

1 Minimum-Variance LASSO–Ridge Ensemble

1.1 Background

This documentation serves to describe the procedures implementing the Minimum-Variance Ridge-LASSO Ensemble (MVRLE) strategy. This strategy is programmed to algorithmically select among the high-dimensional set of stock features within the CTF dataset (Hellum et al., 2025). This data includes monthly stock returns, and 402 features for US stocks spanning the period from 1952 to 2023. The data set consists of a filtered subsample of the data from (Jensen et al., 2023).

1.2 Data Preparation

To prepare the data for the LASSO and Ridge implementations described in later sections, we transform each predictor using a cross-sectional percentile ranking procedure. Specifically, we employ the same transformation function as provided by Theis Ingerslev Jensen in the Common Task Framework repository on GitHub.¹

For each end-of-month date, t , all non-missing observations of a given feature are ranked within the cross-section of stocks. The percentile ranking, $x_{i,t}^{prp}$, is constructed as follows:

$$x_{i,t}^{prp} = \frac{\text{rank}(x_{i,t})}{N_t} \quad (1)$$

Where N_t denotes the number of non-missing observations in that cross-section, $x_{i,t}$ is the original feature value and $\text{rank}(x_{i,t})$ is its rank. Given a tie in cross-sectional ranking, the maximum ranking is employed.

If the original value of a given feature is exactly 0 before the percentile ranking, we manually set its percentile ranking to zero. Thus, these values constitute the minimum percentile ranking. This choice is made, because actual zeros in many instances are interpretable as special cases, or states of the world that are fundamentally different. Accordingly, original zero observations are assigned the lowest possible percentile rank for a given feature. The percentile ranking now spans the interval from 0 to 1. Then we subtract 0.5 from each percentile rank, resulting in the transformation being centered at 0. Lastly, missing values are imputed to be zero after the transformation, as this value corresponds to the cross-sectional median percentile.

1.3 Sample Split

The sample is divided into an out-of-sample test period and an in-sample estimation period using the indicator provided in the dataset. Observations with "ctff_test == TRUE" constitute the test sample, while observations with "ctff_test == FALSE" form the in-sample period.

Within the in-sample period, we further split the data into a training and validation subsample. We define this split to allocate 75% to the training set, whereas the remaining 25% is allocated to the

¹https://github.com/theisij/common-task-framework-SDF/blob/main/models_R/factor-ml/factor_ml.R

validation-set. Because returns are heavily correlated in the cross-section but not in the time series, the allocation of observations into each respective set is based on the end-of-month date. The validation period therefore consists of the most recent proportion of the in-sample time span.

1.4 Return Prediction

We model one-month-ahead excess returns, $r_{i,t+1}$, as a linear function of firm characteristics. This predictive regression can be iterated as follows:

$$r_{i,t+1} = X'_{i,t}\beta + \varepsilon_{i,t+1} \quad (2)$$

where $X_{i,t}$ denotes the transformed cross sectionally ranked characteristics observed on the end of month date t . Given the high dimensionality of the characteristics (402), we estimate the parameter vector β using penalized regression methods from the elastic net framework, implemented via the `glmnet` package in R. The estimator solves:

$$\arg \min_{\beta} \left\{ \frac{1}{N} \sum_{i,t} (r_{i,t+1} - X'_{i,t}\beta)^2 + \lambda \left(\alpha \|\beta\|_1 + \frac{1-\alpha}{2} \|\beta\|_2^2 \right) \right\}, \quad (3)$$

where the first term corresponds to the mean squared error (MSE), and the second term is a regularization penalty. The parameter $\lambda \geq 0$ governs the strength of the shrinkage, while $\alpha \in [0, 1]$ governs the relative weight placed on the ℓ_1 and ℓ_2 penalties. Setting $\alpha = 1$ results in the LASSO estimator, while setting $\alpha = 0$ results in the Ridge estimator. The parameter α , thus governs the tradeoff between sparsity and stability. Specifically, LASSO sets some coefficients exactly equal to zero and shrinks the remaining coefficients toward zero, thereby serving as a valuable selection tool. Ridge, by contrast, does not eliminate predictors, but instead shrinks all coefficients continuously toward zero, which reduces the variance of the estimated coefficients.

Consequently, Ridge tends to produce smoother coefficient paths by distributing weight across correlated predictors. While this can enhance stability, it may come at the cost of including less relevant variables, potentially introducing additional noise into the predictions. Conversely, LASSO may select one predictor among a group of highly correlated variables and set the coefficients of the others exactly equal to zero. Such selection can be unstable, making the resulting strategy more sensitive to sample variation or alternative training-validation splits.

1.4.1 Implementation Details

We estimate two separate predictive regressions corresponding to the LASSO and Ridge specifications. In doing so, we fix the mixing parameter at $\alpha = 1$ for LASSO and $\alpha = 0$ for Ridge, and estimate the models using the `glmnet` package in R. Instead of using cross-validation, we select the parameter λ using a simpler time-series validation procedure. For each model, we estimate the full regularization path on the previously defined training sample and compute predicted returns for the validation period. The value of λ that minimizes the mean squared prediction error (MSPE) on the validation sample is selected.

After selecting the optimal λ , the model is re-estimated using the full in-sample period consisting of the training and validation sets combined. This model is then used to generate return predictions for the out-of-sample period.

1.5 Long short strategy

Based on the out-of-sample model-implied return predictions, we construct separate long–short portfolios for the Ridge- and LASSO-based strategies. For each end-of-month cross-section, assets are sorted according to their predicted next-month excess returns. We take a long position in the top 18th percentile of predicted returns and a short position in the bottom 18th percentile. The 18th percentile cutoff is intentionally chosen to avoid conventional breakpoints such as deciles (10%), quintiles (20%), or terciles (33%). Apart from avoiding these standard thresholds, the exact cutoff is not theoretically motivated, and the results are not sensitive to small variations in this choice. Finally, within each leg of the portfolio, assets are equally weighted. Within each month, all stocks in the long portfolio receive equal positive weights, while all stocks in the short portfolio receive equal negative weights. The portfolio return in the subsequent month is computed as the cross-sectional sum of these weights multiplied by the realized next-month excess returns. Portfolios are rebalanced at the end of each month.

1.6 Ensemble Portfolio Construction

As the excess returns of the two portfolios are unlikely to be independent, we determine the combination weights that minimize the variance of the ensemble portfolio subject to the weights summing to one. We adopt this weighting scheme because mean–variance efficient portfolios constructed using sample estimates of expected returns are known to perform poorly out of sample, whereas global minimum-variance portfolios tend to achieve higher out-of-sample Sharpe ratios (Jagannathan & Ma, 2003).

This convex combination serves to balance the complementary properties of our Ridge- and LASSO-based portfolios. In particular, it seeks to combine the sparsity induced by LASSO with the variance-reducing properties of Ridge, thereby aiming to produce a more robust Sharpe ratio. Using the in-sample monthly portfolio returns, we estimate the covariance matrix of the Ridge and LASSO strategies. In the two-portfolio case, the minimum-variance weight on the LASSO portfolio is given by:

$$w_L = \frac{\sigma_R^2 - \sigma_{LR}}{\sigma_L^2 + \sigma_R^2 - 2\sigma_{LR}} \quad (4)$$

with the Ridge weight defined as $w_R = 1 - w_L$. These weights are estimated using in-sample data only and are subsequently held fixed throughout the out-of-sample test period. In the algorithmic implementation, an additional safeguard imposes equal weights in the event of a near-zero or non-finite denominator. To date, this contingency has not been triggered. The ensemble portfolio excess return, $r_{e,t}$, is defined as the weighted average of the Ridge and LASSO portfolio returns, denoted by r_R and r_L , respectively:

$$r_e = w_L r_L + w_R r_R \quad (5)$$

References

- Hellum, O., Jensen, T. I., Kelly, B. T., & Pedersen, L. H. (2025). The power of the common task framework [Working Paper].
(Available at <https://ssrn.com/abstract=5242901>) doi: 10.2139/ssrn.5242901
- Jagannathan, R., & Ma, T. (2003). Risk reduction in large portfolios: Why imposing the wrong constraints helps. *Journal of Finance*, 58(4), 1651–1683. Retrieved from <https://doi.org/10.1111/1540-6261.00580> doi: 10.1111/1540-6261.00580
- Jensen, T. I., Kelly, B. T., & Pedersen, L. H. (2023). Is there a replication crisis in finance? *Journal of Finance*, 78(5), 2465–2518. Retrieved from <https://doi.org/10.1111/jofi.13249> doi: 10.1111/jofi.13249