Python金融计算（第五讲）：金融市场环境建模与强化学习

谢文杰

华东理工大学 金融学系

2022年春

目录

目录

强化学习简介

强化学习问题，是优化问题，也是控制问题，更是智能体学习问题。随机优化产生于各种各样的问题中，从博弈游戏，到供应链优化等，随机优化思想在不同的领域中得到发展。

- 金融市场中进行复杂交易研究，可以选择强化学习作为技术支撑，在金融金融资产或者证券交易过程就是一个典型的序贯决策过程，而强化学习算法就是为了解决序贯决策问题。
- 通过买卖交易收益最大化，是一个典型的最优化问题。
- 金融数学中有一类非常特殊的随机优化问题：最佳止损问题，其中止损行为可能是出售金融资产或行使期权。

马尔科夫决策过程

马尔科夫决策过程是强化学习模型框架。理解马尔科夫决策过程之前，需要了解什么是马尔科夫过程和马尔科夫性。在随机过程中，如果未来状态只与当前状态有关，而不受历史状态影响，则说明随机过程满足马尔科夫性，即为马尔科夫过程，数学语言描述如下：

定义

如果离散随机过程满足

- $\mathbb{P}(s_{t+1} \mid s_t) = \mathbb{P}(s_{t+1} \mid s_t, \ldots, s_0)$,

则离散随机过程具有马尔科夫性质。

s_t表示t时刻状态。$\mathbb{P}(s_{t+1} \mid s_t)$表示$t$时刻状态$s_t$转移到$t+1$时刻状态$s_{t+1}$的条件概率。$\mathbb{P}(s_{t+1} \mid s_t, \ldots, s_0)$表示在历史状态$s_t, \ldots, s_0$条件下，在$t+1$时刻转移到状态$s_{t+1}$的条件概率。

离散状态之间转移概率

满足马尔科夫性的随机过程能够得到离散状态之间转移概率矩阵P，如下所示：

$$
\begin{array}{c}
 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ \vdots \\ s_{n-1} \\ s_n
\end{array}
\begin{array}{c}
\begin{array}{cccccc} s_1 & s_2 & s_3 & \cdots & s_{n-1} & s_n \end{array} \\
\begin{pmatrix}
p_{11} & p_{12} & p_{13} & \cdots & p_{1,n-1} & p_{1n} \\
p_{21} & p_{22} & p_{23} & \cdots & p_{2,n-1} & p_{2n} \\
p_{31} & p_{32} & p_{33} & \cdots & p_{3,n-1} & p_{3n} \\
p_{41} & p_{42} & p_{43} & \cdots & p_{4,n-1} & p_{4n} \\
p_{51} & p_{52} & p_{53} & \cdots & p_{5,n-1} & p_{5n} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
p_{n-1,1} & p_{n-1,2} & p_{n-1,3} & \cdots & p_{n-1,n-1} & p_{n-1,n} \\
p_{n1} & p_{n2} & p_{n3} & \cdots & p_{n,n-1} & p_{nn}
\end{pmatrix}
\end{array}
$$

矩阵中元素p_{ij}表示从状态s_i转移到状态s_j的转移概率。

离散状态之间转移概率

在状态转移概率矩阵的基础上，能够对随机过程进行大量研究和分析。状态转移概率矩阵有一些基本性质：

$$p_{ij} \geq 0, \tag{1}$$

$p_{ij} = 0$说明从状态s_i不能转移到状态s_j。将马尔科夫过程建模成复杂网络模型，每个状态为网络中一个节点，状态之间的转移行为可以建模成网络连边，$p_{ij} = 0$说明节点i和j之间没有连边$i \to j$，因此状态转移概率矩阵就是有向加权网络的邻接矩阵。如果不满足马尔科夫性质，可以构建状态之间的复杂高阶网络模型。状态转移概率矩阵有另一重要基本性质为：

$$\sum_{j=1}^{n} p_{ij} = 1. \tag{2}$$

马尔科夫奖赏过程（Markov Reward Process，MRP）

马尔科夫随机过程中，状态转移时能够获得回报收益，可以表示成马尔科夫奖赏过程，定义如下

定义

马尔科夫回报过程可以定义为一个四元组 $(\mathcal{S}, P, R, \gamma)$，其中:

- $\mathcal{S}$ 表示状态集合，
- $P : \mathcal{S} \times \mathcal{S} \rightarrow [0, 1]$状态转移函数或状态转移矩阵，
- $R : \mathcal{S} \times \mathcal{S} \rightarrow \mathcal{R}$奖赏函数，$\mathcal{R}$ 为连续区间，$R_{\max} \in \mathbb{R}^+$ (e.g., $[0, R_{\max}]$)，
- $\gamma \in [0, 1)$ 是一个折扣系数。

马尔科夫决策过程（Markov Decision Process ， MDP）

在马尔科夫奖赏过程的基础上加入智能体行为$\mathcal{A}$，可表示成离散马尔科夫决策过程（Markov Decision Process ， MDP）：

定义

马尔科夫决策过程是一个五元组 $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$表示，其中:

- $\mathcal{S}$ 表示状态集合，
- $\mathcal{A}$ 表示动作集合，
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$是状态转移函数，$P(s_t, a_t, s_{t+1})$是状态转移概率，
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{R}$是奖赏函数， $\mathcal{R}$ 为连续区间，$R(s_t, a_t, s_{t+1}) \in \mathcal{R}$， $R_{\max} \in \mathbb{R}^+$ (e.g., $[0, R_{\max}]$),
- $\gamma \in [0, 1)$ 是一个折扣系数。

目录

动态规划方法

动态规划方法是求解马尔科夫决策过程的经典方法。求解马尔科夫决策过程之前需要对一些基本的概念进行理解。马尔科夫决策过程五元组 $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$中，$P$表示状态转移概率，在有些环境下状态转移概率和动作无关，状态转移概率可以写成

$$P_{ss'} = P(S_{t+1} = s' \mid S_t = s) \tag{3}$$

当状态转移概率与动作相关时，$P^a_{ss'}$可以表示为:

$$P^a_{ss'} = P(S_{t+1} = s' \mid S_t = s, A = a) \tag{4}$$

其中S_t 和 S_{t+1} 分别表示相邻时间t和$t+1$时刻环境状态。$P^a_{ss'}$ 表示当前t时刻状态s下，经过动作a后（$a \in \mathcal{A}$），下一个时刻$t+1$转移到状态s'的概率。

策略函数

强化学习核心目标是学习到一个策略π，策略π可以建模成一个函数，将随机过程的状态空间映射到动作空间，表示成$\pi:\mathcal{S}\rightarrow\mathcal{A}$。策略$\pi$与马尔科夫决策过程中行动直接相关，行动影响状态转移和奖励回报。在复杂环境模型确定的情况下，策略输出的动作直接影响了奖励回报。

一般来说，复杂环境模型包括了状态转移函数和回报函数。在问题求解之前需要知道状态转移函数和回报函数，然后通过算法求解最优策略函数π。策略函数π可分成两种类型，

- 一种是随机性策略型，如$\pi:\mathcal{S}\times\mathcal{A}\rightarrow[0,1]$；
- 一种是确定性策略型，如$\pi:\mathcal{S}\rightarrow\mathcal{A}$。

策略函数

从另一个角度也可以将策略分成是连续性策略和离散型策略。下面采用随机策略函数$\pi(a \mid s)$进行举例分析，将状态转移函数进行重写：

$$P^{\pi}_{ss'} = \sum_{a \in A} \pi(a \mid s) P^{a}_{ss'} \tag{5}$$

其中$\pi(a \mid s)$表示状态s下执行不同动作行为a的概率。状态s下可以采取不同的动作行为a，策略函数π输出不同动作行为a的概率$\pi(a \mid s)$，通过遍历所有动作行为累计求和状态转移概率$P^{a}_{ss'}$，最终得到了给定策略函数π时，从当前时刻状态s转移到下一个状态s'的概率$P^{\pi}_{ss'}$。

奖励函数

奖励函数 R 决定了智能体在环境状态s下执行动作a后得到的奖励值R_s^a，可以表示成：

$$R_s^a = E[R_t \mid S_t = s, A_t = a] \tag{6}$$

结合给定的策略函数π，从当前时刻状态s下不同动作下奖励回报期望为：

$$R_s^\pi = \sum_{a \in A} \pi(a \mid s) R_s^a . \tag{7}$$

累积回报

一般来说，马尔科夫决策过程是一个连续决策过程。从初始状态开始，智能体执行动作，获得即时奖励回报，跳转到下一个状态，重新执行动作，获得即时奖励回报，如此循环反复直至终止状态。智能体最终期望获得较高的累积奖励回报，而不是只关心某次单独行动的即时奖励回报。因此定义从当前时刻状态s开始直至终止状态，智能体获得的累计奖励回报为：

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k} \qquad (8)$$

其中γ是折扣系数，且$\gamma \in [0, 1)$。与金融分析中折扣因子类似。从概率角度理解，离当前时刻越远的行为所获得的奖励，存在更大的不确定性，对当前行动的收益影响应该减小权重。对于无限长时间的累积收益情况，G_t不会出现无穷大情况。

状态值函数

在累积回报G_t基础上，可以定义状态值函数$V_\pi(s)$，$V_\pi(s)$ 表示从状态s出发，基于当前策略函数π能够获得的期望回报，具体数学表示为：

$$V_\pi(s) = E_\pi[G_t \mid S_t = s] \tag{9}$$

状态值函数$V_\pi(s)$是状态s获得累积回报G_t的期望，衡量不同状态s的价值，引导智能体通过状态转移跳转到高价值状态。
状态值函数$V_\pi(s)$进行简单推到可以得到

$$\begin{aligned} V_\pi(s) &= E_\pi[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \ldots \mid S_t = s] \\ &= E_\pi[R_t + \gamma(R_{t+1} + \gamma^1 R_{t+2} + \gamma^2 R_{t+3} + \ldots) \mid S_t = s] \\ &= E_\pi[R_t + \gamma G_{t+1} \mid S_t = s] \\ &= E_\pi[R_t + \gamma V_\pi(S_{t+1}) \mid S_t = s] \end{aligned} \tag{10}$$

G_{t+1}的期望值用状态值函数$V_\pi(S_{t+1})$替换，S_{t+1}为随机变量。

状态-动作值函数

同样，可以定义状态-动作值函数$Q_\pi(s,a)$，$Q_\pi(s,a)$ 表示从状态s出发，基于当前策略函数π执行动作a能够获得的期望回报，具体数学表示为：

$$Q_\pi(s,a) = E_\pi[G_t \mid S_t = s, A_t = a] \tag{11}$$

对状态值函数$Q_\pi(s,a)$进行简单推到可以得到

$$\begin{aligned} Q_\pi(s) &= E_\pi[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \ldots \mid S_t = s, A_t = a] \\ &= E_\pi[R_t + \gamma(R_{t+1} + \gamma^1 R_{t+2} + \gamma^2 R_{t+3} + \ldots) \mid S_t = s, A_t = a] \\ &= E_\pi[R_t + \gamma G_{t+1} \mid S_t = s, A_t = a] \\ &= E_\pi[R_t + \gamma Q_\pi(S_{t+1}) \mid S_t = s, A_t = a]. \end{aligned} \tag{12}$$

其中G_{t+1}的期望值用动作-状态值函数$Q_\pi(S_{t+1})$替换，S_{t+1}为随机变量。

状态-动作值函数与状态值函数关系

通过状态值函数和状态-动作值函数定义，可以发现两者之间具有紧密的联系：

$$V_\pi(s) = \sum_{a \in A} \pi(a \mid s) Q_\pi(s, a) \tag{13}$$

上式说明状态s的值函数是状态-动作值函数在策略函数π下执行动作a获得累积收益回报期望值。同样可以得到

$$Q_\pi(s, a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_\pi(s') \tag{14}$$

上式说明状态-动作值函数等于动作a的即时奖励值加上下一个状态s'的值函数的期望$\sum_{s' \in S} P_{ss'}^a V_\pi(s')$，因为是下一个时刻状态值，需要乘上折扣因子$\gamma$。

状态-动作值函数与状态值函数关系

将公式 14代入公式 13，可以得到状态值函数另外一种更加复杂的表示形式

$$V_\pi(s) = \sum_{a \in A} \pi(a \mid s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_\pi(s') \right). \qquad (15)$$

公式中只包含了状态值函数 V_π，无状态-动作值函数 Q_π。

将公式 13代入公式 14，可以得到状态-动作值函数另外一种更加复杂的表示形式

$$Q_\pi(s, a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \left(\sum_{a' \in A} \pi(a' \mid s') Q_\pi(s', a') \right). \qquad (16)$$

公式中只包含了状态-动作值函数 Q_π，无状态值函数 V_π。

状态-动作值函数与状态值函数关系

- 强化学习核心是构建智能体策略函数，通过策略函数输出动作获得累积收益。
- 在更新策略过程中，需要考虑状态-动作值函数Q_π、状态值函数V_π、状态转移函数$P:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$、回报奖励函数$R:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathcal{R}$等。
- 强化学习与监督学习、无监督学习的区别也在于求解过程复杂度更高，且训练过程更加具有挑战。
- 在实际应用和求解过程中会引入一些假设和近似，使得在模型求解和优化实现上更加具有可行性。

Bellman方程

简单的马尔科夫奖赏过程（Markov Reward Process，MRP)中，针对MRP的Bellman方程为：

$$V_\pi(s) = E_\pi[R_t + \gamma V_\pi(S_{t+1}) \mid S_t = s] \tag{17}$$

方程中不包含策略函数和动作。

Bellman方程

下面将简单推导MDP的Bellman方程，写出状态值函数的矩阵形式：

$$\begin{aligned} V_\pi(s) &= \sum_{a\in A}\pi(a\mid s)\left(R_s^a+\gamma\sum_{s'\in S}P_{ss'}^a V_\pi(s')\right) \\ &= \sum_{a\in A}\pi(a\mid s)R_s^a+\gamma\sum_{a\in A}\pi(a\mid s)\left(\sum_{s'\in S}P_{ss'}^a V_\pi(s')\right) \end{aligned} \tag{18}$$

将公式（7）即$R_s^\pi=\sum_{a\in A}\pi(a\mid s)R_s^a$代入后得到

Bellman方程

$$V_\pi(s) = R_s^\pi + \gamma \sum_{a\in A} \pi(a \mid s) \left(\sum_{s'\in S} P_{ss'}^a V_\pi(s') \right)$$
$$= R_s^\pi + \gamma \sum_{s'\in S} \left(\sum_{a\in A} \pi(a \mid s) P_{ss'}^a \right) V_\pi(s') \tag{19}$$

将公式 5即$P_{ss'}^\pi = \sum_{a\in A} \pi(a \mid s) P_{ss'}^a$代入后得到

$$V_\pi(s) = R_s^\pi + \gamma \sum_{s'\in S} P_{ss'}^\pi V_\pi(s') \tag{20}$$

Bellman方程

对于不同的状态s_1，s_2，...，都可以写出类似的Bellman方程

$$
\begin{aligned}
V_\pi(s_1) &= R^\pi_{s_1} + \gamma \sum_{s' \in S} P^\pi_{s_1 s'} V_\pi(s') \\
V_\pi(s_2) &= R^\pi_{s_2} + \gamma \sum_{s' \in S} P^\pi_{s_2 s'} V_\pi(s') \\
&\cdots \\
V_\pi(s_n) &= R^\pi_{s_n} + \gamma \sum_{s' \in S} P^\pi_{s_n s'} V_\pi(s')
\end{aligned} \tag{21}
$$

Bellman方程

将上述方程组写出矩阵形式：

$$\begin{bmatrix} V_\pi(s_1) \\ V_\pi(s_2) \\ V_\pi(s_3) \\ \vdots \\ V_\pi(s_n) \end{bmatrix} = \begin{bmatrix} R_\pi(s_1) \\ R_\pi(s_2) \\ R_\pi(s_3) \\ \vdots \\ R_\pi(s_n) \end{bmatrix} + \gamma \begin{bmatrix} P_{11}^\pi & P_{12}^\pi & \cdots & P_{1n}^\pi \\ P_{21}^\pi & P_{22}^\pi & \cdots & P_{2n}^\pi \\ P_{31}^\pi & P_{32}^\pi & \cdots & P_{3n}^\pi \\ \vdots & \vdots & \cdots & \vdots \\ P_{n1}^\pi & P_{n2}^\pi & \cdots & P_{nn}^\pi \end{bmatrix} \begin{bmatrix} V_\pi(s_1) \\ V_\pi(s_2) \\ V_\pi(s_3) \\ \vdots \\ V_\pi(s_n) \end{bmatrix} \tag{22}$$

Bellman方程

进一步用矩阵符号表示，并求解可得：

$$\begin{aligned} V_\pi &= R_\pi + \gamma P_\pi V_\pi \\ (I - \gamma P_\pi)V_\pi &= R_\pi \\ V_\pi &= (I - \gamma P_\pi)^{-1} R_\pi . \end{aligned} \tag{23}$$

V_π和R_π为列向量，P_π为状态转移概率矩阵。状态值函数V_π可以基于状态转移函数$P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$、回报奖励函数$R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{R}$直接求解。

直接求解的挑战：

- 矩阵太大，存储资源不足。
- 矩阵是否可逆？
- 矩阵求逆复杂度过大。

目录

Q-learning

强化学习目标是训练智能体获得一个智能策略。马尔科夫决策过程为强化学习模型基本结构，求解最优决策策略，是强化学习的核心。强化学习是机器学习的重要分支，继承了机器学习众多算法基本性质。如从哪里学，学什么，怎么学都是需要解决的重要问题。

- 强化学习从数据中学习，更加具体来说是从智能体和环境交互获得的经验数据中学习。
- 智能体和环境的交互过程，可以看做是统计采样的过程。在交互过程中智能体不断尝试，不断采样，不断学习，不断优化自身策略函数。
- 从哪里学？从算法的输入数据中学。
- 学什么？智能体学习策略函数。
- 如何学？强化学习算法的核心主题，也是重点内容。

Q-learning

下面简单介绍时序差分Q-learning算法的伪代码：

Algorithm 11: 时序差分 Q-learning 算法的伪代码

Input: 状态空间 $\mathcal{S}$, 动作空间 $\mathcal{A}$, 折扣系数 γ，以及环境 E，初始化的状态-动作值函数 $Q(s,a)=0$，初始采样策略采用 $\pi(a\mid s)=\frac{1}{|A|}$。

Output: 最优策略 π^*

for $k=0,1,2,3,4,5,...$ **do**
　% 每次循环针对一条轨迹
　初始化状态 s
　for $t=0,1,2,3,4,...,T$ **do**
　　% 采用 ϵ-贪心策略生成轨迹中的一步

$$\pi(s,a)=\begin{cases}1-\epsilon+\frac{\epsilon}{|A|}, & a=\arg\max_a Q(s,a)\\ \frac{\epsilon}{|A|}, & a\neq\arg\max_a Q(s,a)\end{cases}, \tag{4.50}$$

　　产生一步轨迹 $\langle s,a,r,s'\rangle$
　　其中 a 和 r 是基于 ϵ-贪心策略产生的动作和即时奖励。
　　$Q(s,a)\leftarrow Q(s,a)+\alpha(r+\gamma\max_{a'}Q(s',a')-Q(s,a))$
　　智能体进入下一个状态 $s=s'$
　　if s 为终止状态 **then**
　　　开始下一条轨迹采样

% 计算最优策略
for $s\in S$ **do**
　$\pi^*(s)=\arg\max_a Q(s,a)$

Q-learning

算法学习过程为迭代过程，通过智能体不断与环境交互来获得经验数据。每次循环中都会采样一条轨迹，与蒙特卡罗算法不同之处在于，Q-learning算法不会等到采样一整条轨迹后再进行学习和更新策略或值函数，而是采样一步之后，得到一步经验数据$\langle s, a, r, s' \rangle$，其中$a$是基于$\epsilon$-贪心策略产生的动作，$r$和$s'$是环境接收到动作$a$后返回的即时奖励和下一个状态。智能体与环境交互的采样过程使用的ϵ-贪心策略如下所示：

$$\pi(s, a) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|}, & a = \arg\max_a Q(s, a) \\ \frac{\epsilon}{|\mathcal{A}|}, & a \neq \arg\max_a Q(s, a) \end{cases}, \tag{24}$$

Q-learning

Q-learning算法不会等到采样一整条轨迹后再进行学习和更新策略或值函数，而是采样一步之后，得到一步经验数据$\langle s, a, r, s' \rangle$，其中$a$是基于$\epsilon$-贪心策略产生的动作，$r$和$s'$是环境接收到动作$a$后返回的即时奖励和下一个状态。
运用更新公式

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)) \tag{25}$$

更新之后，智能体进入下一个状态$s = s'$，直到状态s为终止状态，重新开始下一条轨迹采样。

目录

基于强化学习的智能交易系统

基于强化学习的智能交易系统框架如图 1所示。

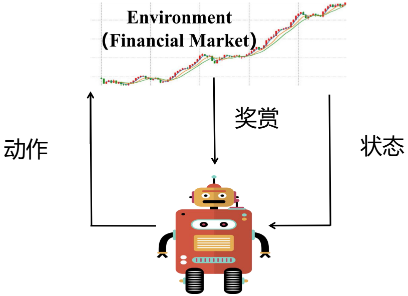


图: 基于强化学习的智能交易系统框架示意图

图展示了强化学习智能体进行智能决策的过程，其中最核心模块为智能体和环境模型（复杂金融市场模型）。

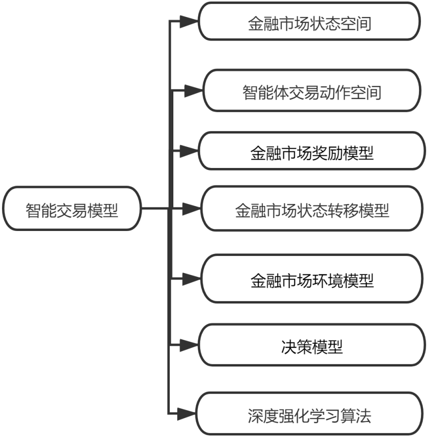
金融市场环境中马尔科夫决策过程建模

定义

基于离散马尔科夫决策过程（Markov Decision Process，MDP)的金融市场环境模型可用一个六元组 $(\mathcal{S}, \mathcal{A}, P, R, \gamma, H)$表示，其中:

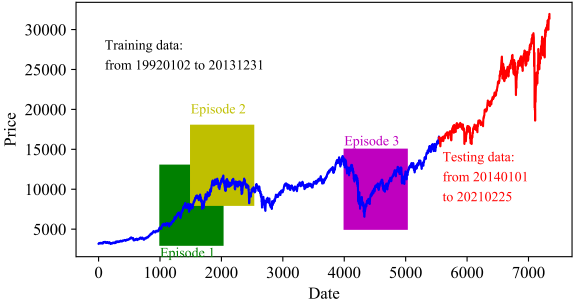
- $\mathcal{S}$ 表示金融市场环境状态集合，
- $\mathcal{A}$ 表示智能体动作集合，
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$是金融市场环境状态转移函数。
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{R}$是金融市场环境奖赏函数， $\mathcal{R}$ 为连续的区间， $R(s_t, a_t, s_{t+1}) \in \mathcal{R}$， $R_{\max} \in \mathbb{R}^+$ (e.g., $[0, R_{\max}]$),
- $\gamma \in [0, 1)$ 是折扣系数。
- H 是投资期限。

金融市场环境中马尔科夫决策过程建模



金融市场状态空间

基于离散马尔科夫决策过程（Markov Decision Process，MDP)的金融市场环境模型六元组中，变量 $\mathcal{S}$ 表示金融市场状态空间。如何刻画金融市场状态，一般采用价格时间序列数据。决策过程中智能体不可能考虑全部的金融市场信息，人类投资者也不可能知晓和处理所有市场信息。



智能体动作空间

金融市场环境模型六元组中，$\mathcal{A}$表示智能体的动作空间，即为金融交易动作。基于环境变量智能体依据当前策略给出交易信号。智能体动作空间描述智能体与金融市场环境交互动作。一般来说智能体动作可以分成2类，一类是连续型的，一类是离散型的。离散型的智能体交易动作$a \in \mathcal{A}$可表示为:

$$a = \begin{cases} -1, & \text{sell} \\ 0, & \text{hold} \\ 1, & \text{buy} \end{cases}, \tag{26}$$

其中 $-1, 0, 1$ 分别表示买入、持有和卖出金融资产。

金融市场奖励模型

金融市场环境模型六元组中，奖励函数 $R(s,a,s')$是智能体学习到优秀策略的重要依据。奖励函数有很多的表示形式，可以由不同部分组成。对于需要奖励的行为增加奖励值，达到引导和训练智能体做出更多类似行为的目的。奖励函数 $R(s,a,s')$可以定义为投资组合总市值的变化量：

$$R(s,a,s') = v' - v \tag{27}$$

当智能体在金融市场状态 s下执行动作 a，转化到金融市场状态 s'， v' 和v 分别表示状态 s' 和s的智能体资产市值。奖励函数 $R(s,a,s')$也可以定义为投资组合市值的对数收益：

$$R(s,a,s') = \log(v') - \log(v). \tag{28}$$

金融市场状态转移模型

- 金融市场环境模型六元组中，$P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ 表示了市场环境的转移函数。在复杂金融系统中无法给出市场状态转移函数的显示形式，因此需要构建一个虚拟的市场环境。
- 金融市场环境模型在状态转移过程中，时间粒度可以重新设计。t时刻金融市场状态可以转移到下一个状态，即$t + \Delta t$时刻状态，其中Δt可以是1天、3天、5天等，或者粒度更细。Δt决定了智能体进行决策的间隔时间，也是智能体调仓的间隔时间。为了减少智能体调仓次数，可以设置较大的Δt。

金融市场折扣因子和市场摩擦

金融市场环境模型六元组中，$\gamma \in [0,1)$ 是折扣因子，与金融中折现因子具有同样的含义。智能体学习过程中每个行为的收益不一样，为了能够学习到获得最大效用的行为策略，需要估计不同行为的期望价值，而期望价值为累计奖励，离当前行为越远的奖励影响越小，所以用折扣系数来实现：

$$R = \sum_{t=0}^{H} \gamma^t R_t. \tag{29}$$

决策模型

金融市场环境数据异常复杂且不可穷尽，模型状态建模过程中只能够对部分可获取的金融市场信息进行分析，此过程可以如图所示。

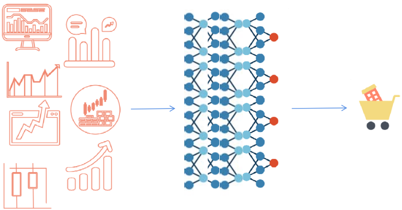


图: 决策模型

图左边部分表示了金融市场环境状态信息，可以是金融技术性指标、宏观经济指标、微观经济指标、舆情指标等。