

文杰 因子 总结

截止到现在（2025.8.5），做出了 5个 符合入库标准的因子（全部为以太坊15m 因子，收益率看10期以后）。
（关于构建模型的相关工作，第六周的时候搭好了回测框架，尚未有成果）

因子

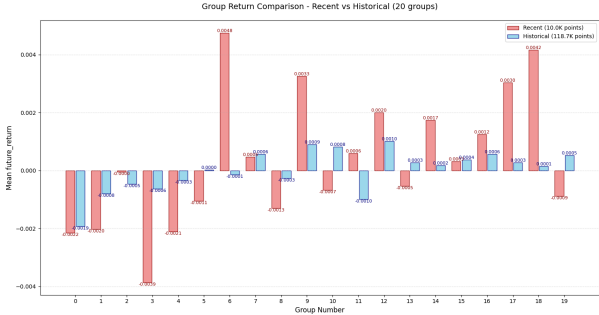
只讨论 人工挖因子 的情况。（因为 机器造因子 既难以解释、我又没有实操过）

造新的 & 改旧的

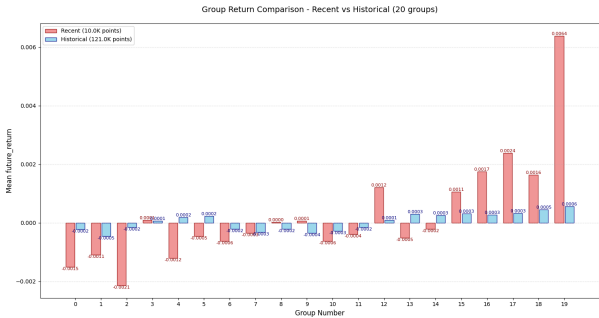
- 按我目前的理解，“尽力把一个 promising 的因子修改到表现出色”比“尽力想出新思路的因子”更重要（或者说更实用）一些。
 - 造新因子 基本只能通过：构建新的具有金融意义的思路
 - 现实情况是，（归根结底还是我太菜/经验不足）造不出那么多全新又有效的新思路
 - 我实习的前两周几乎全耗在“闭门造轮子”的困境里
 - 但 修改已有（表现不如人意的）因子 既可通过金融意义去完善原因子逻辑，又可根据因子表现数据驱动地进行修改
 - 手段更丰富
 - 更具针对性

金融意义驱动 & 数据驱动

假设情景：目前已有有一个80分的因子，主要问题出在因子值分类后头部（因子值大的部分）收益率出现了反向。



我需要修改该因子，使得其总体收益率能够呈现出比较明显的单调性（头尾部更重要，因为在进入到构建量化模型之后头尾部对于训练模型的影响更大）（如下图所示）



- 金融意义驱动 流程：
 - 需要分析出：
 - 按照原先预想，何时因子值最大/最小
 - 在实际情况中，是否可能出现未曾料到的影响因素，使得因子值受此因素的影响被误划为头/尾部
 - 回到原因子构建逻辑，针对分析出的意外情况进行修正
- 常用手段：
 - 动态（自适应）阈值/时间窗口/权重.....
 - 量价/分形市场/信息熵.....复合确认
 - 中性化、取残差
 - 用其他因子作 filter
- 所仰仗的能力 & 知识：
 - 对常用技术指标的熟悉（最基本）
 - 对 金融市场 各种复杂情况的 详尽分析
 - 实际交易经验（对深挖金融逻辑非常有用）

- 数据驱动 流程：
 - 根据收益率/因子值分布图/散点图/累积 IC, Rank_IC 图.....从纯数据角度改进因子实现方式
 - 常用手段：
 - 搜参、调参
 - 加算子（加减乘除、绝对值、开方/平方.....）
 - 强硬截尾/翻转/.....
 - 构造函数。假设因子值为 x ，收益率表现为 $f(x)$ （从收益率图中大概估出来），则可构造 $g(f(x))$ 使得 $g(\cdot)$ 的收益率表现单调递增
 - 注意事项：
 - 为防过拟合风险，数据驱动的诸多操作最好都能具有一定金融意义
 - （比如对于 RSI 指标，对中间部分进行翻转即为相对合理、可解释的）
 - 所仰仗的能力 & 知识：
 - 对数据的敏感程度
 - 对数据的丰富处理手段

目前做出的全部因子

1. keltner_channel

构造

```
def factor(df, ema_period=20, atr_period=10, multiplier=2):
    '''凯尔纳通道：基于ATR的波动通道'''
    ema = df['close'].ewm(span=ema_period, adjust=False).mean()

    # 计算ATR
    high_low = df['high'] - df['low']
    high_close = abs(df['high'] - df['close'].shift(1))
    low_close = abs(df['low'] - df['close'].shift(1))
    true_range = pd.concat([high_low, high_close, low_close], axis=1).max(axis=1)
    atr = true_range.ewm(span=atr_period, adjust=False).mean()

    # 计算通道宽度
    channel_width = multiplier * atr

    factor = (df['close'] - (ema + channel_width)) / (2 * channel_width)

    return factor
```

逻辑

1. 通道突破信号：
 - 因子值 > 0：价格突破上轨（超买信号，可能回调）
 - 因子值 < -1：价格突破下轨（超卖信号，可能反弹）
 - 因子值 $\in [-1, 0]$ ：价格在通道内运行（趋势延续）
2. 波动率调整：
 - ATR 衡量市场波动性，通道宽度动态调整：
 - 高波动 → 通道变宽 → 减少假突破信号
 - 低波动 → 通道变窄 → 提高灵敏度

2. 聪明钱_市场状态过滤

构造

1. 计算每根K线的平均价和涨跌幅绝对值
2. 回溯过去10天（960根15分钟K线）的数据
3. 计算每根K线的S值 = rank(|R|) + rank(V)
4. 按S值降序排序，取成交量累积占比前20%的K线
5. 计算聪明钱VWAP和整体VWAP

6. 因子值 = 聪明钱VWAP / 整体VWAP

1. 结合成交量分布和价格位置双重确认
2. 增加交易意图分析（主动买入/卖出）
3. 使用动态时间窗口

逻辑

- 智能窗口调节：
 - 高波动 → 短窗口 → 捕捉近期主力动向
 - 低波动 → 长窗口 → 识别长期资金布局
- 交易意图解析：
 - 正S值：主力主动做多（量价齐升+大单买入）
 - 负S值：主力主动做空（量价齐跌+大单卖出）
- 趋势自适应：
 - 熊市反转信号：当 `smart_vwap < all_vwap` 时转为看多
 - 牛市强化信号：当 `smart_vwap > all_vwap` 时确认趋势

3. 时间重心偏离分形市场未来数据_快

构造

1. 计算每根K线的收益率
 2. 按日分组计算每日的涨幅/跌幅时间重心
 3. 滚动回归计算跌幅重心偏离
 4. 取20日平均作为因子值
-

1. 实时滚动计算：
 - 替代日级分组，采用滚动扩展窗口计算重心
 - 每个时点仅使用截至当前时刻的当日数据
2. 分维度动态权重：
 - 权重 = 分形值 × 成交量
 - 分形值实时更新： `fractal = std5(range)/std20(range)`
3. 滚动回归隔离：
 - 训练数据严格限定在当前时间点之前
4. 连续信号输出：
 - 前向填充缺失值 → 分钟级连续信号
 - 极值过滤： `abs(factor)>10 → 0` （注释状态）

逻辑

1. 微观结构响应：
 - 每根K线实时更新重心位置
 - 早盘下跌立即反映在因子值中
2. 分形市场适应性：
 - 分形值>1：放大信号敏感度（市场结构变化期）
 - 分形值<1：保持信号稳定（趋势延续期）
3. 规避未来函数：
 - 回归预测仅使用 已发生数据
 - 彻底杜绝训练数据污染

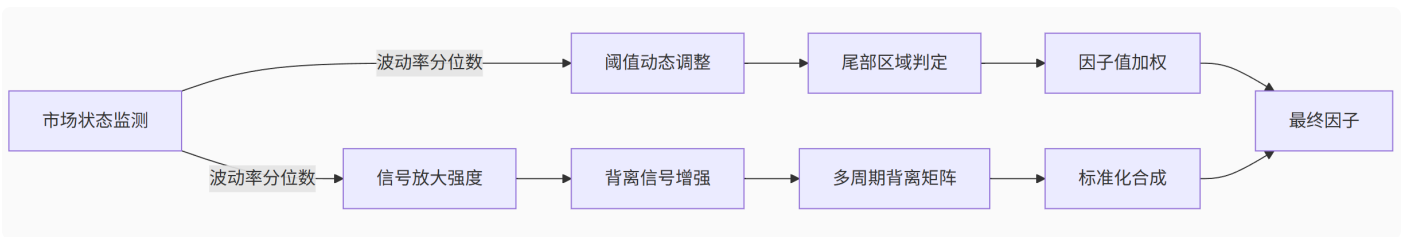
4. VCF

构造

- 核心指标：
 - 多周期成交量均线：[1, 5, 10, 20, 60, 120] 日（换算为15分钟 K 线数）
 - 标准差计算：std(当前成交量, MA5, MA10, MA20, MA60, MA120)
- 转换公式：VCF = ln(1 + σ_{vol_ma})
- 计算流程：
 - 计算6个周期的成交量移动平均
 - 组合成 N×6 特征矩阵
 - 计算每行（每个时点）的标准差
 - 对标准差进行对数变换

- 波动率驱动决策：
 - 波动率分位数同时控制阈值和放大系数
 - 低波动市场：更敏感的信号捕捉 + 更强的放大
 - 高波动市场：更稳健的信号过滤 + 适度放大
- 非线性放大机制：
放大系数 = 基础值 - 波动率分位数 × 衰减系数
 - 背离信号：1.8 → 1.2（波动率0→1）
 - 最终因子：2.0 → 1.5（波动率0→1）
- 信号质量过滤：
 - 剔除绝对值<0.76的弱信号，保留高置信度预测

逻辑



5. RSRJV

构造

- 收益率计算：
 - 基于收盘价计算对数收益率：log_returns = np.log(close[1:]) - np.log(close[:-1])
- 基础参数设定：
 - 常数 mu1 = sqrt(2/π)（标准正态分布绝对值的期望）
 - 其平方倒数 mu1_inv_sq = 1/(mu1²) 用于双幂变差计算
- 滚动窗口计算（窗口=96）：
 - 总已实现波动率（RV）：
 - 窗口内所有收益率平方和：RV = Σ(r_t²)
 - 双幂变差（BV）：
 - 估计连续路径波动：BV = mu1_inv_sq × (n/(n-1)) × Σ(|r_t| × |r_{t+1}|)
 - 方向跳跃波动分离：
 - 上行跳跃波动 RJVP = max(RV₊ - BV/2, 0)
 - 下行跳跃波动 RJVN = max(RV₋ - BV/2, 0)
 - 其中 RV₊ / RV₋ 为正/负收益率平方和
 - 核心指标合成：
 - 跳跃不对称性 SRJV = RJVP - RJVN
 - 归一化因子 RSRJV = SRJV / RV
- 因子序列生成：
 - 原始因子值：factor_series = -RSRJV（负号反转方向）
 - 尾部调整：
 - 对底部20%值取反：factor_series[≤Q_{1.0}%] = -factor_series
 - 剔除顶部8.3%值：factor_series[≥Q_{91.7}%] → NaN

核心机制：跳跃风险不对称溢价

- 高因子值（调整后）：
 - 对应 原始RSRJV极端负值 \rightarrow $SRJV \ll 0$
 - 下行跳跃波动显著强于上行 ($RJVN \gg RJVP$)
 - 反映 市场恐慌加剧：坏消息引发剧烈下跌跳跃
 - 示例：财报暴雷后连续跳空下跌
- 低因子值（调整后）：
 - 对应 原始RSRJV正值 \rightarrow $SRJV > 0$
 - 上行跳跃波动占优 ($RJVP > RJVN$)
 - 暗示 贪婪情绪主导：好消息引发暴涨跳跃
 - 示例：政策利好刺激跳空高开
- 零值附近：
 - 跳跃波动对称 ($RJVP \approx RJVN$)
 - 市场无显著方向性跳跃风险

尾部调整的金融逻辑

1. 底部20%取反：
 - 将 极端恐慌状态（原始RSRJV极负）转换为高因子值
 - 捕捉 "恐慌过度修正" 机会：物极必反
2. 顶部8.3%剔除：
 - 过滤 流动性枯竭 时的异常值（如闪崩/轧空）
 - 避免极端噪音干扰策略稳定性

模型

- 回测计算方式：**单项持仓**
 - **持仓**
 1. 资金被预先分成10等份。
 2. 在任何时刻，所有持仓**方向必须一致**（全多或全空）。
 3. 每次开仓只使用一份资金。
 4. 持仓有固定生命周期（10个 bar），到期强制平仓。
 - **平仓逻辑**
 1. **到期平仓**：每个仓位独立计时，持有10个bar后自动平仓。
 2. **反向信号平仓**：当收到相反信号时，**仅平掉最早开立的一份持仓**
 - **盈亏计算**
 - 完全基于**已实现盈亏 (Realized PnL)**。
 - 只有当一笔交易（无论是到期、反向信号还是期末强制平仓）被关闭时，其产生的真实盈亏才会被计入总权益。