Estimating Expected Returns with Forecast Combinations

Estimación de los rendimientos esperados con combinaciones de previsiones

Robert Richter

Universidad Nacional de Colombia
Facultad de Ciencias Económicas
Bogotá, Colombia
September, 2021

Estimating Expected Returns with Forecast Combinations

Estimación de los rendimientos esperados con combinaciones de previsiones

Robert Richter

Tesis de grado presentada como requisito parcial para optar al título de:
Magister en Administración de Negocios (Universidad Europea de Viadrina)
Magister en Administración (Universidad Nacional de Colombia)

Directora:
Ph.D. Karoll Gomez Portilla

Línea de Investigación:
Seminario de Investigación II
Grupo de Investigación:
Teoría e Investigación aplicada en ciencias Económicas

Universidad Nacional de Colombia
Facultad de Ciencias Económicas
Bogotá, Colombia
September, 2021

For my beloved mother and sister, who were always there to support me.

A hundred times every day I remind myself that my inner and outer life are based on the labors of other men, living and dead, and that I must exert myself in order to give in the same measure as I have received and am still receiving.

Albert Einstein

Resumen

Esta tesis propone aplicar los pronósticos generados por la agregación de expertos como un novedoso predictor de los rendimientos esperados a 2 estrategias de portafolio diferentes: 1) Mean-Variance como propone Markowitz (1952) y 2) contracción de la matriz de covarianza S como en Ledoit and Wolf (2004). Los expertos se construyeron generando pronósticos con Quantile Regression de Generalized Random Forests y versiones automatizadas de Exponential Smoothing y ARIMA. Este estudio evalúa la precisión de los pronósticos de dos algoritmos de agregación de expertos 1) ML-Prod y 2) ML-Poly mediante un estudio de simulación, antes de aplicar el método superior a un portafolio diversificado. Después de evaluar la precisión de los pronósticos, se eligió el algoritmo superior ML-Poly para pronosticar los rendimientos esperados y mostró resultados prometedores fuera de la muestra para los portafolios considerados, devolviendo valores superiores para los parámetros de rendimiento seleccionados y resultados inferiores marginales en términos de ratio de rotación. Mediante el estudio de simulación, también se validaron los resultados de los portafolios.

Palabras clave: (Media-varianza, Shrinkage, Generalized Random Forests, árboles de decisión, ARIMA automatizado, Exponential Smoothing, agregación de expertos, optimización de portafolios).

Abstract

This thesis proposes to apply forecasts produced by expert aggregation as novel predictor of expected returns to 2 different portfolio strategies: 1) mean-variance as proposed by Markowitz (1952) and 2) shrinkage of the covariance matrix S as in Ledoit and Wolf (2004). Experts were built by generating forecasts with quantile regression as in generalized random forests and automatised versions of exponential smoothing and ARIMA. This study evaluates the predictive performance of two forecast combination algorithms 1) ML-Prod and 2) ML-Poly using a simulation study, before applying the superior method to a portfolio scenario. After evaluating prediction accuracy, the superior ML-Poly algorithm was chosen to forecast expected returns and showed promising out-of-sample results for the considered portfolios, returning superior values for the selected performance parameter and only marginal inferior results in terms of turnover ratio. Using the simulation study, the results of the portfolios were also validated.

Keywords: (Mean-Variance, Shrinkage, generalized random forests, decision trees, automatic ARIMA, exponential smoothing, expert aggregation, portfolio optimisation).

Contents

List of Figures

List of Tables

1. Introduction

In modern portfolio theory one of the main problem investors face, is the effective or optimal distribution of assets between different investments in order to achieve the highest possible return without taking too high a risk. Markowitz (1952) paved the way for modern portfolio theory by deriving the mean-variance optimization, which requires estimates for the mean vector and the covariance matrix of excess returns to solve the mean-variance quadratic optimization problem. While his work has been awarded the Nobel Price, it still revealed weaknesses determining expected returns $\hat{\mu}$ and providing a stable Covariance Matrix S. To stabilize mean-variance optimization and reduce the noise in covariance matrix estimators Jagannathan and Ma (2003), Ledoit and Wolf (2004) proposed to shrink S.

These approaches might have improved the portfolio's robustness, but still face challenges when evaluating them out-of-sample. DeMiguel, Garlappi and Uppal (2009) compared the performance of 14 different prominent portfolios to that of the 1/N strategy across seven empirical datasets of monthly returns concluding that in summary none of the various optimizing models consistently delivers a Sharpe ratio or a CEQ return higher than that of the naive portfolio, which also maintains a low turnover. Especially the estimation of μ is known to be more difficult and to have a larger impact on the portfolio weights (Merton, 1980), which concluding lead to a bad out-of-sample performance (Ban et al., 2018; DeMiguel, Garlappi, Nogales et al., 2009).

Due to the growth of financial markets and computational power becoming cheaper, forecasting time series has gained a lot of attention in recent years, with literature producing a variety of approaches. In this context, statistical learning is proven to be effective in improving descriptive, predictive and prescriptive analytics. Hence, it can help recognize patterns in big data, analyse consumer behaviour or simply make better forecasts of future stock prices (Schmidhuber, 2014). Considering estimation errors inherent in the sample expected returns $\hat{\mu}$, a more accurate estimate should lead to improved portfolio results.

One of the most common class of forecasting methods are exponential smoothing algorithms, on which some of the most successful methods are based on. Time series can be found in many different contexts including monthly stock prices, weekly sales of a product, monthly unemployment figures for a region, and quarterly imports of a country. These time series are often characterized by patterns such as upward/downward trends or seasonal variations.

Exponential smoothing algorithms exploit these patterns by estimating forecasts that are weighted combinations of past observations, with recent observations given relatively more weight than older observations (Hyndman et al., 2008).

Another well-known technique is ARIMA (p,d,q), which is used to forecast future equity returns based on historical data of the considered assets. Dong et al. (2020) analysed an automatized variation of the ARIMA algorithm and found that longer sample windows tend to capture a more complete spectrum of the industrial and business cycle by moderating the short-term noise and shocks in the capital market and therefore leads to satisfying forecasting accuracies. Moreover, they found that the degree of integration is mostly 1 for the equities and the time windows they tested, confirming the widely accepted belief that the market is partially efficient and asset prices largely follow random walk. Also, the auto-regression's order was typically 1 or 2, proofing the existence of momentum. The same could be noticed for the order moving average, showing the influence of market noise.

Also to be considered when estimating returns is their distribution, which can be exploited using quantile regression to reflect precise information about features and different points of their distribution function by computing various quantiles. A framework is included in random forests, an ensemble technique proposed by Breiman (2001) that combines various algorithms (e.g. boosting, linear models). They analyse a set of many individual base learners and construct weights for them to forecast new data points (Biau et al., 2008). Random forests are well suited to deal with large real-life tasks, since they can deal with small sample sizes and high-dimensional feature spaces and can easily be parallelized. One of the most recent algorithm known as generalized random forests proposed by Athey et al. (2019) abandons the idea of obtaining the final forecast by averaging estimates over each member of the ensemble and instead treats forests as adaptive nearest neighbour estimates.

While each of the aforementioned algorithms has advantages and disadvantages when adapting on structural breaks in the data, expert aggregation or forecast combinations have frequently been found to produce on average better forecasts than methods based on the individual model. Timmermann (2006) described expert aggregation as a diversification strategy that improves forecasting performance in the same manner as asset diversification leads to a better portfolio performance. Empirical evidence was delivered by Makridakis and Hibon (2000), who forecasted 3003 time series in the so-called M3-competition and found that on average the accuracy of combinations of various methods outperform, the specific models that are being combined.

Following the findings of Makridakis and Hibon (2000), recent expert aggregation algorithms were developed by Gaillard et al. (2016), who established expert-dependent regret bounds and time varying learning rates. Therefore, this thesis proposes to abandon the idea of using the traditional estimates for the expected returns of two different strategies 1) the mean-variance portfolio as proposed by Markowitz (1952) and 2) an extension to the model by shrinking the covariance matrix as in Ledoit and Wolf (2004) and replace them by a combination of forecasts generated by the three aforementioned models to utmost improve the out-of-sample performance.

This thesis is organized highlighting modern portfolio theory in chapter 2 including the estimation of optimal weights for both portfolio strategies. Chapter 3 details the theoretical framework for exponential smoothing, automatic ARIMA, quantile regression and expert aggregation based on the ML-Prod and ML-Poly algorithm by Gaillard et al. (2016). Chapter 4 highlights the methodology of how the forecasts are generated and applied to the portfolios. Chapter 5 first details a simulation study to evaluate the forecasting performance of the proposed expert aggregation algorithms and compares them to automatic ARIMA and exponential smoothing, before applying the superior ML-Poly algorithm to the simulated portfolios and a diversified dataset. Section 5.1 evaluates the out-of-sample portfolio performance of the two traditional approaches in comparison to models applying the chosen estimate as $\hat{\mu}$. For the analysis several performance parameter are determined such as annualized returns, Sharpe ratios, certainty equivalent, turnover and Omega ratio. Each parameter is validated using the simulation study and results are visualized with box-plots and confidence intervals are reported in the appendix table **C-1**. Chapter 6 summarizes the results obtained and discusses these findings.

2. Portfolio selection models

One of the simplest approaches to allocate wealth is investing naive throughout the portfolio. Ignoring all available data the weights are allocated $1/N$ with N being the number of assets and are then rebalanced each month. This approach is simple, but proven to outperform many optimization approaches due to estimation errors (DeMiguel, Garlappi, Nogales et al., 2009). Therefore, this simple but efficient investment strategy is included as additional benchmark.

2.1. Mean-variance optimization

In modern portfolio theory one of the central objectives are the optimal distribution of capital across various investments and the investigation of investment behaviour on the capital markets. Markowitz (1952) laid the foundation of modern portfolio theory by assuming that investors mainly focus on returns and the associated variance/risk. Thus, he proposed to optimize portfolios based on individual risk aversion resulting in the Global Minimum-Variance Portfolio (GMVP) and the Tangency Portfolios. His idea is commonly referred to as Capital Asset Pricing Model and has been further investigated focusing on the relationship between systematic risk and expected return by Lintner (1965), Mossin (1966), Sharpe (1970)).

One of the most discussed topics is the estimation of expected returns $\hat{\mu}$ and the covariance matrix $\hat{\sum}$ to optimize the portfolio. Assuming that historical returns are normally distributed Markowitz (1952) decided to use the means of historical returns to calculate $\hat{\mu}$ and the maximum likelihood estimator to approximate the sample covariance matrix S resulting in the following equations:

$$\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} R_t \tag{2-1}$$

$$S = \frac{1}{T}\sum_{t=1}^{T}(R_t - \hat{\mu})(R_t - \hat{\mu}) \tag{2-2}$$

These parameters are then used to optimize the desired portfolio weights. The GMVP only requires the sample variances to estimate a vector of weights minimizing risk, which is defined as

$$w_{MV} = \frac{\sum^{-1} 1}{1' \sum^{-1} 1} \tag{2-3}$$

In contrast, Tangency Portfolios additionally consider the investors risk aversion and maximize the return for a given level of risk. Their vector of weights is defined as

$$w_T = \frac{\hat{\sigma}_G^2}{\hat{\mu} - r_f} \hat{\sum}^{-1} \hat{\mu}^* \tag{2-4}$$

, where $\hat{\mu}^* = \hat{\mu} - r_f$.

The Portfolios then can be displayed along the efficient frontier with the turning point representing the GMVP, which also separates the efficient from inefficient part of the portfolios. To improve portfolio performance one might additionally introduce short sale constraints, both options are displayed in figure **2-1**.

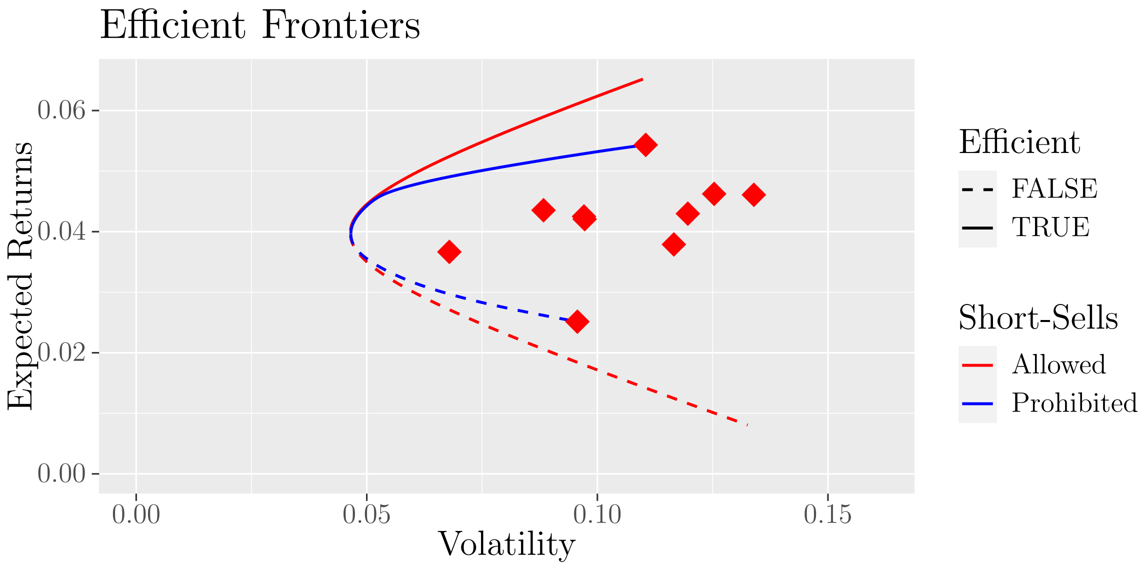

Figure 2-1.: Efficient frontier

Unfortunately this procedure is proven to be less reliable, since the random process that leads to the expected returns cannot be precisely determined (DeMiguel, Garlappi, Nogales et al., 2009). Therefore, the main challenge that portfolio optimization models face is the approximation of estimates, which is particularly noticeable when considering a high amount of assets (Ledoit and Wolf, 2004). Another problem is the tangency portfolio's tendency to assume extreme weights caused by these noisy estimates, which can have a serious impact on the portfolio's out-of-sample performance (DeMiguel, Garlappi and Uppal, 2009). The

GMVP is proven to be more robust due to only considering the variance to optimize the portfolio. Approaches to improve accuracy included constraining weights, shrinking the covariance matrix or considering additional factors.

2.2. Shrinkage of the covariance matrix

To minimize estimation error of the covariance matrix S Ledoit and Wolf (2004) proposed to shrink its extremes towards the centre. Their approach is designed to recognize the most extreme positive coefficients in S and pull them down while the extremely low estimated coefficients are pulled up. The two main questions that arise are: to which goal should one shrink and with which intensity?

The operational *shrinkage estimator* is defined as:

$$\hat{\sum}_{Shrink} = \hat{\delta}^{*}F + (1 - \hat{\delta}^{*})S \tag{2-5}$$

, where $\hat{\delta}^{*}$ is the optimal *shrinkage intensity*, F the *shrinkage target* and S the sample covariance matrix.

The *shrinkage target* F has to meet two requirements simultaneously. 1) It should consider only few degrees of freedom and 2) reflect important characteristics of the parameters to be forecasted. Ledoit and Wolf (2004) decided to use the constant-correlation model, which states that all the (pairwise) correlations are identical. Therefore, other models should be applied when a portfolio contains assets from different classes such as stocks and bonds.

The main challenge is to find an appropriate constant for the *shrinkage intensity* between 0 and 1, that minimizes the compromise of S and F. Hence, the goal is to estimate the minimum between covariance matrix and shrinkage estimator defined as δ^{*}. When choosing the optimal *shrinkage intensity* δ^{*} it is important to consider that shrinkage estimators analysed in (Frost and Savarino, 1986) break down when $N \geq T$ because their loss functions involve the inverse of S. Ledoit and Wolf (2004) proposed to estimate the optimal shrink intensity using a quadratic loss function based on the Frobenius norm approximating the difference between true and sample covariance matrix.

3. Estimating expected returns

From literature arose many different machine learning algorithms to estimate expected returns each with their own advantages and disadvantages, on the other hand methods that combine experts have been found in empirical studies to produce better forecasts on average than the forecasting models considered (Timmermann, 2006). This study proposes to combine experts applying the ML-Poly and ML-Prod algorithm by Gaillard et al. (2016) to further improve forecast accuracy. Experts are built using the automatic ARIMA and exponential smoothing algorithm by Hyndman and Khandakar (2008). Additionally, generalized random forests as in Athey et al. (2019) are constructed and forecasts are generated via quantile regression, where different quantiles include varying information on the distribution of returns. Finally, the forecasts produced by the different algorithms are combined to improve the overall forecast accuracy.

3.1. Exponential smoothing

Exponential smoothing describes a class of forecasting methods, on which some of the most successful algorithms are based on. While time series arise in varying contexts and industries, all have in common that they are often characterized by patterns such as upward/downward trends or seasonal variations. Exponential smoothing algorithms can exploit these characteristics by estimating forecasts that are weighted combinations of past observations, with recent observations given relatively more weight than older observations (Hyndman et al., 2008). They allow considerable flexibility in the specification of the parametric structure. Anderson (2012), Aoki and Havenner (1991), Hannan and Deistler (2012) proposed innovations formulations of the model, which are included in the Forecast package in R.

Each model, referred to as state space models, consists of a measurement equation to describe the observed data and some state equations to determine how unobserved components or states (level, trend, seasonal) change over time (Hyndman et al., 2008). Traditionally, exponential smoothing methods only produce point forecasts, while the underlying model additionally provides a framework for computing prediction intervals and other properties. In state space models the minimum mean squared error forecasts are the estimates from exponential smoothing. For each method exists a model with additive and multiplicative errors with similar point forecasts if the same smoothing parameter values are used, but will produce different prediction intervals.

Let y_t denote the observation at time t, and let x_t be a "state vector" containing unobserved components that describe the level, trend and seasonality of the series. A general linear innovations state space model can be written as

$$y_t = w(x_{t-1}) + r(x_{t-1}) + \epsilon_t, \tag{3-1}$$

$$x_t = F(x_{t-1}) + g(x_{t-1})\epsilon_t, \tag{3-2}$$

where ϵ_t is a white noise series and F, g and w are coefficients. Equation 3-1 describes the relationship between the unobserved states x_{t-1} and the observation y_t and equation 3-2 defines the evolution of states over time. To choose the best model, Hyndman et al. (2008) propose to use a penalized method based on the in-sample fit, since other accuracy measures such as the mean squared error (MSE) might suffer of too few out-of-sample errors. Applications of the automatic forecasting strategy showed that the proposed methodology is particularly good at short-term forecasting, and especially for seasonal short-term series, outperforming the other analysed methods.

In a first step in exponential smoothing the trend component is determined, which is a combination of a level term (ℓ) and a growth term (b). Future trend types then are estimated by combining the level and growth in various ways. Let T_h denote the forecast trend over the next h time periods, and let ϕ denote a damping parameter ($0 < \phi < 1$) (Hyndman et al., 2008). Concluding, the five trend types or growth patterns are defined:

None:	$T_h = \ell$
Additive:	$T_h = \ell + bh$
Additive damped:	$T_h = \ell + (\phi + \phi^2 + ... + \phi^h)b$
Multiplicative:	$T_h = \ell b^h$
Multiplicative damped:	$T_h = \ell b(\phi + \phi^2 + ... + \phi^h)$

A damped trend method is appropriate when there is a trend in the time series, but one believes that the growth rate at the end of the historical data is unlikely to continue more than a short time into the future Hyndman et al. (2008). These equations lead to dim the trend as the length of the forecast horizon increases, which often improves the forecasting accuracy. After a trend is chosen, a seasonal component, either additively or multiplicatively is introduced. Lastly, an error term is added, which is also additively or multiplicatively. Ignoring the error component leads to the following 15 exponential smoothing methods:

Table 3-1.: ETS methods

Trend component	Seasonal Component		
	N (None)	A (Additive)	M (Multiplicative)
N (None)	N,N	N,A	N,M
A (Additive)	A,N	A,A	A,M
A_d (Additive damped)	A_d,N	A_d,A	A_d,M
M (Multiplicative)	M,N	M,A	M,M
M_d (Multiplicative damped)	M_d,N	M_d,A	M_d,M

Cell (N,N) describes the simple exponential smoothing method, cell (A,N) describes Holt's linear method, and cell (A_d,N) describes the damped trend method. Holt-Winters' additive method is given by cell (A,A), and Holt-Winters' multiplicative method is given by cell (A,M). The other cells correspond to less commonly used but analogous methods. Considering the two different error terms (additive, multiplicative) results in two possible state space models for each method in **3-1**. Each model gives equivalent point forecasts when applying the same parameter values, but differs in their prediction intervals. Hence, there are 30 potential models described in this classification. The state space equations for each model of the ETS framework are summarized in table **A-1** appendix A.

3.2. ARIMA

Box and Jenkins (1970) developed the commonly known ARIMA model for forecasting, an extrapolation method that uses historical time series data to generate a forecast. An ARIMA model is expressed by three steps 1) identifying, 2) estimating and 3) diagnosing the underlying model. It combines an auto regressive model (AR) in the first part of the equation with a moving average model (MA) in its second part of the equation. The generalized form of the Autoregressive Integrated Moving Average (ARIMA) to fit non-seasonal data is given by

$$\phi(B)(1-B^d)y_t = c + \theta(B)\epsilon_t, \tag{3-3}$$

where ϵ_t denotes a white noise process with variance σ^2 and a mean of zero, B is the backshift operator and $\phi(.), \theta(.)$ are the polynomial orders of (p, q). According to Brockwell and Davis (2006) causality and invertibility are given by assuming that $\phi(.), \theta(.)$ have no roots for $|z| < 1$.

Equation 3-3 can also be described as a linear function of past values and errors, expressed in

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_p Y_{t-p} + \epsilon_t + \Theta_1 \epsilon_{t-1} + ... + \Theta_q \epsilon_{t-q}, \tag{3-4}$$

where Y_t is the forecasted value, which is expressed as a function of its own lag variables in the past time period along with the summation of its own random error term ϵ_t. In this context p and q are the respective auto regressive and moving average lags. The level at which Y_t becomes stationary is referred to as I (d) indicating the integration order. Hence, ARIMA models (p,d,q) take into account the lag of the dependent variable, the random error arising out of the estimation and order in which the variable becomes stationary, where the order p and q is identified by the auto correlating function (ACF) and partial autocorrelation function (PACF).

Hyndman and Khandakar (2008) state that the main task for automatic ARIMA forecasting is selecting an appropriate model order, that is the values p, q, d, P, Q, D. In contrast to the traditional estimation of p and q, the Automatic ARIMA algorithm as implemented in the Forecast package in R performs a step-wise procedure to optimize the model applying an information criterion. That is, if d and D are known, the orders p, q, P and Q can be chosen via an information criterion such as the AIC:

$$AIC = -2\log(L) + 2(p + q + P + Q + k), \tag{3-5}$$

where $k = 1$ if $c \neq 0$ and 0 otherwise, and L denotes the maximized likelihood of the model fitted to the differenced data $(1 - B^m)^D(1 - B)^d y_t$. Unfortunately, the full model's likelihood y_t is not defined and so the value of AIC for different levels of differencing are not comparable. For a non-seasonal time series Hyndman and Khandakar (2008) suggested to choose the KPSS unit-root test (Kwiatkowski et al., 1992).

1. The data is tested for a unit root

2. if the test result is significant, the differenced data is tested for a unit root

3. The procedure is stopped by when obtaining the first insignificant result

Illustration **3-1** shows the general procedure to estimate an ARIMA model on the left hand-side and displays the application of the automatic ARIMA function in R on the right hand-side.

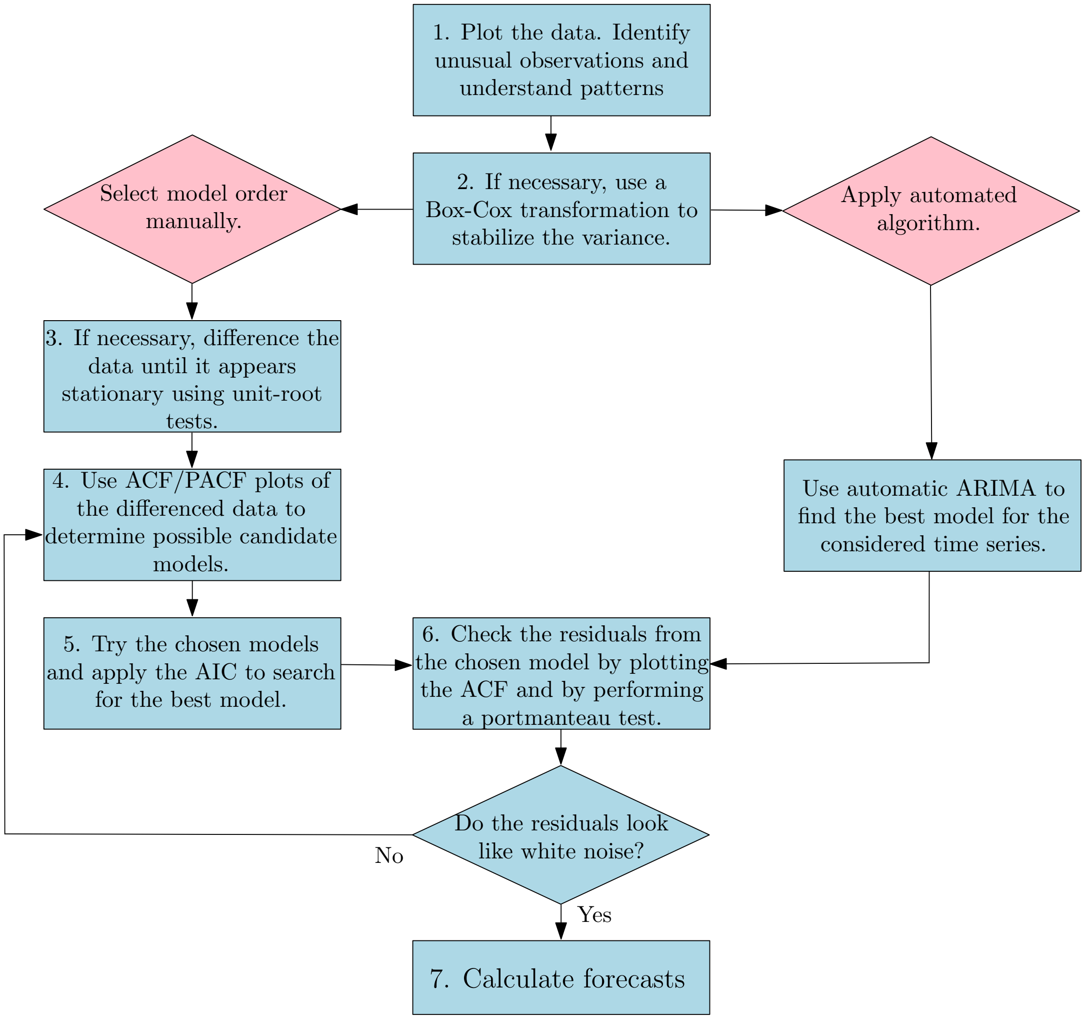
1. Plot the data. Identify unusual observations and understand patterns
2. If necessary, use a Box-Cox transformation to stabilize the variance.
Select model order manually.
Apply automated algorithm.
3. If necessary, difference the data until it appears stationary using unit-root tests.
4. Use ACF/PACF plots of the differenced data to determine possible candidate models.
Use automatic ARIMA to find the best model for the considered time series.
5. Try the chosen models and apply the AIC to search for the best model.
6. Check the residuals from the chosen model by plotting the ACF and by performing a portmanteau test.
Do the residuals look like white noise?
No
Yes
7. Calculate forecasts

Figure 3-1.: General process for forecasting with an ARIMA model

3.3. Generalized random forests

Breiman (2001) introduced random forests an algorithm used for statistical learning, which represents an efficient method for non-parametric conditional mean estimation. They are used given a data-generating distribution for $(X_i, Y_i) \in \mathcal{X} \times \mathbb{R}$ to estimate $\hat{\mu}(x) = \mathbb{E}[Y_i|X_i X_i = x]$. To determine any quantity identified via local moment conditions $\theta(x)$ Breiman (1996) defined 3-6 for given data $(X_i, O_i) \in \mathcal{X} \times \mathcal{O}$.

$$\mathbb{E}[\psi_{\theta(x),\nu(x)}(O_i)|X_i = x] = 0 \quad \text{for all} \quad x \in \mathcal{X}, \tag{3-6}$$

where $\psi(\cdot)$ is a scoring function and $\nu(x)$ an optional nuisance parameter. The forecast of a particular test point x is estimated by averaging forecasts across different trees (Amit and Geman, 1997; Breiman, 1996; Dietterich, 2000; Ho, 1998). Therefore, individual trees are grown by recursively adding axis-aligned splits to the tree, where each split is chosen to maximize the improvement to model fit (Breiman, 1984) also known as greedy recursive partitioning.

The estimates and their asymptotic behaviour have been studied by various statisticians considering confidence intervals (Wager and Athey, 2018), consistency (Arlot and Genuer, 2014; Biau, 2012; Biau et al., 2008; Denil et al., 2014; Lin and Jeon, 2006; Scornet et al., 2015; Wager and Walther, 2015) and second-order asymptotes (Mentch and Hooker, 2016). Regression forests efficiently stabilize forecasts due to their low bias, but high variance (Athey et al., 2019; Scornet et al., 2015). They are written as the average of B noisy tree-based estimates $\hat{\mu}_b(x), \hat{\mu}(x) = B^{-1} \sum_{b=1}^{B} \hat{\mu}_b(x)$ (Bühlmann and Yu, 2002). Since noisy solutions are generally biased, averaging would not improve the model. Another issue of generalizing forest-based methods is their dependency on whether the adaptive neighbourhood function obtained by partitioning adequately captures the heterogeneity in $\theta(\cdot)$ (Breiman, 2001).

One of the most recent approaches include generalized random forests, which were introduced by Athey et al. (2019). In standard classification or regression forests as proposed by Breiman (2001) the trees are randomized using bootstrap (or subsample) aggregation, whereby each tree is grown on a different random subset of the training data, and each variable is restricted by a random split selection, which is available at each step of the algorithm. Athey et al., 2019 treat forests as a type of adaptive nearest neighbour estimator, which makes the model more flexible when applying to statistical extensions and therefore, the idea of obtaining the final forecast by averaging estimates from each member of an ensemble as in Breiman (2001) can be abandoned.

To begin with, one has to estimate solutions for the equation 3-6, given n independent and identically distributed samples, indexed $i = 1, ..., n$. The observable quantity to each sample O_i encodes information relevant to estimating θ, along with a set of auxiliary covariates $X_i(x)$. For non-parametric regression, this observable is defined as $O_i = \{Y_i\}$ with $Y_i \in \mathbb{R}$ and just consists of an outcome, which tends to contain richer information (Athey et al., 2019). The functions $\theta(x)$ are estimated by defining similarity weights $a_i(x)$ that determine the relevance of fitting $\theta(\cdot)$ at x of the training example i. The target of interest is then fitted using the empirical refined version of estimating equation (Athey et al., 2019; Fan et al., 1998; Newey, 1994; Staniswalis, 1989; Stone, 1977; Tibshirani and Hastie, 1987).

However, when applying a forest algorithm as proposed in Breiman (2001) one might face computational limitations. The computation is typically intensive performing the split-selection step, so it's efficient implementation is crucial. Athey et al., 2019 suggested following procedure where the splits are in contrast only solved once per node. Each split starts with a parent node $P \subseteq \mathcal{X}$; given a sample of data $\mathcal{J}$: defined as $(\hat{\theta}_P, \hat{\nu}_P)(\mathcal{J})$ to be the solution to:

$$(\hat{\theta}_P, \hat{\nu}_P)(\mathcal{J}) \in \ \text{argmin}_{\theta,\nu} = \left\{ \left|\left| \sum_{i \in \mathcal{J}: X_i \in P} v_{\theta,\nu}(O_i) \right|\right|_2 \right\}. \tag{3-7}$$

P is then divided into the sub-nodes $C_1, C_2 \subseteq \mathcal{X}$ by an axis-aligned cut improving the θ-estimates utmost possible. The goal is to minimize $err(C_1, C_2) = \sum_{j=1,2} \mathbb{P}[X \in C_j | X \in P]\mathbb{E}[(\hat{\theta}_{C_j}(\mathcal{J}) - \theta(X))^2 | X \in C_j]$, where $\hat{\theta}_{C_j}$ is the parent-node of C_j.
Therefore, an approximate criterion $\tilde{\Delta}(C_1, C_2)$ is optimized by gradient-based estimates for $\hat{\theta}_{C_1}$ and $\hat{\theta}_{C_2}$. For the sub-nodes C applies $\tilde{\theta}_C \approx \hat{\theta}_C$ and they are estimated by 1) determining A_P as any consistent estimate for the gradient of function ψ

$$A_P = \frac{1}{|\{i : X_i \in P\}|} \sum_{\{i: X_i \in P\}} \nabla v_{\hat{\theta}_P, \hat{\nu}_P}(O_i). \tag{3-8}$$

The responding value is then inserted into equation 3-9, where $\hat{\theta}_P$ and $\hat{\nu}_P$ are determined by solving 3-7 once and 2) estimating $\tilde{\theta}_C$ as in:

$$\tilde{\theta}_C = \hat{\theta}_P - \frac{1}{|\{i : X_i \in C\}|} \sum_{\{i: X_i \in C\}} \xi^\top A_P^{-1} v_{\hat{\theta}_P, \hat{\nu}_P}(O_i), \tag{3-9}$$

where $\hat{\theta}_P$ and $\hat{\nu_P}$ are obtained by solving 3-6 once and ξ is a vector that chooses the θ-coordinate from the $(\theta,\, \nu)$ vector. To prepare the last step pseudo-outcomes are created by estimating $\hat{\theta}, \hat{\nu}$ and A_P^{-1}. This step is referred to as

$$\rho_i = \xi^\top A_P^{-1} v_{\hat{\theta}_P, \hat{\nu}_P}(O_i) \in \mathbb{R}. \tag{3-10}$$

The final step is to perform a CART regression split on the pseudo-outcomes p_i maximizing the criterion $\tilde{\Delta}(C_1, C_2)$. After executing the regression step, the observations in each sub-node are relabelled via 3-9 and proceed iteratively.

$$\tilde{\Delta}(C_1, C_2) = \sum_{j=1}^{2} \frac{1}{|\{i : X_i \in C_j\}|} (\sum_{|\{i: X_i \in C_j\}|} \rho_i)^2 \tag{3-11}$$

This approach includes other well-known machine learning algorithms, such as gradient boosting Friedman (2001) and the model-based recursive partitioning algorithm of Zeileis et al. (2008). Empirical results by Athey et al. (2019) show that estimation error from using the approximate criterion instead of the one proposed by Breiman (2001) are within statistical tolerance.

This framework by Athey et al. (2019) provides a flexible method for non-parametric estimation offering quantile regression, conditional average partial effect estimation and heterogeneous treatment effect estimation with instrumental variables.

3.3.1. Quantile regression

For Y a real-valued response variable and X a covariate or predictor variable. A general objective of statistical analysis is to find relations between X and Y. Classical regression analysis tries to estimate $\tilde{\mu}(x)$ of the conditional mean $E(Y|X = x)$ of the response variable Y for $X = x$ (Meinshausen, 2006). Therefore, traditional practise of forecasting the mean stock return assumes a squared loss function

$$L(e_{t+1}) = e_{t+1}^2 \tag{3-12}$$

, where $e_{t+1} = r_{t+1} - \tilde{f}_t$ is the forecast error and $\tilde{f}_t$ the forecast of return r_{t+1}. Considering this loss function, the optimal return estimate is the conditional mean. For the mean absolute error loss $L(e) = |e|$, the optimal forecast is the conditional median.

Unfortunately, the conditional mean only reflects one aspect of the distribution of Y. The conditional distribution function $F(y|X = x)$ is given by the probability that, for $X = x$, Y is smaller than $y \in \mathbb{R}$, $F(y|X = x) = P(Y \leq y|X = x)$.

For given $X = x$ the α-quantile for a continuous distribution function $Q_\alpha(x)$ is defined so that the probability of Y being smaller than $Q_\alpha(x)$ equals exactly α (Koenker, 2005).

$$Q_\alpha(x) = inf\{y : F(Y|X = x) \geq \alpha\} \tag{3-13}$$

Concluding, the quantiles can be applied to give more precise information about features and distribution of the forecasting variable Y than just considering the conditional mean Koenker (2005). To estimate the return using quantiles $\alpha \in (0, 1)$ Koenker and Bassett (1978) proposed the tick loss function

$$L_\alpha(e_{t+1}) = (\alpha - 1\{e_{t+1} < 0\})e_{t+1}. \tag{3-14}$$

Following the first order condition of 3-12 including the forecast $\tilde{f}_t$, the optimal estimate is determined via the conditional quantile $-\alpha + F(\tilde{f}_t) = 0$, where F is the distribution function of returns. Therefore, the optimal quantile depends on the distribution of returns $\tilde{f}_t = F^{-1}(\alpha)$.

3.4. Expert aggregation

To estimate forecasts by combining experts, a learner has to make sequential forecasts over a series of rounds by weighting each expert K (Cesa-Bianchi and Lugosi, 2006; Freund et al., 1997; Gaillard et al., 2016; Littlestone and Warmuth, 1994; Vovk, 1998). For each round $t = 1, ..., T$, the learner forecasts a value by choosing a vector $\boldsymbol{p}_t = p_{1,t}, ..., p_{K,t}$ of positive weights that sum up to one. In a next step the weights $p_{k,t}$ are assigned to each expert k and the weighted average is forecasted.

$$\hat{y}_t = \sum_k p_{k,t} x_{k,t} \tag{3-15}$$

Each expert's k loss $\ell_{k,t} \in [a, b]$ is then cumulated, resulting in the learner's loss $\hat{\ell}_t = \boldsymbol{p}_t^\top \boldsymbol{\ell}_t = \sum_{k=1}^K p_{k,t}\ell_{k,t}$, where $\boldsymbol{\ell}_t = (\ell_{1,t}, ..., \ell_{K,t})$. The learner then minimizes its cumulative loss by controlling his regret $R_{k,T}$ against each expert k, where $R_{k,T} = \sum_{t \leq T}(\hat{\ell}_t - \ell k, t)$. In a worst case scenario, the best bound guaranteed on the standard regret $R_{k,t}$ is of order $O(\sqrt{T \ln K})$ (Cesa-Bianchi and Lugosi, 2006).

Cesa-Bianchi et al. (2007) succeeded in improving the algorithm by providing second-order (variance-like) bounds on the regret, leading to two types of bounds, each with its own advantages and disadvantages. The first formulation in the form of

$$R_{k,t} \leq \frac{\ln K}{\eta} + \eta \sum_{t=1}^{T} \ell_{k,t}^2 \tag{3-16}$$

for all experts k, where $\eta \leq 1/2$ is a parameter of the algorithm (Gaillard et al., 2016). By optimizing η with knowledge of the losses, one would achieve the desired bound

$$R_{k,t} = O\left(\sqrt{\ln K \sum_{t=1}^{T} \ell_{k,t}^2}\right), \tag{3-17}$$

but there is no known method that achieves 3-17 for all experts k without mentioned hindsight. The second bound established by Cesa-Bianchi et al. (2007) is a uniform regret bound, having the drawback of not reflecting that it is harder to compete with some experts than with others. In consequence, Gaillard et al. (2016) proposed to aim for expert-dependent regret bounds and formulated a second-order bound of the form

$$R_{k,t} = O\left(\sqrt{\ln K \sum_{t=1}^{T} (\hat{\ell}_t - \ell_{k,t})^2}\right), \tag{3-18}$$

which holds for all experts k. Furthermore, they developed a variant of the Prod algorithm by Cesa-Bianchi et al. (2007) with two innovations. 1) The analysis for Prod is extended to multiple learning rates η_k similar to a variant of the Hedge algorithm with multiple learning rates as proposed by Blum and Mansour (2007). To prevent that standard tuning techniques for the learning rates lead to an additional $O(\sqrt{T \ln K})$ multiplicative factor, Gaillard et al. (2016) introduced learning rates $\eta_{k,t}$ that vary with time to convert this factor to $O(\ln \ln T)$, which they consider to be consistent.

Another approach is the application of polynomial potentials that can be useful to minimize regret as illustrated in (Cesa-Bianchi and Lugosi, 2003). Gaillard et al. (2016) based their ML-Poly algorithm on them with order $p = 2$. The bound is characterised by a poor dependency on the number of experts K. The adequate dependencies might be achieved by considering polynomial functions of arbitrary orders p (Cesa-Bianchi and Lugosi, 2003). For all sequences of loss vectors $\boldsymbol{\ell}_t \in [0,1]^K$, the cumulative loss of algorithm 4 is determined with learning rates defined as:

$$\eta_{k,t-1} = \frac{1}{1 + \sum_{s=1}^{t-1} (\hat{\ell}_s - \ell_{k,s})^2} \tag{3-19}$$

The opera package by Gaillard et al. (2016) in R provides a wide range of combination algorithms including other approaches. This thesis limits its investigation on applying the ML-prod and ML-poly algorithm to aggregate experts and evaluate the results afterwards.

4. Methodology

The analysis is done performing a rolling window study to generate allocation weights for the mean-variance and shrinkage portfolio. Traditionally, the data of the previous M months is used to estimate the parameters required to implement a particular strategy. Based on a T-month historical data set of returns an estimation window of $M = 120$ months is chosen to estimate the mean and the respective covariance matrices for each month t, starting with $t = M + 1$. These estimates are then applied on the respective functions to determine the relative portfolio weights of each strategy. Using the expected returns in month t and the portfolio weights in month M, the out-of-sample portfolio returns for month t are estimated. This process continues for each month $t + 1$ by determining the portfolio weights for month t, adding the return for the next period $t + 1$ in the data set and discarding the earliest return until the end of the data set is reached. The result is a series of monthly $T - M$ out-of-sample portfolio returns generated by each investment strategy applied.

4.1. A new combined estimate

As mentioned, the reliable estimation of $\hat{\mu}$ is quite challenging when following mean-variance optimization, especially when trying to maximize returns at a given risk level, the estimation error can lead to poor out-of-sample results (DeMiguel, Garlappi, Nogales et al., 2009). Especially affected by this observation is the mean-variance portfolio, which tends to produce extreme weights due to estimation error in the covariance matrix S, which can be improved by shrinking its extremes towards the centre as proposed by Ledoit and Wolf (2004) which minimizes estimation error.

Before aggregating experts different forecasts for the expected return have to be generated. The models to determine the estimate include a state space model for exponential smoothing, automatic ARIMA and generalized random forests to perform quantile regression and estimate a number of quantiles giving more information on the returns distribution and features by applying the proposed variations (Gaillard et al., 2016) of the Prod by Cesa-Bianchi et al. (2007) and the Poly algorithm in Cesa-Bianchi and Lugosi (2003) both with multiple learning rates. The forecasting accuracy of both aggregation models is compared via three statistical errors, before applying the superior estimate to a portfolio scenario.

4.1.1. State space framework

For exponential smoothing Hyndman et al. (2008) suggest to choose the best fitting algorithm by applying Akaike's Information Criterion (AIC), which is based on likelihood rather than one-step forecasts and therefore is able to select between the additive and multiplicative error models. The AIC is defined as:

$$AIC = L^*(\hat{\Theta}, \hat{x}_0) + 2q, \tag{4-1}$$

where q is the number of parameters in θ plus the number of free states in x_0, and $\hat{\Theta}$ and $\hat{x}_0$ denote the estimates of Θ and x_0. The model returning the lowest AIC is then chosen. The resulting algorithm can be described as:

1. For each series, all appropriate models are applied, optimizing the parameters of the model for each scenario.
2. According to AIC the best of the models is chosen.
3. Point forecasts are produced using the best model (with optimized parameters) for as many steps ahead as required.
4. Forecasting results for the best model are obtained either using the analytical results, or by simulating future sample paths for $\{y_{n+1}, ..., y_{n+h}\}$ and finding the $\alpha/2$ and $1-\alpha/2$ percentiles of the simulated data at each forecast horizon. If simulation is used, the sample paths can be generated using the Gaussian distribution for errors (parametric bootstrap) or using the resampled errors (ordinary bootstrap).

The algorithm is implemented using a rolling window of $M = 90$ months to estimate the next value $t + 1$, repeating this process until the end of the dataset is reached. To assure comparability between ARIMA and state space framework, the same methodology is applied for both algorithms.

4.1.2. Automatic ARIMA

The first as expert considered model is an automatized variation of the well-known ARIMA algorithm. Hyndman and Khandakar (2008) developed a method that automatically selects the best fit for the respective model and additionally includes a framework for a variety of exponential smoothing algorithms. They propose to apply a penalized method based on the in-sample fit, since accuracy measures as the mean squared error (MSE) might face issues creating a sufficiently large number of out-of-sample errors to draw reliable conclusions. The automatic ARIMA method performs a step-wise procedure to select the order of $(p, q)(P, Q)$ by applying Akaike's Information Criterion (AIC) and specify the degree of integration d

and D. Instead of directly minimizing the AIC to choose all of the parameters, which might lead to over-differencing. Hyndman and Khandakar (2008) propose to use unit root-tests to first estimate D and d and then proceed to select the values p and q by minimizing the AIC. They defined the resulting procedure as:

Step 1: Four possible models are tested to start with

- ARIMA$(2, d, 2)$ if $m = 1$ and ARIMA$(2, d, 2)(1, D, 1)$ if $m > 1$.
- ARIMA$(0, d, 0)$ if $m = 1$ and ARIMA$(0, d, 0)(0, D, 0)$ if $m > 1$.
- ARIMA$(1, d, 0)$ if $m = 1$ and ARIMA$(1, d, 0)(1, D, 0)$ if $m > 1$.
- ARIMA$(0, d, 1)$ if $m = 1$ and ARIMA$(0, d, 1)(0, D, 1)$ if $m > 1$.

If $d+D \leq 1$, these models are fitted with $c \neq 0$, otherwise $c = 0$. Of these four models, the one with the smallest AIC value is selected, referred to as "current" model. It is denoted by ARIMA(p, d, q) if $m = 1$ or ARIMA$(p, d, q)(P, D, Q)_m$ if $m > 1$

Step 2: Up to thirteen variations on the current model are considered:

- where one of p, q, P and Q is allowed to vary by ± 1 from the current model
- where p and q both vary by ± 1 from the current model;
- where P and Q both vary by ± 1 from the current model;
- where the constant c is included if the current model has $c = 0$ or excluded if the current model has $c \neq 0$.

Whenever a model with lower AIC is found, it becomes the new "current" model and the procedure is repeated. This process finishes when there cannot be found a model close to the current model with lower AIC.

To avoid issues with convergence or near unit-roots, several constraints on the fitted models are introduced:

- The values of p and q are not allowed to exceed the specified upper bounds of 5 in each case.

- The values of P and Q are not allowed to exceed the specified upper bounds of 2 in each case.

- Any model which is "close" to non-invertible or non-causal is rejected. Specifically, the roots of $\phi(B)\Phi(B)$ and $\theta(B)\Theta(B)$ are estimated. If either has a root that is smaller than 1.001 in absolute value, the model is rejected.

- If there are any errors arising in the non-linear optimization routine used for estimation, the model is rejected, since any model that is difficult to fit is probably not a good model for the data.

A valid model is guaranteed to be returned, because the model space is finite and at least one of the starting models will be accepted (the model with no AR or MA parameters). The algorithm produces then forecasts using the selected model. For its application a window length has to be chosen. While short windows might preserve the most recent momentum of returns, a longer window controls short-term noise and shocks in the capital market. Therefore, a rolling window of $M = 90$ Months is chosen to forecast the next value, repeating this process until the end of the dataset is reached. Considering the extensive analysis required to visualize the residuals of all ARIMA fits, this paper refrains from reviewing them.

4.1.3. Quantile regression

Athey et al. (2019) proposed an innovation of random forests by Breiman (2001) that can be used to fit any quantity of interest identified to a set of local moment equations. The method considers a weighted set of nearby training examples, but instead of using classical kernel weighting functions that are prone to dimensionality, an adaptive weighting function is suggested to better account for heterogeneity. They extended the underlying framework to develop new methods for quantile regression, conditional average partial effect estimation and heterogeneous treatment effect estimation. Especially, quantile regression has desirable features for making forecasts since different quantiles of interest contain more information on features and distribution of returns. Hence, for forecast combination it is useful to consider a range of quantiles, instead of focusing just on the mean or median.

Athey et al. (2019) described the generalized random forest algorithm, which predefines all tuning parameters such as the number of trees B to 2000 and the sub-sampling rate $\mathbf{s}$ used in *Subsample* as:

Algorithm 1 Generalized random forests with honesty and subsampling

```
Procedure: GeneralizedRandomForest (set of examples $\mathcal{S}$, test point $x$)
weight vector $\alpha \leftarrow Zeros(|\mathcal{S}|)$ for $b = 1$ to total numbers of trees $B$ do
set of examples $I \leftarrow Subsample(\mathcal{S}, \mathbf{s})$
sets of examples $\mathcal{J}_1, \mathcal{J}_2 \leftarrow Splitsample(\mathcal{I})$
tree $\mathcal{T} \leftarrow GradientTree(\mathcal{J}_1, \mathcal{X})$ $\triangleright$ See the GradientTree algorithm 2
$\mathcal{N} \leftarrow Neighbors(x, \mathcal{T}, \mathcal{J}_2)$ $\triangleright$ Returns those elements of $J_2$ falling into the same leaf as $x$ in tree $\mathcal{T}$
for all example $e \in \mathcal{N}$ do
$\alpha[e] += 1/|\mathcal{N}|$
output $\hat{\theta}(x)$, the solution to the GradientTree algorithm with weights $\alpha/B$
```

The function *Zeros* creates a vector of zeros of length $|\mathcal{S}|$; *Subsample* draws a subsample of size $\int$ from $\mathcal{S}$ without replacement; and *SplitSample* randomly divides a set into two evenly-sized, non-overlapping halves (Athey et al., 2019).

Since the computation of growing trees is typically dominated by the split-selection step, it is critical for this step to be designed as efficient as possible. The authors follow other popular statistical algorithms by choosing a gradient-based approximation, that includes gradient boosting (Friedman, 2001) and the model-based recursive partitioning algorithm (Zeileis et al., 2008), leading to the formulation of the following gradient tree algorithm:

Algorithm 2 Gradient Tree

```
Gradient trees are grown as subroutines of a generalized random forest.
Procedure: GradientTree (set of examples $\mathcal{J}$, domain $\mathcal{X}$)
node $P_0 \leftarrow$ CreateNode($\mathcal{J}, \mathcal{X}$)
queue $\mathcal{Q} \leftarrow$ InitializeQueue $P_0$ while NotNull(node $P \leftarrow$ Pop($\mathcal{Q}$)) do
($\hat{\theta}_P, \hat{\nu}_P, A_P$) $\leftarrow$ SolveEstimatingEquation($P$) ▷ Calculates equations 3-7 and 3-8.
vector $R_P \leftarrow$ GetPseudoOutcomes($\hat{\theta}_P, \hat{\nu}_P, A_P$) ▷ applies equation 3-10 over $P$.
split $\Sigma \leftarrow$ MakeCartSplit($P, R_P$) ▷ optimizes equation 3-11.
if SplitSucceeded($\Sigma$) then
SetChildren($P$, GetLeftChild($\Sigma$), GetRightChild($\Sigma$))
AddToQueue($\mathcal{Q}$ GetLeftChild($\Sigma$))
AddToQueue($\mathcal{Q}$ GetRightChild($\Sigma$))
output tree with root node $P_0$
```

The function *InitializeQueue* initializes a queue with a single element; *Pop* returns and removes the oldest element of a queue $\mathcal{Q}$, unless $\mathcal{Q}$ is empty in which case it returns null. *MakeCartSplit* runs a CART split on the pseudo-outcomes, and either returns two child nodes or a failure message that no legal split is possible (Athey et al., 2019).

The generalized random forest is constructed to exploit cross-information of assets. Therefore, the forests are constructed containing a lagged matrix of all the 40 assets initially considered in **4-3** to explain the remaining 10 stocks in the portfolio. 9 different Quantiles from 0.1 to 0.9 are estimated for each considered asset. A sample of $M = 120$ months is chosen to train the algorithm and generate out-of-sample forecasts for 247 months.

4.1.4. Forecast combination

Lastly, the forecasts generated by ARIMA, quantile regression and the state space framework are combined using the two aforementioned approaches. Figure **4-1** illustrates the aggregation process, with expert advice as inputs to a decision maker, who in turn yields a response.

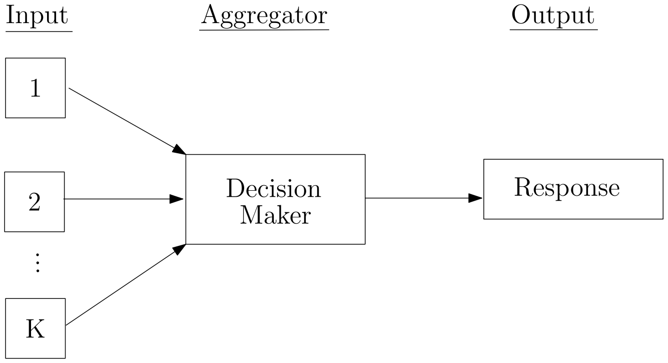

Figure 4-1.: Aggregation of experts

Gaillard et al. (2016) developed innovations of two forecast combination algorithms 1) the ML-Prod by Cesa-Bianchi et al. (2007) and 2) the polynomially weighted average algorithm in Cesa-Bianchi and Lugosi (2003). For 1) they introduce new second-order regret bounds in terms of excess losses, which denote the differences between instantaneous losses suffered by the algorithm and the ones passed by each model, also referred to as experts.

```
Algorithm 3 Prod with multiple learning rates
Parameters: a vector $\eta = (\eta_1, ..., \eta_k)$ of positive learning rates
Initialization: a vector $\omega_0 = (\omega_{1,0}, ..., \omega_{k,0})$ of non-negative weights that sum to 1
For each round $t = 1, 2, ...$
    1. form the mixture $\boldsymbol{p}_t$ defined component-wise by $p_{k,t} = \eta_k \omega_{k,t-1} / \boldsymbol{\eta}^\top \boldsymbol{\omega}_{t-1}$
    2. observe the loss vector $\boldsymbol{\ell}_t$ and incur loss $\hat{\ell}_t = \boldsymbol{p}_t^\top \boldsymbol{\ell}_t$
    3. for each expert $k$ perform the update $\omega_{t,k} = \omega_{k,t-1}(1 + \eta_k(\hat{\ell}_t - \ell_{k,t}))$
```

The second algorithm uses polynomial potentials to minimize the regret with order $p = 2$. Gaillard et al. (2016) state that its bound has the same weak dependency on the number of experts K and on T as the other algorithm. Following Cesa-Bianchi and Lugosi (2003) the right dependencies might be achieved by considering polynomial functions of arbitrary orders p.

Forecasts with both algorithms are generated and then compared with each other and, also with the forecasts generated by ETS and automatic ARIMA. For their comparison various statistical errors are included such as the mean forecast error (ME), the root-mean square deviation (RMSE) and the mean absolute error (MAE). The better performing forecaster is then applied on the respective portfolio strategies. Since the model's estimates will return noisier data than the traditional forecasts and consequently would lead to a high turnover ratio, all the portfolios are constrained, so that their minimum allocation is 0.09 and their

```
Algorithm 4 Polynomially weighted averages with multiple learning rates
Parameter: a rule to sequentially pick positive learning rates $\eta = (\eta_{1,t}, ..., \eta_{k,t})$
Initialization: the vector of regrets with each expert $\boldsymbol{R}_0 = (0, ..., 0)$
For each round $t = 1, 2, ...$
    1. pick the learning rated $\eta_{k,t-1}$ according to the rule
    2. form the mixture $\boldsymbol{p}_t$ defined component-wise by $p_{k,t} = \eta_{k,t-1}(R_{k,t})_+/\boldsymbol{\eta}_{t-1}^\top(\mathbf{R}_{t-1})_+$ where
       $\boldsymbol{x}_+$ denotes the vector of the non-negative parts of the components $\boldsymbol{x}_+$
    3. observe the loss vector $\boldsymbol{\ell}_t$ and incur loss $\hat{\ell}_t = \boldsymbol{p}_t^\top \boldsymbol{\ell}_t$
    4. for each expert $k$ update the regret: $R_{k,t} = R_{k,t-1} + \hat{\ell}_t - \ell_{k,t}$
```

maximum 0.11. The long estimation windows and the strict constraints should lead to stable portfolio weights for all strategies K with low turnovers to minimize involved trading.

4.2. Stock selection

For an optimized scenario, assets that preferably are uncorrelated and historically perform well should be chosen in order to assure the best possible portfolio performance. To generate an overview of the S&P500 and be able to appropriately filter the data an analysis of the whole index is carried out. First, the monthly data from 1990-01-01 to 2020-08-01 of all assets included in the index is downloaded from Yahoo Finance using the quantmod package in R.

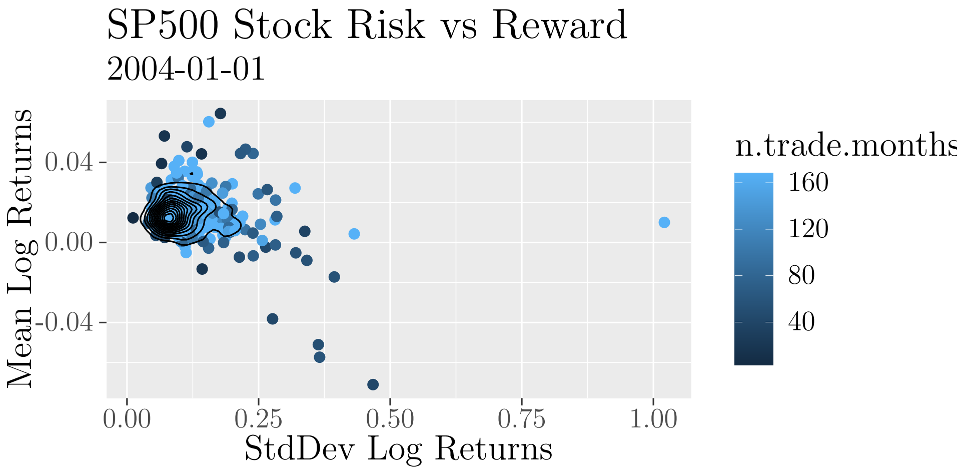

Figure 4-2.: S&P500 analysis

Next, to avoid generating in-sample hindsight the data is filtered for Date $< 2004-01-01$

and each asset's mean log return and standard deviation are determined. The results are visualized in figure **4-2** showing the relation of risk and reward of all stocks to determine appropriate filters to pursue an optimized stock selection. The density shows that the majority of assets finds itself with mean log returns below 0.2 and a standard deviation of below 0.125, while many having at least 120 trading months up to the selected date. Based on these finding the data is filtered for $Date < 2004 - 01 - 01$, 120 minimum trading months and a standard deviation of 0.15. Next, the assets are ranked by their mean log return and limited to 40, leaving numerous stocks.

Then, the remaining 40 assets are hierarchical clustered into 10 groups. Due to advantages when identifying small clusters, an agglomerative clustering algorithm known as AGNES is applied, which includes various clustering techniques (Landau and Chis Ster, 2010).

1. **Complete linkage clustering:** It estimates all pairwise dissimilarities between the elements in cluster 1 and 2, and identifies the largest value of these dissimilarities as the distance between the two clusters. It tends to produce more compact clusters.

2. **Single linkage clustering:** It estimates all pairwise dissimilarities between the elements in cluster 1 and 2, and identifies the smallest of these dissimilarities as a linkage criterion. It tends to produce long, "loose" clusters.

3. **Average linkage clustering:** It estimates all pairwise dissimilarities between the elements in cluster 1 and 2, and identifies the average of these dissimilarities as the distance between the two clusters.

4. **Ward's minimum variance method:** It minimizes the total within-cluster variance. At each step the pair of clusters with the minimum between-cluster distance are merged.

To identify which algorithm measures more accurately the dissimilarities between clusters of observations, the agglomerative coefficient is applied to select the best clustering method. The parameter identifies stronger clustering structures, with values closer to 1 representing a better fit. The following table shows the coefficients returned for each clustering algorithm.

Complete	Single	Average	Ward
0.5293	0.2625	0.4071	0.7066

The ward algorithm returns the best coefficient with 0.7066 and is therefore applied. Figure **4-3** visualizes the resulting hierarchical cluster structure for all 40 assets initially considered. From each group the best performing stock is automatically chosen, assuring most possible portfolio diversification.

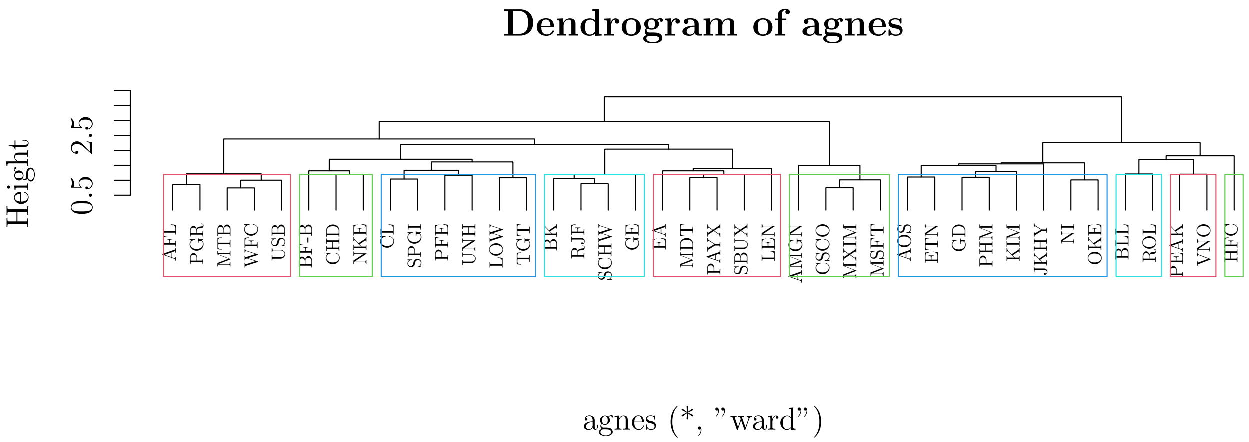

Figure 4-3.: Dendrogram of S&P500

The resulting portfolio of 10 assets is summarized in table **4-1**, giving information on their industry sector, rank, mean log return, standard deviation of log returns cluster group.

Table 4-1.: Stock choice

Symbol	Sector	Rank	Mean.log.return	Sd.log.return	Cluster
BLL	Materials	1	0.0409	0.0985	1
AOS	Industrials	2	0.0401	0.1240	2
BF-B	Consumer Staples	3	0.0380	0.0896	3
HFC	Energy	4	0.0356	0.1090	4
JKHY	Information Technology	5	0.0352	0.1330	5
NKE	Consumer Discretionary	6	0.0351	0.1210	6
CSCO	Information Technology	8	0.0342	0.1340	7
PFE	Health care	10	0.0313	0.0841	8
MXIM	Information Technology	14	0.0270	0.1320	9
GE	Industrials	27	0.0245	0.0729	10

Assets of different industry sectors with levels of risk (standard deviation) varying between 0.0729 and 0.1340 are used. The best performing asset has a monthly mean return of 0.0409 and the worst 0.0245. The data available starts in 1990-02-01 and ends in 2020-08-01, resulting in 367 monthly returns or 247 out-of-sample periods considering the 120 months needed to estimate the parameters to optimize the traditional portfolio models.

4.3. Portfolio performance parameter

Before analysing each portfolio's performance, the forecasting accuracy of the proposed algorithm is evaluated. For performance evaluation of each portfolio 5 indicators are computed 1) out-of-sample returns, 2) Sharpe ratio 3) certainty equivalent return, 4) Turnover and 5) Omega Ratio. A vector for each performance parameter saves the out-of-sample results to each strategy k.

4.3.1. Sharpe ratio

The Sharpe ratio is one of the most used parameters when evaluating the portfolio's performance, since it represents a relationship between returns and variance. The parameter is defined as

$$SR_k = \mu_k - r_f \sigma_k, \tag{4-2}$$

where μ_k is the return generated by strategy k and σ_k its standard deviation (Sharpe, 1964).

4.3.2. Certainty equivalent

The certainty equivalent (CEQ) is a guaranteed return that an investor would accept now instead of taking advantage of the chance of a higher but uncertain return in the future. In other words, the security equivalent is the guaranteed amount of money that a person sees as desirable as a risky asset (DeMiguel, Garlappi, Nogales et al., 2009). The CEQ return is estimated as the risk-adjusted rate of return minus the risk-free rate and is defined for strategy k as

$$C\hat{E}Q_k = \hat{\mu}_k - \frac{y}{2}\hat{\sigma}_k^2, \tag{4-3}$$

where $\hat{\mu}_k$ is the mean and $\hat{\sigma}_k^2$ the variance of out-of-sample excess returns generated by strategy k and y is defined as risk aversion and following common practice set to $y = 1$ (DeMiguel, Garlappi, Nogales et al., 2009).

4.3.3. Turnover ratio

Turnover is defined as the percentage of a portfolio that is sold in a particular month or year. To get a feel for the amount of trading that is required to implement each portfolio strategy, the relative turnover is calculated, i.e. the sum of the absolute value of the trading volume multiplied by 1 by the number of months (DeMiguel, Garlappi, Nogales et al., 2009). Under realistic conditions a high turnover leads to high transaction costs, which is not desirable. The turnover is defined as

$$Turnover = \frac{1}{T-M}\sum_{t=1}^{T-M}\sum_{j=1}^{N}(|\tilde{W}_{k,j,t+1} - \tilde{W}_{k,j,t}|), \tag{4-4}$$

where $\tilde{W}_{k,j,t}$ is the portfolio weight in asset j at time t before rebalancing under strategy k; $\tilde{W}_{k,j,t+1}$ is the portfolio weight after the realignment at $t+1$. For example, in the case of the naive diversification, $\tilde{W}_{k,j,t} = \tilde{W}_{k,j,t+1} = 1/N$, but $\tilde{W}_{k,j,t}$ may differ between t and $t+1$ due to changes in asset prices (DeMiguel, Garlappi, Nogales et al., 2009).

4.3.4. Omega ratio

The Omega ratio can be compared to the Sharpe ratio, but instead of considering only the first two moments of the return distribution, it considers all given moments.

$$\Omega(\theta) = \frac{\int_{\theta}^{\infty}[1-F(r)]dr}{\int_{-\infty}^{\theta}F(r)dr}, \tag{4-5}$$

where F is the cumulative probability distribution function of the returns and θ denotes the target return threshold, defining gains vs losses. A larger ratio is interpreted as that the portfolio provides more gains relative to losses for θ and consequently would be preferred.

5. A simulation experiment

This section highlights a simulation experiment to review the forecasting accuracy of the considered expert aggregation algorithms and select the superior in a portfolio context. Therefore, the data included in the $S\&P500$ is first filtered for months > 340 and a standard deviation of < 0.15 resulting in $n = 247$ of 500 possible assets. The data is then randomly sampled for 40 time series and eventual NA's are removed to construct the generalized random forests using 120 months as training data. Following, 10 stocks are sampled from the remaining 40 assets to construct random portfolios, resulting in data sets of 348 months each. This sampling procedure is repeated 50 times simulating 50 different portfolios. Next, the previously described methodology is applied to generate forecasts with exponential smoothing, automatic ARIMA and quantile regression, which are then used to aggregate experts by applying the ML-Prod and ML-Poly algorithm. Forecasts of exponential smoothing, ARIMA, ML-Prod and ML-Poly are then statistically evaluated and compared to provide insight on their forecasting accuracy. After determining which algorithm performs on average the best, forecasts generated with the superior model are used to optimize the respective portfolio strategies for each month M using the simulated portfolios and the optimal diversified dataset described in section 4.2 to validate if the improved estimate leads to a better out-of-sample portfolio performance.

Table 5-1.: Arithmetic means for forecast accuracy of ARIMA, ETS, ML-Prod and ML-Poly

Forecast Model	Error statistics		
	ME	RMSE	MAE
ETS	0.0063	0.0328	0.0246
ARIMA	0.0048	0.0367	0.0245
ML-Poly	0.0008	0.0365	0.0226
ML-Prod	0.0234	0.0455	0.0364

Table **5-1** reports the arithmetic means for ME, RMSE and MAE obtained for the simulated datasets. Exponential smoothing and automatic ARIMA perform on-par with a marginal difference for the reported mean error statistics. The ML-Prod algorithm does not appear to further improve forecasting accuracy, in contrast to the ML-poly algorithm, that reports superior means for ME with 0.0008, RMSE with 0.0365 and MAE with 0.0226.

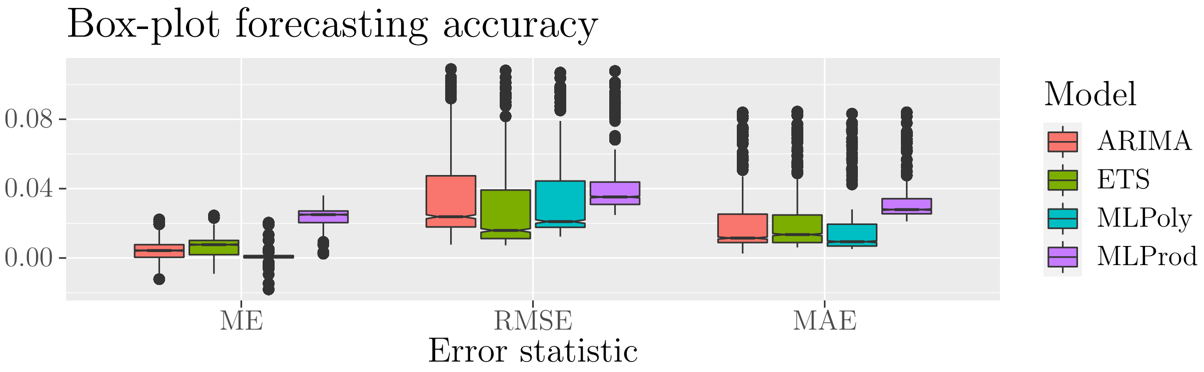

Figure 5-1.: Box-plot forecasting performance

A more detailed insight on forecasting accuracy is visualized in the box-plot **5-1**, which reveals that for all sampled time series every considered algorithm generates precise forecasts. As for expert aggregation with the ML-Prod algorithm forecasting accuracy is not further improved showing inferior quartiles for all three statistics. On the other hand, polynomial potentials appear to be well-suited for this type of expert aggregation with superior accuracy measurements for ME and MAE when reviewing the quartiles and medians, but with a wider interquartile range for the RMSE and superior median when compared to the ML-Prod algorithm. Full statistics including confidence intervals are reported in the appendix **B-1**.

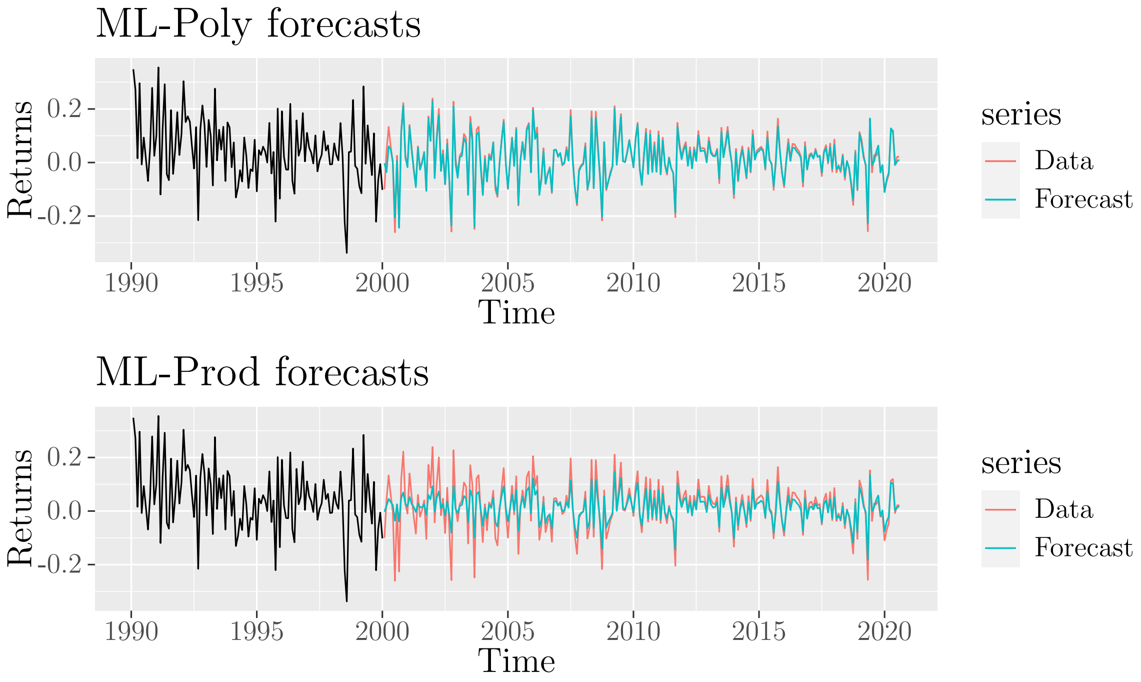

Figure 5-2.: Forecasts of expert aggregation for the HollyFrontier Corporation

The two graphs in figure **5-2** display the forecasts generated by the two expert aggregation algorithms for the HollyFrontier Corporation (HFC), which has a mean log return of 0.0356 and standard deviation of 0.109. The black line shows the estimation window of asset returns with the red line showing the out-of-sample period and each model's forecasts are expressed with the turquoise line.

The figures show that both algorithms improve with increasing number of iteration, which is especially notable for the ML-Prod algorithm, which requires a longer period of time in comparison to the polynomial potentials that seem to fit the data with high accuracy in few iterations. Hence, both should lead to a superior portfolio performance in comparison to the traditional estimates of expected returns.

Next, the on experts allocated weights by each forecast combination algorithms for the optimized dataset are visualized in figure **5-3**, with the left graph displaying weights over time generated by the ML-Prod algorithm and the other illustrating weights estimated with polynomial potentials.

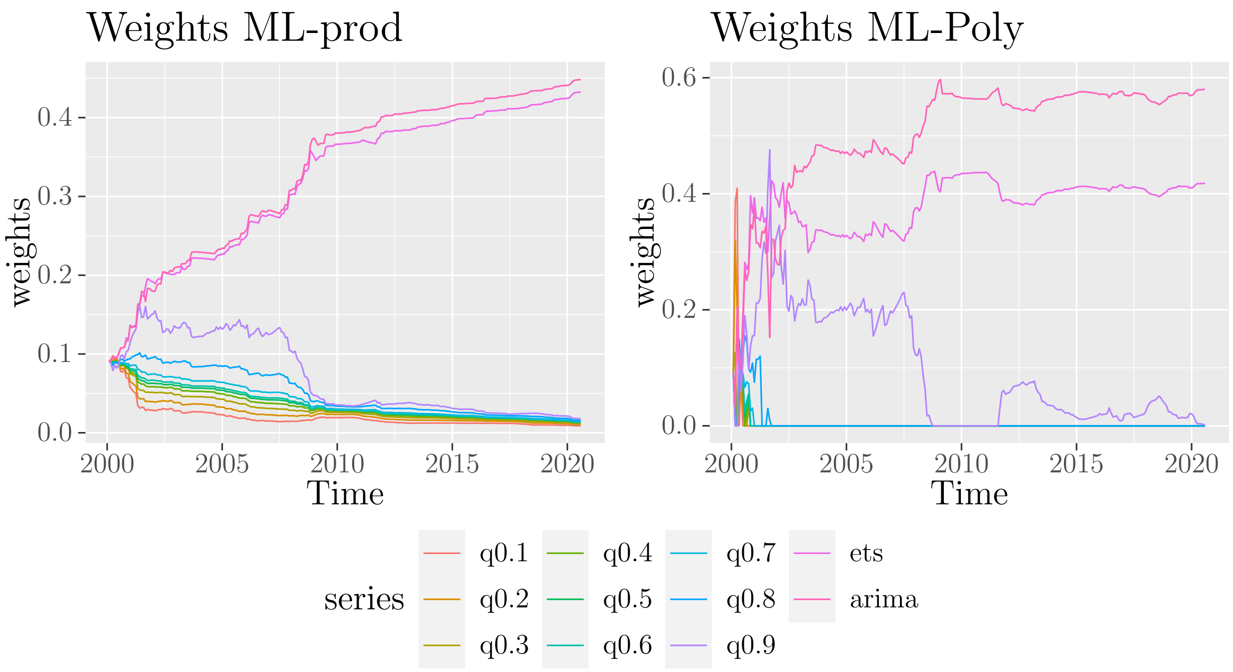

Figure 5-3.: Weights of experts

Lines named q0.x denote the weights on forecasts produced by the different quantiles, arima and ets refer to the automatised versions of these models. Both plots show that the algorithms mainly apply weights on the forecasts produced by automatic ARIMA and exponential smoothing. While the ML-Prod algorithm over time decreases the weights on quantiles, the ML-Poly algorithm is more straightforward but with heavier fluctuating weights on the experts. In this study the superior model is applied to the portfolio strategies, which is following this analysis the ML-Poly algorithm. Additionally, the statistical errors for the 10 time series included in the optimal diversified dataset are reported in table **B-2** appendix B.

5.1. Portfolio evaluation

After evaluating the forecasting performance of each algorithm, the rolling window study previously described is applied, to optimize the presented optimizations strategies K and the results are saved in a data frame of portfolio returns. Using this data, annualized returns, Sharpe ratios, CEQ, turnover and Omega ratios of each strategy K are calculated. Table **5-2** reports the returned parameters for the optimal diversified dataset, where Tang denotes the classical mean-variance portfolio and Shrink its extension by shrinking the covariance matrix. Portfolios ending with .comb are the innovations that implement the proposed forecast combination as expected return. A first look at the table indicates a superior performance for models that include the proposed aggregated expert as $\hat{\mu}$ in terms of returns, Sharpe ratio, CEQ and Omega ratio.

Table 5-2.: Portfolio parameter summary for diversified dataset

Portfolio	Cumulated return	Annualized return	Sharpe ratio	CEQ return	Turnover ratio	Omega ratio
Naive	2.6171	0.1200	0.1307	0.0095	0.0526	1.8123
Tang	2.6484	0.1218	0.1322	0.0096	0.0532	1.8261
Shrink	2.6349	0.1211	0.1318	0.0096	0.0528	1.8237
MV.comb	3.0237	0.1411	0.1478	0.0111	0.0684	1.9385
Shr.comb	3.0307	0.1415	0.1481	0.0111	0.0682	1.9407

Although, the turnover ratios for .comb-models are marginal higher and therefore hypothetically involve more transaction costs, one can argue that their superior returns outweigh the additional expenses. Results from the simulation study are summarized in appendix C table **C-1** including confidence intervals and medians.

5.1.1. Portfolio returns

A first impression of each portfolio's performance is provided by the cumulative returns of each strategy, which are shown in figure **5-4**. The graph clearly shows that the proposed forecaster leads to an improvement in comparison to the returns generated by the traditional portfolios. Introduced constraints on the portfolio weights and the diversification performed to obtain the dataset stabilise both traditional optimization approaches, so that they achieve similar returns when compared to the naive portfolio and furthermore, react well to capital market shocks. In this context, the mean-variance based strategies generate marginal superior annualized returns than the naive portfolio which generates 0.1200.

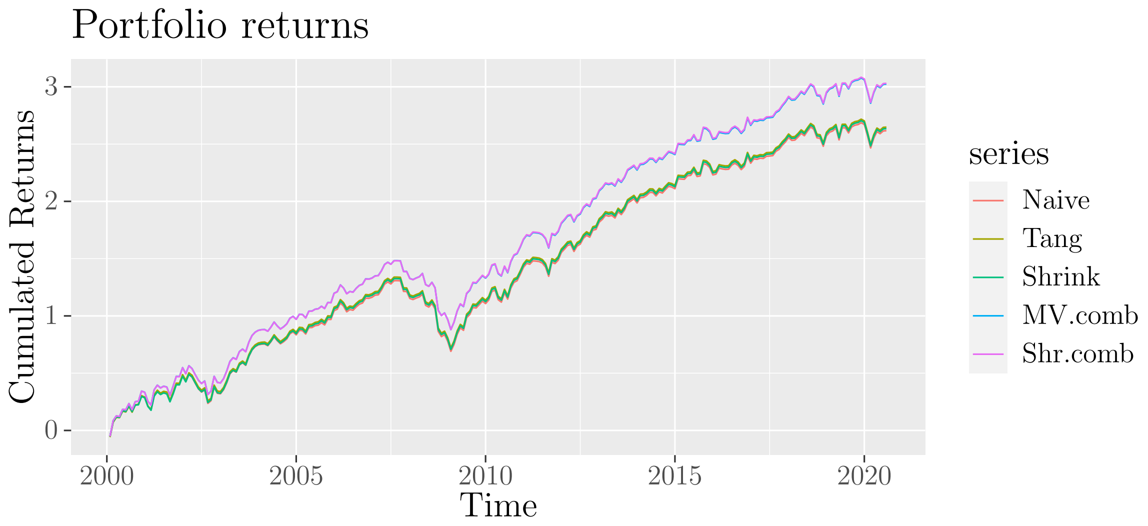

Figure 5-4.: Cumulated portfolio returns

On the other hand, the proposed combined expert leads to an improved performance for both considered portfolios. Comparing them mean-variance and shrinkage portfolio seem to perform equally differing only marginal in their generated returns. In detail, the MV.comb with 0.1411 is marginally outperformed by the shrinkage with 0.1415.

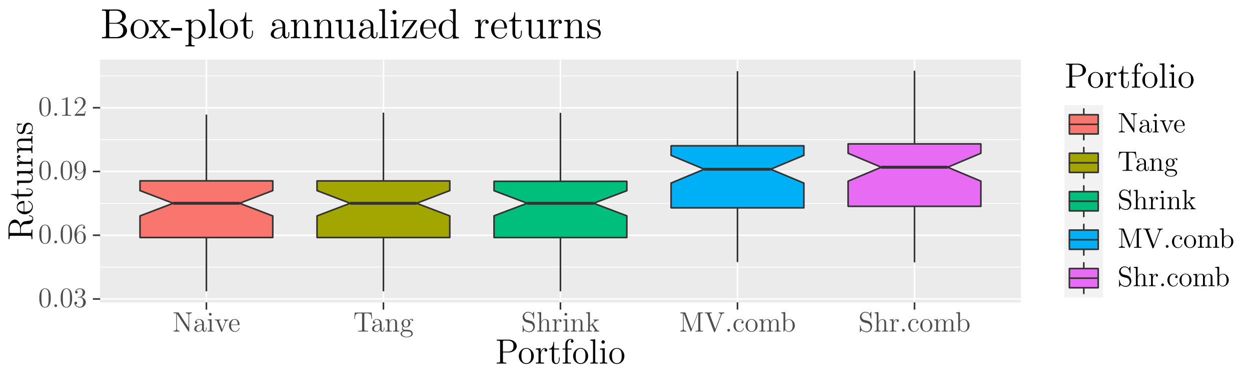

Figure 5-5.: Box-plot of simulated annualized returns

Simulation study results are displayed in the box-plot **5-5**, which confirms previous findings, that the portfolios generate greater portfolio returns with the proposed forecaster as estimate.

5.1.2. Sharpe ratio

The lowest value for the Sharpe ratio is returned by the naive diversification with 0.1307 and is outperformed by the tangency portfolio with 0.1322 and shrinkage with 0.1318, which differ only marginal in their performance, which is less surprising considering the slight differences in returns generated. The proposed estimate leads to superior Sharpe ratios for both, reporting 0.1478 for the mean-variance and 0.1481 in case of the shrinkage portfolio.

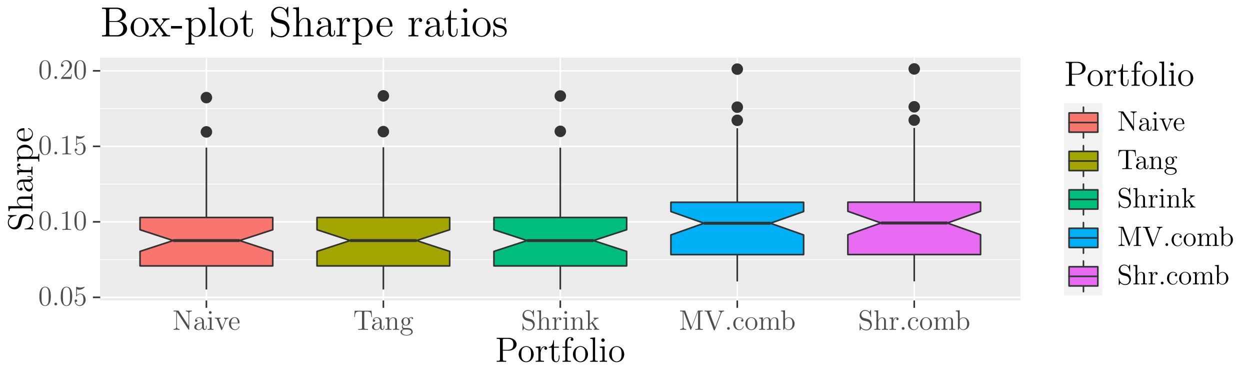

Figure 5-6.: Box-plot of simulated Sharpe ratios

These results are confirmed via confidence intervals of the simulation study and can be accessed in appendix C table **C-1**. The box-plot **5-6** of Sharpe ratios visually confirms that traditional strategies perform on-par and applying the proposed estimate leads to superior values for both strategies in comparison to their base models considering their median, first and third quartile.

5.1.3. Certainty equivalent return

The previously obtained results for Sharpe ratio and cumulated returns are also confirmed for the CEQ return. Reporting a value of 0.0096 for the tangency portfolio, for Shrinkage 0.0096 and the naive diversification 0.0095, which concluding perform quite similar in an optimal scenario. When including the combined forecast as expected return both strategies are improved reporting 0.0111 for the mean-variance and shrinkage portfolio.

Figure **5-7** shows that upper and lower quartiles as well as medians for the improved strategies both are superior in comparison to their base models and confirm the first observation of superior certainty equivalent returns for models that include the combined forecaster. Among the group of strategies using expert aggregation only marginal differences are notable when comparing them to each other with the shrinkage slightly outperforming the tangency portfolio.

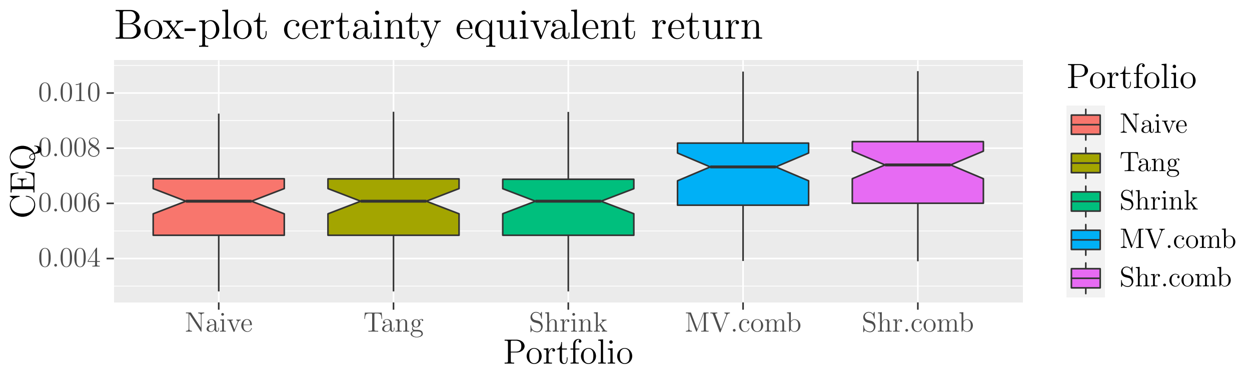

Figure 5-7.: Box-plot of simulated certainty equivalent return

5.1.4. Turnover ratio

Revisiting the turnover ratios in table **5-2**, it can be noted that the strict allocation rules lead to low turnovers for all optimization strategies. Due to price changes in the value of assets, the naive portfolio has to be rebalanced monthly so that its turnover ratio amounts to 0.0526, which is the lowest of all investment strategies. The tangency portfolio with 0.0532 and the related shrinkage approach with 0.0528 return respectively low turnovers due to the stringent allocation rule and the long estimation windows of the covariance matrix and expected returns, leading to stable estimates and portfolio weights.

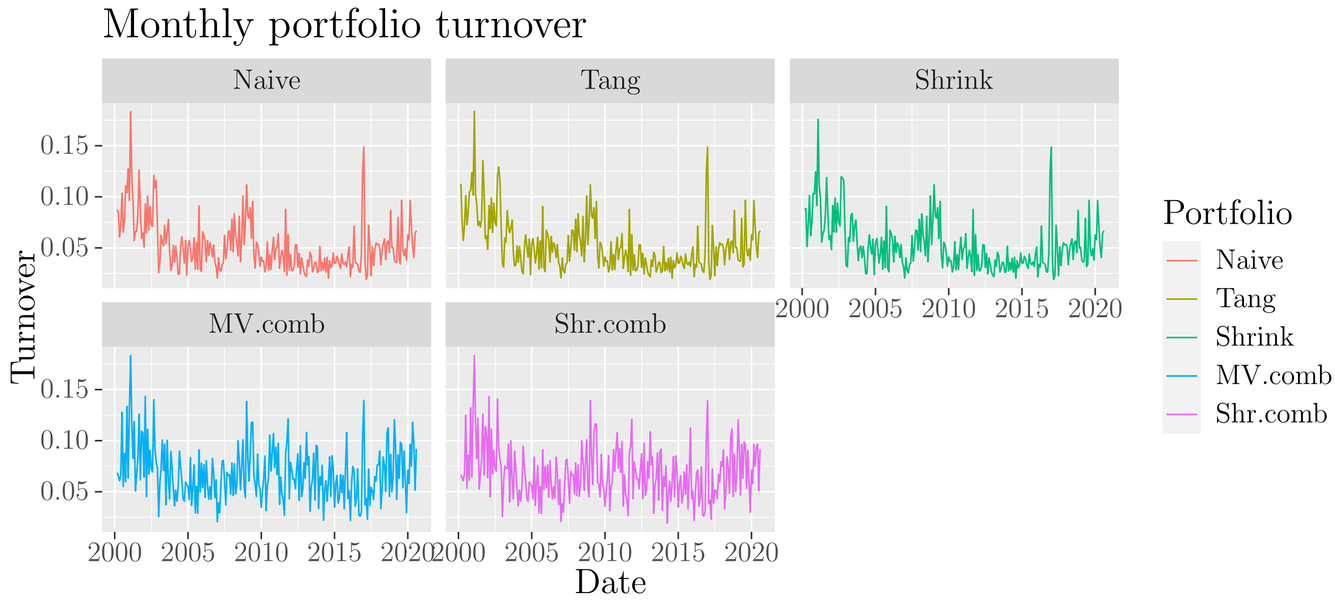

Figure 5-8.: Monthly portfolio turnover

Figure **5-8** displays the monthly turnover of each strategy K. During economic crisis like in 2000 or 2008 all portfolios react and augment their trading. Naive, tangency and shrinkage portfolio all involve marginal less trading over the whole period than models with the combined forecaster. Both strategies with aggregated experts as expected return experienced the same effect during crisis, but with lower peaks in comparison to their average turnover, indicating overall more trading.

The proposed estimator by combining forecasts is quite noisy (see figure **5-2**) and therefore leads to more fluctuating portfolio weights that would produce undesired turnover ratios, if not constrained, which indicates the strong influence of the expected returns on the weights allocated. Due to the allocation constraints introduced, resulting portfolio weights are robust but still vary more than strategies that include the traditional estimates. For the mean-variance portfolio with combined forecaster the turnover ratio is 0.0684, and is less surprising marginal higher than that of the shrinkage portfolio with 0.0682, showing the effect of shrinking the covariance matrix S.

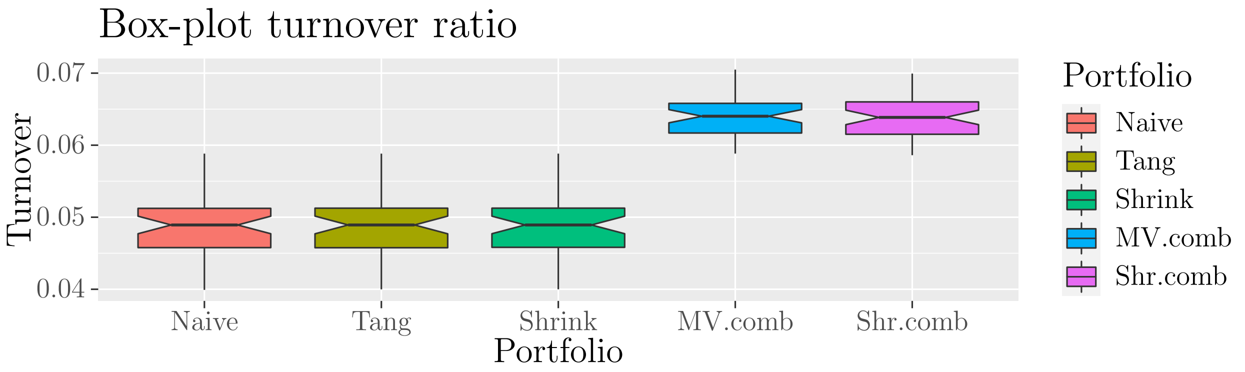

Figure 5-9.: Box-plot of simulated turnover ratios

Figure **5-9** shows that traditional strategies return comparable turnover ratios when considering the first and third quartile, with a marginal greater median for mean-variance and shrinkage optimization. The naive portfolio and traditional approaches are characterised by less turnover than models using the combined forecaster, which leads to a more extensive turnover ratio for the considered models. As demonstrated by the theory, a high portfolio turnover may result in correspondingly high transaction costs. Since these are treated very differently on the financial markets, no assumption was made in this regard.

5.1.5. Omega ratio

The Omega ratios in table **5-2** show that the naive portfolio returns the lowest value with 1.8123, marginally followed by the shrinkage approach with 1.8237 and the tangency portfolio with 1.8261. As for other performance parameters observed, both models with combined forecaster return also in terms of Omega ratio superior values. In detail the proposed estimate leads to improved Omega ratios returning 1.9385 for the mean-variance and 1.9407 for the shrinkage portfolio.

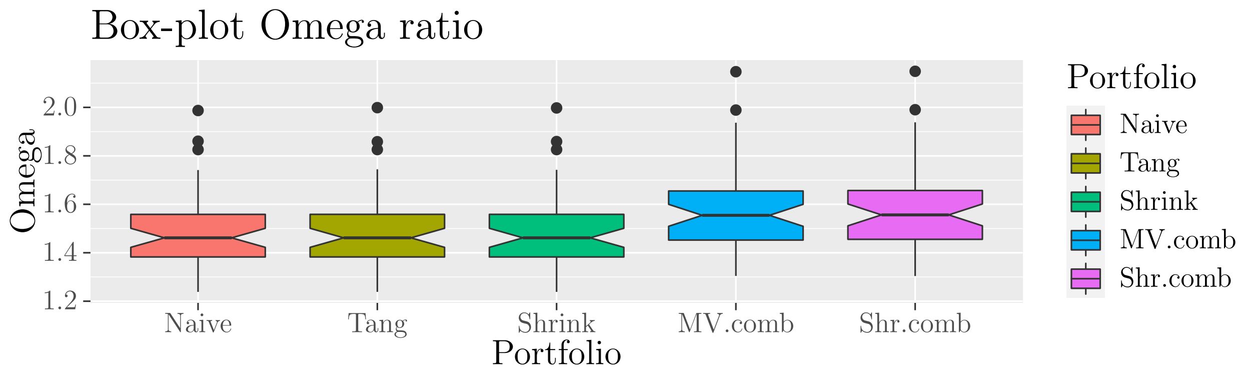

Figure 5-10.: Box-plot of simulated Omega ratios

Omega ratios of the simulated portfolios are displayed in figure **5-10**. Each strategy shows various upward outliers. Naive, tangency and shrinkage portfolio all seem to perform on par, with similar outliers, while both strategies that implement the proposed estimate return superior quartiles and medians in comparison to their base models. Full statistics are reported in appendix C.

6. Conclusion

DeMiguel, Garlappi, Nogales et al. (2009) showed that estimation errors hamper portfolio performance for numerous investment strategies. Especially the erroneous determination of expected returns has a great impact on the portfolio weights, which in conclusion are constructed badly and lead to poor out-of-sample performances. Recent research has therefore focused on relations between assets such as their correlations to optimize portfolios (López de Prado, 2016). In contrast, this study's main objective was to introduce a novel robust estimate of expected returns via expert aggregation to replace the traditional estimates of $\hat{\mu}$ for the tangency and shrinkage portfolio, referred to as base models. Additionally, the naive strategy was included as benchmark, since it is associated with little effort and historically performs well in terms of returns and turnover ratio.

To construct the experts popular forecasting algorithms such as exponential smoothing, ARIMA and quantile regression were applied and afterwards aggregated to construct an improved estimate. Due to the amount of data considered, automatised versions of the first two algorithms were applied. They determine optimal model fits for each rolling window of the time series to forecast the next value. Hyndman and Khandakar (2008) state that linear exponential smoothing methods are all special cases of ARIMA models, while non-linear exponential smoothing do not have equivalent ARIMA counterparts. Both models fit well for linear and non-linear data, and also embrace deterministic trends and stochastic components of the data. Additionally, they include drifts and are able to incorporate random walks.

Gaillard et al. (2016) proposed various expert aggregation methods that improve weights of each forecaster over time. Thus, forecasts were created applying the 1) ML-Prod and 2) ML-Poly algorithm, and after carrying out a simulation study to evaluate the forecasting performance of each model the superior ML-Poly was selected as expected return for portfolio optimization. Due to the noisy estimate, which in consequence would lead to undesired high turnover ratios and to assure comparability between the strategies, all portfolio weights were similarly constrained to a minimum allocation of 0.09 and maximum of 0.11.

To build an optimal diversified portfolio an analysis of the S&P500 was carried out based on hierarchical clustering and ranking assets using their mean log return. In addition, a simulation study was used to confirm all observations made with the optimized dataset. Therefore, the 5 resulting portfolios were tested on their cumulated and annualized returns, Sharpe ratios, CEQ, turnover and Omega ratios. Strategies that included the proposed estimate experienced an improvement in their performance when comparing them to their respective base models, delivering superior statistics for all performance parameter except turnover. Since assumptions regarding resulting transaction costs were not made, one can only argue that the superior returns of the improved models lead to outweigh additional expenses. Differences in performance for the strategies with expert aggregation, can be explained by their covariance matrix estimation, which was kept traditional. When applying the proposed forecast combination the shrinkage portfolio slightly outperformed the mean-variance optimization considering all performance parameter. These observations were also confirmed in the simulation study.

6.1. Discussion

This thesis showed an innovative application of two expert aggregation algorithms in order to minimize estimation errors of return estimates for two in the literature proposed portfolios and improve each one's performance. The results indicated that strategies implementing the improved expected return dominate their respective base models. Combining forecasts showed promising results, however there is still no consensus on how to best combine individual forecasts or which experts to consider. This problematic has been subject to discussions in literature suggesting a mixture of statistical models and expert aggregation when data is sparse and evolving. Despite the time and effort it takes to elicit expert-generated data, the wide range of applications and new methods show general research interest.

For ARIMA forecasting the two major issues are the model estimation and the choice of dataset. First off, there is no clear selection order between different ARIMA models in terms of the lagged values of the AR process and the MA process. Secondly, it is still wildly discussed whether ARIMA forecasting should adopt a time series with a longer estimation window or a shorter one. These challenges can hamper forecasting accuracy and may lead forecasters to incorrectly conclude that the ARIMA model is inferior to other forecasting techniques. Thus, it is improbable that large estimation errors that sometimes are generated by ARIMA models are caused by their inherent weaknesses, but rather by the incorrect determination of appropriate parameters for an ARIMA forecasting model and, as a consequence, the ARIMA models are not trained with the correct data set. Dong et al. (2020) tested the automatic ARIMA algorithm for different rolling windows and found that a long estimation window and a low forecast horizon perform exceptionally well for forecasting purposes. Therefore, the rolling window was chosen to be 90 months and the forecast horizon was limited to just one.

Based on these findings future studies might focus on two aspects. Firstly, one could further improve the experts considered by testing different window lengths for the automatic exponential smoothing algorithm or by including macroeconomic factors to the random forest as predicting variables. Additionally, one might choose to consider additional experts, such as neural networks to further improve forecasting accuracy of the aggregated expert. Due to the amount of time needed to generate experts, research opportunities can also be found in collecting data from experts that are unbiased and in a less-time consuming manner.

Secondly, only few studies focused on rigorously comparing combination forecasting models. Hence, further investigation is required to analyse experts and statistical forecasts to confirm the added value of expert judgement. McAndrew et al. (2019) found that the majority of articles measured success on whether or not the combination scheme could produce a forecast and visually inspected the results. Latter was used due to the lag of ground truth data, but in this case, a simulation study should generate hindsight of the forecasting performance of a novel combination method. Therefore, future research on expert aggregation still needs to find an appropriate parameter to measure forecast accuracy and develop experiments to evaluate novel combination algorithms in comparison to existing methods.

Combining experts to produce forecasts can outperform statistical ensembles when data is sparse, or rapidly evolving. Expert aggregation algorithms are able to gain insight on how forecasts are made and ultimately how to best use the information each expert provides to make crucial decisions about future forecasts.

A. State space framework

Appendix A details the statistical models that underlie the exponential smoothing methods. To distinguish between models with additive and multiplicative errors, a third letter is added to the classification in table **3-1**. Each state space model is labelled ETS(·, ·, ·) for (Error, Trend, Seasonal), leaving the following possibilities for each component: Error = $\{A, M\}$, Trend = $\{N, A, A_d\}$ and Seasonal = $\{N, A, M\}$.

For simple exponential smoothing (A, N, N) with additive errors the forecast equation can be written as $y_t = \ell_{t-1} + e_t$, so that each observation can be determined by the previous level plus an error term. To convert given equation into a state space model, the probability distribution e_t has to be specified. For a model with additive errors, Hyndman et al., 2008 assume that the one-step training errors (residuals) are normally distributed white noise with mean 0 and variance σ^2, which can be formulated as $e_t = \epsilon_t \sim \mathbf{NID}(0, \sigma^2)$. Concluding, the equations of simple exponential smoothing can be written as:

$$y_t = \ell_{t-1} + \epsilon_t \tag{A-1a}$$

$$\ell_t = \ell_{t-1} + \alpha\epsilon_t \tag{A-1b}$$

Together with the statistical distribution of the errors, they form the innovations state space model underlying simple exponential smoothing. A-1a is referred to as measurement equation and shows the relationship between the observations and unobserved states. For simple exponential smoothing observation y_t is a linear function of the level ℓ_{t-1} with a predictable and unpredictable part of y_t and the error ϵ_t. Formula A-1b describes the state equation, which illustrates the evolution of the state through time where α governs the amount of change in successive levels. High values of α leads to rapid changes in level, whereas low values of α allow only smoother changes.

In the same manner, the models with multiplicative errors can be specified by writing the one-step-ahead training errors as relative errors:

$$\epsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \tag{A-2}$$

where $e_t = \epsilon_t \sim \mathbf{NID}(0, \sigma^2)$. All resulting state space equations are listed in **A-1**.

Table A-1.: State space equations for each model in the ETS framework

Additive Error Models			
Trend		Seasonal	
	N	A	M
N	$y_t = \ell_{t-1} + \epsilon_t$ $\ell_t = \ell_{t-1} + \alpha\epsilon_t$	$y_t = \ell_{t-1} + s_{t-m} + \epsilon_t$ $\ell_t = \ell_{t-1} + \alpha\epsilon_t$ $s_t = s_{t-m} + \gamma\epsilon_t$	$y_t = \ell_{t-1}s_{t-m} + \epsilon_t$ $\ell_t = \ell_{t-1} + \alpha\epsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\epsilon_t/\ell_{t-1}$
A	$y_t = \ell_{t-1} + b_{t-1} + \epsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\epsilon_t$ $b_t = b_{t-1} + \beta\epsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \epsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\epsilon_t$ $b_t = b_{t-1} + \beta\epsilon_t$ $s_t = s_{t-m} + \gamma\epsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m} + \epsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\epsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\epsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\epsilon_t/(\ell_{t-1} + b_{t-1})$
A_d	$y_t = \ell_{t-1} + \phi b_{t-1} + \epsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\epsilon_t$ $b_t = \phi b_{t-1} + \beta\epsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \epsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\epsilon_t$ $b_t = \phi b_{t-1} + \beta\epsilon_t$ $s_t = s_{t-m} + \gamma\epsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m} + \epsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\epsilon_t/s_{t-m}$ $b_t = \phi b_{t-1} + \beta\epsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\epsilon_t/(\ell_{t-1} + \phi b_{t-1})$
Multiplicative Error Models			
Trend		Seasonal	
	N	A	M
N	$y_t = \ell_{t-1}(1 + \epsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\epsilon_t)$	$y_t = (\ell_{t-1} + s_{t-m})(1 + \epsilon_t)$ $\ell_t = \ell_{t-1} + \alpha(\ell_{t-1} + s_{t-m})\epsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + s_{t-m})\epsilon_t$	$y_t = \ell_{t-1}s_{t-m}(1 + \epsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\epsilon_t)$ $s_t = s_{t-m}(1 + \gamma\epsilon_t)$
A	$y_t = (\ell_{t-1} + b_{t-1})(1 + \epsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\epsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\epsilon_t$	$y_t = (\ell_{t-1} + b_{t-1} + s_{t-m})(1 + \epsilon_t)$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha(\ell_{t-1} + b_{t-1} + s_{t-m})\epsilon_t$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1} + s_{t-m})\epsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + b_{t-1} + s_{t-m})\epsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m}(1 + \epsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\epsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\epsilon_t$ $s_t = s_{t-m}(1 + \gamma\epsilon_t)$
A_d	$y_t = (\ell_{t-1} + \phi b_{t-1})(1 + \epsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\epsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\epsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1} + s_{t-m})(1 + \epsilon_t)$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\epsilon_t$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\epsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + b_{t-1} + s_{t-m})\epsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m}(1 + \epsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\epsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\epsilon_t$ $s_t = s_{t-m}(1 + \gamma\epsilon_t)$

B. Forecast error statistics

Table B-1.: Confidence intervals and means of accuracy for forecasts generated with automatic ARIMA, ETS and expert aggregation algorithms for time series obtained in the simulation study

Error statistic	ARIMA	ETS	ML-Poly	ML-Prod
ME				
CI 5%	-0.0026	-0.0036	-0.0055	0.0124
Mean	0.0048	0.0063	0.0008	0.0234
CI 95%	0.0143	0.0171	0.0052	0.0318
RMSE				
CI 5%	0.0102	0.0085	0.0139	0.0278
Mean	0.0367	0.0328	0.0365	0.0455
CI 95%	0.1008	0.0985	0.0965	0.0990
MAE				
CI 5%	0.0049	0.0066	0.0054	0.0215
Mean	0.0245	0.0246	0.0226	0.0364
CI 95%	0.0754	0.0749	0.0743	0.0727

Table B-2.: Forecast accuracy of ETS, ARIMA and expert aggregation algorithms for the diversified dataset

	Error statistics					
Forecast model	ARIMA			ETS		
Ticker	ME	RMSE	MAE	ME	RMSE	MAE
AOS	0.0148	0.0195	0.0162	0.0183	0.0197	0.0191
BF-B	0.0183	0.0263	0.0218	0.0189	0.0221	0.0199
BLL	0.0205	0.0622	0.0389	0.0195	0.0367	0.0246
CSCO	0.0008	0.0243	0.0105	0.0006	0.0267	0.0130
GE	-0.0043	0.0374	0.0195	-0.0054	0.0256	0.0140
HFC	0.0256	0.0563	0.0346	0.0281	0.0564	0.0382
JKHY	0.0194	0.0180	0.0118	0.0131	0.0166	0.0141
MXIM	0.0105	0.0281	0.0138	0.0085	0.0158	0.0122
NKE	0.0154	0.0254	0.0206	0.0166	0.0182	0.0176
PFE	0.0045	0.0132	0.0082	0.0023	0.0128	0.0095

	Error statistics					
Forecast model	ML-Poly			ML-Prod		
Ticker	ME	RMSE	MAE	ME	RMSE	MAE
AOS	0.0046	0.0147	0.0099	0.0045	0.0452	0.0308
BF-B	0.0029	0.0182	0.0104	0.0000	0.0386	0.0285
BLL	0.0049	0.0362	0.0199	0.0025	0.0562	0.0335
CSCO	0.0005	0.0275	0.0110	-0.0040	0.0478	0.0258
GE	-0.0026	0.0286	0.0159	-0.0101	0.0463	0.0323
HFC	0.0077	0.0451	0.0265	0.0063	0.0604	0.0389
JKHY	0.0021	0.0160	0.0074	0.0004	0.0419	0.0266
MXIM	0.0036	0.0211	0.0096	-0.0009	0.0459	0.0272
NKE	0.0032	0.0288	0.0115	0.0027	0.0452	0.0269
PFE	0.0007	0.0137	0.0066	-0.0040	0.0381	0.0291

C. Simulation study portfolio performance

Table C-1.: Portfolio performance parameter generated by the simulation study

Portfolio	Naive	Tang	Shrink	Mv.comb	Shr.comb
Annualized return	0.1200	0.1218	0.1211	0.1411	0.1415
CI 5%	0.0470	0.0470	0.0470	0.0613	0.0618
Median	0.0750	0.0750	0.0750	0.0911	0.0920
CI 95%	0.1087	0.1086	0.1087	0.1269	0.1271
Sharpe ratio	0.1307	0.1322	0.1318	0.1478	0.1481
CI 5%	0.0592	0.0592	0.0592	0.0659	0.0661
Median	0.0876	0.0876	0.0876	0.0991	0.0992
CI 95%	0.1488	0.1490	0.1488	0.1649	0.1651
CEQ return	0.0095	0.0096	0.0096	0.0111	0.0111
CI 5%	0.0039	0.0039	0.0039	0.0050	0.0051
Median	0.0061	0.0061	0.0061	0.0073	0.0074
CI 95%	0.0087	0.0086	0.0086	0.0100	0.0100
Turnover ratio	0.0526	0.0532	0.0528	0.0684	0.0682
CI 5%	0.0411	0.0414	0.0413	0.0598	0.0599
Median	0.0489	0.0489	0.0489	0.0640	0.0639
CI 95%	0.0550	0.0550	0.0550	0.0689	0.0687
Omega ratio	1.8123	1.8261	1.8237	1.9385	1.9407
CI 5%	1.3164	1.3164	1.3164	1.3722	1.3740
Median	1.4616	1.4616	1.4616	1.5545	1.5562
CI 95%	1.7878	1.7892	1.7882	1.9120	1.9137

Bibliography

Amit, Y. & Geman, D. (1997). Shape quantization and recognition with randomized trees. *0899-7667*, *9*(7), 1545–1588. https://doi.org/10.1162/neco.1997.9.7.1545

Anderson, B. D. O. (2012). *Optimal filtering*. Dover Publications.

Aoki, M. & Havenner, A. (1991). State space modeling of multiple time series. *Econometric Reviews*, *10*(1), 1–59. https://doi.org/10.1080/07474939108800194

Arlot, S. & Genuer, R. (2014). Analysis of purely random forests bias. https://arxiv.org/pdf/1407.3939

Athey, S., Tibshirani, J. & Wager, S. (2019). Generalized random forests. *0090-5364*, *47*(2), 1148–1178. https://doi.org/10.1214/18-AOS1709

Ban, G.-Y., El Karoui, N. & Lim, A. E. B. (2018). Machine learning and portfolio optimization, (64), 1136–1154.

Biau, G. (2012). Analysis of a random forests model, (13), 1063–1095.

Biau, G., Devroye, L. & Lugosi, G. (2008). Consistency of random forests and other averaging classifiers, (9), 2015–2033.

Blum, A. & Mansour, Y. (2007). From external to internal regret. *Journal of Machine Learning Research*, *8*(47), 1307–1324.

Box, G. E. P. & Jenkins, G. M. (1970). *Time series analysis: Forecasting and control*. Holden Day.

Breiman, L. (1984). *Classification and regression trees* [Breiman, Leo, (author.)]. [Routledge].

Breiman, L. (1996). Bagging predictors [PII: BF00058655]. *08856125*, *24*(2), 123–140. https://doi.org/10.1007/BF00058655

Breiman, L. (2001). Random forests [PII: 354300]. *08856125*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Brockwell, P. J. & Davis, R. A. (2006). *Time series: Theory and methods* (2nd ed., corrected.). New York, Springer.

Bühlmann, P. & Yu, B. (2002). Analyzing bagging [PII: aos30n4r01]. *0090-5364*, *30*(4), 927–961. https://doi.org/10.1214/aos/1031689014

Cesa-Bianchi, N. & Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge University Press.

Cesa-Bianchi, N. & Lugosi, G. (2003). Potential-based algorithms in on-line prediction and game theory [PII: 5120299]. *08856125*, *51*(3), 239–261. https://doi.org/10.1023/A:1022901500417

Cesa-Bianchi, N., Mansour, Y. & Stoltz, G. (2007). Improved second-order bounds for prediction with expert advice [PII: 5001]. *08856125*, *66*(2-3), 321–352. https://doi.org/10.1007/s10994-006-5001-7

DeMiguel, V., Garlappi, L., Nogales, F. J. & Uppal, R. (2009). A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms, *55*(5), 798–812. https://doi.org/10.1287/mnsc.1080.0986

DeMiguel, V., Garlappi, L. & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/ n portfolio strategy? *0893-9454*, *22*(5), 1915–1953. https://doi.org/10.1093/rfs/hhm075

Denil, M., Matheson, D. & Freitas, N. d. (2014). Narrowing the gap: Random forests in theory and in practice, (32), 665–673.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization [PII: 262611]. *08856125*, *40*(2), 139–157. https://doi.org/10.1023/A:1007607513941

Dong, H., Guo, X., Reichgelt, H. & Hu, R. (2020). Predictive power of arima models in forecasting equity returns: A sliding window method [PII: 184]. *Journal of Asset Management*, *21*(6), 549–566. https://doi.org/10.1057/s41260-020-00184-z

Fan, J., Farmen, M. & Gijbels, I. (1998). Local maximum likelihood estimation and inference. *1369-7412*, *60*(3), 591–608. https://doi.org/10.1111/1467-9868.00142

Freund, Y., Schapire, R. E., Singer, Y. & Warmuth, M. K. (1997). Using and combining predictors that specialize (F. T. Leighton, Ed.). In F. T. Leighton (Ed.), *Proceedings of the twenty-ninth annual acm symposium on theory of computing*. the twenty-ninth annual ACM symposium, New York, NY, ACM. ACM Special Interest Group on Algorithms and Computation Theory. El Paso, Texas, United States. https://doi.org/10.1145/258533.258616

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine, (29), 1189–1232. https://www.jstor.org/stable/2699986

Frost, P. A. & Savarino, J. E. (1986). An empirical bayes approach to efficient portfolio selection. *00221090*, *21*(3), 293. https://doi.org/10.2307/2331043

Gaillard, P., Goude, Y. & Nedellec, R. (2016). Additive models and robust aggregation for gefcom2014 probabilistic electric load and electricity price forecasting [PII: S0169207015001545]. *International Journal of Forecasting*, *32*(3), 1038–1050. https://doi.org/10.1016/j.ijforecast.2015.12.001

Hannan, E. J. & Deistler, M. (2012). *The statistical theory of linear systems* (Vol. 70). Philadelphia, SIAM. https://doi.org/10.1137/1.9781611972191

Ho, T. K. (1998). The random subspace method for constructing decision forests. *01628828*, *20*(8), 832–844. https://doi.org/10.1109/34.709601

Hyndman, R. J. & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, *27*(3). https://doi.org/10.18637/jss.v027.i03

Hyndman, R. J., Koehler, A. B., Ord, J. K. & Snyder, R. D. (2008). *Forecasting with exponential smoothing: The state space approach.* Berlin, Heidelberg, Springer. https://doi.org/10.1007/978-3-540-71918-2

Jagannathan, R. & Ma, T. (2003). Risk reduction in large portfolios: Why imposing the wrong constraints helps, *58*(4), 1651–1683. https://doi.org/10.1111/1540-6261.00580

Koenker, R. (2005). Quantile regression [Koenker, Roger (VerfasserIn)]. *Cambridge*, Cambridge University Press. https://doi.org/10.1017/CBO9780511754098

Koenker, R. & Bassett, G. (1978). Regression quantiles [Econometrica, 46(1), 33]. *Econometrica*, *46*(1), 33. https://doi.org/10.2307/1913643

Kwiatkowski, D., Phillips, P. C., Schmidt, P. & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root [PII: 030440769290104Y]. *Journal of Econometrics*, *54*(1-3), 159–178. https://doi.org/10.1016/0304-4076(92)90104-y

Landau, S. & Chis Ster, I. (2010). Cluster analysis: Overview, 72–83. https://doi.org/10.1016/B978-0-08-044894-7.01315-4

Ledoit, O. & Wolf, M. (2004). Honey, i shrunk the sample covariance matrix, (4), 110–119. https://doi.org/10.3905/jpm.2004.110

Lin, Y. & Jeon, Y. (2006). Random forests and adaptive nearest neighbors. *0162-1459*, *101*(474), 578–590. https://doi.org/10.1198/016214505000001230

Lintner, J. (1965). The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *00346535*, *47*(1), 13. https://doi.org/10.2307/1924119

Littlestone, N. & Warmuth, M. K. (1994). The weighted majority algorithm [PII: S0890540184710091]. *Information and Computation*, *108*(2), 212–261. https://doi.org/10.1006/inco.1994.1009

López de Prado, M. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, *42*(4), 59–69. https://doi.org/10.3905/jpm.2016.42.4.059

Makridakis, S. & Hibon, M. (2000). The m3-competition: Results, conclusions and implications [PII: S0169207000000571]. *International Journal of Forecasting*, *16*(4), 451–476. https://doi.org/10.1016/S0169-2070(00)00057-1

Markowitz, H. M. (1952). Portfolio selection, (Vol. 7, No. 1), 77–91.

McAndrew, T., Wattanachit, N., Gibson, G. C. & Reich, N. G. (2019). Aggregating predictions from experts: A scoping review of statistical methods, experiments, and applications [https://github.com/tomcm39/AggregatingExpertElicitedDataForPrediction v0.2: updated funding info]. https://arxiv.org/pdf/1912.11409

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*(Jun), 983–999.

Mentch, L. & Hooker, G. (2016). Quantifying uncertainty in random forests via confidence intervals and hypothesis tests, (17), 1–41.

Merton, R. C. (1980). On estimating the expected return on the market: An exploratory investigation, (8), 323–361.

Mossin, J. (1966). Equilibrium in a capital asset market, (34), 768–783.

Newey, W. K. (1994). The asymptotic variance of semiparametric estimators, *62*(6), 1349. https://doi.org/10.2307/2951752

Schmidhuber, J. (2014). Deep learning in neural networks: An overview, (61), 85–117.

Scornet, E., Biau, G. & Vert, J.-P. (2015). Consistency of random forests. *0090-5364*, *43*(4), 1716–1741. https://doi.org/10.1214/15-AOS1321

Sharpe, W. F. (1964). Capital asset prices: a theory of market equilibirium under conditions of risk, (19), 425–442.

Sharpe, W. F. (1970). *Portolio theory and capital markets*. McGraw-Hill.

Staniswalis, J. G. (1989). The kernel estimate of a regression function in likelihood-based models. *0162-1459*, *84*(405), 276. https://doi.org/10.2307/2289874

Stone, C. J. (1977). Consistent nonparametric regression, (5), 595–620.

Tibshirani, R. & Hastie, T. (1987). Local likelihood estimation. *0162-1459*, *82*(398), 559–567. https://doi.org/10.1080/01621459.1987.10478466

Timmermann, A. (2006). Chapter 4 forecast combinations. In G. Elliott, C. Granger & A. Timmermann (Eds.), *Handbook of economic forecasting* (pp. 135–196). Elsevier. https://doi.org/10.1016/S1574-0706(05)01004-9

Vovk, V. (1998). A game of prediction with expert advice [PII: S0022000097915567]. *Journal of Computer and System Sciences*, *56*(2), 153–173. https://doi.org/10.1006/jcss.1997.1556

Wager, S. & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *0162-1459*, *113*(523), 1228–1242. https://doi.org/10.1080/01621459.2017.1319839

Wager, S. & Walther, G. (2015). Adaptive concentration of regression trees, with application to random forests. https://arxiv.org/pdf/1503.06388

Zeileis, A., Hothorn, T. & Hornik, K. (2008). Model-based recursive partitioning. *1061-8600*, *17*(2), 492–514. https://doi.org/10.1198/106186008X319331