

Pseudocode

Module: FindPos

Purpose: To get the positions

Class: Algorithm

Member Function: `__init__()`

(will be called once after we have TradingWindow Bars ready with us)

Purpose: To initialise attributes of Class Algorithm.

1. Initialize $d=5.0$
2. Initialize $a=1000$
3. $\text{Alpha}=100.0$
4. $\text{Beta}=60.0$
5. Initialize $\text{Gamma}=5.0$
6. Initialize $\text{Dell}=0.007$
7. Initialize $\text{TradingWindow}=80$
8. Initialize $\text{BarsBack}=25$
9. Initialize $\text{RegressionWindow}=20$
10. Initialize $\text{PriceSmoothWindow}=3$
11. Initialize $\text{PositionVector}=\text{Zeros}(1, \text{TradingWindow}-1)$
12. Initialize $\text{Prices}=\text{Close Prices of last TradingWindow Bars}$
13. Initialize $\text{Profit}=\text{Zeros}(1, \text{TradingWindow}-1)$
14. Initialize $\text{CumProfit}=\text{Zeros}(1, \text{TradingWindow}-1)$
15. Initialize $\text{Temp}=\text{Zeros}(1, \text{TradingWindow}-1)$
16. Initialize $\text{PastPos}=-1$
17. Initialize $\text{Threshold}=0.1$.
18. Initialize $\text{Phi}=\text{Zeros}(\text{BarsBack}, \text{TradingWindow})$
19. Initialize $\text{Return}=\text{Zeros}(1, \text{TradingWindow}-1)$
 - 18.1 $\text{SmoothPrices}=\text{MovingAverage}(\text{Prices}, \text{PriceSmoothWindow})$
 - 18.2 $\text{Return}(i)=\text{SmoothPrice}(i)-\text{SmoothPrice}(i-1)$, for all $i=1:\text{TradingWidow}-2$
20. $\text{Weights}=\text{Zeros}(\text{BarsBack}, \text{TradingWindow})$
21. $\text{MaxCumProf}=-1.0$

Member Function: MovingAverage()

Input: Price, PriceSmoothWindow

Purpose: To Compute the Moving Average

1. Compute the Simple Moving Average of Price vector for a PriceSmoothWindow.

Member Function: PositionCalculation()

Input: CurrentBarDate,CurrentBarTime,CurrentBarClosePrice,CurrentTimeIndex

Purpose: To get the position

(will be called for each bar)

1. Append CurrentBarClose to Prices.
2. SmoothPrices=MovingAverage(Prices,SmoothPriceWindow)
3. Append SmoothPrices(CurrentTimeIndex-BarsBack:CurrentTimeIndex) to Phi Matrix.
4. I_Phi_Used=Phi(CurrentTimeIndex-TradingWindow:CurrentTimeIndex+1,:)
5. Normalise the I_Phi_Used matrix by dividing each column with its sum.
6. Convert it to 0 mean.
7. Append SmoothPrice(CurrentTimeIndex-1)- SmoothPrice(CurrentTimeIndex-2) to Returns.
8. Append Zeros(BarsBack,1) to Weights.
9. Append 0.0 to Temp.
- 10.Append 0 to PositionVector
- 11.Append 0.0 to Profit.
- 12.Append 0.0 to CumulativeProfit.
- 13.Initialize I_Lambda=Zeros(1,TradingWindow)
- 14.Initialize I_G=Zeros(1,TradingWindow)
- 15.Initialize Bias=0.0
- 16.Initialize Hessian Matrix
- 17.Solve the QPP
- 18.Get the Lambda and G

19. Compute Weights.
20. Append Weights to Weight matrix
21. Compute Bias.
22. Append $\text{Weights}^T * I_Phi_Used(:, \text{TrsdngWindow}) + \text{Bias}$ to Temp.
23. If $\text{Temp}(\text{CurrentTimeIndex}) > 0.1$, then $\text{CurrentPosition} = 1$,
24. Else If $\text{Temp}(\text{CurrentTimeIndex}) < -0.1$, then $\text{CurrentPosition} = -1$
25. Else $\text{CurrentPosition} = 0$.
26. Append CurrentPosition to PositionVector .
27. Compute $\text{Profit}(\text{CurrentTimeindex}) = \text{PositionVector}(\text{CurrentTimeIndex} - 1) * [\text{Prices}(\text{CurrentTimeIndex}) - \text{Prices}(\text{CurrentTimeIndex} - 1)]$
28. Compute
 $\text{CumulativeProfit}(\text{CurrentTimeIndex}) = \text{CumulativeProfit}(\text{CurrentTimeIndex} - 1) + \text{Profit}(\text{CuurentTimeIndex})$