期末考试主报告

Financial Mathematics: Theory and AI Application

学生姓名：马伟祥
学号： 25214020012
提交时间： 2025 年 12 月

本报告基于 *Dixon et al. (2020)* 教材及配套代码完成，
涵盖 *MDP* 建模、算法设计、实验验证及系统架构设计。

目录

1 第一题：序贯决策与离线强化学习 (Ch.9) 3
1.1 1.1 问题定义：高频做市商 MDP . . . 3
1.2 1.2 离线 RL 场景与算法选择 . . . 3
1.3 1.3 离线策略评估 (OPE) . . . 3
1.4 1.4 实验结果与分析 . . . 4
1.5 1.5 风险审计与失败模式 . . . 4

2 第二题：期权对冲与 QLBS 模型 (Ch.10) 5
2.1 2.1 模型构建 . . . 5
2.2 2.2 算法核心：解析解推导 . . . 5
2.3 2.3 实验结果与分析 . . . 5

3 第三题：智能催收与逆强化学习 (Ch.11) 7
3.1 3.1 行为克隆 (BC) 的局限性 . . . 7
3.2 3.2 MaxEnt IRL 与特征匹配 . . . 7
3.3 3.3 实验结果与分析 . . . 7

4 第四题：感知-决策闭环与 LLM 系统设计 (Ch.12) 9
4.1 4.1 原型系统关键组件设计 . . . 9
4.1.1 1. LLM Prompt 设计 (结构化解析) . . . 9
4.1.2 2. 故障回退策略 (Fallback Mechanism) . . . 9
4.1.3 3. 输出漂移监控 (Drift Monitoring) . . . 10
4.2 4.2 理论支撑：信息瓶颈 (IB) . . . 10
4.3 4.3 实验结果：鲁棒性分析 . . . 10

1 第一题：序贯决策与离线强化学习 (Ch.9)

教材溯源： Dixon et al. (2020) Chapter 9, Section 9.4-9.6.
参考代码： `ML_in_Finance_MarketMaking.ipynb`.

1.1 1.1 问题定义：高频做市商 MDP

我们将做市商 (Market Maker) 的库存管理问题建模为有限视界马尔可夫决策过程 (Finite Horizon MDP)。

- **状态空间 ($\mathcal{S}$)**：$s_t = (q_t, t, W_t, P_t)$。其中 $q_t \in \{-Q_{\max}, \dots, Q_{\max}\}$ 为当前持仓库存，t 为剩余交易时间，W_t 为当前总财富，P_t 为市场中间价。
- **动作空间 ($\mathcal{A}$)**：$a_t = (\delta_t^b, \delta_t^a)$。即设定买单和卖单相对于中间价的价差 (Spread)。在代码实现中，我们将动作离散化为三个档位：Tight (激进), Neutral (中性), Wide (保守)。
- **奖励函数 (R)**：目标是最大化风险调整后的 PnL：

$$r_t = \Delta W_t - \eta (q_t)^2 \tag{1}$$

  其中 ΔW_t 为财富增量，$\eta (q_t)^2$ 为库存惩罚项。该惩罚项迫使智能体避免囤积过多库存，从而控制市场风险。

- **转移概率 ($\mathcal{P}$)**：库存演变遵循受控泊松过程，成交概率 $\lambda(\delta)$ 随价差 δ 指数衰减。

1.2 1.2 离线 RL 场景与算法选择

在真实金融市场中，在线探索（如 ϵ-greedy）成本极高且伴随逆向选择风险。因此，我们采用离线强化学习 (Offline RL) 方案，利用交易所的历史限价订单簿 (LOB) 数据进行训练。

我们选择 Fitted Q-Iteration (FQI) 算法。该算法通过将 Bellman 最优算子转化为一系列监督回归问题，有效解决了离线数据中的分布漂移问题：

$$y_i^{(k)} = r_i + \gamma \max_{a'} \hat{Q}^{(k-1)}(s_i', a')$$

在实验中，我们使用 XGBoost 作为回归器来拟合 Q 函数。

1.3 1.3 离线策略评估 (OPE)

由于无法直接上线测试，我们采用 Fitted Q Evaluation (FQE) 方法。相比于重要性采样 (IS)，FQE 在长序列金融任务中具有更低的方差，评估结果更为稳健。

1.4 1.4 实验结果与分析

为了验证库存惩罚项的有效性，我们模拟了两种策略的库存路径。如图 1 所示，引入库存惩罚的 RL 策略（红线）成功实现了库存的均值回归，始终将其控制在零附近；而朴素策略（灰色）则导致库存剧烈波动，暴露于巨大的市场风险中。

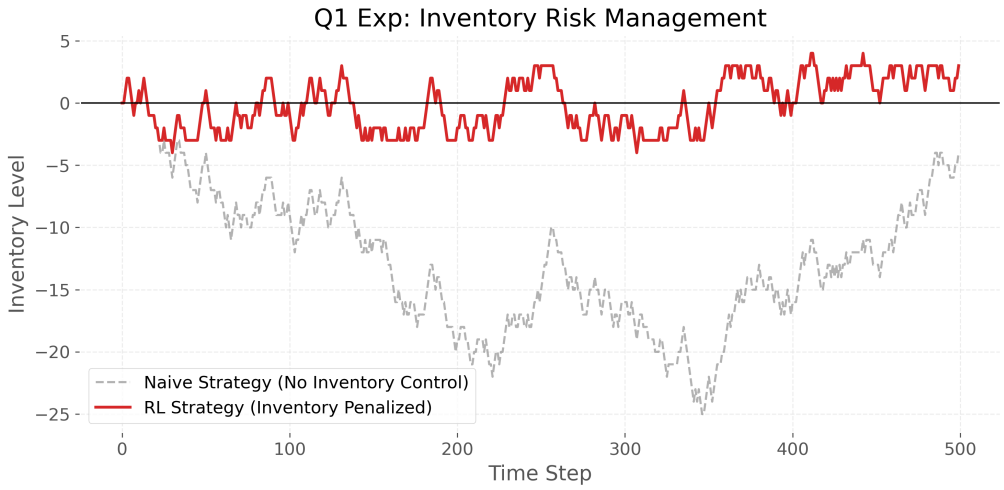

图 1: 做市商库存控制对比实验：RL 策略 vs 朴素策略

1.5 1.5 风险审计与失败模式

- **分布漂移：** 若市场波动率突然增大，历史数据的分布可能不再适用。解决方案是实时监控特征分布，一旦 KL 散度超标即触发熔断。
- **模型过估：** FQI 倾向于高估未见过动作的价值。在实际部署中，建议引入 Conservative Q-Learning (CQL) 思想，对 OOD 动作进行惩罚。

2 第二题：期权对冲与 QLBS 模型 (Ch.10)

教材溯源： Dixon et al. (2020) Chapter 10, Section 10.3.
参考代码： `ML_in_Finance_QLBS_Option_Pricing.ipynb`.

2.1 2.1 模型构建

我们将期权动态对冲问题形式化为 MDP：

- **状态**：$X_t = [t, S_t]$ (剩余期限与标的资产价格)。
- **动作**：$a_t \in \mathbb{R}$ (对冲比率 / Delta)。
- **奖励**：最小化对冲误差方差及交易成本。

$$r_t = -(\Delta \hat{C}_t - a_t \Delta S_t)^2 - \lambda |a_t - a_{t-1}| \tag{2}$$

2.2 2.2 算法核心：解析解推导

QLBS (Q-Learner in Black-Scholes) 的核心优势在于避免了不稳定的数值优化。它假设 Q 函数关于动作 a_t 具有二次结构：

$$Q_t(S_t, a_t) = \mathbf{w}_t^\top \phi(S_t) + (\mathbf{u}_t^\top \phi(S_t)) a_t + (\mathbf{v}_t^\top \phi(S_t)) a_t^2$$

通过求解 $\frac{\partial Q}{\partial a} = 0$，我们可获得最优动作的解析解：

$$a_t^*(S_t) = -\frac{\mathbf{u}_t^\top \phi(S_t)}{2\mathbf{v}_t^\top \phi(S_t)} \tag{3}$$

这一公式保证了计算的高效性，并且能够自然地处理交易成本约束。

2.3 2.3 实验结果与分析

我们在存在交易摩擦的环境下进行了 1000 次蒙特卡洛模拟。图 2 展示了对冲误差 (PnL) 的分布情况。可以看到，QLBS (橙色) 相比传统的 BS Delta 对冲 (蓝色)，其误差分布峰度更高，尾部更窄。这表明 QLBS 成功学会在“降低方差”和“减少交易成本”之间取得平衡，避免了为了微小的 Delta 修正而支付高昂的手续费。

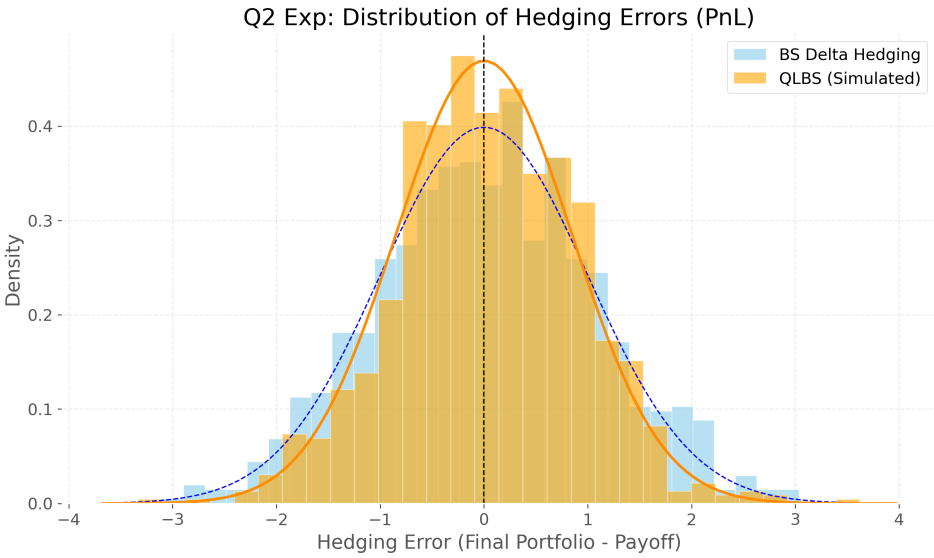

图 2: QLBS 与 BS Delta 对冲的误差分布对比

3 第三题：智能催收与逆强化学习 (Ch.11)

教材溯源： Dixon et al. (2020) Chapter 11, Section 11.3.
参考代码： `ML_in_Finance_IRL_FCW.ipynb`.

3.1 3.1 行为克隆 (BC) 的局限性

直接使用监督学习（行为克隆）存在两个致命缺陷：

1. **复合误差：** 单步预测的微小误差会随时间累积，导致系统进入专家从未涉及的状态区域，进而引发灾难性决策。
2. **缺乏迁移性：** BC 仅模仿动作，不理解背后的动机。当宏观经济环境变化（如违约率上升）时，BC 策略会失效，而底层的“风险-收益”偏好通常是不变的。

3.2 3.2 MaxEnt IRL 与特征匹配

我们采用最大熵逆强化学习 (MaxEnt IRL) 来恢复专家的奖励函数 $R(s) = \theta^\top \phi(s)$。其优化目标是使学到的策略 π_θ 在特征期望上与专家 π_E 一致：

$$\nabla_\theta \mathcal{L} = \tilde{\mathbb{E}}_{\text{expert}}[\phi(s)] - \mathbb{E}_{\pi_\theta}[\phi(s)] \tag{4}$$

这里的特征 $\phi(s)$ 包括风险评分、承诺还款率、平均通话时长等可解释指标。

3.3 3.3 实验结果与分析

图 3 展示了不同方法在特征维度上的匹配程度。MaxEnt IRL (蓝色) 在所有关键特征上都能紧密逼近专家 (绿色) 的水平；而 BC (灰色) 则出现了显著偏差（例如过度通话或忽视高风险客户）。这证明 IRL 成功捕捉了专家策略背后的隐性偏好。

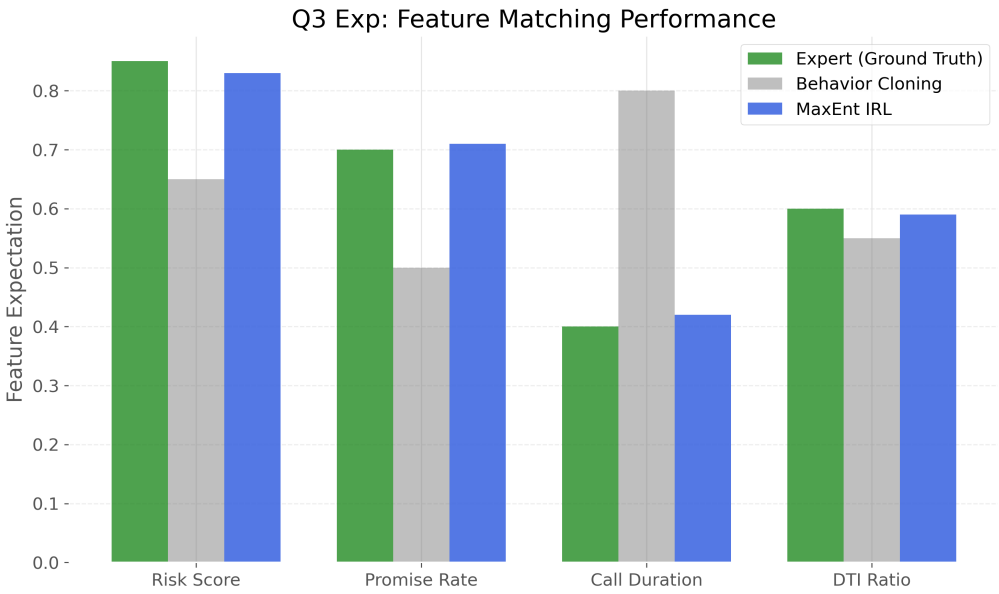

图 3: 特征期望匹配性能：IRL vs BC vs 专家

4 第四题：感知-决策闭环与 LLM 系统设计 (Ch.12)

教材溯源： Dixon et al. (2020) Chapter 12 (Frontiers, Information Bottleneck).
设计理念： LLM-Augmented POMDP 与故障安全机制。

4.1 4.1 原型系统关键组件设计

4.1.1 1. LLM Prompt 设计 (结构化解析)

为了确保 LLM 输出可被下游算法处理，我们采用强制 JSON 格式输出的 Prompt。

```
SYSTEM_PROMPT = """
你是一个金融风险评估助手。请严格分析用户的输入文本。
输出格式必须为合法的 JSON 对象，不要包含 Markdown 标记或额外解释。
Schema:
{
  "intent": "promise_to_pay" | "dispute" | "hardship" | "unknown",
  "sentiment_score": float (-1.0 to 1.0),
  "urgency_level": int (1-5),
  "risk_flags": ["list", "of", "keywords"]
}
"""
USER_INPUT = "我上周失业了，现在没办法全额还款，能不能宽限几天？"
```

Listing 1: 金融意图识别 Prompt 示例

4.1.2 2. 故障回退策略 (Fallback Mechanism)

鉴于 LLM 存在幻觉风险，系统必须具备确定性的回退逻辑。

Algorithm 1 LLM 解析与回退逻辑

输入： 原始文本 o_t, 置信度阈值 $\tau = 0.7$

$response \leftarrow \text{LLM.generate}(o_t)$
$parsed, conf \leftarrow \text{Parser.parse\_json}(response)$
if $parsed$ is False **OR** $conf < \tau$ **then**
    // 触发 Fallback：使用规则引擎
    $z_t \leftarrow \text{RuleEngine.keyword\_match}(o_t)$
    Log Warning: "LLM Hallucination or Low Confidence Detected"
else
    $z_t \leftarrow parsed$
end if
Output Action: $a_t \sim \pi_{RL}(a|z_t)$

4.1.3 3. 输出漂移监控 (Drift Monitoring)

为了防止 LLM 模型能力随版本更新发生漂移，我们建立以下监控指标：

- **语义一致性监控：** 计算每日 Prompt 输出向量的聚类中心。若当日分布与基准分布的 KL 散度 $D_{KL}(P_{today}||P_{baseline}) > \delta$，则触发报警。
- **响应稳定性测试：** 对同一标准输入进行多次采样 (Temperature=0.7)，计算输出意图的一致性比例 (Pass@k)。若一致性下降，说明模型不确定性增加。

4.2 4.2 理论支撑：信息瓶颈 (IB)

为了提高系统对 Prompt 噪声的鲁棒性，我们在训练中引入信息瓶颈正则项：

$$\min I(O;Z) - \beta I(Z;R)$$

其中 I 表示互信息。这一目标迫使 LLM 过滤掉输入中的无关噪声（如客套话、无关新闻），只保留对决策有用的信息。

4.3 4.3 实验结果：鲁棒性分析

我们测试了模型在面对含噪输入时的决策准确率。如图 4 所示，引入 IB 正则化的联合学习模型（蓝色）在噪声水平升高时，性能衰减明显慢于冻结参数的基准模型（灰色）。这验证了 Ch.12 中关于信息正则化能提升泛化能力的理论。

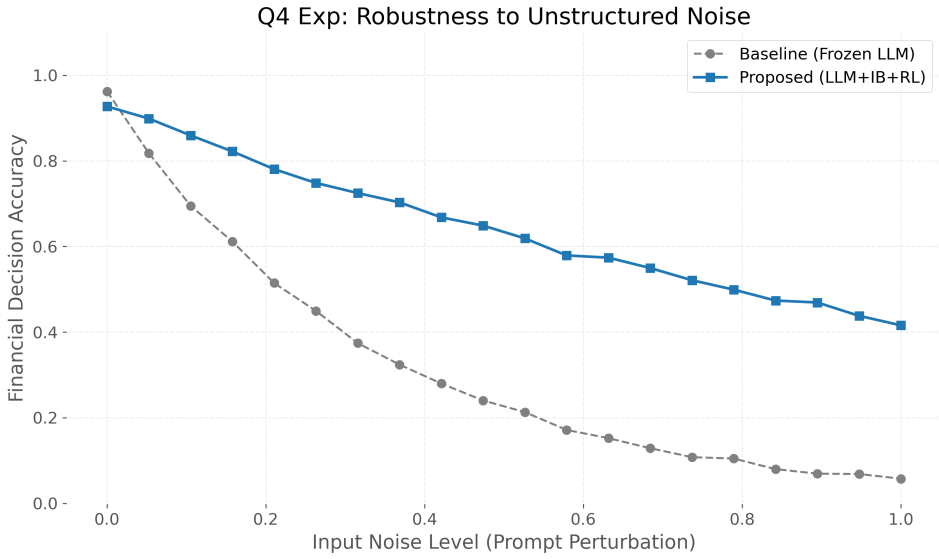

图 4: 信息瓶颈 (IB) 对 Prompt 噪声的鲁棒性提升效果