



加密货币量化交易入门全攻略



(/gitchat/author/589a9524dc9bb5a36dd74ac1)

余博伦 (/gitchat/author...

freeCodeCamp 中国推广者，致力于为技术社区提供良心干货、暖心教程。

查看本场Chat



(/gitchat/activity/5c7bf51875f10b305744b3f2)

本文整理并介绍了从初步了解到入门上手加密货币量化交易的所有关键知识点，包含市场特性分析、交易平台选择、工具框架、技术指标、交易策略、量化程序搭建部署在内一整条知识脉络。在阅读学习完本文之后，读者将能够使用 Python 语言开发出一个最小可用版本的加密货币量化交易机器人程序。

市场特性

经历过 2018 年，相信加密货币和区块链的概念对于互联网和金融投资圈的朋友们已经不再陌生。量化交易的概念乍听起来可能比较高端，而实际上只是利用计算机算法替代人来进行决策交易。在传统的金融及衍生品交易市场中，作为一个普通人，我们要么拿不到交易接口的权限，或者根本达不到资金门槛。然而加密货币交易市场的出现，为每一个技术极客通过量化交易赚取聪明钱提供了可能。

无时间限制

加密货币市场没有交易时间上的限制，不存在开市休市以及法定节假日。任何价格波动会直接市场中实时体现，不存在股市当中的跳空高开或者盘前跳水，买入卖出也没有任何时间上的限制。这对于同样可以24小时连轴运转的量化交易程序来说，是一条非常有价值的优势特点。

无资金门槛

你可以使用任何主流国家的法币兑换成任意数量的加密货币，不存在股市当中“一手”交易的说法。加密货币市场也没有最低入金门槛，无需动辄几十万上百万元的启动资金才能够进行量化交易。



没有基金交易中的高昂手续费，也没有股市券商的佣金。加密货币交易的手续费非常低廉，甚至一些交易平台还会为交易者提供手续费反佣，或者直接推出专门针对量化做市商的合作计划。

高市场波动

相比较以年为单位的股票市场，加密货币市场有更高的波动频率；相比较外汇市场，加密货币市场有更高的波动幅度。而且如今加密货币交易已经有了较为成熟的期货、现货、做多做空的平台和工具，相比较传统交易市场，有着更高频次的交易机会。

竞争力比较优势

加密货币交易市场是一个总市值非常小的新兴交易市场，无需担心与过多华尔街顶级量化团队竞争，任何一名技术极客都有机会凭借自己的才智在加密货币市场中赚取聪明钱。

交易平台

可以进行加密货币交易的平台非常多，在此只选其中两个作为重点推荐：

Binance 币安 (<https://www.binance.com/?ref=20768006>)

币安是目前活跃用户数最多，交易深度最好的加密货币现货交易平台。平台也提供了非常完善的市场数据和交易 API 供量化程序使用，同时也有阶梯式的交易手续费优惠。

官方 API 文档 (<https://github.com/binance-exchange/binance-official-api-docs>)

BitMEX (<https://www.bitmex.com/register/ovxNym>)

BitMEX 是目前全球最为专业，交易量最大的比特币期货交易平台，提供了非常完善的杠杠做空做多的工具。同样也有着非常完善的 API 支持，并且提供交易手续费反佣。

BitMEX API (<https://www.bitmex.com/app/apiOverview>)

工具类库

俗话说，不要重复造轮子。



(/)



目前社区中已经出现了非常多加密货币量化交易相关的工具类库，涵盖了市场数据收集、数据分析、策略回测、交易 API 整合各个环节。在此仅推荐一些优秀的项目供读者参考：

- ccxt (<https://github.com/ccxt/ccxt>) 支持整合了一百多个加密货币交易所的市场数据及交易 API，同时适用于 JavaScript / Python / PHP 三种语言。
- TA-Lib (<http://ta-lib.org/>) 包含了 200 多种内置指标的技术分析类库，支持 C/C++、Java、Perl、Python 等多种语言。
- Crypto Signals (<https://github.com/CryptoSignal/crypto-signal>) 加密货币自动技术分析，交易信号提醒。
- gekko (<https://github.com/askmike/gekko>) Node.js 编写的自动交易及策略回测系统。
- binance-trader (<https://github.com/yasinkuyu/binance-trader>) 专为币安平台定制的交易机器人。
- sample-market-maker (<https://github.com/BitMEX/sample-market-maker>) BitMEX 官方做市机器人示例程序。
- Blackbird (<https://github.com/butor/blackbird>) 最火的比特币无风险套利策略。

量化策略

加密货币市场中适用的量化策略主要可以分为两大类：套利策略和趋势策略。

套利策略

套利策略主要是指利用平台间差价（俗称：搬砖）、期货之间或期货与现货差价（跨期套利，期现套利）、交易对间差价（三角套利，多边套利）等捕捉差价赚取利润的策略。此类策略获利空间小，同时风险也较小。

不过需要注意的是，当前市场中已经出现了非常多的各类型套利交易系统，拿跨平台套利来说，各个交易所之间的差价可能会在几秒内，甚至毫秒级别的时间差上就被抹平。因此套利量化策略更多的是在量化系统性能上的竞争。

本文中不再对此类竞争激烈，获利空间小的策略展开讨论。



趋势策略

趋势策略则是指通过制定一系列出入市规则，由算法来代替人做决策，参与市场的涨跌波动来获取利润的量化策略。

本文后续的内容主要讨论的也是此类策略。趋势策略通过计算机自动根据交易市场的一些关键技术指标数值来判断买入、卖出的时机。这些基于数值数据的逻辑判断，也即是“量化”的含义所在。

当下的趋势策略大多是基于技术指标进行固定模式判断交易的机械策略，也有利用机器学习算法，根据量化策略在市场中的实际表现，优化调参的做法，然而此类机械策略仍然是量化策略的核心。

技术指标

想要开发基于趋势策略的量化程序，自然要对技术指标有所了解。所谓技术指标，就是指描述市场量价关系的数学公式或数学模型，这些公式或模型通常以图形化的方式在交易图表当中展示。

本文将通过 Tradingview (<https://tradingview.go2cloud.org/SH2kr>) 平台，使用 **Pine Script**（在 Tradingview 平台上用于开发技术指标和交易策略的脚本语言）对几个常用的技术指标进行介绍。

- Pine Script 教程 (https://www.tradingview.com/wiki/Pine_Script_Tutorial)
- Pine Script 参考手册 (<https://cn.tradingview.com/study-script-reference/>)

移动平均线 Moving Average (MA)

feibilanceon 发表在TradingView.com, 3月 04, 2019 12:42:50 CST
BITSTAMP:BTCUSD, 1D 3753.14 ▼-36.38 (-0.96%) O:3789.70 H:3806.90 L:3726.00 C:3753.14



移动平均线是指 N 个单位时间内的平均价格线，例如 MA5 是指 5 个单位时间的平均价格线，计算公式为：

```
// 声明当前指标名称属性
//@version=3
study(title="Moving Average", shorttitle="MA", overlay=true)
// 均线函数
pine_sma(x, y) =>
    sum = 0.0
    for i = 0 to y - 1
        sum := sum + x[i] / y
    sum
// 调用均线函数，其中 close 表示单位时间的收盘价
ma5 = pine_sma(close, 5)
// 绘制均线
plot(ma5)
// 直接调用内置函数
plot(sma(close, 10))
```

移动平均线将最基本的统计学概念平均数引入到了量价分析当中，也是技术分析中最常使用的一项技术指标。移动平均线可以同价格走势直接进行比较，也可以组合多条不同单位时间的移动平均线来判断当前趋势。

异同移动平均线 Moving Average Convergence / Divergence (MACD)

feibillanleon 发表在TradingView.com, 3月 04, 2019 12:43:32 CST
BITSTAMP:BTCUSD, 1D 3752.05 ▼-37.47 (-0.99%) O:3789.70 H:3806.90 L:3726.00 C:3752.05



创建于 TradingView

简称 MACD 的异同移动平均线指标是由移动平均线衍生出来的。MACD 使用 DIF 和 DEA 两条线以及 Histogram 柱状图来描述价格趋势运行的速度。

其中 DIF 线表示快慢两条指数移动平均线（EMA）的差值，一般情况下，快线使用 EMA12，慢线使用 EMA26 。

DEA 是基于 DIF 计算出的指数移动平均线，一般取 DIF 的 EMA9 。

Histogram 则是 DIF 与 DEA 直接的差值，在交易图表中一般以柱状图形态表示。MACD 指标的计算代码如下：



布林带当前指标名称属性

//@version=3

study(title="MACD", shorttitle="MACD")

// 指数移动均线函数

pine_ema(x, y) =>

alpha = 2 / (y + 1)

sum = 0.0

sum := alpha * x + (1 - alpha) * nz(sum[1]) // 递归调用, 指数加权

// 计算 MACD

fast_length = 12

slow_length = 26

signal_length = 9

fast_ma = ema(close, fast_length)

slow_ma = ema(close, slow_length)

DIF = fast_ma - slow_ma

DEA = ema(DIF, signal_length)

hist = DIF - DEA

// 绘制指标



布林带 Bollinger Band (BB)

feibilanceon 发表在TradingView.com, 3月 04, 2019 12:43:57 CST

BITSTAMP:BTCUSD, 1D 3753.89 ▼ -35.63 (-0.94%) O:3789.70 H:3806.90 L:3726.00 C:3753.89



创建于 TradingView

布林带是一种利用均值回归原理设计出的价格通道指标，通过价格的均值和标准差描述价格运行的区间范围。布林带的计算代码如下：



```
// 声明当前指标名称属性
//@version=3
study(shorttitle="BB", title="Bollinger Bands", overlay=true)
length = 20 // 单位长度
mult = 2.0 // 标准差乘数
basis = sma(close, length) // 基准移动平均线，布林带中轨
// stdev(x, y) 为标准差计算函数
dev = mult * stdev(close, length)

upper = basis + dev // 布林带上轨
lower = basis - dev // 布林带下轨

//绘制指标
```

示例量化程序开发

接下来，本文将综合上述介绍过的所有知识点，开发出一个最小可用的量化交易机器人程序。此程序将在 Linux 环境下使用 Python 语言进行开发，我们会使用到之前介绍过的一些类库。为了避免不必要的兼容性问题，请读者选用相近的开发环境进行调试。

- Linux 选用 Ubuntu 16.04 64 位发行版
- Python 选用 python3 以上版本

编写代码之前，我们需要安装如下依赖：

```
pip install ccxt numpy TA-Lib schedule
```

市场数据获取

在示例程序中，我们选用 BitMEX (<https://www.bitmex.com/register/ovxNym>) 作为交易平台，它除了有非常完善的 API 支持以外，还提供了测试网络 (<https://testnet.bitmex.com/register/YdA5gq>)，方便开发和调试。



```
# 设置
symbol = 'BTC/USD' # 交易对
timeframe = '1h' # 单位时间选用 1 小时
limit = 100 # 数据条数
params = {'partial': True} # 包含最新一条K线数据
environment = 'testnet' # 选用测试网环境
# 初始化交易所实例
exchange = ccxt.bitmex()
# 使用测试网络
if 'test' in exchange.urls and environment == 'testnet':
    exchange.urls['api'] = exchange.urls['test']
# 获取最近 100 条数据, 包含当前一小时未完成的K线
since = exchange.milliseconds() - (limit - 1) * 60 * 1000
# 获取 K 线数据
candles = exchange.fetch_ohlcv(symbol, timeframe, since, limit, params)
# 输出测试数据
print('{}: O: {} H: {} L: {} C: {}'.format(
    exchange.iso8601(candles[-1][0]),
    candles[-1][1],
    candles[-1][2],
    candles[-1][3],
    candles[-1][4]))
```

技术指标分析

之后我们需要使用获取到的市场数据, 运用技术指标进行计算分析。而后根据计算的结果, 设定一系列条件判断, 也就是所谓的量化策略。

此处我们要使用到之前介绍过的:

- TA-Lib (<http://ta-lib.org/>)

TA-Lib 内含 200 多种常用的技术指标算法, 已经完全能够满足我们的需求。



import logging

import numpy

import talib

from talib import MA_Type

logging.basicConfig(level=logging.DEBUG)

close = numpy.array(list(map(lambda x:x[4], candles))) # 获取K线收盘价

logging.info('close %s', close)

upper, middle, lower = talib.BBANDS(close, matype=MA_Type.SMA) # 计算布林

logging.info('upper %s', upper)

logging.info('lower %s', lower)

if close[-1] > lower[-1] and close[-2] > lower[-2] and close[-3] < lower[-3]:
 logging.warn('Buy')

if close[-1] < upper[-1] and close[-2] < upper[-2] and close[-3] > upper[-3]:
 logging.warn('Sell')



交易平台下单

下单需要获取到账户权限，因此我们需要在交易平台上创建一个拥有下单权限的 API 密钥。

- 创建测试网 API (<https://testnet.bitmex.com/app/apiKeys>)
- 创建真实账户 API (<https://www.bitmex.com/app/apiKeys>)



设置



```
api_key = 'XXXXX-123456789'
```

```
api_secret = 'AbCdEfG1234567891011'
```

```
# 输入你申请到的 API 密钥
```

```
exchange = ccxt.bitmex({  
    'apiKey': api_key,  
    'secret': api_secret,  
    'enableRateLimit': True,  
})
```

```
...
```

```
if close[-1] > lower[-1] and close[-2] > lower[-2] and close[-3] < lower[-3]:  
    logging.warn('Buy')  
    buy_order = exchange.create_market_buy_order(symbol, amount) #市价买单  
    logging.info('buy order %s', buy_order)
```

```
if close[-1] < upper[-1] and close[-2] < upper[-2] and close[-3] > upper[-3]:  
    logging.warn('Sell')  
    sell_order = exchange.create_market_sell_order(symbol, amount) #市价卖  
    logging.info('sell order %s', sell_order)
```

上述代码示例完整版可以在 [gitchat-crypto-quant](https://github.com/discountry/gitchat-crypto-quant) (<https://github.com/discountry/gitchat-crypto-quant>) 获取。

搭建部署

我们可以使用 supervisor (<http://supervisord.org/>) 守护程序来长久运行开发好的量化机器人程序。

```
apt-get install supervisor
```

安装完成后在 `/etc/supervisor/conf.d/` 文件夹下新建一个名为 `bot.conf` 的设置文件：

```
[program:bot]  
command=/usr/local/bin/bot.sh  
autostart=true  
autorestart=true  
stderr_logfile=/var/log/bot.err.log  
stdout_logfile=/var/log/bot.out.log
```

之后读入并重启 supervisor 进程：



```
supervisorctl reread
supervisorctl update
```

现在在命令行输入：

```
supervisorctl status
```

就可以监控机器人程序的运行状况了。你也可以通过：

```
supervisorctl tail bot
```

来查看程序输出的信息。

```
INFO:root:2019-03-04T01:00:00.000Z: O: 3775 H: 3814.5 L:3767 C:3799
INFO:root:2019-03-04T02:00:00.000Z: O: 3799 H: 3799.5 L:3797.5 C:3798.5
INFO:root:2019-03-04T03:00:00.000Z: O: 3798.5 H: 3828 L:3797.5 C:3802
INFO:root:2019-03-04T04:00:00.000Z: O: 3802 H: 3802.5 L:3793 C:3796
INFO:root:2019-03-04T05:00:00.000Z: O: 3796 H: 3796 L:3773 C:3773.5
INFO:root:close [3799. 3798.5 3802. 3796. 3773.5]
INFO:root:upper [3807.5408585 3809.45448661 3812.89071943 3813.57716612 3814.45526567]
INFO:root:lower [3774.4591415 3774.34551339 3774.10928057 3774.62283388 3773.14473433]
```

程序开始运行后，发出交易信号时，也可以在交易平台看到自动下单成交的效果：





1. 中文 Quant 相关资源索引 (<https://github.com/thuquant/awesome-quant>)
2. 英文 Quant 相关资源索引 (<https://github.com/wilsonfreitas/awesome-quant>)
3. 量化交易资源索引 (<https://github.com/EliteQuant/EliteQuant>)
4. Tradingview (<https://tradingview.go2cloud.org/SH2kr>) 最优秀强大的交易图表库，最专业的技术分析社区。
5. metatrader (<https://www.metatrader5.com/>) 最专业的交易终端，可以接入各类交易市场的经纪商，拥有庞大的技术社区进行相关的自动交易和技术指标开发。
6. 发明者 (<https://www.fmz.com/>) 前身是 BotVS，国内最火的量化平台和社区。
7. cryptomon (<https://github.com/cryptomon-io/>) 一个专注于通过机器学习预测加密货币市场的开发组。
8. CoinMetrics (<https://github.com/coinmetrics-io/>) 加密资产分析研究组织。

本文首发于 GitChat，未经授权不得转载，转载需与 GitChat 联系。



16



1

互动评论



说点什么

评论



展护卫

6 个月前

写得内容太少了，每一部分展开来好好写写 都能形成一个专栏。



鼓掌



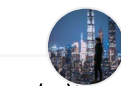
姬小光

3 年前

量化亏损+1



路飞君



3 年前

博士加油



阿永

3 年前

学习学习



起起小兄弟呀🐵🐵🐵🐵

3 年前

博士棒棒哒



[查看更多](#)