

# 策略参数嵌入下的扩展状态集构造与样本空间初始化机制

- 作者: GaoZheng
- 日期: 2025-03-19
- 版本: v1.0.0

## 一、问题定义与目标结构

给定:

- 状态集  $S = \{\sigma_1, \sigma_2, \dots\}$ : 描述账户、市场、策略等状态标签;
- 属性函数  $P: S \rightarrow \mathbb{R}^d$ : 原始状态属性映射;
- 策略系统包含额外**行为参数维度** (如: 下单间距、止损宽度、平仓频率、杠杆系数、风控阈值等);

目标:

- 将这些策略参数**内生嵌入到属性向量中**;
- 扩展状态集为:

$$S^* := \{(\sigma_i, \theta_j) \mid \sigma_i \in S, \theta_j \in \Theta\}$$

- 最终构造结构性样本路径:

$$\text{SamplePaths} := \{\gamma_k \in (S^*)^w\}$$

- 并由演化结果构造匹配的逻辑观察:

$$\text{ObservedValues} := \{y_k \in \mathbb{R} \mid k = 1, \dots, N\}$$

## 二、扩展属性映射: 策略参数嵌入 $P^*$

假设原始属性空间为:

$$P(\sigma_i) = (\text{净值}, \text{敞口}, \text{收益}, \text{波动}) \in \mathbb{R}^4$$

## 策略参数空间为：

$$\Theta := \{\theta = (\delta_{\text{entry}}, \delta_{\text{stop}}, f_{\text{close}}, \lambda)\} \subset \mathbb{R}^4$$

则扩展属性函数为：

$$P^*(\sigma_i, \theta) := (P(\sigma_i), \theta) \in \mathbb{R}^8$$

在实际代码中：

```
Pstar = Association[
  Flatten[
    Table[
      With[{base = P[state], strat = θ},
        (state <> "#" <> ToString[strat]) ->
          Join[base,
            <|"下单间距" -> strat[[1]], "止损" -> strat[[2]],
              "平仓频率" -> strat[[3]], "杠杆" -> strat[[4]]|>]],
        {state, Keys[P]}, {θ, strategyParameterGrid}]]];
```

## 三、构造扩展状态集 $S^*$

设定策略参数网格  $\theta_j \in \Theta$  为一组典型取值，如：

$$\Theta := \{(0.01, 0.01, 0.1, 1), (0.05, 0.02, 0.2, 2), (0.1, 0.05, 0.3, 5)\}$$

扩展状态集合定义为：

$$S^* := \left\{ \sigma_i^{(j)} := (\sigma_i, \theta_j) \mid \sigma_i \in S, \theta_j \in \Theta \right\}$$

即原有18个状态，×3组策略参数  $\rightarrow |S^*| = 54$

## 四、结合偏序时间切割构造 SamplePaths

回顾样本生成方案：

- 将时间序列（如交易记录、持仓状态、指标评估等）分段为  $T_1, \dots, T_{12}$ ;
- 每个时间段使用滑动窗口提取连续状态 + 策略参数，生成局部路径：

$$\gamma_k = \left( \sigma^{(j_1)}, \sigma^{(j_2)}, \dots, \sigma^{(j_w)} \right), \quad \gamma_k \in \Gamma_i$$

在代码中可表达为：

```
SamplePaths = Flatten[
  Table[
    SlidingWindowExtract[
      AnnotatedStateSequence[i], windowSize],
    {i, 1, 12}], 1];
```

其中 `AnnotatedStateSequence[i]` 是对每个周期中状态路径与策略参数联结后的组合状态序列。

## 五、生成 ObservedValues 的方法

每条路径  $\gamma_k$  会对应一个真实的观测值  $y_k$ ，可通过以下几种方式生成：

### 方法一：路径终态收益

$$y_k := \text{路径最后状态对应的收益值}$$

```
ObservedValues =
  Map[(Pstar[#][["收益"]] &@Last[#] &, SamplePaths];
```

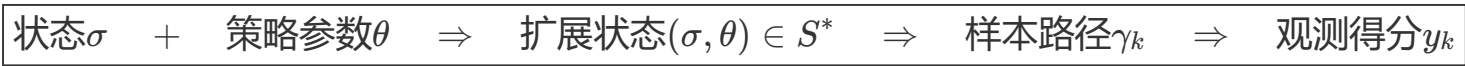
### 方法二：策略评分函数（如：风险调整收益）

$$y_k := \frac{\text{总收益}}{\text{最大回撤} + \epsilon}$$

或利用路径积分逻辑值与真实表现误差回归拟合：

```
ObservedValues =
  Map[ScoringFunction[#, marketReturnData] &, SamplePaths];
```

## 六、总结逻辑闭环



结合路径压强建模与DeriOptimize：

- 可拟合每季度  $w_i$ ，进行局部压强张量优化；
- 利用滑动窗口提取  $\Gamma_i$ ；

- 构建结构样本空间  $\mathcal{D} = \bigcup_i \Gamma_i$ ;
  - 最终支持基于O3路径积分模型的策略系统建模、结构AI训练、结构预测系统部署等。
- 

### 许可声明 (License)

Copyright (C) 2025 GaoZheng

本文档采用[知识共享-署名-非商业性使用-禁止演绎 4.0 国际许可协议 \(CC BY-NC-ND 4.0\)](#)进行许可。