

操作性状态约束下 Markov 假设的系统性失效

Kaifan XIE

2026.02.08

Abstract

本文在统一的操作性状态约束框架下，系统性分析 Markov 假设在五个层面的失效机制：接口层面、信息层面、方法论层面、预测性能层面以及责任接口层面。我们证明：在有限、可观测且无外部记忆的状态接口条件下，Markov 假设不仅不具备普适性，且其常见补救手段在原则上不可证伪、不可操作。所有证明均以展开形式给出，不存在可压缩为单一状态扩展命题的等价替代表述。

1 建模前提与操作性状态约束

定义 1 (操作性状态). 称状态 x_t 为操作性状态，若其满足以下约束：

- 状态信息量有限；
- 状态可由局部规则生成与更新；
- 状态不依赖完整历史即可构造；
- 状态不包含外部或隐藏记忆寄存器。

记所有满足上述条件的状态构成状态空间 \mathcal{X} 。本文中，任何违反上述约束的状态表示均被视为非操作性，且不参与建模讨论。

2 接口层面的 Markov 性

定义 2 (接口 Markov 性). 称过程 $\{x_t\}_{t \geq 0}$ 在接口层面是 Markov 的，若对任意 t ,

$$\Pr(x_{t+1} \mid x_t, x_{t-1}, \dots) = \Pr(x_{t+1} \mid x_t),$$

且上述条件概率完全在 \mathcal{X} 上定义。

定理 1 (接口非 Markov 性). 存在满足操作性状态约束的系统, 其接口状态过程 $\{x_t\}$ 不满足 Markov 性。

Proof. 由于操作性状态不包含完整历史, 必然存在两条不同历史 $h \neq h'$, 使得

$$x_t(h) = x_t(h').$$

若系统未来演化对历史敏感, 则存在事件 A , 使得

$$\Pr(x_{t+1} \in A \mid h) \neq \Pr(x_{t+1} \in A \mid h').$$

由此直接违反接口 Markov 定义。 \square

3 充分统计量的不存在性

定义 3 (充分统计量). 称状态 x_t 为充分统计量, 若其包含未来预测所需的全部信息:

$$\Pr(x_{t+1} \mid x_t, x_{t-1}, \dots) = \Pr(x_{t+1} \mid x_t).$$

定理 2 (操作性充分统计量不存在性). 在操作性状态约束下, 不存在任何 $x_t \in \mathcal{X}$ 能够作为系统未来演化的充分统计量。

Proof. 由操作性约束, x_t 无法区分所有可能历史。因此存在 $h \neq h'$, 使得 $x_t(h) = x_t(h')$ 。若未来演化对历史敏感, 则条件分布不同, 从而 x_t 无法承载全部预测信息。 \square

4 状态扩展的不可证伪性

定义 4 (扩状态命题). 称系统可通过扩状态恢复 Markov 性, 若存在扩展

$$\tilde{x}_t = (x_t, m_t),$$

使 $\{\tilde{x}_t\}$ 为 Markov 过程。

定理 3 (扩状态不可证伪性). 在操作性状态约束下, 扩状态 Markov 命题不可证伪, 因而不构成有效建模假设。

Proof. 允许任意 m_t 时, 可令 m_t 等于完整历史。该扩展不可观测、不可维护、不可验证。因此任何经验失败均可被不可操作变量吸收, 从而失去可证伪性。 \square

5 预测性能的结构性劣化

定义 5 (预测风险). 给定严格合适损失函数 L , 预测器 \hat{P} 的风险定义为

$$\mathcal{R}(\hat{P}) = \mathbb{E}[L(P(\cdot | h_t), \hat{P}(\cdot))].$$

定理 4 (Markov 预测劣化定理). 在操作性状态约束下, 任一 *Markov* 预测器相对于历史敏感预测器存在正的不可消除风险差距。

Proof. 由于不同历史被映射为同一 x_t , Markov 预测器必须输出相同预测, 从而在至少一条历史上产生误差。严格合适损失保证该误差有正下界。 \square

6 状态抽象与责任接口

定义 6 (责任可归属性). 称模型接口支持责任可归属性, 若不同历史成因在接口层面可被区分。

定理 5 (责任消除定理). 在操作性状态约束下, *Markov* 状态抽象在结构上削弱责任可归属性。

Proof. Markov 抽象要求历史在接口层面被压缩。当不同历史映射为同一状态时, 接口无法区分其成因, 责任在结构上不可归属。 \square

7 结论

本文证明: 在操作性状态接口条件下, Markov 假设在多个层面同时失效, 且不存在可操作、可证伪的统一补救策略。因此 Markov 性应被视为一种有明确代价与后果的建模选择, 而非默认成立的结构原理。