Feature-weighted elastic net: using "features of features" for better prediction

J. Kenneth Tay[1], **Nima Aghaeepour**[2,3,4], **Trevor Hastie**[1,4], **Robert Tibshirani**[1,4]

[1]Department of Statistics, Stanford University

[2]Department of Anesthesiology, Pain, and Perioperative Medicine, Stanford University

[3]Department of Pediatrics, Stanford University

[4]Department of Biomedical Data Sciences, Stanford University

Abstract

In some supervised learning settings, the practitioner might have additional information on the features used for prediction. We propose a new method which leverages this additional information for better prediction. The method, which we call the *feature-weighted elastic net* ("fwelnet"), uses these "features of features" to adapt the relative penalties on the feature coefficients in the elastic net penalty. In our simulations, fwelnet outperforms the lasso in terms of test mean squared error and usually gives an improvement in true positive rate or false positive rate for feature selection. We also apply this method to early prediction of preeclampsia, where fwelnet outperforms the lasso in terms of 10-fold cross-validated area under the curve (0.86 vs. 0.80). We also provide a connection between fwelnet and the group lasso and suggest how fwelnet might be used for multi-task learning.

1 Introduction

We consider the usual linear regression model: given n realizations of p predictors $\mathbf{X} = \{x_{ij}\}$ for $i = 1,2, \ldots, n$ and $j = 1,2, \ldots, p$, the response $\mathbf{y} = (y_1, \ldots, y_n)$ is modeled as

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i, \tag{1}$$

with ϵ having mean 0 and variance σ^2. The ordinary least squares (OLS) estimates of β_j are obtained by minimizing the residual sum of squares (RSS). There has been much work on regularized estimators that offer an advantage over the OLS estimates, both in terms of accuracy of prediction on future data and interpretation of the fitted model. One popular regularized estimator is the elastic net (Zou & Hastie 2005) which minimizes the sum of the RSS and a combination of the ℓ_1 and ℓ_2-squared penalties. More precisely, letting $\beta = (\beta_1, \ldots, \beta_p)^T$, the elastic net minimizes the objective function

$$J(\beta_0, \beta) = \frac{1}{2}\|\mathbf{y} - \beta_0 \mathbf{1} - \mathbf{X}\beta\|_2^2 + \lambda\left[\alpha\|\beta\|_1 + \frac{1-\alpha}{2}\|\beta\|_2^2\right] \tag{2}$$

$$= \frac{1}{2}\|\mathbf{y} - \beta_0 1 - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^{p} \left(\alpha|\beta_j| + \frac{1-\alpha}{2}\beta_j^2 \right). \tag{3}$$

The elastic net has two tuning parameters: $\lambda \geq 0$ which controls the overall sparsity of the solution, and $\alpha \in [0, 1]$ which determines the relative weight of the ℓ_1 and ℓ_2-squared penalties. $\alpha = 0$ corresponds to ridge regression (Hoerl & Kennard 1970), while $\alpha = 1$ corresponds to the lasso (Tibshirani 1996). These two tuning parameters are often chosen via cross-validation (CV). One reason for the elastic net's popularity is its computational efficiency: J is convex in its parameters, which means that solutions can be found efficiently even for very large n and p. In addition, the solution for a whole path of λ values can be computed quickly using warm starts (Friedman et al. 2010).

In some supervised learning settings, we often have some information about the features themselves. For example, in genomics, we know that each gene belongs to one or more genetic pathways, and we may expect genes in the same pathway to have correlated effects on the response of interest. Another example is in image data, where each pixel has a specific position (row and column) in the image. We would expect methods which leverage such information on the features to perform better prediction and inference than methods which ignore it. However, many popular supervised learning methods, including the elastic net, do not use such information about the features in the model-fitting process.

In this paper, we develop a framework for organizing such feature information as well as a variant of the elastic net which uses this information in model-fitting. We assume the information we have for each feature is quantitative. This allows us to think of each source as a "feature" of the features. For example, in the genomics setting, the kth source of information could be the indicator variable for whether the jth feature belongs to the kth genetic pathway.

We organize these "features of features" into an auxiliary matrix $\mathbf{Z} \in \mathbb{R}^{p \times K}$, where p is the number of features and K is the number of sources of feature information. Each column of $\mathbf{Z}$ represents the values for each feature information source, while each row of $\mathbf{Z}$ represents the values that a particular feature takes on for the K different sources. We let $\mathbf{z}_j \in \mathbb{R}^K$ denote the jth row of $\mathbf{Z}$ as a column vector.

To make use of the information in $\mathbf{Z}$, we propose assigning each feature a *score* $\mathbf{z}_j^T \theta$, which is simply a linear combination of its "features of features". We then use these scores to influence the weight given to each feature in the model-fitting procedure. Concretely, we give each feature a different penalty weight in the elastic net objective function based on its score:

$$J_{\lambda, \alpha, \theta}(\beta_0, \beta) = \frac{1}{2}\|\mathbf{y} - \beta_0 \mathbf{1} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^{p} w_j(\theta)\left[\alpha|\beta_j| + \frac{1-\alpha}{2}\beta_j^2 \right], \tag{4}$$

where $w_j(\theta) = f(\mathbf{z}_j^T\theta)$ for some function f. θ is a hyperparameter in $\mathbb{R}^K$ which the algorithm needs to select. In the final model, $\mathbf{z}_j^T\theta$ can be thought of as an indication of how influential feature j is on the response, while θ_k represents how important the kth source of feature information is in identifying which features are important for the prediction problem.

The rest of this paper is organized as follows. In Section 2, we survey past work on incorporating "features of features" in supervised learning. In Section 3, we propose a method, the *feature-weighted elastic net* ("fwelnet"), which uses the scores in model-fitting. We then illustrate its performance on simulated data in Section 4 and on a real data example in Section 5. In Section 6, we present a connection between fwelnet and the group lasso, and in Section 7, we show how fwelnet can be used in multi-task learning. We end with a discussion and ideas for future work. The appendix contains further details and proofs.

2   Related work

The idea of assigning different penalty weights for different features in the lasso or elastic net objective is not new. For example, the adaptive lasso (Zou 2006) assigns feature j a penalty weight $w_j = 1/\left|\hat{\beta}_j^{OLS}\right|^\gamma$, where $\hat{\beta}_j^{OLS}$ is the estimated coefficent for feature j in the OLS model and $\gamma > 0$ is some hyperparameter. However, the OLS solution only depends on $\mathbf{X}$ and $\mathbf{y}$ and does not incorporate any external information on the features. In the work closest to ours, Bergersen et al. (2011) propose using weights $w_j = \dfrac{1}{|\eta_j(\mathbf{y}, \mathbf{X}, \mathbf{Z})|^q}$, where η_j is some function (possibly varying for j) and q is a hyperparameter controlling the shape of the weight function. While the authors present two ideas for what the η_j's could be, they do not give general guidance on how to choose these functions which could drastically influence the model-fitting algorithm.

There is a correspondence between penalized regression estimates and Bayesian maximum a posteriori (MAP) estimates with a particular choice of prior for the coefficients. (For example, ridge regression and lasso regression are MAP estimates when the coefficient vector β is given a normal and Laplace prior respectively.) Within this Bayesian framework, some methods have been developed to use external feature information to guide the choice of prior. For example, van de Wiel et al. (2016) take an empirical Bayes approach to estimate the prior for ridge regression, while Velten & Huber (2018) use variational Bayes to do so for general convex penalties.

We also note that most previous approaches for penalized regression with external information on the features only work with specific types of such information. A large number of methods have been developed to make use of *feature grouping information*. Popular methods for using grouping information in penalized regression include the group lasso (Yuan & Lin 2006) and the overlap group lasso (Jacob et al. 2009). IPF-LASSO (integrative lasso with penalty factors) (Boulesteix et al. 2017) gives features in each group its own penalty parameter, to be chosen via cross-validation. Tai & Pan (2007) modify the penalized partial least squares (PLS) and nearest shrunken centroids methods to have group-specific penalties.

Other methods have been developed to incorporate "network-like" or feature similarity information, where the user has information on how the features themselves are related to each other. For example, the fused lasso (Tibshirani et al. 2005) adds an ℓ_1 penalty on the successive differences of the coefficients to impose smoothness on the coefficient profile. Structured elastic net (Slawski et al. 2010) generalizes the fused lasso by replacing the ℓ_2-squared penalty in elastic net with $\beta^T \Lambda \beta$, where Λ is a symmetric, positive semi-definite matrix chosen to reflect some a priori known structure between the features. Mollaysa et al. (2017) use the feature information matrix $\mathbf{Z}$ to compute a feature similarity matrix, which is used to construct a regularization term in the loss criterion to be minimized. The regularizer encourages the model to give the same output as long as the total contribution of similar features is the same. We note that this approach implicitly assumes that the sources of feature information are equally relevant, which may or may not be the case.

It is not clear how most of the methods in the previous two paragraphs can be generalized to more general sources of feature information. Our method has the distinction of being able to work directly with real-valued feature information and to integrate multiple sources of feature information. We note that while van de Wiel et al. (2016) claim to be able to handle binary, nominal, ordinal and continuous feature information, the method actually ranks and groups features based on such information and only uses this grouping information in the estimation of the group-specific penalties. Nevertheless, it is able to incorporate more than one source of feature information.

3   Feature-weighted elastic net ("fwelnet")

As mentioned in the introduction, one direct way to utilize the scores $\mathbf{z}_j^T \theta$ in model-fitting is to give each feature a different penalty weight in the elastic net objective function based on its score:

$$J_{\lambda, \alpha, \theta}(\beta_0, \beta) = \frac{1}{2} \|\mathbf{y} - \beta_0 \mathbf{1} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^{p} w_j(\theta) \left[\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right], \tag{5}$$

where $w_j(\theta) = f(\mathbf{z}_j^T \theta)$ for some function f. Our proposed method, which we call the *feature-weighted elastic net* ("fwelnet"), specifies f:

$$w_j(\theta) = \frac{\sum_{\ell=1}^{p} \exp(\mathbf{z}_\ell^T \theta)}{p \exp(\mathbf{z}_j^T \theta)}. \tag{6}$$

The fwelnet algorithm seeks the minimizer of this objective function over β_0 and β:

$$\begin{aligned}
\left(\hat{\beta}_0, \hat{\beta} \right) &= \operatorname*{argmin}_{\beta_0, \beta} J_{\lambda, \alpha, \theta}(\beta_0, \beta) \\
&= \operatorname*{argmin}_{\beta_0, \beta} \frac{1}{2} \|\mathbf{y} - \beta_0 \mathbf{1} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^{p} w_j(\theta) \left[\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right].
\end{aligned} \tag{7}$$

There are a number of reasons for this choice of penalty factors. First, when $\theta = 0$, we have $w_j(\theta) = 1$ for all j, reducing fwelnet to the original elastic net algorithm. Second, $w_j(\theta) \;\; 1/p$ for all j and θ, ensuring that we do not end up with features having negligible penalty. This allows the fwelnet solution to have a wider range of sparsity as we go down the path of λ values. Third, this formulation provides a connection between fwelnet and the group lasso (Yuan & Lin 2006) which we detail in Section 6. Finally, we have a natural interpretation of a feature's score: if $\mathbf{z}_j^T \theta$ is relatively large, then w_j is relatively small, meaning that feature j is more important for the response and hence should have smaller regularization penalty.

We illustrate the last property via a simulated example. In this simulation, we have $n = 200$ observations and $p = 100$ features which come in groups of 10. The response is a linear combination of the first two groups with additive Gaussian noise. The coefficient for the first group is 4 while the coefficient for the second group is -2 so that the features in the first group exhibit stronger correlation to the response compared to the second group. The "features of features" matrix $\mathbf{Z} \in \mathbb{R}^{100 \times 10}$ is grouping information, i.e. $z_{jk} = 1$ if feature j belongs to group k, and is 0 otherwise. Figure 1 shows the penalty factors w_j that fwelnet assigns the features. As one would expect, the features in the first group have the smallest penalty factor followed by features in the second group. In contrast, the original elastic net algorithm would assign penalty factors $w_j = 1$ for all j.

3.1 Computing the fwelnet solution

It can be easily shown that $\hat{\beta}_0 = \bar{\mathbf{y}} - \sum_{j=1}^p \hat{\beta}_j \bar{\mathbf{x}}_{\cdot j}$. Henceforth, we assume that $\mathbf{y}$ and the columns of $\mathbf{X}$ are centered so that $\hat{\beta}_0 = 0$ and we can ignore the intercept term in the rest of the discussion.

For given values of λ, α and θ, it is easy to solve (7): the objective function is convex in β (in fact it is piecewise-quadratic in β) and $\hat{\beta}$ can be found efficiently using algorithms such as coordinate descent. However, to deploy fwelnet in practice we need to determine the hyperparameter values $\hat{\lambda} \in \mathbb{R}$, $\hat{\alpha} \in \mathbb{R}$ and $\hat{\theta} \in \mathbb{R}^K$ that give good performance. When K, the number of sources of feature information, is small, one could run the algorithm for a grid of θ values, then pick the value which gives the smallest cross-validated loss. Unfortunately, this approach is computationally infeasible for even moderate values of K.

To avoid this computational bottleneck, we propose Algorithm 1 as a method to find $\hat{\beta}$ and $\hat{\theta}$ at the same time. If we think of θ as an argument of the objective function J, Step 3 can be thought of as alternating minimization over θ and β. Notice that in Step 3(c), we allow the algorithm to have a different value of $\hat{\beta}$ for each λ value. However, *we force $\hat{\theta}$ to be the same across all λ values*: Steps (a) and (b) can be thought of as a heuristic to perform gradient descent for θ under this constraint.

We have developed an R package, fwelnet, which implements Algorithm 1. We note that Step 3(c) of Algorithm 1 can be done easily by using the glmnet function in the glmnet R package and specifying the penalty.factor option. In practice, we use the lambda sequence $\lambda_1 > \cdots > \lambda_m$ provided by glmnet's implementation of the elastic net as this range of λ values covers a sufficiently wide range of models. With this choice of λ sequence, we find

that fwelnet's performance does not change much whether we use the component-wise mean or median in Step 3(a), or the mean or median in Step 3(b). Also, instead of running Step 3 until convergence, we recommend running it for a small fixed number of iterations N. Step 3(c) is the bottleneck of the algorithm, and so the runtime for fwelnet is approximately $N +$ 1 times that of glmnet. In our simulation studies, we often find that one pass of Step 3 gives a sufficiently good solution. We suggest treating N as a hyperparameter and running fwelnet for $N = 1$, 2 and 5.

(We also considered an approach where we did not constrain the value of θ to be equal across λ values. While conceptually straightforward, the algorithm was computationally slow and did not perform as well as Algorithm 1 in prediction. A sketch of this approach can be found in Appendix A.)

3.2 Extending fwelnet to generalized linear models (GLMs)

In the exposition above, the elastic net is described as a regularized version of the ordinary least squares model. It is easy to extend elastic net regularization to generalized linear models (GLMs) by replacing the RSS term with the negative log-likelihood of the data:

$$\left(\widehat{\beta}_0, \widehat{\beta}\right) = \operatorname*{argmin}_{\beta_0, \beta} \sum_{i=1}^{n} \ell\left(y_i, \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j\right) + \lambda \sum_{j=1}^{p} \left[\alpha|\beta_j| + \frac{1-\alpha}{2}\beta_j^2\right], \qquad (8)$$

where $\ell(y_i, \beta_0 + \Sigma_j x_{ij}\beta_j)$ is the negative log-likelihood contribution of observation i. Fwelnet can be extended to GLMs in a similar fashion:

$$\left(\widehat{\beta}_0, \widehat{\beta}, \widehat{\theta}\right) = \operatorname*{argmin}_{\beta_0, \beta, \theta} \sum_{i=1}^{n} \ell\left(y_i, \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j\right) + \lambda \sum_{j=1}^{p} w_j(\theta)\left[\alpha|\beta_j| + \frac{1-\alpha}{2}\beta_j^2\right], \qquad (9)$$

with $w_j(\theta)$ as defined in (6). Theoretically Algorithm 1 can be used as-is to solve (9). Because θ only appears in the penalty term and not in the negative log-likelihood, this extension is not hard to implement in code. The biggest hurdle to this extension is a solver for (8) which is needed for Steps 2 and 3(c). Step 3(a) is the same as before, while Step 3(b) simply requires a function that allows us to compute the negative log-likelihood ℓ

Algorithm 1 *Fwelnet algorithm*

1. Select a value of $\alpha \in [0, 1]$ and a sequence of λ values $\lambda_1 > \ldots > \lambda_m$.

2. For $i = 1, \ldots, m$, initialize $\beta^{(0)}(\lambda_i)$ at the elastic net solution for the corresponding λ_i. Initialize $\theta^{(0)} = \mathbf{0}$.

3. For $k = 0, 1, \ldots$ until convergence:

    (a) Set $\Delta\theta$ to be the component-wise mean/median of $\left.\frac{\partial J_{\lambda_i, \alpha}}{\partial \theta}\right|_{\beta=\beta^{(k)}, \theta=\theta^{(k)}}$ over $i = 1, \ldots, m$.

    (b) Set $\theta^{(k+1)} = \theta^{(k)} - \eta\Delta\theta$, where η is the step size computed via backtracking line search to ensure that the mean/median of $J_{\lambda_i, \alpha}\left(\beta^{(k)}, \theta^{(k+1)}\right)$ over $i = 1, \ldots, m$ is less than that for $J_{\lambda_i, \alpha}\left(\beta^{(k)}, \theta^{(k)}\right)$.

    (c) For $i = 1, \ldots, m$, set $\beta^{(k+1)}(\lambda_i) =$ elastic net solution for λ_i where the penalty factor for feature j is $w_j(\theta^{(k+1)})$.

4   A simulation study

We tested the performance of fwelnet against other methods in a simulation study. In the three settings studied, the true signal is a linear combination of the columns of $\mathbf{X}$, with the true coefficient vector β being sparse. The response $\mathbf{y}$ is the signal corrupted by additive Gaussian noise. In each setting, we gave different types of feature information to fwelnet to determine the method's effectiveness.

For all methods, we used cross-validation (CV) to select the tuning parameter λ. Unless otherwise stated, the α hyperparameter was set to 1 (i.e. no ℓ_2 squared penalty) and Step 3 of Algorithm 1 was run for one iteration, with the mean used for Steps 3(a) and 3(b). To compare methods, we considered the mean squared error (MSE) $MSE = \mathbb{E}\left[(\hat{y} - \mu)^2\right]$ achieved on 10,000 test points, as well as the true positive rate (TPR) and false positive rate (FPR) of the fitted models. (The oracle model which knows the true coefficient vector β has a test MSE of 0.) We ran each simulation 30 times to get estimates for these quantities. (See Appendix A for details of the simulations.)

4.1   Setting 1: Noisy version of the true $|\beta|$

In this setting, we have $n = 100$ observations and $p = 50$ features, with the true signal being a linear combination of just the first 10 features. The feature information matrix **Z** has two columns: a noisy version of $|\beta|$ and a column of ones.

We compared fwelnet against the lasso (using the glmnet package) across a range of signal-to-noise ratios (SNR) in both the response **y** and the feature information matrix **Z** (see details in Appendix A.1). The results are shown in Figure 2. As we would expect, the test MSE figures for both methods decreased as the SNR in the response increased. The improvement of fwelnet over the lasso increased as the SNR in **Z** increased. In terms of feature selection, fwelnet appeared to have similar TPR as the lasso but had smaller FPR.

4.2   Setting 2: Grouped data setting

In this setting, we have $n = 100$ observations and $p = 150$ features, with the features coming in 15 groups of size 10. The feature information matrix $\mathbf{Z} \in \mathbb{R}^{150 \times 15}$ contains group membership information for the features: $z_{jk} = 1\{\text{feature } j \in \text{group } k\}$. We compared fwelnet against the lasso and the group lasso (using the grpreg package) across a range of signal-to-noise ratios (SNR) in the response **y**.

We considered two different responses in this setting. The first response we considered was a linear combination of the features in the first group only, with additive Gaussian noise. The results are depicted in Figure 3. In terms of test MSE, fwelnet was competitive with the group lasso in the low SNR scenario and came out on top for the higher SNR settings. In terms of feature selection, fwelnet had comparable TPR as the group lasso but drastically smaller FPR. Fwelnet had better TPR and FPR than the lasso in this case. We believe that fwelnet's improvement over the group lasso could be because the true signal was sparse: fwelnet's connection to the ℓ_1 version of the group lasso (see Section 6 for details) encourages greater sparsity than the usual group lasso penalty based on ℓ_2 norms.

The second response we considered in this setting was not as sparse in the features: the true signal was a linear combination of the first 4 feature groups. The results are shown in Figure 4. In this case, the group lasso performed better than fwelnet when the hyperparameter α was fixed at 1, which is in line with our earlier intuition that fwelnet would perform better in sparser settings. It is worth noting that fwelnet with $\alpha = 1$ performs appreciably better than the lasso when the SNR is higher. Selecting α via cross-validation improved the test MSE

performance of fwelnet, but not enough to outperform the group lasso. The improvement in test MSE also came at the expense of very high FPR.

4.3   Setting 3: Noise variables

In this setting, we have $n = 100$ observations and $p = 100$ features, with the true signal being a linear combination of just the first 10 features. The feature information matrix $\mathbf{Z}$ consists of 10 noise variables that have nothing to do with the response. Since fwelnet is adapting to these features, we expect it to perform worse than comparable methods.

We compare fwelnet against the lasso across a range of signal-to-noise ratios (SNR) in the response $\mathbf{y}$. The results are depicted in Figure 5. As expected, fwelnet has higher test MSE than the lasso, but the decrease in performance is not drastic. Fwelnet attained similar FPR and TPR to the lasso.

5   Application: Early prediction of preeclampsia

Preeclampsia is a leading cause of maternal and neonatal morbidity and mortality, affecting 5 to 10 percent of all pregnancies. The biological and phenotypical signals associated with late-onset preeclampsia strengthen during the course of pregnancy, often resulting in a clinical diagnosis after 20 weeks of gestation (Zeisler et al. 2016). An earlier test for prediction of late-onset preeclampsia will enable timely interventions for improvement of maternal and neonatal outcomes (Jabeen et al. 2011). In this example, we seek to leverage data collected in late pregnancy to guide the optimization of a predictive model for early diagnosis of late-onset preeclampsia.

We used a dataset of plasma proteins measured during various gestational ages of pregnancy (Erez et al. 2017). For this example, we considered time points   20 weeks "early" and time points > 20 weeks as "late". We had measurements for between 2 to 6 time points for each of the 166 patients for a total of 666 time point observations. Protein measurements were log-transformed to reduce skew. We first split the patients equally into two buckets. For patients in the first bucket, we used only their late time points (83 patients with 219 time points) to train an elastic net model with $\alpha = 0.5$ to predict whether the patient would have preeclampsia. From this late time point model, we extracted model coefficients at the λ hyperparameter value which gave the highest 10-fold cross-validated (CV) area under the curve (AUC). For patients in the second bucket, we used only their early time points (83 patients with 116 time points) to train a fwelnet model with the absolute values of the late time point model coefficients as feature information. When performing CV, we made sure that observations from one patient all belonged to the same CV fold to avoid "contamination" of the held-out fold. One can also run the fwelnet model with additional sources of feature information for each of the proteins.

We compare the 10-fold CV AUC for fwelnet run with 1, 2 and 5 minimization iterations (i.e. Step 3 in Algorithm 1) against the lasso as a baseline. Figure 6 shows a plot of 10-fold CV AUC for these methods against the number of features with non-zero coefficients in the model. The lasso obtains a maximum CV AUC of 0.80, while fwelnet with 2 minimization iterations obtains the largest CV AUC of 0.86 among all methods.

We note that the results were somewhat dependent on (i) how the patients were split into the early and late time point models, and (ii) how patients were split into CV folds when training each of these models. We found that if the late time point model had few non-zero coefficients, then the fwelnet model for the early time point data was very similar to that for the lasso. This matches our intuition: if there are few non-zero coefficients, then we are injecting very little additional information through the relative penalty factors in fwelnet, and so it will give a very similar model to elastic net. Nevertheless, we did not encounter cases where running fwelnet gave worse CV AUC than the lasso.

6   Connection to the group lasso

One common setting where "features of features" arise naturally is when the features come in non-overlapping groups. Assume that the features in $\mathbf{X}$ come in K non-overlapping groups. Let p_k denote the number of features in group k, and let $\beta^{(k)}$ denote the subvector of β which belongs to group k. Assume also that $\mathbf{y}$ and the columns of $\mathbf{X}$ are centered so that $\hat{\beta}_0 = 0$. In this setting, Yuan & Lin (2006) introduced the group lasso estimate as the solution to the optimization problem

$$\underset{\beta}{\text{minimize}}\frac{1}{2}\left\|\mathbf{y} - \mathbf{X}\beta\right\|_2^2 + \lambda \sum_{k=1}^{K} \left\|\beta^{(k)}\right\|_2. \tag{10}$$

The ℓ_2 penalty on features at the group level ensures that features belonging to the same group are either all included in the model or all excluded from it. Often, the penalty given to group k is modified by a factor of $\sqrt{p_k}$ to take into account the varying group sizes:

$$\hat{\beta}_{gl,2}(\lambda) = \underset{\beta}{\text{argmin}}\frac{1}{2}\left\|\mathbf{y} - \mathbf{X}\beta\right\|_2^2 + \lambda \sum_{k=1}^{K} \sqrt{p_k}\left\|\beta^{(k)}\right\|_2. \tag{11}$$

Theorem 1 below establishes a connection between fwelnet and the group lasso. For the moment, consider the more general penalty factor $w_j(\theta) = \frac{\sum_{\ell=1}^{p} f(\mathbf{z}_\ell^T\theta)}{pf(\mathbf{z}_j^T\theta)}$, where f is some function with range $[0,+\infty)$. (Fwelnet makes the choice $f(x) = e^x$.)

Theorem 1. *If the "features of features" matrix $\mathbf{Z} \in \mathbb{R}^{p \times K}$ is given by $z_{jk} = 1\{$feature $j \in$ group $k\}$, then minimizing the fwelnet objective function* (7) *jointly over β_0, β and θ reduces to*

$$\underset{\beta}{argmin}\frac{1}{2}\left\|\mathbf{y}-\mathbf{X}\beta\right\|_2^2 + \lambda'\sum_{k=1}^{K}\sqrt{p_k\left[\alpha\left\|\beta^{(k)}\right\|_1 + \frac{1-\alpha}{2}\left\|\beta^{(k)}\right\|_2^2\right]}$$

$$= \begin{cases} \underset{\beta}{argmin}\ \frac{1}{2}\left\|\mathbf{y}-\mathbf{X}\beta\right\|_2^2 + \lambda'\sum_{k=1}^{K}\sqrt{p_k}\left\|\beta^{(k)}\right\|_2 & \text{if } \alpha = 0, \\ \underset{\beta}{argmin}\ \frac{1}{2}\left\|\mathbf{y}-\mathbf{X}\beta\right\|_2^2 + \lambda'\sum_{k=1}^{K}\sqrt{p_k}\left\|\beta^{(k)}\right\|_1 & \text{if } \alpha = 1, \end{cases}$$

for some $\lambda' \geq 0$.

We recognize the $\alpha = 0$ case as minimizing the residual sum of squares (RSS) and the group lasso penalty, while the $\alpha = 1$ case is minimizing the RSS and the ℓ_1 version of the group lasso penalty. The proof of Theorem 1 can be found in Appendix B.

7 Using fwelnet for multi-task learning

We turn now to an application of fwelnet to *multi-task learning*. In some applications, we have a single model matrix $\mathbf{X}$ but are interested in multiple responses $\mathbf{y}_1, \ldots \mathbf{y}_B$. If there is some common structure between the signals in the B responses, it can be advantageous to fit models for them simultaneously. This is especially the case if the signal-to-noise ratios in the responses are low.

We demonstrate how fwelnet can be used to learn better models in the setting with two responses, $\mathbf{y}_1$ and $\mathbf{y}_2$. The idea is to use the absolute value of coefficients of one response as the external information for the other response. That way, a feature which has larger influence on one response is likely to be given a correspondingly lower penalty weight when fitting the other response. Algorithm 2 presents one possible way of doing so.

We tested the effectiveness of Algorithm 2 (with step 2 run for 3 iterations) on simulated data. We generate 150 observations with 50 independent features. The signal in response 1 is a linear combination of features 1 to 10, while the signal in response 2 is a linear combination of features 1 to 5 and 11 to 15. The coefficients are set such that those for the common features (i.e. features 1 to 5) have larger absolute value than those for the features specific to one response. The signal-to-noise ratios (SNRs) in response 1 and response 2 are 0.5 and 1.5 respectively. (See Appendix C for more details of the simulation.)

We compared Algorithm 2 against: (i) the *individual lasso (ind_lasso)*, where the lasso is run separately for $\mathbf{y}_1$ and $\mathbf{y}_2$; and (ii) the *multi-response lasso (mt_lasso)*, where coefficients belonging to the same feature across the responses are given a joint ℓ_2 penalty. Because of the ℓ_2 penalty, a feature is either included or excluded in the model for all the responses at the same time.

The results are shown in Figure 7 for 50 simulation runs. Fwelnet outpeforms the other two methods in test MSE as evaluated on 10,000 test points. As expected, the lasso run individually for each response performs well in the response with higher SNR but poorly

in the response with lower SNR. The multi-response lasso is able to borrow strength from the higher SNR response to obtain good performance on the lower SNR response. However, because the models for both responses are forced to consist of the same set of features, performance suffers on the higher SNR response. Fwelnet has the ability to borrow strength across responses without being hampered by this restriction.

Algorithm 2 *Using fwelnet for multi-task learning*

1. Initialize $\beta_1^{(0)}$ and $\beta_2^{(0)}$ at the `lambda.min` elastic net solutions for $(\mathbf{X}, \mathbf{y}_1)$ and $(\mathbf{X}, \mathbf{y}_2)$ respectively, that is, the value of the hyperparameter λ which minimizes cross-validated error.

2. For $k = 0, 1, \ldots$ until convergence:

    (a) Set $\mathbf{Z}_2 = \left|\beta_1^{(k)}\right|$. Run fwelnet with $(\mathbf{X}, \mathbf{y}_2, \mathbf{Z}_2)$ and set $\beta_2^{(k+1)}$ to be the `lambda.min` solution.

    (b) Set $\mathbf{Z}_1 = \left|\beta_2^{(k+1)}\right|$. Run fwelnet with $(\mathbf{X}, \mathbf{y}_1, \mathbf{Z}_1)$ and set $\beta_1^{(k+1)}$ to be the `lambda.min` solution.

---

8   Discussion

In this paper, we have proposed organizing information about predictor variables, which we call "features of features", as a matrix $\mathbf{Z} \in \mathbb{R}^{p \times K}$, and modifying model-fitting algorithms by assigning each feature a score, $\mathbf{z}_j^T \theta$, based on this auxiliary information. We have proposed one such method, the feature-weighted elastic net ("fwelnet"), which imposes a penalty modification factor $w_j(\theta) = \dfrac{\sum_{\ell=1}^{p} \exp(\mathbf{z}_\ell^T \theta)}{p \exp(\mathbf{z}_j^T \theta)}$ for the elastic net algorithm.

There is much scope for future work:

- *Choice of penalty modification factor.* While the penalty modification factors $p_j(\theta)$ we have proposed works well in practice and has several desirable properties, we make no claim about its optimality. We also do not have well-developed theory for the current choice penalty factor.

- *Extending the use of scores beyond the elastic net.* The use of feature scores $\mathbf{z}_j^T \theta$ in modifying the weight given to each feature in the model-fitting process is a general idea that could apply to any supervised learning algorithm. More work needs to be done on how such scores can be incorporated, with particular focus on how θ can be learned through the algorithm.

- *Whether θ should be treated as a parameter or a hyperparameter, and how to determine its value.* In this paper, we introduced θ as a hyperparameter for (7). This formulation gives us a clear interpretation for θ: θ_k is a proxy for how important the kth source of feature information is for identifying which features are important. With this interpretation, we do not expect θ to change across λ values.

  When θ is treated as a hyperparameter, we noted that the time needed for a grid search to find its optimal value grows exponentially with the number of sources of feature information. To avoid this computational burden, we suggested a descent algorithm for θ based on its gradient with respect to the fwelnet objective function (Step 3(a) and 3(b) in Algorithm 1). There are other methods for hyperparameter optimization such as random search (e.g. Bergstra & Bengio (2012)) or Bayesian optimization (e.g. Snoek et al. (2012)) that could be applied to this problem.

One could consider θ as an argument of the fwelnet objective function to be minimized over jointly with β. One benefit of this approach is that it gives us a theoretical connection to the group lasso (Section 6). However, we will obtain different estimates of θ for each value of the hyperparameter λ, which may be undesirable for interpretation. The objective function is also not jointly convex in θ and β, meaning that different minimization algorithms could end up at different local minima. In our attempts to make this approach work (see Appendix A), it did not fare as well in prediction performance and was computationally expensive. It remains to be seen if there is a computationally efficient algorithm which treats θ as a parameter to be optimized for each λ value.

An R language package fwelnet which implements our method is available at https://www.github.com/kjytay/fwelnet.

Acknowledgements:

Appendix

Appendix

A   Alternative algorithm with θ as a parameter

Assume that $\mathbf{y}$ and the columns of $\mathbf{X}$ are centered so that $\hat{\beta}_0 = 0$ and we can ignore the intercept term in the rest of the discussion. If we consider θ as an argument of the objective function, then we wish to solve

$$
(\hat{\beta}, \hat{\theta}) = \operatorname*{argmin}_{\beta, \theta} J_{\lambda, \alpha}(\beta, \theta)
$$

$$
= \operatorname*{argmin}_{\beta, \theta} \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\beta \right\|_2^2 + \lambda \sum_{j=1}^{p} w_j(\theta) \left[\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right].
$$

J is not jointly convex β and θ, so reaching a global minimum is a difficult task. Instead, we content ourselves with reaching a local minimum. A reasonable approach for doing so is to alternate between optimizing β and θ: the steps are outlined in Algorithm 3.

Algorithm 3 *Minimizing the fwelnet objective function via alternating minimization*

1. Select a value of $\alpha \in [0, 1]$ and a sequence of λ values $\lambda_1 > \ldots > \lambda_m$.

2. For $i = 1, \ldots, m$:

    (a) Initialize $\beta^{(0)}(\lambda_i)$ at the elastic net solution for λ_i. Initialize $\theta^{(0)} = \mathbf{0}$.

    (b) For $k = 0, 1, \ldots$ until convergence:

        i. Fix $\beta = \beta^{(k)}$, update $\theta^{(k+1)}$ via gradient descent. That is, set $\Delta\theta = \left. \frac{\partial J_{\lambda, \alpha}}{\partial \theta} \right|_{\beta = \beta^{(k)}, \theta = \theta^{(k)}}$ and update $\theta^{(k+1)} = \theta^{(k)} - \eta \Delta\theta$, where η is the step size computed via backtracking line search to ensure that $J_{\lambda, \alpha}\left(\beta^{(k)}, \theta^{(k+1)}\right) < J_{\lambda, \alpha}\left(\beta^{(k)}, \theta^{(k)}\right)$.

        ii. Fix $\theta = \theta^{(k+1)}$, update $\beta^{(k+1)}$ by solving the elastic net with updated penalty factors $w_j(\theta^{(k+1)})$.

Unfortunately, Algorithm 3 is slow due to repeated solving of the elastic net problem in Step 2(b)ii for each λ_i. The algorithm does not take advantage of the fact that once α and θ are fixed, the elastic net problem can be solved quickly for an entire path of λ values. We have also found that Algorithm 3 does not predict as well as Algorithm 1 in our simulations.

A Details on simulation study in Section 4

A.1 Setting 1: Noisy version of the true β

1. Set $n = 100$, $p = 50$, $\beta \in \mathbb{R}^{50}$ with $\beta_j = 2$ for $j = 1, \ldots, 5$, $\beta_j = -1$ for $j = 6, \ldots, 10$, and $\beta_j = 0$ otherwise.

2. Generate $x_{ij} \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, p$.

3. For each $SNR_y \in \{0.5, 1, 2\}$ and $SNR_Z \in \{0.5, 2, 10\}$:

    a. Compute $\sigma_y^2 = \left(\sum_{j=1}^{p} \beta_j^2\right)/SNR_y$.

    b. Generate $y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i$, where $\varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_y^2)$ for $i = 1, \ldots, n$.

    c. Compute $\sigma_z^2 = \text{Var}(|\beta|)/SNR_Z$.

    d. Generate $z_j = |\beta_j| + \eta_j$, where $\eta_j \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_z^2)$. Append a column of ones to get $\mathbf{Z} \in \mathbb{R}^{p \times 2}$.

A.2 Setting 2: Grouped data setting

1. Set $n = 100$, $p = 150$.

2. For $j = 1, \ldots, p$ and $k = 1, \ldots 15$, set $z_{jk} = 1$ if $10(k-1) < j \leq 10k$, $z_{jk} = 0$ otherwise.

3. Generate $\beta \in \mathbb{R}^{150}$ with $\beta_j = 3$ or $\beta_j = -3$ with equal probability for $j = 1, \ldots, 10$, $\beta_j = 0$ otherwise.

4. Generate $x_{ij} \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, p$.

5. For each $SNR_y \in \{0.5, 1, 2\}$:

    a. Compute $\sigma_y^2 = \left(\sum_{j=1}^{p} \beta_j^2\right)/SNR_y$.

    b. Generate $y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i$, where $\varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_y^2)$ for $i = 1, \ldots, n$.

A.3 Setting 3: Noise variables

1. Set $n = 100$, $p = 100$, $\beta \in \mathbb{R}^{100}$ with $\beta_j = 2$ for $j = 1, \ldots, 10$, and $\beta_j = 0$ otherwise.

2. Generate $x_{ij} \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, p$.

3. For each $SNR_y \in \{0.5, 1, 2\}$:

    a. Compute $\sigma_y^2 = \left(\sum_{j=1}^{p} \beta_j^2\right)/SNR_y$.

    b. Generate $y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i$, where $\varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_y^2)$ for $i = 1, \ldots, n$.

c. Generate $z_{jk} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ for $j = 1, \ldots, p$ and $k = 1, \ldots 10$. Append a column of ones to get $\mathbf{Z} \in \mathbb{R}^{p \times 11}$.

B Proof of Theorem 1

First note that if feature j belongs to group k, then $\mathbf{z}_j^T \theta = \theta_k$, and its penalty factor is

$$w_j(\theta) = \frac{\sum_{\ell=1}^p f(\mathbf{z}_\ell^T \theta)}{p f(\mathbf{z}_j^T \theta)} = \frac{\sum_{\ell=1}^p f(\theta_\ell)}{p f(\theta_k)} = \frac{\sum_{\ell=1}^K p_\ell f(\theta_\ell)}{p f(\theta_k)},$$

where p_ℓ denotes the number of features in group ℓ. Letting $v_k = \dfrac{f(\theta_k)}{\sum_{\ell=1}^K p_\ell f(\theta_\ell)}$ for $k = 1, \ldots, K$, minimizing the fwelnet objective function (7) over β and θ reduces to

$$\underset{\beta, \theta}{\text{minimize}} \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\beta \right\|_2^2 + \frac{\lambda}{p} \sum_{k=1}^K \frac{1}{v_k} \left[\alpha \left\| \beta^{(k)} \right\|_1 + \frac{1-\alpha}{2} \left\| \beta^{(k)} \right\|_2^2 \right].$$

For fixed β, we can explicitly determine the v_k values which minimize the expression above. By the Cauchy-Schwarz inequality,

$$\frac{\lambda}{p} \sum_{k=1}^K \frac{1}{v_k} \left[\alpha \left\| \beta^{(k)} \right\|_1 + \frac{1-\alpha}{2} \left\| \beta^{(k)} \right\|_2^2 \right] = \frac{\lambda}{p} \left(\sum_{k=1}^K \frac{1}{v_k} \left[\alpha \left\| \beta^{(k)} \right\|_1 + \frac{1-\alpha}{2} \left\| \beta^{(k)} \right\|_2^2 \right] \right)$$

$$\left(\sum_{k=1}^K p_k v_k \right) \geq \frac{\lambda}{p} \left(\sum_{k=1}^K \sqrt{p_k \left[\alpha \left\| \beta^{(k)} \right\|_1 + \frac{1-\alpha}{2} \left\| \beta(k) \right\|_2^2 \right]} \right)^2. \tag{12}$$

Note that equality is attainable for (12): letting $a_k = \sqrt{\dfrac{\left[\alpha \left\| \beta^{(k)} \right\|_1 + \frac{1-\alpha}{2} \left\| \beta^{(k)} \right\|_2^2 \right]}{p_k}}$, equality occurs when there is some $c \in \mathbb{R}$ such that

$$c \cdot \frac{1}{v_k} \left[\alpha \left\| \beta^{(k)} \right\|_1 + \frac{1-\alpha}{2} \left\| \beta^{(k)} \right\|_2^2 \right] = p_k v_k \text{ for all } k,$$

$$v_k = \sqrt{c} a_k \text{ for all } k.$$

Since $\sum_{k=1}^K p_k v_k = 1$, we have $\sqrt{c} = \dfrac{1}{\sum_{k=1}^K p_k a_k}$, giving $v_k = \dfrac{a_k}{\sum_{k=1}^K p_k a_k}$ for all k. A solution for this is $f(\theta_k) = a_k$ for all k, which is feasible for f having range $[0, \infty)$. (Note that if f only has range $(0, \infty)$, the connection still holds if $\lim_{x \to -\infty} f(x) = 0$ or $\lim_{x \to +\infty} f(x) = 0$: the solution will just have $\theta = +\infty$ or $\theta = -\infty$.)

Thus, the fwelnet solution is

$$\underset{\beta}{\arg\min} \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\beta \right\|_2^2 + \frac{\lambda}{p} \left(\sum_{k=1}^K \sqrt{p_k \left[\alpha \left\| \beta^{(k)} \right\|_1 + \frac{1-\alpha}{2} \left\| \beta^{(k)} \right\|_2^2 \right]} \right)^2. \tag{13}$$

Writing in constrained form, (13) becomes minimizing $\frac{1}{2}\left\|\mathbf{y} - \mathbf{X}\beta\right\|_2^2$ subject to

$$\left(\sum_{k=1}^{K}\sqrt{p_k\left[\alpha\|\beta^{(k)}\|_1 + \frac{1-\alpha}{2}\|\beta^{(k)}\|_2^2\right]}\right)^2 \leq C \text{ for some constant } C,$$

$$\sum_{k=1}^{K}\sqrt{p_k\left[\alpha\|\beta^{(k)}\|_1 + \frac{1-\alpha}{2}\|\beta^{(k)}\|_2^2\right]} \leq \sqrt{C}.$$

Converting back to Lagrange form again, there is some $\lambda' \quad 0$ such that the fwelnet solution is

$$\underset{\beta}{\operatorname{argmin}}\frac{1}{2}\left\|\mathbf{y} - \mathbf{X}\beta\right\|_2^2 + \lambda'\sum_{k=1}^{K}\sqrt{p_k\left[\alpha\|\beta^{(k)}\|_1 + \frac{1-\alpha}{2}\|\beta^{(k)}\|_2^2\right]}.$$

Setting $\alpha = 0$ and $\alpha = 1$ in the expression above gives the desired result.

C   Details on simulation study in Section 7

1.   Set $n = 150$, $p = 50$.

2.   Generate $\beta_1 \in \mathbb{R}^{50}$ with

$$\beta_{1,j} = \begin{cases} 5 \text{ or } -5 \text{ with equal probability} & \text{for } j = 1, \dots, 5, \\ 2 \text{ or } -2 \text{ with equal probability} & \text{for } j = 6, \dots, 10, \\ 0 & \text{otherwise.} \end{cases}$$

3.   Generate $\beta_2 \in \mathbb{R}^{50}$ with

$$\beta_{2,j} = \begin{cases} 5 \text{ or } -5 \text{ with equal probability} & \text{for } j = 1, \dots, 5, \\ 2 \text{ or } -2 \text{ with equal probability} & \text{for } j = 11, \dots, 15, \\ 0 & \text{otherwise.} \end{cases}$$

4.   Generate $X_{ij} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ for $i = 1, \dots, n$ and $j = 1, \dots, p$.

5.   Generate response 1, $\mathbf{y}_1 \in \mathbb{R}^{150}$, in the following way:

    a.   Compute $\sigma_1^2 = \left(\sum_{j=1}^{p}\beta_{1,j}^2\right)/0.5$.

    b.   Generate $y_{1,i} = \sum_{j=1}^{p}x_{ij}\beta_{1,j} + \varepsilon_{1,i}$ where $\varepsilon_{1,i}\overset{i.i.d.}{\sim}\mathcal{N}(0,\sigma_1^2)$ for $i = 1, \dots, n$.

6.   Generate response 2, $\mathbf{y}_2 \in \mathbb{R}^{150}$, in the following way:

    a.   Compute $\sigma_2^2 = \left(\sum_{j=1}^{p}\beta_{2,j}^2\right)/1.5$.

    b.   Generate $y_{2,i} = \sum_{j=1}^{p}x_{ij}\beta_{2,j} + \varepsilon_{2,i}$, where $\varepsilon_{2,i}\overset{i.i.d}{\sim}\mathcal{N}(0,\sigma_2^2)$ for $i = 1, \dots, n$.

References

Bergersen LC, Glad IK & Lyng H (2011), 'Weighted lasso with data integration', Statistical Applications in Genetics and Molecular Biology 10(1).

Bergstra J & Bengio Y (2012), 'Random search for hyper-parameter optimization', Journal of Machine Learning Research 13, 281–305.

Boulesteix A-L, De Bin R, Jiang X & Fuchs M (2017), ' IPF-LASSO: Integrative L1 -Penalized Regression with Penalty Factors for Prediction Based on Multi-Omics Data ', Computational and Mathematical Methods in Medicine 2017, 1–14.

Erez O, Romero R, Maymon E, Chaemsaithong P, Done B, Pacora P, Panaitescu B, Chaiworapongsa T, Hassan SS & Tarca AL (2017), 'The prediction of early preeclampsia: Results from a longitudinal proteomics study', PLoS ONE 12(7), 1–28.

Friedman J, Hastie T & Tibshirani R (2010), 'Regularization Paths for Generalized Linear Models via Coordinate Descent', Journal of Statistical Software 33(1), 1–24. [PubMed: 20808728]

Hoerl AE & Kennard RW (1970), 'Ridge regression: Biased estimation for nonorthogonal problems', Technometrics 12(1), 55–67.

Jabeen M, Yakoob MY, Imdad A & Bhutta ZA (2011), 'Impact of interventions to prevent and manage preeclampsia and eclampsia on stillbirths', BMC Public Health 11(S3), S6.

Jacob L, Obozinski G & Vert J-P (2009), Group Lasso with overlap and graph Lasso, in 'Proceedings of the 26th annual international conference on machine learning', pp. 433–440.

Mollaysa A, Strasser P & Kalousis A (2017), 'Regularising non-linear models using feature side-information', Proceedings of the 34th International Conference on Machine Learning pp. 2508–2517.

Slawski M, zu Castell W & Tutz G (2010), 'Feature selection guided by structural information', Annals of Applied Statistics 4(2), 1056–1080.

Snoek J, Larochelle H & Adams RP (2012), Practical Bayesian optimization of machine learning algorithms, in 'Advances in neural information processing systems', pp. 2951–2959.

Tai F & Pan W (2007), 'Incorporating prior knowledge of predictors into penalized classifiers with multiple penalty terms', Bioinformatics 23(14), 1775–1782. [PubMed: 17483507]

Tibshirani R (1996), 'Regression Shrinkage and Selection via the Lasso', Journal of the Royal Statistical Society. Series B (Methodological) 58(1), 267–288.

Tibshirani R, Saunders M, Rosset S, Zhu J & Knight K (2005), 'Sparsity and smoothness via the fused lasso', Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67(1), 91–108.

van de Wiel MA, Lien TG, Verlaat W, van Wieringen WN & Wilting SM (2016), 'Better prediction by use of co-data: adaptive group-regularized ridge regression', Statistics in Medicine 35(3), 368–381. [PubMed: 26365903]

Velten B & Huber W (2018), 'Adaptive penalization in high-dimensional regression and classification with external covariates using variational Bayes', arXiv preprint arXiv:1811.02962.

Yuan M & Lin Y (2006), 'Model Selection and Estimation in Regression with Grouped Variables', Journal of the Royal Statistical Society. Series B (Statistical Methodology) 68(1), 49–67.

Zeisler H, Llurba E, Chantraine F, Vatish M, Staff AC, Sennström M, Olovsson M, Brennecke SP, Stepan H, Allegranza D, Dilba P, Schoedl M, Hund M & Verlohren S (2016), 'Predictive value of the sFlt-1:PlGF ratio in women with suspected preeclampsia', New England Journal of Medicine 374(1), 13–22. [PubMed: 26735990]

Zou H (2006), 'The adaptive lasso and its oracle properties', Journal of the American Statistical Association 101(476), 1418–1429.

Zou H & Hastie T (2005), 'Regularization and variable selection via the elastic net', Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67(2), 301–320.

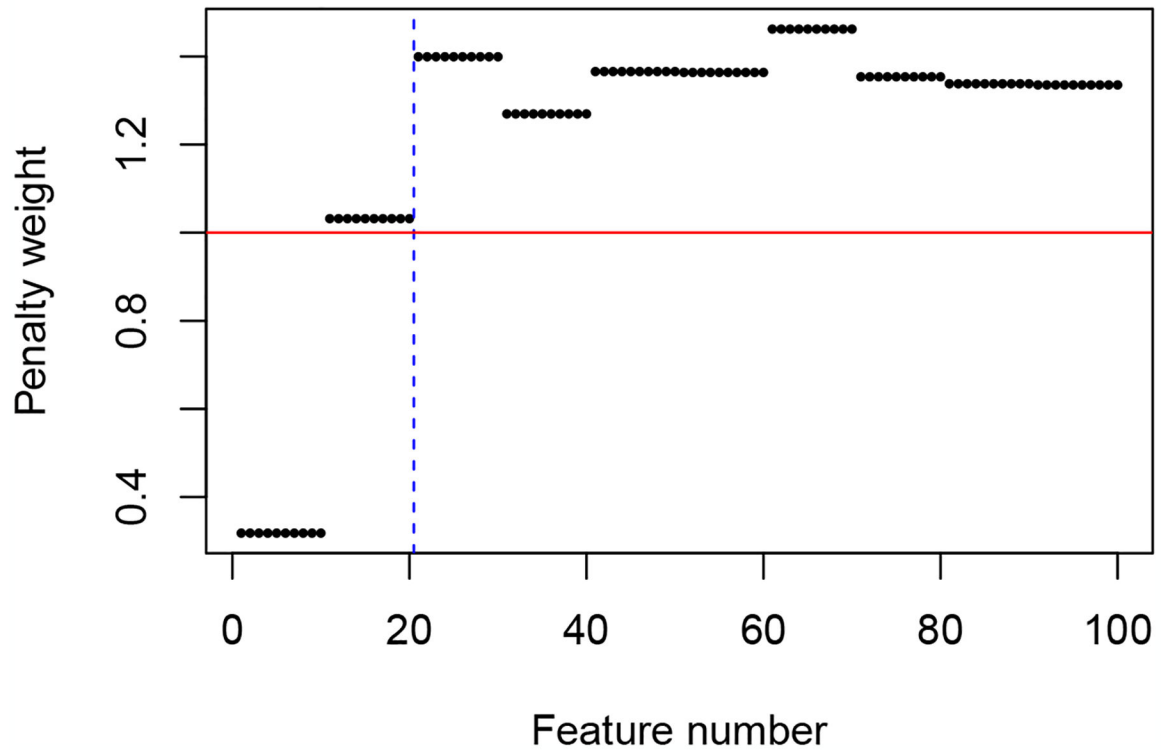# Penalty weight for each feature

Figure 1:

Penalty factors which fwelnet assigns to each feature. $n = 200$, $p = 100$ with features in groups of size 10. The response is a noisy linear combination of the first two groups, with signal in the first group being stronger than that in the second. As expected, fwelnet's penalty weights for the true features (left of blue dotted line) are lower than that for null features. In elastic net, all features would be assigned a penalty factor of 1 (horizontal red line).
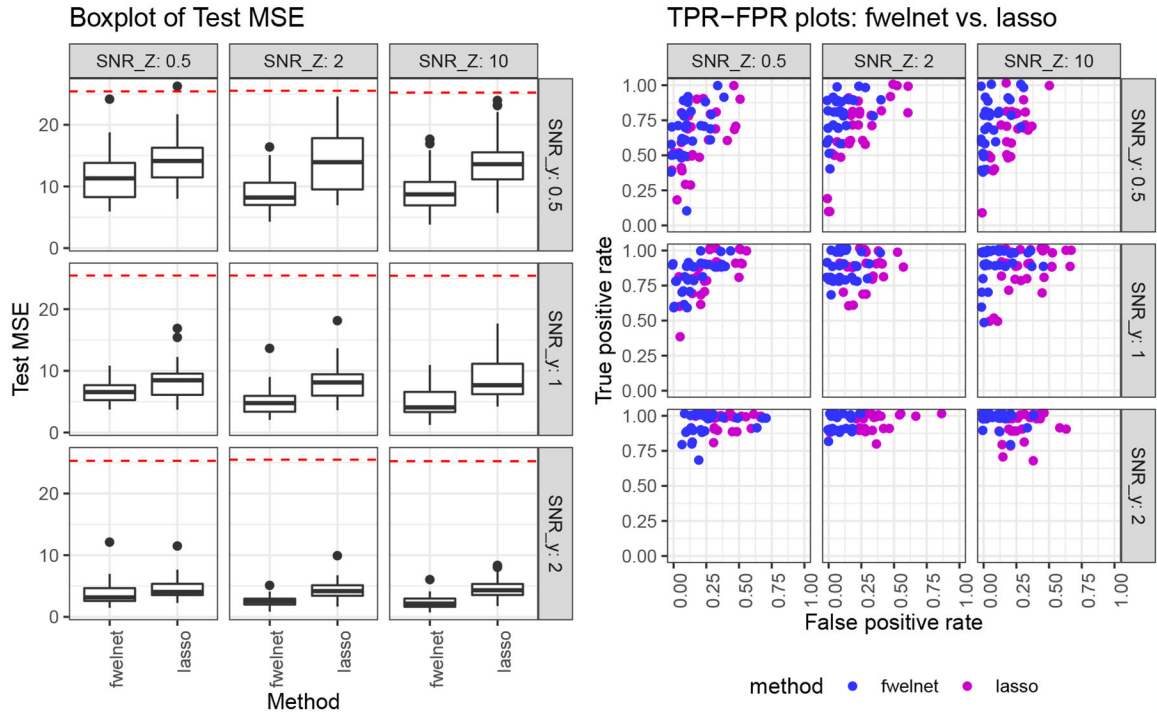
Figure 2:
"Feature of features": noisy version of the true $|\beta|$. $n = 100$, $p = 50$. The response is a linear combination of the first 10 features. As we go from left to right, the signal-to-noise ratio (SNR) for **y** increases; as we go from top to bottom, the SNR in **Z** increases. The panel on the left shows the raw test mean squared error (MSE) figures with the red dotted line indicating the median null test MSE. In the figure on the right, each point depicts the true positive rate (TPR) and false positive rate (FPR) of the fitted model for one of 30 simulation runs. Fwelnet outperforms the lasso in test MSE, with the improvement getting larger as the SNR in **Z** increases. Fwelnet appears to have similar TPR to the lasso but has significantly smaller FPR.
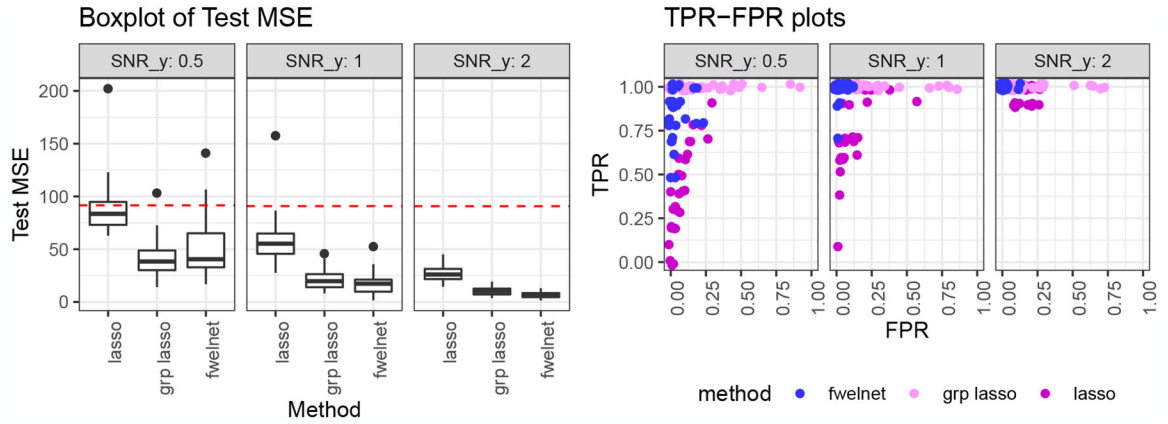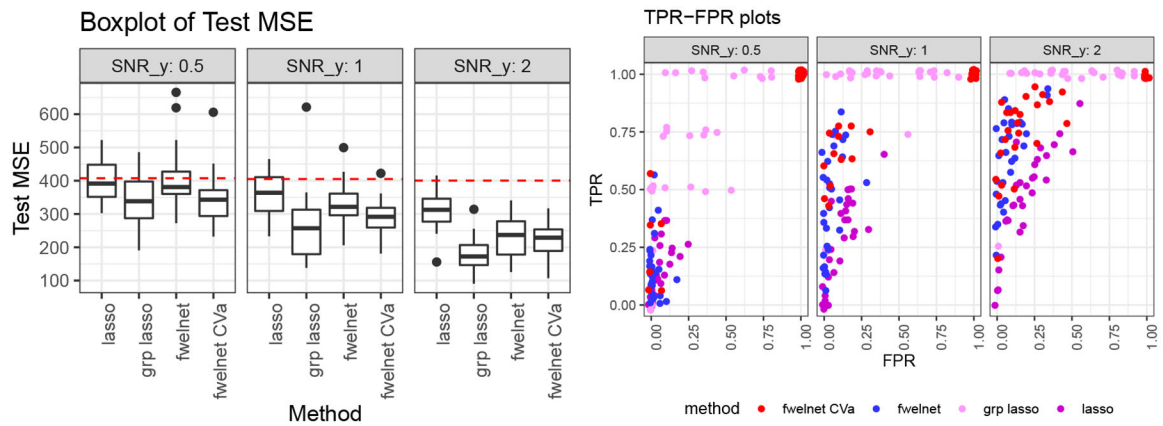
Figure 3:

"Feature of features": grouping data. $n = 100$, $p = 150$. The features come in groups of 10, with the response being a linear combination of the features in the first group. As we go from left to right, the signal-to-noise ratio (SNR) for **y** increases. The figure on the left shows the test mean squared error (MSE) results with the red dotted line indicating the median null test MSE. In the figure on the right, each point depicts the true positive rate (TPR) and false positive rate (FPR) of the fitted model for one of 30 simulation runs. Fwelnet outperforms the group lasso in terms of test MSE at higher SNR levels. Fwelnet has higher TPR than the lasso, and lower FPR than the group lasso.

Tay et al.

Page 20

Figure 4:

"Feature of features": grouping data. $n = 100$, $p = 150$. The features come in groups of 10, with the response being a linear combination of the first 4 groups. As we go from left to right, the signal-to-noise ratio (SNR) for **y** increases. The figure on the left shows the test mean squared error (MSE) results with the red dotted line indicating the median null test MSE. Fwelnet sets $\alpha = 1$ while fwelnet CVa selects α via cross-validation. In the figure on the right, each point depicts the true positive rate (TPR) and false positive rate (FPR) of the fitted model for one of 30 simulation runs. Group lasso performs best here. Cross-validation for α improves test MSE performance but at the expense of having very high FPR.

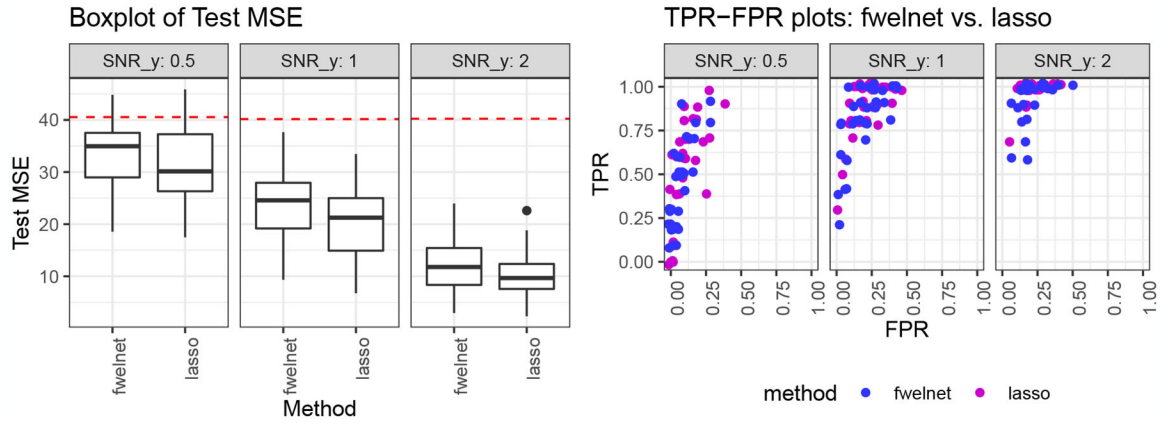*Stat Sin*. Author manuscript; available in PMC 2023 April 25.

Figure 5:
"Feature of features": 10 noise variables. $n = 100$, $p = 100$. The response is a linear combination of the first 10 features. As we go from left to right, the signal-to-noise ratio (SNR) for **y** increases. The figure on the left shows the test mean squared error (MSE) results, with the red dotted line indicating the median null test MSE. In the figure on the right, each point depicts the true positive rate (TPR) and false positive rate (FPR) of the fitted model for one of 30 simulation runs. Fwelnet only performs slightly worse than the lasso in test MSE, and has similar TPR and FPR as the lasso.
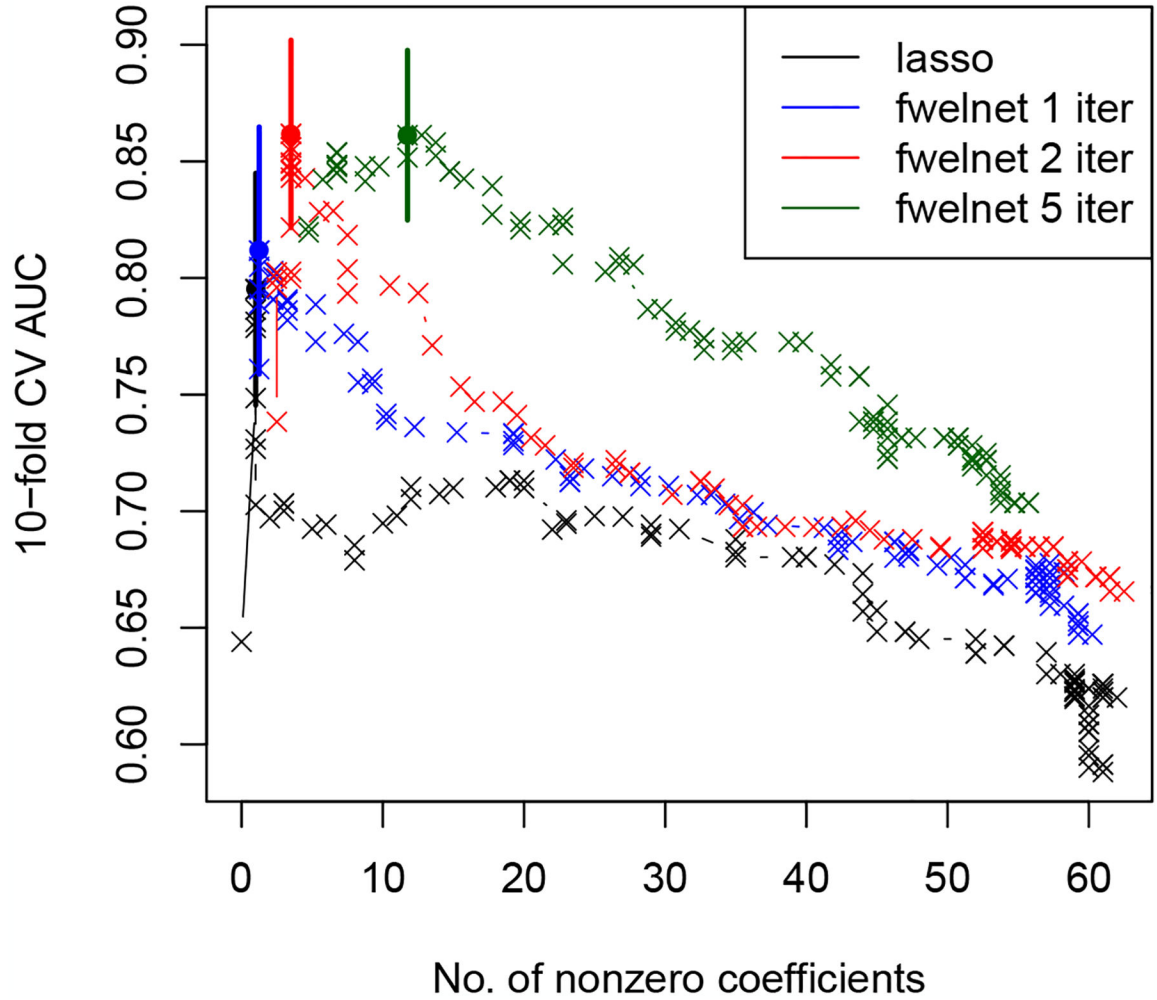
10−fold CV AUC vs. model size

Figure 6:
Early prediction of pre-eclampsia: Plot of 10-fold cross-validated (CV) area under the curve (AUC). 10-fold CV AUC is plotted against the number of non-zero coefficients for each model trained on just early time point data. The baseline method is the lasso; we run fwelnet for 1, 2 and 5 minimization iterations. For each method/algorithm, the model with highest CV AUC is marked by a dot. To reduce clutter in the figure, the ±1 standard error bars are drawn for just these models. Fwelnet with 2 minimization iterations has the largest CV AUC.
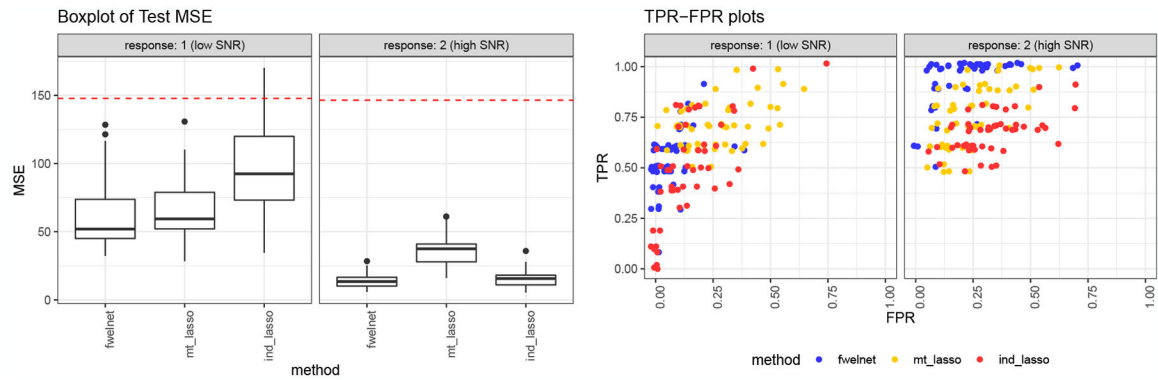
Figure 7:
Application of fwelnet to multi-task learning. n = 150, p = 50. Response 1 is a linear combination of features 1 to 10, while response 2 is a linear combination of features 1 to 5 and 11 to 15. The signal-to-noise ratios (SNR) for responses 1 and 2 are 0.5 and 1.5 respectively. The figure on the left shows the raw test mean squared error (MSE) figures with the red dotted line indicating the median null test MSE. The figure on the right shows the true positive rate (TPR) and false positive rate (FPR) of the fitted model (each point being one of 50 simulation runs). Fwelnet outperforms the individual lasso and the multi-response lasso in test MSE for both responses. Fwelnet also has better TPR and FPR than the other methods.