

实验 5: 智能投顾实验指南

一、实验目的

学习投资组合基本流程, 了解投资组合算法框架, 并实现经典投资组合策略。

二、实验步骤

1. 学习并编译示例代码

运行示例代码, 分析示例代码中 EW 策略的表现, 理解示例代码中的评价指标。

- 环境配置: 安装示例代码所需的 python 包(pytorch, scipy, pandas, numpy, matplotlib, cvxopt), 建议使用 pip 安装。
- 数据集配置: 下载数据集, 将 dataset 文件夹放置在 findme 目录下。
- 示例代码运行: 运行 findme/experiments/fintech_class_experiments/run_ew.py 文件。示例运行结果如图下所示, 依次是累计收益率、年化利率、最大回撤、波动率、夏普比率、卡玛比率、成交量。

```
Final cm for _Portfolio is: 0.9979809848119663
Final annualized_percentage_yield for _Portfolio is: -0.004270738181057054
Final max_draw for _Portfolio is: 0.12251570116019692
Final volatility for _Portfolio is: 0.18040154759236163
Final sharpe_ratio for _Portfolio is: -0.023673511885315343
Final calmar_ratio for _Portfolio is: -0.03485870088987858
Final turnovers for _Portfolio is: 0.018292482942342758
```

2. 实现马科维茨投资组合策略(MV)、EG 算法、ONS 算法

使用 cvxopt 库中的二次规划函数、优化函数, 根据理论课中讲述的算法流程, 实现马科维茨投资组合策略(MV)、EG 算法、ONS 算法。分析三个策略的表现, 利用评价指标对三个算法进行性能对比。实验的数据集统一使用 ashare_sse300 数据集。

- MV 算法实现: 在 findme/compute_platform/computing/base_strategy/modules/optimizor/mvo.py 文件中编写代码, 实现 cvxopt.solvers.qp 中需要用到的 P, q, G, h, A, b 矩阵。

```
def mean_variance_optimizer(mean, covariance, y):
    """
    Args:
        mean: numpy.ndarray, shape is [stock_num]
        covariance: numpy.ndarray, shape is [stock_num, stock_num]
    """
    k = mean.shape[0]

    #####
    # todo: 编写代码, 构造二次规划问题。实现cvxopt.solvers.qp中需要用到的P,q,G,h,A,b矩阵
    #####

    portfolio = solvers.qp(P, q, G, h, A, b)['x']
    return np.squeeze(portfolio)
```

- b. EG 算法实现：在 findme/compute_platform/computing/base_strategy/modules/optimizor/eg.py 文件中编写代码，计算投资组合的权重 weight。

```
def weight_compute(self, price_relative, last_weight):
    """
        price_relative: torch.tensor, the shape is [stock_num]
        last_weight: torch.tensor, the shape is [stock_num]
    """
    #####
    # todo: 编写代码, 计算当前round的weight
    #####

    # 返回投资组合权重
    return weight
```

- c. ONS 算法实现：在 findme/compute_platform/computing/base_strategy/modules/optimizor/ons.py 文件中编写代码，依次计算梯度、A、b 和投资组合权重 weight。

```
def weight_compute(self, price_relative, last_A, last_b, last_weight):
    #####
    # todo: 编写代码, 计算梯度

    # todo: 编写代码, 更新A

    # todo: 编写代码, 更新b

    # todo: 编写代码, 使用projection_in_norm函数进行单纯形投影投影, 获得weight
    #####
    return A, b, weight
```

- d. 运行程序，获得实验结果：三个算法程序运行入口在 findme/experiments/fintech_class_experiments/目录下，依次是 run_mv.py, run_eg.py, run_ons.py。原则上不需要修改 run 脚本中的代码。实验结果会自动存储在 findme/exp_result/results/fintech_class_experiments/目录下。

```
zmy@descenter33-SYS-70496P-TRT:~/ymy_workplace/findme/exp_result/results/fintech_class_experiments$ ls
eg  ew  mvo  ons
```

```
zmy@descenter33-SYS-70496P-TRT:~/ymy_workplace/findme/exp_result/results/fintech_class_experiments/ew/Portfolio_ashare_sse300_1662171704_1736622/metric$ ls
metric.pt                                     OnlineDecision_1__cm_reward_Portfolio.csv   OnlineDecision_1__volatility_Portfolio.csv
n_mode_call.pt                               OnlineDecision_1__max_draw_Portfolio.csv     OnlineDecision_1__weight_Portfolio.csv
OnlineDecision_1__annualized_percentage_yield_Portfolio.csv  OnlineDecision_1__sharpe_ratio_Portfolio.csv
OnlineDecision_1__calmar_ratio_Portfolio.csv   OnlineDecision_1__turnovers_Portfolio.csv
```

3. 对 EG 算法和 ONS 算法进行 regret 的数学推导

阅读参考文献中的 EG 和 ONS 算法相关文献,理解其中的 regret 相关定理,并且完成数学推导及证明。

三、提交内容

1. **实验报告:** 包含上述三项实验内容。理论推导可 latex 编辑、word 或者手写拍照。
2. **实验代码:** 程序中的编写的策略代码文件 mvo.py, eg.py, ons.py。

注意: 请在学在浙大平台上提交, 请勿上传数据集。

参考文献:

1. EG 算 法 : ON-LINE PORTFOLIO SELECTION USING MULTIPLICATIVE UPDATES[Mathematical Finance'98]
2. ONS 算法: Algorithms for Portfolio Management based on the Newton Method [ICML'06]