【电子与信息科学 / Electronics and Information Science】

随机状态下基于期望经验回放的 Q 学习算法

张 峰, 钱 辉, 董春茹, 花 强

河北省机器学习与计算智能重点实验室, 河北大学数学与信息科学学院, 河北保定 071002

摘 要: 强化学习的经验回放方法在减少状态序列间相关性的同时提高了数据的利用效率, 但目前只能用于确定性的状态环境. 为在随机状态环境下充分利用经验回放, 且能够保持原有的状态分布, 提出一种基于树的经验存储结构来存储探索过程中的状态转移概率, 并根据该存储方式, 提出基于期望经验回放的 Q 学习算法. 该方法在保证算法复杂度较低的情况下, 可实现对环境状态转移的无偏估计, 减少 Q 学习算法的过估计问题. 在经典的机器人随机行走问题中进行实验, 结果证明, 相比于基于均匀回放方法和优先回放的经验回放方法, 基于期望经验回放 Q 学习算法的收敛速度约提高了 50%.

关键词: 人工智能; 机器学习; 强化学习; 经验回放; Q 学习算法; 随机环境; 收敛; 过估计

中图分类号: TP181

文献标志码: A

doi: 10.3724/SP.J.1249.2020.02202

An expected experience replay based Q-learning algorithm with random state transition

ZHANG Feng, QIAN Hui, DONG Chunru, and HUA Qiang

Hebei Key Laboratory of Machine Learning and Computational Intelligence,
College of Mathematics and Information Science, Hebei University, Baoding 071002, Hebei Province, P. R. China

Abstract: The experience replay method in reinforcement learning algorithms reduces the correlation between state sequences by sampling randomly and increases the efficiency of data utilization. However, presently it can only be used in the deterministic environment. In order to use the experience replay efficiently in a dynamic random environment and keep the original state transition distribution unchanged, we propose a tree-based experience storage structure to store the state transition probability in the process of exploration and provide an expected experience replay based Q-learning algorithm which realizes an unbiased estimation of transition distribution. The main advantage of proposed algorithm lies in that it can keep the transition distribution unchanged without increasing the algorithm complexity. Additionally, it eliminates the overestimation of Q value in an efficient way. Experimental results in the classical random walking problem of robot verify that the proposed algorithm improves the convergence speed by about 50%.

Key words: artificial intelligence; machine learning; reinforcement learning; memory replay; Q-learning algorithm; stochastic environment; convergence; over estimation

Received: 2019-04-10; **Accepted:** 2019-05-15

Foundation: Natural Science Foundation of Hebei Province (F2017201020, F2018201115); Key Science and Technology Foundation of the Education Department of Hebei Province (ZD2019021); Youth Fund of Hebei Education Department (QN2017019)

Corresponding author: Professor HUA Qiang. E-mail: huaq@hbu.edu.cn

Citation: ZHANG Feng, QIAN Hui, DONG Chunru, et al. An expected experience replay based Q-learning algorithm with random state transition [J]. Journal of Shenzhen University Science and Engineering, 2020, 37(2): 202-207. (in Chinese)

Q学习算法^[1]是一种用来解决马尔可夫决策问题^[2]的流行算法.与动态规划的值迭代和策略迭代算法相比,Q学习算法能应用于对模型未知的环境中,通过对环境采样,基于Bellman最优方程,采用Bootstrap方式来迭代更新状态-动作的Q值与最优策略.Q学习算法的更新公式^[2]为

$$Q_{t+1}(s,a) = \alpha(r + \gamma \max_a Q_t(s',a) - Q_t(s,a)) \quad (1)$$

其中, $Q_{t+1}(s,a)$ 是在 $t+1$ 时刻对状态-动作价值的估计; $Q_t(s,a)$ 为 t 时刻状态-动作价值的估计; s 为当前状态; s' 为执行动作 a 后的后续状态; α 为学习率; $\max_a Q_t(s',a)$ 为执行动作 a 后达到的后续状态的最大状态-动作价值;根据Bellman最优方程理论,利用式(1)对Q值迭代更新,可保证Q值收敛到其真实值^[2].其中, $r + \gamma[\max_a Q_t(s',a) - Q_t(s,a)]$ 为时序差分误差(temporal-difference error, TD-error),即迭代过程中相邻两次Q值估计的偏差, r 为执行动作 a 后获得的立即回报.

Q学习算法的收敛速度与经验数量指数相关^[3].为进一步提高Q学习算法的收敛速度,研究人员提出了很多Q学习更新算法.如 $Q(\lambda)$ 算法^[4]、后向Q学习算法^[5]等.但是这些改进算法通常会将实时采样数据更新后就丢弃,造成两个问题:①对于那些不容易被采样的状态,仅使用1次会造成数据利用效率降低;②Bellman方程中每个Q值的更新依赖于后续状态中的Q值,使用逆序访问序列的收敛速度更快,而标准形式的Q学习算法中Q值更新是按正序访问序列方式更新的,收敛速度较慢.针对这些问题,LIN^[6]提出经验回放算法.随后MINH等^[7]将其扩展到深度学习领域,提出深度Q网络(deep Q-network, DQN)算法,通过构造经验池保存历史经验,再采用经验回放的方法,在必要时再次使用历史经验.该算法打破了状态转移之间的强关联性,不仅保证了监督学习中数据独立同分布的假设,还加快了算法收敛速度,提高了数据利用率.

目前关于深度强化学习的研究中,针对经验回放主要采取两种方式:①均匀回放,即将采样序列 $\langle s, a, r, s' \rangle$ 保存到经验池中,等概率选择经验池中的经验,并应用式(1)更新Q值;②优先回放,即优先回放TD-error值较大的经验,以提高收敛速度,优先回放的经验结构与均匀回放的经验结

构一致.但是,这些方法只适用于状态转移确定的情况,当系统的状态转移不确定时,两种方法都不能保证回放经验的状态转移分布与实际状态转移分布一致.另外,在DQN算法中,由于Q学习算法总是贪心地选择后续状态动作对中的最大Q值,因此可能会出现Q值过估计现象^[8].为此,本研究提出一种新的采样序列存储结构,改进了迭代计算公式,并提出基于状态-动作对的期望Q值经验回放方法.该方法在保证算法复杂度较低的情况下,实现对于环境状态转移的无偏估计,并在一定程度上减少了Q学习算法的过估计问题.

1 经验回放

经验回放通过agent对环境的多次采样来提高强化学习算法的收敛速度并解决更加复杂的问题^[7].由于经验可存储在agent的经验池中,减少了等待环境反馈的时间,减小了agent与环境交互的代价.相比策略梯度类算法^[9],基于经验回放的值评估算法能将采样效率提高约10倍^[10].与基于模型的算法不同的是,经验回放机制的所有状态转移序列都来自真实环境,避免了因模型拟合准确度问题而造成的偏差.

但是,在不确定性的系统中,即使agent执行相同的动作,agent观察到的环境也可能会转移到不同的状态.与不同后续状态的转移概率不同,仅经验回放选择的状态转移分布与实际环境一致时,才能保证最终的Q值是准确值.

DQN算法通过经验回放减少了状态转移序列的相关性,并通过批量选择经验进行小批量更新,并多次回放经验,提高了经验利用率^[11].为进一步提升算法速度,优先经验回放^[12]通过给每个状态转移序列加权,将权重设为TD-error的幂函数,再以大概率选择回放TD-error值大的序列,从而加快收敛速度.但是,此方法会令状态转移分布与实际环境不一致,必须对经验的更新值进行修正,让每个更新值都乘以重要性采样权重^[12].当每个经验都需按权重回放时,又会极大地增加算法的复杂度.因此,为减少复杂度,在优先经验回放中新增了sum-tree结构,用来保存经验的回放优先级,以实现快速获取经验.

由以上分析可见,均匀回放不需进行重要性采样加权,算法简单,但学习效率较慢.在经验池足够大的情况下,可认为经验的状态转移概率与环境

一致, 因此采用均匀经验回放可近似保证状态转移分布与实际状态转移分布的一致性. 基于优先回放的 Q 学习算法学习效率有所提高, 但是相对复杂, 对于随机性较大的动态系统, 优先经验回放打破了原有的经验分布, 用于处理不确定环境时, 效果可能更差. 因此, 本研究通过改进现有经验存储方式, 设计基于期望经验回放的 Q 学习算法. 该方法在保证算法复杂度较低的情况下, 可实现对于不确定环境状态转移的无偏估计, 并保证用于计算 Q 值的状态转移概率与实际环境的状态转移概率一致.

2 期望经验回放

为解决优先经验回放所带来的问题, 本研究重点关注优先回放算法造成的有偏回放以及复杂的数据结构问题. 为使经验回放的状态转移分布与真实的状态转移分布一致, 考虑以采样获得的状态转移频率 μ_n/n 作为环境状态转移分布 P 的估计值.

由伯努利大数定律

$$\lim_{n \rightarrow \infty} P\left(\left|\frac{\mu_n}{n} - P\right| < \varepsilon\right) = 1 \quad (2)$$

可知, 估计值 μ_n/n 为后续状态分布概率的无偏估计. 其中, ε 为任意实数. 因此可将此估计值带入迭代公式, 获得动作价值更好的估计.

为保存状态转移频率, 本研究将经验存储结构修改为 $\langle s, a, \langle \langle s'_0, r_0, c_0 \rangle, \langle s'_1, r_1, c_1 \rangle \dots \rangle \rangle$, 其中, s'_0 与 s'_1 是在当前状态 s 下执行动作 a 后可能到达的后续状态; r_0 与 r_1 是转移到下一个状态的即时期望回报; c_0 与 c_1 中保存了从当前状态 s'_0 转移到下一状态 s'_1 的次数.

采用这种经验存储结构, 可实现在特定状态下执行同一动作时, 无需更新整个经验, 只更新经验池中后续状态的访问次数和即时期望回报值. 每个经验样本相当于一个完整的状态转移子树, 当采样次数足够大时, 可直接以等概率方式回放经验, 且多次回放经验仍是无偏估计. 更新公式修订为

$$Q_{i+1}(s, a) = Q_i(s, a) + \alpha[\hat{Q}_i(s, a) - Q_i(s, a)] \quad (3)$$

其中, $\hat{Q}_i(s, a)$ 是本次采样得到的动作评估值, 是经验池中本次动作获得的回报值与所有后续状态值的期望. 转移到后续状态的转移概率估计值由本次采样的样本中对转移频次的统计得到, 计算公式为

$$\begin{aligned} \hat{Q}_i(s, a) &= \sum_{s'} \{ \hat{p}(s' | s, a) [R(s, a, s') + \gamma V(s')] \} = \\ &= \sum_{s'} \left\{ \frac{c(s')}{\sum_{s'} c(s')} [R(s, a, s') + \gamma V(s')] \right\} = \\ &= \sum_{s'} \left\{ \frac{c(s')}{\sum_{s'} c(s')} [R(s, a, s') + \gamma \max_a Q_i(s', a)] \right\} \end{aligned} \quad (4)$$

其中, $\hat{p}(s' | s, a)$ 为后续状态转移概率的估计值; $R(s, a, s')$ 是转移到对应状态获得的立即回报期望值; γ 为折扣因子; $V(s')$ 为状态 s' 的状态值; $c(s')$ 是当前状态转移到后续状态 s' 的次数; $Q_i(s', a)$ 是对应后续状态的最大动作值. 根据式(4), 本研究设计的基于状态-动作期望经验回放算法如图 1.

algorithm1: 期望经验回放算法

input: 采样次数: episodes; 学习率: α ; 开始回放步数: startStep; 回放周期: period; 回放数量: m

output: Q 值

- (1) 初始化 Q 值表为任意值: 所有状态-动作的 Q 值均为 0. $Q(\text{terminal-state}, \cdot) = 0$;
- (2) step = 0;
- (3) for each episode do
- (4) 初始化当前状态 s ;
- (5) for each step of episode do
- (6) 从当前 Q 值表中依据策略 (例如 ε -greedy) 选择动作 a ;
- (7) 执行动作 a , 观察回报 r 与 s' ;
- (8) step = step + 1;
- (9) 以 s, a 作为主键, 将 (s, a, r, s') 保存到经验池: 若经验池中无 (s, a) 记录则新增 1 条, 若有记录, 对当前记录更新;
- (10) if step >= startStep then
- (11) if step mod period == 0 then
- (12) 从经验池中选择 m 个经验;
- (13) for each 经验 do
- (14) 用更新公式(2)对 Q 值更新;
- (15) $s = s'$
- (16) end for
- (17) end for

图 1 期望经验回放算法

Fig. 1 Expected memory replay-based Q-learning algorithm

与原均匀经验回放与优先经验回放算法相比, 本研究算法具有以下优点:

- 1) 式(4)中状态转移概率的估计值 $\hat{p}(s' | s, a)$

是对动态随机环境转移的无偏估计, 无需进行加权调整.

2) 经验池中的每个样本包含多个序列, 增加了经验池的使用效率.

3) 每次经验回放相当于回放多个转移序列的期望, 经验利用率比原均匀回放算法更高, 而经验池的结构比优先经验回放更简单且易于实现.

4) 由于使用了后续状态价值的期望, 由式(5)可知, 每次更新的 TD-error 值都更稳定, 这在一定程度上减小了 Q 学习算法对动作价值的过估计问题^[13].

$$\min_{s'} \max_a Q(s', a) \leq E(\max_a Q(s', a)) \leq \min_{s'} \max_a Q(s', a) \quad (5)$$

在采用 TD 评估方式更新 Q 值的方法中, 评估值的方差可分为立即回报方差、状态转移方差和后续估计值方差之和^[14]. 采用期望经验回放, 将使得状态转移方差变为 0, 加快了算法收敛速度.

由于在采样初期, $\hat{p}(s'|s, a)$ 的误差可能较高, 且多次回放会导致算法不稳定. 为此, 采用递增回放数量的方式解决该问题. 由此得到均匀回放经验的 Q 学习算法、带优先经验回放 Q 学习算法, 以及动作期望经验回放的 Q 学习算法的更新时间复杂度分别为 $O(1)$ 、 $O(\lg N)$ 和 $O(\lg N)$, 采样时间复杂度分别为 $O(1)$ 、 $O(\lg N)$ 和 $O(1)$. 其中, N 为经验池中样本数量. 可见, 基于期望经验回放的 Q 学习算法的采样效率明显低于基于优先回放的 Q 学习算法. 下面将通过实验验证算法可在未提高采样复杂度的情况下, 仍能以较快的速度收敛到最优策略, 且方差较小.

3 实 验

3.1 实验设置

设计一个简单的带有障碍的机器人行走环境 (如图 2), 并在此环境下分别测试原始 Q 学习算

图 2 机器人随机行走问题

Fig. 2 Robot random walk

法、均匀回放算法、优先回放算法及本研究算法收敛到最优策略的速度. 假设有一个机器人处在一个 3 行 4 列的网格中, 其任务是到达位置 12, 在位置 8 处有一个陷阱, 一旦机器人进入则会损坏; 位置 6 处有一根柱子, 机器人若碰到柱子则会停在原地. 网格周围都是高墙, 机器人碰到墙, 就会停留在原地.

机器人可站在图 2 的空白区域, 可向上下左右走, 但限于控制精度, 仅有 0.8 的概率走到预定位置. 例如, 若机器人在位置 3 往上走, 则有 0.8 的概率走到位置 7, 0.1 的概率走到位置 2 或位置 4; 若机器人从位置 1 往上走, 则有 0.8 的概率走到位置 5, 0.1 的概率走到位置 2 和位置 1.

任务目标是在任意空白位置, 求机器人行走到达位置 12 的最佳轨迹. 为使机器人避开陷阱, 到达位置 12, 可将到达位置 12 的回报设为 1 000, 到达其他位置的回报设为 -1, 折扣设为 1. 同时, 为提供一个比较的基准, 基于此设置, 先利用值迭代算法求得该问题真实的最优策略如图 3.

图 3 机器人行走环境的最优策略
Fig. 3 The optimal policy in the robot walk environment

根据 Bellman 最优方程, 得到最优策略下对应于各状态动作价值如表 1.

表 1 最优策略下动作价值
Table 1 Action values for the true optimal policy

状态	上	右	下	左
1	993.3	991.5	992.2	992.5
2	991.1	990.0	991.1	992.1
3	988.0	981.6	988.8	990.7
4	196.1	881.6	980.7	889.6
5	994.7	993.7	992.6	993.7
7	896.9	197.9	890.5	989.7
9	995.1	996.0	994.1	994.8
10	996.4	997.4	996.4	995.3
11	997.7	998.6	990.6	995.7

分别用经典 Q 学习算法、带均匀回放经验、带优先经验回放和动作期望经验回放的 Q 学习算法对机器人行走环境求解最优策略, 每种算法测试 1 000 次, 通过比较算法收敛到最优策略所需的采样数量, 以及算法的收敛速度.

3.2 实验结果

由于不同算法适合的超参数不同, 本研究对每个算法分别进行多次实验, 取其平均最优结果, 再统计各算法收敛到最优策略时所需采样次数的均值与标准差. 收敛测试结果表明, 经典 Q 学习算法、随机经验回放算法、优先经验回放算法和期望经验回放算法在各自最优参数下, 收敛到最优策略所需的平均采样次数及其标准差分别为 $(6\ 734 \pm 533)$ 、 $(2\ 200 \pm 255)$ 、 $(2\ 678 \pm 847)$ 和 $(1\ 260 \pm 386)$ 次. 可见, 期望经验回放算法收敛到最优策略所需的平均采样次数仅为原始 Q 学习算法的 1/5、方差较小且收敛速度比均匀回放算法和优先回放算法均提高了约 50%. 优先回放算法使回放的状态转移分布与实际环境不一致, 因此方差最大. (详细测试结果请扫描论文页末右下角二维码查看.) 为保证实验的可复现性, 提供期望经验回放算法取得最好结果的参数为: 学习率 $\alpha = 0.005$, 开始回放步数 $\text{startStep} = 2\ 000$, 回放周期 $\text{period} = 4$, 回放数量 replaySize 从 8 递增到 32, 即在每回放 100 次后回放数量加 4. 实验表明, 采用动作期望经验回放的方式, 明显提高了算法收敛到最优策略的速度, 对于动态环境更有优势.

3.3 算法的进一步分析

在算法迭代过程中, 所求策略总是会先收敛到一个次优的策略. 图 4 给出了一个收敛过程中最先出现 1 个次优策略的例子.

→	→	→	12
↑	6	↑	8
↑	←	←	←

图 4 收敛过程中的次优策略

Fig. 4 An example of suboptimal policy in convergence process

由图 4 可见, 此策略是到达目标的最短路径策略, 直观上是最优策略. 但在状态 4 和 7 处, 最短

路径策略风险较高, 这是因为算法迭代过程中分别对状态 4 向左和状态 7 向上的动作产生了过估计. 由式(4)可知, 本研究算法对状态 4 和 7 的动作价值估计更稳定, 且在迭代过程中保留了动作价值的序关系, 保证了算法能够快速收敛到最优策略.

表 2 和表 3 分别给出了基于均匀回放的 Q 学习算法和基于期望回放算法收敛到最优策略后的平均动作价值与真实动作价值的误差统计.

表 2 基于均匀回放的算法收敛后动作价值的误差

Table 2 Action value errors based on uniform replay

状态	上	右	下	左
1	-2.1	-2.7	-2.5	-2.5
2	-3.3	-5.6	-3.3	-2.3
3	-29.6	-88.8	-17.2	-4.9
4	-16.2	-117.4	-117.6	-28.3
5	-2.0	-2.4	-2.5	-2.4
7	-3.1	-0.5	-7.9	-15.3
9	-2.3	-1.9	-2.3	-2.3
10	-2.1	-1.8	-2.1	-2.2
11	-1.7	-1.7	-14.2	-3.7

表 3 基于期望回放算法收敛后动作价值的误差

Table 3 Action value errors based on expected replay

状态	上	右	下	左
1	-17.2	-30.2	-23.8	-22.0
2	-33.4	-41.8	-33.2	-24.9
3	-62.0	-115.0	-52.4	-36.7
4	-16.4	-119.7	-122.7	-48.4
5	-11.7	-15.0	-21.1	-15.0
7	-12.5	-5.1	-40.7	-50.0
9	-10.8	-9.5	-13.6	-11.3
10	-8.8	-7.9	-8.9	-10.5
11	-7.6	-7.1	-46.2	-13.6

由表 2 和表 3 可见, 当算法收敛到最优策略时, 对于所有状态, 动作期望回放算法最终的动作价值比均匀回放算法的动作价值要低, 但都收敛到了最优策略, 说明动作期望算法有效解决了 Q 学习算法的动作价值过估计问题.

结 语

提出一种基于树的经验存储结构,并将随机环境的后续状态转移频次保留到经验存储的结构中来估计实际环境的转移分布,设计了具体算法.实验结果表明,该算法可行且有效.相比于原有经验回放方式,本研究提出的基于期望的经验回放模型存储更加高效,数据利用率更高.由于算法采用了更稳定的迭代计算公式,有效降低了Q学习算法的过估计问题,对于动态系统的策略收敛速度更快.下一步,我们将深入研究包括如何用近似模型对状态转移概率进行拟合,以及如何设计有效的算法以使算法收敛更加稳定.另外,拟采用参考经验回放的连续思想^[15]对算法进行扩展.

基金项目: 河北省自然科学基金资助项目(F2017201020, F2018201115);河北省教育厅科学技术研究重点资助项目(ZD2019021);河北省教育厅青年基金资助项目(QN2017019)

作者简介: 张 峰(1976—),河北大学副教授、博士.研究方向:强化学习和智能决策. E-mail: amyfzhang@yahoo.com

引 文: 张 峰,钱 辉,董春茹,等.随机状态下基于期望经验回放的Q学习算法[J].深圳大学学报理工版,2020,37(2):202-207.

参考文献 / References:

- [1] WATKINS C H. Learning from delayed rewards [D]. London: King's College, 1989: 89-95.
- [2] SUTTON R S, BARTO A G. Reinforcement learning: an introduction [M]. Cambridge, USA: MIT Press, 2018.
- [3] SZEPESVÁRI C. The asymptotic convergence-rate of Q-learning [C]// Advances in Neural Information Processing Systems (NIPS). Cambridge, USA: MIT Press, 1998: 1064-1070.
- [4] HWANG I, YOUNG J J. $Q(\lambda)$ learning-based dynamic route guidance algorithm for overhead hoist transport systems in semiconductor fabs [J]. International Journal of Production Research, 2019(3): 1-23.
- [5] ALIMORADI M R, KASHAN A H. A league championship algorithm equipped with network structure and backward Q-learning for extracting stock trading rules [J]. Applied Soft Computing, 2018, 68: 478-493.
- [6] LIN Longji. Self-improving reactive agents based on reinforcement learning, planning and teaching [J]. Machine Learning, 1992, 8(3/4): 293-321.
- [7] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning [J]. Nature, 2015, 518(7540): 529-533.
- [8] HASSELT H V, GUEZ A, SILVER D. Deep reinforcement learning with double Q-Learning [C]// Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix, USA: AAAI Press, 2016: 2094-2100.
- [9] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms [EB/OL]. (2017-07-20)[2017-08-28]. <https://arxiv.org/abs/1707.06347>
- [10] GU Shixiang, LILLICRAP T, SUTSKEVER I, et al. Continuous deep Q-learning with model-based acceleration [C]// International Conference on Machine Learning. New York. USA: [s. n.], 2016: 2829-2838.
- [11] MAHMOOD A R, van HASSELT H, SUTTON R S. Weighted importance sampling for off-policy learning with linear function approximation [C]// Advances in Neural Information Processing Systems. [S. l.]: Neural Information Processing Systems Foundation, Inc., 2014: 3014-3022.
- [12] SCHAUL T, QUAN J, ANTONOGLOU I, et al. Prioritized experience replay [EB/OL]. (2015-11-18)[2019-04-01]. <https://arxiv.org/abs/1511.05952>.
- [13] VAN HASSELT H. Double Q-learning [C]// Advances in Neural Information Processing Systems. [S. l.]: Neural Information Processing Systems Foundation, Inc., 2010: 2613-2621.
- [14] ARJONA-MEDINA J A, GILLHOFFER M, WIDRICH M, et al. RUDDER: return decomposition for delayed rewards [DB/OL]. (2018-06-20)[2019-09-10]. <https://arxiv.org/abs/1806.07857>.
- [15] ROLNICK D, AHUJA A, SCHWARZ J, et al. Experience replay for continual learning [C]// Advances in Neural Information Processing Systems. [S. l.]: Neural Information Processing Systems Foundation, Inc., 2019: 348-358.

【中文责编: 英 子; 英文责编: 木 柯】

补充材料