



实证技术分析

Evidence-Based Technical Analysis



Bush

CONTENTS

PROFESSIONAL · LEADING · VALUE-CREATING

▶ PART 1

基于技术分析的量化投资

▶ PART 2

基于技术指标的量化投资

▶ PART 3

基于通道技术的量化投资

专业来自101%的投入!



基于技术分析的量化投资



金程教育
GOLDEN FUTURE

- 技术分析简介
- 实证技术分析
- 技术分析的量化投资之路



- 以股票价格涨跌的直观行为表现作为主要研究对象，以预测股价波动形态和趋势为主要目的，从股价变化的K线图表及技术指标入手，对股票市场波动规律进行分析的方法总和。
- 技术分析的目的不仅是对金融市场数据中的重复模式进行研究，还要对市场未来的价格走势做出预测。技术分析包括很多种分析方法、形态、信号、指标以及交易策略。

4-4

专业来自101%的投入!



- Prices are determined by the interaction of supply and demand.
- Only participants who actually trade affect prices, and better-informed participants tend to trade in greater volume.
- Price and volume reflect the collective behavior of buyers and sellers.
- Market prices reflect both rational and irrational investor behavior.
- Investor behavior is reflected in trends and patterns that trend to repeat and can be identified and used for forecasting prices.

5-4

专业来自101%的投入!



- **Advantages of technical analysis:**
 - Actual price and volume data are observable.
 - Technical analysis itself is objective (although require subjective judgment), while much of the data used in fundamental analysis is subject to assumptions or restatements.
 - It can be applied to the prices of assets that do not produce future cash flows, such as commodities.
 - It can also be useful when financial statement fraud occurs.
- **Disadvantage:**
 - The usefulness is limited in markets where price and volume data might not truly reflect supply and demand, such as in illiquid markets and in markets that are subject to outside manipulation.
 - 大众接受的技术分析是通过形式各异的民间传说获得，而不是依靠客观的统计实证获得的

6-4

专业来自101%的投入!



- 传统技术分析主要依赖不能够明确定义的分析方法和图形形态。分析结果反映分析师对分析方法的个人理解。这有可能使得不同分析师运用同一种方法对一组市场数据进行分析，得到完全不同的结论。
- 主观的技术分析方法是不可检验的，因此其有效性无法对到实证检验。
- 技术分析必须发展为一种缜密的观察性科学，才能体现出这种理论的重要性。科学的方法是挖掘市场中有价值的知识，以及确定哪种技术分析具有预测力的唯一理性方法。这种方法称为实证技术分析。
- 实证技术分析以客观观察和统计推断（科学的方法）为基础，重新定义了被持续质疑的奇幻思维、容易上当受骗的痴迷者以及随机游走三者之间的关系。

7-4

专业来自101%的投入!



- 实证技术分析可以借助量化投资的方法实现。量化投资可以把技术分析应用到市场数据中，它所提供的信号和预测是非常清晰的，这就使得对历史数据以及判断绩效的准确程度方面成为可能。这种方法被称为回测测试。
- 对技术分析的方法进行回测测试是可以重复实现的。这种实验可以通过统计数据对盈利状况进行测试，并且对客观的方法进行反驳。回测测试可以发现哪种技术分析方法是有效的，哪种是无效的。

8-4

专业来自101%的投入!

CONTENTS

PROFESSIONAL · LEADING · VALUE-CREATING

▶ PART 1

基于技术分析的量化投资

▶ PART 2

基于技术指标的量化投资

▶ PART 3

基于通道技术的量化投资

专业来自101%的投入!



- 基于MACD指标的量化投资策略
- 基于WVAD指标的量化投资策略
- 基于RSI指标的量化投资策略
- 基于MFI指标的量化投资策略
- 基于CCI指标的量化投资策略

10-4

专业来自101%的投入!



基于MACD指标的量化投资策略



金程教育
GOLDEN FUTURE

- MACD指标的基本原理及操盘策略
- MACD指标的应用举例
- 基于MACD指标的量化投资策略的代码及注释
- 基于MACD指标的量化投资策略的回测结果
- 本节程序学习：数据加载

11-4

专业来自101%的投入!



基于MACD指标的量化投资策略的基本原理及操作步骤



金程教育
GOLDEN FUTURE

- 基本原理：
 - MACD称为指数平滑移动平均线，是从双指数移动平均线发展而来的，由两线一柱组合起来形成，快速线为DIF，慢速线为DEA，柱状图为MACD。
 - 快的指数移动平均线（EMA12）减去慢的指数移动平均线（EMA26）得到快线DIF，慢速线DEA是由DIF的9日指数加权移动平均值。MACD柱由快速线DIF和慢速线DEA做差得到。
 - MACD可以反映股票近期价格走势的能量和变化强度。当MACD从负数转向正数，是买的信号。当MACD从正数转向负数，是卖的信号。当MACD以大角度变化，表示快的移动平均线和慢的移动平均线的差距非常迅速的拉开，代表了一个市场大趋势的转变。
- 操作策略：
 - 买入规则：DIF上穿DEA，买入股票
 - 卖出规则：DIF下穿DEA，卖出股票

12-4

专业来自101%的投入!



13-4

专业来自101%的投入!



```
#加载函数库
import talib #Technical Analysis Library
import numpy as np

#初始化回测环境
start = '20140701' # 回测起始时间 注：支持两种日期表述形式 ('2015-01-01', '20150101')
end = '20150501' # 回测结束时间
benchmark = 'HS300' # 策略参考标准
universe = set_universe('HS300', start) # 证券池：可供选择的股票的范围；注：取start日的成分股防止survival bias
freq = 'd' # 用日线回测的策略
refresh_rate = 1 # 每天调一次仓，即每个交易日都会运行第三部分的handle_data函数

#初始化投资者（账户）参数
#accounts为字典类型，代表投资者所有的账户，而字典中每一个键代表一个账户，而每一个键对应的值为该账户的初始情况，如本程序中的键为fantasy_account（股票账户），值为相应配置
accounts = {
    'fantasy_account': AccountConfig(account_type='security', capital_base=1000000) #初始化投资者的股票账户：投资品种为股票，初始投资金额为1千万
}

#初始化策略参数
max_history_window = 60 # 计算指数平滑所考虑的时间窗口为60天
fastperiod = 12 # 快线周期
slowperiod = 26 # 慢线周期
signalperiod = 9 # Signal平滑周期

#初始化回测环境，指明创建账户的工作，全局只运行一次
def initialize(context):
    return

#handle_data函数是策略的核心函数，包含了所有策略算法的内容，包括数据获取，交易信号生成，订单委托等逻辑。
#handle_data函数无论是回测还是模拟交易场景，这个函数会根据回测频率 freq 的设置被调用。当freq='d'时，每天被调用一次，当freq='m'时，每分钟被调用一次。
def handle_data(context):
    timing(context)
```

14-4

专业来自101%的投入!



```
def timing_Basic_MACD(context):
    ##basic_macd算法在DIF上穿DEA时买入，在DIF下穿DEA时卖出 （操盘规则：预测）
    ##基于仓位管理的需要，每只股票最大投资额为100000元 （仓位管理：风控）

    #数据获取（通用部分）：投资者账户，可供投资股票，价格数据，持仓数据，账户金额数据
    account = context.get_account('fantasy_account')
    current_universe = context.get_universe(exclude_halt=True)
    history = context.history(current_universe, 'closePrice', max_history_window, rtype='array')
    security_position = account.get_positions()
    cash = account.cash

    #择时策略部分：获取当前时点的buylist
    buylist = []
    for sec in current_universe:
        close = history[sec]['closePrice']
        DIF, DEA, MACD = talib.MACD(close, fastperiod=fastperiod, slowperiod=slowperiod, signalperiod=signalperiod) #talib提供的MACD计算函数，计算DIF,DEA以及MACD的取值
        if MACD[-2]<0 and MACD[-1]>0 and sec not in security_position: # DIF上穿DEA，且无持仓
            #将股票sec加入买入列表
            buylist.append(sec)
        elif MACD[-2]>0 and MACD[-1]<0 and sec in security_position: # DIF下穿DEA，且有持仓
            #全部卖出
            order_to(sec, 0)
            cash += security_position[sec].amount * context.current_price(sec) # 估计卖出股票后的账户金额，注：context.current_price(sec)是获取sec股票当前价格
            # security_position[sec].amount获取购买股票的数量

    #交易执行部分
    d = min(len(buylist), int(cash) // 100000) # 可以买入的股票数量，如果资金不够，只买入部分
    for sec in buylist[d:]:
        order(sec, 100000 / context.current_price(sec)) #基于仓位管理的需要，每只股票最大投资额为100000元

def timing(context):
    timing_Basic_MACD(context) #基于基础MACD算法的择时策略
```

15-4

专业来自101%的投入!



基于MACD指标的量化投资策略的回测结果



金程教育
GOLDEN FUTURE

年化收益率 94.0% 基准年化收益率 161.9% 阿尔法 -9.3% 贝塔 0.63 夏普比率 5.03 收益波动率 17.9% 信息比率 -2.36 最大回撤 5.4% 换手率 11.26

回测详情

开始交易



16-4

专业来自101%的投入!



本节程序学习：数据加载



金程教育
GOLDEN FUTURE

- 优矿的数据加载机制

```
#数据获取（通用部分）：投资者账户，可供投资股票，价格数据，持仓数据，账户金额数据
account = context.get_account('fantasy_account')
current_universe = context.get_universe(exclude_halt=True)
history = context.history(current_universe, 'closePrice', max_history_window, rtype='array')
security_position = account.get_positions()
cash = account.cash

#获取投资者的股票账户 (fantasy_account)
#获取当前除停牌外的所有可供投资股票 (universe)
#拿过去60个交易日的收盘价来估算MACD
#字典型数据，上一k线结束后的有效证券头寸，即持仓数量大于0的证券及其头寸
#获取股票账户可用于投资的现金额度
```

17-4

专业来自101%的投入!



基于WVAD指标的量化投资策略



金程教育
GOLDEN FUTURE

- WVAD指标的基本原理及操盘策略
- WVAD指标的应用举例
- 基于WVAD指标的量化投资策略的代码及注释
- 基于WVAD指标的量化投资策略的回测结果
- 本节程序学习：信号机制

18-4

专业来自101%的投入!



➤ 基本原理：

- WVAD (William's variable accumulation distribution)，是一种将成交量加权的量价指标。用于测量从开盘价至收盘价期间，**买卖双方各自爆发力的程度**。
- 该理论精髓在于重视一天中开盘到收盘之间的价位，而将此区域之上的价位视为压力，区域之下的价位视为支撑，求取此区域占当天总波动的百分比，以便测量当天的成交量中，有多少属于此区域。成为实际有意义的交易量。
- 如果区域之上的压力较大，将促使WVAD变成负值，代表卖方的实力强大，此时应该卖出持股。如果区域之下的支撑较大，将促使WVAD变成正值，代表买方的实力雄厚，此时应该买进股票。WVAD正负之间，由于模拟测试所选用的周期相当长，测试结果也以长周期成绩较佳。因此，长期投资者适合使用，如同EMV使用法则一样，应该在一定的投资期限内，不断的根据WVAD讯号将交易买卖，以求得统计盈亏概率的成果。
- WVAD的计算公式如下

$$WVAD = \sum \frac{Close_t - Open_t}{High_t - Low_t} * Volume_t$$

➤ 操作策略：

- 买入规则：DIF上穿DEA，买入股票
- 卖出规则：DIF下穿DEA，卖出股票

19-4

专业来自101%的投入！



WVAD指标的应用举例



20-4

专业来自101%的投入！



基于WVAD指标的量化投资策略的代码及注释



```
#加载函数库：注：由于该策略使用优矿提供的signal机制，所以不用引入talib
import numpy as np
import pandas as pd

#初始化回测环境
start = '20160501' # 回测起始时间 注：支持两种日期表述形式 ('2015-01-01', '20150101')
end = '20161101' # 回测结束时间
benchmark = 'HS300' # 策略参考标准
universe = set_universe('HS300', start) # 证券池：可供选择的股票的范围；注：取start日的成分股防止survival bias
freq = 'd' # 用日线回测的策略
refresh_rate = 1 # 每天调一次仓，即每个交易日都会运行第三部分的handle_data函数

#初始化投资者（账户）参数
#accounts为字典类型，代表投资者所有的账户，而字典中每一个键代表一个账户，而每一个键对应的值为该账户的初始情况，如本程序中的键为fantasy_account（股票账户），值为相应配置
accounts = {
    'fantasy_account': AccountConfig(account_type='security', capital_base=10000000) #初始化投资者的股票账户：投资品种为股票，初始投资金额为1千万
}

#初始化策略参数
buy_threshold = 0 # 买入阈值
sell_threshold = 0 # 卖出阈值

#初始化回测环境，指明创建账户时的工作，全局只运行一次
#建立策略的wvad信号机制，默认向前遍历的时间参数是24天
def initialize(context):
    wvad = Signal('WVAD') #定义名为WVAD的信号，T默认取值为24天
    context.signal_generator = SignalGenerator(wvad) #产生信号，Quartz 框架会自动触发依赖图分析以及在回测区间内计算信号取值。

#handle_data函数是策略的核心函数，包含了所有策略算法的内容，包括数据获取，交易信号生成，订单委托等逻辑。
#handle_data函数无论是回测还是模拟交易场景，这个函数会根据回测频率 freq 的设置被调用。当freq='d'时，每天被调用一次，当freq='m'时，每分钟被调用一次。
def handle_data(context):
    #todo(context)
```

21-4

专业来自101%的投入！



基于WVAD指标的量化投资策略的代码及注释



金程教育
GOLDEN FUTURE

```
def timing_Basic_WVAD(context):
    ##basic_WVAD算法在WVAD>0时买入，在WVAD<0时卖出（操盘规则：预测）
    ##基于仓位管理的需要，每只股票最大投资额为100000元（仓位管理：风控）

    #数据获取（通用部分）：投资者账户，可供投资股票，持仓数据，账户金额数据；注：由于使用系统提供的信号机制，不必获取价格数据
    account = context.get_account('fantasy_account')
    current_universe = context.get_universe(exclude_halt=True)
    #history = context.history(current_universe, 'closePrice', max_history_window, rtype='array')
    security_position = account.get_positions()
    cash = account.cash

    #定时策略部分：获取当前时点的buylist
    buylist = []
    for sec in current_universe:
        if context.signal_result['WVAD'][sec] > buy_threshold and sec not in security_position:
            buylist.append(sec)
        elif context.signal_result['WVAD'][sec] < sell_threshold and sec in security_position:
            cash += security_position[sec].amount * context.current_price(sec)

    #交易执行部分
    d = min(len(buylist), int(cash) // 100000) # 可以买入的股票数量，如果资金不够，只买入部分
    for sec in buylist[:d]:
        order(sec, 100000 / context.current_price(sec)) # 基于仓位管理的需要，每只股票最大投资额为100000元

def timing(context):
    timing_Basic_WVAD(context) # 基于WVAD指标的基础算法
```

22-4

专业来自101%的投入!



基于WVAD指标的量化投资策略的回测结果

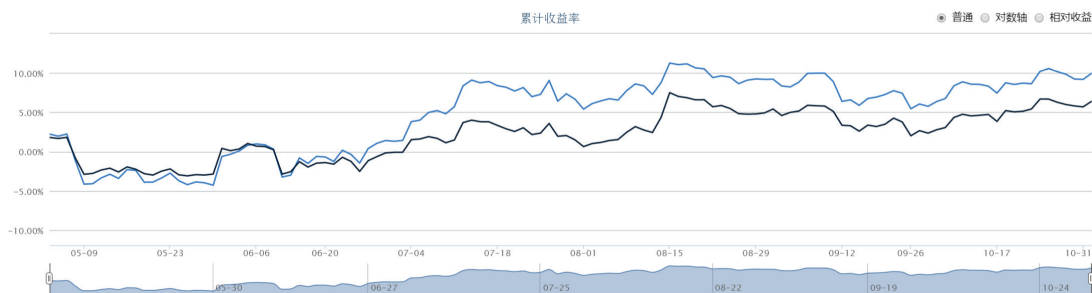


金程教育
GOLDEN FUTURE

年化收益率 21.5% 基准年化收益率 13.6% 阿尔法 5.8% 贝塔 1.16 夏普比率 1.28 收益波动率 16.8% 信息比率 1.32 最大回撤 6.4% 换手率 0.00

回测详情

开始交易



23-4

专业来自101%的投入!



本节程序学习：信号机制



金程教育
GOLDEN FUTURE

- 建立策略的wvad信号机制，默认向前遍历的时间参数是24天

```
wvad = Signal('WVAD') # 定义名为WVAD的信号，T默认取值为24天
context.signal_generator = SignalGenerator(wvad) # 产生信号，Quartz 框架会自动触发依赖分析以及回测区间内计算信号取值。
```

- 信号的使用

```
for sec in current_universe:
    if context.signal_result['WVAD'][sec] > buy_threshold and sec not in security_position:
        buylist.append(sec)
    elif context.signal_result['WVAD'][sec] < sell_threshold and sec in security_position:
        cash += security_position[sec].amount * context.current_price(sec)
```

遍历所有可供投资股票；注：如该策略执行了选股策略，该部分应遍历被选中股票
股票sec的WVAD信号的取值大于买入阈值（0），买方强势，且无持仓
股票sec的WVAD信号的取值小于卖出阈值（0），卖方强势，且有持仓
估计卖出股票后的账户金额，注：context.current_price(sec) 是获取sec股票当前价格

24-4

专业来自101%的投入!



- RSI指标的基本原理及操盘策略
- RSI指标的应用举例
- 基于RSI指标的量化投资策略的代码及注释
- 基于RSI指标的量化投资策略的回测结果
- 本节程序复习：交易指令

25-4

专业来自101%的投入!



- **基本原理：**
 - RSI (Relative Strength Index)，又称为相对强弱指数，由威尔斯·威尔德 (Welles Wilder)最早应用于期货买卖，后来人们发现在众多的图表技术分析中，是根据最近一段时间上涨和下跌幅度的比较得到的一种动量类指标，后应用于股市交易中，**主要用于识别资产对的超买超卖情况**
 - RSI的公式为：
$$RSI = 100 - 100 / (1 + RS)$$
 - ✓ 其中 RS = 一段时期内平均上涨幅度 / 一段时期内平均下跌幅度，一般采用14个交易日作为移动窗口
 - ✓ RSI的值域是0~100，一般认为低于30处于超卖期，高于70处于超买期
- **操作策略：**
 - 买入条件：RSI<30超卖, 买入股票
 - 卖出条件：RSI>70超买, 卖出股票

26-4

专业来自101%的投入!



27-4

专业来自101%的投入!



基于RSI指标的量化投资策略的代码及注释



金程教育
GOLDEN FUTURE

```
#加载函数库
import talib #Technical Analysis Library
import numpy as np

#初始化回测环境
start = '20140701' # 回测起始时间 注：支持两种日期表述形式 ('2015-01-01', '20150101')
end = '20150501' # 回测结束时间
benchmark = 'HS300' # 策略参考标准
universe = set_universe('HS300', start) # 证券池：可供选择的股票的范围；注：取start日的成分股防止survival bias
freq = 'd' # 用日线回测的策略
refresh_rate = 1 # 每天调一次仓，即每个交易日都会运行第三部分的handle_data函数

#初始化投资者（账户）参数
#accounts为字典类型，代表投资者所有的账户，而字典中每一个键代表一个账户，而每一个键对应的值为该账户的初始情况，如本程序中的键为fantasy_account（股票账户），值为相应配置
accounts = {
    'fantasy_account': AccountConfig(account_type='security', capital_base=10000000) #初始化投资者的股票账户：投资品种为股票，初始投资金额为1千万
}

#初始化策略参数
N = 14 # 移动窗口长度
buy_threshold = 30 # 买入阈值
sell_threshold = 70 # 卖出阈值

#初始化回测环境，指明创建账户时的工作，全局只运行一次
def initialize(context):
    return

#handle_data函数是策略的核心函数，包含了所有策略算法的内容，包括数据获取，交易信号生成，订单委托等逻辑。
#handle_data函数无论是回测还是模拟交易场景，这个函数会根据回测频率 freq 的设置被调用。当freq='d'时，每天被调用一次，当freq='m'时，每分钟被调用一次。
def handle_data(context):
    timing(context)
```

28-4

专业来自101%的投入！



基于RSI指标的量化投资策略的代码及注释



金程教育
GOLDEN FUTURE

```
def timing(context):
    timing_Basic_RSI(context) #基于基础RSI算法的择时策略

def timing_Basic_RSI(context):
    ##timing_Basic_RSI算法在RSI<30超卖时买入；在RSI>70超买时卖出 （操盘规则：预测）
    ##基于仓位管理的需要，每只股票最大投资额为100000元 （仓位管理：风控）

    #数据获取（通用部分）：投资者账户，可供投资股票，价格数据，持仓数据，账户金额数据
    account = context.get_account('fantasy_account')
    current_universe = context.get_universe(exclude_halt=True)
    history = context.history(current_universe, 'closePrice', N+1, rtype='array')
    security_position = account.get_positions()
    cash = account.cash

    #择时策略部分：获取当前时点的buylist
    buylist = []
    for sec in current_universe:
        close = history[sec]['closePrice']
        rsi = talib.RSI(close, timeperiod=N)[-1]
        if rsi < buy_threshold and sec not in security_position:
            buylist.append(sec)
        elif rsi > sell_threshold and sec in security_position:
            order_to(sec, 0)
            cash += security_position[sec].amount * context.current_price(sec)

    #交易执行部分
    d = min(len(buylist), int(cash // 100000)) # 可以买入的股票数量，如果资金不够，只买入部分
    for sec in buylist[:d]:
        order(sec, 100000 / context.current_price(sec)) #基于仓位管理的需要，每只股票最大投资额为100000元

    #获取投资者的股票账户（fantasy_account）
    #获取当前除停牌外的所有可供投资股票（universe）
    # 拿过去N+1个交易日的收盘价来估算RSI
    #字典型数据，上一K线结束后的有效证券头寸，即持仓数量大于0的证券及其头寸
    #获取股票账户可用于投资的现金额度

    #初始化购买股票列表
    #遍历所有可供投资股票；注：如该策略执行了选股策略，该部分应遍历被选中股票
    #获取股票sec过去max_history_window（60）天的收盘价
    #talib提供的RSI计算函数，计算参数为N天的RSI指标
    # RSI超买，且无持仓
    #将股票sec加入购买列表
    # RSI超买，且有持仓
    # 全部卖出
    # 估计卖出股票后的账户金额，注：context.current_price(sec) 是获取sec股票当前价格
    # security_position[sec].amount获取购买股票的数量
```

29-4

专业来自101%的投入！



基于RSI指标的量化投资策略的回测结果

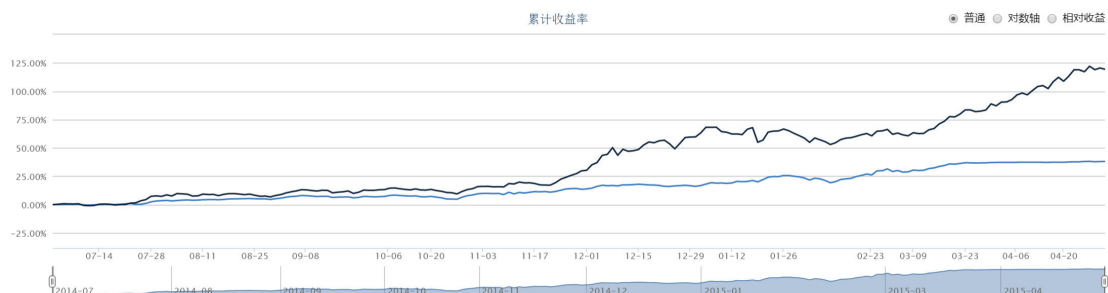


金程教育
GOLDEN FUTURE

年化收益率	基准年化收益率	阿尔法	贝塔	夏普比率	收益波动率	信息比率	最大回撤	换手率
48.5%	161.9%	4.9%	0.25	4.48	9.9%	-3.05	5.1%	3.71

回测详情

开始交易



30-4

专业来自101%的投入！



```
#3.程序复习：交易指令
order_to(sec, 0)
cash += security_position[sec].amount * context.current_price(sec)

# 全部卖出
# 估计卖出股票后的账户金额，注： context.current_price(sec) 是获取sec股票当前价格
# security_position[sec].amount获取购买股票的数量

#交易执行部分
d = min(len(buylist), int(cash) // 100000) # 可以买入的股票数量，如果资金不够，只买入部分
for sec in buylist[:d]:
    order(sec, 100000 / context.current_price(sec)) #基于仓位管理的需要，每只股票最大投资额为100000元
```

31-4

专业来自101%的投入!



- MFI指标的基本原理及操盘策略
- MFI指标的应用举例
- 基于MFI指标的量化投资策略的代码及注释
- 基于MFI指标的量化投资策略的回测结果
- 本期程序复习：证券池设定

32-4

专业来自101%的投入!



- 基本原理：
 - MFI (Money Flow Index) 是一种通过资金的流入与流出来预测超买和超卖的指标，其主要使用上涨的天数、下跌的天数、成交量增加幅度、成交量减少幅度等四项数据来进行计算。MFI越大，资金流入越大，所以当MFI > 80时，一般认为资金短期过热；反之，当MFI < 20时，一般认为资金短期冷却
 - MFI的计算公式如下：
 - ✓ $TP(\text{Typical Price}) = (\text{close} + \text{highest} + \text{lowest}) / 3$
 - ✓ 当日现金流 $MF = TP * \text{volume}$
 - ✓ $MR_t = \frac{\sum_{i=t-T+1}^t I(TP_i > TP_{i-1})MF_i}{\sum_{i=t-T+1}^t I(TP_i < TP_{i-1})MF_i}$
 - ✓ $MFI = 100 - 100 / (1 + MR)$
 - ✓ 若当日TP高于前一日TP，定义MF为正；反之则MF为负；特别地，若当日TP等于前一日TP，则当日MF被舍去。之后其中T一般取14
- 操作策略：
 - 买入条件：MFI<20超卖, 买入股票
 - 卖出条件：MFI>80超买, 卖出股票

33-4

专业来自101%的投入!



34-4

专业来自101%的投入!



基于MFI指标的量化投资策略的代码及注释



```
#加载函数库;注:由于该策略使用优矿提供的signal机制,所以不用引入talib
import numpy as np
import pandas as pd

#初始化回测环境
start = '20160201' # 回测起始时间 注:支持两种日期表述形式('2015-01-01','20150101')
end = '20170201' # 回测结束时间
benchmark = 'HS300' # 策略参考标准
universe = DynamicUniverse('HS300') # 证券池:可供选择的股票的范围;选择每一期的HS300股票组成每一期的证券池
freq = 'd' # 用日线回测的策略
refresh_rate = 1 # 每天调一次仓,即每个交易日都会运行第三部分的handle_data函数

#初始化投资者(账户)参数
#accounts为字典类型,代表投资者所有的账户,而字典中每一个键代表一个账户,而每一个键对应的值为该账户的初始情况,如本程序中的键为fantasy_account(股票账户),值为相应配置
accounts = {
    'fantasy_account': AccountConfig(account_type='security', capital_base=1000000) #初始化投资者的股票账户: 投资品种为股票,初始投资金额为1千万
}

#初始化策略参数
buy_threshold = 20 # 买入阈值
sell_threshold = 80 # 卖出阈值

#初始化回测环境,指明创建账户时的工作,全局只运行一次
#建立策略的CCI信号机制,选择向前遍历的时间参数是20天
def initialize(context):
    mfi = Signal('MFI') # 优矿的MFI的参数是14,即通过考察过去14天的资金流情况分析超买和超卖
    context.signal_generator = SignalGenerator(mfi) #产生信号cci20, Quartz 框架会自动触发依赖图分析以及在回测区间内计算信号取值。

#handle_data函数是策略的核心函数,包含了所有策略算法的内容,包括数据获取,交易信号生成,订单委托等逻辑。
#handle_data函数无论是回测还是模拟交易场景,这个函数会根据回测频率 freq 的设置被调用。当freq='d'时,每天被调用一次,当freq='m'时,每分钟被调用一次。
def handle_data(context):
    timing(context)
```

35-4

专业来自101%的投入!



基于MFI指标的量化投资策略的代码及注释



```
def timing_basic_mfi(context):
    ##basic_cci算法在mfi<20,超买时买入,在cci>80,超卖时卖出 (操盘规则:预测)
    ##基于仓位管理的需要,每只股票最大投资额为100000元 (仓位管理:风控)

#数据获取(通用部分):投资者账户,可供投资股票,持仓数据,账户金额数据;注:由于使用系统提供的信号机制,不必获取价格数据
account = context.get_account('fantasy_account')
current_universe = context.get_universe(exclude_halt=True)
mfi = context.signal_result['MFI']
security_position = account.get_positions()
cash = account.cash

#择时策略部分:获取当前时点的buylist
buylist = []
for sec in current_universe:
    if mfi[sec] < buy_threshold and sec not in security_position:
        buylist.append(sec)
    elif mfi[sec] > sell_threshold and sec in security_position:
        order_to(sec, 0)
        cash += security_position[sec].amount * context.current_price(sec) # 估计卖出股票后的账户金额,注:context.current_price(sec)是获取sec股票当前价格
        # security_position[sec].amount获取购买股票的数量

#交易执行部分
d = min(len(buylist), int(cash) // 100000) # 可以买入的股票数量,如果资金不够,只买入部分
for sec in buylist[:d]:
    order(sec, 100000 / context.current_price(sec)) #基于仓位管理的需要,每只股票最大投资额为100000元

def timing(context):
    timing_basic_mfi(context) #基于mfi指标的择时策略
```

36-4

专业来自101%的投入!



基于MFI指标的量化投资策略的回测结果



金程教育
GOLDEN FUTURE

年化收益率 4.9% 基准年化收益率 15.5% 阿尔法 -4.0% 贝塔 0.57 夏普比率 0.45 收益波动率 10.8% 信息比率 -1.19 最大回撤 7.8% 换手率 1.80

回测详情

开始交易



37-4

专业来自101%的投入!



本节程序复习：交易指令



金程教育
GOLDEN FUTURE

#3. 程序复习：证券池的设置

universe = DynamicUniverse('HS300') # 证券池：可供选择的股票的范围；选择每一期的HS300股票组成每一期的证券池

universe = set_universe('HS300', start) # 证券池：可供选择的股票的范围；注：取start日的成分股防止survival bias

universe = StockScreener(Factor.PE.nlarge(10)) + ['000001.XSHE', '600000.XSHG'] # 设置证券池为PE值最大的10个股票，并加上平安银行和浦发银行

38-4

专业来自101%的投入!



基于CCI指标的量化投资策略



金程教育
GOLDEN FUTURE

- CCI指标的基本原理及操盘策略
- CCI指标的应用举例
- 基于CCI指标的量化投资策略的代码及注释
- 基于CCI指标的量化投资策略的回测结果

39-4

专业来自101%的投入!



➤ 基本原理：

- CCI (Commodity Channel Index)，又称为顺势指标，是美国股市技术分析家唐纳德·蓝伯特(Donald Lambert)于20世纪80年代提出。CCI指标用于判断资产是否处于超买或超卖情况的一种震荡指标，专门测量股价、外汇或者贵金属交易是否已超出常态分布范围。属于超买超卖类指标中较特殊的一种。波动于正无穷大和负无穷大之间。但是，又不需要以0为中轴线，这一点也和波动于正无穷大和负无穷大的指标不同。
- CCI的公式为：

$$CCI_n = \frac{\text{Typical Price} - SMA_n(TP)}{c \cdot \sigma_n(TP)}$$

- ✓ 其中 Typical Price = (最高价 + 最低价 + 收盘价) / 3，n一般采用20或40，c一般采用0.015
- ✓ CCI的变化范围很广，在n=20，c=0.015的情况下，一般认为低于-100为超卖，高于100为超买（正负1.5倍标准差）

➤ 操作策略：

- 买入条件：cci<-100超卖, 买入股票
- 卖出条件：cci>100超买, 卖出股票

40-4

专业来自101%的投入!



CCI指标的应用举例



41-4

专业来自101%的投入!



基于CCI指标的量化投资策略的代码及注释



```
##加载函数库；注：由于该策略使用优矿提供的signal机制，所以不用引入talib
import numpy as np
import pandas as pd

##初始化回测环境
start = '20150601' # 回测起始时间 注：支持两种日期表述形式（'2015-01-01'，'20150101'）
end = '20160601' # 回测结束时间
benchmark = 'HS300' # 策略参考标准
universe = DynamicUniverse('HS300') # 证券池：可供选择的股票的范围；选择每一期的HS300股票组成每一期的证券池
freq = 'd' # 用日线回测的策略
refresh_rate = 1 # 每天调一次仓，即每个交易日都会运行第三部分的handle_data函数

##初始化投资者（账户）参数
#accounts为字典类型，代表投资者所有的账户，而字典中每一个键代表一个账户，而每一个键对应的值为该账户的初始情况，如本程序中的键为fantasy_account（股票账户），值为相应配置
accounts = {
    'fantasy_account': AccountConfig(account_type='security', capital_base=100000000) #初始化投资者的股票账户：投资品种为股票，初始投资金额为1千万
}

##初始化策略参数
N = 20 # 移动窗口长度
buy_threshold = -100 # 买入阈值
sell_threshold = 100 # 卖出阈值

##初始化回测环境，指明创建账户时的工作，全局只运行一次
#建立策略的CCI信号机制，选择向前遍历的时间参数是20天
def initialize(context):
    cci = Signal('CCI20') #CCI5, CCI10, CCI20, CCI80等4中不同参数的预计算CCI，最常用的参数为40和20
    context.signal_generator = SignalGenerator(cci) #产生信号cci20, Quartz 框架会自动触发依赖图分析以及在回测区间内计算信号取值。

##handle_data函数是策略的核心函数，包含了所有策略算法的内容，包括数据获取，交易信号生成，订单委托等逻辑。
#handle_data函数无论是回测还是模拟交易场景，这个函数会根据回测频率 freq 的设置被调用。当freq='d'时，每天被调用一次，当freq='m'时，每分钟被调用一次。
def handle_data(context):
    timing(context)
```

42-4

专业来自101%的投入!



基于CCI指标的量化投资策略的代码及注释



金程教育
GOLDEN FUTURE

```
def timing_basic_cci(context):  
    ##basic_cci算法在cci<-100,超卖时买入,在cci>100,超买时卖出 (操盘规则:预测)  
    ##基于仓位管理的需要,每只股票最大投资额为100000元 (仓位管理:风控)  
  
    #数据获取(通用部分):投资者账户,可供投资股票,持仓数据,账户金额数据;注:由于使用系统提供的信号机制,不必获取价格数据  
    account = context.get_account('fantasy_account')  
    current_universe = context.get_universe(exclude_halt=True)  
    cci = context.signal_result['CCI20']  
    security_position = account.get_positions()  
    cash = account.cash  
  
    #获取投资者的股票账户 (fantasy_account)  
    #获取当前除停牌外的所有可供投资股票 (universe)  
    #获取证券池 (HS300) 中股票的CCI20信号  
    #字典型数据,上一k线结束后的有效证券头寸,即持仓数量大于0的证券及其头寸  
    #获取股票账户可用于投资的现金额度  
  
    #择时策略部分:获取当前时点的buylist  
    buylist = []  
    for sec in current_universe:  
        if cci[sec] < buy_threshold and sec not in security_position:  
            buylist.append(sec)  
        elif cci[sec] > sell_threshold and sec in security_position:  
            order_to(sec, 0)  
            cash += security_position[sec].amount * context.current_price(sec) # 估计卖出股票后的账户金额,注:context.current_price(sec) 是获取sec股票当前价格  
            # security_position[sec].amount获取购买股票的数量  
            # 全部卖出  
            # 股票sec的cci信号的取值大于卖出阈值(100),处于超买期,且有持仓  
  
    #交易执行部分  
    d = min(len(buylist), int(cash) // 20000) # 可以买入的股票数量,如果资金不够,只买入部分  
    for sec in buylist[:d]:  
        order(sec, 20000 / context.current_price(sec)) #基于仓位管理的需要,每只股票最大投资额为100000元  
  
def timing(context):
```

43-4

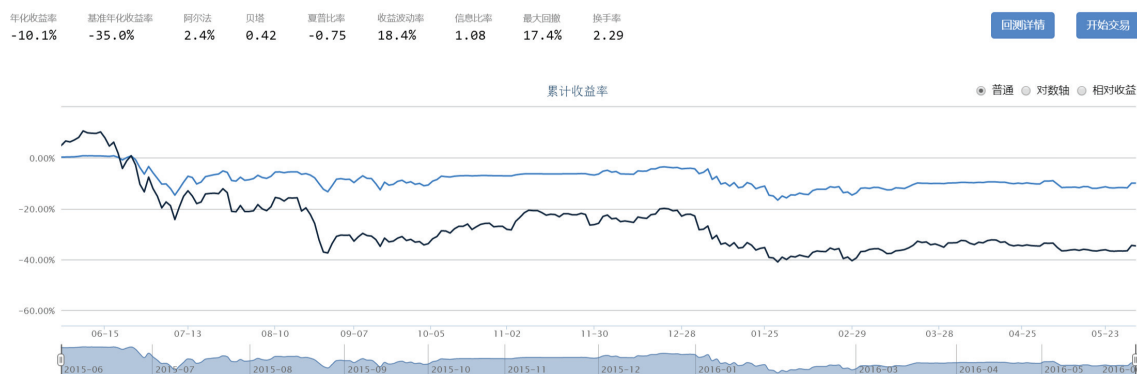
专业来自101%的投入!



基于CCI指标的量化投资策略的回测结果



金程教育
GOLDEN FUTURE



44-4

专业来自101%的投入!

CONTENTS

PROFESSIONAL · LEADING · VALUE-CREATING

▶ PART 1

基于技术分析的量化投资

▶ PART 3

基于技术指标的量化投资

▶ PART 2

基于通道技术的量化投资

专业来自101%的投入!



- 基于唐奇安通道的量化投资策略
- 基于布林带的量化投资策略

46-4

专业来自101%的投入!



- 唐奇安通道的基本原理及操盘策略
- 唐奇安通道的应用举例
- 基于唐奇安通道的量化投资策略的代码及注释
- 基于唐奇安通道的量化投资策略的回测结果

47-4

专业来自101%的投入!



- 基本原理：
 - 唐奇安通道指标（Donchian Channel）是由著名海龟交易员Richard Donchian发明的，是海龟算法的基础。该指标用周期（一般都是20，有的平台系统设置时可以改变的，有的则设置的不可以）内的最高价和最低价来显示市场价格的波动性，当其通道窄时表示市场波动较小，反之通道宽则表示市场波动比较大。周期内的最高价和最低价作为通道的上下轨。当价格突破通道上轨时，说明股价运动强势，应买入股票，当价格向下突破通道下轨时，空头市场较为强势，应卖出股票
 - 唐奇安通道的计算公式如下：
 - ✓ 上轨：N日内最高价
 - ✓ 下轨：N日内最低价
- 操作策略：
 - 买入条件：如果收盘价上穿唐奇安通道上轨，买入；
 - 卖出条件：如果收盘价下穿唐奇安通道下轨，卖出；

48-4

专业来自101%的投入!



49-4

专业来自101%的投入!



```
#加载函数库
import talib #Technical Analysis Library
import numpy as np

#初始化回测环境
start = '20150701' # 回测起始时间 注:支持两种日期表述形式 ('2015-01-01', '20150101')
end = '20160501' # 回测结束时间
benchmark = 'HS300' # 策略参考标准
universe = DynamicUniverse('HS300') # 证券池:可供选择的股票的范围;选择每一期的HS300股票组成每一期的证券池
freq = 'd' # 用日线回测的策略
refresh_rate = 1 # 每天调一次仓,即每个交易日都会运行第三部分的handle_data函数

#初始化投资者(账户)参数
#accounts为字典类型,代表投资者所有的账户,而字典中每一个键代表一个账户,而每一个键对应的值为该账户的初始情况,如本程序中的键为fantasy_account(股票账户),值为相应配置
accounts = {
    'fantasy_account': AccountConfig(account_type='security', capital_base=1000000) #初始化投资者的股票账户: 投资品种为股票,初始投资金额为1千万
}

#初始化策略参数
N = 20 # 移动窗口长度

#初始化回测环境,指明创建账户时的工作,全局只运行一次
def initialize(context):
    return

#handle_data函数是策略的核心函数,包含了所有策略算法的内容,包括数据获取,交易信号生成,订单委托等逻辑。
#handle_data函数无论是回测还是模拟交易场景,这个函数会根据回测频率 freq 的设置被调用。当freq='d'时,每天被调用一次,当freq='m'时,每分钟被调用一次。
def handle_data(context):
    timing(context)
```

50-4

专业来自101%的投入!



```
def timing_Basic_DC(context):
    ##basic_TQ算法在收盘价上穿唐奇安通道上轨,突破阻力位买入;如果收盘价下穿唐奇安通道下轨,突破支撑位则卖出 (操盘规则:预测)
    ##基于仓位管理的需要,每只股票最大投资额为20000元 (仓位管理:风控)

    #数据获取(通用部分):投资者账户,可供投资股票,价格数据,持仓数据,账户金额数据
    account = context.get_account('fantasy_account')
    current_universe = context.get_universe(exclude_halt=True)
    history = context.history(current_universe, 'closePrice', N, rtype='array')
    highest_price = context.history(current_universe, 'highPrice', N+1, rtype='array')
    lowest_price = context.history(current_universe, 'lowPrice', N+1, rtype='array')
    security_position = account.get_positions()
    cash = account.cash

    #择时策略部分:获取当前时点的buylist
    buylist = []
    for sec in current_universe:
        close = history[sec]['closePrice']
        up = highest_price[sec]['highPrice']
        down = lowest_price[sec]['lowPrice']
        if close[-2] < up[-2].max() and close[-1] > up[-1].max() and sec not in security_position: # 收盘价上穿OCL上轨,且无持仓(突破通道上轨阻力位)
            buylist.append(sec) # 将股票sec加入购买列表
        elif close[-2] > up[-2].min() and close[-1] < up[-1].min() and sec in security_position: # 收盘价下穿OCL下轨,且有持仓(突破通道下轨支撑位)
            order_to(sec, 0) # 全部卖出
            cash += security_position[sec].amount * context.current_price(sec) # 估计卖出股票后的账户金额,注:context.current_price(sec)是获取sec股票当前价格

    #交易执行部分
    d = min(len(buylist), int(cash // 20000))
    for sec in buylist[d:]:
        order(sec, 20000 / context.current_price(sec))

    #初始化购买股票列表
    #遍历所有可供投资股票;注:如该策略执行了选股策略,该部分应遍历被选中股票
    #
    #获取投资者的股票账户(fantasy_account)
    #获取当前除停牌外的所有可供投资股票(universe)
    #通过去N(N=20)个交易日的收盘价来估算布林带
    #上轨道由前21个交易日的最高价组成
    #下轨道由前21个交易日的最低价组成
    #字典型数据,上一x结束后的有效证券头寸,即持仓数量大于0的证券及其头寸
    #获取股票账户可用于投资的现金额度

    #可以买入的股票数量,如果资金不够,只买入部分
    #基于仓位管理的需要,每只股票最大投资额为20000元

def timing(context):
    timing_Basic_DC(context) #基于唐奇安通道的选股方法
```

51-4

专业来自101%的投入!



基于唐奇安通道的量化投资策略的回测结果



金程教育
GOLDEN FUTURE

年化收益率 -8.7% 基准年化收益率 -34.8% 阿尔法 -7.8% 贝塔 0.12 夏普比率 -1.57 收益波动率 7.9% 信息比率 0.74 最大回撤 7.9% 换手率 1.17

回测详情

开始交易



52-4

专业来自101%的投入!



基于Bull（布林带）的量化投资策略



金程教育
GOLDEN FUTURE

- 布林带的基本原理及操盘策略
- 布林带的应用举例
- 基于布林带的量化投资策略的代码及注释
- 基于布林带的量化投资策略的回测结果

53-4

专业来自101%的投入!



基于布林带的量化投资策略的基本原理及操作步骤



金程教育
GOLDEN FUTURE

- 基本原理：
 - BOLL（Bollinger Bands）是美国股市分析家约翰·布林根据统计学中的标准差原理设计出来的一种非常简单实用的技术分析指标。它使用均值及标准差来刻画股价的波动范围及未来走势的技术指标，一般而言，股价的运动总是围绕某一价值中枢（如均线、成本线等）在一定的范围内变动，布林线指标正是在上述条件的基础上，引进了“股价通道”的概念，其认为股价通道的宽窄随着股价波动幅度的大小而变化，而且股价通道又具有变异性，它会随着股价的变化而自动调整。正是由于它具有灵活性、直观性和趋势性的特点，BOLL指标渐渐成为投资者广为应用的市场上热门指标由于其需要表现取值范围，所以由三条线构成：上轨、中轨和下轨，因此也被称之为布林带。
 - BOLL的计算公式如下：
 - ✓ 中轨：N日的简单移动平均
 - ✓ 上轨：中轨 + k倍N日标准差
 - ✓ 下轨：中轨 - k倍N日标准差
- 操作策略：
 - 买入条件：如果收盘价上穿BOLL上轨，买入；
 - 卖出条件：如果收盘价下穿BOLL中轨，卖出；

54-4

专业来自101%的投入!



55-4

专业来自101%的投入!



```
#加载函数库
import talib #Technical Analysis Library
import numpy as np

#初始化回测环境
start = '20140701' # 回测起始时间 注:支持两种日期表述形式('2015-01-01','20150101')
end = '20150501' # 回测结束时间
benchmark = 'HS300' # 策略参考标准
universe = DynamicUniverse('HS300') # 证券池:可供选择的股票的范围;选择每一期的HS300股票组成每一期的证券池
freq = 'd' # 用日线回测的策略
refresh_rate = 1 # 每天调一次仓,即每个交易日都会运行第三部分的handle_data函数

#初始化投资者(账户)参数
#accounts为字典类型,代表投资者所有的账户,而字典中每一个键代表一个账户,而每一个键对应的值为该账户的初始情况,如本程序中的键为fantasy_account(股票账户),值为相应配置
accounts = {
    'fantasy_account': AccountConfig(account_type='security', capital_base=10000000) #初始化投资者的股票账户:投资品种为股票,初始投资金额为1千万
}

#初始化策略参数
N = 20 # 移动窗口长度
k = 1.96 # 标准差倍数

#初始化回测环境,指明创建账户时的工作,全局只运行一次
def initialize(context):
    return

#handle_data函数是策略的核心函数,包含了所有策略算法的内容,包括数据获取,交易信号生成,订单委托等逻辑。
#handle_data函数无论是回测还是模拟交易场景,这个函数会根据回测频率 freq 的设置被调用。当freq='d'时,每天被调用一次,当freq='m'时,每分钟被调用一次。
def handle_data(context):
    timing(context)
```

56-4

专业来自101%的投入!



```
def timing_Basic_Bull(context):
    ##basic_Bull算法在收盘价上穿BOLL上轨,突破阻力位买入;如果收盘价下穿BOLL中轨,突破支撑位则卖出 (操盘规则:预测)
    ##基于仓位管理的需要,每只股票最大投资额为200000元 (仓位管理:风控)

    #数据获取(通用部分):投资者账户,可供投资股票,价格数据,持仓数据,账户金额数据
    account = context.get_account('fantasy_account')
    current_universe = context.get_universe(exclude_halt=True)
    history = context.history(current_universe, 'closePrice', N+1, rtype='array')
    security_position = account.get_positions()
    cash = account.cash

    #择时策略部分:获取当前时点的buylist
    buylist = []
    for sec in current_universe:
        close = history[sec]['closePrice']
        up, mid, low = talib.BBANDS(close, timeperiod=N, nbdevup=k, nbdevdn=k)
        if close[-2] < up[-2] and close[-1] > up[-1] and sec not in security_position:
            buylist.append(sec)
        elif close[-2] > mid[-2] and close[-1] < mid[-1] and sec in security_position:
            order_to(sec, 0)
            cash += security_position[sec].amount * context.current_price(sec)

    #交易执行部分
    d = min(len(buylist), int(cash) // 20000)
    for sec in buylist[:d]:
        order(sec, 20000 / context.current_price(sec))

    #获取投资者的股票账户(fantasy_account)
    #获取当前持仓中除停牌外的所有可供投资股票(universe)
    #拿过去N+1(N=31)个交易日的收盘价来估算布林带
    #字典型数据,上一k线结束后的有效证券头寸,即持仓数量大于0的证券及其头寸
    #获取股票账户可用于投资的现金额度

    #初始化购买股票列表
    #遍历所有可供投资股票;注:如该策略执行了选股策略,该部分应遍历被选中股票
    #获取股票sec过去N(30)天的收盘价
    #talib提供的布林带计算函数,计算布林带上轨(up)、中轨(mid)和下轨(down)取值
    #收盘价上穿BOLL上轨,且无持仓(突破通路上轨阻力位)
    #将股票sec加入购买列表
    #收盘价下穿BOLL中轨,且有持仓(突破中轨支撑位)
    #全部卖出
    #估计卖出股票后的账户金额,注:context.current_price(sec)是获取sec股票当前价格

    #可以买入的股票数量,如果资金不够,只买入部分
    #基于仓位管理的需要,每只股票最大投资额为200000元

def timing(context):
    timing_Basic_Bull(context) #基于布林带的基础算法
```

57-4

专业来自101%的投入!



基于布林带的量化投资策略的回测结果

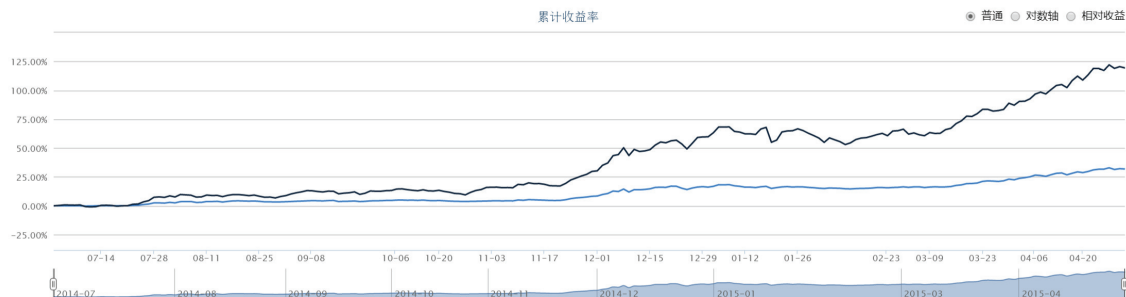


金程教育
GOLDEN FUTURE

年化收益率 40.3% 基准年化收益率 161.9% 阿尔法 -14.2% 贝塔 0.32 夏普比率 4.19 收益波动率 8.6% 信息比率 -3.91 最大回撤 3.1% 换手率 2.30

回测详情

开始交易



58-4

专业来自101%的投入!



金程教育
GOLDEN FUTURE

Thank you!

GOLDEN FUTURE GOLDEN FUTURE
GOLDEN FUTURE GOLDEN FUTURE
GOLDEN FUTURE GOLDEN FUTURE
GOLDEN FUTURE GOLDEN FUTURE

专业来自101%的投入!