配对交易——相对价值套利的算法实现与实验分析

陈炜艺 2010011352

清华大学 交叉信息研究院 指导老师：唐平中

1. 前期工作与目前结果

我在上班学期的主要工作是进行策略后台 Python 库的搭建和配对交易算法的学习。配对交易属于计算金融领域，介于计算机和金融之间，基于计算机模型和语言进行金融分析和预测。而我的主要研究内容是采用 Machine Learning 的方法进行股票预测，但为了进行实验需要建立好后台包括交易策略的研究与模拟测试等。上半学期的八周期间我都给自己作了计划并顺利完成目标，下面分八块介绍我的前期工作和每周结果。我担心如果不是兼具金融和计算机知识的人，看我的实习内容可能感觉晦涩难懂，所以我把通俗易懂的结果写在具体工作内容之前供老师审核。

1.1. 第一周：建立 QSTK 开发环境与熟悉 QSTK

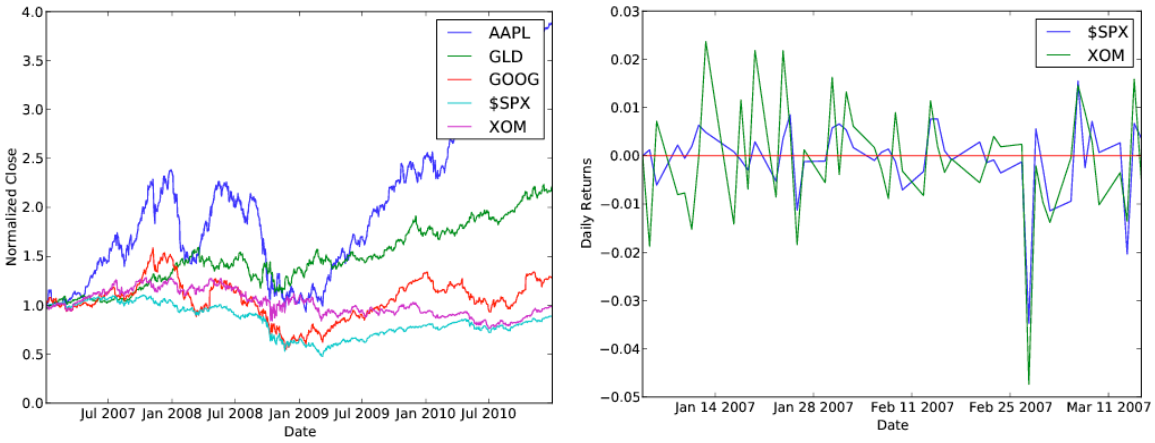
Quantitative Software Tool Kit 是我们实验室基于 Python 的开源构架，设计目的是为了支持投资组合的建立和管理。实验室初步建立开发 QSTK 主要是供金融、计算机的学生和有编程经验的数量分析员使用。它目前还不是一个桌面开发平台，仅仅只是是应用的基础架构，支持模型、测试和交易的工作流程。我的第一个任务就是安装这套开源架构，熟悉其中的代码，为后面基于这份架构的开发和实验做准备。

1.2. 第二周：投资组合的评估与优化的 Python 库建立

目前结果：初步了解股票历史数据的获得方法，以及如何使用 Python 和它的 Numpy 库进行股票组合的优化。这是导师给我布置的任务，实际上这并不是现实中迅速建立一个强大投资组合的方法，只是导师希望借此能让我对股票资产有些自己的想法，自主创造和优化一个局限在 2011 年和 4 支股票的短时间小型投资组合。

工作内容：

1、用 QSTK 代码进行股票价格数据的时间序列分析。我们使用的编程语言是 Python，选择原因是不象 C 那么复杂，几行代码就能实现十分强大的功能，并且本身有很多库支持数据的操作和展示。代码将作为附件一同提交。在报告中展现分析代码过于冗余，此处用两幅图说明工作内容。



以上都是用 Python 的 Plot 库实现的图案绘制，左图是各股票基于首日价格的价格走线图，右图是日回报图。

2、Python 函数模拟包含 4 支股票的投资组合过程和评估该投资组合的表现。函数的输入参数是开始时间、结束时间、股票代码（如 GOOG、AAPL、GLD、XOM）和在模拟投资开始时在各股票上的资产分配比例（如 0.2、0.3、0.4、0.1），函数的输出是投资组合日回报的标准差、平均日回报、夏普比例和总回报。

举例说明，调用函数的输入代码为
vol, daily_ret, sharpe, cum_ret = simulate(startdate, enddate, ['GOOG','AAPL','GLD','XOM'], [0.2,0.3,0.4,0.1])

3、用上述的 similate()函数做一个投资组合优化器。换言之，用 for 循环将所有在 4 支股票上可能的资产分配集合元素循环一遍，纪录最优投资组合，并输出。可能存在的配比模式即配比之和为 1，为简化实验，分度值仅设为 10%，如［1.0，0.0，0.0，0.0］,［0.1，0.1，0.1，0.7］。“最优”投资组合是夏普比例（Sharpie Ratio）最高的投资组合。

4、绘制最佳投资组合的图像，并和 SPY 指数作对比。

1.3. 第三周：事件分析（Event studies）库的建立

目前结果：成功学会如何基于历史信息进行事件分析、如何读入和处理不同类型的历史数据、如何评估事件分析的结果。这个任务中我的工作是用“事件分析”评估股票信息对未来价格的影响，其中 QSTK 中事件分析器的代码附于附件。

工作内容：

1、复习和理解 QSTK 关于事件分析的已成代码。我在开始写自己的事件分析器代码前 QSTK 中就已经有部分未完成的雏形，经过加工和完善后，它能让开发者自主描述市场事件，然后从统计的角度观察这些事件如何影响股票价格。事件分析器采用的算法是扫描一遍特殊事件的历史数据，然后计算该事件在股票价格过去和基于一段回顾的未来的影响。

2、做一个在 S&P500 指数的专有“已知”事件的事件分析，再比较它对两组不同领域股票的影响。事件的定义是股票的实际收盘价跌下$5。严格地说，如果 price[t - 1] ≥ 5, price[t] < 5，那么该事件发生于 t 时刻。将这个事件分析在时间区间 2008.1.1 到 2009.12.31 测试，对比两种 S&P500 清单的结果：A）S&P500 采用 2008 年 500 支股票的清单，B）S&P500 采用 2010 年的 500 支股票的清单（S&P500 是美国股票的一个指数，综合了股市中 500 支股票的信息，并且每年这 500 支股票都有小许变化，因此 08 和 10 年会是不同的股票清单）。时间分析的结果很显然会因为采用清单的不同而有所变化，我和教授就此讨论了下其中的原因。

3、后来我又自己定义了一些事件，并且用之前完善的事件分析器进行实验。关于这部分产生了很多自己的猜测，并和导师讨论了不少问题，由于未经验证此处不作具体内容的阐述，但就我笔记上和导师讨论过的问题作下介绍：A）有没可能通过自己的事件分析赚钱？B）如果可能的话，我咨询了下实业公司采用的投资策略，一起讨论入市和出市的时机以及持有期的细节问题。C）同时也对导师说的策略进行了抽象的风险评估。D）从期望的角度说，每次交易的回报会是多少？E）每年会有多少次机会发生定义事件？F）有没有什么方法可以在此基础上降低风险？

4、就之前写的程序，提交报告结果给导师，即一些数据的纪录，如
For the $5.0 event with S&P500 in 2012, we find **176** events. Date Range = 1st Jan 2008 to 31st Dec 2009.
For the $5.0 event with S&P500 in 2008, we find **326** events. Date Range = 1st Jan 2008 to 31st Dec 2009.

1.4. 第四周：市场模拟（Market Simulation）库的建立

目前结果：已成功完成市场模拟器，即可以接受交易订单，同时纪录投资组合的价值，并将之储存在.csv 文件中。同时也进行了另一个项目进行股票投资组合表现的评估。

工作内容：

1、制作市场模拟器，也就是写一个叫 market.py 的文件，可以接受象下面这样的一条命令行：
python marketsim.py 1000000 orders.csv values.csv

其中数字代表开始投资时刻的现金，orders.csv 是纪录了订单的输入文件，每条订单包括年、月、日、股票代码、买或卖和交易股票数量，如（2008, 12, 3, AAPL, BUY, 130），（2008, 12, 8, AAPL, SELL, 130）。模拟器需要每天通过实际收盘价计算投资组合的总价值，再将结果打印在 values.csv 文件中。values.csv 的结果输出会像是（2008, 12, 3, 1000000），（2008, 12, 4, 1000010），（2008, 12, 5, 1000250）以此类推。

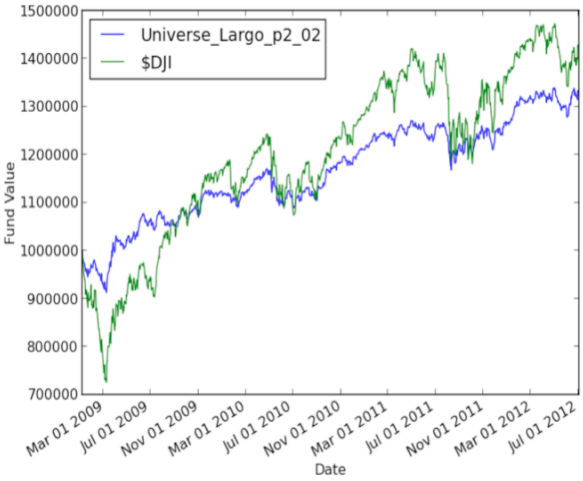
2、制作一个投资分析工具，叫 analyze.py，它能接受如下的一条命令行：
python analyze.py values.csv $SPX

该工具会从 values.csv 读入每日投资组合价值，并且绘制图像输出。它会使用命令行中的股票指数符号作为基准进行比较（在这个例子中是$SPX）。简单地说就是 analyze.py 可以绘制全交易期间的投资组合历史价值的图像，并且输出投资组合的日回报标准差、平均日回报、夏普比例和总回报，如其中一个导师提供的 orders.csv 作为输入的输出结果如下：

The final value of the portfolio using the sample file is -- 2011,12,20,1133860

Details of the Performance of the portfolio:
Data Range: 2011-01-11 to 2011-12-20

Sharpe Ratio of Fund: 1.2154
Sharpe Ratio of $SPX: 0.0183
Total Return of Fund: 1.1338
Total Return of $SPX: 0.9775
Standard Deviation of Fund: 0.0071
Standard Deviation of $SPX: 0.01498
Average Daily Return of Fund: 0.0005493
Average Daily Return of $SPX: 1.7223e-05



以上所阐述的输入输出文件，以及 Python 代码文件将作为附件提交。

1.5. 第五周：事件分析库和市场模拟库的连接

目前结果：这部分主要是将前面不同的项目进行结合，也就是将事件分析的输出用于建立更加完善的测试后台。再具体一点，从前面做过实验的定义事件中任意挑一个，用之前的事件分析器（Event Profiler）进行评估和调整，最后用之前的模拟器（simulator）进行后台测试。

工作内容：

1、重新修改事件分析器使得它能根据事件的发生输出一系列的交易订单。之前写的事件分析器只是在事件矩阵（可想象成横排是股票符号，纵列是时间的表格）中放置一个 1 标志事件的发生，现在是要让输出变成订单模式，如：
Date, AAPL, BUY, 100
Date + 5 days, AAPL, SELL, 100

2、将该输出作为输入传递给市场模拟器

3、通过市场模拟器得出交易策略的表现，如总回报、平均日回报、日回报标准差和交易时期的夏普比例。

4、根据导师的要求做了两个实验。

a) 实验一：采用在任务 2 中实现的实际收盘价$5 事件和 2012 年的 S&P500 数据。该实验和一个博士实习生分别独立实验，导师希望通过如此检查我们能否得到相同的答案以保证我们各自的事件分析器改造没有问题。当时要求的输入数据是：
启动资金: $50,000；开始日期: 1 January 2008；结束日期: 31 December 2009；当事件发生的时候，在当日买入 100 股股票；持有 5 天后自动卖出。

b) 实验二：设计自己的事件和交易策略，对于两个实验都要生成图像以便观察，同时象之前一样计算夏普比例、总回报和日收入的标准差，如下——

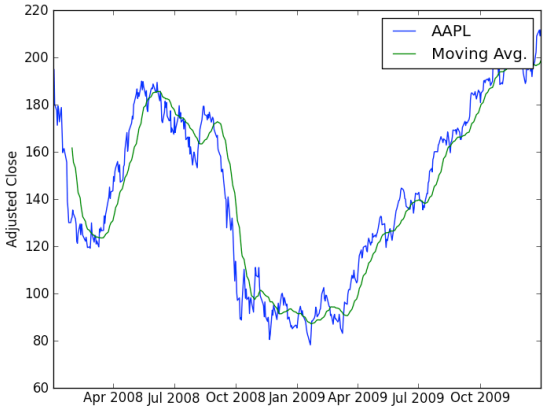
The final value of the portfolio using the sample file is -- 2009,12,28,54824.0
Details of the Performance of the portfolio
Data Range : 2008-01-03 16:00:00 to 2009-12-28 16:00:00
Sharpe Ratio of Fund : 0.527865227084
Sharpe Ratio of $SPX : -0.184202673931
Total Return of Fund : 1.09648
Total Return of $SPX : 0.779305674563
Standard Deviation of Fund : 0.0060854156452
Standard Deviation of $SPX : 0.022004631521
Average Daily Return of Fund : 0.000202354576186
Average Daily Return of $SPX : -0.000255334653467

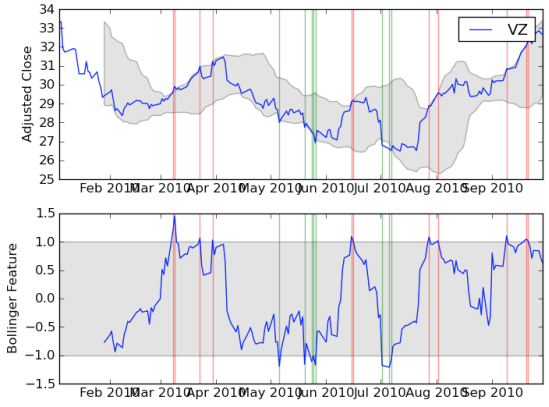
1.6. 第六周：实现财务指标（Financial Indicators）库，以 Bollinger Band 为例

目前结果：开始 technical indicator 的研究，完成 Bollinger Band 的案例实现，并为之后再进行一些最前卫的 indicator 实现做准备。

工作内容：

1、用 20 日回顾实现 Bollinger Band 指标。编写代码生成图形展示移动平均线（下左图），股票价格和 Bollinger Band 的高低线。Bollinger Band 的高线代表期望值加一倍的标准差，低线代表期望值减一倍的标准差。传统上应该是 2 倍的标准差，但导师希望我把 Bollinger Band 做得细一点所以只用 1 倍的标准差。





2、调整输出使得 indicator 产生的值在-1 和 1 之间。实际上，这些值是有可能被超过的，但我想导师的目的是希望+1 代表价格是高于期望一倍标准差的位置，-1 代表价格是低于期望一倍标准差的位置，为了实现这个变化，其实只要简单地加入下面这行代码——

Bollinger_val = (price - rolling_mean) / (rolling_std)

输出图像如右上图所示。

3、实现一些我自己设计的 Indicator，并把播报值调整到-1 和 1 之间。我进行了两个实验，实验一仍然采用上述的 Bollinger Band，并同时把输出值调整到－1 和 1 之间。然后生成了一个时间从 Jan 1, 2010 到 Dec 31, 2010 的 GOOG 图像，即输入数据是 Symbol: GOOG; Startdate: 1 Jan 2010; Enddate: 31 Dec 2010; 20 period lookback。

实验二是关于 relative strength 的 Indicator 实现，这部分是我最近才开始的工作，也是实验室着手使用的最新预测工具，涉及公司机密，我签署了不泄露合同所以不能在该实习报告中有所展示。但所作的工作步骤和上述 Bollinger Band 的实现是一样的，只是稍显复杂，以下是 Bollinger Band 实验的一个输出结果例子（采用 Python 中的 pandas 库可以直接计算 Bollinger band 的值）——

	AAPL	GOOG	IBM	MSFT
2010-12-23 16:00:00	1.185009	1.298178	1.177220	1.237684
2010-12-27 16:00:00	1.371298	1.073603	0.590403	0.932911
2010-12-28 16:00:00	1.436278	0.745548	0.863406	0.812844
2010-12-29 16:00:00	1.464894	0.874885	2.096242	0.752602
2010-12-30 16:00:00	0.793493	0.634661	1.959324	0.498395

1.7. 第七周：以 Bollinger Band 为例用财务指标进行事件分析

目前结果：这周内容是 event study 和 technical indicator 的结合。导师说他相信某些股票的 technical indicator 的走势与市场走势相反将比其它 indicators 更具有合理性，这部分就是要验证这个假设。这个项目是想寻找和某些特定 Bollinger Band 相关的事件，这些 Bollinger Band 的特点是它们的值和现市场中的值有明显的区别。

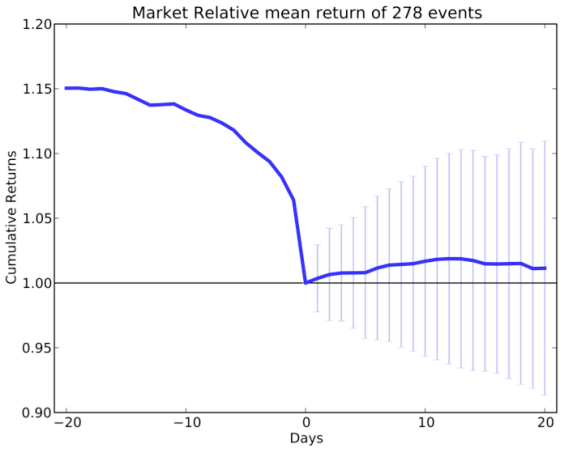
工作内容：

1、用 20 日回顾实现布林带 indicator。还是一样的定义，布林带的高线代表期望值加一倍标准差，低线代表期望值减一倍标准差。同时把 indicator 的输出值调整在－1 和 1 之间。

2、采用以下特征进行事件分析：Bollinger value for the equity today ≤ -2.0, Bollinger value for the equity yesterday ≥ 2.0, Bollinger value for SPY today ≥ 1.0.

因此我们是在寻找股票跌穿布林带低线，同时市场走势却是稳定地向上走。这说明肯定有些特殊的事情发生在个股上。

3、使用上一项目做出来的 indicator，再按上述方法进行事件分析，试图挖掘一些有意思的结果。该部分的实验导师也给了统一数据如下：Event 如 2 中所写，Startdate: 1 Jan 2008, Enddate: 31 Dec 2009; 20 day lookback for Bollinger Bands; Symbol list: SP5002012; Adjusted close. 以下是事件分析的一个结果。



1.8. 第八周：Indicators 的大规模实现

目前结果：这是我目前正在进行的项目，即将目前金融分析领域常用的 indicator 工具用 python 代码大规模实现，并嵌入 QSTK 之中。配对交易的机器学习需要训练集，因此需要 indicator 进行数据的收集以产生样式（pattern）的识别，这是我目前的任务。这个任务我已经做了有一周了，也是代码写得最多、金融词汇了解最多的一个项目。收获当然主要是编程经验的增加，金融分析领域概念的拓展，以及一些前卫 indicator 的认识。就如前面有提到的一点，现实验室里已经对 relative strength 的 indicator 做过测试，作为 indicator 对股票的走势有 70%左右的预测度，所以正打算将它大规模实现。我了解算法和该 indicator 的过程中也学习了金融分析领域最可靠的预测方法。

内容：我会附自己这周写的代码 lrfeatures.py 于附件中，代码长达五十多页，是我和另一个博士生 Tingyu 共同合作的结果，大部分是我写的，Tingyu 写的函数有注释标注。这部分内容其实和 Bollinger Band 的流程相似，只是不仅仅实现 Bollinger Band 这一种 indicator，而是实现了 104 种包含 384 种变化的 indicator 集合，方便以后进行事件分析的时候可以综合各种 indicators 一起使用。

2. 后续工作和目标

软件库开发（Developer）工作，包括策略评估指标、策略优化、事件分析库、市场模拟哭、库与库之间的连接和财务指标的实现，都已经基本结束，后八周可以进入量化分析阶段开始策略开发（Quant）工作。以下是大致计划——

2.1. 第九周－第十周：

根据 Quantitative method and analysis 介绍的经典配对交易策略模拟构架，实现三步模型：

- 一、数据挖掘（Data Mining）算法进行高协整（Co-integration）配对的搜索
- 二、机器学习（Machine Learning）算法进行高盈利性（Profitability）配对的筛选
- 三、后台测试（Back test）策略，生成金融交易报告

初期采用简单的算法进行配对的发现、选择和交易，研究几种直接、自融资交易法则的盈利效果。 虽然配对策略只是挖掘股票信息的临时成分，但可以理论证明利润并不只是像文献所述的仅有均值回

归，对此也将该论文算法层面陈述。

2.2. 第十一周－第十三周：

运用前八周建造的后台软件库对模型进行参数调整和测试，实验实现以 Python 为主， 代码将兼顾风险因子，以便合理研究结果的稳定性，即不只包含广泛使用的因子如价格数据，同时也兼顾低频的机构因素如破产风险，均加以量化。另外，在后台模拟测试（back test）代码中也会考虑金融微观因素如买入与卖出滑移（slippage），卖空利率（short-selling interest）和交易成本（transaction cost），更趋近真实交易。

2.3. 第十四周－第十六周：

在基于以上算法的模型构造和代码实现后，我将进入实验分析阶段进行各模型的搭配模拟测试效果，包括改造回归模型，多种数据挖掘算法对配对进行搜索选择，多种机器学习算法对高利润型配对进行筛选，以及对量化因素可靠性的时间序列分析，最终以金融分析中的夏普比率（Sharpe ratio）进行策略优劣的比较。完成该部分试验后，撰写最终版论文介绍收益性效果最好（即夏普比率最高）的算法搭配组合和策略模型，展现分析报告。