程序化设计技术文档

目录

[anaconda使用 1](#_Toc482781903)

[spyder使用 1](#_Toc482781904)

[notebook jupyter使用 2](#_Toc482781905)

[tushae使用 2](#_Toc482781906)

[talib使用 2](#_Toc482781907)

[技术指标 2](#_Toc482781908)

[K线组合 5](#_Toc482781909)

[进阶 7](#_Toc482781910)

[米筐使用 8](#_Toc482781911)

[几个市场常用策略 8](#_Toc482781912)

[BM效应 8](#_Toc482781913)

[Graham number 格雷厄姆数字价值投资法 9](#_Toc482781914)

[A股市场的ETF轮动策略 9](#_Toc482781915)

[基本面选股器: Piotroski F-Score ranking system 9](#_Toc482781916)

[基于NPL对爬取的舆情数据进行情感分析 9](#_Toc482781917)

[仓位控制与止损 9](#_Toc482781918)

# 软件与框架

## anaconda使用

## spyder使用

## notebook jupyter使用

## tushae使用

首先导入tushare库：import tushae as ts

获取股票K线数据，ts.get\_k\_data(‘share code’)

获取指数K线数据，ts.get\_k\_data(‘share code’,index=True)

例：

#获得上证指数数据

ts.get\_k\_data('000001',index=True)

## talib使用

首先导入talib库：import talib as ta

### 技术指标

例：

#获取收盘价数据，拿到的是Dataframe，转换成ndarray

dcclose=ts.get\_k\_data('000001',index=True)['close'].values

#调用talib ma均线指标，获得30日和120日ma均线

ma30 = ta.MA(dcclose, timeperiod=30, matype=0)

ma120 = ta.MA(dcclose, timeperiod=120, matype=0)

#通过matplotlib画图，画出均线

import matplotlib.pyplot as plt

plt.plot(ma30)

plt.plot(ma120)

#图如下：



常用的talib指标为：

|  |  |  |  |
| --- | --- | --- | --- |
| 指标 | 描述 | Python Api 接口或算法 |  |
|  |  |  |  |
| BIAS | 乖离率 | 乖离率=[(当日收盘价-N日平均价)/N日平均价]\*100% |  |
| BOLL BBANDS | 布林轨道 | upperband, middleband, lowerband = BBANDS(close, timeperiod=5, nbdevup=2, nbdevdn=2, matype=0) |  |
| CCI | 顺势指标 | real = CCI(high, low, close, timeperiod=14) |  |
| KDJ  STOCH | 随机指标 | slowk, slowd = STOCH(high, low, close, fastk\_period=5, slowk\_period=3, slowk\_matype=0, slowd\_period=3, slowd\_matype=0) |  |
| MA | 移动平均线 | real = MA(close, timeperiod=30, matype=0) |  |
|  |  |  |  |
| MACD | 指数平滑移动平均线指标 | macd, macdsignal, macdhist = MACD(close, fastperiod=12, slowperiod=26, signalperiod=9) |  |
| MTM  MOM | 动量指标 | real = MOM(close, timeperiod=10) |  |
|  |  |  |  |
| PSY | 情绪指标 | 无 |  |
| RSI | 相对强弱指标 | real = RSI(close, timeperiod=14) |  |
| WR  WILLR | 威廉指标 | real = WILLR(high, low, close, timeperiod=14) |  |

编程习惯：

上面列表中，都是直接API接口，实际使用中，先要导入talib库，建议使用import talib as ta ，这样每个指标使用的时候，都要加上“ta.”比如MACD指标就要写成：macd, macdsignal, macdhist = ta.MACD(close, fastperiod=12, slowperiod=26, signalperiod=9)

MA指标就要写成：real = talib.MA(close, timeperiod=30, matype=0)

意外的是，最不提倡的导入方法“from talib import \*”是无法生效的。

如前面所讲，talib需要array数据格式，大部分网上拿到的数据都是DataFrame格式，因此存在一个数据转换过程！转换可以使用.value 方法实现。

如果是在米筐等测试框架中，在handle\_bar()中使用该指标数据，那么要取矩阵最后一个数据，也就是用[-1]定位。如前面拿到MA指标的real数据，最后一位就是real[-1]

### K线组合

Talib也支持K线组合的

如：

import tushare as ts

import talib as ta

dcdata =ts.get\_k\_data('000001',index=True)

open,high,low,close = dcdata['open'].values,dcdata['high'].values,dcdata['low'].values,dcdata['close'].values

#Engulfing Pattern 吞噬模式（阴阳并线）

integer = ta.CDLENGULFING(open, high, low, close)

输出的integer是一个numpy.ndarray ,可以看到数据里有100和-100

type (integer)

Out[81]: numpy.ndarray

print(integer)

[0 0 0 ..., 0 0 0]

100 in integer

Out[83]: True

-100 in integer

Out[84]: True

常用K线形态组合列表：

|  |  |  |
| --- | --- | --- |
| K线形态组合名 | 描述 | Python接口 |
| CDL3BLACKCROWS | 三只黑乌鸦 | integer = CDL3BLACKCROWS(open, high, low, close) |
| CDL3LINESTRIKE | 三线打击 | integer = CDL3LINESTRIKE(open, high, low, close) |
| CDL3WHITESOLDIERS | 白三兵（红三兵） | integer = CDL3WHITESOLDIERS(open, high, low, close) |
| CDLADVANCEBLOCK | 上升受阻（） | integer = CDLADVANCEBLOCK(open, high, low, close) |
| CDLBELTHOLD | 捉腰带线 | integer = CDLBELTHOLD(open, high, low, close) |
| CDLENGULFING | 吞噬模式（阴阳并线） | integer = CDLENGULFING(open, high, low, close) |
| CDLLONGLEGGEDDOJI | 长十字星 | integer = CDLLONGLEGGEDDOJI(open, high, low, close) |
| CDLLONGLINE | 长实体 | integer = CDLLONGLINE(open, high, low, close) |
| CDLSHORTLINE | 短蜡烛 | integer = CDLSHORTLINE(open, high, low, close) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

参见：<https://www.ricequant.com/community/topic/2393/>

如前面一样，调用的时候建议使用import talib as ta,函数前添加“ta.”

### 进阶

可以查看talib所有的函数：

print(ta.get\_functions())

print(ta.get\_function\_groups())

每个函数可以通过help（）或“？”来获得帮助，

如：

**help(ta.MA)**

Help on built-in function MA in module talib.func:

MA(...)

MA(real[, timeperiod=?, matype=?])

Moving average (Overlap Studies)

Inputs:

real: (any ndarray)

Parameters:

timeperiod: 30

matype: 0 (Simple Moving Average)

Outputs:

real

**ta.MA?**

Docstring:

MA(real[, timeperiod=?, matype=?])

Moving average (Overlap Studies)

Inputs:

real: (any ndarray)

Parameters:

timeperiod: 30

matype: 0 (Simple Moving Average)

Outputs:

real

Type: builtin\_function\_or\_method

**抽象函数abstract**

talib使用熟练之后，可以使用abstract抽象API，每个函数的输入都是固定格式的arrays .

import numpy as np

# note that all ndarrays must be the same length!

inputs = {

'open': np.random.random(100),

'high': np.random.random(100),

'low': np.random.random(100),

'close': np.random.random(100),

'volume': np.random.random(100)

}

output=ta.abstract.MA(inputs)

plt.plot(output)



还可以通过Function使用：

newma=ta.abstract.Function('sma')

newma(inputs,5)

Out[83]:

array([ nan, nan, nan, ..., 0.50493402,

0.52459283, 0.51730766])

通过字符串赋值，就可以在自己的程序中自由的构建和使用指标了！

## 米筐使用

## 人工智能与机器学习

prophet

# 几个市场常用策略

## BM效应

## Graham number 格雷厄姆数字价值投资法

https://www.ricequant.com/community/topic/285/

## A股市场的ETF轮动策略

<https://www.ricequant.com/community/topic/283/>

## 基本面选股器: Piotroski F-Score ranking system

<https://www.ricequant.com/community/topic/297/>

## 基于NPL对爬取的舆情数据进行情感分析

https://www.ricequant.com/community/topic/3339//2

# 仓位控制与止损