2.2.2 Lambda sequence

In general, the solution to (2.2) is computed over a decreasing sequence of values for the tuning parameter λ, beginning with the smallest value λ_{max} for which the entire coefficient vector $\widehat{\beta} = \mathbf{0}_p$ (Friedman et al., 2010). To determine λ_{max}, we turn to the Karush-Kuhn-Tucker (KKT) optimality conditions for (2.2). These conditions can be written as

$$\begin{aligned} &\frac{1}{v_j}\sum_{i=1}^{n} w_i X_{ij}\left(y_i - \sum_{j=1}^{p} X_{ij}\hat{\beta}_j\right) = \lambda\gamma_j, \\ &\gamma_j \in \begin{cases} \text{sign}(\hat{\beta}_j) & \text{if} \quad \hat{\beta}_j \neq 0 \\ [-1,1] & \text{if} \quad \hat{\beta}_j = 0 \end{cases}, \qquad \text{for } j = 1,\ldots,p \end{aligned} \tag{2.20}$$

where γ_j is the subgradient of the function $f(x) = |x|$ evaluated at $x = \hat{\beta}_j$. From (2.20), we can solve for the smallest value of λ such that the entire vector $(\hat{\beta}_1, \ldots, \hat{\beta}_p)$ is 0. This is given by

$$\lambda_{max} = \max_j \left\{ \left| \frac{1}{v_j}\sum_{i=1}^{n} w_i X_{ij} y_i \right| \right\}, \quad j = 1,\ldots,p \tag{2.21}$$

Following Friedman et al. (2010), we can choose $\tau\lambda_{max}$ to be the smallest value of tuning parameters λ_{min}, and construct a sequence of K values decreasing from λ_{max} to λ_{min} on the log scale. The defaults are set to $K = 100$, $\tau = 0.01$ if $n < p$ and $\tau = 0.001$ if $n \geq p$. The optimal value of λ can be chosen using 5-fold or 10-fold cross-validation. For least-squares loss, this corresponds to choosing the λ which minimizes the mean squared error.

2.2.3 Warm starts

The way in which we have derived the sequence of tuning parameters using the KKT conditions allows us to exploit warm starts which has been shown to lead to computational speedups (Friedman et al., 2010). That is, the solution $\widehat{\Theta}$ for λ_k is used as the initial value $\Theta^{(0)}$ for λ_{k+1}.

2.2.4 Adaptive lasso

It has been shown that the lasso estimator can be produce biased estimates for large coefficients and give inconsistent variable selection results at the optimal λ for prediction, i.e., many noise features are included in the prediction model (Zou, 2006). To overcome the bias problems of the lasso, Zou (2006) proposed the adaptive lasso which allows a different amount of shrinkage for each regression coefficient using adaptive weights. Adaptive weighting has been shown to construct oracle procedures (Fan & Li, 2001), i.e., asymptotically, it performs as well as if the true model were given in advance. The adaptive lasso can be described as a two-stage procedure:

1. Calculate the initial regression estimates $\widehat{\boldsymbol{\beta}}_{init}$ from (2.2)
2. Refit (2.2) using penalty factors v_j equal to $1/\left|\hat{\beta}_{init,j}\right|$ for $j = 1, \ldots, p$.

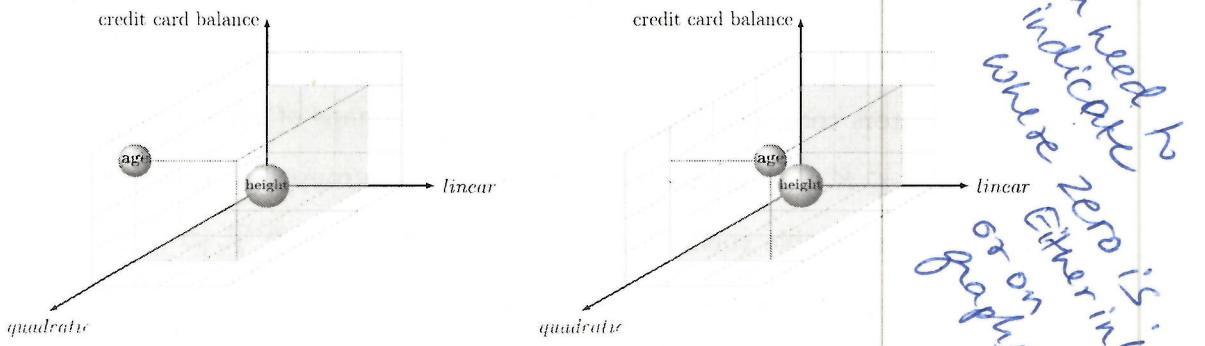
As we can see from the weights, the adaptive lasso will shrink larger coefficients less which leads to consistent variable selection results under weaker conditions than the lasso (Bühlmann & Van De Geer, 2011). We detail the adaptive lasso procedure in Algorithm 2.

Algorithm 2: Adaptive lasso algorithm

1. For a decreasing sequence $\lambda = \lambda_{max}, \ldots, \lambda_{min}$, fit the lasso with $v_j = 1$ for $j = 1, \ldots, p$
2. Use cross-validation or a data splitting procedure to determine the optimal value for the tuning parameter: $\lambda^{[opt]} \in \{\lambda_{max}, \ldots, \lambda_{min}\}$
3. Let $\hat{\beta}^{[opt]}_{init,j}$ for $j = 1, \ldots, p$ be the coefficient estimates corresponding to the model at $\lambda^{[opt]}$
4. Set the weights to be $v_j = \left(\left|\hat{\beta}^{[opt]}_{init,j}\right|\right)^{-1}$ for $j = 1, \ldots, p$
5. Refit the lasso with the weights defined in step 4), and use cross-validation or a data splitting procedure to choose the optimal value of λ

2.3 Group Lasso

One main drawback of the lasso is that it ignores the grouping structure of the design matrix. When given a predetermined grouping of non-overlapping variables, we would want might



(a) Lasso (b) Group Lasso

Figure 2.1: The selected components from Model (2.22) for (a) the lasso and (b) the group lasso. In this toy example, the lasso selects only the quadratic term for age while the group lasso selects both linear and quadratic terms.

all members of the group to be either zero or non-zero. For example, when dealing with categorical predictors where each factor is expressed through a set of indicator variables, removing an irrelevant factor is equivalent to setting the coefficients of the indicator variables to 0. In an additive model where each variable is projected on to a set of basis function, e.g. $f_j(X_j) = \sum_{\ell=1}^{m_j} \psi_{j\ell}(X_j)\beta_{j\ell}$, we would want all $\{\beta_{j\ell}\}_{\ell=1}^{m_j}$ to be either zero or non-zero. This key difference between the lasso and group lasso penalty is shown in Figure 2.1. Suppose we want to predict an individual's credit card balance from their age and height using the following additive model:

$$\text{credit card balance} = \beta_0 + \beta_{11}\text{age} + \beta_{12}\text{age}^2 + \beta_{21}\text{height} + \beta_{22}\text{height}^2 + \varepsilon \tag{2.22}$$

In Figures 2.1a and 2.1b we see that both the lasso and group lasso set the linear and quadratic terms for height $(\hat{\beta}_{21}, \hat{\beta}_{22})$ to 0. However, the lasso estimates only a nonzero quadratic term for age $(\hat{\beta}_{11} = 0, \hat{\beta}_{12} \neq 0)$ while the group lasso estimates both linear and quadratic terms to be nonzero $(\hat{\beta}_{11} \neq 0, \hat{\beta}_{12} \neq 0)$. We now provide details on the group lasso estimator.

Assume that the predictors in the design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ belong to K groups and define the

cardinality of index set I_k to be p_k. These groups are known *a priori* such that $(1, 2, \ldots, p) = \bigcup_{k=1}^{K} I_k$, and are also non-overlapping, i.e., $I_k \bigcap I_{k'} = \emptyset$ for $k \neq k'$. Therefore, group k contains p_k predictors corresponding to $\mathbf{X}^{(k)}$, i.e., the columns of the design matrix X_j for $j \in I_k$, and $1 \leq k \leq K$. The intercept belongs to its own group, i.e., $I_1 = \{1\}$. The group lasso partitions the variable coefficients into K groups $\boldsymbol{\beta} = ([\boldsymbol{\beta}^{(1)}]^\intercal, [\boldsymbol{\beta}^{(2)}]^\intercal, \cdots, [\boldsymbol{\beta}^{(K)}]^\intercal)^\intercal$, where $\boldsymbol{\beta}^{(k)}$ denotes the segment of $\boldsymbol{\beta}$ corresponding to group k. For least-squares loss, the group lasso estimator (Yuan & Lin, 2006) is given by:

$$\widehat{\boldsymbol{\beta}}(\lambda) = \underset{(\beta_0, \boldsymbol{\beta})}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} w_i (y_i - \beta_0 - (\mathbf{X}\boldsymbol{\beta})_i)^2 + \lambda \sum_{k=1}^{K} v_k \|\boldsymbol{\beta}^{(k)}\|_2 \tag{2.23}$$

where $\left\|\boldsymbol{\beta}^{(k)}\right\|_2 = \sqrt{\sum_{j \in I_k} \beta_j^2}$ and $\lambda > 0$ is the tuning parameter. As in the lasso estimator (2.2), there are both observation weights w_i, and penalty factors $v_k \geq 0$ which control the relative strength of the terms within the group lasso penalty. These penalty factors are often set to $\sqrt{p_k}$ (Yuan & Lin, 2006). Note that the same penalty factor is applied to all the coefficients in a group. Solving the group lasso estimator is more challenging than the lasso since there is no closed form solution for (2.23). In the next section, we detail a majorization-minimization (MM) type algorithm (Lange et al., 2000; Y. Yang & Zou, 2015) used to solve (2.23).

2.3.1 Groupwise majorization descent algorithm

This description of the groupwise majorization descent (GMD) algorithm used to solve (2.23) follows mainly from Y. Yang & Zou (2015). The main difference here is that we consider a more general loss function of the form

$$L(\boldsymbol{\beta} \mid \mathbf{D}) = \frac{1}{2} [y - \widehat{y}]^\top \mathbf{W} [y - \widehat{y}] \tag{2.24}$$

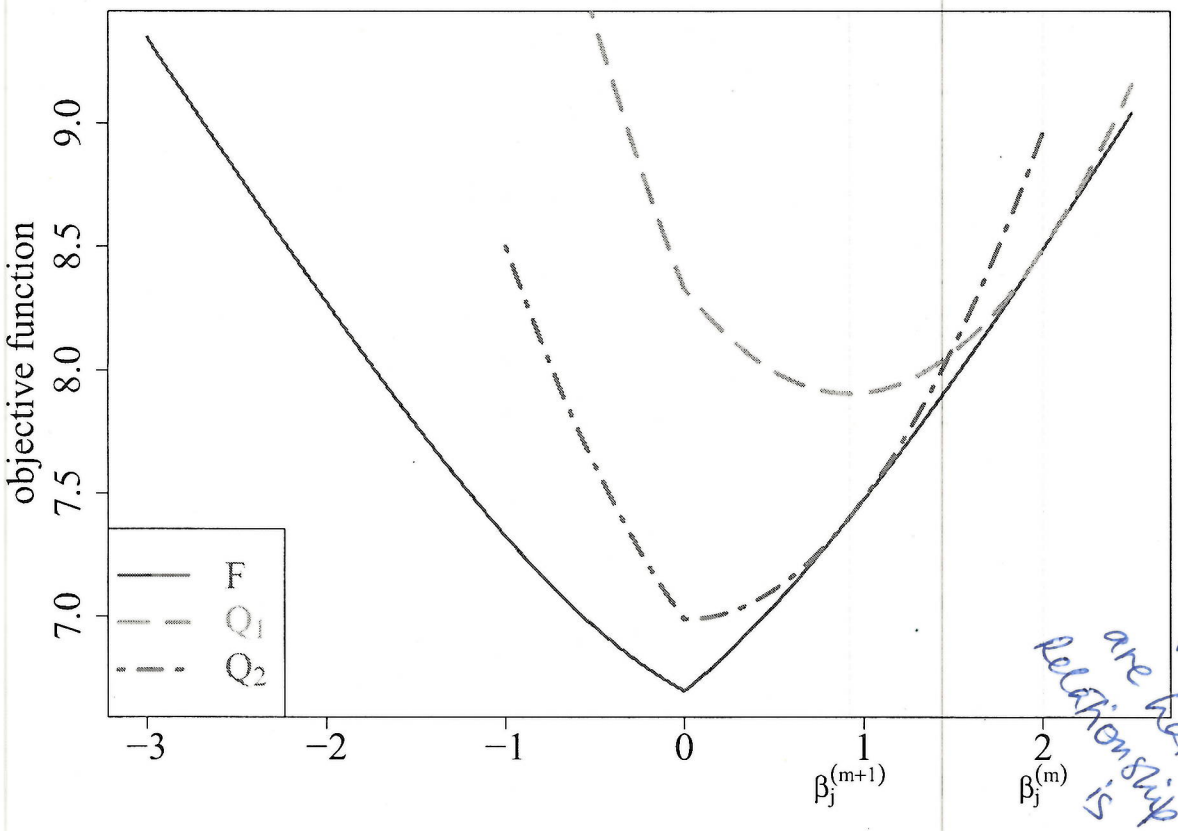


Figure 2.2: Illustration of the quadratic majorization technique

In Figure 2.2 we provide an illustration of the quadratic majorization technique for updating a parameter β_j. The solution lies at the minimum of the F curve but we cannot solve for this directly since there is no closed form solution. Instead we majorize F using Q_1 which is a function consisting of the quadratic approximation of F plus the penalty term evaluated at $\beta_j^{(m)}$. The minimum of Q_1, for which there is a closed form solution, corresponds to the next iteration $\beta_j^{(m+1)}$. We then majorize again using Q_2 and solve for the minimum. This process is repeated until convergence.

2.3.2 Lambda sequence

Similar to Section 2.2.2, we compute the solution to (2.23) over a decreasing sequence of values for the tuning parameter λ starting with λ_{max}. From the update formula (2.32) we

Algorithm 3: The GMD algorithm for group lasso with least-squares loss function given by (2.24).

1. For $k = 1, \ldots, K$, compute γ_k, the largest eigenvalue of $\mathbf{H}^{(k)} = (\mathbf{X}^{(k)})^\top \mathbf{W} \mathbf{X}^{(k)}$
2. Initialize $\widetilde{\boldsymbol{\beta}}$.
3. Repeat the following cyclic groupwise updates until convergence:
   — for $k = 1, \ldots, K$, do step (3.1)–(3.3)
   3.1 Compute $U(\widetilde{\boldsymbol{\beta}}) = -\nabla L(\widetilde{\boldsymbol{\beta}}|\mathbf{D}) = -(\boldsymbol{y} - \hat{\boldsymbol{y}})^\top \mathbf{W} \mathbf{X}^{(k)}$
   3.2 Compute $\widetilde{\boldsymbol{\beta}}^{(k)}(\text{new}) = \frac{1}{\gamma_k}\left(U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\right)\left(1 - \frac{\lambda v_k}{\|U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\|_2}\right)_+$.
   3.3 Set $\widetilde{\boldsymbol{\beta}}^{(k)} = \widetilde{\boldsymbol{\beta}}^{(k)}(\text{new})$.

have that for all k

$$\begin{cases} \widetilde{\boldsymbol{\beta}}^{(k)} = \frac{1}{\gamma_k}\left(U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\right)\left(1 - \frac{\lambda v_k}{\|U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\|_2}\right) & \text{if } \|U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\|_2 > \lambda v_k \\ \widetilde{\boldsymbol{\beta}}^{(k)} = \mathbf{0} & \text{if } \|U^{(k)} + \gamma_k \widetilde{\boldsymbol{\beta}}^{(k)}\|_2 \leq \lambda v_k \,. \end{cases}$$

We can then directly obtain the KKT conditions for $k = 1, \ldots, K$:

$$\begin{aligned} -U^{(k)} + \lambda v_k \cdot \frac{\widetilde{\boldsymbol{\beta}}^{(k)}}{\|\widetilde{\boldsymbol{\beta}}^{(k)}\|_2} = \mathbf{0} \qquad & \text{if } \widetilde{\boldsymbol{\beta}}^{(k)} \neq \mathbf{0}, \\ \left\|U^{(k)}\right\|_2 \leq \lambda w_k \qquad & \text{if } \widetilde{\boldsymbol{\beta}}^{(k)} = \mathbf{0} \,. \end{aligned} \tag{2.33}$$

Using (2.33) we can solve for the smallest value of λ such that the entire vector $\left\{\widehat{\boldsymbol{\beta}}^{(k)}\right\}_{k=1}^{K}$ is 0. This is given by

$$\lambda_{max} = \max_k \frac{1}{v_k}\left\|U^{(k)}\right\|_2, \quad k = 1, \ldots, K, \; v_k \neq 0 \tag{2.34}$$

2.3.3 Warm starts and adaptive group lasso

Warm starts can also be implemented for the group lasso as described in Section 2.2.3 for the lasso. Furthermore, the adaptive group lasso can be computed using Algorithm 2; the

Can you make it clearer what is published and what is your algorithm improvements?