

金融科技導論 HW1

許博翔

October 15, 2023

0 Estimating the Result

作業給的 `rrEstimate.py` 是跑過所有 2017-10-05 到 2022-10-03 的收盤價來衡量最後的結果，也就是從 2017-10-05 就開始交易 0050，但是我們實際上是要從 2023-10-16 開始交易，而前面的資料只當作參考，因此我重寫了一遍 `rrEstimate.py`，將衡量結果的方式改成是在 2022-10-14 到 2023-10-13 這一年中進行交易，而 2022-10-14 以前的資料就只用做計算 MA、RSI、K 線、D 線等用途。

使用重寫過的 `rrEstimate.py` 跑作業給的範例 strategy 的結果是 20.19%。

1 MA

我使用 `myStrategy1.py`，其中短期均線為 $maShort$ 、長期均線為 $maLong$ ，策略是當 $maShort - maLong > \alpha$ 時就買進，當 $maShort - maLong < -\beta$ 時就賣出。

其中 $maShort$ 是 $nmaShort$ 天以來的平均值，而 $maLong$ 是 $nmaLong$ 天以來的平均值，我使用 `run1.sh` 爆搜所有的 $1 \leq nmaShort \leq 10, 11 \leq nmaLong \leq 20, -3 \leq \alpha \leq 3, -3 \leq \beta \leq 3$ 之後，用 `getRes.py` 找出最好的 30 組結果，發現最好的 15 組結果滿足為 $7 \leq nmaShort \leq 9$ 與 $12 \leq nmaLong \leq 15, \alpha = -1$ ，而對於任意的 $-3 \leq \beta \leq 1$ 並無差異。而這些最好的 15 組結果介於 31.22% 到 33.71% 之間，比範例的結果多賺了 67%。

從這邊的結果得知，使用 $(nmaShort, nmaLong) = (8, 13)$ 有較好的結果，於是對於之後在做混合指標的判斷，將會採用這些參數。

2 RSI

我使用 `myStrategy2.py`，其中往前分析的天數為 n 、超買指標為 ob 、超賣指標為 os ，策略是當這 n 天的 $RSI > ob$ 時就賣出，當 $RSI < os$ 時就買進。我使用 `run2.sh` 爆搜所有的 $10 \leq n \leq 20, 0.1 \leq ob \leq 0.9, 0.1 \leq os \leq 0.9$ 之後，用 `getRes.py` 找出最好的 30 組結果，發現最好的 2 組結果為 $n = 20$ ，而 (ob, os) 分別為 $(0.8, 0.3), (0.7, 0.3)$ 。這些最好的 2 組結果為 34.08% 與 33.88%，比範例的結果多賺了 69%。

從這邊的結果得知，使用 $n = 20$ 有較好的結果，於是對於之後在做混合指標的判斷，將會採用這些參數。

3 K 線、D 線

我使用 `myStrategy3.py`，其中 RSV 是以前 n 日的計算，而

當日 K 值 = r 前日 K 值 + $(1 - r)$ 當日 RSV

當日 D 值 = r 前日 D 值 + $(1 - r)$ 當日 K 值

策略是當 $K > D$ 時就買進， $K < D$ 時就賣出。

在使用 `run3.sh` 爆搜 n, r 值時，發現最好的 30 個結果用的 n, r 落差很大，並非與結果高度相關，因此我最後選擇使用一般最常使用的 $(n, r) = (9, \frac{2}{3})$ 。

4 Linear Regression

在綜合考量前述的那些指標，我使用簡單的線性 model：linear regression 來計算如合分配那些指標的比重。

具體而言，設 x_{ij} 是第 i 天資料的第 j 項指標（其中 $x_{i1} = 1$ 為常數項），而 y_i 是第 i 至第 $i + 1$ 天的漲幅，則目標是找到 w_1, w_2, \dots, w_k （有 k 項指標）使得

$$\sum_i (y_i - \sum_{j=1}^k x_{ij} w_j)^2 \text{ 最小。}$$

令 X 為一矩陣，其中第 i 列第 j 行是 x_{ij} ，而 $y := \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ ，則目標是找到 w 使得 $\|Xw - y\|$ 最小，而 w 就會是使得 Xw 是 y 到 X 的 column space 上的投影

點的解，可以知道 $w = X^\dagger y$ ，其中 X^\dagger 是 X 的 pseudo inverse。

另外，爲了避免 noise data，我使用 20-fold cross validation，也就是隨機排列 data 後將其中的 $\frac{19}{20}$ 拿來 linear regression，剩下的 $\frac{1}{20}$ 拿來計算誤差，取誤差最小所對應到的 w 。

5 綜合指標

可以注意到前面所使用的 α, β 等常數並沒有拿來放在綜合指標，因爲這些常數的會全部都會被包含在 linear regression 所使用的常數項中。我使用的指標包括常數項 1、當日的收盤價、 $maShort, maLong, RSI, K$ 值與 D 值。

使用 mixed.py 搭配 linear regression 算出來的結果爲 $w = [0.00823331, -0.04023871, 0.08947945, -0.05024477, 0.12391539, 0.00555206, -0.00444235]^T$ 。

接著，使用以下策略：

令 $y = w^T [1, \text{當日的收盤價}, maShort, maLong, RSI, K \text{ 值}, D \text{ 值}]^T$ ，即預測出來隔天與今天的價差，當 $y > a$ 時，買進；當 $y < b$ 時，賣出。

將其寫成 myStrategy4.py 之後再用 run4.sh 爆搜 a, b ，得到在 $(a, b) = (0, -0.1)$ 時有最好的結果 26.54%。

6 策略結果比較

最後要選擇使用的 model 作爲上傳的 strategy，其中爆搜參數的部份有可能會對於這一年內的資料 overfitting，所以我拿了其他也不錯的參數的結果來比較。

使用 model	最好結果	前 $\frac{1}{4}$ 好的結果平均	前 $\frac{1}{4}$ 好的結果標準差
範例	20.19%	X	X
MA	33.71%	25.54%	2.34%
RSI	34.08%	21.91%	2.32%
K 線、D 線	23.16%	18.52%	1.60%
Mixed	26.54%	22.54%	2.08%

可以看到 MA 與 RSI 最好的結果差不多，而 Mixed 少了很多，但是 MA 的平均超過 RSI 很多，而標準差除了 K 線、D 線很小之外剩下的都差不多，所以我最後選擇了 MA 作爲上傳的 strategy。