doi:10.3969/j.issn.1000-436x.2013.11.015

基于优先级扫描 Dyna 结构的贝叶斯 Q 学习方法

于俊[1]，刘全[1,2]，傅启明[1]，孙洪坤[1]，陈桂兴[1]
（1. 苏州大学 计算机科学与技术学院，江苏 苏州 215006；2. 吉林大学 符号计算与知识工程教育部重点实验室，吉林 长春 130012）

摘 要：贝叶斯 Q 学习方法使用概率分布来描述 Q 值的不确定性，并结合 Q 值分布来选择动作，以达到探索与利用的平衡。然而贝叶斯 Q 学习存在着收敛速度慢且收敛精度低的问题。针对上述问题，提出一种基于优先级扫描 Dyna 结构的贝叶斯 Q 学习方法—Dyna-PS-BayesQL。该方法主要分为 2 部分：在学习部分，对环境的状态迁移函数及奖赏函数建模，并使用贝叶斯 Q 学习更新动作值函数的参数；在规划部分，基于建立的模型，使用优先级扫描方法和动态规划方法对动作值函数进行规划更新，以提高对历史经验信息的利用，从而提升方法收敛速度及收敛精度。将 Dyna-PS-BayesQL 应用于链问题和迷宫导航问题，实验结果表明，该方法能较好地平衡探索与利用，且具有较优的收敛速度及收敛精度。

关键词：强化学习；马尔科夫决策过程；优先级扫描；Dyna 结构；贝叶斯 Q 学习
中图分类号：TP181 **文献标识码：**A **文章编号：**1000-436X(2013)11-0129-11

Bayesian Q learning method with Dyna architecture and prioritized sweeping

YU Jun[1], LIU Quan[1, 2], FU Qi-ming[1], SUN Hong-kun[1], CHEN Gui-xing[1]
(1. School of Computer Science and Technology, Soochow University, Suzhou 215006, China;
2. Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China)

Abstract: In order to balance this trade-off, a probability distribution was used in Bayesian Q learning method to describe the uncertainty of the Q value and choose actions with this distribution. But the slow convergence is a big problem for Bayesian Q-Learning. In allusion to the above problems, a novel B ian Q learning algorithm with Dyna architecture and prioritized sweeping, called Dyna-PS-BayesQL was proposed. The algorithm mainly includes two parts: in the learning part, it models the transition function and reward function according to collected samples, and update Q value function by Bayesian Q-learning, in the programming part, it updates the Q value function by using prioritized sweeping and dynamic programming methods based on the constructed model, which can improve the efficiency of using the historical information. Applying the Dyna-PS-BayesQL to the chain problem and maze navigation problem, the results show that the proposed algorithm can get a good performance of balancing the exploration and exploitation in the learning process, and get a better convergence performance.

Key words: reinforcement learning; Markov decision process; prioritized sweeping; Dyna architecture; Bayesian Q learning

收稿日期：2013-05-18；**修回日期：**2013-07-20
基金项目：国家自然科学基金资助项目(61070223, 61103045, 61070122, 61272005)；江苏省自然科学基金资助项目(BK2012616)；江苏省高校自然科学研究基金资助项目(09KJA520002, 09KJB520012)；吉林大学符号计算与知识工程教育部重点实验室基金资助项目(93K172012K04)
Foundation Items: The National Natural Science Foundation of China(61070223, 61103045, 61070122, 61272005); The Natural Science Foundation of Jiangsu Province(BK2012616); The High School Natural Foundation of Jiangsu Province(09KJA520002, 09KJB520012); The Foundation of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University(93K172012K04)

1 引言

强化学习又称为增强学习、再励学习或激励学习，是一类学习环境状态到动作的映射方法。在与环境的交互学习中，agent选择动作，并作用于环境，环境对此做出响应，产生新的场景，同时给出评价性反馈信号。该评价性反馈信号通常称为奖赏或强化信号，agent的目标就是极大化期望累积奖赏值[1]。强化学习的一个基本特征是agent与环境的交互，agent不断探索环境和感知奖赏，利用获得的奖赏实现序贯决策的优化，因此与监督学习和无监督学习比较，在解决序贯决策优化方面，强化学习有着很大的优势[2~4]。

马尔科夫决策过程(MDP, Markov decision process)是研究一类具有随机动态系统的最优序贯决策问题，即最优化问题。MDP经常用来对强化学习进行建模，系统在每个时间步所处的状态是随机的，从当前状态按照一定的概率迁移到下一状态，并且下一状态仅仅取决于当前的状态和迁移概率，与以前的状态无关，即无后效性。状态的转移规律与选用的动作，两者交互作用决定系统的发展进程[5]。基于强化学习的基本框架，Watkins于1989年在其博士学位论文中首次提出Q学习算法，Q学习算法以求解具有延迟回报的序贯决策优化问题为目标，并且成为强化学习的经典算法之一[6]。

Sutton于1990年提出Dyna结构，agent每次与环境交互都会产生真实经验，利用真实经验进行在线学习的同时，也将利用获得的环境知识来完善环境的模型，再通过模型产生的模拟经验来规划更新状态的值函数[7]。当所建立的模型趋近于真实环境时，能够加快算法的收敛速度。近年来，Sutton Szepesvári等人将Dyna结构进行扩展，并与线性函数逼近相结合，且在此基础之上，证明算法能够收敛到一个唯一解[8]。优先级扫描思想是基于状态动作对的重要程度来确定优先级的，以优先级的顺序来更新值函数，并利用回溯方法来更新其所有前继状态动作对的优先级[9]。优先级扫描能够在有限的样本情况下，对值函数进行较好的估计。

强化学习需要解决的重要难点之一是平衡探索与利用(exploration-and-exploitation)，即使用已知最优动作还是探索未测试动作[10]。在任意状态下，动作的选择对算法性能的表现都有着非常重要的影响，agent需要根据先前知识选择执行的动作，常用的平衡探索与利用的方法有：Semi-uniform方法和Boltzmann方法[11]。Semi-uniform方法是贪心策略的延伸，以较大概率选择最优动作（利用），而以较小的概率随机选择动作（探索），Semi-uniform方法的一个缺点是在探索时，所有的动作是以等概率的形式被选择的。Boltzmann方法也涉及到概率，它将动作的选择与值函数联系在一起，利用温度参数调整动作的选择概率（温度参数的初始值可以设置得较高以提高探索，慢慢地降低温度以减少探索，提高利用）。Boltzmann方法的一个缺点是温度参数的初始值设定是不确定的。较复杂的方法有区间估计方法(interval estimation) [12,13]，Myopic-VPI (myopic value of perfect information)方法[14,15]。区间估计方法和Myopic-VPI方法可以归纳为动作的选择是基于需要估计的状态值再加上额外的探索奖赏，在这一点上有些类似于塑造奖赏[16]。区间估计方法的额外探索奖赏设为置信区间宽度的一半，它的缺点是探索奖赏的确定仅仅与该状态的状态值有关。Myopic-VPI方法的额外探索奖赏通过计算价值增益的方法来得到，比较探索可获得的预期收益与采用已知最优动作可获得的预期收益来帮助选择动作。

本文针对强化学习中的探索与利用的平衡问题，提出一种基于优先级扫描Dyna结构的贝叶斯Q学习方法——Dyna-PS-BayesQL (Bayesian Q learning method with Dyna architecture and prioritized sweeping)。该方法在学习部分利用贝叶斯Q学习进行在线学习，即抽样学习，并对环境动态性进行建模；在规划部分，利用优先级扫描和动态规划对需要估计的状态动作对的值函数进行规划更新，以改进Dyna结构规划部分。在整个方法中，利用Myopic-VPI方法来优化动作选择。将Dyna-PS-BayesQL应用在链问题(chain problem)和迷宫导航问题(maze problem)中，实验结果表明，基于优先级扫描Dyna结构的贝叶斯Q学习方法能较好地平衡探索与利用，且有较优的收敛速度和精确度。

2 相关理论

2.1 马尔科夫决策过程

在强化学习中，agent与环境交互的过程通常被模型化为马尔科夫决策过程(MDP)或部分可感知马尔科夫决策过程(POMDP)。本文以随机可数马尔科夫决策过程为研究对象，随机可数马尔科夫决策可

以用一个五元组来表示 $M=<X,U,f,r,g>$，X 是环境的可数状态集合，如离散状态集合；U 是 agent 能采取的动作集合；f：$X\times U\times X\to[0,1]$ 是状态转移函数，它对后继状态的不确定性进行了模型化，表示为在离散时间步 t，状态 x_t 下执行动作 u_t 到达下一状态 x_{t+1} 的概率，即 $f(x,u,x')=Pr\{x_{t+1}=x'\mid x_t=x,u_t=u\}$，对于任意的状态 x 和动作 u，函数 f 必定满足 $\sum_{x'}f(x,u,x')=1$；$r:X\times U\times X\to j$，是关于状态动作对的立即奖赏函数，$r(x,u,x')$ 表示在状态 x 处，agent 执行动作 u，到达后继状态 x' 获得的奖赏值；$g\in[0,1)$ 是折扣率。MDP 的本质是：当前状态向下一个状态迁移的概率和奖赏只取决于当前状态和选择的动作，而与历史状态、动作无关。

当强化学习问题满足 MDP 框架时，通常用累积折扣奖赏来表示值函数。累积折扣奖赏是从当前时间步开始将后续时间步的立即奖赏折扣相加的总和。形式上，设 r_t 是在时间步 t 获得的奖赏，g 是折扣，则累积折扣奖赏可以表示为 $r_t+gr_{t+1}+g^2r_{t+2}+\text{L}$。在 Q-学习算法中，动作值函数的更新公式为 $Q(x,u)=Q(x,u)+a[r+g\max_{u'}Q(x',u')-Q(x,u)]$，其中，$a$ 为步长参数。当环境模型已知时，可以给出更加精确的动作值函数更新公式，即 $Q(x,u)=r(x,u)+g\sum_{x'\in X}f(x,u,x')\max_{u'\in U}Q^*(x',u')$。

2.2　Dyna 结构以及优先级扫描

Dyna 结构的强化学习算法整合了学习和规划[17]。学习是基于 agent 与当前环境交互获得的真实经验，而规划是基于当前所学习到的环境模型，两者并行执行，agent 高效地利用获得的经验来构建和完善模型，并且利用此模型产生模拟经验来进行规划，以达到提升状态的值函数和策略的收敛速度。学习和规划的核心都是更新值函数。算法 1 给出了 Dyna-Q 学习算法的一般流程。

算法 1　Dyna-Q 学习算法

1) 初始化 $\text{Q}(x,u)$，$\text{Model}(x,u)$；

2) Repeat：

step1　选择状态 x 以及选择状态 x 的动作 u，(如根据 Semi-uniform 方法)；

step2　执行动作 u，观察下一状态 x' 和立即奖赏 r；

step3　$\text{Q}(x,u)\leftarrow\text{Q}(x,u)+a[r+g\max_{u'}\text{Q}(x',u')-\text{Q}(x,u)]$；

step4　$\text{Model}(x,u)\leftarrow x',r$；

step5　重复 N 次

① $x\leftarrow$ 随机先前观察到的状态；

② $u\leftarrow$ 随机先前在 x 上采取的动作；

③ $x',r\leftarrow\text{Model}(x,u)$；

④ $\text{Q}(x,u)\leftarrow\text{Q}(x,u)+a[r+0g\max_{u'}\text{Q}(x',u')-\text{Q}(x,u)]$；

3) Until 到达目标或训练结束。

Dyna 结构在规划部分是随机地选取先前观察到的状态进行规划，而这种随机的规划方式通常会降低算法的收敛速度。优先级扫描则根据状态的重要程度来确定状态值函数的更新顺序，其基本思想是在一个队列中存放着更新值可能发生较大变化的状态动作对，并以变化的大小进行优先级排序。当队列最上面的状态动作对的值函数被更新时，它前继的每一个状态动作对的优先级也将被更新。如果这个变化大于设定的阈值，那么这个状态动作对就以新的优先级插入队列。通过这种方式，可以对状态值函数进行有效更新，并同步更新相关状态动作对的优先级，加快算法的收敛速度。

2.3　贝叶斯推理

贝叶斯推理通常用来解决学习参数未知的概率模型问题。贝叶斯推理提供了推理的一种概率手段，它基于如下的假定：待考查的量遵循某概率分布，且可根据这些概率以及观察到的样本数据利用贝叶斯规则进行推理，以做出最优的决策。贝叶斯的基本观点是：任一未知量 q 都可以看作随机变量，可用一个概率分布来描述，这个分布称为先验分布；在获得样本之后，总体分布、样本与先验分布通过贝叶斯公式计算结合起来得到一个关于未知量 q 的新分布，即后验分布；任何关于 q 的统计推断都应该基于 q 的后验分布进行。

在经典统计中，随机变量 X 依赖于参数 q 的概率函数记为 $p(x;q)$，它表示参数空间 Q 中不同的 q 对应于不同的分布。在贝叶斯推理中，记为 $p(x\mid q)$，它表示随机变量 q 取某个定值时总体的条件概率函数。根据参数 q 的先验信息确定先验分布 $\text{p}(q)$。为了产生样本 $X=(x_1,\text{L},x_n)$，首先从先验分布 $\text{p}(q)$ 产生一个样本 q_0，然后从 $p(x\mid q_0)$ 产生一组样本，这时样本 $X=(x_1,\text{L},x_n)$ 的联合条件概率函数为 $p(X\mid q_0)=p(x_1,\text{L},x_n\mid q_0)=\prod_{i=1}^{n}p(x_i\mid q_0)$。为了把先验信息综合进去，因此不能只考虑 q_0，对 q 的其他值发生的可能性也要加以考虑，所以利用样本和参

数 q 的联合分布 $h(X,q)=p(X|q)\mathrm{p}(q)$，由此可以积分得到 X 的边际概率函数 $m(X)=\int_Q h(X,q)\mathrm{d}q=\int_Q p(X|q)\mathrm{p}(q)\mathrm{d}q$。根据贝叶斯规则，参数 q 的后验 $\mathrm{p}(q|X)=\dfrac{h(X,q)}{m(X)}=\dfrac{p(X|q)\mathrm{p}(q)}{\int_Q p(X|q)\mathrm{p}(q)\mathrm{d}q}$。后验分布是利用总体分布、样本对先验分布 $\mathrm{p}(q)$ 作调整的结果，它要比 $\mathrm{p}(q)$ 更接近 q 的实际情况。

3 Dyna 结构的贝叶斯 Q 学习算法及分析

3.1 贝叶斯 Q 学习

贝叶斯 Q 学习是对 Q 学习的一个扩展，由 Dearden 等人提出，用概率分布表示每个状态动作对的 Q 值，因此可以得到更优的动作选择策略。利用 Q 值的分布来计算每个状态动作对的信息价值增益(VPI，value of perfect information)，即通过信息价值增益和 Q 值期望来改进动作选择的策略，以平衡探索和利用的问题。在贝叶斯框架中，需要考虑在 Q 值上的先验分布，agent 通过与环境的交互获得经验，并更新这些先验，使用贝叶斯 Q 学习方法需要一些前提条件，形式上，$R_{x,u}$ 是一个随机变量，代表在状态 x 下执行动作 u 接受到的累积折扣奖赏，$Q^*(x,u)=\mathrm{E}[R_{x,u}]$ 就是需要逼近的动作值函数。

1) $R_{x,u}$ 是满足高斯分布的一个随机变量，$R_{x,u}$ 的均值为 $m_{x,u}$，$m_{x,u}=Q^*(x,u)$，方差为 $s^{-2}_{x,u}$，$R_{x,u}$ 的精度为 $t_{x,u}=1/s^{-2}_{x,u}$。为了方便，用均值 $m_{x,u}$ 和精度 $t_{x,u}$ 来表示 $R_{x,u}$ 的参数，则概率密度函数为 $p(y|m_{x,u},t_{x,u})=\left(\dfrac{t_{x,u}}{2\mathrm{p}}\right)^{1/2}\exp\left[-\dfrac{1}{2}t_{x,u}(y-m_{x,u})^2\right]$。类似地，如果 $Y=y_1,y_2,\mathrm{L},y_n$ 是 n 个关于 $R_{x,u}$ 的独立样本，则它们的联合概率密度函数为 $p(Y|m_{x,u},t_{x,u})=\left(\dfrac{t_{x,u}}{2\mathrm{p}}\right)^{n/2}\exp\left[-\dfrac{1}{2}t_{x,u}\sum_{i=1}^{n}(y_i-m_{x,u})^2\right]$。

2) 对于参数未知的高斯分布的变量 $R_{x,u}$，$R_{x,u}$ 参数的先验 $p(m_{x,u},t_{x,u})$ 服从高斯伽玛分布[18]，即给定 $t_{x,u}$ 值时，$m_{x,u}$ 的条件分布是均值为 m_0，精度为 $l_0t_{x,u}$ 的高斯分布，条件概率密度函数为 $\mathrm{p}_1(m_{x,u}|t_{x,u})\propto t_{x,u}^{1/2}\exp\left[-\dfrac{1}{2}l_0t_{x,u}(m_{x,u}-m_0)^2\right]$；关于 $t_{x,u}$ 的边际分布是参数为 a_0 和 b_0 的伽玛分布，边际概率密度函数为 $\mathrm{p}_2(t_{x,u})\propto t_{x,u}^{a_0-1}\mathrm{e}^{-b_0t_{x,u}}$。根据贝叶斯推理，$m_{x,u}$ 和 $t_{x,u}$ 联合后验密度函数满足式(1)

$$\mathrm{p}(m_{x,u},t_{x,u}|Y)\propto p(Y|m_{x,u},t_{x,u})\mathrm{p}_1(m_{x,u}|t_{x,u})\mathrm{p}_2(t_{x,u})\propto t_{x,u}^{a_0+(n+1)/2-1}\left[-\frac{t_{x,u}}{2}\left(\begin{matrix}l_0[m_{x,u}-m_0]^2+\\ \sum_{i=1}^{n}(y_i-m_{x,u})^2\end{matrix}\right)-b_0t_{x,u}\right] \quad (1)$$

用四元组参数来表示高斯伽玛分布，$p(m_{x,u},t_{x,u})$：$NG(m_0,l_0,a_0,b_0)$。高斯伽玛分布隶属共轭家族，所以给定一个高斯伽玛分布先验，在任意个独立样本序列之后，它的后验分布仍然是高斯伽玛分布。

3) 当 $x\neq x'$ 或 $u'\neq u$ 时，均值为 $m_{x,u}$，精度为 $t_{x,u}$ 的先验分布独立于均值为 $m_{x',u'}$，精度为 $t_{x',u'}$ 的先验分布，且 $m_{x,u}$ 和 $t_{x,u}$ 的后验分布是独立于 $m_{x',u'}$ 和 $t_{x',u'}$ 的后验分布。根据 Bellman 公式，先后 2 个状态动作对的值函数的先验及后验是相互关联的，但为了求解的方便，仍然假设上述关系成立。

贝叶斯 Q 学习不同于普通的 Q 学习，对于每一个状态动作对 (x,u)，存储的是一组四元组 (m_0,l_0,a_0,b_0)，而不是存储的点估计 Q 值，即用一个分布对 Q 值进行描述，并结合信息价值增益选择动作。

Myopic-VPI 动作选择方法的主要思想是比较探索可获得的预期收益与采用已知最优动作可获得的预期收益来选择策略。所以只有能够改变 agent 行为策略的新知识才有意义。新知识改变策略的情况有 2 种：1）之前被认为是次优的动作被新知识揭示为最优动作，即执行非最优动作，发现该动作比最优动作好。假设 u_1 是之前的最优的动作，即对于所有的其他动作 u' 有 $\mathrm{E}(m_{x,u_1})\geq\mathrm{E}(m_{x,u'})$，但新知识表明 u 是更好动作，$m^*_{x,u}\geq\mathrm{E}(m_{x,u_1})$，这种情况下价值增益的值可以用 $m^*_{x,u}-\mathrm{E}(m_{x,u_1})$ 来衡量。2）执行最优动作，但新知识表明最优动作在当前策略下并不是最优，那么可以得到当前的最优动作为之前的第二优动作。假设 u_1 是之前最优的动作，u_2 是之前第二优动作，$\mathrm{E}(m_{x,u_1})\geq\mathrm{E}(m_{x,u_2})$，但新知识表明 u_1 不再是最优动作，那么得到 $\mathrm{E}(m_{x,u_2})\geq m^*_{x,u}$，在这种情况下价值增益的值可以用 $\mathrm{E}(m_{x,u_2})-m^*_{x,u}$ 来衡量。Dearden 等人定义了学习真实 $m^*_{x,u}$ 所获得的价值增益，如式(2)所示。

$$\mathrm{gain}_{x,u}(m^*_{x,u})=\begin{cases} m^*_{x,u}-\mathrm{E}(m_{x,u_1}), \\ u\neq u_1, m^*_{x,u}>\mathrm{E}(m_{x,u_1}) \\ \mathrm{E}(m_{x,u_2})-m^*_{x,u}, \\ u=u_1, m^*_{x,u}<\mathrm{E}(m_{x,u_2}) \\ 0, \text{其他} \end{cases} \tag{2}$$

其中，u_1 是之前策略的最优动作，u_2 是之前策略的第二优动作。因为 agent 事先并不知道 $m^*_{x,u}$ 的值，所以对当前动作的信息价值增益的计算转化为计算式(3)。

$$VPI(x,u)=\int_{-\infty}^{\infty}\mathrm{Gain}_{x,u}(m)\Pr(m_{x,u}=m)\mathrm{d}m \tag{3}$$

$VPI(x,u)$ 的计算十分复杂，因为 $m_{x,u}$ 先验是用高斯伽玛分布描述的，所以利用由 Teacy，Chalkiadakis 等人提出的截断偏差函数将积分计算转换成易于求解的形式[19]。

式(3)积分等式计算分为以下 2 种情况。

1) 当 $u=u_1$ 时，

$$VPI(x,u)=\mathrm{B}_r(\mathrm{E}[m_{x,u_2}])+(\mathrm{E}[m_{x,u_2}]-\mathrm{E}[m_{x,u_1}])\cdot \Pr(m_{x,u_1}\mid m_{x,u_1}<\mathrm{E}[m_{x,u_2}])$$

2) 当 $u\neq u_1$ 时，

$$VPI(x,u)=\mathrm{B}_r(\mathrm{E}[m_{x,u_1}])+(\mathrm{E}[m_{x,u}]-\mathrm{E}[m_{x,u_1}])\cdot \Pr(m_{x,u}\mid m_{x,u}>\mathrm{E}[m_{x,u_1}])$$

其中：

$$\mathrm{B}_r(x)=\frac{\mathrm{G}(a_0-1/2)\sqrt{b_0}\,(1+\frac{l_0(x-m_0)^2}{2b_0})^{-a_0+1/2}}{\mathrm{G}(a_0)\mathrm{G}(1/2)\sqrt{2l_0}}$$

$$\Pr(m<x)=T\left((x-m_0)\left(\frac{l_0a_0}{b_0}\right)^{\frac{1}{2}}:2a_0\right)$$

其中，$T(x:d)$ 是自由度为 d 的累积 t 分布。

为了计算状态 x 下选择的动作，设在状态 x 处执行探索动作 u 所需的花费为状态 x 下最优的动作值与采取的动作 u 值的差，即 $\max_{u'}\mathrm{E}[Q(x,u')]-\mathrm{E}[Q(x,u)]$，也就是说，选择能够使得 $VPI(x,u)-(\max_{u'}\mathrm{E}[Q(x,u')]-\mathrm{E}[Q(x,u)])$ 最大的动作，可以看出该策略的动作就是使 $\mathrm{E}[Q(x,u)]+VPI(x,u)$ 最大的动作，即应该选取的动作 $u=\arg\max_u(\mathrm{E}[Q(x,u)]+VPI(x,u))$。

3.2　Dyna-PS-BayesQL 算法

Dyna-PS-BayesQL 算法的详细描述如算法 2 所示，在学习部分，利用贝叶斯 Q 学习进行在线学习，对其环境进行动态建立模型，记录迁移概率 $f(x,u,x')$，奖赏函数 $r(x,u)$。在规划部分，对 Dyna 结构的规划部分进行了改进，利用优先级扫描和动态规划方法对需要估计的值函数进行规划更新。

算法 2　Dyna-PS-BayesQL 算法

1) 初始化每个状态动作对参数 (m_0,l_0,a_0,b_0) 的值，迁移函数 $f(x,u,x')$ 以及奖赏函数 $r(x,u)$ 置零，队列 *PQueue* 清空；

2) Repeat：

step1　选择状态 x；

step2　根据 Myopic-VPI 动作选择方法选择状态 x 的动作 u；

step3　执行动作 u，观察下一状态 x' 和立即奖赏 r；

step4　根据状态的迁移 (x,u,x',r) 更新迁移函数 $f(x,u,x')$ 以及奖赏函数 $r(x,u)$；

step5　根据贪心策略，选出状态 x' 的当前最优动作 u'；

step6　根据 x',u' 的高斯函数随机出 n 个后续状态的样本回报值，$R^1_{x',u'},R^2_{x',u'},\cdots,R^n_{x',u'}$；

step7　根据立即奖赏 r 和后继状态样本回报，计算 v_1,v_2；

step8　更新状态动作对 (x,u) 的高斯伽玛参数值 (m_0,l_0,a_0,b_0)；

step9　$p\leftarrow|r(x,u)+g\sum_{x'\in X}f(x,u,x')*\max_{u'\in U}m(x',u')-m(x,u)|$;

step10　如果 $p>q$，则将 (x,u) 插入到 *PQueue* 中，赋予优先级 p；

step11　重复 N 次，且 *PQueue* 非空，

① 从 *PQueue* 中选出使优先级最大的状态动作对 (x,u)，并且出队；

② $m(x,u)=m(x,u)+a[r(x,u)+g\sum_{x'\in X}f(x,u,x')\max_{u'\in U}m(x',u')-m(x,u)]$;

③ 重复，对于所有能够迁移到 x 的状态动作对 $(\bar{x},\bar{u})$；$p\leftarrow|r(\bar{x},\bar{u})+g\sum_{x\in X}f(\bar{x},\bar{u},x)\max_{u\in U}m(x,u)-m(\bar{x},\bar{u})|$; 如果 $p>q$，则将 $(\bar{x},\bar{u})$ 插入到 *PQueue* 中，赋予优先级 p；

3) Until 到达目标或训练结束。

算法 2 的 step1、step2、step3 用来产生一个 MDP 样本，在 step4 中，根据真实经验样本 (x,u,x',r) 更

新迁移概率 $f(x,u,x')$ 和 $r(x,u)$，来构建和完善模型，为算法的规划部分提供最新的模型。在 step5 中使用贪心策略选择出后继状态 x' 的最优动作 u'。

在 step6 中，设 $R^1_{x',u'}, R^2_{x',u'}, \cdots, R^n_{x',u'}$ 是由后继状态动作对 (x',u') 的高斯函数随机选出 n 个样本回报值，根据矩估计方法对值函数分布的参数进行更新。假设当前的立即奖赏为 r，且从相应的高斯分布中随机取出 n 个后续状态 x' 的样本回报值为 $R^1_{x',u'}, R^2_{x',u'}, \cdots, R^n_{x',u'}$，其中，$u'$ 表示在状态 x' 下的最优动作。根据 Bellman 公式，更新 $R_{x,u}$ 的四元组参数的样本值为 $r+gR^1_{x',u'}, r+gR^2_{x',u'}, \cdots, r+gR^n_{x',u'}$。则

$$v_1 = \mathrm{E}[r+gR_{x',u'}] = r+g\mathrm{E}[R_{x',u'}]$$

$$v_2 = \mathrm{E}[(r+gR_{x',u'})^2] = r^2+2gr\mathrm{E}[R_{x',u'}]+g^2\mathrm{E}[R^2_{x',u'}]$$

根据定理 1，更新状态动作对 (x,u) 的高斯伽玛分布的参数，即 m_0, l_0, a_0, b_0。

定理 1 假设 $x_1, x_2, \cdots, x_n$ 是 n 个独立样本，$p(m,t): NG(m_0, l_0, a_0, b_0)$ 即给定 t 值时 m 的条件分布是均值为 m_0 精度为 $l_0 t$ 的高斯分布 $(l_0>0)$，t 的边际分布是参数为 a_0 和 b_0 的伽玛分布 $(a_0>0$ 且 $b_0>0)$，则

$$\begin{cases} m_1 = \dfrac{l_0 m_0 + n\bar{x}}{l_0+n} \\ l_1 = l_0 + n \\ a_1 = a_0 + \dfrac{n}{2} \\ b_1 = b_0 + \dfrac{1}{2}\sum_{i=1}^{n}(x_i-\bar{x})^2 + \dfrac{nl_0(\bar{x}-m_0)^2}{2(l_0+n)} \end{cases} \tag{4}$$

其中，$\bar{x} = \dfrac{1}{n}\sum_{i=1}^{n} x_i$。

为了便于计算，将式(4)转化为与一阶原点矩，二阶原点矩相关的式子。

证明 设一阶原点矩 $v_1 = \dfrac{1}{n}\sum_{i=1}^{n} x_i$，二阶原点矩 $v_2 = \dfrac{1}{n}\sum_{i=1}^{n} x_i^2$，因此

$$\begin{aligned} \sum_{i=1}^{n}(x_i-\bar{x})^2 &= \sum_{i=1}^{n}(x_i - \frac{1}{n}\sum_{i=1}^{n} x_i)^2 \\ &= \sum_{i=1}^{n}(x_i^2 - \frac{2}{n}x_i\sum_{i=1}^{n}x_i + \frac{1}{n^2}(\sum_{i=1}^{n}x_i)^2) \\ &= \sum_{i=1}^{n} x_i^2 - \frac{2}{n}\sum_{i=1}^{n}x_i\sum_{i=1}^{n}x_i + \frac{1}{n}(\sum_{i=1}^{n}x_i)^2 \\ &= \sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n}x_i)^2 \\ &= nv_2 - nv_1^2 \\ &= n(v_2 - v_1^2) \end{aligned}$$

所以，对 (m_1, l_1, a_1, b_1) 的更新可以转化为式(5)

$$\begin{cases} m_1 = \dfrac{l_0 m_0 + nv_1}{l_0+n} \\ l_1 = l_0 + n \\ a_1 = a_0 + \dfrac{n}{2} \\ b_1 = b_0 + \dfrac{1}{2}n(v_2 - v_1^2) + \dfrac{nl_0(v_1-m_0)^2}{2(l_0+n)} \end{cases} \tag{5}$$

在 step9~step11 中，将 Bellman 误差作为优先级排序的标准。优先级函数的值越大，则状态的值函数的变化越大，对该状态进行更新的意义就越重要，在下次扫描中就优先更新这些状态。每次扫描时，优先对靠前的状态进行规划更新。这样改进 Dyna 规划部分，只对重要的状态进行更新，提升收敛的速度；而不像原始的 Dyna 结构，随机进行规划更新，效率比较低。

3.3 Dyna-PS-BayesQL 算法分析

Dyna-PS-BayesQL 算法收敛性取决于均值 $m_{x,u}$ 能否收敛到真实的 Q 值，均值的方差 $\mathrm{Var}[m_{x,u}]$ 能否收敛于 0。如果这两点得到证明，那么基于优先级扫描 Dyna 结构的贝叶斯 Q 学习算法的收敛也就得到了证明。

定理 2 对于有限状态和动作空间的 MDP，当满足如下条件时：

1) 动作选择策略保证算法对状态和动作空间进行无限遍历；

2) 学习因子满足随机逼近的收敛性条件，即

$$0 \le a_t \le 1, \quad \sum_{t=0}^{\infty} a_t = \infty, \quad \sum_{t=0}^{\infty} a_t^2 < \infty$$

表格型 Q 学习算法的动作值函数估计将以概率 1 收敛到 MDP 的最优动作值函数。定理 2 已由 Watkins 和 Dayan 证明，可参见文献[4]。

Dyna-PS-BayesQL 算法中，使用的是矩估计方法。矩估计方法的本质是用样本矩去替换总体矩，用样本矩的函数去替换相应的总体矩的函数。对于参数未知的情况下，可以用样本均值估计总体均

值，用样本方差，估计总体方差。

定理 3　在每种状态下如果每个动作被尝试无限次，那么使用矩估计方法的 Dyna-PS-BayesQL 算法在状态动作对 (x,u) 的均值 $m_{x,u}$ 将收敛到状态动作对 (x,u) 的真实 Q 值，状态动作对 (x,u) 均值的方差 $\mathrm{var}[m_{x,u}]$ 将收敛到 0。

证明　设总体为 X 且方差存在，$X_1, X_2, \mathrm{L}, X_n$ 是独立同分布于总体的一组随机样本，因为这些样本值都是独立同分布的，且有一个共同上界，根据大数定律可知，当 $n \to +\infty$ 时，这些样本的均值将收敛到它们的期望值，而在 Dyna-PS-BayesQL 算法中期望值就是真实 Q 值。对于样本均值的方差有 $\mathrm{var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{var}(X_i) \le \frac{c}{n}$，其中，常数 c 是 $\mathrm{var}(X_i)$ 的上界，当 $n \to +\infty$ 时，由夹逼定理可得，均值的方差收敛于 0。

为了保证使用 Myopic-VPI 动作选择方法进行动作选择时，每个动作都能被选中，在进行动作选择时引入了一些随机因素，例如以较大概率 $(1-e)$ 利用 Myopic-VPI 方法进行动作选择，而以较小的概率 e 随机选择动作，以达到遍历每个状态动作对。

综合使用上面定理 2、定理 3 以及引入的随机因素即可证明 Dyna-PS-BayesQL 算法能够收敛，且收敛到最优 Q 值函数。

4　实验及结果分析

为了验证 Dyna-PS-BayesQL 算法性能，将算法应用在链问题和迷宫导航问题中，在每个问题上，将 Dyna-PS-BayesQL 算法与 Q 学习 semi-uniform 算法、Q 学习 Boltzmann 算法以及贝叶斯 Q 学习的 Myopic-VPI 算法进行了比较。

4.1　链问题

链问题包含了 5 个状态，在每个状态下 agent 有 2 个动作 a 和 b 可以选择，如图 1 所示。状态 1 为初始状态，并且在每个状态下 agent 以 0.2 的概率滑动，即执行相反的动作。由图 1 可知，在每个状态下最优的动作是动作 a，即使有些时候以一定的概率滑动到动作 b，这是因为虽然在状态 1~状态 4 下执行动作 a 获得的立即奖赏是 0，但是只要状态 5 达到就可以获得 10 的立即奖赏。在这 5 个状态中执行动作 b 获得 2 的立即奖赏。其中，折扣因子 $g = 0.99$。

在链问题中，一般的学习算法会陷入在初始状态，会偏向于选择动作 b，以获取到一个较小的奖赏，获得的是次优的策略，如果算法收敛太快，agent 将不会发现获得更高奖赏的路径。因此针对链问题，需要高效地探索动作以及精确地估计累积折扣奖赏。在该实验中，将 Dyna-PS-BayesQL 算法与 Q 学习 semi-uniform 算法、Q 学习 Boltzmann 算法以及贝叶斯 Q 学习的 Myopic-VPI 算法相比较，每个算法独立重复 30 次实验，结果取平均值。Dyna-PS-BayesQL 算法中优先级中的规划步数 N 设为 10，优先级阈值 q 初始值设为 0.5，且每步以 0.99 倍的衰减速度衰减以提高算法的时间性能。算法性能衡量的标准是在一定步数内所获得的累积奖赏的大小，即将每一步获得的立即奖赏相加。实验结果如图 2 和表 1 所示。

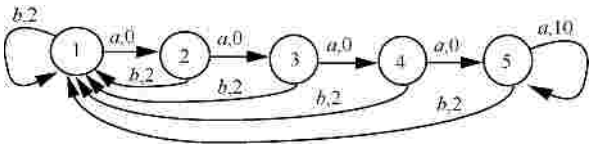

图 1　链问题示意

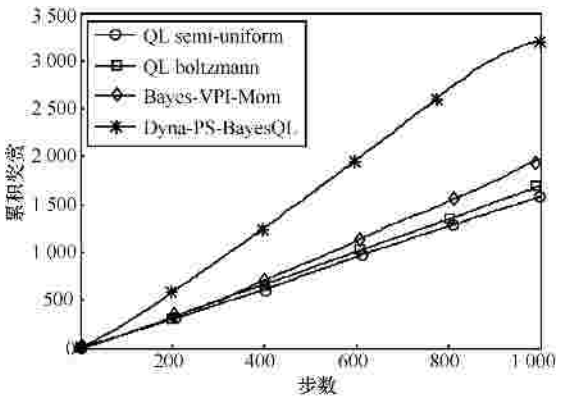

图 2　4 种算法在每一步的累积奖赏曲线比较

表 1　链问题 4 种算法在一定步数获得的累积奖赏比较

链问题	步数	
	500	1 000
QL semi-uniform	783	1582
QL Boltzmann	829.8	1 692.8
Bayes-VPI-Mom	905.8	1 973.5
Dyna-PS-BayesQL	1 603.5	3 209.7

表 1 给出的是 4 种算法在 500 步和 1 000 步获得的累积奖赏，由表 1 可以看出 Dyna-PS-BayesQL 算法明显优于其他 3 种算法，这主要是因为在

Dyna-PS-BayesQL 算法中每个状态动作对的值函数的更新都是更新一个分布，可以得到更加可靠的值函数，在规划部分使用优先级扫描和动态规划来加快值函数的收敛速度。为了更加清楚地看出累积奖赏增长的趋势，图 2 给出了 4 种算法在每一步的累积奖赏。

4.2 迷宫导航问题

在迷宫导航问题中，agent 的目标是收集旗子并把它们带回目标终点。迷宫如图 3 所示，S 标记的是初始状态，G 标记的是目标状态，F 标记的是待收集的旗子所在的位置。到达目标状态 G 所得的奖赏是由收集到的旗子数目决定的，其余所有状态转移的奖赏都为 0。Agent 一旦到达目标状态，就重新回到初始状态，开始新的情节。在该问题中，一共有 264 个状态，这是因为有 33 个可到达的位置，3 个旗子的组合一共有 8 种可能，并且 agent 可以向左、向右、向上、向下移动，这些动作可以让 agent 到达它们相邻状态。除非碰到障碍物或到达迷宫的边界时，agent 会留在原地不动。为了使问题更加复杂，假设 agent 将会以 0.1 的概率偶然的滑到与执行动作垂直的方向上。该问题的难点是在到达目标状态之前需要做充分的探索来收集所有的 3 个旗子。其中，折扣因子 $g = 0.95$。

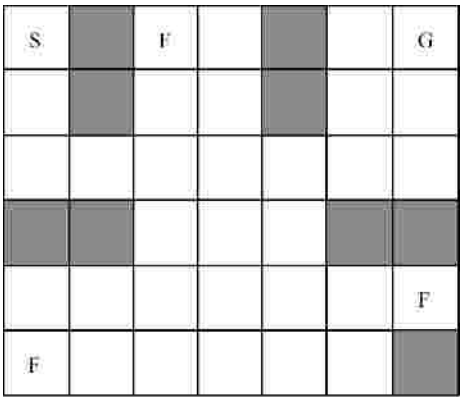

图 3 迷宫导航问题示意

迷宫导航问题的状态空间要比链问题的状态空间大很多，但是随机性比链问题要小，用迷宫导航问题来评估各种探索策略在规模扩大时的性能。在实验中，仍然将 Dyna-PS-BayesQL 算法与 Q 学习 semi-uniform 算法、Q 学习 Boltzmann 算法以及贝叶斯 Q 学习的 Myopic-VPI 算法相比较。算法性能衡量的标准是在 20000 步数内所收集到的旗子总和，到达终止状态后，agent 重新从初始状态 S 出发，开始一个新的情节。实验结果如表 2 所示，表 2 给出了在 10 000 步和 20 000 步的 4 种算法收集到旗子对比情况。

表 2 迷宫导航问题 4 种算法在一定步数收集到的旗子总和

迷宫导航问题	步数	
	10 000	20 000
QL semi-uniform	72	549
QL Boltzmann	56	148
Bayes-VPI-Mom	69	160
Dyna-PS-BayesQL	334	1 246

图 4~图 7 分别给出了这 4 种算法性能详细的曲线信息，每个图的上部分描述了 20 000 步中相应算法能执行的情节数，以及每个情节数的步数，每个图的下半部分描述了该情节中所收集到的旗子个数。

图 4 是在迷宫导航问题中使用 QL semi-uniform 算法的情况，随机探索的概率设为 0.1。由图 4 可以看出在 20 000 步中一共执行了 572 个情节，在算法前 60 个情节中，agent 处于探索阶段，每个情节数的步数比较多，且通过前后情节数步数比较，发现步数振荡比较大，收集到的旗子个数也不稳定。在 100 个情节后，虽然每一步仍以 0.1 的概率随机探索，但 agent 每次的探索步数开始趋于一个稳定的值，在 17 步左右震荡且振荡比较小，每次收集到的旗子个数也是比较稳定，仅仅只能够收集一个旗子，实际上仅仅收集到了迷宫导航问题中的最上面的一个旗子，还有 2 个旗子已经几乎无法收集到，这并不满足问题的要求。所以 QL semi-uniform 算法并不能较好地平衡探索与利用。

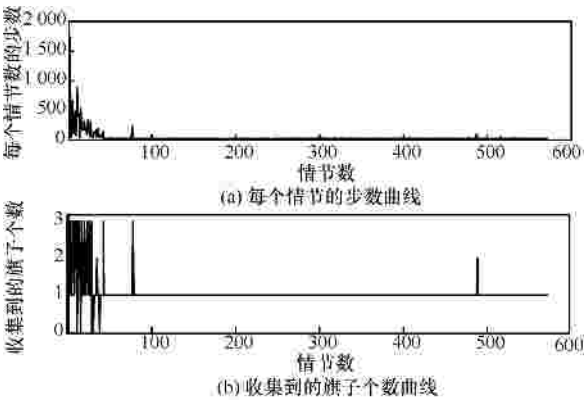

图 4 QL semi-uniform 算法在每一个情节的步数以及收集到的旗子个数曲线

图 5 是在迷宫导航问题中使用 QL Boltzmann 算法的情况，初始的温度参数设为 20，并且在每个

情节结束时，以 0.9 倍的衰减速度衰减以降低探索提高利用。由图 5 可以看出在 20 000 步中一共执行了 93 个情节。在算法的前 80 个情节中，agent 处于探索阶段，通过与 QL semi-uniform 算法对比，发现它收集到的旗子没有 QL semi-uniform 算法多，每个情节的步数比 QL semi-uniform 算法多，表明 agent 尝试努力着去收集 3 个旗子，因此，QL Boltzmann 算法探索的能力要比 QL semi-uniform 算法强。在 80 个情节后，QL Boltzmann 算法也趋于稳定，依然只能够收集到迷宫导航问题中的最上面的一个旗子，还有 2 个旗子已经几乎无法收集到。所以 QL Boltzmann 算法在应对平衡探索与利用时，探索的能力稍微比 QL semi-uniform 算法好，但依然不太理想。

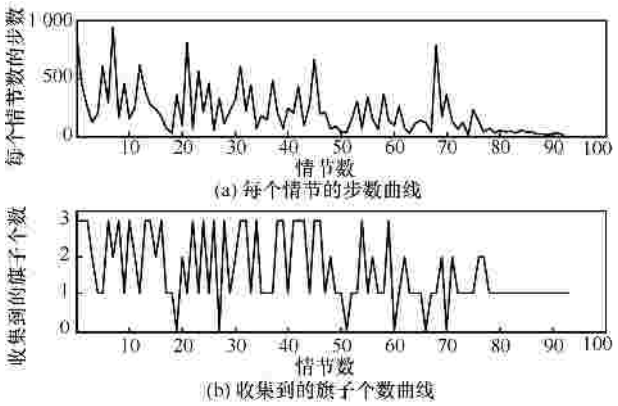

图 5　QL Boltzmann 算法在每一个情节的步数以及收集到的旗子个数曲线

图 6 是在迷宫导航问题中使用 Bayes-VPI-Mom 算法的情况，由图 6 可以看出在 20 000 步中一共执行了 91 个情节，agent 收集的旗子个数在不断震荡。由表 2 可以看出，与 QL Boltzmann 算法相比，前 10 000 步中收集了 56 个旗子，20 000 步中收集了 160 个旗子，都要比 QL Boltzmann 算法好，所以 Bayes-VPI-Mom 算法在这 20 000 步中一直尝试着平衡探索与利用，所以 Bayes-VPI-Mom 算法在应对平衡探索与利用时，要比 QL Boltzmann 算法好。

图 7 是在迷宫导航问题中使用 Dyna-PS-Bayes QL 算法的情况，优先级中的规划步数 N 设为 20，优先级阈值 q 初始值设为 0.01，且每个情节以 0.99 倍的衰减速度衰减。由图 7 可以看出在 20 000 步中一共执行了 583 个情节，agent 初期每个情节数的步数比较多，收集到的旗子个数也不稳定，但情节的步数在快速振荡下降，慢慢地趋于发现最优策略，在 250 个情节后，agent 每次的执行步数开始趋于一个稳定的值，在 35 步左右震荡且振荡比较小，并且每次能够收集到迷宫导航问题中的全部 3 个旗子。由表 2 可以看出 Dyna-PS-BayesQL 算法在 10 000 步中，20 000 步中收集的旗子，要比 QL semi-uniform 算法、QL Boltzmann 算法、Bayes-VPI- Mom 算法都要多，更重要的是 Dyna-PS-BayesQL 算法能够收集到问题中的全部旗子并把旗子以接近理论最小值的步数带回目标状态。在这一过程中，agent 到达目标的次数以及收集到的旗子个数均比其他 3 种算法多，这表明 Dyna-PS-BayesQL 算法能够较好地平衡探索和利用，且具有较好的收敛速度和精确度。

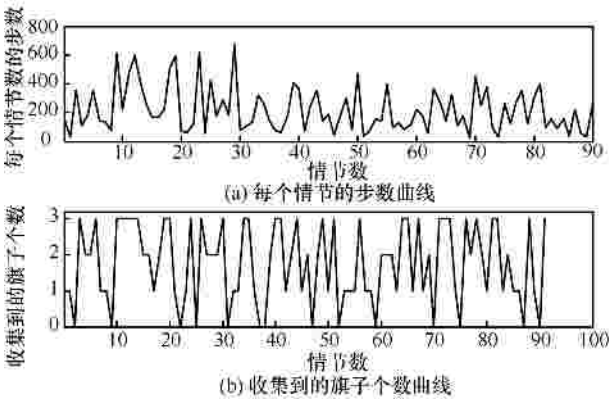

图 6　Bayes-VPI-Mom 算法在每一个情节的步数以及收集到的旗子个数曲线

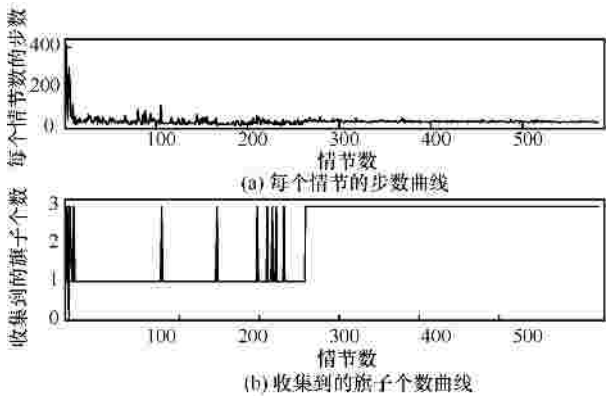

图 7　Dyna-PS-BayesQL 算法在每一个情节的步数以及收集到的旗子个数曲线

针对所有状态动作对先验参数的设定问题，采用一种通用的方法。在 Dyna-PS-BayesQL 算法初始阶段，因为对先验均值的不确定，所以将高斯伽玛分布的方差设定为一个较大值，即初始阶段均值 m 的方差 $\frac{b}{l\ (a\ -1)}$ 值较大，然后根据 agent 与环境交互获得的样本更新参数。初始阶段所有状态动作对的先验设定为 $(m_0, l_0, a_0, b_0) = (1, 2, 2, 200)$，以在初始

状态 S 采取向下动作为例，图 8 给出了该状态动作对下均值 m 在初始阶段、2 000 步、5 000 步和 10 000 步时的概率密度分布图。算法独立重复 30 次，结果取平均值，2 000 步的均值 m 约等于 1.17，5 000 步的均值 m 约等于 1.72，10 000 步的均值 m 约等于 2.46，由此可见，随着迭代步数的增加，均值 m 的后验逐渐变好，且 m 的后验分布比先验分布在其均值附近更为集中，表明结果更为准确。

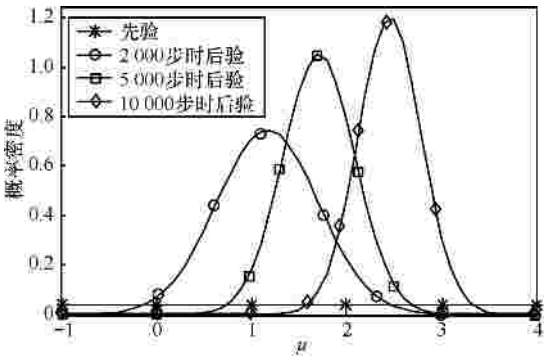

图 8 均值 m 的先验和后验概率密度

下面探讨先验方差对实验结果的影响，以参数 b 为例，参数 b 取不同值时，方差大小也将会不同，表 3 中统计了算法初始值 b_0 取为 20、50、200 时，执行 10 000 步收集到 3 个旗子且回到终点的次数情况，算法独立重复 30 次，结果取平均值。由表 3 可见，初始阶段均值 m 的方差较大时，效果较好。当参数 l 和 a 取不同值时，也将会有类似结果。

表 3 迷宫导航问题参数 b 初始值不同时的情况

参数	平均次数
(1, 2, 2, 20)	50.5
(1, 2, 2, 50)	55.9
(1, 2, 2, 200)	69.1

5 结束语

本文针对强化学习中的贝叶斯 Q 学习收敛速度慢且收敛精度低的问题，提出一种基于优先级扫描 Dyna 结构的贝叶斯 Q 学习方法。将优先级扫描的思想与 Dyna 结构相结合，并用于贝叶斯 Q 学习算法。在学习部分，利用贝叶斯 Q 学习进行在线学习，并对环境动态性建模，利用 Myopic-VPI 方法选择动作，以平衡强化学习中的探索与利用。在规划部分，利用优先级扫描方法和动态规划方法对需要估计的值函数进行规划更新，提高对历史经验信息的利用以及规划学习的效率，从而提升算法的收敛速度和收敛精度。从理论上对 Dyna-PS-BayesQL 算法进行了分析。将该算法应用于经典的链问题和迷宫导航问题，通过与 Q 学习 semi-uniform 算法、Q 学习 Boltzmann 算法以及贝叶斯 Q 学习的 Myopic-VPI 算法进行比较分析，实验结果表明 Dyna-PS-BayesQL 算法能较好地平衡探索与利用，且有较优的收敛速度和精确度。但是本文所提出的方法需要设定所有状态动作对的先验参数，而在实际问题中，难以获得较准确的先验参数，因此下一步的工作是通过寻找较优的值函数表示方法并结合贝叶斯推理来求解最优策略。

参考文献：

[1] SUTTON R S, BARTO A G. Reinforcement Learning: An Introduction[M]. Cambridge: MIT Press, 1998.

[2] 徐昕. 增强学习与近似动态规划[M]. 北京: 科学出版社, 2010.
XU X. Reinforcement Learning and Approximate Dynamic Programming[M]. Beijing: Science Press, 2010.

[3] 刘全, 傅启明, 龚声蓉等. 最小状态变元平均奖赏的强化学习方法[J]. 通信学报, 2011, 32(1): 66-71
LIU Q, FU Q M, GONG S R, *et al*. Reinforcement learning algorithm based on minimum state method and average reward[J]. Journal on Communications, 2011, 32(1):66-71.

[4] 肖飞, 刘全, 傅启明等. 基于自适应势函数塑造奖赏机制的梯度下降 Sarsa (?) 算法[J]. 通信学报, 2013, 34(1): 77-88.
XIAO F, LIU Q, FU Q M, *et al*. Gradient descent Sarsa(?) algorithm based on the adaptive potential function shaping reward mechanism[J]. Journal on Communications, 2013,34(1):77-88.

[5] SZEPESVÁRI C. Algorithms for Reinforcement Learning[M]. San Rafael: Morgan Claypool, 2010.

[6] WATKINS C. Learning From Delayed Rewards[D]. Cambridge: Kings's College, University of Cambridge, 1989.

[7] SUTTON R S. Dyna, an integrated architecture for learning, planning, and reacting[J]. SIGART Bulletin, 1991, 2: 160-163.

[8] SUTTON R S, SZEPESVÁRI C, GERAMIFARD A, *et al*. Dyna-style planning with linear function approximation and prioritized sweeping[A]. Proceedings of the 24th Conference on Uncertai y in Artificial Intelligence[C]. Finland: AUAI, 2008.

[9] WINGATE D, SEPPI K D. Prioritized methods for accelerating MDP solvers[J]. Journal of Machine Learning Research, 2005, 6: 851-881.

[10] MEULEAU N, BOURGINE P. Exploration of multi-state environments: local measures and back-propagation of uncertainty[J]. Machine Learning, 1999, 35(2):117-154.

[11] COGGAN M. Exploration and exploitation in reinforcement learning[A]. Proceedings of the 4th International Conference on Computational Intelligence and Multimedia Applications[C]. Japan, 2001.

[12] ALEXANDER L, STREHL, MICHAEL L. A theoretical analysis of model-based interval estimation[A]. Proceedings of the 22nd International Conference on Machine Learning[C]. New York: ACM, 2005.

[13] MEULEAU N, BOURGINE P. Exploration of multi-state environments: local measures and back-propagation of uncertainty[J]. Machine Learning, 1999, 35(2):117-154.

[14] DEARDEN R, FRIEDMAN N, RUSSELL S. Bayesian Q learning[A]. Proceedings of 15th International Conference on Artifi ial Intelligence[C]. Menlo Park: AAAI Press, 1998.

[15] DEARDEN R, FRIEDMAN N, ANDRE D. Model based Bayesian exploration[A]. Proceedings of 15th Conference on Uncertainty in Artificial Intelligence[C]. San Francisco: Morgan Kaufmann, 1999.

[16] ASMUTH J, MICHAEL L, *et al*. Potential-based shaping in model-based reinforcement learning[A]. Proceedings of the 23th AAAI Conference on Artificial Intelligence[C]. Chicago: AAAI Press, 2008.

[17] PENG J, WILLIAMS R J. Efficient learning and planning within the dyna framework[J]. Adaptive Behavior, 1993, 2: 437-454.

[18] DEGROOT M, SCHERVISH M. Probability and Statistics[M]. New York: Person Edition, 2010.

[19] TEACY W, CHALKIADAKIS G, FARINELLI A. Decentralised Bayesian reinforcement learning for online agent collaboration[A]. Proceedings of 11th International Joint Conference on tonomous Agents and Multi-Agent Systems[C]. Spain: IFAAMAS, 2012.

作者简介：

于俊（1989-），男，江苏泰州人，苏州大学硕士生，主要研究方向为强化学习和贝叶斯推理。

刘全（1969-），男，内蒙古牙克石人，苏州大学教授、博士生导师，主要研究方向为强化学习、智能信息处理和自动推理。

傅启明（1985-），男，江苏淮安人，苏州大学博士生，主要研究方向为强化学习、贝叶斯推理和遗传算法。

孙洪坤（1988-），男，江苏淮安人，苏州大学硕士生，主要研究方向为强化学习。

陈桂兴（1990-），男，江西赣州人，苏州大学硕士生，主要研究方向为强化学习和模式识别。