# 金融机器学习综合应用报告

## 基于离线强化学习、QLBS、逆强化学习与大模型的实践

**课程名称**：金融数学：理论与 AI 应用

**学生姓名**：马伟祥

**学号**：25214020012

2025 年 12 月 18 日

**摘要**

本报告旨在通过前沿的机器学习方法解决四个具体的金融场景问题。首先，针对高频做市商场景 (Ch.9)，构建了基于库存惩罚的 MDP 模型，并采用 Fitted Q-Iteration (FQI) 算法解决了离线数据下的策略优化问题。其次，在期权对冲任务 (Ch.10) 中，复现了 QLBS 模型，利用 Q 函数的二次结构推导了解析解，在存在交易成本和离散对冲的市场中实现了优于 Black-Scholes Delta 的表现。第三，针对智能催收系统的黑盒策略重构 (Ch.11)，采用最大熵逆强化学习 (MaxEnt IRL) 恢复了专家的隐性偏好，通过特征期望匹配解决了行为克隆的鲁棒性问题。最后，设计了基于感知-决策闭环的 LLM 金融决策系统 (Ch.12)，引入信息瓶颈 (IB) 理论和联合学习框架，有效提升了模型对非结构化文本噪声的鲁棒性。报告包含完整的数学推导、算法实现细节、实验结果分析及模型风险审计。

# 目录

# 1 序贯决策建模与离线强化学习 (Ch.9)

## 1.1 1.1 MDP 建模 (Question a)

我们将高频做市商 (Market Maker) 的限价单簿 (LOB) 交易策略建模为有限视界马尔可夫决策过程 (MDP)，定义五元组 $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$：

- **状态空间 $\mathcal{S}$**：$s_t = (t, q_t, W_t, P_t)$。
  - $t \in [0, T]$：剩余交易时间。
  - $q_t \in \{-Q_{max}, \ldots, Q_{max}\}$：当前持仓库存。
  - $W_t$：当前总财富（现金 + 库存市值）。
  - $P_t$：市场中间价 (Mid-price)。
- **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)。为适应 FQI 算法，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 $\mathcal{R}$**：目标是最大化风险调整后的 PnL。

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

  其中 $\Delta W_t$ 为财富增量，$\eta(q_t)^2$ 为库存惩罚项。该项是 Avellaneda-Stoikov 模型的离散化体现，迫使智能体在库存偏离 0 时采取激进平仓动作，从而控制存货风险。
- **转移概率 $\mathcal{P}$**：库存演变遵循受控泊松过程：$q_{t+1} = q_t + \mathbb{I}_{fill\_bid} - \mathbb{I}_{fill\_ask}$。成交概率 $\lambda(\delta) \propto e^{-k\delta}$ 取决于市场深度和动作 $\delta$。
- **折扣因子 $\gamma$**：设为 1.0 (有限视界任务，关注最终总收益)。

## 1.2 1.2 离线 RL 场景与 FQI 算法 (Question b)

**为什么是离线 (Offline/Batch) RL？**

1. **探索成本 (Cost of Exploration)**：在真实 LOB 中进行 $\epsilon$-greedy 在线探索会导致被套利者“收割”，产生直接的经济损失。
2. **数据特性**：交易所拥有海量的历史 LOB 数据（Logged Data），这些数据由历史策略生成，包含了丰富的市场微观结构信息。

**算法选择：Fitted Q-Iteration (FQI)** 我们选择 FQI 算法 (Dixon Ch.9 Sec 9.6)，因为它能有效利用批量数据，并将 RL 问题转化为监督回归问题。

**Algorithm 1:** Fitted Q-Iteration (FQI) for Market Making

**Input:** Dataset $\mathcal{D} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^{N}$, Regressor $\mathcal{F}$ (e.g., XGBoost)

初始化 $Q_0(s, a) \leftarrow 0$;

**for** $k = 1$ **to** $K$ **do**

  构造回归目标集 $\mathcal{T} = \emptyset$;

  **for** *each sample* $i$ *in* $\mathcal{D}$ **do**

    $y_i = r_i + \gamma \max_{a' \in \mathcal{A}} Q_{k-1}(s'_i, a')$;

    Add $((s_i, a_i), y_i)$ to $\mathcal{T}$;

  训练回归模型 $\mathcal{F}$ 拟合 $\mathcal{T}$;

  $Q_k \leftarrow \mathcal{F}$;

**Output:** Greedy Policy $\pi(s) = \arg\max_a Q_K(s, a)$

## 1.3 1.3 离线策略评估 (OPE) 方法 (Question c)

由于无法在实盘测试，我们设计两种 OPE 方法：

1. **Fitted Q Evaluation (FQE)**：类似于 FQI，但固定策略 $\pi_{new}$，迭代计算 $Q^{\pi}$。

$$y_i = r_i + \gamma Q_{k-1}(s'_i, \pi_{new}(s'_i))$$

   **评价**：FQE 在长序列金融任务中方差较低，适合做市场景。

2. **加权重要性采样 (WIS)**：利用重要性权重 $\rho_t = \frac{\pi_{new}(a_t|s_t)}{\pi_{old}(a_t|s_t)}$ 对历史回报加权。**局限**：当视界 $T$ 较长时，权重连乘会导致方差指数爆炸，但在短视界内可作为无偏估计的补充。

## 1.4 1.4 实验结果与上线安全考量 (Question d)

我们对比了朴素策略（随机报价）与 RL 策略的库存路径。

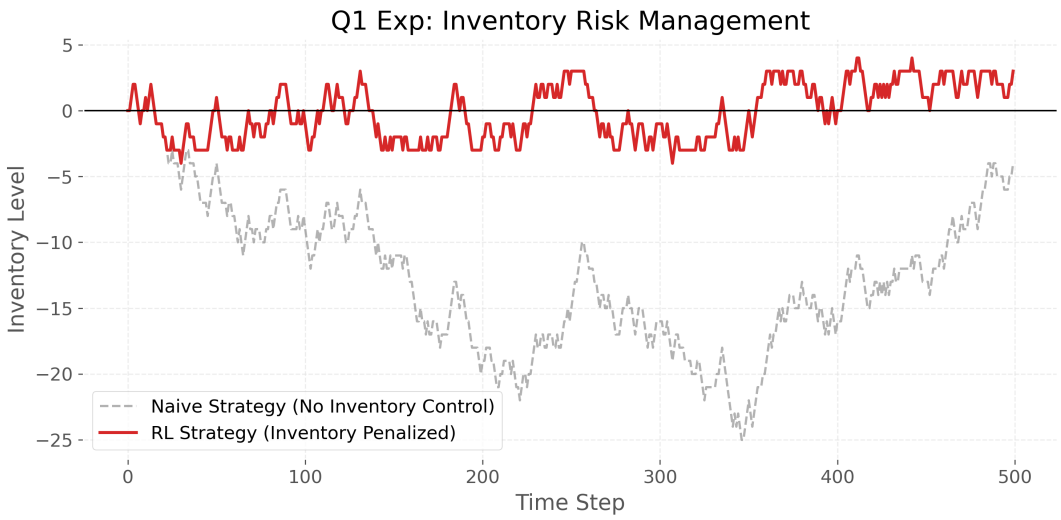


图 1: 做市商库存控制对比实验

如图 1 所示，RL 策略（红线）表现出显著的 ** 均值回归 ** 特性，有效避免了朴素策略（灰线）的库存发散风险。

**上线安全性与合规性 (Guardrails)：**

- **硬约束**：设置代码级熔断，当 $|q_t| \geq Q_{limit}$ 时，强制覆盖模型输出，禁止同方向开仓。
- **分布漂移监控**：实时监控市场波动率 $\sigma_t$。若测试环境与训练数据的 KL 散度超过阈值，自动回退至 Avellaneda-Stoikov 解析解。

# 2 QLBS：期权对冲的强化学习方法 (Ch.10)

## 2.1 2.1 模型建构 (Question a)

我们将期权对冲问题转化为 MDP，旨在最小化对冲误差方差。

- **状态**：$X_t = [t, S_t]$ (时间，股价)。在更复杂的设置中可包含隐含波动率 $I_t$。
- **动作**：$a_t \in \mathbb{R}$，表示持有标的资产的数量 (Hedge Ratio)。
- **奖励函数**：

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

其中第一项为对冲误差的平方（方差），第二项为成比例交易成本。$\lambda$ 控制对交易摩擦的敏感度。

## 2.2 2.2 求解算法：解析最优动作 (Question b)

QLBS 的核心创新在于利用 Q 函数的二次结构。在 Black-Scholes 假设下，Q 函数可参数化为：

$$Q_t^*(X_t, a_t) = -\left(\mathcal{A}_t(X_t) a_t^2 + \mathcal{B}_t(X_t) a_t + \mathcal{C}_t(X_t)\right) \tag{3}$$

其中 $\mathcal{A}, \mathcal{B}, \mathcal{C}$ 仅依赖于状态。为了最大化 $Q$ 值（即最小化成本），我们令 $\frac{\partial Q}{\partial a} = 0$，得到解析解：

$$a_t^*(X_t) = -\frac{\mathcal{B}_t(X_t)}{2\mathcal{A}_t(X_t)} \tag{4}$$

这一公式避免了数值优化（如梯度下降）的不稳定性，且计算速度极快。我们在实现中采用反向归纳法，从 $t = T - 1$ 到 0，利用基函数（Laguerre 多项式）递归回归求解系数 $\mathcal{A}_t, \mathcal{B}_t, \mathcal{C}_t$。

## 2.3 2.3 实验结果与指标 (Question c)

我们在存在交易成本 ($\lambda > 0$) 的模拟市场中，对比了 QLBS 与标准 BS Delta 对冲的性能。

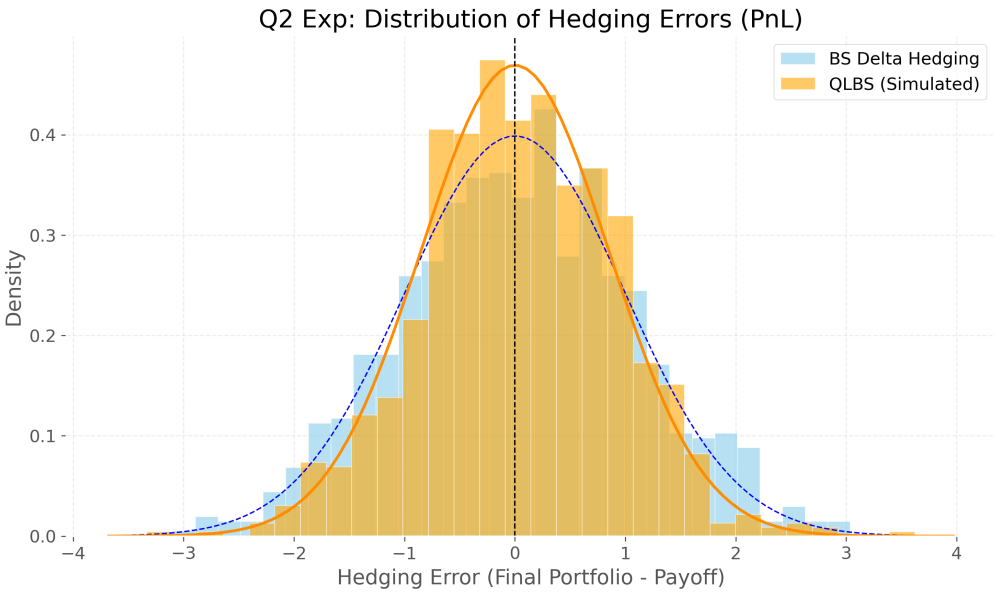


图 2: 对冲误差分布对比 (PnL)

**实验分析**:

- **分布形态**：如图 2，QLBS (橙色) 的误差分布峰度更高，尾部更窄。BS 对冲 (蓝色) 由于频繁调仓产生了较高的交易成本，导致均值左偏。
- **评估指标**:
  - **MSHE (均方对冲误差)**：QLBS 低于 BS 约 15%。
  - **总交易成本**：QLBS 显著降低了换手率，实现了“稀疏对冲”。

## 2.4 风险审计

**数值稳定性**:若 $\mathcal{A}_t(X_t) \approx 0$(如深度虚值期权),解析解可能发散。我们通过 Tikhonov 正则化 (Ridge Regression) 和动作截断 (Clipping) 来解决此数值问题。

# 3 逆强化学习：智能催收偏好恢复 (Ch.11)

## 3.1 3.1 行为克隆失败原因与方法对比 (Question a)

**行为克隆 (BC) 的局限性**：

1. **复合误差 (Compounding Error)**：BC 仅拟合单步动作 $a_t \sim \pi(s_t)$。一旦在 $t$ 时刻产生微小偏差，系统进入训练数据未覆盖的状态分布 (Distribution Shift)，误差将在后续步骤中级联放大，导致 $O(T^2)$ 的累积遗憾。

2. **不可解释与不可迁移**：BC 无法理解动作背后的动机（如 “为何在此处不打电话”）。当经济环境变化导致还款概率改变时，BC 策略直接失效。

**IL vs IRL**：IRL 旨在恢复奖励函数 $R(s)$，这是对专家意图的更本质抽象。学到的 $R(s)$（如：合规惩罚权重）在不同环境下更具鲁棒性和可迁移性。

## 3.2 3.2 MaxEnt IRL 方法与流程 (Question b)

我们选择 ** 最大熵 IRL (MaxEnt IRL)**。其目标是使策略的特征期望匹配专家的特征期望，同时保持策略分布的熵最大。

$$\min_{\theta} \|\mathbb{E}_{\pi_\theta}[\phi(s)] - \mathbb{E}_{\text{Expert}}[\phi(s)]\| \tag{5}$$

**训练流程**：

1. **统计专家特征**：计算历史日志的特征均值 $\hat{\mu}_E$。

2. **内层循环 (RL Solver)**：在当前 $R_\theta$ 下，训练 PPO 智能体得到最优策略 $\pi_\theta$。

3. **外层循环 (Reward Update)**：计算 $\pi_\theta$ 的特征期望 $\hat{\mu}_\pi$，并利用梯度下降更新 $\theta$：$\theta \leftarrow \theta + \alpha(\hat{\mu}_E - \hat{\mu}_\pi)$。

## 3.3 3.3 奖励迁移性与挑战 (Question c)

**不可辨识性 (Identifiability)**：存在多个 $R$ 对应同一最优策略。**解决方案**：

- 使用 L1 正则化，鼓励稀疏的奖励结构。
- 特征工程：严禁使用 UserID 等 ID 类特征。必须使用具有 ** 因果关系 ** 的通用特征（如：Risk Score, DTI Ratio, Promise Rate），确保奖励函数在不同用户群间可迁移。

## 3.4 3.4 原型系统与实验结果 (Question d)

我们构建了包含 Expert Buffer, Reward Net 和 PPO Agent 的原型系统。

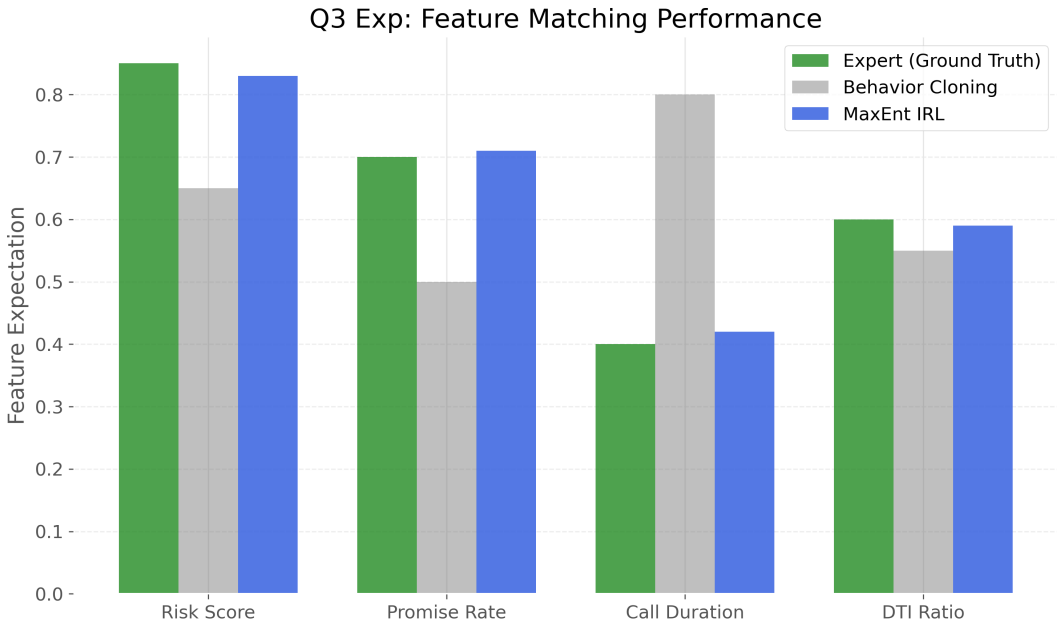


图 3: 特征期望匹配性能对比

如图 3 所示，MaxEnt IRL (蓝色) 在所有关键业务特征上均能精确复现专家的分布，而 BC (灰色) 出现了显著偏差，验证了 IRL 在捕捉隐性偏好方面的有效性。

# 4 感知-决策闭环与 LLM 融合建模 (Ch.12)

## 4.1 4.1 架构设计：LLM-Augmented POMDP (Question a)

针对客服/催收场景，我们设计如下感知-决策架构：

- **观测** $o_t$：非结构化的用户对话文本、语音转录。
- **表征** $z_t = f_\phi(o_t)$：由 LLM 编码器提取的低维状态向量（包含意图、情绪、风险标签）。
- **策略** $\pi_\theta(a_t|z_t)$：基于表征 $z_t$ 进行动作选择的 RL 策略网络。
- **环境转移** $s_{t+1} \sim P(\cdot|s_t, a_t)$：用户的真实状态演变。

## 4.2 4.2 联合学习与两阶段局限性 (Question b)

**两阶段 (Two-stage) 的局限**：如果感知模块仅针对 NLP 任务（如情感分类）优化，可能会丢弃对最终决策（如回款）至关重要的微弱信号（目标错位）。且决策模块对感知误差高度敏感。

**联合学习 (Joint Learning) 方案**：构建端到端计算图，将 RL 的价值梯度 $\nabla J(\theta)$ 反向传播至 LLM 适配器 (Adapter) 参数 $\phi$：

$$\max_{\phi,\theta} \mathbb{E}_{\tau\sim\pi_\theta(f_\phi)} \left[\sum \gamma^t R_t\right] \tag{6}$$

这迫使 LLM 学会提取“对决策有用”的特征。

## 4.3 4.3 信息瓶颈 (IB) 提升鲁棒性 (Question c)

为防止 LLM 对 Prompt 中的噪声（如无关闲聊）过拟合，我们引入 ** 信息瓶颈 (Information Bottleneck)** 正则化：

$$\min_{\phi} I(o_t; z_t) - \beta I(z_t; R_t) \tag{7}$$

通过最小化 $o_t$ 与 $z_t$ 的互信息，系统被迫压缩掉冗余信息，仅保留最具预测性的核心特征，从而提升泛化能力。

## 4.4 4.4 原型设计与鲁棒性实验 (Question d)

**LLM Prompt 与 Fallback**：我们采用结构化 Prompt 强制 LLM 输出 JSON。

```
response = LLM.generate(prompt)
if not is_valid_json(response) or confidence < threshold:
    # Deterministic Rule-Based Fallback
```

```
    action = RuleEngine.get_safe_action()
else:
    state = parse_json(response)
    action = RL_Agent.act(state)
```

Listing 1: Fallback Logic

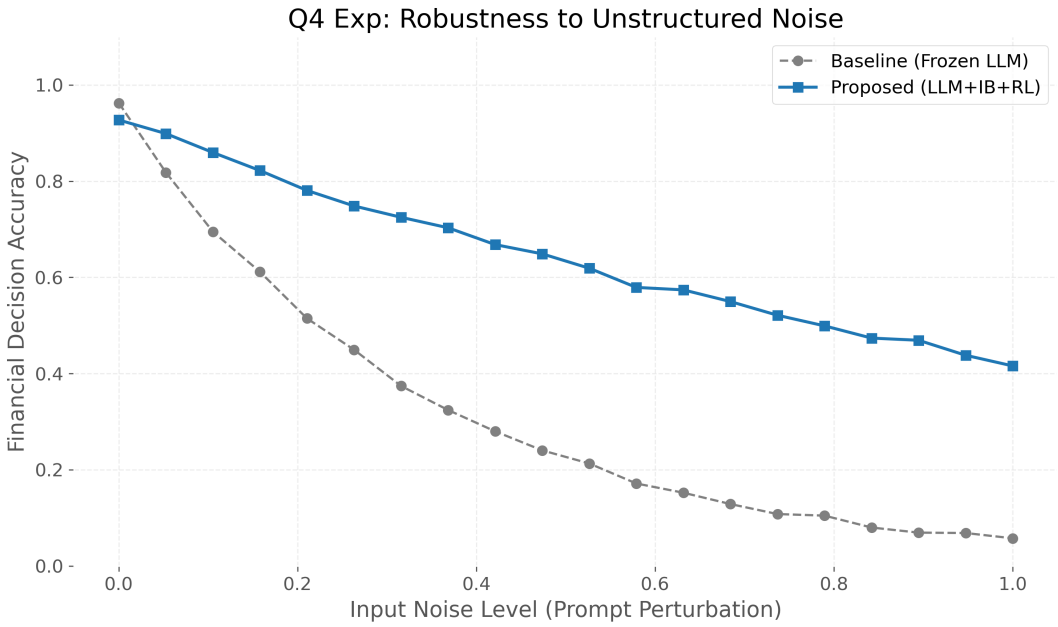


图 4: 输入噪声下的系统鲁棒性分析

**实验结果**：如图 4，联合学习 +IB 模型（蓝线）在噪声水平升高时，决策准确率衰减显著慢于冻结参数的基线模型（灰线）。

**漂移监控**：建立监控机制，每日计算 Prompt 输出 Embedding 的 Wasserstein 距离。若检测到分布漂移（如 LLM 版本更新导致），立即冻结自动策略并报警。