

DESIGN DOCUMENT

Objective:

To get the trading position for the market.

Description:

The trading algorithm is based in the following formulation: -

$$\min \left\{ \frac{1}{2} (\|\omega\|^2 + A\theta^2) + \alpha \sum_{t=1}^T a_t - \beta \sum_{t=1}^T b_t + \gamma \sum_{t=1}^T (c_t + d_t) \right\} \quad (1)$$

Subject to:-

$$r_t [\omega^T \phi(x^{t-1}) + \theta] + (a_t - b_t) = 0 \quad \text{such that } a_t, b_t \geq 0$$

$$\delta \omega^T [\phi(x^t) - \phi(x^{t-1})] + (c_t - d_t) = 0 \quad \text{such that } c_t, d_t \geq 0$$

The formulation named as “OMEGA RATIO FORMULATION” aims at minimizing the losses (contained in a_t), transaction costs (contained in c_t and d_t) and maximizing the profits (contained in b_t).

The ϕ 's represents the samples. Each sample is BarsBack dimensional i.e. each sample has past BarsBack features which can simply be the closing prices or some complex Indicator or a combination of both.

The T is the TradingWindowSize ($T \geq 1$).

α, β, γ are the weighing factors for the cumulative losses, profits and transaction costs respectively. The condition $\alpha > \beta$ ensures that minimizing losses is more important than maximizing profits. So we carefully implement this condition while selecting the parameters.

r is the return at time t given by $r_t = \text{price}_t - \text{price}_{t-1}$

δ is the transaction cost incurred for the scrip being traded.

The above formulation is our primal formulation. After we apply KKT conditions and convert the above into dual formulation, it becomes:-

$$\begin{aligned} \text{Min } \frac{1}{2} \sum_{t=1}^T \sum_{u=1}^T \{ & \lambda_t r_t \lambda_u r_u \phi(x^{t-1})^T \phi(x^{u-1}) + \lambda_t r_t g_u \delta [\phi(x^{t-1})^T \phi(x^u) - \\ & \phi(x^{t-1})^T \phi(x^{u-1})] + g_t g_u \delta^2 [\phi(x^t)^T \phi(x^u) - \phi(x^t)^T \phi(x^{u-1}) - \phi(x^u)^T \phi(x^{t-1}) + \\ & \phi(x^{t-1})^T \phi(x^{u-1})] + \frac{1}{A} \lambda_t r_t \lambda_u r_u + \lambda_u r_u g_t \delta [\phi(x^{u-1})^T \phi(x^t) - \phi(x^{u-1})^T \phi(x^{t-1})] \} \end{aligned} \quad (2)$$

Constrained to the following conditions: -

$$\beta \leq \lambda_t \leq \alpha$$

$$-\gamma \leq g_t \leq \gamma$$

Here, $\Phi(x^t)^T \Phi(x^u) = K(x^t, x^u)$ or *Kernel*(x^t, x^u). The kernel function can be one of the many commonly used kernels such as rbf kernel, polynomial kernel etc.

Minimizing equation (2) gives set of λ 's and g 's which gives us ω and θ .

$$\omega = \sum_{t=1}^T \lambda_t r_t \Phi(x^{t-1}) + \sum_{t=1}^T g_t \delta (\Phi(x^t) - \Phi(x^{t-1}))$$

$$\theta = \frac{1}{A} \sum_{t=1}^T \lambda_t r_t$$

Ideally, the current position in the market is given by $\text{sgn}(\omega^T \Phi(x^t) + \theta)$. But to inculcate the confidence of trader into our strategy and to rule out the possibility of bad trades, we compute the current position as: -

If TempPosition[t]> threshold and TempPosition[t-1]>threshold and
 HighPrice[t]>HighPrice[t-1] and shortmovingaverage[t]>longmovingaverage[t],
 Then Current Position=1. {i.e If we are confident, enter the trade}

 ElseIf TempPosition[t]> threshold and Position[t-1]==1 and
 shortmovingaverage[t]>longmovingaverage[t], Then Current Position=1
 {Continue your position as long as trend is positive and Trader is confident}

 ElseIf TempPosition[t]< -threshold and Temposition[t-1]<-threshold and
 HighPrice[t]<HighPrice[t-1] and shortmovingaverage[t]<longmovingaverage[t], Then
 Current Position=-1 . {i.e If we are confident, enter the trade}

 ElseIf TempPosition[t]< -threshold and position[t-1]==-1 and
 shortmovingaverage[t]<longmovingaverage[t], Then Current Position=-1
 {Continue your position as long as trend is negative and Trader is confident}

 Else, Current Position=0

Where , $(\omega^T \phi(x^t) + \theta / || \omega ||)$ can be termed as TempPosition. Division by
 $|| \omega ||$ helps to normalize the output of trader.

Thus alpha, beta, gamma, dell, A, threshold, kernelfunc, TradingWindowSize,
 BarsBack, BarTimeInterval are the parameters of algorithm.

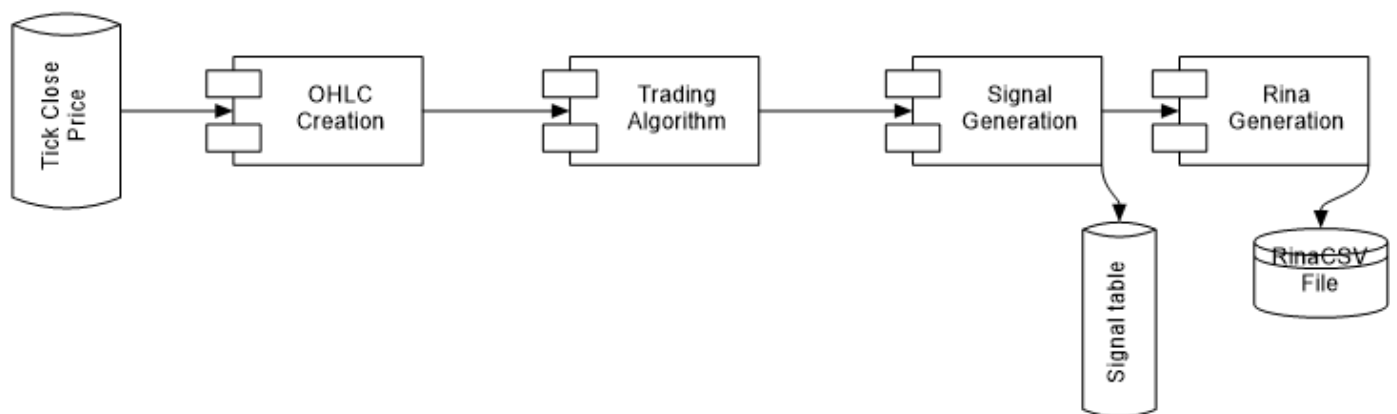
So the algorithm requires past T closing prices to compute the current position.
 The output of the of the trading algorithm is ternary with 1, -1, 0 as the 3 possible
 states representing the long, short, and neutral position in the market
 respectively.

The preprocessing and post processing can be done can be done in an outside
 module. Preprocessing involves reading the data prices for each tick and

preparing the OHLC bars for the time interval specified by BarTimeInterval. Sequence of states can be converted into Trading Signals post processing. We stand Long as long as the output is +1, Short as long as the output is -1 and Neutral whenever the output is 0. Static Trailing and Stop Loss can be applied after we take positions in the market.

After we get the position for current time instant or current bar, we repeat the procedure for next bar by shifting the entire T sized Trading Window to the right exactly by 1 bar.

Component Relationship Diagram:



Design Considerations:

Assumptions:

- The input stream is available for every tick in a continuous manner.
- The signal file is generated for every entry and exit of the trade.
- The RinaCSV file is generated at the end of the trading session.
- The ternary output of Trader is stored for reference purpose.

Constraints:

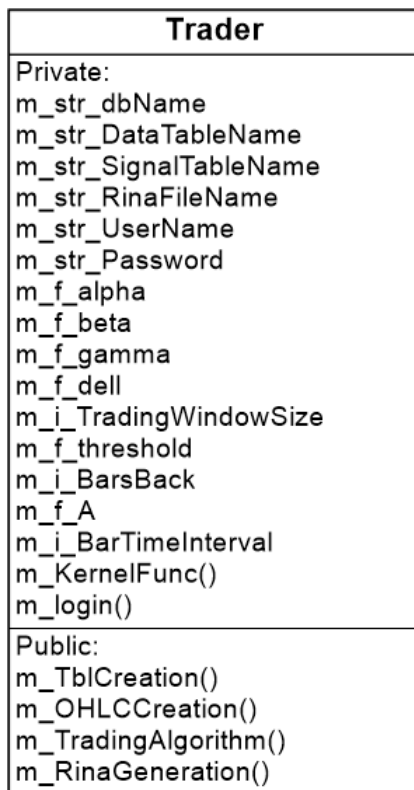
- The input to algorithm is rounded off to 2 places of decimal.
- The rounding off can be done using methods OHLCCreation before calling the algorithm.

Optimizer Used:

ISMO algorithm is being used to optimize the dual objective.

DataSet Used:

A class with the following UML diagram:-



Milestones

- Pseudocode --- Estimated date for completion: June 19, 2015
- Code and Use Cases --- Estimated date for completion: June 25, 2015

Flowchart:

Trader without adaptive trailing

