Two Machine Learning Approaches for Statistical Arbitrage Pairs Selection

Brian Chan, Nhi Truong, Carina Zhang

Introduction

Pairs trading is a strategy that assumes returns between two stocks A and B are linearly dependent,

$$\frac{dA_t}{A_t} = \alpha dt + \beta \frac{dB_t}{B_t} + dX_t, \quad (1)$$

with the spread being mean-reverting

$$dX_t = \kappa(m - X_t)dt + \sigma dB_t. \quad (2)$$

Following [1], the strategy relies on the signal $s_t = \frac{X_t - m}{\sigma}$. When $|s_t| > 1.25$, we short the overpriced stock and long the underpriced one. We closed when $-0.25 < s < 0.5$.

Previous literature focuses mainly on estimating (1). We applied two methods: 1) factor modeling with PCA and 2) CAE to find stock clusters with similar features, from which we picked the most optimal pairs.

Data

Stock data for top 500 companies in 2010-11 is obtained via Kaggle. Only stocks with non-missing values are admitted (467). We use 2010 data to train and 2011 data to test.

Methodology

Stage 1: Choose stock pairs using method described in next column.

Stage 2: Run trading simulation with ARIMA modeling to get Sharpe Ratios (SR).

Stage 3: Repeat stage 2 for random pairs; bootstrap to compute p-values.

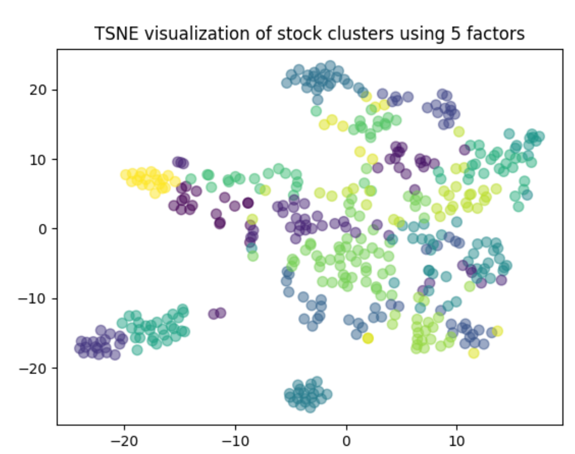
ML Approaches for Choosing Pairs

1. Factors Model & PCA

Let X be the return matrix of N stocks over T days. We use factor model

$$\underbrace{X}_{N\times T} = \underbrace{\Lambda}_{N\times L}\underbrace{F}_{L\times T} + \underbrace{e}_{N\times T}.$$

We estimate Λ as L largest eigenvectors obtained from PCA.



2. Convolutional AutoEncoder (CAE)

Time Series as Images

Following [4], we use Gramian Agular Field (GAF): scale stock price S_t to be in $[-1, 1]$, then GAF is the matrix $[\cos(\phi_i + \phi_j)]_{1\le i,j\le T}$ with $\phi_t = \arccos(S_t)$.

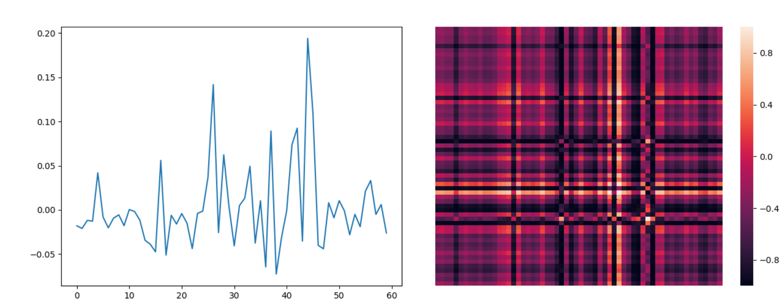
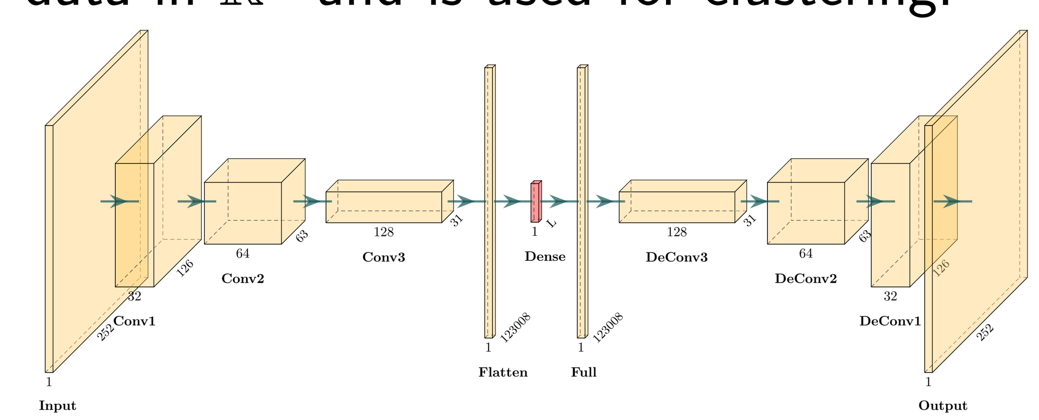


Figure: GAF of a return series

CAE Architecture

We used the CAE proposed by [2]. The stacked convolutional layers learned hierarchical features. The output of the red layer is compressed data in $\mathbb{R}^L$ and is used for clustering.



Final Screening: Use regression to compute α, β and dX_t in (1). Approximate (2) with ARIMA model and use Kalman filter to estimate κ, m, σ. Choose pairs with the smallest p-values from ARIMA estimates.

Results

Hyperparameter Tuning

Hyperparameters: number of factors in approach 1, number of features in approach 2, number of clusters in both approaches. Three approaches for choosing number of factors: **thresholding**, **information criterion**, **random matrix theory** [3]. From the figures, the optimal number of clusters $K = 20$ and the optimal number of $L = 5$. Underfitting (resp. overfitting) seems to be an explanation for lower SR with lower (resp.) higher values of K, L in both models.

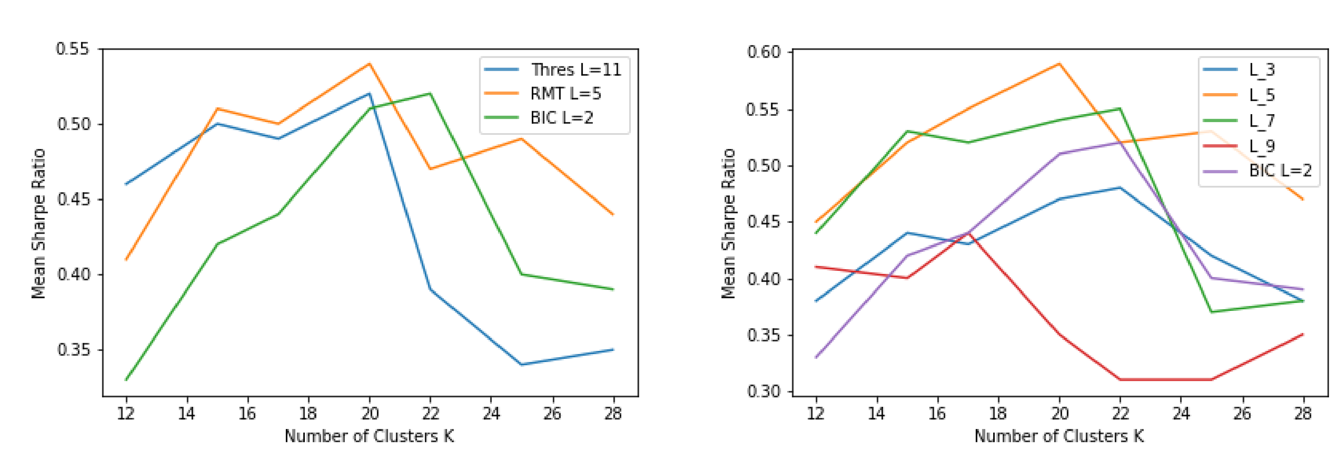


Figure: Left: Factor model, Right: CAE

Performance

We presented below the train and test results of the two methods (with optimal $L = 5$, $K = 20$) versus the baseline of random picking pairs in the same industry.

Method	Metrics	Train	Test
Factor Model	Mean SR	0.54	0.33
	p-value	10^{-6}	0.009
CAE	Mean SR	0.59	0.39
	p-value	$< 10^{-6}$	0.003
Baseline (Industry)	Mean SR	0.18	0.17
	p-value	0.34	0.31

Across, strategy $\beta \approx 0.01$ (market neutral) but high $\alpha \approx 0.1$.

Discussion

- Using both approaches to cluster stocks (Factor Models with PCA or CAE) outperforms the baseline method of randomly choosing stocks from the industry.
- Using the optimal $L = 5$ and $K = 20$, the p-values obtained in both train and test simulations reject the null hypothesis of random picking at the 1% level.
- CAE is a much more complicated model but does only slightly better than factor model and PCA.

Extension

- We implemented rolling window train-test and had great portfolio-wise SR (≈ 1.2), but need more run time for bootstrapping.
- We haven't found time to tune other hyperparameters in CAE such as epochs, batch size.
- The only form of regularization in CAE right now is the low number of features in compression.

Selected References

[1] M. Avellaneda and J. Lee, in Quant. Finance 10(7), 2010.
[2] X. Guo, X. Liu, E. Zhu, and J. Yin, in ICONIP, 2017.
[3] A. Onatski, in Rev. Econ & Stats, 2010.
[4] Z. Wang, T. Oates, in Proc. 29th AAAI Conf. Artif. Intell., 2015.