Statistica Sinica **33** (2023), 259-279
doi:https://doi.org/10.5705/ss.202020.0226

FEATURE-WEIGHTED ELASTIC NET: USING "FEATURES OF FEATURES" FOR "BETTER PREDICTION"

J. Kenneth Tay, Nima Aghaeepour, Trevor Hastie and Robert Tibshirani

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 that 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 the test mean squared error, and usually gives an improvement in terms of the true positive rate or false positive rate for feature selection. We also compare this method with the group lasso and Bayesian estimation. Lastly, we apply the proposed method to the early prediction of preeclampsia, where *fwelnet* outperforms the lasso in terms of the 10-fold cross-validated area under the curve (0.84 vs. 0.80, respectively), and suggest how *fwelnet* might be used for multi-task learning.

Key words and phrases: Feature information, model selection/variable selection, , prediction.

1. Introduction

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

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

with ϵ having mean zero and variance σ^2. 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 OLS estimates, both in terms of prediction accuracy and interpreting the fitted model. One popular regularized estimator is the elastic net (Zou and Hastie (2005)). Letting $\beta = (\beta_1, \ldots, \beta_p)^T$, the elastic net minimizes the objective function

Corresponding author: J. Kenneth Tay, Department of Statistics, Stanford University, Stanford, CA 94305, USA. E-mail: kjytay@stanford.edu.

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

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. Setting $\alpha = 0$ corresponds to the ridge regression (Hoerl and Kennard (1970)), whereas $\alpha = 1$ corresponds to the lasso (Tibshirani (1996)). One reason for the elastic net's popularity is its computational efficiency: J is convex in its parameters, so solutions can be found efficiently, even for very large n and p. In addition, the solution for an entire path of λ-values can be computed quickly using warm starts (Friedman, Hastie and Tibshirani (2010)).

In some settings, we have 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. Methods that leverage such information are likely to perform better prediction and inference than methods that ignore it. However, many popular methods, including the elastic net, do not use such information in the model-fitting process.

In this study, we develop a framework for organizing such feature information, and propose a variant of the elastic net that uses this information in model fitting. We assume that the feature information is quantitative, allowing 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. Let $\mathbf{z}_j \in \mathbb{R}^K$ denote the jth row of $\mathbf{Z}$ as a column vector. We propose assigning each feature a *score* $\mathbf{z}_j^T \theta$, that is, a linear combination of its "features of features," and using these scores to influence the penalty weight in the elastic net penalty:

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

where $w_j(\theta) = f(\mathbf{z}_j^T \theta)$ for some function f. Here, 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 viewed as an indication of how influential feature j is on the response.

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 present connections to the group lasso and Bayesian estimation in Section 4, and illustrate fwelnet's performance on simulated data in Section 5 and on a real-data example in Section 6. 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 online Supplementary Material contains further details and proofs.

2. Related Work

The idea of assigning different penalty weights to features in the lasso or elastic net objective is not new. The adaptive lasso (Zou (2006)) assigns feature j a penalty weight $w_j = 1/|\hat{\beta}_j^{OLS}|^\gamma$, where $\hat{\beta}_j^{OLS}$ is the estimated OLS coefficent for feature j and $\gamma > 0$ is some hyperparameter. However, the OLS solution depends only on $\mathbf{X}$ and $\mathbf{y}$, and does not incorporate any external information. In the work closest to ours, Bergersen, Glad and Lyng (2011) propose using the weights $w_j = 1/|\eta_j(\mathbf{y}, \mathbf{X}, \mathbf{Z})|^q$, 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 η_j 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 prior for the coefficients. Within this Bayesian framework, some methods propose using 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 a ridge regression, whereas Velten and Huber (2021) use variational Bayes to do so for general convex penalties.

Most previous approaches for penalized regression with external information on the features only work with specific types of such information. Several methods have been developed to use *feature grouping information.* Here, popular methods include the group lasso (Yuan and Lin (2006)) and the overlap group lasso (Jacob, Obozinski and Vert (2009)). The integrative lasso with penalty factors (IPF-Lasso) (Boulesteix et al. (2017)) gives each group its own penalty parameter, chosen using cross-validation (CV). Tai and Pan (2007) modify the penalized partial least squares (PLS) and nearest shrunken centroids methods to have group-specific penalties.

Other methods incorporate "network-like" or feature similarity information. The fused lasso (Tibshirani et al. (2005)) adds an ℓ_1-penalty to the successive differences of the coefficients to impose smoothness on the coefficient profile. The structured elastic net (Slawski, zu Castell and Tutz (2010)) generalizes the fused lasso by replacing the ℓ_2-squared penalty in the 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. Li and Li (2008) present a special case of the structured elastic net, where Λ is equal to the normalized Laplacian matrix of the feature network graph. Mollaysa, Strasser and Kalousis (2017) use the feature information matrix $\mathbf{Z}$ to compute a feature similarity matrix, which in turn is used to construct a penalty term in the loss criterion. Note that their 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 prior works can be generalized to generic 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. While van de Wiel et al. (2016) claim to be able to handle binary, nominal, ordinal, and continuous feature information, their method actually ranks and groups features based on such information, and only uses this grouping information. Nevertheless, the method is able to incorporate more than one source of feature information.

3. Feature-weighted Elastic Net ("Fwelnet")

One way to use the scores $\mathbf{z}_j^T \theta$ in model fitting is to give each feature a different penalty weight in the elastic net objective, based on its score:

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

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

$$w_j(\theta) = \frac{\sum_{\ell=1}^{p} \exp\left(\mathbf{z}_\ell^T \theta\right)}{p \exp(\mathbf{z}_j^T \theta)}. \tag{3.1}$$

The fwelnet algorithm minimizes this objective function over β_0 and β:

$$(\hat{\beta}_0, \hat{\beta}) = \underset{\beta_0, \beta}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{y} - \beta_0 \mathbf{1} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^{p} w_j(\theta) \left[\alpha|\beta_j| + \frac{1-\alpha}{2}\beta_j^2\right]. \tag{3.2}$$

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.

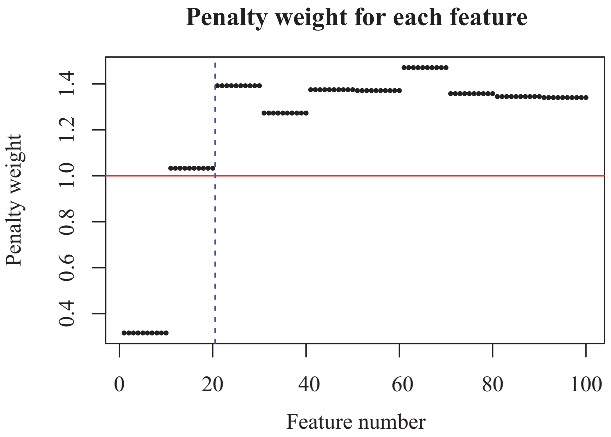

Figure 1. Penalty factors that fwelnet assigns to each feature. $n = 200$, $p = 100$ with features in groups of size 10. The response is a noisy linear combination of the first two groups, with the signal in the first group being stronger than that in the second. As expected, fwelnet's penalty weights for the true features (left of blue dotted line) are lower than those for the null features. The elastic net assigns all features a penalty factor of one (horizontal red line).

Second, $w_j(\theta) \geq 1/p$, for all j and θ, ensuring that no features have a negligible penalty. This allows the fwelnet solution to have a wider range of sparsity across λ hyperparameter values. Third, this formulation provides theoretical connections, which we detail in Section 4. Finally, a feature's score has a natural interpretation: if $\mathbf{z}_j^T\theta$ is relatively large, then w_j is relatively small, meaning that feature j is more important to the response, and hence should have a smaller 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 first group exhibits a stronger correlation to the response than that of the second group. The "features of features" matrix $\mathbf{Z} \in \mathbb{R}^{100\times 10}$ is grouping information; that is, $z_{jk} = 1\{\text{feature } j \text{ belongs to group } k\}$. Figure 1 shows the penalty factors w_j that fwelnet assigns the features. (The hyperparameter θ is determined using Algorithm 1, described in Section 3.1.) As expected, the features in the first group have the smallest penalty factor, followed by the features in the second group. In contrast, the original elastic net algorithm assigns penalty factors $w_j = 1$, for all j.

3.1. Computing the fwelnet solution

It can be easily shown that $\hat{\beta}_0 = \overline{\mathbf{y}} - \sum_{j=1}^p \hat{\beta}_j \overline{\mathbf{x}}_{\cdot j}$. Henceforth, we assume that $\mathbf{y}$ and the columns of $\mathbf{X}$ are centered such that $\hat{\beta}_0 = 0$; thus, 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 (3.2): the objective function is convex in β, and $\hat{\beta}$ can be found efficiently using algorithms such as the 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 that 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 solving the following minimization problem:

$$\begin{aligned}
&\underset{\beta(\lambda_i),\theta(\lambda_i)}{\text{minimize}} \ \frac{1}{m}\sum_{i=1}^m \Bigg[\frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta(\lambda_i)\|_2^2 \\
&+ \lambda_i \sum_{j=1}^p w_j(\theta(\lambda_i)) \left(\alpha|\beta_j(\lambda_i)| + \frac{1-\alpha}{2}\beta_j(\lambda_i)^2\right)\Bigg] \\
&\text{subject to} \ \ \theta(\lambda_1) = \cdots = \theta(\lambda_m),
\end{aligned}$$

where $\lambda_1 > \lambda_2 > \cdots > \lambda_m$ is a path of λ hyperparameter values. Here, we think of θ as an argument of the objective function J. Furthermore, we view minimizing J as a joint function of β and θ as a heuristic to obtain a good value of θ. However, to maintain the interpretation of θ as a hyperparameter, *we force $\hat{\theta}$ to be the same across all λ-values.* We propose an alternating minimization (Algorithm 1) to solve this minimization problem. Step 3(c) finds the optimum solution for $\beta(\lambda_1), \ldots, \beta(\lambda_m)$ for given values of $\theta(\lambda_1), \ldots, \theta(\lambda_m)$; Steps 3(a) and 3(b) perform a gradient descent for $\theta(\lambda_1), \ldots, \theta(\lambda_m)$ projected to the constraint set.

Because of the backtracking line search in Step 3(b) and the fact that Step 3(c) solves a convex problem, Algorithm 1 is guaranteed to converge, albeit to a stationary point. However, because Step 2 initializes $\hat{\beta}(\lambda_i)$ at the elastic net coefficients, we usually end up with a good solution. In our simulations, convergence was almost always reached within 20 iterations, and often one to three passes gave a sufficiently good solution.

Algorithm 1 Fwelnet algorithm.

```
1. Select a value of $\alpha \in [0,1]$ and a sequence of $\lambda$-values $\lambda_1 > \cdots > \lambda_m$.
2. For $i = 1, \ldots, m$, initialize $\beta^{(0)}(\lambda_i)$ at the elastic net solution for the corresponding $\lambda_i$. Initialize $\theta^{(0)} = \mathbf{0}$.
3. For $k = 0, 1, \ldots$ until convergence:
   (a) Set $\Delta\theta$ to be the component-wise mean of $(\partial J_{\lambda_i,\alpha}/\partial\theta)|_{\beta=\beta^{(k)},\theta=\theta^{(k)}}$ over $i = 1, \ldots, m$.
   (b) Set $\theta^{(k+1)} = \theta^{(k)} - \eta\Delta\theta$, where $\eta$ is the step size computed using a backtracking line search to ensure that the mean of $J_{\lambda_i,\alpha}\left(\beta^{(k)}, \theta^{(k+1)}\right)$ over $i = 1, \ldots, m$ is less than that of $J_{\lambda_i,\alpha}\left(\beta^{(k)}, \theta^{(k)}\right)$.
   (c) For $i = 1, \ldots, m$, set $\beta^{(k+1)}(\lambda_i)$ = elastic net solution for $\lambda_i$, where the penalty factor for feature $j$ is $w_j(\theta^{(k+1)})$.
```

Remark 1. We also considered an approach where θ was not constrained to be the same 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 is given in the Supplementary Material S1.

We have developed an R package, `fwelnet`, that implements Algorithm 1. Step 3(c) of Algorithm 1 can be performed easily using the `glmnet` function in the `glmnet` R package and specifying the `penalty.factor` option. In practice, we use the sequence $\lambda_1 > \cdots > \lambda_m$ provided by `glmnet`'s implementation of the elastic net, because this range of λ-values covers a sufficiently wide range of models. (In our package, we allow the user to replace the component-wise mean with the component-wise median in Step 3(a), and to replace the mean with the median in Step 3(b). We find that these options do not change the performance much when the default sequence is used, so we recommend using the defaults.)

3.2. Extending fwelnet to generalized linear models

It is easy to extend the elastic net to generalized linear models (GLMs) by replacing the RSS term with the negative log-likelihood of the data:

$$(\hat{\beta}_0, \hat{\beta}) = \underset{\beta_0,\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \ell\left(y_i, \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j\right) + \lambda \sum_{j=1}^{p} \left[\alpha|\beta_j| + \frac{1-\alpha}{2}\beta_j^2\right], \quad (3.3)$$

where $\ell(y_i, \beta_0 + \sum_j x_{ij}\beta_j)$ is the negative log-likelihood contribution of observation

i. Fwelnet can be extended to GLMs in a similar fashion:

$$(\hat{\beta}_0, \hat{\beta}, \hat{\theta}) = \underset{\beta_0, \beta, \theta}{\mathrm{argmin}} \sum_{i=1}^{n} \ell \left(y_i, \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j \right) + \lambda \sum_{j=1}^{p} w_j(\theta) \left[\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right], \tag{3.4}$$

with $w_j(\theta)$ defined in (3.1). Algorithm 1 can be used as-is to solve (3.4). Because θ only appears in the penalty term, this extension can be implemented easily. We can rely on `glmnet` for Steps 2 and 3(c), Step 3(a) is the same as before, and Step 3(b) simply requires a function that allows us to compute ℓ.

4. Theoretical Connections

4.1. 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 that belongs to group k. Assume too that $\mathbf{y}$ and the columns of $\mathbf{X}$ are centered, such that $\hat{\beta}_0 = 0$. In this setting, Yuan and Lin (2006) introduced the group lasso estimate as the solution to the optimization problem

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

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 varying group sizes:

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

Theorem 1 establishes the connection between fwelnet and the group lasso.

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* (3.2) *jointly over* β_0, β, *and* θ *reduces to*

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

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

for some $\lambda' \geq 0$*.*

The $\alpha = 0$ case minimizes the RSS and the group lasso penalty, and the $\alpha = 1$ case minimizes the RSS and the ℓ_1 version of the group lasso penalty. The proof of Theorem 1 can be found in the Supplementary Material S2.

4.2. Connection to Bayesian estimation

Regularized estimators can often be thought of as the Bayes posterior mode for a given prior distribution. For example, it is well known that if the prior and likelihood are given by

$$\beta \overset{i.i.d.}{\sim} \mathcal{N}(0, \tau^2\mathbf{I}), \text{ and } \mathbf{y} \mid \mathbf{X}, \beta \sim \mathcal{N}\left(\mathbf{X}\beta, \sigma^2\mathbf{I}\right),$$

respectively, for some $\tau^2, \sigma^2 > 0$, then the posterior distribution for β is minimized at the ridge regression solution for $\lambda = \sigma^2/(2\tau^2)$. If feature information is available, a better prior might be one where the β_j are exchangeable, conditional on the $\mathbf{z}_j$, that is, $\beta_j \overset{i.i.d.}{\sim} G(\cdot \mid \mathbf{z}_j)$, for some prior distribution G. One possible choice is

$$\beta_j \overset{ind.}{\sim} \mathcal{N}\left(0, v_j^2\tau^2\right), \quad v_j^2 = \frac{p\exp\left(\mathbf{z}_j^T\theta\right)}{\sum_{\ell=1}^{p}\exp\left(\mathbf{z}_\ell^T\theta\right)}, \tag{4.1}$$

for some fixed θ. With this prior, τ^2 is the average prior variance for β_j, and v_j^2 modulates the prior variance for each coefficient based on its feature information. The expression for v_j^2 is simply softmax applied to $\mathbf{z}_j^T\theta$ (scaled by p), a function commonly used to convert a vector of real values to a probability vector. Features with larger scores $\mathbf{z}_j^T\theta$ have correspondingly larger v_j^2, meaning they are more likely to have larger coefficients in the model. Straightforward computation shows that the posterior mode for β is the fwelnet solution (3.2) with $\alpha = 0$ and $\lambda = \sigma^2/(2\tau^2)$. Algorithm 1 can be viewed as an empirical Bayes approximation to estimate θ. For other values of α, the fwelnet solution corresponds to the posterior mode for the following prior on β:

$$p(\beta_j) \propto \exp\left[-v_j^2\tau^2\left(\alpha|\beta_j| + \frac{1-\alpha}{2}\beta_j^2\right)\right], \quad v_j^2 = \frac{p\exp\left(\mathbf{z}_j^T\theta\right)}{\sum_{\ell=1}^{p}\exp\left(\mathbf{z}_\ell^T\theta\right)}.$$

This connection also presents a way to incorporate feature information in a fully Bayesian framework: instead of estimating θ from the data, we can impose a prior on it. This direction also gives us an explicit way to encode beliefs about the relative importance of the sources of side information for the predictive model.

5. 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 CV to select the tuning parameter λ. Unless otherwise stated, the α hyperparameter was set to one (i.e., no ℓ_2-squared penalty). To compare the methods, we considered the mean squared error (MSE) $MSE = \mathbb{E}[(\hat{y}-\mu)^2]$ achieved on 10,000 test points, as well as the true positive rate (TPR) and false positive rate (FPR) of the fitted models. (Note that $\hat{y}$ denotes the model's prediction, whereas μ denotes the true underlying signal. The oracle model, which knows the true coefficient vector β, can compute μ exactly, and hence has a test MSE of zero.) We ran each simulation 30 times to obtain estimates for these quantities. (See the Supplementary Material S3 for details of the simulations.)

5.1. Setting 1: Noisy version of the true $|\boldsymbol{\beta}|$

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 $\mathbf{Z}$ has a single column: a noisy version of $|\beta|$.

We compared fwelnet against the lasso (using the `glmnet` package) and the adaptive lasso (using the OLS solution as the pilot estimator) across a range of signal-to-noise ratios (SNRs) in both the response $\mathbf{y}$ and the feature information matrix $\mathbf{Z}$ (see the Supplementary Material S3.1). The results are shown in Figure 2. As expected, the test MSE figures for the methods decreased as the SNR in the response increased. Fwelnet performed best, with its improvement over the other methods increasing as the SNR in $\mathbf{Z}$ increased, up to a point. In terms of feature selection, fwelnet appeared to have a similar TPR, but a smaller FPR.

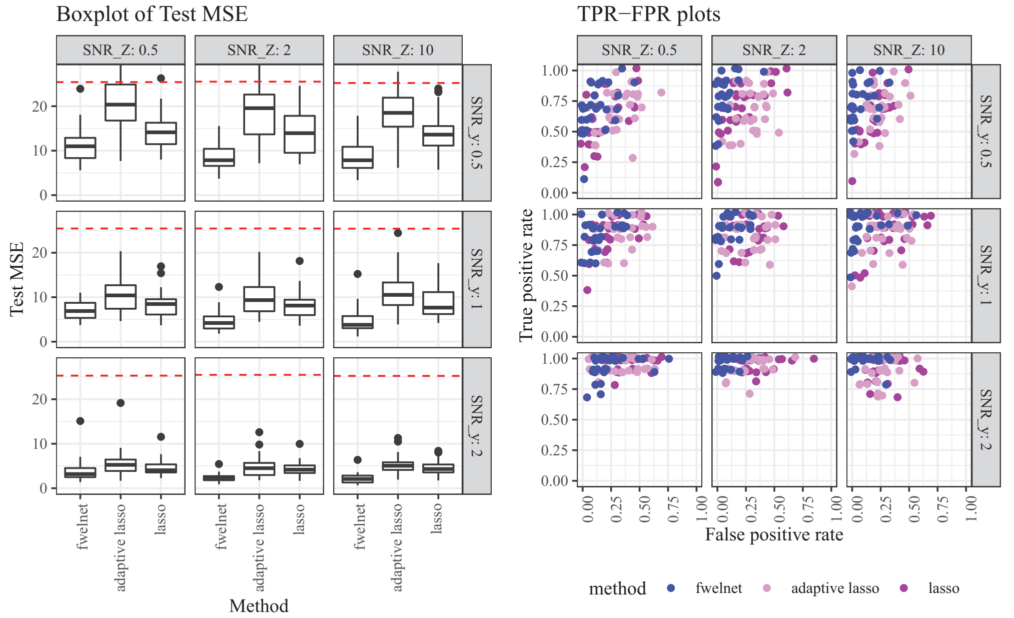

Figure 2. "Feature of features": noisy version of the true $|\beta|$. $n = 100$, $p = 50$. The response is a linear combination of the first 10 features. The SNR for $\mathbf{y}$ increases from left to right; the SNR in $\mathbf{Z}$ increases from top to bottom. The left panel shows the test MSE figures, with the red dotted line indicating the median null test MSE. In the figure on the right, each point depicts the TPR and FPR of the fitted model for one of 30 simulation runs. Fwelnet performs best in terms of the test MSE, with the improvement increasing as the SNR in $\mathbf{Z}$ increases, up to a point. Fwelnet appears to have a similar TPR, but a significantly smaller FPR.

5.2. 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, adaptive lasso, and group lasso (using the `grpreg` package) across a range of SNRs in the response $\mathbf{y}$. (For the adaptive lasso, we used the lasso solution, with λ chosen using CV, as the pilot estimator, because the OLS solution is unidentified in this setting.)

We considered two different responses in this setting. The first response 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 the test MSE, fwelnet was competitive with the group lasso. In terms of feature selection, fwelnet had a comparable TPR to that of the group lasso (except in the lowest SNR setting),

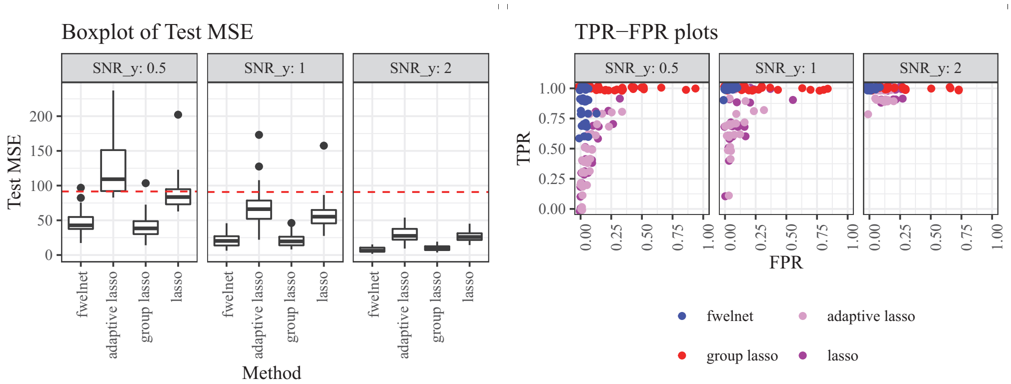

Figure 3. "Feature of features": grouping data. $n = 100$, $p = 150$. The features come in groups of 10, with the response being a linear combination of the features in the first group. The SNR for $\mathbf{y}$ increases from left to right. The figure on the left shows the test MSE results, with the red dotted line indicating the median null test MSE. In the figure on the right, each point depicts the TPR and FPR of the fitted model for one of 30 simulation runs. Fwelnet performs comparably with the group lasso in terms of the test MSE. Fwelnet has a higher TPR than the lasso and a lower FPR than the group lasso.

but a drastically smaller FPR. Fwelnet had a better TPR and FPR than the lasso in this case.

The second response was not as sparse in the features: the true signal was a linear combination of the first four feature groups. The results are shown in Figure 4. In this case, fwelnet with α fixed at one lags the group lasso slightly in terms of the test MSE. Note that fwelnet with $\alpha = 1$ performs appreciably better than the lasso when the SNR is higher. Selecting α using CV improved the test MSE performance of fwelnet slightly, but not enough to outperform the group lasso; it also came at the cost of a very high FPR.

5.3. Setting 3: Noise variables

In this setting, we have $n = 100$ observations and $p = 80$ 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. Because fwelnet is adapting to these features, we expect it to perform worse than comparable methods.

We compare fwelnet against the lasso and the adaptive lasso (using the OLS solution as the pilot estimator): the results are depicted in Figure 5. As expected, fwelnet has a higher test MSE than that of the lasso, but the decrease in performance is not drastic. The adaptive lasso performs much more poorly than

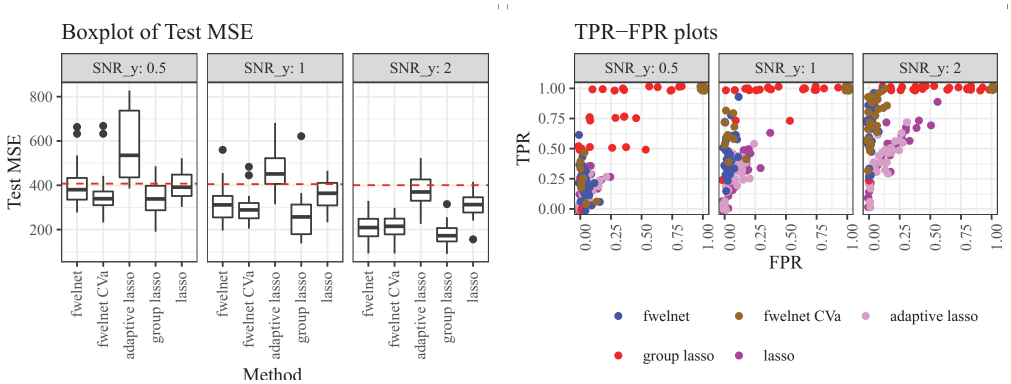

Figure 4. "Feature of features": grouping data. $n = 100$, $p = 150$. The features come in groups of 10, with the response being a linear combination of the first four groups. The SNR for $\mathbf{y}$ increases from left to right. The left figure shows the test MSE results, with the red dotted line indicating the median null test MSE. Fwelnet sets $\alpha = 1$, while fwelnet CVa selects α using CV. In the figure on the right, each point depicts the TPR and FPR of the fitted model for one of 30 simulation runs. The group lasso performs best here. CV for α improves the test MSE performance slightly, but at the expense of a very high FPR.

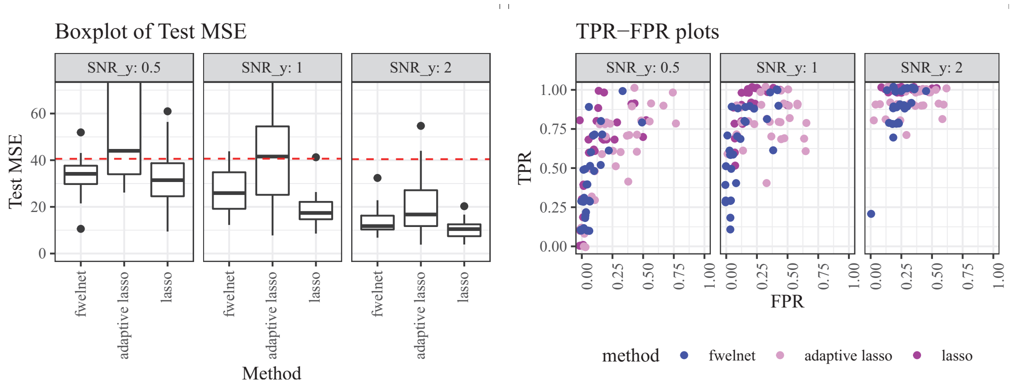

Figure 5. "Feature of features": 10 noise variables. $n = 100$, $p = 80$. The response is a linear combination of the first 10 features. The SNR for $\mathbf{y}$ increases from left to right. The left figure shows the test MSE results, with the red dotted line indicating the median null test MSE. In the right figure, each point depicts the TPR and FPR of the fitted model for one of 30 simulation runs. Fwelnet performs only slightly worse than the lasso in terms of the test MSE, and has a similar TPR and FPR to those of the lasso.

the other methods. This is likely due to unstable least squares estimates for the weights owing to p being close to n. Fwelnet attained a similar FPR and TPR to those of the lasso.

6. 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 late-onset preeclampsia has substantially higher clinical value, because it enables interventions for improved maternal and neonatal outcomes (Jabeen et al. (2011)). In this example, we leverage protein data collected in late pregnancy, which is closer to the onset of preeclampsia, but of lower clinical utility, to learn about the proteins most helpful for this prediction task. Then, we use this information to build a model using protein measurements from early in the pregnancy. Note that the data from late pregnancy is only used to train the model: for prediction on new patients, we need only the samples collected during early pregnancy.

We used a data set of 1,125 plasma proteins, measured during various gestational ages of pregnancy (Erez et al. (2017)). The SOMAScan platform used in this data set produces targeted measurements of a broad range of proteins that are broadly related to various aspects of human biology. To maintain the exploratory nature of the study, we did not select specific proteins that are expected to be related to preeclampsia, based on prior studies. We considered time points ≤ 20 weeks as "early," and time points > 20 weeks as "late." The data set consists of 166 patients, each with two to six time points, for a total of 666 time-point observations. Protein measurements were log-transformed to reduce skewness. We used the following procedure to build a predictive model, based on early time-point data only:

1. Patients were split randomly into two equal-sized buckets. For patients in the first bucket, we used only their late time points (83 patients with 219 time points). For patients in the second bucket, we used only their early time points (83 patients with 116 time points).

2. We trained an elastic net logistic regression model on the late time points for patients in the first bucket to predict whether the patient would have preeclampsia (using the log-transformed protein measurements as predictors). Here, α was set to 0.5 and λ was selected using CV. We extracted the model coefficients at the λ-value that gave the highest CV area under the curve (AUC).

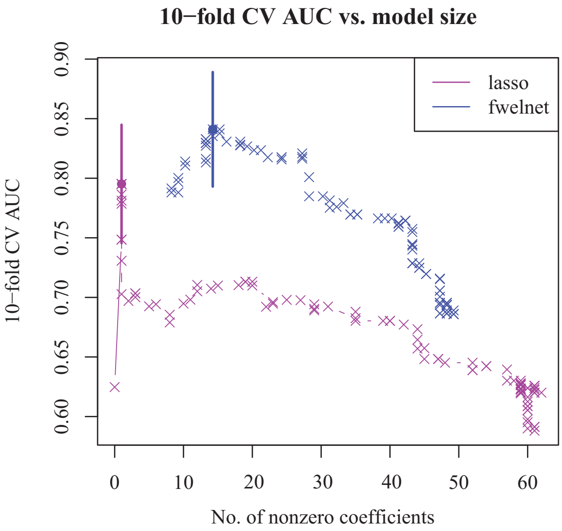

Figure 6. Early prediction of preeclampsia: Plot of the 10-fold CV AUC, plotted against the number of nonzero coefficients for each model, trained on early time-point data only. For each method, the model with the highest CV AUC is marked by a dot. To reduce clutter in the figure, the ±1 standard error bars are drawn for just these models. Fwelnet achieved a higher CV AUC for the same model size, that is, the number of features with nonzero coefficients.

3. We trained a fwelnet logistic regression model on the early time points for patients in the second bucket, using the absolute values of the late time-point model coefficients as feature information. Here, α was set to one, and we computed the 10-fold CV AUC for the entire path of λ-values.

When performing CV in Steps 2 and 3, 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.

Figure 6 shows a plot of the 10-fold CV AUC for the fwelnet model in Step 3 and the baseline lasso model against the number of features in the model. The lasso obtains a maximum CV AUC of 0.80, and fwelnet obtains the largest CV AUC of 0.84.

In running the workflow several times, we noted that the results were somewhat dependent on (i) how the patients were split into the two buckets in Step 1, and (ii) how patients were split into CV folds when training the models in Steps 2 and 3. We found that if the late-time point model had few nonzero coefficients, then the fwelnet model for the early time-point data was very similar to the lasso.

This matches our intuition: few nonzero coefficients means injecting very little additional information through fwelnet's relative penalty factors. Nevertheless, we did not encounter cases in which running fwelnet resulted in a worse CV AUC than that of the lasso.

7. Using Fwelnet for Multi-task Learning

We now apply fwelnet to *multi-task learning*. Here, 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 responses, it can be advantageous to fit models for them simultaneously. This is especially useful if the responses have a low SNR.

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 values of the coefficients of one response as the external information for the other response. That way, a feature that has a 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.

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 $\lambda$ that minimizes cross-validated error.
2. For $k = 0, 1, \ldots$ until convergence:
    (a) Set $\mathbf{Z}_2 = |\beta_1^{(k)}|$. 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 $\mathbf{Z}_1 = |\beta_2^{(k+1)}|$. Run fwelnet with $(\mathbf{X}, \mathbf{y}_1, \mathbf{Z}_1)$ and set $\beta_1^{(k+1)}$ to be the lambda.min solution.
```

We tested the effectiveness of Algorithm 2 (with step 2 run for three iterations) on simulated data. We generated 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 one to five) have larger absolute values than those for the features specific to one response. The SNRs in response 1 and response 2 are 0.5 and 1.5, respectively. (See the Supplementary Material S4 for more details of the simulation.)

We compared Algorithm 2 against the following: (i) the *individual lasso*

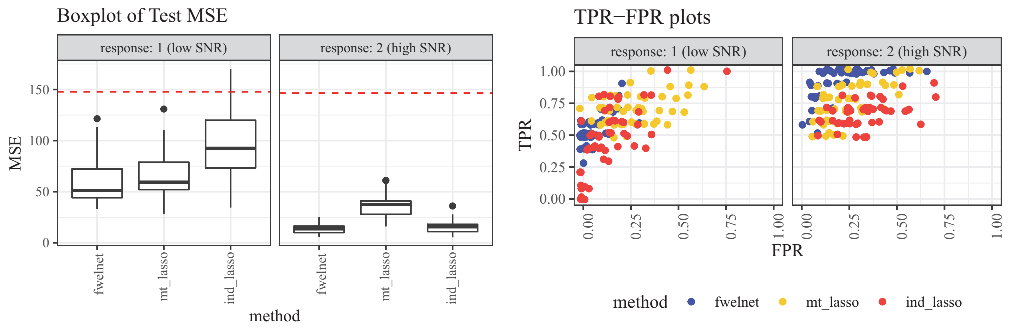

Figure 7. Application of fwelnet to multi-task learning. $n = 150$, $p = 50$. Response 1 is a linear combination of features 1 to 10, while response 2 is a linear combination of features 1 to 5 and 11 to 15. The SNRs for the responses are 0.5 and 1.5, respectively. The left figure shows the test MSE figures, with the red dotted line indicating the median null test MSE. The right figure shows the TPR and FPR of the fitted model (each point being one of 50 simulation runs). Fwelnet outperforms the individual lasso and the multi-response lasso in terms of the test MSE for both responses. Fwelnet also appears to have a better FPR than the other methods and a better TPR than 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)* (Obozinski, Taskar and Jordan (2010)), where the 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.

Figure 7 shows the results for 50 simulation runs. Fwelnet outperforms the other two methods in terms of the test MSE, as evaluated on 10,000 test points. The individual lasso performs well for the higher SNR response, but poorly for the lower SNR response. 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 have 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.

8. Discussion

In this paper, we have proposed a method for exploiting external information about predictor variables. We do this by organizing these “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, “fwelnet,” which imposes a penalty modification factor $w_j(\theta) =$

$\sum_{\ell=1}^{p} \exp(\mathbf{z}_\ell^T \theta)/p \exp(\mathbf{z}_j^T \theta)$ for the elastic net algorithm.

This method is widely applicable in that there are no restrictions on the type of feature information that can be incorporated into $\mathbf{Z}$, as long as it is real-valued. As such, we recommend using fwelnet whenever feature information is available (e.g., grouping information, prior guesses on feature importance). When the feature information is relevant to the prediction problem, in that it has some signal on how important a feature is to predicting the response, we expect fwelnet to outperform competing methods. At the same time, simulation setting 3 (Section 5.3) shows that using irrelevant feature information can be detrimental to the fit. In practice, we recommend using domain knowledge to guide the selection of side information for the model. We also recommend fitting the vanilla elastic net and comparing the CV error of the two methods: this comparison will show whether the feature information was relevant to the prediction problem.

There is much scope for future work:

- *Interpretation of* $\mathbf{z}_j^T\theta$ *and* θ. As noted in the Introduction, $\mathbf{z}_j^T\theta$ can be viewed as an indication of how influential feature j is on the response, because a larger $\mathbf{z}_j^T\theta$ corresponds to a smaller penalty weight $w_j(\theta)$ (see Equations (3.1) and (3.2)).

  The interpretation for θ is not as straightforward. When $\mathbf{Z} \in \mathbb{R}^{p\times K}$ is orthonormal, we can interpret θ_k as the relative importance of the kth source of feature information for identifying important features for the prediction problem. However, this interpretation becomes less clear when there are correlations between the columns of $\mathbf{Z}$. In the extreme case, where there is multicollinearity in $\mathbf{Z}$, θ is not identified, even though $\mathbf{z}_j^T\theta$ is unique. These are the same issues one faces when interpreting OLS coefficients in the presence of feature correlations.

- *Different choices of side information* $\mathbf{Z}$. We have explored a few different choices of side information, including prior coefficient estimates and group membership. It would be interesting to evaluate fwelnet's effectiveness when using other types of side information. One natural extension of group membership is probabilistic group membership, where each feature is assigned a probability distribution across the K groups. Another extension is overlapping groups, where each row of $\mathbf{Z}$ need not sum to one. Then, $\mathbf{Z}$ as a $p \times p$ similarity matrix is another option, which can be thought of as a combination of the two extensions above, with group j being associated with feature j, and the degree of group membership measured by how similar each feature is to feature j.

- *Whether θ should be treated as a parameter or a hyperparameter, and how to determine its value.* We introduced θ as a hyperparameter for (3.2). This gives us the clear interpretation for θ described above. However, the grid search computation to find its optimal value grows exponentially with the number of sources of feature information. To avoid this growth, we suggested a descent algorithm for θ based on its gradient with respect to the fwelnet objective function. Other methods for hyperparameter optimization can be applied, including the random search (e.g., Bergstra and Bengio (2012)) and Bayesian optimization (e.g., Snoek, Larochelle and Adams (2012)).

  One could consider θ as an argument of the fwelnet objective function to be minimized over jointly with β. This approach gives us a theoretical connection to the group lasso (Section 4.1). However, we 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, so different minimization algorithms could end up at different local minima. Our attempts to make this approach work (see the Supplementary Material S1) did not fare as well in terms of prediction performance and was computationally expensive.

- *Choice of penalty modification factor.* While the penalty modification factor $w_j(\theta)$ we have proposed works well in practice and has several desirable properties, we make no claim about its optimality.

- *Extending the use of scores beyond the elastic net.* The use of feature scores $\mathbf{z}_j^T\theta$ in modifying feature weights 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.

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

Supplementary Material

The online Supplementary Material provides the following: (i) details on an alternative algorithm with θ as a parameter; (ii) details on the simulation study in Section 5; (iii) a proof for Theorem 1; and (iv) details on the simulation study in Section 7.

Acknowledgments

We would like to thank Nikolaos Ignatiadis, the associate editor, and the reviewers for their helpful discussions and comments. Nima Aghaeepour was supported by the Bill & Melinda Gates Foundation (OPP1189911, OPP1113682), the National Institutes of Health (R01AG058417, R01HL13984401, R61NS114926, R35GM138353-01), the Burroughs Wellcome Fund, and the March of Dimes Prematurity Research Center at Stanford University. Trevor Hastie was partially supported by the National Science Foundation (DMS-1407548 and IIS1837931) and the National Institutes of Health (5R01 EB001988-21). Robert Tibshirani was supported by the National Institutes of Health (5R01 EB001988-16) and the National Science Foundation (19 DMS1208164).

References

Bergersen, L. C., Glad, I. K. and Lyng, H. (2011). Weighted Lasso with data integration. *Statistical Applications in Genetics and Molecular Biology* **10**, Article 39.

Bergstra, J. and 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. and Fuchs, M. (2017). IPF-LASSO: Integrative L_1-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. et al.(2017). The prediction of late-onset preeclampsia: Results from a longitudinal proteomics study. *PLoS ONE* **12**, e0181468.

Friedman, J., Hastie, T. and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**, 1–22.

Hoerl, A. E. and Kennard, R. W. (1970).Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **42**, 80–86.

Jabeen, M., Yakoob, M. Y., Imdad, A. and Bhutta, Z. A. (2011). Impact of interventions to prevent and manage preeclampsia and eclampsia on stillbirths. *BMC Public Health* **11**, S6.

Jacob, L., Obozinski, G. and Vert, J.-P. (2009). Group Lasso with overlap and graph Lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 433–440. Association for Computing Machinery, New York.

Li, C. and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics* **24**, 1175–1182.

Mollaysa, A., Strasser, P. and Kalousis, A. (2017). Regularising non-linear models using feature side-information. In *Proceedings of the 34th International Conference on Machine Learning*, 2508–2517. The MIT Press, Cambridge.

Obozinski, G., Taskar, B. and Jordan, M. I. (2010). Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing* **20**, 231–252.

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

Snoek, J., Larochelle, H. and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, 2951–2959. Curran Associates Inc., New York.

Tai, F. and Pan, W. (2007). Incorporating prior knowledge of predictors into penalized classifiers with multiple penalty terms. *Bioinformatics* **23**, 1775–1782.

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

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

van de Wiel, M. A., Lien, T. G., Verlaat, W., van Wieringen, W. N. and Wilting, S. M. (2016). Better prediction by use of co-data: Adaptive group-regularized ridge regression. *Statistics in Medicine* **35**, 368–381.

Velten, B. and Huber, W. (2021). Adaptive penalization in high-dimensional regression and classification with external covariates using variational Bayes. *Biostatistics* **22**, 348–364.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **68**, 49–67.

Zeisler, H., Llurba, E., Chantraine, F., Vatish, M., Staff, A. C., Sennström, M. et al. (2016). Predictive value of the sFlt-1:PlGF ratio in women with suspected preeclampsia. *New England Journal of Medicine* **374**, 13–22.

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

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**, 301–320.

J. Kenneth Tay

Department of Statistics, Stanford University, Stanford, CA 94305, USA.

E-mail: kjytay@stanford.edu

Nima Aghaeepour

Department of Anesthesiology, Pain, and Perioperative Medicine, and Department of Pediatrics, and Department of Biomedical Data Sciences, Stanford University, Stanford, CA 94305, USA.

E-mail: naghaeep@stanford.edu

Trevor Hastie

Department of Statistics, Stanford University, and Department of Biomedical Data Sciences, Stanford University, Stanford, CA 94305, USA.

E-mail: hastie@stanford.edu

Robert Tisbshirani

Department of Statistics, Stanford University, and Department of Biomedical Data Sciences, Stanford University, Stanford, CA 94305, USA.

E-mail: tibs@stanford.edu

(Received June 2020; accepted May 2021)