Research Notes

Tao Zhu

zhutaoer@outlook.com

创建于: 2025 年 12 月 24 日

更新于: 2025 年 12 月 24 日

目录

1 正文示例章节

这是一个简单的正文内容展示。你可以看到，普通文本使用的是默认的字号和对齐方式。你可以自由添加内容并调整格式。

1.1 子章节标题

这是一个子章节的内容展示。你可以在这里添加更多详细的文本内容，或者分段介绍不同的主题。

1.2 列表示例

这里是无序列表和有序列表的示例：

无序列表

- 项目 1
- 项目 2
- 项目 3

有序列表

1. 第一项
2. 第二项
3. 第三项

1.3 英文字体展示

The font I chose is Palatino, a classic serif font designed by German type designer Hermann Zapf in 1948. Compared to *Times New Roman*, Palatino has softer stroke contrasts and a more open character layout, making it less compact and serious, resulting in a more comfortable reading experience.

1.4 三线表示例

这是一个简单的三线表展示：

表 1: Notations

Symbol	Description
$\mathcal{S}$	State space of the embodied agent
$\mathcal{A}$	Discrete action space
$T(s' \mid s, a)$	State transition function
$R(s, a)$	Reward function (negative distance to goal)
γ	Discount factor, $\gamma \in (0, 1)$
$V(s)$	Value function of state s
$\pi(s)$	Policy mapping states to actions
V^*	Optimal value function
π^*	Optimal policy
θ	Convergence threshold for value iteration
Δ	Maximum change in value function during iteration
s_{goal}	Target position in the environment
s_{pos}	Current position of the agent

*There are some variables that are not listed here and will be discussed in detail in each section.

1.5 伪代码示例

Algorithm 1: Value Iteration Algorithm for Embodied Navigation

Input: EMDP $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$, convergence threshold θ

Output: Optimal value function V^* and policy π^*

Initialize $V(s) \leftarrow 0$ for all $s \in \mathcal{S}$

repeat

  $\Delta \leftarrow 0$

  for *each* $s \in \mathcal{S}$ **do**

    $v \leftarrow V(s)$

    $V(s) \leftarrow \max_{a \in \mathcal{A}} \left[R(s,a) + \gamma \sum_{s'} T(s' \mid s,a) V(s') \right]$

    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$

  end

  /* end for each state */

until $\Delta < \theta$

/* end repeat */

for *each* $s \in \mathcal{S}$ **do**

  $\pi^*(s) \leftarrow \arg\max_{a \in \mathcal{A}} \left[R(s,a) + \gamma \sum_{s'} T(s' \mid s,a) V^*(s') \right]$

end

/* end for policy extraction */

return V^*, π^*

2 代码示例

这是一个代码块的展示，支持语法高亮：

Listing 1: Python 代码示例

```python
def hello_world():
    print("Hello, World!")
```

3 插图示例

图 1: 示例图片

4 彩色信息框示例

定义 4.1: 具身马尔可夫决策过程（EMDP）

一个具身马尔可夫决策过程定义为五元组 $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$，其中：

- $\mathcal{S} = \mathbb{R}^2 \times [0, 2\pi)$ 为状态空间（位置 + 朝向）；
- $\mathcal{A} = \{\text{前进}, \text{左转}, \text{右转}\}$ 为离散动作空间；
- $T(s' \mid s, a)$ 为状态转移函数；
- $R(s, a) = -\|s_{\text{pos}} - s_{\text{goal}}\|$ 为稀疏奖励（负欧氏距离）；
- $\gamma \in (0, 1)$ 为折扣因子。

假设 4.1: 可观测性

智能体在每一步都能精确观测当前状态 $s_t \in \mathcal{S}$。

假设 4.2: 确定性动力学

状态转移是确定性的，即 $s_{t+1} = f(s_t, a_t)$，无环境噪声。

该算法的正确性由以下定理保证。

引理 4.1: 贝尔曼最优性

值函数 V^* 是贝尔曼最优算子 $\mathcal{T}$ 的唯一不动点，即 $V^* = \mathcal{T} V^*$，其中

$$\mathcal{T} V(s) := \max_{a \in \mathcal{A}} \left[R(s, a) + \gamma \sum_{s'} T(s' \mid s, a) V(s') \right].$$

定理 4.1: 值迭代收敛性

值迭代算法生成的序列 $\{V_k\}$ 以指数速率收敛到 V^*，即

$$\|V_k - V^*\|_\infty \leq \frac{2\gamma^k}{1 - \gamma} \|V_1 - V_0\|_\infty.$$

证明. 由于状态空间 $\mathcal{S}$ 在假设下为有限（或可离散化），且 $\gamma < 1$，贝尔曼算子 $\mathcal{T}$ 是 γ-压缩映射。由巴拿赫不动点定理，迭代 $V_{k+1} = \mathcal{T} V_k$ 收敛到唯一不动点 V^*，且误差界如上所述。 □

例子 4.1: 2D 网格世界导航

考虑一个 5×5 的网格世界，机器人从左下角 $(0,0)$ 出发，目标为右上角 $(4,4)$。状态 $s = (x, y, \theta)$，其中 $\theta \in \{0, \pi/2, \pi, 3\pi/2\}$。动作 $\mathcal{A} = \{$前进, 左转, 右转$\}$ 控制朝向与位置。在 **??** 下，值迭代可精确计算到达目标的最短路径。