Developing Trading Algorithms on Quantopian

Lucy Wu

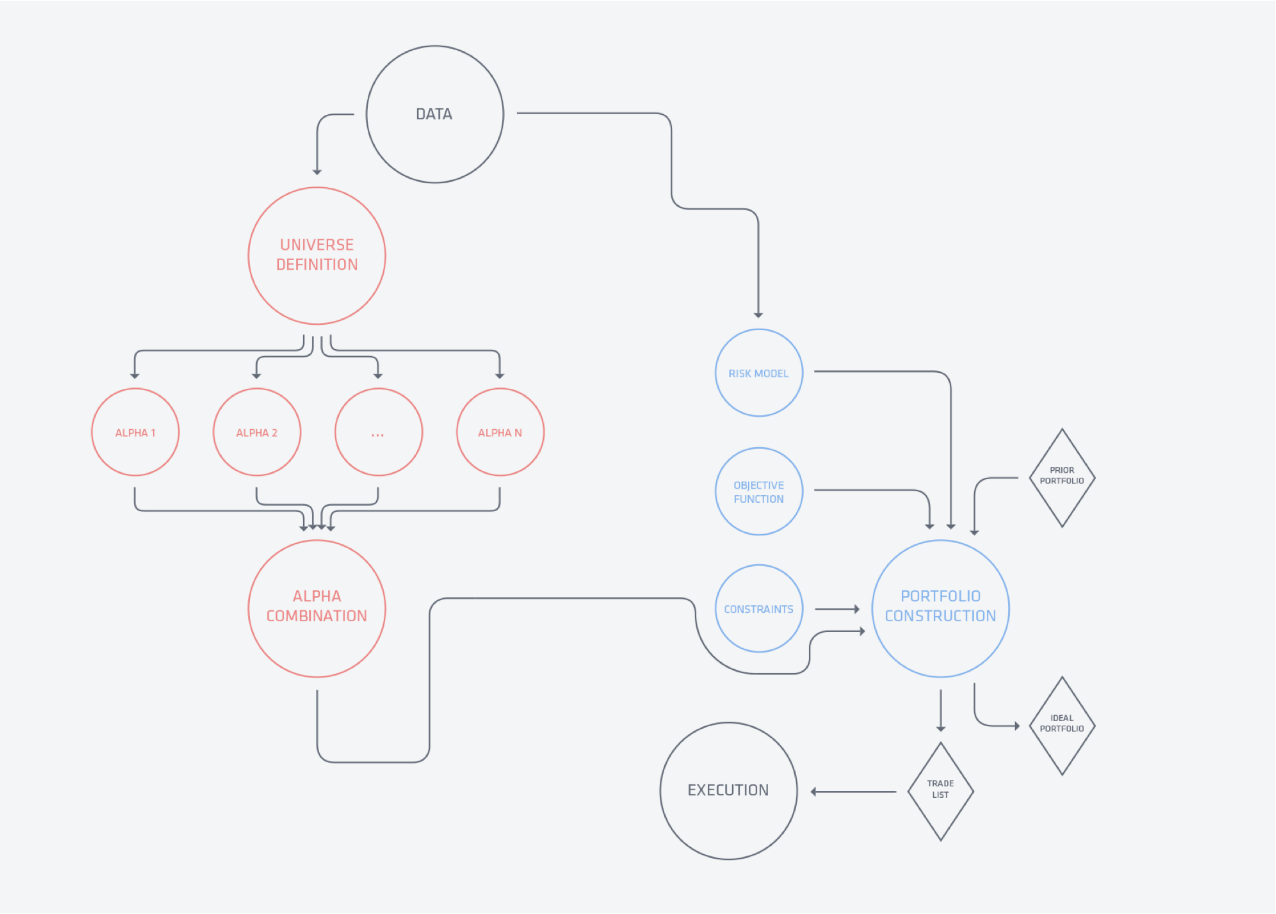
November 13, 2018

Overview

- What is a trading algorithm?
- Research: Creating a “thesis”
- Building: Writing algorithms on Quantopian
  - Mean reversion example
- Backtesting: Interpreting backtests on Quantopian

What is a trading algorithm?

Data → Executing Trades



What is a trading algorithm?

Very basically…

1. Develop “thesis”
2. Collect data needed by thesis
3. Create algorithm that executes trades based on this data
4. Backtest and analyze results
5. Iterate *without overfitting*

*For daily/minutely long-short equity algorithms – *not high-frequency trading*

Why Quantopian?

Free, well-maintained data

Fundamentals, pricing, alternative datasets

Adjusted for stock splits/mergers/dividends

Pre-built tools

Backtesting engine

Risk analysis

Signal analysis

Community

Forums

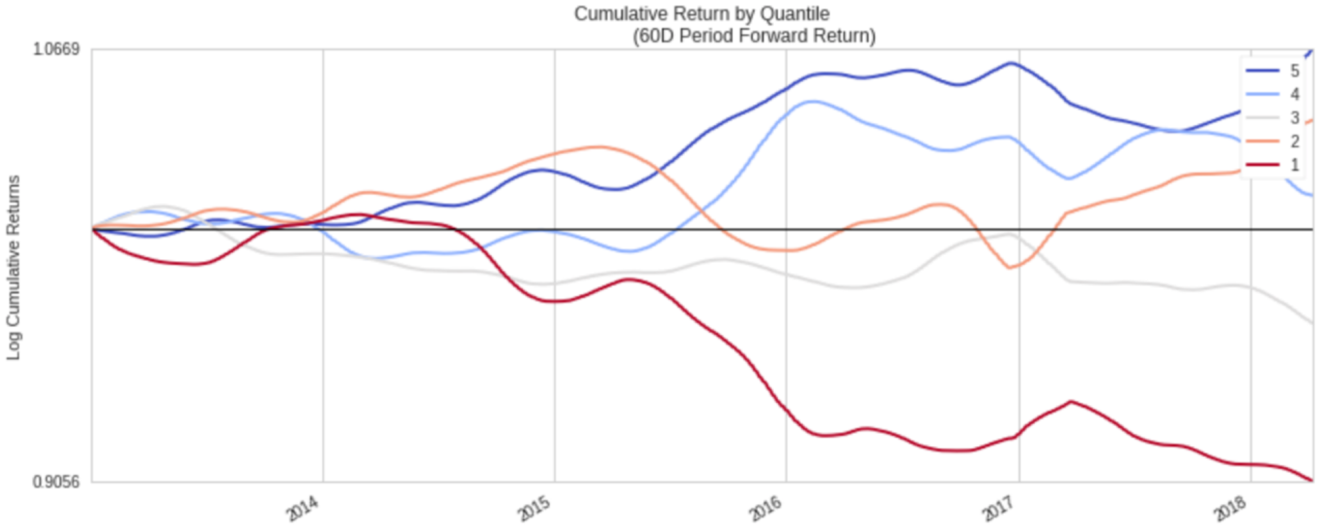
Contests!

Choosing a Thesis

Examples

- Mean reversion
- The “weekend effect”
- Companies that substantially change their 10-Ks and 10-Qs will generate lower returns than companies that do not change their 10-Ks or 10-Qs.

- Investigate using Alphalens:



*For daily/minutely long-short equity algorithms – *not high-frequency trading*

Importing Data

- Via `Pipeline` objects
  - Define universe
  - Choose dataset and fields from dataset

Algorithm Structure

- `initialize()`
  - Initializing global parameters
  - Scheduling functions like `rebalance()`
  - Attaching a Pipeline
- `before_trading_start()`
  - Daily data retrieval and computations
- `rebalance()`
  - Executing trades

Simple Momentum Example

- If price is above the 5-day simple moving average: Buy
- If price is below the 5-day simple moving average: Sell

Backtest Analysis

- Things to look at:
  - Returns and variability of returns
  - Correlation with the market (beta)
  - Sector exposure
  - “Style” risk
  - Position concentration
  - Turnover

Other things to consider...

- Using Optimize API
- Recording variables
- Using non-pricing data
- Importing your own data (Self-Serve Data)