Detection of mean-reverting phenomenon from American stocks with unsupervised techniques

C. Billet
*TELECOM Nancy**
claire.billet@telecomnancy.eu

S. Krzyzaniak
*Credit Agricole SA.**
stephan.krzyzaniak@credit-agricole-sa.fr

R. Sellem
*Credit Agricole SA.**
remi.sellem@credit-agricole-sa.fr

Abstract

This study aims to present different techniques in order to capture the mean reverting phenomenon on the market. The data used are the return of 398 assets extracted from American stocks over the period 2004-2017. To capture this structure, it is necessary to use algorithms such as principal component analysis, sparse principal component analysis, which eliminates noise compared to the first technique. Neural networks are also tested, autoencoders and LSTMs. These four algorithms enable us to reconstruct the returns. This reconstruction is compared with actual returns. The difference between actual returns and their reconstructions is considered as an anomaly. Indeed, if a daily actual return differs from the reconstructed return, this means that the real value does not correspond to the market structure. The mean-reverting in a market assumes that if a return deviates from the expected value then it will return to its standard value imposed by the market structure. Thus, this anomaly suggests taking a position of buying or selling for the concerned action. We construct strategies using anomalies that make it possible to compose portfolios on a daily basis. As result, the study shows that the strategies are generally successful over the test period, which validates the existence of mean-reverting on the market.

1 Introduction

In investment, it is important to have robust and winning strategies. This study cares about the existence of mean-reverting phenomenon in the market and to be able to capture this phenomenon with unsupervised algorithms.
Our approach to detect the mean-reverting phenomenon is: first we want construct a model to capture the market structure, then we detect anomalies in this structure, finally we show that the anomalies are corrected in time.
This work follows the Antoine Isnardy's study [6], who tested different autoencoder models to capture the mean-reverting phenomenon.
The mean-reverting is the returns variations around a mean reversion, this means that if a return deviates from this mean then it will return to its standard value. So, we can set up strategies on this phenomenon.
This study aims to make a benchmark of several solutions like PCAs, Autoencoders and LSTMs in order to capture the mean-reverting phenomenon and built robust and winning strategies based on the hypothesis of the existence of the mean-reverting phenomenon in the market.
The rest of the paper is organized as follows. Section 2 presents the experiment as a whole and step by step. Section 3 reports the results of this experiment and the paper is concluded in Section 4.

2 Experiment

This section is to provide an overview of the study and present it step by step.

2.1 Experiment presentation

The aim of the study is to detect the mean-reverting phenomenon. The mean-reverting strategy exploit difference between the value of an asset and its reference value. When the value exceeds the reference value, positively or negatively, then the value will revert to reference value. We consider this difference as an anomaly, the problem is thus an anomaly detection problem. The aim is therefore to find a function f modeling market's structure, f must re-built the market, like:

$$\begin{array}{rccc} f: & \mathbb{R}^n & \to & \mathbb{R}^n \\ & r=(r_1,...,r_n) & \mapsto & \tilde{r}=(\tilde{r}_1,...,\tilde{r}_n) \end{array} \quad (1)$$

*The views expressed in this paper are solely the responsibility of the authors and should not be interpreted as reflecting the official positions of TELECOM Nancy or Credit Agricole SA. The usual disclaimer nonetheless applies, and any remaining errors remain ours.

Electronic copy available at: https://ssrn.com/abstract=3074360

We calculate f using machine learning algorithm, f is not the identity. The reconstruction of the asset i is noted $\tilde{r}_i$, the real return of the i^{th} asset is expressed as: $r_i = f(r_1, ..., r_n)_i + \varepsilon_i = \tilde{r}_i + \varepsilon_i$, with ε_i the reconstruction error.

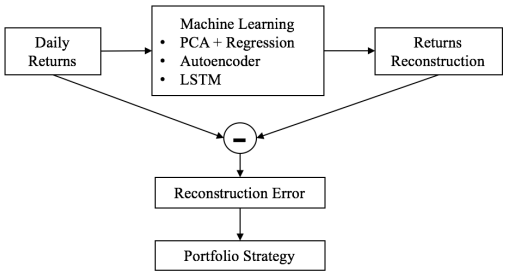

Figure 1: Global view of the study

The figure 1 present the general principle of the study. The study is to detect the mean-reverting phenomenon from an anomaly detection problem. To this end we try to reconstruct the daily returns with machine learning algorithms like a Principal Component Analysis (PCA), an Autoencoder or a LSTM. This reconstruction is compared to the actual returns and the difference between this both returns is considered as an anomaly. We use this anomaly to build portfolio strategy.

2.2 Datasets

As dataset, we use American stocks in the period 2004-2017. So, we need to have assets quoted on the stock exchange since 2004 and not stopped before 2017. It is why we select 398 assets on 500. From 2004 to 2014, data are the training dataset, 10% of this dataset is used for validation and the rest, from 2014 to 2017 is held out as the testing dataset. 31% of the assets have a positive average return in the period 2004-2017. The average return of an asset during that period is -0,0082% and the standard deviation is 0,036% so the returns oscillate around 0.

2.3 Machine Learning for anomaly detection

In this part, we present the four machine learning algorithms we use for anomaly detection. We begin with Principal Component Analysis and Principal Component Analysis on a sparse covariance matrix, then we introduce our Autoencoder model and our LSTM models.

Principal Component Analysis (PCA)
A Principal Component Analysis transforms correlated variables into uncorrelated variables called principal components. With this technique, we bring out patterns and select the most important information in dataset. In our study, we use a PCA or a sparse PCA followed by a regression. What we call a sparse PCA, it is a PCA on a sparse covariance matrix. D'Aspremont uses semidefinite optimization to select covariance and he shows the difference between a covariance matrix and the sparse covariance matrix. The variable selection is more important with the sparse covariance matrix and the patterns in dataset are stronger [4] [5].

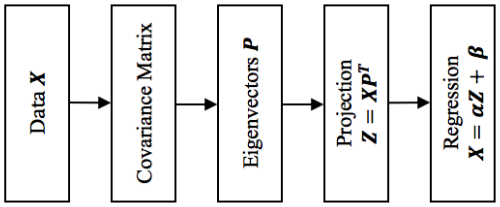

Figure 2: Principal Component Analysis model

Figure 2 shows the process of our PCA and regression. We begin with the calculation of covariance matrix, sparse or not. With our dataset, we obtain a covariance matrix with 79 202 parameters to calibrate. We extract of this matrix the eigenvectors. We choose 83 eigenvectors because they explain 80% variance. We project data $\mathbf{X}$ in 83 eigenvectors matrix $\mathbf{P}$ and we obtain $\mathbf{Z} = \mathbf{XP^T}$.
Then we proceed with the linear regression with which we rebuilt the real returns $\mathbf{X}$ with $\mathbf{Z}$, the regression is as follows: $\mathbf{X} = \alpha\mathbf{Z} + \beta$. The linear regression estimate α and β coefficients to find the actual returns.

Autoencoder
An autoencoder is a data compression algorithm with a compression function and a decompression function [2]. Each function is implemented with neural network. We create an autoencoder with two fully-connected layers. The first layer is the encoder and the second is the decoder to rebuilt the input. We can see our model in figure 3, every day the autoencoder has daily returns of 398 assets and he tries to reconstruct this daily returns.
We evaluate the anomaly detection performance of Autoencoder in terms of Mean Absolute Error. The number of neurons for the hidden layer is chosen from the number of eigenvectors selected in PCA and then we test increasing and reducing the width of this layer to achieve the best performance. By increasing the width of the hidden layer we increase the number of parameter too, as we can see in table 1.

Long-Short Term Memory
Thanks to a Long-Short Term Memory network, called LSTM, we hope to capture a time structure in financial

Electronic copy available at: https://ssrn.com/abstract=3074360

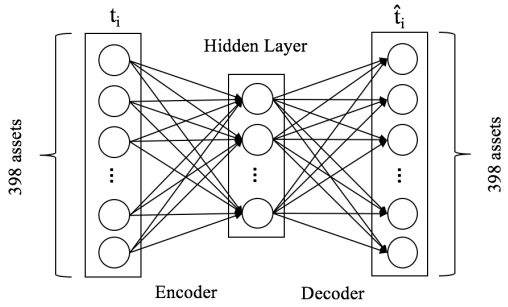

Figure 3: Autoencoder model for study

Hidden Layer Width	Learning time (s)	Cost	Parameters[1] number
(30)	51.56	0.0091	24 308
(83)	61.9	0.0077	66 549

Table 1: Learning time, cost and parameters number for each autoencoder model

series. Indeed, with its four gates and its cell state, a LSTM is special kind of recurrent neural network, capable of learning long-term dependencies [3].

As we want to capture a time structure, the neural networks have to learn on a sliding window. We study the returns of each day of a week and we realize the returns are more important on Monday and Friday. This study determines the size of a sliding windows, it must have a size of 5 days minimum to take a full trading week into account.

To answer to our anomaly detection problem, we implement a LSTM *many-to-one*, it means that we learn on many variables to build one variable. We construct a LSTM that it learns of 5 successive trading days and rebuilt the 5th day, as we can see in figure 4. One day is made up of the return of 398 assets.

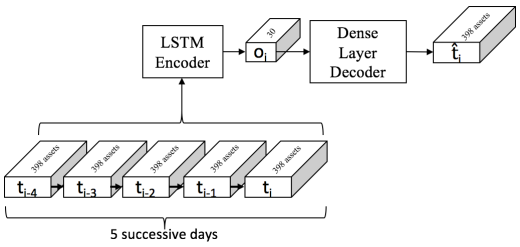

Figure 4: LSTM model for study

The aim of the study is to obtain robust strategies based on the hypothesis of the existence of the mean-reverting phenomenon. Even if the reconstruction is perfect but strategies are poor, the solution is unacceptable. With 600 neurons on the LTSM, the reconstruction is almost perfect but the number of parameters is 2.5 times taller than our dataset, it is overfitting.

As we can see in table 2, the parameters number increases considerably with the neural network width. The structure the most interesting in our case is the first structure thanks to the number of parameters being reasonable against our data.

LSTM dimension	Learning time (s)	Cost	Parameters number[2]
(30)	15.95	0.0091	63 818
(80)	61.9	0.0084	185 518
(600)	233	0.0031	2 636 798
(600,80,30)	334	0.101	2 641 178

Table 2: Learning time, cost and parameters number for each LSTM model

We test another structure of the LSTM, see figure 5, we construct a predictive LSTM [1]. Instead of learning on 5 successive days and reconstructing the last day of this 5 days, we learn on 5 successive days and we predict the 398 returns of the next day.

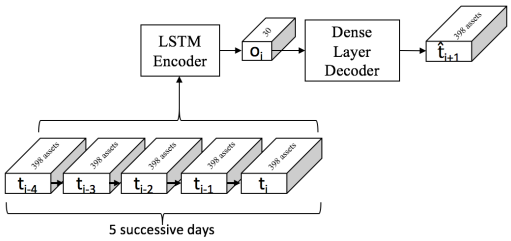

Figure 5: Predictive LSTM model for study

2.4 Error detection method

Thanks to machine learning algorithms we obtain the returns reconstruction $(\tilde{r}_1,...,\tilde{r}_n)$ of real returns $(r_1,...,r_n)$. our problem is a anomaly detection problem and we use this anomalies to set up strategies. The error is also defined as the difference between the reconstructed return and the actual return that means $\forall i \in [1,n], \varepsilon_i = \tilde{r}_i - r_i$. With this error definition, that can be deduced is that if:

- $\varepsilon_i \ll 0$, the actual return is superior to estimated return, this asset is most cost-effective than expected, it has to be sold.
- $\varepsilon_i \gg 0$, the real return is lower to estimated return, this asset is less cost-effective than expected, it has to be bought.

There are different approaches to take decision to sale or buy an asset like relative error or endogenous threshold.

Relative error

The relative error calculate the error importance i.e. $\frac{\varepsilon_i}{|r_i|} = \frac{\bar{r}_i - r_i}{|r_i|}$. We can measure the error against the real return, so we use the importance of the error against its expected value.

Endogenous threshold

The second that we use to consider if an error is important or not, is the endogenous threshold. We calculate two thresholds for each asset. Each threshold is a quantile calibrated using the training error for each asset, thus they take the behavior of each asset into account. So, we have two quantiles on the distribution of errors of i^{th} asset, $q_{i,\frac{\alpha}{2}}$ at $\frac{\alpha}{2}\%$ and $q_{i,1-\frac{\alpha}{2}}$ at $1-\frac{\alpha}{2}\%$. When the test error is in the first quantile, we sell this asset and when the test error is in the second quantile, we buy this asset.

Define this threshold can to take the behavior of each action into account and ensure that each asset participates $\alpha\%$ of time in strategies.

2.5 Financial strategies

To evaluate algorithms performance and to check the existence of mean-reverting phenomenon, we use financial strategies.

Study strategy

The first strategy is a study strategy that means we study net long/short portfolio. A net long portfolio is a portfolio built with bought assets whereas a net short portfolio is a portfolio built with sold assets. It is a question of studying the robustness of this strategy i.e. the impact of addition of an asset in one of portfolio.

To determinate the importance of the error and the decision to sell or buy, we use the relative error and each asset in this portfolio has equal weight.

Production strategy

The second financial strategy is the production strategy. This strategy uses the endogenous threshold to take a decision to sell or buy an action. With this technique, we have a variable number of assets in each strategy. The assets are allowed in a portfolio with Equally-weighted Risk Contributions. The weight for an asset is inversely proportional to its volatility, its risk and it is as follows: $w_i = \frac{\sigma_i^{-1}}{\sum_{j=1}^{n} \sigma j^{-1}}$.

To compare the strategies results, we construct a benchmark that is a net long portfolio with the 398 assets of the market weighted by the ERC method [7].

3 Experiment Results

In this section, we present our results throughout the error distributions and the strategies.

3.1 Error distributions

The first point to mention is the error distributions are centered as we can see in table 3 with the means of distributions. The sign of error is an indication to sell or buy and action, if it is negative we have to sell this asset and buy if the error is positive. So, it is important to have a centered distribution that does not affect the purchase or the sale of an asset. No bias is introduced.

Algorithm and time	Mini Mean	Maxi Mean
PCA train	-5.2×10^{-18}	4.6×10^{-19}
PCA test	-1.5×10^{-18}	1.8×10^{-18}
Sparse PCA train	-2.6×10^{-18}	3.7×10^{-18}
Sparse PCA test	-2.2×10^{-18}	2.2×10^{-18}
Autoencoder train	-0.006	0.005
Autoencoder test	-0.007	0.006
LSTM train	-0.002	0.004
LSTM test	-0.008	0.008

Table 3: Error means for each algorithm

3.2 Strategy results

The aim is to have robust and winning strategies in order to confirm the existence of mean-reverting phenomenon. We evaluate each strategy with the Sharpe ratio to know its robustness and its gain. The Sharpe ratio is defined as $SR = \frac{\bar{r}}{\sigma}$ that is the ratio between the performance and the volatility of the strategy.

Study strategy

For the study strategy, we plot matrix of Information Ratio i.e. we have matrix where each cell is the Information Ratio of the strategy when you buy l assets against you sell s assets every day. For example, we study the robustness when we buy 50 assets and we sell 40 assets each day.

In the table 4, we report the means of Information Ratio for each test period. We note that with a PCA we obtain a good strategy in 2014 but these results are not robust in time because the Information Ratio decreases in time. The sparse PCA gives similar results even if the results appear little more robust in time. With the autoencoder with 30 neurons, we have robust and winning strategy in time because the deviations between each period are low. By contrast, with the autoencoder with 83 neurons, the deviations are more important, the market is very volatile

in 2015 and this autoencoder does not succeed to capture this structure and make winning strategies. The LSTM succeed to capture the structure of this volatile year but for the other year, the success is not the same. Here we can note that the predictive LSTM is not a good structure as an answer to our problem.

Algorithm	Mean of Ratio Sharpe			
	2014	2015	2016	2014-2017
PCA	0.15	0.06	0.03	0.07
Sparse PCA	0.14	0.05	0.07	0.08
Autoencoder 30	0.14	0.11	0.08	0.10
Autoencoder 83	0.16	0.04	0.09	0.9
LSTM	0.09	0.18	0.07	0.11
Pred LSTM	0.007	0.017	-0.01	0.0001

Table 4: Mean of Information Ratio for each algorithm with study strategy

Production strategy

Comparing this four next tables, tables 5, 6, 7, we can say all algorithms, beyond the predictive LSTM, allow to obtain stronger and more efficient strategies than the benchmark.

With the production strategy, we choose a parameter α to calculate the size of the two quantiles. α has a value in 0.3, 0.6, 0.9, 0.95. When α is 0.9 or 0.95 we take in consideration almost all assets so all the market. By contrast, when we choose α at 0.3 or 0.6, we have less assets in the strategy, this strategy is also riskier. Thus, these two values of α are the most interesting in order to obtain a robust and winning strategies.

In table 5 relevant to 2014, we remark that the PCAs and the autoencoders have good values of Information Ratio. The Information Ratio for the LSTMs are lower than the benchmark for α in 0.3, 0.6, 0.9.

Algorithm	α			
	0.3	0.6	0.9	0.95
PCA	0.20	0.18	0.24	0.23
Sparse PCA	0.10	0.09	0.07	0.08
Autoencoder 30	0.12	0.19	0.18	0.18
Autoencoder 83	0.11	0.19	0.22	0.20
LSTM	0.05	0.10	0.10	0.11
Pred LSTM	-0.02	-0.02	-0.01	-0.001
Benchmark	0.102			

Table 5: Information Ratio for production strategy in 2014

In table 6 relevant to 2015, we can see the values for the PCA and autoencoders stay robust even if this year is very volatile. The LSTMs are again under our expectations.

Algorithm	α			
	0.3	0.6	0.9	0.95
PCA	0.05	0.10	0.17	0.18
Sparse PCA	0.03	0.09	0.09	0.09
Autoencoder 30	0.09	0.14	0.12	0.13
Autoencoder 83	0.04	0.11	0.10	0.10
LSTM	0.01	0.02	0.06	0.10
Pred LSTM	-0.03	-0.02	-0.02	-0.02
Benchmark	0.007			

Table 6: Information Ratio for production strategy in 2015

In table 7 relevant to 2016, the results for the PCA are poorer than the other years and this confirms that the PCA is not robust in time. The sparse PCA succeeds this year.

Algorithm	α			
	0.3	0.6	0.9	0.95
PCA	0.09	0.07	0.07	0.07
Sparse PCA	0.14	0.14	0.13	0.13
Autoencoder 30	0.09	0.12	0.11	0.10
Autoencoder 83	0.11	0.12	0.12	0.12
LSTM	0.1	0.07	0.07	0.07
Pred LSTM	0.04	0.03	0.03	0.03
Benchmark	0.077			

Table 7: Information Ratio for production strategy in 2016

The two autoencoders allow to obtain robust Information Ratio in time unlike the PCAs. The LSTM obtains poorer results than the other algorithms and even sometimes poorer than the benchmark. We can confirm with this tables that the predictive LSTM is not a model for our data and our problem.

4 Conclusion and Discussion

On one hand, the autoencoder with 30 neurons learns with less parameters than the other algorithms and the autoencoder with 83 neurons, reminded in table 8. On the other hand, with this autoencoder the strategies are robust in time and they are winning. So, the autoencoder is the best answer to our study to detect the existence of mean-reverting phenomenon. We can confirm too that the mean-reverting phenomenon exists in the American stocks considering these strategies are winning.

With its ability to capture a time structure, we hoped LSTM learn the returns structure of financial series, but it does not. We have 398 returns on 2500 days i.e. about 1 million of data, 2500 days is certainly not enough to calibrate the LSTM. Indeed, at each step the LSTM have

Algorithm	Parameters number
PCA	79 202
sparse PCA	79 202
Autoencoder 30	24 308
Autoencoder 83	66 549
LSTM	63 818

Table 8: Parameters number for each algorithm

to reconstruct 398 information with an historical of 5 days and long-term memory. To conclude the structure of the LSTM does not match with the data and an autoencoder with 30 neurons is enough to obtain robust and winning strategies with the hypothesis of the existence of the mean-reverting phenomenon.

References

[1] BROWNLEE, J. Time series prediction with lstm recurrent neural networks in python with keras. *Available at: machinelearningmastery.com* (2016).

[2] CHOLLET, F. Building autoencoders in keras. *The Keras Blog, blog.keras.io* (2016).

[3] CHRISTOPHER, O. Understanding lstm networks. *Available at colah's blog, http://colah.github.io/posts/2015-08-Understanding-LSTMs/* (2015).

[4] D'ASPREMONT, A. Quelques applications de la programmation semidfinie. *MATAPLI* (2013).

[5] D'ASPREMONT, A., BANERJEE, O., AND GHAOUI, L. E. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and its Applications, 30(1), pp. 56-66* (2006).

[6] ISNARDY, A., AND KRZYZANIAK, S. Autoencodeurs appliques la construction de portefeuilles mean-revering. *Available at SSRN: 2952833* (2016).

[7] MAILLARD, S., RONCALLI, T., AND TEILETCHE, J. Equally-weighted risk contributions: a new method to build risk balanced diversified portfolios.

Notes

[1]Parameters number for a dense layer formula: $input\_dim \times output\_dim + output\_dim$

[2]Parameters number for a LSTM layer formula: $4 \times ((input\_dim + 1) \times output\_dim + output\_dim^2)$