# Comparison of Supervised & Unsupervised Learning in \betas Estimation between Stocks and the S&P500

J. Wei, Y. Hassid, J. Edery, A. Becker, Stanford University

#### I. Introduction

THE goal of our project is to analyze the relationships between stocks within the S&P500, and use various machine learning techniques that we've studied in class to do so. Ideally, the analysis will lead us to the discovery of effective ways to calculate the βs between stocks and the S&P500, which would ultimately allow us to obtain a proper hedge ratio for trading the stock vs. the S&P500.

To premise, for most of our analyses, we will use the log returns of the data, as in financial modeling of data typically people do not look at the absolute values of the data but instead at the percent returned for the asset (i.e.  $S_t / S_{t-1}$ ), and taking the logarithm allows us to introduce some linearity to the problem.

At the end of our project, we modeled the relationship between the stocks with the S&P500 using a stochastic differential equation, and by focusing on the idiosyncratic component of this model we were able to detect a signal to trade off of. We end our project with an analysis of our various trading strategies and their respective Profit and Losses (PnL).

Our data comes courtesy of EvA Hedge Fund in San Francisco, CA, and consists of intra-day data (specifically, 15 minute time intervals) of the S&P500 and its components.

#### A. Motivation: About the $\beta$

The  $\beta$  is the coefficient between the *market* component and the *constituent* component, namely it can be expressed as the linear relationship between:

$$LogRet(S_{Market}) \sim \beta * LogRet(S_{Component})$$

The  $\beta$  in turn has several natural interpretations as well as uses — though all of the interpretations and uses are intimately related. The natural interpretation is that the  $\beta$  can be used as a hedge ratio between the constituent component (i.e. the stock or the sector of stocks) and the market component (i.e. the S&P500 index). This ratio tells us the amount a trader would have to buy/sell the market/stock to remain "risk neutral".

Another use for the  $\beta$  is for forecasting and prediction. By modeling the linear relationship between the log returns, one could then use the estimated  $\beta$  to forecast future values of

the time series. By doing so, one could determine when the stock is trading too cheap or rich to the market, and then take advantage of this inefficiency by initiating a trade. This difference between the stock and the market can be used as a "signal" in high-frequency trading, and is the main motivation behind the stochastic differential equation proposed by Avellaneda et al<sup>1</sup>, and which is explained in more detail in the following section.

## B. Back-testing methodology: Modeling

We used a quantitative approach to stock pricing based on relative performance within industry sectors or PCA factors, which has been presented in [1]. The stock prices are noted  $S_i(t), \ldots, S_N(t)$  where t is time, and where the indices are noted  $I_j(t)$ . In the case of supervised learning,  $I_j(t)$  represents the price of the  $j^{th}$  factor used to span the market. The stock returns are modeled according to the following stochastic differential equation:

$$\frac{dS_i(t)}{S_i(t)} = \sum_{i=1}^{N} \beta_{ij} \frac{dI_j(t)}{I_j(t)} + dX_i(t)$$

It is composed of:

- a systematic component  $\sum_{j=1}^{N} \beta_{ij} \frac{dI_j(t)}{I_j(t)}$ , driven by the returns of the indices
- An idiosyncratic component  $dX_i(t)$ . It is assumed to be the increment of a stationary stochastic process which models price fluctuations corresponding to over-reactions or other idiosyncratic fluctuations in the stock price which are not reflected the industry sector.

The  $dX_i(t)$  component is modeled as an Ornstein-Uhlembeck process, i.e. follows this model:

$$dX_i(t) = \kappa_i(m_i - X_i(t))dt + \sigma_i dW_i(t), \kappa_i > 0$$

where  $\kappa_i$  is an indicator of the mean-reversion speed. This process is stationary and auto-regressive. In particular, the increment  $dX_i(t)$  has unconditional mean zero.

# C. Back-testing methodology: trading strategy

We focus only on the process  $X_i(t)$  and define the dimensionless variable:

$$s_i = \frac{X_i(t) - m_i}{\sigma_i}$$

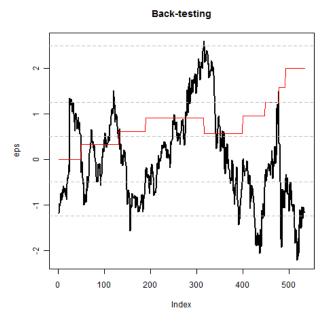
where  $m_i = E\{X_i(t)\}$  and  $\sigma_i = Var\{X_i(t)\}$ . These values are computed with a sliding window before the time of estimation t.

The s-score measures the distance to equilibrium in units standard deviations of  $X_i(t)$ , the cointegrated residual which is computed from  $dX_i(t)$ . In other words, it describes how far away a given stock is from the theoretical equilibrium value associated with our model.

Our basic trading signal based on mean-reversion is

open long position (buy)if  $s_i < -s_{bo}$ open short position (sell)if  $s_i > +s_{so}$ close long position if  $s_i > -s_{bc}$ close short position if  $s_i < -s_{sc}$ 

where the hedge values are determined empirically. In practice, we used  $s_{bo} = s_{so} = 1.5$  and  $s_{bc} = s_{sc} = 0.5$ . Such a trading strategy is illustrated by the figure below:



where the axis represents the time, the black line is  $s_i$  and the red line is the PnL. Notice that the PnL fluctuates only on specific times, corresponding to points where  $s_i$  crosses one of the hedges define above.

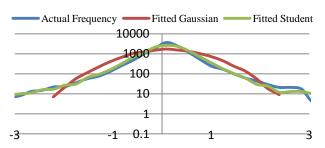
Notice also that the PnL may drop because we used two additional cut-offs  $s_{slc}$  and  $s_{slo}$  both equal to 2.5 which represent 'stop-loss' hedges, at which we close our position, despite the fact that the cointegrated residual does not returns to the mean. This accounts for jumps in the stock value which aren't taken into account in our mean-reverting model, but which breach our risk limits. By closing our position, it results in a money loss and explains the drops of the PnL, but is necessary in practice because of risk limits.

#### II. SUPERVISED LEARNING

# A. The different norms

The main difficulties with our dataset concern its distribution. Over the two-year period we have, we cover a period of big growth and the financial crisis. The first problem is that our dataset may not be stable, i.e. the betas in those different periods may differ, due to fundamental market conditions. With the analogy of house prices, we can imagine that the house prices have fundamentally changed between those periods, meaning that it may not be possible to learn a model from one period and apply it to the other. The other difficulty is that the distribution is not Gaussian, there are a lot of jumps and big variations, and a T-distribution seems to be a better fit.

## Modeling the dataset distribution



In order to deal with those problems, we have selected different objective function to minimize.

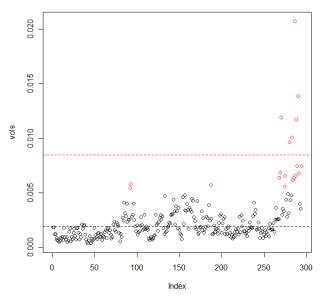
The usual one involves the L2-norm, but it does not seem robust to the non-Gaussian case, and does not adapt to the fundamental changes in the dataset.

The L1-norm has the property of being much more robust to outliers. For example, in a simple 1D setting, using the L2-norm, we know the Maximum Likelihood estimator of a series of number is the mean. However using the L1-norm, this estimate is the median, which is more robust to outliers.

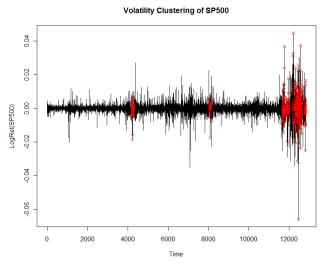
In order to deal with the instability and the fundamental changes, we can divide our dataset into two different clusters: one for the highly volatile days, and one for the other days. We can use a K-means or a mixture of Gaussians to separate the different days. We have observed that during the crisis, we had much more volatile days than before, and in practice one could either turn off their trading strategies during high-volatility days, or one could design a separate trading strategy that handles highly-volatile days well and switch between the two. In our setting, we tested one strategy which uses the same  $\beta$  for both periods of time (high-volatility and low-volatility) and a separate strategy which only predicts during low-volatility days. To calculate volatility, we were careful to only use half-a-days worth of

data, as in practice the objective is to "guess" whether or not a day will have low-volatility and high-volatility as early into the day as possible. Hence, this naturally introduces a possible *misclassification error rate* which we calculated to be 5.78%. Once we had computed all half-day volatilities, running K-means gave us:

## Volatility Classification with 2-Means



From which we can see a very clear *volatility clustering* effect when illuminating all "high-volatility" days:



Using the mean squared error (MSE) as a measurement for strength of prediction, we compare the two strategies and find that the MSE for the first strategy (same  $\beta$  for all days) was 1.73 times higher than the MSE for the second strategy (only forecasting on low-volatility days).

However, despite these positive results, for the purposes of PnL and back-testing we chose to implement the naïve strategy of using the same  $\beta$  for all days. Finally, we note that we could also combine the methods, using Mixture of

Gaussians for the weights and the L1-norm for example. Below is a summary of the different supervised methods we used as well as their respective objective functions:

Norm	Objective
L2-Norm	$\min_{\beta} \sum_{i,j} (Y_{ij} - \beta_j^t F_i)^2$
L1-Norm	$\min_{eta} \sum_{i,j} \left  Y_{ij} - oldsymbol{eta}_j^t F_i \right $
Mixture	$\min_{\beta} \sum_{i,j} \frac{1}{\sigma_i^2} (Y_{ij} - \beta_j^t F_i)^2$

# B. Choosing the norms and parameters

We need to choose different parameters such as the ratio of Test Size to Training Size, the numbers of factors we use or the overall time interval we can use. Let's plot the test error/training error or the influence of each factor to choose.

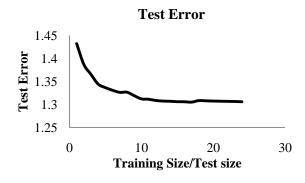


Fig. 1. Test error with respect to the ratio Training Size / Test set size using the usual L2-norm.

The improvement in the Test error is small after a ratio of 10, which we will use for the backtesting. Furthermore, the Cross-validation error is fairly stable over time with the total set size, meaning that we can expect the same performance for different time interval.

Finally, we have assessed the importance of each factor by the increase in error when it is excluded. The financial sector is the most important, as the financials played an important role during the crisis, and the Telecom sector is the least important.

#### C. Back-testing results

In order to backtest our models, we have used the trading strategy described in part B.

The results are very different depending on the period we choose, and our strategy is not able to yield good returns in a consistent manner. During the crisis, and precisely at the time of Lehman's failure, we have a huge loss. On average we get a yearly return of 13% a year, which is not that bad for such a period, and a Sharpe ratio of 2.8 over ten days, which is the standard. The mixture of Gaussians method gives the best returns and Sharpe Ratio.

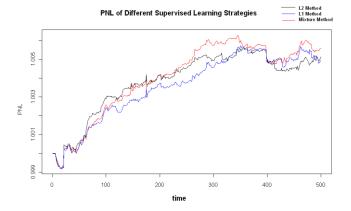


Fig. 2. PnL comparison of L1, L2, and Mixture of Gaussians supervised learning methods. Mixture of Gaussians (red) performed best.

## III. UNSUPERVISED LEARNING

#### A. Motivation

Though supervised learning provides a sufficient framework to find the  $\beta s$  and trade, we would like our strategy to rely on indices chosen to be the most relevant ones to explain the considered log return stock, rather than pre-defined.

To address this problem, we used the Principal Component Analysis, which enables us to identify the driving forces of the market and predict the evolution of the stock in terms of very few indices which explain most of the market's variance. The advantage of the Principal Component Analysis as an unsupervised learning technique is that we make no assumption on the predictor variables; the algorithm finds itself the linear factors that best explained the response.

The commonly used L2-PCA maximizes:

$$\arg\max_{\boldsymbol{W}} \left\| \boldsymbol{W}^T \boldsymbol{X} \right\|_2$$

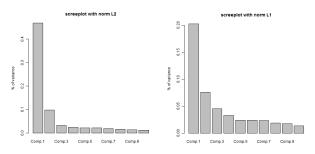
When the maximization problem is expressed in terms of the L2 norm, the result is unfortunately very sensitive in presence of outliers and could result in a skewed estimation of the betas. Thus, we modified the maximization problem in terms of the L1 norm, which provides a Robust Principal Component Analysis.

$$\underset{W}{\operatorname{arg\,max}} \left\| W^{T} X \right\|_{1}$$

Though the traditional L2-PCA was performed through R's standard package, we used the pcaPP package in R to perform the computation of the L1-PCA. It provides the Robust Principal Components using the Grid search algorithm, presented in [2].

#### B. Results

Our graphs show the eigenvalues of the PCA components for the L1 and L2 norms. As expected, the part of the variance explained by the first components is higher with the L2 norm than with the L1 norm.

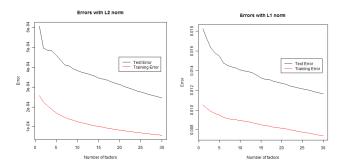


We defined the training and test error of our model:

$$Error - L2 = \frac{1}{N \times M} \sum_{i,j} (Y_i^j - \beta_j^i F_i)^2$$

$$Error - L1 = \frac{1}{N \times M} \sum_{i,j} |Y_i^j - \beta_j^t F_i|$$

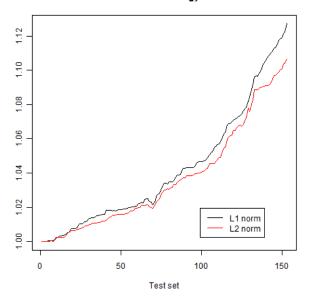
The graphs below represent the difference of a particular stock and its estimated value against the number of factors used in the estimation, averaged over all the S&P500 stocks. As expected, the error decreases when the number of factors increases.



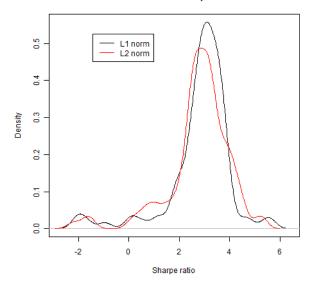
# C. Back Testing of strategy

In order to test if we should use the L1 norm rather than the L2 norm, we performed a back-testing strategy on the stocks using the two different  $\beta s$  estimations. We train the PCA on 2/3 of the data to estimate the  $\beta s$  and the factors, we then execute our statistical arbitrage strategy on the test set with estimated  $\beta s$  and factors.

#### PnL of strategy



#### Distribution of Sharpe ratio



The graph on the top represents the Profit and Losses (PnL) of the trading strategy presented in the introduction, starting at 1.00. The graph on the bottom presents the distribution of Sharpe ratios obtained over the different S&P500 stocks (5 factors were used in the PCA).

We obtained that the L1 norm has a better PnL and Sharpe ratio distribution in many cases. In our example, the PnL is 1.127 for L1 norm compared to 1.106 for L2 norm after 150 iterations and the mean of the Sharpe ratios are of 2.81 for the L1 norm and 2.77 for L2 norm. This result is very encouraging and proves that our intuition that the outliers affect negatively the estimation of the  $\beta s$  was true.

#### IV. CONCLUSION

For our project, we investigated the relative performance of several supervised learning methods (L1-regression, L2regression, and Mixture of Gaussians), as well as several unsupervised learning methods (L1-PCA and L2-PCA). Because of the fat-tailed nature of many financial time series, there is a higher tendency for outliers and intuition tells us that by using a more robust statistic – namely the L1norm – we could have more success in our modeling. Using PnL and Sharpe Ratio as a metric for determining the effectiveness of a given  $\beta$ , we determined that for both supervised and unsupervised learning methods, the L1-norm indeed performed better and in fact had similar returns for both cases. More specifically, in the case of supervised learning methods, we saw that the Mixture of Gaussians for the weights with the L1-norm objective function had a yearly return of about 13%, while the L1-PCA for the unsupervised learning method had a yearly return of about 12.7%.

For future improvements, we would like to re-implement our back-testing algorithm so that it tries to test for mid-day volatility and take this into account before deciding to trade. We believe that by eliminating highly volatile days (such as the period surrounding the Lehman incident) we can improve our overall returns and also improve upon the consistency of our returns.

#### REFERENCES

- [1] Avellaneda, M. and Lee, J.-H. (2008). 'Statistical Arbitrage in the U.S. Equities Market' (11 July)
- [2] Croux C, Filzmoser P, Oliveira M (2007). 'Algorithms for Projectionpursuit Robust Principal Component Analysis.' Chemometrics and Intelligent Laboratory Systems