

法律声明

本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。



加班主任入群

《进阶！量化交易实战：迭代式的量化策略研发》第5期

第4课：趋势型策略的设计与实现 主 讲：许银涛

内容介绍



趋势型策略原理以及股票池机制的设计

趋势型择时信号的设计方法与编程实现

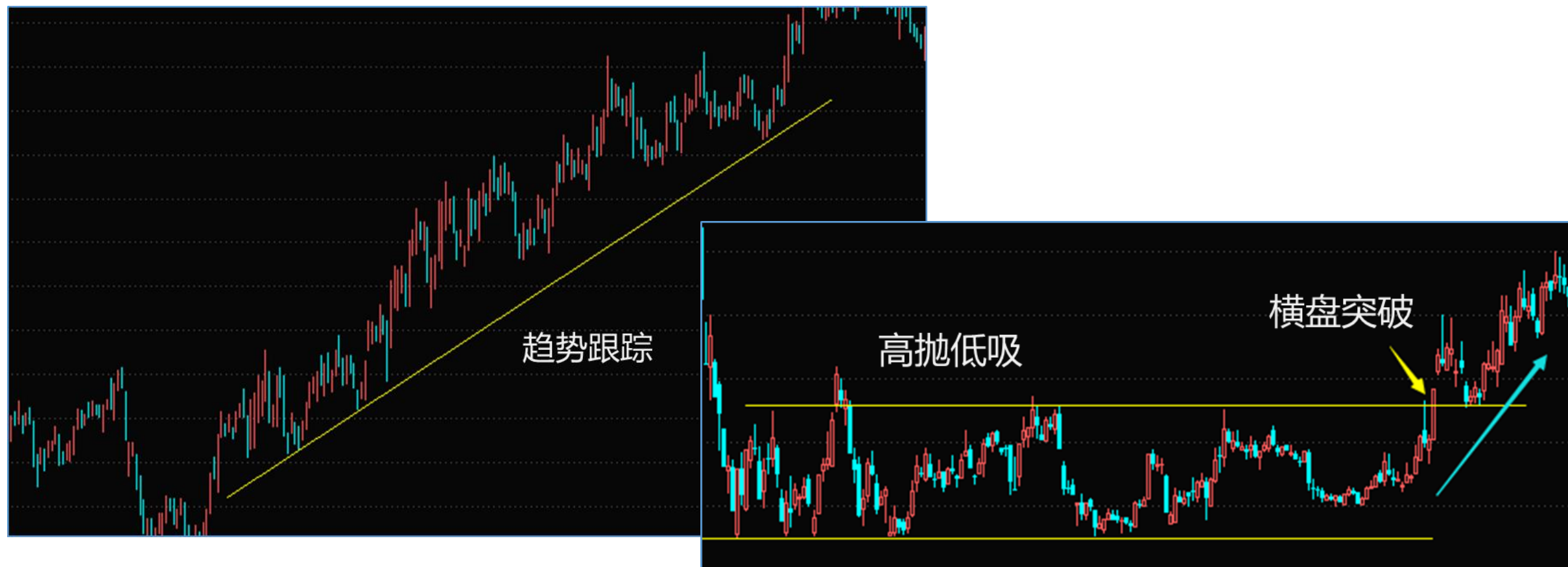
编写一个趋势策略并做回测与性能评估

优化趋势型策略的思路及方法

复习一下

趋势型策略原理以及股票池机制的设计

常见交易策略类型



趋势与三个关系

供需关系

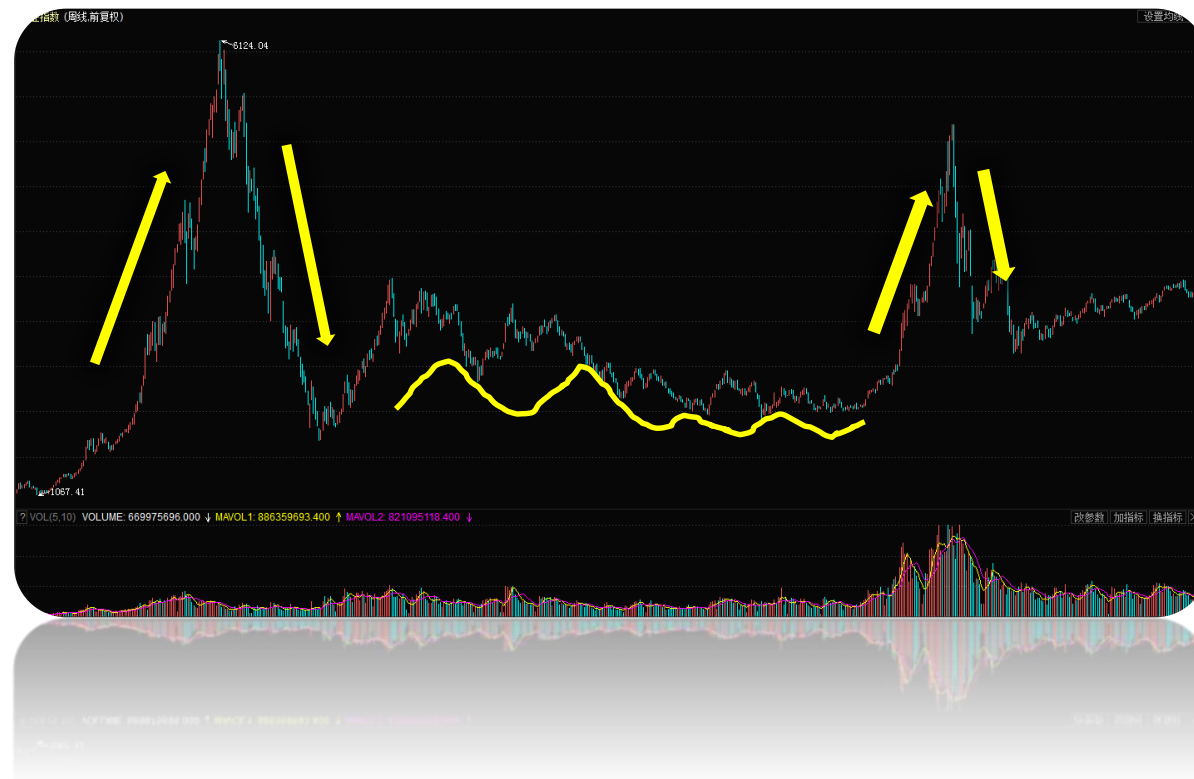
- 趋势的动力来自于供需关系的不平衡

因果关系

- 趋势形成之前需要准备过程

努力与结果

- 成交量的增长没有使价格大幅增长，这是走势停止行为



什么级别的趋势



更小的级别



股票池机制 – 温故知新

□ 选股条件

- 剔除总市值排名最小的10%的股票
- 剔除PE(TTM) 小于0或大于100的股票
- 剔除ST股票
- 取25日跌幅前10%的股票

□ 再平衡周期

- 25个交易日

□ 容量

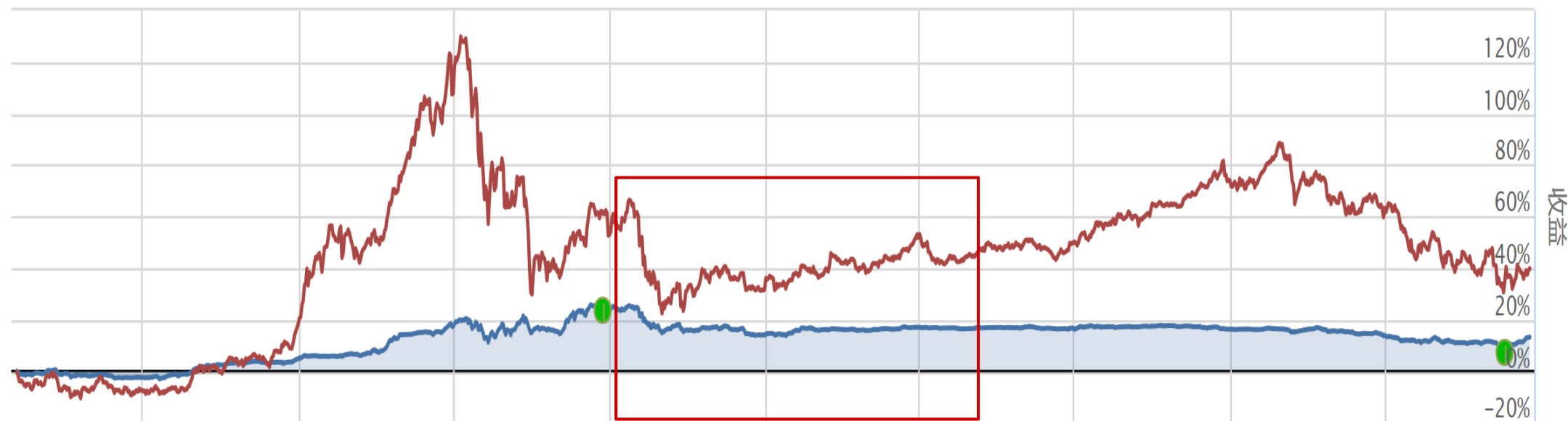
- 无限制

股票池检验

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤 ?	其他指标
13.21%	2.64%	39.81%	-0.020	0.189	-0.225	0.528	1.636	12.947%	>

缩放: 1个月 1年 全部 ■ 策略收益 ■ 基准收益 ■ 超额收益 ● 普通轴 ● 对数轴 □ 超额收益 ?

时间: 2013-12-30 - 2018-11-16



启动 延续 转折

趋势型择时信号的设计方法与编程实现

均线、MACD

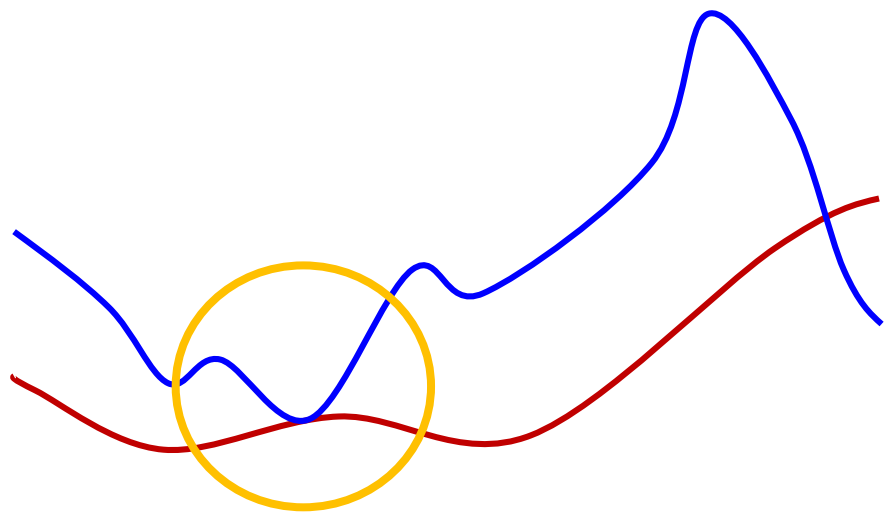


均线系统



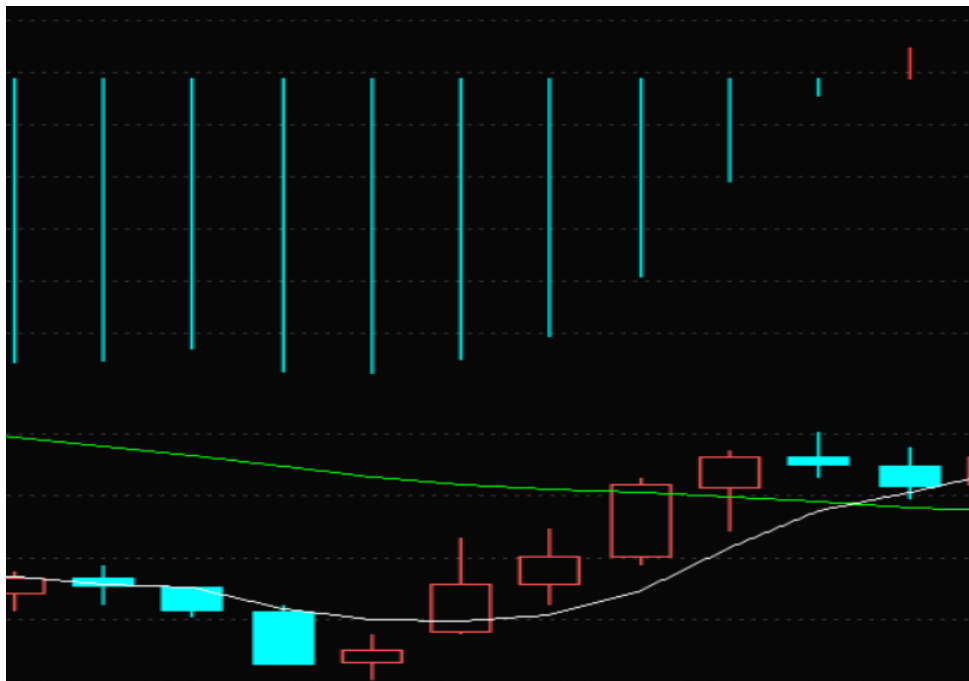
金叉 死叉

```
def cross(short_mean, long_mean):  
    """  
    判断短时均线和长时均线的关系。  
  
    Args:  
        short_mean 短时均线，长度不应小于3  
        long_mean 长时均线，长度不应小于3。  
  
    Returns:  
        1 短时均线上穿长时均线  
        0 短时均线和长时均线未发生交叉  
        -1 短时均线下穿长时均线  
    """  
    delta = short_mean[-3:] - long_mean[-3:]  
    if (delta[-1] > 0) and ((delta[-2] < 0) or ((delta[-2] == 0) and (delta[-3] < 0))):  
        return 1  
    elif (delta[-1] < 0) and ((delta[-2] > 0) or ((delta[-2] == 0) and (delta[-3] > 0))):  
        return -1  
    return 0
```



金叉

1. $\text{delta}[-1] > 0$ and $\text{delta}[-2] < 0$



2. $\text{delta}[-1] > 0$ and $\text{delta}[-2] == 0$ and $\text{delta}[-3] < 0$



3. $\text{delta}[-1] > 0$ and $\text{delta}[-2] == 0$ and $\text{delta}[-3] == 0$?

背离



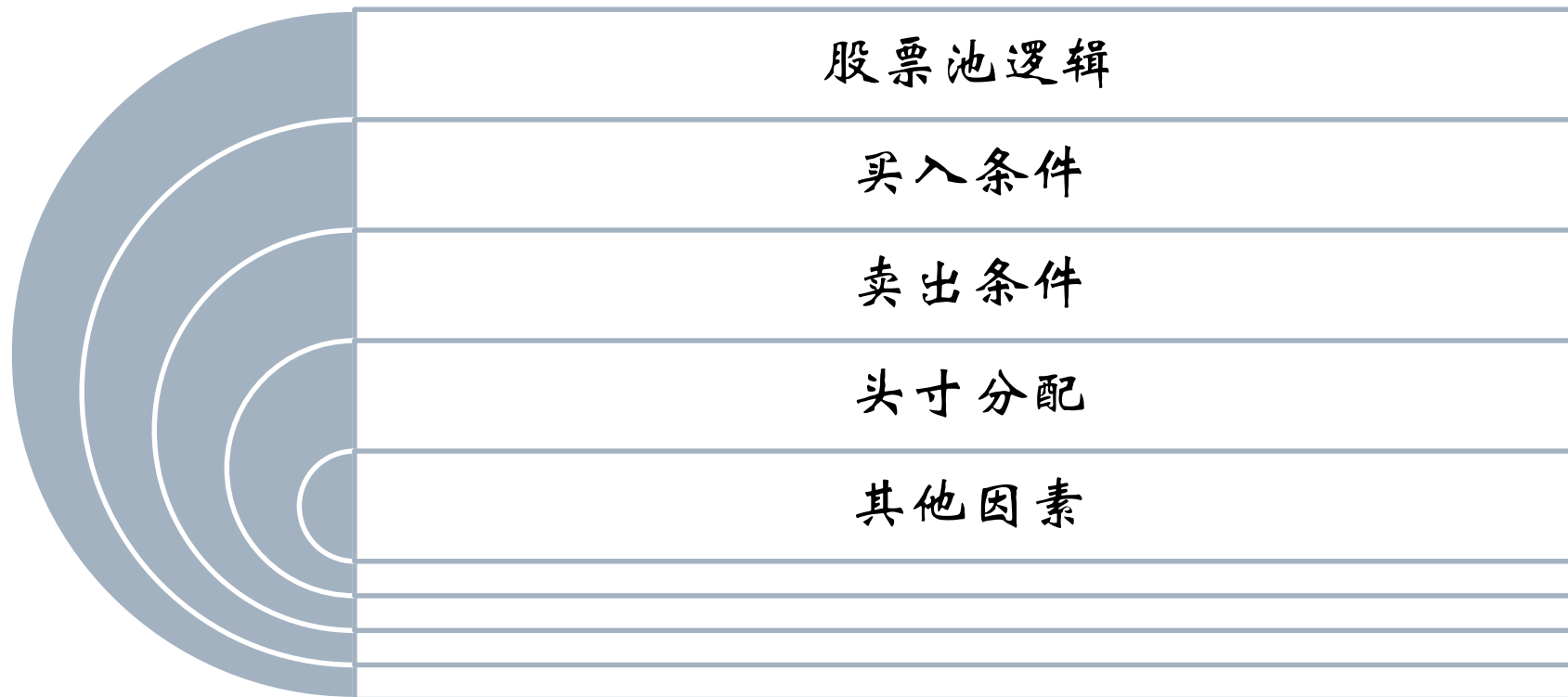
鱼与熊掌不可兼得

- ☐ 大级别背离比小级别背离稳定吗?
- ☐ 大级别背离比小级别背离滞后吗?
- ☐ 稳定与及时哪个更重要?
- ☐ 错了怎么办?
- ☐ 对了怎么出场?

理想很丰满 现实很骨感

编写一个趋势策略并做回测与性能评估

策略描述



股票池逻辑

□ 选股条件

- 剔除总市值排名最小的10%的股票
- 剔除PE(TTM) 小于0或大于100的股票
- 剔除ST股票
- 取25日跌幅前10%的股票

□ 再平衡周期

- 25个交易日
- Attention: 再平衡时不进行买卖操作!

□ 容量

- 无限制

买入条件

□ 条件一：

- 15分钟bar, MA5上穿MA30

□ 条件二：

- 15分钟bar, MACD底背离 (DIF vs. Close)

或者

- 5分钟bar, MACD连续底背离



卖出条件

- 对应于买入条件一：
 - 15分钟bar, MA5下穿MA30
- 对应与买入条件二：
 - 15分钟bar, MACD顶背离
 - 或者
 - 5分钟bar, MACD连续顶背离

头寸分配

- 均仓
- 根据历史不完全统计，采用一个相对“不激进”的均仓方式

其他因素

- ☐ 要不要止损？
- ☐ 要不要止盈？
- ☐ 要不要加仓？
- ☐ 有没有可改进的逻辑？

代码包

□ 买卖条件一：均线系统

- 股票池-交易信号均线.py

□ 买卖信号二：MACD背离系统

- 股票池-交易信号MACD.py

□ 运行环境：一创聚宽 <https://ycjq.95358.com/>

□ 依赖：

- `from PL_jukuan_macd_signal import *` # 导入自定义的，用于信号检测的module
- `from PL_signal_statistics import *` # 导入用于统计信号胜率和赔率的module

休息5分钟

买卖条件一：均线系统（1）

```
# 导入函数库
import jqdata
import pandas as pd
import numpy as np
import math
import talib as tl

# 两次处理交易逻辑的窗口大小
TRADE_BAR_DURATION = 15
# 操作时的分钟线级别
UNIT = str(TRADE_BAR_DURATION) + 'm'
# 股票池计算涨跌幅的窗口大小
CHANGE_PCT_DAY_NUMBER = 25
# 更新股票池的间隔天数
CHANGE_STOCK_POOL_DAY_NUMBER = 25

# 买卖信号的长时均线窗口大小
LONG_MEAN = 30
# 买卖信号的短时均线窗口大小
SHORT_MEAN = 5
# 标的调整出股票池后是否卖出
CLOSE_POSITION = False
```



买卖条件一：均线系统（2）

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)
    # 过滤掉order系列API产生的比error级别低的log
    log.set_level('order', 'error')

    ### 股票相关设定 ###
    # 设定滑点为0
    set_slippage(FixedSlippage(0))
    # 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')

    init_global(context)

    # 开盘前运行
    run_daily(before_market_open, time='before_open', reference_security='000300.XSHG')
    # 交易
    run_daily(trade, time='every_bar', reference_security='000300.XSHG')
    # 收盘后运行
    run_daily(after_market_close, time='after_close', reference_security='000300.XSHG')
```

买卖条件一：均线系统（3）

```
def init_global(context):  
    '''  
    初始化全局变量  
    '''  
    # 距上一次股票池更新的天数  
    g.stock_pool_update_day = 0  
    # 股票池，股票代码  
    g.stock_pool = []  
    # 距离上一次处理交易逻辑的bar的个数  
    g.bar_number = 0  
    # 每只标的需要买入的头寸  
    g.position = context.portfolio  
  
def trade(context):  
    '''  
    交易函数  
    '''  
    # 调用平仓处理逻辑。  
    if CLOSE_POSITION:  
        close_position(context)  
  
    # 间隔 TRADE_BAR_DURATION 调用一次买入函数  
    if g.bar_number % TRADE_BAR_DURATION == 0:  
        buy(context)    # 建仓  
        sell(context)  
    g.bar_number = (g.bar_number + 1) % TRADE_BAR_DURATION  
    pass
```

买卖条件一：均线系统（4）

```
def after_market_close(context):  
    '''  
    收盘后处理  
  
    1. 更新股票池  
    2. 更新股票池后，获取需要新买入的标的列表  
    '''  
    if g.stock_pool_update_day % CHANGE_STOCK_POOL_DAY_NUMBER == 0:  
        # 更新股票池  
        stock_pool(context)  
    g.stock_pool_update_day = (g.stock_pool_update_day + 1) % CHANGE_STOCK_POOL_DAY_NUMBER  
    record(pos=(context.portfolio.positions_value / context.portfolio.total_value * 100))  
    pass  
  
def close_position(context):  
    '''  
    平仓逻辑，当持仓标的不在股票池中时，平仓该标的  
    '''  
    for code in context.portfolio.positions.keys():  
        if code not in g.stock_pool:  
            if is_low_limit(code):  
                continue  
            # 标的已经不在股票池中尝试卖出该标的的股票  
            order_ = order_target(security=code, amount=0)  
            if order_ is not None and order_.filled:  
                log.info("交易 卖出 平仓", code, order_.filled)
```

买卖条件一：均线系统（5）

```
def buy(context):  
    """  
    买入逻辑。  
    股票池中的标的在发生短时均线上穿长时均线时买入  
  
    注意：  
    涨停无法买入  
    停牌无法买入  
    已经持仓的股票无法买入  
    """  
    for code in g.stock_pool:  
        if code in context.portfolio.positions.keys():  
            continue  
        current_data = get_current_data()[code]  
        if current_data == None:  
            continue  
        if is_high_limit(code):  
            continue  
        count = max(LONG_MEAN, SHORT_MEAN) + 3  
        close_data = attribute_history(security=code, count=count, unit=UNIT, fields=['close'], skip_paused=True, df=True, fq='pre')['close']  
        if (list(np.isnan(close_data)).count(True) > 0) or (len(list(close_data)) < count):  
            continue  
  
        # 长时均线  
        long_mean = pd.rolling_mean(close_data, LONG_MEAN)  
        # 短时均线  
        short_mean = pd.rolling_mean(close_data, SHORT_MEAN)  
  
        # 短时均线上穿长时均线时买入  
        if (cross(short_mean, long_mean) > 0):  
            order_ = order_value(security=code, value=g.position)  
            if order_ is not None and order_.filled > 0:  
                log.info("交易 买入", code, "买入仓位", g.position, "买入的股数", order_.filled)  
    pass
```

买卖条件一：均线系统（6）

```
def cross(short_mean, long_mean):  
    """  
    判断短时均线和长时均线的关系。  
  
    Args:  
        short_mean 短时均线，长度不应小于3  
        long_mean 长时均线，长度不应小于3。  
  
    Returns:  
        1 短时均线上穿长时均线  
        0 短时均线和长时均线未发生交叉  
        -1 短时均线下穿长时均线  
    """  
    delta = short_mean[-3:] - long_mean[-3:]  
    if (delta[-1] > 0) and ((delta[-2] < 0) or ((delta[-2] == 0) and (delta[-3] < 0))):  
        return 1  
    elif (delta[-1] < 0) and ((delta[-2] > 0) or ((delta[-2] == 0) and (delta[-3] > 0))):  
        return -1  
    return 0
```


买卖条件二：MACD背离系统（1）

```
def trade(context):  
    """  
    交易函数  
    """  
    # 调用平仓处理逻辑。  
    if CLOSE_POSITION:  
        close_position(context)  
  
    if g.macd_cache_long is not None:  
        # 15分钟级别交易  
        if g.bar_number % LONG_TRADE_BAR_DURATION == 0:  
            update_macd_cache(context, LONG_UNIT)  
            buy(context, LONG_UNIT) # 买入  
            sell(context, LONG_UNIT) # 卖出  
        # 5分钟级别交易  
    if g.macd_cache_short is not None:  
        if g.bar_number % SHORT_TRADE_BAR_DURATION == 0:  
            update_macd_cache(context, SHORT_UNIT)  
            buy(context, SHORT_UNIT) # 买入  
            sell(context, SHORT_UNIT) # 卖出  
    g.bar_number = g.bar_number + 1  
    pass
```

买卖条件二：MACD背离系统（2）

```
is_sell = False
sell_type = None
if unit == LONG_UNIT:
    # 15分钟级别交易顶背离卖出
    if current_tm in g.top_divergence_long[code].keys():
        is_sell = True
        sell_type = "LONG "
elif unit == SHORT_UNIT:
    # 5分钟级别交易 连续顶背离卖出
    if current_tm in g.top_divergence_short[code].keys():
        # 发生了5分钟级别的顶背离，需要判断是否发生了连续的顶背离
        for i in range(len(g.top_divergence_short[code][current_tm])):
            item_obj = g.top_divergence_short[code][current_tm][i]
            pre_dif_limit_tm = item_obj['pre_dif_limit_tm']

            for time_key in g.top_divergence_short[code].keys():
                if is_sell:
                    break
                for j in range(len(g.top_divergence_short[code][time_key])):
                    last_dif_limit_tm = g.top_divergence_short[code][time_key][j]
                    if pre_dif_limit_tm == last_dif_limit_tm:
                        is_sell = True
                        sell_type = "SHORT"
                        break
if is_sell:
    order_ = order_target(security=code, amount=0)
    if (order_ is not None) and (order_.filled > 0):
        log.info("交易 卖出", code, sell_type, "成交均价", order_.price, "卖出的股数", order_.filled, "平均成本", order_.avg_cost)
```

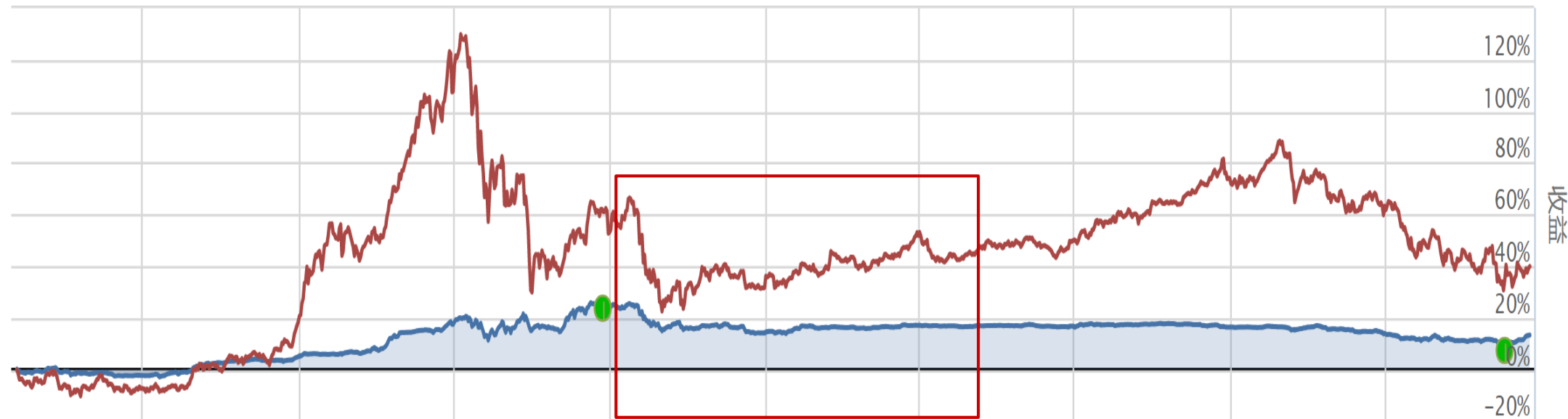
是提高胜率 还是提高赔率

优化趋势型策略的思路及方法

股票池检验

缩放: 1个月 1年 全部 ■ 策略收益 ■ 基准收益 ■ 超额收益 ● 普通轴 ● 对数轴 □ 超额收益 ?

时间: 2013-12-30 - 2018-11-16



均线系统回测

收益概述

策略收益

-6.68%

策略年化收益

-6.83%

基准收益

-11.28%

Alpha

-0.025

Beta

0.536

Sharpe

-0.733

胜率

0.374

盈亏比

1.021

最大回撤

11.48%

其他指标

缩放: 1个月 1年 全部

全部

策略收益

基准收益

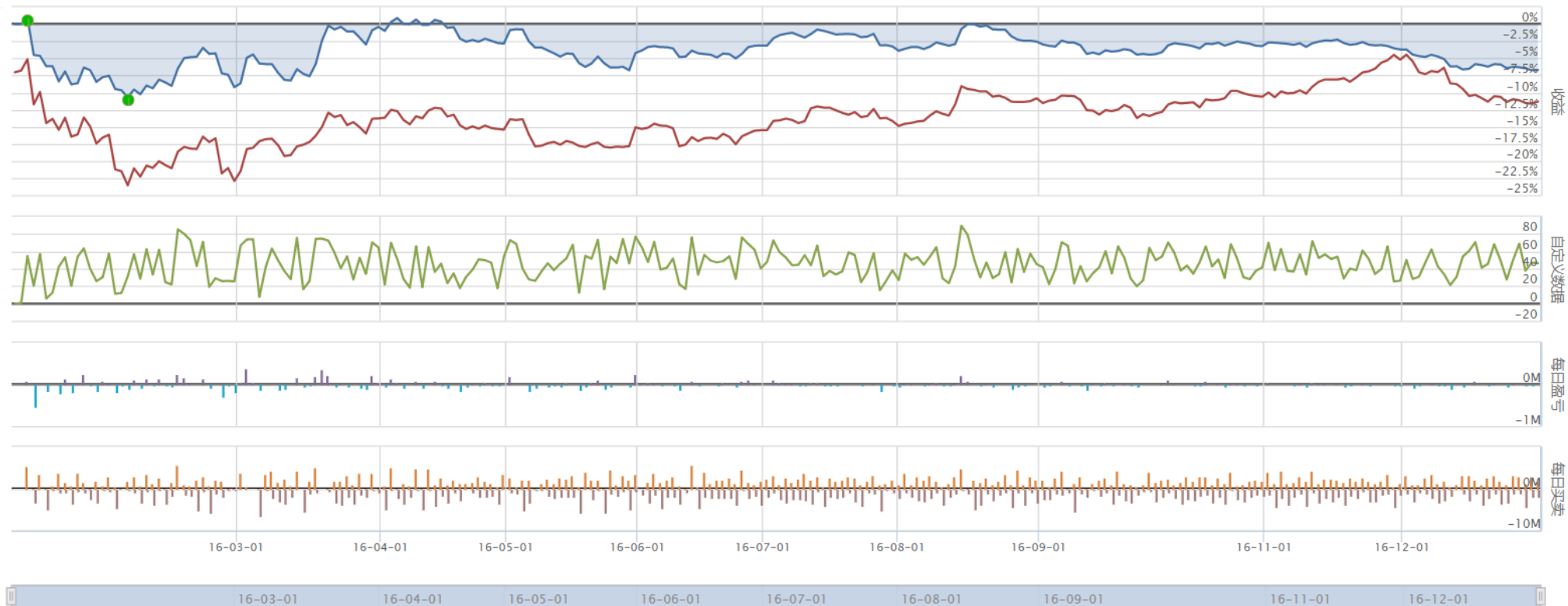
超额收益

普通轴

对数轴

超额收益

时间: 2016-01-04 - 2016-12-30



MACD背离系统回测

收益概述

策略收益
9.92%

策略年化收益
10.18%

基准收益
-11.28%

Alpha
0.133

Beta
0.459

Sharpe
0.439

胜率
0.665

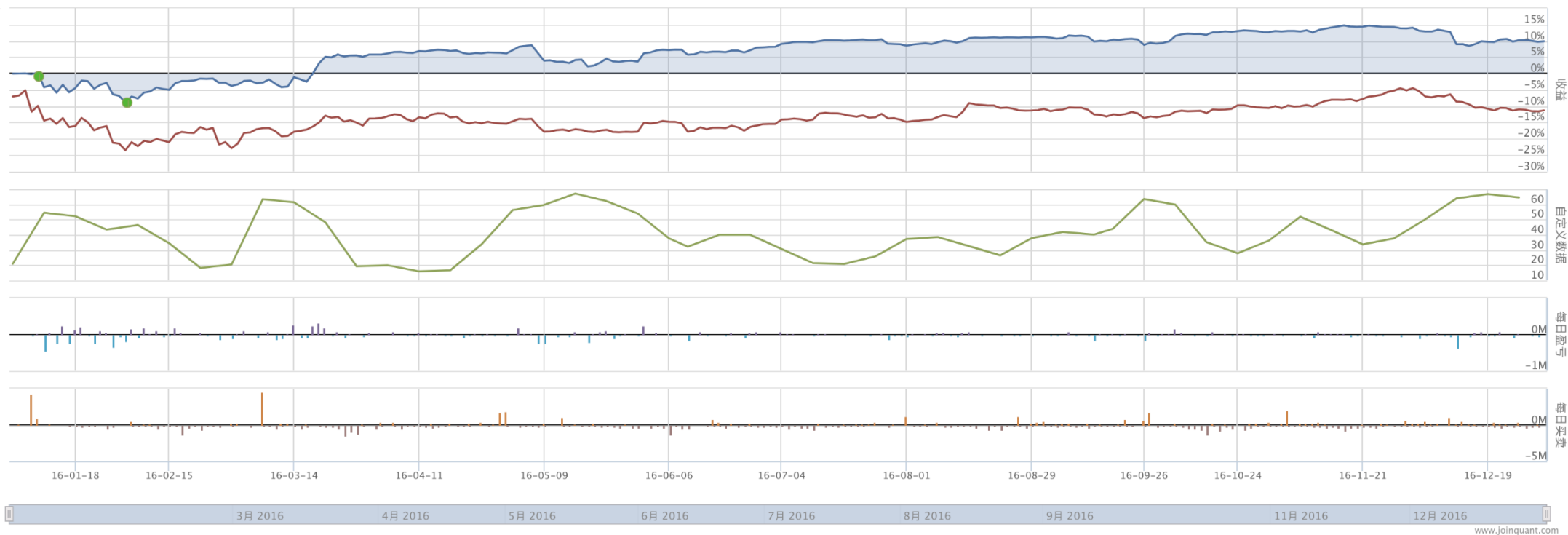
盈亏比
2.800

最大回撤
9.042%

其他指标

缩放: 1个月 1年 全部 ■ 策略收益 ■ 基准收益 ■ 超额收益 Powered by joinquant.com ● 普通轴 ○ 对数轴 □ 超额收益

时间: 2016-01-04 - 2016-12-30



收益 vs. 回撤

- ☐ 是没抓到趋势吗？
- ☐ 是没有锁定收益吗？
- ☐ 是没有及时止损吗？
- ☐ 是信号太慢了吗？
- ☐ 是信号太不准了吗？

胜率赔率决定了期望收益，但还差什么？

交易成本的影响

持仓周期、资金利用率与年化收益

交易机会的多少与交易频次

资金管理与风险控制

课后作业

- 理解课程代码包，通过调整参数进行更多回测
- 思考：想要提高趋势型策略的赔率，怎样从退出和头寸上入手

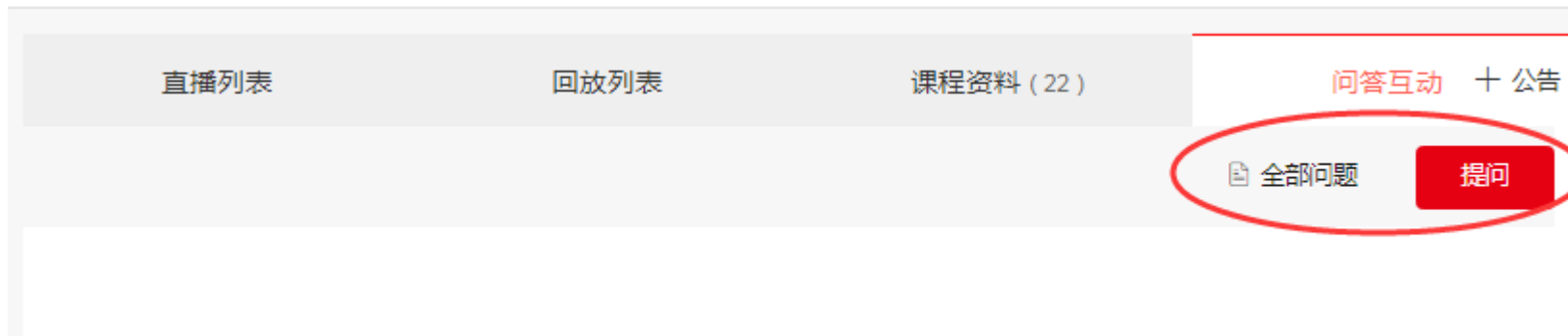
后续课程

□ 量化体系中的风险控制

问答互动

在所报课的课程页面，

- 1、点击“全部问题”显示本课程所有学员提问的问题。
- 2、点击“提问”即可向该课程的老师 and 助教提问问题。



联系我们

小象学院：互联网新技术在线教育领航者

— 微信公众号：**小象学院**



THANKS