Stock Picking & Structural Breaks

Jeppe Sndergaard Johansen (pcv439)

August 21, 2019

Introduction & Overview

This paper investigates strategies for stock picking, where stock picking is assumed to be at any period only having a single stock, in an environment of structural breaks. The investigation follows the structure of:

1. Model outline.
2. A description of data.
3. Model identification and fitting the parameters. A data set is simulated using these results.
4. An ensemble of algorithms are discussed, and the best performing algorithm is found using the simulated data set.
5. The best performing algorithm is applied on the real data set, and compared to a set of benchmarks.

DGP in classic CAPM

- Returns are assumed to follow a DGP with constant covariance matrix and expected returns
- In this application i've restricted the assumption s.t. the returns are normally distributed.

$$\mathbf{R}^{(t)} \sim N(\mu, \Omega) \quad (1)$$

DGP under assumptions of structural breaks

- I modify the assumption, and say at any point in time a structural break can occur.
- A structural break implies a new covariance matrix and vector of expected returns are drawn

$$b^{(t)} = bern(p) \tag{2}$$

$$\mu^{(t)} \sim d_\mu \qquad \Omega^{(t)} \sim d_\Omega, \qquad \text{if } b = 1 \tag{3}$$

$$\mu^{(t)} = \mu^{(t-1)} \qquad \Omega^{(t)} = \Omega^{(t-1)}, \qquad \text{if } b = 0 \tag{4}$$

$$\mathbf{R}^{(t)} \sim N(\mu^{(t)}, \Omega^{(t)}) \tag{5}$$

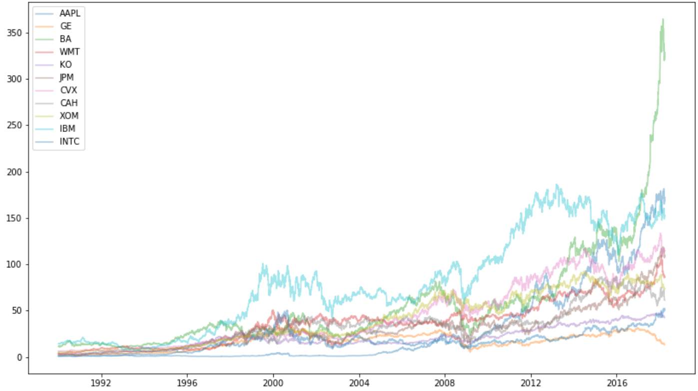
Data

- The data consists of 11 stocks for approximately 30 years ($\approx$ 7000 observations).
- The data is acquired through a public API provided by **Quandle**.
- The individual stocks represents a wide variety of companies from different sectors.
- To get stationarity in the data, the data is transformed by:

$$r_i^{(t)} = \frac{a_i^{(t)}}{a_i^{(t-1)}} - 1 \tag{6}$$

- The risk free asset is assumed to be of 2% annually throughout the paper.

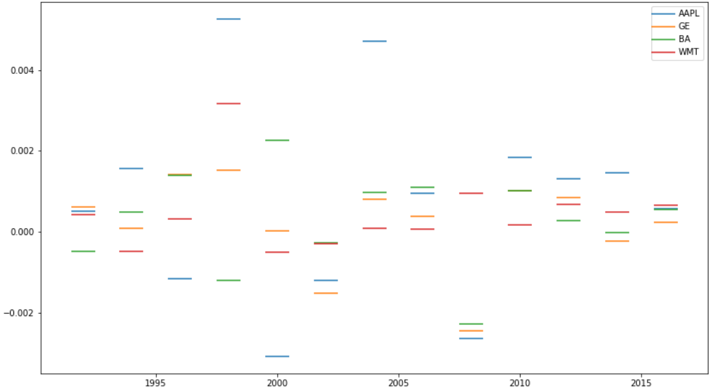
Historical traces



Summary statistics of Stocks

	AAPL	GE	BA	WMT	KO	JPM	CVX	CAH	XOM	IBM	INTC
mean	0.0011	0.0004	0.0006	0.0006	0.0005	0.0008	0.0005	0.0007	0.0005	0.0005	0.0009
std	0.0282	0.0175	0.0187	0.0165	0.0141	0.0241	0.0153	0.0188	0.0145	0.0174	0.0240
min	-0.5187	-0.1279	-0.1763	-0.1018	-0.1047	-0.2073	-0.1249	-0.2454	-0.1395	-0.1554	-0.2202
25%	-0.0128	-0.0079	-0.0092	-0.0079	-0.0064	-0.0101	-0.0078	-0.0084	-0.0072	-0.0079	-0.0115
50%	0.0001	0.0000	0.0001	0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0001	0.0004
75%	0.0144	0.0087	0.0104	0.0087	0.0072	0.0107	0.0089	0.0097	0.0082	0.0087	0.0130
max	0.3322	0.1970	0.1546	0.1107	0.1388	0.2510	0.2085	0.2039	0.1719	0.1316	0.2012

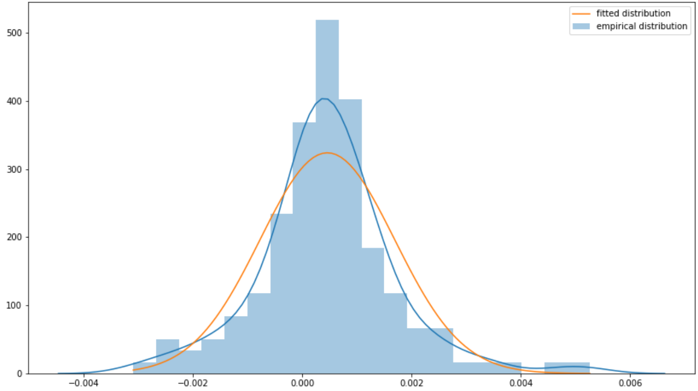
Structural Breaks (Visual inspection)



Identification strategy

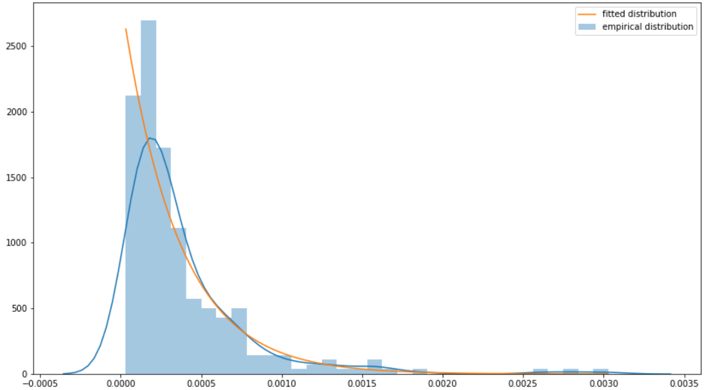
- Sampling from the structural model, requires the distributions to be specified, and the parameters of the distributions to be estimated.
- The d_μ is assumed to be normal, and d_Ω is found by a transformation of variances and correlations.
- the variances is assumed to be exponentially distributed, and the correlations is assumed to be normally distributed.
- When the model is identified, $11 \times 2.000.000$ observations are simulated. This data set can be used to training and tuning different algorithms on.

Distribution of Returns, d_μ

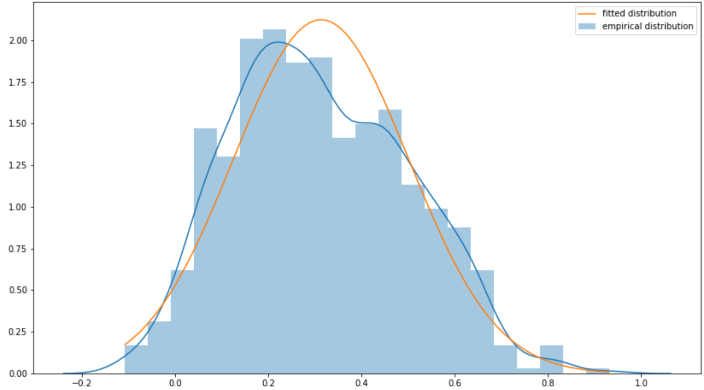


Distribution of Variances, σ_i^2

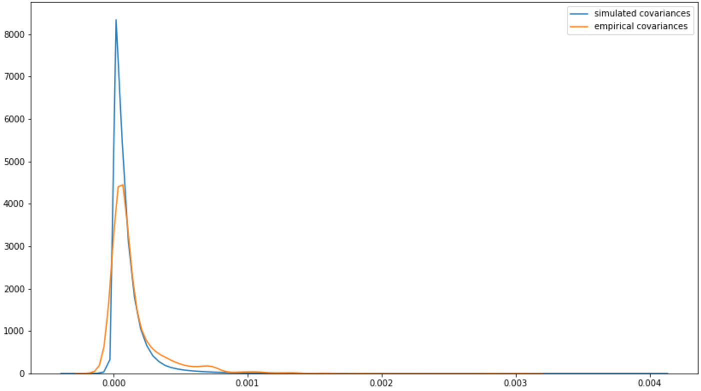
Needing to sample the covariance matrix, I find the distribution og variances, and the correlation matrix.



Distribution of individual correlations, $\rho_{i,j}$



Simulated Covariances vs. Empirical Covariances, $\sigma_{i,j}$



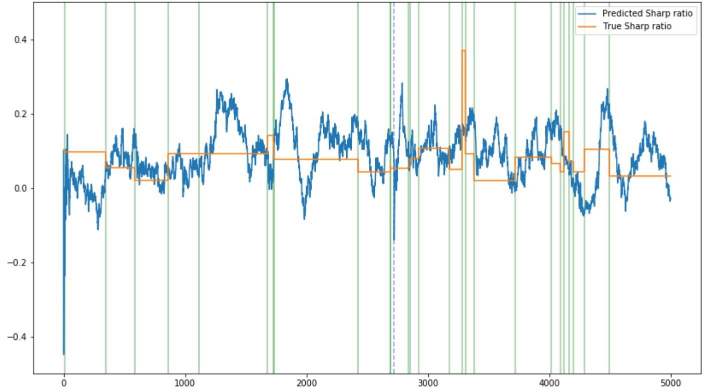
Overview of 3 algorithms

- **LSTM**: Neural network based algorithm used for patern representation in time series problems. Used in self driving cars, and state of the art chess computers.
- **Naive Rolling Sharpe** Use the rolling Sharpe ratios on the different stock to pich the stock with highest sharpe ratio
- **Rolling Sharpe** An extension of the algorithm above, but with the capability to forget its distant past.

Strategy for algorithm selection

- The three algorithms are tested on the simulated data set. The best performing algorithm (lowest *mean squared error*) is used on the real data set.
- Since the true variance and std. deviations in the simulated dataset are known the true Sharpe ratio can be calculated for each simulated stock in each time step.
- The *mean squared error* is found between the true Sharpe ratio and the Sharpe ratio predicted by the algorithm.
- The best performing algorithm is found to be **Rolling Sharpe**

Predicted vs. actual Sharpe ratios



Analysis

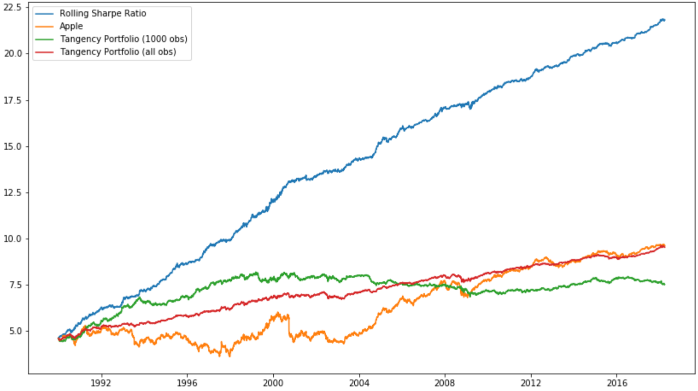
- Taking the **Rolling Sharpe** algorithm to the real data we need benchmarks.
- 4 benchmarks is used: 1) The *Apple* stock, 2) The tangency portfolio calculated on the first 1000 observations, 3) The tangency portfolio calculated on the entire sample, 4) Perfect foresight.

Performance Comparison (part 1)

	Expected Return	Std	Sharpe Ratio
rolling sharpe	0.00263	0.02028	0.12562
perfect stock pick	0.02515	0.02285	1.09734
apple	0.00111	0.02824	0.03661
tangency portfolio (1000 obs)	0.00058	0.01830	0.02719
tangency portfolio (all obs)	0.00077	0.01246	0.05545

Performance Comparison (part 2)

Counterfactual portfolios, using different strategies



Performance Comparison (part 3)

Monte carlo simulation, using moments of portfolios from real data.