Eötvös Loránd University
Corvinus University of Budapest

Inverse Regularisation as a New Machine Learning Concept: a Simulation and Empirical Study

MSc Thesis

Tamás Jászai

MSc in Actuarial and Financial Mathematics
Quantitative Finance Major

Supervisor:
Milán Badics

Budapest, 2022

Acknowledgements

I thank my supervisor, Milán Badics for the enormous support he has given me in defining the research question of my project, for the feedback he has given me on my work, and for helping me present my results in a (hopefully) clear manner.

Additionally, I thank my parents, whom I owe immeasurable gratitude for all the support they have given me.

NYILATKOZAT

Név: Jászai Tamás

ELTE Természettudományi Kar, szak: Biztosítási és pénzügyi matematika

NEPTUN azonosító: LBVMRL

Szakdolgozat címe:

Inverse Regularisation as a New Machine Learning Concept: a Simulation and Empirical Study

A **szakdolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2022.05.30.

Jászai Tamás

a hallgató aláírása

Contents

Tables and Figures

Introduction

In financial and economic forecasting, especially when there are a significant number of potentially important predictors and a low signal-to-noise ratio, it is a common approach to use shrinkage estimators or regularisation methods to decrease the variance of the forecasting method. Recently, the LASSO has seen widespread applications in financial forecasting (Rapach, Strauss, and Zhou, 2013), (Chinco, Clark-Joseph, and Ye, 2019), (Freyberger, Neuhierl, and Weber, 2020), (Kozak, Nagel, and Santosh, 2020).

Another of the more popular and successful of these shrinkage estimators is the so called 'forecast combination' method. It consists of estimating univariate linear regressions using OLS, then generating forecasts from the univariate regressions, and finally taking the - usually equal-weighted - average of these forecasts to get the 'final' forecast of the variable of interest[1] (Rapach, Strauss, and Zhou, 2010), (Rapach, 2013). The estimator is equivalent to setting some restrictions on the coefficients and the covariance matrix of a multivariate linear regression, which show that it is in fact a very strong form of shrinkage (Rapach, Strauss, and Zhou, 2010). This method, which I will call 'univariate OLS' from now on[2], has been applied to forecasting the US equity premium (Rapach, Strauss, and Zhou, 2010), (Rapach, 2013), (Elliott, Gargano, and Timmermann, 2013), (Zhang, Wei, Ma, et al., 2019), (Rapach and Zhou, 2020) crude oil prices (Zhang, Ma, and Y. Wang, 2019), GDP (Chauvet and Potter, 2013), among many others.

In this paper, I argue that the uOLS proves to be a too strong form of shrinkage for data generating processes with signal-to-noise ratios and predictor correlation structures that are practically relevant to financial and economic forecasting. For some data generating processes, especially those that have at least a mediocre signal-to-noise ratio and do not have excessive predictor correlation, the uOLS has too high a bias. The uOLS unfortunately cannot optimise the bias-variance trade-off to achieve a lower expected squared error.

In this paper, I suggest the usage of a wider class of methods, which give back the uOLS as a special case, but include a hyperparameter that can be set to optimise the

[1] Note that the term 'forecast combination' often has a wider meaning, as in Timmermann (2006), for example. Here I refer to a narrower meaning of the term, which is used in Rapach, Strauss, and Zhou (2010) and Rapach (2013), for example.

[2] This is meant to a) avoid the different usages of the term 'forecast combination and b) to emphasise the close relationship between uOLS and a new method I will introduce shortly, 'uNCL'.

bias-variance trade-off. I call these methods *inverse regularisation method* or *inverse regularisers* (*IR-s*), because they work in a reverse fashion when compared to traditional 'regularisation' methods, such as the LASSO or ridge regression. What I mean by this is that IR methods start from a high bias low variance estimator, the uOLS and decrease its bias at the cost of increasing its variance by setting the value of a hyperparameter. In contrast, the LASSO and ridge work the other way around, by starting from a low bias high variance 'supermodel' that includes all of the available independent variables and decreases the variance at the cost of introducing additional bias by setting the value of a hyperparameter (Tibshirani, 1996).

There have been several IR methods in the literature, however, they have not been grouped as such (Diebold and Shin, 2019), (Elliott, Gargano, and Timmermann, 2013), (Elliott, Gargano, and Timmermann, 2015), (Boot and Nibbering, 2019). To build upon my new definition of the IR-s, I introduce a new categorisation of IR methods into two groups. I group IR methods into those that optimise the bias-variance trade-off by slightly altering the estimation or structure of the 'individual models'[3], which I call 'stage one IR-s', and those that optimise the bias-variance trade-off by estimating non-equal combination weights for the individual models, which I call 'stage two IR-s'[4].

My first major contribution to the forecasting literature is to use this new categorisation to compare the performance of popular and highly representative methods from both category of IR-s as well as 'traditional' regularisation methods in a large scale simulation study. I compare a well-known and highly competitive stage one IR, the complete subset regression (CSR) of Elliott, Gargano, and Timmermann (2013), Elliott, Gargano, and Timmermann (2015) and Boot and Nibbering (2019), two stage two IR-s that are highly representative of these category of methods, the ELASSO and ERidge of Diebold and Shin (2019) and two traditional regularisation methods, the LASSO (Tibshirani, 1996) and ridge regression (Hoerl and Kennard, 1970). The CSR (Elliott, Gargano, and Timmermann, 2013), stage two IR-s similar to the ELASSO and ERidge (Rapach, Strauss, and Zhou, 2010), and especially LASSO have been used in financial applications, a through comparison an evaluation of these methods together is lacking from the literature. Additionally, my results also include an inherent comparison with the uOLS, as it is a special case of all IR methods.

I find that the stage one IR-s are highly successful and tend to dominate the stage two IR-s for the majority of the examined data generating processes. This suggests that it is much better to alter the estimation or the structure of the individual models than to estimate combination weights. I note a close, and obviously not accidental similarity between

[3] The 'individual models' refer to the models whose predictions are aggregated by some weighting; in the case of the uOLS, the individual models are the univariate regressions, each with a different predictor.

[4] Note that these two groups are rather 'attributes' in the sense that they are not necessarily mutually exclusive. Nevertheless, in practice, most methods fall into only one of these groups and I also do not consider any methods that fall into both categories in this paper.

this finding and the well-known ’forecast combination puzzle’, which is the stylised fact that an equal weighted combination of forecasts tends to outperform more sophisticated combination schemes (Smith and Wallis, 2009).

My second finding is that the stage one IR-s, which ’inverse regularise’ the uOLS, tend to outperform the LASSO and ridge regression, which ’regularise’ the multivariate ’supermodel’ that includes all predictors, for most of the DGPs considered. This is an important finding, because it shows that inverse regularisation approaches are superior to the very popular ’normal’ regularisation approaches.

I make another contribution to the literature by introducing a new stage one IR method, and comparing its performance with the other methods. This new method replaces the OLS in the estimation of the univariate models of the uOLS with a training algorithm called ’negative correlation learning’ (NCL). The NCL algorithm has been present in the machine learning community since the late 90s, where it is mainly used to train ensembles of neural networks (Liu and Yao, 1999). The algorithm is meant to train a set of neural networks that are ’diverse’ in the sense that their predictions have low covariance (Brown, Wyatt, and Tino, 2005). Because the ’diversity’ of the individual networks can usually be increased (equivalent to a decrease in their covariance), a trade-off between the accuracy of the individual models and their diversity emerges. The NCL algorithm, thus, trains a set of neural networks that are accurate in aggregate but not by themselves. The algorithm itself is based on a decomposition of the squared error of a linear combination of estimators called the ’ambiguity decomposition’ (Krogh and Vedelsby, 1994). This decomposition, although it applies in a much wider context than neural networks, has been mostly overlooked in the econometric and forecasting literature to the best of my knowledge.

I compare the predictive performance of this new method, which I call ’uNCL’[5], with the other methods. I find that it usually has comparable performance with the other stage one IR method, CSR, and as such is usually one of the two best performing methods in the simulations.

Additionally, I estimate the bias-variance and a bias-variance-covariance decomposition of the uNCL through the simulations. The bias-variance decomposition shows that the uNCL acts as an IR method; it has a decreasing bias and increasing variance as it moves away from the uOLS by increasing the value of its hyperparameter. The estimate of the bias-variance-covariance decomposition shows that the uNCL has covariance that is mostly flat as a function of its hyperparameter. In its applications to the training of ensembles of neural networks, the NCL algorithm has a covariance curve that is decreasing in its hyperparameter (Brown, Wyatt, and Tino, 2005). Thus, my results indicate a significant deviation in the behavior of the NCL algorithm as I move the uNCL setting. The

[5] Abbreviation for ’univariate negative correlation learning’

NCL optimises an accuracy-diversity trade-off when applied to neural networks (Brown, Wyatt, and Tino, 2005); in my uNCL application, it optimises the bias-variance trade-off instead.

In recent years, machine learning models have seen widespread applications in financial forecasting. Notable, examples include Gu, Kelly, and Xiu (2020) in the cross section of equity returns, Bianchi, Buchner, and Tamoni (2021) in the cross section of bond returns, Hollstein and Prokopczuk (2022) in the time-series predictability of factor portfolios, and Avramov, Cheng, and Metzker (2021), which applies neural networks. Most importantly, the LASSO has seen extensive applications in forecasting the US equity premium (Rapach, Strauss, and Zhou, 2013), (Chinco, Clark-Joseph, and Ye, 2019), (Freyberger, Neuhierl, and Weber, 2020), (Kozak, Nagel, and Santosh, 2020). To extend upon this literature and provide an empirical comparison of IR-s and traditional regularisation methods, I also forecast the US equity premium, using a standard set of macroeconomic and financial predictors from Welch and Goyal (2007). I find that the stage one IR methods, the uNCL and CSR give the best performance of the considered models, with an R^2_{OOS} over 3% and over 5% for the uNCL if the nonnegativity restriction of Campbell and Thompson (2007) are imposed on the forecasts, and the hyperparameters are chosen optimally. The stage two IR-s, the LASSO and ridge perform poorly. The ranking of the uNCL and CSR as the two best performing models is robust to validating the hyperparameters from the data. Also, I find that there is a huge drop in the performance of most models after the 90s. CSR, and the uNCL are the least affected by this drop, remaining on par with or slightly outperforming the historical average even after the 90s, unlike other models. The results from the empirical application are in agreement with the results from the simulations, and indicate that the stage one IR-s, CSR and the uNCL are superior in forecasting performance to traditional and stage two IR methods. These finding suggests a more widespread application of the so-far somewhat overlooked CSR and the new uNCL in future studies.

The paper is structured as follows. Section II introduces the theoretical foundations, concepts and models of my study in more detail, as well as a short review of the literature of forecasting the US equity premium. Section III describes the simulation study. Section IV is devoted to the empirical application of forecasting the US quarterly equity premium. Section V concludes.

Review and Theoretical Foundations

II.1 uOLS as a strong form of shrinkage

The uOLS, which I describe here, has seen widespread applications in forecasting, including forecasting the equity premium (Rapach, Strauss, and Zhou, 2010), (Rapach, 2013), (Rapach and Zhou, 2020), (Elliott, Gargano, and Timmermann, 2013), (Zhang, Wei, Ma, et al., 2019), crude oil prices (Zhang, Ma, and Y. Wang, 2019), GDP (Chauvet and Potter, 2013) among many others. Notably, it is also a special case of the CSR of Elliott, Gargano, and Timmermann (2013), introduced later.

Suppose one has a variable one wants to forecast, y_t, $t = 1, 2, \ldots, T$ and p number of predictors, $x_{p,t}$, $t = 0, 1, \ldots, T-1$. Also suppose that the true model has the following form:

$$y_t = \beta_0 + \sum_{i=1}^{p} \beta_i x_{i,t-1} + \epsilon_t \tag{II.1}$$

Where ϵ is a random variable and the βs are some unknown deterministic constants, and the Gauss-Markov assumptions are satisfied: the predictors are exogen ($E(\epsilon_i|X) = 0$ $Var(\epsilon_i|X) = \sigma$ for some constant σ and the noise terms have zero cross-product, $E(\epsilon_l \epsilon_k|X) = 0$ for $l \neq k$, and X is a $Tx(p+1)$ matrix with 1s in its first column and the i-th ($i > 1$) column equal to the predictor vector $\boldsymbol{x_i} = (x_{i,0}, x_{i,1}, \ldots, x_{i,T-1})$ and is assumed the be full rank.

Suppose one wants to estimate a model of the form in equation II.6. An obvious solution is to minimise the mean squared error:

$$MSE_{KS} = \sum_{t=1}^{T} (y_t - \hat{\beta}_0 - \sum_{i=1}^{p} \beta_i x_{i,t-1})^2 \tag{II.2}$$

This is minimised by setting the coefficients to:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{KS}} = (X^T X)^{-1} X^T \boldsymbol{y} \tag{II.3}$$

Where $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\ldots}, \hat{\beta}_p)$ is the vector of coefficients and X^T denotes the transpose of X. This model, which uses all of the predictors and minimises the mean squared error is often called the 'kitchen sink' (KS) model in financial applications.

Because the Gauss-Markov assumptions are met, this estimator is the best unbiased linear estimator. However, its variance is dependant upon the variance of the noise term ϵ and the degree of linear dependence between the predictor. Financial time series are often very noisy and have a substantial degree of multicollinearity in their predictors. As such, one may reduce the expected mean squared error of the forecasts with some restrictions that reduce the variance of the estimator at the cost of introducing some bias.

Suppose one restricts the X^TX covariance matrix to be diagonal. That is, let $\overline{X^TX}$ be a nx(T-1) matrix such that its diagonal elements are equal to the corresponding diagonal element of the unrestricted covariance matrix X^TX, and that its off-diagonal elements are all equal to zero. Let us modify the OLS estimator by replacing X^TX with $\overline{X^TX}$:

$$\boldsymbol{\hat{\beta}_{rest1}} = \overline{X^TX}^{-1} X^T \boldsymbol{y} \tag{II.4}$$

Note that the restricted coefficient vector can be estimated by univariate linear regressions with OLS. Its first element is equal to the sum of the intercepts of of the p univariate regressions, and the i-th ($i > 1$) element is equal to the coefficient of the variable x_i from the univariate regression corresponding to the variable (Rapach, Strauss, and Zhou, 2010).

Additionally, impose another restriction. Instead of estimating using the coefficients from equation II.4 to generate the forecasts, divide them by the number of predictors p:

$$\boldsymbol{\hat{\beta}_{rest2}} = \frac{\boldsymbol{\hat{\beta}_{rest1}}}{p} \tag{II.5}$$

And the forecasts are calculated the following way:

$$\hat{y_t} = \hat{\beta}_{rest2,0} + \sum_{i=1}^{p} \hat{\beta}_{rest2,i} x_{i,t-1} + \epsilon_t \tag{II.6}$$

Note that this is equivalent to taking a simple average of the univariate forecasts:

$$\begin{aligned}
\frac{\sum_{i=1}^{p} \hat{y}_{i,t}}{p} &= \sum_{i=1}^{p} \frac{\hat{\beta}_{univar,i,0} + \hat{\beta}_{univar,i} x_{i,t}}{p} \\
&= \frac{\sum_{i=1}^{p} \hat{\beta}_{univar,i,0}}{p} + \sum_{i=1}^{p} \frac{\hat{\beta}_{univar,i}}{p} x_{i,t} \\
&= \frac{\hat{\beta}_{rest1,0}}{p} + \sum_{i=1}^{p} \frac{\hat{\beta}_{rest1,i}}{p} x_{i,t} \\
&= \hat{\beta}_{rest2,0} + \sum_{i=1}^{p} \hat{\beta}_{rest2,i} x_{i,t} \qquad \text{(II.7)}
\end{aligned}$$

Where $\hat{y}_{i,t}$ at the beginning is the forecast from the i-th univariate regression, $\hat{\beta}_{univar,i,0}$

is the intercept from the i-th univariate regression, and $\hat{\beta}_{univar,i,0}$ is the regression coefficient from the i-th univariate regression. The first equality follows by the definition of the $\hat{y}_{i,t}$-s. The third equality is uses the fact that the univariate regression coefficients are equal to the regression coefficients after restriction 1 is imposed, and the fact that the sum of the univariate regression intercepts equals the intercept of the regression with restriction 1 imposed. The last equation follows from the definition of the $\hat{\beta}_{rest2,i}$-s.

To sum up, I have shown that the uOLS method, which estimates univariate regressions with each of the predictors and then takes the simple average of the forecasts generated by the univariate regression, is equivalent to estimating the KS model with two restrictions. The first restriction is to estimate the regression coefficients by first ignoring the (cross-)covariances of the predictors, and the second restriction is to divide the estimated coefficients by the number of the univariate models. The first restriction decreases the number of the parameters that have to be estimated from the data. Without the restriction, the covariance matrix has p variance and $\frac{p(p-1)}{2}$ covariance parameters; with the restriction imposed, the only the p variances have to be estimated. The second restriction shrinks the coefficients to zero, similar to other regularisation method such as the LASSO or ridge regression (Rapach, Strauss, and Zhou, 2010), (Rapach and Zhou, 2020), (Tibshirani, 1996), (Hoerl and Kennard, 1970).

II.2 The concept and categorisation of inverse regularisation

An important characteristic of the shrinkage of uOLS is that it is 'set at a certain level'; the degree of shrinkage is not optimised in any way. In comparison, consider the LASSO or ridge regressions, which, using the notation from the previous subsection, estimate a linear model of the form in equation II.6, but instead of minimising the mean squared error, minimise the mean squared error plus a penalty term:

$$MSE_{penalised} = \sum_{t=1}^{T}(y_t - \hat{\beta}_0 - \sum_{i=1}^{p}\beta_i x_{i,t-1})^2 + \lambda\sum_{i=1}^{p}|\beta_i|^s \tag{II.8}$$

Where $s = 1$ gives the LASSO and $s = 2$ gives ridge regression.

The additional penalty term, which is the s-norm of the coefficient vector (not including the intercept), is multiplied by the hyperparameter λ. This penalty term shrinks the coefficients to zero. A higher lambda puts more emphasis on the penalty term in comparison to the mean squared error, and thus means a stronger degree of shrinkage & regularisation. A stronger regularisation results in lower variance but higher bias. A value of λ that optimises the degree of regularisation and the bias-variance trade-off can be estimated from the data(Tibshirani, 1996), (Hoerl and Kennard, 1970).

In contrast, the uOLS has no hyperparameter comparable to the λ of the LASSO and ridge. This means that the degree of the shrinkage or regularisation of the uOLS method is fixed. In the next sections, I show through a simulation study and an empirical application to forecasting the quarterly US equity premium that this fixed level of regularisation is indeed often suboptimal. This begs the question: can we improve upon the uOLS method by somehow reducing the level of regularisation?

Note that these methods would in many ways have the opposite effect on the uOLS method in comparison to the effect of the LASSO or ridge on the KS model. Whereas the LASSO and ridge increase the bias and decrease the variance of the KS model, this modification would have a smaller bias and higher variance than the uOLS benchmark. Additionally, the LASSO and ridge regressions are meant to prevent the KS model from overfitting; on the other hand, the uOLS is likely underfit and the sought method is meant to prevent that. To emphasize that the sought group of methods change the uOLS very much opposite to how the *regularisation* methods such as the LASSO or ridge change the KS model, I refer to these methods as *inverse regularisation* methods or *inverse regularisers*, or in short, *IR*-s.

There are several methods already present in the literature that can be considered inverse regularisers. However, these methods have not been adequately compared. To achieve this end, I propose a new categorisation of these methods and carry out a large scale comparison in the following sections.

The basis of the categorisation is the two restrictions, or two stages of the estimation of the uOLS method. The first stage is the estimation of the individual model, which is equivalent to imposing the diagonal covariance matrix restriction. Intuitively, this stage and restriction ignores *all* of the relationship between the predictors in the estimation of the individual models. Naturally, one way to inverse regularise the uOLS method is to not ignore *all* of the relationship between the predictors, but retain *some*. As such, stage one inverse regularisers are the methods that either estimate or define the individual models in a way that generalises the uOLS, allows for less regularisation and use more of the sample-relationship between the predictors in the estimation. Notable examples of stage one inverse regularisers include the complete subset regression of Elliott, Gargano, and Timmermann (2013), Elliott, Gargano, and Timmermann (2015) and Boot and Nibbering (2019), and the uNCL method, a new approach that I introduce in this paper.

The second stage of the calculation of the uOLS forecasts is the aggregation of the individual models. This is done by taking the simple average of the forecasts of the individual models, which is also equivalent to second restriction from the previous subsection (Rapach, Strauss, and Zhou, 2010). Taking the simple average of the individual forecasts ignores the difference in the biases and variances of the individual forecasts as well as their correlation structure. Forecasts with higher bias, higher variance and a high positive correlation with other forecasts contribute more to the expected squared error of the combined forecasts. As such, it might be possible to estimate combination weights and a bias term (as an intercept) to aggregate by. This could improve the performance of the uOLS method by reducing its bias and incorporating more information. I group these inverse regularisers that change the simple average combination method of the uOLS stage two inverse regularisers.

II.3 Stage two methods

This section introduces some important examples of stage II inverse regularisers. The list is not meant to be exhaustive at all, but instead focuses on the ELASSO and ERidge of Diebold and Shin (2019), two relatively new methods that are good representatives of this group and are evaluated in the simulation study in the later part of this paper.

II.3.1 Bates-Granger regression

Before introducing the ELASSO and ERidge, let us first consider the so called Bates-Granger regression (Bates and Granger, 1969). Suppose we have p individual forecasts of the variable of interest y_T, $f_{i,T}, \quad i = 1, 2, \ldots, p$ (which may come from the univariate OLS models, as in uOLS) that we want to aggregate in the following form:

$$f_{FC,T} = \alpha + \sum_{i=1}^{i} \beta_p f_{i,T} \tag{II.9}$$

Here, the β_i-s are the combination weights and α is a constant offset. We want to estimate the parameters that minimise the expected mean squared error of the combined forecast $f_{FC,T}$ over y_T. Define the error of the i-th individual forecast, $e_{i,t}$ as $e_{i,t} \coloneqq y_t - f_{i,t}$. If the covariance matrix of the errors and the bias terms of the individual forecasts $f_{i,t}$ are not time dependent, and the usual Gauss-Markov assumptions are met, then estimating the regression in equation II.9 with OLS is optimal within the class of unbiased estimators (Timmermann, 2006). Notably, the OLS-intercept is equal to the weighted average of the biases of the individual forecasts is expectation, thus the combination is unbiased.

In practice, the Bates-Granger regression combination of the individual forecasts usually underperforms the simple average combination (Timmermann, 2006). This stylised empirical finding is often referred to as the ”forecast combination puzzle”. The solution to this puzzle to a great extent lies in the fact that estimating the combination weights by the Bates-Granger regression introduces additional estimation error into the model (Smith and Wallis, 2009). On the other hand, this additional estimation error can be avoided by assuming the simple average weights, which are usually actually reasonable close to optimal in most applications (Smith and Wallis, 2009), (Claeskens et al., 2016), (Genre et al., 2013).

In the search for combination weights that outperform the simple average combination, one must abandon the class of unbiased models. Perhaps the largest class of these approaches consists of 'shrinkage' estimators. These estimators shrink the combination weights to the simple average in some way. In the following section, I introduce ELASSO & ERidge of Diebold and Shin (2019), two relatively new methods that are great representatives of this class.

II.3.2 ELASSO and ERidge

Let us consider the combination weight regression from equation II.9 of the form:

Instead of minimising the sample MSE (Bates-Granger approach) let us minimise a penalised form of the MSE:

$$MSE_{penalised} = \sum_{t=1}^{T}\left(y_t - \hat{\alpha} - \sum_{i=1}^{p}\beta_i f_{i,t}\right)^2 + \lambda\sum_{i=1}^{p}\left|\beta_i - \frac{1}{p}\right|^s \tag{II.10}$$

The case $s = 1$ is called the 'egalitarian LASSO' (ELASSO) and $s = 2$ is called 'egalitarian ridge (ERidge) (Diebold and Shin, 2019).

The estimation of the ELASSO or ERidge can be easily traced back to the estimation of a 'regular' LASSO or ridge model. Let $\overline{f}_t = \frac{1}{p}\sum_{i=1}^{p} f_{i,t}$ be the simple average combination. Then:

$$\begin{aligned}
MSE_{penalised} &= \sum_{t=1}^{T}\left(y_t - \hat{\alpha} - \sum_{i=1}^{p}\hat{\beta}_i f_{i,t}\right)^2 + \lambda\sum_{i=1}^{p}\left|\hat{\beta}_i - \frac{1}{p}\right|^s \\
&= \sum_{t=1}^{T}\left(y_t - \overline{f}_t + \overline{f}_t - \hat{\alpha} - \sum_{i=1}^{p}\hat{\beta}_i f_{i,t}\right)^2 + \lambda\sum_{i=1}^{p}\left|\hat{\beta}_i - \frac{1}{p}\right|^s \\
&= \sum_{t=1}^{T}\left((y_t - \overline{f}_t) - \hat{\alpha} + \sum_{i=1}^{p}(\frac{1}{p} - \hat{\beta}_i)f_{i,t}\right)^2 + \lambda\sum_{i=1}^{p}\left|\hat{\beta}_i - \frac{1}{p}\right|^s \\
&= \sum_{t=1}^{T}\left((y_t - \overline{f}_t) - \hat{\alpha} - \sum_{i=1}^{p}\delta_i f_{i,t}\right)^2 + \lambda\sum_{i=1}^{p}|\delta_i|^s
\end{aligned} \tag{II.11}$$

Where $\delta_i := \hat{\beta}_i - \frac{1}{p}$. What the equations tell us is that the ERidge or ELASSO coefficients and intercept can be easily estimated by fitting a regular LASSO or ridge of the $y_t - \overline{f}_t$-s on the individual forecasts $f_{i,t}$. For ridge, this regression has an analytic solution (Hastie, Tibshirani, and Friedman, 2001); in the case of the LASSO, numerical procedures, most commonly coordinate gradient descent is used (Friedman, Hastie, and Tibshirani, 2010).

In comparison to the regular LASSO and ridge, the ELASSO and ERidge are different only in the penalty term, which has an additional $\frac{1}{p}$ subtraction inside the absolute value signs. Intuitively, this difference means that ELASSO and ERidge penalises the combination weights β_i based on their distance (s-norm) from $\frac{1}{p}$, the equal weight combination, thus shrinking the combination weights towards the simple average. The hyperparameter λ determines the severity of the shrinkage. If $\lambda = 0$, we get back the Bates-Granger regression weights; on the other hand, as λ tends to infinity, we get back the equal weighted combination.

The extreme case of the Bates-Granger regression can be viewed as using all of the available information in the covariance structure and sample means of the individual forecasts $f_{i,t}$. The other extreme case, the simple average of the individual forecasts is equivalent to using no sample information about the means or covariance structure of the individual models and assuming they perform equally and have zero bias. In the context of the uOLS, using the ELASSO or ERdige to estimate the combination weights and the "offset" instead of using taking the simple average translates to inverse regularising the estimator, and, as we will see in the simulations, to optimising the bias-variance trade-off.

II.4 Stage one methods

In this subsection, I describe two stage one inverse regularisation methods, the complete subset regression of Elliott, Gargano, and Timmermann (2013) and my novel application of the negative correlation learning algorithm, originally introduced by Liu and Yao (1998) and later popularised by Brown, Wyatt, and Tino (2005).

II.4.1 Complete subset regression

Suppose we have p predictor variables $x_{i,t} \quad i = 1, 2, \dots, p$ and the variable we want to forecast, y_t, as previously. The complete subset regression (CSR) method consists of estimating individual models with OLS by regressing the y_t-s on each possible subset of size k of the predictors $x_{i,t}$, and then taking a simple average of the individual models. Formally, we estimate the j-th individual model in the following form:

$$y_t = \beta_0 + \sum_{i \in C(p,k,j)} \beta_i x_{i,t-1} \tag{II.12}$$

Where $C(p, k, j)$ is the j-th element of a set consisting of all possible combinations of size k of $\{1, 2, \dots, p\}$. For a given k and p, there are $\frac{p!}{(p-k)!k!}$ such combinations. These individual models are estimated by minimising the mean squared error (OLS). The combined forecast is a simple average of the individual forecasts, and can be written as:

$$\hat{y}_t = \frac{1}{\frac{p!}{(p-k)!k!}} \sum_{j=1}^{\frac{p!}{(p-k)!k!}} \left(\hat{\beta}_{0,j} + \sum_{l=1}^{p} \hat{\beta}_{l,j} x_{l,t-1} \right) \tag{II.13}$$

Where $\hat{\beta}_{0,j}$ is the estimated intercept term from the j-th individual model and $\hat{\beta}_{l,j}$ is the estimated coefficient of the l-th predictor from the j-th individual model. Note that if $\hat{\beta}_{l,j} = 0$ if the j-th individual model does not include the predictor $x_{l,t}$.

Note that with $k = 1$, the set of all possible combinations of size $k = 1$ can be ordered such that $C(p, k, j) = j$. It is easy to see that in this case, CSR is equivalent to uOLS. On the other hand, if $k = p$, there is only one subset of the p predictors with cardinality $k = p$. In this case CSR is equivalent to the KS model. As such, CSR can be viewed as a transition from the uOLS to the KS model as we increase k.

CSR is also a generalisation of uOLS. uOLS estimates the individual models by ignoring all covariances between the predictors. On the other hand, CSR (with $k > 2$) ignores most of the covariances but some when it estimates the individual models. Thus, it incorporates more information about the sample relationships of the predictors than the uOLS. In the second step, both the uOLS and CSR simply divide by the number of individual models. Elliott, Gargano, and Timmermann (2013) also shows that as we increase k the bias of the combined forecast decreases but the variance increases. As such, k may be considered a hyperparameter that can be optimised over some validation set to optimise the level of shrinkage.

Elliott, Gargano, and Timmermann (2013) also show analytically that the CSR is a shrinkage estimator and its relationship to the OLS coefficients. Let us define:

$$\boldsymbol{\hat{\beta}_{p,k}} = \left(\frac{1}{\frac{p!}{(p-k)!k!}} \hat{\beta}_{1,j}, \ldots, \frac{1}{\frac{p!}{(p-k)!k!}} \hat{\beta}_{p,j} \right) \tag{II.14}$$

Which is equivalent to the coefficient vector of CSR model from equation II.13 after bringing the division by the number of individual models inside the summation. Additionally, let $\hat{\beta}_{KS} = (\hat{\beta}_{1,KS}, \ldots, \hat{\beta}_{p,KS})^T$ be the coefficient vector of the kitchen sink model, where $\hat{\beta}_{l,KS}$ is the l-th coefficient estimate from the kithen sink model. Define $S_i \in R^{pxp}$ be a pxp matrix whose elements are all zero except the j-th diagonal elements if the j-th predictor is included in the i-th individual model. In this case, let the diagonal element be equal to one. Denote the time-independet covariance matrix of the predictors by Σ_X. Then, assuming that $\boldsymbol{\hat{\beta}_{KS}} \xrightarrow{p} \boldsymbol{\beta}$ for some $\boldsymbol{\beta} \in R^p$ as the sample size $T \to \infty$, the CSR coefficient estimates are a function of the OLS coefficient estimates (Elliott, Gargano, and Timmermann, 2013):

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{\boldsymbol{p,k}} &= \Lambda_{p,k}\hat{\boldsymbol{\beta}}_{\boldsymbol{KS}} + o_p(1) \\ \Lambda_{p,k} &= \frac{1}{\frac{p!}{(p-k)!k!}} \sum_{i=1}^{\frac{1}{\frac{p!}{(p-k)!k!}}} \left(S_i^T \Sigma_X S_i\right)^{-1} \left(S_i^T \Sigma_X\right) \end{aligned} \tag{II.15}$$

Here, if $\Lambda_{p,k}$ is diagonal, the CSR coefficients are approximately (with some error $o_p(1)$) equal to the corresponding KS coefficients multiplied by some scalar. In practice, $\Lambda_{p,k}$ is hardly ever diagonal and the CSR coefficients depend on all of the OLS coefficients (Elliott, Gargano, and Timmermann, 2013).

II.4.2 Univariate negative correlation learning

In this section, I introduce the negative correlation learning (NCL) algorithm and describe its novel application in this paper (uNCL). First, I show how to NCL algorithm is based on a lesser known decomposition of the squared error, the ambiguity decomposition. Then, I present the explanation of Brown, Wyatt, and Tino (2005) on why NCL performs well in the setting it was originally used in. Last, I describe my novel uNCL method, and note several differences between it and the original NCL.

II.4.2.1 The ambiguity Decomposition and the NCL algorithm

The ambiguity decomposition states that at a single data point, the quadratic error of the combined forecast is less or equal to the weighted average quadratic error of the combined forecasts. Formally:

$$(y_t - f_{FC,t})^2 = \sum_{i=1}^{p} w_i (y_t - f_{i,t})^2 - \sum_{i=1}^{p} w_i (f_{i,t} - f_{FC,t})^2 \tag{II.16}$$

Where y_t is a single arbitrary data point, $f_{i,t}, i = 1, 2, \ldots, p$ are the individual forecasts one combines, w_i are the weights corresponding to the forecasts $f_{i,t}$, $f_{FC,t} \coloneqq \sum_{i=1}^{p} w_i f_{i,t}$ is the combined forecast. For a derivation, see Krogh and Vedelsby (1994).

The ambiguity decomposition says that the squared error at a single data point is not equal to the weighted sum of the squared errors of the individual forecasts, but contains an additional term. The additional term, usually called the ambiguity term, takes up high values if the individual forecasts take up substantially different values from the combined forecast. The ambiguity term is subtracted from the weighted squared errors of the individual models, so a high value of the ambiguity term is advantageous. As such, the decomposition says that if two collections of individual forecasts have the same weighted squared error, but the first collection has a higher dispersion of the individual forecasts around the weighted mean forecasts than the second collection, then its combination has a smaller squared error.

This suggests that one shouldn't only care about the having a collection of individual forecasts that are accurate by themselves, but also a collection that is highly dispersed. Notice that most combination-based approaches do not care about the dispersion of the individual forecasts, or at least not directly. For example, the uOLS method estimates the individual models with OLS, which tries to maximise the accuracy of the univariate model by minimising its squared error over the train set. Similarly, in the machine learning community, where individual neural networks are often fitted and then aggregated by a weighted combination of the forecasts, the most common approach is to estimate the individual neural networks by minimising the squared error over the training set by some gradient-based algorithm.

Alternatively, the ambiguity decomposition presents an idea to improve performance by encouraging disperse forecasts. Adding up the squared errors over the train set $t = 1, 2, \dots, T$ we get:

$$\sum_{t=1}^{T}(y_t - f_{FC,t})^2 = \sum_{t=1}^{T}\left(\sum_{i=1}^{p} w_i(y_t - f_{i,t})^2 - \sum_{i=1}^{p} w_i(f_{i,t} - f_{FC,t})^2\right) \tag{II.17}$$

Taking the inner sum only for a fixed i, $i \in \{1, 2, \dots, p\}$, we get:

$$Contribution_{f_i} = \sum_{t=1}^{T}\left(w_i(y_t - f_{i,t})^2 - w_i(f_{i,t} - f_{FC,t})^2\right) \tag{II.18}$$

Assuming simple average weighting $f_{FC,t} = \frac{1}{p}\sum_{i=1}^{p} f_{i,t}$ this becomes:

$$Contribution_{f_i} = \frac{1}{p}\sum_{t=1}^{T}\left((y_t - f_{i,t})^2 - (f_{i,t} - f_{FC,t})^2\right) \tag{II.19}$$

I call this expression $Contribution_{f_i}$ because it can be seen as the contribution of the i-th individual model to the error of the combined forecast $Contribution_{f_i}$. It follows trivially from the definition of $Contribution_{f_i}$ that adding up the contributions of all the individual models over the train set is equal to the squared error of the combined forecasts over the train set:

$$\sum_{i=1}^{p} Contribution_{f_i} = \sum_{t=1}^{T}(y_t - f_{FC,t})^2 = MSE_{f_{FC}} \tag{II.20}$$

Naturally, one might minimise $Contribution_{f_i}$ when fitting the individual model i instead of the traditional approach of minimising the squared error, that is, only the first term of $Contribution_{f_i}$. Negative correlation learning is a generalisation of this idea. Instead of minimising the $Contribution_{f_i}$, NCL minimises the following expression for some $\lambda \in [0, 1]$:

$$NCL_{loss} = \frac{1}{p} \sum_{t=1}^{T} \left((y_t - f_{i,t})^2 - \lambda (f_{i,t} - f_{FC,t})^2 \right) \tag{II.21}$$

The only difference between NCL_{loss} and $Contribution_{f_i}$ is that the ambiguity term is multiplied by the hyperparameter λ.

This expression is minimised by a gradient-descent based algorithm.

From now on, assume that the individual forecasts come from some parametrised model with parameter vector $\boldsymbol{w_i}$ and that the parameters are time-independent. Also assume that the forecasts $f_{i,t}$ depend on time only through some predictors $\boldsymbol{x_{i,t}}$. As such, the individual forecasts are a function of the parameters $\boldsymbol{w_i}$ and the predictors $\boldsymbol{x_{i,t}}$; $f_{i,t} = g_i(\boldsymbol{w_i}, \boldsymbol{x_{i,t}})$ for some function g_i. We want to choose the value of the parameters $\boldsymbol{w_i}$ of the individual forecasts $f_{i,t}$ such that they minimise the NCL_{loss} of the i-th individual model.

Assuming that the combined forecasts $f_{FC,t}$ have a zero partial derivative with respect to g[1], we have[2]:

$$\frac{\partial NCL_{loss}}{\partial g_i(\boldsymbol{w_i}, \boldsymbol{x_{i,t}})} = \sum_{t=1}^{T} \left((g_i(\boldsymbol{w_i}, \boldsymbol{x_{i,t}}) + \lambda \sum_{j \neq i} (g_j(\boldsymbol{w_j}, \boldsymbol{x_{j,t}}) - f_{FC,t}) \right) \tag{II.22}$$

Using the chain rule, the gradient of the NCL_{loss} of the i-th individual model with respect to the parameters $\boldsymbol{w_i}$ is:

$$\frac{\partial NCL_{loss}}{\partial w_{i,k}} = \sum_{t=1}^{T} \left((g_i(\boldsymbol{w_i}, \boldsymbol{x_{i,t}}) + \lambda \sum_{j \neq i} (g_j(\boldsymbol{w_j}, \boldsymbol{x_{j,t}}) - f_{FC,t}) \right) \frac{\partial g_i(\boldsymbol{w_i}, \boldsymbol{x_{i,t}})}{\partial w_{i,k}} \tag{II.23}$$

Where $w_{i,k}$ is the k-th element of $\boldsymbol{w_i}$.

The gradient from equation II.23 is used to estimate the values of the parameters $\boldsymbol{w_i}$. However, the gradient of the i-th individual model is also a function of the values of the other models, both directly (through $g_j(\boldsymbol{w_j}, \boldsymbol{x_{j,t}})$ and indirectly (through the combined forecast $f_{FC,t}$). These values depend of $\boldsymbol{w_j}$, $j = 1, 2, \ldots, p$, $j \neq i$. If all $\boldsymbol{w_j}$, $j = 1, 2, \ldots, p$, $j \neq i$ values are set at the time we train the i-th model, then we train model i last. Consequently, at the time the parameters of model j ($j \neq i$) were set, at least one models parameters had not been set (the parameters of model i). Actually, if we train the individual models sequentialy, at the time the parameters of model l are estimated, there are only $l - 1$ individual models with parameters already set. Because the NCL_{loss} of

[1] This assumption is clearly not met, because $f_{FC,t} = \frac{1}{p} \sum_{i=1}^{p} g_i(\boldsymbol{w_i}, \boldsymbol{x_{i,t}})$ however, it actually yields the correct result. For more detail see Brown, Wyatt, and Tino (2005)

[2] I omit multiplication by a constant. This is inconsequential, because it does not change the optimal values of the parameters $\boldsymbol{w_i}$

model i depends on the values of the parameters of the other models, this is problematic.

To solve this problem, NCL updates the parameters of the individual models not sequentially, but parallel. The process is as follows. First, the p number of individual models and their parametrisation is chosen. Then, initial values of the parameters are set for each model (this can be done randomly or by setting the initial parameters to some preset constant). At this point, a loop starts. Using the current values of the parameters, the forecasts of each of the individual models are calculated and then are combined by equal weights to yield the combined forecasts. These individual forecasts and the combined forecast is used to calculate the gradients of each individual model parallel. Then, the parameters of each model are updated at the same time by subtracting the gradients from equation II.23 times the *learning rate*. At this point the loop is started over. The loop usually ends either after a initially specified number of iterations or if the error does not improve on the train or some validation data.

II.4.2.2 The hyperparameter lambda

From equation II.21, NCL minimises the following function:

$$NCL_{loss} = \frac{1}{p}\sum_{t=1}^{T}\left((y_t - f_{i,t})^2 - \lambda(f_{i,t} - f_{FC,t})^2\right) \tag{II.24}$$

Which, as noted previously, differs from the contribution of the individual model i to the error of the combined forecast only in that the ambiguity term is multiplied by the hyperparameter $\lambda \in [0, 1]$.

The boundaries on the value λ can take are suggested by the ambiguity decomposition. If $\lambda = 1$, NCL minimises $Contribution_{f_i}$ directly. A λ higher than 1 would, base on the ambiguity decomposition, result in individual forecasts that are too much dispersed around the combined forecast. The other extreme, when $\lambda = 0$, is equivalent to ignoring the ambiguity term and only minimising the mean squared error. As such, the value of λ determines the degree to which the estimation takes into account not only the error of the individual model, but also its relationship to the other models; the higher the lambda, the more ambiguity is taken into account.

Naturally, one might think that the optimal value of λ should always be 1, because that minimises the contribution of the individual models to the combined forecast. Although this view in intuitive, it is wrong. Reeve and Brown (2018) prove that there is always a $\lambda < 1$ value for which the combined forecast has a squared error lower than for $\lambda = 1$. Brown, Wyatt, and Tino (2005) also shows in a cross sectional simulation study that the MSE decreases as a function of λ, except for a close neighbourhood of $\lambda = 1$, where it sharply increases. Reeve and Brown (2018) suggest that the weak performance of NCL when $\lambda = 1$ is a result of overfitting; just as minimising of the MSE over the train set can overfit a model, minimising the contribution of an individual model directly (which is equivalent to NCL with $\lambda = 1$) overfits the model to the sampling errors. Later, my own simulations and the empirical application will also show that my uNCL method also performs poorly when $\lambda = 1$ in line with the previous results from the literature.

II.4.2.3 Managing diversity: an interpretation of NCL

In the ensemble learning literature, it is a well-known empirical finding that a combination of individual predictions work best when the individual models are substantially different from one another in some respect. As such, many ensemble learning methods, for example bagging or boosting, are motivated by the desire to create a 'different' or 'diverse' set of individual predictions (Brown, Wyatt, Harris, et al., 2005). In a very similar fashion, NCL was originally introduced to train a set of neural networks with 'diverse' forecasts (Liu and Yao, 1998).

What the 'diversity' of the individual forecasts means can be best captured by the lesser known bias-variance-covariance decomposition for regression problems (Brown, Wyatt, and Tino, 2005). Suppose we have a dataset of size T of the input vectors $\boldsymbol{x_t}$ and the variable we want to forecast, y_t. That is, we have the data $(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \dots, (\boldsymbol{x_T}, y_T)$ drawn independently form the joint distribution of the predictors x_t and variable of interest y_t, $p(\boldsymbol{x}, y)$. We want to forecast the y_t-s as a parametric function of the predictors $\boldsymbol{x_t}$-s: $\hat{y_t} = g(\boldsymbol{w}, \boldsymbol{x_t})$. We want to set the parameters to minimise the expected squared error:

$$E[(g(\boldsymbol{w}, \boldsymbol{x_t}) - y_t)^2] = \int g(\boldsymbol{w}, \boldsymbol{x_t}) - y_t)^2 p(\boldsymbol{x_t}, y_t) d(\boldsymbol{x_t}, y_t) \qquad \text{(II.25)}$$

From now on, I omit the $\boldsymbol{w}$ and $\boldsymbol{x_t}$ arguments of g for the purposes of brevity, unless I deem the simpler notation confusing.

According to the bias-variance decomposition, assuming a noise level of zero for the sake of simplicity, the expected squared error from the previous equation decomposes into:

$$E[(g - y_t)^2] = E[g - y_t]^2 + E[(g - E[g])^2] = bias^2 + variance \qquad \text{(II.26)}$$

In general, there is a trade-off between the two components; decreasing one usually leads to an increase of the other. This is the bias-variance trade-off.

Now assume that we have p individual estimators $g_i(\boldsymbol{w_i}, \boldsymbol{x_{i,t}})$. Also, define the previous estimator g as the equal weighted average of the individual estimator g_i-s:

$$g(\boldsymbol{w}, \boldsymbol{x_t}) = g(\boldsymbol{w_1}, \boldsymbol{w_2}, \dots, \boldsymbol{w_p}, \boldsymbol{x_{1,t}}, \boldsymbol{x_{2,t}}, \dots, \boldsymbol{x_{p,t}}) = \frac{\sum_{i=1}^{p} g_i(\boldsymbol{w_i}, \boldsymbol{x_{i,t}})}{p} \qquad \text{(II.27)}$$

Because g is a linear combination of the g_i-s, the bias variance decomposition further decomposes into a bias-variance-covariance decomposition (Brown, Wyatt, and Tino, 2005):

$$E[(g - y_t)^2] = \overline{bias} + \frac{1}{p}\overline{variance} + \frac{p-1}{p}\overline{covariance} \tag{II.28}$$

Where the $\overline{bias}$, $\overline{variance}$ and $\overline{covariance}$ terms are the averages of the biases, variances and covariances of the individual models that g is the aggregate of:

$$\begin{aligned}
\overline{bias} &= \frac{\sum_{i=1}^{p} bias_i}{p} = \frac{\sum_{i=1}^{p} E\left[g_i - y_t\right]}{p} \\
\overline{variance} &= \frac{\sum_{i=1}^{p} variance_i}{p} = \frac{\sum_{i=1}^{p} E\left[(g_i - E\left[g_i\right])^2\right]}{p} \\
\overline{covariance} &= \frac{\sum_{i=1,j=1,j\neq i}^{p} covariance_{i,j}}{p(p-1)} = \\
&= \frac{\sum_{i=1,j=1,i\neq j}^{p} E\left[(g_i - E\left[g_i\right]) \cdot (g_j - E\left[g_j\right])\right]}{p(p-1)}
\end{aligned} \tag{II.29}$$

Where I have omitted the dependency of the g_i-s on their parameter vectors $\boldsymbol{w_i}$ and the predictors $\boldsymbol{x_{i,t}}$. It is quite easy to see that this bias-variance-covariance decomposition results from the bias-variance decomposition of g using the linearity of bias and the formula for the variance of a sum of random variables:

$$\begin{aligned}
bias(g) = E[g - y_t] &= E\left[\frac{\sum_{i=1}^{p} g_i - y_t}{p}\right] = E\left[\frac{bias_i}{p}\right] = \overline{bias} \\
variance(g) = variance\left(\sum_{i=1}^{p} \frac{g_i}{p}\right) &= \frac{1}{p^2} variance\left(\sum_{i=1}^{p} g_i\right) \\
&= \frac{1}{p^2}\left(\sum_{i=1}^{p} variance_i + \sum_{i=1,j=1,i\neq j}^{p} covariance_{i,j}\right) \\
&= \frac{1}{p}\overline{variance} + \frac{p-1}{p}\overline{covariance}
\end{aligned} \tag{II.30}$$

The decomposition, similar to the ambiguity decomposition, suggests that one should not only care about the performance of the individual models by themselves (represented by the $\overline{bias}$ and $\overline{variance}$ terms in the decomposition), but should also care about the relationship of the individual models to one another. In the case of the ambiguity decomposition, the relationship of the individual forecasts is measured by their 'dispersion' around the mean forecasts; in the bias-variance-covariance decomposition, it is captured by their covariance.

Brown, Wyatt, and Tino (2005) make a connection between the ambiguity and bias-variance-covariance decomposition. They show that:

$$
\begin{aligned}
E\left[\frac{1}{p}\sum_{i=1}^{p}(g_i - y_t)^2\right] &= \overline{bias}^2 + \Omega \\
E\left[\frac{1}{p}\sum_{i=1}^{p}(g_i - g)^2\right] &= \Omega - \left[\frac{1}{p}\overline{variance} + \frac{p-1}{p}\overline{covariance}\right]
\end{aligned}
$$

Where the two expressions on the left are the expectation of the two terms, the mean squared error and the ambiguity term of the ambiguity decomposition. The expression Ω is the interaction between the two sides:

$$\Omega = \overline{variance} + \frac{1}{p}\sum_{i=1}^{p}(E[g_i] - E[g])^2 \tag{II.31}$$

Ω is present in both the mean squared error term and the ambiguity term, so when we substract the two terms from one another, it cancels out and we get the bias-variance-covariance decomposition back.

The NCL minimises a the mean squared error minus the ambiguity term times λ. Multiplying the ambiguity term by λ, then subtracting it out from the mean squared error term and taking expectation, we have:

$$
\begin{aligned}
&E\left[\frac{1}{p}\sum_{i=1}^{p}(g_i - y_t)^2 - \lambda\frac{1}{p}\sum_{i=1}^{p}(g_i - g)^2\right] \\
&= \overline{bias}^2 + (1-\lambda)\Omega + \lambda * \left[\frac{1}{p}\overline{variance} + \frac{p-1}{p}\overline{covariance}\right]
\end{aligned} \tag{II.32}
$$

Substituting Ω from equation II.31, we have:

$$
\begin{aligned}
&E\left[\frac{1}{p}\sum_{i=1}^{p}(g_i - y_t)^2 - \lambda\frac{1}{p}\sum_{i=1}^{p}(g_i - g)^2\right] \\
&= \overline{bias}^2 + \frac{p - \lambda * (p-1)}{p}\overline{variance} + \frac{\lambda(p-1)}{p}\overline{covariance} \\
&+ \frac{1-\lambda}{p}\sum_{i=1}^{p}(E[g_i] - E[g])^2
\end{aligned} \tag{II.33}
$$

This equation show the relationship between the NCL_{loss} and the bias-variance-covariance decomposition. Not taking into account the last term, which is not present in the bias-variance-covariance decomposition, the hyperparameter λ can be given a new interpretation. If $\lambda_1 > lambda_2$, there is a greater emphasis on the covariance term and a weaker emphasis on the variance term for λ_1 than λ_2. The case of $\lambda = 1$ gives back the bias-variance-covariance decomposition, whereas the other extreme, $\lambda = 0$ puts zero emphasis on the covariance term. Following this reasoning, Brown, Wyatt, and Tino (2005) claims NCL is to be interpreted as a method that optimises the trade-off between the accuracy of the individual models (as measured by the $\overline{bias}$ plus $\frac{1}{p}\overline{variance}$) and their 'diversity' (as measured by the average covariance of the individual models, $\frac{p-1}{p}\overline{covariance}$. They call this trade-off "the accuracy-diversity trade-off" and the optimisation of this trade-off by NCL "managing diversity". Apart from theory, some empirical results also suggest that NCL, when applied to neural networks, leads to lower average covariance of the individual forecasts.

II.4.2.4 Applications of NCL: neural networks and the new uNCL

In this subsection, I describe how NCL have previously been applied, and how my own application of the algorithm, which I call uNCL, differs from it.

The NCL learning algorithm, as described previously, did not assume anything about the individual models $g_i(\boldsymbol{w_i}, \boldsymbol{x_{i,t}})$, apart from them being parametric. Despite this fact, previous application, the algorithm has been applied only to very specific methods. The algorithm was developed in the late 1990s, specifically with the narrow aim of applying it to train a diverse set of *neural networks* (Liu and Yao, 1999). Later applications also remained largely concentrated in the neural network literature, see for example S. Wang, Tang, and Yao (2009), Liu, Zhao, and Pei (2014), Liu, Yao, and Higuchi (2000), Sheng et al. (2017)[3]. The algorithm has also been applied to classification problems (S. Wang, Chen, and Yao, 2010), and lately to deep neural networks (Z. Shi et al., 2018), (Buschjager, Pfahler, and Morik, 2020). In contrast to my study, NCL has mainly been applied to cross-sectional, rather than time series data, with the exceptions of Waleed et al. (2009) and Liu and Yao (1998).

My uNCL method breaks this tradition of applying NCL in a rather narrow context, despite the fact that it is based on a theoretical result (the ambiguity decomposition) that applies in a much wider context.

Consider the uOLS method. As described previously, this method consists of a) estimating univariate predictive regressions of the variable of interest with each of the available predictors with OLS, b) generating forecasts of the variable of interest from the univariate regressions, and then c) taking a simple average of the individual forecasts from step b) as the final forecast.

I propose a simple modification to the uOLS method. Instead of estimating the univariate regressions with OLS, I estimate them with the NCL algorithm, keeping everything else equal. I call this proposed method uNCL.

This differs from the usual neural network based application of NCL in several respects.

[3] As an interesting exception, see the NCL-based version of SVM in Hu and Mao (2009)

First, the individual models of uNCL are simple, linear models, whereas neural networks are highly non-linear and complex. On the other hand, uNCL trains linear models. It is conceivable that NCL was able to train a diverse set of forecasts with neural networks because each neural network could learn different patterns on the data due to the flexible nature of these models. uNCL trains much less flexible linear models that may not be able to specialise well to different patterns in the dataset.

Additionally, the neural network NCL has been applied to highly non-linear simulated and empirical datasets[4]. When the dataset comes from a more complex underlying model and has substantial nonlinearities, it probably has more distinct patterns that the individual models can 'specialise' to with NCL. uNCL will be applied to a linear dataset in the simulations in this paper, which might also hamper its ability to train a diverse set of individual forecasts.

Perhaps most notably, there is a stark difference in the way the predictors are handled by uNCL and previous neural networks-based applications of the NCL algorithm. The individual neural networks that make up the ensemble in NN-based applications of NCL always have the same architecture; the same number of nodes, activation functions, number of hidden layers, etc[5]. Most importantly, all of the individual NNs have all of the predictors as their inputs. This is in contrast to uNCL, which only gives one of the predictor as an input to each individual model.

To illustrate the importance of this difference in inputs, consider the NCL_{loss} of the individual models of uNCL. The NCL_{loss} of the individual model i from equation II.21 and the $NCL_{gradient}$ from equation II.23 are:

$$\begin{aligned} NCL_{loss} &= \sum_{t=1}^{T} \left((y_t - f_{i,t})^2 - \lambda (f_{i,t} - f_{FC,t})^2 \right) \\ \frac{\partial NCL_{loss}}{\partial w_i} &= \sum_{t=1}^{T} \left((f_{i,t} + \lambda \sum_{j \neq i} (f_{j,t} - f_{FC,t}) \right) \frac{\partial f_{i,t}}{\partial w_i} \end{aligned} \tag{II.34}$$

The individual models of uNCL are of the form:

$$f_{i,t} = \hat{\beta}_{0,i} + \hat{\beta}_i x_{i,t} \tag{II.35}$$

And the combined forecast can be expressed as follows:

$$f_{FC,t} = \frac{\sum_{i=1}^{p} \hat{\beta}_{0,i}}{p} + \frac{\sum_{i=1}^{p} \hat{\beta}_i x_{i,t}}{p} \tag{II.36}$$

[4] See the 'Friedman-data' in Brown, Wyatt, and Tino (2005) for an example.

[5] The only difference between the individual networks is that their training starts from different initial parameter values.

The derivative at the end of the gradient is:

$$\begin{aligned}
\frac{\partial f_{i,t}}{\partial \hat{\beta}_{0,i}} &= \frac{\partial(\hat{\beta}_{0,i} + \hat{\beta}_i x_{i,t})}{\partial \hat{\beta}_{0,i}} = 1 \\
\frac{\partial f_{i,t}}{\partial \hat{\beta}_i} &= \frac{\partial(\hat{\beta}_{0,i} + \hat{\beta}_i x_{i,t})}{\partial \hat{\beta}_i} = x_{i,t}
\end{aligned}$$

Substituting these expressions into the NCL_{loss}, we get:

$$\begin{aligned}
NCL_{loss} &= \sum_{t=1}^{T} \left[(y_t - \hat{\beta}_{0,i} - \hat{\beta}_i x_{i,t})^2 - \lambda \left(\hat{\beta}_{0,i} + \hat{\beta}_i x_{i,t} - \frac{\sum_{j=1}^{p} \hat{\beta}_{0,j}}{p} - \frac{\sum_{j=1}^{p} \hat{\beta}_i x_{i,t}}{p} \right)^2 \right] \\
&= \sum_{t=1}^{T} \left[(y_t - \hat{\beta}_{0,i} - \hat{\beta}_i x_{i,t})^2 - \lambda \left(\frac{(p-1)(\hat{\beta}_{0,i} + \hat{\beta}_i x_{i,t}) - (\sum_{j=1,j\neq i}^{p} \hat{\beta}_{0,j} + \hat{\beta}_j x_{j,t})}{p} \right)^2 \right]
\end{aligned}$$

And substituting back into the $NCL_{gradient}$ yields:

$$\begin{aligned}
\frac{\partial NCL_{loss}}{\partial \hat{\beta}_{0,i}} &= \sum_{t=1}^{T} \left(\hat{\beta}_{0,i} + \hat{\beta}_i x_{i,t} - y_t + \lambda \sum_{j=1,j\neq i}^{p} \hat{\beta}_{0,j} + \hat{\beta}_j x_{j,t} - \frac{\sum_{k=1}^{p} \hat{\beta}_{0,k} + \hat{\beta}_k x_{k,t}}{p} \right) \\
&= \sum_{t=1}^{T} \left(\frac{p - \lambda(p-1)}{p} (\hat{\beta}_{0,i} + \hat{\beta}_i x_{i,t}) + \frac{\lambda}{p} \sum_{j=1,j\neq i}^{p} \hat{\beta}_{0,j} + \hat{\beta}_j x_{j,t} \right) \\
\frac{\partial NCL_{loss}}{\partial \hat{\beta}_i} &= \sum_{t=1}^{T} \left(\frac{p - \lambda(p-1)}{p} (\hat{\beta}_{0,i} + \hat{\beta}_i x_{i,t}) - y_t + \frac{\lambda}{p} \sum_{j=1,j\neq i}^{p} \hat{\beta}_{0,j} + \hat{\beta}_j x_{j,t} \right) x_{i,t} \qquad \text{(II.37)}
\end{aligned}$$

The NCL_{loss} equation tells us that the loss of the individual model i, which corresponds to the predictor $x_{i,t}$, also depends on the other predictors to some degree.

The gradient equations describe the relationship more intuitively. The gradient has a term that depends only on the parameters that are optimised in the individual model i, that is, $\hat{\beta}_{0,i}$ and $\hat{\beta}_i$, and a term that depends on the parameters that are optimised in the other individual models. The parameter λ determines the degree of emphasis put on each of the terms; a higher value of λ means more emphasis on the parameters of the other model. As such, while uNCL *directly* only optimises the parameters belonging to a single predictor during the training of an individual model, it also takes the other predictors into account *indirectly* to some degree. Because, in contrast to uOLS, uNCL incorporates some information about all the predictors when estimating the parameters corresponding to a single predictor, uNCL may behave similar to or most like a stage I inverse regulariser.

II.5 Forecasting the US equity premium

Forecasting the equity premium has been a considerably difficult task, and most simple linear regressions based on a single variable, or multivariate linear regressions based on several variables do not outperform the historical average benchmark out-of-sample (Welch and Goyal, 2007). The general consensus tends to be that such methods can forecast the equity premium, although the forecastable component is rather small (Rapach, 2013). However, as Campbell and Thompson (2007) argue, an R^2_{OoS} as low as 0.5% can be significant in an economic sense.

In recent years, a number of different approaches, suitable for analyzing noisy data, with many, potentially spurious predictors have seen widespread applications in finance. Rapach, Strauss, and Zhou (2013) apply the LASSO to forecasting the equity premium. Gu, Kelly, and Xiu (2020) apply a large set of machine learning tools to analyze the time series predictability of monthly individual stock returns. Chinco, Clark-Joseph, and Ye (2019) use the LASSO to predict individual stock returns one minute ahead. Freyberger, Neuhierl, and Weber (2020) apply a non-parametric version of the LASSO to analyze nonlinear relationships between numerous firm characteristics and the cross section of stock returns. Kozak, Nagel, and Santosh (2020) apply the LASSO to forecast the stochastic discount factor with a large set of firm characteristics. Han et al. (2020) apply the ELASSO of Diebold and Shin (2019) to forecast cross sectional returns using firm characteristics.

Besides the time series and cross section of equity returns, other areas of finance and macroeconomic have also seen a large number of applications of the same or similar techniques with considerable success. Oil markets have been a particularly active field recently. Crude oil markets have been an active Zhang, Ma, B. Shi, et al. (2018) and Zhang, Ma, and Wei (2019) use an iterated version of the forecast combination approach to forecast oil prices and oil futures return volatility, respectively. Zhang and Y. Wang (2022) apply a forecast combination and PCA hybrid to forecast oil futures market returns. Zhang, W., and Y. Wang (2022) use a LASSO and PCA hybrid to forecast crude oil return volatility. Zhang, Wei, Zhang, et al. (2019) compare the LASSO and forecast combination in forecasting oil market volatility. Apart from oil market, Elliott, Gargano, and Timmermann (2015) and Huang et al. (2022) apply CSR and a modified PCA-based technique on macroeconomic data, respectively.

As the previous list shows, probably the most popular method in finance has been the LASSO and its variants recently (Rapach and Zhou, 2020), (Elliott, Gargano, and Timmermann, 2013), (Gu, Kelly, and Xiu, 2020), (Freyberger, Neuhierl, and Weber, 2020), (Kozak, Nagel, and Santosh, 2020). The LASSO is usually given all of a large set of potential predictors as inputs, and is thus a penalised version of the 'kitchen sink' model, which is simply a multivariate linear regression that uses all of the predictors.

However, I note that the LASSO can be applied in other ways. For example, Rapach and Zhou (2020) estimates univariate regressions with each predictor, and then uses the LASSO the select a subset of the forecasts generated by the univariate regression to be aggregated by equal weighting. They find that this application of the LASSO is actually superior to directly using the LASSO on all predictors as penalised version of the 'kitchen sink' in forecasting the US equity premium.

I believe that the equity premium, and in general the finance literature has given relatively too much attention to the LASSO as used to penalise the kitchen sink. The popularity of the method is understandable, because it results in 'sparse' models with only a handful of predictors with non-zero coefficients, which lends itself to easy interpretation[6]. However, techniques based on the uOLS, which I call IR-s, can achieve superior forecasting performance.

I note that another approach to improve forecasts of the equity premium has been to impose restrictions of the forecasts. Campbell and Thompson (2007) restricts the signs of the coefficients in univariate regressions based on theoretical considerations, and also restricts the forecasts to be positive. They find that the forecasts outperform the historical average benchmark after the restrictions are imposed, although they do not outperform without the restrictions (Welch and Goyal, 2007). Other papers build on this idea and develop more advanced restrictions. Pettenuzzo, Timmermann, and Valkanov (2014) use the nonnegativity restriction to alter the posterior distribution of the parameters. Zhang, Wei, Ma, et al. (2019), Dai et al. (2020) and Li and Tsiakas (2017), among many others, apply similar restrictions. The general finding of the literature is that imposing some theoretically motivated restrictions tends to improve the performance of most forecasts.

An important aspect of the predictable component of the US equity premium is that predictability is clustered in short periods with high predictability and longer periods of weak or no predictability (Farmer, Schmidt, and Timmermann, 2022), (Baltas and Karyampas, 2018), (Haase and Neuenkirch, 2022). Most often, the periods of high predictability correspond with recessions, and the periods of low predictability with expansions, and predictability is thought to be connected to the predictability of the business cycle (Rapach, Strauss, and Zhou, 2010). As such, it is usual practice to also evaluate the forecasting models separately in expansive and recessive periods of the business cycle, and to examine the time-dependence of the performance of the forecasting model. The rolling mean squared error plot, which plots the mean squared error of the forecasting model up to the time on the x axis, is a commonly used graphical tool to illustrate the latter (Campbell and Thompson, 2007).

[6] I note that this observed sparsity actually may lend itself to misinterpretation, as Giannone, Lenza, and Primiceri (2021) show, the set of predictors with non-zero coefficients is often unstable with the LASSO

Simulation Study

This section presents the simulation study that I carried out to answer my research questions. I aim to a) provide a large scale comparison of the stage I and stage II inverse regularisation methods from the previous section, b) show that the uNCL works as a stage I inverse regulariser, and to c) show that stage I inverse regularisers tend to outperform both stage II inverse regularisers and normal regularisers such as the LASSO or ridge.

The following section is structured as follows. First, I define the data generating process and the methods I compare. Then, I present the MSEs of the methods with fixed parameter values. This is followed by robustness check to optimising the hyperparameters from the data. Having laid out the main results, I offer an explanation by estimating the bias-variance and bias-variance-covariance decomposition of each method.

III.1 The data generating process

I assume that there are 8 predictors $x_{i,t}$ of the variable of interest, y_t, where $i = 1, 2, \dots, 8$ denotes the index of the predictor and $t = 1, 2, \dots, T$ denotes 'time'. I assume a constant time series length of $T = 300$ for all of the simulations, which closely mimics the length of the empirical time series from the application[1].

Following the approach of Elliott, Gargano, and Timmermann (2013), I generate the $x_{i,t}$-s from a multivariate normal distribution with zero mean $\boldsymbol{\mu} = (0, 0, \dots, 0)$ and covariance matrix Σ equal to

$$\rho = \begin{pmatrix} 1 & \rho & \rho & \rho & \dots & \rho \\ \rho & 1 & \rho & \rho & \dots & \rho \\ \rho & \rho & 1 & \rho & \dots & \rho \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \rho & \rho & \dots & 1 \end{pmatrix} \tag{III.1}$$

That is, the variances of the x_i-s are normalized to one and I assume a constant

[1] By fixing the length of the time series, I follow Elliott, Gargano, and Timmermann (2013) in not considering the effect the length of the time series has on the methods.

covariance ρ. In my simulations, I consider $\rho = 0, 0.4, 0.8$ to study the performance of the methods in low (no), medium and high covariance structures.

The variable of interest y_{t+1} is a linear function of the $x_{i,t}$-s plus a random Gaussian noise:

$$y_{t+1} = c \sum_{i=1}^{8} x_{i,t} + e_t \tag{III.2}$$

Where e_t follows the standard normal distribution with $Corr(e_t, x_{i,t}) = 0$, $i = 1, 2, \dots, 8$ and c is a constant that determines the *signal-to-noise ratio*[2]. If c is greater (smaller), a relatively greater (smaller) part of the variables of interest y_{t+1} are explained by the $x_{i,t}$-s, that is, there is a stronger *signal.* I choose the different values of c such that they correspond to R^2-s of 1%, 2.5%, 5%, 10% and 25%[3]. I note that this specification means that all of the predictors $x_{i,t}$ have the same coefficient of c in the data generating process. As such, my simulation does not deal with either predictors with different predictive power or spurious predictors.

I generate 100 time series for each of the possible combination of the parameters (the noise level $R^2 = 1\%, 2.5\%, 5\%, 10\%, 25\%$ and the predictor correlation $\rho = 0, 0.4, 0.8$).

I estimate several different models that all use the predictors $x_{i,t}$ to forecast y_{t+1}. Following the approach used by Welch and Goyal (2007), Rapach, Strauss, and Zhou (2010), Rapach (2013) and Elliott, Gargano, and Timmermann (2013) among many others in applications to forecasting the US equity premium, I evaluate the methods with an expanding window. More specifically, I first divide the sample of 300 observations into an initial estimation sample of the first 100 observations, which is only used for model estimation, and an evaluation sample of the last 200 observations. The first observation in the evaluation sample, y_{102} is forecast by fitting each method on the initial estimation sample of the first 100 observations ($y_{t+1}, \boldsymbol{x_t}, t = 1, 2, \dots, 100$, and then using this fitted models to generate the forecast of y_{102}. The next observation y_{103} in the evaluation sample is forecast by now fitting the models on the expanded dataset $y_{t+1}, \boldsymbol{x_t}, t = 1, 2, \dots, 101$, and using these fitted models to generate the forecasts of y_{103}. I proceed in these manner by iteratively reestimateing each model on an expanded data to forecast the proceeding y_t-s.

Importantly, this expanding window estimation and evaluation scheme only uses information available at time t to forecast the variable y_{t+1}. In this sense, it is akin to a

[2] This approach is similar to that of Elliott, Gargano, and Timmermann (2013). The difference is that they determine the signal-to-noise ratio by setting the standard error of the residual e_t, and keep the coefficients of the independent variables fixed. In contrast, I keep the standard error of the residuals constant at 1 and set the coefficients of the predictors to get the desired R^2.

[3] Most papers report an R^2_{OOS} measure below 5% for the models they consider; see Rapach (2013). As this is a measure of out-of-sample forecasting performance, the actual R^2 of the underlying data generating process should be somewhat, but not excessively, higher. This is covered by the range of R^2 values I consider in this study.

real-time forecasting, and as such it is adequate way to compare forecasting performance.

I calculate the squared error $(\hat{y_{i,t}} - y_t)^2$ of the forecasts $y_{i,t}, t = 102, 103, \ldots, 301$ for each method i. Then, the squared errors are averaged for each method to get the mean squared error for the given method for the given time series:

$$MSE_{i,j} = \frac{1}{200} \sum_{t=101}^{300} (\hat{y}_{i,t+1} - y_{t+1})^2 \tag{III.3}$$

Where i is the index of the method and j is the index of the time series. I carry out this process for each of the 100 time series while keeping the parameters of the time series, the R^2 and the predictor cross-correlation ρ constant. Then, I take an average of the MSE of each method over the 100 time series to get the mean squared error of each method:

$$MSE_i = \frac{1}{100} \sum_{j=1}^{100} MSE_{i,j} \tag{III.4}$$

This MSE_i value is normalised by dividing it with the average MSE of the uOLS over the 100 time series:

$$\frac{MSE_{i,normalised}}{MSE_{uOLS}} = \frac{\sum_{j=1}^{100} \sum_{t=101}^{300} (\hat{y}_{i,j,t+1} - y_{j,t+1})^2}{\sum_{t=101}^{300} (\hat{y}_{uOLS,j,t+1} - y_{j,t+1})^2} \tag{III.5}$$

Where $MSE_{i,normalised}$ is the normalised MSE of method i, the performance measure I use, MSE_{uOLS} is the mean squared error of the uOLS method averaged over the 100 individual time series, j is the index of the time series, and $\hat{y}_{i,j,t+1}$ of the variable of interest $y_{j,t+1}$ from method i for the time series j[4]. The individual forecasts, as described previously, come from an expanding window estimation scheme that simulates real-time out-of-sample forecasting and updates the dataset used to estimate the model each period to include all of the available past information.

The reasoning behind this normalisation is that this makes it easier to interpret the resulting values. The special cases of some models (namely, uNCL with $\lambda = 0$ and CSR with $k = 1$) have a MSE of 1 after normalisation, and the MSEs of the other methods can be interpreted as having a relatively higher or lower MSE than the uOLS.

I use $MSE_{i,normalised}$ to compare the performance of the models. For the models that have hyperparameters, I first estimate the normalised mean squared errors for a grid of hyperparameter values, to compare how the methods perform with different values of the hyperparameters. After the simulation, I have a normalised MSE value for each of the methods as a function of the predictor cross-correlation ρ, the DGP R^2 and the value of the hyperparameter of the method.

[4] Note that both the numerator and denominator should include a division by $\frac{1}{200}$ and $\frac{1}{100}$ to average over the data points and the time series, but they cancel each other out.

Later, I evaluate each of the methods with parameter values that are not fixed, but validated from previous out-of-sample performance from the data. To do this, I also need an initial 'hyperparameter validation' window, which I set at length 30. This means that the first 30 out-of-sample forecasts, $y_{102}, y_{103}, \dots, y_{131}$, which I generate as previously, are not used in the calculated of the $MSE_{i,normalised}$ of the validation data. Instead, the $MSE_{i,normalised}$ is calculated only using the validated forecasts of the last 170 data points[5]. I use and expanding window to validate the hyperparameters as well. This means that I use the first 30 out-of-sample forecasts $\hat{y}_{102,\alpha}, \hat{y}_{103,\alpha}, \dots, \hat{y}_{131,\alpha}$ to validate the hyperparameter α for the first observation, y_{132} in the validated evaluation sample. Subsequently, I always expand the validation sample by the most recent out-of-sample forecasts with each fixed hyperparameter value to validate the hyperparameters for the subsequent y_t values. I always choose the validated forecast to be the forecast generated by the hyperparameter value α that has the lowest mean squared error on the corresponding validation window.

Now, I provide a brief description of each of the methods I compare.

III.2 The methods

Kitchen sink (KS)

The kitchen sink (KS) model is a linear regression of the y_{t+1}-s on all of the lagged predictors, the $x_{i,t}$-s, and a constant:

$$y_{t+1} = \hat{\beta}_0 + \sum_{i=1}^{8} \hat{\beta}_i x_{i,t} \tag{III.6}$$

The parameters $\hat{\beta}_j$, $j = 0, 2, \dots, 8$ are estimated with OLS, that is, to minimising the mean squared error over the estimation sample using an analytic solution[6].

Stage one inverse regularisers

uNCL

The uNCL method consists of estimating a univariate models with a constant with the uNCL algorithm as described in the previous section. The univariate models of uNCL are of the form:

[5] Obviously, this means that some of the numbers in the numerators from the previous equations of the MSE_i and $MSE_{i,normalised}$ change in accordance with the change in the evaluation sample size from 200 to 170. I believe these changes are trivial, thus I do not describe them in more detail.

[6] Computationally, I use the *lm* function from the programming language R to estimate the model.

$$y_{t+1} = \hat{\beta}_{0,i} + \hat{\beta}_i x_{i,t} \tag{III.7}$$

And the parameters $\hat{\beta}_{0,i}$ and $\hat{\beta}_i$ $(i = 1, 2, \ldots, 8)$ are estimated by minimising the NCL loss of each univariate model:

$$NCL_{loss} = \sum_{t=1}^{m} \Big((y_{t+1} - \hat{\beta}_{0,i} - \hat{\beta}_i x_{i,t})^2 - \lambda(\hat{\beta}_{0,i} + \hat{\beta}_i x_{i,t} - f_{FC,t+1})^2 \Big) \tag{III.8}$$

Where $f_{FC,t+1} = \frac{1}{8}\sum_{i=1}^{8} \hat{\beta}_{0,i} + \hat{\beta}_i x_{i,t}$, the average of the individual forecasts and m is the indice of the last period used in the estimation. The NCL_{loss} is minimised *parallel* by gradient descent, as described in more detail in the previous section. The gradient descent stops either after a maximal number of iterations $iter_{max}$ is reached, or if the MSE over the estimation sample does not improve by at least a predefined *threshold* between two consequent iterations. Additionally, the algorithm does not stop before a number of iterations $iter_{min}$ is reached. A fourth parameter of the gradient descent is the learning rate, which I set to 0.5. I set $iter_{max} \coloneqq 500$, $iter_{min} \coloneqq 10$, $threshold \coloneqq 10^{-4}$[7].

uNCL has a hyperparameter λ, which falls between 0 and 1. A higher value of λ corresponds to estimating the individual models by putting more emphasis on the ambiguity term, while a lower value corresponds to putting more emphasis on the squared error term, thus estimating individual models that are accurate by themselves. I estimate uNCL with $\lambda = 0, 0.1, 0.2, \ldots, 1$. For λ, the NCL_{loss} is equal to the MSE_{loss}, that is, there is no 'weight' on the ambiguity term and uNCL is equal to uOLS. As such, I do not use the NCL algorithm for $\lambda = 0$, but instead use the R's 'lm' function, which is based on the analytic solution to minimising the mean squared error.

Complete Subset Regression (CSR)

The complete subset regression of Elliott, Gargano, and Timmermann (2013) consists of estimating individual models by regressing the y_{t+1}-s all possible subsets of the original 8 predictors with k elements (k is fixed and $k \leq 8$) and a constant, and then taking the simple average of the forecasts of the individual models. I estimate the individual models with R's 'lm' function and consider all values of $k = 1, 2, \ldots, 8$. Note that CSR with $k = 1$ is equivalent to uNCL with $\lambda = 0$ and uOLS, and CSR with $k = 8$ is equivalent ot the KS model.

[7] Preliminary simulations indicate that the algorithm converges with this choice of values for the parameters

Stage two inverse regularisers

ELASSO and ERidge

The ELASSO and ERidge forecasts are weighted combinations of the univariate forecasts from the univariate regressions of regressing the y_{t+1} on a single $x_{i,t}$ and a constant, plus an constant offsetting term. As such, the ELASSO is a generalisation of the uOLS (or, equivalently, of uNCL with $\lambda = 0$) by allowing for unequal combination weights.

The combination weights and offsetting terms are chosen by minimising the penalised squared error:

$$MSE_{penalised} = \sum_{t=1}^{m} \left(y_{t+1} - \hat{\alpha} - \sum_{i=1}^{8} \beta_i f_{i,t} \right)^2 + \lambda \sum_{i=1}^{8} \left| \beta_i - \frac{1}{8} \right|^s \tag{III.9}$$

Where $f_{i,t}$ is the forecast from the individual model i at time t, $\hat{\alpha}$ is the offsetting parameter, β_i is the combination weight corresponding to individual model i and λ is a hyperparameter. If $s = 1$, we get the ELASSO, and if $s = 2$, we get ERidge. Both models are reformulated such that they can be estimated by the normal LASSO and ridge regressions as shown in the previous section[8].

The grid for λ is chosen for each estimation period the following way. First, a maximal value λ_{max} of λ is calculated. Then, the lambda grid is chosen such that the smallest lambda is zero, the highest lambda is λ_{max} and the fourth power of the λ values form an equidistant partition of the interval $[0, \lambda_{max}^4]$ with n_λ number of elements. I use $n_\lambda = 40$.

For ELASSO, λ_{max} is the lowest value of λ for which the combination weights are all equal (or, for the corresponding LASSO model, this is the lowest value of λ for which all of the coefficients are zero). For ERidge, λ_{max} is equal to the lowest value of λ for which the combination weights from the corresponding E-elastic net regression with $\alpha = 10^{-3}$ are all equal (or, equivalently, λ_{max} is equal to the lowest value of λ for which all of the coefficients in the corresponding elastic net regression with $\alpha = 10^{-3}$ are equal to zero)[9].

This way of choosing a grid for λ has several advantages to simply supplying a λ sequence to the estimation. First, there isn't really a definite method of choosing a lambda sequence. As such, it is usually hard to determine whether a λ sequence 'makes sense'. The method I use has the advantage that it gives back the two extreme cases, the equal combination weights case (if $\lambda = \lambda_{max}$) and the weights from the Bates-Granger regression (if $\lambda = 0$. Note that λ_{max} and consequently the λ sequence is recalculated when the window is expanded, so the λ sequence is generally not the same for the windows[10].

[8] Computationally, these reduced LASSO and ridge regressions are estimated with R's 'glmnet' function from the package bearing the same name

[9] This is motivated by the fact that for ERidge (or, equivalently, E-elastic net with $\alpha = 0$) the combination weights are never exactly equal to zero, so max lambda is chosen from an E-elastic net regression which is almost, but not exactly equal to the ridge regression.

[10] This way of choosing the λ sequence is also a slight modification of the method suggested in Frey-

Also note that the univariate forecasts $f_{i,t}$ are needed to estimate the ELASSO and ERidge. It is important that these $f_{i,t}$ values are 'out-of-sample', that is, they do not use any information not available at time $t-1$ to get $f_{i,t}$, the estimate of y_t. This means that the univariate forecasts that I estimate ERidge or ELASSO on have to come from a series of univariate regressions with rolling or expanding windows. To keep in line with the other methods, I use and expanding window, with the initial estimation period equal to the first 70 data points. The first value that I forecast with the ELASSO and ERidge is y_{102}, so I use the 'out-of-sample' univariate forecasts of $f_{i,t}$, $t = 72, 73, \ldots, 101$ initially. Later, I also expand the window of univariate forecasts that I estimate the ELASSO and ERidge on[11].

Regularisers

LASSO and Ridge

The LASSO and ridge regressions estimate a linear regression of the same form as the kitchen sink:

$$y_{t+1} = \hat{\beta}_0 + \sum_{i=1}^{8} \hat{\beta}_i x_{i,t} \tag{III.10}$$

But, the parameters $\hat{\beta}_i$, $i = 0, 2, \ldots, 8$ are estimated by minimising the penalised mean squared error:

$$MSE_{penalised} = \sum_{t=1}^{m} \left(y_t - \hat{\alpha} - \hat{\beta}_0 - \sum_{i=1}^{8} \hat{\beta}_i \right)^2 + \lambda \sum_{i=1}^{8} \left| \hat{\beta}_i \right|^s \tag{III.11}$$

Where $s = 1$ is the LASSO and $s = 2$ is the ridge regression. The hyperparameter λ determines the degree of shrinkage; a higher value of λ corresponds to a stronger shrinkage. The extreme case of $\lambda \to \infty$ means a model with only a constant and $\lambda = 0$ is equivalent to estimating the model with OLS, so the result is the kitchen sink model.

I fit both the LASSO and ridge with R's glmnet function. The supplied λ sequence is chosen in a similar fashion to the ELASSO and ERidge methods. The only difference is that the λ values with indices $1, 2, \ldots, (n_\lambda - 1)$ are chosen so that their natural logarithms are equidistant on $[ln(10^{-4}, ln(\lambda_{max})]$[12]. I use $n_\lambda = 49$ and add $\lambda = 0$ at the end of the sequence with index 50 to make sure that I get the KS model as a special case of the LASSO and ridge. I note again that the λ sequence in recalculated each time the window

berger, Neuhierl, and Weber (2020). For more details, see Appendix A.1.1

[11] This means, for example, that I estimate the ELASSO and ERidge on $f_{i,t}$, $t = 72, 73, \ldots, i$ to forecast y_{i+1}, with $i > 102$.

[12] Note that this is the way R's glmnet package generates the λ sequence by default. It is also recommended by Friedman, Hastie, and Tibshirani (2010)

is expanded. As such, I do not show results for exact λ values. Instead, I fix the indice that I use for the λ value in each recalculation, and calculate results for these fixed indices on the λ sequence.

III.2.1 Simulation results for fixed hyperparameter values

Figure III.1 plots the normed MSEs of the uNCL, CSR, ELASSO, ERidge and KS methods. The KS method is included to emphasise that it is a special case of CSR, but not of uNCL. The values on the x axis mean the values of the λ hyperparameter for the uNCL. For CSR, k start at 1 on the left and increases to 8 on the right. The ELASSO and ERidge both have the case of λ_{max} on the left and their λ decreases to the right. This difference of plotting uNCL and CSR with increasing hyperparameter values and plotting the ELASSO and ERidge with decreasing parameter values is meant to emphasise the role the parameters play. For the uNCL and CSR, a higher hyperparameter means lower shrinkage. In contrast, it is the opposite for the ELASSO and ERidge. As described previously, the actual normed MSEs are only estimated for a finite discrete grid of values[13]; the lines are an linear interpolation between normed MSEs that are actually estimated in the sample. Also, note that the actually estimated MSEs are placed such that they have an equal distance to their neighbours; therefore, although the ELASSO and ERidge do not have equidistant hyperparameter values, they are plotted at equal distances.

There are several interesting observations to be made. First, we see that all of the methods can improve on the uOLS benchmark (have a normed MSE lower than 1) if the DGP is not very noisy (R^2-s are not very small) and the predictors are not very highly correlated (ρ is small). This is not very surprising; all of these methods, as will be shortly demonstrated, can be labeled inverse regularisers in the sense that they decrease the level of shrinkage or regularisation inherent in the uOLS. Because a lower level of regularisation is usually sufficient if the data is not very noisy and predictors are not highly correlated, the observed result follows. Also note that US equity premium and its predictors are probably closest to the mediocre predictor correlation $\rho = 0.4$ and mediocre signal-to-noise ratio around 5%(Rapach, 2013). The plot shows that both CSR and the uNCL have a normalised MSE below 1 in this case, so I expect them to outperform the uOLS in the application in the next section.

Second, an interesting observation is that the ELASSO and especially the ERidge hardly ever outperform the uOLS (assuming $\lambda \neq \lambda_{max}$ for these models), while both the uNCL and CSR often do. This finding shows that changes to the 'estimation phase' of the uOLS (stage I inverse regularisers) tend to perform better than changes to the 'aggregation phase' (stage II inverse regularisers) do, at least when the hyperparameters are optimally

[13] The ELASSO and ERidge also do not have a fixed hyperparameter grid, but it somewhat changes each time the window is expanded.

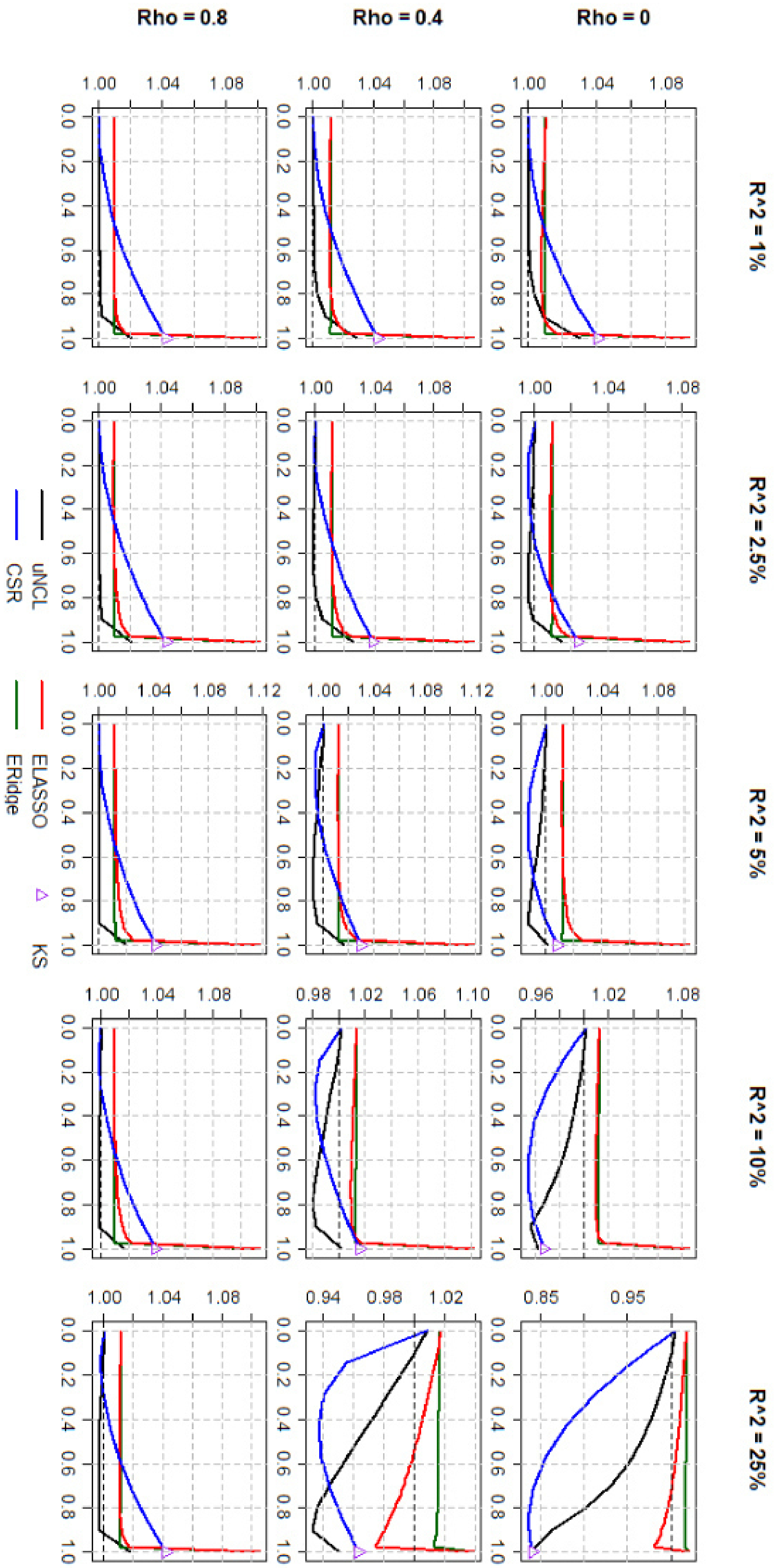

Figure III.1: The normed MSE of the uNCL, CSR, ELASSO, ERidge and KS models with fixed hyperparameters. The y axis shows the normed MSE. The x axis shows the λ value of the uNCL. For CSR, k increases to the right. For the ELASSO and ERidge, λ decreases to the right. Note that the actual normed MSEs are calculated only at a set of discrete values; I linearly interpolate these values to get the lines presented on the plot. The dashed black line at 1 emphasises the comparison to the performance of the uOLS.

chosen. Furthermore, the fact that the stage II inverse regularisers hardly ever outperform the uOLS, and even when they do, they are dominated by the uNCL and CSR suggest that stage II inverse regularisation is not a good approach to optimise the shrinkage of the uOLS. I note that this finding bears a close relationship to the previously mentioned 'forecast combination puzzle', a widespread and many times corroborated finding the the forecasting literature that an equal weighted combination of forecasts usually performs better than more advanced techniques that estimate the combination weights from the data.

Figure III.2 presents the normed MSEs of the uNCL, CSR, LASSO and ridge regression methods. The uNCL and CSR is included to make a comparison with figure III.1 easier. The x axis once again shows the λ values for uNCL, and CSR's k increases to the right. The LASSO and ridge have $\lambda = \lambda_{max}$ on the left, which is equivalent to the constant model, and $\lambda = 0$ on the right, which is equivalent to the kitchen sink model.

Once again, both the LASSO and ridge can mostly outperform the uOLS only when the data generating process is not very noisy and the predictors are not very highly correlated. This is due to the fact that the LASSO and ridge can have different degrees of shrinkage with different values of their hyperparameters. If the DGP is not very noise and predictors are not very highly correlated, a lower level of shrinkage is optimal than the shrinkage of the uOLS.

Another interesting observation is that the uNCL and CSR methods dominate the LASSO and ridge if the hyperparameters of all models are chosen optimally. This finding is the one of the main contributions of the paper; with noisy datasets, it is better to inverse regularise the uOLS than to use traditional regularisers that 'start' from the kitchen sink, like the LASSO or ridge does.

Notably, both the uNCL and CSR have very similar MSE values if their hyperparameters are chosen optimally. I also want to highlight that while the degree of outperformance - in some cases roughly only 0.5% - seems small at first sight, even this performance improvement can be substantial in practical applications. For example, Campbell and Thompson (2007) note that an R^2 gain of around 0.5% is already significant in an economic sense when talking about equity premium predictability.

III.2.2 Hyperparameter optimisation

In the last subsection, I compared the performance of the models with fixed hyperparameters. However, in practical applications, the optimal value of the hyperparameter is unknown and has to be estimated from the data, usually be previous performance or some information criterion. This introduces and additional source of error, namely the increased forecasting error that results from selecting a suboptimal hyperparameter value. The magnitude of this additional error my vary between the different models, thus

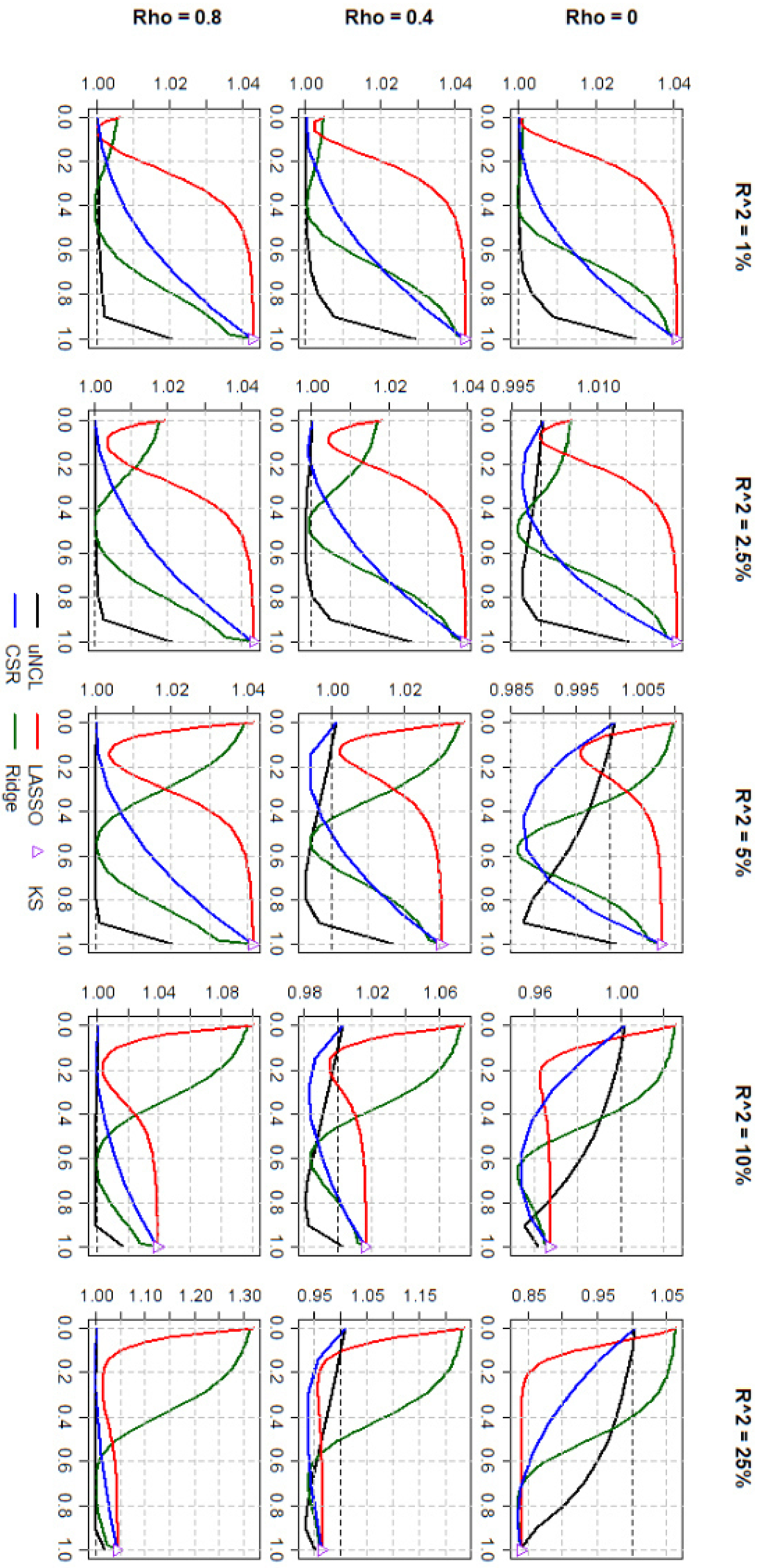

Figure III.2: The normed MSE of the uNCL, CSR, LASSO, ridge and KS models with fixed hyperparameters. The y axis shows the normed MSE. The x axis shows the λ value of the uNCL. For CSR, k increases to the right. For the LASSO and ridge, λ decreases to the right. Note that the actual normed MSEs are calculated only at a set of discrete values; I linearly interpolate these values to get the lines presented on the plot. The dashed black line at 1 emphasises the comparison to the performance of the uOLS.

Normalised MSEs with Validation						
DGP params	Models	$R^2 = 1\%$	$R^2 = 2.5\%$	$R^2 = 5\%$	$R^2 = 10\%$	$R^2 = 25\%$
$\rho = 0$	uNCL	**99.96**	100.18	**99.16**	**95.67**	**84.34**
	CSR	100.41	**99.58**	99.97	96.69	84.63
	ELASSO	100.76	101.58	101.73	101.02	98.33
	ERidge	100.70	101.54	101.46	101.20	100.50
	LASSO	100.81	100.13	101.17	98.13	85.06
	Ridge	100.45	99.63	99.93	96.53	84.45
$\rho = 0.4$	uNCL	100.22	100.44	**99.75**	**99.19**	93.53
	CSR	**100.18**	**100.41**	99.90	99.82	**93.33**
	ELASSO	101.02	101.46	101.36	101.32	97.13
	ERidge	100.88	101.37	101.02	101.52	101.60
	LASSO	100.61	101.19	101.03	101.34	94.91
	Ridge	100.35	100.61	100.06	99.94	93.41
$\rho = 0.8$	uNCL	100.31	100.85	100.17	**100.89**	**99.76**
	CSR	**100.00**	**100.07**	**99.81**	101.28	99.80
	ELASSO	101.06	101.75	100.88	102.24	100.89
	ERidge	100.92	101.52	100.64	102.08	100.82
	LASSO	100.28	100.57	100.62	102.56	101.18
	Ridge	100.08	100.25	100.15	101.79	100.09

Table III.3: Normalised MSEs with Validation. The table present the normalised MSE of each model with validated hyperparameters for each (R^2, ρ) pair. The R^2 of the DGP is on the top, while the predictor correlation ρ is on the right. A normalised MSE value above (below) 100 indicates that the method beats (is beaten by) uOLS when the hyperparameters are validated.

a robustness analysis of the previous results is necessary.

Table III.3 summarizes the $100 * MSE_{i,normalised}$[14] values for each method i with validated hyperparameters results for each (ρ, R^2) pair and each model. The red bold values indicate the best performing model on the given (ρ, R^2) pair.

The table shows that the previous findings with the optimal hyperparameters stay the same with hyperparameter values optimised from the data. First, the models generally perform better with higher R^2-s and lower ρ-s. Second, the ERidge and ELASSO hardly ever outperform the uOLS, meaning that stage II inverse regularisation is not very competitive. Third, for each (ρ, R^2) pair, uNCL and CSR show similar performance and outperform the LASSO and ridge. This indicates that inverse regularising the uOLS with stage I inverse regularisers is better than regularising the kitchen sink model directly with the LASSO or ridge, even if the optimal values of the hyperparameters are not known a priori and have to be estimated from past performance.

Table III.5 shows the differences between the normalised MSE of each method with its hyperparameter value that is optimal when fixed on the whole evaluation sample and the normalised MSE that I have presented in table III.3 for each (R^2, ρ) parameter pairs.

[14] The values are multiplied by 100 so that the results are easy to interpret as a percentage of the MSE of the uOLS.

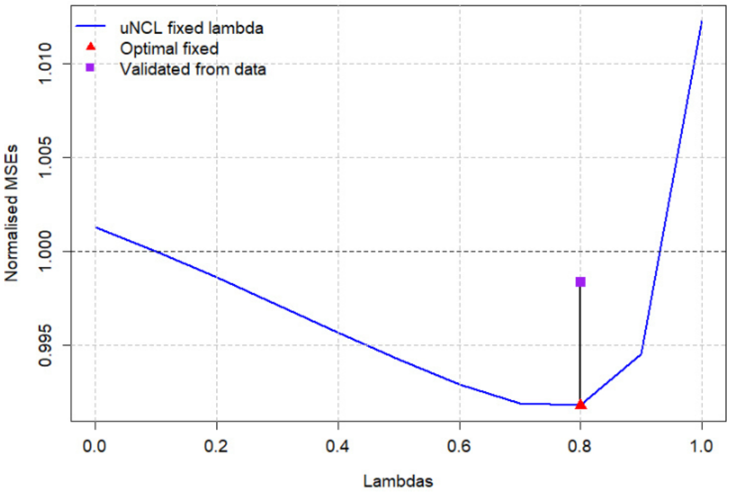

Figure III.4: Graphical illustration of the difference between the 'optimal' and validated normalised MSEs. The illustration plots the normalised MSE curve of the uNCL for a fixed DGP (R^2, ρ) (blue line) as a function of the hyperparameter λ. The red triangle denotes the optimal value of the λ hyperparameter when fixed, along with the corresponding normalised MSE value. The pruple square denotes the normalised MSE of the validated model, and is placed at the same coordinate on the x axis to provide an easy comparison with the normalised MSE with optimal λ. The length of the black line is the measure shown in III.5

Figure III.4 gives a graphical illustration for this metric for the $R^2 = 5\%$ and $\rho = 0.4$ DGP. The blue line is the (linear interpolation of) the uNCL normalised MSEs with fixed λ values over the evaluation sample. The red triangle shows the 'optimal' lambda and the corresponding normalised MSE, when the hyperparameter is once again fixed on the whole evaluation sample. The purple square illustrates the normalised MSE with hyperparameter validation each time the window is expanded. It is placed above the 'optimal' normalised MSE for convenience. The length of the black line between the validated and 'optimal' normalised MSEs is the metric I present in table III.5.

Note that the 'optimal' value, as defined above, is only optimal in the sense that it has better performance when fixed on the whole evaluation sample than any other the performance of any other hyperparameter value *when set on the whole estimation sample*. The actually optimal value of the hyperparameter is not a constant on the whole sample, but should be lower near the end of the evaluation period, because the models are fitted on more data points by then. The fact that the 'optimal' model is optimal only in this narrow sense means that negative values (which indicate that the validated model performs better than the 'optimal') can appear (and do appear) in the table. This happens in the few cases when validation the hyperparameter each time the window is expanded can successfully capture the decreasing need for shrinkage at the later period of the evaluation sample.

Comparing the two best performing models, the uNCL and the CSR, we see that the performance of uNCL (CSR) usually deteriorates to a smaller degree than the performance of CSR (uNCL), if the predictor correlation ρ is low/high. In line with these results, table

Difference of 'optimal' and validated normalised MSEs						
DGP params	Models	$R^2 = 1\%$	$R^2 = 2.5\%$	$R^2 = 5\%$	$R^2 = 10\%$	$R^2 = 25\%$
$\rho = 0$	uNCL	-0.04	0.61	0.70	0.31	**1.00**
	CSR	0.40	**0.01**	1.50	1.67	1.46
	ELASSO	-0.08	0.73	0.95	0.12	1.57
	ERidge	**-0.19**	0.67	**0.65**	**0.08**	1.63
	LASSO	0.67	0.05	1.81	2.23	1.52
	Ridge	0.44	0.04	1.60	1.58	1.31
$\rho = 0.4$	uNCL	0.28	0.53	0.46	0.92	0.36
	CSR	**0.20**	0.45	0.46	1.37	**-0.17**
	ELASSO	0.38	0.53	0.56	0.89	0.62
	ERidge	0.20	**0.43**	**0.20**	**0.61**	0.50
	LASSO	0.33	0.80	0.74	1.75	0.05
	Ridge	0.32	0.60	0.58	1.39	-0.15
$\rho = 0.8$	uNCL	0.31	0.86	0.19	**1.07**	0.38
	CSR	**0.00**	**0.06**	**-0.20**	1.40	0.27
	ELASSO	0.37	0.79	0.11	1.19	0.17
	ERidge	0.21	0.55	-0.15	1.02	**-0.01**
	LASSO	0.16	0.35	0.18	2.03	0.53
	Ridge	0.12	0.23	0.12	1.85	0.46

Table III.5: Differences between the normalised MSE of each method with its hyperparameter value that is optimal when fixed on the whole evaluation sample and the normalised MSE when the hyperparameter is validated each time the window is expanded. A smaller value indicates that hyperparameter optimisation is less detrimental to the performance of the model.

III.3 indicated that uNCL (CSR) has a lower normalised MSE than CSR (uNCL), if the predictor correlation ρ is low (high).

Additionally, I note that the performance for the LASSO and ridge regressions usually deteriorates to a higher degree than that of the uNCL and CSR for the lower predictor correlation and/or higher R^2 parameter pairs, that is, when weaker shrinkage is beneficial and the uOLS can be improved upon. As such, the results indicate that uNCL and CSR perform better at those DGPs because of a lower risk of hyperparameter estimation error. This decreased sensitivity to validation error is a significant advantage in practical applications.

It is worth mentioning that the ELASSO and more prominently ERidge have a small reduction in their performance due to hyperparameter optimisation in many cases. This is most likely due to the fact that these methods have roughly the same performance for a wide range of the hyperparameter sequence, which could be seen prior in figure III.1.

III.3 Error decompositions

In the theoretic review as well as previously in the evaluation of the simulation results, I noted that the compared methods all optimise the bias-variance trade-off by setting the value of a hyperparameter. This is a well-known property of the LASSO (Tibshirani, 1996), Ridge (Hoerl and Kennard, 1970), CSR methods (Elliott, Gargano, and Timmermann, 2013), but has not been demonstrated in the case of my novel method, the uNCL. The main goal of this subsection is to show that the uNCL indeed optimises the bias-variance trade-off, and that it does that quite effectively. This is shown by the estimation and comparison of the bias-variance decomposition of the squared error of each method for each hyperparameter value and DGP. Additionally, the subsection also estimates the bias-variance-covariance decomposition of the uNCL and CSR methods. The decomposition of the squared error of the uNCL is shows that the NCL algorithm does not lead to an effective reduction of the covariance component in this case, which is in contrast to previous applications of the NCL algorithm in machine learning in the training of ensembles of neural networks (Brown, Wyatt, and Tino, 2005).

Additionally, most empirical applications that compare the different methods only show the performance of the considered methods, but do not aim to give a comprehensive explanation of the underlying reasons for the results, or why certain methods outperform the others. In this section, I use the estimated bias-variance decomposition of the methods to measure the efficiency of each method in reducing the bias. This 'efficiency' is measured as the corresponding increase in variance; a low increase in variance indicates that the method is efficient in reducing the bias. I show that the superior performance of the uNCL stems from its ability to decrease the bias of the uNCL at a lower cost than the other methods.

III.3.1 Bias-variance trade-off estimation

To estimate the bias-variance decomposition of each method, I carry out an additional simulation. The data is simulated with a slight modification. Previously, the 100 simulated time series all were sampled from the underlying distribution. However, this is not adequate to the estimation of the variance component of the squared error. To see why, consider the variance component:

$$Var_{i,\boldsymbol{x_t}} = E\left[(f_i(\boldsymbol{x_t}) - E[f_i(\boldsymbol{x_t})^2]\right] \tag{III.12}$$

Where $f(\boldsymbol{x_t}$ is the fitted value from method i at the predictor vector $\boldsymbol{x_t}$. This could be estimated from the data with the following formula:

$$\hat{Var}_{i,\boldsymbol{x_t}} = \frac{1}{n}\sum_{k=1}^{n}\left(f_i(\boldsymbol{x_t}) - \frac{\sum_{j=1}^{n} f_i(\boldsymbol{x_t})}{n}\right)^2 \tag{III.13}$$

Where $\frac{\sum_{j=1}^{n} f_i(\boldsymbol{x_t})}{n}$ is the average of the fitted values at point $\boldsymbol{x_t}$, and n is the number of points used in the estimation.

The problem with the previous method of generating the data is that the average of the fitted values at point $\boldsymbol{x_t}$ cannot be estimated, because there is only one fitted value for each $\boldsymbol{x_t}$. Thus, I generate the DGP in a different way. I first generate 4 $\boldsymbol{x_t}$, $t = 1, 2, \ldots, 300$ sequences for the predictors, instead of generating 100 different sequence as previously. Then, I generate 100 different sequences of the error terms $\boldsymbol{\epsilon_t}$, $t = 1, 2, \ldots, 300$ and generate the $\boldsymbol{y_{t+1}}$, $t = 1, 2, \ldots, 300$ sequences by adding 25 of these 100 error sequences to each of the 4 $\boldsymbol{x_t}$ sequences[15]. This ensures that I have 100 time series with different $\boldsymbol{y_{t+1}}$ sequences, but they have the same expected value for each 25-element batches. Using the 25 time series from a given batch, I can estimate the variance from the data by the previous formula:

$$\hat{Var}_{i,\boldsymbol{x_t}} = \frac{1}{25}\sum_{l=1}^{25}\left(f_{i,l}(\boldsymbol{x_t}) - \frac{\sum_{j=1}^{25} f_{i,j}(\boldsymbol{x_t})}{25}\right)^2 \tag{III.14}$$

Where t is a fixed value, and thus the predictors $\boldsymbol{x_t}$ are the same. The fact that I resampled the residual errors ϵ_t in the DGP means that the fitted values $f_{i,l}(\boldsymbol{x_t})$ will be different, but their expectation can be estimated by the within-batch-average $\frac{\sum_{j=1}^{n} f_{i,j}(\boldsymbol{x_t})}{25}$.

Taking an average of $\hat{Var}_{i,\boldsymbol{x_t}}$ over the the time gives an estimate of the within-batch-average variance:

$$\hat{Var}_{i,j} = \frac{101}{300}\sum_{t=101}^{300} \hat{Var}_{i,\boldsymbol{x_t}} \tag{III.15}$$

Where $j = 1, 2, 3, 4$ indexes the batch. Taking an average over the 4 batches gives the final estimate of the variance component of the model.

The squared bias is estimated on the same 'batched' data. The estimate of the squared bias at point $\boldsymbol{x_t}$ from a given batch is:

$$\hat{Bias}^2_{i,\boldsymbol{x_t}} = \frac{1}{25}\sum_{l=1}^{25}\left[f_{i,l}(\boldsymbol{x_t}) - E[y_{t+1}]\right]^2 \tag{III.16}$$

$$= \frac{1}{25}\sum_{l=1}^{25}\left[f_{i,l}(\boldsymbol{x_t}) - \sum_{k=1}^{8}\beta_k x_{k,t}\right]^2 \tag{III.17}$$

Where I use the definition of $y_{t+1} = \sum_{k=1}^{8}\beta_k x_{k,t} + e_t$ and $E[e] = 0$ to get the result.

This squared bias term is averaged over the $t = 101, 102, \ldots, 300$ and the batches $j = 1, 2, 3, 4$ to get the estimate of the bias of method i:

$$\hat{Bias}_i = \frac{1}{4 * 25 * 200}\sum_{j=1}^{4}\sum_{l=1}^{25}\sum_{t=101}^{300}\left[f_{i,l,j}(\boldsymbol{x_t}) - \sum_{k=1}^{8}\beta_k x_{k,t}\right]^2 \tag{III.18}$$

[15] Naturally, I use each $\boldsymbol{\epsilon_t}$ sequence only once.

Where $f_{i,l,j}$ is the forecast from model i on element l of batch j.

I estimate both the squared bias and the variance component for each of the methods and for each value of the hyperparameters and DGP parameters (R^2, ρ).

III.3.2 Bias-variance trade-off results

Figure III.6 plots the estimated bias-variance decomposition of the uNCL for each fixed value of the hyperparameter (x axis), and each DGP parameter pair (R^2, ρ).[16] The plot proves that the uNCL optimises the bias-variance trade-off for most parameter pairs (R^2, ρ) by setting the value of the λ hyperparameter. A lower λ value, which is closer (or, with $\lambda = 0$, equivalent) to the uOLS usually has a higher squared bias but small variance, whereas uNCL with a high value of has a small squared bias with increased variance.

The plots also show that the uNCL can reduce the squared bias to essentially zero for most of the DGP parameters, albeit at a relatively low cost in variance. Table III.7 show the estimated variance of the uNCL with $\lambda = 1$ divided by the estimated variance of the kitchen sink model, and multiplied by 100 for each DGP parameter pair. Intuitively, the table shows the variance of the uNCL as a percentage of the variance of the kitchen sink model. The values prove the efficiency of uNCL as a bias-decreasing method; the uNCL with $\lambda = 1$ has a variance considerably lower than the kitchen sink, even though it has essentially the same bias (zero).

III.3.3 The cost of bias reduction

In this subsection, I demonstrate that the uNCL is more efficient than the competing methods in reducing the squared bias component. I show this by calculating the 'cost' of reducing the bias as the corresponding increase in variance. This is motivated by the fact that a method may be viewed as 'efficient' in reducing the bias if the bias reduction causes a relatively small increase in the variance. Such a method can achieve good performance by optimising the bias-variance trade-off efficiently.

I define the cost of bias reduction (CoBR) between the hyperparameter values α_1 and α_2 as:

$$CoBR_{\alpha_1,\alpha_2} = -\frac{Var_{\alpha_1} - Var_{\alpha_2}}{Bias^2_{\alpha_1} - Bias^2_{\alpha_2}} \tag{III.19}$$

Where $^2_{\alpha_i}$ is the squared bias of the given method at the hyperparameter value α_i, and $Var_{_i}$ is the variance of the given method at the hyperparameter value α_i[17]. Intuitively, the numerator is the change in variance, while the denominator is the change in the squared

[16] The bias-variance decompositions of the other models are presented in Appendix **??**

[17] The CoBR is not constant for different values of the hyperparameter, that is why the measure is defined such that it depends on the hyperparameters.

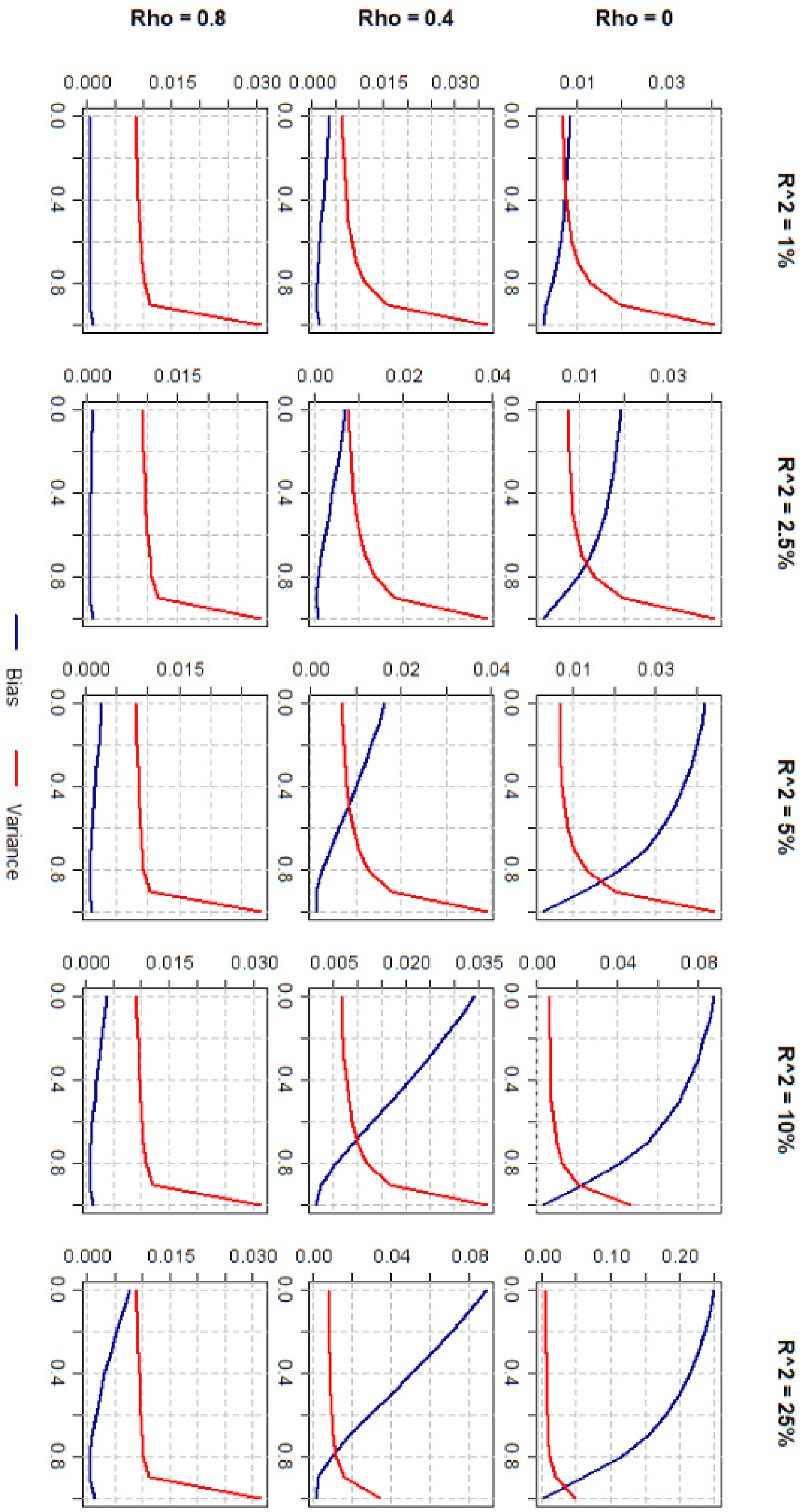

Figure III.6: Bias-variance error decomposition of the uNCL. The x axis has the λ hyperparameter values in increasing order.

	$R^2 = 1\%$	$R^2 = 2.5\%$	$R^2 = 5\%$	$R^2 = 10\%$	$R^2 = 25\%$
$\rho = 0$	80.62	82.12	85.24	87.76	91.70
$\rho = 0.4$	74.07	74.97	74.70	75.39	74.47
$\rho = 0.8$	57.89	58.68	57.27	58.74	57.08

Table III.7: Ratio of uNCL and kitchen sink variances. The ratio is multiplied by 100 so that it can be interpreted as a percentage.

bias as we move the hyperparameter from α_1 to α_2. The expression is multiplied by 1 so that the result remains positive for the easy of interpretation - because the squared bias and variance usually change in the opposite directions, the numerator and denominator usually have different signs. The CoBR can be intuitively understood as the necessary increase in variance that comes with a 'unit' decrease in bias. Because the expected squared error is the sum of the squared bias and the variance (plus an irreducible noise term), the move from α_1 to α_2 is beneficial in terms of MSE if the CoBR is below one. In this case, the bias reduces to a higher degree than the variance as the hyperparameter is moved.

I measure the CoBR for the uNCL, CSR, LASSO, ridge and ELASSO for all 'adjacent' hyperparameter values. That is, I calculate $CoBR_{0,0.1}, CoBR_{0.1,0.2}, \ldots, CoBR_{0.9,1}$ for the uNCL, $CoBR_{1,2}, CoBR2, 3, \ldots, CoBR_{7,8}$ for the CSR and $CoBR_{\lambda_{max},\lambda_{max-1}}, \ldots, CoBR_{\lambda_2,0}$ for the LASSO, ridge and ELASSO. I use the previously estimated squared bias and variance components to calculate the CoBR values.

Figure III.8 plots the CoBR values of the uNCL, CSR, LASSO, ridge and ELASSO over the squared bias terms. The CoBR values are plotted over the squared bias corresponding to the 'lower' of the CoBR hyperparameter values in the case of the uNCL and CSR. For example, the $CoBR_{0.1,0.2}$ of the uNCL is plotted over the squared bias of the uNCL with the lower hyperparameter value, so over the squared bias of uNCL with $\lambda = 0.1$. For the LASSO, ridge and ELASSO, the CoBR is plotted over the squared bias corresponding to the hyperparameter with the lower index (that is, the hyperparameter that is closer to λ_{max}). The x axis has the squared biases limited on the left and right by the minimal and maximal squared biases of the uNCL on the DGP (R^2, ρ). The y axis has values from 0 to only 2.5 to ensure that the 'neighbourhood' of 1 is visible with more detail[18]. As noted previously, the CoBR value of 1 is important because a CoBR below 1 means that the expected squared error would decrease by moving the hyperparameter from λ_1 to λ_2. Any CoBR outside the neighbourhood of 1 is thus not very interesting, because the corresponding hyperparameter value is usually very far from the optimal.

In general, the plots show that the CoBR values of the uNCL are smaller than the

[18] In some cases, the limits on the y axis mean that much of the CoBR values are not shown on the plot. This is most visible for (R^2, ρ), where no value is plotted. This is not really a problem; in this case, setting the hyperparameters to the value with the highest bias is obviously the best choice, so a comparison of CoBR values is not very interesting.

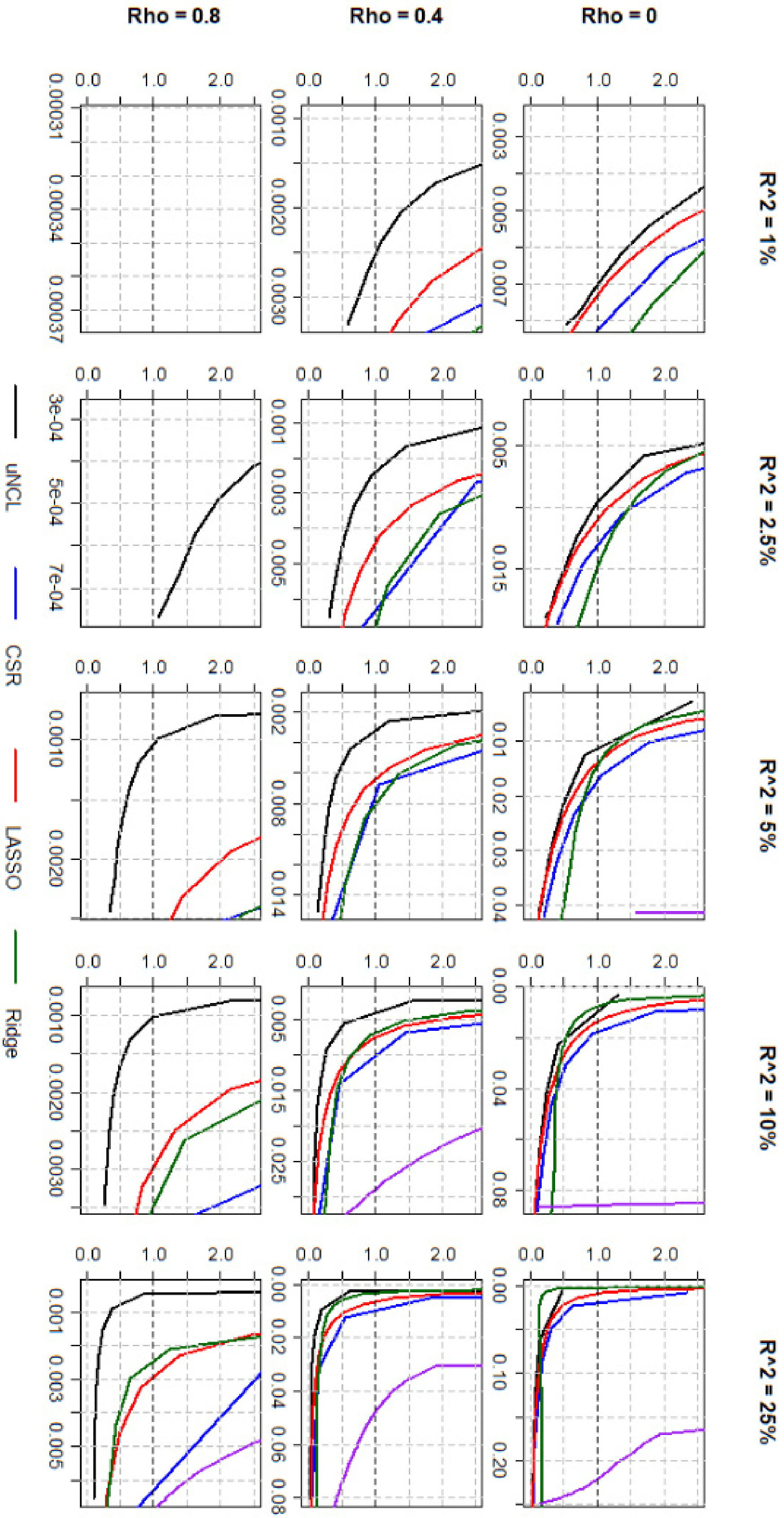

Figure III.8: Cost of Bias Reduction (CoBR) plots. The x axis is the squared bias, and has the minimum and maximum squared bias value of the uNCL as limits on the plot. The y axis plots the CoBR values. On the plot, each CoBR value is plotted over the squared bias of the lower hyperparameter value. For example, $CoBR_{0.1,0.2}$ of the uNCL is plotted over the squared bias of uNCL with $\lambda = 0.1$. For the LASSO, ridge and ELASSO, 'lower' hyperparameter value means the hyperparameter value with the lower index. The y axis has the limits $(0, 2.5)$, so that the neighbourhood of 1 can be seen well on the plots.

CoBR values of most of the other methods for a fixed level of squared bias. Notably, the CoBR of the uNCL is usually lower than those of the traditional regularisation methods, indicating better performance. Graphically, this is seen by the fact that the black curve tends to take up lower values than the other curves when the squared bias on the x axis is kept fixed. The fact that uNCL has a relatively low CoBR means that it can efficiently decrease the squared bias while keeping the corresponding increase in variance at minimal. The plots also show that the black curve tends to cross the horizontal dashed line at 1 at lower levels of the squared bias. Because a $CoBR > 1$ indicates that raising the hyperparameter value would introduce more variance than the amount it would reduce the squared bias by, the fact that uNCL has a $CoBR = 1$ at low levels of the squared bias shows that uNCL has a lower squared bias than the other methods in optimum.

III.3.4 The bias-variance-covariance decomposition

In this subsection, I present the bias-variance-covariance decomposition of the uNCL and CSR models[19].

I estimate the decomposition on the same data that I used to estimate the bias-variance decomposition. The estimate of the squared bias remains the same. I estimate the variance of the individual model l from the j-th batch at point $\boldsymbol{x_t}$[20] with the following expression:

$$\hat{Var}_{i,\boldsymbol{x_t}} = \frac{1}{25}\sum_{k=1}^{25}\left(f_l(\boldsymbol{x_t}) - \frac{\sum_{j=1}^{25} f_l(\boldsymbol{x_t})}{25}\right)^2 \tag{III.20}$$

Note that this expression looks the same as equation III.14, but there is important difference. In equation III.14, f_i is the forecast from the i-th 'final' model; here, f_l is the forecast from the individual model with index l. The individual models are later aggregated by equal weighting to get the uNCL or CSR 'final' forecast.

The variance estimate from batch j from the previous equation III.20 is averaged over the 4 batches, and then the evaluation sample $t = 101, 102, \ldots, 300$ to get the estimate of the variance of the individual forecast l.

The covariance of the individual models is estimated in a similar manner. Namely, the covariance of individual model l and k on the j-th batch is estimated by:

[19] The LASSO and the ridge do not have a bias-variance-covariance decomposition, because they are not an aggregation of individual models. The ELASSO and ERidge do have an bias-variance-covariance decomposition, but it is difficult to compare with the decompositions of the uNCL and CSR, because ELASSO and ERidge have non-equal weights. Thus, only the bias-variance-covariance decomposition of the uNCL and CSR is estimated and shown here.

[20] Although I use vector notation here, the input of the individual model may be of length 1, for example, in the case of the uNCL.

$$\begin{aligned}\hat{Covar}_{l,k,\boldsymbol{x_{l,t}},\boldsymbol{x_{k,t}}} =& \frac{1}{25}\sum_{n=1}^{25}\left(f_{l,n}(\boldsymbol{x_{l,t}}) - \frac{\sum_{m=1}^{25} f_{l,m}(\boldsymbol{x_{l,t}})}{25}\right) \\ & * \left(f_{k,n}(\boldsymbol{x_{k,t}}) - \frac{\sum_{m=1}^{25} f_{k,m}(\boldsymbol{x_{k,t}})}{25}\right) \end{aligned} \tag{III.21}$$

Where $f_{l,n}$ and $f_{k,n}$ are the forecasts from the individual models l and k for the n-th sequence in the batch, respectively, and $\boldsymbol{x_{l,t}}$ and $\boldsymbol{x_{k,t}}$ are the predictors used in these models. This expression is averaged over the batches $j = 1, 2, 3, 4$, the evaluation sample $t = 101, 102, \ldots, 300$ and the individual model pairs (l, k) to get the estimate of the average covariances of the individual models.

Figure III.9 plots the bias-variance-covariance decomposition of the uNCL for each fixed $\lambda = 0, 0.1, \ldots, 1$ hyperparameter value and (R^2, ρ) DGP parameter pair. The plots show that the covariance component is essentially 'flat' in λ for most of the hyperparameter space. It has a sharp decrease from $\lambda = 0.9$ to $\lambda = 1$ if the predictor correlation parameter ρ is not equal to zero, but this sharp decrease is also followed by a sharp increase in the average variances of the individual models. As a result, the decrease in the covariance component does not lead to an decrease in MSE; the increase in the variance component is usually higher than the decrease in the covariance component[21].

These results on the relationship of the covariance component and the λ hyperparameter are in disagreement with the explanation that $Brown, Wyatt, AndTino$(2005) gives about the strong performance of ensembles of neural networks trained by the NCL algorithm. The strong performance of uNCL seen in my simulations is not the result of optimising the 'accuracy' of the individual models (the sum of the bias and variance component from the bias-variance-covariance decomposition) against the 'diversity' of the individual models (the covariance component), but a result of decreasing the bias of the uOLS relatively effectively. This, as shown previously, is better understood as an optimisation of the bias-variance trade-off.

I already noted previously in the previous section on the theoretical foundations that the uNCL differs from previous applications of the NCL algorithm to neural networks in several way. These include a) applying the NCL algorithm to rigid linear models instead of flexible non-linear models, b) testing on a linear DGP and c) the difference in how the predictors are used, meaning that the individual models use only one of the predictors in uNCL, but all of the predictors in previous applications. I do not aim to determine which

[21] The fact that the increase in the variance component is usually higher than the decrease in the covariance component can also be read from the bias-variance plots of the uNCL; there, the variance component is the sum of the variance and covariance component from the bias-variance-covariance decomposition. Because the variance in the bias-variance plots usually increase when moving from $\lambda = 0.9$ to $\lambda = 1$, the increase in the variance component in the bias-variance-covariance decomposition is higher than the decrease in the covariance component in most cases.

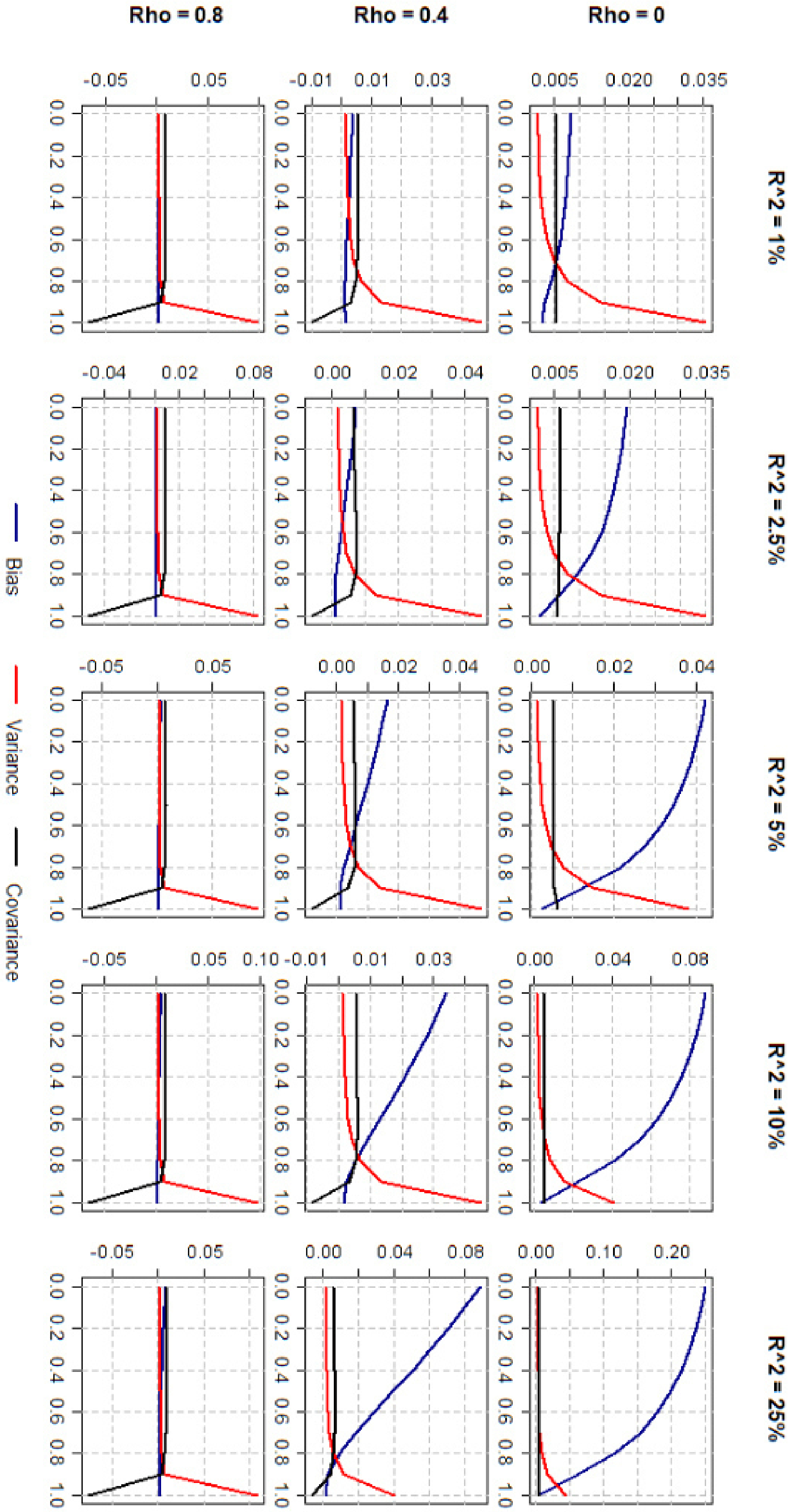

Figure III.9: Bias-variance-covariance error decomposition of the uNCL. The x axis has the λ hyper-parameter values in increasing order.

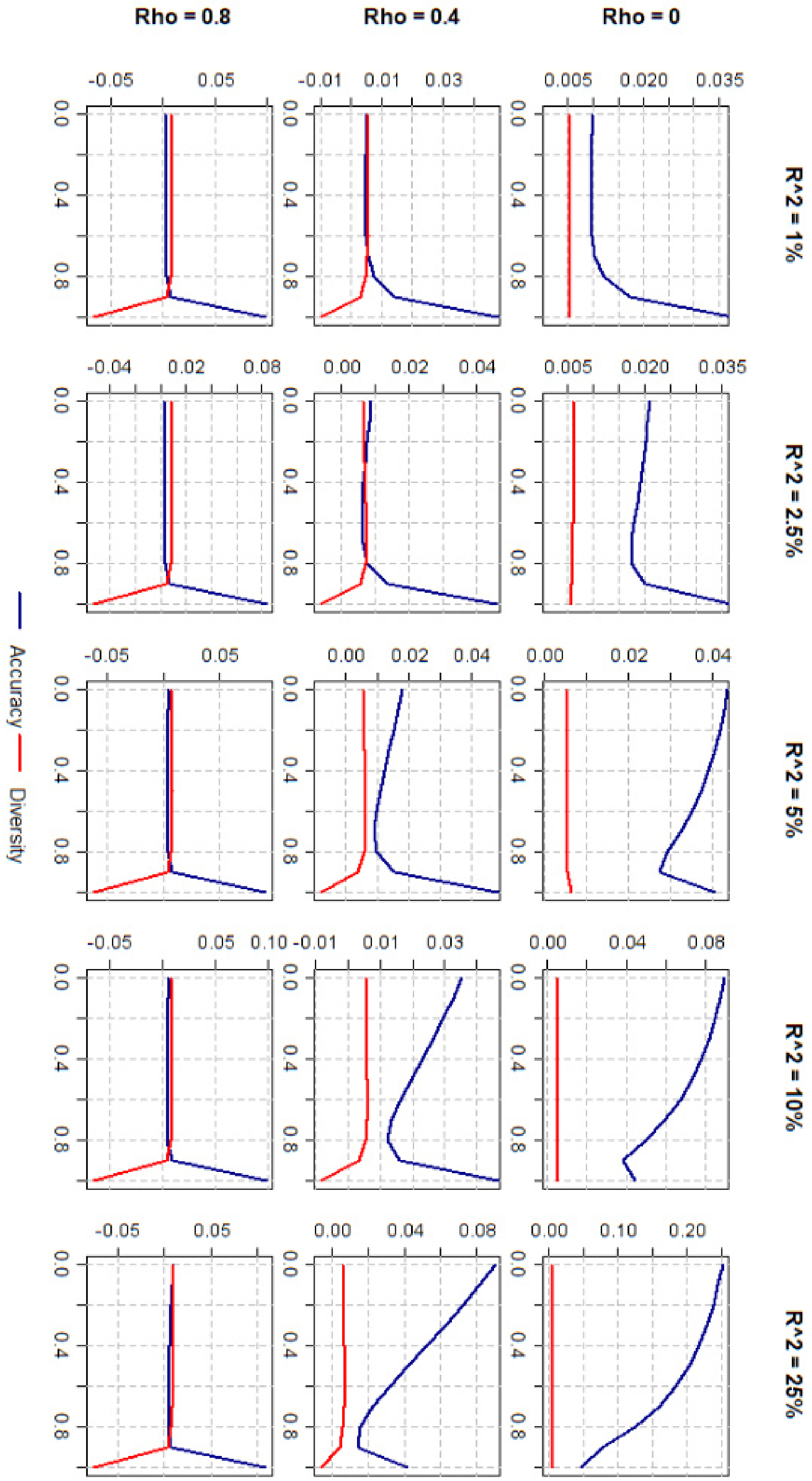

Figure III.10: Accuracy-Diversity decomposition of the uNCL. The x axis has the λ hyperparameter values in increasing order. 'Accuracy' is defined as the sum of the bias and variance terms from the bias-variance-covariance decomposition, while 'diversity' is defined as the covariance component, following Brown, Wyatt, and Tino (2005). The plots show that uNCL cannot really be regarded as an optimisation of the accuracy of the individual models against their diversity.

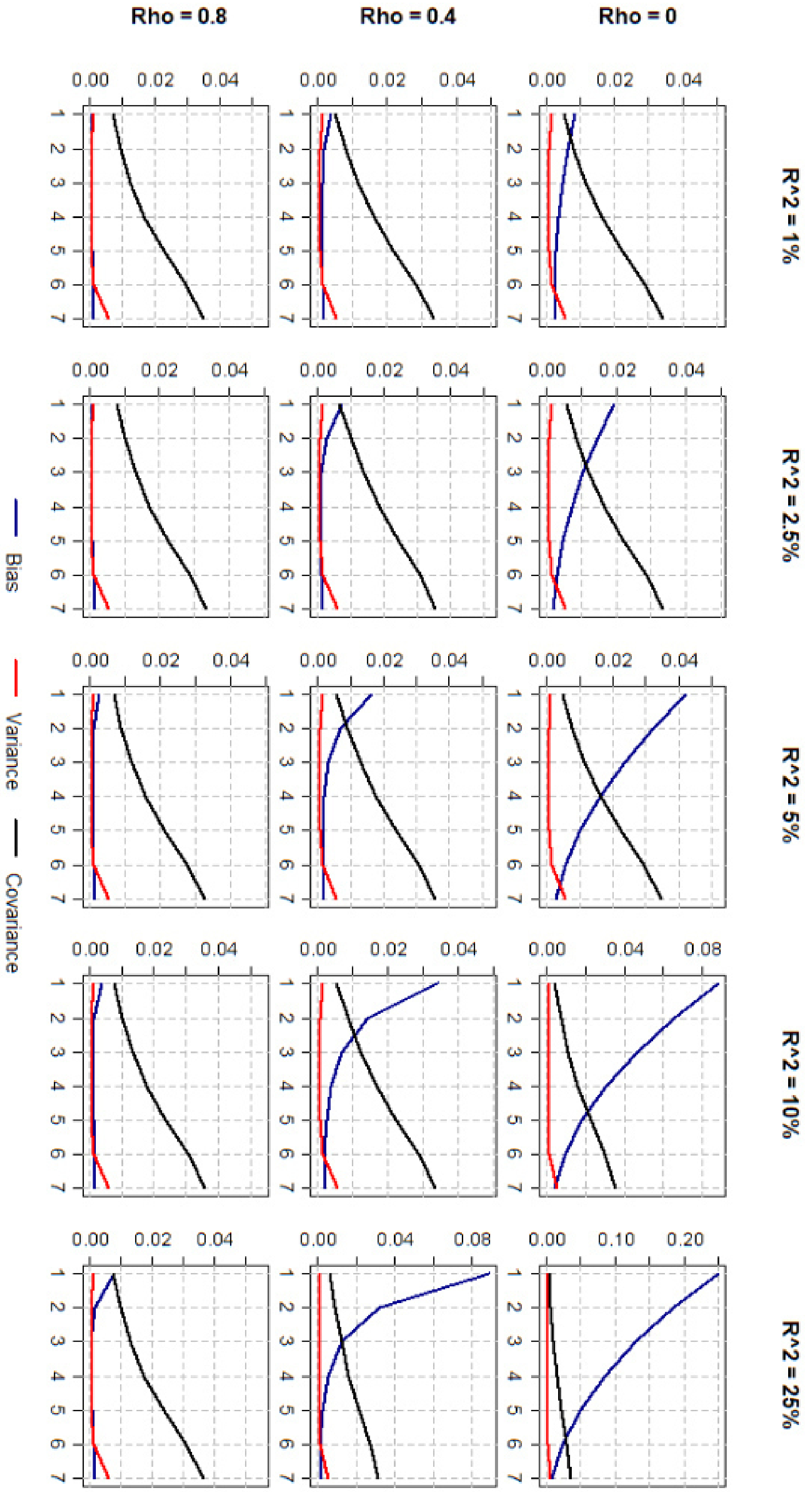

Figure III.11: Bias-variance-covariance decomposition of CSR. The x axis has the dimension of the individual models, k in increasing order. Note that $k = 8$ is not included, because it has no covariance value (there is only one individual model).

of these explanations, if any, causes the different behaviour of the NCL algorithm in this study.

Figure III.11 plots the estimated bias-variance-covariance decomposition of CSR. The plot shows that CSR has a substantial effect on the covariance component, which increases monotonically in k[22], while the variance component is mostly flat. This is in contrast to the uNCL, which has a mostly flat covariance and monotonically increasing variance component. This shows that although the two methods both act as inverse regularisers and optimise the bias-variance trade-off, they do it in a rather different way. In the terms of the bias-variance-covariance decomposition, the uNCL decreases the bias at the cost of increasing the variance, while the covariance is mostly flat; the CSR, in contrast, decreases the bias at the cost of increasing the covariance, while the variance component is mostly flat.

[22] The monotonically increasing covariance component is easy to make sense intuitively. Increasing k means that the individual models share more of the same predictors; thus, their 'information sets' have a higher overlap and their forecasts are less diverse.

Empirical Application

In this section, I apply the stage I and stage II inverse regularisers and the traditional regularisers, meaning the LASSO and ridge regression to forecasting the US equity premium.

IV.1 Data and forecasting methods

I forecast the US (log) equity premium. As noted in the previous review, this dataset has a relatively low signal-to-noise ratio and many predictors, and is thus suitable for regularisation or inverse regularisation methods. Notably, the LASSO has been applied widely to forecast the US equity premium (Rapach, Strauss, and Zhou, 2013), (Freyberger, Neuhierl, and Weber, 2020), (Kozak, Nagel, and Santosh, 2020), (Elliott, Gargano, and Timmermann, 2013). The aim of this application is to validate the results of my simulation in a dataset with suitably low signal-to-noise ratio and many predictors, and to provide a through comparison of the considered methods.

I use a version of the data originally used in Welch and Goyal (2007), and later also in several other studies, including Campbell and Thompson (2007), Rapach, Strauss, and Zhou (2010) and Elliott, Gargano, and Timmermann (2013). Notably, Rapach, Strauss, and Zhou (2010) use the uOLS on this same database, and Elliott, Gargano, and Timmermann (2013) uses CSR and the LASSO on a subset of this database, which gives a natural comparison with their results. The dataset is available at Amit Goyal's website, and consists of quarterly data of S&P500 returns (including dividends), a risk free rate, and 15 other macroeconomic or financial variables that are often used to forecast the equity premium. For a more detailed description of the 15 predictors, see Welch and Goyal (2007).

I define the log equity premium as the log of the returns on the S&P500 (including dividends) minus the log of the lagged US three-month treasury bill yields. I use quarterly data from 1947Q2 to 2020Q4 for the log equity premium and from 1947Q1 to 2020Q3 for the predictors[1].

[1] Elliott, Gargano, and Timmermann (2013) also forecasts the quarterly US log equity premium with a subset of my predictors, and their dataset starts at the same quarter as mine.

I forecast the log equity premium with the 15 predictors with each of the 6 methods examined in the simulations. For CSR, $k = 1, 2, \ldots, 15$. For the LASSO, ridge, ELASSO and ERidge I determined the λ sequence just like in the simulations, and I also use $n_\lambda = 40$ for the LASSO and ridge and $n_\lambda = 50$ for the ELASSO and ERidge. For uNCL, I use $[0, 0.1, \ldots, 0.9, 1]$ as the λ sequence, just like in the simulations. However, I set the stopping threshold at 10^{-10}, the learning rate at 0.1 and the maximum number of iterations at 100000[2]. Additionally, I trained the uNCL models in an increasing order of λ-s. This meant that I could use the final parameters from the model with hyperparameter $\lambda - 0.1$ as a starting point in the gradient descent of the model with hyperparameter λ to speed up convergence. I train all of the models with and intercept included.

Following Welch and Goyal (2007), Campbell and Thompson (2007), Rapach, Strauss, and Zhou (2010), Rapach (2013) and Elliott, Gargano, and Timmermann (2013) among many others, I use and expanding estimation window, and estimate each model each time the window is expanded. I use data from 1947Q1 to 1964Q4 as an initial estimation window, and evaluation starts at 1965Q1[3].

For the ELASSO and ERidge, an initial combination weight estimation window is also needed. To this end, I estimate the uOLS out-of-sample forecasts for 1960Q1 to 1964Q4 as well (again, using an expanding window), and estimate the combination weights initially on this 1960Q1 to 1964Q4 quasi out-of-sample data. Later, this combination weight estimation window is expanded the same way as the estimation window of the other methods.

In addition to the 'unrestricted' forecasts from each model, I also evaluate a restricted version of the forecasts of each method. Following Campbell and Thompson (2007), I set negative forecasts to zero. Campbell and Thompson (2007) suggest that theoretical considerations imply that the log equity premium should always be positive, and they show that setting negative forecasts to zero improves upon unrestricted forecasts of the log equity premium. While $Rapach, Strauss, AndZhou$(2010) find that the nonnegativity constraint is never binding for the uOLS method, the regularisers and inverse regularisers I compare in this study might have substantially more varied forecasts than the relatively smooth forecasts of the uOLS. As such, negative values come up, and the nonnegativity constraint can reduce "wrong-way" variance.

[2] I had to lower the learning rate (from 0.5, which I used in the simulations) to 0.1, because the algorithm did not converge with a higher learning rate. This resulted in a slow convergence to the optimum, so I had to increase the maximum number of iterations.

[3] The initial estimation window is the same as in Elliott, Gargano, and Timmermann (2013), and the evaluation window also starts at the same quarter (albeit ends later, due to the availability of more data).

IV.2 Forecast evaluation

Following the usual practice of the literature on forecasting the equity premium, as can be see in Rapach, Strauss, and Zhou (2010), Rapach (2013), and Campbell and Thompson (2007), for example, I compute out-of-sample R^2-s as a measure of performance for each method and each hyperparameter value. This metric is defined as (Campbell and Thompson, 2007):

$$R^2_{OoS} = 1 - \sum_{k=q_0}^{q} \frac{(r_k - \hat{r}_k)^2}{(r_k - \overline{r}_k)^2} \tag{IV.1}$$

Where q_0 and q are the starting and ending indices of the out-of-sample forecast evaluation period, respectively, r_k is the actual log equity premium at time k, $\hat{r}_k$ is the forecast of the log equity premium at time k, and $\overline{r}_k$ is the historical average log equity premium (the average of the actual log equity premiums from 1947Q2 to $k-1$).

This measure is analogous the the in-sample R^2, but is 'out-of-sample' in the sense that the models used to generate the forecast $\hat{r}_k$ do not use r_k and the corresponding predictors for the estimation, and the historical average also do not contain r_k. As such, this measure does not contain any information in the evaluation that was not available when r_k would be estimated. Additionally, this measure gives an intuitive idea on how much information the predictors contain about the variable to be forecast; the R^2_{OoS} roughy shows the portion of the variance that is explained by the forecasts. Notably, a R^2_{OoS} above (below) 0 means that the forecasts have better (worse) forecasting performance than the historical average benchmark.

I also test if the different forecasting methods have a significantly lower out-of-sample MSE than the historical average benchmark. To this end, I carry out Clark & West tests that test the null hypothesis that $R^2_{OoS} \leq 0$ against the alternative hypothesis that R^2_{OoS} for nested models.

Finally, I evaluate the performance of each method in an economic sense. I follow the approach of Welch and Goyal (2007), Campbell and Thompson (2007), Rapach, Strauss, and Zhou (2010) and Elliott, Gargano, and Timmermann (2013) among others, and evaluate the utility gains that a mean-variance investor would have realised if they had reallocated their investments between the S&P500 index and short-term US treasury bills based on the equity premium forecasts or the historical average at the end of each quarter. An investor with risk aversion parameter γ who forecasts the equity premium with the historical average allocates the following share of their portfolio to equities at the end of period t:

$$w_{avg,t+1} = \frac{1}{\gamma} \frac{\hat{r}_{t+1}}{\hat{\sigma}_{t+1}} \tag{IV.2}$$

Where $\overline{r}_{t+1}$ is the forecast of the equity premium at time $t+1$ and $\hat{\sigma}^2_{t+1}$ is the volatility forecast of the investor. Following Rapach, Strauss, and Zhou (2010), I assume that the investor forecasts the volatility with its historical average over the last ten years and that $\gamma = 3$. I also follow them in ruling out short selling or excessive risk taking. As such, if $w_{avg,t+1} < 0$, I set it to zero, and if $w_{avg,t+1}$ is over 1.5, I set it to 1.5.

The investor realises an average utility level of

$$\overline{\nu}_{avg} = \overline{\mu}_{avg} - \frac{1}{2}\gamma\overline{\sigma}^2_{avg} \tag{IV.3}$$

Where $\overline{\mu}_{avg}$ is the sample mean and $\overline{\sigma}^2_{avg}$ is the sample variance of the portfolio allocated according to the trading strategy defined above.

I calculate the average utility gains $\overline{\nu}_{avg}$ for all of the equity premium forecasts generated by any of the methods I consider, both with fixed and validated hyperparameter values. I also calculate the average utility the investor who uses historical returns up till time t the reallocate her portfolio at the end of quarter t. Let us denote this average utility by $\overline{\nu}_{hist}$. Then, I measure the economic significance of the forecasts as the excess utility the investor who allocates her portfolio based on the forecasts realises over the investor who uses the historical average to reallocate her portfolio, multiplied by 400:

$$CER = 400 * (\overline{\nu}_{avg} - \overline{\nu}_{hist}) \tag{IV.4}$$

I multiply the difference so that the result can be interpreted as an annual measure in percentages[4] Note that this measure is a 'certainty equivalent return' gain, meaning that the investor is indifferent between the returns of the historical average-based portfolio plus this certainty equivalent return, and the returns of the forecast-based portfolio.

IV.3 Results with fixed hyperparameters

Figure IV.1 plots the R^2_{OoS} of the unrestricted and nonnegativity restricted forecasts for each method and each considered hyperparameter value. The x axis labels are the λ of the uNCL. For CSR, k increases to the left from $k = 1$ to $k = 15$. For the LASSO, ridge, ELASSO and ERidge, the leftmost value is $\lambda = \lambda_{max}$ and the rightmost is $\lambda = 0$. The lines are linear interpolations between the point for which I estimated the models and have exact R^2_{OoS} values. Note that some of the methods have very large negative R^2_{OoS} values for some the the hyperparameters considered, which I do not plot for the sake of giving a better visual comparison of the models at the hyperparameter values that have roughly similar performance. Nevertheless, Appendix B.1 contains a table with the R^2_{OoS},

[4] I multiply by 4 to annualise to quarterly data, and multiply by 100 so that the result can be interpreted as percentages.

Clark & West test p-value and CER gains for each method and each hyperparameter value.

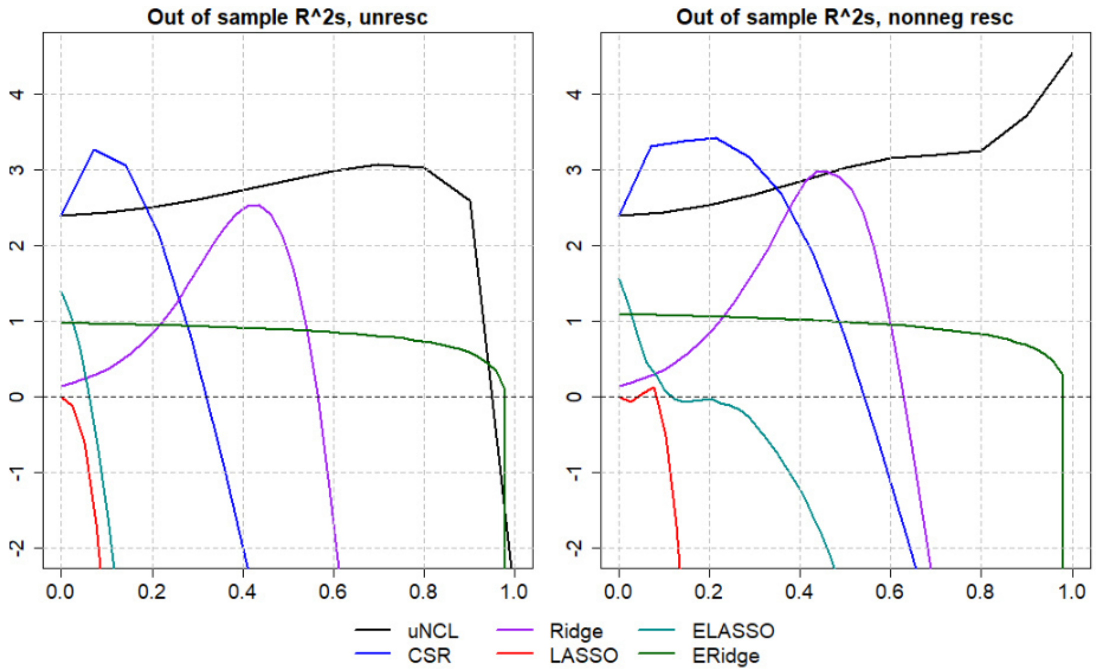

Figure IV.1: Out-of-sample R^2-s of the unrestricted and nonnegativity restricted forecasts. The plot on the left is of the unrestricted, while the plot on the right is of the nonnegativity restricted forecasts. X axis values show the λ values of the uNCL. For CSR, k increases to the right. For the ELASSO, ERidge, LASSO and ridge, the leftmost value corresponds to $\lambda = \lambda_{max}$ and the rightmost value ot $\lambda = 0$. The lines are linear interpolations between the actual parameter values I estimated the models with.

The figure shows that CSR and uNCL are the two best performing methods if their hyperparameters are set optimally. CSR with $k = 2$ performs the best among the unrestricted forecasts, and $uNCL$ with $\lambda = 1$ is the best among the nonnegativity restricted forecasts. From the other methods only ridge regression has a performance relatively close to that of CSR and uNCL. These results are in line with the findings in simulations, where I showed that stage I inverse regularisers tend to outperform stage II inverse regularisers and traditional regularisers like the LASSO or ridge regression.

The plots also indicate that the nonnegativity constraints can improve the performance of CSR, uNCL and ridge regression substantially. These are, as noted previously, exactly the models that perform relatively well even without the restriction. It is also worth noting that the optimal value of the hyperparameters differ for the unrestricted and nonnegativity restricted forecasts. The general trend is that the hyperparameters that are optimal for the nonnegativity restricted forecasts mean a weaker shrinkage than the optimal hyperparameter values for the unrestricted forecasts. Intuitively, the nonnegativity restriction reduces 'wrong' variance, and therefore decreases the cost of decreasing the bias in exchange for a higher variance.

Figure IV.2 plots the Clark-West test (Clark and West, 2007) p-values (in percentages) for each model and each hyperparameter. The plots have the same structure as IV.1, with

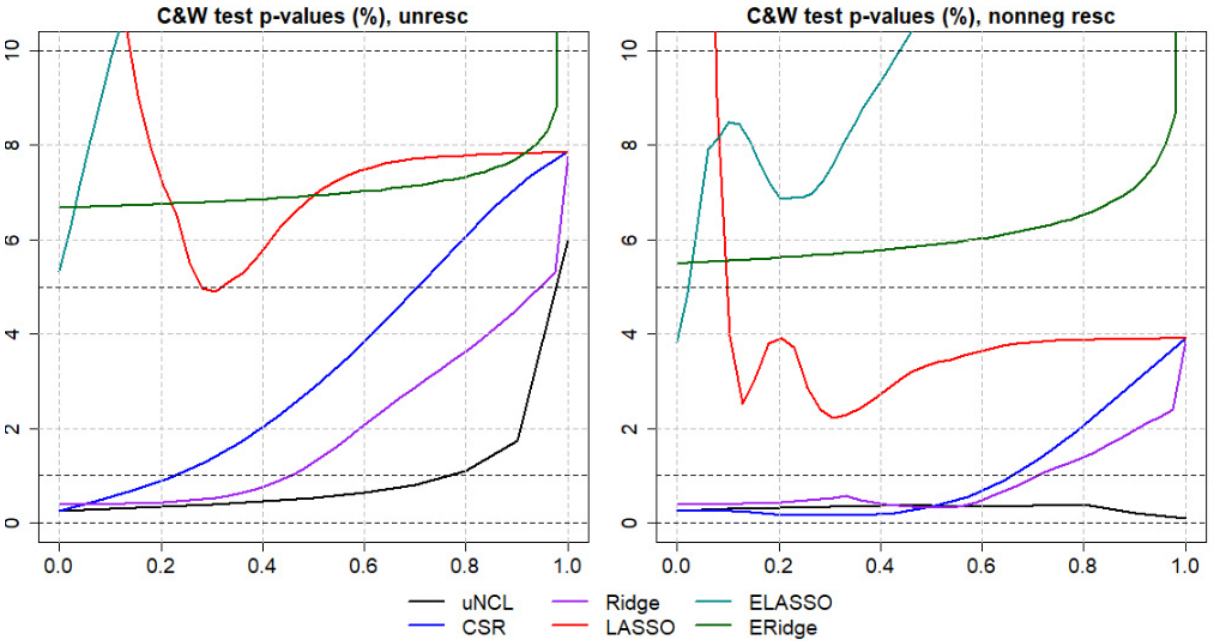

Figure IV.2: C&W test p-values (%) of the unrestricted and nonnegativity restricted forecasts.The plot on the left is of the unrestricted, while the plot on the right is of the nonnegativity restricted forecasts. The plot uses the same structure as figure IV.1. Only p-values between 0 and 10 are shown on the plots.

the only difference that only values between 0 and 10 are shown on the y axis. The plots show that the LASSO, ELASSO and ERidge are usually (or, except for the LASSO, never) significantly outperform the historical average benchmark at the 5% significance level. The other 3 methods, CSR, ridge and the uNCL are significant for most hyperparameter values at the 5% level, and for some hyperparameters even at the 1% level. Adding the nonnegativity restrictions improve the significance of the forecasts of the 3 best performing model and the LASSO, but the ELASSO and ERidge still do not outperform at the 5% level. These results show that the stage I inverse regularisers perform very well and often better than the competing methods not only in terms of their R^2_{OoS}, but also in statistical significance.

Figure IV.3 plots the annual certainty equivalent returns that an investor would have realised if she allocated her portfolio between equities and treasury bill over the investor who allocated her portfolio between the same assets, but based on the prior historical average of the equity premium.

Note that there is only one plot for the CER values, because the unrestricted and nonnegativity restricted forecasts generate the same portfolio allocation in each period. This is due to the fact that the two forecasts differ only when the unrestricted forecast is negative. In this case, both the unrestricted negative and the restricted zero forecast allocate a weight of 0 to equities.[5]

[5] Because treasury bills are dominate a zero or negative yield, but more volatile asset, the optimal weight on equities would be negative for the investor in these cases. However, short selling is ruled out by construct so a weight of 0 is given to equities.

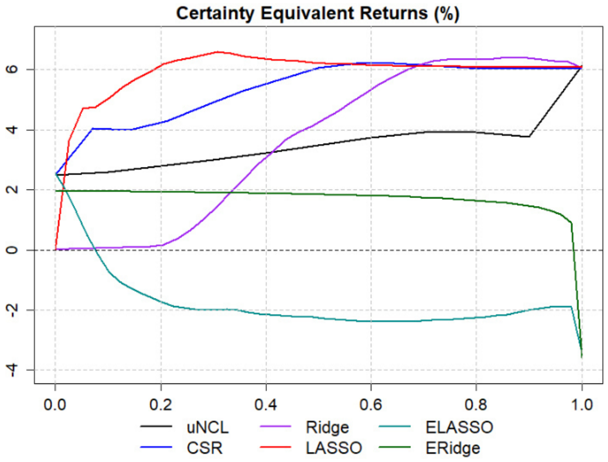

Figure IV.3: Certainty equivalent returns for the investor who allocated her portfolio between equities and the treasury bills based on model-based forecasts of the equity premium would have realised over the investor who allocated her portfolio based on the prior historical equity premium. The plot has the same structure as IV.1. Only values between -4 and 7 are plotted. Note that the CER values are the same for the unrestricted and nonnegativity restricted forecasts.

Interestingly, the CER values tend to be the highest for a low level of shrinkage (except for the stage II inverse regularisers, which do not perform well). The absolute highest value is achieved by the LASSO. This is in contrast to the previous R^2_{OoS} and Clark-West test results, where a medium level of shrinkage produced the best results, and the LASSO and ridge with small λ parameters and CSR with high k parameters performed poorly. This is due to the fact that I limit the equity weights to lie in the interval $[0, 1.5]$. This ensures that outlier forecasts do not have a strong effect on the investment decision of the investor, and a large leverage or short selling do not cause a significant drop in investment returns if the forecasts are off. In other words, restricting the equity weights to lie in the interval $[0, 1.5]$ reduces the negative effect a highly variable forecast has on investment returns, and makes having a small bias forecast more important. As such, a low level of shrinkage produces the best results for the investor. To test this hypothesis, a widened the restriction so that equity results must lie in the much wider interval of $[-5, 5]$, which led to highly deteriorating (usually negative CER values) performance for the low shrinkage models[6].

To sum up, the stage I inverse regularisers uNCL and CSR perform remarkably well when their hyperparameters are set optimally. For the unrestricted (nonnegativity restricted) uNCL, this is $\lambda = 0.7$ ($\lambda = 1.0$) and for the CSR, it is $k = 2$ ($k = 3$). The uNCL and CSR outperform both the stage II inverse regualrisers ELASSO and ERidge, and the traditional regularisation methods LASSO and ridge. The uNCL and CSR fare well in both statistical and economic tests of significance as well, and outperform the historical average.

[6] For the sake of brevity, the actual results of this robustness check are not presented in my paper.

	Unrestricted			Nonnegatitivty rest	
Method	R^2_{OoS}	p-value	CER	R^2_{OoS}	p-value
uNCL	1.60	6.39	3.13	2.45	1.64
CSR	0.52	4.23	3.65	1.07	1.51
Ridge	-1.54	1.86	3.86	-1.92	1.77
LASSO	-5.08	0.49	3.37	-6.62	2.42
ELASSO	-3.07	38.09	1.6	0.35	13.87
ERidge	-0.19	18.53	1.19	-0.08	19.17
uOLS	1.60	2.55	2.23	1.60	2.55

Table IV.4: Results for validated hyperparameters. The left three columns plot the R^2_{OoS} values (Campbell and Thompson, 2007), the Clark & West test (Clark and West, 2007) p-values and the certainty equivalents gains (CER) of the unrestricted forecasts. The right two columns plot the R^2_{OoS} and Clark & West test p-value for the nonnegativity restricted forecasts. Note that the CER values are the same for the nonnegativity restricted and unrestricted forecasts, so I do not show the same CER value for the nonnegativity restricted forecasts in a different columns.,

IV.4 Results with validated hyperparameters

In practical forecasting applications, the optimal values of the hyperparameters are not known prior to forecasting, but have to be estimated form the data. Therefore, I validate the hyperparameters of each method from the data to check the robustness of my results. I validate the hyperparameters the same way I did in the simulations. I generate out-of-sample forecasts with an expanding estimation window. This means that I use all data available before and no data available after time $t + 1$ to estimate the models that forecast the log equity premium at time $t+1$. I forecast the data sequentially; after having forecast the log equity premium at time $t + 1$ with a model that was estimated on data available up to (and including) time t, I forecast the equity premium at time $t + 2$ with a model estimated on an 'expanded' dataset that includes all available data up til time $t + 1$. I use the data from 1947Q1 to 1964Q4 for the equity premium and from 1946Q4 to 1964Q3 for the predictors as an initial estimation window that is only used for the estimation of the models (this remains the same as in the previous section with the fixed hyperparameters). Out of sample forecasting thus starts at 1965Q1. The first 20 out of sample forecasts are used as an initial validation window, meaning the first quarter I validate the hyperparameters for is 1970Q1. I validate the hyperparameters at time t by choosing the hyperparameter that has the lowest mean squared error on prior data. By using all available prior data at all times to measure the past performance of the method to validate the hyperparameters, I use an expanding window for the validation as well.

Table IV.4 presents the performance of the models with validated hyperparameters for the unrestricted and nonnegativity restricted forecasts. The performance of all models decreases sharply in terms of the R^2_{OoS} for all methods when compared to their R^2_{OoS} values with the hyperparameters chosen optimally. Only 2 of the 7 methods I examine have a positive R^2_{OoS} values. The uOLS forecast, which is included as a benchmark, performs at the same rate in R^2_{OoS} as the uNCL, and outperforms the only other method that has a positive R^2_{OoS}. Additionally, uNCL does not outperform the historical average significantly at the 5% level.

The nonnegativity restriction improves the performance of the uNCL, CSR and ELASSO. However, CSR and ELASSO has an R^2_{OoS} that is still lower than that of the uOLS. On the other hand, we see that the uNCL outperforms th uOLS with the nonnegativity restrictions by about 0.85%.

Interestingly, most of the methods produce economically significant forecast that give rise to a sizeable CER when used for portfolio allocation. Excluding the stage II inverse regularisers, which, as usual, do not perform well, the methods all have CERs above 3%, which is a substantial improvement over the 2.23% CER of the uOLS. In general, the methods that can produce a low bias - high variance forecast perform well in asset allocation, because of the restriction on the portfolio equity weights to lie in the interval $[0, 1.5]$.

Figure IV.5 plots the actual log equity premium and its validated unrestricted and nonnegativity restricted forecasts from the 3 best performing model, the uNCL, uOLS and CSR. The plots clearly show that the stage I inverse regularisers uNCL and CSR produce more varying forecasts, whereas the uOLS forecasts are quite smooth. After about 1985, the uOLS forecast is almost constant. The uNCL and CSR forecasts also become less time-dependent as time goes on, but at a much slower rate. The unconstrained uNCL and CSR forecasts become almost constant only after 2010, and the nonnegativity constrained forecasts - which perform better than the unconstrained forecasts in terms of R^2_{OoS} - show substantial time-dependence even at the end of the evaluation period.

The validated forecasts become more stable for two reasons. One, the model estimation period is longer for later periods, which reduces the variance. Two, the hyperparameters chosen in the validation indicate a stronger level of shrinkage at the end of the sample. This is shown in figure IV.6, which plots the chosen values of the hyperparameters over time. Because the chosen hyperparameters are exactly the hyperparameters that give the best performance on past data, an increasing (decreasing) trend on the plot implies that a lower (stronger) form of shrinkage is better on the newly added data[7]. The fact that a stronger shrinkage is optimal at later periods is interesting, because due to the variance-reducing effect of using longer windows for model estimation at later periods, a weaker form of shrinkage should be optimal if the underlying data generating process stayed the same. A weaker form of shrinkage might be optimal at later periods because at least some of the predictors may have lost at least some of their predicting capabilities over time. Alternatively, the predicting power could have remained, but the direction (positive or negative) of the relationship may have changed. These results are in agreement

[7] Note that the plot is structured such that values closer to min indicate stronger, and values closer to max indicate weaker shrinkage.

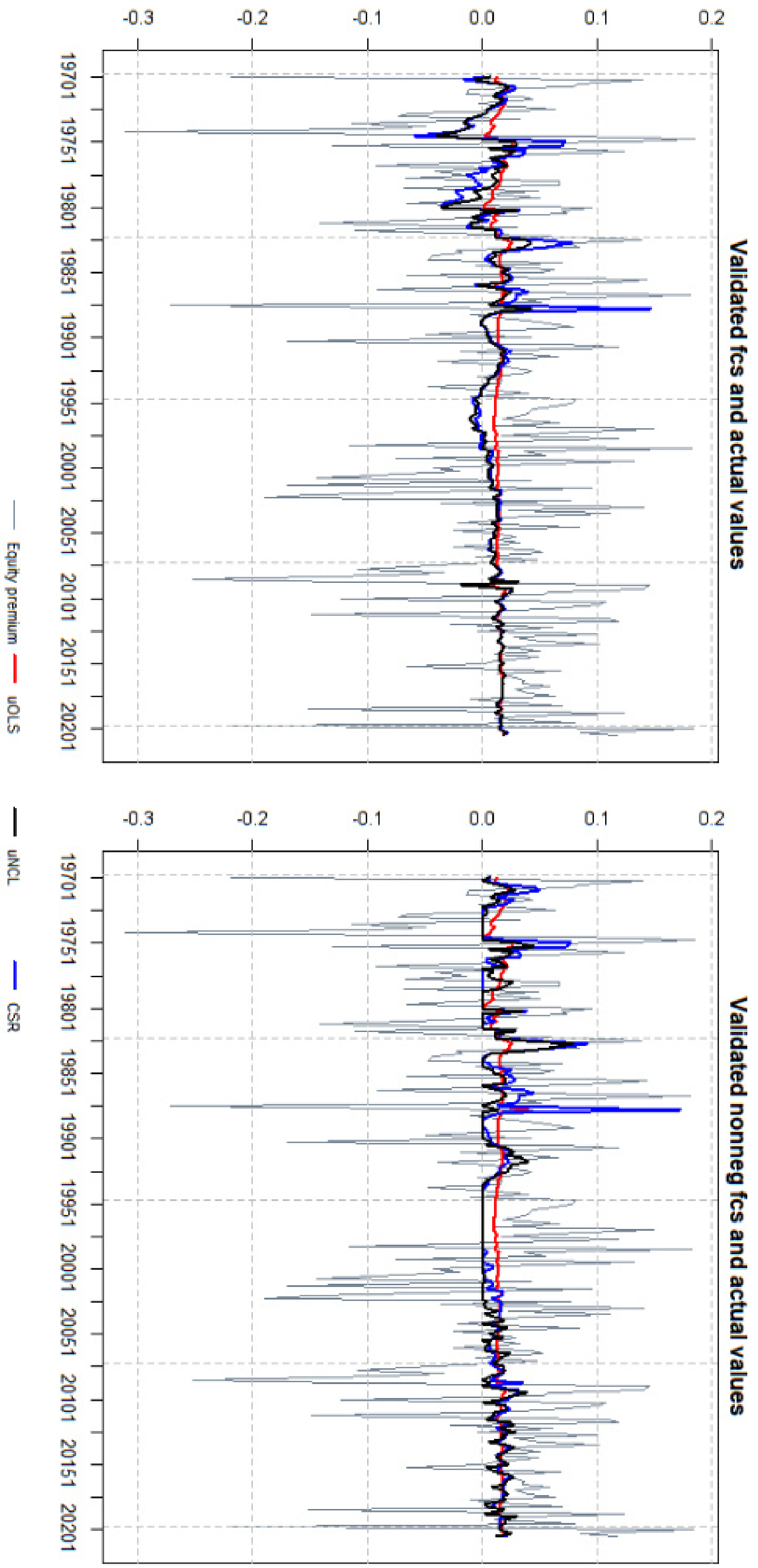

Figure IV.5: Actual log equity premiums and its validated out-of-sample forecasts from the uOLS, uNCL and CSR models without restrictions (left) and with nonnegativity restrictions (right). The plots show that the nonnegativity restricted forecasts are much more volatile even near the end of the evaluation sample.

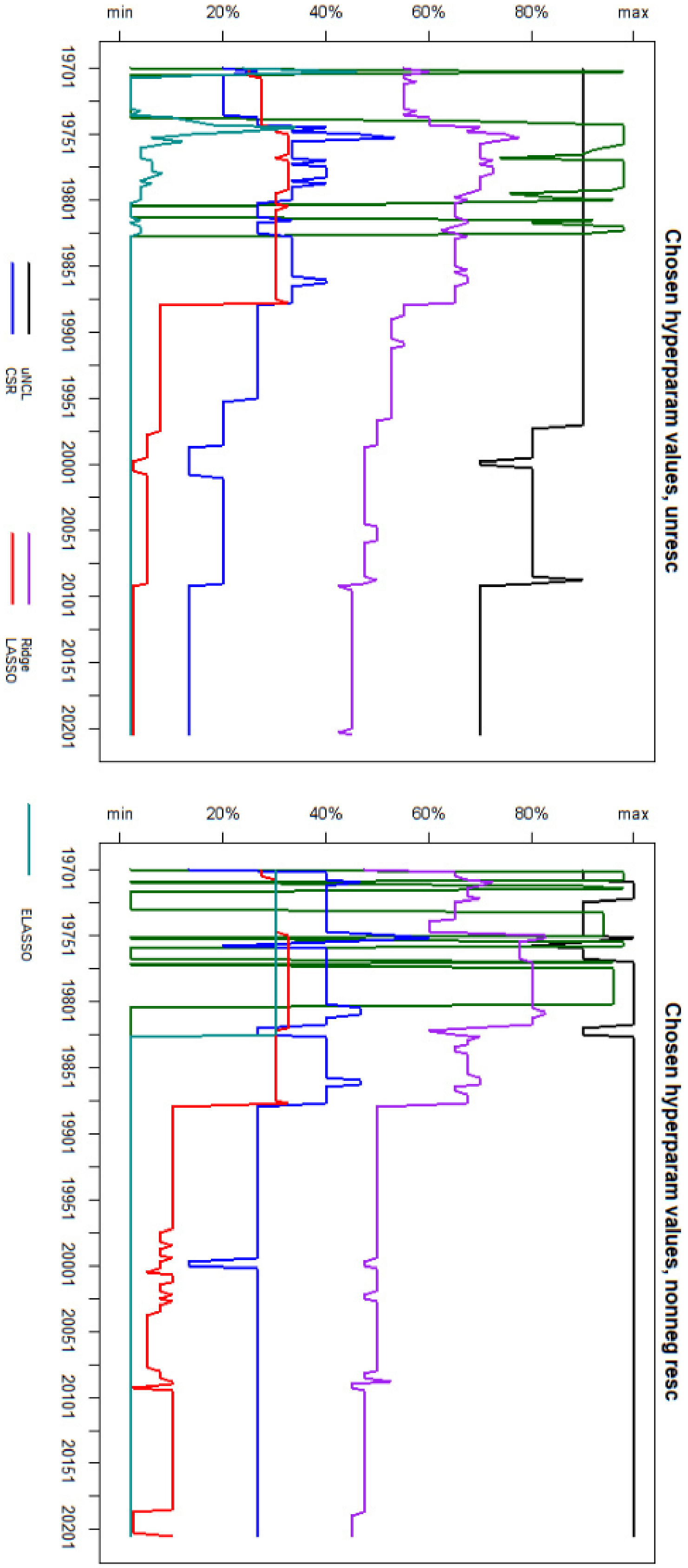

Figure IV.6: Chosen hyperparameter values over time. The values on the y axis are given as a percentage of the maximal value of the hyperparameter for the uNCL and CSR. For the LASSO, ridge, ELASSO and ERidge, the values on the y axis are given as the 'index of the chosen hyperparameter value' divided by the number of hyperparameter values ($n = 40$ for the LASSO and ridge and $n = 50$ for the ELASSO and ERidge). Note that $\lambda = \lambda_{max}$ is indexed by 1 and $\lambda = 0$ is indexed by the highest index for these models.

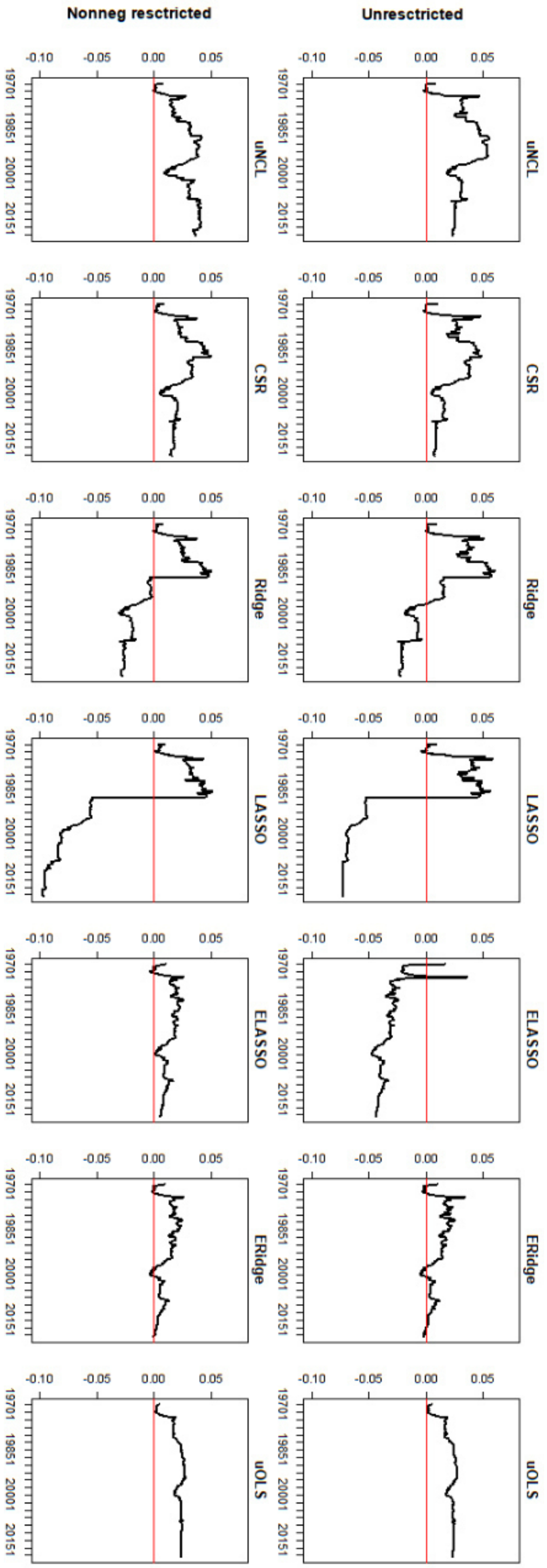

Figure IV.7: Rolling historical average and model-based forecast MSE differences. The plots at time t show the difference of the MSE of the historical average and the forecasts generated by a given model with validated hyperparameter values. If the difference is positive (negative) at time t, the model-based forecast outperforms (underperforms) the historical average when evaluated on data prior to and including time t.

with previous research that indicate the presence of structural breaks in the relationship between the equity premium and other macroeconomic and financial variables.

Figure IV.7 shows the difference of the MSE of the prior historical average forecast and the forecasts produced by a given method with validated hyperparameters on data prior to and including the values on the x axis. A positive (negative) value indicates that the model-based forecasts outperformed (underperformed) the historical average when evaluated on data prior to and including the time written on the x axis. This plot is often used as a graphical illustration of the change of the performance of the models over time. The rate at which the lines move up (down) can be interpreted as the rate by which the model-based forecast outperforms (underperforms) the historical average. If the degree of outperformance were constant over the evaluation sample, we would expect to see a plot that moves up linearly.

The plots do not even roughly resemble the linear relationship. The forecasts of most models perform well in the early parts of the sample. The LASSO and ridge has a sudden drop of performance around 1985, after which both models consistently underperform the historical average. The uNCL and CSR have a largely overlapping period of consistent low performance in the 90s. This is followed by a period lasting till the end of the evaluation period during which the unrestricted uNCL and both the unrestricted and nonnegativity restricted CSR have a performance roughly equal to the performance of the historical average. After the 1990s, only the nonnegativity restricted uNCL can outperform the historical average (and the uOLS), although even for this method most of the outperformance is concentrated in or around the dot-com bubble and the financial crisis of 2008.

IV.5 Results for recessions and expansions

A common finding in the empirical literature (Rapach, 2013), (Haase and Neuenkirch, 2022), and also explained by some theoretic models (Cujean and Hasler, 2017) on equity premium forecasting is that predictability is concentrated in recessions, and most predictors have limited or even non-existent forecasting ability over the historical average in expansions.

I compare the predictive power of the six methods I examine in this study. I use data from the National Bureau of Economic Research's Business Cycling Database to classify each quarter as a recessive or expansive period. Then, I calculate a R^2_{OoS} value for each method for the expansive and recessive quarters separately, by only taking into account the quarters that are recessive or expansive:

$$R^2_{OoS,type} = 1 - \sum_{i \in I(type)} \frac{(r_k - \hat{r}_k)^2}{(r_k - \overline{r}_k)^2} \tag{IV.5}$$

	Unrestricted		Nonnegativity restricted	
	R^2_{OoS}			
Method	Recessive	Expansive	Recessive	Expansive
uNCL	5.28	-1.57	6.27	-0.84
CSR	5.69	-3.93	6.11	-3.28
Ridge	6.39	-8.38	4.71	-7.63
LASSO	9.44	-17.59	5.62	-17.18
ELASSO	-1.68	-4.26	4.15	-2.94
ERidge	4.47	-4.21	4.59	-4.10
uOLS	3.14	0.28	3.14	0.28

Table IV.8: Results for recessive and expansive periods for the validated forecasts. Evaluation sample is from 1970Q1 to 2020Q4.

Where type is either 'recession' or 'expansion', and $I(type)$ is the indices of the quarters with $type$.

Table IV.8 shows the results. My findings are in agreement with the literature; all models produce more accurate forecasts in recessive periods by a considerable margin. Actually, the forecasts of all models - except the uOLS - even underperform the historical average in expansive periods. This finding holds both for the unrestricted and nonnegativity restricted forecasts - the nonnegativity restricted forecasts tend to perform better both in recessions and expansions, the increased performance does not cluster to economic conditions.

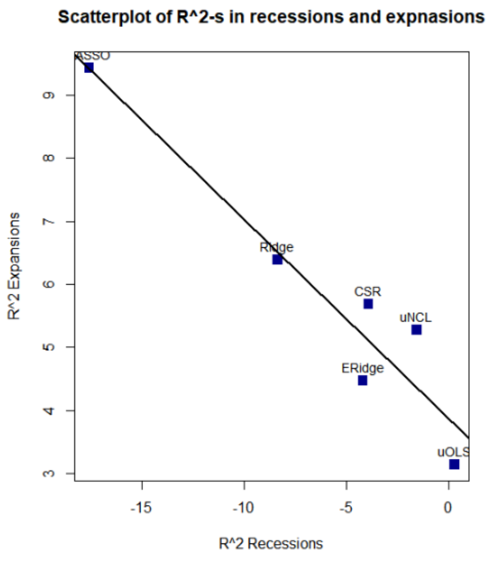

Figure IV.9: Scatter plot of R^2_{OoS} values during recession and expansions. The x axis has the R^2_{OoS} values of the unrestricted forecasts during recessions, and the y axis has the R^2_{OoS} values of the unrestricted forecasts during expansions. The black line is the regression line of the two values. Note the negative relationship; the methods that perform well in recessions tend to perform badly in expansive periods.

Figure IV.9 plots the R^2_{OoS} values during expansions on the x and during recessions on the x axis for the unrestricted forecasts of each method except the ELASSO, which is an outlier because it has a negative R^2_{Oos} in both expansions and recessions. The plot also includes the regression line from the linear regression of the $R^2_{Oos,recession}$-s on the $R^2_{Oos,expansions}$-s. The plot shows a close fit, which has a regression β significant at the 1% level and a relatively very fit with a regression R^2 of 90.62%. This tells us that there is a quite strong relationship between the performance of the models during recessions and expansion; increasing the performance in one usually means decreasing the performance in the other. Notably, the good performance of both the uNCL and CSR is shown by the fact that the both have positive residuals, that is, they perform better in recessions than what would be expected of them based on their performance in expansions.

IV.6 Summary

In this section, I forecast the US equity premium with the two stage one IR-s, two stage two IR-s and the two traditional regularisation methods. The aim of this section was to give a comparison of the performance of the three different groups of methods in forecasting the US equity premium, and to find out if the simulation results are robust to the application of the methods to a suitably noisy and high dimensional empirical forecasting problem.

The results are in agreement with the simulations. The stage two IR-s do not outperform the equal-weighted uOLS, which was seen in the simulations but also in previous studies on the forecasting the equity premium (see Rapach, Strauss, and Zhou (2010) for an example). Additionally, the two best performing methods are the uNCL and CSR, which both outperform the traditional regularisation methods, the LASSO and ridge regression. Actually, only the uNCL, uOLS and CSR have a positive R^2_{OoS} with validated hyperparameters, and only the uNCL outperforms the uOLS (with nonnegativity restricted forecasts) over the entire evaluation sample. This shows that the stage one IR-s, especially the uNCL, can outperform the very popular LASSO by a substantial margin even in empirical applications.

Additionally, the rolling MSEs of all methods increase sharply at the begginning of the sample, which is followed by a sudden drop in the 90s. After the 90s, the performance of all forecasting models becomes much worse than previously. In fact, onyl the uOLS, uNCL and CSR seem to more-or-less have at least the same performance as the historical average, and only the nonnegativity-restrited uNCL seems to outperform the historical average, albeit only by a small margin. These results give further proof to the presence of structural breaks and the diminishing degree of predictability in equity premiums, which have been observed in previous studies as well

Conclusions

In this paper, I have introduced the notion of inverse regularisation (IR) methods, which generalise the uOLS and make it possible to optimise the bias-variance trade-off by setting the value of a hyperparameter. I also introduced a new categorisation of previous IR-s, such as the complete subset regression of Elliott, Gargano, and Timmermann (2013), Elliott, Gargano, and Timmermann (2015) and Boot and Nibbering (2019), and the ELASSO and ERidge of Diebold and Shin (2019) into stage one and stage two IR-s. My paper, using the categorisation introduced above, made two additional contributions to the forecasting literature.

First, I carried out a large scale comparison of stage one stage two IR-s as well as traditional regularisation methods like the LASSO and ridge regressions. The results of the simulation and empirical application both indicate that a) stage one IR-s generally perform well, b) stage one IR-s generally outperform stage two IR-s and c) stage one IR-s generally outperform traditional regularisation methods like the LASSO and ridge.

As such, the results suggest that the stage I IR-s, meaning the complete subset regression of Elliott, Gargano, and Timmermann (2013), which has inexplicably few applications so far, and my univariate negative correlation learning (uNCL) should see more use in future applications.

Second, I introduced a new forecasting method, which I call univariate negative correlation learning (uNCL). The method is an application of the negative correlation learning algorithm introduced by Liu and Yao (1999) and later popularised by Brown, Wyatt, and Tino (2005), which is a popular algorithm to train ensembles of neural networks in the machine learning literature. My paper examined the performance of uNCL, and showed that it usually performs on par or better than other methods in the simulations.

Additionally, the empirical application indicates that uNCL is the only method that can outperform the uOLS benchmark in terms of R^2OoS, and that it is the only method that has substantial predictive power after the 90s (if nonnegativity restrictions are imposed).

Interestingly, my simulations also establish uNCL as a stage one IR method. This is done by estimating the bias-variance decomposition of uNCL, whereby I show that uNCL optimises the bias-variance trade-off by setting the value of its hyperparameter,

λ. This result shows a serious break with previous literature on the NCL algorithm in the machine learning community. There, it was shown that uNCL trains a set of neural networks that have low covariance, but at the cost of decreasing the individual accuracy of the neural networks(Brown, Wyatt, and Tino, 2005). As such, NCL trains a set of neural networks that are inaccurate by themselves, but work well in aggregate. This lead to an accuracy-diversity trade-off, where the accuracy (squared bias plus individual network variance) of the individual models is optimised against the diversity (covariance) of the models(Brown, Wyatt, and Tino, 2005). My simulation show by estimating the bias-variance and bias-variance-covariance decomposition of uNCL that my method does not optimise this accuracy-diversity trade-off, and the covariance component is mostly flat in the hyperparameter λ.

Although my paper showed that uNCL works differently from previous applications of the NCL algorithm, it did not answer where exactly this deviation comes from. I noted some differences between uNCL and previous applications. These were a) the linearity of the models in uNCL, b) the linearity of the data generating process in my simulations, and c) the fact that I only give a subset (exactly one) of the predictors as inputs to each individual model, whereas previous applications gave all of the predictors as inputs to each individual model. I believe further research should be carried out to investigate which (if any) of these differences can be regarded as the 'point of deviation' for the uNCL.

Additionally, I note that a 'supermodel' combination of uNCL and CSR has not been considered in this paper. The uNCL assumes that the individual models are univariate; however, the NCL algorithm could be applied to train multivariate, such as 2-variate, 3-variate, etc. models as well. As such, this 'supermodel' would have two hyperparameters. One of the would be the λ, which is the hyperparameter of the NCL algorithm; the other, k would determine the number of predictors in each individual model. In this paper, I only considered the special case where either $k = 1$ or $\lambda = 0$; further research should be carried out to test the performance of the more general 'supermodel'.

I also applied the IR methods to forecasting the US equity premium. The results are in agreement with the simulations, showing that stage one IR-s perform the best. Notably, the uNCL is the only method that outperforms the uOLS (with nonnegativity restrictions imposed). Furthermore, whereas most models underperform the historical average benchmark after the 90s, the nonnegativity restricted uNCL actually achieves a slightly better performance than the historical average even in the later half of the evaluation sample.

Research in finance has widely applied the LASSO recently (Rapach, Strauss, and Zhou, 2013), (Freyberger, Neuhierl, and Weber, 2020), (Gu, Kelly, and Xiu, 2020), (Elliott, Gargano, and Timmermann, 2013), (Kozak, Nagel, and Santosh, 2020). My results indicate that from the perspective of forecasting power, the LASSO is suboptimal. I show that both my new method, the uNCL, and the complete subset regression of Elliott, Gargano, and Timmermann (2013), which, somewhat surprisingly, has not seen many applications, outperforms it in most cases. I note that Rapach and Zhou (2020) also finds that selecting a subset of the univariate forecasts from those making up the uOLS by the LASSO, and then taking an equal-weighted average of the selected forecasts is superior to simply penalising the multivariate 'kitchen sink' model by the LASSO in forecasting the equity premium. This result bears a close resemblance to my findings. They both imply that the popularity of the LASSO in financial applications is to a degree unearned from a predictive ability standpoint. Instead, my paper suggests that the concept of 'inverse regularisation', that is, optimising the bias-variance trade-off implicit in the uOLS, is superior to traditional regularisation methods like the LASSO or ridge regression.

Simulation Appendix

A.1 Error Decompositions

A.1.1 Lambda sequence for ELASSO and ERidge

It is also worth mentioning that this method of choosing the λ sequence is a slight modification of the method R's glmnet package uses to choose the λ sequence by default and this λ sequence is recommended by Friedman, Hastie, and Tibshirani (2010). The modification is that I choose the λ values so that their *fourth powers* are equidistant on $[0, maxlambda^4]$, whereas the default method chooses the λ values so that their natural logarithms are equidistant on $[ln(10^{-4}), ln(\lambda_{max})]$. The change is motivated by the fact that the natural logarithm-based method would result in a λ sequence that has many values near the max (because of the concavity of the ln function) whereas I want more values near the min, 0. This makes it possible to check whether a low or medium level of shrinkage towards equal weights, or in other words, a stronger inverse regularisation leads to superior performance.

A.1.2 Lambda sequence for LASSO and ridge

A.1.3 Simulation figures

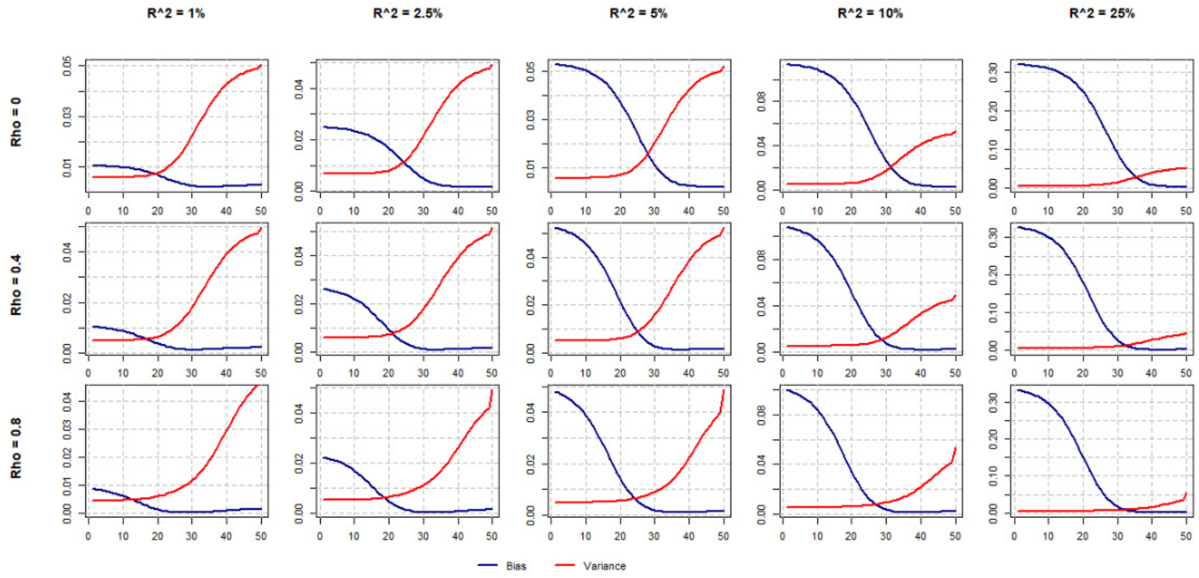

Figure A.1: Bias-variance decomposition of the LASSO. The x axis has the indices of the λ hyperparameter. $\lambda = \lambda_{max}$ is denoted by index 1, while $\lambda = 0$ is denoted by the highest index.

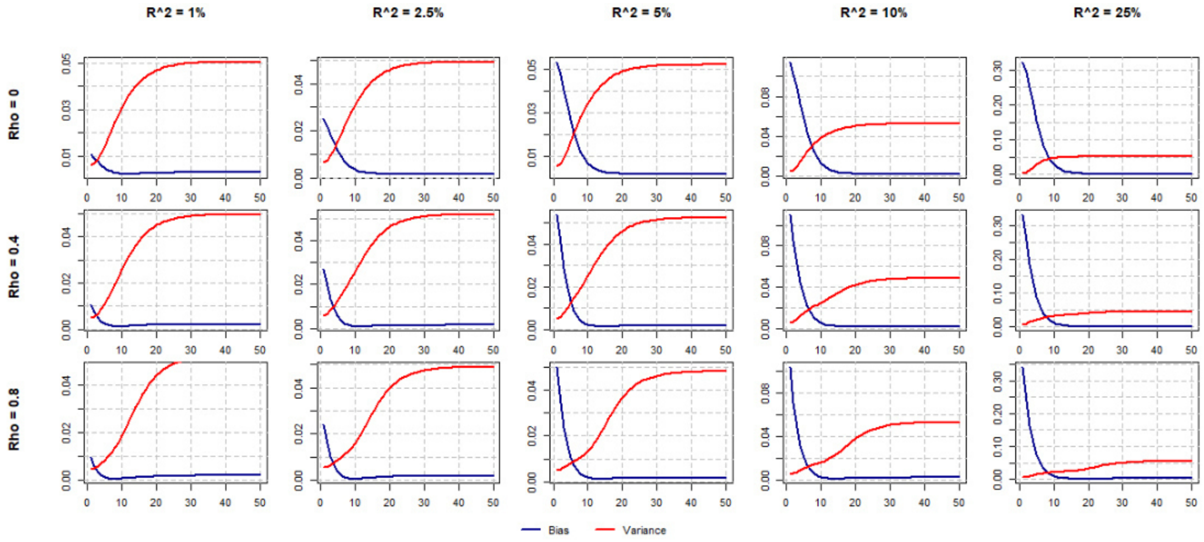

Figure A.2: Bias-variance decomposition of the ridge regression. The x axis has the indices of the λ hyperparameter. $\lambda = \lambda_{max}$ is denoted by index 1, while $\lambda = 0$ is denoted by the highest index.

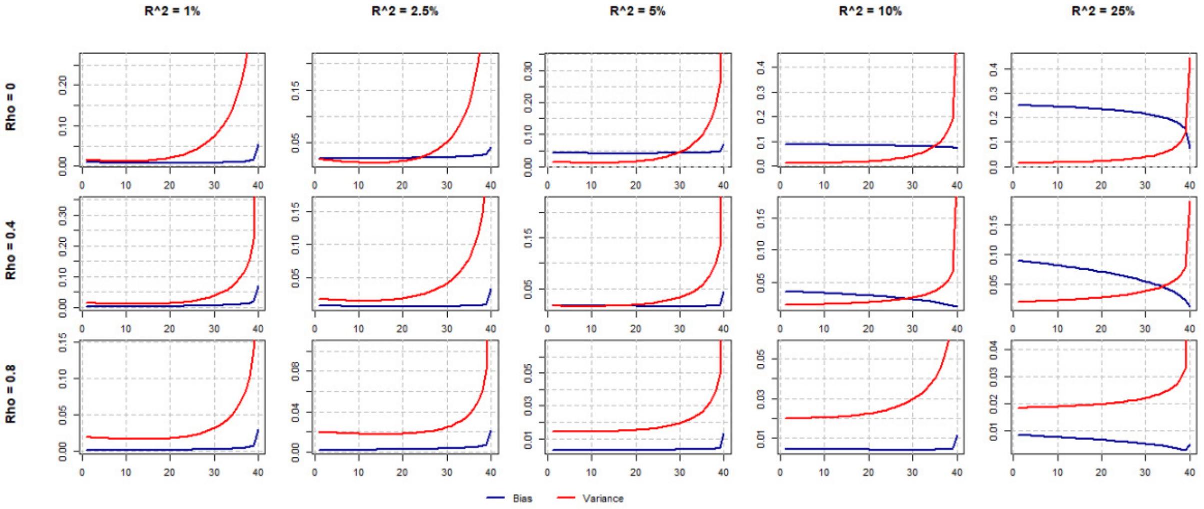

Figure A.3: Bias-variance decomposition of the ELASSO. The x axis has the indices of the λ hyper-parameter. $\lambda = \lambda_{max}$ is denoted by index 1, while $\lambda = 0$ is denoted by the highest index. Note that some extremely high values of the variance component are not plotted on the right end of the plot. This is meant to make the range of the y axis smaller, so that the values actually on the plot are easier to interpret.

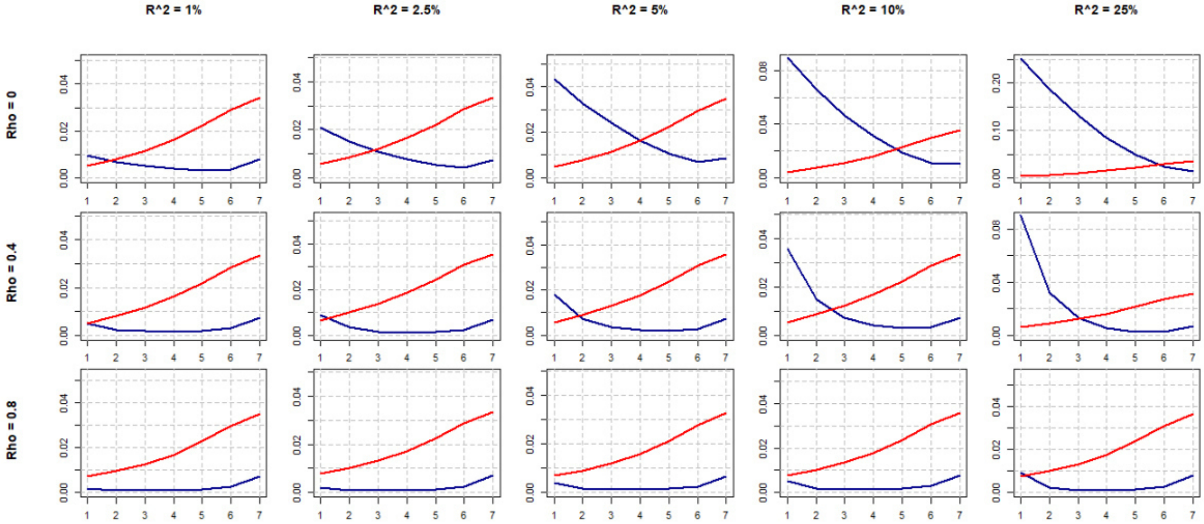

Figure A.4: Accuracy-diversity decomposition of CSR. The x axis has the dimension of the individual models, k in increasing order. Note that $k = 8$ is not included, because it has no covariance value (there is only one individual model). 'Accuracy' is defined as the sum of the bias and variance components from the bias-variance-covariance decomposition, while 'diversity' is defined as the covariance component, following Brown, Wyatt, and Tino (2005)

Empirical Application Appendix

B.1 Results with Fixed Hyperparameters

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
R^2_{OoS}	2.40	2.44	2.51	2.61	2.73	2.86	2.99	3.07	3.03	2.59	-2.66
p-value	0.27	0.30	0.35	0.40	0.46	0.54	0.65	0.81	1.10	1.73	5.98
CER	2.49	2.59	2.79	2.99	3.23	3.49	3.75	3.92	3.93	3.77	6.13

Table B.1: uNCL unrestricted evaluation results.

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
R^2_{OoS}	2.40	2.44	2.54	2.68	2.84	3.03	3.15	3.20	3.25	3.71	4.55
p-value	0.27	0.29	0.32	0.34	0.36	0.36	0.34	0.37	0.39	0.22	0.09

Table B.2: uNCL unrestricted evaluation results.

k	1	2	3	4	5	6	7	8	9	10
R^2_{OoS}	2.40	3.27	3.06	2.14	0.79	-0.85	-2.73	-4.82	-7.12	-9.60
p-value	0.27	0.46	0.68	0.95	1.29	1.73	2.25	2.87	3.56	4.31
CER	2.49	4.05	3.99	4.30	4.81	5.29	5.68	6.06	6.21	6.20
k	11	12	13	14	15					
R	-12.23	-14.96	-17.74	-20.54	-23.36					
p-value	5.11	5.92	6.69	7.38	7.88					
CER	6.14	6.06	6.04	6.03	6.05					

Table B.3: CSR nonnegativity restricted evaluation results.

k	1	2	3	4	5	6	7	8	9	10
R^2_{OoS}	2.40	3.31	3.38	3.42	3.17	2.69	1.91	0.76	-0.56	-2.02
p-value	0.27	0.28	0.24	0.17	0.15	0.17	0.22	0.35	0.55	0.89
k	11	12	13	14	15					
R^2_{OoS}	-3.57	-5.11	-6.63	-8.15	-9.71					
p-value	1.37	1.94	2.58	3.25	3.92					

Table B.4: CSR nonnegativity restricted evaluation results.

	Unrestricted			Nonnegatitivty rest	
λ index	R^2_{OoS}	p-value	CER	R^2_{OoS}	p-value
1	1.40	5.34	2.57	1.56	3.83
2	1.09	6.21	1.97	1.22	4.88
3	0.64	7.15	1.23	0.83	6.32
4	0.04	8.09	0.48	0.45	7.92
5	-0.70	8.99	-0.16	0.28	8.17
6	-1.59	9.84	-0.72	0.09	8.49
7	-2.65	10.61	-1.07	-0.03	8.45
8	-3.88	11.36	-1.27	-0.06	8.08
9	-5.30	12.05	-1.45	-0.06	7.59
10	-6.91	12.70	-1.59	-0.03	7.17
11	-8.72	13.29	-1.74	-0.02	6.87
12	-10.75	13.83	-1.86	-0.08	6.89
13	-13.00	14.34	-1.93	-0.12	6.90
14	-15.51	14.83	-1.98	-0.17	6.98
15	-18.28	15.31	-1.99	-0.27	7.24
16	-21.33	15.77	-2.00	-0.40	7.61
17	-24.66	16.17	-1.99	-0.57	8.03
18	-28.30	16.54	-2.01	-0.74	8.43
19	-32.28	16.89	-2.08	-0.92	8.81
20	-36.62	17.21	-2.12	-1.11	9.16
21	-41.34	17.50	-2.15	-1.33	9.49
22	-46.47	17.77	-2.18	-1.57	9.83
23	-52.05	18.03	-2.20	-1.85	10.20
24	-58.11	18.29	-2.22	-2.17	10.59
25	-64.70	18.56	-2.24	-2.53	11.00
26	-71.85	18.83	-2.27	-2.93	11.41
27	-79.59	19.06	-2.30	-3.38	11.83
28	-88.02	19.30	-2.33	-3.88	12.25
29	-97.18	19.53	-2.36	-4.44	12.67
30	-107.15	19.75	-2.39	-5.06	13.09
31	-118.00	19.97	-2.40	-5.75	13.51
32	-129.85	20.20	-2.39	-6.51	13.93
33	-142.77	20.45	-2.39	-7.37	14.37
34	-156.92	20.69	-2.38	-8.32	14.81
35	-172.44	20.92	-2.36	-9.39	15.26
36	-189.57	21.19	-2.34	-10.59	15.75
37	-208.54	21.48	-2.33	-11.95	16.24
38	-229.61	21.79	-2.30	-13.49	16.75
39	-253.16	22.12	-2.27	-15.22	17.28
40	-279.59	22.46	-2.25	-17.21	17.83
41	-309.38	22.78	-2.22	-19.50	18.42
42	-343.28	23.11	-2.19	-22.14	19.03
43	-382.40	23.50	-2.14	-25.27	19.66
44	-428.09	23.86	-2.07	-29.02	20.31
45	-482.63	24.19	-2.00	-33.71	21.08
46	-549.61	24.64	-1.94	-39.62	21.87
47	-634.42	24.96	-1.90	-47.44	22.69
48	-748.92	25.14	-1.89	-58.75	23.62
49	-914.23	26.22	-1.90	-76.49	24.64
50	-62298.30	71.28	-3.43	-1234.2	67.08

Table B.5: ELASSO evaluation results.

	Unrestricted			Nonnegatitivty rest	
λ index	R^2_{OoS}	p-value	CER	R^2_{OoS}	p-value
1	0.98	6.67	1.97	1.09	5.49
2	0.97	6.68	1.97	1.09	5.51
3	0.97	6.69	1.96	1.09	5.52
4	0.97	6.70	1.96	1.08	5.53
5	0.97	6.70	1.96	1.08	5.54
6	0.96	6.71	1.95	1.08	5.55
7	0.96	6.72	1.95	1.07	5.57
8	0.96	6.73	1.94	1.07	5.58
9	0.95	6.74	1.94	1.07	5.59
10	0.95	6.74	1.94	1.06	5.61
11	0.95	6.75	1.93	1.06	5.62
12	0.95	6.76	1.93	1.05	5.63
13	0.94	6.77	1.92	1.05	5.65
14	0.94	6.78	1.92	1.05	5.67
15	0.93	6.79	1.91	1.04	5.68
16	0.93	6.80	1.91	1.04	5.70
17	0.93	6.81	1.90	1.03	5.71
18	0.92	6.83	1.90	1.03	5.73
19	0.92	6.84	1.89	1.03	5.75
20	0.91	6.85	1.89	1.02	5.77
21	0.91	6.86	1.88	1.02	5.79
22	0.90	6.88	1.87	1.01	5.81
23	0.90	6.89	1.87	1.00	5.83
24	0.89	6.91	1.86	1.00	5.85
25	0.89	6.92	1.85	0.99	5.88
26	0.88	6.94	1.85	0.99	5.90
27	0.88	6.96	1.84	0.98	5.93
28	0.87	6.97	1.83	0.97	5.96
29	0.86	6.99	1.82	0.96	5.99
30	0.86	7.01	1.81	0.96	6.02
31	0.85	7.03	1.80	0.95	6.05
32	0.84	7.06	1.79	0.94	6.09
33	0.83	7.08	1.78	0.93	6.13
34	0.82	7.11	1.77	0.92	6.17
35	0.81	7.14	1.75	0.91	6.22
36	0.80	7.17	1.74	0.89	6.27
37	0.78	7.20	1.72	0.88	6.32
38	0.77	7.24	1.70	0.87	6.38
39	0.75	7.28	1.68	0.85	6.44
40	0.74	7.33	1.66	0.83	6.51
41	0.72	7.38	1.63	0.81	6.60
42	0.69	7.45	1.61	0.79	6.69
43	0.66	7.52	1.57	0.76	6.81
44	0.63	7.60	1.53	0.73	6.94
45	0.59	7.71	1.48	0.69	7.10
46	0.53	7.84	1.42	0.64	7.30
47	0.46	8.03	1.33	0.57	7.57
48	0.34	8.31	1.19	0.46	8.01
49	0.09	8.86	0.92	0.29	8.65
50	-61622.64	71.46	-3.58	-1259.83	67.93

Table B.6: ERidge evaluation results.

	Unrestricted			Nonnegatitivty rest	
λ index	R^2_{OoS}	p-value	CER	R^2_{OoS}	p-value
1	0.00	50.05	0	0.00	50.05
2	-0.13	38.70	3.63	-0.07	36.97
3	-0.61	32.86	4.72	0.05	22.19
4	-1.68	25.03	4.75	0.13	9.01
5	-3.31	17.01	5.07	-0.52	3.97
6	-4.96	10.66	5.43	-1.71	2.52
7	-7.55	9.06	5.71	-3.73	3.10
8	-10.14	7.96	5.94	-5.77	3.79
9	-12.12	7.15	6.18	-6.86	3.92
10	-13.89	6.52	6.32	-7.65	3.71
11	-15.10	5.50	6.40	-7.90	2.87
12	-15.93	4.96	6.50	-7.89	2.41
13	-16.49	4.90	6.59	-7.70	2.22
14	-17.25	5.11	6.53	-7.74	2.29
15	-17.94	5.28	6.45	-7.87	2.43
16	-18.59	5.59	6.39	-8.03	2.62
17	-19.13	5.93	6.35	-8.13	2.81
18	-19.72	6.29	6.32	-8.36	3.01
19	-20.30	6.57	6.28	-8.63	3.19
20	-20.83	6.81	6.25	-8.84	3.31
21	-21.31	7.04	6.22	-9.00	3.40
22	-21.72	7.20	6.20	-9.12	3.46
23	-22.04	7.34	6.18	-9.23	3.54
24	-22.30	7.45	6.16	-9.33	3.63
25	-22.51	7.54	6.14	-9.41	3.70
26	-22.67	7.61	6.13	-9.48	3.75
27	-22.80	7.67	6.12	-9.53	3.80
28	-22.90	7.70	6.11	-9.57	3.83
29	-22.98	7.72	6.11	-9.59	3.84
30	-23.04	7.75	6.10	-9.61	3.86
31	-23.09	7.76	6.10	-9.63	3.87
32	-23.13	7.78	6.10	-9.64	3.88
33	-23.17	7.80	6.09	-9.65	3.89
34	-23.19	7.81	6.09	-9.66	3.89
35	-23.21	7.82	6.09	-9.66	3.90
36	-23.23	7.83	6.08	-9.67	3.90
37	-23.24	7.83	6.08	-9.67	3.90
38	-23.25	7.84	6.08	-9.67	3.91
39	-23.26	7.84	6.08	-9.67	3.91
40	-23.29	7.86	6.08	-9.68	3.92

Table B.7: LASSO evaluation results.

	Unrestricted			Nonnegatitivty rest	
λ index	R^2_{OoS}	p-value	CER	R^2_{OoS}	p-value
1	0.15	0.39	0	0.15	0.39
2	0.19	0.39	3.63	0.19	0.39
3	0.24	0.39	4.72	0.24	0.39
4	0.30	0.40	4.75	0.30	0.40
5	0.37	0.40	5.07	0.37	0.40
6	0.47	0.41	5.43	0.47	0.41
7	0.58	0.42	5.71	0.58	0.42
8	0.71	0.43	5.94	0.71	0.43
9	0.88	0.44	6.18	0.88	0.44
10	1.06	0.46	6.32	1.06	0.46
11	1.27	0.48	6.40	1.27	0.48
12	1.51	0.51	6.50	1.51	0.51
13	1.75	0.54	6.59	1.75	0.54
14	2.00	0.59	6.53	2.00	0.59
15	2.22	0.64	6.45	2.31	0.49
16	2.41	0.72	6.39	2.60	0.43
17	2.52	0.80	6.35	2.85	0.39
18	2.53	0.91	6.32	2.97	0.38
19	2.41	1.04	6.28	2.97	0.34
20	2.12	1.20	6.25	2.90	0.32
21	1.65	1.37	6.22	2.75	0.32
22	0.95	1.56	6.20	2.46	0.33
23	0.04	1.77	6.18	1.95	0.38
24	-1.08	1.98	6.16	1.27	0.45
25	-2.41	2.20	6.14	0.44	0.55
26	-3.88	2.42	6.13	-0.47	0.66
27	-5.47	2.62	6.12	-1.47	0.79
28	-7.11	2.83	6.11	-2.46	0.93
29	-8.75	3.02	6.11	-3.41	1.06
30	-10.34	3.22	6.10	-4.24	1.16
31	-11.83	3.41	6.10	-4.97	1.26
32	-13.21	3.60	6.10	-5.62	1.37
33	-14.45	3.81	6.09	-6.17	1.50
34	-15.57	4.02	6.09	-6.65	1.64
35	-16.56	4.25	6.09	-7.05	1.79
36	-17.45	4.50	6.08	-7.39	1.94
37	-18.23	4.76	6.08	-7.67	2.10
38	-18.92	5.02	6.08	-7.90	2.24
39	-19.54	5.30	6.08	-8.12	2.40
40	-23.26	7.76	6.08	-9.64	3.87

Table B.8: Ridge evaluation results.

Magyar nyelvű összefoglaló

A pénzügyi ökonometria területén az elmúlt években egyre nagyobb népszerűséggel bírnak a gépi tanulási módszerek az empirikus alkalmazásokban. Így például a részvényportfoliók keresztmetszetét Gu, Kelly és Xiu (2020), kötvényportfoliók keresztmetszetét Bianchi, Buchner és Tamoni (2021), valamint Hollstein és Prokopczuk (2022) faktorportfóliók hozamának előrejelezhetőségét vizsgálja gépi tanulási módszerek alkalmazásával. Emellett a gépi tanulási módszereket a részvények kockázati prémiumának idősoros előrejelzésére is használták az elmúlt években (Rapach, Strauss és Zhou, 2013, Chinco, Clark-Joseph és Ye, 2019, Freyberger, Neuhierl és Weber, 2020, Kozak, Nagel és Santosh, 2020). Ezen tanulmányok mindegyikben szerepel, és általánosságban a lineáris modellek körén belül a legnépszerűbbnek bizonyul a LASSO.

A dolgozatom ehhez az irodalomhoz több ponton is hozzájárul. A kiindulási pontja az empirikus pénzügyi alkalmazásokban 'előrejelzések kombinálása' (forecast combination) névvel illetett módszer. Ennek lényege, hogy ha egy adott változót több változóval szeretnénk előrejelezni, akkor külön-külön mindegyik változóval becsülünk egy egyváltozós lineáris regressziós modellt (amelynek paramétereit OLS-sel, azaz a becsült értékek mintán való négyzetes hibájának minimalizálásával kapunk meg), majd az ezen modellekből kapott előrejelzések átlagát vesszük, mint a 'végső' előrejelzésünket. Ez a módszer lényegében egy a regressziós együtthatók 0 felé való regularizásával ekvivalens (Rapach, Strauss és Zhou, 2010), és jellemzően zajos idősorok és sok, egymással korreláló prediktor esetén teljesít jól. A dolgozatomban azt állítom, hogy ennek a módszernek bizonyos esetekben túl nagy a torzítása a(z egyébként viszonylag kicsi) varianciájához képest, és ezért nem teljesít jól. Ellentétben a hagyományos regularizációs módszerekkel, mint például a korábban említett, nagyon népszerű LASSO vagy ridge regresszió, az 'előrejelzések kombinálása' módszernek nincsen egy további hiperparamétere, amellyel képes lenne a 'torzítás-variancia átváltást' (bias-variance trade-off) optimalizálni. Ezért olyan módszereket vizsgálok, amelyek a) az 'előrejelzések kombinálása' módszer általánosításai, b) rendelkeznek egy hiperparaméterrel, amellyel képesek a torzítás-variancia átváltást optimalizálni, és c) az 'előrejelzések kombinálása' módszert a legnagyobb torzítású és legkisebb varianciájú speciális esetként adják vissza. Az ilyen módszereket az 'előrejelzések kombinálásának inverz regularizációja' névvel illetem, és a címben is erre utalok. Ezeket az inverz reguralizációs (IR) módszereket két nagy csoportba kategorizáltam. Első szakaszosnak ('stage I') nevezem azokat a módszereket, amelyek az 'előrejelzések kombinálásán' az 'első szakaszban', azaz az egyedi modelleken keresztül (például másféle becslése a modelleknek, vagy másféle struktúra) változtatnak. Második szakaszosnak ('stage II') nevezem azokat a módszereket, amelyek az előrejelzések kombinálásán úgy változtatnak, hogy továbbra is az egyváltozós, OLS-sel becsült regressziók becsült értékeit átlagoljuk, azonban nem egyenlő, hanem attól eltérő súlyokkal.

Az első nagy hozzájárulásom az irodalomhoz az előrejelzések kombinálásának inverz reguralizációja, mint fogalomrendszer bevezetése, és az irodalomban már meglévő egyes módszerek besorolása a fent említett kategóriákba. A második nagy hozzájárulásom, hogy szimulált adatsorokon összehasonlítok bizonyos első szakaszos ('complete subset regression (CSR)' nevű módszer, Elliot és Timmermann (2013)-ből, illetve egy saját módszer, az 'uNCL'), második szakaszos (ELASSO és ERidge, Diebold és Shin (2019)-ből), és 'hagyományos' regularizációs módszereket (LASSO és ridge regressziók). A szimuláció eredményei azt mutatják, hogy a) csak az első szakaszos IR módszerek képesek javítani a 'előrejelzések kombinálása' módszer előrejelzési hibáját és b) a legjobban teljesítő két módszer a vizsgált esetekben jellemzően a két első szakaszos IR módszer, a CSR és az uNCL, megelőzve az empirikus applikációkban nagyon népszerű LASSO-t és ridge regressziót. Az eredményeket először az adott módszerek hiperparamétereinek optimális megválasztása esetén vizsgálom, majd azt is megmutatom, hogy a módszereket pontosságának sorrendjén az sem változtat, ha a hiperparaméterek értékét az adatokon való kvázi 'múltbeli' teljesítményük alapján választom meg.

A harmadik nagy hozzájárulásom az irodalom az, hogy kidolgozok egy új előrejelzési módszert, amelynek az 'egyváltozós negatív korrelációs tanulás ('univariate negative correlation learning', uNCL) nevet adom. A módszer egy, a gépi tanulási irodalomban neurális hálókból álló kombinált modellek ('ensemble'-k) tanítására használt algoritmus, a 'negatív korrelációs tanítás' ('negative correlation learning', NCL) újszerű alkalmazása. Ennek a módszernek a lényege, hogy az egyedi neurális hálók paramétereit nem úgy választja meg, hogy külön-külön minimalizálja a hálók előrejelzésének négyzetes hibáját, hanem figyelembe veszi az egyes hálók közötti kapcsolatot is. Így olyan hálókat tanít, amelyek külön-külön pontatlanok, de kombinált modellként pontosabbak, mintha szimplán a négyzetes hiba minimalizálásával tanítottuk volna az egyes modelleket. A dolgozatomban ezt az eljárást alkalmaztam az előrejelzések kombinálása módosítására azáltal, hogy az egyváltozós regressziók paramétereit nem OLS-sel, hanem az NCL algoritmussal becsültem meg. A szimuláció eredményei alapján az így kapott módszer jellemzően a két legjobban teljesítő módszer valamelyike. Emellett azt is megmutatom, hogy az uNCL jelentősen különbözőbben 'viselkedik', mint az NCL viselkedett a korábbi, neurális hálókon való alkalmazásokban. Míg a neurális hálók esetében a módszer az egyes hálók pontosságát feláldozva az egyes hálók kovarianciájának ('diverzitásának') csökkentésével egy úgynevezett ’pontosság-diverzitás átváltás’ optimalizálásával ért el jobb eredményeket, addig az uNCL esetében az egyedi modellek előrejelzéseinek kovarianciája közel konstans, és ehelyett a torzítás-variancia átváltás optimalizálásával javít az előrejelzések kombinálása teljesítményén.

A szimulációs elemzés mellett a dolgozat tartalmaz egy részletes empirikus applikációt is. Ebben a szimulációban is használt módszerek teljesítményét hasonlítom össze egy jól ismert és praktikus szempontból is releváns adatsoron, az Egyesült Államok részvénypiaci kockázati prémiumának előrejelzésén. Az alkalmazás eredményei validálják a szimulációs eredményeket; a módszerek sorrendje és a fő megállapítások változatlanok a szimulációhoz képest. A legjobban teljesítő módszer az általam kidolgozott uNCL, a második legjobban teljesítő módszer pedig a másik első szakaszos IR módszer, a CSR. Az eredmények robusztusak a hiperparaméterek értékeinek múltbeli teljesítményen alapuló megválasztására, a kiértékelési időszakra és a negatív előrejelzések lehetőségének kizárására.

A dolgozat legfőbb tanulsága az, hogy bizonyos első szakaszos inverz regularizációs módszerek, így a CSR és az általam kifejlesztett uNCL jellemzően jobban teljesítenek, mint az pénzügyi előrejelzések empirikus irodalmában az elmúlt években legnépszerűbb lineáris módszer, a LASSO. Ezáltal az eredmények azt indikálják, hogy az említett módszereknek sokkal nagyobb szerepet kellene adni a jövőbeli empirikus alkalmazásokban. Emellett azt is mutatják az eredmények, hogy a LASSO egyedüli ’benchmark’-ként való szerepeltetése nem megfelelő gyakorlat annak gyakran szuboptimális teljesítménye miatt.

Szószedet

Angol	Magyar
accuracy-diversity trade-off	pontosság-diverzitás átváltás
bias-variance trade-off	torzítás-variancia átváltás
bias-variance-covariance trade-off	torzítás-variancia-kovariancia átváltás
certainty equivalent return (CER)	biztos hozam-egyenértékes
equity premium	részvénypiac kockázati prémiuma
complete subset regression	összes részhalmaz regresszió
inverse reguralisation	inverz regularizáció
forecast combination	előrejelzések kombinálása
data generating process (DGP)	adatgeneráló folyamat
machine learning	gépi tanulás
mean squared error (MSE)	átlagos négyzetes hiba
negative correlation learning	negatív korrelációs tanítás
neural network	neurális háló
neural network ensemble	kombinált neurális hálók
ordinary least squares (OLS)	legkisebb négyzetek módszere
shrinkage	(regressziós együtthatók) 'zsugorítása'
regularisation	regularizáció
univariate negative correlation learning	egyváltozós negatív korrelációs tanítás

Bibliography

Avramov, D., Cheng, S., and Metzker, L. (2021). Machine learning versus economic restrictions: Evidence from stock return predictability. *Available at SSRN.*

Baltas, N. and Karyampas, D. (2018). Forecasting the equity risk premium: The importance of regime-dependent evaluation. *Journal of Financial Markets*, 38, p. 83–102.

Bates, J. and Granger, C. (1969). The Combination of Forecasts. *Journal of the Operational Research Society*, 20(4), pp. 451–468.

Bianchi, D., Buchner, M., and Tamoni, A. (2021). Bond risk premiums with machine learning. *The Review of Financial Studies*, 34(2), p. 1046–1089.

Boot, T. and Nibbering, D. (2019). Forecasting using random subspace methods. *Journal of Econometrics*, 209(2), pp. 391–406.

Brown, G., Wyatt, J., Harris, R., et al. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1), p. 5–20.

Brown, G., Wyatt, J., and Tino, P. (2005). Managing Diversity in Regression Ensembles. *Journal of Machine Learning Research*, 6, p. 1621–1650.

Buschjager, S., Pfahler, L., and Morik, K. (2020). Generalized negative correlation learning for deep ensembling. *arXiv preprint arXiv:2011.02952.*

Campbell, J. and Thompson, S. (2007). Predicting Excess Stock Returns Out of Sample: Can Anything Beat the Historical Average? *The Review of Financial Studies*, 21(4), pp. 1509–1531.

Chauvet, M. and Potter, S. (2013). "Forecasting Output". In: *Handbook of Economic Forecasting.* Ed. by G. Elliott and A. Timmermann. Vol. 2. Handbook of Economic Forecasting. Elsevier, pp. 141–194.

Chinco, A., Clark-Joseph, A., and Ye, M. (2019). Sparse signals in the cross-section of returns. *The Journal of Finance*, 74(1), p. 449–492.

Claeskens, G. et al. (2016). The forecast combination puzzle: A simple theoretical explanation. *International Journal of Forecasting*, 32(3), p. 754–762.

Clark, T. and West, K. (2007). Approximately normal tests for equal predictive accuracy in nested models. *Journal of Econometrics*, 138(1), p. 291–311.

Cujean, J. and Hasler, M. (2017). Why does return predictability concentrate in bad times? *The Journal of Finance*, 72(6), p. 2717–2758.

Dai, Z. et al. (2020). Forecasting stock market returns: New technical indicators and two-step economic constraint method. *The North American Journal of Economics and Finance*, 53, p. 101216.

Diebold, F. and Shin, M. (2019). Machine learning for regularized survey forecast combination: Partially-egalitarian LASSO and its derivatives. *International Journal of Forecasting*, 35(4), p. 1679–1691.

Elliott, G., Gargano, A., and Timmermann, A. (2013). Complete subset regressions. *Journal of Econometrics*, 177(2), pp. 357–373.

— (2015). Complete subset regressions with large-dimensional sets of predictors. *Journal of Economic Dynamics and Control*, 54, pp. 86–110.

Farmer, L., Schmidt, L., and Timmermann, A. (2022). Pockets of predictability. *Journal of Finance, forthcoming.*

Freyberger, J., Neuhierl, A., and Weber, M. (2020). Dissecting characteristics nonparametrically. *The Review of Financial Studies*, 33(5), p. 2326–2377.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1), p. 1–22.

Genre, V. et al. (2013). Combining expert forecasts: Can anything beat the simple average? *International Journal of Forecasting*, 29(1), p. 108–121.

Giannone, D., Lenza, M., and Primiceri, G. (2021). Economic predictions with big data: The illusion of sparsity.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5), p. 2223–2273.

Haase, F. and Neuenkirch, M. (2022). Predictability of bull and bear markets: A new look at forecasting stock market regimes (and returns) in the US. *International Journal of Forecasting.*

Han, Y. et al. (2020). Firm characteristics and expected stock returns. *Available at SSRN.*

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning.* Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Hoerl, Arthur E and Kennard, Robert W (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), p. 55–67.

Hollstein, F. and Prokopczuk, M. (2022). Managing the Market Portfolio. *Management Science, forthcoming.*

Hu, G. and Mao, Z. (2009). "Bagging ensemble of SVM based on negative correlation learning". In: *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems.* Vol. 1, p. 279–283.

Huang, D. et al. (2022). Scaled PCA: A new approach to dimension reduction. *Management Science*, 68(3), p. 1678–1695.

Kozak, S., Nagel, S., and Santosh, S. (2020). Shrinking the cross-section. *Journal of Financial Economics*, 135(2), p. 271–292.

Krogh, A. and Vedelsby, J. (1994). "Neural Network Ensembles, Cross Validation and Active Learning". In: NIPS'94. Denver, Colorado: MIT Press, pp. 231–238.

Li, J. and Tsiakas, I. (2017). Equity premium prediction: The role of economic and statistical constraints. *Journal of Financial Markets*, 36, pp. 56–75.

Liu, Y. and Yao, X. (1998). "Time Series Prediction by Using Negatively Correlated Neural Networks". In: SEAL'98. Berlin, Heidelberg: Springer-Verlag, pp. 333–340.

— (1999). Ensemble learning via negative correlation. *Neural Networks*, 12(10), pp. 1399–1404.

Liu, Y., Yao, X., and Higuchi, T. (2000). Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, 4(4), p. 380–387.

Liu, Y., Zhao, Q., and Pei, Y. (2014). "From low negative correlation learning to high negative correlation learning". In: *2014 International Joint Conference on Neural Networks (IJCNN)*, p. 171–174.

Pettenuzzo, D., Timmermann, A., and Valkanov, R. (2014). Forecasting stock returns under economic constraints. *Journal of Financial Economics*, 114(3), pp. 517–553.

Rapach, D. (2013). "Forecasting Stock Returns". In: *Handbook of Economic Forecasting.* Ed. by G. Elliott and A. Timmermann. Vol. 2. Handbook of Economic Forecasting. Elsevier, pp. 328–383.

Rapach, D., Strauss, J., and Zhou, G. (2010). Out-of-Sample Equity Premium Prediction: Combination Forecasts and Links to the Real Economy. *Review of Financial Studies*, 23(2), pp. 821–862.

— (2013). International stock return predictability: What is the role of the United States? *The Journal of Finance*, 68(4), p. 1633–1662.

Rapach, D. and Zhou, G. (2020). Time-series and cross-sectional stock return forecasting: new machine learning methods. *Machine Learning for Asset Management: New Developments and Financial Applications*, p. 1–33.

Reeve, H. and Brown, G. (2018). Diversity and degrees of freedom in regression ensembles. *Neurocomputing*, 298, p. 55–68.

Sheng, W. et al. (2017). A niching evolutionary algorithm with adaptive negative correlation learning for neural network ensemble. *Neurocomputing*, 247, p. 173–182.

Shi, Z. et al. (2018). "Crowd Counting with Deep Negative Correlation Learning". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5382–5390.

Smith, J. and Wallis, K. (2009). A Simple Explanation of the Forecast Combination Puzzle. *Oxford Bulletin of Economics and Statistics*, 71(3), p. 331–355.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), p. 267–288.

Timmermann, A. (2006). "Chapter 4 Forecast Combinations". In: ed. by G. Elliott, C. Granger, and A. Timmermann. Vol. 1. Handbook of Economic Forecasting. Elsevier, p. 135–196.

Waleed, A. et al. (June 2009). "MLP, Gaussian Processes and Negative Correlation Learning for Time Series Prediction". In: pp. 428–437.

Wang, S., Chen, H., and Yao, X. (2010). "Negative correlation learning for classification ensembles". In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*, p. 1–8.

Wang, S., Tang, K., and Yao, X. (2009). "Diversity exploration and negative correlation learning on imbalanced data sets". In: *2009 International Joint Conference on Neural Networks*, p. 3259–3266.

Welch, I. and Goyal, A. (2007). A Comprehensive Look at The Empirical Performance of Equity Premium Prediction. *The Review of Financial Studies*, 21(4), pp. 1455–1508.

Zhang, Y., Ma, F., Shi, B., et al. (2018). Forecasting the prices of crude oil: An iterated combination approach. *Energy Economics*, 70, p. 472–483.

Zhang, Y., Ma, F., and Wang, Y. (2019). Forecasting crude oil prices with a large set of predictors: Can LASSO select powerful predictors? *Journal of Empirical Finance*, 54, pp. 97–117.

Zhang, Y., Ma, F., and Wei, Y. (2019). Out-of-sample prediction of the oil futures market volatility: A comparison of new and traditional combination approaches. *Energy Economics*, 81, p. 1109–1120.

Zhang, Y., W., and Wang, Y. (2022). Forecasting crude oil market volatility using variable selection and common factor. *International Journal of Forecasting.*

Zhang, Y. and Wang, Y. (2022). Forecasting crude oil futures market returns: A principal component analysis combination approach. *International Journal of Forecasting.*

Zhang, Y., Wei, Y., Ma, F., et al. (2019). Economic constraints and stock return predictability: A new approach. *International Review of Financial Analysis*, 63, pp. 1–9.

Zhang, Y., Wei, Y., Zhang, Y., et al. (2019). Forecasting oil price volatility: Forecast combination versus shrinkage method. *Energy Economics*, 80, p. 423–433.