

相关信息

1. 考试时间：2023 年 11 月 07 日，第十周星期二，09:55~11:55；
2. 考试地点：徐汇校区四 105 教室；
3. 任课老师：林黎；
4. 还没交两次作业的同学，之后会开补考通道，但是会扣分数；
5. 课程考试总体难度不大，上课认真听讲的同学一定可以通过考试；
6. 题型：判断题 25 题，大部分比较简单，小部分陷阱题；选择题 5 题；计算题 2 题；简答题 2 题。
7. 本手稿中宋体字是 PPT 和课本综合整理，楷体字是个人理解，**两部分都不保证正确性**，如有观点冲突欢迎交流。
8. 习题少部分由同学讨论出具，大部分由 ChatGPT 出具，同样**不保证正确性**。

版本更新

1. 更新了 1.3 中的图片，修正了 7.8 的错误（2023.11.4）；
2. 更新了 1.Test 和 1.Answer 中错误的部分（2023.11.4）；
3. 更新了 2.13 和 2.14 中的公式错误（2023.11.4）；
4. 更新了第三章-第六章部分（2023.11.4）；
5. 更新了第七章-第八章部分（2023.11.5）。

如有其他错误，请联系我一并更正。

最后：

感谢林黎老师的辛勤付出！

感谢周学勤同学的辛勤付出！该手稿中的习题大部分是周学勤同学命制的习题。

希望该手稿可以节约您宝贵的复习时间。

衷心祝愿计金（双）200 班所有同学前程似锦。

第一讲 金融机器学习导论

1.1 学习金融机器学习的必要性：

- ①**金融操作需要根据预先制定的规则做出决策：**金融市场转变为超高速、超连接的信息交换网络，需要计算机在规则基础上做出操作，如期权定价、算法执行或风险监控。
- ②**金融算法化进程的需求：**人类不善于冷静做决策，但确定算法的机器则不会。
- ③**计算机算力更高：**计算机算力显著高于人类，可以轻易找出隐藏在金融数据中人类所找不到的结构和模式。

1.2 不同于其他机器学习的应用领域，**金融数据的信噪比更低**，且金融数据往往与时间关联密切，所以金融机器学习必须当作一个专门的学科来看待。

1.3 传统机器学习转向金融机器学习时存在陷阱：

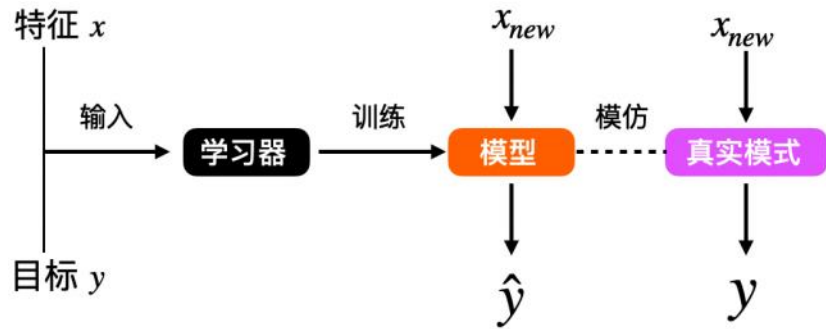
类别	陷阱	可选应对方案
1. 认识论	单打独斗的西西弗斯范式	团队协作的元策略范式
2. 认识论	依据回测来筛选策略	依据特征重要性分析
3. 数据处理（标签）	固定时长标签法	三限法
4. 数据处理（标签）	同时决定交易方向与交易规模	元标签法
5. 数据处理（信息）	每一个示例看作独立同分布 iid	最大化去重赋权抽样
6. 模型训练	交叉验证时有信息泄露	净化与隔离处理
7. 模型评估	根据真实历史路径进行回测	组合净化交叉验证 CPCV
8. 模型评估	以回测结果为目标导致过拟合	组合对称交叉验证 CSCV

单打独斗的西西弗斯范式：全权委托投资组合管理人做投资决策时，不会遵循一个特别的定律或基本原则，他们主要依赖他们自己的判断或直觉而不是逻辑；如果采用全权委托投资组合管理人的方式做量化项目，容易使所有员工都单打独斗，这就是所谓的让每个员工日复一日搬石头上山的西西弗斯范式，这种范式的投入产出比极低。

团队协作的元策略范式：每个数据分析师的角色专注于特定的任务，成为该任务的独特人选，但也具备全局视野。新的发现是团队整体努力的结果而非特定的个人为全部策略负责。

1.4 **机器学习：**是人工智能的一个分支；是一种计算机利用学习算法，从经验数据产生捕捉规律模式的模型，最后利用模型来改进预测的方法体系。

机器学习的运行原理：



1.5 **算法：**为达到目标而遵循一组规则集合，包括从输入到输出的所有步骤；

1.6 **机器学习算法：**狭义的指代学习器，即模型的学习过程，也称学习算法。广义的指机器学习运作中每一个环节的所涉及的算法。（例如数据准备环节）

1.6 **模型**：将输入 x_{new} 映射到预测 \hat{y} 的程序，也称为预测器；

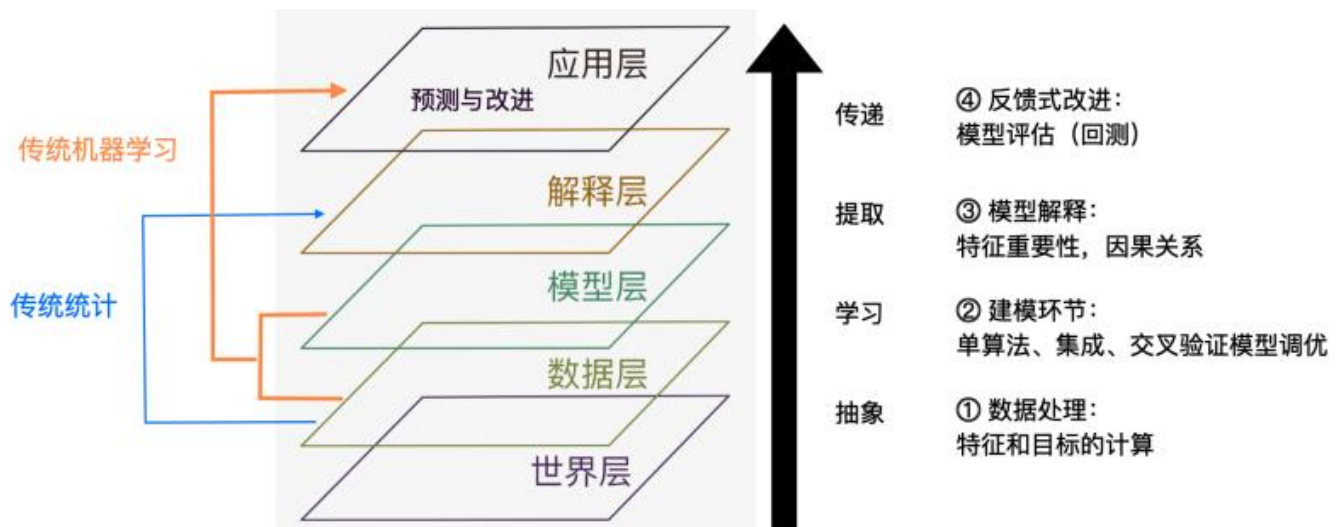
1.7 **黑箱模型**：通过查看了参数也无法理解的模型，如很多非线性模型；

1.8 **数据集**：数据集指一个**结构化的二维表格**，包括计算机要学习的数据。分为**目标列组**（y）和**特征列组**（X）；

1.9 **变量**：数据集中的每一列。目标列又称因变量，特征列又称自变量；

1.10 **实例**：数据集中的每一行，也称为数据点、样本点、观测点、示例。

1.11 **特征**：特征表现为数据集中的列，很多时候特征的含义是明确的。如果特征很难解释，那模型的行为就很难理解。



1.12 **训练集**：用于输入学习器进行训练获得模型的数据集；

1.13 **测试集**：学得模型后，用来测试检验模型效果的数据；

1.14 **模型的泛化能力**：模型在新样本上的预测表现。训练集只是样本空间的很小采样，强泛化能力的模型能很好的适应整个样本空间。

1.15 **假设空间**：潜在的真实模式称之为假设，学习过程是在所有的假设组成的空间中进行搜索的过程，找到与训练集匹配的假设。假设的经验对应物就是模型。（模型是来自于从经验数据捕捉特定模式）

1.16 **归纳偏好**：机器学习算法在学习过程中对某种类型假设的偏好称为“归纳偏好”，它是学习算法的“价值观”。

1.17 **好模型的两个要求**：（1）与训练集和测试机的匹配程度好；（2）整个样本空间的适应度高。

1.18 **损失函数**是评价单个样本拟合度的方法，常见的损失函数有 0-1 Loss；Log Loss；Mean Squared Error (MSE)：

$$L(y, f(x)) = \begin{cases} 1, & y \neq f(x) \\ 0, & y = f(x) \end{cases}$$

$$L(y, f(x)) = -\ln P_f(y|x)$$

$$L(y, f(x)) = (y - f(x))^2$$





1.19 **匹配程度**：是在平均意义下的损失情况，是一种经验的损失

$$\mathbb{E}^{in}(f) = \frac{1}{N} \sum_i^N L(y_i, f(x_i))$$

1.20 **适应程度**：是在期望意义下的损失情况，是一种推断的损失

$$\mathbb{E}^{out}(f) = \sum_{x \in \mathcal{X}} L(y, f(x)) P(y, f(x))$$

1.21 混淆矩阵:

		真实情况					
	总量				Prevalence	1-Prevalence	先验概率
预测值		True Positive	False Positive	TP+FP	Positive Predictive Value Precision	False Discovery Rate	后验概率
		False Negative	True Negative	FN+TN	False Omission Rate	Negative Predictive Value	后验概率
		TP+FN	FP+TN				
		True Positive Rate Sensitivity Recall Prob. of Detection	False Positive Rate Fall-out Prob. of False Alarm		Accuracy	LR+	
		False Negative Rate Miss Rate	True Negative Rate Specificity			LR-	
		条件概率	条件概率			DOR	

二级指标:

	公式	意义
准确率 <i>ACC</i>	$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$	分类模型所有判断正确的结果占总观测值的比重
查准率 <i>PPV</i>	$\text{Precision} = \frac{TP}{TP+FP}$	在模型预测是Positive的所有结果中，模型预测对的比重
查全率 <i>Sen</i>	$\text{Sensitivity} = \text{Recall} = \frac{TP}{TP+FN}$	在真实值是Positive的所有结果中，模型预测对的比重
特异度 <i>TNR</i>	$\text{Specificity} = \frac{TN}{TN+FP}$	在真实值是Negative的所有结果中，模型预测对的比重

三级指标: F1 值是查全率 Sen 和查准率 PPV 的调和平均值, 综合了 PPV 和 Sen 的产出结果。F1 值的取值范围是从 0 到 1 的, 1 代表模型输出最好, 0 代表模型输出最差。

$$F1 = \frac{2}{\frac{1}{Sen} + \frac{1}{PPV}} = \frac{2 \cdot Sen \cdot PPV}{Sen + PPV}$$

1.22 参数: 由机器学习而得的参数, 依赖于训练集。

1.23 超参数: 无法自己学习出来, 通过人为给定。来源于学习算法, 是人为事先设定的, 无法自己学习出来, 是模型与身俱来带有的。

1.24 学习算法的选择由归纳偏好来决定, 偏好通常来自于某种理论 (如奥卡姆剃刀); 学习算法同时确定了有哪些超参数。

奥卡姆剃刀：如果有多个假设（模型）与观测一致，则选择最简单的那一个。奥卡姆剃刀只是一个可能的归纳偏好原则。

1.25 在既定的算法下，模型的选择其实就是对超参数取值的选择，通过训练来完成。

1.26 没有免费午餐定理：

没有免费午餐定理 (No Free Lunch Theorem, NFL): 如果所有需要机器学习解决的问题 (T) 等概率出现，则平均误差与学习算法无关！即

$$\mathbb{E}^{out}(\mathcal{A}_a | X, T) = \sum_f \sum_{x \in \mathcal{X}-X} P(x) L(T(x), f(x)) P(f | X, \mathcal{A}_a)$$

$$\mathbb{E}^{out}(\mathcal{A}_b | X, T) = \sum_f \sum_{x \in \mathcal{X}-X} P(x) L(T(x), f(x)) P(f | X, \mathcal{A}_b)$$

那么

$$\sum_T \mathbb{E}^{out}(\mathcal{A}_a | X, T) = \sum_T \mathbb{E}^{out}(\mathcal{A}_b | X, T)$$

因此，脱离具体问题，空谈“哪种机器学习算法好”没有意义

1.27 欠拟合：模型并没有很好的捕捉到训练样本的一般性质，通过增加模型复杂度可以避免欠拟合

1.28 过拟合：模型把训练样本有的特质当作潜在样本的一般性质，造成模型匹配度很高，但泛化的适应度却变低的现象。原因：模型 f 的复杂度 $> T$ （需要机器学习解决的问题）的复杂度，模型复杂度与算法和超参数取值有关。如果 $N \neq NP$ 过拟合就不可避免，只能缓解。

1.29 交叉验证：选择模型时，不调整算法，选择匹配度和适应度均较好的超参数取值。该方法无法克服由学习算法复杂度导致的过拟合。

1.30 正则化方法：调整算法，在学习算法中增加控制模型复杂度的惩罚项。该方法增加了算法的超参数个数，也会引入过拟合。

1.31 外部交叉验证：综合评价在超参数个数固定时，学习算法复杂度导致的过拟合性。

外部交叉验证时，每一轮使用的是不同的超参数值，因此只能检验学习算法的性能，而不能用于确定模型。

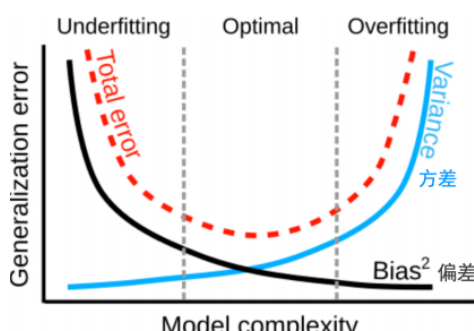
1.32 偏差-方差分解：解释模型泛化能力（适应程度）的工具。

假设学习算法 \mathcal{A} 在训练集 X 上获得模型 f ，真实的规律为 $T, y = T + \varepsilon$ ，且 $\mathbb{E}(\varepsilon) = 0$ 。
记 $\bar{f} = \mathbb{E}^X(f)$ ，那么

$$\mathbb{E}^X[(f - y)^2] = (\bar{f} - T)^2 + \mathbb{E}^X[(f - \bar{f})^2] + \mathbb{E}^X(\varepsilon^2)$$

$$\mathbb{E}^{out}(f) = f \text{ 偏差}^2 + f \text{ 方差} + \text{噪声}^2$$

- 偏差：度量期望预测与真实结果的偏离程度，表示拟合能力
- 方差：同样大小的训练集变动导致的模型预测变化，表示模型对数据的敏感性
- 噪声：数据中真实规律起作用的程度，表示学习的难度



偏差和方差此消彼长，不能同时消减。模型复杂度提高，偏差下降但是方差上升。

1.Test 判断

- 1)数据集一定是结构化的。
- 2)数据集分为目标列组和特征列组。
- 3)特征的含义必须是明确的且具有金融意义的。
- 4)特征含义的明确与否对模型可解释性具有重要意义。
- 5)训练集和测试集共同构成了样本空间。
- 6)模型就是假设，假设就是模型。
- 7)F1 值越接近 0 往往代表模型的输出结果较好。
- 8)F1 值的取值范围是全体实数。
- 9)模型在训练集和测试集上的超参数往往是不一样的。
- 10)奥卡姆剃刀原则是我们在金融机器学习中必须遵循的准则。
- 11)Random Forest 算法是比 SVM 算法更好的算法。
- 12)如果两个模型都是 SVM 算法，他们的超参数也是一样的。
- 13)过拟合和欠拟合都是可以避免的。
- 14)过拟合和欠拟合都是不可避免的。
- 15)模型复杂度只与算法有关。
- 16)外部交叉验证可以用于确定模型。
- 17)外部交叉验证可用于评价超参数带来的过拟合问题。
- 18)交叉验证可用于评价算法确定的情况下，不同超参数取值带来的过拟合问题。
- 19)适应程度描述的是模型的泛化能力。
- 20)在算力足够的情况下，应当尽量增加模型复杂度。

1.Test 计算

若 TP=43，FP=8，TN=41，FN=6，请计算 ACC、Sen、PPV、F1 值（过程和结果均保留六位小数）。

1.Answer 判断

- | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|
| 1)√ | 2)√ | 3)× | 4)√ | 5)× | 6)× | 7)× | 8)× | 9)× | 10)× |
| 11)× | 12)× | 13)× | 14)× | 15)× | 16)× | 17)× | 18)√ | 19)√ | 20)× |

1.Answer 计算

ACC=0.857143； Sen=0.877551； PPV=0.843137； F1=0.860000

第二讲 金融数据结构化算法

2.1 结构化数据与非结构化数据：

结构化数据：是以二维表结构来实现逻辑表达的数据。数据的每一行代表产生一个金融事件（或一次观测），数据的每一列代表事件（或观测）包含的变量。尽量保证每一行观测都包含独立的信息。

非结构化数据：某种未被人为整理的高频数据。如新闻数据、电子邮件/聊天/微信/微博、网页、搜索数据、卫星/监控数据、视频会议、刷卡记录、App 生成数据等等。

2.2 金融数据结构化的动机：

①数据的独立同分布性（i.i.d）是机器学习算法能够有效识别规律、进行预测的前提，因为学习算法一般只接受结构化的数据；

②结构化数据更易于交流展示、储存和关联。

2.3 标准结构化法：标准结构化法的作用是将一系列以不规则频率观测的数据，化为通过规律性采样获得的衍生均匀序列。标准结构化方法有等时长采样、等时点采样、等额采样、等量采样等。

2.4 等时长采样（Time bar）：

一种以固定的时长间隔来记录数据的结构化方法。通常是将非结构化的 tick 级数据转化为结构化的周期数据。

tick 级数据：tick 级数据指的是以交易为单位记录的价格和成交量的时间序列数据。在股票、期货、外汇等金融市场中，每个交易的价格和成交量都被认为是一个 tick。以下图片表示了一组 tick 级数据

stockcode	time	price	bi	volv	vol	direction	buy1	buy2	buy3	buy4	buy5	sell1	sell2	sell3	sell4	sell5
CN600000	2020-01-02 09:32:35	12.51	34	1117028	893	B	12.50	12.49	12.48	12.47	12.46	12.51	12.52	12.53	12.54	12.55
CN600000	2020-01-02 09:32:38	12.51	32	885964	709	B	12.51	12.50	12.49	12.48	12.47	12.52	12.53	12.54	12.55	12.56
CN600000	2020-01-02 09:32:41	12.50	30	968788	775	S	12.49	12.48	12.47	12.46	12.45	12.50	12.51	12.52	12.53	12.54
CN600000	2020-01-02 09:32:44	12.52	27	1015536	812	B	12.50	12.49	12.48	12.47	12.46	12.52	12.53	12.54	12.55	12.56
CN600000	2020-01-02 09:32:47	12.52	21	343044	274	B	12.51	12.50	12.49	12.48	12.47	12.52	12.53	12.54	12.55	12.56
CN600000	2020-01-02 09:32:50	12.51	10	210160	168	S	12.50	12.49	12.48	12.47	12.46	12.51	12.52	12.53	12.54	12.55
CN600000	2020-01-02 09:32:53	12.52	19	526712	421	B	12.51	12.50	12.49	12.48	12.47	12.52	12.53	12.54	12.55	12.56
CN600000	2020-01-02 09:32:56	12.51	12	155212	124	S	12.51	12.50	12.49	12.48	12.47	12.52	12.53	12.54	12.55	12.56
CN600000	2020-01-02 09:32:59	12.51	9	125156	100	S	12.51	12.50	12.49	12.48	12.47	12.52	12.53	12.54	12.55	12.56
CN600000	2020-01-02 09:33:02	12.52	29	2322920	1856	B	12.51	12.50	12.49	12.48	12.47	12.52	12.53	12.54	12.55	12.56
CN600000	2020-01-02 09:33:05	12.52	42	848852	678	B	12.52	12.51	12.50	12.49	12.48	12.53	12.54	12.55	12.56	12.57
CN600000	2020-01-02 09:33:08	12.53	62	1120744	895	B	12.52	12.51	12.50	12.49	12.48	12.53	12.54	12.55	12.56	12.57
CN600000	2020-01-02 09:33:11	12.51	26	315684	252	S	12.51	12.50	12.49	12.48	12.47	12.52	12.53	12.54	12.55	12.56

等时长采样时，需要把一段相同时间里的一组数据进行汇总，计算以下指标：

- ① OHLC: open, high, low, close 价格
- ② Volumn: 总成交量
- ③ 时间加权平均价 TWAP:

$$TWAP(t_i, t_j) = \frac{1}{n} \sum_{t_{i,s}=1}^{t_{i,s}=n} price(t_{i,s})$$

- ④ 成交量加权平均价 VWAP:

$$VWAP(t_i, t_j) = \frac{\sum_{t_{i,s}=1}^{t_{i,s}=n} price(t_{i,s}) * volume(t_{i,s})}{\sum_{t_{i,s}=1}^{t_{i,s}=n} volume(t_{i,s})}$$

等时长采样往往具有以下缺点：

- ① 相同周期内金融市场的信息含量有显著不同；
- ② 等时长采样下的金融序列的统计学特性很差：不是正态分布（高斯分布）、不同实例并非独立同分布 i.i.d。

2.5 等时点采样 (Tick bar):

一种以固定的报价变动数量(tick)间隔来记录数据的结构化方法。

等时点采样时, 需要把一段相同时点里的一组数据进行汇总, 计算 OHLC、总成交量、TWAP 和 VWAP 等指标。

等时点采样的优点: 能够获得接近 i.i.d 正态分布的回报。

2.6 等量采样 (Volume bar):

一种以固定的成交量(股)来记录数据的结构化方法。

等量采样时, 需要把一段相同交易量里的一组数据进行汇总, 计算 OHLC、总成交量、TWAP 和 VWAP 等指标。

等量采样的优点: 能够获得比等时点采样获得接近于 i.i.d 正态分布的回报。

2.7 等额采样 (Value bar):

一种以固定的成交额(元)来记录数据的结构化方法。

等额采样时, 需要把一段相同交易额里的一组数据进行汇总, 计算 OHLC、总成交量、TWAP 和 VWAP 等指标。

等额采样的优点: 当价格有大幅波动时, 采样的回报更加平稳, 容易记录趋势; 屏蔽股本变化带来的影响。

2.8 信息驱动的结构化方法: 信息驱动的结构化方法的目的是在新信息进入市场时进行更加频繁的采样, 信息匮乏时采样频率相应放慢, 从而将采样和信息的流动同步, 获得等信息量的衍生序列。

2.9 时点平衡性采样 (Time Imbalance bars):

时点平衡性采样本质上是捕捉不平衡现象, 用于衡量买卖力量的不平衡情况。当分时不平衡超出预期的时候进行采样。

买卖不平衡度被定义为:

$$b_t = \begin{cases} b_{t-1} & \text{if } \Delta p_t = 0 \\ \frac{|\Delta p_t|}{\Delta p_t} & \text{if } \Delta p_t \neq 0 \end{cases}$$

这里可以理解为, 力量对比买>卖时, 价格上升, $b_t=1$; 力量对比卖大于买时, 价格下降, $b_t=-1$ 。

然后, 将买卖不平衡度累加起来表示在一段时间内买/卖力量的持续强弱关系:

累计买卖不平衡度被定义为:

$$\theta_{t,j} = \sum_{i,s=1}^{t,j} b_{t,i,s} \quad \theta_T = \sum_{t=1}^T b_t$$

最后, 在买卖双方力量对比的持续强弱关系, 也即累计买卖不平衡度偏离某一平均水平的时候, 我们考虑将该处进行一个切割。那么, 我们首先要找到这个平均水平(期望):

$$\begin{aligned} \mathbb{E}_0(\theta_T) &= \mathbb{E}_0 \left[\sum_{t=1}^T b_t \right] \\ &= \mathbb{E}_0 \left[\sum_{t=1}^{+\infty} b_t \cdot \phi(T > t-1) \right] = \sum_{t=1}^{+\infty} \mathbb{E}_0(b_t) \cdot \mathbb{E}_0(\phi(T > t-1)) \\ &= \mathbb{E}_0(b_t) \cdot \sum_{t=1}^{+\infty} \mathbb{E}_0(\phi(T > t-1)) = \mathbb{E}_0(b_t) \cdot \sum_{t=1}^{+\infty} P(T > t-1) \\ &= \mathbb{E}_0(b_t) \cdot \sum_{t=0}^{+\infty} P(T > t) = \mathbb{E}_0(b_t) \cdot \mathbb{E}_0(T) \\ &= \mathbb{E}_0(T) \cdot (P(b_t = 1) - P(b_t = -1)) \end{aligned}$$

得到这样一个平均水平之后, 我们只需考虑是否偏离了这个平均水平(绝对值大于期望的绝对值):

$$|\theta_T| \geq |\mathbb{E}_0(\theta_T)|$$

每当出现一个这样的时刻, 我们就将序列在该时刻切断, 距离上一次切断的长度就是 T:

$$T^* = \arg \min_T \{ |\theta_T| \geq \mathbb{E}_0(T) \cdot [P(b_t = 1)\mathbb{E}_0(v_t | b_t = 1) - P(b_t = -1)\mathbb{E}_0(v_t | b_t = -1)] \}$$

2.10 成交量平衡性采样 (Volume Imbalance bars): 将累积的成交量平衡度限定在一定范围内的 tick 数据进行汇总

可以预计的是, 虽然在某一时刻, 买卖力量的对比总可以认为有强弱之分, 但是对于不同的时刻来说, 重要性在持续对比中是不一样的, 这里认为成交量高的时刻更重要, 所以有必要用成交量给予加权。

$$\theta_T = \sum_{t=1}^T b_t v_t$$

然后求期望, 并截断求得 T 即可。

$$T^* = \arg \min_T \{|\theta_T| \geq \mathbb{E}_0(T) \cdot [P(b_t = 1)\mathbb{E}_0(v_t | b_t = 1) - P(b_t = -1)\mathbb{E}_0(v_t | b_t = -1)]\}$$

2.11 成交额平衡性采样 (Value Imbalance bars): 将累积的成交额平衡度限定在一定范围内的 tick 数据进行汇总

这里认为成交额高的时刻更重要, 所以有必要用成交额给予加权。

$$\theta_T = \sum_{t=1}^T b_t q_t = \sum_{t=1}^T b_t v_t \text{price}(t)$$

然后求期望, 并截断求得 T 即可。

$$T^* = \arg \min_T \{|\theta_T| \geq \mathbb{E}_0(T) \cdot [P(b_t = 1)\mathbb{E}_0(q_t | b_t = 1) - P(b_t = -1)\mathbb{E}_0(q_t | b_t = -1)]\}$$

2.12 时点游程采样 (Ticks Runs bars): 将单向的买卖持续度限定在一定范围内的 tick 数据进行汇总

之前我们认为, 在一段时间内买卖强弱力量的对比之和体现了市场微观上的信息量。但时点游程认为, 如果某一段时间内有密集的单向买/卖力量, 说明市场微观上的信息量较多。或许该时间段买/卖的次数较高的一项可以被用来体现这一点

$$\theta_T = \max \left\{ \sum_{t|b_t=1}^T b_t, \sum_{t|b_t=-1}^T -b_t \right\}$$

然后求期望, 并截断求得 T 即可。

$$T^* = \arg \min_T \{|\theta_T| \geq \mathbb{E}_0(T) \cdot \max[P(b_t = 1), 1 - P(b_t = 1)]\}$$

2.13 交易量游程采样 (Volume Runs bars): 将单向的成交量持续度限定在一定范围内的 tick 数据进行汇总

在时点游程基础上, 认为成交量高的时刻更重要, 所以有必要用成交量给予加权。

$$\theta_T = \max \left\{ \sum_{t|b_t=1}^T b_t v_t, \sum_{t|b_t=-1}^T -b_t v_t \right\}$$

然后求期望, 并截断求得 T 即可。

$$T^* = \arg \min_T \{|\theta_T| \geq \mathbb{E}_0(T) \cdot \max[P(b_t = 1)\mathbb{E}_0(v_t | b_t = 1), [1 - P(b_t = 1)]\mathbb{E}_0(v_t | b_t = -1)]\}$$

2.14 交易额游程采样 (Value Runs bars): 将单向的成交额持续度限定在一定范围内的 tick 数据进行汇总

在时点游程基础上, 认为成交额高的时刻更重要, 所以有必要用成交额给予加权。

$$\theta_T = \max \left\{ \sum_{t|b_t=1}^T b_t q_t, \sum_{t|b_t=-1}^T -b_t q_t \right\}$$

然后求期望, 并截断求得 T 即可。

$$T^* = \arg \min_T \{|\theta_T| \geq \mathbb{E}_0(T) \cdot \max[P(b_t = 1)\mathbb{E}_0(q_t | b_t = 1), [1 - P(b_t = 1)]\mathbb{E}_0(q_t | b_t = -1)]\}$$

2.15 在实际操作中, 我们采用已样本的采样间隔 T 的指数移动平均来近似代替 T 的期望。

之所以不用平均值, 应该是因为要体现出“时序”。即: 越接近“当前”的值在平均中的权重应越高。

2.16 在实际操作中, 我们采用已样本中买入性 tick 所占的比例的指数移动平均来近似代替 $P(b_t=1)$ 。

2.17 多产品序列结构化处理的动机：

- ① 有时我们需要对权重随时间动态调整的工具的时间序列进行建模；
- ② 有时需要对不定期支付息票或股息的金融产品进行处理；
- ③ 有时我们需要对受制于公司行为的产品进行处理；
- ④ 有时需要对研究中能够使时间序列的性质发生变化的事件进行处理。

在这种情况下，我们往往将复杂的多产品数据转化为一个回报类似 ETF 的单一数据（ETF 分时法）

2.18 ETF 分时法

假设结构化采样的时点为 $t = 1, 2, \dots, T$ ，组合中的金融资产为 $i = 1, 2, \dots, I$ 。当结构化采样产生后，已知如下数据

- $o_{i,t}$ ：原始的开盘价
- $p_{i,t}$ ：原始的收盘价
- $\varphi_{i,t}$ ：时点上一个点的价值
- $v_{i,t}$ ：时点上的成交量
- $d_{i,t}$ ：时点上支付的息票，息票与资金成本之差 (carry)，股利分红

这里还需要补充一个假设：在每一个 t 的离散取值，所有的金融资产都可以自由交易。也就是说，即便 $t-1$ 到 t 的这段时间市场无法交易，所有金融资产都应该可以在 $t-1$ 和 t 时刻交易。

w_t 表示在采样时点的子集 $B \subseteq \{1, 2, \dots, T\}$ 中资产组合的重新平衡权重，转换为单一回报的组合价值 K_t 按照如下方式计算

$$K_t = K_{t-1} + \sum_i^I h_{i,t-1} \varphi_{i,t} (\delta_{i,t} + d_{i,t})$$
$$\delta_{i,t} = \begin{cases} p_{i,t} - o_{i,t}, & \text{如果 } t-1 \in B \\ \Delta p_{i,t}, & \text{否则} \end{cases} \quad h_{i,t-1} = \begin{cases} \frac{w_{i,t-1} K_{t-1}}{o_{i,t} \varphi_{i,t-1} \sum_i^I |w_{i,t-1}|}, & \text{如果 } t-1 \in B \\ h_{i,t-2}, & \text{否则} \end{cases}$$

- $h_{i,t}$ 表示在时刻 t 资产 i 的持有量（合约数量）
- $\delta_{i,t}$ ：资产 i 从 $t-1$ 到 t 的市场价值变化
- $\frac{w_{i,t}}{\sum_i^I |w_{i,t}|}$ 表示每个资产的占比分配

这里有一些抽象，需要记住我们正在研究一个期货（以股指期货举例好理解）。其他的不难理解，重点在于这个式子

$$h_{i,t-1} = \frac{w_{i,t-1} K_{t-1}}{o_{i,t} \varphi_{i,t-1} \sum_i^I |w_{i,t-1}|},$$

把他改写成这样就好理解了：

$$h_{i,t-1} p_{i,t-1} \varphi_{i,t-1} = \frac{w_{i,t-1}}{\sum_i^I |w_{i,t-1}|} K_{t-1}$$

左边的三个分别表示：期货合约的数量、股票指数开盘价、股指期货合约乘数（股票指数变动 1 个点，股指期货变动 200/300 元），这样得到了第 i 个产品的美元价格。右边则是将 $t-1$ 时刻的组合价值用 PCA 方法重新分配到每个资产，分母则用来去杠杆（将 w 向量的模场归一化）。由于 $p_{i,t-1}$ 没有办法实时得到，用 $o_{i,t}$ 代替。

然后用 PCA 求出权重：

- 使用主成分分析法 (Principle Component Analysis)
- $\mu_{1 \times N}$ ：收益均值向量
- $V_{N \times N}$ ：协方差矩阵
- 第一步：进行谱分解： $VX = X\Lambda$
- 第二步：引入 w_i ， $\sigma^2 = w^T V w = w^T X \Lambda X^T w = \beta^T \Lambda \beta$

■ 第三步，主成分分解：第 n 个成分的风险为 $R_n = \beta_n^2 \Lambda_{n,n} \sigma^{-2} = (X^T w)_n^2 \lambda_n \sigma^{-2}$ ， R_n 是第 n 个主成分的风险

■ 第四步，计算 β ： $\beta_n = \sigma \sqrt{\frac{R_n}{\lambda_n}}$

■ 第五步，计算 w_i ： $w = X\beta$

2.19 结构化抽样的动机：

① 包括 SVM 在内的很多机器学习算法无法对样本进行一个有效缩放，不支持规模较大的数据；

② 只有数据具有足够的信息量时，机器学习算法才能形成具有较高预测精度的学习器。

这里的结构化抽样介绍缩减抽样和事件驱动抽样。

2.20 缩减抽样：抽取出符合金融机器学习算法建模要求的数据量。其中，线性等分抽样是通过恒定的步长进行序列抽样，缺点是步长任意且结果受随机种子而变化；均匀抽样是以均匀分布进行随机抽样，缺点是数量任意且结果受随机种子而变化，二者的共同优点是比较简单，但不一定包含那些在预测能力或信息内容方面最相关数据样本。

2.21 事件驱动抽样：重要的事件往往带有较强预测能力的信息，事件可能与某些宏观经济统计数据的发布、波动率飙升、曲线差值与平衡水平的显著偏离等有关。这里介绍一种事件驱动抽样算法——CUSUM 过滤器。

设结构化的数据为 $y_t, t = 1, 2, \dots, T$ ，蓄势累积指标 $S_t = \max\{S_{t-1} + (y_t - \mathbb{E}_{t-1}(y_t)), 0\}$ 。如果累积指标大于阈值，对 t 进行抽样： $S_t \geq h$ 。

变体：对称抽样

$$\begin{aligned} S_t^+ &= \max\{S_{t-1}^+ + (y_t - \mathbb{E}_{t-1}(y_t)), 0\}, & S_0^+ &= 0 \\ S_t^- &= \min\{S_{t-1}^- + (y_t - \mathbb{E}_{t-1}(y_t)), 0\}, & S_0^- &= 0 \end{aligned}$$

在简单应用时，可以取 $\mathbb{E}_{t-1}(y_t) = y_{t-1}$

2.Test 判断

- 1)SQL 数据库中的数据往往都是结构化的。
- 2)财经网站上，股价 K 线的 HTML 代码是结构化的。
- 3)金融市场中，信息在时间上的分布是均匀的。
- 4)等时长采样在统计学特性上比等时点采样更好，这主要是因为他的采样结果更加符合 i.i.d
- 5)等量采样在统计学特性上比等时点采样更好，这主要是因为他的采样结果更加符合 i.i.d
- 6)等额采样在统计学特性上比等量采样更好，这主要是因为他的采样结果更加符合 i.i.d
- 7)等时点采样是一种以固定的时长间隔来记录数据的结构化方法
- 8)等额采样需要将一段相同成交量里的一组数据进行汇总，并计算相关指标
- 9)信息驱动的结构化方法的目的是在新信息进入市场时进行更加舒缓的采样，信息匮乏时采样频率相应加快，从而将采样和信息的流动同步，获得等信息量的衍生序列。
- 10)在实际操作中，我们采用已样本的采样间隔 T 的平均值来近似代替 T 的期望。

2.Test 简答

【思考】如何计算 $\mathbb{E}_0(v_t | b_t = 1)$? $\mathbb{E}_0(q_t | b_t = 1)$?

2.Answer 判断

1) \checkmark 2) \times 3) \times 4) \times 5) \checkmark 6) \times 7) \times 8) \times 9) \times 10) \times

2.Answer 简答

前者可以估算为前线所得购买成交量的指数加权移动平均数；后者可以估算为前线所得购买成交额的指数加权移动平均数。

第三讲 金融机器学习中的标签

3.1 金融机器学习打标签的动机：非监督机器学习允许直接从自变量集中提取可预测的模式，但监督型机器学习需要有因变量集 y ，因变量的量化通过打标签完成。

3.2 固定时限标签法（FH 法）：固定时间内对于某个股票，如果其收益高于阈值 c ，那么被分为正例（用+1 表示）；低于阈值 $-c$ ，那么被分为负例（用-1 表示）；如果在 $-c$ 和 c 之间，被分为第三类（用 0 表示）。

$$y_i = \begin{cases} -1, & \text{如果 } r_{t_i,0,t_i,0+h} < -c \\ 0, & \text{如果 } |r_{t_i,0,t_i,0+h}| \leq c \\ +1, & \text{如果 } r_{t_i,0,t_i,0+h} > c \end{cases}, \quad r_{t_i,0,t_i,0+h} = \frac{p_{t_i,0+h}}{p_{t_i,0}} - 1$$

固定时限标签法的问题：阈值 c 的选择是不变的，但价格波动率却随着时间变化而变化，因此：

在波动率很大时，价格很容易突破 $[-c,+c]$ ，很少样本会被标注为 0，大量 1；

而波动率很小时，价格不容易突破 $[-c,+c]$ ，很多样本会被标注为 0，少量 1。

可以用以下方法弥补：

① 使用等额采样或者等量采样；

之所以使用等额采样或者等量采样，是因为采样方式选用这两种采样方式的时候，波动率更接近常量（具有同方差性）。

② 使用滚动指数加权移动平均（EWMA）来估计波动率的动态阈值。

即便使用了这种弥补方法，依然存在问题：通常的策略会存在止损限额（如保证金制度），固定时限标签法无法反映价格变动的轨迹，因此不能将止损止盈机制纳入到策略中来。

3.3 三限标签法（TB 法）：是一种路径依赖的标签法。通常设立两个价格上的水平界限和一个时间上的垂直界限。

① 水平界限用来止盈和止损，其值均为估计波动性的动态函数；

② 垂直界限是一种过期时间限制，是以已产生线的数量的形式进行定义的。

③ 上水平线先被触及，标注为+1；下水平线先被触及，标注为-1；垂直界限先被触及，标注为 0 或者 $\text{Sign}(r)$ 。

需要注意的是，上下两条水平界限并不一定非要对称设置。另外，考虑到三个屏障各自可能无法生效，我们可以定义 $[\text{pt}, \text{sl}, \text{t1}]$ 分别表示止盈、止损、垂直三个界限是否生效。共有 8 种可能的状态。

3 种比较有用的状态：

① $[1,1,1]$ ：标准设置。希望实现盈利，但对损失和持有期限有最大限度。

② $[0,1,1]$ ：没有止盈条件。要么止损退出，要么到持有期限退出。

③ $[1,1,0]$ ：没有终止条件。除非达到了止盈或者止损线，不然不会推出。

3 中不太符合实际的状态：

④ $[0,0,1]$ ：等价于固定时间水平标识法。

⑤ $[1,0,1]$ ：持有头寸直到持有期结束，不考虑中间未实现的损失。

⑥ $[1,0,0]$ ：持有头寸直至盈利。但这可能表示在若干长的时间内都处于亏损头寸。

2 种不符合逻辑的状态：

⑦ $[0,1,0]$ ：无目标状态，停留于一个头寸直至被终止。

⑧ $[0,0,0]$ ：无屏障状态。头寸永远锁定，没有任何标签产生。

3.4 元标签法 (Meta-Labeling): 三限标签法只能训练出预测投资方向 (side) 的模型, 无法确定投资的仓位(size)。可通过在原始模型基础上进行二次标签来训练确定仓位的模型。

第一次, 根据三限标签训练初级模型确定投资方向, 优化出高查全率 **Sen** 的模型。在初级模型确定方向的数据集中重新打出是否入场的二分类标签, 训练次级模型, 以优化查准率 **PPV** 为目标, 通过模型给出的概率来确定仓位。

元标签法的意义:

- ① 仅需次级模型选择机器学习模型即可, 初级模型可以使用任意模型。包括: 机器学习模型, 计量经济学模型复杂统计模型, 基本面模型, 技术分析模型, 甚至是主观定性观点。
- ② 由于大多数的正例情况已经由初级模型捕捉, 次级模型相当于再对初级模型的阳性判定一次。这样可以同时兼顾查全率与查准率。这会使元标签法的模型具有更高的 F1 值。

使用元标签法的优势:

- ① 提升了模型的可解读性。先通过简单模型 (如基本面或者人的看法) 来确定头寸方向, 随后再使用复杂模型 (如机器学习模型) 提高预测精度。可以适用于量化基本面的建模;
- ② 兼顾定性和定量, 一定程度上限制了过拟合;
- ③ 将判断方向和确定仓位分开, 可生成复杂策略: 虽然用同一个次级模型来确定买卖头寸, 但是使用完全不同的初级模型确定买卖时机;
- ④ 准确预测的小头寸但大头寸预测不准确会叫人破产。开发一个仅针对关键决策 (控制规模) 准确性的机器学习算法非常重要。

3. Test 判断

- 1) 固定时限标签法 (FH 法) 将股票分为三类: 正例、负例和第三类。
- 2) 在波动率很大时, 固定时限标签法容易将样本标注为第三类。
- 3) 等额采样和等量采样是用来弥补固定时限标签法中样本不均衡的问题。
- 4) 等额采样和等量采样之所以可以用来弥补固定时限标签法中样本不均衡的问题, 是因为这两种采样方法得到的波动率较大, 第三类标签数量不会很多。
- 5) 滚动指数加权移动平均 (EWMA) 可以用来估计波动率的动态阈值。
- 6) 固定时限标签法可以反映价格变动的轨迹。
- 7) 固定时限标签法可以纳入止损止盈机制到策略中。
- 8) 三限标签法是一种路径依赖的标签法, 通常设两个价格上的水平界限和一时间上的垂直界限。
- 9) 三限标签法中, 垂直界限用来实现止盈和止损, 其值均为估计波动性的动态函数。
- 10) 垂直界限先被触及时, 在三限标签法中会标注为+1。
- 11) 三限标签法中, 上下两条水平界限必须对称设置。
- 12) [pt, sl, t1] 分别表示止盈、止损、垂直三个界限是否生效。共有 8 种可能的状态。
- 13) 在三限标签法中, [1, 1, 0] 表示没有止盈条件, 要么止损退出, 要么到持有期限退出。
- 14) 在三限标签法中, [0, 0, 1] 表示等价于固定时间水平标识法。
- 15) 在三限标签法中, [1, 0, 0] 表示持有头寸直至盈利。但这可能在若干长的时间内都处于亏损头寸。
- 16) 在三限标签法中, 八种可能的状态都是常用且符合逻辑的。
- 17) 在元标签法中, 次级模型优化的目标是查全率 **Sen**。
- 18) 元标签法中, 大多数正例情况已经由初级模型捕捉, 因此次级模型相当于再对初级模型的阈值进行调整。
- 19) 在元标签法中, 初级模型和次级模型都必须使用机器学习算法。
- 20) 元标签法可以准确预测小头寸和大头寸的投资方向和仓位大小。

3. Answer 判断

- | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1) √ | 2) × | 3) √ | 4) × | 5) √ | 6) × | 7) × | 8) √ | 9) × | 10) × |
| 11) × | 12) √ | 13) × | 14) × | 15) √ | 16) × | 17) × | 18) √ | 19) × | 20) × |

第四讲 金融机器学习的样本权重

4.1 确定样本权重的动机:

- ① 如果规定每一个特征结果都在下一个观察到的特征开始前或者开始时确定的, 会产生粗糙模型, 其特征采样频率将受到用于确定结果的范围的限制。
- ② 如果采用路径依赖型标签技术, 无论采取怎样的措施, 对结果范围进行调整消除重叠都不是一个好方法。
- ③ 基于以上两点, 观察并不是由 iid 过程产生的, 但大多数机器学习有关的研究文献都是在 iid 假设的基础上进行讨论的。因此, 许多机器学习算法的假设在金融时间序列下是不实际的。

4.2 样本重叠: 如果两个样本的因变量标签 y_i, y_j 包含一项或多项共同回报 $rt-1, t$ 。

4.3 样本独特: 样本 i 的因变量标签不与任何其他样本包含有共同回报 $rt-1, t$

4.4 重叠矩阵: $[1]T \times N$, 是一个二进制数组, 当且仅当 $[ti, 0, ti, 1]$ 与 $[t-1, t]$ 重合时该位置为 1, 否则为 0。

4.5 重叠度、独特度和平均独特度:

重叠度: 特定时点 t 上重叠的样本数量 $c_t = \sum_{i=1}^N 1_{t,i}$

独特度: 特定样本 i 在特定时点上的独特程度 $u_{t,i} = \frac{1_{t,i}}{c_t}$

平均独特度: 特定样本 i 的总体独特度评价 $\bar{u}_i = \frac{\sum_t u_{t,i}}{\sum_t 1_{t,i}}$

4.6 抽样重叠的原因: 假如一共有 N 个待抽样本, 有放回的抽取 I 次, 非重叠的样本最大数量为 K 。对于 K 个样本中的任意一个样本, 放回式抽取 I 次后出现 i 次的概率是

$$\mathbb{P}(i) = C_I^i \left(\frac{1}{K}\right)^i \left(1 - \frac{1}{K}\right)^{I-i}$$

$$I \rightarrow \infty, P(i) \rightarrow \frac{\left(\frac{I}{K}\right)^i e^{-\frac{I}{K}}}{i!}, \text{服从 } \lambda = \frac{I}{K} \text{ 的 Poisson 分布, 均值为 } I/K > 1$$

4.7 去重序贯抽样算法: 去重序贯抽样算法可以产生重叠度较低的抽样 (有重叠度, 但是重叠的可能性越来越小), 这使得去重序贯抽样的样本构成更加接近 i.i.d。

Algorithm 1: 去重序贯抽样算法 (De-overlapping Sequential Bootstrap; DSB)

Result: 产生重叠度较低的抽样, 尽量满足 i.i.d

确定重叠矩阵 $[1_{t,i}]_{T \times N}$, $t = 1 \cdots T, i = 1 \cdots N$, 按 $i \sim U[1, I]$ 抽出第一个样本 i_1 , 抽样序列 $\phi^{(1)} = \{i_1\}$;

while $k \leq I$ **do**

$$(1) \text{ 计算 } u_{t,i}^{(k)} = \frac{1_{t,i}}{1 + \sum_{j \in \phi^{(k-1)}} 1_{t,j}}, \bar{u}_i^{(k)} = \frac{\sum_t u_{t,i}^{(k)}}{\sum_t 1_{t,i}};$$

$$(2) \text{ 更新抽样的概率 } p_i^{(k)} = \frac{\bar{u}_i^{(k)}}{\sum_i \bar{u}_i^{(k)}};$$

$$(3) \text{ 抽取 } k \text{ 轮的样本 } i_k, \phi^{(k)} = \phi^{(k-1)} \cup \{i_k\}$$

end

4.8 去重序贯抽样算法的例子：

假设有一个标签集 $\{y_i\}_{i=1,2,3}$ ，其中标签 y_1 是回报 $r_{0,3}$ 的一个函数，标签 y_2 是回报 $r_{2,4}$ 的一个函数，标签 y_3 是回报 $r_{4,6}$ 的一个函数。结果的重叠以此指示矩阵 $\{1_{t,i}\}$ 表示：

$$\{1_{t,i}\} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

此过程以 $\varphi^{(0)} = \emptyset$ 开始，概率均匀分布， $\delta_i = \frac{1}{3}, \forall i = 1, 2, 3$ 。假设

我们随机从 $\{1, 2, 3\}$ 中抽取一个数字，且抽出的数字为 2。在从 $\{1, 2, 3\}$ 中抽取第二个数字之前（记住，引导程序抽取过程中允许重复），我们需要对概率进行调整。到目前为止，被抽取的观察集为 $\varphi^{(1)} = \{2\}$ 。第一个特征的平均唯一性为 $\bar{u}_1^{(2)} = \left(1 + 1 + \frac{1}{2}\right) \frac{1}{3} = \frac{5}{6} < 1$ ，第

二个特征的平均唯一性为 $\bar{u}_2^{(2)} = \left(\frac{1}{2} + \frac{1}{2}\right) \frac{1}{2} = \frac{1}{2} < 1$ 。第二次抽取的概率为

$\delta^{(2)} = \left\{\frac{5}{14}, \frac{3}{14}, \frac{6}{14}\right\}$ 。有两点值得一提：（1）被抽出的概率最低的是第

一个被抽出的特征，此种情况下的重叠（率）最高；（2）在 $\varphi^{(1)}$ 以外的两个可能抽出的特征中，被抽出的可能性最高的是 $\delta_3^{(2)}$ ，因为这个标签与 $\varphi^{(1)}$ 不存在重叠。假设第二个被抽出的数字是 3，我们将第三个同时也是最后一个被抽到的 $\delta^{(3)}$ 的概率的计算留在练习部分讨论。代码片

4.9 回报归因权重计算：如果对非重叠结果进行同等考虑，则高度重叠的结果将具有不成比例的权重。

而同时，与较大绝对回报相关联的标签则会比那些绝对回报较小的标签具有更加显著的重要性。

换言之，我们需要回报的绝对值较大因变量对应的样本权重较大，但重叠性较大时权重应该减小。

对于第 i 个样本，标签的时间跨度为 $[t_{i,0}, t_{i,1}]$ ，则样本权重应该为

$$\tilde{w}_i = \left| \sum_{t=t_{i,0}}^{t_{i,1}} \frac{r_{t-1,t}}{c_t} \right|$$

$$w_i = \frac{\tilde{w}_i I}{\sum_{j=1}^I \tilde{w}_j}$$

算法无论对初始标签还是元标签都适用。

4.Test 计算

1. 分别计算出各个时间点的重叠度，以及各个样本的平均独特度数。

$t \backslash i$	1	2	3	4	5
t_1	x				
t_2	x	x			
t_3	x	x	x		
t_4		x	x		
t_5			x		
t_6				x	
t_7				x	x
t_8				x	x

$t \backslash i$	1	2	3	4	5	6	7
t_1				x			
t_2			x	x			
t_3	x		x	x			
t_4	x		x	x		x	
t_5	x	x	x	x		x	
t_6		x				x	x
t_7		x					x
t_8		x			x		
t_9					x		

2. 写出去重序贯抽样算法的过程，假定抽 3 次，最多有 2 个不同元素，抽取顺序为 {2,3}

$t \backslash i$	1	2	3
1	x		
2	x		
3	x	x	
4		x	
5			x
6			x

3.计算回报归因权重（请将权重缩放至 I=2，结果保留 4 位小数）:

$t \backslash i$	1	2	3
1	2.5%		
2	2%		
3	3%	-0.5%	3%
4		-2%	1%
5			0.5%
6			-1%

4. Answer 计算

1.

	$u_{t,i}$				
c_t	1	2	3	4	5
1	1	0	0	0	0
2	0.5	0.5	0	0	0
3	0.33	0.33	0.33	0	0
2	0	0.5	0.5	0	0
1	0	0	1	0	0
1	0	0	0	1	0
2	0	0	0	0.5	0.5
2	0	0	0	0.5	0.5
	0.61	0.44	0.61	0.66	0.5

	$u_{t,i}$						
c_t	1	2	3	4	5	6	7
1	0	0	0	1	0	0	0
2	0	0	1/2	1/2	0	0	0
3	1/3	0	1/3	1/3	0	0	0
4	1/4	0	1/4	1/4	0	1/4	0
5	1/5	1/5	1/5	1/5	0	1/5	0
3	0	1/3	0	0	0	1/3	1/3
2	0	1/2	0	0	0	0	1/2
2	0	1/2	0	0	1/2	0	0
1	0	0	0	0	1/2	0	0
	$\frac{47}{180}$	$\frac{23}{60}$	$\frac{77}{240}$	$\frac{137}{360}$	$\frac{1}{2}$	$\frac{47}{180}$	$\frac{5}{12}$

2.

Figure 1 illustrates the proposed algorithm through three stages of a 3x3 grid state, labeled $u_{t,i}^{(1)}$, $u_{t,i}^{(2)}$, and $u_{t,i}^{(3)}$.

Stage 1: $u_{t,i}^{(1)}$

	1	2	3
1	1	0	0
1	1	0	0
1	1	1	0
1	0	1	0
1	0	0	1
	1	1	1
	1/3	1/3	1/3

Below the grid, the label $\bar{u}_i^{(1)}, p_i^{(1)}$ is shown.

Stage 2: $u_{t,i}^{(2)}$

	1	2	3
1	1	0	0
1	1	0	0
2	0.5	0.5	0
2	0	0.5	0
1	0	0	1
1	0	0	1
	5/6	0.5	1
	5/14	3/14	6/14

Below the grid, the label $\bar{u}_i^{(1)}, p_i^{(1)}$ is shown. The label $\phi^{(1)} = \{2\}$ is shown below the grid.

Stage 3: $u_{t,i}^{(3)}$

	1	2	3
1	1	0	0
1	1	0	0
2	0.5	0.5	0
2	0	0.5	0
2	0	0	0.5
2	0	0	0.5
	5/6	0.5	0.5
	10/22	6/22	6/22

Below the grid, the label $\bar{u}_i^{(1)}, p_i^{(1)}$ is shown. The label $\phi^{(2)} = \{2, 3\}$ is shown below the grid.

3. $w_1=2.0625, w_2=-0.4375, w_3=0.3750$

第五讲 金融机器学习的分数差分处理

5.1 进行分数差分的动机：

- ① 金融序列数据通常信噪比很低；
- ② 金融预测同时依赖于信号的时序记忆性和平稳性；
- ③ 平稳性对于预测的意义：监督学习算法要求 y 标签的变量是平稳的，否则无法将未知新观测对应到历史已知观测；
- ④ 整数差分造成了过度差分，虽然保证了平稳性，但进一步降低了时序记忆性，信噪比更低；
- ⑤ 缺乏时序记忆性时，使用再复杂的统计技术都无法完成预测，只能贡献错误发现；
- ⑥ 分数差分能够同时兼顾记忆性和平稳性。

5.2 整数差分与 F-差分的原理：

F-差分 (Fractional Differentiation) 是整数差分的自然推广。

■ 滞后算子 \mathcal{L} 作用与序列 X_t , $\mathcal{L}X_t = X_{t-1}$, $\mathcal{L}^k X_t = X_{t-k}$

■ $(1 - \mathcal{L})^2 X_t = X_t - 2X_{t-1} + X_{t-2}$

■ $(1 - \mathcal{L})^{-1} = 1 + \mathcal{L} + \mathcal{L}^2 + \mathcal{L}^3 + \dots$

■ $X_t = X_{t-1} + u_t$, $X_t - \mathcal{L}X_t = u_t$, $X_t = (1 - \mathcal{L})^{-1}u_t$

举例：为了让计量序列平稳，我们往往处理的是收益率而非价格序列。这里的收益率 $(1 + r_t)$ 可以看作一个价格序列的滞后算子

$$P_t = (1 + r_t)P_{t-1}$$

这样进行差分的序列会只保留一个增量信息，去掉了之前若干时间积累的存量信息 P_t 。这样的信息处理不总是对的（如 A 股散户有持有低价股票的倾向），因为丢失的信息不总是无用的。

■ d 为任意整数, $(1 + \mathcal{L})^d = \sum_{k=0}^d C_d^k \mathcal{L}^k 1^{d-k} = \sum_{k=0}^{\infty} C_d^k \mathcal{L}^k 1^{d-k} = \sum_{k=0}^{\infty} C_d^k \mathcal{L}^k$

■ 将差分算子式展开

$$\begin{aligned}(1 - \mathcal{L})^d &= \sum_k C_d^k (-\mathcal{L})^k = \sum_k \frac{\prod_{i=0}^{k-1} (d-i)}{k!} (-\mathcal{L})^k \\&= \sum_k (-\mathcal{L})^k \prod_{i=0}^{k-1} \frac{d-i}{k-i} \\&= 1 - d\mathcal{L} + \frac{d(d-1)}{2!} \mathcal{L}^2 - \frac{d(d-1)(d-2)}{3!} \mathcal{L}^3 + \dots\end{aligned}$$

■ 将整数 d 拓展为任意实数，差分算子同样表示为

$$(1 - \mathcal{L})^d = 1 - d\mathcal{L} + \frac{d(d-1)}{2!} \mathcal{L}^2 - \frac{d(d-1)(d-2)}{3!} \mathcal{L}^3 + \dots$$

■ 因此，选择 d 为非整数，F-差分将保留序列的记忆性

$$\tilde{X}_t = \sum_{k=0}^{\infty} \omega_k X_{t-k}$$

其中, $\omega = \left\{ 1, -d\mathcal{L}, \frac{d(d-1)}{2!} \mathcal{L}^2, -\frac{d(d-1)(d-2)}{3!} \mathcal{L}^3, \dots, (-1)^k \prod_{i=0}^{k-1} \frac{d-i}{k-i} \dots \right\}$

$$\omega_k = -\omega_{k-1} \frac{d-k+1}{k}$$

这里指的是两点：1. 分数阶差分 ($d < 1$) 比整数阶差分 ($d=1, 2, 3, \dots, N$) 丢失信息更少；2. 差分算子式有很好的数学性质，使得它可以拓展 d 为有理数的时候而没有数学上的谬误。

思考： $d=1$ 、 $d=1/3$ 、 $d=4/3$ 三种情况，哪种能够保存序列最多的信息？

5.3 固定窗口 F-差分的原理:

- 固定窗口 F-差分 (Fixed Window Fractional Difference, FFD): 序列 X_t 长度有限, ω_k 值有限, 选取一个固定的回溯窗口 $[t - l^*, t]$ 中的数据来计算 F-差分
- 相当于选取了一个阈值 τ , 其中 $|\omega_{l^*}| \geq \tau$, 但 $|\omega_{l^*+1}| \leq \tau$,
- 计算方法

$$\tilde{X}_t = \sum_{k=0}^{l^*} \tilde{\omega}_k X_{t-k}, \quad t = T - l^* + 1, \dots, T$$

选取了一个阈值的话, 考虑 $|\omega_k|$ 是单调减序列, 这样这个级数才能收敛。

固定窗口 F-差分的结果:

- ① 获得平稳的时间序列 X_t ;
- ② 分布可能不再是正态分布;
- ③ 分布可能具有一定的偏度和峰度;
- ④ 选取合适的 d 值, 就可以得到平稳序列。

5.4 平稳性与记忆性平衡:

- 考虑一个原始序列 $\{X_t\}_{t=1, \dots, T}$, 在运用 FFD 时, 可优化选取一个最小的 d^* 使得 F-差分得到的序列 $\{X_t\}_{t=1, \dots, T}$ 通过 ADF 检验
- ADF 检验: H_0 原始序列存在单位根非平稳, H_1 原始序列平稳
- 可选取显著性水平 $\alpha = 0.05$ (95% 的置信度) 来进行 ADF 检验

5. Test 判断

- 1) 金融序列数据通常信噪比很高。
- 2) 金融预测仅依赖于信号的时序记忆性。
- 3) 监督学习算法要求 y 标签是时序记忆性的, 否则无法将未知新观测对应到历史已知观测
- 4) 整数差分造成了过度差分, 降低了平稳性。
- 5) 想要完成预测必须要有充足的时序记忆性。
- 6) 分数差分能够同时兼顾记忆性和平稳性。
- 7) F 差分是整数差分的自然推广。
- 8) 可以借用 FFD 中固定窗口的概念来使得整数差分保持记忆性。
- 9) FFD 中序列 X_t 长度无限, 但是权重 w_t 长度有限。
- 10) FFD 结果必是无偏无峰的正态分布。
- 11) 通过选取合适的差分参数 d 值, 可以使得 FFD 结果具有平稳性。
- 12) 可以通过单位根检验 ADF 来验证平稳性, 其中原假设 H_0 为原始序列存在单位根非平稳。

5 Answer 判断

- 1) \times 2) \times 3) \times 4) \times 5) \checkmark 6) \checkmark 7) \checkmark 8) \times 9) \times 10) \times 11) \checkmark 12) \checkmark

5. Test 简答

【思考】 $X_t = \rho X_{t-1} + u_t$, $|\rho| < 1$ 如何表示为 u_t 滞后项的和?

5. Answer 简答

$$X_t - \rho X_{t-1} = u_t \quad \Rightarrow \quad X_t - \rho L X_{t-1} = u_t$$

令 $L' = \rho L$, 问题回归原型, 结果为 $X_t = (1 - \rho L)^{-1} u_t$ (前提: $|\rho L| < 1$)

第六讲 金融机器学习的模型集成

6.1 偏差-方差分解:

这里林老师的 PPT 没有解析, 给出西瓜书的推导过程:

对学习算法除了通过实验估计其泛化性能, 人们往往还希望了解它“为什么”具有这样的性能. “偏差-方差分解” (bias-variance decomposition) 是解释学习算法泛化性能的一种重要工具.

偏差-方差分解试图对学习算法的期望泛化错误率进行拆解. 我们知道, 算法在不同训练集上学得的结果很可能不同, 即便这些训练集是来自同一个分布. 对测试样本 \mathbf{x} , 令 y_D 为 \mathbf{x} 在数据集中的标记, y 为 \mathbf{x} 的真实标记, $f(\mathbf{x}; D)$ 为训练集 D 上学得模型 f 在 \mathbf{x} 上的预测输出. 以回归任务为例, 学习算法的期望预测为

$$\bar{f}(\mathbf{x}) = \mathbb{E}_D[f(\mathbf{x}; D)], \quad (2.37)$$

使用样本数不同的不同训练集产生的方差为

$$\text{var}(\mathbf{x}) = \mathbb{E}_D \left[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2 \right], \quad (2.38)$$

噪声为

$$\varepsilon^2 = \mathbb{E}_D \left[(y_D - y)^2 \right]. \quad (2.39)$$

期望输出与真实标记的差别称为偏差(bias), 即

$$\text{bias}^2(\mathbf{x}) = (\bar{f}(\mathbf{x}) - y)^2. \quad (2.40)$$

为便于讨论, 假定噪声期望为零, 即 $\mathbb{E}_D[y_D - y] = 0$. 通过简单的多项式展开合并, 可对算法的期望泛化误差进行分解:

$$\begin{aligned} E(f; D) &= \mathbb{E}_D \left[(f(\mathbf{x}; D) - y_D)^2 \right] \\ &= \mathbb{E}_D \left[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}) + \bar{f}(\mathbf{x}) - y_D)^2 \right] \\ &= \mathbb{E}_D \left[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2 \right] + \mathbb{E}_D \left[(\bar{f}(\mathbf{x}) - y_D)^2 \right] \\ &\quad + \mathbb{E}_D \left[2(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))(\bar{f}(\mathbf{x}) - y_D) \right] \\ &= \mathbb{E}_D \left[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2 \right] + \mathbb{E}_D \left[(\bar{f}(\mathbf{x}) - y_D)^2 \right] \\ &= \mathbb{E}_D \left[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2 \right] + \mathbb{E}_D \left[(\bar{f}(\mathbf{x}) - y + y - y_D)^2 \right] \\ &= \mathbb{E}_D \left[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2 \right] + \mathbb{E}_D \left[(\bar{f}(\mathbf{x}) - y)^2 \right] + \mathbb{E}_D \left[(y - y_D)^2 \right] \\ &\quad + 2\mathbb{E}_D \left[(\bar{f}(\mathbf{x}) - y)(y - y_D) \right] \\ &= \mathbb{E}_D \left[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2 \right] + (\bar{f}(\mathbf{x}) - y)^2 + \mathbb{E}_D \left[(y_D - y)^2 \right], \end{aligned} \quad (2.41)$$

于是,

$$E(f; D) = \text{bias}^2(\mathbf{x}) + \text{var}(\mathbf{x}) + \varepsilon^2, \quad (2.42)$$

也就是说, 泛化误差可分解为偏差、方差与噪声之和.

如果你觉得西瓜书的推导难以看懂，下面是南瓜书的解析：

[解析]：第 1-2 步：减一个 $\bar{f}(x)$ 再加一个 $\bar{f}(x)$ ，属于简单的恒等变形；

第 2-3 步：首先将中括号里面的式子展开

$$\mathbb{E}_D \left[(f(x; D) - \bar{f}(x))^2 + (\bar{f}(x) - y_D)^2 + 2(f(x; D) - \bar{f}(x))(\bar{f}(x) - y_D) \right]$$

然后根据期望的运算性质： $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$ 可将上式化为

$$\mathbb{E}_D \left[(f(x; D) - \bar{f}(x))^2 \right] + \mathbb{E}_D \left[(\bar{f}(x) - y_D)^2 \right] + \mathbb{E}_D \left[2(f(x; D) - \bar{f}(x))(\bar{f}(x) - y_D) \right]$$

第 3-4 步：再次利用期望的运算性质将第 3 步得到的式子的最后一项展开

$$\mathbb{E}_D \left[2(f(x; D) - \bar{f}(x))(\bar{f}(x) - y_D) \right] = \mathbb{E}_D \left[2(f(x; D) - \bar{f}(x)) \cdot \bar{f}(x) \right] - \mathbb{E}_D \left[2(f(x; D) - \bar{f}(x)) \cdot y_D \right]$$

首先计算展开后得到的第一项

$$\mathbb{E}_D \left[2(f(x; D) - \bar{f}(x)) \cdot \bar{f}(x) \right] = \mathbb{E}_D \left[2f(x; D) \cdot \bar{f}(x) - 2\bar{f}(x) \cdot \bar{f}(x) \right]$$

由于 $\bar{f}(x)$ 是常量，所以由期望的运算性质： $\mathbb{E}[AX + B] = A\mathbb{E}[X] + B$ （其中 A, B 均为常量）可得

$$\mathbb{E}_D \left[2(f(x; D) - \bar{f}(x)) \cdot \bar{f}(x) \right] = 2\bar{f}(x) \cdot \mathbb{E}_D[f(x; D)] - 2\bar{f}(x) \cdot \bar{f}(x)$$

由公式 (2.37) 可知： $\mathbb{E}_D[f(x; D)] = \bar{f}(x)$ ，所以

$$\mathbb{E}_D \left[2(f(x; D) - \bar{f}(x)) \cdot \bar{f}(x) \right] = 2\bar{f}(x) \cdot \bar{f}(x) - 2\bar{f}(x) \cdot \bar{f}(x) = 0$$

接着计算展开后得到的第二项

$$\mathbb{E}_D \left[2(f(x; D) - \bar{f}(x)) \cdot y_D \right] = 2\mathbb{E}_D[f(x; D) \cdot y_D] - 2\bar{f}(x) \cdot \mathbb{E}_D[y_D]$$

由于噪声和 f 无关，所以 $f(x; D)$ 和 y_D 是两个相互独立的随机变量，所以根据期望的运算性质： $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$ （其中 X 和 Y 为相互独立的随机变量）可得

$$\begin{aligned} \mathbb{E}_D \left[2(f(x; D) - \bar{f}(x)) \cdot y_D \right] &= 2\mathbb{E}_D[f(x; D) \cdot y_D] - 2\bar{f}(x) \cdot \mathbb{E}_D[y_D] \\ &= 2\mathbb{E}_D[f(x; D)] \cdot \mathbb{E}_D[y_D] - 2\bar{f}(x) \cdot \mathbb{E}_D[y_D] \\ &= 2\bar{f}(x) \cdot \mathbb{E}_D[y_D] - 2\bar{f}(x) \cdot \mathbb{E}_D[y_D] \\ &= 0 \end{aligned}$$

所以

$$\begin{aligned} \mathbb{E}_D \left[2(f(x; D) - \bar{f}(x))(\bar{f}(x) - y_D) \right] &= \mathbb{E}_D \left[2(f(x; D) - \bar{f}(x)) \cdot \bar{f}(x) \right] - \mathbb{E}_D \left[2(f(x; D) - \bar{f}(x)) \cdot y_D \right] \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

第 4-5 步：同第 1-2 步一样，减一个 y 再加一个 y ，属于简单的恒等变形；

第 5-6 步：同第 2-3 步一样，将最后一项利用期望的运算性质进行展开；

第 6-7 步：因为 $\bar{f}(x)$ 和 y 均为常量，所以根据期望的运算性质可知，第 6 步中的第 2 项可化为

$$\mathbb{E}_D \left[(\bar{f}(x) - y)^2 \right] = (\bar{f}(x) - y)^2$$

同理，第 6 步中的最后一项可化为

$$2\mathbb{E}_D \left[(\bar{f}(x) - y)(y - y_D) \right] = 2(\bar{f}(x) - y) \mathbb{E}_D[(y - y_D)]$$

由于此时假设噪声的期望为零，也即 $\mathbb{E}_D[(y - y_D)] = 0$ ，所以

$$2\mathbb{E}_D \left[(\bar{f}(x) - y)(y - y_D) \right] = 2(\bar{f}(x) - y) \cdot 0 = 0$$

偏差：度量拟合能力，不现实的假设导致此值变大。机器学习算法未能识别特征和结果之间的重要关系时贡献偏差。

方差：表示模型对数据的敏感性。算法错误地将噪声误认为信号，而非建模训练集中的一般模式时贡献方差。

噪声：表示学习的难度。这是无法由任何模型解释的不可减少的。

模型集成视角：进行模型集成可以有效的减小偏差或者方差。

6.2 装袋式集成:

装袋式集成 (Bagging) 的原理如下:

- 通过有放回的随机抽样生成 N 个训练数据集。
- 使用 N 个估计器, 分别从一个训练集中拟合一个估计器。这些估计器是相互独立的, 因此可以并行拟合模型。
- 模型集成:
 - 连续因变量: 从 N 个模型中得到的单个预测进行简单平均。
 - 分类因变量: 由对这个观察值进行分类为该类别的估计器数量所占比例 (投票数) 确定, 也可以计算平均概率值

装袋式集成可以降低方差:

装袋式集成的最大优势就是降低模型的方差, 从而缓解过拟合 $\varphi_i[c]$: 第 i 个集成模型的预测, 各预测间的平均关联为 $\bar{\rho}$

$$\begin{aligned} V\left[\frac{1}{N} \sum_{i=1}^N \varphi_i[c]\right] &= \frac{1}{N^2} \sum_{i=1}^N \left(\sum_{j=1}^N \sigma_{i,j}\right) = \frac{1}{N^2} \sum_{i=1}^N \left(\sigma_i^2 + \sum_{j \neq i}^N \sigma_i \sigma_j \rho_{i,j}\right) \\ &= \frac{1}{N^2} \sum_{i=1}^N (\bar{\sigma}^2 + \underbrace{\sum_{j \neq i}^N \bar{\sigma}^2 \bar{\rho}}_{=(N-1)\bar{\sigma}^2 \bar{\rho} \text{ for a fixed } i}) = \frac{\bar{\sigma}^2 + (N-1)\bar{\sigma}^2 \bar{\rho}}{N} = \bar{\sigma}^2 \left(\bar{\rho} + \frac{1-\bar{\rho}}{N}\right) \end{aligned}$$

$$\bar{\sigma}^2 = \frac{1}{N} \sum_{i=1}^N \sigma_i^2, \bar{\rho} = \frac{\sum_{i \neq j}^N \sigma_i \sigma_j \rho_{i,j}}{\bar{\sigma}^2 N(N-1)}, \text{Var}\left[\frac{1}{N} \sum_{i=1}^N \varphi_i[c]\right] = \bar{\sigma}^2 \left(\bar{\rho} + \frac{1-\bar{\rho}}{N}\right)$$

当 $\bar{\rho} < 1$ 可以降低方差

装袋式集成可以降低偏差:

对于分类模型, 偏差相当于准确率数学期望。设单个模型的准确率为 p , 若有 k 个类别, 按照少数服从多数原则, 分类正确的必要条件为

$$\mathbb{P}\left(\text{Vote}_i > \frac{N}{k}\right) = 1 - \mathbb{P}\left(\text{Vote}_i \leq \frac{N}{k}\right) = 1 - \sum_{i=0}^{\lfloor N/k \rfloor} C_N p^i (1-p)^{N-i}$$

- $N > \frac{p}{(p - \frac{1}{k})^2}$, 则如果 $p > \frac{1}{k}$, $\mathbb{P}\left(\text{Vote}_i > \frac{N}{k}\right) > p$
- 如果 $p < \frac{1}{k}$, 无论 N 多大, $\mathbb{P}\left(\text{Vote}_i > \frac{N}{k}\right) < p$ 弱分类器无法变强

这里需要尤其注意的是, 弱分类器无法变强, 当 $p < \frac{1}{k}$ 时, 无论如何增加分类器的数量结果也不会变好。“三个臭皮匠, 比单个臭皮匠更臭”。

6.3 装袋式集成的样本重叠问题: 进行模型集成可以有效的减小偏差或者方差。

①可放回的随机抽样可能会产生相同的样本, 这使得相关性接近 1, 导致无法削减方差。这个问题可以通过去重顺序贯抽样算法来缓解。

②训练集样本中有放回地随机采样与袋外采样非常相似, 在这种情况下可能造成准确性被高估。这可以通过不打乱顺序的分层 K 折交叉验证缓解, 使用时优先选择较低的 K 值。

6.4 随机森林——近似装袋式集成的决策树：

随机森林与袋装法有一定的共性，体现在从自举的数据子集中对每一个估计器进行单独训练。

随机森林与袋装法的最关键区别在于，随机森林包含了第二层随机过程：在考虑每个分支节点的优化时，仅评估随机抽取（无放回）的子样本属性，其目的是进一步减少估计器之间的相关性。

随机森林的优点：

- ①与袋装法类似，随机森林可以在没有过拟合的情况下降低预测方差；
- ②随机森林可以评估特征重要性；
- ③随机森林提供了袋外准确性的估算（但在金融领域很可能被夸大）；
- ④与袋装法类似，随机森林不一定会比单一决策树表现出更低的偏差。

6.5 正确使用随机森林：

应对金融机器学习中的样本的重叠性问题，RF 的参数需要合理选择

- 将最大特征数 (Python: `max_features`, R: `mtry`) 设为较低的值，强制树之间产生差异
- 将叶节点的最小样本数 (Python: `min_weight_fraction_leaf`, R: `min.node.size`) 设为一个较大的值，形成 Early stop
- 将每个模型的样本量设定为样本间的平均独特度与总样本量的乘积： $\text{avg}U \times N$ 。
- 使用 DSB 进行抽样

6.6 助推式集成算法：

助推式集成的思路是逐步通过弱学习器生成偏差小的强学习器。步骤如下

- (1) 初始化为均匀权重，使用随机抽样生成一个训练集
- (2) 使用该训练集拟合一个学习器；
- (3) 如果单个学习器的准确率大于接受阈值 (例如在二元分类器中为 50%，即比随机分类好)，则保留该学习器，否则丢弃；
- (4) 给误分类的样本更多的权重，给正确分类的样本更少的权重；
- (5) 重复前面的步骤直到生成 N 个学习器；
- (6) 合成的预测值是 N 个模型的个体预测值的加权平均，其中权值由个体学习器的准确性确定

助推式集成算法 (Boosting) 和袋装式继承算法 (Bagging) 的比较：

- ① 助推式串行计算，无法并行；
- ② 很弱的分类器将被放弃；
- ③ 每轮迭代时样本的权重都不相同；
- ④ 每个学习器都有一个不同的权重；
- ⑤ 主要用于解决欠拟合问题。金融应用中主要面临过拟合问题，应以装袋式集成为主。

6.Test 判断

- 1)进行模型集成可以使得假设更接近于真实规律，但不能减小算法对于数据集的敏感度。
- 2)模型集成通过将性质截然不同的弱学习模型组合起来，取长补短，从而获得一个强学习模型。
- 3)装袋集成中 N 个训练集的生成需要采用不放回的方式，否则会出现重叠度过高的问题。
- 4)装袋式集成相比于助推式集成的一大优点就是可以并行学习。
- 5)装袋式集成形成的 N 个估计器中，即使样本各个训练集之间样本存在重叠，但估计器必定是相互独立的。
- 6)对于分类模型，偏差相当于准确率的数学期望。
- 7)在装袋式集成中，若有 k 个类别，单模型准确率为 p，那么分类正确的必要条件是 $P(\text{Vote}_i > \frac{N}{k})$
- 8)装袋式集成将足够多的弱分类器组合在一起，最终必定能得到一个更强的分类器。
- 9)装袋式集成中，当各预测之间的平均关联 $\rho = 1$ 时，
- 10)装袋式集成中，袋内袋外分别对应于训练集和测试集。
- 11)助推式集成的思路是通过弱学习器生成方差小的强学习器。
- 12)AdaBoost 是装袋式集成算法。
- 13)助推式集成中每次给正确分类的样本分配更多权重，使得算法能够“加深印象”
- 14)助推式集成只能降低偏差。

6.Answer 判断

- | | | | | | | | | | |
|-------|-------|-------|-------|------|------|------|------|------|-------|
| 1) × | 2) × | 3) × | 4) √ | 5) √ | 6) √ | 7) √ | 8) × | 9) × | 10) √ |
| 11) × | 12) × | 13) × | 14) × | | | | | | |

第七讲 金融机器学习的交叉验证

7.1 交叉验证：

交叉验证的目的在于确定机器学习算法的泛化误差，以防止过拟合。然而，交叉验证是标准机器学习技术应用到金融问题上失败的又一例证。交叉验证会通过超参数调优来促成过拟合。

基本思想：将数据分割成训练集和测试集，训练集和测试集不相交。使用训练集来训练模型，使用测试集来评估模型的性能。通过优化模型超参数来选择最优模型。

关键假设：预测数据满足 i.i.d，因此测试集与训练集仅仅的差异在于噪声大小，具有相同的结构规律。

在交叉验证中调超参，导致进行多次交叉验证，但引入了**多重测试偏误**，很难反应泛化误差。

多重测试偏误是指在进行多次假设检验或模型选择时，由于进行了多次比较，从而使得显著性水平增加，从而可能导致错误地拒绝原假设或选择错误的模型或超参数。例如，考虑使用多重交叉测试来选择最佳分类器或最优超参数。如果我们比较多个分类器或超参数设置，并且根据它们在不同测试集上的性能进行排名，则可能会选择一个在所有测试集上具有最高性能的分类器或超参数设置。然而，由于我们进行了多次比较，可能会错误地选择具有过拟合性能的分类器或超参数设置。

因此标准交叉验证采用重抽样交叉验证法：[训练集+验证集]*多次+测试集。

7.2K 折交叉验证 (K-fold CV)：是一种用于评估机器学习模型性能的方法。它将数据集分成 K 个大小相等的子集，其中一个子集被保留作为测试集，其他 K-1 个子集用作训练集。这个过程会重复 K 次，每次选择不同的子集作为测试集，最后将 K 次评估结果的平均值作为模型的性能指标。

留一 K 折交叉验证 (LOOCV)：是一种 K 折交叉验证的特例，在训练集中取 K=N 的时候，每次只留下一个样本作为验证集，其他的全部作为训练集。LOOCV 一共执行 N 次 CV (N 为训练集+测试集样本总数)

LOOCV 的优点：首先它不受训练集和验证集划分(这里指将大训练集划分为训练集和验证集)的影响，因为每一个样本都单独的做过验证集，同时，其用了 N-1 个样本训练模型，几乎用到了所有样本信息。**LOOCV 缺点：**计算量过大。

重复 K 折交叉验证：重复的进行 N 次 K 折交叉验证。每次将建模样本都分为 K 个尺度相当的子集。重复 K 折交叉验证一共进行 NK 次 CV，这比 LOOCV 更耗费算力。

7.3 蒙特卡洛交叉验证：重复的对建模样本进行 B 次训练集/验证集劈分，劈分时采用固定的比例划分建模样本。

7.4 自助交叉验证：有放回抽取 B 个与建模数据等长的样本作为训练集，未抽到的样本进入验证集(袋外)。BCV 衡量的平均模型方差要低于 K 折 CV，但平均有 63.2 % 的样本会出现一次，模型偏差与 2 折交叉验证相似。

63.2%是怎么来的：63.2%约等于 $1-1/e$ 。

7.5 标准 CV 法失效的原因：(1) 重抽样 CV 法仍然不能完全缓解多重测试偏误；(2) 因为样本重叠，预测数据不再是 i.i.d.

■ 重抽样 CV 面对非 i.i.d 的后果：

- 信息泄露 (leakage)：训练集的信息泄露到了验证集中。
- 假设 t 在训练集中， $t+1$ 在验证集中， t 和 $t+1$ 对应的因变量标签来此重叠数据，因此
$$Y_{t+1} \approx Y_t。$$

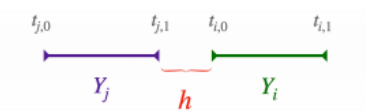
- 由于序列相关性 $X_{t+1} \approx X_t$
- 此时，及时 X 是预测能力很弱的变量，仍然有 $\mathbb{E}(Y_{t+1} | X_{t+1}) \approx \mathbb{E}(Y_t | X_t)$ ，CV 的评价被高估

可以用以下方法弥补传统 CV 法的缺陷：

- 净化法（数据层面）： Y_j 属于验证集， Y_i 属于训练集，且 Y_i 与 Y_j 有时间重叠，则在训练时清除 Y_i
- 避免容易过拟合的分类器（算法层面）：从源头上直接避免模型过拟合，比如采用装袋式集成
 - 注意设置 Early stop 机制
 - 注意避免训练时的样本重叠，采用 DSB
 - 注意采用 AvgU×N 生成单个分类器，保证分类器的多样性

7.6 净化-隔离 CV 法：

- 净化 CV 法 (Purged-CV)：将会与验证集内的数据产生重叠的训练集内的那部分数据清除（仅清除训练集中的数据）
- 验证集中的标签 Y_j , $Y_j = f([t_{j,0}, t_{j,1}])$
- 训练集中的标签 Y_i , $Y_i = f([t_{i,0}, t_{i,1}])$
- 如果 $[t_{i,0}, t_{i,1}]$ 与 $[t_{j,0}, t_{j,1}]$ 产生重叠，则在训练时清除 $[t_{i,0}, t_{i,1}]$
- 隔离机制 (Embargo)：在劈分数据集时，因为时间序列具有序列相关的性质，应该将验证集切分处之后一段时间内的数据也从训练集中删除。因为 h 不充分大，仍有可能 $Y_j \approx Y_i$



- h 可取 $0.01T$
- 附加了隔离机制的净化 CV 法称为净化-隔离 CV 法

7.Test 判断

- 1)CV 法能评估和控制欠拟合。
- 2)自主交叉验证衡量的模型偏差与二折交叉验证类似。
- 3)自主交叉验证衡量的模型方差要高于 k 折交叉验证。
- 4)对于特征而言, x_i 和 x_j 有时间重叠, 会造成信息泄露。
- 5)在净化 CV 法中, 需要将验证集和训练集中重叠的数据都去掉。
- 6)交叉验证是用于评估模型泛化能力的一种技术。
- 7)在交叉验证过程中, 数据被分成多个大小相同的部分, 并且每一部分都会被用作测试集一次。
- 8)交叉验证可以完全防止模型过拟合的问题发生。
- 9)留一交叉验证 (LOOCV) 在处理非常大的数据集时是效率最高的交叉验证方法。
- 10)K 折交叉验证 (K-fold CV) 中的 K 代表了数据将被分成多少份。
- 11)在 K 折交叉验证中, 每一份数据只会被用一次作为测试集, 其余的 K-1 份数据则作为训练集。
- 12)如果数据集中的数据点是时间序列相关的, 那么采用简单的随机划分为训练集和测试集的方法不会有任何问题。

7.Test 简答

- 1.假定训练集和验证集共有 N 个样本, 回答下列交叉验证共需要执行多少次:
(1)K 折交叉验证; (2)留一 K 折交叉验证; (3)重复 P 次的 K 折交叉验证。
- 2.试证明: BCV 衡量的平均模型平均有 63.2 % 的样本会出现一次。
- 3.对于特征而言, X_i 和 X_j 有时间重叠, 是否会造成信息泄露?

7.Answer 判断

1) × 2)√ 3)× 4) × 5)× 6)√ 7)√ 8)× 9)× 10)√ 11)√ 12)×

7.Answer 简答

- 1.(1)K; (2)N; (3)PK;

2.

由于每一个bootstrap训练集是有放回抽样, 对于每一个观测来说, 当样本量很大的时候, 它不被抽中的概率为 $\lim_{n \rightarrow \infty} (1 - \frac{1}{n})^n = \frac{1}{e}$ 。那么对于每一个bootstrap训练集, 大约有 $\frac{1}{e}$ 的原始数据不在其中。

3.

不一定, 只要 Y_i 和 Y_j 之间相互独立 (就不属于信息泄露)。若发生泄露, 则必须满足 $(X_i, Y_i) \approx (X_j, Y_j)$, 而单独满足 $X_i \approx X_j$ 或 $Y_i \approx Y_j$ 是不够的。

第八讲 金融机器学习回测算法

8.1 回测的目的：

回测的目标是通过金融机器学习模型生成的投资策略在过去的表现来推断策略在未来的表现。因此，回测要非常完整的评估在特定场景下各种变量的效果，包括投资规模、投资周期、成本变化等。

需要特别注意的是，回测是一个假设，而绝不是一个实验，不能证明任何东西也不能保证得到任何东西。

8.2 回测的常见陷阱：量化投资七宗罪

幸存者偏差：回测数据仅包含当前活跃资产，忽略了随时间推移由于破产、摘牌或被并购的资产；

前视偏差：使用了在历史上看还尚未公开的信息进行决策；

事后诸葛亮：事后错误的寻找因果关系来证实随机（不自知）的模式；

数据窥探：在测试集上训练模型；

交易成本：设定不切实际的交易成本进行模拟；

异常值控制：对极端情况情况笠选或裁剪；

做空：做空的可行性不进行正确的评估。

8.3 回测的常见陷阱：马尔科斯回测定律

马尔科斯回测第一定律：回测不是研究工具，特征重要性才是。特征重要性有助于人们理解机器学习算法获得的模式的本质，但并不涉及如何使用它们盈利。回测是基于研究结果的基础上评价盈利的可能性。

马尔科斯回测第二定律：回测时研究就像开车时喝酒，不要在回测中去训练模型。回测的意义是剔除不好的模型，而不是去通过不断回测改进他们。通过回测去调模型会产生选择偏差，从而浪费时间。

在所有的研究流程结束后才能进行回测，回测结果不好必须重新开始研究。

8.4 回测的常见陷阱：选择偏差风险

选择性偏差：通过反复回测，调整模型参数，最后筛选出一个策略表现最好的回测结果。

风险 1：无法保证最好的回测结果不是来自于运气；

风险 2：无法保证是否模型足够复杂以至于总有一套参数可以完美的拟合历史数据（拟合了噪声）

回测过拟合：选择性偏差造成的效应。

8.5 对选择偏差的评估算法：CSCV 算法

组合对称交叉验证：通过交叉验证方法来评估回测过拟合的概率，该算法仅适用于评估模型的选择性偏差的风险，并不是完整的回测。

该算法与开发模型时评估使用的 CV 不同，在开发完模型之后再使用。

该算法需要对模型的所有超参数组合进行总体评估。

对称划分训练集和验证集：保证了样本内和样本外数据的平衡性。

对称划分：样本内和样本外数据一样多。

Algorithm 1: 组合对称交叉验证算法 CSCV

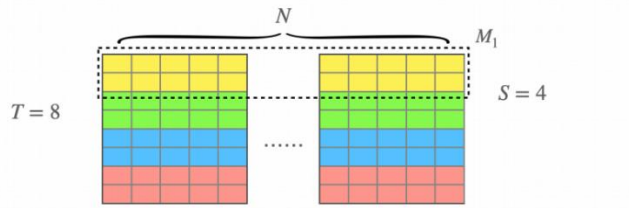
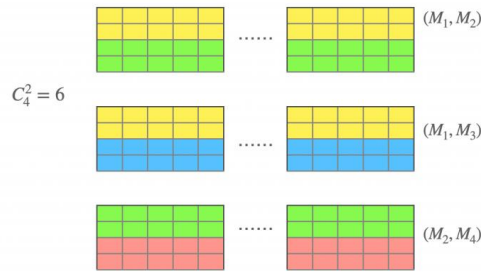
Result: 产生回测过拟合概率 PBOstep1: 确定性能矩阵 $M_{T \times N}$: N 为可选模型数量 (超参组合总量);step2: 将性能矩阵 M 按照行分割为 S 个子矩阵 $M_s, s = 1, 2, \dots, S$;step3: 构造 M_s 的可能组合, 每组的大小为 $S/2$, 一共有 $C_s = C_{S/2}^{S/2}$ 个可能组合;step4: $\lambda = \{ \}$;**for** $c \in C_s$ **do** 对比 c 号性能的 IS 的表现和 OOS 表现差异, 计算对数几率比 λ_c ; $\lambda = \lambda \cup \{ \lambda_c \}$ **end**step5: $PBO = \int_{-\infty}^0 f(\lambda) d\lambda$

Step1:

- $M_{T \times N}$ 的第 n 列对应从 1 到 T 的时间段内, 第 n 号策略在每一个时间 t 上的损益情况
- 第 n 号策略由第 n 组模型超参组合所唯一确定
- $M_{T \times N}$ 的同一行对应了所有策略在时间 t 各自执行的同步结果
- 要求所有策略在 t 上都要用明确的结果, 因此 $[t, t+1]$ 是所有策略执行频率的最小公倍数

Step2:

- S 必须是一个偶数
- 子矩阵 M_s 的维度为 $\frac{T}{S} \times N$

**Step3:**

特别注意: 分成了 $C(S, S/2)$ 组, 这个林老师上课特别提到过。

Step4:

- (1) 对于每一个 c 将其对应的子矩阵组合设定为训练集 J , J 的维度应为是 $\frac{T}{2} \times N$
- (2) 将没有选入训练集的子矩阵组合在一起设定为测试集 \bar{J} , \bar{J} 的维度应为是 $\frac{T}{2} \times N$
- (3) 构造一个长度为 N 的向量 R , $R[n] = R_n$ 给出 J 中第 n 列计算得到的策略损益的回测统计量
- (4) 确定训练集内的最优表现策略 $n^* = \arg \max \{R_n\}$
- (5) 构造一个长度为 N 的向量 \bar{R} , $\bar{R}[n] = \bar{R}_n$ 给出测试集 \bar{J} 中第 n 列计算得到的策略损益的回测统计量
- (6) 确定训练集中最优策略的回测统计量 \bar{R}_{n^*} 的在测试集 \bar{J} 中的排序相对位置
 $\omega_c \in [0, 1]$. ω_c 越大, 则说明 IS 中最优策略在 OOS 中表现依然越好。
- (7) 计算对数几率比 $\lambda_c = \log \left(\frac{\omega_c}{1 - \omega_c} \right)$

Step5:

- 当训练集的最优策略表现比样本外一半的策略好时, $\omega_c = 0.5$, $\lambda_c = 0$
- 对于 λ_c 构成的分布, 有 $\int_{-\infty}^{+\infty} f(\lambda) d\lambda = 1$
- : $PBO = \int_{-\infty}^0 f(\lambda) d\lambda$ 计算出的是 IS 中最优策略表现劣于 OOS 所有策略表现中位数的概率。
- 如果选择性偏差越大, 则计算出的 PBO 也会越大。

这里“分布”的意思: 每做一遍样本内/外分割, 就会有一个 λ_c 。一共会生成 $C(S, S/2)$ 个 λ_c , 这 $C(S, S/2)$ 个 λ_c 构成了一个 λ_c 的经验分布。

8.6 历史型单路径回测算法 (HSP):

历史模拟: 假设策略在曾经的历史数据执行一遍, 作为当未来历史重演时策略的真实表现。历史只有一次, 所以模拟出也是一条路径上策略的表现, 故称之为历史型单路径回测。

合理性: 只要避免使用后视数据, 历史模拟表现可以看作是假设策略在历史中执行的实际表现。

缺点: ①因为仅在一条历史路径上进行回测, 很容易陷入选择性偏差, 导致策略回测过拟合;

②一段回测历史中往往包含了多个明显不同的市场环境, 比如包含快牛市、股灾、灾后反弹, HSP 只能是这些环境按历史顺序演化后策略的总体结果。

③未来环境按不同顺序不同时间长度出现时, HSP 无法给出评估结果。

8.7 场景型交叉验证回测算法 (NCV):

动机: 克服 HSP 的选择性偏差问题, 数据通过 CV 算法拆分为不同的环境, 然后推断特定环境下策略未来的表现。

场景型交叉验证: 对于每个回测场景, 使用除该场景时间窗口内的所有数据训练模型, 生成策略。在该场景中模拟策略效果。首先模拟在各种不同的环境下未来的表现, 然后再按这些环境出现的历史顺序重新拼装起来产生路径上的模拟效果。

优点:

①使用 CV 的思想进行回测, 可以支持多个不同的场景 (k 个测试集);

②生成策略的训练集样本大小一致, 利用的信息量一致, 后期有可比性;

③每个场景对应一个唯一的测试集, 不像 HSP 需要设定预热窗口。这样使得每个数据都能参与回测。

缺点:

①没有明确的历史型解释, 它的结果不是完整的模拟策略在过去的表现以反映未来;

②NCV 结果经过拼装后还是只在一条路径 (代表历史路径), 只形成一次推断。当需要以历史为基础建立多种场景下的策略执行效果的统计分布时, NCV 无法实现。

③在回测时可能出现信息泄露问题。

8.8 组合清除交叉验证回测算法 (CPCV): 是 NCV 的推广, 将一条历史路径扩展到多条。

CPCV 法基于 Purged-CV 来构建场景型交叉验证, 避免了 NCV 的信息泄露问题。

Algorithm 2: 组合清除交叉验证回测算法: CPCV 算法

Result: 产生在 φ 条模拟的回测路径上, 策略执行效果评价指标

step1: 将 T 个样本分为 N 组, 前 $N - 1$ 个样本容量为 T/N , 最后一个为

$$T - \lfloor T/N \rfloor (N - 1);$$

step2: 从 N 个组中取 k 个作为测试集, 一共有 $S = C_N^k$ 个组合;

step3: 对每一个组合使用 Purged 和 Embargo 算法进行清除和隔离;

step4: 执行 CV;

for $s \in 1 : S$ **do**

 | 在第 s 组的训练集中训练策略模型, 并在测试集中进行预测

end

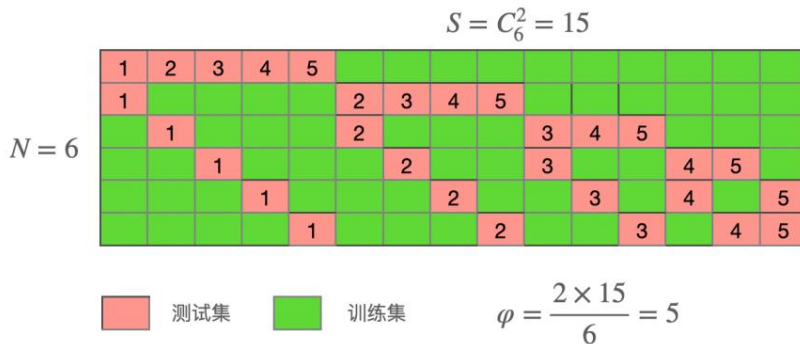
step5: 执行 Back test。拼接构造 $\varphi[N, k] = \frac{C_N^k k}{N}$ 条模拟路径;

for $i \in 1 : \varphi$ **do**

 | 在第 i 条路径上完整的计算执行策略的效果 (夏普率、胜率)

end

计算 φ 条模拟路径上的效果分布



一个解释视频: <https://youtu.be/hDQssGntmFA?t=248>

- $k = 1$: $\varphi = \frac{C_N^1 \times 1}{N} = 1$ CPCV 退化为 Purged-Embargo CV
- $k = 2$: $\varphi = \frac{C_N^2 \times 2}{N} = N - 1 \approx N$ 所以要生成 φ 条路径, 那么就划分为 $\varphi + 1$ 组
- $k = 2$: 选 $N = T$, 一共将有 $T - 1$ 条回测路径, 策略将会在 $1 - \frac{2}{T}$ 比例上进行训练。
- $k \rightarrow \frac{N}{2}$: 路径为最多。

8.Test 判断

1)在回测中，我们需要对超参数进行不断的调整，从而使得模型的预测效果提升，进而挖掘出更为符合显示的因果关系。

2)回测可以在研究中途过程中进行，因为这可以使得我们更早发现错误。

3)因为幸存者偏差，回测数据只包括当前活跃资产，这会造成回测过拟合。

4)场景型交叉验证回测算法有着明确的历史解释。

5)场景型交叉验证回测算法可以通过拼接形成多条历史路径。

8.Test 计算

1.在 CSCV 算法中， $T=12, N=4, S=6$ 。请计算：

(1)子矩阵一共有多少种可能的组合？

(2)子矩阵的维度是多少？

(3)训练集中最优回测统计量在测试集中的排序相对位置为 0.2（从前往后），请问 λ_c 为多少？

(4)假设 λ 分布如下所示：1.5, -2.0, 1.0, -4.5, 3.0, 2.5, -1.0, 0.5, 2.0, -3.0, 0.5, -1.5, 4.0, -0.5, 0.0, -2.0, 1.0, -2.5, 2.0, -1.0，请问 PBO 为多少？

2.在 CPCV 算法中，将 100 个样本分为 10 组，从中选择 k 个作为测试集，请计算：

(1)当 CPCV 算法退化为 Purged-Embargo CV 时， k 是多少？测试集组合有多少个？路径有多少条？

(2)当 $k = 2$ 时，测试集组合有多少个？路径有多少条？

(3)当 $k = 2$ 时，如果我们想有 4 条路径，应该将样本分为几组？每组有几个样本？

(4)当 $k = 2$ 时，如果我们想像“留一交叉验证”那样进行回测，应该将样本划分为多少组？每组有几个样本？测试集组合有多少个？路径有多少条？策略训练/回测的比例是多少？

(5)当路径条数最多时， k 是多少？测试集组合有多少个？路径有多少条？

8.Answer 判断

1) × 2) × 3) × 4) × 5) ×

8.Answer 计算

1.(1)20; (2)8; (3)-1.386290; (4)0.45;

2.(1)1; 10; 1; (2)45; 9; (3)5; 20; (4)100; 1; 4950; 99; 0.98; (5)5; 252; 126;