scaling with Bergman divergences[J]. Pattern Recognition, 2011, 44(5): 1137-1154.
- [8] 仵博, 吴敏. 一种基于信念状态压缩的实时 POMDP 算法[J]. 控制与决策, 2007, 22(12): 1417-1420.
- Wu B, Wu M. Real-time POMDP algorithm based on belief states space compression[J]. Control and Decision, 2007, 22(12): 1417-1420.
- [9] Roy N. Finding approximate POMDP solutions through belief compression[D]. Pittsburgh, USA: Carnegie Mellon University, 2003.
- [10] Paquet S. Distributed decision-making and task coordination in dynamic uncertain and real-time multiagent environments[D]. Quebec, Canada: Laval University, 2006.
- [11] Boutilier C, Dearden R, Goldszmidt M. Stochastic dynamic programming with factored representations[J]. Artificial Intelligence, 2000, 121(1/2): 49-107.
- [12] Poupart P, Kim K E, Kim D. Closing the gap: Improved bounds on optimal POMDP solutions[C]//Proceedings of the Twenty-First International Conference on Automated Planning and Scheduling. San Diego, CA, USA: AAAI, 2011: 194-201.

作者简介:

仵 博 (1979-), 男, 博士生, 副教授. 研究领域为序贯决策, 机器学习, 无线传感器网络.

吴 敏 (1963-), 男, 博士, 教授, 博士生导师. 研究领域为过程控制, 鲁棒控制, 智能系统.

(上接第 712 页)

- [17] 刘朝华, 张英杰, 章兢, 等. 基于免疫双态粒子群的混沌系统自抗扰控制 [J]. 物理学报, 2011, 60(1): 019501-1-019501-9.
- Liu Z H, Zhang Y J, Zhang J, et al. Active disturbance rejection control of a chaotic system based on immune binary-state particle swarm optimization algorithm[J]. Acta Physica Sinica, 2011, 60(1): 019501-1-019501-9.
- [18] 江全元, 邹振宇, 曹一家. 基于免疫遗传算法的多机电力系统 UPFC 控制器 [J]. 电工技术学报, 2006, 21(7): 60-68.
- Jiang Q Y, Zou Z Y, Cao Y J. UPFC in multi-machine power systems based on immune genetic algorithm[J]. Transactions of China Electrotechnical Society, 2006, 21(7): 60-68.
- [19] de Castro L N, Von Zuben F J. Learning and optimization using the clonal selection principle[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(3): 239-251.
- [19] Dasgupta D. Advances in artificial immune systems[J]. IEEE Computational Intelligence Magazine, 2006, 1(4): 40-49.

- [20] Juang C F. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2004, 34(2): 997-1006.
- [21] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295.

作者简介:

童成意 (1965-), 男, 硕士生, 副教授. 研究领域为嵌入式系统及其应用, 智能控制理论与应用, 电力系统与控制.

刘朝华 (1983-), 男, 博士. 研究领域为计算机控制技术, 智能控制理论与应用, 智能系统.