## Of Zeros and Ones

Rauschenberger, A.

2021

***document version***
Publisher's PDF, also known as Version of record

# Chapter 6

# Stacked elastic net

This chapter is based on "Pre- dictive and interpretable models via the stacked elastic net" (Rauschenberger et al., 2020a).

Rauschenberger, A., Glaab, E. & van de Wiel, M.A. (2020). "Predictive and interpretable models via the stacked elastic net." *Bioinformatics*, btaa535. doi: 10.1093/bioinformatics/btaa535.

R package `starnet`: "Stacked Elastic Net".
`https://CRAN.R-project.org/package=starnet`
`https://github.com/rauschenberger/starnet`

**Machine learning in the biomedical sciences should ideally provide predictive and interpretable models. When predicting outcomes from clinical or molecular features, applied researchers often want to know which features have effects, whether these effects are positive or negative, and how strong these effects are. Regression analysis includes this information in the coefficients but typically renders less predictive models than more advanced machine learning techniques. Here we propose an interpretable meta-learning approach for high-dimensional regression. The elastic net provides a compromise between estimating weak effects for many features and strong effects for some features. It has a mixing parameter to weight between ridge and lasso regularisation. Instead of selecting one weighting by tuning, we combine multiple weightings by stacking. We do this in a way that increases predictivity without sacrificing interpretability. The R package `starnet` is available from CRAN.**

## 6.1 Background

High-dimensional regression requires regularisation. The elastic net (Zou and Hastie, 2005) generalises ridge ($L_2$) and lasso ($L_1$) regularisation, and overcomes some of their shortcomings. Adapting the sparsity of the model to the sparsity of the signal, it often improves predictions. One issue with the elastic net is that it has two tuning parameters: either two regularisation parameters $\lambda_1$ and $\lambda_2$ for ridge and lasso, or one regularisation parameter $\lambda$ and one mixing parameter $\alpha$ for moderating between ridge and lasso. Tuning both $\alpha$ and $\lambda$ is notoriously hard due to the flat cross-validated likelihood landscape (van de Wiel et al., 2019). Alternatively, fixing $\alpha$ close to the lasso might be a good solution, because this introduces stability (Friedman et al., 2010). As an alternative to tuning or fixing $\alpha$, we propose to combine multiple values of $\alpha$, using stacked generalisation (Wolpert, 1992). Each $\alpha$ renders one model with one estimated effect for each feature. Instead of selecting one $\alpha$ for making predictions, stacking combines the predictions from multiple $\alpha$ (Figure 6.1). The resulting ensemble model (multiple $\alpha$) might be more predictive than any of the constituent models (single $\alpha$) but is less interpretable due to multiple effects for each feature (one for each $\alpha$). Rather than combining the predicted values from the base learners, we propose to combine their linear predictors. This allows us to rewrite the complex model (with multiple effects for each feature) as a simple model (with one effect for each feature). The stacked elastic net thereby increases predictivity while maintaining the interpretability of the regression coefficients. Furthermore, feature selection is possible after model fitting (Hahn and Carvalho, 2015). In the following, we introduce the stacked elastic net, analyse simulated and experimental high-dimensional data, and discuss possible extensions.

## 6.2 Methods

### 6.2.1 Base learners

The data consist of one outcome and $p$ features for $n$ samples, possibly in a high-dimensional setting ($p \gg n$). For example, the outcome might represent a clinical variable, and the features might represent molecular data. Let the $n \times 1$ vector $\boldsymbol{y}$ denote the outcome, and let the $n \times p$ matrix $\boldsymbol{X}$ denote the features. We index samples by $i \in \{1, \ldots, n\}$ and features by $j \in \{1, \ldots, p\}$. In the generalised linear model framework, we have

$$\mathbb{E}[y_i] = h^{-1}\left(\beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}\right) ,$$

where $h(\cdot)$ is a link function, $\beta_0$ is the unknown intercept, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\intercal$ are the unknown slopes. Penalised maximum likelihood estimation involves determining

$$\{\hat{\beta}_0, \hat{\boldsymbol{\beta}}\} = \underset{\beta_0, \boldsymbol{\beta}}{\operatorname{argmax}} \Big\{ L(\boldsymbol{y}; \beta_0, \boldsymbol{\beta}) - \rho(\lambda, \alpha; \boldsymbol{\beta}) \Big\} ,$$

where $L(\boldsymbol{y}; \beta_0, \boldsymbol{\beta})$ is the likelihood, and $\rho(\lambda, \alpha; \boldsymbol{\beta})$ is the elastic net penalty. The likelihood depends on the type of regression model (e.g. Gaussian, binomial or Poisson), and the penalty function is

$$\rho(\lambda, \alpha; \boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} \left( \frac{1-\alpha}{2} \beta_j^2 + \alpha |\beta_j| \right) ,$$

where $\lambda$ is the regularisation parameter ($\lambda \geq 0$), and $\alpha$ is the elastic net mixing parameter ($0 \leq \alpha \leq 1$). The limits correspond to ridge ($\alpha = 0$) and lasso ($\alpha = 1$) regularisation. We consider $m$ different values for $\alpha$, which are equally spaced in the unit interval and indexed by $k \in \{1, \ldots, m\}$. For each $\alpha_k$, we use 10-fold cross-validation for tuning $\lambda_k$. We consider an exponentially decreasing sequence of values for $\lambda_k$, starting with the intercept-only model ($\lambda_k \to \infty$) and

stopping with the (almost) unpenalised model ($\lambda_k \to 0$). In short, we select the optimal $\lambda_k^\star$ for each $\alpha_k$. We retain the corresponding cross-validated linear predictors in the $n \times m$ matrix $\hat{\boldsymbol{H}}^{(\text{cv})}$.

### 6.2.2 Meta learner

We then regress the outcome on the cross-validated linear predictors:

$$\mathbb{E}[y_i] = h^{-1}\left(\omega_0 + \sum_{k=1}^{m} \omega_k \hat{H}_{ik}^{(\text{cv})}\right) ,$$

where $\omega_0$ is the unknown intercept, and $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m)^\intercal$ are the unknown slopes. The intercept might allow the meta learner to reduce systematic errors from strongly correlated base learners. Since the slopes are weights, we constrain them to the unit interval, i.e. $0 \leq \omega_k \leq 1$ for all $k \in \{1, \ldots, m\}$. They weight the linear predictors from the different elastic net mixing parameters. Penalised conditional maximum likelihood estimation involves determining

$$\{\hat{\omega}_0, \hat{\boldsymbol{\omega}}\} = \underset{\omega_0, \boldsymbol{\omega}}{\operatorname{argmax}} \Big\{ L(\boldsymbol{y}; \omega_0, \boldsymbol{\omega}) - \rho(\lambda; \boldsymbol{\omega}) \Big\} ,$$

where $L(\boldsymbol{y}; \omega_0, \boldsymbol{\omega})$ is the likelihood conditional on $\hat{\boldsymbol{H}}^{(\text{cv})}$, and $\rho(\lambda; \boldsymbol{\omega})$ is the lasso penalty

$$\rho(\lambda; \boldsymbol{\omega}) = \lambda \sum_{k=1}^{m} |\omega_k| \ .$$

Using the same cross-validation folds as for the base learners, we select the optimal regularisation parameter $\lambda^\star$ for the meta learner. Accordingly, in the two consecutive cross-validation loops, we use the same training sets for estimating the base and meta parameters ($\beta_0$ and $\boldsymbol{\beta}$ given $\alpha_k$ for all $k$; $\omega_0$ and $\boldsymbol{\omega}$), and the same validation sets for tuning the base and meta hyperparameters ($\lambda_k$ for all $k$; $\lambda$).

The tuned elastic net is a special case of the stacked elastic net: if the intercept equals zero ($\omega_0 = 0$), one weight equals one ($\omega_k = 1$), and all other weights equal zero ($\omega_{l \neq k} = 0$), the meta learner simply

selects one mixing parameter ($\alpha_k$). In a broader sense, van der Laan et al. (2007) distinguish between *cross-validation selection* and *super-learning*, which consist of selecting one or combining multiple base learners, respectively.

### 6.2.3 Combination

Given the cross-validated parameters $\boldsymbol{\lambda}^\star = (\lambda_1^\star, \ldots, \lambda_m^\star)^\intercal$ and $\lambda^\star$, we refit the base and meta learners to all folds. For the base learners, let the $1 \times m$ vector $\hat{\boldsymbol{\beta}}_0$ and the $p \times m$ matrix $\hat{\boldsymbol{\beta}}$ denote the estimated intercepts and slopes, respectively. For the meta learner, the estimates are $\hat{\omega}_0$ and $\hat{\boldsymbol{\omega}} = (\hat{\omega}_1, \ldots, \hat{\omega}_m)^\intercal$. We then use the estimates from the base and meta learners to predict the outcome of previously unseen samples.

If sample $i$ has the feature vector $\boldsymbol{X}_{i\circ} = (X_{i1}, \ldots, X_{ip})^\intercal$, base learner $k$ returns the linear predictor $\hat{\eta}_{ik} = \hat{\beta}_{0k} + \sum_{j=1}^p \hat{\beta}_{jk} X_{ij}$. The meta learner combines the linear predictors from all base learners:

$$\begin{aligned}\hat{\eta}_i^\star &= \hat{\omega}_0 + \sum_{k=1}^m \hat{\omega}_k \hat{\eta}_{ik} = \hat{\omega}_0 + \sum_{k=1}^m \hat{\omega}_k \left( \hat{\beta}_{0k} + \sum_{j=1}^p \hat{\beta}_{jk} X_{ij} \right) \\ &= \hat{\beta}_0^\star + \sum_{j=1}^p \hat{\beta}_j^\star X_{ij} \ ,\end{aligned}$$

where $\hat{\beta}_0^\star = \hat{\omega}_0 + \sum_{k=1}^m \hat{\omega}_k \hat{\beta}_{0k}$ and $\hat{\beta}_j^\star = \sum_{k=1}^m \hat{\omega}_k \hat{\beta}_{jk}$. Since the *stacked* linear predictor is a function of *pooled* estimates, we perform stacking without loss of interpretability. For each feature, the corresponding pooled estimate represents the estimated effect on the outcome. Due to ridge regularisation in one of the base learners, however, all pooled estimates might be different from zero. Stacking worsens the variable selection property of the elastic net, but we still have the option to select variables after model fitting (see below).

### 6.2.4 Extension

Decoupling shrinkage and selection (Hahn and Carvalho, 2015) allows us to perform feature selection after model fitting. The idea is to approximate the fitted linear predictor $\hat{\boldsymbol{\eta}}^\star = \boldsymbol{X}\hat{\boldsymbol{\beta}}^\star$ by $\boldsymbol{X}\hat{\boldsymbol{\gamma}}$, where $\hat{\boldsymbol{\beta}}^\star$ is dense but $\hat{\boldsymbol{\gamma}}$ is sparse. Instead of including many features ($\sum_{j=1}^{p} \mathbb{I}[\hat{\beta}_j^\star \neq 0] \leq p$), we only want to include some features ($\sum_{j=1}^{p} \mathbb{I}[\hat{\gamma}_j \neq 0] \ll p$). This can be achieved by regressing the fitted linear predictor on the features and estimating a sparse model:

$$\mathbb{E}[\hat{\eta}_i^\star] = \gamma_0 + \sum_{j=1}^{p} \gamma_j X_{ij} \ ,$$

where $\gamma_0$ is the unknown intercept, and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)^\intercal$ are the unknown slopes. Penalised maximum likelihood estimation involves determining

$$\{\hat{\gamma}_0, \hat{\boldsymbol{\gamma}}\} = \underset{\gamma_0, \boldsymbol{\gamma}}{\operatorname{argmax}} \Big\{ L(\hat{\boldsymbol{\eta}}^\star; \gamma_0, \boldsymbol{\gamma}) - \rho(\lambda; \boldsymbol{\gamma}) \Big\} \ ,$$

where $L(\hat{\boldsymbol{\eta}}^\star; \gamma_0, \boldsymbol{\gamma})$ is the Gaussian likelihood, and $\rho(\lambda; \boldsymbol{\gamma})$ is the adaptive lasso penalty (Zou, 2006)

$$\rho(\lambda; \boldsymbol{\gamma}) = \lambda \sum_{j=1}^{p} \frac{|\gamma_j|}{|\hat{\beta}_j^\star|} \ .$$

The absolute values of the dense estimates ($\hat{\boldsymbol{\beta}}^\star$) operate as weights for the sparse estimates ($\hat{\boldsymbol{\gamma}}$). As $\lambda$ increases from 0 to $\infty$, the number of non-zero coefficients decreases from $\min(n, p)$ to 0. We can cross-validate $\lambda$, or adjust $\lambda$ in order that the model includes a specific number of non-zero coefficients (e.g. $\sum_{j=1}^{p} \mathbb{I}[\hat{\gamma}_j \neq 0] = 10$). We expect this approximation to work well when the pooled estimates are relatively sparse, i.e. include few values far from zero and many values close to zero. Such a situation is fairly natural for the stacked elastic net because it pools mainly sparse and strongly correlated models. Nevertheless, post-hoc feature selection might significantly decrease the predictive performance of the stacked elastic net, and should therefore be used with caution.

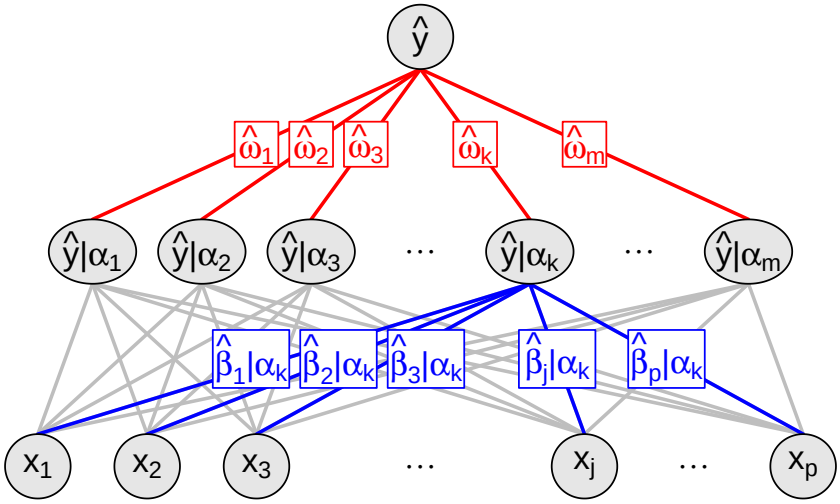


Figure 6.1: Stacked elastic net. After predicting the outcome from the features given an elastic net mixing parameter (bottom), we combine the predictions from multiple elastic net mixing parameters (top).

## 6.3 Simulation

### 6.3.1 Prediction accuracy

To examine the predictive performance of the stacked elastic net, we conducted a simulation study. We compared ridge, lasso, tuned elastic net, and stacked elastic net regularisation.

In three different scenarios, we repeatedly simulated high-dimensional data. In each iteration, we sampled several $n$-dimensional vectors from the standard Gaussian distribution, namely three signal variables $(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3)$ and $p$ noise variables $(\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_p)$. We constructed the outcome from the signal variables, and the features from the signal and noise variables. In all scenarios, the $n$-dimensional outcome vector equals the sum of the three signal variables $(\boldsymbol{y} = \boldsymbol{z}_1 + \boldsymbol{z}_2 + \boldsymbol{z}_3)$. The $n \times p$ feature matrix $\boldsymbol{X}$, however, depends on the scenario (Table 6.1). Let $\boldsymbol{x}_j$ denote the $j^{\text{th}}$ column of $\boldsymbol{X}$, for any $j$ in $\{1, \ldots, p\}$. Each feature equals a weighted sum of one signal variable and one noise variable: $\boldsymbol{x}_j = \sqrt{\pi}\boldsymbol{z}_l + \sqrt{1-\pi}\boldsymbol{\epsilon}_j$, where the weight $\pi$ is in the unit interval, and the index $l$ equals 1 or 2. The weight $\pi$ determines whether the feature is weakly $(\pi = 0.1)$, moderately $(\pi = 0.5)$, or strongly $(\pi = 0.9)$ correlated with the signal variable, and consequently weakly, moderately, or highly predictive of the outcome. In the first scenario, one feature is strongly correlated with $\boldsymbol{z}_1$, and another feature is strongly correlated with $\boldsymbol{z}_2$. In the second scenario, 50% of the features are weakly correlated with $\boldsymbol{z}_1$, and the other 50% are weakly correlated with $\boldsymbol{z}_2$. And in the third scenario, 5% of the features are moderately correlated with $\boldsymbol{z}_1$, and another 5% of the features are moderately correlated with $\boldsymbol{z}_2$. The weighting ensures that all features have unit variance: $\text{Var}(\boldsymbol{x}_j) = \pi\text{Var}(\boldsymbol{z}_l) + (1-\pi)\text{Var}(\boldsymbol{\epsilon}_j) = 1$ because $\text{Var}(\boldsymbol{z}_l) = 1$, $\text{Var}(\boldsymbol{\epsilon}_j) = 1$ and $\text{Cov}(\boldsymbol{z}_l, \boldsymbol{\epsilon}_j) = 0$.

In each scenario, we simulated the outcome ($n \times 1$ vector $\boldsymbol{y}$) and the features ($n \times p$ matrix $\boldsymbol{X}$) each 100 times, where $n = 10\,000$ and $p = 500$. We assessed the predictive performance using 100 samples for training and validation (internal 10-fold cross-validation) and

9 900 samples for testing (hold-out). Figure 6.2 shows the mean squared error for the test set under different flavours of elastic net regularisation (ridge, lasso, tuning, stacking). These out-of-sample errors are estimates of the predictive performance on previously unseen data, with lower values indicating better predictions. Lasso outperforms ridge if the signal is sparse (1st scenario), but ridge outperforms lasso if the signal is dense (2nd scenario). Approaching the performance of the optimal elastic net mixing parameter, tuning is slightly worse than lasso in the sparse case (1st scenario), slightly worse than ridge in the dense case (2nd scenario), or better than both in the intermediate case (3rd scenario). We observe that stacking outperforms tuning in all three scenarios. Stacking is even slightly better than lasso in the sparse case and slightly better than ridge in the dense case. The most important gains relative to the best competitor occur in the intermediate case. In the three scenarios, stacking is the best approach in 79%, 67% and 88% of the iterations, respectively.

Next, we tested whether stacking leads to significantly better predictions than ridge, lasso and tuning. For this purpose, we calculated the pairwise differences in out-of-sample mean squared error, applied the two-sided Wilcoxon signed-rank test, and used the Bonferroni-adjusted 5% significance level ($p$-value $\leq 0.05/9$). Stacking significantly outperforms tuning in all three scenarios. Moreover, stacking is significantly better than ridge and lasso, but not significantly different from ridge if the signal is dense (2nd scenario). In practice, we often do not know whether ridge or lasso is more suitable for the data at hand. An advantage of the elastic net is that it automatically adapts to the sparsity level of the signal.

For comparison, we also examined the elastic net with the fixed mixing parameters $\alpha = 0.05$ (close to ridge) and $\alpha = 0.95$ (close to lasso). As expected, the ridge-like elastic net performs better than ridge if the signal is sparse (1st scenario) and worse than ridge if the signal is dense (2nd scenario). The results for the lasso-like elastic net are similar to those for the lasso. Indeed, it has previously been found that the elastic net without simultaneous tuning of both

penalties can mimic ridge or lasso regression (Waldron et al., 2011).

Some applications require models with a limited number of selected features. We therefore verified how post-hoc feature selection affects the predictive performance of the stacked elastic net. Figure 6.3 shows the generalisation error for different numbers of non-zero coefficients. Models with many selected features tend to be more predictive than models with few selected features. While the stacked elastic net outperforms the lasso given a small number of non-zero coefficients, this difference vanishes for large numbers of non-zero coefficients. Post-hoc feature selection increases predictivity if the signal is sparse (1st scenario) and otherwise decreases predictivity (2nd and 3rd scenarios).

Table 6.1: Scenarios for constructing features $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{500})$ from signal $(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3)$ and noise $(\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_{500})$.

| | signal + noise | noise | signal + noise |
|---|---|---|---|
| (1) | $\underset{j=1}{\boldsymbol{x}_j = \sqrt{0.9}\boldsymbol{z}_1 + \sqrt{0.1}\boldsymbol{\epsilon}_j}$ | $\underset{\forall j \in \{2,\ldots,499\}}{\boldsymbol{x}_j = \boldsymbol{\epsilon}_j}$ | $\underset{j=500}{\boldsymbol{x}_j = \sqrt{0.9}\boldsymbol{z}_2 + \sqrt{0.1}\boldsymbol{\epsilon}_j}$ |
| (2) | $\underset{\forall j \in \{1,\ldots,250\}}{\boldsymbol{x}_j = \sqrt{0.1}\boldsymbol{z}_1 + \sqrt{0.9}\boldsymbol{\epsilon}_j}$ | – | $\underset{\forall j \in \{251,\ldots,500\}}{\boldsymbol{x}_j = \sqrt{0.1}\boldsymbol{z}_2 + \sqrt{0.9}\boldsymbol{\epsilon}_j}$ |
| (3) | $\underset{\forall j \in \{1,\ldots,25\}}{\boldsymbol{x}_j = \sqrt{0.5}\boldsymbol{z}_1 + \sqrt{0.5}\boldsymbol{\epsilon}_j}$ | $\underset{\forall j \in \{26,\ldots,475\}}{\boldsymbol{x}_j = \boldsymbol{\epsilon}_j}$ | $\underset{\forall j \in \{476,\ldots,500\}}{\boldsymbol{x}_j = \sqrt{0.5}\boldsymbol{z}_2 + \sqrt{0.5}\boldsymbol{\epsilon}_j}$ |

### 6.3.2 Estimation accuracy

If we knew the effects of the features on the outcome, we could not only examine the prediction accuracy but also the estimation accuracy of the stacked elastic net. We adapted the simulation study to make this possible: (1) Simulating the features ($n \times p$ matrix $\boldsymbol{X}$) from the multivariate Gaussian distribution $\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with a

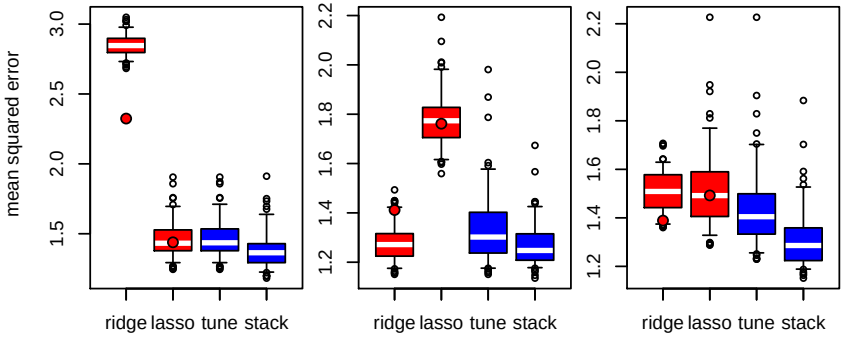


Figure 6.2: Out-of-sample mean squared error in the first (left), second (centre), and third (right) scenario. The filled circles indicate the medians from the ridge-like ($\alpha = 0.05$) and lasso-like ($\alpha = 0.95$) elastic net. (The boxes show the interquartile ranges, and the whiskers show the ranges from the $5^{\text{th}}$ to the $95^{\text{th}}$ percentiles.)

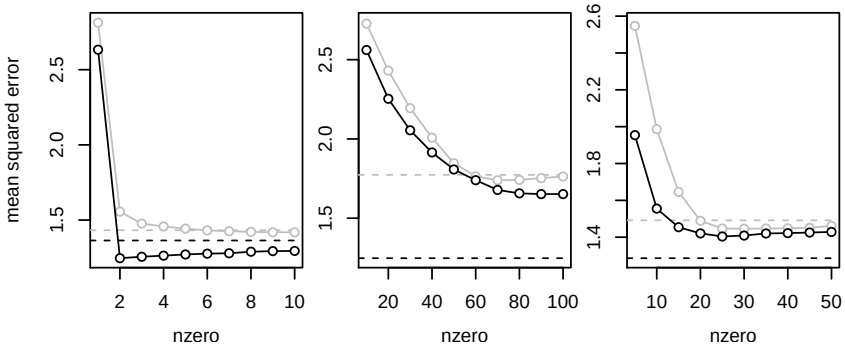


Figure 6.3: Median out-of-sample mean squared error against number of non-zero coefficients, for the lasso (grey) and the stacked elastic net with post-hoc feature selection (black), in the first (left), second (centre), and third (right) scenario. The dashed lines indicate the medians from the unrestricted versions.

constant mean and correlation structure, namely $\mu_j = 0$, $\Sigma_{jj} = 1$ and $\Sigma_{jk} = 0.1$ for all $j$ and $k \neq j$ in $\{1, \ldots, p\}$. (2) Generating the effects ($p \times 1$ vector $\boldsymbol{\beta}$) by setting most coefficients to zero and some coefficients to one, namely 5 (sparse scenario), 50 (dense scenario), or 20 (mixed scenario). (3) Obtaining the outcome ($n \times 1$ vector $\boldsymbol{y}$) by summing up the linear predictor and the residuals ($\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$), where the residuals are Gaussian noise with half the (sample) standard deviation of the linear predictor.

In each scenario, we simulated 100 times the feature matrix $\boldsymbol{X}$, the coefficient vector $\boldsymbol{\beta}$, and the outcome vector $\boldsymbol{y}$, for 100 training and validation samples (but no testing samples). We measure the difference between the true coefficients $\boldsymbol{\beta}$ and the estimated coefficients $\hat{\boldsymbol{\beta}}$ with the mean absolute error and the mean squared error. For true coefficients equal to zero, stacking is less accurate than tuning. This matches our expectations because stacking leads to denser models than tuning. For true coefficients different from zero, however, stacking is more accurate than tuning. The median decrease in mean absolute error (mean squared error) is 30.4% (15.4%) in the sparse scenario, 3.1% (4.6%) in the dense scenario, and 0.4% (1.1%) in the mixed scenario. Stacking is significantly more accurate than tuning in the sparse and dense scenarios in terms of both metrics, according to the two-sided Wilcoxon signed rank test at the Bonferroni-adjusted 5% level ($p$-value $\leq 0.05/3$).

Additionally, we also examined the selection accuracy. We allowed the stacked elastic with post-hoc feature selection, the lasso, and the lasso-like elastic net ($\alpha = 0.95$) to include at most 10 features in the model. To compare the selection accuracy, we calculate the precision $\text{TP}/(\text{TP} + \text{FP})$, where $\text{TP} = \sum_{j=1}^{p} \mathbb{I}[\hat{\beta}_j \neq 0 \cap \beta_j \neq 0]$ and $\text{FP} = \sum_{j=1}^{p} \mathbb{I}[\hat{\beta}_j \neq 0 \cap \beta_j = 0]$, with $\text{TP} + \text{FP} \leq 10$. Compared to the lasso, the stacked elastic net selects more features among those with an effect ($\overline{\text{TP}}$: $4.4 > 3.7$), and less features among those without an effect ($\overline{\text{FP}}$: $5.2 < 6.1$). Accordingly, the stacked elastic net has a higher mean precision than the lasso in the sparse ($57\% > 51\%$), dense ($36\% > 27\%$), and mixed ($49\% > 36\%$) sce-

nario. The lasso-like elastic net performs slightly worse than the lasso.

## 6.4 Application

### 6.4.1 Benchmark data sets

To further examine the performance of the stacked elastic net, we analysed experimental genomics data. The R package `plsgenomics` includes three preprocessed gene expression sets for binary or multinomial classification, namely tumour against normal colon tissue (Alon et al., 1999), two kinds of leukaemia (Golub et al., 1999), and four types of small-blue-round-cell tumours (Khan et al., 2001). For the last, we reduced the multinomial problem to four one-versus-rest binary problems. All three data sets are high-dimensional: the first covers 62 samples and 2 000 features, the second covers 38 samples and 3 051 features, and the third covers 83 samples and 2 308 features. We did not perform any further preprocessing to ensure reproducibility and comparability. To obtain robust and almost unbiased estimates of the predictive performance, we used repeated nested cross-validation with 10 repetitions, 10 external folds, and 10 internal folds. Table 6.2 shows the median cross-validated logistic deviance for the six binary classification problems. The stacked elastic net decreases the loss, as compared to ridge, lasso, and tuning, except for lasso on the colon data set. Under post-hoc feature selection with the number of non-zero coefficients determined by cross-validation, stacking remains competitive.

Figure 6.4 shows the median cross-validated loss for different elastic net mixing parameters. For "leukaemia" and "SRBCT", the loss decreases between 0 (ridge) and some $\alpha$, and then increases between this $\alpha$ and 1 (lasso). The optimal elastic net mixing parameter, across all cross-validation repetitions, is $\alpha = 0.95$ for "colon", $\alpha = 0.2$ for "leukaemia", and $\alpha = 0.4$ for "SRBCT". If we had known these values before the analysis, we would have minimised the cross-validated loss. Searching for the optimal $\alpha$ in each cross-validation

iteration, we either find or miss the optimal $\alpha$. This is why the tuned elastic net never outperforms the elastic net with the optimal $\alpha$ for a single split. By contrast, the stacked elastic net may outperform the elastic net with the optimal $\alpha$. We observe this for two out of three applications, namely "leukaemia" and "SRBCT".

Table 6.2: Median cross-validated logistic deviance for three classification problems (rows) under different regularisation methods (columns), with the class frequencies (0/1) in the first two columns, and the results for post-hoc feature selection in parentheses.

| | #0 | #1 | ridge | lasso | tune | stack | |
|---|---|---|---|---|---|---|---|
| colon | 22 | 40 | 0.900 | 0.820 | 0.878 | 0.848 | (0.840) |
| leukaemia | 27 | 11 | 0.252 | 0.199 | 0.145 | 0.039 | (0.165) |
| SRBCT1 | 54 | 29 | 0.369 | 0.164 | 0.111 | 0.078 | (0.140) |
| SRBCT2 | 72 | 11 | 0.111 | 0.035 | 0.047 | 0.001 | (0.007) |
| SRBCT3 | 65 | 18 | 0.258 | 0.052 | 0.052 | 0.004 | (0.005) |
| SRBCT4 | 58 | 25 | 0.338 | 0.102 | 0.070 | 0.015 | (0.070) |

### 6.4.2 Normal/tumour classification

The Cancer Genome Atlas (The Cancer Genome Atlas Research Network et al., 2013) provides genomic data for 33 cancer types. We retrieved the upper quartile normalised RSEM (RNA-Seq by expectation-maximisation) TPM (transcript per million) gene expression values (R package `curatedTCGAData`), merged replicated measurements (R package `MultiAssayExperiment`), and extracted the sample definitions from the barcodes (R package `TCGAutils`). We retained "solid tissue normal" (collected near the tumour) and "primary solid tumour" samples. For each cancer type, we retained the 2 000 most variably expressed genes, and standardised their expression values.

For cancer types with at least five normal and five tumour sam-

ples, we repeatedly trained and validated models with approximately 90% of the samples, and tested the models with approximately 10% of the samples. Table 6.3 shows the cross-validated logistic deviance under different regularisation methods. Here, lasso performs better than ridge for 13 out of 15 cancer types, and stacking performs better than tuning for 11 out of 15 cancer types. The mean decrease in cross-validated logistic deviance from tuning to stacking is 7.5%, and the two-sided Wilcoxon signed rank test returns a $p$-value of 0.06. Post-hoc feature selection with the number of non-zero coefficients determined by cross-validation leads to competitive models, except for cholangiocarcinoma (CHOL). The problem with this cancer type might be the small sample size together with the fact that normal and tumour samples are derived from the same patients. In any case, results for such small sample sizes are inherently unreliable.

## 6.5 Discussion

The elastic net is the method of choice for many biomedical applications, because it renders predictive and interpretable models. It weights between ridge and lasso regularisation, but the optimal weighting is often unknown. Instead of selecting one weighting by tuning, we combine multiple weightings by stacking. According to our empirical analyses, this improves the predictive performance of the elastic net in various settings. The increase in computational cost is negligible, because the only addition is the low-dimensional regression of the outcome on the cross-validated linear predictors. The equivalence between stacking linear predictors and pooling regression coefficients allows us to increase the predictive performance while maintaining the interpretability of the regression coefficients.

In contrast to the lasso, the stacked elastic net might or might not perform feature selection. It selects features unless the meta learner includes the base learner with pure ridge regularisation, but it tends to select more features than the tuned elastic net, because it combines multiple base learners. The stacked elastic net selects

Table 6.3: Cross-validated logistic deviance for binary classification problems (rows) under different regularisation methods (columns), with the class frequencies (0/1) in the first two columns, and the results for post-hoc feature selection in parentheses.

| | #0 | #1 | ridge | lasso | tune | stack | |
|---|---|---|---|---|---|---|---|
| BLCA | 19 | 389 | 0.242 | 0.248 | 0.265 | 0.240 | (0.240) |
| BRCA | 112 | 977 | 0.271 | 0.197 | 0.187 | 0.266 | (0.276) |
| CHOL | 9 | 27 | 0.754 | 0.069 | 0.061 | 0.023 | (0.645) |
| ESCA | 11 | 172 | 0.394 | 0.296 | 0.302 | 0.259 | (0.260) |
| HNSC | 44 | 475 | 0.279 | 0.291 | 0.264 | 0.267 | (0.269) |
| KICH | 25 | 41 | 0.853 | 0.755 | 0.608 | 0.581 | (0.584) |
| KIRC | 72 | 460 | 0.198 | 0.155 | 0.162 | 0.159 | (0.159) |
| KIRP | 32 | 257 | 0.390 | 0.250 | 0.299 | 0.255 | (0.252) |
| LIHC | 50 | 321 | 0.362 | 0.330 | 0.304 | 0.323 | (0.330) |
| LUAD | 59 | 457 | 0.262 | 0.245 | 0.253 | 0.206 | (0.203) |
| LUSC | 51 | 450 | 0.161 | 0.076 | 0.078 | 0.073 | (0.073) |
| PRAD | 52 | 444 | 0.444 | 0.438 | 0.464 | 0.442 | (0.488) |
| STAD | 35 | 383 | 0.339 | 0.237 | 0.230 | 0.237 | (0.237) |
| THCA | 59 | 434 | 0.431 | 0.393 | 0.463 | 0.458 | (0.459) |
| UCEC | 10 | 360 | 0.088 | 0.072 | 0.085 | 0.062 | (0.059) |

a feature if and only if the meta learner selects a base learner that selects this feature. It is therefore possible to impose feature selection by excluding the base learner with pure ridge regularisation ($\alpha > 0$). As this might fail to render sufficiently sparse models, we suggest to perform post-hoc feature selection (Hahn and Carvalho, 2015) but recommend to verify by cross-validation whether imposing sparsity makes the model much less predictive.

An extension of the stacked elastic net would be to use a fused penalty (Tibshirani et al., 2005) for the meta learner, because the base learners are related in regard to the elastic net mixing parameter. Another extension would be to combine two ensemble techniques, namely stacking and bagging. While stacking involves fitting different models to the same samples and *weighting* the predictions, bagging involves fitting the same model to different bootstrap samples and *averaging* the predictions. Since random (bagged) regressions seem to be competitive with random forests (Song et al., 2013), we could potentially combine stacking and bagging to make elastic net regression even more predictive without making it less interpretable.

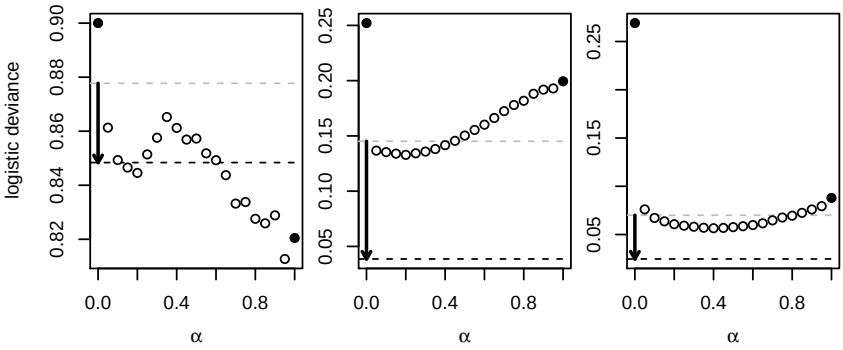


Figure 6.4: Median cross-validated logistic deviance against the elastic net mixing parameter, for "colon" (left), "leukaemia" (centre), and "SRBCT" (right). The filled circles indicate ridge ($\alpha = 0$) and lasso ($\alpha = 1$) regularisation. The dashed lines indicate tuning (grey) and stacking (black). (For "SRBCT" we show the mean over the medians from the four binary problems.)

**Software:** The R package `starnet` is available from CRAN: `https://CRAN.R-project.org/package=starnet`.

**Acknowledgements:** We are grateful to Léon-Charles Tranchevent for helpful discussions, and to Maharshi Vyas for technical support. This work was supported by the Luxembourg National Research Fund (FNR) as part of the National Centre for Excellence in Research on Parkinson's disease (I1R-BIC-PFN-15NCER) and the ERA-Net ERACoSysMed JTC-2 project PD-Strat (INTER/11651464).

**Author contributions:** The authors contributed to this research by developing the method (AR, MAW), preparing the manuscript (AR, EG), and revising the manuscript critically (EG, MAW). All authors read and approved the final manuscript.