Annals of Operations Research (2025) 346:319–340
https://doi.org/10.1007/s10479-024-06257-1

ORIGINAL - SURVEY OR EXPOSITION

Enhancing Markowitz's portfolio selection paradigm with machine learning

Marcos López de Prado[1,2] · Joseph Simonian[3,4] · Francesco A. Fabozzi[5] · Frank J. Fabozzi[6]

Received: 19 June 2024 / Accepted: 29 August 2024 / Published online: 7 October 2024

Abstract
In this paper we describe the integration of machine learning (ML) techniques into the framework of Markowitz's portfolio selection and show how they can help advance the robust mathematical strategies necessary for modern financial markets. By combining traditional econometrics with cutting-edge ML methodologies, we show how to enhance portfolio management processes including alpha generation, risk management, and optimization of risk metrics like conditional value at risk. ML's capacity to handle vast and complex datasets allows for more dynamic and informed decision-making in portfolio construction. Moreover, we discuss the practical applications of these techniques in real-world portfolio management, highlighting both the potential enhancements and the challenges faced by portfolio managers in implementing ML strategies.

Keywords Machine learning · Signal generation · Feature selection · Portfolio optimization · Generative Language models · Natural language processing

JEL Classification C6 · G11

✉ Frank J. Fabozzi
ffabozz1@jhu.edu

Marcos López de Prado
ml863@cornell.edu

Joseph Simonian
jsimonian@autoinvestec.com

Francesco A. Fabozzi
francesco.fabozzi@yale.edu

1 Abu Dhabi Investment Authority (ADIA), Abu Dhabi, UAE

2 Cornell University, New York, NY, USA

3 Autonomous Investment Technologies, Newton, MA, USA

4 FDP Institute, New York, NY, USA

5 Yale's International Center for Finance, New Haven, CT, USA

6 Carey Business School, Johns Hopkins University, Baltimore, MD, USA

1 Introduction

Harry Markowitz is among the individuals responsible for providing a rigorous quantitative foundation for investment management in the 20th century. By marrying portfolio selection to optimization and statistical theory, Markowitz provided a new path forward for investment researchers seeking a robust mathematical approach to navigating financial markets. His work ultimately became one of the fundamental pillars of modern portfolio theory. The complexity of today's markets, with a seemingly infinite amount of data being generated at rapid speeds, has required investors to look for new mathematical arrows to deploy in their methodological quivers. Indeed, investors have recognized for some time that the methods of data science and machine learning (ML) can be a valuable addition to their investment process (López de Prado, 2022).

ML has become a fundamental tool in modern portfolio management by significantly enhancing traditional approaches to alpha generation and risk management that previously relied solely on econometric models − regression models, time-series models (autoregressive integrated moving average (ARIMA) (Box & Jenkins, 1976), generalized autoregressive conditional heteroscedasticity (GARCH) (Bollerslev, 1986), vector autoregression models (VAR) (Sims (1972, 1980), and cointegration models. ML provides portfolio and risk managers with more sophisticated ways to build frameworks for trading, portfolio construction, stress testing, scenario analysis, and the optimization of risk metrics like conditional value at risk (CVaR).[1]

ML originates from what Breiman (2001a) calls the "algorithmic modeling culture" while traditional econometrics and statistics come from the "data modeling culture". The data modeling culture has hitherto been the dominant culture in traditional statistical practice and is primarily concerned with providing information about the relationships that exist between input and response variables and assumes that the data are generated by a specific stochastic process. In contrast, the algorithmic modeling culture assumes that the relationships between input and response variables are essentially too complex to uncover, and consequently is primarily concerned with devising methods to successfully predict responses from inputs. ML thus provides a complimentary set of analytic tools for investment practitioners.

What is ML? Simply put, it is a field of study that combines the use of statistics and computing to discover or impose order in complex data to enhance informed decision-making. It is an inherently practical endeavor, just like finance, and so it is especially suited to investment applications. ML comprises a family of computational techniques that facilitate the automated learning of patterns and the formation of predictions from data. Although there are many types of ML frameworks and algorithms, they share the following three components: (1) a method for extracting and representing the essential features of the data under consideration; (2) a process and period of model training; and (3) an objective function derived ("learned") during the training period and applied to a test dataset (not used during training). Furthermore, ML algorithms are generally designed to solve one of two types of problems: a classification-type problem, in which the goal is to categorize data into different types, or a regression-type problem, in which the goal is to predict a quantity for a variable given the values for a set of predictor variables. Both types of problems are prevalent in finance, so ML can be viewed as a natural extension to investment practitioners' existing tool set.

The objective of this paper is to provide an overview of how practitioners are using ML in portfolio and risk management including signal generation, feature selection, portfolio

[1] See Simonian et. al (2018) and Simonian and Fabozzi (2019) for detailed arguments regarding the added value of machine learning relative to traditional econometrics.

construction and optimization, and stress testing. We conclude this survey with the challenges portfolio managers face in applying ML.

2 Signal generation

An investment signal is a piece of information or set of data points that can potentially inform an investment decision. Investment signals can be derived from a variety of sources, such as econometric or statistical models, ML models, news articles, company filings with the Securities and Exchange Commission (SEC), economic indicators, and market sentiment. Generally, signals can be classified as quantitative or qualitative. Data for quantitative signals include historical price movements, economic indicators, and company financial performance measures. Qualitative signals include news events, management discussions in 10-K filings, and market sentiment. In assessing qualitative signals, it is necessary to interpret and judge the meaning of the signal. For example, in the case of market sentiment as determined by a natural language processing (NLP) algorithm, the input is either positive, negative, or neutral.

The identification of signals is central to active portfolio management. Econometric tools have long been used for that purpose. They include regression analysis (linear and logistic regression analysis) and time-series analysis (e.g., ARIMA). ML methods for signal generation include (1) supervised learning algorithms (e.g., support vector machines,[2] random forecasts,[3] (2) unsupervised learning algorithms (e.g., clustering methods such as k-means[4] and hierarchical clustering[5]), (3) reinforcement learning[6] (e.g., Q-learning,[7] SARSA,[8] deep Q-networks,[9] and policy gradient methods[10]), (4) neural networks and deep learning algorithms (e.g., convolution neural networks and autoencoders,[11] recurrent neural networks,[12] and long short-term memory networks[13]).

Unsupervised learning algorithms include those encompassing clustering and dimension reduction, in which the goal is to draw inferences and define hidden structures from input data. Unsupervised algorithms are distinguished by the fact that the input data are not categorized or classified. Rather, the algorithm is expected to provide a structure for the data. In contrast,

[2] A description of support vector machines is provided by Cortes & Vapnik (1995) and Vapnik et al. (1997). An application to portfolio construction/optimization is provided by Gupta, Mehlawat, and Mittal (2012) and Silva (2024), portfolio rebalancing by Sahu and Kumar (2024), and risk management by Singh (2009).

[3] See Breiman (2001b).

[4] For a description of k-means methods and their application to portfolio construction, see Aslam, Bhuiyan, and Zhang (2023).

[5] See López de Prado (2016) and Raffinot (2018).

[6] See Sood et al. (2023) for the application of reinforcement learning to portfolio management.

[7] Seminal papers in the development of Q-learning include Watkins and Dayan (1992), Tesauro (1995), Hasselt (2010), Schulman, et. al (2015), Wang et. al (2016), and Ha and Schmidhuber (2018). For an application of Q-Learning to portfolio management, see Gao et al. (2023).

[8] For more background on the SARSA algorithm, see Rummery and Niranjan (1994) and Sutton and Barto (1998).

[9] For an application of deep Q networks to portfolio management, see Gao et al. (2023).

[10] See Zheng, Jiang, and Su (2021) for a description of gradient policy methods and its application to portfolio management.

[11] For an application of convolution neural networks to portfolio management, see Gao et al. (2023).

[12] For an application of the recurrent neural network algorithm to portfolio construction, see Cao et al. (2023).

[13] This algorithm, developed by Hochreiter and Schmidhuber (1997), has been applied to portfolio optimization by Martínez-Barbero, Cervelló-Royo, and Ribal (2024).

supervised learning algorithms use input variables that are clearly defined. With supervised learning, the goal is to produce rules and/or inferences that can be reliably applied to new data, whether for classification or regression-type problems. Neural networks are motivated by the functioning of the human brain. Their basic design consists of a collection of data processors organized in layers, called neurons (or nodes). Information is processed via the responses of neurons to external inputs. These responses are then passed on to the next layer, eventually ending up as final output. Reinforcement learning is a branch of machine learning which studies how agents learn to maximize rewards in specific environments. By an "environment" we mean a specific state of the world, actions that an agent can take, and rewards and punishments attached to specific actions. Reinforcement learning is based on the idea that agents learn based on the pattern of rewards and punishments experienced in previous states. As they learn these patterns, they can make more informed decisions in the future.

While both econometric and ML models can be used for prediction, ML models are typically focused on pattern recognition and prediction, while econometric models have typically been geared toward providing statistical explanations of past economic and market events.

Whatever type of model one uses, the signals it produces are supposed to be informationally additive to an investment process. There are many ways of measuring signals' informational value. For example, a well-known statistical test is based on the idea of *Granger causality* (GC) (Granger, 1969), which stipulates that for a given predictor x and target variable y we can say that x *Granger-causes* y if:

(1) x temporally precedes y
(2) x provides more predictively useful information than a rival "naïve" predictor

One way of formally expressing (2) is $P(Y_{t+1}|X_t, Y_t) > P(Y_{t+1}|Y_t)$, which can be interpreted as saying that given a target variable Y, the probability of predicting its one-step-ahead value Y_{t+1} is higher given temporally prior information about a predictor variable X in addition to temporally prior information about Y alone.

ML also provides us with metrics to evaluate a signal or model's predictive efficacy. The majority are based on tests from statistical classification (analogous to type I and II tests in hypothesis testing). Thus, they evaluate signals based on their ability to classify events correctly. For example, a correct (true positive) classification could be to predict that an asset produces a positive return in a future point in time (e.g., one month after the prediction is made) and observe that the asset does in fact produce a positive return as predicted. Some major measures of predictive accuracy are:

$$Precision = \frac{\text{True Positives}}{\text{True Positives} \ + \ \text{False Positives}}$$

A higher precision score indicates fewer false positives, meaning the model is more precise in its positive predictions.

$$Recall = \frac{\text{True Positives}}{\text{True Positives} \ + \ \text{False Negatives}}$$

Higher recall scores imply that the model captures a larger proportion of actual positives.

$$F1 = 2 \times \frac{\text{Precision} \ \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

A high precision at the expense of a low recall is undesirable, similar to a high recall at the expense of a low precision. The F1 score is the harmonic mean of precision and recall.

It informs whether an algorithm has performed well on both measurements. F1 scores range from 0 to 1.

$$Accuracy = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}}$$

A higher accuracy score (closer to 1) suggests that the model makes fewer mistakes, correctly predicting more instances.

In addition to the well-worn methods for testing signals, there are more novel approaches that draw on both ML and traditional statistics. For example, Simonian (2020) introduces a modular ML framework for model validation in which the output from one procedure serves as the input to another procedure within a single validation framework. The framework uses an econometric model in the first module to classify data in an economically intuitive way. Proceeding modules apply data science techniques to evaluate the predictive characteristics of the model components. It then applies the framework to the fundamental law of active management (Grinold, 1989; Grinold and Kahn, 2000), a well-known formal characterization of portfolio managers' alpha generation process. In contrast to standard applications of the law, in which it has been used to evaluate a manager's existing active management process, the framework recasts the law as a means to test investment signals for their potential use, individually or collectively, in a manager's investment process. To illustrate how this application works, an example is provided using the well-known Fama–French factors as test signals.

3 Feature selection

Feature or factor selection is one of the most important aspects of building investment models. This is because the features that explain phenomena are often also the most likely candidates for predictive signals. A popular regression-based analytical framework for feature selection is the least absolute shrinkage and selection operator (LASSO).[14] LASSO provides a methodology that can help portfolio managers or analysts both select variables for and mitigate against overfitting in (i.e., regularize) our models, to enhance their predictive accuracy and interpretability. Within a regression framework, the objective of LASSO is to find the best model by minimizing the squared errors between the portfolio returns R_i and a model $\widehat{R}_i$ fitted to explain portfolio returns. LASSO is described formally in the following manner:

$$\underset{\beta \in R^p}{\operatorname{argmin}}\left[\sum_{i=1}^{K}\left(R_i - \widehat{R}_i\right)^2\right] = \sum_{i=1}^{K}\left(R_i - \alpha - \sum_{j=1}^{P}\beta_j F_{ij}\right)^2 + \lambda\sum_{j=0}^{P}\left|\beta_j\right| \quad (1)$$

where R_i is the portfolio return, $\boldsymbol{\alpha_i}$ is an intercept, $\boldsymbol{F_{j,t}}$ are factors, and $\boldsymbol{\beta_i}$ is a coefficient of factor loadings that measures the sensitivity of the portfolio return to the factors. The regularization term $\lambda\sum_{j=0}^{P}\left|\beta_j\right|$ penalizes the absolute values of the coefficients. The magnitude of the penalty is controlled by the regularization parameter λ. When λ takes a value of zero, we have the ordinary least squares (OLS) model. However, a sufficiently large value of λ will force some of the coefficients β_j to become zero, consequently excluding them from the model.

There are many convenient and practical ways to use LASSO in the context of model development. For example, when building a multi-asset model, there are typically numerous

[14] Santosa and Symes (1986).

potential factors that could provide some informational value. However, in keeping with the goal of parsimony for factor models, it would be impractical to use a large set of factors, only some of which would be the primary drivers of portfolio behavior. It would be much more efficient to select a subset of a more expansive array of risk factors. To do this via LASSO, one would simply have to select a large number of factors (say, 60), specify the final number of factors the portfolio manager wants in the model (say, 8), and run the procedure described above. Then LASSO will provide the subset of risk factors that have the most explanatory and predictive power while simultaneously having the lowest correlation with one another.

Feng, Giglio, and Xiu (2020) and Freyberger, Neuhierl, and Weber (2020) employ LASSO to create data-driven combinations of stock return predictors from large sets of signals. Similarly, Rapach et al. (2019) use LASSO to identify the most significant predictors and create optimal combinations among a large set of industry and market returns. Aside from LASSO, ML frameworks can also be used for feature selection. These include elastic nets, and artificial neural networks (ANNs). ML methods that can capture both nonlinear patterns and interaction between features are ANNs, support vector machines, and tree-based methods. As noted by Bartram et al. (2021), ML methods can be used to identify potentially complex nonlinear relationships and a large number of features. Rapach et al. (2013) use adaptive elastic nets to explore the predictive ability of lagged US market returns on global indexes.

Gu et al. (2020) provide an example of the application of elastic nets to the selection and combination of both fundamental and technical signals to predict stock returns. Messmer and Audrino (2022) conclude that adaptive LASSO outperforms both OLS and LASSO when used to select from a vast number of features. Kozak et al. (2019) adopt a shrinkage approach to build a stochastic discount factor summarizing the joint explanatory power of a large set of stock characteristics. As an alternative to these data-driven approaches, Bew et al. (2019) and Papaioannou and Giamouridis (2020) consider expert predictions as model inputs instead of stock characteristics.

Finally, we note that in addition to helping us with feature selection, many ML algorithms such as random forests, can also tell us which features are the primary drivers of an asset or portfolio performance. These relativized feature importance (RFI) values indicate the importance of a factor in predicting asset or portfolio returns when compared to the other factors in a set. Because RFI values sum to unity, they are naturally viewed as weights. RFI values can thus plausibly be used to offer guidance in portfolio construction. More generally, permutation feature importance methods such as mean decrease accuracy (MDA) help identify the features responsible for the cross-validated performance (see López de Prado (2018) and López de Prado (2020) for a discussion).

Aside from finding new features and signals, ML methods can also be used to make the application of well-known signals more effective. For example, consider traditional factor models, which are by now widespread in finance and form an integral part of investment practice. The most common models in the investment industry are linear, a development that is no doubt the result of their familiarity and relative simplicity. Linear models, however, often fail to capture important information regarding asset behavior. Research by Simonian and Wu et al (2019) however shows that ML algorithms such as random forests, can be used to produce factor frameworks that improve upon more traditional models in terms of their ability to account for nonlinearities and interaction effects among variables, as well as their higher explanatory power.

Another example of ML methods being used to improve upon existing econometric models is the development of ML-driven regime-switching models. The latter models have become popular among academic researchers to analyze the interaction between financial assets and the broader economy, especially the significant, at times abrupt, changes in behavior they

undergo at specific points in time. Regime models range from the simple demarcations of regimes based on a set of statistical characteristics (e.g., high/ low volatility regimes), to models driven by sophisticated econometric theory, in which the transition from one regime to another is assumed to follow a process of some type. Well-known models falling into the latter category include those described by Hamilton (1989), Filardo (1994), Kim and Nelson (1998), and Guidolin and Timmermann (2007, 2008), among others. While well-known among investors, regime models nevertheless have a long history of poor predictive performance. As a more forecast-focused counterpoint to traditional regime-switching models, Simonian and Wu (2019) use spectral clustering,[15] a graph-theoretic approach to classifying data, to implement a regime-based trading model. They show that their framework not only produces predictively effective macro signals, but also classifies regimes in an economically intuitive way. Yazdani (2020) further builds on this regime-related research.

Typically, signal generation involves more than one signal. Three often used ML tools employed for multiple signal generation are regression trees, panel trees, and XGBoost for capturing non-linearities and interactions of factors. An additional advantage is their interpretability which is critical for ensuring legal and ethical compliance, transparency, and insights into which input features are most predictive. Signal generation can be enhanced by using NLP. These techniques can be employed to extract factors from unstructured data such as earnings calls and financial reports. These are critical for dynamic asset allocation as will be explained later for updating factors. Simonian and Wu (2019) also demonstrate, by means of a simple example, how combining signals from more than one ML framework can produce viable strategies. They present an example that combines signals produced from a random forest algorithm with signals produced by another framework known as *association rule learning* (Agrawal et. al, 1993) to build an equity sector rotation strategy. They further show how investors can use ML-generated signals in combination with traditional statistical signals (e.g., simple volatility measures) to enhance their investment strategies.

4 Portfolio optimization and construction

The Markowitz (1952) mean–variance optimization model revolutionized the way investors approach portfolio construction. Markowitz's insight was twofold. First, he cast the investment problem as an optimization problem, with its solution characterized by the efficient frontier. Second, he developed an algorithm to estimate the efficient frontier, known as the critical line algorithm.

While the model's core principles are widely recognized, it is important to note that despite its pioneering contributions, the Markowitz model faces several limitations. These limitations primarily arise from the computational and data constraints of its time, as well as its foundational assumptions. To understand why, consider an investment universe with N assets, where the expected value of returns is represented by an array μ, and the covariance of returns is represented by matrix V. We would like to minimize the variance of a portfolio with allocations ω, measured as $\omega\prime V\omega$, subject to achieving a target $\omega\prime a$, where a characterizes the optimal solution. In general terms, the problem can be stated as

$$\min_{\omega} \frac{1}{2}\omega\prime V\omega$$

[15] The fundamental ideas behind spectral clustering were introduced in the seminal papers by Hall (1970), Donath and Hoffman (1973), and Fiedler (1973). For a historical overview of spectral clustering, see Spielman and Teng (2007).

$$\text{s.t.} : \omega\prime a = \overline{a}$$

This problem can be expressed in Lagrangian form as

$$L[\omega, \lambda] = \frac{1}{2}\omega\prime V\omega - \lambda(\omega\prime a - \overline{a})$$

with first-order conditions

$$\frac{\partial L[\omega, \lambda]}{\partial \omega} = V\omega - \lambda a$$
$$\frac{\partial L[\omega, \lambda]}{\partial \lambda} = \omega' a - \overline{a}$$

Setting the first-order (necessary) conditions to zero, we obtain that $V\omega - \lambda a = 0$, $\omega = \lambda V^{-1}a$, and $\omega' a = a'\omega = \overline{a}$, $\lambda a' V^{-1}a = \overline{a}$, $\lambda = \frac{\overline{a}}{a' V^{-1}a}$, thus

$$\omega^* = \overline{a}\frac{V^{-1}a}{a\prime V^{-1}a}$$

The second-order (sufficient) condition confirms that this solution is the minimum of the Lagrangian,

$$\begin{vmatrix} \frac{\partial L^2[\omega,\lambda]}{\partial \omega^2} & \frac{\partial L^2[\omega,\lambda]}{\partial \omega \partial \lambda} \\ \frac{\partial L^2[\omega,\lambda]}{\partial \lambda \partial \omega} & \frac{\partial L^2[\omega,\lambda]}{\partial \lambda^2} \end{vmatrix} = \begin{vmatrix} V\prime & -a\prime \\ a & 0 \end{vmatrix} = a\prime a \geq 0$$

The common approach to estimating ω^* is to compute

$$\widehat{\omega}^* = \overline{a}\frac{\widehat{V}^{-1}\widehat{a}}{\widehat{a}\prime\widehat{V}^{-1}\widehat{a}}$$

where $\widehat{V}$ is the estimated V, and $\widehat{a}$ is the estimated a. In general, replacing each variable with its estimate will lead to unstable solutions, that is, solutions where a small change in the inputs will cause extreme changes in $\widehat{\omega}^*$. In particular, when the determinant of $\widehat{V}$ is relatively small, the entries in $\widehat{V}^{-1}$ are large, and the solution $\widehat{\omega}^*$ becomes extremely sensitive to even small changes in $\widehat{a}$. This is problematic, because in many practical applications there are material costs associated with the re-allocation from one solution to another. The next two sections explain the major sources of Markowitz instability, and how the ML literature has addressed them.

4.1 Noise-induced instability

Consider a matrix of independent and identically distributed (IID) random observations X, of size $T \times N$, where the underlying process generating the observations has zero mean and variance σ^2. The matrix $C = T^{-1}X\prime X$ has eigenvalues λ that asymptotically converge (as N→ + ∞ and T→ + ∞ with 1<T/N< + ∞) to the Marcenko-Pastur probability density function (PDF),

$$f[\lambda] = \begin{cases} \frac{T}{N}\frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{2\pi\lambda\sigma^2} & if\lambda \in [\lambda_-, \lambda_+] \\ 0 & if\lambda \notin [\lambda_-, \lambda_+] \end{cases}$$

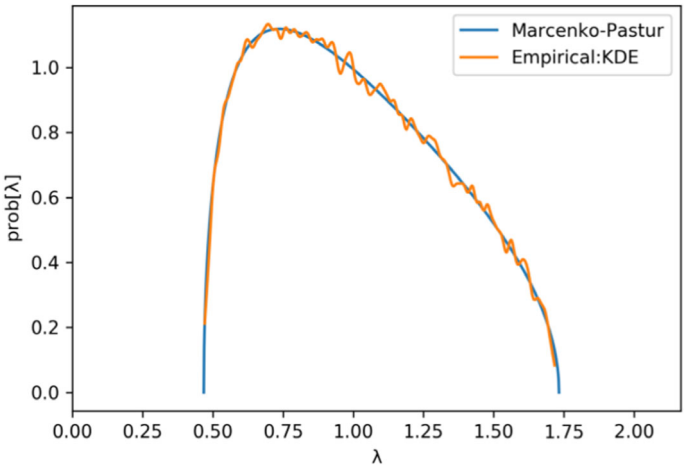

Fig. 1 A visualization of the Marcenko-Pastur theorem

where the maximum expected eigenvalue is $\lambda_+ = \sigma^2\left(1 + \sqrt{\frac{N}{T}}\right)^2$, and the minimum expected eigenvalue is $\lambda_- = \sigma^2\left(1 - \sqrt{\frac{N}{T}}\right)^2$. When $\sigma^2 = 1$, then C is the correlation matrix associated with X.

Eigenvalues $\lambda \in [\lambda_-, \lambda_+]$ are consistent with random behavior due to the finite sample size. Specifically, we can associate eigenvalues $\lambda \in [0, \lambda_+]$ with estimation error. One problem is that in finance most eigenvalues fall under the Marcenko-Pastur distribution,[16] and can be considered insignificant. The implication is that neither C^{-1} nor V^{-1} can be estimated robustly. Accordingly, solutions are only optimal in-sample, not out-of-sample (Fig. 1).

Ledoit and Wolf (2004a, 2004b) introduced a popular method to shrink a numerically ill-conditioned covariance matrix. By making the covariance matrix closer to a diagonal, shrinkage reduces its condition number. However, shrinkage accomplishes that without discriminating between noise and signal. As a result, shrinkage can further eliminate an already weak signal. While shrinkage makes the covariance matrix more stable, it treats all sources of variance the same way, reducing both random noise and signal patterns. This uniform approach might weaken understated but important signals in the data, which are essential for making accurate analyses and informed decisions. In practice, choosing the right amount and type of shrinkage is very important. Consequently, while the Ledoit-Wolf shrinkage method is a helpful tool for stabilizing covariance matrices, it needs to be applied carefully due to this limitation and the specific characteristics of the data.

A method to filter out the noisy part of the covariance matrix by identifying and removing the components associated with the eigenvalues that match the predictions of random matrix theory (RMT) for purely random data was proposed by Potters et al (2005). RMT has proven highly effective in analyzing the structure of stock return correlation matrices. This framework

[16] The Marcenko-Pastur theorem is a fundamental result in RMT that describes the asymptotic behavior of the eigenvalues of large random covariance matrices. A probability distribution describing this asymptotic behavior of the eigenvalues of large random covariance matrices is referred to as the Marcenko-Pastur distribution.

helps in identifying and distinguishing genuine market signals from random noise in financial data, providing deeper insights into the relationships between different stocks. By applying RMT, researchers can better understand the complex interactions within financial markets, leading to improved portfolio optimization strategies. Potters et al (2005) propose analyzing the eigenvalues of the empirical covariance matrix and identifying the ones that deviate from the predictions of RMT as containing the true correlation information for data that is purely random. The eigenvectors that are identified as corresponding to the non-deviating, noisy eigenvalues predicted by RMT are considered random and are filtered out. After the filtering, the resulting covariance matrix constructed using the deviating eigenvalues then captures the true correlations that lead to better performance in applications such as portfolio optimization compared to the standard empirical covariance matrix.

Building on the ideas of Potters et al (2005), López de Prado (2020) proposes using ML to fit the function f[λ] to the empirical eigenvalue distribution, which allows deriving an implied variance (σ^2) that the the random eigenvectors present. That is, by fitting f[λ] to the eigenvalue distribution separates the random and non-random components. This approach derives the variance that is explained by the random eigenvectors present in the correlation matrix, and it will determine the cut-off levelλ_+, adjusted for the presence of non-random eigenvectors. Because we know what eigenvalues are associated with noise, we can shrink only those, without diluting the signal (Fig. 2).

Let $\{\lambda_n\}_{n=1,\ldots,N}$ be the set of all eigenvalues, ordered descending, and i be the position of the eigenvalue such that $\lambda_i > \lambda_+$ and $\lambda_{i+1} \leq \lambda_+$. Then we set $\lambda_j = \frac{1}{N-i}\sum_{k=i+1}^{N}\lambda_k$, $j = i+1, \ldots, N$, hence preserving the trace of the correlation matrix. Given the eigenvector decomposition $VW = W\Lambda$, we form the de-noised correlation matrix C_1 as

$$C_1 = \left(\text{diag}\left[\tilde{C}_1\right]\right)^{-1/2} \tilde{C}_1 \left(\text{diag}\left[\tilde{C}_1\right]\right)^{-1/2}$$

where $\tilde{C}_1 = W\tilde{\Lambda}W'$ and $\tilde{\Lambda}$ is the diagonal matrix holding the corrected eigenvalues. The reason for the second transformation is to re-scale the matrix $\tilde{C}_1$, so that the main diagonal of C_1 is an array of 1s.

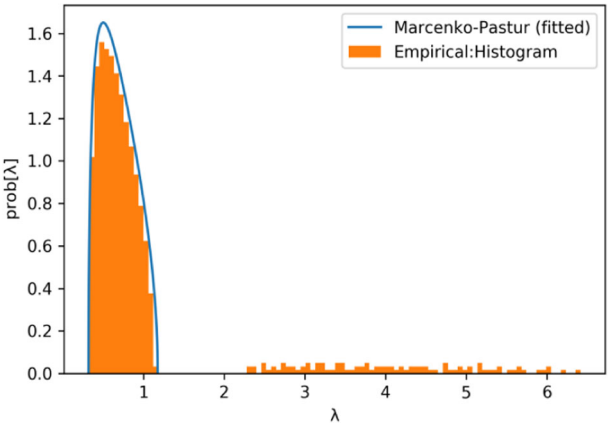

Fig. 2 Fitting the Marcenko-Pastur PDF on a noisy covariance matrix through a Kernel Density Estimator

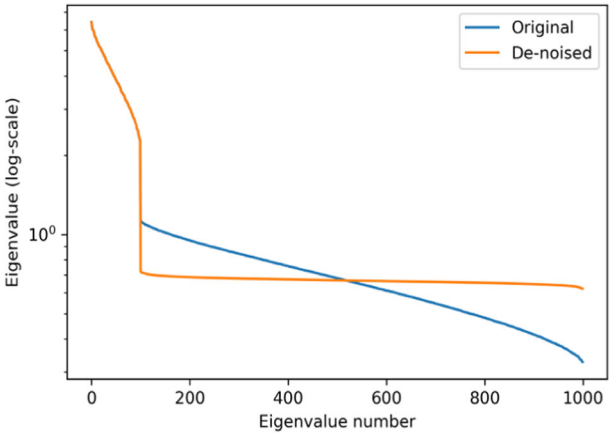

Fig. 3 A comparison of eigenvalues before and after applying the residual eigenvalue method

Through Monte Carlo experiments, López de Prado (2020) shows that, applying this denoising procedure, the maximum Sharpe ratio portfolio incurs only 0.04% of the root mean square error (RMSE)—a metric employed for assessing the accuracy and performance of predictive models − incurred by the maximum Sharpe ratio portfolio without de-noising. That is a 94.44% reduction in RMSE from de-noising alone, compared to a 70.77% reduction using Ledoit-Wolf shrinkage. While shrinkage is somewhat helpful in the absence of de-noising, it adds no benefit in combination with de-noising. This is because shrinkage dilutes noise but also dilutes some of the signal in the process (Fig. 3).

4.2 Input uncertainty

As explained earlier, a major source of Markowitz's instability is the need to invert a covariance matrix $\widehat{V}$ that is likely estimated on an insufficient number of observations T. Consequently, as $\lambda_{\_}$ approaches zero, the determinant of $\widehat{V}$ also approaches zero. When this occurs, the condition number of the covariance matrix becomes very high, thereby magnifying any estimation error on a. To circumvent this problem, some practitioners have opted for dropping $\widehat{a}$ entirely, giving rise to so-called risk-parity approaches, which essentially compute the allocations $\widehat{\omega}^*$ that assign equal risk to each asset. While dropping $\widehat{a}$ reduces some of the instability in $\widehat{\omega}^*$, the issue remains that risk parity requires the inversion of an ill-conditioned matrix $\widehat{V}$.

To address this flaw of risk parity, López de Prado (2016) introduced a ML-based approach known as hierarchical risk parity (HRP) to derive risk-parity allocations without the need to invert $\widehat{V}$. HRP involves three steps. First, HRP applies a hierarchical clustering to the estimated covariance $\widehat{V}$. The purpose of doing so helps in identifying the covariance matrix's underlying structure as well as grouping assets that have similar risk profiles together. Second, HRP reorders the rows and columns of $\widehat{V}$, so that the largest values lie along the main diagonal. This reordering makes it easier to identify clusters that have higher inter-asset correlations. Doing so enhances the effectiveness of risk allocation. Third, HRP splits allocations through recursive bisection of the reordered covariance matrix. By iteratively splitting the

covariance matrix into smaller subsets, HRP generates risk-parity allocations that reflect the diversification benefits across different asset classes.

Through Monte Carlo experiments, López de Prado (2016) shows that HRP solutions exhibit lower out-of-sample variance than the traditional Markowitz's minimum variance solutions. Furthermore, Antonov et al (2024) derived analytical values for the noise of allocation weights coming from the estimated covariance and provide empirical support that HRP is indeed less noisy (and thus more robust) than the traditional Markowitz's minimum variance solution, further highlighting its usefulness in portfolio optimization.

A modified HRP method has been proposed by Molyboga (2020). This HRP method replaces the traditional sample covariance matrix with an exponentially weighted covariance matrix, incorporating Ledoit-Wolf shrinkage. Doing so, enhances diversification both within and among clusters through an equal volatility allocation strategy. Moreover, this method improves temporal diversification by implementing volatility targeting within portfolios.

In a pioneering investigation of risk budgeting (RB), Koumou (2024) proposed an RB methodology grounded in a filtered similarity matrix derived from hierarchical clustering. The advantages of this approach, termed Hierarchical Risk Budgeting (HRB), include visualizability, adaptability, and resilience. Leveraging these characteristics, Koumou illustrates how HRB can be adapted to integrate asset expected returns without imposing additional constraints for error estimation management. Empirical tests demonstrate HRB's improvements over the original HRP, particularly when accounting for asset return expectations.

A cutting-edge approach combining the Long Short-Term Memory (LSTM) model with Support Vector Regression (SVR) to gauge investor sentiment for integration into the Black–Litterman model[17] was proposed by Punyaleadtip et al. (2024). Applying the proposed method to analyze both the SET Index of the Thai equity market and the Dow Jones Index of the US equity market, they find support for the method's effectiveness by consistently outperforming traditional buy-and-hold strategies in both stock markets. This innovative combining of LSTM and SVR not only improves the predictive power of sentiment analysis but also offers valuable insights for portfolio management in dynamic market environments.

Owen (2023) identified portfolios constructed using hierarchical clustering, combined with a three-month buy-and-hold, long-only strategy, generated higher out-of-sample risk-adjusted returns compared to those derived from traditional Markowitz portfolio strategies. Additionally, portfolios constructed using ML techniques exhibited significantly reduced frequency of portfolio rebalancing, thereby reducing trading costs. This finding demonstrates the potential benefits of applying ML-based techniques to optimize portfolio and minimize transaction costs.

Lastly, Menvouta et al. (2023) developed the minCluster portfolio, an innovative methodology that combines the optimization of downside risk measures, hierarchical clustering, and cellwise robustness to identify inherent hierarchical structures within datasets. The minCluster portfolio not only mitigates downside risk through the application of tail risk measures but also consistently outperforms traditional mean–variance and other hierarchical clustering-based strategies in both simulated and real data scenarios.

4.3 Signal-induced instability

Another limitation of Markowitz's mean–variance optimization framework is what López de Prado (2020) described as "Markowitz's curse": Markowitz's solutions become more

[17] For a detailed description of the Black-Litterman model and its offshoots, see Kolm, Ritter, and Simonian (2021).

unstable when they are needed the most. To see how, consider a correlation matrix between two securities,

$$C = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

where ρ is the correlation between their returns. Matrix C can be diagonalized as $CW = W\Lambda$ as follows, where

$$W = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}, \Lambda = \begin{bmatrix} 1+\rho & 0 \\ 0 & 1-\rho \end{bmatrix}$$

The trace of C is $tr(C) = \Lambda_{1,1} + \Lambda_{2,2} = 2$, so ρ sets how big one eigenvalue gets at the expense of the other. The determinant of C is given by $|C| = \Lambda_{1,1}\Lambda_{2,2} = (1+\rho)(1-\rho) = 1-\rho^2$. The determinant reaches its maximum at $\Lambda_{1,1} = \Lambda_{2,2} = 1$, which corresponds to the uncorrelated case, $\rho = 0$. The determinant reaches its minimum at $\Lambda_{1,1} = 0$ or $\Lambda_{2,2} = 0$, which corresponds to the perfectly correlated case, $|\rho| = 1$. The inverse of C is $C^{-1} = W\Lambda^{-1}W\prime = \frac{1}{|C|}\begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix}$. The implication is that, the more ρ deviates from zero, the bigger one eigenvalue becomes relative to the other, causing $|C|$ to approach zero, which makes the values of C^{-1} explode. This happens regardless of the N/T ratio, hence this source of instability is independent from noise (which was discussed in an earlier section).

Matrix C is just a standardized version of V, and the conclusions we drew on C^{-1} apply to the V^{-1} used to estimate ω^*. When securities within a portfolio are highly correlated ($-1 < \rho \ll 0$ or $0 \ll \rho < 1$), C has a high condition number, and the values of V^{-1} explode. This is problematic in the context of portfolio optimization, because ω^* depends on V^{-1}, and unless $\rho \approx 0$, we must expect an unstable solution to the convex optimization program. In other words, Markowitz's solution is guaranteed to be numerically stable only if $\rho \approx 0$, which is precisely the case when we don't need it! The reason we needed Markowitz was to handle the $\rho \not\approx 0$ case, but the more we need Markowitz, the more numerically unstable is its estimation of ω^*.

To address Markowitz's curse, López de Prado (2020) proposed a ML-based algorithm that he called Nested Clustered Optimization (NCO), which utilizes hierarchical clustering techniques as a basic component of the approach. The NCO algorithm involves five steps. First, the correlation matrix is de-noised and clustered for the purpose of identifying relationships and patterns. Second, the NCO algorithm computes intra-cluster allocations, optimizing allocation strategies within each identified cluster. Third, the solutions from the first two steps are used to reduce the dimensionality of the system, to one row/column per cluster. Fourth, inter-clustered allocations are computed on the reduced system, optimizing allocation between clusters. Fifth, the final asset allocation is the dot product of the inter-cluster and intra-cluster allocations. Monte Carlo experiments show that NCO computes the maximum Sharpe ratio portfolio with 45.17% of Markowitz's RMSE (i.e. a 54.83% reduction in RMSE). The combination of shrinkage and NCO yields an 18.52% reduction in the RMSE of the maximum Sharpe ratio portfolio, which is better than shrinkage but worse than NCO. Once again, NCO delivers substantially lower RMSE than Markowitz's solution, and shrinkage adds no value. It is easy to test that NCO's advantage widens for larger portfolios.

4.4 IID normality

A third limitation of Markowitz's mean–variance optimization framework lies in its reliance on the assumption that asset returns are IID as Gaussian. To address this limitation, forward-looking returns and risk measures can be introduced into the optimization process. Moreover, incorporating additional statistical moments—such as skewness and kurtosis—allows for a more comprehensive assessment of the risk associated with extreme price movements.

Pinelis and Ruppert (2022) demonstrate that ML can yield economically and statistically significant benefits in portfolio allocation between market indices and risk-free assets. The study implements optimal portfolio rules that adjust to time-varying expected returns and volatilities using two random forest models. The first model forecasts monthly excess returns incorporating macroeconomic factors such as payout yields, while the second estimates prevailing volatilities. The findings underscore the advantages of reward-risk timing via ML, which surpass traditional buy-and-hold strategies in terms of utility, risk-adjusted returns, and maximum drawdowns. This research serves as a comprehensive framework for applying machine learning to both return and volatility timing.

Uysal and Mulvey (2021) explore a ML methodology for regime-based asset allocation. Their approach includes two main components: (1) regime modeling and prediction and (2) the formulation of a regime-based strategy to improve risk-parity portfolio performance. They employ supervised learning algorithms, notably random forest, to predict the likelihood of recessions or market downturns, using a substantial macroeconomic dataset. Their out-of-sample validation, spanning from 1973 to 2020, confirms the accuracy of their recession forecasts in the United States. By integrating these predictions with a dynamic investment overlay strategy, their approach significantly enhances risk-adjusted returns in both two-asset and multi-asset portfolios, even amidst the rising interest rates of the late 1970s.

A novel reinforcement learning framework designed for the autonomous and efficient management of time-dynamic risk budgets is proposed by Han (2023). Marking a significant departure from traditional risk management strategies, this novel approach leverages advanced ML techniques to adaptively allocate risk across different asset classes. Han demonstrates that this learning agent not only enhances target performance measures but also exhibits the capability to approximate optimal risk budget deviations through its learned policy. These findings suggest promising applications for the integration of reinforcement learning in portfolio management, potentially revolutionizing the way risk is managed and allocated in dynamic market environments.

Bosancic et al (2024) developed a novel approach for selecting features within a regime-aware portfolio model. Their work focuses on six Fama–French equity factors. Two regimes are proposed: growth and crash for each factor. The algorithm labels regimes over a 20-year rolling-horizon training window by applying the continuous statistical jump model. The optimal features are selected every six months during the testing period from January 1990 to December 2023. The empirical results provide evidence of the ability to protect drawdown during crash episodes and thereby maximize Sharpe ratios and other statistics. The positive results occur for both individual factors and a portfolio of factors.

4.5 Single-period (myopic) optimization

A fourth limitation of Markowitz's mean–variance optimization framework lies in its design as a single-period model, which contrasts sharply with the multi-period investment horizons typical of most investors. This limitation of the model significantly impacts the model's

applicability to real-world scenarios, where investment decisions and market conditions evolve over time. ML offers a promising solution for addressing this limitation. It does so by facilitating a dynamic, multi-period approach to portfolio optimization. ML algorithms can continuously update the inputs for the portfolio optimization process—such as expected returns, variances, and covariances—based on recent market data and trends. By providing this flexibility for including new information as it becomes available, it allows portfolio managers to rebalance their asset allocation or current portfolio to adapt more effectively to changing market conditions.

Rosenberg et al. (2016) address a multi-period portfolio optimization challenge using the quantum annealer from D-Wave Systems − a type of quantum computing device designed to solve optimization problems.[18] They develop a problem formulation, explore various integer encoding schemes, and provide numerical examples that highlight high success rates. Their approach notably avoids the need for covariance matrix inversion and includes transaction costs, encompassing both permanent and temporary market impacts. The discrete multi-period portfolio optimization problem tackled is highlighted as significantly more complex than its continuous variable counterpart.

In ML, a hyperparameter is a parameter whose value is set prior to the learning process begins. Unlike model parameters, which are learned from the training data during the training process, hyperparameters are not directly learned from the data. Instead, a hyperparameter is either specified by the modeler or selected via some search process. The hyperparameters can have a significant impact on the performance and behavior of a ML model because they control aspects of the model's architecture − complexity, regularization, and optimization strategy.[19] For achieving good performance and generalization in ML models, selecting the appropriate hyperparameter values is critical.

Typically, selecting hyperparameters involves experimentation and tuning through various methods. Such methods include grid search, random search, or more sophisticated optimization techniques like Bayesian optimization. A systematic methodology for hyperparameter optimization by integrating recent developments in automated ML and multi-objective optimization is proposed by Nystrup et al. (2020). Their approach focuses on optimizing hyperparameters within a training set to achieve optimal outcomes, constrained by market-determined realized costs. In their application to both single- and multiperiod portfolio selection, sequential hyperparameter optimization demonstrates superior risk-return trade-offs compared to traditional methods such as manual, grid, and random hyperparameter searches, while requiring fewer function evaluations. Notably, the solutions generated exhibit greater stability from in-sample training to out-of-sample testing, indicating a reduced likelihood of solutions that perform well in training by chance.

Abdelhakmi and Lim (2024) explore a multi-period generalization in which the time horizons of expert views do not align with those of a dynamically-trading investor. Utilizing an underlying graphical structure that links asset prices and expert views, they derive the conditional distribution of asset returns under a geometric Brownian motion price process. Additionally, they demonstrate that this distribution can be expressed via a multi-dimensional Brownian bridge. They introduce a new price process modeled as an affine factor model, where the conditional log-price process functions as a vector of factors. The authors provide an

[18] Quantum annealers, such as those developed by D-Wave Systems, use quantum bits to represent the variables in the optimization problem so as to more efficiently solve for certain classes of problems than classical computers.

[19] Examples of hyperparameters include the learning rate, the number of hidden layers and units in a neural network, the selection of the kernel in a support vector machine, and the depth and size of decision trees in a random forest algorithm.

explicit formula for the optimal dynamic investment policy and examine the hedging demand prompted by the introduction of a new covariate. More broadly, the study illustrates that Bayesian graphical models offer an effective framework for integrating complex information structures into the Black-Litterman model.

5 Stress testing

Stress testing a portfolio involves simulating the portfolio's performance under extreme market conditions with the goal of identifying the portfolio's potential vulnerabilities. Econometrics models have long-been used to define scenarios (or regimes) for stress testing based on historical return data and economic and financial market conditions. However, it is well-known that econometric models are limited in their ability to capture nonlinear relationships and detect anomalies from the data for generating and, as a result, failing to provide realistic scenarios for stress testing.

ML algorithms, such as the various clustering approaches that are available, can be used to detect extreme/anomalous events that are more meaningful than those identified by traditional econometric models. Clustering methods include k-means clustering, hierarchical clustering, Gaussian mixture models (GMMs), density-based spatial clustering of applications with noise (DBSCAN), and spectral clustering. Enhanced by ML techniques such as these, Monte Carlo simulation can then employ historical data to optimize the simulations by improving their computational efficiency and generating more realistic scenarios.

Let us discuss in detail how two of these ML clustering techniques, GMMs and DBSCAN, have been used for portfolio risk management. Consider first GMMs. These models can be used to create "conditional forecasts" based on the occurrence of certain market conditions, such as a drastic change in interest rates and foreign-exchange rates, spikes in the price of oil, or an unexpected change in macroeconomic conditions.

The assumption of GMMs is that the data points are generated from a mixture of multiple Gaussian distributions. Each distribution is associated with a different underlying market condition or regime. This feature of GMMs make them particularly appealing for scenario analyses that correspond to a particular change. Botte and Bao (undated), for example, explain how GMMs are used for portfolio stress testing by Two Sigma, a large hedge fund, by generating four market conditions (scenarios). Each of the four market conditions is depicted by a 17-dimensional Gaussian distribution, with the distributions varying across the four market conditions and having unique factor means, volatilities, and correlation structures. GMM derived market conditions are data-driven and therefore may not directly correspond to conventional market environments. Botte and Bao (n.d.) note the advantages and disadvantages of this approach. The advantage is that it can identify unexpected insights into market behavior, offering the portfolio manager with information that might not be apparent without the application of this ML technique. The disadvantage is that it can be challenging to apply to the market conditions that GMM identifies, making them appear abstract and disconnected from conventional market analyses.

DBSCAN excels not only in identifying clusters within datasets but also in managing outliers, which is particularly advantageous in the context of portfolio stress testing. This capability is crucial for pinpointing assets or securities in a portfolio that exhibit atypical behavior compared to the majority under specific market conditions. By effectively highlighting these outliers, DBSCAN facilitates a more comprehensive approach to stress testing, allowing portfolio managers to assess extreme risks and unexpected responses to stressful

market environments. This contributes to a more robust and resilient framework for understanding and mitigating potential vulnerabilities within the portfolio.

We note that it is also possible to combine approaches when building stress testing frameworks. For example, Simonian (2022) combines the random cut forest (Guha et al., 2016) and random forest algorithms, to build a model that classified and predicts large near-term stress episodes and drawdowns.

6 Natural language processing and generative language models

Natural Language Processing (NLP) algorithms apply ML algorithms to linguistic tasks. Computational analysis of language includes two main domains: syntactical analysis and semantical analysis. Syntactical analysis is used to determine the grammatical structure of a phrase or sentence by parsing the syntax of the words using a formal lexicon and set of grammatical rules. Semantical analysis tries to determine the meaning of words and interpret words within a particular sentence and phrase structures.

The basic approaches to NLP include rules-based NLP algorithms and ML-based algorithms.[20]

Rules-based NLP algorithms were the earliest NLP applications and utilized simple if–then decision trees, with preprogrammed rules for answering specific and limited sets of questions. One type of ML-based algorithm is statistical NLP which uses both supervised and unsupervised ML algorithms to automatically extract, classify, and label the elements of a piece of textual data, and then assigns a probability value to each possible meaning of the data being analyzed. Sentiment analysis, for example, primarily utilizes supervised learning. Deep learning NLP algorithms, however, have by now become the leading frameworks for conducting NLP as they can process an immense amount of unstructured data and provide output with a high degree of precision.

One branch of NLP is concerned with building generative language models (GLMs),[21] like GPT (Generative Pre-trained Transformer). These models can process large amounts of unstructured data from various sources such as regulatory filings, analyst reports, earnings call transcripts, and social media. By analyzing the sentiment expressed in these sources and converting them into sentiment scores, GLMs can gauge the market sentiment towards the overall financial market or specific asset classes and sectors. Positive sentiments can indicate bullish conditions, whereas negative sentiments might suggest bearish market conditions. Portfolio managers can use these insights to manually adjust their asset allocations accordingly or integrate sentiment scores into a quantitative model. As an example of the latter, a quantile-based trading strategy can utilize sentiment scores to create long-short equity portfolios. Scores can be used to rank stocks and then select the stocks in the top quantile to purchase and the stocks in the bottom quantile to sell short.

Another application of GLMs that is useful for portfolio managers in adjusting their exposure to an individual security is detecting material events. An occurrence of something that has resulted at a certain time and in a certain place that is associated with one or more participants is referred to as an event. Event extraction refers to the process of detecting one or more events in a text. There are events in financial markets that can move the price of a single security or the market overall. Examples include the announcement of a new product, the signing of a major sales agreement, and the signing of a labor agreement. This information is

[20] For a further discussion of the topics mentioned in this section, see FA Fabozzi (2024).

[21] GLMs are discussed in detail in FA Fabozzi (2024).

transmitted to the market via text or a company spokesperson announcement. Marinov (2019) provides a real-world example of six identified events that resulted in a major movement in Tesla's stock price. By continuously monitoring and processing news feeds and other text sources, these models can alert portfolio managers about events that require immediate action or consideration, thus potentially enhancing portfolio performance.

Finally, GLMs can be used to create simulated market scenarios for training other ML models. These scenarios can significantly improve the training and robustness of ML used in portfolio management and other financial decision-making processes, allowing for a more dynamic and responsive adaptation to evolving market conditions.

7 Challenges

Despite the increased use of ML techniques in portfolio construction, implementation can at times be challenging. Admittedly, some of these challenges apply equally to traditional portfolio construction methods such as mean–variance portfolio optimization and its extensions. For example, data are the key to both traditional and ML models. Data are prone to errors, can be costly in the case of ML, and can be difficult to access.

Overfitting remains a significant challenge in developing reliable and effective ML-based portfolio strategies that are designed to perform well across diverse market conditions and over time. In the realm of ML, overfitting often arises from incorporating too many features, overly extensive training on the data, the intensive training of the model itself, and usage of the test set multiple times (see Bailey et al., 2014, for a discussion). These practices lead to models that perform well with training data but perform poorly when applied to new, out-of-sample data. When this occurs, it makes ML tools less effective for constructing portfolios expected to perform robustly in real-world market conditions. Despite the availability of techniques like data splitting, regularization, and cross-validation to mitigate overfitting, tackling the overfitting issue effectively remains a formidable challenge. Traditional mean–variance models used in portfolio optimization encounter a similar problem. These models can perform well with historical return data but struggle to generalize effectively to new data. Essentially, a portfolio optimized using the mean–variance approach might show excellent performance based on past data but can underperform when faced with future market conditions. This problem underscores the broader difficulty in creating models that not only fit past data well but also adapt robustly to new, unforeseen market scenarios.

Scaling ML models for portfolio construction presents several challenges, particularly as the number of asset classes or new financial instruments in which the portfolio manager may be permitted to invest increases and the variety and volume of data increase. While a ML model may perform well with a limited scope of data and simpler decision-making processes, as the scope of investment decisions broadens to include a wider array of asset classes or financial instruments, the complexity of the models and the computational load increase significantly. The scaling challenge is not just a matter of handling larger datasets. It involves dealing with the increasing complexity of databases and model intricacies such that it does not compromise the existing model's accuracy or performance.

Finally, as ML models become more prevalent in financial decision-making, the need for regulatory frameworks that can address the unique challenges posed by this technology grows. Ensuring that these models are fair, transparent, and compliant with existing financial regulations is crucial to their acceptance and continued use in areas like portfolio construction. In ML-based credit scoring models, for example, regulatory bodies like the U.S.

Consumer Financial Protection Bureau and the European Union's General Data Protection Regulation have set guidelines to address fairness, transparency, and the right to explanation of automated decision-making processes. As ML increasingly becomes integral to portfolio management, securities regulators globally are likely to take a closer look at how these technologies are implemented, drawing parallels to their oversight of ML in credit decision-making, requiring asset management firms to (1) clearly explain the choices made by their algorithms, especially when these decisions affect client investments and returns, (2) allow regulators to inspect and verify the model's compliance with industry standards and legal requirements, and (3) implement stress testing periodically to ensure that models can handle extreme market conditions without causing undue risk to investors.

References

Abdelhakmi, A., & Lim, A. (2024). A multi-period Black-Litterman model. Working paper. Retrieved from https://arxiv.org/abs/2404.18822

Agrawal, R., Imieliński, T., & Swami, A. (1993) Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM–SIGMOD '93, Washington, D.C., May 1993, pp. 207–216. New York: ACM Press

Antonov, A., Lipton, A., & López de Prado, M. (2024). Overcoming Markowitz's instability with the help of the hierarchical risk parity (HRP): Theoretical evidence. Working paper. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4748151

Aslam, B. R., Bhuiyan, R. A., & Zhang, C. (2023). Portfolio construction with k-means clustering algorithm based on three factors. *MATEC Web of Conferences, 377*, 02006. https://doi.org/10.1051/matecconf/202337702006

Bailey, D., Borwein, J., López de Prado, M., & Zhu, J. (2014). Pseudo-mathematics and financial charlatanism: The effects of backtest overfitting on out-of-sample performance. *Notices of the American Mathematical Society, 65*(5), 458–471.

Bartram, S.M., Branke, J., De Rossi, G., & Motahari, M (2021). Machine learning for active portfolio management. *Journal of Financial Data Science, 3*(3), 9–30.

Bew, D., Harvey, C. R., Ledford, A., Radnor, S., & Sinclair, A. (2019). Modeling analysts' recommendations via Bayesian machine learning. *Journal of Financial Data Science, 1*(1),75–98.

Bollerslev, T. (1986). generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics, 31*(3), 307–327.

Bosancic, T., Nie, Y., & Mulvey, J. (2024). Regime-aware factor allocation with optimal feature selection. *The Journal of Financial Data Science, 6*(3), 10.

Botte, A., & Bao, D. (2021) A Machine Learning Approach to Regime Modeling. Two Sigma. Retrieved from https://www.twosigma.com/articles/a-machine-learning-approach-to-regime-modeling/

Box, G. E. P., & Jenkins, G. M. (1976). *Time series analysis: forecasting and control* (Revised). Holden Day.

Breiman, L. (2001a). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science, 16*(3), 199–231.

Breiman, L. (2001b). Random Forests. *Machine Learning, 45*(1), 5–32.

Cao, X., Francis, A., Pu, Z., Zhang, V., Katsikis, P., Stanimirovic, I., Brajevic, I., & Li, S. (2023). A novel recurrent neural network based online portfolio analysis. *Expert Systems with Applications, 233*(15), 120934.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.

Donath, W. E., & Hoffman, A. J. (1973). Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development, 17*(5), 420–425.

Fabozzi, F. A. (2024). Leveraging generative language models for portfolio construction and financial forecasting (Doctoral dissertation). Stevens Institute of Technology

Feng, G., Giglio, S., & Xiu, D. (2020). Taming the factor zoo: A test of new factors. *Journal of Finance, 75*(3), 1327–1370.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal, 23*(2), 298–305.

Filardo, A. J. (1994). Business-cycle phases and their transitional dynamics. *Journal of Business & Economic Statistics, 12*(3), 299–308.

Freyberger, J. Neuhierl, A., & Weber, M. (2020). Dissecting characteristics nonparametrically.*The Review of Financial Studies, 33*(5), 2326–377

Gao, Y., Gao, Z., Hu, Y., Song, S., Jiang, Z., & Su, J. (2023). A Framework of Hierarchical Deep Q-Network for Portfolio Management. arXiv. Retrieved from https://arxiv.org/pdf/2003.06365

Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica, 37*(3), 424–438.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies 33*(5), 2227–2273.

Guha, S., N. Mishra, G. Roy, & O. Schrijvers. 2016. Robust random cut forest based anomaly detection on streams. In: *Proceedings of the 33rd International Conference on Machine Learning*, pp. 2712–2721. New York: JMLR

Guidolin, M., & Timmermann, A. (2007). Asset allocation under multivariate regime switching.*Journal of Economic Dynamics and Control 31*(11), 3503–3544.

Guidolin, M., & Timmermann, A. (2008). International asset allocation under regime switching, skew, and kurtosis preferences.*The Review of Financial Studies, 21*(2), 889–935.

Guidolin, M., Hyde, S., McMillan, D., & Ono, S. (2009). Non-linear predictability in stock and bond returns: When and where is it exploitable? *International Journal of Forecasting, 25*(2), 373–399.

Grinold, RC. (1989). The fundamental law of active management. *The Journal of Portfolio Management, 15*(3), 30–37.

Grinold, R. C., & Kahn, R. (2000). *Active portfolio management,* 2nd ed. New York, NY: McGraw-Hill.

Gupta, P., Mehlawat, M., & Mittal, G. (2012). Asset portfolio optimization using support vector machines and real-coded genetic algorithm. *Journal of Global Optimization, 53*(2), 1–19.

Ha, D., & Schmidhuber, J. (2018). Recurrent experience replay in distributed reinforcement learning.In: Proceedings of the 35th International Conference on Machine Learning (ICML-18), 1955–1964

Hall, K. M. (1970). An r-dimensional quadratic placement algorithm. *Management Science, 17*(3), 219–229.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica, 57*(2), 357–384.

Han, S. (2023). Risk budgeting portfolio optimization with deep reinforcement learning. *The Journal of Financial Data Science, 5*(4), 86–99.

Hasselt, H. V. (2010). Double Q-learning. *Advances in Neural Information Processing Systems (NeurIPS), 23*, 2613–2621.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Junfeng, W., Yaoming, L., Wenqing, T., & Yun, C. (Forthcoming). Portfolio management based on a reinforcement learning framework. Journal of Forecasting

Kim, C. J., & Nelson, C. R. (1998). Business cycle turning points, a new coincident index, and tests of duration dependence based on a dynamic factor model with regime switching. *The Review of Economics and Statistics, 80*(2), 188–201.

Koumou, G. (2024). Hierarchical risk budgeting. *The Journal of Financial Data Science, 6*(2), 35–53.

Kozak, S., Nagel, S., & Santosh, S. (2019). Shrinking the cross-section. *Journal of Financial Economics, 135*(2), 271–292.

Ledoit, O., & Wolf, M. (2004a). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis, 88*(2), 365–411.

Ledoit, O., & Wolf, M. (2004b). Honey i shrunk the sample covariance matrix. *The Journal of Portfolio Management, 20*(4), 110–119.

Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory, 28*(2), 129–137.

López de Prado, M. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management, 42*(4), 59–69.

López de Prado, M. (2018). *Advances in financial machine learning*. Hoboken, NJ: Wiley.

López de Prado, M. (2020). *Machine learning for asset managers*. Cambridge University Press.

López de Prado, M. (2022). Beyond econometrics: A roadmap towards financial machine learning. *The Journal of Financial Data Science, 4*(3), 1–21.

Marinov, S. (2019). Natural language processing in finance: Shakespeare without the monkeys. Man institute. Retrieved from https://www.man.com/maninstitute/shakespeare-without-the-monkeys

Markowitz, M. (1952). Portfolio selection. *Journal of Finance, 7*(1), 77–91.

Martínez-Barbero, X., Cervelló-Royo, R., & Ribal, J. (2024). Long short-term memory neural networks: Testing on upward and downward European. Computational Economics. TO COME

Menvouta, E., Serneels, S., & Verdonck, T. (2023). Portfolio optimization using cellwise robust association measures and clustering methods with application to highly volatile markets. *The Journal of Finance and Data Science, 9*, 100097.

Messmer, M., & Audrino, F. (2022). The lasso and the factor zoo: Predicting expected returns in the cross-section. *Forecasting, 4*(4), 969–1003.
Molyboga, M. (2020). A modified hierarchical risk parity framework for portfolio management. *The Journal of Financial Data Science, 2*(3), 128–139.
Nystrup, P., Lindström, E., & Madsen, H. (2020). Hyperparameter optimization for portfolio selection. *The Journal of Financial Data Science, 2*(3), 40–54.
Owen, S. (2023). An analysis of conditional mean-variance portfolio performance using hierarchical clustering. *The Journal of Finance and Data Science, 9*, 100112.
Papaioannou, G., & Giamouridis, D. (2020). Enhancing alpha signals from trade ideas data using supervised learning. In E. Jurczendo (Ed.), *Machine learning for asset management*. Wiley. Hoboken, NJ. 167-189.
Pinelis, M., & Ruppert, D. (2022). Machine learning portfolio allocation. *The Journal of Finance and Data Science, 8*, 35–54.
Potters, M., Bouchaud, J. B., & Laloux, L. (2005). Financial applications of random matrix theory: Old laces and new pieces. Working paper. Retrieved from https://arxiv.org/abs/physics/0507111
Punyaleadtip, K., Kridsda, P., & Kijsirikul, N. (2024). Black-Litterman portfolio management using the investor's views generated by recurrent neural networks and support vector regression. *The Journal of Financial Data Science, 6*(1), 85–105.
Raffinot, T. (2018). Hierarchical clustering-based asset allocation. *Journal of Portfolio Management, 44*(2), 89–99.
Rapach, D., Strauss, J., Tu, J., & Zhou, G. (2019). Industry return predictability: A machine learning approach. *The Journal of Financial Data Science, 1*(3), 9–28.
Rapach, D., & Zhou, G. (2013). Forecasting stock returns. In *Handbook of economic forecasting, 2*, Elsevier, 328-383.
Rosenberg, G. P., Haghnegahdar, P., Goddard, P. M., López de Prado, P., Carr, J., & Wu, J. (2016). Solving the optimal trading trajectory problem using a quantum annealer. *IEEE Journal of Selected Topics in Signal Processing, 10*(6), 1053–1060.
Rummery, G. A., & Niranjan, M. (1994). On-line Q-learning using connectionist systems. Technical report, Cambridge University Engineering Department
Sadon, A. N., Ismail, S., Jafri, N. S., & Shaharudin, S. M. (2021). Long short-term vs gated recurrent unit recurrent neural network for google stock price prediction. In: 2021 2nd international conference on artificial intelligence and data sciences (AiDAS) (Ipoh), 1–5
Sahu, B. R. B., & Kumar, P. (2024). Portfolio rebalancing model utilizing support vector machine for optimal asset allocation. *Arabian Journal for Science and Engineering*. https://doi.org/10.1007/s13369-024-08850-9
Santosa, F., & Symes, W. W. (1986). Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing, 7*(4), 1307–1330.
Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. Proceedings of the 32nd International Conference on Machine Learning (ICML-15), 1889–1897
Silva, N. F., Pedro de Andrade, W., Santos da Silva, M., de Melo, Kely, & Tonelli, A. O. (2024). Portfolio optimization based on the pre-selection of stocks by the support vector machine model. *Finance Research Letters., 61*, 105014.
Simonian, J. (2020). Modular machine learning for model validation: An application to the fundamental law of active management. *The Journal of Financial Data Science, 2*(2), 41–50.
Simonian, J. (2022). Forests for Fama. *The Journal of Financial Data Science, 4*(1), 145–157.
Simonian, J., & Fabozzi, F. J. (2019). Triumph of the empiricists: The birth of financial data science. *The Journal of Financial Data Science, 1*(1), 10–13.
Simonian, J., & Wu, C. (2019). Minsky vs. machine: New foundations for quant-macro investing. *The Journal of Financial Data Science, 1*(2), 94–110.
Simonian, J., López de Prado, M., & Fabozzi, F. J. (2018). Order from chaos: How data science is revolutionizing investment practice. *The Journal of Portfolio Management, 45*(1), 1–4.
Sims, C. A. (1972). Money, income, and causality. *American Economic Review, 62*(4), 540–552.
Sims, C. A. (1980). Macroeconomics and reality. *Econometrica, 48*, 1–48.
Singh, S. (2009). Portfolio risk management using support vector machine. *In Modeling Computation and Optimization*. https://doi.org/10.1142/9789814273510_0018
Sood, S., Papasotiriou, K., Vaiciulis, M., & Balch, T. (2023). Deep reinforcement learning for optimal portfolio allocation: A comparative study with mean-variance optimization. In: Proceedings of the International Conference on Automated Planning and Scheduling.
Spielman, D. A., & Teng, S. (2007). Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and Its Applications, 421*(2), 284–305.
Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.

Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM, 38*(3), 58–68.
Uysal, S., & Mulvey, J. (2021). A machine learning approach in regime-switching risk parity portfolios. *The Journal of Financial Data Science, 3*(2), 87–108.
Vapnik, V. N. (1997). Gerstner, Wulfram; Germond, Alain; Hasler, Martin; Nicoud, Jean-Daniel (Eds.) The support vector method. In Artificial Neural Networks—ICANN'97 (pp. 261–271). Berlin, Heidelberg: Springer
Wang, Z., Schaul, T., Hessel, M., Hasselt, H. V., Lanctot, M., & de Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. Proceedings of the 33rd International Conference on Machine Learning (ICML-16), 1995–2003
Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning, 8*(3–4), 279–292.
Yazdani, A. (2020). Derivation of a dynamic market risk signal using kernel PCA and machine learning. *The Journal of Financial Data Science, 2*(3), 73–85.
Zhang, H., Jiang, Z., & Su, J. (2021). A deep deterministic policy gradient-based Strategy for Stocks Portfolio Management. arXiv. Retrieved from https://arxiv.org/pdf/2103.11455