LASSO Ensembles

BACHELOR THESIS

written by

Alexander Glötzl

University of

Regensburg

supervised by

Prof. Dr. Rainer Spang

22nd August, 2024

Preamble

Abstract

In bioinformatics, datasets often have a high dimensionality, necessitating feature reduction to focus on the most relevant variables. Regularized regression methods, such as LASSO, are commonly used to address this issue by simplifying models and reducing the risk of overfitting through regularization. However, this thesis demonstrates that, with a high number of features or a low signal-to-noise ratio, regularized regression can struggle to identify the true underlying patterns in the data. For this purpose, the LASSO training process is examined in detail, highlighting the difficulties in achieving good test performance in high-dimensional settings. The thesis then introduces ensemble learning for LASSO models and compares its performance to that of individual LASSO models. Extensive testing is conducted on both synthetic and biological LAMIS datasets. The findings show that LASSO ensembles do not outperform single LASSO models in these settings. This is because, in high-dimensional cases, individual LASSO models tend to favor false positive features that offer better training performance, overshadowing the true model coefficients.

Required prior knowledge

The reader should be familiar with basic machine learning concepts and terminology, including:

- Samples and features: Understanding that data points are samples with multiple features.
- Overfitting and underfitting: Concepts describing model performance in relation to training and generalization.
- Regularized linear regression: Techniques like LASSO (Least Absolute Shrinkage and Selection Operator), which add regularization to linear regression models.
- Mean Squared Error (MSE): A performance metric used to evaluate the accuracy of regression models.

The figures and the code for the package *cv.glmnets* are implemented in R and can be accessed in the GitLab repositories [2] and [3]. While this code provides additional detail for those interested, it is not necessary for understanding the main content of the thesis. The final chapter presents probabilistic estimates using hypergeometric distributions and binomial coefficients, which require a basic understanding of probability theory.

Contents

1. Introduction

Typical bioinformatics datasets often contain thousands of features, which can lead to data sparsity in high dimensions. This makes it challenging for models to make accurate predictions, as the number of samples may be insufficient to effectively sample the feature space. Regularized regression, such as LASSO (Least Absolute Shrinkage and Selection Operator), is commonly used to address this issue by penalizing models with too many coefficients. This helps to simplify models and reduce overfitting.

This thesis will show that, when dealing with very high-dimensional data and a low signal-to-noise ratio, the LASSO algorithm may struggle to identify the correct underlying patterns in the dataset. Therefore, in this thesis the function *cv.glmnet* of the R package GlmNet for regularized regression is studied in detail and results will be presented, that illustrate the challenge of estimating the correct model coefficients and follow the training procedure of the function step by step. After understanding the problem, inspired by the paper "Classification of high dimensional data using LASSO ensembles" of Urda et. al. [7] the question is asked, can LASSO ensembles improve predictive performance for high-dimensional data with high noise? The paper uses different combine strategies for LASSO ensembles and shows a statistically significant improvement over base models. Even though the benefits of ensemble learning over a single machine learning model is well documented, besides the mentioned paper from 2017 there is no detailed analysis of ensemble learning for linear regression models available in literature yet.

Therefore a new package *cv.glmnets* is introduced, which combines multiple LASSO base models to build an ensemble, where each base model has only a subset of all features available. In this thesis the ensemble algorithm is tested extensively for different combine strategies, true β signatures, number of available features and datasets, i.e. both a synthetic and biological LAMIS dataset. It will be shown that LASSO ensembles cannot outperform the classic LASSO model significantly on the tested datasets since for a high number of available features the base model is likely to find and prefer false positive features that have a better training performance than real model coefficients. Thus this thesis shows why single LASSO model estimates break down with increasing noise and dimensionality and that LASSO ensembles are not suitable to improve upon this challenge.

2. Methods

2.1 Linear Regression

Simple linear unregularized regression describes the relation between target variable y and the predictor variables X.

$$y \approx \hat{y} = \beta_0 + \sum_{j=1}^{p} \beta_j X \tag{2.1}$$

where $\hat{y}$ is the predicted value of y, β_0 is the intercept, β_j are the estimated model coefficients for the p features, and X is the data matrix with dimensions $n \times p$, where n is the number of samples.

The algorithm tries to minimize the difference between predicted $\hat{y}$ and the target y by solving the cost function

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y_i})^2 \tag{2.2}$$

where MSE is an abbreviation for Mean Squared Error. In the simple unregularized case, the cost function for linear regression is convex and even has an analytical solution. Since this thesis focuses solely on linear regression, logistic regression and its cost function, which are used for categorical classification, are not considered.

2.2 LASSO

The solution to the above equation will generally include all features. However, for high dimensional datasets with up to 10,000 features, feature selection is necessary to prevent overfitting and ensure a more comprehensive model. To address this, the LASSO (Least Absolute Shrinkage and Selection Operator) algorithm introduces a regularization term to the cost function J

$$J(\beta, \lambda) = \text{MSE} + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.3}$$

where λ is the regularization parameter ranging from 0 to $+\infty$. λ defines how much the model will be regularized, i.e. restricted in the feature space. For example $\lambda = \infty$ would return a maximum restricted model with only the intercept coefficient remaining since β_0 is excluded from equation 2.3. On the other hand, $\lambda = 0$ represents a model, where there is no regularization at all and which typically results in an overfitted model. The additional term $\lambda \sum_{j=1}^{p} |\beta_j|$ in equation 2.3 is often called L1 regularization.

The regularization term can be understood as a constraint on the magnitude of β values. Specifically, in the 2 dimensional feature space this constraint resembles a diamond shape around the origin point (see figure 2.1) described by $|\beta_1| + |\beta_2| \leq$ const. The bigger the λ value, the smaller the diamond shape gets and the more solutions there are in the vicinity of the optimal $\hat{\beta}$ solution (see cone around $\hat{\beta}$ in figure 2.1). At a certain λ, the cone will intersect the diamond for the first time, determining the solution to the regularized regression equation, see 2.3. In higher dimensions, the diamond is more likely to be intersected at one of its corners than along one of its edges. Intersecting at a corner corresponds to having exactly one model coefficient non-zero, with the rest being zero. Therefore, when solving the regularized linear regression, depending on how many corners and edges are intersected, multiple β_j values will be set exactly to 0. Therefore, as λ increases, LASSO effectively performs continuous subset selection [4].

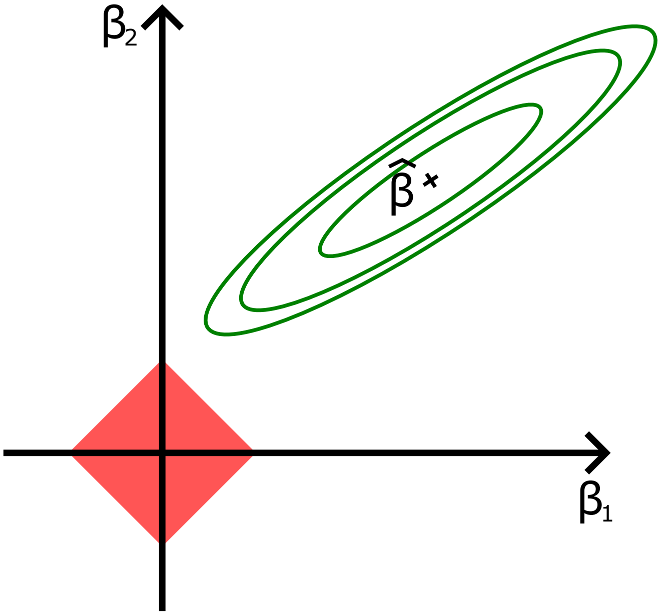


Figure 2.1: Figure shows the feature space of the LASSO model in 2D. The magnitude of model coefficients β_1 and β_2 are restricted by the regularization term in equation 2.3. This constraint resembles a diamond shape $|\beta_1| + |\beta_2| \leq$ const (red). Increasing the λ value, increases the number of possible β combinations around the unregularized solution $\hat{\beta}$, that all solve the model equally good (green ellipsoidal lines). By varying λ at some point these lines will intersect with the outer diamond, which returns a solution for the LASSO model under the regularization constraint. The drawing is inspired by figure in [4], page 244.

2.3 Best Subset Selection

Another seemingly simple method for feature selection would be to try out every combination of t features, where t is the assumed number of true model coefficients, which is chosen by the user. "Best Subset Selection" library *BeSS* by R. R. Hocking and R. N. Leslie [5] tries out every combination of t features out of n total features, e.g. if $t = 2$ and $n = 128$ then the possible combinations are $X1$-$X2$, $X1$-$X3$, …, $X1$-$X128$, $X2$-$X3$, …, $X127$-$X128$. The best performing subset of features on the training set is picked by the final model. For 2 feature combinations out of 128 features this accumulates to $\binom{128}{2} = \frac{128 \cdot 127}{2} = 8128$ combinations. For larger number of available features and more than 2 assumed real model coefficients this number would soon become too large to try out every combination, therefore the *BeSS.one* algorithm uses heavy internal optimization procedures to achieve a fast running time.

2.4 pBeta

The motivation for this thesis was that my supervisor, Tobias Schmidt, has observed that for large number of features the R-package *cv.glmnet* fails to find the correct underlying true model coefficient signature β_{true} for high dimensional, noisy data (see chapter 3). To quantify how well a model finds the true model coefficients β_{true}, the metric *pBeta* is introduced.

$$p_\beta = \frac{\|\tilde{\beta}\|}{\|\hat{\beta}\|} \tag{2.4}$$

The variable *pBeta* describes the ratio of $\|\tilde{\beta}\|$ to $\|\hat{\beta}\|$, where $\hat{\beta}$ denotes the vector of "non-zero model coefficient estimates" and $\tilde{\beta}$ denotes the vector of "true non-zero model coefficient estimates". "True" in this context means, that the corresponding feature is also part of non-zero β_{true}. The β coefficients are estimates from the model at λ_{min}. The higher *pBeta*, the higher the mass of true model coefficients is in the total amount of estimated model coefficient signature. Following is an example on how to calculate *pBeta* with two non-zero entries in β_{true}

$$\begin{aligned}
&\text{true } \beta & \beta_{true} &= \begin{pmatrix} 2 \\ -1 \\ 0 \\ 0 \end{pmatrix} && \text{(2.5a)} \\
&\text{indices}_{\beta_{true,non\text{-}zero}} & &= [1, 2] && \text{(2.5b)} \\
&\beta \text{ estimate} & \hat{\beta} &= \begin{pmatrix} 2.2 \\ -1.4 \\ -0.4 \\ 0.2 \end{pmatrix} && \text{(2.5c)} \\
&\text{true } \beta \text{ estimate} & \tilde{\beta} &= \begin{pmatrix} 2.2 \\ -1.4 \end{pmatrix} && \text{(2.5d)}
\end{aligned}$$

With *pBeta* defined as in equation 2.4 the example 2.5 has a *pBeta* value of

$$\text{pBeta} = \frac{\|\tilde{\beta}\|}{\|\hat{\beta}\|} = \frac{|2.2| + |-1.4|}{|2.2| + |-1.4| + |-0.4| + |0.2|} \approx 0.86 \qquad (2.6)$$

where the Manhatten norm is used. For high number of features *pBeta* becomes close to zero as will be shown in chapter 4.1. This phenomenon is titled "Signal Drowning".

2.5 GlmNet

Unlike unregularized regression the final model heavily depends on the selected λ value. The R-package *cv.glmnet* by Jerome Friedman, Trevor Hastie et al.[1] determines the best λ value by doing a 5-fold cross-validation on the training data. The λ value that performs best on average on all 5 test folds is called λ_{min}. Taking the standard deviation of the 5 individual best λ values on the 5 folds and then adding it to the λ_{min} value results in the so called λ_{1se} value. This value is generally considered to give better model interpretation than λ_{min} because it has fewer elements of β, that are non-zero.

λ goes through a range of values. Starting with the highest value, the algorithms picks its first feature, which has the highest correlation with y. Then continuously more features are added to an active set in a way that minimizes the remaining error between y and the already active set of features best (*Least Angle Regression* chapter in [4]). For every λ step the performance is measured on the test fold, therefore negating overfitting for a model with too many features. If λ reaches zero, every feature will be added to the model and LASSO returns the least squares fit.

2.6 GlmNets

In chapter 4.1, it will be demonstrated that the "Best Subset Selection" algorithm can find model coefficients that are closer to β_{true} than the estimates of *cv.glmnet.* For this purpose, the package *cv.glmnets* (plural) was developed, which trains 100 LASSO base models, each on a randomly selected subset of features, and combines them into an ensemble. Four types of ensembles are created: max-, mean-, top10-, and weighted-ensemble. These ensembles differ in how they combine the top base models: max-ensemble uses only the single best model, mean-ensemble averages all 100 models, top10-ensemble averages the top 10 models, and weighted-ensemble combines all 100 models, giving more weight to those with lower CVM error. The following equation 2.7 shows how the weighted-ensemble is calculated.

$$
\begin{aligned}
&\text{weighted-ensemble}(\lambda_i) &&= \frac{1}{\text{norm}} \cdot \left(w_{\text{bm }1}\beta_{\text{bm }1(\lambda_i)} + w_{\text{bm }2}\beta_{\text{bm }2(\lambda_i)} + \ldots + + w_{\text{bm }100}\beta_{\text{bm }100(\lambda_i)}\right) \\
&\text{with} \qquad w_{\text{bm }1} &&= 1/\text{MSE}_{\text{bm }1}(\lambda_i) \\
&\text{and} \qquad \text{norm} &&= w_{\text{bm }1} + w_{\text{bm }2} + \ldots + w_{\text{bm }100}
\end{aligned}
\tag{2.7}
$$

Figure 2.2 shows the process of selecting the best models from 100 base models. The *cv.glmnets* package generates these models by running *cv.glmnet* 100 times with different feature subsets. Each model's cross-validation mean error (CVM) is calculated using fixed fold indices to ensure comparability, i.e. a model that is only good because it has by chance a fold that results in a low error is avoided this way. Every model is forced to have 100 λ values by overriding the stopping criteria of *cv.glmnet.* Therefore ensuring, that the base models can be combined to ensembles along the same λ indices. The CVM results for 100 λ values in addition to λ_{min} and λ_{1se} are stored in a matrix, which is then sorted row-wise to identify the models with the lowest CVM errors. The ensembles are built by averaging their base model predictions. For weighted-ensemble a weight is added to every individual base model according to its CVM error (see equation 2.7). Since the λ values are not fixed for the base models during training, the λ index, i.e. 1 to 100 with λ_{min} and λ_{1se} being index 101 and 102 respectively, is used to combine the base models instead. The ensemble with the lowest MSE across all λ values is selected as the *best* ensemble. The corresponding λ value is referred to as λ_{best}. Following this procedure in figure 2.2 one of the lowest two λ values (the bottom two rows) in the right-hand side matrix would represent λ_{best}, having the overall lowest CVM error. In the code snippet 1 *cv.glmnets* is written out in pseudo code. As a side note, each base model uses the same feature subset across all 10 folds in cross-validation. If different subsets were used for each fold, it could result in a mix of "good" and "bad" predictions, leading to a poorer overall model with many non-zero model coefficients in the β vector. By fixing the feature subset, a model that happens to include the β_{true} vector is likely to have a significantly lower MSE than the others, which can improve the performance of the max-ensemble.

[1] Jerome Friedman, Trevor Hastie, Rob Tibshirani, Balasubramanian Narasimhan, Kenneth Tay, Noah Simon, Junyang Qian and James Yang

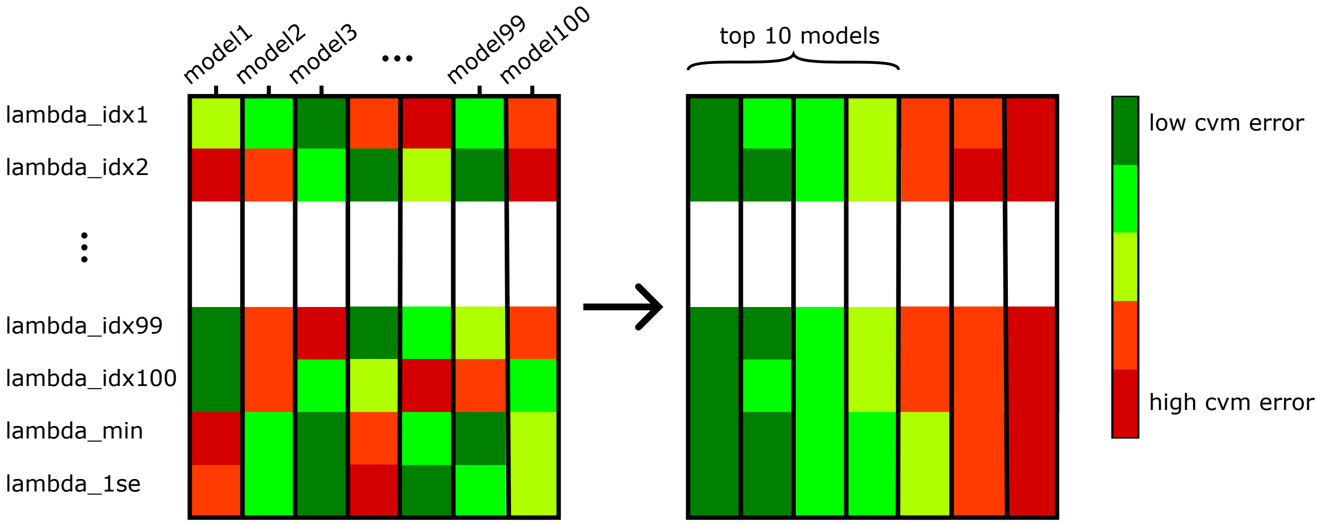


Figure 2.2: The 2D cross-validation mean error (CVM) data matrix illustrates the selection process for base models in the ensembles. The matrix has 100 columns representing all base models, and 102 rows corresponding to 100 λ values, along with λ_{min} and λ_{1se}, whose indices may vary in position for each base model. The right-hand side of the matrix is ordered row-wise according to the lowest CVM error for each base model.

```
GlmNets:
    number_base_models <- 100
    number_lambdas <- 100
    # foldid is fixed
    foldid <- [1,2,...,10,1,2,...,10]
    cvm <- array(0, dim=(number_lambdas + 2, number_base_models))
    # For each i from 1 to number_models
    for i = 1 to number_models :
        bm_i <- cv.glmnets(X[, feature_subset_i], y, foldid)
        cvm_error[all_lambda_values, i] <- [cvm_error_bm_i (i=1..100),
          cvm_error_bm_i[lambda.min], cvm_error_bm_i[lambda.1se]]
    end for
    # For each lambda value, take the top n base models with the lowest CVM error
    # and build an ensemble with them according to the ordered CVM matrix.
    # Order along the models/rows so smallest CVM error is on the left side of
    # (the matrix see figure 2.2)
    cvm_ordered <- sort(cvm)
    for combine_strategy in [mean-, max-, top10-, weighted-ensemble] :
        top_bm <- [100 if (mean or weighted), 1 if max, 10 if top10]
        # sum() is applied row-wise on cvm_ordered
        lambda_best_idx <- index(min(sum(cvm_ordered[:, 1:top_bm]))
        ensemble_i <- (bm_33[lambda_best_idx] + bm_10[lambda_best_idx] + ...+
          bm_top_bm[lambda_best_idx]) / top_bm
    end for

return [mean-ensemble, max-ensemble, weighted-ensemble, top10-ensemble]
```

Algorithm 1: Pseudo code for *cv.glmnets*.

2.7 Subfeature Coverage

When running the *cv.glmnets* function, 100 base models are typically trained using a random subset of 20% of the features. These hyperparameters are not selected deterministically, so it's useful to assess how well these subsets cover the relevant features. This section presents a probability-based analysis to evaluate the sufficiency of subfeature coverage with X base models.

Before using *cv.glmnets*, users often estimate the number of relevant features in their dataset. For instance, in a biological dataset with over 10,000 features, past experience might suggest that at most t features are truly relevant to the prediction target. The key question then is: How many models X need to be trained with a subset size s to ensure that every combination of the t relevant features is included at least once in a pool of n total features?

In other words, for each model, we select a subset s from the total n features and check for any combination of t features. If a combination is present, we mark it off the list. This process is repeated until every combination of t features has been included in at least one subset. With each draw, $\binom{s}{t}$ possible combinations of t features are covered. In total, there are $\binom{n}{t}$ possible combinations of size t.

In the paper "The Coupon Subset Collection Problem" [1], equation 2.8 addresses the question: how many subsets need to be drawn to have every coupon at least once? This is analogous to asking how many Panini football sticker packs you need to buy to collect at least one of every football player sticker.

$$E[X] = \binom{n}{s} \sum_{j=1}^{n} (-1)^{j+1} \frac{\binom{n}{j}}{\binom{n}{s} - \binom{n-j}{s}} \tag{2.8}$$

Figure 2.3 shows the process step by step.

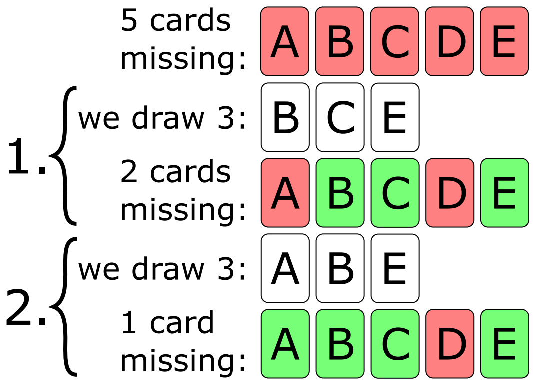


Figure 2.3: Illustration of the coupon subset collection problem, where expected number of subsets needed until each coupon is contained in at least one of the subsets is of interest. Here, with $n = 5$ and $s = 3$, equation 2.8 yields approximately 3 expected subset draws to cover all coupons at least once. Only the first two subset draws are shown.

For this thesis, equation 2.8 was extended to address the feature combination subset problem. To make it applicable, n needs to be replaced by all possible combinations of features of size t, namely $\binom{n}{t}$, and the subset draw of equation 2.8 needs to be replaced by the number of feature combinations, that are drawn with each subset draw, i.e. $\binom{s}{t}$.

$$n_{new} = \binom{n}{t} \tag{2.9}$$

$$s_{new} = \binom{s}{t} \tag{2.10}$$

Figure 2.4 illustrates the process with number of features $n = 5$, subset size $s = 3$ and relevant features $t = 2$. There are in total $\binom{n}{t} = \binom{5}{2} = 10$ pairwise feature combinations and with each subset $\binom{s}{t} = \binom{3}{2} = 3$ of these combinations are covered.

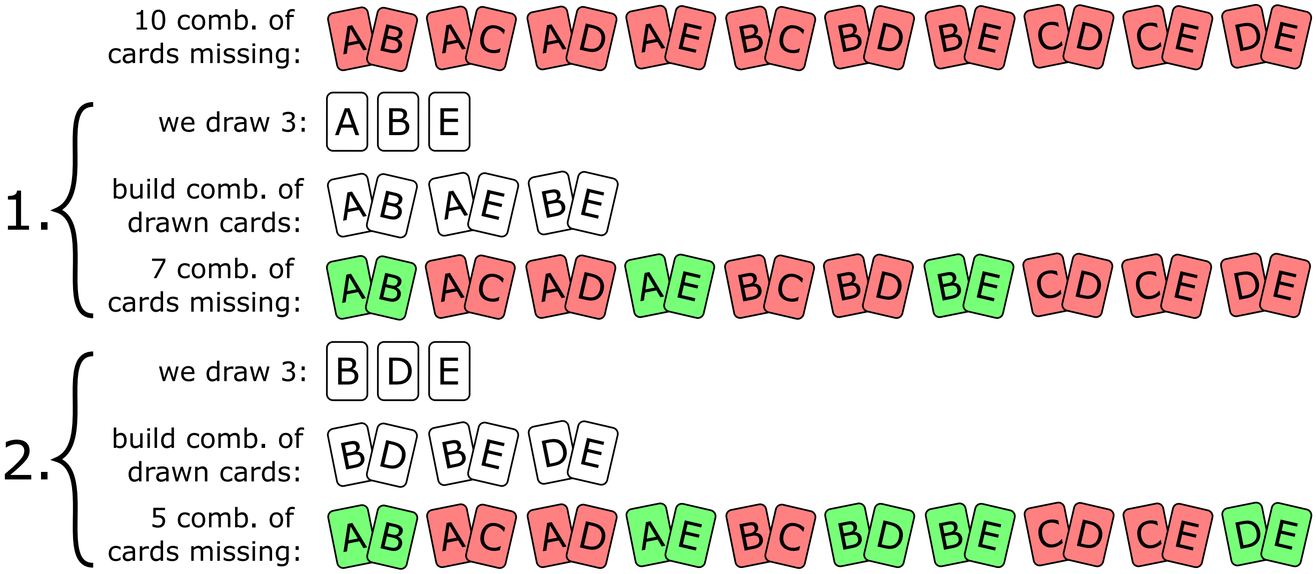

10 comb. of cards missing:
AB AC AD AE BC BD BE CD CE DE
1.
we draw 3:
A B E
build comb. of drawn cards:
AB AE BE
7 comb. of cards missing:
AB AC AD AE BC BD BE CD CE DE
2.
we draw 3:
B D E
build comb. of drawn cards:
BD BE DE
5 comb. of cards missing:
AB AC AD AE BC BD BE CD CE DE

Figure 2.4: Illustration of the adopted coupon subset collection problem, where combinations of $t = 2$ relevant features are considered. The $n = 5$ different features are: A, B, C, D and E, with a feature subset size of $s = 3$. Instead of single coupons, combination of features $\binom{n}{t} = \binom{5}{2} = 10$ are of interest. Each subset covers $\binom{s}{t} = \binom{3}{2} = 3$ combinations of total pairwise feature combinations. Only the first two subset draws are shown.

In this context, n_{new} and s_{new} correspond to n and s of the original equation 2.8. With $n = 5$, $s = 3$ $t = 2$, this leads to $n_{new} = 10$ and $s_{new} = 3$, resulting in an expected number of base models required to cover all feature combinations being approximately 9. However, with parameters like $n = 100$, $s = 10$ and $t = 5$, this results in $n_{new} \approx 17 \cdot 10^{12}$ and $s_{new} = 252$. Calculating the expected number of draws with equation 2.8 for these values becomes computationally infeasible for a standard computer. Therefore, achieving 100% coverage is not practical, and equation 2.8 should be viewed as a guideline for very small sample sizes.

3. Data

3.1 Simulated Dataset

For this thesis, two different datasets were created. An artificial dataset of dimension $n \times p$ is created by sampling $n \cdot p$ values from a normal distribution with mean zero and standard deviation one, resulting in independent and identically distributed values. Therefore the i-th row of this data matrix corresponds to the i-th sample with p features. The target value y for each sample was created by defining a fixed set of non-zero β values of length r and setting the remaining $p - r$ features to 0, e.g. in the following chapter 4.1, $\beta_{true} = [\,2; -1; 0; \dots ;0\,]$.

This β_{true} determines the relationship between data and target y, meaning only the features corresponding to the non-zero values in β_{true} affect y. Additionally, an error term sampled from a normal distribution with a standard deviation of $\sigma = 1.8^2$ was added to y, resulting in a signal-to-noise ratio of $1 : 1.8^2$. This ratio was chosen because it creates a scenario where LASSO performs well in low-dimensional feature spaces but struggles in higher dimensions.

$$y = \beta_{true} X + \epsilon \tag{3.1}$$

$$\beta_{true} = \begin{bmatrix} \beta_1 & \beta_2 & 0 & 0 & \dots & 0 \end{bmatrix} \tag{3.2}$$

with β_{true} number of non-zero coefficients chosen to be two and ϵ representing the added error to simulate real data. Number of non-zero coefficients in β_{true} can be any number between 1 and p, e.g. in chapter 4.2.1 four and eight non-zero, true model coefficients were selected for β_{true}.

3.2 LAMIS

In addition to the artificially constructed dataset, a real dataset called LAMIS is also used for training. LAMIS is a biological dataset consisting of samples with different genes as features. These genes, known as lymphoma-associated macrophage interaction signatures (LAMIS), are used to predict large B-cell lymphoma. The dataset contains 20,502 genes recorded from a cohort of 466 patients enrolled in clinical trials by the German High-Grade Non-Hodgkin Lymphoma Study Group (DSHNHL) [6]. The data was normalized by centering it around the mean of each gene and dividing by the corresponding standard deviation (z-score).

The response variable in LAMIS is survival time, which typically requires Cox regression rather than linear regression. To enable comparison with the artificial dataset, the new target variable y was created the same way as in 3.1 allowing results from *cv.glmnet* to be compared against a defined β_{true}.

Both datasets are typical examples of high-dimensional biological data with thousands of features, making predictions challenging. Furthermore, the signal-to-noise ratio for the target variable y was set to $1 : 1.8^2$, which, as discussed in the next chapter, adds to the difficulty of making accurate predictions.

4. Results

4.1 Adding features causes Signal Drowning

In chapter 2.4 *pBeta* was introduced in order to define a measure for Signal Drowning. Therefore the main goal of this section is to reproduce the intermediate steps of *cv.glmnet* algorithm and to understand why Signal Drowning is happening for high number of features.

Figure 4.1 illustrates the *pBeta* values for the simulated dataset with two non-zero β_{true} coefficients, as discussed in chapter 3. For β_{true} with non-zero indices $X_1 = 2$ and $X_2 = -1$, *pBeta* starts at 1 with 2 features, decreases to 0.59 with 16 features, remains between 0.59 and 0.42 up to 2048 features, and then drops to zero when the number of features exceeds 4096. This pattern is consistent across multiple runs of *cv.glmnet*. In the following section, we will analyze and compare why this "Signal Drowning" phenomenon occurs with a high number of features.

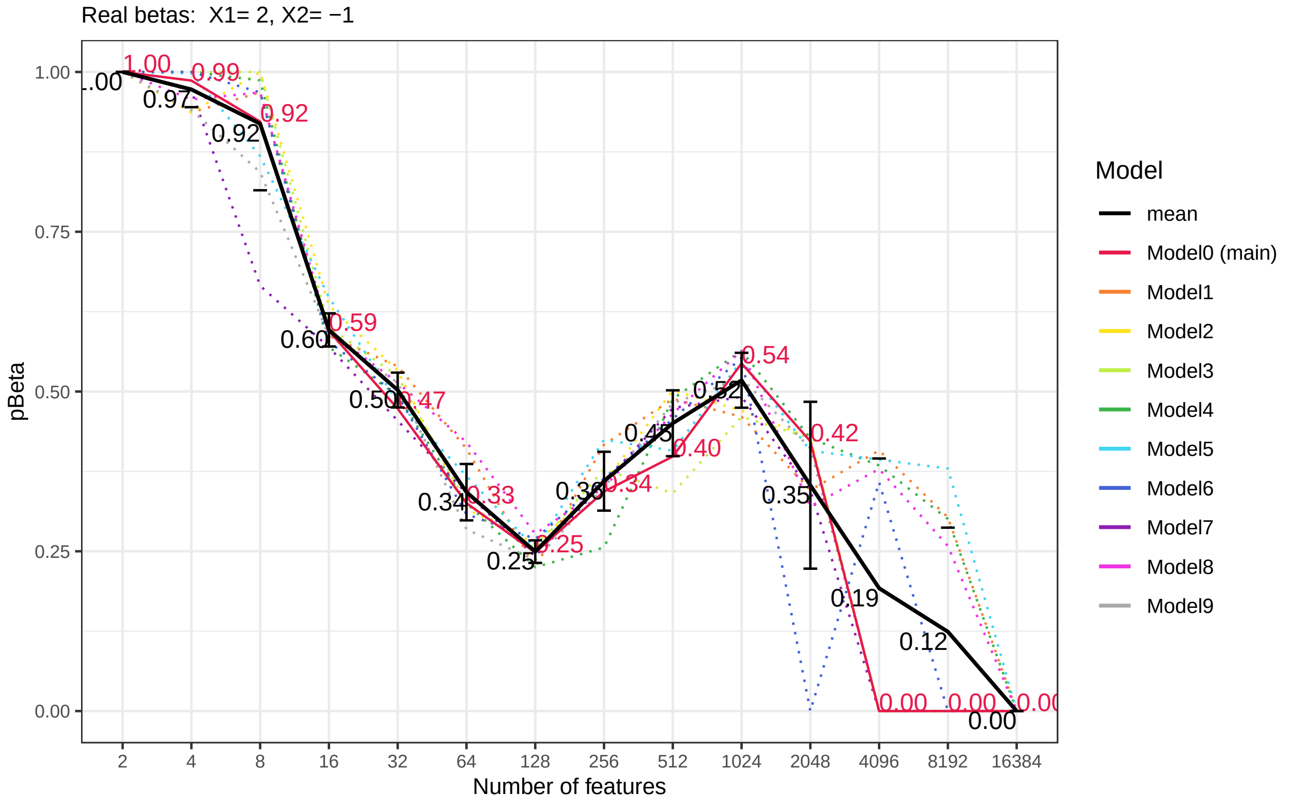


Figure 4.1: On the y-axis *pBeta* for λ_{min} is plotted, while the x-axis represents the number of features. Each tick on the x-axis corresponds to a model trained with a different number of features. For the dataset with 2 true non-zero model coefficients, $\beta_{\text{true,non-zero}} = [\,2; -1\,]$, up to 16,382 false features are added to reach Signal Drowning at 4,096 features. For instance, with 128 features, only the first two coefficients of β_{true} contribute to the target y. The *pBeta* value was calculated 10 times for each number of features, with the order of samples reshuffled before each run, which affected cross-validation splits and *pBeta* values. The black line in the figure represents the mean *pBeta* across all ten runs, while the bars indicate the standard deviation. The red line marks *Model 0*, selected as the primary model for further analysis in subsequent chapters. The remaining nine models, though not analyzed further, demonstrate the stability of *pBeta* for a given number of features.

4.1.1 5-fold Cross-Validation Plots

In figure 4.1, we choose to compare two different models obtained for two different number of features, namely 128 and 4096 number of features for the simulated dataset. The model with 128 number of features identifies some of the real β_{true} entries because the *pBeta* value is not zero. However, at 4096 features, the model fails to capture the β_{true} vector, resulting in a *pBeta* value of zero. The function *cv.glmnet* identifies the optimal λ value using a 5-fold cross-validation (default is 10-fold) on the training data. During each fold, the data is split into a training set (80%) and a test set (20%).

In Figure 4.2, each subplot shows the mean squared error (MSE) on the test set for a specific fold with 128 features, with the bottom right subplot representing the average MSE across all 5 folds. *cv.glmnet* determines the global λ_{min} based on this averaged MSE. For each fold, the λ value that minimizes the MSE is called the local λ_{min}. Figure 4.3 illustrates a well-defined minimum (green vertical line) for all test folds. The global λ_{min} is the minimum of the averaged MSE curve from all 5 folds, shown in the final subplot of Figure 4.2.

Adding the standard deviation of the 5 local λ values to the global λ_{min} results in λ_{1se}, which is generally preferred for better model interpretability as it tends to result in fewer non-zero coefficients. Additionally, the λ value that corresponds to exactly n non-zero coefficients (besides the intercept) is highlighted in the plots as the vertical dashed red line, referred to as $\lambda_{n\neq 0}$. In Figures 4.1 and 4.2, two non-zero coefficients are considered, so $\lambda_{n\neq 0}$ represents the model with exactly two non-zero model coefficients. If there is no model with exactly two β coefficients the next higher available number of non-zero coefficients is picked.

In practice, the number of true non-zero coefficients is unknown, so $\lambda_{n\neq 0}$ illustrates how the model would perform, if it were forced to select only the two most important features. This concept will be further explored in a later chapter by subsetting the number of features, allowing the model to exclude false features by chance.

In Figure 4.2, the five different cross-validation folds are displayed. Table (a) 4.1 provides an excerpt of the β coefficients for each of these folds, labeled as X to match the *cv.glmnet* nomenclature. Despite *pBeta* being only 25% for 128 features (as shown in figure 4.1), the model identifies the β_{true} coefficients β_1 and β_2 ($\beta_{true}[1:2] = [-2;1]$) in every fold. Table (b) 4.1 shows the results when the model is forced to select only two features at $\lambda_{n\neq 0}$. Here, X_1 is selected in 3 out of 5 folds, while X_2 is selected only twice. The MSE varies significantly across the folds. Notably, fold 2, which correctly identifies both X_1 and X_2, still has a higher MSE than fold 5, which selects X_{15} over X_2. Table (c) 4.1 displays the β coefficients for the final model shown in the sixth subplot of figure 4.2. This model includes 29 non-zero coefficients with a global $\lambda_{min} = 0.31$. The table also includes a column showing the coefficients identified by the *BeSS.one* package, which uses a best subset selection approach. Interestingly, *BeSS.one* accurately identifies the true model coefficients when limited to two-feature combinations. In contrast, for $\lambda_{n\neq 0}$, the LASSO model selects X_{37} instead of X_2.

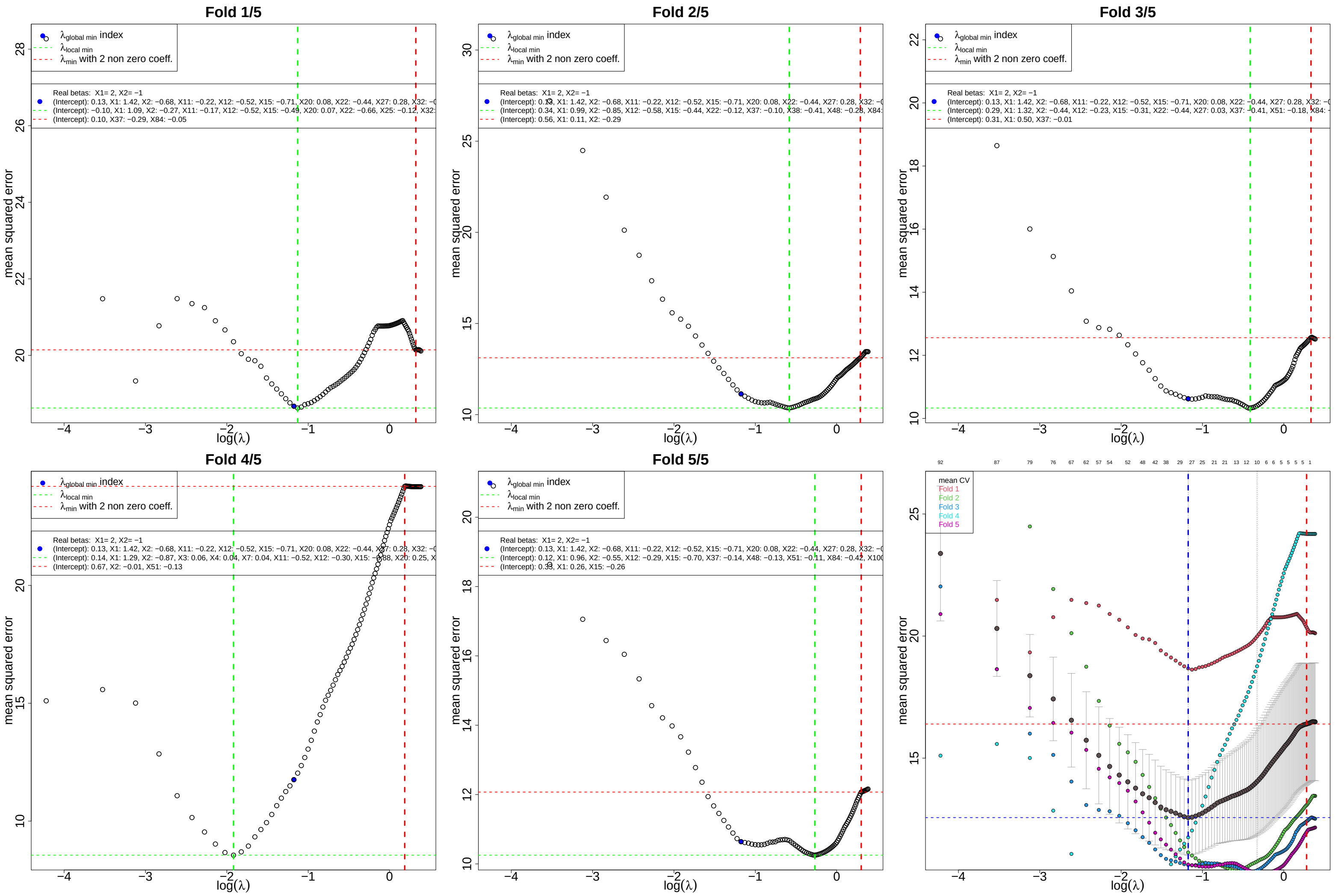


Figure 4.2: In the figure, each subplot represents the test fold of a 5-fold cross-validation using *cv.glmnet* with true coefficients $\beta_{true} = [2; -1; 0; \ldots; 0]$ across 128 features. The green vertical line in each fold's plot indicates the local λ_{min}, where the mean squared error (MSE) is at its lowest. The red vertical line shows the $\lambda_{2\neq0}$, corresponding to the model with exactly two non-zero coefficients. The legend in each subplot lists the model coefficients selected in that fold, matching those shown in table 4.1 (a & b). The bottom right subplot aggregates the results by averaging the MSE across all five folds. The global λ_{min} is identified as the minimum of this average MSE curve, marked by the blue vertical line.

	Fold1	**Fold2**	**Fold3**	**Fold4**	**Fold5**
non-zero	25.00	14.00	11.00	49.00	9.00
CVM	18.63	10.36	10.33	8.55	10.25
local λ_{min}	0.32	0.56	0.66	0.15	0.76
(Intercept)	-0.10	0.34	0.29	0.14	0.12
X1	1.09	0.99	1.32	1.29	0.96
X2	-0.27	-0.85	-0.44	-0.87	-0.55
X3	0.00	0.00	0.00	0.06	0.00
X4	0.00	0.00	0.00	0.04	0.00
X7	0.00	0.00	0.00	0.04	0.00
X11	-0.17	0.00	0.00	-0.52	0.00
X12	-0.52	-0.58	-0.23	-0.30	-0.29
X15	-0.49	-0.44	-0.31	-0.88	-0.70
X20	0.07	0.00	0.00	0.25	0.00
X22	-0.66	-0.12	-0.44	-0.26	0.00
X23	0.00	0.00	0.00	0.27	0.00
...	...	...	...	...	...

(a)

	Fold1	**Fold2**	**Fold3**	**Fold4**	**Fold5**
non-zero	2.00	2.00	2.00	2.00	2.00
CVM	20.14	13.12	12.56	24.20	12.07
2 non-zero λ	1.38	1.34	1.40	1.21	1.35
(Intercept)	0.10	0.56	0.31	0.67	0.33
X1	0.00	0.11	0.50	0.00	0.26
X2	0.00	-0.29	0.00	-0.01	0.00
X15	0.00	0.00	0.00	0.00	-0.26
X37	-0.29	0.00	-0.01	0.00	0.00
X51	0.00	0.00	0.00	-0.13	0.00
X84	-0.05	0.00	0.00	0.00	0.00

(b)

	all Folds	**all Folds**	**all Folds**
	λ_{min}	$\lambda_{2\ non-zero}$	BeSS
non-zero	29.00	2.00	2.00
CVM	12.57	16.40	11.59
λ	0.31	1.32	1.19
(Intercept)	0.13	0.40	0.21
X1	1.42	0.16	1.88
X2	-0.68	0.00	-1.51
X11	-0.22	0.00	0.00
X12	-0.52	0.00	0.00
X15	-0.71	0.00	0.00
X20	0.08	0.00	0.00
X22	-0.44	0.00	0.00
X27	0.28	0.00	0.00
X32	-0.09	0.00	0.00
X37	-0.47	-0.02	0.00
...	...	...	...

(c)

Table 4.1: The table shows the β coefficients labeled as X_0*(Intercept)*$, X_1, \ldots, X_{128}$ for 128 available features across three scenarios: (a) the five folds with their local λ_{min}, (b) the five folds with their $\lambda_{n\neq0}$, and (c) the sixth subfigure in Figure 4.2.

For comparison, Figure 4.3 shows the 5-fold cross-validation with 4096 features, where the *pBeta* value was zero, indicating the model couldn't correctly identify features X_1 and X_2. The figure shows no clear minimum in any fold, with the dashed green vertical line always near the right side, indicating heavy regularization. Table 4.2 (a) reveals that each fold finds a solution with one to nine non-zero coefficients, but no fold correctly identifies both β coefficients. Interestingly, the combined solution in the sixth plot only finds the intercept-only model at $\lambda = 1.52$, despite three folds picking at least one non-zero model coefficient at the same λ value. Even when the model is informed that there are only two true features among the 4096, table 4.2 (c) shows that both $\lambda_{n\neq0}$ and *BeSS.one* fail to identify the second true β coefficient X_2, instead selecting feature X_{1983}, which has a β coefficient of -1.32, similar to $\beta_{true}[2] = -1$.

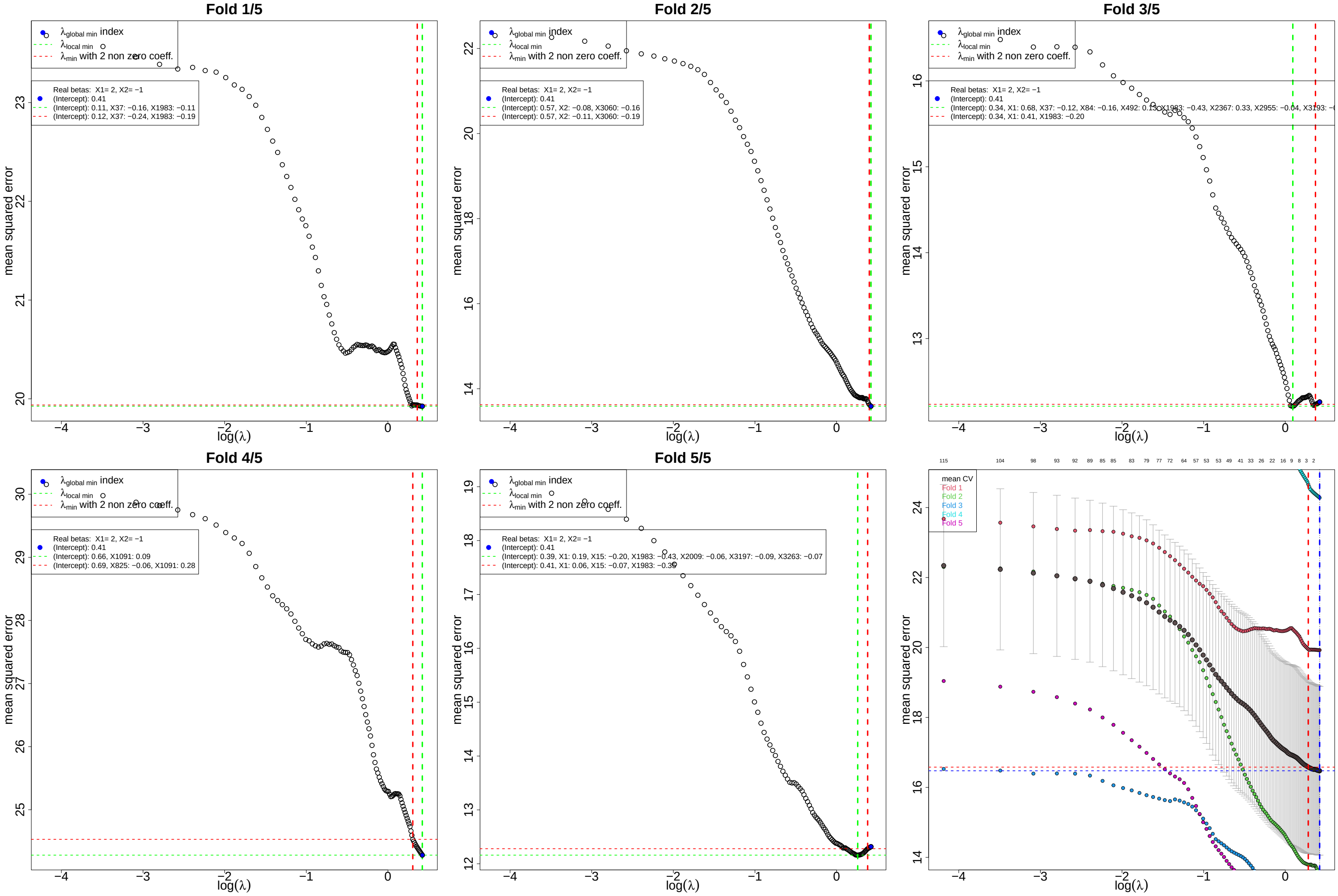


Figure 4.3: In the figure, each subplot represents the test fold of a 5-fold cross-validation using *cv.glmnet* with $\beta_{true} = [2; -1; 0; \ldots; 0]$ across 4096 features. The green vertical line in each fold's plot indicates the local λ_{min}, where the mean squared error (MSE) is at its lowest. The red vertical line shows the $\lambda_{n\neq0}$, corresponding to the model with exactly two non-zero coefficients. The legend in each subplot lists the model coefficients selected in that fold, matching those shown in table 4.1 (a & b). The bottom right subplot aggregates the results by averaging the MSE across all five folds. The global λ_{min} is identified as the minimum of this average MSE curve, marked by the blue vertical line.

	Fold1	**Fold2**	**Fold3**	**Fold4**	**Fold5**
non-zero	2.00	2.00	9.00	1.00	6.00
CVM	19.92	13.59	12.21	24.28	12.16
local λ_{min}	1.52	1.52	1.10	1.52	1.30
(Intercept)	0.11	0.57	0.34	0.66	0.39
X1	0.00	0.00	0.68	0.00	0.19
X2	0.00	-0.08	0.00	0.00	0.00
X15	0.00	0.00	0.00	0.00	-0.20
X37	-0.16	0.00	-0.12	0.00	0.00
X84	0.00	0.00	-0.16	0.00	0.00
X492	0.00	0.00	0.13	0.00	0.00
X1091	0.00	0.00	0.00	0.09	0.00
X1983	-0.11	0.00	-0.43	0.00	-0.43
X2009	0.00	0.00	0.00	0.00	-0.06
X2367	0.00	0.00	0.33	0.00	0.00
X2955	0.00	0.00	-0.04	0.00	0.00
X3060	0.00	-0.16	0.00	0.00	0.00
...	...	...	...	...	...

(a)

	Fold1	**Fold2**	**Fold3**	**Fold4**	**Fold5**
non-zero	2.00	2.00	2.00	2.00	3.00
CVM	19.94	13.62	12.23	24.53	12.28
2 non-zero λ	1.43	1.49	1.45	1.36	1.46
(Intercept)	0.12	0.57	0.34	0.69	0.41
X1	0.00	0.00	0.41	0.00	0.06
X2	0.00	-0.11	0.00	0.00	0.00
X15	0.00	0.00	0.00	0.00	-0.07
X37	-0.24	0.00	0.00	0.00	0.00
X825	0.00	0.00	0.00	-0.06	0.00
X1091	0.00	0.00	0.00	0.28	0.00
X1983	-0.19	0.00	-0.20	0.00	-0.35
X3060	0.00	-0.19	0.00	0.00	0.00

(b)

	all Folds	**all Folds**	**all Folds**
	λ_{min}	$\lambda_{2\ non-zero}$	BeSS
non-zero	1.00	2.00	2.00
CVM	16.47	16.58	12.17
λ	1.52	1.33	0.89
(Intercept)	0.41	0.42	0.48
X1	0.00	0.13	1.40
X1983	0.00	-0.18	-1.32

(c)

Table 4.2: The table presents β coefficients labeled as X_0*(Intercept)*$, X_1, \ldots, X_{4096}$ for 4096 available features across three scenarios: (a) local λ_{min} for the five folds, (b) $\lambda_{n\neq 0}$ for the five folds, and (c) the sixth subfigure from figure 4.3. The "non-zero" row indicates the non-zero β coefficients, excluding β_0. In (b), the model is estimated for λ with two non-zero β coefficients, with the next higher number selected if no exact solution exists (e.g., fold 5 has 3 non-zero coefficients). In (c), at λ_{min}, a single non-zero β coefficient (either $X1$ or $X1983$) is identified, but it is smaller than 0.05 and thus not displayed.

4.1.2 MSE Curves

In the previous chapter, the model did not find the correct solution anymore for high enough number of features, i.e. 4096 features. One can think of two different reasons why this might be the case. Firstly, the correct combination of features is never considered to build the model. During training LASSO picks the first feature, which explains the training error best. This feature could be false and just by chance correlates with y best on the training set. With decreasing λ the model adds new features successively in order to explain the remaining training error continuously more in respect to the already picked feature(s) [4]. The model repeats this for all λ values but since the correct combination of features is not picked first, the false model can end up with a bigger error than the combination of real coefficients β_{true}, even though each of the true features do not correlate with y strongly on their own but do so combined. The false model has a bad performance on the test fold for every λ and in the end an intercept-only model is returned as final model since β_0 is exempt from the regularization in equation 2.3, therefore only providing an offset or an average of the y targets as the simplest prediction possible. For 4096 number of features the final model was the intercept-only model (see table 4.2 (c)) even though all of the folds did find a solution close to but not exactly the same as the intercept only model. For example fold 1 picks X_{37} and X_{1983} over X_1 and X_2. Therefore another option is that the model with the false features simply performs better in training, even in a 5-fold cross-validation, than a model with the correct features. This could be because the picked features correlate to the true features by chance and actually look better than the real features on the test fold. Conclusively, the first case could be solved by feature subsetting (chapter 4.2), i.e. leaving false features out randomly and therefore forcing the model to pick the correct features by chance. In the second case, if false features outperform true features, then feature subsetting looks less promising since the algorithm has no chance of differentiating a true feature from a wrong feature combination by looking at their respective training error. This can happen if the signal is too weak in comparison to the noise in the data.

Figures 4.4 and 4.6 illustrate the training, cross-validation, and test MSE across different values of λ for 128 and 4096 number of features respectively. On the right-hand side scale, the MSE values are plotted. As expected, the training MSE curve decreases as λ becomes smaller. This occurs because, as λ decreases, more features are included in the model, enabling it to fit the training data more closely. However, this comes at a cost: the model becomes increasingly tailored to the specific training data, losing its ability to generalize to new, unseen data. As a result MSE of test data and cross-validation of the test folds is increasing. This process is called overfitting.

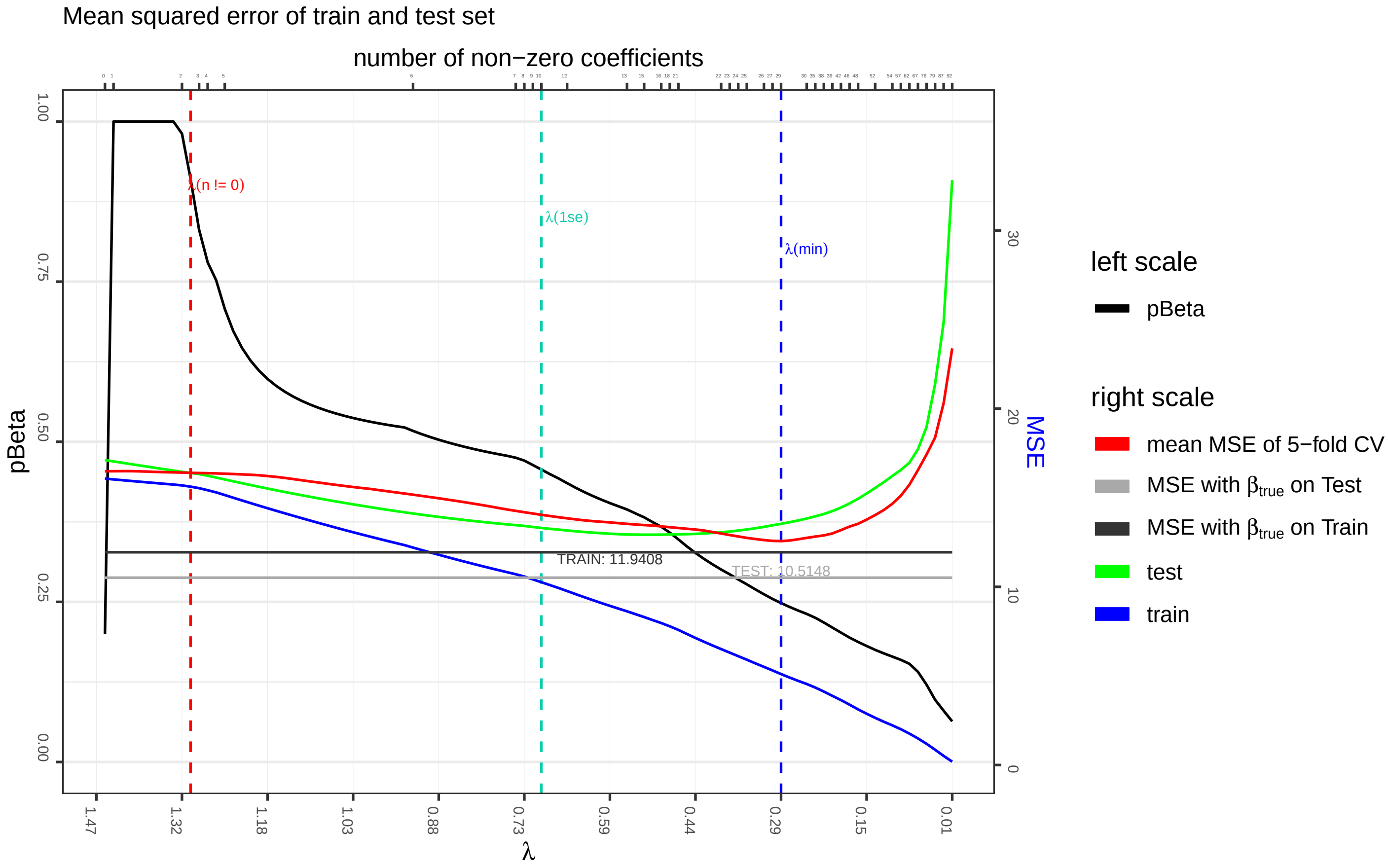


Figure 4.4: Figure shows *pBeta* on the left-hand side scale (black line) for a model with 128 features and 2 real model coefficients. On the right-hand side scale MSE for training, test and cross-validation mean (CVM) error for the LASSO model is shown. The three vertical lines indicate special values of λ, i.e. λ_{min}, λ_{1se} and $\lambda_{n\neq0}$. The top scale of the plot shows the number of non-zero coefficients in the model for each λ value.

In theory, the best MSE that a model can achieve corresponds to the MSE of a model using the true coefficients, where β_{true} is set as $X_1 = 2$, $X_2 = -1$, and all other coefficients are zero. If the model correctly identifies only these true model coefficients during training, without any false positives, its predictions on new data would be optimal. This optimal prediction MSE is represented by the black horizontal line (for the training set) and the grey horizontal line (for the test set) in figure 4.4. As expected, both lines show similar MSE values, indicating the best possible performance if only the true features are identified.

Interestingly, the training MSE drops below the MSE of the true model, β_{true}. This suggests that the model begins to identify features that improve training performance but are not part of the true feature set. This is confirmed by the β_{true} values on the left-hand side scale of the plot, which track the proportion of correct features in the model. As $pBeta$ decreases, meaning the model increasingly includes incorrect features, the MSE for the test and cross-validation datasets rises. This increase in MSE indicates the model's overfitting to the training data.

In figure 4.4, the cross-validation error curve marked in red shows that the model achieves its minimum at $\lambda_{min} = 0.29$. According to table 4.1 (c), at this value of λ_{min}, the model has 29 non-zero coefficients. While this includes the true features X_1 and X_2, it also includes many false positive features. By definition, λ_{min} minimizes the cross-validation mean (CVM) error curve. However, despite achieving the lowest error, the $pBeta$ value at λ_{min} is only 0.25. When examining the MSE, these additional features at λ_{min} provide a better fit to the training data, improving the model's ability to explain the error in y even on the test fold. Typically, we would expect such a model to fail on the test data due to overfitting. However, in this case, the model with additional false positives still outperforms the sparser model at λ_{1se} in terms of test MSE. The non-sparse model at λ_{min} does incur a penalty in the L1 term in the equation 2.3. This means that a non-sparse solution must be considerably better than a sparse solution, such as the one obtained at λ_{1se}, to be chosen. When the model at $\lambda_{n\neq 0}$ is forced to pick two β coefficients, it only finds $X_1 = 0.16$ as the correct feature, however since the second feature $X_{37} = -0.02$ has a small magnitude the overall $pBeta$ value is quite high. In figure 4.4 if $\lambda > 1.30$ the model would have a $pBeta = 1$ before λ is so big that the model would consist of the intercept-only model, where $pBeta$ goes down to zero.

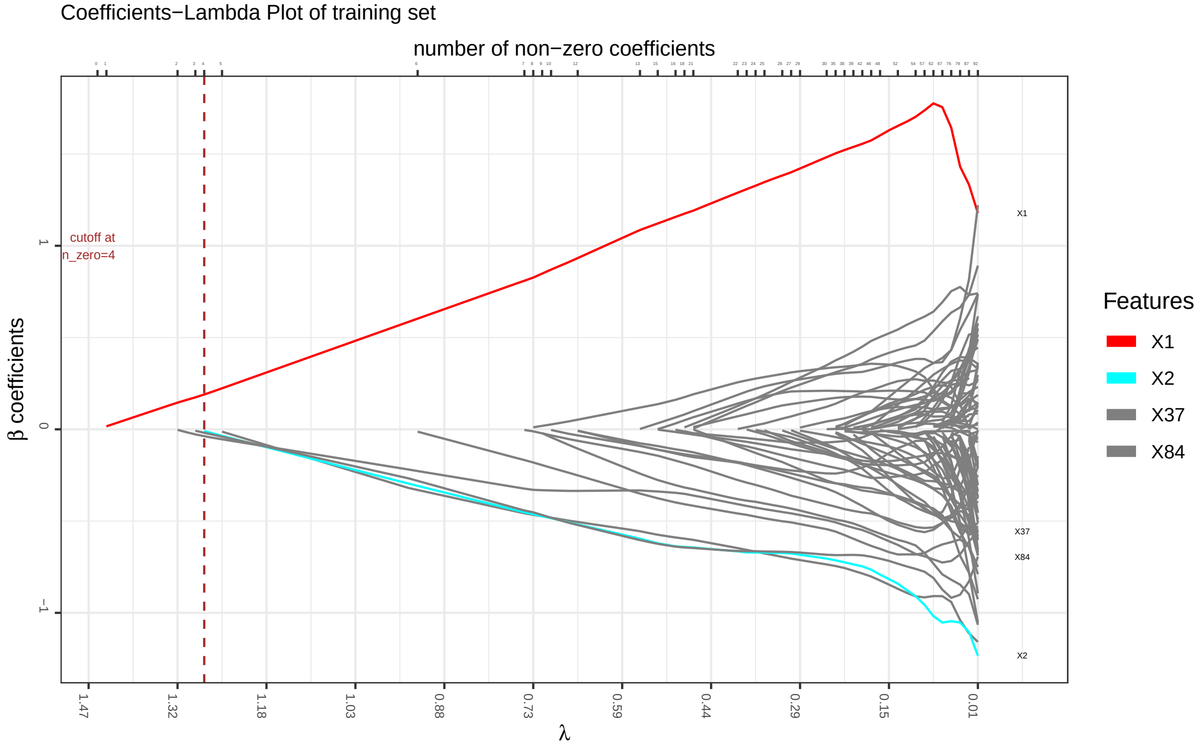


Figure 4.5: Figure shows the magnitude of model coefficients along the λ pathway for 128 features, going from maximal regularization for big λ to minimal regularization for small λ values on the training data. A fixed cutoff was selected to limit the features shown in the legend for readability reasons. On the top x-axis the number of non-zero coefficients is marked. The real features are $X_1 = 2$ and $X_2 = -1$. The first four features found by the model however are X_1, X_{37}, X_{84} and then X_2.

In table 4.1, when dealing with 128 features, the model consistently identifies X_1 but struggles more with X_2. This discrepancy arises because X_1 has a larger magnitude compared to X_2 ($X_1 = 2$ and $X_2 = -1$), making X_2 harder for the model to detect. This difficulty is also reflected in figure 4.5, which illustrates the magnitude of the β values selected by the model as λ decreases. In figure 4.5, as λ approaches zero, the regularization effect diminishes, leading to the inclusion of more features in the model. In theory, when lambda = 0, the model would be unregularized, including every available feature. The top of the figure notes the number of non-zero model coefficients as λ changes. For readability, the legend only shows the model coefficients for the four most significant features rather than all 128. As expected, X_1 is the first feature selected by the model, likely because it explains the training error best. Interestingly, X_2 is only identified as the fourth most significant feature, after features X_{37} and X_{84}. This sequence suggests that while X_2 is a true predictor, its lower magnitude causes it to be overshadowed by other features that better explain the remaining error on training data at earlier stages of the model's selection process.

In figure 4.6, the training MSE curve for 4096 features is similar to that with 128 features, but the cross-validation mean (CVM) reaches its minimum at the intercept-only model. Table 4.3 provides the detailed train and test errors for the 4096-feature case. Table 4.2 (c) confirms that the model does not find any correct features at λ_{min}. Interestingly, the cross-validation mean (CVM) curve doesn't align with the test error curve, which shows a clear dip around $\lambda \approx 0.90$, unlike the CVM curve. The behavior of the CVM curve varies based on how the samples in the folds are distributed during training. For low λ values, the CVM error is even bigger than the test error, which is unusual since the model is optimized on cross-validation data for a specific λ value, i.e. samples of a training fold are part of the test in another fold. Unlike the intercept-only model at λ_{min}, the model at $\lambda_{n \neq 0}$ (red vertical line) picks X_1 and X_{1983} as the features that explain the response variable best, achieving $pBeta \approx 0.44$. This behavior is confirmed in figure 4.7, here the first four features that are found are X_{1983}, X_1, X_{84} and X_{37}, with the correct feature X_2 appearing only in 5th place.

	Train	Test
MSE:		
$\hat{y} = \beta_{\text{true}} X$	11.9408	10.5148
$\hat{y} = \hat{\beta}_{\lambda(\text{min})} X$	16.0716	17.1160
$\hat{y} = \hat{\beta}_{\lambda(\text{1se})} X$	16.0716	17.1160
$\hat{y} = \hat{\beta}_{\lambda(n \neq 0)} X$	15.2301	16.6242
$\hat{y} = \hat{\beta}_{\text{BeSS}} X$	12.1742	14.7266

Table 4.3: Each row shows a different way to predict $\hat{y}$. MSE is the mean squared difference between y and $\hat{y}$, which is calculated for the model with 4096 number of features for train and test simulated dataset for different λ values. Function *BeSS.one* does not have a regularization and therefore no λ value, just the corresponding β coefficients.

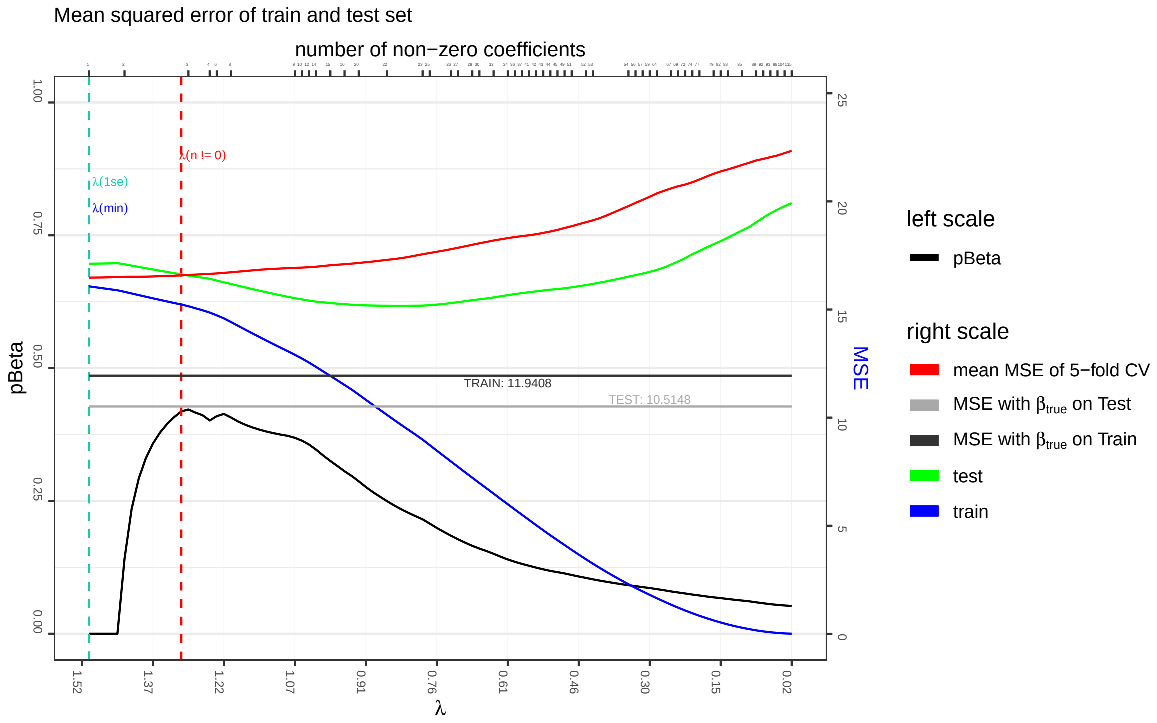


Figure 4.6: Figure shows *pBeta* (for 4096 features with 2 real model coefficients) on the left scale (black line) and training, test and cross-validation mean error (CVM) on the right scale for the function *cv.glmnet*. The three vertical lines indicate special values of λ, i.e. λ_{min}, λ_{1se} and $\lambda_{n\neq0}$

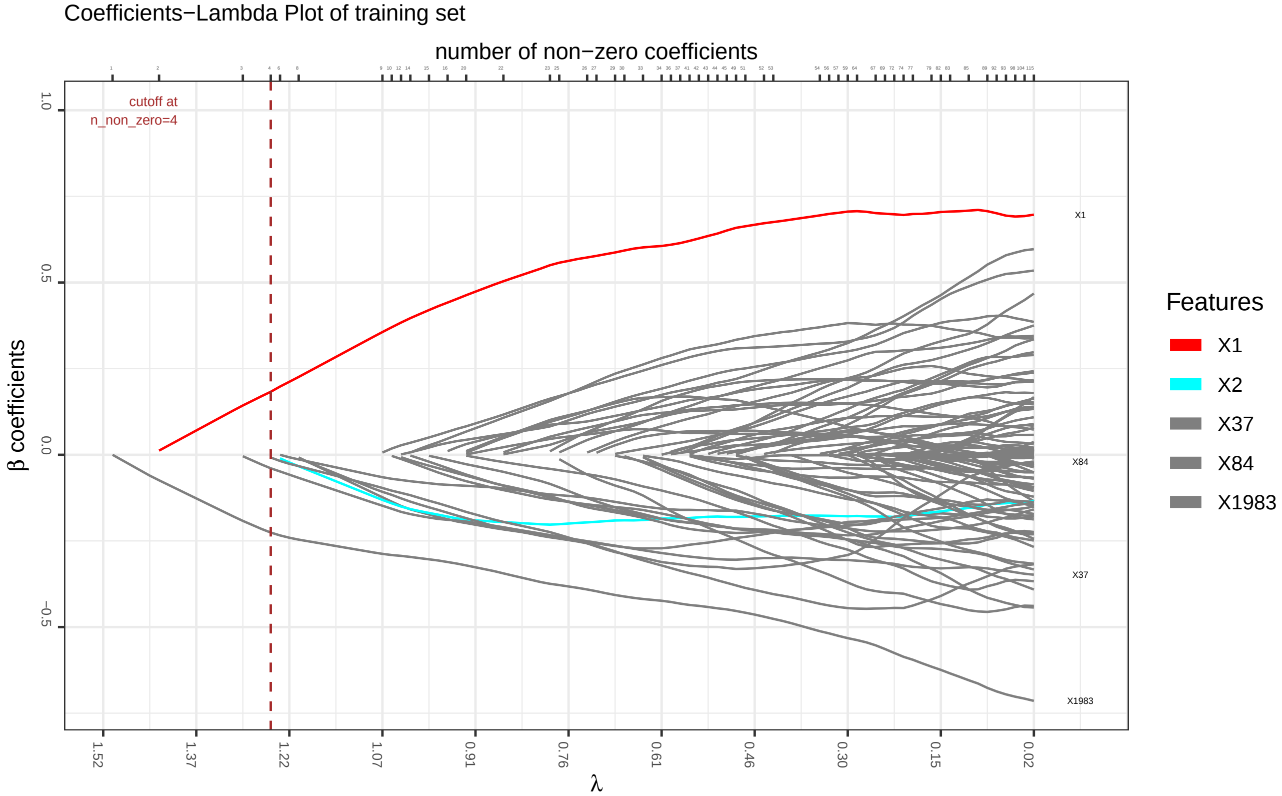
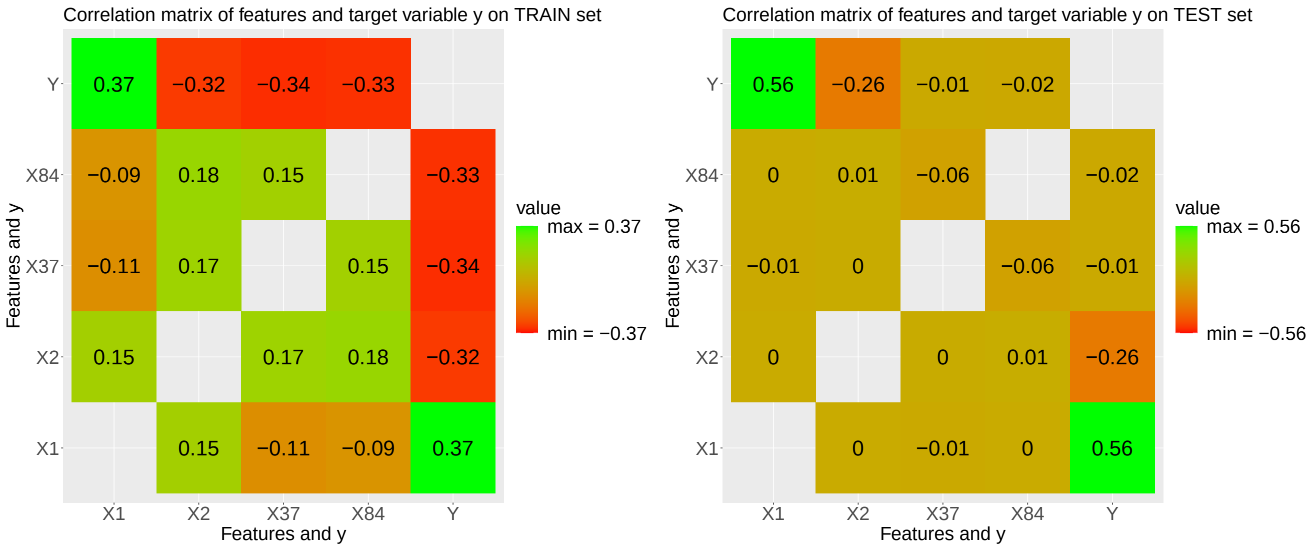


Figure 4.7: Figure shows the magnitude of beta coefficients along the λ pathway for 4096 features, going from maximal regularization for big λ and minimal regularization for small λ values on the training data. A random cutoff was selected to limit the features shown in the legend for readability reasons. On the top x-axis the number of non-zero coefficients is marked. The real features are $X_1 = 2$ and $X_2 = -1$. The first four features found by the model however are X_{1983}, X_1, X_{84} and X_{37} and then X_2.

Figure 4.8 and 4.9 help explain why some features perform better on the training data than β_{true}. These figures show a correlation matrix for the non-zero β values of β_{true}, the features, which are picked early by the model (see figures 4.5 and 4.7) and the target variable y. As expected, in figure 4.8 the target variable y is correlated highest to X_1 ($\beta_{true}[1:2] = [2; -1]$). Similarily, y is anticorrelated to X_2, X_{37} and X_{84}. However, X_{37} and X_{84} have a slightly higher correlation with y than X_2, which may explain why the model prefers these features in the coefficient plot 4.5 over the real feature X_2. These false positive features do not fail on the test fold because they correlate with y.

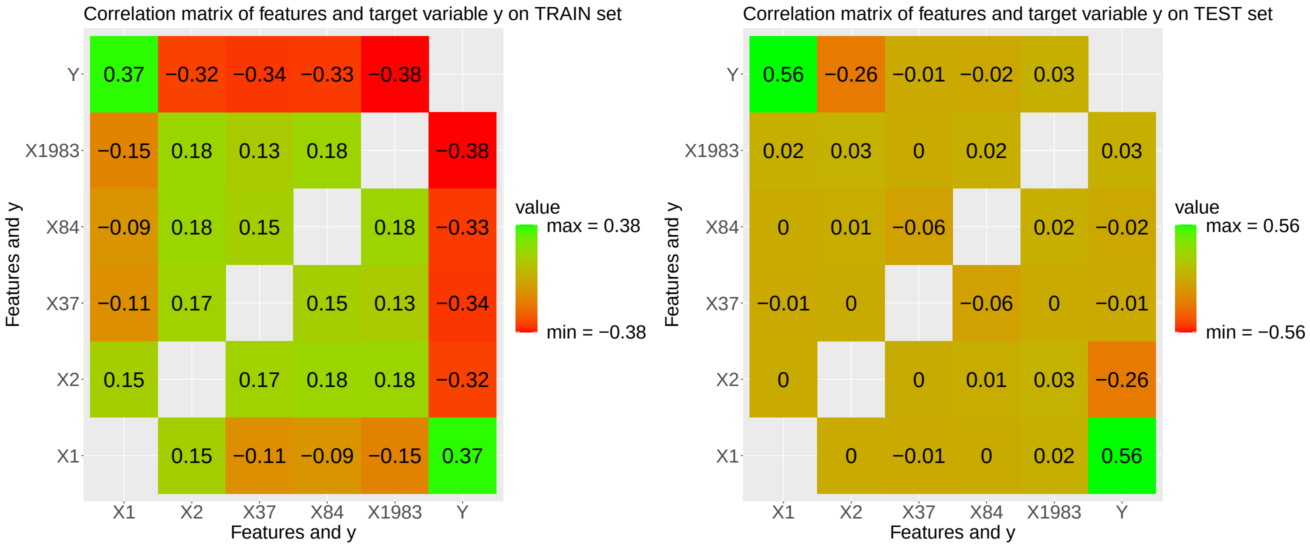
In figure 4.9 (a) X_1 correlates with y, but the incorrect feature X_{1983} has a slightly higher anticorrelation than X_1 by chance. Again, this confirms why the model picks feature X_{1983} first in figure 4.7 and X_1 only second. X_2 shows the weakest absolute correlation with y in comparison to all the false positive features X_{1983}, X_{37} and X_{84}. Interestingly, on the test set false features stop correlating with y in figure 4.9 (b). It seems that with so many features the model will always find highly correlating ones optimized to a specific dataset. So even in cross-validation these features perform well since cross-validation is performed on the training data. As expected, on the test data the true features X_1 and X_2 do not change their correlation much. The next chapter will explore the idea of randomly subsetting features to force the model to exclude false features like X_{1983} and increase the chances of selecting true features such as X_1 and X_2.



(a) train

(b) test

Figure 4.8: Correlation between real and early picked features by the model with 128 number of features on train (a) and test set (b). The correlation matrix is normalized in magnitude by its absolute biggest appearing value (excluding correlations of variables with themselves, which are always 1). Therefore the scale is not from -1 to 1 but in the range of $\pm|corr_{max}|$ in order to have a higher contrasting color scheme.



(a) train

(b) test

Figure 4.9: Correlation between real and *early picked* features by the model with 4096 number of featues on train (a) and test set (b). The correlation matrix is normalized in magnitude by its absolute biggest appearing value (excluding correlations of variables with themselves, which are always 1). Therefore the scale is not from -1 to 1 but in the range of $\pm|corr_{max}|$ in order to have a higher contrasting color scheme.

4.2 LASSO Ensembles do not prevent Signal Drowning

In the previous chapter, we discussed the difficulty of identifying correct models as the number of features increases, and the role of feature correlation was explored in this challenge. Two potential solutions are proposed to address this issue.

The first idea is to ensure that the model considers every feature as the initial feature during the training process. If the model correctly identifies the first feature, the likelihood of finding the subsequent true features increases. This can be implemented by resetting each β value in the penalty term in equation 2.3 to zero for different training runs, making each feature more accessible to the model once. The model with the lowest MSE would ideally have identified the correct initial features. In figure 4.5 however, the model for 128 number of features has identified X_1 correctly but X_2 only after two false positive features as the 4th most significant model coefficient. Another approach at solving the failed training would be to train multiple models separately, where each model is presented with only a subset of features during training. These individual models, or base models (short bm), are combined by averaging the β coefficients across all base models. More precisely, averaging the predictions of every base model or averaging the β values of every base model to have a single prediction is mathematically the same. This technique is akin to ensemble learning and is therefore labeled LASSO Ensemble. The idea of the ensemble is that base models, that are only presented false features, will either have poor performance and can be selectively disregarded (in certain combine strategies) or base models with wrong features result in the intercept-only model with no contribution when averaged. Ideally by feature subsetting, base models with correctly selected features should emerge and have a MSE close to the MSE of the model with β_{true}. For example figure 4.4 for 128 features generally shows a lower MSE at λ_{min} than any value of the test and CVM error in figure 4.6 for 4096 features. Especially since table 4.1 (c) for 128 features shows in the last column that *BeSS.one* is still able to find both correct features X_1 and X_2.

Figure 4.10 displays results for 4096 features, with the same layout to figure 4.6, but with the added comparison of ensemble models against the classic "Full-Feature LASSO" model, which is shown in the background (grey) with its key λ values (λ_{min}, λ_{1se} and $\lambda_{n\neq0}$) highlighted.

The ensembles were created using a 20% feature subset (819 features out of 4096). As detailed in Chapter 2.6, each ensemble's λ_{best} was determined through 10-fold cross-validation, involving training 100 models for each of the 100 λ values, where λ_{min} and λ_{1se} are included in those 100 λ values and are not trained separately. The λ sequences for the base models within each ensemble do not align perfectly, so they are combined using their λ indices. The x-axis in figure 4.10 represents the 100 λ values from the Full-Feature LASSO model, and the base models' λ indices are matched accordingly.

In figure 4.10, colored vertical dashed lines indicate the λ_{best} index for each ensemble. For instance, the max-ensemble's λ_{best} is found at index 41, represented by the red vertical line. If λ_{best} aligns with λ_{min} or λ_{1se}, which may vary across base models, their index is set to 101 or 102, respectively, and are placed on the far right of the plot, indicated by λ_{min} or λ_{1se} labels near the vertical λ_{best} lines and their training and test errors for each ensemble are marked by dots along their respective lines, e.g. for mean-, top10- and weighted-ensemble $\lambda_{best} = \lambda_{min}$ in figure 4.10.

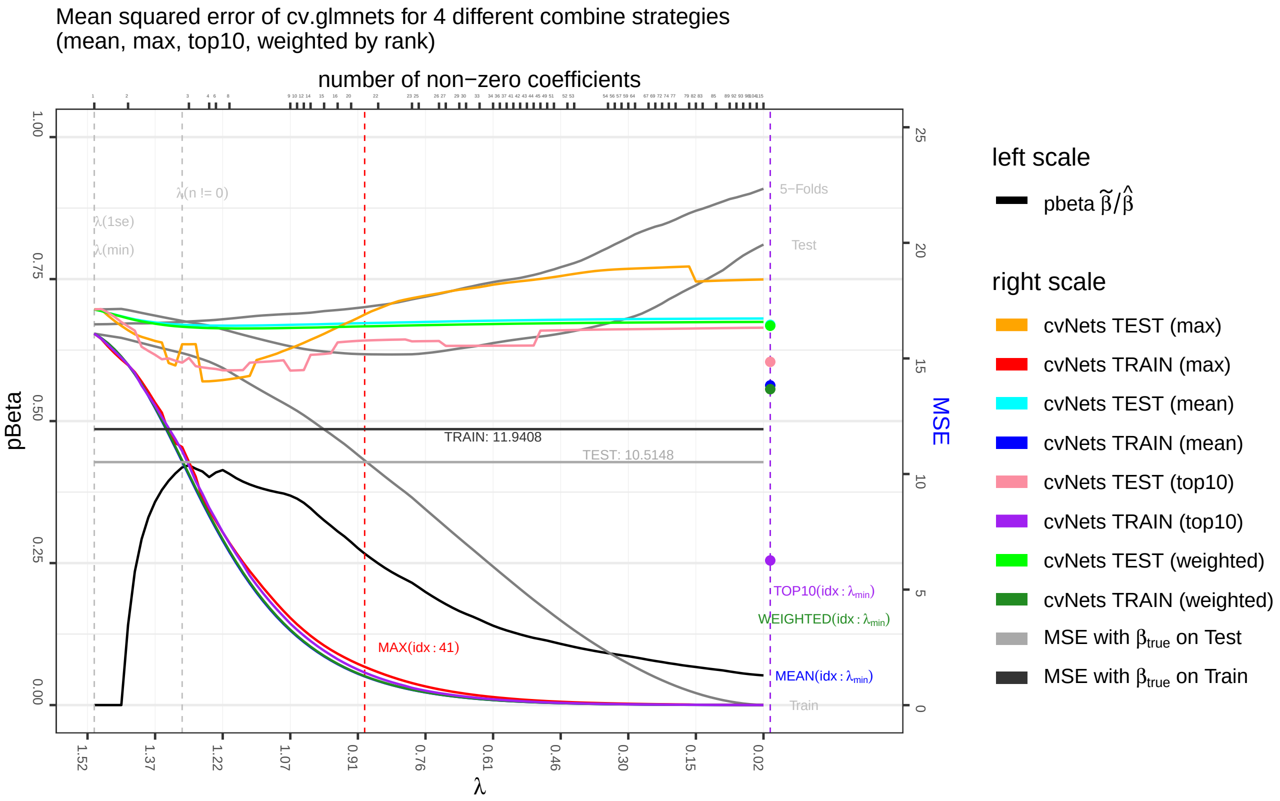


Figure 4.10: Figure compares ensembles to the Full-Feature LASSO model for 4096 number of features. The Full-Feature LASSO model was introduced in figure 4.6 and is greyed out here in the background. Training and test performance is displayed for every combine strategy of *cv.glmnets*. The vertical lines indicate the index position of λ_{best}. The x-axis shows the λ sequence of the Full-Feature LASSO model and for each λ step the ensembles are mapped accordingly using the λ index of their base models. For clarification, the light blue dot is slightly visible under the light green dot and training and test error of max-ensemble are to be read from their respective intersection (orange and red curves) at λ_{best}.

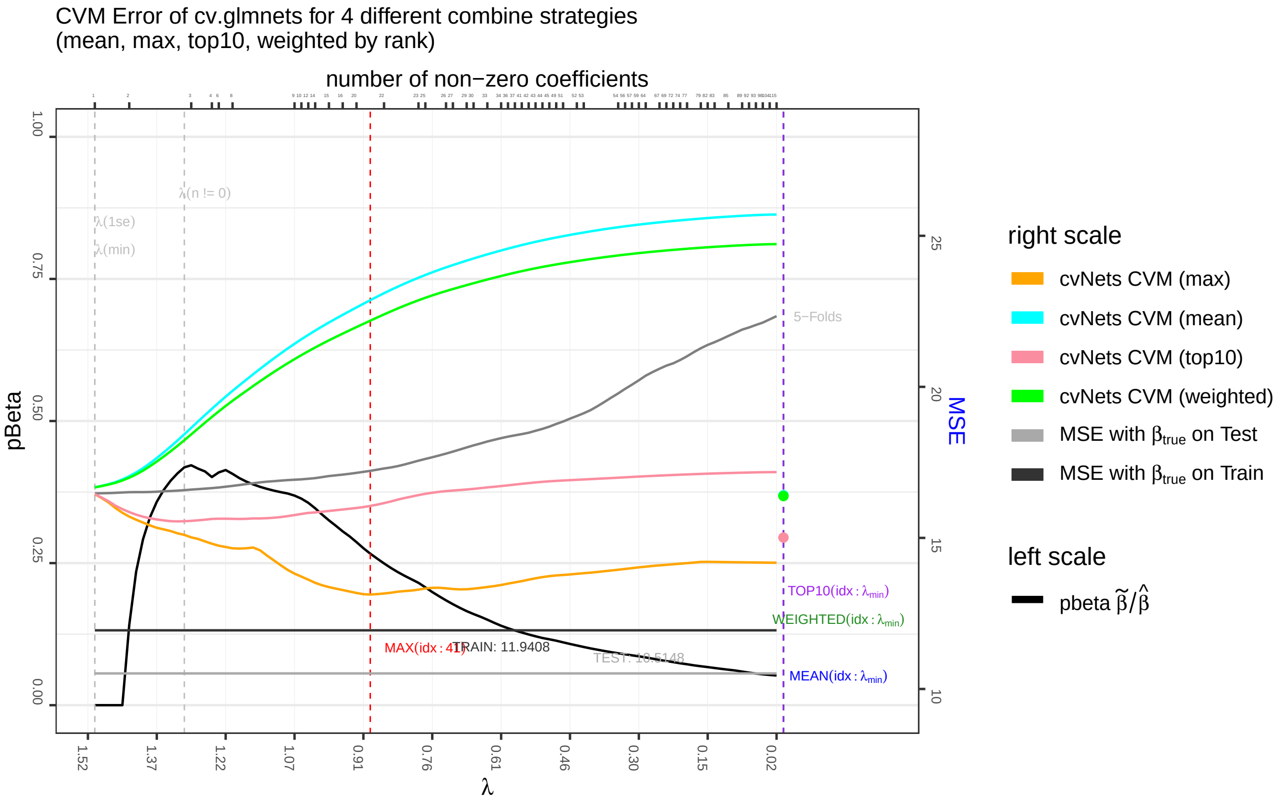


Figure 4.11: Same as figure 4.10 but only with the CVM errors of the ensembles shown.

In figure 4.11, the CVM error for all four ensembles is plotted across different λ indices, showing how λ_{best} is selected during training. The max-ensemble reaches its minimum CVM error at the 41st λ index, confirming the earlier observation. The other ensembles, however, have their λ_{best} at the 101st index, meaning their base models consist of model coefficients at λ_{min}. Table 4.4 (a) lists the CVM errors for the mean-ensemble at selected λ indices (full table in the appendix 5). It confirms that the lowest CVM error occurs at $\lambda_{best} = \lambda_{min}$ with a value of 16.41, which is lower than $\lambda_1 = 16.67$ and the CVM error curve of mean-ensemble in figure 4.11 shows a monotonic behavior with a minimum at the highest λ value. In the same table, the top base models with the lowest CVM errors are listed, starting with the 33rd, 10th, 73rd, 60th and 99th model. For the max-ensemble in table 4.4 (b), the CVM error at the 41st index is 13.13, identical to the CVM error at λ_{min}. This consistency is expected since the max-ensemble's λ_{best} will always match the λ_{min} of the best-performing base model, which, in this case, is the 33rd base model.

	λ_1	λ_2	λ_{100}	λ_{min}	λ_{1se}
CVM	16.67401929	16.72328362	25.70427969	16.4134412	16.66123331
best λ index	0	0	0	101	0
basemodels	m85, m66, m73, m94, m24	m85,m66,m73, m94, m7	m10, m33, m65, m27, m52	m33, m10, m73, m60, m99	m33, m85, m66, m73, m94
(Intercept)	0.408516957	0.408980247	0.392549827	0.403618319	0.408130665
X1	0	0.013746214	0.24912278	0.144240951	0.008326053
X2	0	-0.002827676	-0.086948977	-0.029047586	-0.003484713
X3	0	0	0.000152203	0	0
X4	0	0	$-1.67\text{E}-06$	0	0
X5	0	0	0.003863821	0	0
X6	0	0	0	0	0
X7	0	0	0	0	0
X8	0	0	0.006041621	0	0
X9	0	0	0	0	0
X10	0	0	-0.000583277	0	0
X11	0	0	-0.001927113	0	0
X12	0	0	-0.046180153	-0.000416036	0
...	...	...	...	...	...

mean-ensemble (a)

	$lambda_1$	$lambda_{41}$	$lambda_{100}$	λ_{min}	λ_{1se}
CVM	16.44510111	13.12579551	14.17898265	13.12579551	15.18112798
best λ index	0	41	0	0	0
basemodels	model85	model33	model10	model33	model33
(Intercept)	0.408516957	0.250201105	0.471104812	0.250201105	0.369887793
X1	0	0.893667948	1.131563989	0.893667948	0.832605318
X2	0	-0.29360801	-0.6725785996	-0.29360801	-0.3484713
X3	0	0	0	0	0
X4	0	0	0	0	0
X5	0	0	0	0	0
X6	0	0	0	0	0
X7	0	0	0	0	0
X8	0	0	0	0	0
X9	0	0	0	0	0
X10	0	0	0	0	0
X11	0	0	0	0	0
...	...	...	...	...	...

max-ensemble (b)

Table 4.4: Table shows mean-ensembles (a) and max-ensembles (b) for some of the 102 λ indices (with 2 non-zero true model coefficients) and their respective CVM error. Row "best λ idx" shows a non-zero entry at the column with the lowest CVM error, i.e. λ_{best}.

As expected, training error consistently decreases with reduced regularization, as shown in figure 4.10. To focus on the more relevant data, the training error has been removed, and only the test error is highlighted in figure 4.12.

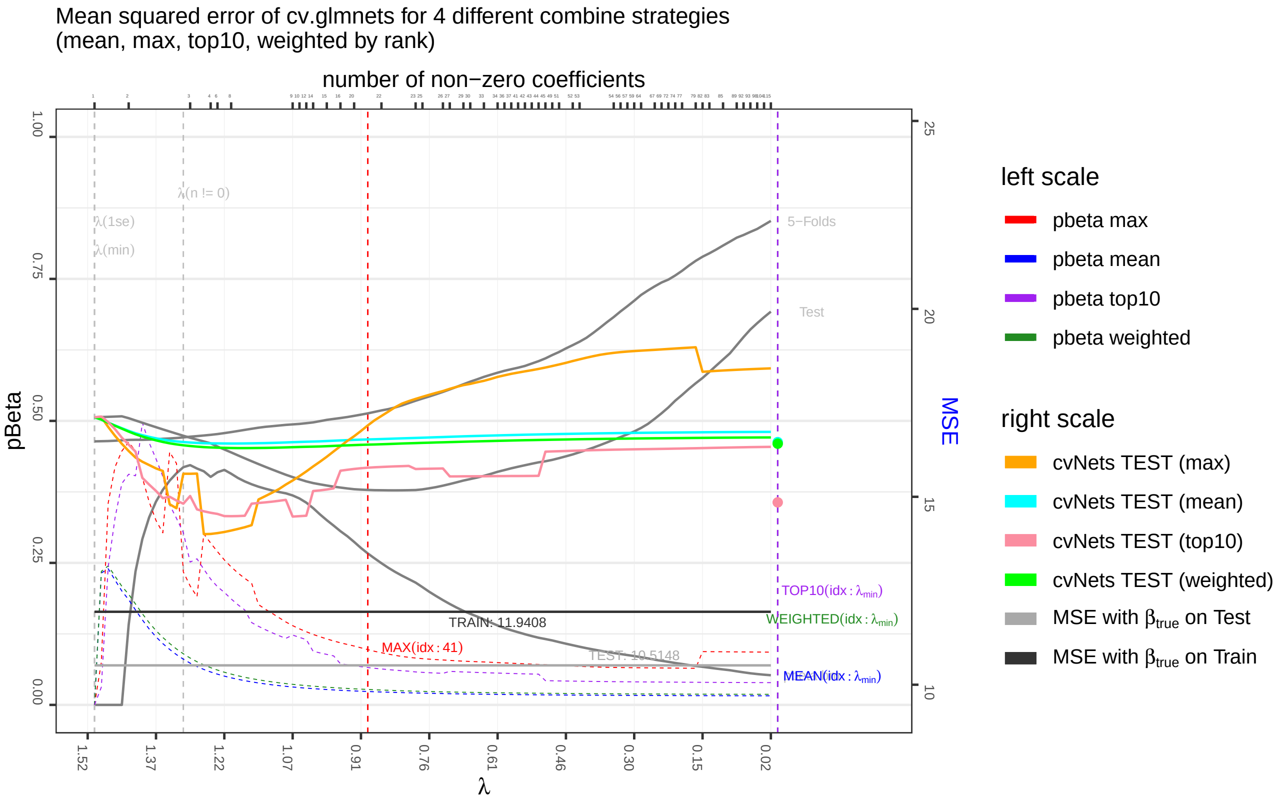


Figure 4.12: Figure compares MSE of *cv.glmnets* with Full-Feature LASSO model (grey in background) for 4096 number of features on test data. Additionally the *pBeta* values of the ensembles are shown for every λ index except λ_{min} and λ_{1se}, which would correspond to the 101st and 102nd λ indices. These can be seen in the table *Best ensembles* 4.5 in row *pBeta*.

Figure 4.12 compares the performance of LASSO ensembles on a test set to the Full-Feature LASSO model with 4096 features (figure 4.6), which is shown in grey in the background. Additionally, the *pBeta* value for all ensembles is displayed on the left-hand side scale. The test error curves for the mean- and weighted-ensembles are smooth, while the curves for the max- and top10-ensembles are more erratic. This difference is likely due to the number of models each ensemble comprises: mean- and weighted-ensembles consist of 100 base models each, which leads to smoother averaging of β values and predictions. In contrast, the max- and top10-ensembles have only 1 or 10 base models, respectively, making them more susceptible to fluctuations in test error as the λ value changes. For instance, when the max-ensemble shows a jump in test error, it indicates a switch to a different base model with a lower cross-validation mean error at that particular λ index. The test error for the Full-Feature LASSO model with 4096 features is 17.12 for both λ_{min} and λ_{1se}, as calculated in table 4.3, since it represents the intercept-only model in both cases. In figure 4.12, all ensembles exhibit a lower test error than the Full-Feature LASSO model at both λ_{min} and λ_{1se}, with the top10-ensemble even outperforming the Full-Feature LASSO model at every λ step. In summary, for 4096 number of features on the simulated data test set with $\beta_{true} = [2; -1]$, top10-ensemble (pink) performs best, weighted- (light green) closely followed by mean-ensemble (light blue) perform second best, surprisingly max-ensemble (orange) has the worst ensemble test performance but is still slightly better than the Full-Feature LASSO model at λ_{min} (grey vertical line at far left).

Table 4.5 shows the top base models of all ensembles at λ_{best}. Since $\lambda_{best} = \lambda_{min}$ for every combine strategy the top base models are the same for all ensembles as well, specifically 33, 10, 73, 60 and 99. As already mentioned, λ_{best} for max-ensemble is also λ_{min}.

	mean	max	weighted by rank	top10
non-zero	327	62	327	221
CVM	16.4134412	13.12579551	16.38621009	15.01095807
λ_{best}	101	41	101	101
pBeta	0.151	0.097	0.149	0.134
basemodels	m33, m10, m73, m60, m99	model 33	m33, m10, m73, m60, m99	33, 10, 73, 60, 99, 52, 35, 85, 22, 94
(Intercept)	0.403618319	0.250201105	0.403159542	0.401200106
X1	0.144240951	0.893667948	0.153098811	0.65508931
X2	-0.029047586	-0.29360801	-0.03172973	-0.180831958
X3	0	0	0	0
X4	0	0	0	0
X5	0	0	0	0
X6	0	0	0	0
X7	0	0	0	0
X8	0	0	0	0
X9	0	0	0	0
X10	0	0	0	0
X11	0	0	0	0
X12	-0.000416036	0	-0.000420337	0
X13	0	0	0	0
X14	0	0	0	0
X15	-0.028623346	-0.360987768	-0.030667877	-0.159296791
X16	0	0	0	0
X17	0	0	0	0
X18	0	0	0	0
X19	0	0	0	0
X20	0	0	0	0
...	...	...	...	...

Table 4.5: Table 4.5 presents all ensembles at λ_{best}, including their CVM error, top base models (truncated for readability for the mean- and weighted-ensembles), and the beginning of their β signature. Due to the potential for each ensemble to have up to 4096 non-zero model coefficients, the table is abbreviated, with the full version available in the appendix 5 under ”Best ensembles” (for 4096 number of features).

Figure 4.13 presents a heatmap of the feature subsets selected by each base model. To keep the figure manageable, it only includes the β_{true} and first β coefficients that the model identifies early on, as depicted in figure 4.7. All the top base models with the lowest CVM error - 33, 10, 73, 60, and 99 - have identified feature X_1. Both base models 33 and 60 have correctly identified features X_1 and X_2, as well as the false feature X_{1983}, which has a high correlation with the target variable y (see figure 4.9). Base models 10 and 99 have also identified X_1 and X_2, along with one additional false feature: X_{37} and X_{84}, respectively. Base model 73 identified X_{1983} instead of X_2. By chance, feature subsetting sometimes results in base models containing only true features (β_{true}) without highly correlating false positives. Unfortunately, base model 28, which correctly identified only X_1 and X_2 without any of the highly correlating false features, has a lower CVM error than the top base models mentioned above.

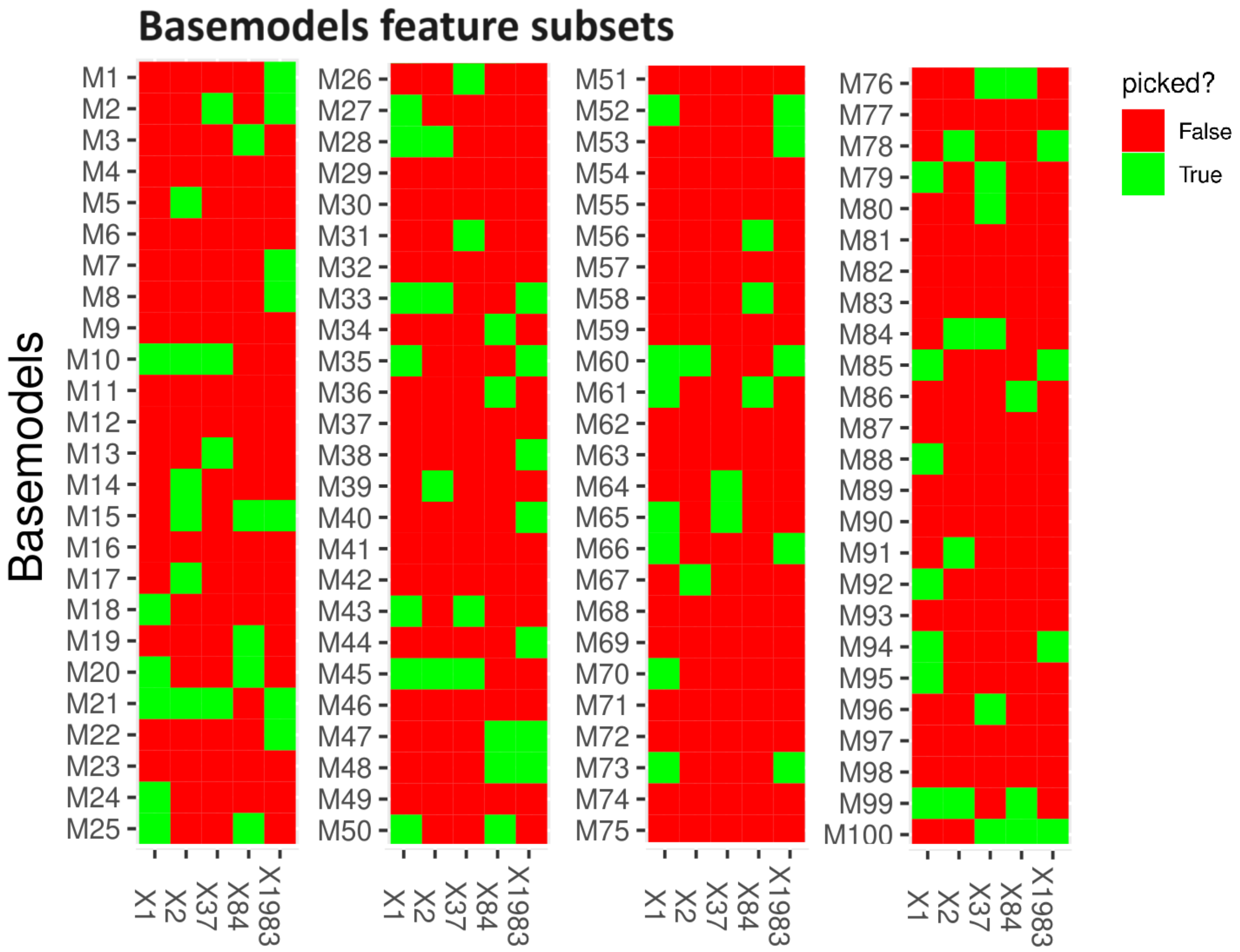


Figure 4.13: Heatmap shows *cv.glmnets* feature subsetting of all base models for 4096 number of features. Since for 4096 features the table would be too broad only real features and features that are picked early in figure 4.7 are shown.

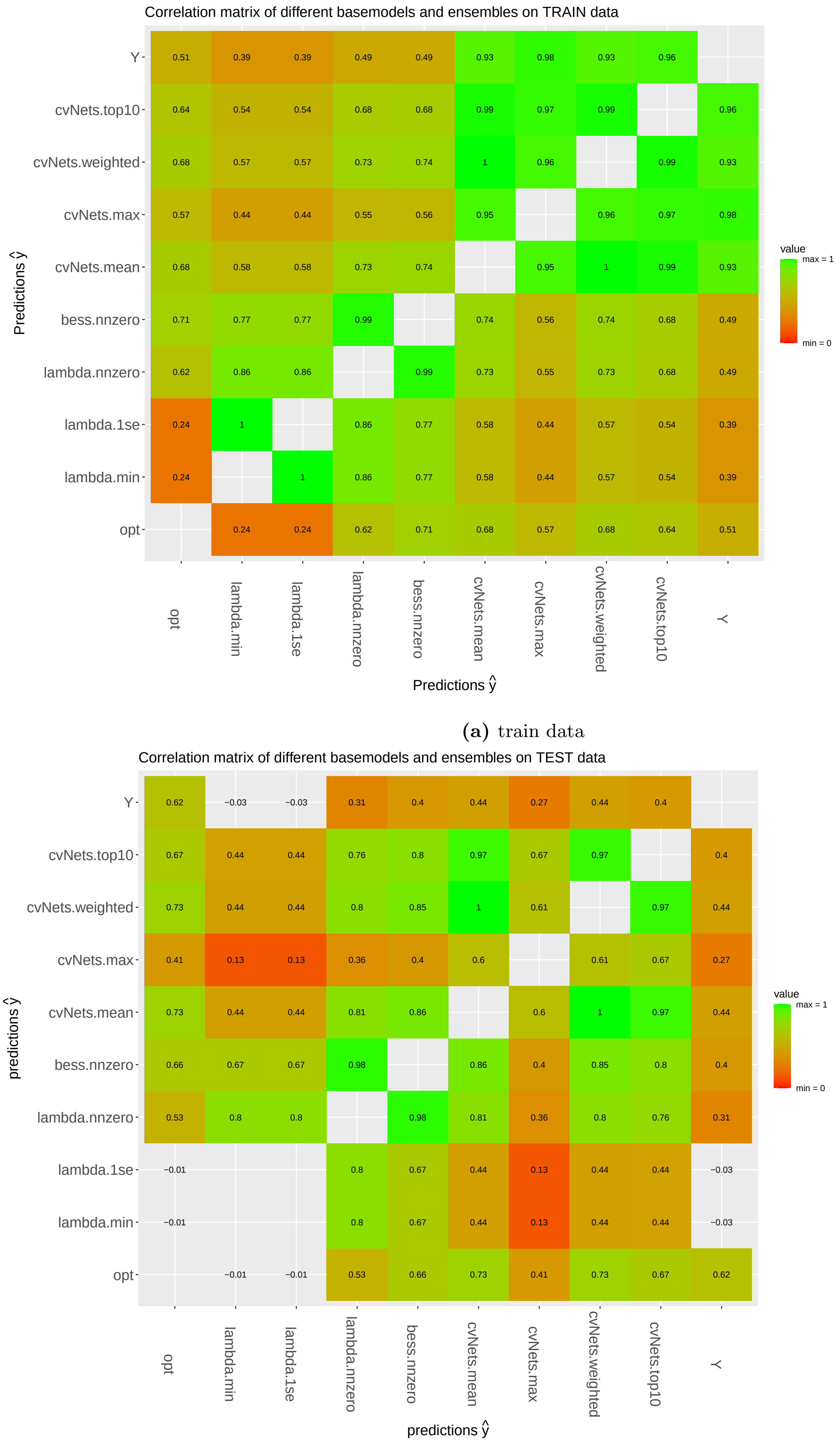
The probability that a base model with 4096 number of features and a feature subsetting size of 20% of 4096 draws by chance both X_1 and X_2 or one of those features is respectively

$$P(X_1 \text{ and } X_2) = \frac{\binom{2}{2} \cdot \binom{4094}{817}}{\binom{4096}{819}} \approx 4\% \tag{4.1}$$

$$P(X_1 \text{ or } X_2) = \frac{\binom{2}{1} \cdot \binom{4094}{818}}{\binom{4096}{819}} \approx 32\% \tag{4.2}$$

with $20\% \cdot 4096 \approx 819$. According to equation 4.1, with 100 base models, only 4 are expected to include both X_1 and X_2 in their feature subsets. However, in figure 4.13, there are actually 7 base models, namely 10, 21, 28, 33, 45, 60 and 99, that contain both features, suggesting that the LASSO ensembles might be performing better than statistically expected. For subsets containing either X_1 or X_2 (but not both), the expected number of base models is 32, whereas figure 4.13 shows 29. Overall, the probability of an ensemble including one or more true features is $4\% + 32\% = 36\%$. This highlights the idea of ensembles: combining many weak learners (base models with only one true feature) to enhance overall performance. However, base model 33, with the lowest CVM error at $\lambda_{best} = \lambda_{min}$, has a $pBeta = 0.0973$ - better than the $pBeta = 0$ of the Full-Feature LASSO model at λ_{min}, but still not ideal. Ensemble $pBeta$ values are similar (see table 4.5), ranging between 10% and 15%. In conclusion, for 4096 number of features feature subsetting helps lowering the test error in figure 4.12, however since there are so many features available to the model during training there will always be false features, that correlate highly with y, even if the highest correlating features are randomly removed from the feature space. Otherwise base model 28 would perform better. This can be confirmed by looking at $pBeta$ and the number of *non-zero* model coefficients in table 4.5 of the ensembles at λ_{best}. A lot of false positive features are picked even by the max-ensembles, which results in a poor $pBeta$ value. If it were not for the high correlation with y in training (see figure 4.9) LASSO ensemble would likely perform better.

Instead of using MSE, figure 4.14 compares the correlation between the predicted values $\hat{y}$ and the target variable y for the Full-Feature LASSO model at different λ values and for ensembles at λ_{best}. On the training set, all ensembles show significant overfitting, with correlations close to 1, but this drops sharply on the test set. Notably, the top10-ensemble, which had the lowest test error in figure 4.12, shows a slightly lower correlation with y (0.40) on the test set compared to the mean-ensemble and weighted-ensemble (both at 0.44), despite having the best test MSE. While MSE and correlation are related, they measure different aspects: MSE captures the average squared distance between y and $\hat{y}$ along the y-axis, whereas correlation measures the degree to which y and $\hat{y}$ co-vary.



(a) train data

(b) test data

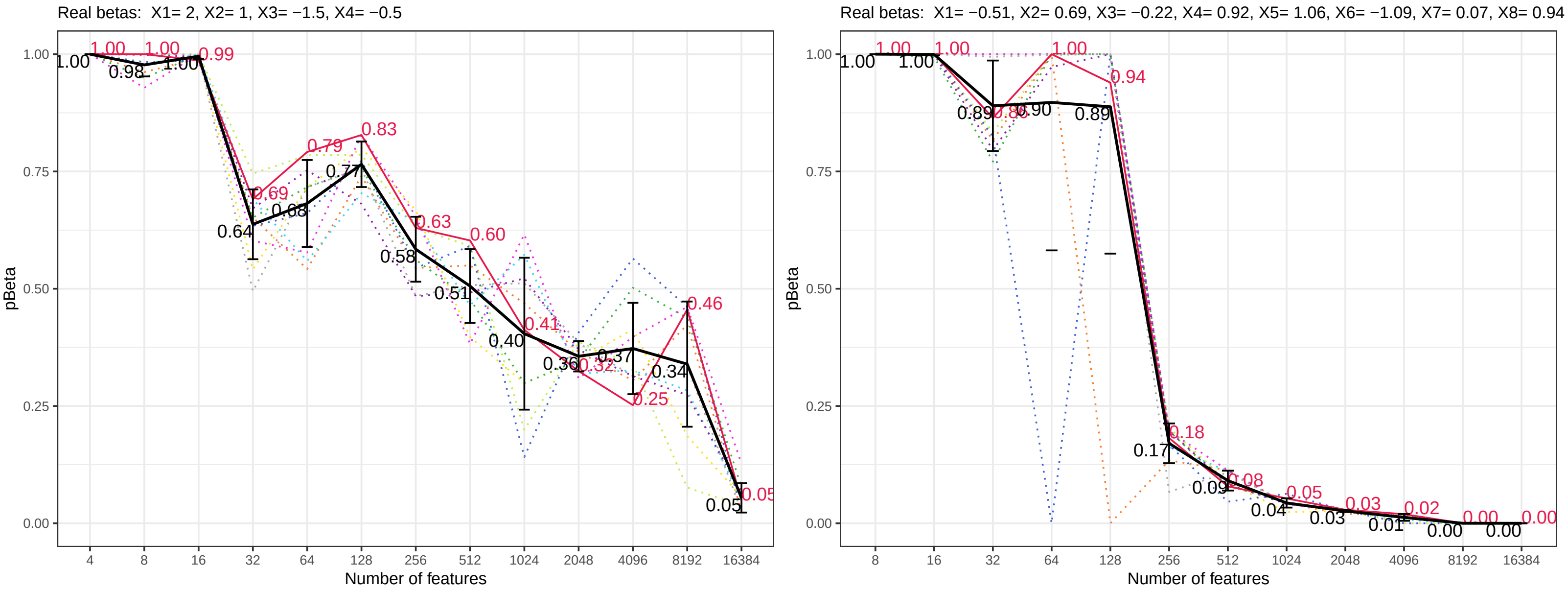
Figure 4.14: Two correlation heatmaps for train (a) and test data (b). The correlation is calculated between different estimated predictions $\hat{y}$ of base models at λ_{min}, λ_{1se}, $\lambda_{n\neq0}$, $BeSS.One$ model, every ensembles at λ_{best} and the target variable y.

4.2.1 New Model Configurations

Up to this point, the focus has been on models with two non-zero real model coefficients ($\beta_{true}[1,2] = [2;-1]$). Next, the analysis will be extended to cases with 4 and 8 true non-zero model coefficients. Finally, the LAMIS cohort dataset, introduced in chapter 3, will be utilized to explore models with 2, 4, and 8 non-zero coefficients, providing insights into the ensembles' performance on real-world data.

4 and 8 Real Coefficients

For 4 non-zero real coefficients $\beta_{true} = [2; 1; -1.5; -0.5]$ was selected to be small enough to ensure that for high number of features the model would have a low *pBeta* since this is the region *cv.glmnets* hopes to improve upon. Similarly, for 8 non-zero real model coefficients $\beta_{true} = \left[-0.51;\ 0.69;\ -0.22;\ 0.92;\ 1.06;\ -1.09;\ 0.07;\ 0.94\right]$ was chosen. More precisely, a random value betweeen -1.2 and 1.2 was sampled. In figure 4.15 for 16384 number of features *pBeta* values are $pBeta = 0.05$ (a) and $pBeta = 0$ (b) for 4 and 8 real model coefficients respectively. Therefore, figures and tables in this chapter exclusively show LASSO models for 16384 number of features since there *pBeta* is close the 0 in both cases.

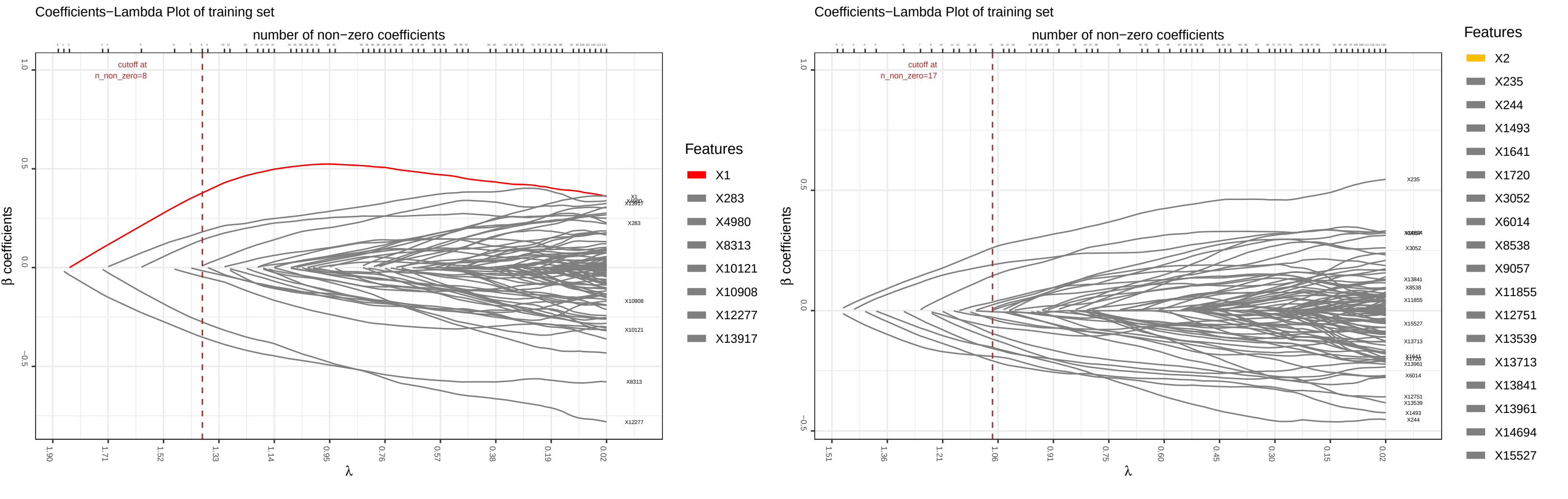


(a) 4 non-zero β_{true} coefficients

(b) 8 non-zero β_{true} coefficient

Figure 4.15: *pBeta* for non-zero $\beta_{true} = \left[2;\ 1;\ -1.5;\ -0.5\right]$ (a) and non-zero $\beta_{true} = \left[-0.51;\ 0.69;\ -0.22;\ 0.92;\ 1.06;\ -1.09;\ 0.07;\ 0.94\right]$ (b) on the simulated dataset shows Signal Drowning at approximately 16384 number of features for both (a) and (b).

As the number of real model coefficients in β_{true} increases, each coefficient's contribution to the overall prediction y diminishes proportionally. In models with 4 or 8 real model coefficients, each feature's effect strength is reduced compared to models with only 2 real coefficients. This reduction makes true features more easily replaceable by false positives. However, the replaceability of individual coefficients depends primarily on their magnitude and the number of available features. In the Full-Feature LASSO model with 4 real coefficients, which is comparable in magnitude to those with 2 real coefficients but with four times as many features, only X_1 (the true feature with the highest absolute value) is identified by the model, as shown in figure 4.16 (a). In the model with 8 real β coefficients, the model eventually identifies X_2, but it occurs so late that it is not visible in figure 4.16 (b).

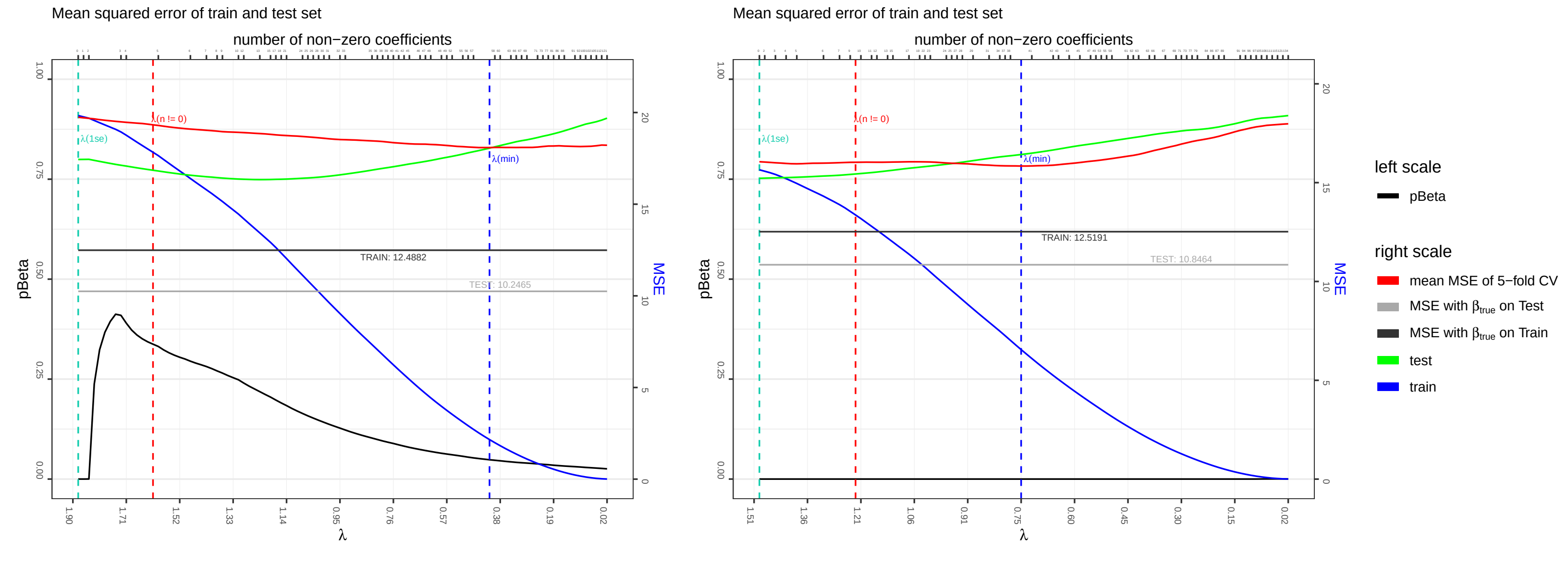


(a) 4 real coefficients

(b) 8 real coefficients

Figure 4.16: β coefficient magnitude of Full-Feature LASSO models with 4 (a) and 8 (b) true model coefficients at 16384 number of features for the simulated dataset respectively.

Figure 4.17 illustrates the training, testing, and cross-validation errors of the Full-Feature LASSO model with 16384 features, for cases with 4 (a) and 8 (b) non-zero true β coefficients. Unlike the Full-Feature LASSO model with 2 true β coefficients, the models here do not default to the intercept-only model at λ_{min}. However, the *pBeta* value is still very low for the model with 4 true β coefficients in (a) and is zero for the model with 8 true β coefficients in (b) at λ_{min}. If the model in (a) is constrained to select only 4 true coefficients at $\lambda_{n\neq0}$, then *pBeta* improves to approximately 0.33.

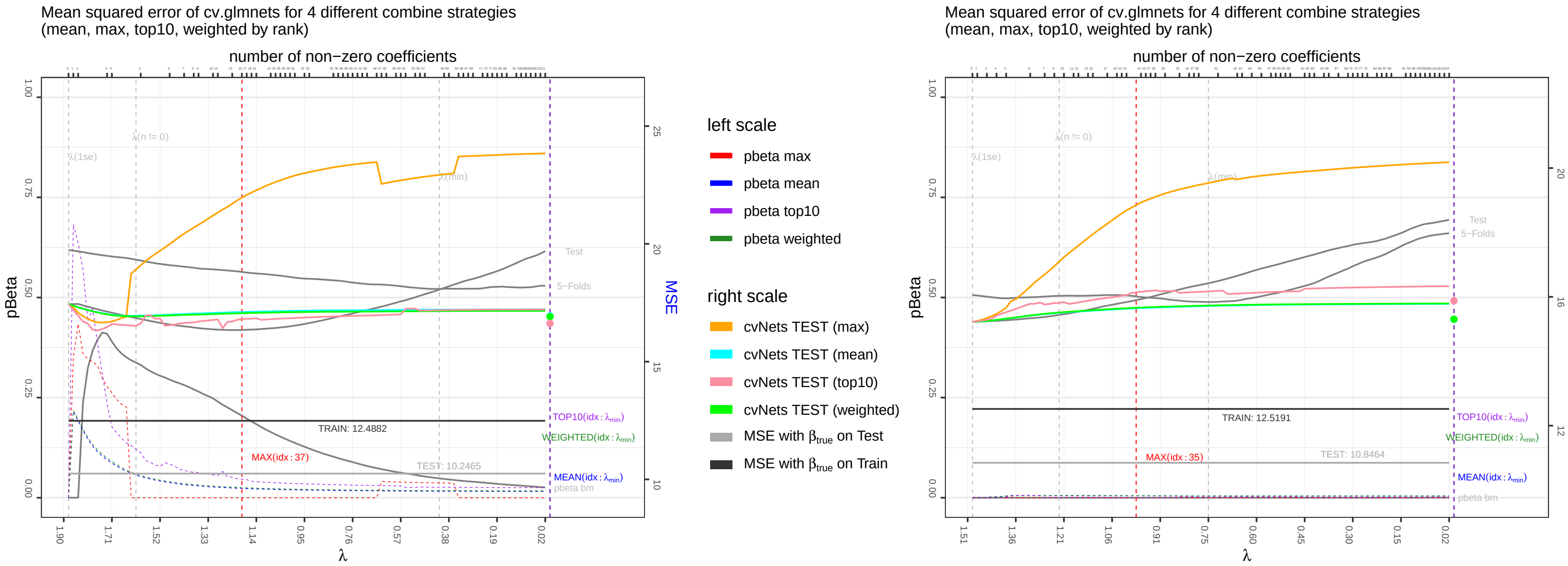


(a) 4 real coefficients

(b) 8 real coefficients

Figure 4.17: Train, test and cross-validation mean error (CVM) of LASSO base model with 4 (a) and 8 true beta coefficients (b) for 16384 number of features. The vertical lines show different λ values, i.e. λ_{min}, λ_{1se} and $\lambda_{n\neq0}$.

For the case of 16384 features, the test error behavior in figure 4.18 for LASSO ensembles with 4 and 8 non-zero β_{true} coefficients is similar to that observed in the ensembles with 2 real β_{true} coefficients shown in figure 4.12. Best lambda λ_{best} is again λ_{min} at index 101, where max-ensemble has its λ_{best} by definition at λ_{min}, occuring at index 37 for 4 real model coefficients and 35 for 8 real model coefficients. Apparently, ensembles of base models at λ_{min} perform best on the cross-validation set. This is partly expected since λ_{min} has by definition the lowest CVM error. On the other hand, a higher regularization could produce a $pBeta \approx 0.4$ for 4 real coefficients in figure 4.18 (a). Therefore ensembles at a low λ index or λ_{1se} (index $= 102$) with a higher $pBeta$ value should ideally outperform the ensembles with a low $pBeta$ value in cross-validation. However, the correlation plot in figure 4.19 reveals that the false positive features, which are picked early by the model as shown in figure 4.16, maintain a high correlation with the target variable y on the training set again. This high correlation of false positives suggests that despite their incorrectness, these features are contributing significantly to the model's prediction, which complicates the process of identifying and relying on the true features even with higher regularization.



(a) 4 real coefficients

(b) 8 real coefficients

Figure 4.18: Comparison of Full-Feature LASSO model (grey in background) with ensembles for 4 (a) and 8 (b) real model coefficients with 16384 number of features on the simulated dataset.

Table 4.6 shows all ensembles at λ_{best}. For 4 real model coefficients (a) the order of best performing base models on the cross-validation set is given in row labeled base models. Since all four ensembles have $\lambda_{best} = \lambda_{min}$ (for max-ensemble λ index 37 is λ_{min}) the order of the base models does not change, i.e. base model 85, 42, 73, 12, 100, 26, 60, 69, 22 and 83 are the top 10 base models for λ_{min}.

In Figure 4.20 (a), base model 12 successfully identifies 3 out of the 4 real coefficients (X_1, X_2 and X_4), but it ranks only 4th in terms of the lowest cross-validation mean error among all base models. Interestingly, the highest-ranked base model, model 85, fails to identify any of the real features. This outcome highlights the increased difficulty that base models face when trying to identify correct features as the number of real coefficients and total features increases. As a result, feature subsetting becomes less effective when dealing with a larger number of features and a higher number of real model coefficients.

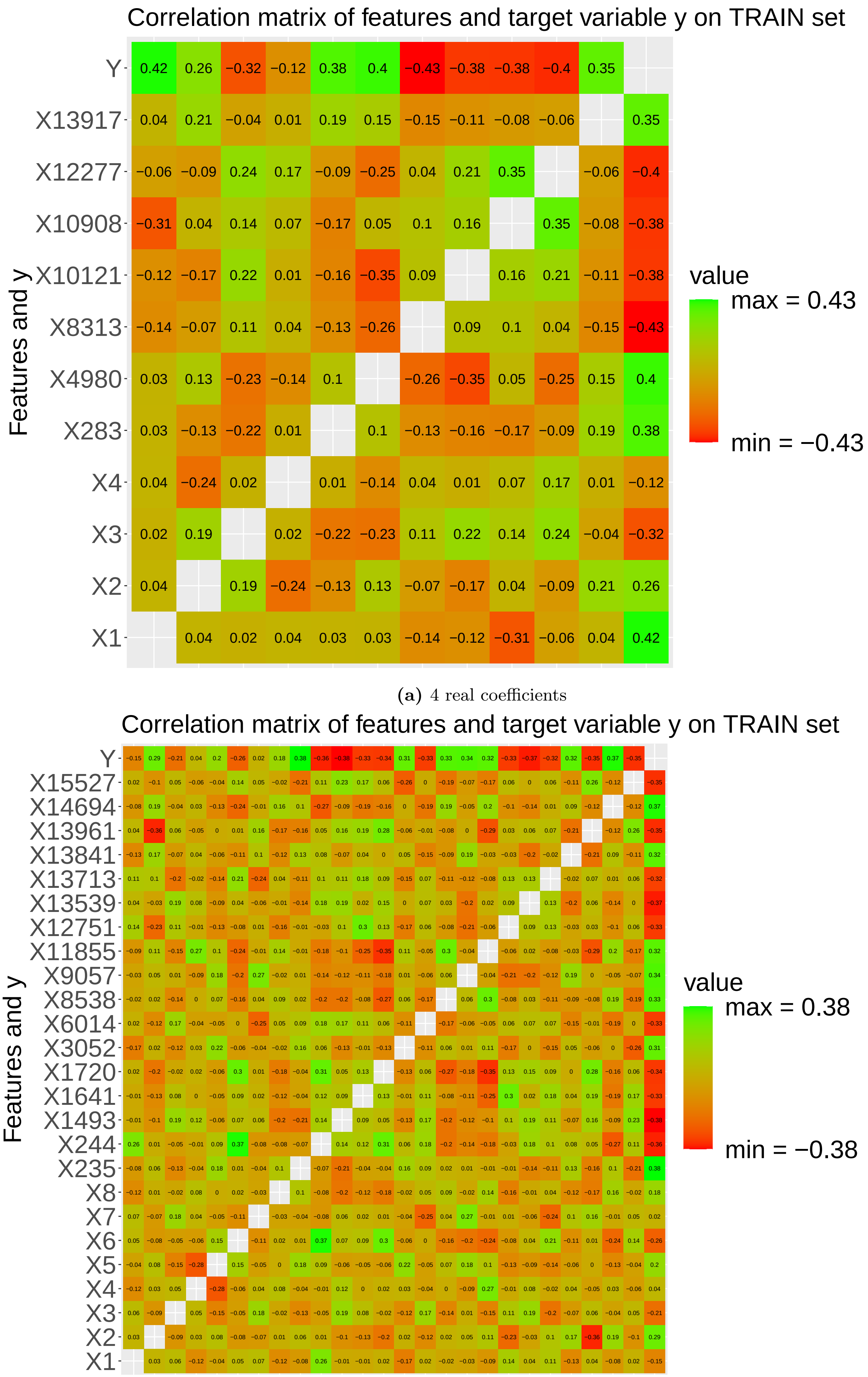
	mean	max	weighted	top10
non-zero	882	58	882	484
CVM	18.76	14.85	18.64	15.87
λ_{best}	101	37	101	101
pBeta	0.03	0	0.04	0.04
basemodels	m85, m42, m73, m12, m100	model 85	m85, m42, m73, m12, m100	85, 42, 73, 12, 100, 26, 60, 69, 22, 83
(Intercept)	-0.44	-0.49	-0.44	-0.52
X1	0.16	0	0.17	0.46
X2	0.01	0	0.01	0.01
X3	-0.04	0	-0.04	-0.04
X4	0	0	0	0
X5	0	0	0	0
X6	0	0	0	0
X7	0	0	0	0
X8	0	0	0	0
X9	0	0	0	0
X10	0	0	0	0
X11	0	0	0	0
X12	0	0	0	0
X13	0	0	0	0
...	...	...	...	...

4 coefficients (a)

	mean	max	weighted	top10
non-zero	532	64	532	247
CVM	15.42	12.16	15.39	13.92
λ_{best}	101	35	101	101
pBeta	0	0	0	0
basemodels	m90, m66, m21, m32, m1	model 90	m90, m66, m21, m32, m1	90, 66, 21, 32, 1, 24, 89, 78, 37, 41
(Intercept)	-0.07	0.1	-0.07	-0.09
X1	0	0	0	0
X2	0.00	0	0.00	0.00
X3	0	0	0	0
X4	0	0	0	0
X5	0	0	0	0
X6	0	0	0	-0.00
X7	0	0	0	0
X8	0	0	0	0
X9	0	0	0	0
X10	0	0	0	0
X11	0	0	0	0
X12	0	0	0	0
X13	0	0	0	0
...	...	...	...	...

8 coefficients (b)

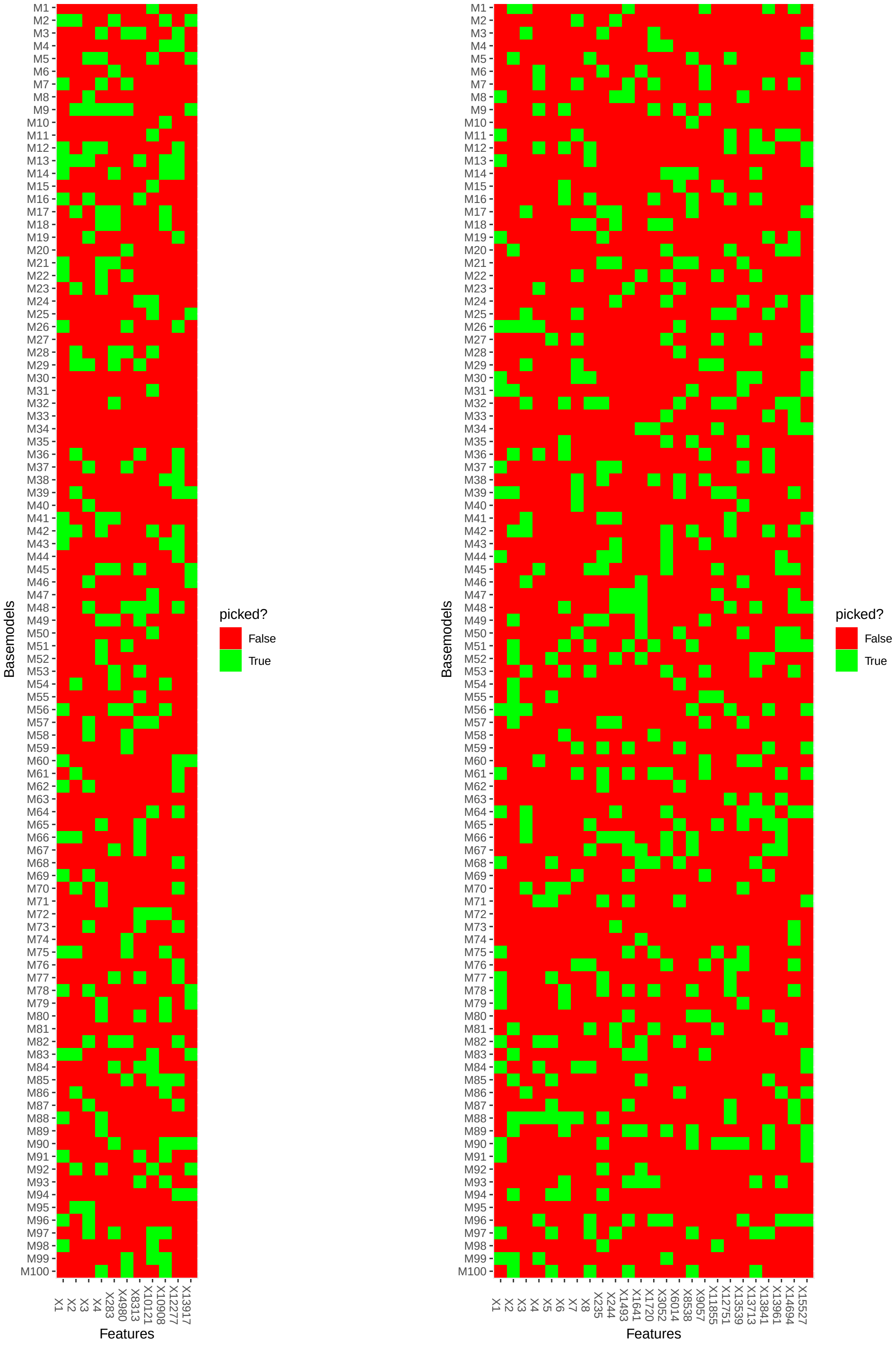
Table 4.6: All ensembles at λ_{best} for 4 (a) and 8 (b) real model coefficients respectively. Full table of model coefficients can be viewed in the appendix 5 under *Best ensembles*.



(a) 4 real coefficients

(b) 8 real coefficients

Figure 4.19: Heatmap of correlation true and false positive features with variable y. The color scale ranges from $\pm|corr_{max}|$.



(a) 4 real coefficients

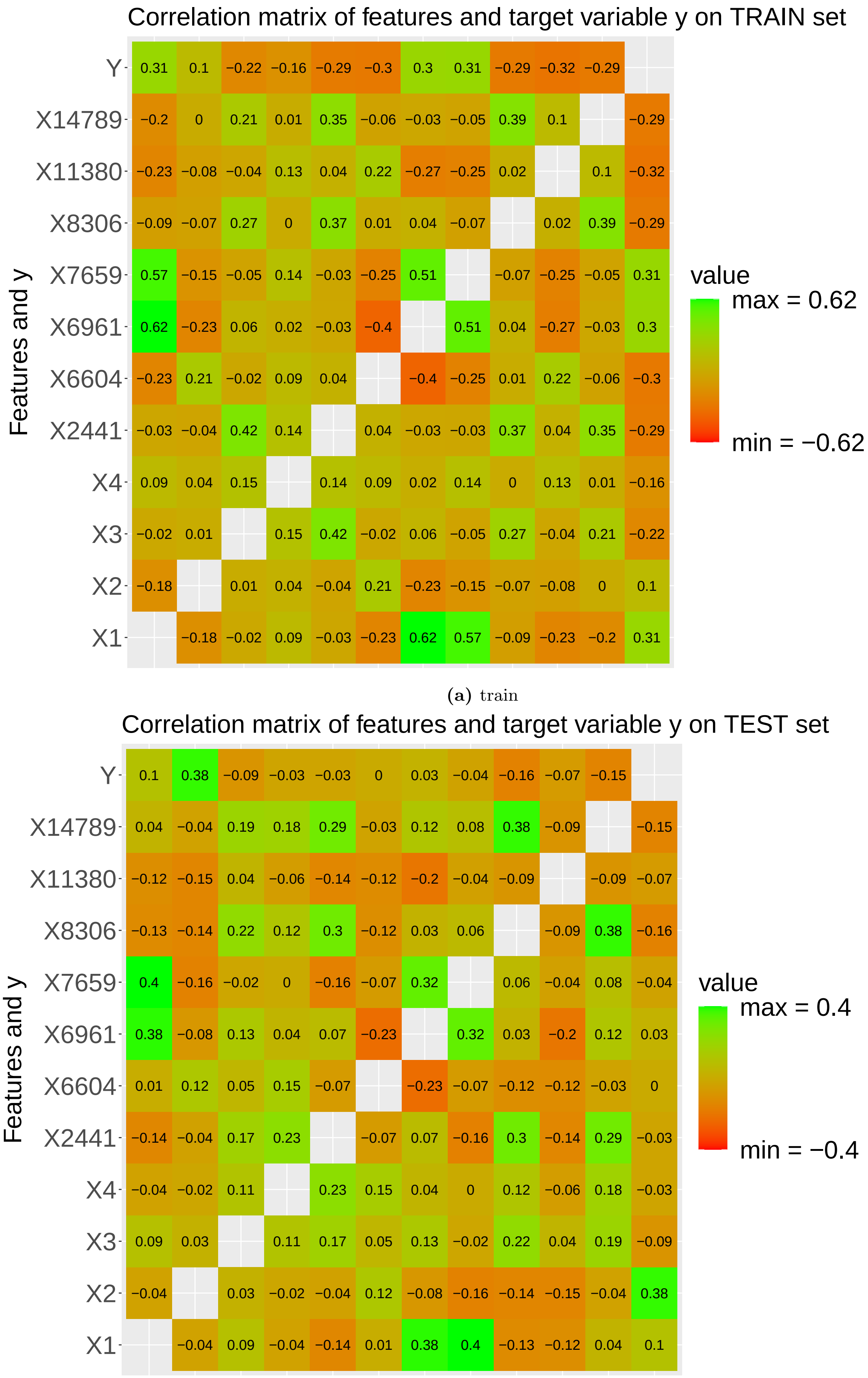
(b) 8 real coefficients

Figure 4.20: Heatmap for every base model showing if true feature or any of the high correlating false postive features are picked. Top base models for 4 real coefficients are 85, 42, 73, 12, 100, 26, 60, 69, 22 and 83. For 8 real coefficients they are 90, 66, 21, 32, 1, 24, 89, 78, 37 and 41.

CvNets with LAMIS data

In the LAMIS dataset, feature correlations can occur due to biological interactions, such as one gene influencing the presence of another. For this analysis, true model coefficients were assigned values of $\beta_{true} = \left[\, 1.4;\ -0.7 \,\right]$ for 2 non-zero coefficients, $\left[\, 1;\ 0.7;\ -0.8;\ -0.5 \,\right]$ for 4 non-zero coefficients, and $\left[\, -0.68;\ -0.06;\ -0.09;\ 0.13;\ -0.16;\ 0.76;\ -0.46;\ 0.71 \,\right]$ for 8 non-zero coefficients. Interestingly, the LAMIS data required these β_{true} coefficients to be smaller in magnitude to achieve $pBeta \approx 0$, compared to the simulated dataset.

Figure 4.21 illustrates the feature correlation for the LAMIS cohort with 4 real coefficients at 16384 features, showing results for both the training and test set. While false features performed better on the training set than on the test set, they do not perform as well as in the simulated dataset (as seen in figure 4.9). In the simulated data, a false positive feature often showed the highest correlation with the target variable y, whereas in the LAMIS dataset, X_1 exhibited strong correlations with X_{6961} and X_{7659} across both training and test sets, possibly due to underlying gene dependencies. This correlation suggests that X_1 could easily be substituted by X_{6961} or X_{7659}, making it surprising that $pBeta \approx 0$ was reached later than expected, i.e. requiring smaller β_{true} coefficients to reach $pBeta \approx 0$ sooner. The remaining plots for the LAMIS dataset exhibit similar patterns to those discussed earlier and are included in the appendix 5 report.



(a) train

(b) test

Figure 4.21: Feature correlation with 4 real coefficients at 16384 number of features for train (a) and test (b) set on the LAMIS cohort.

4.2.2 LASSO Ensembles conclusion

In comparing LASSO models with 2, 4, and 8 real coefficients for both simulated and LAMIS datasets, six configurations were evaluated by examining the test errors of the Full-Feature LASSO model at λ_{min} from *cv.glmnet* against ensembles at λ_{best} from *cv.glmnets*. Figure 4.22 presents these test errors across all configurations. The number of features in each configuration was selected to achieve a $pBeta \approx 0$ for the Full-Feature LASSO model. For the first configuration ($\beta_{true}[1,2] = [2;-1]$), the LASSO ensemble did not significantly outperform the Full-Feature LASSO model, and the $pBeta$ value remained low. Among the six configurations, the top10-ensemble yielded the lowest test MSE in three instances, while the weighted-ensemble was the best in two configurations. In one configuration, the Full-Feature LASSO model achieved the lowest test error. Interestingly, mean- and weighted-ensemble perform almost identically over all 6 configurations. In the LAMIS dataset, Full-Feature LASSO model performed similarly to the ensembles, with LAMIS showing slightly lower test errors across all three configurations, despite having the same signal-to-noise ratio as the simulated dataset. This indicates that although LASSO ensembles can offer some improvements, their advantages are modest and vary depending on the dataset.

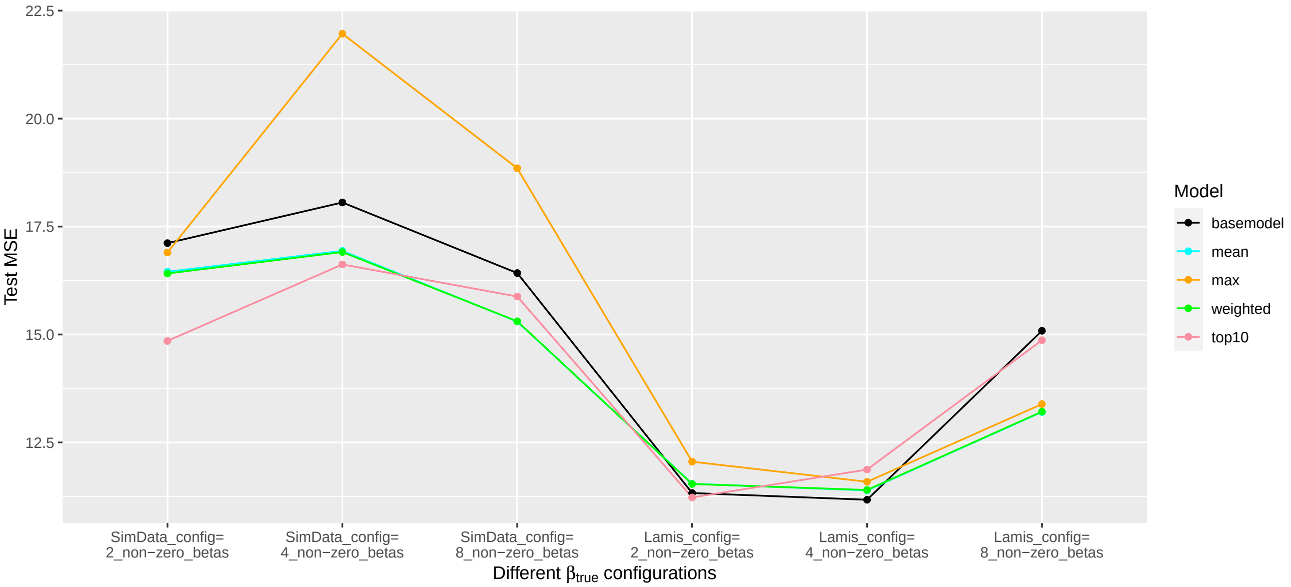


Figure 4.22: The figure compares Full-Feature LASSO models at λ_{min} and ensembles at λ_{best} across various configurations, which combine different datasets (a simulated dataset and the real LAMIS dataset) and varying numbers of real, non-zero model coefficients. The x-axis represents these configurations, while the corresponding number of features - 4096, 16384, 16384, 8192, 16384, and 4096, from left to right - are selected based on when $pBeta \approx 0$ first occurs for each setup (as illustrated in figures 4.1 and 4.15).

5. Discussion

This thesis explored how the LASSO algorithm finds its solutions and found that, for the tested simulated and biological datasets, LASSO ensembles did not outperform the Full-Feature LASSO model. However, LASSO ensembles may still be effective on other datasets, especially those with lower noise levels. The high signal-to-noise ratio used to achieve $pBeta \approx 0$ might be unrealistic for real-world scenarios. This is particularly noteworthy since Urda et al. [7] found LASSO ensembles to be superior to classic LASSO models in their breast cancer dataset. The LASSO ensemble algorithm's hyperparameter tuning was not rigorously tested in this study - for example, by increasing the number of base models from 100 to 500 or varying the feature subsetting from 20% to 50%, which was consistently fixed at 20%. The base models were combined based on their λ indices. However, exploring a more sophisticated alignment approach could potentially lead to improved results. In the LAMIS dataset, smaller β_{true} coefficients were required in comparison to the simulated dataset to achieve $pBeta \approx 0$, despite strong feature correlations that would suggest the opposite—that $pBeta$ should reach 0 more quickly. Future research could investigate how correlation structures in the data influence LASSO performance. Additionally, an algorithm that tests each feature as the starting point in *cv.glmnet* could improve upon single LASSO models by better identifying the correct combination of true coefficients as the active set. However, the challenge of false positive features is likely to persist even with this approach.

Appendix

Accompanying this thesis is an extensive appendix report containing figures and tables of all configurations generated by the main script in the GitLab repository [2]. The full appendix is available in three ways:

- via Google Cloud (public):
  `https://drive.google.com/drive/folders/1M0awtdG5XEKhcN9KuYERx9W_0fW2F_ob?usp=sharing`
- on the internal Spang compute server ”r4” (/data/lasso-ensembles)
- on the GitLab repository [2]
  `https://gitlab.spang-lab.de/bachelorthesis/ws2223_agloetzl_lasso-ensembles/-/tree/main/data`

For access details, please contact Tobias Schmidt (tobias2.schmidt@ukr.de), Michael Huttner (michael.huttner@ukr.de) or Christian Kohler (christian.kohler@ur.de).

Glossary

β_{true} The non-zero entries of β_{true} vector are the only contributors to the target variable y (see equation 3.1).. 9

base models Base models refers to the building block of LASSO ensembles. They are the classic LASSO model with feature subsetting, i.e. not all features are used for training.. 5

cv.glmnet Package for linear regularized regression. The resulting model returns the estimated model coefficients and λ_{min}. λ_{min} is determined in cross-validation on the test folds. Cv.glmnets calls cv.glmnet internally when creating the base models.. 5

cv.glmnets Cv.glmnets is the proposed package of this thesis in order to prevent "Signal Drowning". It creates by default 100 base models, which each contain different subsets of features. These base models are then combined by different strategies to max-, mean-, top10- and weighted-ensembles.. 5

CVM CVM stands for cross-validation mean error and is the mean error of the test folds in cross-validation, therefore simulating holdout test data, that the model has not seen before.. 5

ensemble In this thesis, LASSO ensembles can be built in four different ways from its base models:

- max-ensemble: ensemble consists only of best performing LASSO base model.
- mean-ensemble: ensemble consists of the average of every LASSO base model.
- top10-ensemble: ensemble consists of the top 10 best performing LASSO base models.
- weighted-ensemble: ensemble consists of every LASSO base model, giving more weight to those with lower CVM error (see equation 2.7.)

. 5

false positive False positive is an error in binary classification, e.g. a false positive feature is a feature, that is picked by the model during training but is not a non-zero entry of β_{true}.. 20

Full-Feature LASSO Full-Feature LASSO refers to the original LASSO model from *cv.glmnet* without feature subsetting. The goal of the ensembles is to beat the Full-Feature LASSO model, i.e. have a lower test MSE.. 22

LASSO LASSO is a algorithm for regularized linear regression with L1 penalty. It is an acronym for Least Absolute Shrinkage and Selection Operator.. 2

pBeta pBeta describes the ratio of $\|\tilde{\beta}\|$ to $\|\hat{\beta}\|$, where $\hat{\beta}$ denotes the vector of "non-zero coefficient estimates" and $\tilde{\beta}$ denotes the vector of "true non-zero coefficient estimates".. 3

Signal Drowning For increasing number of available features and with a low signal-to-noise ratio, the package *cv.glmnet* has a higher chance of not finding the correct solution anymore. The correct solution in this context means, that β_{true} is not estimated correctly by the model and consequently *pBeta* is close to zero.. 4

List of Figures

List of Tables

Bibliography

[1] Ilan Adler and Sheldon M. Ross. The coupon subset collection problem. J. Appl. Prob. 38, 737–746, 2001. University of California, Berkeley.

[2] Glötzl Alexander. Accompanying repository for creating the report of figures. `https://gitlab.spang-lab.de/bachelorthesis/ws2223_agloetzl_lasso-ensembles`, 2024. University of Regensburg.

[3] Glötzl Alexander and Schmidt Tobias. Lasso ensembles. `https://gitlab.spang-lab.de/sct39258/glmnets`, 2024. University of Regensburg.

[4] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning.* Springer, 2008.

[5] R. R. Hocking and R. N. Leslie. *Selection of the Best Subset in Regression Analysis.* Technometrics , Nov. 1967, Vol. 9, No. 4 (Nov., 1967), pp. 531-540, 1967.

[6] Staiger, A.M., Altenbuchinger, M., Ziepert, and M. et al. A novel lymphoma-associated macrophage interaction signature (lamis) provides robust risk prognostication in diffuse large b-cell lymphoma clinical trial cohorts of the dshnhl. `https://doi.org/10.1038/s41375-019-0573-y`, 2020.

[7] Daniel Urda, Leonardo Franco, and Jose M. Jerez. Classification of high dimensional data using lasso ensembles, 2017.

Acknowledgments

Ein großes Dankeschön geht an meinen Betreuer Tobias Schmidt. Unzählige Male haben seine Erklärungen und Skizzen mir bei dieser Arbeit weitergeholfen. Nicht nur auf Bezug zu LASSO und Maschinellem Lernen, sondern generell wie man ein besserer Programmierer wird.

Ein großes Dankeschön gilt auch Rainer für die Ausrichtung und Zielsetzung vorallem am Anfang der Arbeit. Nicht nur die fachliche Kompetenz, sondern auch die lockere Art während unserer Treffen wird mir in Erinnerung bleiben.

Ich bin immer gerne ans Institut an meinen Arbeitsplatz gekommen und das lag hauptsächlich am herzlichen Lehrstuhl, mit dem es immer etwas zu quatschen gab. Also Danke Tobi, Michi, Lena, Linda, Andreas, Zahra, Paul, Claudio, Rainer, Sharon, Christian, Lukas und Lukas für die schöne Zeit.

Erklärung zur Bachelorarbeit

Ich habe die Arbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und bisher keiner anderen Prüfungsbehörde vorgelegt. Außerdem bestätige ich hiermit, dass die vorgelegten Druckexemplare und die vorgelegte elektronische Version der Arbeit identisch sind, dass ich über wissenschaftlich korrektes Arbeiten und Zitieren aufgeklärt wurde und dass ich von den in § 27 Abs. 5 vorgesehenen Rechtsfolgen Kenntnis habe.

Regensburg, den 22. August 2024

....
Alexander Glötzl