# A Comprehensive Study on Group Lasso Performance

James Yang

March 7, 2023

## 1 Group Lasso Problem

The lasso problem solves the following optimization problem:

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \ f(\beta) + \lambda \|\beta\|_1$$

where $f : \mathbb{R}^p \to \mathbb{R}$ is a convex function and $\lambda > 0$. Given data $X \in \mathbb{R}^{n \times p}$ and $y \in \mathbb{R}^n$, the most popular choices for $f$ are the Generalized Linear Model (GLM) negative log-likelihoods under the Gaussian and Binomial family:

$$\text{Gaussian:} \quad f(\beta) := \frac{1}{2}\|y - X\beta\|_2^2 \tag{1}$$

$$\text{Binomial:} \quad f(\beta) := \sum_{i=1}^{n} \left( y_i x_i^\top \beta - \log\left(1 + e^{x_i^\top \beta}\right) \right) \tag{2}$$

The celebrated lasso objective gained popularity for its interpretability in regression settings as in the above since it tends to shrink some coefficients to zero exactly.

The group lasso problem [Yuan and Lin, 2006] is an extension of the lasso problem that attempts to capture group structure within the features. Specifically, the group lasso attempts to zero-out *groups of coefficients* rather than a single coefficient. The interpretation is that if a group of coefficients is zero, the whole group is uninformative in predicting $y$ [Meier et al., 2008]. This overcomes a major fallback of lasso with categorical features since lasso selects individual categorical feature (factor) instead of a whole group of features (e.g. one-hot encoded factors).

> James: Cite early literature of group lasso.

In preparation of posing the group lasso problem, we begin with some notation. Let $1 \leq G \leq p$ be the number of groups among the $p$ features. Let $p_i$ be the group size of group $i$ for each $1 \leq i \leq G$ such that $\sum_{i=1}^{G} p_i = p$. Define $\beta = [\beta_1 \, \beta_2 \, \ldots \, \beta_G]^\top \in \mathbb{R}^p$ where each $\beta_i \in \mathbb{R}^{p_i}$. The group lasso problem is to minimize the following objective:

> James: Cite GWAS and LD stuff that it makes a lot of sense to group genes.

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \ f(\beta) + \lambda \sum_{j=1}^{G} \|\beta_j\|_2 \tag{3}$$

for some convex function $f : \mathbb{R}^p \to \mathbb{R}$ and $\lambda > 0$. Similar to the lasso, given data $X$ and $y$, we may use the same Gaussian and Binomial log-likelihoods (1), (2), respectively, for $f$.

## 2 Optimizer Algorithm

In this section, we discuss the algorithm to solve the optimization problem (3). Let $X \in \mathbb{R}^{n \times p}$, $y \in \mathbb{R}^n$ be the feature matrix and response vector, respectively. Let $G$ be the number of groups and $p_i$ be the size of group $i$ for $1 \leq i \leq G$. Further let $X = [X_1 \, X_2 \, \ldots \, X_G]$ where $X_i \in \mathbb{R}^{n \times p_i}$ is the feature matrix corresponding to group $i$ features only.

In general, if the loss function $f$ is of the form $f(\beta) \equiv f(y, X\beta)$, the optimization problem (3) can be reparametrized in terms of $\tilde{\beta} \in \mathbb{R}^p$ where $\tilde{\beta}_i := Q_i^\top \beta_i$ where $Q_i \in \mathbb{R}^{p_i \times p_i}$ is the orthogonal matrix from the eigen-decomposition of $X_i^\top X_i = Q_i D_i Q_i^\top$ for each $i = 1, \ldots, G$. Concretely, it suffices to solve

$$\underset{\tilde{\beta} \in \mathbb{R}^p}{\text{minimize}} \ f(y, \tilde{X}\tilde{\beta}) + \lambda \sum_{j=1}^{G} \|\tilde{\beta}_j\|_2 \tag{4}$$

where $\tilde{X} = [X_1 Q_1 \, X_2 Q_2 \, \ldots \, X_G Q_G]$. Note that this was only possible because the group lasso penalty is invariant under the transformation from $\beta \to \tilde{\beta}$. Then, given a solution $\tilde{\beta}^*$ to the new problem (4), we have the solution to the original problem $\beta^*$ given by $\beta_i^* := Q_i^\top \tilde{\beta}_i^*$. Hence, without loss of generality, we assume that our feature matrix $X$ is of the form $\tilde{X}$. In particular, we assume that $X_i^\top X_i = D_i$ is diagonal for all $1 \leq i \leq G$. We drop all tilde for notational simplicity.

Computationally, it may seem like we must construct $\tilde{X}$ from $X$ before solving (4). However, we will shortly see that the fitting algorithm can be amended to dynamically compute the necessary components of $\tilde{X}$. In particular, we only need access to $\tilde{X}_i$, $D_i$, and $\tilde{X}_i^\top \tilde{X}_j$ for $i$ in the strong set and $j$ in the active set. In comparison to other proposed fitting procedures, we believe that the extra cost to compute $\tilde{X}_i$ and $\tilde{X}_i^\top \tilde{X}_j$ is comparable to the cost of the other procedures. For example, Yuan and Lin [2006], Meier, Van De Geer, and Böhlmann [2008] rely on the fact that each group feature matrix $X_i$ is full-rank and therefore admits a QR decomposition with an invertible $R$. Under this assumption, they solve (4) with $\tilde{X} = [Q_1 Q_2 \ldots Q_G]$ and transform the solution back to the original scale by applying $R_i^{-1}$ to the $i$th group of coefficients. Note that these other approaches also apply a one-time decomposition of the group feature matrices. The benefit of this approach is that the algorithm is aided by a simple closed-form expression as part of the blockwise coordinate descent algorithm. However, there are two major downsides to this approach. The first is that, while the aforementioned authors primarily focused on the application of group lasso on categorical data such as one-hot encoded data that easily satisfy the full-rank assumption, this assumption may easily be violated in other applications. For example, . The second is that this method does not simplify computations when we add a ridge penalty as well (i.e. elastic net), since in general, it introduces a Tikhonov regularization in the transformed problem.

Similar to other approaches [Yuan and Lin, 2006, Meier et al., 2008, Tseng, 2001, Liang et al., 2022], we use blockwise coordinate descent for its excellent speed and simplicity. Section 2.1 describes this procedure in greater detail. However, we take a completely different approach to fitting each block update. The essence of implementing a highly performant optimizer lies in solving each block update (6) as fast as possible. Aside from the orthonormalization method described above as in Yuan and Lin [2006], Meier, Van De Geer, and Böhlmann [2008], many works in literature that describe group lasso fitting procedures use the general descent method called Proximal Gradient Descent (PGD), or Iterative Soft-Thresholding Algorithm (ISTA) in this context, to solve (6) [Liang et al., 2022, Beck and Teboulle, 2009, Klosa et al., 2020, Wright et al., 2009, Loris, 2009, Hastie et al., 2016, O'Donoghue and Candés, 2015]. A Nesterov acceleration can be applied on ISTA to get Fast-ISTA (FISTA) [Beck and Teboulle, 2009]. Section 2.2 describes the (F)ISTA procedure, Sections 2.3- describes our proposed method to solve (6), and Section gives a comparison of the two methodology.

> James: Reference to later section?

> James: mention GWAS; small n, bigger group size

> James: ?

> James: ?

## 2.1 Blockwise Coordinate Descent

The (convex) optimization problem (4) has a separable regularization, so blockwise coordinate descent can be applied Tseng [2001]. Hence, we iterate through each block $j = 1, \ldots, G$, keeping all other blocks fixed, and solve the following problem:

$$\underset{\beta_j \in \mathbb{R}^{p_j}}{\text{minimize}} \ f(y, X\beta) + \lambda \|\beta_j\|_2 \tag{5}$$

until convergence is reached. The final solution vector $\beta$ then solves (4).

For the remainder of the paper, we only discuss the Gaussian loss (1). For a general $f$, we may apply Iterative Reweighted Least Squares (IRLS) by iteratively reweighting the data and solving (4) with the Gaussian loss. The Gaussian loss yields the optimization problem:

> James: Cite IRLS. Yuan and Meier?

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \; \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{G} \|\beta_j\|_2$$

with the block update for group $j$ as:

$$\underset{\beta_j \in \mathbb{R}^{p_j}}{\text{minimize}} \; \frac{1}{2}\beta_j^\top D_j \beta_j - (X_j^\top y - \sum_{i \neq j} X_j^\top X_i \beta_i)^\top \beta_j + \lambda \|\beta_j\|_2$$

For notational simplicity, define $v_j := X_j^\top y - \sum_{i \neq j} X_j^\top X_i \beta_i$. Then, dropping the subscript $j$, each block update is of the form:

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \; \frac{1}{2}\beta^\top D\beta - v^\top \beta + \lambda \|\beta\|_2 \tag{6}$$

where $D \in \mathbb{R}^{p \times p}$ is diagonal with non-negative entries, $v \in \mathbb{R}^p$ is any vector, and $\lambda > 0$.

## 2.2 Proximal Gradient Descent (PGD)

In this section, we describe the PGD method, or ISTA, to solve the block update (6). The progression is similar to that of Hastie et al. [2016]. Define

$$f(\beta) = \frac{1}{2}\beta^\top D\beta - v^\top \beta$$

as the convex (Gaussian) loss function in (6). ISTA has guaranteed convergence so long as the step-size is chosen properly. Any step-size $\nu$ can be chosen such that $\nu \leq \frac{1}{L}$ where $L$ is the Lipschitz constant for $\nabla f$. Since

$$\nabla f(\beta) = D\beta - v$$
$$\implies \|\nabla f(\beta) - \nabla f(\tilde{\beta})\|_2 = \|D(\beta - \tilde{\beta})\|_2 \leq \|D\|_{\text{op}}\|\beta - \tilde{\beta}\|_2$$

we have $L \equiv \|D\|_{\text{op}} \equiv \lambda_{\max}(D)$ as the largest eigenvalue of $D$. In particular, since $D$ is diagonal, $L \equiv \max_i D_{ii}$.

The ISTA procedure is given in Algorithm 1. By applying a standard Nesterov acceleration [Beck and Teboulle, 2009], we have the FISTA procedure given in Algorithm 2. Additionally, we may perform adaptive restarts based on the proximal gradient updates [O'Donoghue and Candés, 2015] for even faster convergence as outlined in Algorithm 3.

---

**Algorithm 1:** ISTA

    **Data:** $D$, $v$, $\lambda$, $\beta^{(0)}$

1   $\nu \leftarrow \left(\max_i D_{ii}\right)^{-1}$;

2   **while** *not converged* **do**

3      $w \leftarrow \beta^{(k-1)} + \nu\left(v - D\beta^{(k-1)}\right)$;

4      $\beta^{(k)} \leftarrow \left(1 - \frac{\nu\lambda}{\|w\|_2}\right)_+ w$;

---

Note that the computational cost of Algorithms 1, 2, 3 are all $O(pk)$ where $k$ is the number of iterations until convergence. Since we must output a vector in $\mathbb{R}^p$, any optimizer must necessarily have $O(p)$ operations. Hence, the optimal optimizer of complexity $O(pk)$ must prioritize lowering the number of iterations until convergence. While each algorithm is an improvement from the previous, there is still much improvement left. Section 2.3 shows a novel approach to solving (6) with a significant decrease in iterations while maintaining $O(p)$ operations per iteration.

---
**Algorithm 2:** FISTA
---
**Data:** $D$, $v$, $\lambda$, $\beta^{(0)}$

**1** $\nu \leftarrow \left( \max_i D_{ii} \right)^{-1}$;

**2** $\eta^{(1)} \leftarrow \beta^{(0)}$;

**3** $t_1 = 1$;

**4 while** *not converged* **do**

**5** $\quad w \leftarrow \eta^{(k)} + \nu \left( v - D\eta^{(k)} \right)$;

**6** $\quad \beta^{(k)} \leftarrow \left( 1 - \frac{\nu\lambda}{\|w\|_2} \right)_+ w$;

**7** $\quad t_{k+1} \leftarrow \frac{1+\sqrt{1+4t_k^2}}{2}$;

**8** $\quad \eta^{(k+1)} \leftarrow \beta^{(k)} + \frac{t_k-1}{t_{k+1}}(\beta^{(k)} - \beta^{(k-1)})$;
---

---
**Algorithm 3:** FISTA with Adaptive Restart
---
**Data:** $D$, $v$, $\lambda$, $\beta^{(0)}$

**1 while** *not converged* **do**

**2** $\quad$ Carry out Algorithm 2 with inputs $D$, $v$, $\lambda$, $\beta^{(0)}$ until

$$\nabla f(\eta^{(k)})^\top (\beta^{(k)} - \beta^{(k-1)}) > 0 \qquad (7)$$

$\quad$ ;

**3** $\quad$ **if** *(7) holds at step k* **then**

**4** $\quad\quad$ Reset $\beta^{(0)} \leftarrow \beta^{(k)}$;
---

## 2.3 Newton's Method (NM)

In this section, we propose a novel and simple method to solve (6). By solving the sub-gradient problem of (6), we see that the solution $\beta^\star$ must satisfy

$$\beta = \begin{cases} \left( D + \frac{\lambda}{\|\beta\|_2} I \right)^{-1} v, & \|v\|_2 > \lambda \\ 0, & \|v\|_2 \leq \lambda \end{cases} \qquad (8)$$

Without loss of generality, we consider the first case when $\beta^\star \neq 0$. Without knowing $\|\beta^\star\|_2$, there is no closed-form solution for $\beta^\star$. In an endeavor to find an expression for $\|\beta^\star\|_2$, we take the $\ell_2$ norm on both sides to get that $\|\beta^\star\|_2$ must satisfy

$$\sum_{i=1}^{p} \frac{v_i^2}{(D_{ii}\|\beta\|_2 + \lambda)^2} = 1 \qquad (9)$$

Unfortunately, there is no closed-form solution for (9) either. However, we will shortly show that there is an efficient algorithm to numerically solve (9). In turn, once we find $\|\beta^\star\|_2$, we may substitute it into (8) to get the full solution $\beta^\star$ in $O(p)$ time. We note in passing that (8) and (9) were previously discovered [Hastie et al., 2016], however, there is no mention of how to quickly solve for $\beta^\star$ or $\|\beta^\star\|_2$.

We now discuss how to solve (9). First, define $\varphi : [0, \infty) \to \mathbb{R}$ by

$$\varphi(h) := \sum_{i=1}^{p} \frac{v_i^2}{(D_{ii}h + \lambda)^2} - 1$$

so that (9) is now a root-finding problem for $\varphi$. Upon differentiating $\varphi$,

$$\frac{d\varphi(h)}{dh} = -2 \sum_{i=1}^{p} \frac{v_i^2 D_{ii}}{(D_{ii}h + \lambda)^3} \leq 0$$

4

where the inequality is strict if not all $v_i^2 D_{ii} = 0$. Note that if all $v_i^2 D_{ii} = 0$, then we must have that $\|v\|_2 \leq \lambda$. Otherwise,

$$\varphi(h) = \sum_{i:D_{ii}=0} \frac{v_i^2}{\lambda^2} - 1 = \lambda^{-2} \sum_{i=1}^{p} v_i^2 - 1 > 0$$

In particular, $\varphi$ is constant and strictly positive so it has no roots. So, if $\|v\|_2 > \lambda$, then there exists a vector $\beta$ where $\varphi(\|\beta\|_2) = 0$, which is a contradiction. Hence, without loss of generality, we may assume $\varphi'$ is strictly negative, so that $\varphi$ is strictly decreasing. Consequently, since $\varphi(0) > 0$ by hypothesis, there exists a (unique) root. Further, it is easy to see that $\varphi$ is convex since it is a sum of convex functions.

Since $\varphi$ is convex, this suggests solving (9) via a one-dimensional Newton's Method. Specifically, with $h^{(0)}$ as the initial starting point, for $k \geq 1$,

$$h^{(k)} = h^{(k-1)} - \frac{\varphi(h^{(k-1)})}{\varphi'(h^{(k-1)})} \tag{10}$$

We claim that Newton's Method is guaranteed to converge for any initial point $h^{(0)}$. Indeed, for every $k \geq 1$, by convexity of $\varphi$,

$$\varphi(h^{(k)}) \geq \varphi(h^{(k-1)}) + \varphi'(h^{(k-1)})(h^{(k)} - h^{(k-1)}) = 0$$

Along with (10), this shows that $h^{(k)}$ is an increasing sequence for $k \geq 1$ and bounded above by $h^\star$, the root of $\varphi$, by monotonicity. Hence, $h^{(k)}$ converges to some limit $h^{(\infty)}$. From (10), taking limit as $k \to \infty$ and using that $\varphi'$ is non-zero,

$$h^{(\infty)} = h^{(\infty)} - \frac{\varphi(h^{(\infty)})}{\varphi'(h^{(\infty)})} \implies \varphi(h^{(\infty)}) = 0$$

which shows that $h^{(\infty)}$ is the root of $\varphi$. In summary, Newton's Method gives *guaranteed convergence* to the root.

Since $\varphi$ is convex, we also have quadratic convergence rate and each iteration costs $O(p)$. In principle, our setting is one of the best settings for Newton's Method to succeed. While FISTA (Algorithm 2) and the adaptive restarted version (Algorithm 3) also give the same convergence rate, Figure shows a clear [James: add figure ref] improvement for the Newton's Method in practice. One very important benefit of Newton's Method is that the optimization domain remains one-dimensional *regardless of the input dimensions*. However, FISTA optimizes over $\beta \in \mathbb{R}^p$ directly, so its performance is heavily impacted by the dimension.

It is widely known that descent methods including Newton's Method is highly sensitive to the initialization. The question that remains is: how to choose the initial starting point $h^{(0)}$? Since $\varphi(0) > 0$, it is sufficient to choose $h^{(0)} = 0$. However, in practice, even this can be dramatically improved. Section 2.4 describes our most performant modification to the vanilla Newton's Method.

We end this section with a remark that if $h^{(0)}$ is such that $\varphi(h^{(0)}) < 0$, then we do not have the guarantee that $\varphi(h^{(k)}) \leq 0$, i.e. it is possible that $\varphi(h^{(k-1)}) < 0$ while $\varphi(h^{(k)}) > 0$. Of course, once $\varphi(h^{(k)}) > 0$, then we have the guaranteed convergence as before. In practice, $\varphi$ tends to have a flat tail that if $h^{(0)}$ is chosen such that $\varphi(h^{(0)}) < 0$, then the next Newton step will dramatically place $h^{(1)}$ towards 0, sometimes being largely negative. In this case, $h^{(1)}$ must be projected back to $[0, \infty)$. However, we do not recommend this approach because this possible large jump will usually set the user back to the case of applying Newton's Method with $h^{(0)} = 0$.

## 2.4   Newton's Method with Adaptive Bisection (NMAB)

In this section, we improve the Newton's Method described in Section 2.3. Specifically, we first discuss how to find lower and upper bounds $h_\star, h^\star \in [0, \infty)$, respectively, such that the root lies in $[h_\star, h^\star]$. We then discuss an adaptive bisection method that preceeds the Newton's Method to find a clever initial point $h^{(0)}$.

We first begin with the derivation of $h_\star$ and $h^\star$. Note that the root-finding problem for $\varphi$ is equivalent to finding the largest value $h > 0$ such that $\varphi(h) \geq 0$, or

$$\sum_{i=1}^{p} \frac{v_i^2}{(D_{ii}h + \lambda)^2} \geq 1$$

Let $a(h), b(h) \in \mathbb{R}^p$ be defined by $a_k(h) := D_{kk}h + \lambda$ and $b_k(h) := \frac{|v_k|}{a_k(h)}$ for each $k = 1, \ldots, p$. Then, by Cauchy-Schwarz,

$$\|v\|_1 := \sum_k |v_k| = \sum_k a_k(h)b_k(h) \leq \|a(h)\|_2 \|b(h)\|_2$$

Hence, if $\|a(h)\|_2 \leq \|v\|_1$, then $\varphi(h) \geq 0$. We see that $\|a(h)\|_2 \leq \|v\|_1$ if and only if

$$\sum_{i=1}^{p} (D_{ii}h + \lambda)^2 \leq \|v\|_1^2$$

This inequality can be solved for $h$ using the quadratic formula. Let $\tilde{h}$ be the solution. Then, we have a potentially tighter lower bound $h_\star := \max(\tilde{h}, 0)$ than zero.

Next, we discuss $h^\star$. Similar to $h_\star$, we wish to find the smallest $h$ such that

$$\sum_{i=1}^{p} \frac{v_i^2}{(D_{ii}h + \lambda)^2} \leq 1$$

Approximating this problem, since

$$\sum_{i=1}^{p} \frac{v_i^2}{(D_{ii}h + \lambda)^2} = \sum_{i:D_{ii}=0} \frac{v_i^2}{\lambda^2} + \sum_{i:D_{ii}>0} \frac{v_i^2}{(D_{ii}h + \lambda)^2} \tag{11}$$

$$\leq \sum_{i:D_{ii}=0} \frac{v_i^2}{\lambda^2} + h^{-2} \sum_{i:D_{ii}>0} \frac{v_i^2}{D_{ii}^2} \tag{12}$$

by setting

$$h^\star := \sqrt{\frac{\sum_{i:D_{ii}>0} \frac{v_i^2}{D_{ii}^2}}{1 - \lambda^{-2} \sum_{i:D_{ii}=0} v_i^2}}$$

we have that $\varphi(h^\star) \leq 0$. This shows that the root must lie in $[h_\star, h^\star]$.

In practice, $\varphi$ may decay incredibly rapidly near $h \approx 0$ and have an extremely flat tail. To protect against slow convergence of Newton's Method in this case, we may use $h_\star$ and $h^\star$ to first perform a bisection method to find the initial starting point. The key idea is to use bisection first for a few iterations to avoid the region of slow changes in the Newton iterations. Then, once sufficiently close to the root, we apply the Newton's Method for the fast guaranteed convergence. Although one may use a simple bisection method of splitting the interval $[h_\star, h^\star]$ in halves until sufficiently small, we propose an *adaptive bisection method* that can reduce the number of bisections.

We describe the adaptive bisection method. Ideally, we would like to know if the root is closer to $h_\star$ or $h^\star$. If we believe that the root is much closer to $h_\star$, we do not have to bisect $[h_\star, h^\star]$ with the mid-point, but perhaps with a point much closer to $h_\star$. Likewise, if the root were much closer to $h^\star$, we would like to bisect at a point closer to $h^\star$. In essence, we would like to quantify a *prior* of the root being closer to $h_\star$. With this in mind, note that in (12), we approximated the problem of finding the smallest $h$ such that $\varphi(h) \leq 0$ by solving the problem for an upper bound of $\varphi$. Then, the more accurate the approximation, the closer $h^\star$

6

is to the root. The approximation essentially came from using the fact that $D_{ii}h^\star \leq D_{ii}h^\star + \lambda$ for $D_{ii} > 0$. This motivates us to consider the worst approximation error (rate)

$$w := \max_{i:D_{ii}>0} \frac{\lambda}{D_{ii}h^\star + \lambda} = \frac{\lambda}{D_\star h^\star + \lambda} \in [0,1]$$

where $D_\star := \min_{i:D_{ii}>0} D_{ii}$. If $w$ is small, the approximation in (12) is tight, which implies that the root is close to $h^\star$. Hence, $1 - w$ represents the prior that the root is close to $h^\star$, or in other words, $w$ is the prior that the root is close to $h_\star$. We bisect at the new point $\tilde{h} := wh_\star + (1 - w)h^\star$. If $\varphi(\tilde{h}) \geq 0$, we use $h^{(0)} := \tilde{h}$ as the initial point for the Newton Method. Otherwise, we set $h^\star := \tilde{h}$ and repeat the argument. Algorithm 4 summarizes this procedure in its most basic form.

---

**Algorithm 4:** Newton Method with Adaptive Bisection

**Data:** $D$, $v$, $\lambda$, $\epsilon$

**1** Compute $h_\star$ that solves: $\sum_{i=1}^{p}(D_{ii}h + \lambda)^2 \leq \|v\|_1^2$;

**2** Compute $h^\star := \sqrt{\dfrac{\sum\limits_{i:D_{ii}>0} \frac{v_i^2}{D_{ii}^2}}{1-\lambda^{-2}\sum\limits_{i:D_{ii}=0} v_i^2}}$;

**3** $D_\star \leftarrow \min\limits_{i:D_{ii}>0} D_{ii}$;

**4** $w \leftarrow \frac{\lambda}{D_\star h^\star + \lambda}$;

**5** $h \leftarrow wh_\star + (1 - w)h^\star$;

**6 while** $\varphi(h) < 0$ *and* $|\varphi(h)| > \epsilon$ **do**

**7**    $h^\star \leftarrow h$;

**8**    $w \leftarrow \frac{\lambda}{D_\star h^\star + \lambda}$;

**9**    $h \leftarrow wh_\star + (1 - w)h^\star$;

**10 if** $|\varphi(h)| \leq \epsilon$ **then**

**11**    $\beta^\star \leftarrow (D + \frac{\lambda}{h}I)^{-1}v$;

**12**    return $\beta^\star$;

**13 else**

**14**    Apply Newton's Method starting at $h^{(0)} = h$;

---

Algorithm 4 can be further optimized. For example, if $h^\star - h_\star$ is below some threshold (e.g. 0.1), then we may skip bisection entirely and start Newton's Method at $h^{(0)} = h_\star$. This is because the adaptive bisection may move too slowly if the range is too small. On a similar note, one may promote enough movement towards $h_\star$ by taking the max of $w$ with a minimal probability (e.g. 0.05). This will ensure that at least 5% of $h_\star$ is taken if the prior is too strongly suggestive that the root is close to $h^\star$. It is always better to overshoot towards $h_\star$ such that $\varphi$ is non-negative, so that Newton's Method can quickly converge down to the root, than to slowly bisect to the root.

# References

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. doi: 10.1137/080716542. URL https://doi.org/10.1137/080716542.

Trevor Hastie, Rob Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity*. Chapman & Hall, 2016.

J. Klosa, N. Simon, P.O. Westermark, V. Liebscher, and D. Wittenburg. Seagull: lasso, group lasso and sparse-group lasso regularization for linear regression models via proximal gradient descent. *BMC Bioinformatics*, 21(407), 2020. doi: 10.1186/s12859-020-03725-w.

Xiaoxuan Liang, Aaron Cohen, Anibal Solón Heinsfeld, Franco Pestilli, and Daniel J. McDonald. sparsegl: An r package for estimating sparse group lasso, 2022. URL `https://arxiv.org/abs/2208.02942`.

Ignace Loris. On the performance of algorithms for the minimization of $\ell_1$-penalized functionals. *Inverse Problems*, 25(3):035008, jan 2009. doi: 10.1088/0266-5611/25/3/035008. URL `https://dx.doi.org/10.1088/0266-5611/25/3/035008`.

Lukas Meier, Sara Van De Geer, and Peter Böhlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society B*, 70(1):53–71, 2008. doi: 10.1111/j.1467-9868.2007.00627.x.

Brendan O'Donoghue and Emmanuel Candés. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15:715–732, 2015. doi: https://doi-org.stanford.idm.oclc.org/10.1007/s10208-013-9150-3.

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494, 2001. doi: https://doi-org.stanford.idm.oclc.org/10.1023/A:1017501703105.

Stephen J. Wright, Robert D. Nowak, and MÁrio A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009. doi: 10.1109/TSP.2009.2016892.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society B*, 68(1):49–67, 2006. doi: 10.1111/j.1467-9868.2005.00532.x.