FE520: Python application in Finance: Pairs Trading

Bing Yu
byu6@stevens.edu

Jian Yang
jyang19@stevens.edu

Zhe Zhao
zzhao6@stevens.edu

1 Schedules

From	To	Progress
10/07/2014	10/28/2014	Build connection with Interactive Brokers
10/29/2014	11/4/2014	Code the naive version of the Pairs Trading Strategy
11/5/2014	11/11/2014	First improvement: kappa filter and volume adjustment.
11/12/2014	11/25/2014	Second improvement: use a basket of stocks instead of one stock.
11/26/2014	12/02/2014	Reports and Final Presentation

2 Objectives

In this project, we aim at building a pairs trading strategy in python. We will connect the strategy with Interactive Brokers and trade on paper accounts.

The first plan is to trade on both daily and intraday base. Our data for model calibration comes from Thomson Reuters Tick History, and we use zipline as our back testing system. The algorithm is based on a book written by Dr. Rupka Chatterjee [1]. We also want to make our own improvement.

3 Project Description

3.1 IB API Connection

In order to implement this strategy in reality, we need to connect to some platform to get QUOTES and send ORDERS. And IB is one of the best choice.

Since IB doesn't have an offical python API, we are using the C# API as well as a third party library "Python for dot net" together. Through this library, we can seamlessly call any dot net function and object from python.

Since we are integrating two different languages and rely on some third party software, we would like to make the connection provide as much functionality as it can. Some parts of this project have to be done in C#. Here is a list of functionality that can be provided by this connection:

- Read symbol files from CSV.
- Get quote data and save to containers.
- Record trading results.

Besides these parts, in python we will process historical and quote data, and calibrate the parameters in our model. Also whenever we have improvements, they will be implemented in python as well. We suppose there are still a lot work to do in python for this project.

3.2 Data Source

In this project, we will use three different types of data to trade. We will download the daily data from Yahoo Finance and download the 1 minute data and hourly data from Thomson Reuters Tick History.

3.3 Algorithm

We aim to implement a Factor Neutral ETF Statistical Arbitrage strategy between a stock and a sector ETF. By monitoring the appearance of the pairs and comparing with the trading rules, we can trade automatically.

The strategy roughly consists fo these parts:

Peform linear regression on stock and ETF returns

$$r_t^{stk} = \alpha + \beta r_t^{etf} + \epsilon_t$$

Define residual process as

$$X_t = \Sigma_{k=1}^{t} \epsilon_k$$

Comparing this process to the Ornstein-Uhlenbeck(OU) process

$$dX_t = \kappa(m - X_t)dt + \sigma dW_t$$

According to this model, we will open positions whenever Xt goes far away from the long time average m, and will wait for it comes back and make money.

3.4 Back Testing

We will simulate Algorithmic Trading and conduct the backtesting in Zippline under the python environment. The backing testing process is carried out in the following steps:

- Import the historical data
- Process our models through the data resources
- Compare the spread, also known as the hedge ratio between the two ETFS that we are choosing,
- Calculate the Z-score which is the standard score of the spread
- Create long and short market signals and update pairs contain those signals

4 Improvements

Given that naive pairs trading is hard to make money, we want to somehow find ways improve this trading strategy. Here is a potential list

- We will try to implement two filters on stock returns: Kappa filter and volume adjustment filter.
- Instead of using one stock and one ETF as a pair, we will use a basket of stocks and one ETF as a pair.
- Further adjustments.

References

[1] Rupak Chatterjee. *Practical Methods of Financial Engineering and Risk Management: Tools for Modern Financial Professionals.* Apress, 2014.