**强化学习理论学习与代码实现**

# 马尔科夫决策过程和贝尔曼等式（上）

在本章中将介绍有限马尔科夫决策过程（finite MDPS）的形式化问题，该问题涉及评价反馈，又涉及到关联方面—在不同的情形下选择不同的动作。MDPs是对序贯决策问题的一种典型的形式化表达，其中动作不仅会影响瞬时奖励，还会影响之后的情形或者说是状态，再以此影响未来的奖励。因此MDPs会涉及延迟奖励，需要在瞬时和延迟奖励之间作出权衡。在MDPs中，我们评估各个状态s下各个动作a的值q\*(s,a)，或者估计给定最优动作选择下的各个状态的值V\*(s)。这些依赖状态的量，对于准确地将长期结果的信誉（credit）分配给各个动作选择来说是必不可少的。

## 学习目标

* 理解代理-环境接口；
* 理解什么是马尔科夫决策过程（MDPs）以及如何解释转移图（变迁图）；
* 理解值函数、行为值函数和策略函数；
* 理解值函数、行为值函数的贝尔曼等式和贝尔曼最优等式。

## 代理-环境接口

MDPs旨在为从交互中学习以实现目标的问题提供一个直观的框架。学习者和决策者被称为代理（agent），代理所能交互的其之外的所有事物，称为环境（environment）。两者持续进行交互：代理选择动作，然后环境对动作做出反馈并将新的情形呈现给代理。环境也会带来一种特殊的数值奖励，随着时间的推移，代理通过其行为选择寻求最大化其累积值。

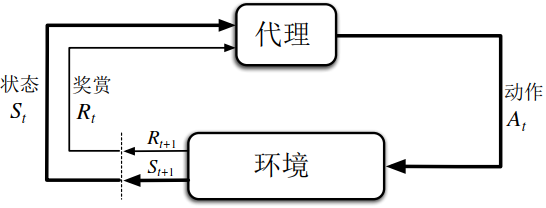


图1 马尔可夫决策过程中的代理-环境交互

更具体地说，代理和环境在一系列离散的时间步t=0,1,2,3,…中交互。在每一个时间步t，代理接收环境状态St∈*S*的某一表达，并在此基础上选择一个动作At∈*A*(s)，一个时间步之后，在某种程度上作为其动作的结果，代理收到一个数值奖励Rt+1∈*R*，并进入下一状态St+1，因此马尔科夫决策过程与代理一起产生可如下的序列或轨迹：

S0,A0,R1,S1,A1,R2,S2,A2,R3,... (1)

在有限MDP中，状态集、动作集与奖励集(*S*;*A*;*R*)，都只有有限个元素。在这一情况下，随机变量Rt与St都有明确的离散概率分布，且该分布仅依赖于前一状态与动作。也就是说，在给定前一状态与动作的情况下，这两个随机变量的特定值s’∈*S*及r∈*R*，出现的概率为：

 (2)

函数p定义了MDP的动态性，等式中等号上的点提醒我们这是一个定义式，而非由先前的定义推导出的事实。动态函数p：*S*×*R*×*S*×*A*→[0;1]是有4个参数的、一般的确定性函数。其中“|”表示条件概率，但在这里其仅用于提醒我们：p指明了选择各个s与a的概率分布，且

 (3)

在马尔科夫决策过程中，由p给出的概率可以完全描述环境的动态（完全可观的），也就是说，St和Rt取得各个可能值的概率只由前一个状态与动作，即St−1和At−1的具体值决定，而不受更早的状态或动作的任何影响。状态必须包括代理与环境的先前交互中所有能对未来产生影响的各方面的信息，如果符合此条件，那么我们称这样的状态拥有马尔科夫性质（Markov property）。

MDP框架抽象而灵活，可以以许多不同的方式应用于不同的问题。例如，时间步无需为真实时间的固定间隔，可以指任意连续的决策和动作阶段。动作可以是底层控制，例如加到机械臂马达上的电压，也可以是高层的决策，例如是否去吃午饭或去读研究生。**再例如，在我需要研究的调度问题中，动作可以是选择调度规则，或是直接选择工序。**一般来说，动作可以是任何我们希望学得如何决策的决定，而状态可以是任何我们认为可以帮助决策的信息。

关于代理与环境的界限，一般遵循的原则是：任何不能被代理任意更改的内容都被认为是在代理的外部，因此是其环境的一部分。我们并没有假设环境中的一切事物对于代理而言是未知的，例如，代理通常知道很多关于奖励是如何作为其行动和状态的函数来计算的，但是我们总是认为奖励计算对代理来说是外部的，因为它定义了代理所面临的任务，因此代理没有对其进行任意修改的能力。事实上，在一些情况下，代理可能知道环境运作的一切信息，但其面临的强化学习任务仍然非常困难，例如我们完全知道魔方这样的益智游戏的规则，但我们仍然无法解决它们。代理与环境间的边界代表了代理绝对控制能力的限制，而非其知识的限度。

## 目标和奖励

在强化学习中，代理的目的或目标可以通过一个被称为“奖励（reward）”的特殊信号进行形式化，且该信号从环境传递给代理。在每个时间步中，奖励是一个简单数字Rt∈*R*。代理的目标是使其获得的奖励总量最大化，这意味着最大化的不是瞬时奖励，而是长期累积奖励。我们可以把这个非正式的想法明确地表述为奖励假说（reward hypothesis）：

我们所说的目标和目的都可以看作是：最大化接收到的标量信号(称为奖励)的累积和的期望值。

使用奖励信号来形式化目标的概念，是强化学习最显著的特征之一。

尽管这种表示刚开始看起来有局限性，但是在实际应用中却被证明是及其灵活且具有广泛适用性。了解这一点的最好方法是考虑它是如何被使用的，或者可以如何被使用的例子。例如，为了让机器人学会走路，研究人员为机器人的每一步向前移动提供了与之成比例的奖励。在让机器人学习如何从迷宫中逃脱时，在逃跑前每走一步，奖励通常是1，以鼓励代理尽可能快地逃离。为了让机器人学会寻找和回收空汽水罐，人们可能会在大部分时间里给它零奖励，然后每回收一个汽水罐就给它+1奖励。当机器人撞到东西或有人对它大喊大叫时，也会给它负面的奖励。对于一个学习下跳棋或国际象棋的代理来说，自然的奖励是赢给予+1，输给予-1，平局和所有非终点位置给予0分。

## 回报和片段

我们已经说过，代理的目标是最大化其长期获得的累积奖励，一般而言，我们希望最大化期望回报，其中回报（return）记为Gt，被定义为奖励序列的某一特定函数。在最简单的情况下回报就是奖励的和：

 (7)

其中T是最后的时间步。这种方法当代理-环境的交互可以自然地分解为子序列时才有意义，这样的子序列称为片段（episode）。每一个片段都有一个终止状态，这个状态之后意味着整个过程重置成初始状态。

另一方面，在许多情形下，代理与环境的交互不能自然地划分为可辨别的分节，而是无限地继续下去。例如，可以很自然地以此种方式形式化一个持续进行的过程控制任务，或一个长生命周期的机器人应用。我们将这样的任务称为连续式任务（continuing task）。对连续式任务来说，上面的回报公式存在一定问题，因为最终时步T=∞,且我们希望最大化的回报可以很轻易地达到无穷（例如假设代理在每一时间步都接收+1的奖励），因此我们将采用一种概念上很复杂但数学表示上很简单的回报定义。

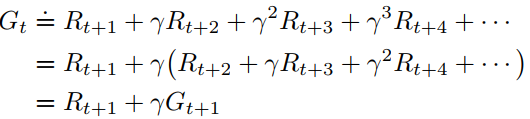
我们需要的另一个概念是折扣（discounting），根据此方法，代理试图选择可以最大化未来累积折扣奖励的动作：

 (8)

其中γ为折扣率且0≤γ≤1。

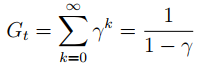
折扣率决定了未来奖励的当前值：在k时间步之后接收的奖励，仅是其瞬时接收时的γk-1倍。如果γ<1,只要奖励序列{Rk}是有限的，上式的无限项之和也会是有限值；如果γ=0,代理只关心最大化瞬时奖励而变得短视：在这种情况下，代理的目标为选择At以仅仅最大化Rt+1。如果代理的每一个动作恰巧仅影响瞬时奖励而不会影响未来奖励，那么一个短视的代理可以通过分别最大化每一个瞬时奖励来最大化上式。但是一般来说，将瞬时奖励最大化会减少获得未来奖励的机会，从而降低回报。随着γ趋近于1，回报目标更强烈地考虑到未来的奖励回报，代理也变得更有远见。

相邻时间步中的回报彼此相关，其相关的方式对强化学习的理论与算法来说十分重要：

 (9)

请注意如果我们定义GT=0，那么这对所有的t<T的时间步都成立，即使终止发生于t+1，这使得由奖励序列计算回报变得容易。

请注意折扣累积回报由无穷多项组成，但如果奖励为非0常数，且γ<1，那么回报值仍是有限的值。例如，如果奖励恒定为+1，那么回报为

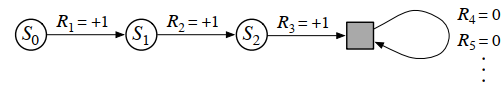
 (10)

## 片段任务和连续任务的统一表示法

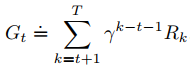
在前面的章节中，已经描述了两种类型的强化学习任务，一种是代理-环境交互可以自然地分解成一系列单独的片段（片段任务），而另一种不能（连续任务）。前面的情况在数学上稍微简单一些，因为每一个动作仅影响其后获得的有效数量的奖励。由于通常需要考虑两种问题，因此有必要建立一种表达，能够同时精确地讨论这两种情况。

为了精确表达片段任务，需要一些额外的表达。片段任务不再是一个长长的时间步序列，而是考虑其为一系列片段，每个片段都包含有限的时间步序列，且每个片段的时间步都从0开始编号。因此，不仅仅用St指代在时间t的状态，还需要用St,i指代片段i在时间t的状态（类似地，有At,i，Rt,i，πt,i，Ti等）。然后，结果表明当我们讨论片段任务时，几乎不会区分不同的片段，而是考虑特定的一个片段，或所讨论的对所有片段都成立。在实际中一般会省略片段的索引，但要知道，使用St时实际上代表的是St,i。

除此之外，还需要另外一个约定以获得涵盖片段任务和情节任务的单一表示。前面已经分别定义了回报作为有限项的和以及无限项的和。如果将片段终止视为进入了一个特殊的吸收状态（absorbing state），即仅转移到自身且获得的所有奖励均为0的状态，那么就可以将两种任务统一起来。例如，考虑如下的转移图：



这里，实心方框代表与片段结束相对应的特殊吸收状态，从S0开始，奖励序列为+1，+1，+1，0,0,0，…，对其进行求和，就能得到与前T个奖励和一样的结果，引入折扣后结论同样成立。可以这样定义一般性的回报（如下），当不需要片段编号时就忽略，同时也保留了当需要求和时γ=1的可能。

 (11)

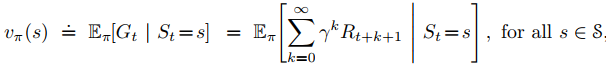
其中T=∞和γ=1都可以（但不能两者同时满足）。

## 策略与值函数

几乎所有的强化学习算法都会涉及值函数-状态函数（或状态-行为对）的评估，也就是评估代理在给定状态下表现好坏（或者给定状态下执行给定动作的表现好坏）。这里表现好坏是根据期望未来奖励定义的，或准确地说是根据期望回报，当然代理未来获得的期望奖励取决于所采取的动作。因此，值函数被定义为同具体的动作方式—即策略相关。

正式地讲，策略是从状态到每个可选动作的概率的映射，如果代理在时间t遵循策略π，那么π(a|s)是当St=s时At=a的概率。如同p一样，π也是普通函数，“|”为每个s∈S定义了a∈A(s)的概率分布。强化学习方法指明了代理 策略如何因经验而改变。

策略π下状态s的值函数记为vπ(s)，为从状态s开始，遵循策略π之后的期望回报。对于MDP，可以有如下vπ的正式定义：

 (12)

其中E代理遵循策略π时随机变量的期望值，t是任意时间步。注意终止状态（有的话）的值总是为0。我们称函数vπ为策略π的状态值函数。

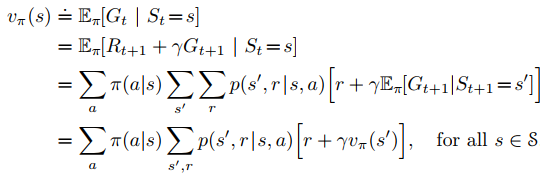
类似地，将策略π下在状态s采取动作a的值qπ(s,a)定义为从s开始，采取动作a，遵循策略π之后的期望回报：

 (13)

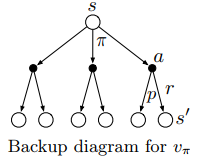
qπ为策略π的行为值函数。

值函数vπ和qπ可以通过经验来估计。例如。如果一个代理遵循策略π，并且对于每一个经历的状态，计算其后实际回报的平均值，那么当状态的经历次数趋于无穷时，平均值将收敛至状态值。如果对每一个状态中的各个动作分别计算均值，那么这些均值也会类似地收敛到qπ(s,a)。这种评估方法称为蒙特卡洛方法（Monte Carlo methods），因为此类方法涉及到对大量实际回报值的随机样本进行平均。

一条贯穿于强化学习与动态规划始终的，关于值函数的基本性质为：其满足类似于中的递归关系。对于任意策略π和任意状态s来说，下面的一致性条件在s的值与其可能后续状态的值间成立：

 (14)

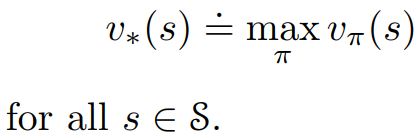
其中隐含着以下几点:动作a来自于集合A(s)，下一状态s’来自于集合S，奖励r来自于集合R。请注意在最后一个等式中，我们是怎么将两个和式—一者为在s’的所有值上求和，另一者为在r的所有值上求和—归并到对两者的所有可能值求和。我们常常使用这种归并后的和式来简化公式。也请注意最后那个表达式，我们可以轻易地将其作为期望值来理解。实际上其为在三个变量a，s’与r的所有值上的和式。对每一个三元组，我们计算其概率π(a|s)p(s’,r|s,a)，并将该概率作为方括号内值的权重，然后在所有的概率上求和来得到期望值。

上式即为针对vπ的贝尔曼方程（Bellman equtation），它表达了状态值与其后续状态状态之间的关系。试想下从某一个状态向前看，看到了所有可能的后一状态的情形，如右图所示。空心圆表示状态，实心圆表示状态-动作对，从顶部的根节点即状态s开始，基于策略π代理可以选择某动作集合（图中为3个动作）中的任意一个。从任意动作出发，环境根据函数p给定的动态性可以响应以下一状态s’中的一个和奖励r。贝尔曼方程对所有的可能进行加权（发生概率）平均，一状态的值一定等于其下一状态的(折扣的)值与随同的奖励这两者之和的期望值。

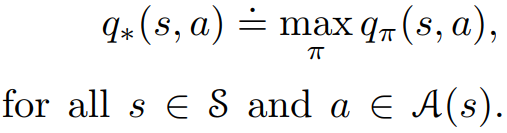
值函数vπ是其贝尔曼方程的唯一解。我们将上面出现过的那种图称为反向图（backup diagram），因为其绘制出的关系构成了更新或反向（backup）操作的基础，而该操作为强化学习方法的核心。这些操作将值的信息从某一状态的后一状态（或动作-状态对），反向传输到该状态（或状态-动作对）。

## 最优策略和最优值函数

粗略地讲，解决强化学习任务意味着找到一个策略，实现长期上的高额奖励。对于有限MDP，可以按照以下方式准确定义最优策略。值函数定义了策略上的偏序关系，对于所有状态，如果一个策略π的期望回报大于等于π’的回报，那么就认为π优于或等于π’，换言之，对于所有状态s∈S，当且仅当vπ(s)≥vπ’(s)时才有π≥π’。至少存在一个策略优于或等于其他策略，这就是最优策略。尽管会存在多个最优，将所有均表示为π\*，它们具有相同的状态值函数，称为最优状态值函数v\*，表示为：

 (15)

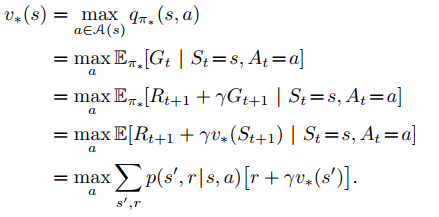
最优策略同样具有相同的最优动作值函数q\*：

 (16)

对于状态-动作对(s,a)，该函数给出了在状态s采取动作a，之后遵循最优策略所能获得的期望回报，因此可以来v\*定义q\*：

 （17）

因为v\*为某一策略的值函数，其必须满足状态值贝尔曼方程给出的自治条件。然而因为是最优值函数，v\*的一致性条件可以一种不参照任何特定策略的特殊形式写出，这就是v\*的贝尔曼方程，或贝尔曼最优方程。直观地看，贝尔曼最优方程表达了这样一个事实：最优策略下一个状态的值一定等于该状态下最好行为的期望回报：

 (18)

最后两个等式是针对v\*的贝尔曼最优方程的两种形式，针对q\*的贝尔曼最优方程为：

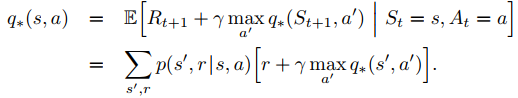
 (19)

图2从图形上展示了针对v∗与q∗的贝尔曼最优性方程中所考虑的未来状态与动作的范围。这些反向图和之前呈现的针对vπ与qπ的反向图类似，只是在选择点处添加了添加了弧线来表明进行了最优选择，而不是计算给定策略下的期望值。图中左侧的方向图从图形上表示贝尔曼最优性方程(18)，而右侧的从图形上表示(19)。

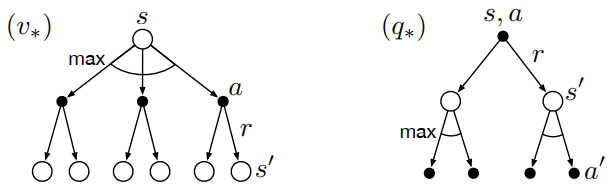


图2 针对v∗和q∗的反向图

对于有限MDP，v\*的贝尔曼最优方程有唯一解。贝尔曼最优方程实际上是方程组，每个状态对应一个方程，所以如果有n个状态，就有n个方程，包含n个未知数。如果环境的动态p已知，那么理论上就可以使用任意一种求解非线性方程组的方法求解v\*，也可以q∗相关的方程组。

一旦求得v\*，就相对容易确定最优策略。对于每一个状态s，在贝尔曼方程中可以有一个或多个动作获取得最大，任何只给这些动作分配非零概率的策略都是最优策略。你可以将此视为单步搜索，如果你已经知道了最优值函数v\*，那么在单步搜索后表现最好的动作即为最优动作。另一种理解视角为：任何对最优值函数贪婪的策略都是最优策略。在计算机科学的领域中，术语贪婪用于描述仅基于局部或即时的考量进行选择的搜索或决策过程，而不考虑这样的选择是否会在将来阻碍我们做出更好的选择。因此，其用于形容仅基于短期结果来进行动作选择的策略。v\*的优美之处在于，如果我们将其用于对动作的短期结果进行评估——或更确切地说，单步结果——那么从我们所关注的长期的角度看，贪婪策略事实上是最优的，因为v\*已经将所有可能的未来动作所产生的关于奖赏的影响考虑在内。通过v\*，最优的长期期望回报转换成了可以在每个状态上通过局部、即时的计算而获得的量。因此，单步搜索即可得到长期而言最优的动作。

如果我们知道了q∗，那么选择最优动作甚至可以更简单。有了q∗，代理甚至都不必进行单步搜索：对任一状态s，代理可以轻松地发现最大化q∗(s,a)的动作。动作值函数实际上已经存储了所有单步搜索的结果。其以值的形式提供了长期期望回报，每一状态-动作对都可以在局部即时获取该值。因此，以构建动作-状态对上而非仅状态上的函数为代价，最优动作值函数使得我们可以选择最优动作而不必了解任何可能的后一状态与其值的信息，即不必了解关于环境的动态的任何信息。

## 最优与近似

我们已经定义了最优值函数与最优策略，很明显，学得了最优策略的代理将表现得非常好，但在实际中这很少发生。在我们所关心的各种问题中，只有付出极大的计算代价才能获得最优策略，最优性仅是代理可以在不同程度上接近的一种理想化。正如我们前面所讨论的，即使我们拥有对环境的完整而准确的模型，通过单纯地解贝尔曼最优性方程来得到最优策略通常也是不现实的。举个例子，棋类游戏，如国际象棋，仅占到人类各种经验的极小的一部分，但是规模庞大的定制计算机也无法计算出最优下法。代理所面临的问题中极为关键的一个方面为其拥有多少算力，更具体地说，其能在一个时步中完成多少计算。

内存的大小也是一个重要的制约，为了建立对值函数、策略以及模型的估计，通常需要大量的内存。对于状态集小而有限的任务，那么使用数组或表格来进行估计是可行的，其中各个状态(或状态-动作对)都有对应的表项，我们将此称为表格型情形，并将对应的方法称为表格型方法。而在我们感兴趣的许多实际情形下，状态的数量太多而远不能放入表格内。在这种情形下，必须使用某种简洁的参数化函数表示方法，来对函数进行近似。

关于强化学习问题的框架迫使我们将就于使用近似，然而这也为我们提供了一些实现有用近似的独特机会。例如，在近似求解最优动作时，可能有很多状态代理遇到的概率非常小，以至于在这些状态上就算选择了次优动作也对代理接收到的奖励总量几乎没有影响。再例如，Tesauro的西洋双陆棋程序拥有卓越的技巧，尽管其在与高手的对局中对未出现过的棋面有可能做出极糟糕的决策。事实上，极有可能TD-Gammon在游戏状态集中很大一部分都做出糟糕的决策。强化学习的在线特性使得可以以这样的方式来近似最优策略：投入更过精力为经常经历的状态学习好的决策，而以牺牲掉不常见状态为代价。这是将化学习区别于其他近似求解MDP方法的一个关键特性。

## 总结

强化学习是指从交互中学习如何决策以达到目标。强化学习代理及其环境在一系列离散时间步上交互。动作是由代理做出的选择；各状态是作出选择的基础；奖励是评估选择的基础。代理内部的一切都是代理完全知道和可控制的；代理外部的一切是不完全可控的，但代理可能完全已知也可能不完全可知。策略是一种随机规则，根据该规则，代理选择作为状态函数的动作。代理的目标是使其在一段时间内获得的报酬最大化。

当上面描述的强化学习设置用定义良好的转移概率来表示时，它就构成了一个马尔可夫决策过程（MDP）。有限MDP具有有限状态、动作和奖励集合的MDP。

回报是未来的奖赏的函数，也是代理寻求最大化的目标(从期望的角度)。取决于任务的特性以及我们是否希望对延迟的奖励进行折扣，其有数种不同的定义。非折扣形式适合于片段任务，在片段任务中代理与环境间的交互可以自然地划分成不同的片段；折扣形式适合于连续任务，在连续任务中交互不能自然地划分为不同的片段，而是无限地持续下去。我们希望统一两类任务中回报的定义，使得同一套公式可以同时适用于片段情形与连续情形。

一略的值函数为每个状态(或状态-动作对)给出了在该策略下，从该状态(或状态-动作对)起可以获得的期望回报。最优值函数为每个状态(或状态-动作对)给出了各个策略中所能获得的最大期望回报，值函数为最优的策略即为最优策略。对一个给定的MDP，虽然针对状态或状态-动作对的最优值函数是唯一的，但可能有多个最优策略。任何就最优值函数而言贪婪的策略一定是最优策略。贝尔曼最优方程为最优值函数必须满足的特殊一致性条件，并且从理论上说，我们可以从中解得最优值函数，然后利用最优值函数我们可以相对轻松地确定最优策略。

即使代理有完整而准确的环境模型，代理通常无法进行足够多的计算来彻底对其进行利用，在实际感兴趣的多数情形下，状态通常太多而无法放入表中，此时必须进行函数近似。