

# 数字货币量化 交易研究报告

—— 2018年9月 ——



TokenClub  
— 研究院 —

# 目录

<b>1. 量化交易综述</b>	<b>5</b>
1.1 量化交易概述	5
1.2 传统金融市场的量化交易	5
1.3 数字货币量化交易	6
<b>2. 量化交易的优势与风险</b>	<b>7</b>
2.1 量化交易的优势	7
2.1.1 严格的纪律性	7
2.1.2 完备的系统性	8
2.1.3 妥善运用套利思想	9
2.1.4 靠概率取胜	9
2.2 量化交易的风险	9
2.2.1 历史数据的完整性	10
2.2.2 策略模型设计上的缺陷	10
2.2.3 系统故障、风险控制缺失	10
2.2.4 过去并不代表未来	10
2.2.5 过度拟合	11
2.2.6 交易成本敏感性	11
2.3 非专业投资者如何评估量化策略	12
<b>3. 数字货币的量化策略</b>	<b>13</b>
3.1 高频交易	13
3.1.1 高频交易理论	13
3.1.2 高频交易特征	14

3.1.3 高频交易分类	15
3.1.4 高频交易优点	15
3.1.5 高频交易缺点	15
3.2 趋势交易与趋势理论	16
3.2.1 趋势理论概述	16
3.2.2 趋势理论分析方法	17
3.2.2.1 根据成交量判断趋势的变化	17
3.2.2.2 盘局可以代替中级趋势	18
3.2.2.3 主要趋势不会立马反转	19
3.2.3 趋势线	19
3.2.4 利用趋势轨道决定买卖点	20
3.2.5 趋势理论的缺陷	20
3.3 马丁格尔策略 BTC 应用	21
3.3.1 马丁格尔策略理论	21
3.3.2 马丁格尔策略优缺点	22
3.3.3 马丁格尔策略回测 BTC	22
3.3.3.1 策略的自动反手机制	23
3.3.3.2 策略核心代码	24
3.3.3.3 回测及分析	25
3.4 动态平衡策略 BTC 量化	30
3.4.1 动态平衡策略理论	30
3.4.2 动态平衡策略优点	31
3.4.3 动态平衡策略缺点	32
3.4.4 动态平衡策略回测 BTC	32
3.4.4.1 策略逻辑	32
3.4.4.2 策略框架	33

3.4.4.3 下单模块	34
3.4.4.4 撤单模块	35
3.4.4.5 回测及分析	36
3.5 运用机器学习预测币价	39
3.5.1 数据收集	40
3.5.2 训练、测试、随机游走模型	42
3.5.3 结果分析比较	47
<b>4. 数字货币量化套利场景</b>	<b>48</b>
4.1 搬砖套利	48
4.1.1 跨交易所两角硬搬套利	48
4.1.2 跨交易所两角对冲套利	49
4.1.3 交易所站内三角循环套利	52
4.1.4 跨交易所三角套利	52
4.1.5 跨国套利	54
4.1.6 打新套利	55
4.2 期现套利	55
4.2.1 期货的优势	55
4.2.1.1 回测期货可以双向交易	55
4.2.1.2 期货带有杠杆	56
4.2.1.3 可对冲套利	56
4.2.1.4 对冲风险	56
4.2.2 期现套利	56
4.2.3 跨期套利	58
4.2.3.1 牛市套利	58
4.2.3.2 熊市套利	58
4.2.3.3 BTC 期货分析	59

4.2.4 统计套利	61
4.2.5 跨品种套利	62
4.2.6 跨市套利	62
<b>5. 量化开发语言与开发库</b>	<b>63</b>
5.1 Python	63
5.1.1 NumPy	63
5.1.2 Pandas	64
5.1.3 SciPy	64
5.1.4 Matplotlib	65
5.2 R 语言	66
5.2.1 数据管理	67
5.2.2 指标计算	68
5.2.3 回测交易	69
5.2.4 投资组合	69
5.2.5 风险管理	70
5.3 其他编程语言	70
<b>6. 数字货币量化投资未来展望</b>	<b>71</b>
6.1 国内外量化投资现状	71
6.2 数字货币量化投资发展趋势	72
6.3 对数字货币量化投资领域的期待	73
<b>风险提示</b>	<b>74</b>



# 1. 量化交易综述

## 1.1 量化交易概述

量化交易(Quantitative Trading)也称自动化交易、程序化交易、算法交易。是指借助先进的统计学方法和数学模型，利用计算机技术来进行程序化交易的投资方式。量化交易从庞大的历史数据中海选能带来超额收益的多种“大概率”事件以制定策略，用数学模型验证及固化这些规律和策略，然后严格执行已固化的策略来指导投资，以求获得可以持续的、稳定且高于平均收益的超额回报。

量化交易起源于上世纪七十年代的股票市场，之后迅速发展和普及，尤其是数字货币的兴起，导致量化交易逐渐成为主流。量化交易有很多种，包括跨平台搬砖、趋势交易、对冲套利、三角套利、跨期套利、ETF 轮动套利等。

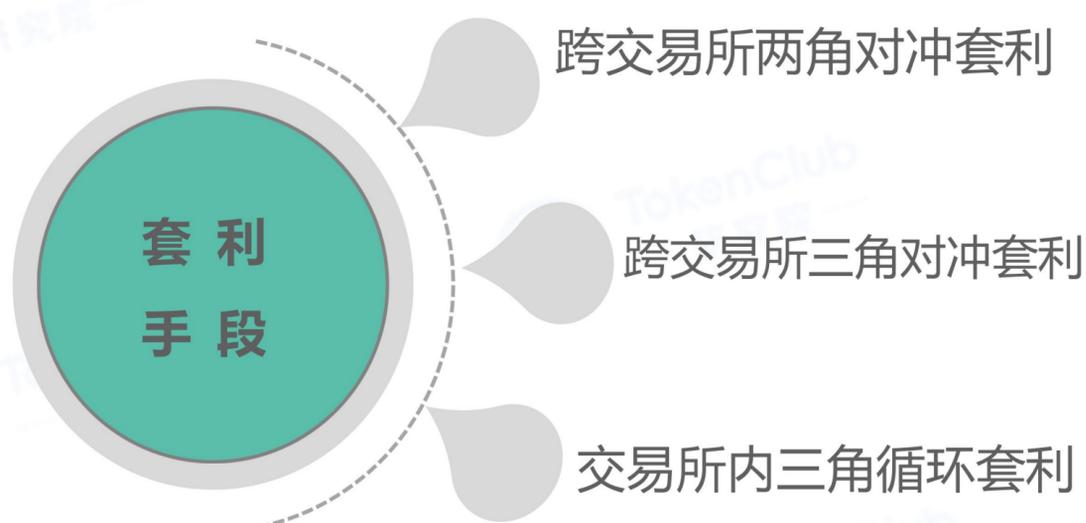
## 1.2 传统金融市场的量化交易

这里说的传统金融市场是主要是股市，基金，期权，期指市场，股票交易主要以手动交易为主，但在国内成熟的期货市场，量化自动化交易的比重已经超过 50%。在股指期货中，高频交易量很大，而高频交易与量化自动化关系密切，大量的高频交易靠人工是很难实现的，虽然量化自动化不一定是高频交易，但高频交易大部分都是靠量化自动化来实现的。当前我国股指期货也有一些高频交易模式，但使用的比例很低，随着市场参与者的增多，金融期货品种的不断增多，相信熟悉

和选择量化自动交易的人会越来越多，量化自动化交易在股指期货中的发展前景还是十分乐观的。

### 1.3 数字货币量化交易

相对于传统金融市场的量化交易，由于数字货币市场是 7\*24 小时 T+0 交易，而国内 A 股一天交易 4 小时，每周交易 20 小时，还有其他法定假日不交易，数字货币一天就顶上传统金融市场一周的时间；其次交易所众多（交易所之间搬砖、币对之间搬砖），交易所没有牌照限制，交易所为了提高交易量也积极提供量化交易的机会；另外在传统金融市场很多应用不好的策略都能在数字货币市场得到很好的发挥（纯粹的自由市场经济）。导致数字货币市场的量化交易异常火爆，简直就是宽客（量化交易者）的天堂，聚集了众多量化投资者，甚至出现“量化交易万能论”。



## 2. 量化交易的优势与风险

量化交易在数字货币市场有“量化交易万能论”、“熊市跑出牛市的收益率”等神话，真的那么厉害？接下来分析一下量化交易在数字货币市场上占有的优势以及要承担的风险。

### 2.1 量化交易的优势

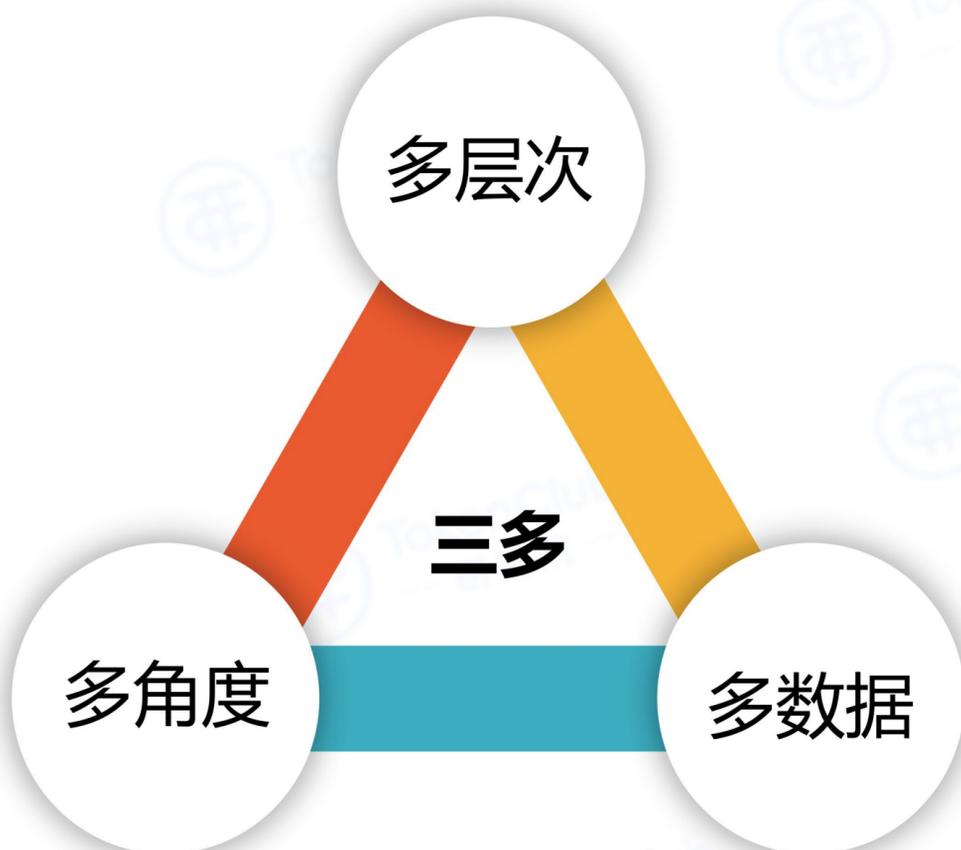
量化交易的优势主要体现在去除人性主观上由情感等因素带来操作上的失误，以及运用大数据、云计算、人工智能等先进技术在“大概率”上取胜，具体表现在以下几点：

#### 2.1.1 严格的纪律性

数字货币量化交易有着严格的纪律性，可以克服人性的弱点，如贪婪、恐惧、侥幸心理，也可以克服认知偏差。一个好的投资方法应该是一个“透明的盒子”。我们的每一个决策都是有理有据的，特别是有数据支持的。如果有人质问我，某年某月某一天，你为什么购买某个数字货币的话，我会打开数字货币量化交易系统，系统会显示出当时被选择的这个数字货币与其他的数字货币相比在成长面上、估值上、资金上、买卖时机上的综合评价情况，而且这个评价是非常全面的，比普通投资者拍脑袋或者简单看某一个指标买卖更具有说服力。

## 2.1.2 完备的系统性

完备的系统性具体表现为“三多”。



首先表现在多层次，包括在大类资产配置、行业选择、精选数字货币三个层次上都有对应的模型；其次是多角度，量化交易的核心投资思想包括宏观周期、市场结构、估值、成长、盈利质量、分析师盈利预测、市场情绪等多个角度；再者就是多数据，就是海量数据的处理。现有的数字货币有 2000 多个，交易所 230 多所，交易对更是上万对，而且这个数量还在继续增长，在这种数量级下，强大的量化交易的信息处理能力能反映它的优势，能捕捉更多的投资机会，拓展更大的投资机会。

### 2.1.3 妥善运用套利思想

量化交易正是在找估值洼地，通过全面、系统性的扫描捕捉错误定价、错误估值带来的机会。定性投资大部分精力是在做基本面分析；与定性投资不同，量化交易大部分精力花在定型上，分析哪里是估值洼地，哪一个币种被低估了，买入低估的，卖出高估的。

### 2.1.4 靠概率取胜

这表现为两个方面，一是量化交易不断的从历史中挖掘有望在未来重复的历史规律并且加以利用。二是依靠一组数字货币取胜，而不是一个或几个数字货币取胜。二是在数字货币实际操作过程中，运用概率分析，提高买卖成功的概率和仓位控制。

## 2.2 量化交易的风险

在数字货币市场中量化交易一般会经过海量历史数据仿真测试和模拟操作等手段对量化策略进行检验，并依据一定的风险管理算法进行仓位和资金配置，实现风险最小化和收益最大化；但也有存在很多的潜在风险，甚至说存在很多陷阱都不为过，主要分为以下几种：

## 2.2.1 历史数据的完整性

行情数据不完整可能导致模型与行情数据不匹配。行情数据自身风格转换，也可能导致模型失败，如交易流动性，价格波动幅度，价格波动频率等，而这一点是目前量化交易难以克服的。

## 2.2.2 策略模型设计上的缺陷

在策略模型设计上没有考虑仓位和资金配置没有安全的风险评估和预防措施，可能导致资金、仓位和模型的不匹配，而发生爆仓现象。而且不同的策略模型在不同市场行情中也会表现出不同的风险特征。

## 2.2.3 系统故障、风险控制缺失

在量化交易系统日常运转中，可能会出现程序错误、网络中断、硬件故障、突然停电或黑客攻击等都会导致量化交易系统的出现问题。

## 2.2.4 过去并不代表未来

在大多数情况下，通过回测可以证明你的量化交易系统在过去的表现。但是一个量化交易系统是不可能会适应所有数字货币交易所的。但市场出现巨大的变化时，量化交易有可能会失效；甚至有可能会遭遇死循环，进而导致了市场的异常波动。除了这个还有交易对流动性差、交易机制单一、交易员操作失误等都会影响量化交易系统。

## 2.2.5 过度拟合

为了使一个策略的回测表现令人满意，宽客往往会不断调优样本内参数，直至样本外数据回测结果达到心中的标准，一个策略总会被“优化”得近乎完美，这就出现了过度拟合现象。结果就是样本内稳定赚钱的策略，到了样本外就稳定亏钱。

## 2.2.6 交易成本敏感性

量化交易策略进行测试时，如果不计入手续费，资金曲线可能会产生巨大差异，甚至不计手续费的盈利策略，在计入手续费后也可能产生亏损。手续费相当于在原本的资金曲线上叠加了一条斜率为负的直线（在每次交易都是固定手数的情况下）。

## 2.3 非专业投资者如何评估量化策略

上面列的几种风险主要是基于量化交易系统上的风险，但是还有很多非量化交易系统自身的风险。

目前市场上的量化基金质量参差不齐，甚至有打着量化的旗号实则是庞氏骗局的团队，作为一个非专业的投资者要怎么简单的衡量评估量化策略、模型和团队？

- 1. 要正确对看待量化交易，量化交易不等于稳赚不赔。
- 2. 清楚量化策略的局限性，币圈的变化实在太快了。再好的投资策略，也是有时效性的，甚至一种投资方法，昨天还让你赚的盆满钵满，可能今天就会让你损失惨重。行业里那些已经被公开的量化策略，大多数是已经过时或者收益率很低的策略。
- 3. 从生态角度正确的看待量化团队，有些量化团队的定位是，帮助项目方绘制出完美的 K 线图，稳定币价走势，合理的制造交易热度塑造投资者信心，减少行情波动带来的投资者恐慌。有些团队则更多的通过项目交易、对敲，赚取手续费或交易挖矿分红，或者通过高频交易进行搬砖套利。同时也有一部分团队，与项目方勾结，操控币价，洗劫韭菜，谋取血淋淋的超高收益，“割韭菜”才是他们真正的主业。
- 4. 散户想参与量化交易难度颇高，简单策略大量盈利的时代曾经有过，但已是过去时，随着传统的金融机构逐步踏入数字货币量化市场，目前量化策略的难度正在快速上升。
- 5. 作为普通投资者，也务必看清量化领域的风险，不要盲目地加入其中。熊市之中，无论是普通投资者还是业内大佬，任何人都很难找到真正的捷径。

## 3. 数字货币的量化策略

量化策略是指使用计算机作为工具，通过一套固定的逻辑来分析、判断和决策。量化策略分为主动投资和被动投资。其中主动投资策略会积极寻找市场中的套利机会，适时根据投资者的判断调整资产组合，以求收益最大化；被动投资与主动投资相对，采用市场投资组合和主流数字货币的方式管理资产。下面列出币圈常见的量化策略：

### 3.1 高频交易

#### 3.1.1 高频交易理论

高频交易是量化交易的一种形式，以速度见长，它利用复杂的计算机技术和系统，以毫秒级的速度执行交易，且日内短暂持仓。

### 3.1.2 高频交易特征

高频交易有如下几个关键特征：



处理分笔交易数据和算法交易是高频交易的重要流程，高频交易通过对市场分笔交易数据进行采集、处理，分析市场在微观上潜在的交易机会，一旦确认交易机会，通过算法交易及时下单进场。高资金周转率和日内交易也是高频交易的特点，交易过程中，资金快速进场出场，一秒钟内可以发生多次下单、撤单的动作。资金在整个交易过程中高速流转，提高了市场流动性。

### 3.1.3 高频交易分类



### 3.1.4 高频交易优点

高频交易的优点是能够实现高收益并且降低风险的理想交易。因为持仓时间很短，因此规避了一些无法预测的市场风险，这样交易状况更稳定。

### 3.1.5 高频交易缺点

较平交易的缺点主要表现在其资金容量有限制，对交易软件和硬件的要求极高。具体如下：

高频交易并不是适合每个人，高频交易策略对计算机和网络的性能要求极高，以至于有些交易机构将自己的服务器群组（server farms）安置到了离交易所计算机很近的地方，以缩短交易指令通过光缆以光速旅行的距离。高频交易需要的设

备和计算能力对中小投资者是一种不可逾越的门槛，下图是专业高频设备的一些速度优化手段，都是在微秒级的优化（1秒等于1百万微秒）。

速度优化				
部位	一般现状	优化手段	优化目标	终极目标
物理距离	5-20ms	COLOCATION	300us	150us
报单系统	150us	采用FPGA系统或县城优化精简系统	40us	10us
网卡	100us	万兆网卡	10us	0
交换机	200us	纳米交换机	0.2us	~0
硬件处理	50us	(HP380&X980) 高频极速交易服务器	35us	0
架构延时	50us	客户程序与报单系统同一个服务器	5us	0
线路	30us/km	电缆>裸光纤线>无线微波>激光>中微子>光速	3.3us/km	~0

高频交易有一个致命的弱点，就是其资金容量很有限，由于非常简单，你可以交易这么短，就是一秒内，市场的筹码就是这么点，你能够投入多少资金呢？

## 3.2 趋势交易与趋势理论

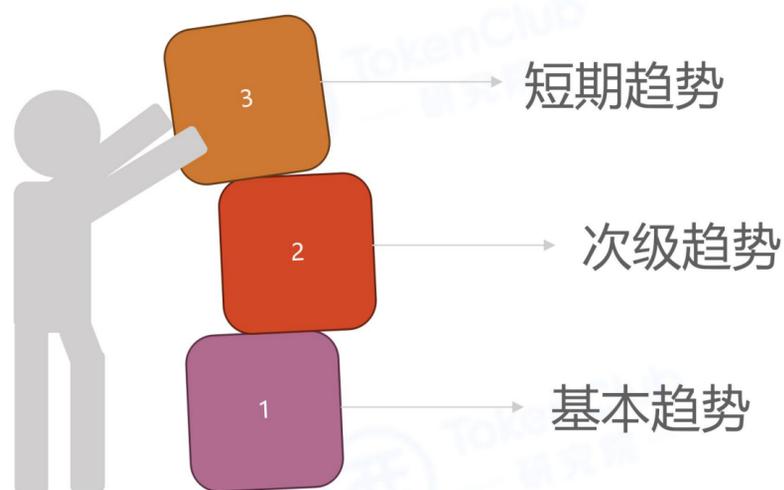
所谓趋势交易就是价格以趋势的方式演变，我们都是知道这个市场是有趋势的，只是这个趋势什么时候产生我们不得而知。我们透过对价格的不断跟踪，用支付成本不断尝试的办法去追踪中级以上的趋势，当趋势发生以后，量化交易系统会去跟随这个趋势。趋势交易都是围绕着趋势理论来，通过趋势理论画趋势线等。

### 3.2.1 趋势理论概述

趋势理论是指一旦市场形成了下降(或上升)的趋势后，就将沿着下降(或上升)的方向运行。

根据趋势理论，数字货币价格运动有三种趋势，其中最主要的是数字货币的基本趋势，即币价广泛或全面性上升或下降的变动情形。对投资者来说，基本趋势持

续上升就形成了多头市场，持续下降就形成了空头市场。币价运动的第二种趋势称为币价的次级趋势。因为次级趋势经常与基本趋势的运动方向相反，并对其产生一定的牵制作用，因而也称为币价的修正趋势。币价运动的第三种趋势称为短期趋势，反映了币价在几天之内的变动情况。修正趋势通常由3个或3个以上的短期趋势所组成。



在三种趋势中，长期投资者最关心的是币价的基本趋势，其目的是想尽可能地在多头市场上买入数字货币，而在空头市场形成前及时地卖出数字货币。投机者则对币价的修正趋势比较感兴趣。他们的目的是想从中获取短期的利润。短期趋势的重要性较小，且易受人为操纵，因而不便作为趋势分析的对象。

## 3.2.2 趋势理论分析方法

### 3.2.2.1 根据成交量判断趋势的变化

成交量会随着主要的趋势而变化。因此，根据成交量也可以对主要趋势做出一个判断。通常，在多头市场，价位上升，成交量增加；价位下跌，成交量减少。在空头市场，当价格滑落时，成交量增加；在反弹时，成交量减少。当然，这条规

则有时也有例外。因此，只是根据几天的成交量是很难得出正确的结论，只有在持续一段时间的整个交易的分析中才能够得出。在趋势理论中，为了判定市场的趋势，最终结论性信号只由价位的变动产生。成交量仅仅是在一些有疑问的情况下提供解释的参考。

### 3.2.2.2 盘局可以代替中级趋势

一个盘局出现于一种或两种指数中，持续了两个或三个星期，有时达数月之久，价位仅在约5%的距离中波动。这种形状显示买进和卖出两者的力量是平衡。当然，最后的情形之一是，在这个价位水准的供给完毕了，而那些想买进的人必须提高价位来诱使卖者出售。另一种情况是，本来想要以盘局价位水准出售的人发觉买进的气氛削弱了，结果他们必须削价来处理他们的数字货币。因此，价位往上突破盘局的上限是多头市场的征兆。相反价位往下跌破盘局的下限是空头市场的征兆。一般来说，盘局的时间愈久，价位愈窄，它最后的突破愈容易。

盘局常发展成重要的顶部和底部，分别代表着出货和进货的阶段，但是，它们更常出现在主要趋势时的休息和整理的阶段。在这种情形下，它们取代了正式上的次级波动。很可能一种指数正在形成盘局，而另一种却发展成典型的次级趋势。在往上或往下突破盘局后，有时在同方向继续停留一段较长的时间，这是不足为奇的。

### 3.2.2.3 主要趋势不会立马反转

在反转趋势出现之前主要趋势仍将发挥影响，不会立马反转，当然，在反转信号出现前，提前改变对市场的态度，就好比赛跑时于发出信号前抢先跑出。这条规则也意味着在趋势反转信号已经明朗化以后，一个人还应再迟延一下他的行动，而是说在经验上，我们等到已经确定了以后再行动较为有利，以避免在还没有成熟前买进(或卖出)。自然，币价主变动趋势是在经常变化着的。多头市场并不能永远持续下去，空头市场总有到达底的一天。当一个新的主要趋势第一次由两种指数确定后，如不管短期间的波动，趋势绝大部分会持续，但愈往后这种趋势延势下去的可能性会愈小。这条规则告诉人们：一个旧趋势的反转可能发生在新趋势被确认后的任何时间，作为投资人，一旦作出委托后，必须随时注意市场。

### 3.2.3 趋势线

所谓趋势线，就是根据币价上下变动的趋势所画出的线路，画趋势线的目的，即依其脉络寻找出恰当的卖点与买点。趋势线可分为上升趋势线，下降趋势线与横向整理趋势线。

币价在上升行情时，除了在连接币价波动的低点画一直线外，也应在连接币价波动的高点画一直线，于是币价便在这两条直线内上下波动，这就是上升趋势轨道。币价在下跌行情时，除了连接币价波动的高点画一直线外，也要在币价波动的低点画一条直线，币价在这两条直线内上下波动，这就是下跌趋势轨道。币价在横向整理时可形成横向箱型趋势线。

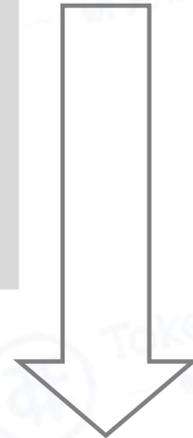
### 3.2.4 利用趋势轨道决定买卖点

无论是在上升或下跌趋势轨道中，当币价触及上方的压力线时，就是卖出的时机；当币价触及下方的支撑线时，就是买进的时机。



若在上升趋势轨道中，发现币价突破上方的压力线时，证明新的上升趋势线即将产生

同理，若在下跌趋势中，发现币价突破下方的支撑线时，可能新的下跌趋势轨道即将产生



币价在上升行情时，一波的波峰会比前一波峰高，一波的波谷会比前一波谷高；而在下跌行情时，一波的波峰比前一波峰低，一波的波谷会比前一波谷低。处于上升趋势轨道中，若发现币价无法触及上方的压力线时，即表示涨势趋弱了。

### 3.2.5 趋势理论的缺陷

趋势理论主要目标乃探讨币市的基本趋势。一旦基本趋势确立，趋势理论假设这种趋势会一路持续，直到趋势遇到外来因素破坏而改变为止。好像物理学里牛顿定律所说，所有物体移动时都会以直线发展，除非有额外因素力量加诸其上。但有一点要注意的是，趋势理论只推断股市的大势所趋，却不能推动大趋势里面的升幅或者跌幅将会达到哪个程度。趋势理论注重长期趋势，对中期趋势，特别是

在不知是牛还是熊的情况下，不能带给投资者明确启示。

## 3.3 马丁格尔策略 BTC 应用

### 3.3.1 马丁格尔策略理论

马丁格尔 (Martingale) 策略的基本概念就是利用“机率”的方法来达到自己的胜算越来越高，然后依照“典型启发” (Representativeness Heuristic) 理论“连续六次机率”的一个临界点，来让自己在市场上处于不败之地。但是重点就是只能挑“长期震荡”的货币市场，否则很容易赔光资金。

简化版：先开仓，指定一个盈利点平仓，如果平不了，超过止损点，就再加仓拉低成本价，一直加到盈利出场，如果加仓加到没钱或者币再加了，就开始反手做，原来是做空的，就反手做多，这样一直反复。也可以不让自动反手，让程序一直等着解套。

比如现在 10 元均价开的多仓，目标盈利是 5 毛钱，程序挂一个 10.5 元卖出的单，同时挂一个低价加仓的买单，如果价格跌了，就加仓到当前价格的附近，调整均价，再挂一个目标盈利单，如果还不行，继续加仓拉低均价，实在不行，被套了，就反手开始做空单。

### 3.3.2 马丁格尔策略优缺点

马丁格尔策略的优点和诱惑力是在每一次赔钱后加码双倍，因此下一次赢钱时，你总能赢回之前所有的损失，外加一单位赌注(可以在纸上演算)；然而在一连串的损失发生时，赌注规模会增加得非常快速，对资金量的要求和连续亏损后的成本激增，连续亏损后极大的破产风险，你可能在赌赢之前，就破产了。

由于马丁格尔策略的破产的风险性过大，于是出现了反马丁格尔策略 (Anti-Martingale)。反马丁格尔策略跟马丁格尔策略都是一种赌博策略 (Betting Strategy)。但是反马丁格尔策略跟马丁格尔策略不管是在概念上还是在实际的操作都是完全相反，下表是两策略特点的比较。

马丁格尔策略	反马丁格尔策略
亏损时加码	获利时加码
在第二次进场之后的平均价格成本会高于目前市场价格	在第二次进场之后的平均价格成本会低于目前市场价格
小额短期获利后会有大额亏损可能	小额短期亏损后会有大额获利可能
潜在无限亏损之可能性	潜在无限获利之可能性
无停损仓位，帐户亏损后加码再进场	有停损仓位，帐户亏损后不会加码进
浮动亏损会造成整个帐户资金调度困	浮动利益会帮助整个帐户资金调度宽

### 3.3.3 马丁格尔策略回测 BTC

基于马丁格尔策略理论写一个量化程序来回测 BTC。

### 3.3.3.1 策略的自动反手机制

如果当前做多仓，盈利出局，会自动再做多仓，如果被套，反手拿币做空仓，空仓盈利后，继续做空仓，一直到加仓加不动了，反手开始做多仓，这样一直循环。只要上次是盈利的就一直保持着上次的开仓方向，策略会自动计算需要加仓的量的大小跟目标盈利点的新值，需要注意的：

此策略有完整的恢复机制，可以实际操作或者学习使用

此策略 100%赚钱的前提是：你得准备好足够的资金加仓

如果没有足够的资金，就让自动反手做，会产生浮动盈亏

如果你有足够的资金，就不需要自动反手了，一直让程序加仓

策略可以通过调整参数，实现高频的盘口型策略

不适合期货，期货这样搞容易爆仓，所以只能现货来操作

浮动盈亏指  $((\text{当前币数量} - \text{初始币数量}) * \text{当前币价格}) + (\text{当前钱} - \text{初始钱})$

### 3.3.3.2 策略核心代码

```
320 while (true) {
321     Sleep(Interval);
322     var ticker = EnsureCall(exchange, "GetTicker");//获取此刻行情
323     var orders = EnsureCall(exchange, "GetOrders");//获得所有未完成的订单
324     LastTicker = ticker;//保存此刻行情
325     var nowAccount = GetAccount(exchange); //获取此刻账户信息, 不等待冻结
326     var diff = nowAccount.Stocks + nowAccount.FrozenStocks - preAccount.Stocks; //此刻账户总市值 - 加仓前账户市值
327     //如果 当前未完成的订单数不等于 订单计数 (就是有完成的订单) 或 diff的绝对值大于等于 最小交易数
328     if (orders.length != order_count || Math.abs(diff) >= minStock) {
329         StripOrders(exchange);//取消所有未完成的订单
330         nowAccount = GetAccount(exchange, true);//获取此刻账户信息
331         //Log(nowAccount);//测试用
332         var diffAmount = nowAccount.Stocks - initAccount.Stocks;//计算币差量
333         var diffMoney = nowAccount.Balance - initAccount.Balance;//计算钱差量
334         if (Math.abs(diffAmount) < minStock) {
335             AllProfit = _N(AllProfit + (holdAmount * ProfitGoal), 4); //更新盈利
336             LogProfit(AllProfit, "平仓完成, 达到目标盈利点, 单次盈利", _N(holdAmount * ProfitGoal, 4));
337             initAccount = nowAccount;//更新initAccount
338             isFinished = true;//外层while循环 结束标记 为true
339             if (!canOpen) { //如果不能加仓
340                 Counter.f++; //解套计数累计1
341             }
342             break;//跳出当前while
343         }
344         var newHoldPrice = 0;//新持仓价
345         var newHoldAmount = 0;//新持仓量
346         if (TradeType == ORDER_TYPE_BUY) { //如果为 多 方向
347             newHoldAmount = _N(diffAmount, 4);
348             newHoldPrice = _N((-diffMoney) / diffAmount, 4);
349         } else { //空 方向
350             newHoldAmount = _N(-diffAmount, 4);
351             newHoldPrice = _N(diffMoney / (-diffAmount), 4);
352         }
353         // if open again, we need adjust hold positions's price
354         var isAdd = false;
355         if (newHoldAmount > holdAmount) { //如果新持仓量 大于 持仓量
356             holdPrice = newHoldPrice;//更新 持仓均价
357             isAdd = true;// 设置可以加仓
358         }
359         holdAmount = newHoldAmount;//更新持仓量
360         if (!isAdd) { //如果不能加仓 持仓前账户调整
361             // 重置 initAccount
362             initAccount = {
363                 Stocks : nowAccount.Stocks,
364                 Balance : nowAccount.Balance,
365                 FrozenBalance : nowAccount.FrozenBalance,
366                 FrozenStocks : nowAccount.FrozenStocks,
367             };
368             if (TradeType == ORDER_TYPE_BUY) { //多仓方向
369                 initAccount.Stocks -= holdAmount;
370                 initAccount.Balance += holdAmount * holdPrice;
371             } else {
372                 initAccount.Stocks += holdAmount;
373                 initAccount.Balance -= holdAmount * holdPrice;
374             }
375             initAccount.Stocks = _N(initAccount.Stocks, 4); //处理数值
376             initAccount.Balance = _N(initAccount.Balance, 4); //处理数值
377             Log("持仓前账户调整为: ", initAccount);
378         }
379         Log((TradeType == ORDER_TYPE_BUY ? "多仓" : "空仓"), (isAdd ? "加仓后" : "平仓后"), "重新调整持仓, 均价: ", holdPrice, "数量", holdAmount);
380         Log("买一:", ticker.Buy, "卖一:", ticker.Sell, "上次成交价:", ticker.Last);
381         Log(nowAccount);
382         break;
383     }
384 }
```

上图是策略核心代码，代码中有详细的注释，也是按照马丁格尔策略理论进行调仓，反手等操作。

通过修改参数可以调整策略运行机制，下图是本策略核心参数。

参数	描述	默认值
OpType	首次开仓方向	做多 做空
OpAmount	开仓数量	0.1
OpMode	开仓方式	吃单 挂单
MaxSpace@OpMode==1	挂单失效距离	0.5
SlidePrice	下单滑动价(元)	0.1
MaxAmount	开仓最大单次下单量	0.3
AddGoal	加仓间距(元)	1
AddLine	加仓均价目标(元)	0.8
ProfitGoal	平仓目标(元)	0.5
Interval	轮询间隔(秒)	1
RestoreIt	恢复进度	FALSE
RestoreType@RestoreIt	持仓方向	做多 做空
RestorePrice@RestoreIt	持仓均价	0
RestoreAmount@RestoreIt	持仓数量	0
RestoreProfit@RestoreIt	上次盈利	0
SaveLocal	保存本地日志	FALSE
AutoReverse	自动反手	TRUE
MinStock	最小交易币数	0.01

### 3.3.3.3 回测及分析

接下来测试一下这个简单的动态平衡策略，以 OKCoin 国际 2018 年 1~7 月份数据作为训练数据，以 OKCoin 国际 2018 年 8 月份数据做为样本外测试数据。

训练时主要针对下面几个参数进行了调优，如下图：

参数	描述	最小值	最大值	步长
AddGoal	加仓间距(元)	1	2	0.2
AddLine	加仓均价目标(元)	0.1	2	0.2
ProfitGoal	平仓目标(元)	0.1	2	0.2

就是对上面几个参数按步长进行排列组合，排列组合测试轮数计算公式：

$$(\text{Math.ceil}((\text{参数 1 最大值}-\text{参数 1 最小值})/\text{参数 1 步长})+1)*(\text{Math.ceil}((\text{参数 2 最大值}-\text{参数 2 最小值})/\text{参数 2 步长})+1)*(\text{Math.ceil}((\text{参数 2 最大值}-\text{参数 2 最小值})/\text{参数 2 步长})+1)*\dots$$

说明：Math.ceil 函数为向上取整。

在本次调优中测试轮数为

$$(\text{Math.ceil}((2-1)/0.2)+1)*(\text{Math.ceil}((2-0.1)/0.2)+1)*(\text{Math.ceil}((2-0.1)/0.2)+1)=726$$

训练结果如下图所示：

#	加仓间距(元)	加仓均价目标(元)	平仓目标(元)	耗时(秒)	交易次数	胜率(%)	夏普	预估收益	利润
0	1	0.1	0.1	5.499	14774	100	0.118	-302795.559	106.13
1	1	0.1	0.3	4.119	19292	100	0.13	-514483.538	398.46
2	1	0.1	0.5	3.913	16716	100	0.125	-437213.256	613.35
3	1	0.1	0.7	4.094	17694	100	0.131	-485741.457	921.97
4	1	0.1	0.9	4.001	17690	100	0.131	-486101.238	1178.1
5	1	0.1	1.1	3.894	17690	100	0.131	-486099.957	1439.9
6	1	0.1	1.3	4.085	17690	100	0.131	-486094.711	1701.7
7	1	0.1	1.5	4.731	25502	100	0.153	-617099.294	2545.95
...	...	...	...	...	...	...	...	...	...

经过分析，发现几个参数各自区间，如下图配置所示是为较优结果。

2018-01-01 00:00:00 - 2018-08-01 00:00:00

1分钟

模拟级 Tick

OKCoin国际

BTC

余额 10000

余币 300

绘制行情图表

Maker 0.15

Taker 0.2

OKCoin国际 / BTC x

### 策略参数

首次开仓方向

做多

开仓数量

0.1

调优

开仓方式

吃单

下单滑动价(元)

0.1

调优

开仓最大单次下单量

0.3

调优

加仓间距(元)

1

调优

加仓均价目标(元)

1

调优

平仓目标(元)

1

调优

轮询间隔(秒)

1

调优

恢复进度

(勾上为True)

保存本地日志

(勾上为True)

自动反手

(勾上为True)

最小交易币数

0.01

调优

1 ~ 7 月份回测参数配置图

回测日志 - 进度: 100.0 % - 耗时(秒): 2.778 秒 - 日志总数: 23047 - 交易次数: 7583 - 超过最大限度 8000 - 已完成 (Tick 120.53 KB)

账户信息

名称	定价货币	交易品种	手续费	预估收益 <span style="font-size: 0.8em;">?</span>
OKCoin_EN	USD	BTC	USD: 265459.65	75527.312

状态信息

成功: 463 次, 被套: 0 次, 当前账户 钱: 87164.236 币: 9999.757 平仓盈亏: 1332.122, 浮动盈亏: 75285.3259 (初始账户 钱: 10000 币: 10000)

1 ~ 7 月份回测结果绩效图



1 ~ 7 月份回测收益曲线图

接着用上述参数配置去测试 8 月份数据：

回测日志 - 进度: 100.0 % - 耗时(秒): 1.117 秒 - 日志总数: 834 - 交易次数: 275 - 已完成 (Tick 1.67 MB)

账户信息

名称	定价货币	交易品种	手续费	预估收益
OKCoin_EN	USD	BTC	USD: 27.43	-1029.3508

状态信息

成功: 1 次, 被套: 0 次, 当前账户 钱: 56.678 币: 4.28 平仓盈亏: 0.28, 浮动盈亏: -87.322 (初始账户 钱: 10000 币: 3)

8 月份回测结果绩效图



8 月份回测收益曲线图



BTC k线图

从上述回测结果看，虽然在训练阶段调出了“最优”参数，能在1~7月份盈利，但是当拿“最优”参数去回测8月份数据的时候，结果是亏损的。当然还有其他参数没调，以及这三个参数也还有其他区间没调。

## 3.4 动态平衡策略 BTC 量化

### 3.4.1 动态平衡策略理论

动态平衡策略最先是由沃伦·巴菲特的导师本杰明·格雷厄姆在《聪明的投资者》一书中提出。描述的是一种股票债券动态平衡的交易模式。

这种交易模式非常简单：



把手中 50% 的资金投资于，剩下 50% 投资于债券基金。即股票和债券两者各占一半

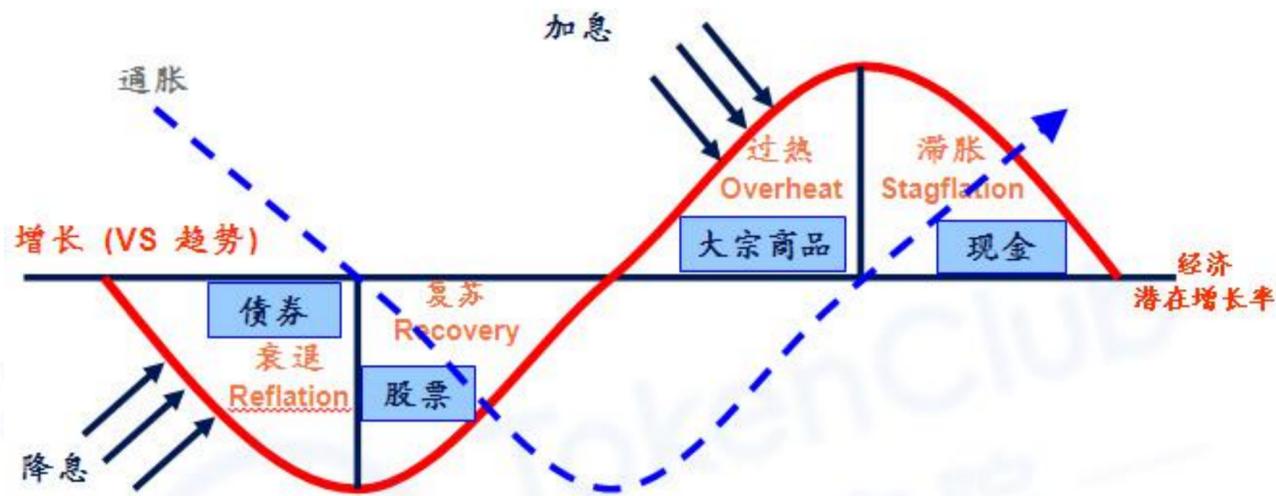


根据固定间隔时间或市场变化进行一次资产再平衡，使股票资产和债券资产的比例恢复到初始的 1:1

这就是整个策略的全部逻辑，包含了什么时候买卖，以及买卖多少。

在这个方法中，债券基金的波动率其实很小，远远低于股票波动率，所以债券在这里被当做“参照锚”，也就是说，用债券来衡量股票究竟是涨得太多了，还是涨得太少了。如果，股票价格上涨，会使得股票的市值大于债券的市值，当两者市值比率超过设定的阈值时，则对总仓位进行重新调整，卖出股票，并且买入债券，使股债市值比例恢复至初始的 1:1。

反之，股票价格下跌，会使得股票的市值小于债券的市值，当两者市值比率超过设定的阈值时，则对总仓位进行重新调整，买入股票，并且卖出债券，使股债市值比例恢复至初始的 1:1。



就这样，在动态平衡股票和债券之间的比例，就够享受到股票成长的果实，并且减少了资产波动率。作为价值投资的先驱，格雷厄姆为我们提供了一个很好的思路。

### 3.4.2 动态平衡策略优点

动态平衡策略实际是做出了高抛低吸的操作。在币价上涨时，占比上升，卖出；在币价下跌时，占比下降，买入。类似于网格策略。同时，相比于网格，此策略不会出现在暴涨暴跌时变成满仓或者空仓的状态。而且，由于不会达到满仓状态，在遇到特殊行情时，投资者可以十分灵活的使用剩余的现金，而不会出现满仓没有资金可用的尴尬局面。

### 3.4.3 动态平衡策略缺点

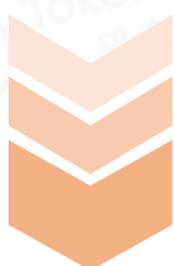
动态平衡由于本质上是做高抛低吸动作，在震荡行情中会表现的更好。而在单边上涨的行情中，很难跑赢基准。

### 3.4.4 动态平衡策略回测 BTC

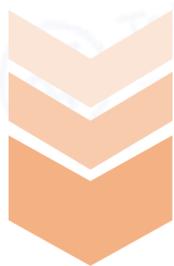
既然这是一个完整的策略，为何我们不把它用在数字货币上呢？

下面演示一个在 BTC 的上应用例子：

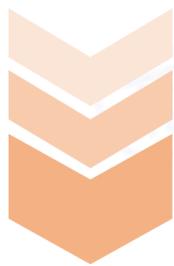
#### 3.4.4.1 策略逻辑



按照当前的 BTC 的价值，账户余额保留 ¥ 5000 现金和 0.1 个 BTC，即现金和 BTC 市值的初始比例是 1:1



如果 BTC 的价格上涨至 ¥ 6000，即 BTC 市值大于账户余额，并且其之间的差超过设定的阈值，就卖掉 $(6000-5000)/6000/2$  个币。说明 BTC 升值了，把钱兑换回来



如果 BTC 的价格下跌至 ¥ 4000，即 BTC 市值小于账户余额，并且其之间的差超过设定的阈值，就买入 $(5000-4000)/4000/2$  个币。说明 BTC 贬值了，把 BTC 买回来

就这样，不管 BTC 是升值还是贬值，始终动态保持账户余额和 BTC 的市值相等。如果 BTC 贬值了就买一些，等再涨回来，就再卖一些，就好像天平一样。

### 3.4.4.2 策略框架

```

1 //策略参数
2 var threshold = 0.05; //阈值
3 var LoopInterval = 60; //轮训间隔 (秒)
4 var MinStock = 0.001; //最小交易量
5 var XPrecision = 4; //量精度
6 var ZPrecision = 8; //价格精度
7
8 // 撤单函数
9 function CancelPendingOrders() {
10
11 }
12
13 // 下单函数
14 function onTick() {
15
16 }
17
18 // 主函数
19 function main() {
20     // 过滤非重要信息
21     SetErrorFilter("GetRecords:|GetOrders:|GetDepth:|GetAccount|:Buy|Sell|timeout");
22     while (true) { // 轮询模式
23         if (onTick()) { // 执行 onTick 函数
24             CancelPendingOrders(); // 取消未成交的挂单
25             Log(_C(exchange.GetAccount)); // 打印当前账户信息
26         }
27         Sleep(LoopInterval * 1000); // 休眠
28     }
29 }

```

整个策略框架其实很简单，一个 main 主函数、一个 onTick 下单函数、一个 CancelPendingOrders 函数、以及必要参数。

程序参数如下：

参数	描述	默认值
threshold	阈值	0.005
Interval	出错重试间隔(毫秒)	2000
LoopInterval	轮询间隔(秒)	60
MinStock	最小交易量	0.001
XPrecision	量精度	4
ZPrecision	价格精度	8

### 3.4.4.3 下单模块

```
34 // 下单函数
35 function onTick() {
36     var acc = _C(exchange.GetAccount); // 获取账户信息
37     var ticker = _C(exchange.GetTicker); // 获取 Tick 数据
38     var spread = ticker.Sell - ticker.Buy; // 获取 Tick 数据的买卖价差
39     // 账户余额与当前持仓价值的差值的 0.5倍
40     var diffAsset = (acc.Balance - (acc.Stocks * ticker.Sell)) / 2;
41     var ratio = diffAsset / acc.Balance; // diffAsset / 账户余额
42     LogStatus('ratio:', ratio, _D()); // 打印 ratio和当前时间
43     if (Math.abs(ratio) < threshold) { // 如果 ratio的绝对值小于指定阈值
44         return false; // 返回 false
45     }
46     if (ratio > 0) { // 如果 ratio大于 0
47         var buyPrice = _N(ticker.Sell + spread, ZPrecision); // 计算下单价格
48         var buyAmount = _N(diffAsset / buyPrice, XPrecision); // 计算下单量
49         if (buyAmount < MinStock) { // 如果下单量小于最小交易量
50             return false; // 返回 false
51         }
52         exchange.Buy(buyPrice, buyAmount, diffAsset, ratio); // 买入下单
53     } else {
54         var sellPrice = _N(ticker.Buy - spread, ZPrecision); // 计算下单价格
55         var sellAmount = _N(-diffAsset / sellPrice, XPrecision); // 计算下单量
56         if (sellAmount < MinStock) { // 如果下单量小于最小交易量
57             return false; // 返回 false
58         }
59         exchange.Sell(sellPrice, sellAmount, diffAsset, ratio); // 卖出下单
60     }
61     return true; // 返回 true
62 }
```

下单交易逻辑条理清晰，所有的注释都已经写到代码里面了。

主要流程如下：

- 获取账户信息
- 获取 Tick 数据
- 计算 Tick 数据买卖价差
- 计算账户余额和 BTC 市值价差
- 计算买卖条件、下单价格、下单量
- 下单

### 3.4.4.4 撤单模块

```
11 // 撤单函数
12 function CancelPendingOrders() {
13     Sleep(1000); // 休眠 1秒
14     var ret = false;
15     while (true) {
16         var orders = null;
17         // 持续获取未成交订单数组, 如果返回异常, 则继续获取
18         while (!(orders = exchange.GetOrders())) {
19             Sleep(1000); // 休眠 1秒
20         }
21         if (orders.length == 0) { // 如果订单数组为空
22             return ret; // 返回撤单状态
23         }
24         for (var j = 0; j < orders.length; j++) { // 遍历未成交订单数组
25             exchange.CancelOrder(orders[j].Id); // 依次取消未成交订单
26             ret = true;
27             if (j < (orders.length - 1)) {
28                 Sleep(1000); // 休眠 1秒
29             }
30         }
31     }
32 }
```

撤单模块就更简单了，步骤如下：

- 停顿 1 秒，撤单太快会被交易所封号
- 持续获取未成交订单数组，如果返回异常，则继续获取
- 如果未成交订单数组为空，即时返回撤单状态
- 如果有未成交的订单，则遍历整个数组，并依次根据订单号撤单

### 3.4.4.5 回测及分析

接下来测试一下这个简单的动态平衡策略，看看到底有没有效果。方法也跟前面的马丁格尔回测分析一样。以 OKCoin 国际 2018 年 1 ~ 7 月份数据作为训练数据，以 OKCoin 国际 2018 年 8 月份数据做为样本外测试数据。选取如下两个参数进行调试。

参数	描述	最小值	最大值	步长
LoopInterval	轮询间隔(秒)	10	90	10
MinStock	最小交易量	0.001	0.1	0.01

在本次调优中测试轮数为

$$(\text{Math.ceil}((90-10)/10)+1)*(\text{Math.ceil}((0.1-0.01)/0.01)+1)=90$$

训练结果如下图所示：

#	轮询间隔(秒)	最小交易量	耗时(秒)	交易次数	胜率(%)	夏普	预估收益	利润
0	10	0.01	56.024	14544	100	0.145	20244.51	929009.32
1	10	0.02	65.823	7624	100	0.203	18936.59	824820.4
2	10	0.03	49.843	3496	100	0.217	16892.18	73307.2472
3	10	0.04	47.17	1986	100	0.127	15537.05	624921.7416
4	10	0.05	49.253	1180	100	0.127	14127.09	53164.536
5	10	0.06	49.426	712	100	0.279	12457.72	356738.9166
6	10	0.07	50.24	534	100	0.265	12266.1	355531.3109
7	10	0.08	61.159	346	100	0.251	10855.89	295533.0098
8	10	0.09	83.148	268	100	0.216	10367.32	256431.4879
9	10	0.1	72.208	186	100	0.17	9620.13	195372.6
10	20	0.01	58.478	14366	100	0.221	20480.91	233747.5737
11	20	0.02	67.978	7840	100	0.257	19514.24	193489.8826
...	...	...	...	...	...	...	...	...

经过分析，发现几个参数各自区间，如下图配置所示是为较优结果。

1 ~ 7 月份回测参数配置图

回测日志 - 进度: 100.0 % - 耗时(秒): 52.903 秒 - 日志总数: 13452 - 交易次数: 6724 - 超过最大限度 8000 - 已完成 (Tick 11.32 MB)

账户信息

名称	定价货币	交易品种	手续费	预估收益
OKCoin_EN	USD	BTC	USD: 2888.229	18327.6054

1 ~ 7 月份回测结果绩效图

初始净值：10000, 累计收益：183.276%, 年化收益(365天)：314.865%, 夏普比率：0.195, 年化波动率：16.157%, 最大回撤：37.447%



1~7月份回测收益曲线图

回测日志 - 进度: 100.0% - 耗时(秒): 8.083 秒 - 日志总数: 636 - 交易次数: 317 - 已完成 (Tick 1.70 MB)

账户信息

名称	定价货币	交易品种	手续费	预估收益
OKCoin_EN	USD	BTC	USD: 116.057	1615.42865

8月份回测结果绩效图

初始净值：10000, 累计收益：16.154%, 年化收益(365天)：187.433%, 夏普比率：0.273, 年化波动率：6.860%, 最大回撤：3.780%



8月份回测收益曲线图

从上述回测结果看，交易量较小，行情越震荡下，收益越高，符合动态平衡策略特征。对于一个这么简单的投资策略，这个投资回报率已经超过绝大多数梭哈群众了。

该动态平衡策略，只有一个核心参数（threshold 阈值），是一个很简单的投资方法，追求的不是超额的收益，而是稳健的收益。与趋势策略相反，动态平衡策略却是逆势而动。在市场热的时候减仓降温，市场冷清的时候加仓蛰伏，有点类似宏观经济调控。其实，动态平衡策略正是秉承了价格不可预测的观念，同时又捕捉价格波动的一门手艺。动态平衡策略的关键核心在设定和调整资产配置比例，还有触发阈值。

动态平衡策略最重要的是投资思想，你甚至可以把单个 BTC 资产换成一篮子数字货币组合。

### 3.5 运用机器学习预测币价

近年来不近是区块链技术火热，人工智能也豪不逊色，特别是机器学习。机器学习如此强大，那怎么跟数字货币交集起来？怎么来预测比特币的价格？目前市面上已有一些价格预测模型，但是由于比特币市场价格波动较大，不确定太多，基本都很难在生产环境中体现出效果。接下来演示一下的预测比特币和 TCT 的价格流程。

### 3.5.1 数据收集

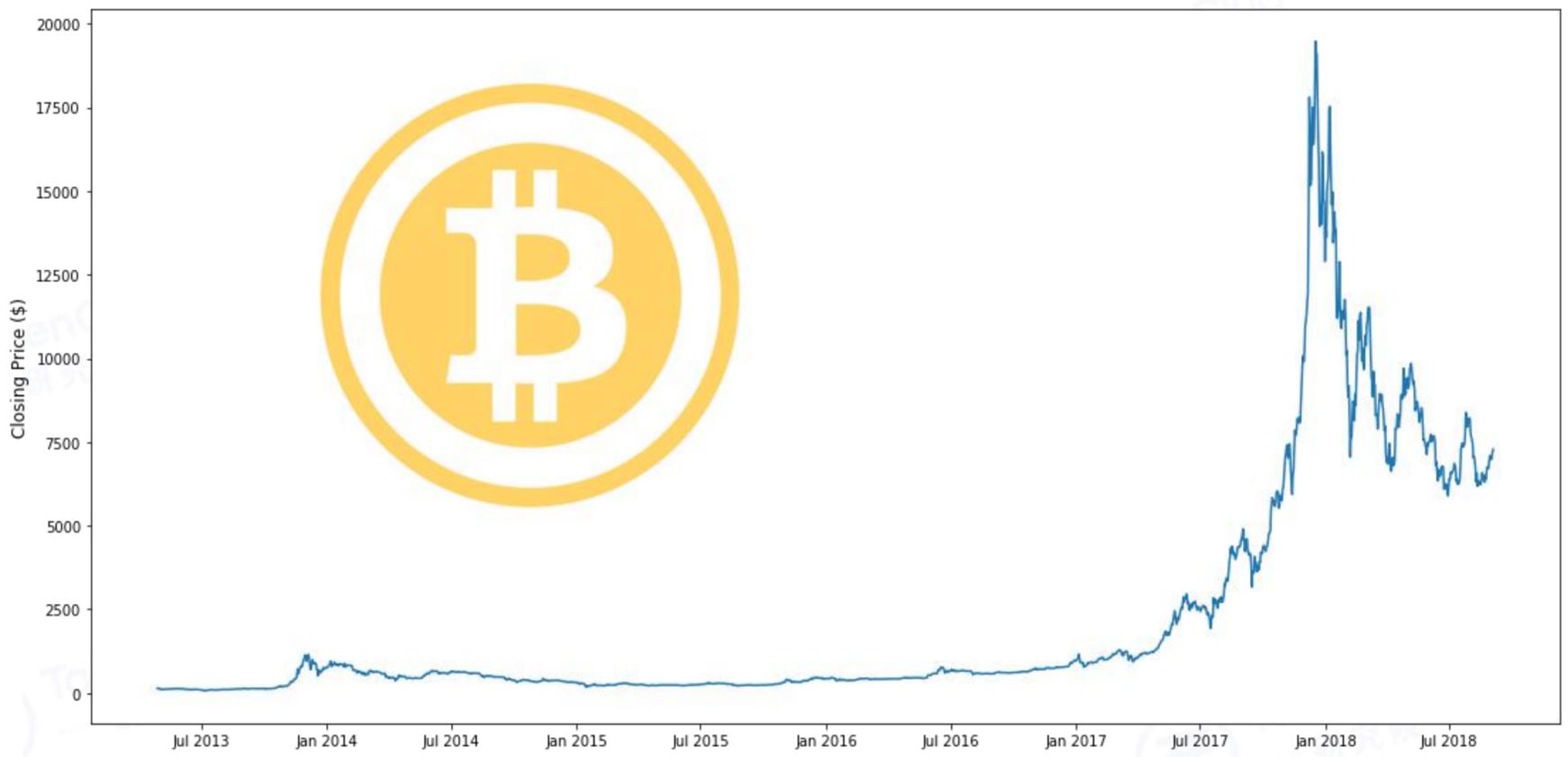
在搭建模型之前，需要获取一些数据让模型学习。在深度学习中，在严重缺乏数据的情况下，模型基本不太可能取得成功。从 coinmarketcap 提取数据。我们只考虑比特币和 TCT 的数据，按照这种方式添加其他币种也不难。下图是完成导入数据代码。

```
import pandas as pd
import time
import seaborn as sns
import matplotlib.pyplot as plt
import datetime
import numpy as np

# get market info for bitcoin from the start of 2016 to the current day
bitcoin_market_info = pd.read_html("https://coinmarketcap.com/currencies/bitcoin/historical-data/?start=20130428&end="
                                   +time.strftime("%Y%m%d"))[0]
# convert the date string to the correct date format
bitcoin_market_info = bitcoin_market_info.assign(Date=pd.to_datetime(bitcoin_market_info['Date']))
# when Volume is equal to '-' convert it to 0
bitcoin_market_info.loc[bitcoin_market_info['Volume']=='-', 'Volume']=0
# convert to int
bitcoin_market_info['Volume'] = bitcoin_market_info['Volume'].astype('int64')
# sometime after publication of the blog, coinmarketcap starting returning asterisks in the column names
# this will remove those asterisks
bitcoin_market_info.columns = bitcoin_market_info.columns.str.replace("*", "")
# look at the first few rows
# bitcoin_market_info.head()

# get market info for TCT from the start of 2016 to the current day
tct_market_info = pd.read_html("https://coinmarketcap.com/currencies/tokenclub/historical-data/?start=20130428&end="
                               +time.strftime("%Y%m%d"))[0]
# convert the date string to the correct date format
tct_market_info = tct_market_info.assign(Date=pd.to_datetime(tct_market_info['Date']))
# sometime after publication of the blog, coinmarketcap starting returning asterisks in the column names
# this will remove those asterisks
tct_market_info.columns = tct_market_info.columns.str.replace("*", "")
# look at the first few rows
# tct_market_info.head()
```

为了能快速看到数据是准确的，我们绘制出两种币的收盘价走势图：



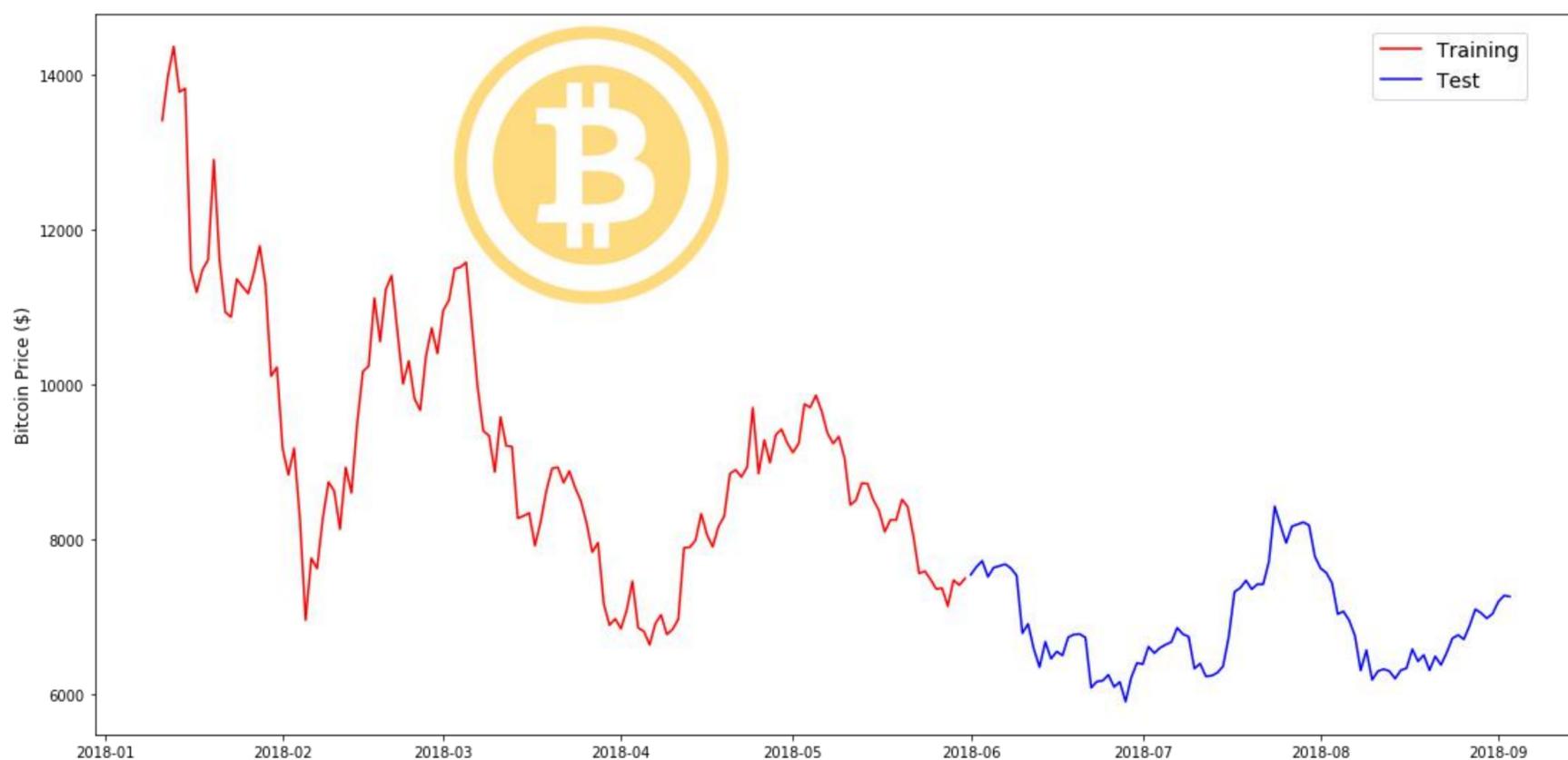
比特币收盘价走势图



TCT 收盘价走势图

### 3.5.2 训练、测试、随机游走模型

数据已经有了，那么现在需要一个模型。在深度学习中，数据通常被分为训练集和测试集两部分。我们用训练集让模型进行学习，然后用测试集评估模型的性能。对于时间序列模型，我们一般对一个时间序列进行预测，然后对另一个时间序列进行测试。比如，我把截止日期设为2018年6月1日，那么就会用在这个日期之前的时间训练模型，用这个日期之后的数据评估模型。



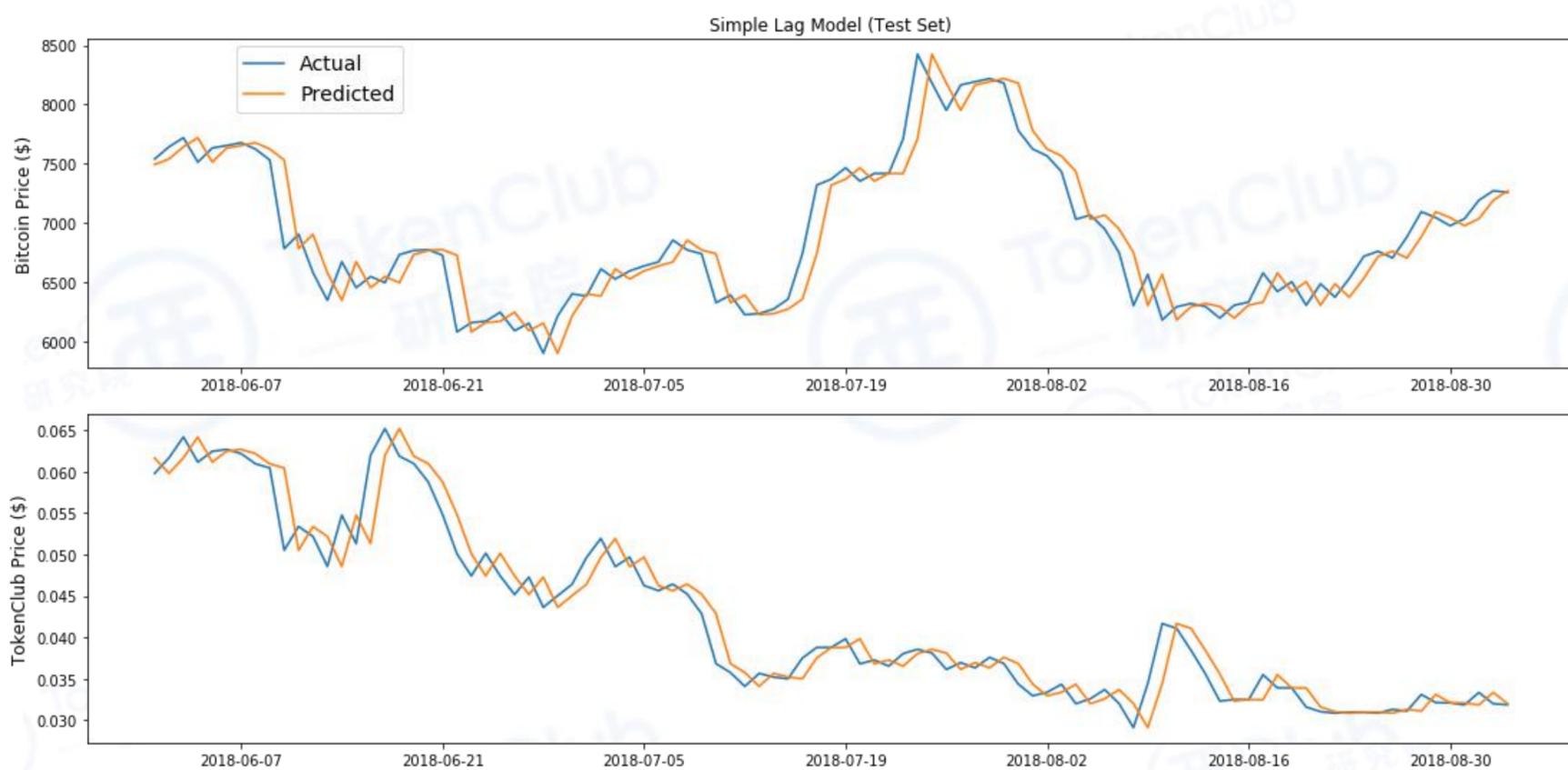
比特币训练测试收盘价走势图



TCT 训练测试收盘价走势图

你可以看到训练时间大部分包含虚拟货币价格较低的时期。这样，训练数据或许无法代表测试数据，会损害模型泛化不可见数据的能力（你可以试着让数据更平稳些）。在使用我们的机器学习模型前，有必要讨论一个更简单些的模型。最基本的模型就是设定明天的价格等于今天的价格（我们称之为滞后模型）。我们在数学上以如下方式定义该模型：

$$PredPrice_t = ActualPrice_{t-1}$$



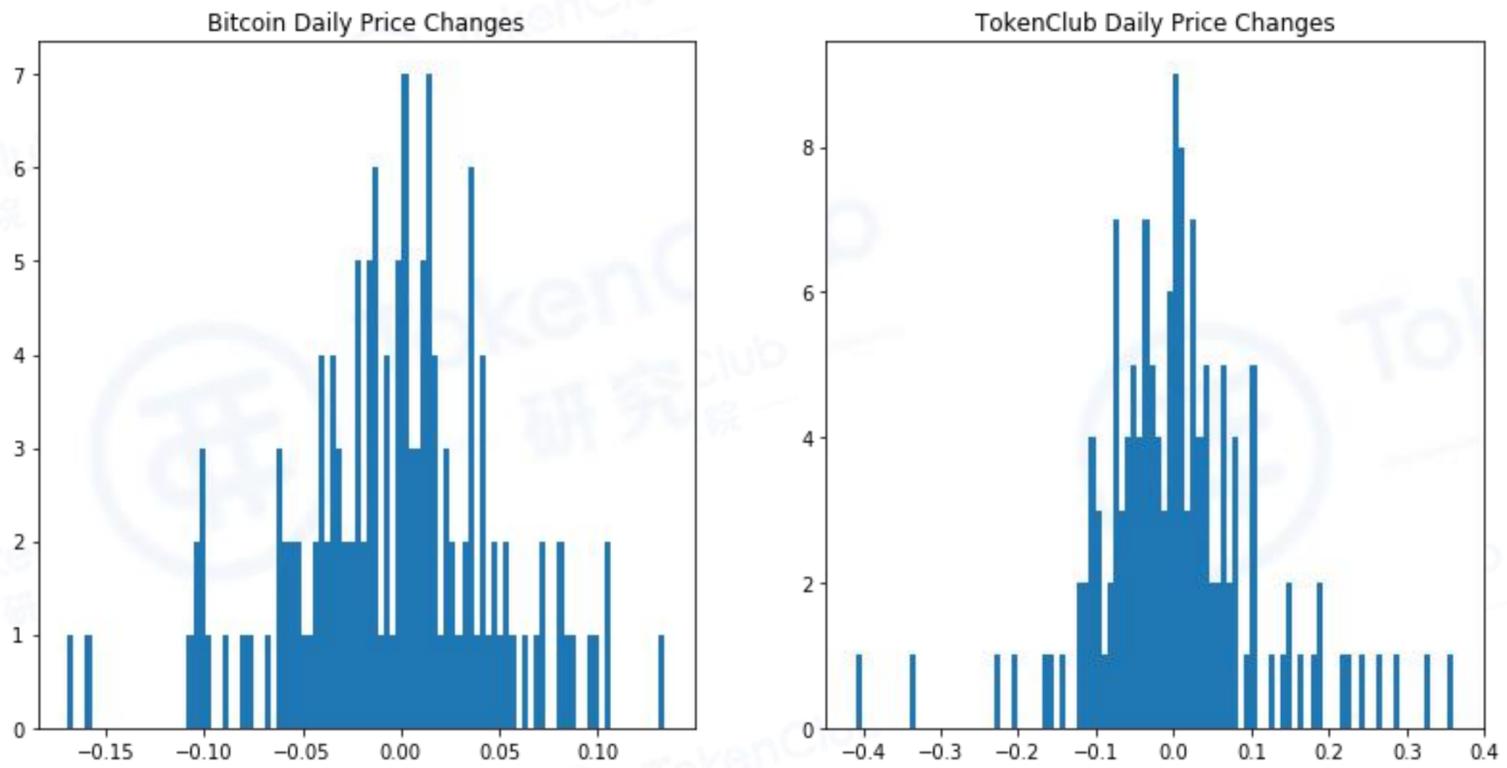
比特币、TCT 滞后模型收盘价走势图

扩展这种琐碎的滞后模型，数字货币的价格通常被视为随机游走，可以用以下数学术语来定义：

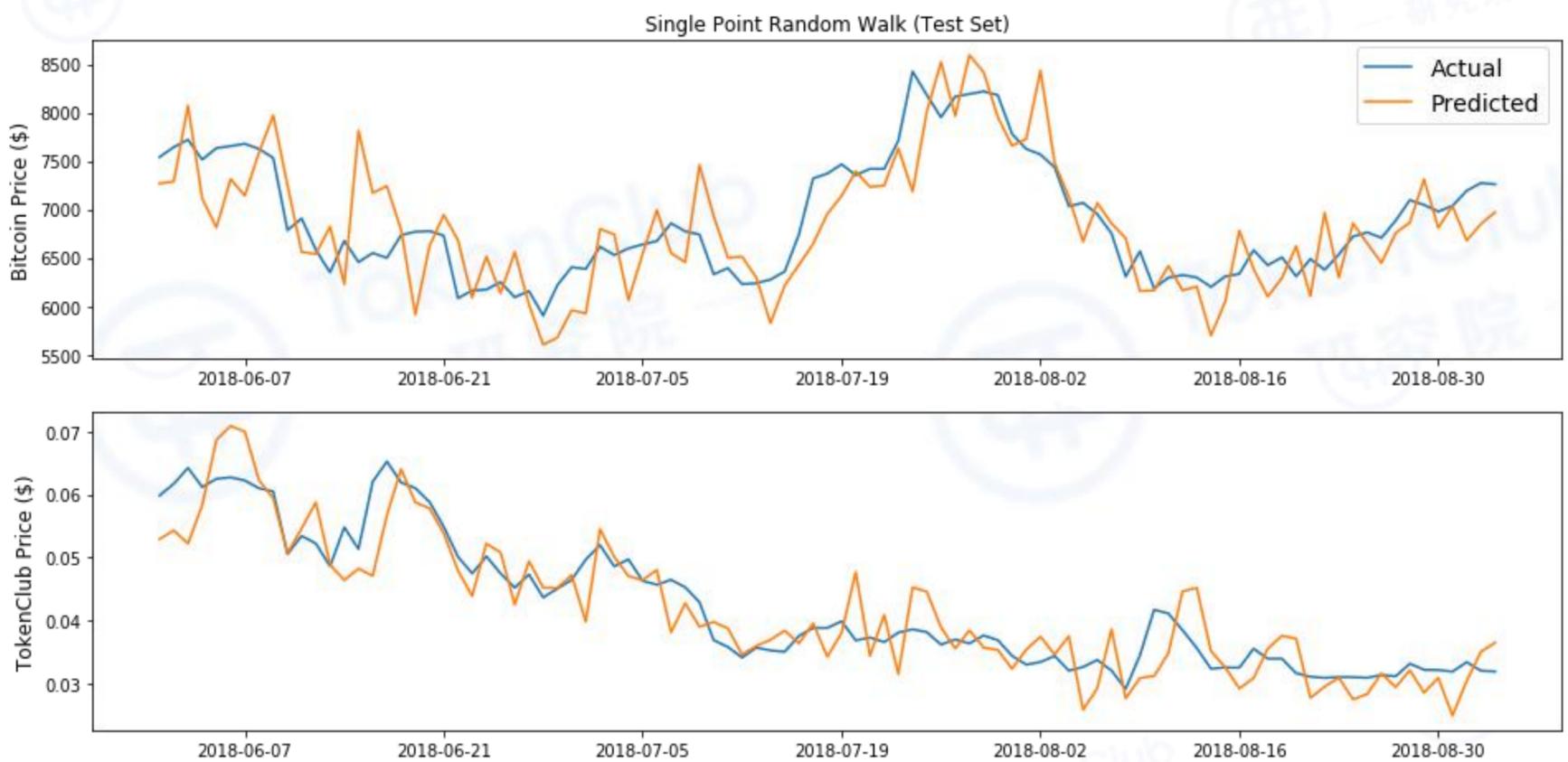
$$PredPrice_t = ActualPrice_{t-1} * \epsilon, \epsilon \sim N(\mu, \sigma)$$

从训练集中确定 $\mu$ 和 $\sigma$ ，将随机游走模型应用在比特币和 TCT 的测试集上。

首先,我们可能希望确保价格的每日变化遵循正态分布。我们将绘制值的直方图。



比特币、TCT 收盘价正太分布直方图

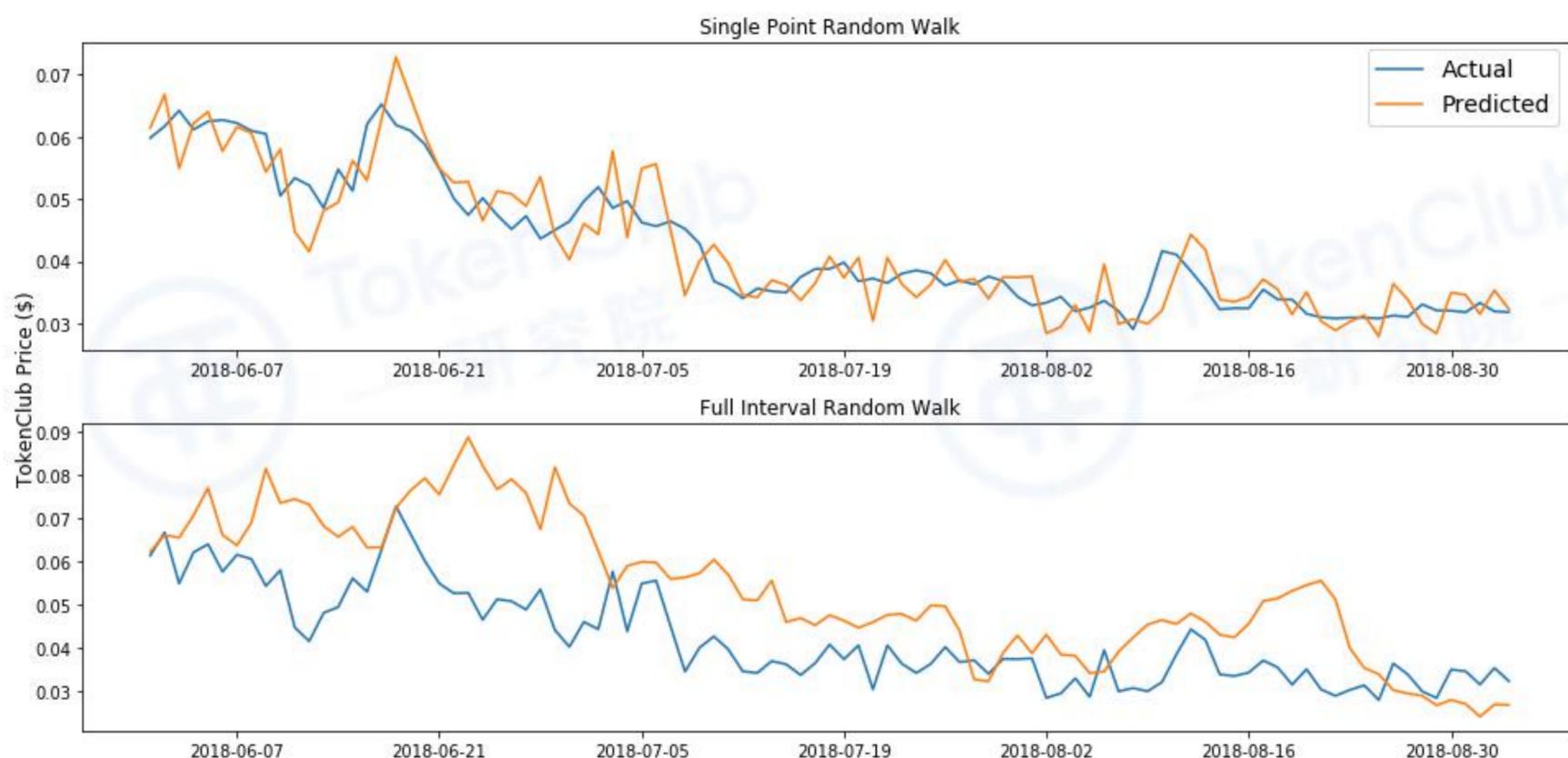


比特币、TCT 单点随机游走收盘价走势图

从上面的预测线可以看出：除了少许部分扭曲，基本上紧贴每个虚拟货币的实际价格变动情况。模型甚至捕捉到了6月中旬和8月下旬的涨跌情况。然而正如另一篇预测比特币价格的博客中所说，只在某个点预测未来的模型常常会让人误以为它很准确，因为在接下来的预测中模型没有将误差因素考虑在内。不管误差大小如何，它在每个时间点本质上会被重置，因为输入模型的是真实价格。比特币的随机游走尤其具有误导性，因为y轴的值可以很大，这让预测线看起来很平滑。不幸的是，单点预测在评估时序模型中相当普遍。所以通过多点预测衡量模型准确度是个更好的做法。这样以来，之前预测的误差不会被重置，而是会被计算入接下来的预测中。这样，我们在数学上这样定义：

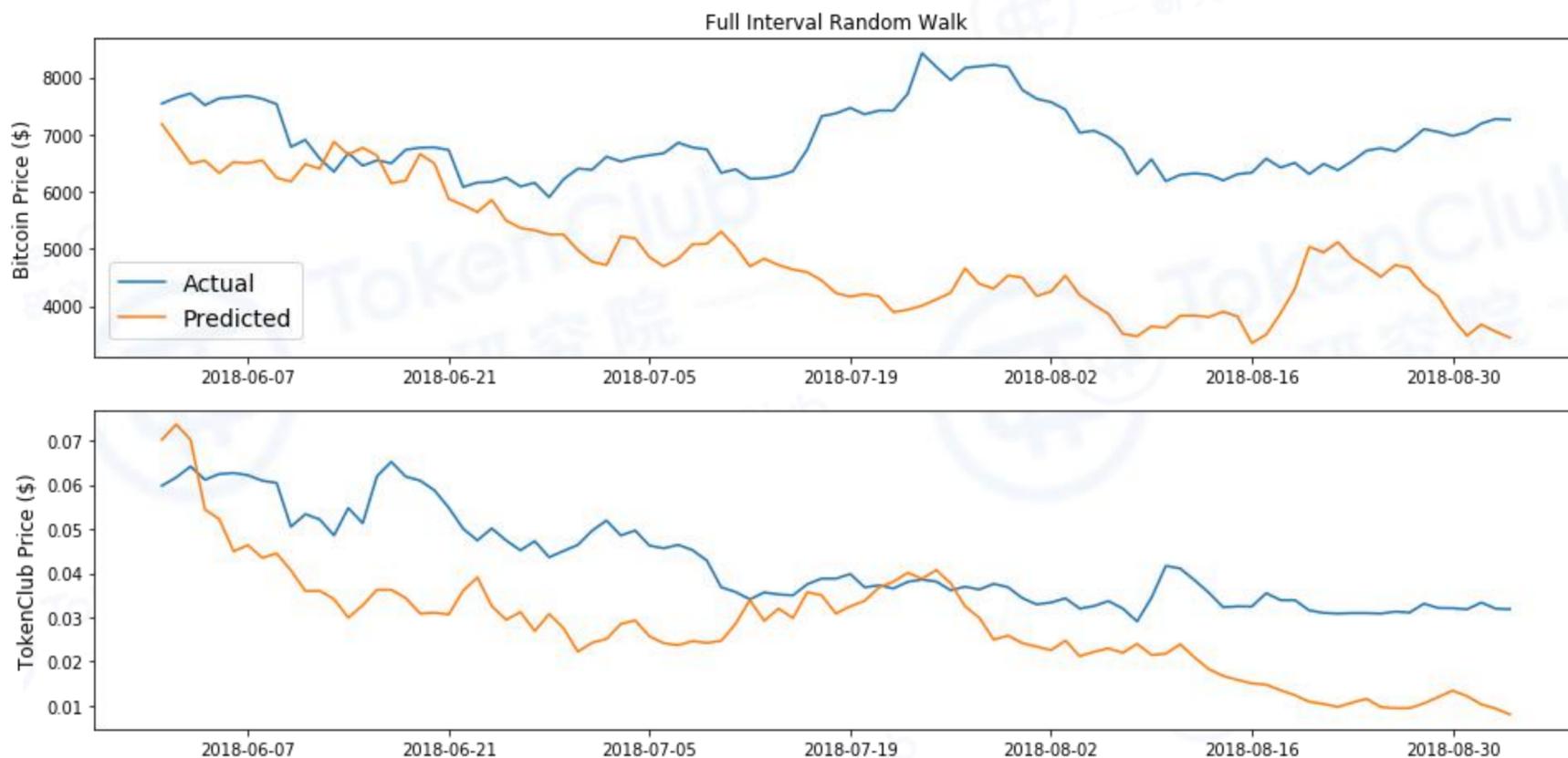
$$PredPrice_t = PredPrice_{t-1} * \epsilon, \epsilon \sim N(\mu, \sigma) \& PredPrice_0 = Price_0$$

来试试多点预测随机游走模型，预测整个测试集的收盘价格。



比特币、TCT 多点随机游走收盘价走势图

模型预测对随机种子非常敏感，上图是预测多次之后看起来比较好的预测图。



用单点和多点随机游走模型预测 TCT 收盘价走势图

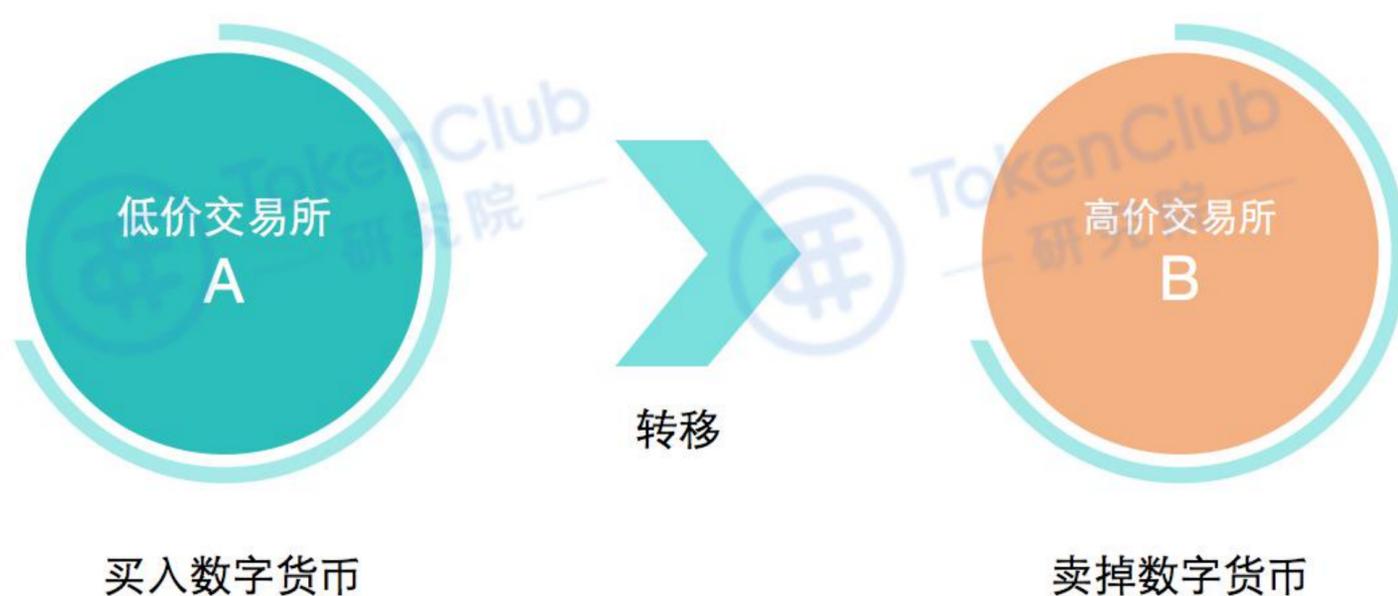
### 3.5.3 结果分析比较

经过简单一轮测试以及分析，可以很明显地看到单点随机游走模型更稳定一下。但是单点分析也没有实际性能让人信服的东西。机器学习还有很多其他模型可以预测价格，比如 TSTM 及其扩展模型。

机器学习预测价格还需要更多的数据，像基本面数据、搜索指数、新闻等数据再加上优秀的模型才能更可能地准确。

## 4. 数字货币量化套利场景

### 4.1 搬砖套利



搬砖套利就是同一数字货币在不同市场有价差时，在价低的平台买入数字货币，再到价高的平台卖出数字货币；那么出现套利机会的时候，数字货币量化交易系统会在毫秒级内在完成套利操作，实现搬砖套利，从而赚取低风险差价的一种套利交易行为。

#### 4.1.1 跨交易所两角硬搬套利

所谓的跨交易所两角硬搬套利就说是直接从低价交易所买入，然后在高价交易所卖出的过程。

假设交易所 OKEX、Gate 交易对 TCT/USDT 价格分别为 10、5，存在价差有套利机会。遵循高卖低买原则，首先假设用户在 OKEX 存有 10 个 TCT，在 OKEX 上卖出 10 个 TCT 得到 100 个 USDT，将 USDT 提币到 Gate，在 Gate 全部买入 TCT，则得到 20 个 TCT。完成一次套利增加了 10 个 TCT，USDT 没有变化，如果价差一直存在可以循环继续套利。假如用户在两个交易所没有余额，则可以先在 OKEX 存入人民币进行上述交易。为了方便计算没有使用实际价格以及不考虑交易手续费、提币等成本。具体流程图如下：

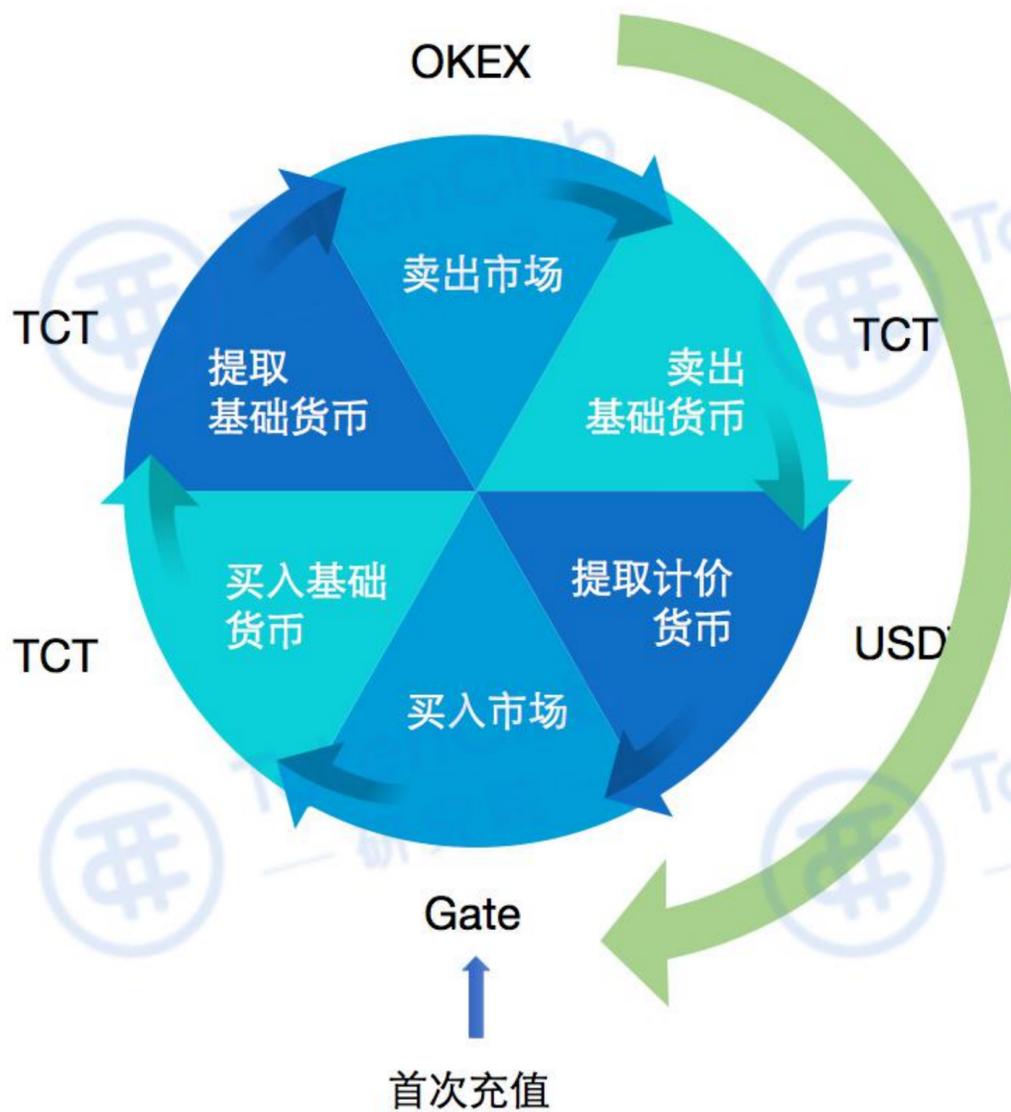


### 4.1.2 跨交易所两角对冲套利

跨交易所两角对冲套利又叫直接套利或双边套利，跟上一条跨交易所两角硬搬套利差不多，也是通过发现同一交易对（如:TCT/USDT）在两个不同交易所间存在价差关系，进行高卖低买从中套取差价利润的行为；但是中间使用了对冲规避风

险，对冲盈亏相抵数量相当，同时在两边市场买入和卖出相同的基础货币，确保转入计价币种，赚取的也是计价币种。

假设交易所 OKEX、Gate 交易对 TCT/USDT 价格分别为 2、1，存在价差有套利机会，在 OKEX TCT 余额 5 个，在 Gate USDT 10 个。引入对冲思想，每次对冲量 1 个 TCT，遵循高卖低买原则。为了方便计算没有使用实际价格以及不考虑交易手续费、提币等成本。套利过程如下：



- 黑色箭头：当高价市场持有交易币或低价市场持有计价币、法币时，可以进行双向循环对冲套利。
- 蓝色箭头：当在高价市场无基础货币 TCT 或低价市场无计价货币 USDT 时，从

事套利需先在低价市场进行法币充值。

➤绿色箭头：当循环套利收益大于套利成本，且存在法币账户路径下，可以进行循环套利。

流程说明：

阶段 对冲量 (1TCT)	买入市场 (Gate)				卖出市场 (OKEX)				总资产	
	买入 价格	买入 数量	剩余 USDT	剩余 TCT	卖出 价格	卖出 数量	剩余 USDT	剩余 TCT	USDT	TCT
准备对冲	0	0	10	0	0	0	0	5	10	5
第1次	1	1	9	1	2	1	2	4	11	5
第2次	1	1	8	2	2	1	4	3	12	5
第3次	1	1	7	3	2	1	6	2	13	5
第4次	1	1	6	4	2	1	8	1	14	5
第5次	1	1	5	5	2	1	10	0	15	5

OKEX的TCT为0不具备对冲条件，进行提币，将Gate的5个USDT提至OKEX 并买入5个TCT继续下一轮对冲操作。

第 1 轮对冲:①在 Gate 买入个 TCT 花费 1 个 USDT 则在 OKEX 账户变为持有 1 个 TCT、9 个 USDT ; ②在 OKEX 卖出 1 个 TCT 得到 2 USDT ,则在 OKEX 账户变为持有 4 个 TCT ,持有 1 个 USDT。

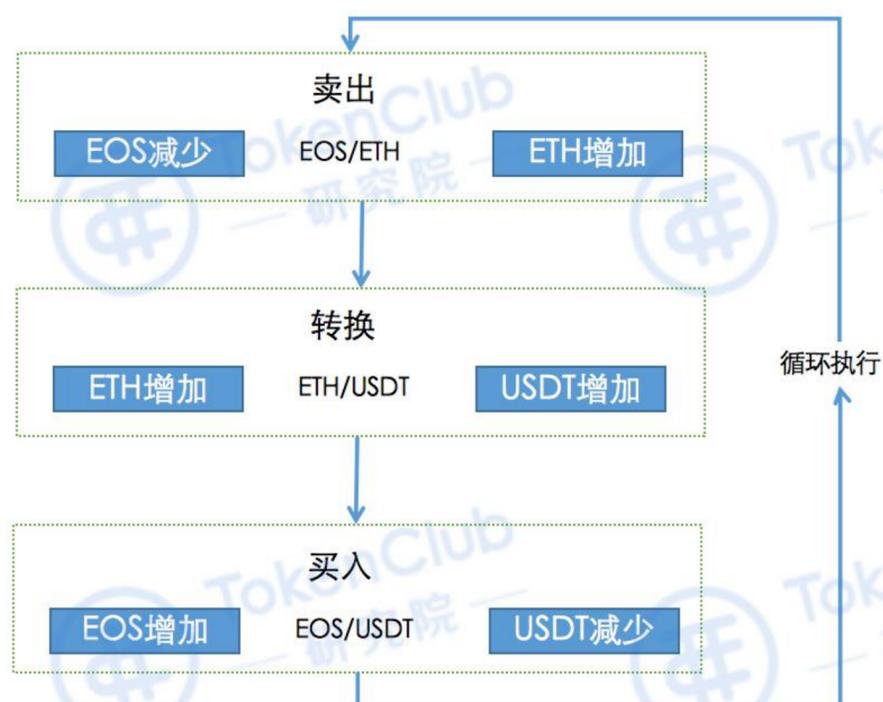
经过 1 轮 ( ①、②不分先后,可同时进行) 对冲交易 ( 对冲量 1 个 TCT ) , 该用户 TCT 仍为 5 个 , USDT 数量变为 11 个 , 净赚 1 个 USDT。

### 4.1.3 交易所站内三角循环套利

三角套利是利用数字货币交叉汇率定价进行的套利。在数字货币市场上，几乎所有的数字货币兑比特币或 ETH 都有一个兑换率。不同交易所对比特币或 ETH 汇率报价不同于其他交易所报出的汇率。如果差异大到足以与购买及出售数字货币货币的交易成本相比，就出现了无风险套利机会。

交易所站内三角循环套利基于三角套利原理，同一交易所三个交易对形成的差价进行交易，不需要提币动作可无限循环进行。以 1 万块投入来算，即使一次套利收益仅 0.1%，如果一天执行 1 千次套利，则一天收益变成 2.17 万年化收益，永动不需要提币充币。如  $\text{EOS/USDT}=10$ 、 $\text{EOS/ETH}=0.5$ 、 $\text{ETH/USDT}=100$ ，则通过  $\text{EOS/USDT}$ 、 $\text{EOS/ETH}$  交叉定价形成  $\text{ETH/USDT}$  的价格为 5，与  $\text{ETH/USDT}$  的当前价 100 存在价格差，遵循高卖低买原则，可进行套利。

假设初始持有 2 个 EOS，花费 2 个 EOS 通过卖出  $\text{EOS/ETH}$  得到 1 个 ETH，再将 1 个 ETH 通过买入  $\text{USDT/ETH}$  得到 100 个 USDT，接着再将 100 个 USDT 通过买入  $\text{EOS/USDT}$  得到 10 个 EOS，循环一圈完成三角套利 EOS 从初始的 2 个变为 10 个。



## 4.1.4 跨交易所三角套利

跨交易所三角套利又叫间接套利或多边套利，是同一时间在三个或三个以上的交易市场上利用三种或三种以上的货币汇率差价进行交易，三角套利是利用交叉汇率定价错误进行的套利。相比站内三角套利跨市三角套利收益更高机会更多。

例如：

OKEX : EOS/USDT = 5 \$

Gate: EOS/ETH = 0.02 \$

Huobi: ETH/USDT = 275\$

则通过 EOS/USDT、EOS/ETH 交叉定价形成 ETH/USDT 价格为 250 \$( A / B = ETH/USDT = 250 \$ ) ，与交易所 C 的 ETH/USDT = 275\$存在价格差，可进行套利。

Gate 交易所卖出 EOS 100 个，获得 ETH 2 个；C 交易所卖出 ETH 2 个，获得 USDT = 550 \$；OKEX 交易所买入 550 \$ EOS ，即 110 个 EOS ，套利利润为 10 个 EOS。



跨市三角套利，必须三个交易所都存有足量的 USDT、EOS、ETH，并且每次操作之后，需要把消耗的币种互相转移补充，保持一个平衡。

跨市三角套利由于交易所众多，机会的出现概率也会相应增大，但仍然要注意平台风险，尽量在大交易所之间操作。当然套利还需要考虑来回手续费以及币价涨跌的各种损失。

### 4.1.5 跨国套利

跨国套利原理跟上面套利原理差不多。一般来说，交易所都是以本国货币来计价的，一旦一个国家因为某些原因造成价格波动，那其他币种会因为差价来套利，但是货币之间在国际上有管制，再加上比特币的交易也不是瞬间完成的，所以会有个差价的时间段，但是随着时间这个差价会逐渐减少。

比如韩国人莫名地突然大量买比特币，但韩国人只持有韩元，所以短时间会造成韩国比特币交易所价格升高。那中国人看到利益，会花人民币低价从国内交易所购得比特币，在韩国的交易所卖掉，得到大量韩元再由韩元换成人民币，这样能挣不少人民币。

现实例子：据 Coinhills 数据显示，2018 年 9 月 5 日 04:53，BTC 报价 9.78 亿伊朗里亚尔，约合 2.32 万美元，创历史新高。当前，BTC 全球均价为 7361 美元，伊朗 BTC 溢价高达 1.58 万美元。此前伊朗央行官员暗示加密货币禁令将于 9 月结束，伊朗央行附属机构表示，伊朗未来的加密货币将由法币里亚尔支持。

## 4.1.6 打新套利

打新套利跟传统金融市场差不多；当数字货币在交易所新上市，那么对数字货币的需求通常会上升。那么这里就有个套利机会了。例如，当一种新数字货币在像 OKEX 这样的交易所上市时，它会经历价值的增长，并且很多人都想购买它；从其他已上市的交易所购买并以较高的价格在 OKEX 上出售，这就是数字货币打新套利。

## 4.2 期现套利

说期现套利之前，我们先从期货跟现货的优势体现在哪里，然后再具体谈谈怎么套利。

### 4.2.1 期货的优势

#### 4.2.1.1 回测期货可以双向交易

现货只能做多，币价涨了才能赚钱。而数字货币一直在波动，在下跌的时候投资者错过了赚钱的机会，不平仓还会带来巨大的亏损。

### 4.2.1.2 期货带有杠杆

在现货市场中，比特币涨1美金就赚1美金，而期货市场中，利用杠杆比特币涨1美金可以赚5美金甚至更多，可实现资金最大化利用。

### 4.2.1.3 可对冲套利

若数字货币市场上期货价格比现货价格高，则可以买入现货，同时做空等价值的期货，等待估值回归或交割，可以赚到其中的差价。

### 4.2.1.4 对冲风险

对宽客而言，利用数字货币期货可以对冲风险。在卖出现货的时候，可买入期货进行对冲，在买回现货后再平掉期货，实现风险对冲，减少因为价格波动带来的风险，省去了资金来回调动的的时间。

## 4.2.2 期现套利

期现套利指的是，利用期货现货存在差价进行套利。理论上期货与现货应该存在一种固有等评价关系，但实际上，期货价格常受多种因素影响而偏离其合理的理论价格。由于交割日时比特币期货与比特币现货最后完全一致，当比特币期货和比特币现货一旦出现不合理的差价时，就可以通过做空期货，同时做多现货来获得无风险的差价收益。

案例：假设 OKEX 的比特币期货那天最高价格是 12000 美金，而 OKEX 的比特币现货价格是 10000 美金

这时候可以在 OKEX 买入 1 个比特币现货，花费 10000 美金，同时时间在 OKEX 做空当周比特币期货，做空价 12000 美金的

#### 盈亏计算方法：

1、假如当周期货交割时 OKEX 比特币币价涨到 15000 美金。

期货亏损： $(15000-12000)/15000=0.2$  BTC，剩余 0.8 BTC，此时币价涨到 15000 美金，在现货能卖  $0.8*15000=12000$  美金。实际收益  $12000-10000=2000$  美金。

2、假如当周期货交割时 OKEX 比特币币价跌到 9000 美金。

期货收益： $(12000-9000)/9000 \approx 0.33$  BTC，剩余 1.33BTC，此时币价跌到 9000 美金，在现货能卖  $(1+(12000-9000)/9000)*9000=12000$  美金。实际收益  $12000-10000=2000$  美金。

通过上面的案例可以看到，只要有差价存在，下单后就可以赚钱。

特别说明：全程为了方便计算没有计算建仓成本等，没有使用杠杆，实际在 OKEX 的期货合约收益计算以张数计算，而一张合约所锚定的，并不是一定数量的数字

货币，而是价值 10 美元（btc 是 100 美元）的某种数字货币，即合约的面值为 10 美元（btc 合约为 100 美元）。通过这样的设计，可以达到合约持仓的杠杆率始终不变。

### 4.2.3 跨期套利

跨期套利是利用同一交易所同时买入、卖出数字货币的不同交割月份的期货合约，在未来的有利时机将这些不同交割月份的比特币合约对冲平仓获利。主要有如下两种方式：

#### 4.2.3.1 牛市套利

牛市套利（买近卖远套利、多头套利）：从价差的角度看，做牛市套利的投资者看多币市，认为较远交割期的数字货币期货涨幅将大于近期数字货币期货的涨幅，或者说较远期的数字货币期货跌幅将小于数字货币期货的跌幅；从价值判断的角度看，即是认为远期的数字货币期货的价格应高于当前远期的数字货币期货的交易价格，当前远期的数字货币期货的价格被低估。我们都可以买入较近的数字货币期货同时卖出较远的数字货币期货，这样盈利的可能性比较大。

#### 4.2.3.2 熊市套利

熊市套利（卖近买远套利、空头套利）：从价差的角度看，做熊市套利的投资者看空币市，认为较远交割期的数字货币期货跌幅将大于近期数字货币期货的跌

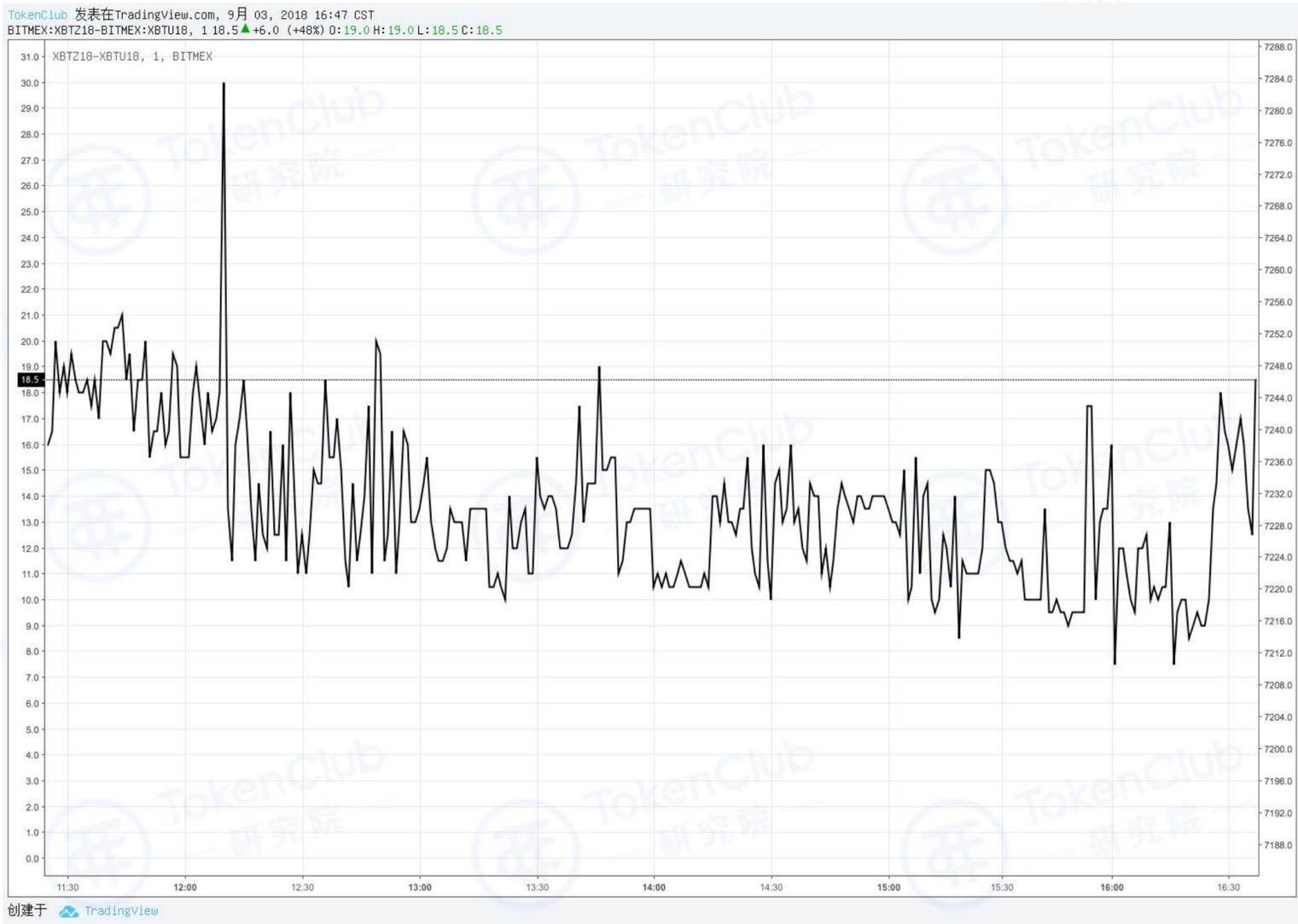
幅，或者说较远期的数字货币期货涨幅将小于数字货币期货的涨幅；从价值判断的角度看，即是认为远期的数字货币期货的价格应低于当前远期的数字货币期货的交易价格，当前远期的数字货币期货的价格被高估。我们都可以卖出较近的数字货币期货同时买入较远的数字货币期货，这样盈利的可能性比较大。

但无论采取哪种操作模式，其本质均是对不同交割期的合约同时进入低买高卖，即同时买入价值被低估的合约而卖出价值被高估的合约。

#### 4.2.3.3 BTC 期货分析

跨期套利需要达到的目标是，无论价格怎么波动，我的收益只和价差的变化有关，如果做空价差，那么价差缩小即盈利，价差扩大即亏损，和价格没有关系。但是，上文所述的开仓方法是错误的。因为它还和一个价格有关系，那就是现货价格。下面我们以一个当周及季度期货举例说明：

取 BitMEX 的 XBTU18（比特币合约 2018 年 9 月 28 日到期）以及 XBTZ18（比特币合约 2018 年 12 月 28 日到期）一分钟数据，价差图（远期-近期）、收盘价走势图如下：



价差图



收盘价走势图

收盘价走势图中红色线为 XBTZ18 合约，黑色线为 XBTU18 合约。可以很明显的看到，在不同期数中价格在变动时会出现明显的价差。

与期现套利的理论上无风险套利不同，在进行跨期套利时，需要对未来价格走势进行预判，如果判断错误且行情不反转，价差可能一直维持在不利的位位置，难以盈利平仓。

## 4.2.4 统计套利

统计套利的基本思路是运用统计分析工具对一组相关联的价格之间的关系的历史数据进行研究分析，研究该关系在历史上的稳定性，并估计其概率分布，确定该分布中的极端区域，即否定域，当真实市场上的价格关系进入否定域时，则认为该种价格关系不可长久维持，套利者有较高成功概率进场套利。

统计套利 交易策略	第一类：居于均值回归的配对交易策略
	第二类：对因子模型判断资产价值高/低估的交易策略
	第三类：指数跟踪类的投资组合建立交易策略

## 4.2.5 跨品种套利

跨品种套利：指交易者在两个价格联动性较强的不同指数期货合约上开立相反头寸的交易方式，这两个指数期货可以是在同一个交易所交易的，也可以是在不同交易所交易的。

## 4.2.6 跨市套利

跨市场套利指在不同的数字货币交易所同时买进和卖出相同交割期数的同种或类似的数字货币期货合约，以赚取价差利润的套利方式，又称市场间价差。

投机者利用同一商品在不同数字货币交易所的期货价格的不同，在两个交易所同时买进和卖出期货合约以谋取利润的活动。

## 5. 量化开发语言与开发库

要进行量化交易，必然要通过写代码来实现，对于初学者碰到的第一个问题就是工具的选择。首先大部分交易员本来不会写程序，选择任何一个语言进行策略开发，都有不小的学习成本。更重要的是，选择了一门语言，接下来开发环境、人员招聘、数据接口与平台、甚至同类人群之间的交流、遇到问题后的支持，都跟着被“套牢”。接下来列举一下常用多量化交易编程语言，以及相关的开发库。

### 5.1 Python

Python 从一开始就是开源的，有各种第三方的库可以现成使用。这些底层功能库让程序员省去了“造轮子”的时间，让我们可以集中精力做真正的策略开发工作。

Python 是使用人群最多、社区最活跃的语言之一，也是最受宽客欢迎的语言之一。Python 也是机器学习领域被使用最多的语言之一。像 tensorflow、scikit-learn、Theano 等等对 python 都有极好的支持

下面是几个量化中常用的 python 库：

#### 5.1.1 NumPy

科学计算包，向量和矩阵处理超级方便。

NumPy 是 Python 很多科学计算与工程库的基础库，在量化数据分析中最常使

用的 pandas 也是基于 NumPy 封装的。可以说 NumPy 就是量化数据分析领域中的基础数组，学会使用 NumPy 是量化分析的关键一步。

NumPy 底层实现中使用了 C 语言和 Fortran 语言的机制分配内存。可以理解为它的输出是一个非常大且连续的并由同类型数据组成的内存区域，所以你可以构造一个比普通列表大得多的数组，并且灵活高效地对数组中所有元素进行并行化操作。

## 5.1.2 Pandas

Pandas 天生为处理金融数据而开发的库。几乎所有的主流数据接口都支持 Pandas。Python 量化必备神器。

pandas 在 Python 金融数据分析、量化交易等领域起到了至关重要的作用，pandas 的两大主要工具 DataFrame 和 Series 的框架设计与 API 参考方向，都是源于统计分析语言 R。

## 5.1.3 SciPy

开源算法和数学工具包，与 Matlab 和 Scilab 等类似方便、易于使用、专为科学和工程设计的 Python 工具包。它包括统计、优化、整合、线性代数模块、傅里叶变换、信号和图像处理、常微分方程求解器等等。基本可以代替 Matlab，但是使用的话和数据处理的关系不大，数学系，或者工程系相对用的多一些。

近期发现有个 statsmodel 可以补充 scipy.stats，时间序列支持完美。

## 5.1.4 Matplotlib

Python 中最著名的绘图系统，很多其他的绘图例如 seaborn（针对 pandas 绘图而来）也是由其封装而成。

绘制的图形可以大致按照 ggplot 的颜色显示，但是还是感觉很鸡肋。但是 matplotlib 的复杂给其带来了很强的定制性。其具有面向对象的方式及 Pyplot 的经典高层封装。

需要掌握的是：



散点图，折线图，条形图，直方图，饼状图，箱形图的绘制。



绘图的三大系统：pyplot，pylab(不推荐)，面向对象



坐标轴的调整，添加文字注释，区域填充，及特殊图形 patches 的使用

## 5.2 R 语言

通常编程语言的设计，都是为了解决软件开发和程序实现的问题。但 R 语言在开始时，就被设计为主要解决数据的问题。

量化投资就是对数据进行各种数据处理、数据分析，从而找到数据的规律。所以，有很多从事量化投资的人，把 R 语言用来构建量化交易的模型，进行回测，风险管理等，最后把研究成果开源并贡献给 R 语言的社区，为后面的人提供了非常大的帮助。R 语言在量化投资领域，已经有很多年的积累，很多的算法已经成型。从投资研究到交易分析，再到风险管理，有着完整的体系结构。

R 与跟 Python 相比：本质上 Python 是面向程序设计的语言，而 R 是面向数据的语言。

由于 R 语言的开发库众多，下面按功能来列举 R 语言开发库：

## 5.2.1 数据管理

包括数据集抓取、存储、读取、时间序列、数据处理等，涉及 R 包：

zoo(时间序列对象)	Quantmod (数据下载和图形可视化)	timeSeries(Rmetrics 系 时间序列对象)
WindR(Wind 数据接口)	RQuantLib (QuantLib 数据接口)	timeDate(Rmetrics 系时 间序列处理)
rredis(Redis 访问接口)	xts(时间序列处理)	Rmongodb (MongoDB 访问接口)
rhive(Hive 访问接口)	RJDBC(数据库访问接口)	SparkR (Spark 访问接口)
data.table(数据处理)	Rhadoop (Hadoop 访问接口)	flImport(Rmetrics 系数据 访问接口)

## 5.2.2 指标计算

包括金融市场的技术指标的各种计算方法，涉及 R 包：

TTR(技术指标)

TSA(时间序列计算)

urca(单位根检验)

fRegression  
(Rmetrics 系回归分析)

fUnitRoots  
(Rmetrics 系单位根检验)

fGarch

(Rmetrics 系 Garch 模型)

fNonlinear

(Rmetrics 系非线性模型)

fOptions

(Rmetrics 系期权定价)

fCopulae

(Rmetrics 系财务分析)

FinancialInstrument  
(金融产品)

Quantstrat

(策略模型和回测)

Blotter

(账户管理)

fTrading

(Rmetrics 系交易分析)

## 5.2.3 回测交易

包括金融数据建模，并验证用历史数据验证模型的可靠性，涉及 R 包。

FinancialInstrument  
(金融产品)

Quantstrat  
(策略模型和回测)

Blotter  
(账户管理)

fTrading  
(Rmetrics 系交易分析)

## 5.2.4 投资组合

对多策略或多模型进行管理和优化，涉及 R 包：

PortfolioAnalytics  
(组合分析和优化)

stockPortfolio  
(股票组合管理)

fAssets  
(Rmetrics 系组合管理)

## 5.2.5 风险管理

对持仓进行风险指标的计算和风险提示，涉及 R 包：

PerformanceAnalytics  
(风险分析)

fPortfolio  
(Rmetrics 系组合优化)

fExtremes  
(Rmetrics 系数据处理)

## 5.3 其他编程语言

C、C++的特点是速度最快，但要想用好，必须对计算机底层架构、编译器等等有较好的理解，这是非计算机专业的人很难做到的，对于做量化交易来说更是没有必要。C/C++没有完备的第三方库。如果为了做大量的计算，要自己实现、维护、优化相应的底层算法。但是由于 C/C++语言速度优势，通常被用来开发高频交易系统。

## 6. 数字货币量化投资未来展望

随着数字货币的发展的浪潮，未来数字货币市场的量化交易必定会越来越成熟稳定，我们通过国内外量化现状看数字货币投资未来发展趋势，以及对量化投资市场的些许期待。

### 6.1 国内外量化投资现状

传统金融市场的量化投资在海外发展有 30 多年发展历程，其投资业绩稳定，市场规模和份额不断扩大，得到越来越多投资人的认可。事实上，传统金融市场的量化投资在国内并不新鲜，但真正的量化基金在国内的发展还处于初级阶段。

相比较传统金融市场的量化交易，国内数字货币量化交易一点不比国外差，甚至国内可能还更强一点。因为最大的交易所是中国的，最大的矿机也是中国的，炒币最多的也是中国的。目前国外数字货币量化领域鲜有知名人物，但在传统量化领域我们知道有詹姆斯·西蒙斯、雷煞·达里奥等人，比国内强很多。

## 6.2 数字货币量化投资发展趋势

随着数字货币量化投资的快速发展，投资领域正在出现以下新趋势：

- 1. 算法模型同质化。量化交易的算法模型虽各具特色，但同一理念的模型不断改进最终将趋同。
- 2. 软件竞争硬件化。当算法模型趋于极限，包括超级计算机、超级芯片在内各种 IT 设备的硬件性能、质量、机房、服务器托管、与交易中心的点距离等都可能成为秒杀的关键因素，这同时也给高频交易的监管提出了新挑战。
- 3. 竞争对手同类化。向量化交易转型的高技术门槛必然导致投资者结构的调整，市场将从主观与量化交易之间的博弈转向各类量化交易之间的博弈。
- 4. 投资范围全域化。单一品种、单一资产、单一市场无法实现量化交易优势最大化，概率论青睐全品种、全时空、多市场、高频率的投资组合。
- 5. 交易人才 IT 化。老牌金融中心伦敦正在演变为 IT 公司集散地，包括高盛、大小摩根、花旗、瑞信等在内众多欧美顶尖投行都在不计血本打造 IT 团队，专攻投资模型研发。
- 6. 量化团队集约化。2017 年 8 月有消息称，大小摩根和高盛筹划共建大数据公司，合力提供证券产品参考数据(Securities Product Reference Data)。
- 7. 博弈手段对抗化。日趋激烈的竞争派生对手之间的干扰、幌骗甚至黑客攻击。

## 6.3 对数字货币量化投资领域的期待

A

整个市场投资者会更加理性，不要再指望着像去年那样动不动百倍千倍的收益，未来投资人越来越理性之后可以接受一年百分之三十到五十的投资回报，这个可能是个比较大的利好。

B

未来趋势就是大量专业机构投资人和专业做金融的人士会进入量化投资领域，带来更加专业的东西，像以前那样找个名人站台就能割韭菜的情况，我相信会越来越减少。未来的币圈，不管是做产品、做项目还是发基金，越来越需要较正规的专业性人士参与。

## 风险提示

本报告所采用的数据均来自合规渠道，分析逻辑基于作者的职业理解，本报告清晰准确地反映了作者的研究观点，力求独立、客观和公正，结论不受任何第三方的授意或影响，特此声明。

本报告的信息来源于已公开的资料，TokenClub 研究院对该等信息的准确性、完整性或可靠性不作任何保证。在任何情况下，本报告中的信息或所表述的意见均不构成对任何人的投资建议。

本报告版权仅为 TokenClub 研究院所有，未经书面许可，任何机构和个人不得以任何形式翻版、复制、发表或引用。如征得 TokenClub 研究院同意进行引用、刊发的，需在允许的范围内使用，并注明出处为“TokenClub 研究院”，且不得对本报告进行任何有悖原意的引用、删节和修改。



TokenClub 是国内领先的数字货币投资社区，致力于构建一个自治、信任、高效的数字资产投资服务生态。

“TokenClub 研究院”是 TokenClub 旗下研究区块链的专业机构，专注于区块链行业研究、项目评级。



扫码关注  
TokenClub 研究院



扫码下载  
TokenClub APP