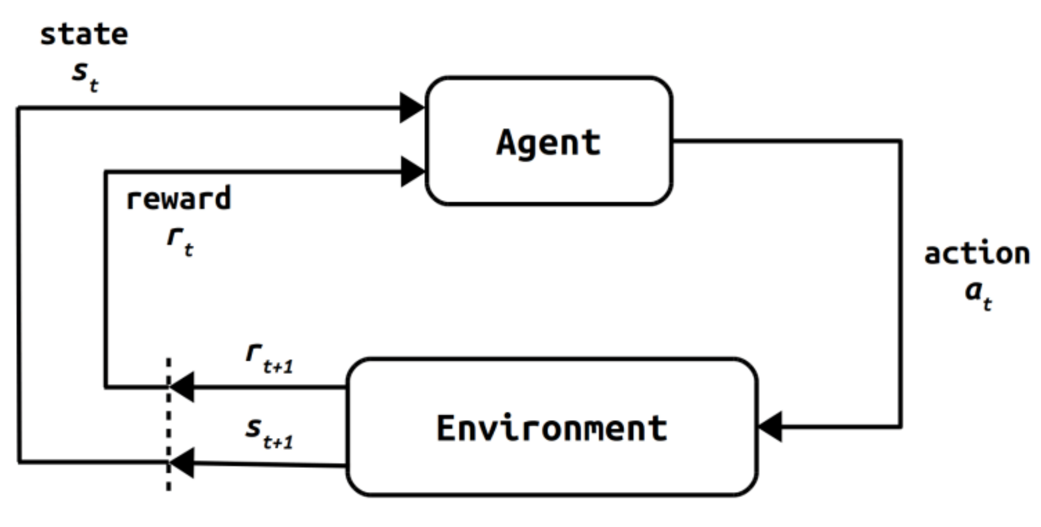
Reinforcement Learning for Portfolio Management

强化学习

- 环境
- 智能体



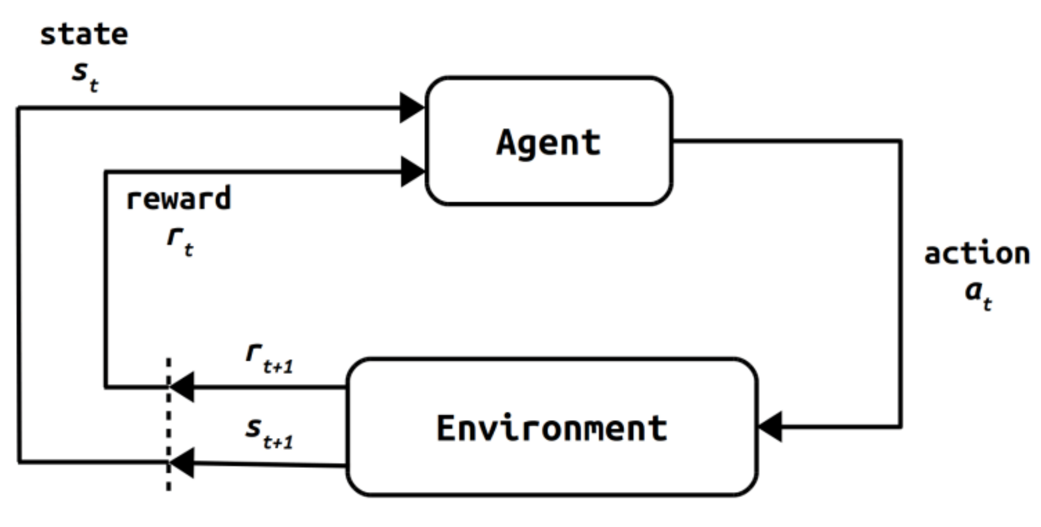
R.S. Sutton, A.G. Barto,
Reinforcement Learning: An
Introduction,
MIT Press, Cambridge, MA, 1998

状态空间

- 50*11
- 50: 可投资的股票数目
- 11: 股票的维度：['zopen', 'zhigh', 'zlow', 'zadjcp', 'zclose', 'zd_5', 'zd_10', 'zd_15', 'zd_20', 'zd_25', 'zd_30']

奖励

- reward = portfolio_value(t+1) - portfolio_value(t)



state s_t

reward r_t

Agent

action a_t

r_{t+1}

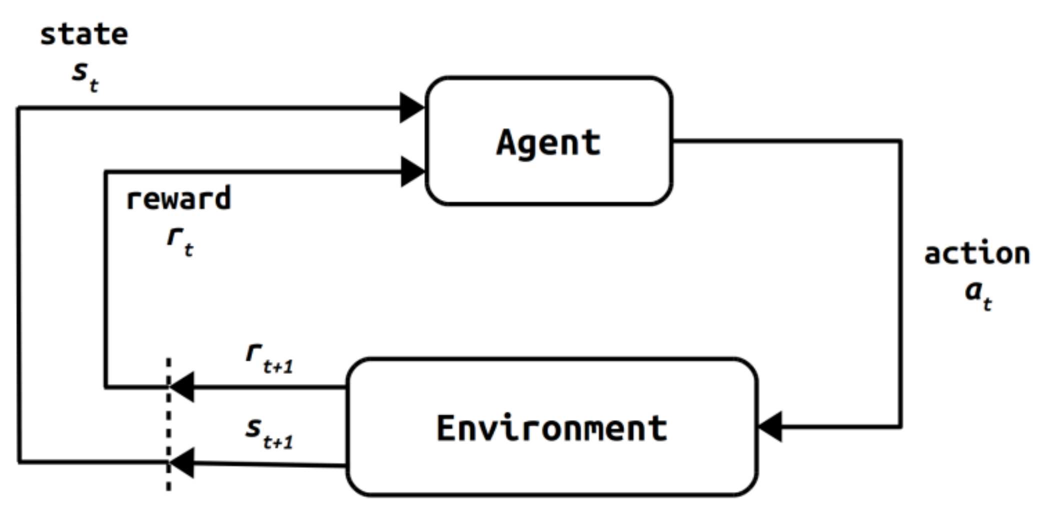
s_{t+1}

Environment

R.S. Sutton, A.G. Barto,
Reinforcement Learning: An
Introduction,
MIT Press, Cambridge, MA, 1998

行动空间

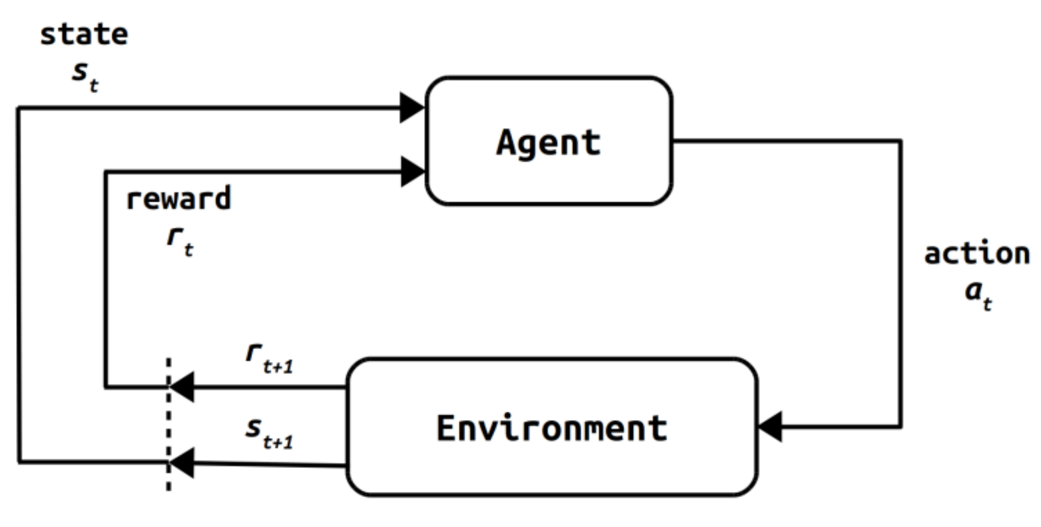
- (51,)
- 50: 可投资的股票数目+1:不投资



R.S. Sutton, A.G. Barto,
Reinforcement Learning: An Introduction,
MIT Press, Cambridge, MA, 1998

智能体

- 状态空间(50,11) ->行动空间 (51,)
- 用反馈的奖励来训练智能体

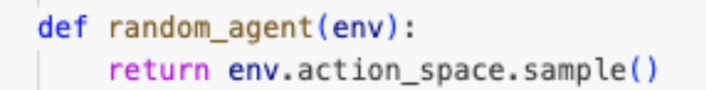


R.S. Sutton, A.G. Barto,
Reinforcement Learning: An Introduction,
MIT Press, Cambridge, MA, 1998

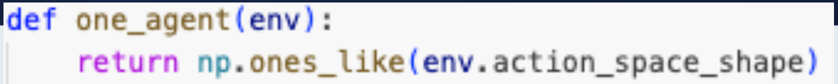
智能体设计

- Policy Optimization: 策略优化
  – Policy Gradient: 策略梯度
- State Optimization: 状态优化
  – Q-learning: Q学习
- Actor-Critic Methods: 演员-评论家方法

随机策略揭示问题

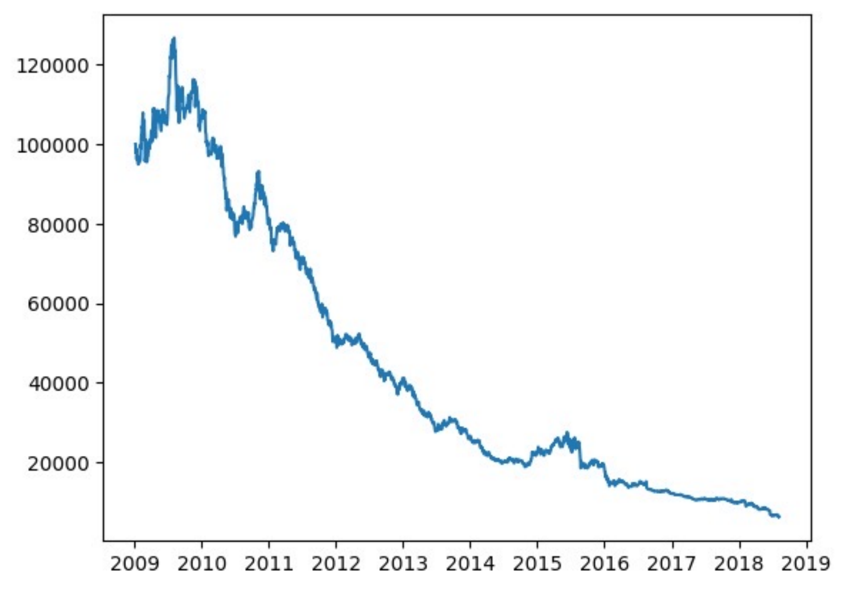


```python
def random_agent(env):
    return env.action_space.sample()
```

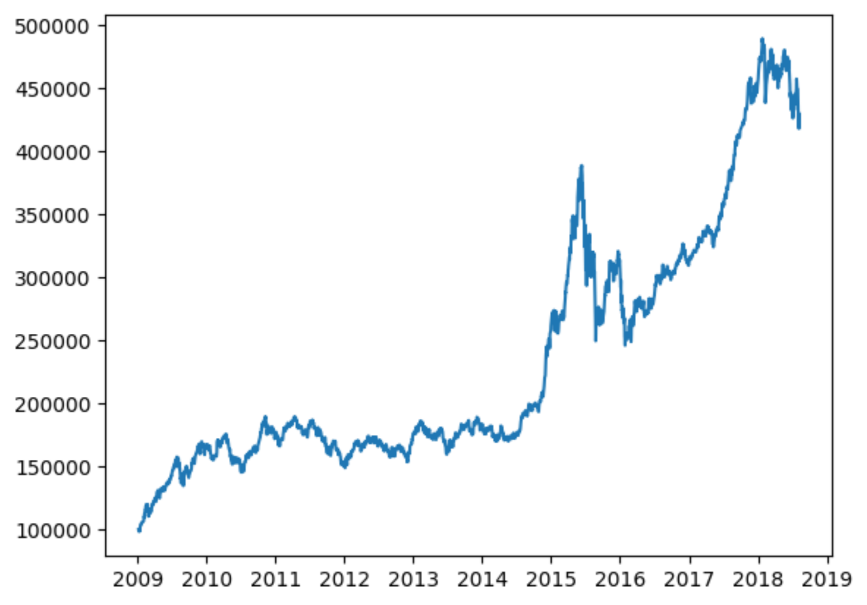


```python
def one_agent(env):
    return np.ones_like(env.action_space_shape)
```

- Fully random vs Evenly distributed



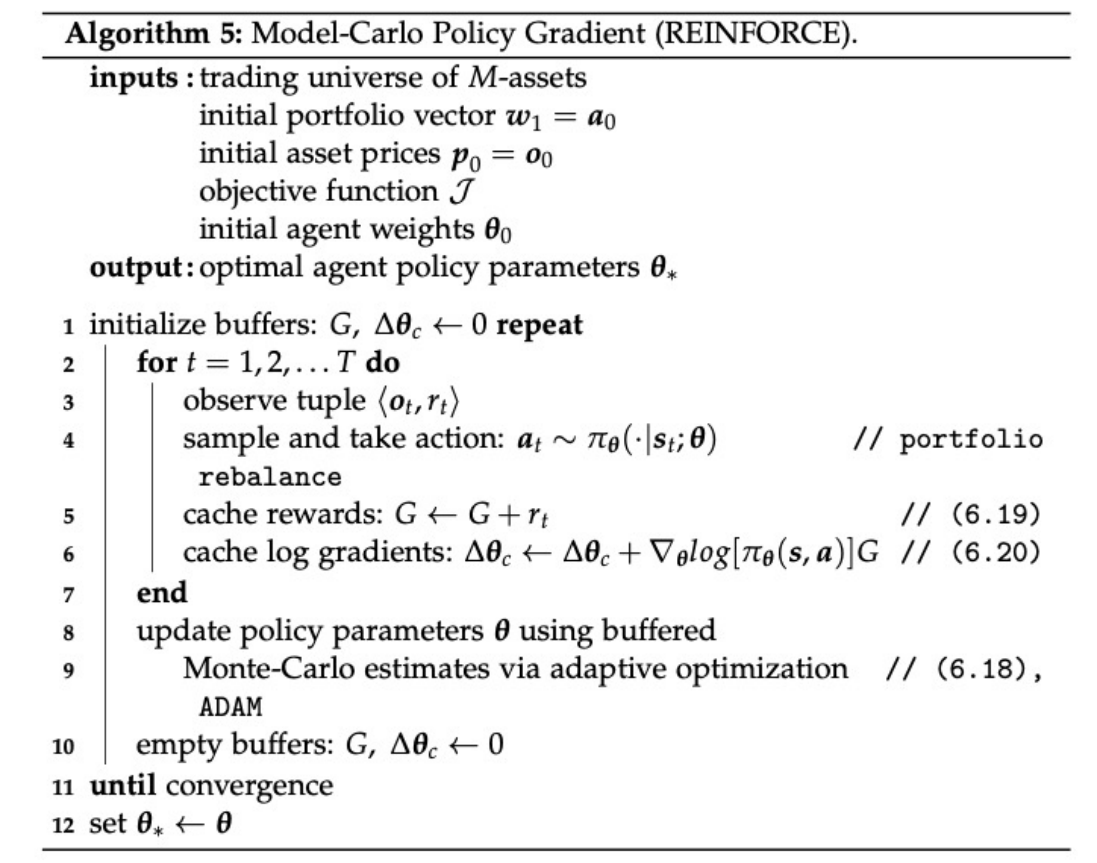
portfolio_value -93.676294%



portfolio_value +323.096709%

Policy Gradient: 策略梯度

- 策略网络结构
  - – input状态空间
  - – out行动空间
- 训练策略网络
  - – Step1. MC采样
  - – Step2. batch训练



Algorithm 5: Model-Carlo Policy Gradient (REINFORCE).

```
inputs : trading universe of $M$-assets
         initial portfolio vector $\boldsymbol{w}_1 = \boldsymbol{a}_0$
         initial asset prices $\boldsymbol{p}_0 = \boldsymbol{o}_0$
         objective function $\mathcal{J}$
         initial agent weights $\boldsymbol{\theta}_0$
output:  optimal agent policy parameters $\boldsymbol{\theta}_*$

initialize buffers: $G,\ \Delta\boldsymbol{\theta}_c \leftarrow 0$ repeat
    for $t = 1, 2, \dots T$ do
        observe tuple $\langle \boldsymbol{o}_t, r_t \rangle$
        sample and take action: $\boldsymbol{a}_t \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{s}_t;\boldsymbol{\theta})$        // portfolio
          rebalance
        cache rewards: $G \leftarrow G + r_t$                                 // (6.19)
        cache log gradients: $\Delta\boldsymbol{\theta}_c \leftarrow \Delta\boldsymbol{\theta}_c + \nabla_{\boldsymbol{\theta}} log[\pi_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})]G$  // (6.20)
    end
    update policy parameters $\boldsymbol{\theta}$ using buffered
        Monte-Carlo estimates via adaptive optimization   // (6.18),
          ADAM
    empty buffers: $G,\ \Delta\boldsymbol{\theta}_c \leftarrow 0$
until convergence
set $\boldsymbol{\theta}_* \leftarrow \boldsymbol{\theta}$
```

网络设计

- 简单全连接
- Lstm

简单全连接网络

Lstm

问题和未来方向

- 环境:
  - 数据集:
    - 考虑停牌股票
    - 考虑股票的基本面信息

- 智能体:其他的智能体结构
  - 网络结构：可扩展性（现在只能支持50支股票）

引用

- R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998
- Filos, A. (2019). Reinforcement Learning for Portfolio Management. ArXiv [q-Fin.PM]. Retrieved from http://arxiv.org/abs/1909.09571
- Sun, S., Qin, M., Wang, X., & An, B. (2023). PRUDEX-Compass: Towards Systematic Evaluation of Reinforcement Learning in Financial Markets. Transactions on Machine Learning Research. Retrieved from https://openreview.net/forum?id=JjbsIYOuNi

谢谢！