

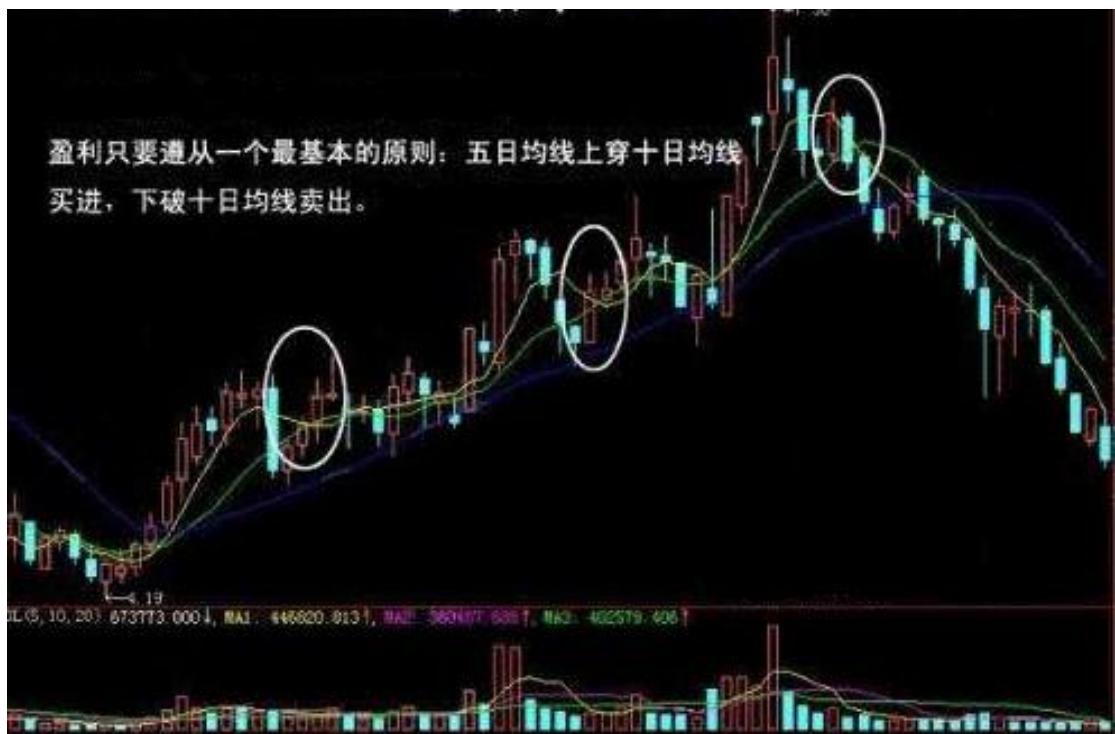
指标和策略介绍：

本策略为均线择时策略，采用 TRIX 指标判断长期趋势，双均线判断短期趋势，主要还是以追涨杀跌来获取收益。

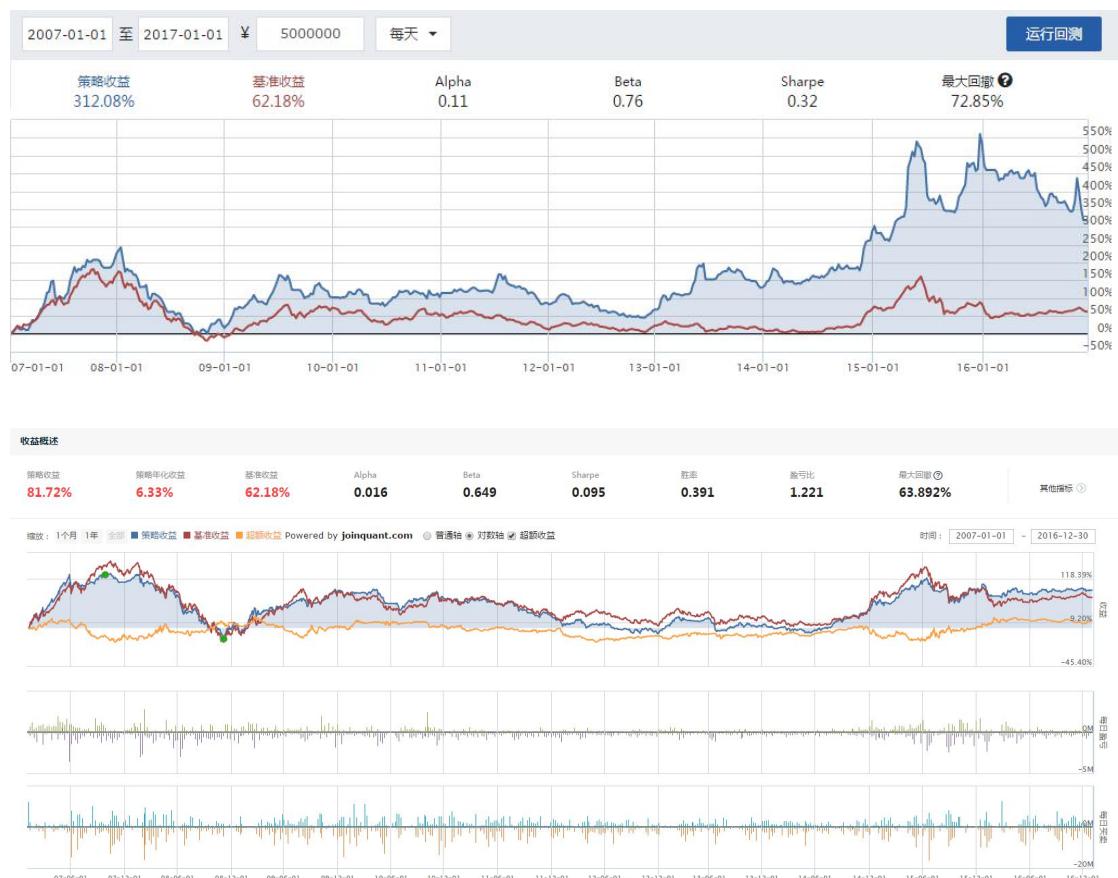
TRIX，中文名称：三重指数平滑移动平均，根据移动平均线理论，对一条平均线进行三次平滑处理，再根据这条移动平均线的变动情况来预测股价的长期走势。长线操作时采用本指标的讯号，可以过滤掉一些短期波动的干扰，避免交易次数过于频繁，造成部分无利润的买卖，及手续费的损失。



双移动平均线，“平均”是指最近 n 天收市价格的算术平均线；“移动”是指我们在计算中，始终采用最近 n 天的价格数据。因此，被平均的数组（最近 n 天的收市价格）随着新的交易日的更迭，逐日向前推移。本策略采用最常用的以收盘价计算的简单移动平均线，分为 5 日线和 10 日线。该指标是反映价格运行趋势的重要指标，其运行趋势一旦形成，将在一段时间内继续保持。

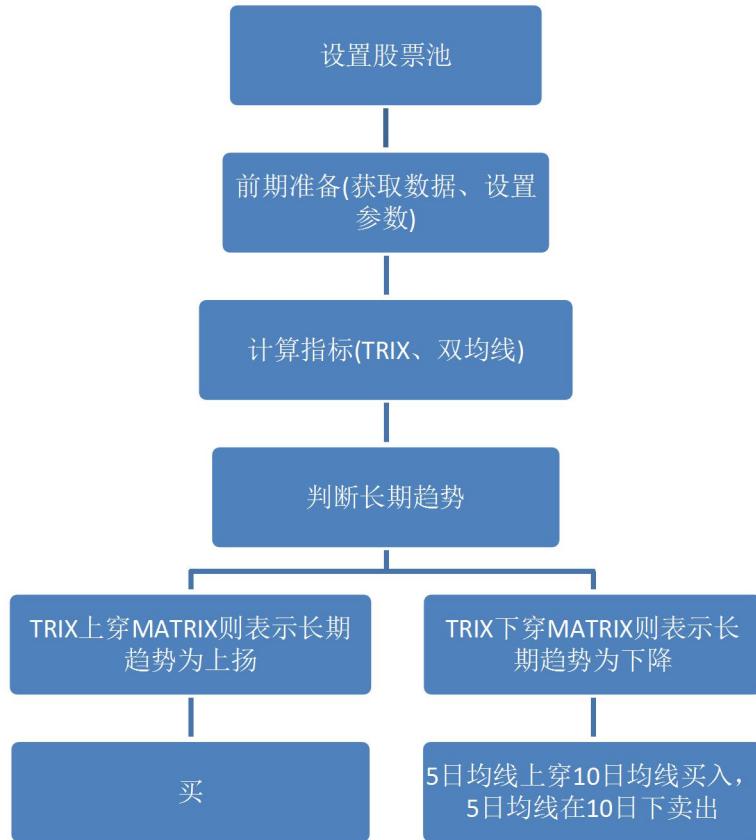


本策略回测的时间均为 2007-01-01 到 2017-01-01，回测股票池分别为沪深 300 前 5 只股票、前 100 只股、指数股，资金为 5000000、50000000、1000000，回测平台为聚宽，回测结果如下：





策略流程:



正文:

首先设置策略的参数, N 代表求 N 日平均值 TR , M 代表保留 M 日的均值信息 $TRIX$, num 代表保留 num 天的 $TRIX$ 的简单移动平均 $MATRIX$ 。此处策略所使用的参数仅供参考, 不代表最优值。

```

N = 20 # 计算TR时的N
M = 49 # 计算MATRIX时的M
num=5
length_of_data = N+M+num # 取closeprice的天数，为了足够计算MATRIX、TRIX

```

我们在使用双移动平均线指标时，采用五日均线与十日均线。

短期移动平均线由下而上穿过上升的长期移动平均线时，我们称这种交叉为黄金交叉，一般来说这是一个值得重视的买入信号，而当长期移动平均线自上而下穿越短期移动平均线时，我们视之为死亡交叉，这一现象往往是一个值得重视的卖出信号。

```

MA5=close_price['close'][-5:].mean()
MA10=close_price['close'][-10:].mean()
ma5=close_price['close'][-6:-1].mean()
ma10=close_price['close'][-11:-1].mean()

```

TRIX 指标是根据移动平均线理论，对一条平均线进行三次平滑处理，再根据这条移动平均线的变动情况来预测股价的长期走势。

1. TR=收盘价的 N 日指数移动平均；
- 2.TRIX=(TR-昨日 TR)/昨日 TR*100；
- 3.MATRIX=TRIX 的 M 日简单移动平均；

```

price_array={}
for i in range(0,M+num):
    price_array[i]=close_price['close'][i:i+N]
#TR=收盘价的N日指数移动平均
TR={}
for i in range(0,M+num):
    TR[i]=np.mean(price_array[i])
#TRIX=(TR-昨日TR)/昨日TR*100
TRIX={}
for i in range(1,M+num):
    if TR[i]==0:
        TRIX[i]=0
        continue
    TRIX[i]=(TR[i]-TR[i-1])/TR[i]*100
#MATRIX=TRIX的M日简单移动平均
MATRIX={}
for i in range(0,num):
    TRIX_sum=0
    for j in range(1,M):
        TRIX_sum=TRIX_sum+TRIX[i+j]
    MATRIX[i]=TRIX_sum/M

```

TRIX 指标是属于中长线技术指标，其最大的优点就是可以过滤短期波动的干扰，以避免频繁操作而带来的失误和损失。TRIX 指标的一般研判标准主要集中在 TRIX 线和 TRMA 线的交叉情况的考察上。我们假设过去的 num 天，TRIX 高于 MATRIX 的日子多于 TRIX 低于 MATRIX 的日子，且昨日的 MATRIX 高于 num 天前，则认为长期趋势为上涨。相反则是下降。

```

length=0
for i in range(0,num-1):
    if TRIX[M+i]>MATRIX[i]:
        length=length+1
    else:
        length=length-1
if length>0 and MATRIX[num-1]>MATRIX[0]:
    trend[security]='up'
elif length<0 and MATRIX[num-1]<MATRIX[0]:
    trend[security]='down'

```

当长期趋势为上涨时，买入相应的股票，当长期趋势为下降，且股票为持有状态，则在五日均线上穿十日均线时买入，五日均线低于十日均线时卖出

```

if trend[security]=='up':
    order_value(security,cash)
elif trend[security]=='down' and security in context.portfolio.positions:
    if MA5<MA10:
        order_target(security,0)
    elif MA5>MA10 and ma5<ma10 :
        order_value(security,cash)

```

策略优缺点：

1. 可以考虑对参数进行优化，
2. 该策略在牛市时获益不够理想，但在熊市时止损效果尚可，
3. 策略在计算时直接将缺失值记录为 0，换用前一个时间段的数据效果会更佳
4. 策略采用字典形式存储数据，这一点比使用列表更精确

附录：

```

# 导入函数库
import jqdata
import pandas as pd
import numpy as np

# 初始化函数，设定基准等等
def initialize(context):
    # 设定沪深 300 作为基准
    set_benchmark('000300.XSHG')
    # 开启动态复权模式(真实价格)
    set_option('use_real_price', True)
    # 股票类每笔交易时的手续费是：买入时佣金万分之三，卖出时佣金万分之三加千分之一印花税，每笔交易佣金最低扣 5 块钱
    set_order_cost(OrderCost(close_tax=0.001, open_commission=0.0003,

```

```

close_commission=0.0003, min_commission=5), type='stock')

#获取沪深 300 股票池
stock_set=get_index_stocks('000300.XSHG')
#此处可增加选股条件
q = query(
    valuation.code, # 股票代码
).filter(
    valuation.code.in_(stock_set),#只对设定股票池执行
)
fdf = get_fundamentals(q)
#取前 50 只股
fdf=fdf.head(50)
#获取股票列表
stock_list=list(fdf['code'])
#趋势指标 trend 的数据类型为字典
trend={}
#初始化
for security in stock_list:
    trend[security]='None'
#按天运行
def handle_data(context, data):
    N = 20    # 计算 TR 时的 N
    M = 49    # 计算 MATRIX 时的 M
    num=3    #保存 num 天 MATRIX
    length_of_data = N+M+num  # 取 closeprice 的天数, 为了足够计算 MATRIX、TRIX
    #资金为当前账户可用资金
    cash=context.portfolio.available_cash/50
    #遍历股票池中每只股票
    for security in stock_list:
        #获取收盘价数据
        close_price=attribute_history(security,length_of_data,'1d','close'),skip_paused=True)
        #转化 nan 值为 0
        where_are_nan = np.isnan(close_price)
        where_are_inf = np.isinf(close_price)
        close_price[where_are_nan] = 0
        close_price[where_are_inf] = 0
        #计算双均线
        MA5=close_price['close'][-5:].mean()
        MA10=close_price['close'][-10:].mean()
        ma5=close_price['close'][-6:-1].mean()
        ma10=close_price['close'][-11:-1].mean()
        #用字典保存收盘价数据
        price_array={}

```

```

for i in range(0,M+num):
    price_array[i]=close_price['close'][i:i+N]
#TR=收盘价的 N 日指数移动平均
TR={}
for i in range(0,M+num):
    TR[i]=np.mean(price_array[i])
#TRIX=(TR-昨日 TR)/昨日 TR*100
TRIX={}
for i in range(1,M+num):
    if TR[i]==0:
        TRIX[i]=0
        continue
    TRIX[i]=(TR[i]-TR[i-1])/TR[i]*100
#MATRIX=TRIX 的 M 日简单移动平均
MATRIX={}
for i in range(0,num):
    TRIX_sum=0
    for j in range(1,M):
        TRIX_sum=TRIX_sum+TRIX[i+j]
    MATRIX[i]=TRIX_sum/M
#取昨日收盘价
current_price=data[security].price
#判断长期趋势
length=0
for i in range(0,num-1):
    #判断某日加速增长或加速下跌
    if TRIX[M+i]>MATRIX[i]:
        length=length+1
    else:
        length=length-1
#累计 num 天， 加速上涨的天数大于下降， 且昨日涨幅均值高于之前
if length>0 and MATRIX[num-1]>MATRIX[0]:
    #记为长期趋势为上升
    trend[security]='up'
elif length<0 and MATRIX[num-1]<MATRIX[0]:
    trend[security]='down'
#当长期趋势为上升时， 买入即可
if trend[security]=='up':
    order_value(security,cash)
#长期趋势为下降时
elif trend[security]=='down' and security in context.portfolio.positions:
    #五日均线低于十日均线
    if MA5<MA10:
        order_target(security,0)

```

```
#均线金叉，短期将有一个上涨，买入  
elif MA5>MA10 and ma5<ma10 :  
    order_value(security,cash)
```