Asset Management: Advanced Investment Final Project

Paride Iadonisi[1], Marco Inglin[2], Lorenzo Noci[3]

***Abstract*—In this paper we implement a Long-Short-Time-Memory (LSTM) based neural network to estimate future stocks prices. Then, we construct optimal portfolios through the Markowitz Mean-Variance (MV) method. We found that the LSTM portfolio performs better than the standard MV portfolio, which uses past data as proxy for future returns and volatility. Those findings are true for each output window we tested. While the LSTM method is sensitive to the applied output window it is much less than the standard Markowitz optimization. We experiment several techniques for returns estimation, including regularizing the covariance matrix and combining different estimations of returns and covariance matrix to reduce uncertainty in estimation in a Bayesian fashion. To analyze the portfolio weights, we use a clustering method for stocks based on non-negative matrix factorization (NNMF) with which we capture the latent trends of the market. We show that a low-risk portfolio has a well diversified allocation of stocks according to the latent trends found, and that these latent trends actually well represent the market.**

I. Asset Allocation

One of the main contribution to quantitative portfolio management was provided by Markowitz [1] with his paper "Portfolio Selection", in which he lays out the mean-variance (MV) analysis, a tool to select portfolios. The main idea behind the analysis is to select portfolios based on the expected returns and the risk an investor is willing to take, the higher the risk the higher the expected performance. As shown in Fig. 1. the standard deviation decreases by adding stocks to the portfolio, in the limit the risk is given only by the covariance which is the non-diversifiable risk, thus the risk which can't be eliminated by increasing the number of assets to the portfolio. The diversification power of stocks is given by the fact that the correlation between stocks is less then one.

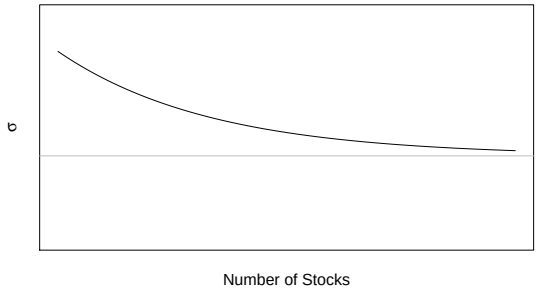
A. Minimum Variance portfolio

For a given number n of risky assets we can find the minimum variance portfolio for all portfolios with mean return μ_p by optimization of the following equation:

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{\Sigma}w - \boldsymbol{\mu}^T\boldsymbol{w} \\ \text{subject to} \quad & \mathbf{1}^T\boldsymbol{w} = 1, \\ & -1 \leq w_i \leq 1, \forall i \end{aligned} \tag{1}$$

1 Master of Arts in Business and Administration, UZH.
2 Master of Arts in Banking and Finance, UZH.
3 Master of Science in Data Science, ETH.

Fig. 1. Diversification Effect.



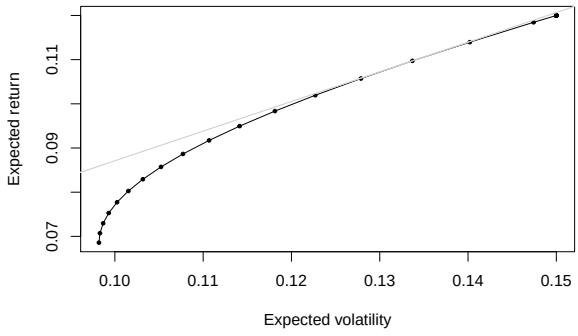
Note that we allow the stock weights to be less than 0 to allow short-selling.

If we collect all portfolios that solve this minimization problem for different $\boldsymbol{\mu}_p$ we obtain the minimum variance frontier, also called efficient frontier. If we add a risk-free asset with return R_f, then the weight assigned to the risky portfolio is given by:

$$\mathbf{w}_p = \frac{\boldsymbol{\mu}_p - R_f}{(\boldsymbol{\mu}^e)^T\Sigma^{-1}\boldsymbol{\mu}^e}\Sigma^{-1}\boldsymbol{\mu}^e \tag{2}$$

where $\boldsymbol{\mu}^e$ is the vector of excess return of the assets over the risk-free rate and $1–\mathbf{1}^T\mathbf{w}_p$ is the fraction invested in the risk-free asset.

Fig. 2. Markowitz Efficient Frontier.



In Fig. 2. we can see the efficient frontier and the line with all combination of risky and risk-free assets; the so called cap-

ital allocation line. The line is tangent to the efficient frontier, whereby the tangent point is called the tangent portfolio. The tangent portfolio is entirely composed of risky asset.

B. Tobin's Separation Theorem

According to Tobin's Separation Theorem, the composition of the fraction invested in the risky portfolio is independent of the targeted return of the entire portfolio, and thus of the investors risk aversion. Therefore the only difference between investors is the fraction they invest in the risk free asset. Risk averse individual will have a high proportion of of risk-free assets while less risk avers individual can have zero risk-free or even take on a loan, thus leveraging their investment. Furthermore the tangent portfolio is composed of all assets of the investment universe with the respective market capitalization, therefore it is often called the market portfolio [2].

C. Markowitz Weaknesses

The Mean Variance analysis comports some weaknesses, such as restricting the risk measure to variance, or excluding subjective views from the portfolio allocation process, relying only on data driven views. In this subsection we are going to analyze how to overcome three of the main problems of the Mean Variance optimization. The extreme portfolios weights, the high sensitivity on inputs which leads to large weight movement in the optimal portfolio and the lack of quantification in the confidence in estimated returns.

The Mean-Variance portfolio is extremely sensitive to changes in the input data, this is caused by the errors in the estimation of expected returns [3]. One way to improve the estimation is the resampling method [4]. The main idea behind this method is to estimate a large set of efficient frontiers, which are close to the original one, but are constructed using different allocation to the various assets. This reduces the reliance on only few assets to construct the MV portfolio, which is the case in the standard Mean-Variance model. From this set of efficient frontiers we calculate the average asset weights to construct the new portfolio. This new portfolio, based on resampling has now the advantage to have smother changes in the weights as we select different risk aversion and optimize for them[4].

To overcome the problem that Markowitz allows only data driven views, therefore no possibility to introduce subjective views of the market and that leaves no opportunity to quantify the confidence in the estimations, is possible to make use of the Bayesian statistics [5]. The effect of including prior information is to reduce the degree to which the estimated means are over-fitted, in particular in data–scarce situations. The investor finds itself in a trade–off between the available data and the data distribution. The subjective views – also called priors – play an increasing rule in situation of low data availability. There are three types of priors: informative prior elicitation, non–informative prior distribution and conjugate prior distribution. Informative prior elicitation are prior beliefs that substantially modify the information of the sample, not only by its mean but also its variance. Non–informative prior distribution, are more vague and are modeled often through uniform distribution. Conjugate prior distribution, guarantees instead to align the class of the posterior distribution with the prior distribution. Generally, the prior used to optimize the portfolio, are non–informative, that reduces estimation risk, but the method does not reduce the estimation error [6]. While, by using informative priors should be possible to structure a portfolio with superior returns that reduce also the estimation error. Thereby, in the following, we are going through – a so called – 'empirical Bayes approach to the efficient portfolio selection' [6]. The objective of the method is to maximize the following likelihood function by determining the prior parameters values ν (of the variance) and τ (of the return), given $\mathbf{S}$, $\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Omega}}^{-1}$ and $\hat{\boldsymbol{\mu}}_0$:

$$
\begin{aligned}
g\left(\mathbf{S}, \hat{\boldsymbol{\mu}} \mid \hat{\boldsymbol{\Omega}}^{-1}, \hat{\boldsymbol{\mu}}_0, \nu, \tau\right) = \\
\int_{\boldsymbol{\mu}} \int_{\boldsymbol{\Lambda}} f\left(\mathbf{S}, \hat{\boldsymbol{\mu}}, \boldsymbol{\Lambda}, \boldsymbol{\mu} \mid \boldsymbol{\Omega}^{-1}, \boldsymbol{\mu}_0, \tau, \nu\right) d\boldsymbol{\Lambda} d\boldsymbol{\mu} \\
\int_{\boldsymbol{\mu}} \int_{\boldsymbol{\Lambda}} f_{nw}\left(\mathbf{S}, \hat{\boldsymbol{\mu}}, \mid \boldsymbol{\Lambda}, \boldsymbol{\mu}\right) \cdot \\
\cdot f_{nw}\left(\boldsymbol{\Lambda}, \boldsymbol{\mu} \mid \hat{\boldsymbol{\Omega}}, \boldsymbol{\mu}_0, \nu, \tau\right)
\end{aligned}
\tag{3}
$$

Where the matrix $\mathbf{S}$ represents the sum of the squared errors of the estimated covariance matrix, $\hat{\boldsymbol{\Omega}}^{-1}$ is the prior inverse covariance matrix and $\hat{\boldsymbol{\mu}}_0$ is the prior return vector. $\boldsymbol{\Lambda}$ is defined as the inverse of the cov–matrix.
Then the estimation of $\hat{\nu}$ and $\hat{\tau}$ determine the posterior covariance matrix of the returns $\boldsymbol{\Sigma''}$ and the posterior returns $\boldsymbol{\mu''}$ through:

$$
\begin{aligned}
\boldsymbol{\mu''} &= \boldsymbol{\mu}_0 \left(\boldsymbol{\omega}_\tau\right) + \hat{\boldsymbol{\mu}} \left(1 - \boldsymbol{\omega}_\tau\right) \\
\boldsymbol{\Sigma''} &= \left[\frac{\nu + T}{\nu + T - 2}\right] \left[1 + \frac{1}{\tau + T}\right] \left[\boldsymbol{\Omega}\boldsymbol{\omega}_\nu + \hat{\boldsymbol{\Sigma}} \left(1 - \boldsymbol{\omega}_\nu\right) + \right. \\
&\quad \left. + \boldsymbol{\omega}_\tau \left(\frac{T}{\nu + T}\right) (\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}_0)(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}_0)'\right],
\end{aligned}
\tag{4}
$$

Where

$$
\begin{aligned}
\boldsymbol{\omega}_\tau &= \frac{\tau}{(\tau + T)} \\
\boldsymbol{\omega}_\nu &= \frac{\nu}{(\nu + T)}
\end{aligned}
\tag{5}
$$

Where T is the amount of sample information, thereby it is possible to find an equilibrium for every given T. The weight strengths are defined by the prior parameters τ and ν, whereby high weights increase the information shock of the views on the estimated $\boldsymbol{\mu''}$ and $\boldsymbol{\Sigma''}$. If we now compute an optimal

portfolio from the posterior return vector and the posterior covariance matrix, it will be referred as the 'Empirical Bayes investment rule' [6]. As anticipated before [6] it is proven in over 50 portfolio construction that the method allows superior returns compared to the Bayes Diffuse or the classical method.

To notice regarding the method, is that the Black and Litterman model [7] is often described as Bayesian but the authors do not elaborate on the connection with Bayesian statistics.

The last problem is the estimation of μ, it has been long assumed that the maximum likelihood estimator (MLE) was optimal, Stein however demonstrated that it was inadmissible for p $\geq$ 3 [8]. The so called James-Stein Estimator has a smaller risk than the MLE and is defined as follow. Supposed $\boldsymbol{\theta}$ is the unknown mean of a p-variate normally distributed random variable $\mathbf{Y}$. If σ^2 is known, the estimator is defined as:

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{JS}} = (1 - \frac{(m-2)\sigma^2}{||y||^2} y \tag{6}$$

If p $\geq$ 3, the James-Stein estimator achieves a lower mean squared error than the MLE [9].

II. Financial Markets

Being a social science, finance, does not have constants and true parameters. There is no such a thing as a constant speed of light or an universal formula such as $e = mv^2$. The parameters in finance are based on an estimation of others people estimations, which change through time based on new relevant signals. Some signals may be easier to insert in the estimation equation such as the disclosure of the dividends of a firm, which are numerical. Some signals, on the contrary, are qualitative, based for example on politics and countries relations. Some signals may be imperfect because of media reporting, noise or simply agent–firm information asymmetry, those are more difficult to quantify.

The stock price is based on the discounted estimated future dividends. A rational investor, as said above, does not know its true values, but formulates an expectation about it. The expectation changes continuously with every new signal. The investor takes into consideration the new signals and reassesses its expectations. The learning process is based on the Bayes' rule [10]. The rule states that before valuing any signal of a parameter (δ), an investor considers the parameter as normally distributed with mean δ_0 and variance σ_0^2. The posterior beliefs about δ is still normally distributed but with a mean $\tilde{\delta}_T$ and variance $\tilde{\sigma}_T^2$ where:

$$\tilde{\delta}_T = \delta_0 \frac{\frac{1}{\sigma_0^2}}{\frac{1}{\sigma_0^2} + \frac{T}{\sigma^2}} + \bar{s} \frac{\frac{T}{\sigma^2}}{\frac{1}{\sigma_0^2} + \frac{T}{\sigma^2}} \tag{7}$$

$$\tilde{\sigma}_T^2 = \frac{1}{\frac{1}{\sigma_0^2} + \frac{T}{\sigma^2}} \tag{8}$$

where $\bar{s}$ is the arithmetic mean of the signal and T the number of independent signals about δ. As the formula shows, the variance does not depend on the realization of the signal, but on T, which decrease the uncertainty and therefore the variance of δ with every new relevant signal emitted.[1] Trough the perspective of this rule it is possible to explain a number of different markets behavior, such as young firms exhibiting high volatility and high valuation, the presence of 'bubbles', which is typically attributed to market irrationality or mispricing, fund flows as reaction of fund performance and, of particular interest for the purpose of the class project, the return predictability [10].

Before we can talk about returns and volatility predictability we have to take a look at stock valuation. If we price a stock with the well-known Gordon growth formula, the price of a stock is given by:

$$P = \frac{D}{r - g} \tag{9}$$

Where the discount rate r and the dividend growth rate g are constant. D is the next period's dividend of the firm. Investors, however, do not know the real value of g, since it can only be learned through time.[2] Uncertainty about g increases the stock price as [11] and [12] claim. The two papers argue that the uncertainty about g makes the distribution of future dividends right-skewed, therefore increasing the expectation of discounted future dividends. From this theory we can also understand why young firms, compared to more mature firms, have higher valuations. For example, the g of a new start–up has high uncertainty because the firm can potentially become the next Nestlé (there is almost unlimited upside potential) or declare bankruptcy in a few periods (maximum loss 100%). Instead, a mature firm already gave major information about g through its lifetime, the uncertainty about g is much lower. The firm has a lower probability to change position in the market and become the leader of its industry or declare bankruptcy in the near future, because its market is already defined. In this case the market values the young firm higher than the mature firm (ceteris paribus), because the chances that in the future the start–up becomes leader in its industry is comparatively higher than a more mature firm.

In theory the volatility of the stock return for known and constant r and g is the same as the volatility of the dividend growth rate. However, in reality, the two volatility do not match. In fact, return volatility are more than three times higher than dividend growth volatility. The reason is explained by [13], who starts by taking g as unknown, as it is in reality. Investors learn about g over time – for example – trough the disclosure of the annual dividend. An increase of the dividend pushes the price up. The positive price shock is not only given by the reassessment of the current dividend but also by higher

[1] Later it will be shown that it is not always the case in the real world.

[2] Every time the firm discloses the dividends of the following period, the investor can compute the past dividend growth and reassess the expectation of the future. The investor, however, does not know the dividend of the whole lifetime of the firm. See also Bayes' rule.

dividends expectation in the future. It means that both D and g increase giving to the stock price a – so known – double kick to the price. Formally the double kick is explained as follows [13]:

$$Vol = DGV \cdot \left[1 + \left(\frac{\partial log(P/D)_t}{\partial \tilde{g}_t}\right) m_t\right] \tag{10}$$

Where Vol is the return volatility and DGV is the dividend growth volatility. As said before the uncertainty about g decrease over time, meaning that the volatility of the return decrease as well (see Bayes' rule).

The negative relationship of T and the volatility of the stock return however, is not valid in reality. In the market we can observe extended periods of sustained high as well as low volatility in the stock return [10]. The models of [11] and [13] do not consider this possibility since they assume g to be constant. But if, as in reality, we consider the possibility of unobservable regime shifts, the uncertainty about g can fluctuate rather than converge deterministically to a constant. In fact, if we consider a signal that dramatically changes the market situation (a so called regime shift) and the dividend growth realization is far from its current estimation, then the posterior expectation about g will be higher than before, because past data becomes less useful for forecasting since it is from a different regime. The regime shifting not only increases the variance but also the equilibrium discount rate, doubling the shock [10].

Now that we had a look at stock pricing through parameters uncertainty and learning process through signals, we can dive in to the prediction of volatility and returns. [13] shows that returns are predictable. He proves empirically that when the aggregate P/D ratio is low, future returns are high. However, it is also specified that the learning–induced predictability is only possible ex–post. In fact, [14] distinguish between the forecasting possibility of 'econometricians' and 'real–time investors'. The latter sees the underling parameters as random and unstable over time as new information are absorbed in the learning process. While, for an econometrician the data–generating process does not contains any random parameters. The presented model [14] proves that returns are effectively predictable by looking at past returns, as [13] does,[3] real–investors however, only learn the new parameters at the moment of disclosure, therefore they are at the mercy of the randomness of the signals.

III. Machine Learning

It's been known for long that simple multilayer feedforward neural networks (FFN) are universal function approximators [15]. However, a shortcoming of FFN is that it builds new features based on the whole input (in a sense, it has too many degree of freedom). When the input is a time-varying sequence of events, we would like to force our network to scan the input sequentially to learn features from past events, and how these are correlated with future ones. Recurrent neural networks (RNN) are especially designed to deal with sequences, and they do it through *parameter sharing* across several time steps in the input. In this way, we do no not have to learn a separate set of feature for events that occur at different position in the input sequence [16]. For instance suppose that we want to detect positive and negative divergence for a given asset, in time window of τ days. Thanks to parameter sharing, an RNN will extract this information independently of the fact that the divergence point is at the i-th or j-th input, where $1 \leq i, j \leq \tau$. In a RNN, the hidden units represent the state of the dynamical system it models, and therefore can be written as:

$$\boldsymbol{h}^{(t)} = f(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^t, \boldsymbol{\theta}) \tag{11}$$

where f is the function that the network aims to learn, $\boldsymbol{x}^t$ is the input at time t and $\boldsymbol{\theta}$ represents the parametrization of the function f. In a standard RNN, we represent the dynamical system with three sets of weights: $\boldsymbol{W}$ (hidden-to-hidden connections), $\boldsymbol{V}$ (hidden-to-output) and $\boldsymbol{U}$ (input-to-hidden), and we compute the following quantities to update the state and computing the output:

$$\begin{aligned} \boldsymbol{h}^{(t)} &= \sigma(\boldsymbol{W}\boldsymbol{h}^{(t)} + \boldsymbol{U}\boldsymbol{x}^{(t)} + \boldsymbol{b}) \\ \boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \end{aligned} \tag{12}$$

where $\boldsymbol{b}, \boldsymbol{c}$ are the bias vectors, and σ is a non-linear activation function. Interestingly, the final state $\boldsymbol{h}^{(\tau)}$ represents a "lossy" version of the encoded sequence (this is due to parameter sharing, as the weights $\boldsymbol{W}$ and $\boldsymbol{U}$ are sequentially applied to the whole input sequence). Unfortunately, standard RNN suffer from the famous "Long Term Dependencies" problem [17], which cause the fact that is hard to optimally train an RNN through back-propagation (or its variant back-propagation through time BPTT). The second problem, linked to parameter sharing and the training procedure (BPTT), is the exploding-vanishing gradients problem. If I have a long sequence (large τ), then if the largest singular value σ_{max} of the weight matrix $\boldsymbol{W}$ is greater than one, then the repeated application of $\boldsymbol{W}$ cause the norm of the gradients to explode. That's why we chose the LSTM, a variant of standard RNN that aims at solving the mentioned problems. The main idea behind a LSTM cell is that we want to allow the information to flow unchanged through the cell state. This is done through the usage of *gates*. Note that we have two state for the LSTM: the cell state and the h-state. In detail, when the input at time

[3]The models used in the two papers are slightly different but they relay on the same principles.

t is processed, we compute the following:

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma(\boldsymbol{x}_t \boldsymbol{U}^i + \boldsymbol{h}_{t-1} \boldsymbol{W}^i) \\
\boldsymbol{f}_t &= \sigma(\boldsymbol{x}_t \boldsymbol{U}^f + \boldsymbol{h}_{t-1} \boldsymbol{W}^f) \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{x}_t \boldsymbol{U}^o + \boldsymbol{h}_{t-1} \boldsymbol{W}^o) \\
\tilde{\mathbf{C}}_t &= \tanh(\boldsymbol{x}_t \boldsymbol{U}^g + \boldsymbol{h}_{t-1} \boldsymbol{W}^g) \\
\boldsymbol{C}_t &= \sigma(\boldsymbol{f}_t * \boldsymbol{C}_{t-1} + \boldsymbol{i}_t * \tilde{\boldsymbol{C}}_t) \\
\boldsymbol{h}_t &= \tanh(\boldsymbol{C}_t) * \boldsymbol{o}_t
\end{aligned}$$

Where here σ is the sigmoid function and $*$ is the element-wise multiplication. We can recognize three gates: $\boldsymbol{i}_t$ is called the input layer, $\boldsymbol{f}_t$ is the forget layer and $\boldsymbol{o}_t$ the output layer. Note that the gates are all in the same form and the sigmoid function "squashes" the values of the input vector between 0 and 1. As the gates appear only as matrix multiplication in the forward pass, the gates define what information should pass and what should not. Note that the cell state $\boldsymbol{C}_t$ depends on the forget layer multiplied by the state at time $t-1$ (intuitively this parts decides which parts of the cell state to forget). It also depends on the input layer $\boldsymbol{i}_t$ times a function of the input $\boldsymbol{x}_t$ and and the h-state at $t-1$. Therefore this factor decide which parts of the input $\boldsymbol{x}_t$ is relevant and therefore which information to keep in the cell state. Finally, note that the LSTM has many more weights to optimize than standard RNN. In particular, in LSTM we have 4 sets of weights in the input-hidden connections (denoted by $\boldsymbol{U}$) and 4 sets of weights in the hidden-hidden connections (denoted by $\boldsymbol{W}$). LSTM will be at the core of our model to make stock prices predictions.

A. Non-negative matrix factorization (NNMF)

The NNMF is a matrix factorization technique that is particularly useful to compress non-negative data (like images or stock prices). Given a non-negative data matrix $\boldsymbol{X} \in \mathbb{R}^{nxp}$, we look for two matrices $\boldsymbol{A}$ (compressed representation, or *scores*) and $\boldsymbol{H}$ (the *atoms* or dictionary) that are optimal to the following optimization problem:

$$\arg\min_{\boldsymbol{A} \in \mathbb{R}^{nxq}, \boldsymbol{H} \in \mathbb{R}^{qxp}} \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{H}\|_2^2 \text{ under constraints } \boldsymbol{A} \geq 0, \boldsymbol{H} \geq 0$$

where the symbol $\geq$ is intended element-wise. Intuitively we want to minimize the squared reconstruction error under the non-negative constraints. If we add a L_1 penalty term on the dictionary matrix $\boldsymbol{H}$ we can impose sparsity on $\boldsymbol{H}$, and therefore only few atoms will be involved in the lossy reconstruction of $\boldsymbol{X}$. In the case that $\boldsymbol{X}$ is a matrix containing n daily observations of p stock prices, we can interpret the matrix $\boldsymbol{A}$ as the matrix containing the $q << p$ main latent trends that describe the original p stocks. Furthermore, every reconstructed asset (i.e. every column of $\boldsymbol{AH}$) is a linear combination of the latent trends identified. Here we follow the idea of [18] and we cluster the stocks based on the highest score for that stock. In other words, considering the component matrix $\boldsymbol{H}$, we select for each column i, the highest value, which represents the main trend that composes the i-th stock. We will explain better the implementation in section IV.

IV. Experiments

A. Approach

Our approach consists of two different parts: first of all we use an LSTM based neural network to learn the stock prices for a fixed future time window [19]. Then, on a separate dataset, we use the trained model to make predictions on new data points, with these predictions we compute the vector of expected returns and the sample covariance matrix. Then we optimize the portfolio weights using the quadratic optimization problem described in Markowitz portfolio selection theory (1). We keep the portfolio weights for a fixed given period p. Furthermore, we re-optimize the network parameters with given frequency (which is also a hyperparameter of the model). We do this in order to consider new trading days that have passed. The algorithm is summarized in Alg. 1.

Algorithm 1: Portfolio optimization procedure

```
input : train_data, test_data, input_window,
        output_window
model = LSTM(input_window, output_window)
model = model.train(train_data)
for i, entry in enumerate(test_data) do
    predictions=model.predict(entry)
    exp_returns, cov_matrix =
      get_estimations(predictions)
    portfolio_weights=optimize(exp_returns, cov_matrix)
    if optimize_again(i) then
        model=model.train([train_data, test_data[:i])
    end
end
```

Finally, we do portfolio weights analysis. We measure portfolio diversification using non-negative matrix factorization to capture the main market latent trends and assigning each stock to one of these latent trend. This part is explained in detail in subsection III-A.

B. Model

We built a neural network with one LSTM layer that extracts sequential features from the data. We fix the number of hidden recurrent units to 200. After that we extract the output of the full sequence, reshape it to a tensor of shape [batch_size, timesteps$\cdot$hidden_units] where timesteps is length of the sequence (i.e. the input window) and we use a dense layer to combine all the LSTM outputs and map them to a vector of shape [batch_size, output_window $\cdot$ n_stocks]. Then we reshape this tensor to match the label's shape. We used the mean squared error as loss function. During training, we additionally use a dropout layer with dropout probability of

0.5 on the LSTM outputs to improve generalization. We used the Adam optimizer with fixed learning rate of 0.00002.

C. Data

Due to the complexity and degrees of freedom of the neural network, we first split the dataset in in two parts: with the first part (training set) we train the LSTM model, and with the second part (test set) we optimize the portfolio weights. With the train set, we compute the train samples by applying a rolling window with window size of 100 days. For the labels, we apply a shifted rolling window of size output_window. We will consider output_window a hyperparameter to tune. Therefore our LSTM model is trained by considering the past 100 days and predicting the stock prices for a time window of output_window.

D. Evaluation Metrics

To run our experiment we used logarithmic returns, and to annualize the daily returns and the daily volatility we used the following equations:

$$r_{pf} = \frac{252}{n} \cdot \sum_{d=1}^{n} log(r_d + 1) \tag{13}$$

$$\sigma_{pf} = \sqrt{252} \cdot std(log(r_d + 1)) \tag{14}$$

where:
n = number of trading days in the considered window
$252 =$ number of trading days in a year
r_d = daily returns vector
std = function that takes a vector in input and computes the standard deviation
Note the the logarithm function is applied element-wise to the vector.
To optimize our portfolio we used a modified Sharpe Ratio (SR_m), as we do not consider the risk free rate in our investment universe but only a collection on 14 different stocks.

$$SR_m = \frac{r_{pf}}{\sigma_{pf}} \tag{15}$$

E. Results

We ran the following experiments (we keep the frequency of portfolio weights optimization constant to 30 days):

1) We trained the LSTM with output window of 25 days, computing the expected returns based on the LSTM predictions and the covariance matrix based on the past daily prices for 100 days. Furthermore we never re-optimize the LSTM. The resulting annualized return was 0.1886, the annualized volatility 0.1722 and the modified SR was 1.096.
2) We trained the LSTM with different output windows that vary from $[10, 15, 20, 25, 30, 35]$ days, computing both the expected returns and the the covariance matrix based on the LSTM predictions. Furthermore we re-optimize the LSTM every 120 days. The results in terms of the chosen evaluation metrics are shown in Table I. Interestingly by increasing the size of the time window predicted by the LSTM, the annualized volatility decreases. At first this can be thought as counter-intuitive, as a predictions are more uncertain if future time period is larger. However, from a numerical and statistical perspective, a larger time window guarantees a better (stable) estimate of the expected returns and sample covariance matrix (this can be explained by the law of large numbers). Furthermore we also argue that as we trained a relatively large model (200 hidden units), the network actually predicts with more or less the same accuracy the different time windows.
3) We trained the LSTM as in experiment 2 but we re-optimize the returns every 120 days and we modify the covariance matrix by adding 0.00001 on the diagonal. The results are shown in Table III.
4) To reduce uncertainty in estimation, similar to what described in the first section (4), we assume a prior on both the expected returns and covariance matrix. We compute such priors with a time window of the 40 latest daily observations. Then we compute the new candidate expected returns using the predictions made by the LSTM. We used an output time window of 15 and 20 days to train the LSTM and re-optimize the LSTM every 120 days. Then we combine the prior believes with the predictions using a simple weighted average (we assign a weight of 0.6 for the prior). The results are shown in IV.

TABLE I
LSTM MARKOWITZ MV PORTFOLIO OPTIMIZATION

Output window	Annualized returns	Annualized volatility	Modified Sharpe ratio
10	-0.0695	0.4678	-0.1487
15	0.1915	0.3128	0.6125
20	0.1108	0.2241	0.4944
25	0.0967	0.1335	0.7244
30	0.1019	0.1429	0.7130
35	**0.1072**	**0.0893**	**1.2002**

We compare the results of our experiments with the baseline built in the following way: we use Markowitz optimization portfolio theory (1), where both the expected returns and the covariance matrix are computed with a time windows of the past 100 days. As we can see from Table I and II the LSTM Markowitz MV portfolio optimization performs better for each output windows than the standard model, as measured for the modified Sharpe Ratio. Also the the LSTM without re-optimization has a higher modified SR than the baseline for the same output window. We can further observe that the an annualized volatility decreases also for the baseline with higher output window. It is interesting to notice that the baseline has a much higher Δ between the smallest an the

TABLE II
STANDARD MARKOWITZ MV PORTFOLIO OPTIMIZATION

Output window	Annualized returns	Annualized volatility	Modified Sharpe ratio
10	-0.2092	0.3828	-0.5466
15	-0.2075	0.3132	-0.6627
20	-0.0376	0.2510	-0.1500
25	0.1223	0.2266	0.5400
30	0.1357	0.2126	0.6384
35	**0.1798**	**0.2080**	**0.8642**

highest return than the LSTM, this shows us that the LSTM model is much less sensitive to the output window, an thus a more reliable method.

TABLE III
LSTM MARKOWITZ MV PORTFOLIO OPTIMIZATION WITH MODIFIED COVARIANCE MATRIX

Output window	Annualized returns	Annualized volatility	Modified Sharpe ratio
10	0.2634	0.3445	0.7646
15	0.0192	0.2473	0.0778
20	**0.1836**	**0.1861**	**0.9869**

Table III shows the results of the third experiment. The shrinkage, through the constant modification on the diagonal of the covariance matrix, helps to find the optimal portfolio.[4] This is because the number of observations is low compared to the number of stock analyzed, resulting in a ill-conditioned inverse covariance matrix [20].

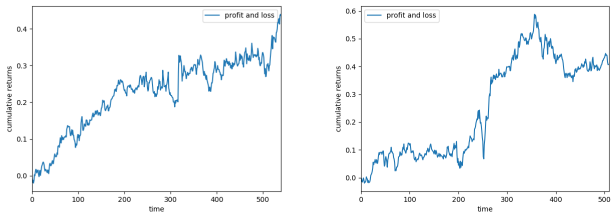
TABLE IV
LSTM MARKOWITZ MV PORTFOLIO OPTIMIZATION WITH BAYESIAN APPROACH FOR COVARIANCE MATRIX

Output window	Annualized returns	Annualized volatility	Modified Sharpe ratio
15	0.1175	0.1581	0.7432
20	**0.1143**	**0.0932**	**1.2262**

Table IV shows the results of the LSTM using Bayesian approach, we can see that with this method we reached the highest RS of all our experiments. Note that that prior has the effect of reducing uncertainties in estimation, and therefore the optimized portfolio has a lower annualized volatility compared to the previous experiments (considering the same output window).

[4]The ad-hoc shrinkage method used requires prior information to work better.

Fig. 3. Cumulative return



In Fig. 3. we can see the cumulative return of two of our experiments, the one on the left hand side has been achieved with the LSTM model of experiment 1, while the one on the right with the standard Markowitz optimization. We can clearly see that the LSTM has a smoother upwards path.

F. Portfolio analysis and NNMF

As we explained in section I, diversification has a major importance in portfolio construction. The portfolio variance decreases till the systematic risk, diversifying the idiosyncratic. We found it interesting that Markowitz optimize its portfolios considering the covariance matrix without necessarily diversifying over multiple stocks. Diversification helps decreasing the risk of the portfolio of stocks when the correlation is different than one. Thus, corner solutions do exist in optimized portfolios [21]. In order to verify not only the diversification of the stocks by plotting the weights of the stocks in the optimal portfolios, but also if the optimal portfolio allocates weights to the different latent trends, we use the aformentioned NNMF technique as measure of diversification.

In detail, we first scale the stock data such that all the values are between 0 and 1 (min-max scaling). Then, for each set of weights, we apply Alg. 2. In the first step, we get the main latent trend for each stock (i.e. the trend with highest coefficient for all the stocks). We call this coefficient vector ψ. Then for each latent trend, we sum all the weights of stocks that have that trend as the main one. As a result we have portfolios composed only by the three major trends.
Fig. 4. shows the optimal portfolio weights composed of the three latent trends of experiment one, while Fig. 5. is the same plot but for the second experiment. A negative weight means a short selling of the asset.

A diversified portfolio will have allocated similar weights (positive or negative) to different latent trends. We see that Fig. 4. has different periods where the diversification is low, such as period 5, 6, 11, 12 and 17. While, we can recognize maybe only two periods with low latent trend diversification in experiment two: period 6 and 8.
From those plots we cannot identify if the portfolios contain corner solutions, because a high weight on one latent trend could be the result of a diversification of stocks within the

Fig. 4. Experiment one, the high risk portfolios.

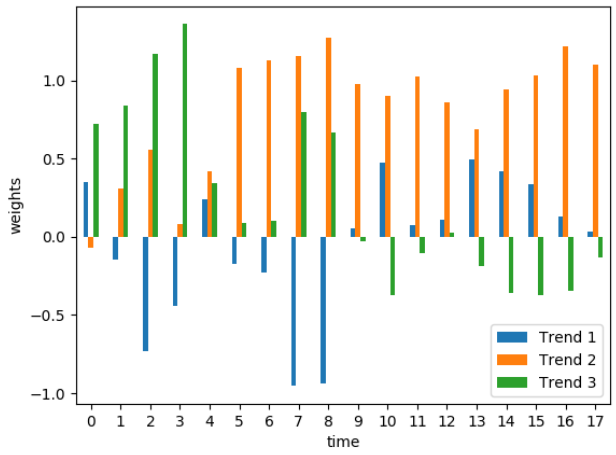
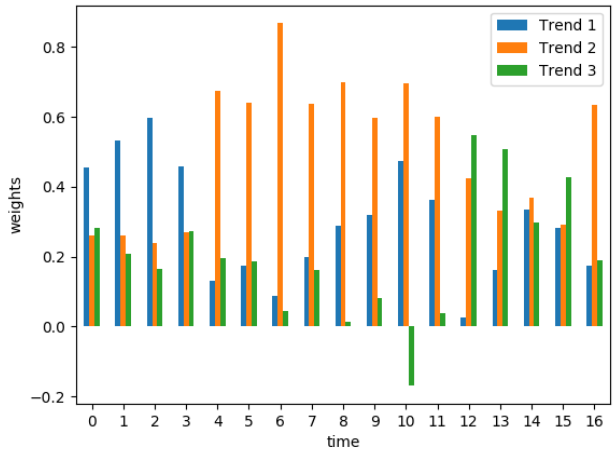


Fig. 5. Experiment two with output window of 35 days, the low risk portfolios.



same main latent trend, which we will not be able to identify. What we can see, is that the position on the trend do not change massively during time. For example, we do not see massive changes in the weight of latent trend 2 in the first as in the second experiment. While massive changes of latent trend weights are present between the two experiments, as we can see latent trend one is primarily negative in the first periods for experiment one. In experiment two, instead, the latent trend one is the biggest long position in the first four periods.

The interpretation is that **the latent trends effectively represent the main trends of the market**. In fact experiment 1 has an annualized volatility of 0.1722 and we have reported a not well diversified allocation among the latent trend. On the contrary the portfolio weights analysis conducted for experiment 2 (Fig. 5.) has an annualized volatility of 0.0893 and present a much more diversified allocation of weights among the three latent trends. Again, this suggests that the latent trends are correctly captured by the NNMF.

Algorithm 2: Latent trends assignments

input : latent trends A, dictionary H, and portoflio weights ω

ψ =np.argmax(H, axis=0) #get main trend for each stock

allocations = [0,0,0]

for *i, weight in enumerate(ω)* **do**

  trend=ψ[i]

  allocations[trend] += weight

end

V. Conclusion

We have shown that for the given data our LSTM model outperforms classic Markowitz portfolio weights optimized only with the past data. The interpretation that we give is that past data alone are not always representative of future returns, and it is trivial to say that meaningful and accurate predictions of stock prices can be much more powerful. On the other hand, from an estimation point of view, we have found that the LSTM neural network is highly sensitive to hyperparameters, like learning rate, input window and output window, number of hidden recurrent units. Therefore the optimization process is more troublesome and requires much more data than classic machine learning techniques. Furthermore, we have shown that NNMF can be used as a measure of diversification of our portfolio. This claim is justified by the fact that high risk portfolios present weights that are not well allocated among the various latent trends, while low-risk portfolios have weights that are much better diversified among the latent trends.

Computation power limits did not let us experiment even larger output time windows. Future works might test how large the prediction time window can be accurately predicted by the LSTM (we think that at an extremely large output window the predictions will not be accurate anymore and the risk will stop decreasing or the expected returns will become unreasonably low). Furthermore, it would be interesting to run our LSTM experiment in a bear market to stress–test the reliance of the method. About the NNMF, we only use it as evaluation metric of the portfolio weights. It would be interesting to try to integrate the latent trends in the portfolio weights choice (i.e. integrate it in the Markowitz portflio optimization framework).

Note

Code is submitted in a separate zip file.

REFERENCES

[1] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[2] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[3] V. K. Chopra and W. T. Ziemba, "The effect of errors in means, variance, and covariance on optimal portfolio choice," *Journal of Portfolio Management*, vol. 19, no. 2, pp. 6–11, 1993.

[4] P. Jorion, "Portfolio optimization in practice," *Financial Alalysts Journal*, vol. 48, no. 1, pp. 68–74, 1992.

[5] A. Bade, G. Frahm, and U. Jaekel, "A general approach to bayesian portfolio optimization," *Mathematical Methods of Operations Research*, vol. 70, no. 2, p. 337, 2009.

[6] P. A. Frost and J. E. Savarino, "An empirical bayes approach to efficient portfolio selection," *Journal of Financial and Quantitative Analysis*, vol. 21, no. 3, pp. 293–305, 1986.

[7] F. Black and R. Litterman, "Global portfolio optimization," *Financial analysts journal*, vol. 48, no. 5, pp. 28–43, 1992.

[8] C. Stein, "Inadmissibility of the usual estimator for the mean of a multivariate normal distribution," *Conference paper*, 1956.

[9] W. James and C. Stein, "Estimation with quadratic loss," *Matematical Reviews*, vol. 1, p. 361379, 1961.

[10] L. Pástor and V. Pietro, "Learning in financial markets," *Annu. Rev. Financ. Econ.*, vol. 1, no. 1, pp. 361–381, 2009.

[11] ——, "Stock valuation and learning about profitability," *The Journal of Finance*, vol. 58, no. 5, pp. 1749–1789, 2003.

[12] L. Pástor and P. Veronesi, "Was there a nasdaq bubble in the late 1990s?" *Journal of Financial Economics*, vol. 81, no. 1, pp. 61–100, 2006.

[13] A. G. Timmermann, "How learning in financial markets generates excess volatility and predictability in stock prices," *The Quarterly Journal of Economics*, vol. 108, no. 4, pp. 1135–1145, 1993.

[14] J. Lewellen and J. Shanken, "Learning, asset-pricing tests, and market efficiency," *The Journal of finance*, vol. 57, no. 3, pp. 1113–1145, 2002.

[15] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359 – 366, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0893608089900208

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[17] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *Trans. Neur. Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994. [Online]. Available: http://dx.doi.org/10.1109/72.279181

[18] A. Pazienza, S. Francesca Pellegrino, S. Ferilli, and F. Esposito, "Clustering underlying stock trends via non-negative matrix factorization," 09 2016.

[19] S. Obeidat, "Adaptive portfolio asset allocation optimization with deep learning," *International Journal on Advances in Intelligent Systems*, vol. 11 no 1 & 2, 2018.

[20] O. Ledoit and M. Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *Journal of multivariate analysis*, vol. 88, no. 2, pp. 365–411, 2004.

[21] G. M. De Athayde and R. G. Flôres Jr, "Finding a maximum skewness portfolioa general solution to three-moments portfolio choice," *Journal of Economic Dynamics and Control*, vol. 28, no. 7, pp. 1335–1352, 2004.