Penalization for sparsity - Optimization techniques

Benjamin Labrecque
Department of Computer Science
McGill University
Montreal, QC H3A 2A7
benjamin.labrecque@mail.mcgill.ca

1 Introduction

A sparse statistical model is one having only a small number of nonzero weights. A sparse model can be much easier to estimate and interpret than a dense model - a classic case of *less is more*. With datasets growing rapidly in this age, the number of features measured on a person or image can be large or even exceed the number of observations.

Medical scientists study the genomes of humans to predict diseases and their best treatments. Online movie and book providers study customer ratings to recommend or sell them new movies and books. Social networks collect information on users to deliver the best possible user experience. There is a crucial need to make sense of this information, which leads us to the assumption of simplicity. One form of simplicity is sparsity, which is the focus of this project. I focus specifically on linear regression with Lasso and Group-Lasso penalization.

We observe N observations of an outcome variable y_i and p associated predictor variables (or features) $x_i = (x_{i1}, ..., x_{ip})^T$. The goal is to figure out which predictors play an important role for future predictions when presented with new data points. Using a linear regression model we make the following assumption,

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

where β_0 and $\beta = (\beta_1, \beta_2, ..., \beta_j)$ are unknown parameters to be estimated and e_i is an error term. We can use the method of least squares to estimate the parameters by minimizing the least-squares objective function

$$\underset{\beta_0,\beta}{\text{minimize}} \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_{ij})^2 \tag{2}$$

Generally, all the least-squares estimates will be nonzero. This makes the final model hard to interpret. Especially, if $p > N$ then there are infinitely many sets that make the objective function equal to zero. In addition, these solutions will most certainly overfit the data. To avoid this, there is a need to regularize the estimation process by adding constraints to the estimates. In the $\ell_1 - regularized$ or $lasso\ regression$ we obtain the estimates by solving the following problem

$$\underset{\beta_0,\beta}{\text{minimize}} \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_{ij})^2 \text{ subject to } \|\beta\|_1 \leq t \tag{3}$$

where $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$ is the ℓ_1 norm of β and t a user-specified parameter. We can think of t as a constraint on the ℓ_1 norm of the parameter vector, and the lasso finds the best fit within this constraint.

Equation (3) is often represented using matrix-vector notation.

$$\underset{\beta_0,\beta}{\text{minimize}} \left\{ \frac{1}{2N} \|\mathbf{y} - \beta_0 \mathbf{1} - \mathbf{X}\beta\|_2^2 \right\} \text{ subject to } \|\beta\|_1 \leq t \tag{4}$$

where $\mathbf{1}$ is the vector of N ones, $\mathbf{y} = (y_1, ..., y_N)$ the N-vector of responses, $\mathbf{X}$ an $N \times p$ matrix with $x_i \in \mathbb{R}^p$ as its i^{th} row and $\|\cdot\|_2$ the usual Euclidean norm on vectors.

It is often convenient to represent the lasso problem in Lagrangian form

$$\underset{\beta_0,\beta}{\text{minimize}} \left\{ \underbrace{\frac{1}{2N} \|\mathbf{y} - \mathbf{X}\beta\|_2^2}_{:=g(\beta)} + \underbrace{\lambda \|\beta\|_1}_{:=h(\beta)} \right\} = g(\beta) + h(\beta) \tag{5}$$

for some $\lambda \geq 0$. By Lagrangian duality there is a one-to-one correspondence between the solution of (4) and (5): for each value of t where $\|\beta_1\| \leq t$ there is a corresponding λ that yields the same solution of the Lagrangian form (5). Conversely, the solution $\hat{\beta}_\lambda$ to problem (5) solves the bound problem (4) with $t = \|\hat{\beta}_\lambda|_1$.

The ℓ_q norm with $q = 1$ has special properties. If the constraint t is small enough, the lasso yields sparse solution vectors, whereby only some coordinates are nonzero. This does not occur when $q > 1$; when $q < 1$ the solutions are sparse but the problem is not convex which makes the minimization computationally very challenging. The value $q = 1$ is thus the smallest value that yields a convex problem. The convexity and sparsity assumptions greatly simplify the computations and allow for efficient and scalable algorithms with even millions of parameters.

Furthermore, we note that in equation (5) $g(\beta)$ is convex and differentiable whereas $h(\beta)$ is convex but non-differentiable. We can thus not run vanilla gradient descent. We can make a quadratic approximation to g and leave h alone thus computing,

$$\begin{aligned}
\beta^{(k+1)} &= \underset{\beta}{\text{argmin}} \left\{ g(\beta) + \nabla g(\beta)^T (\beta - \beta^{(k)}) + \frac{1}{2t_k} \|\beta - \beta^{(k)}\|_2^2 + h(\beta) \right\} \\
&= \underset{\beta}{\text{argmin}} \left\{ \frac{1}{2t_k} \|\beta - (\beta^{(k)} - t_k \nabla g(\beta^{(k)}))\|_2^2 + h(\beta) \right\} \\
&= \text{prox}_{t_k}\big(\beta^{(k)} - t_k \nabla g(\beta^{(k)})\big)
\end{aligned} \tag{6}$$

where $\text{prox}_t(x)$ is a proxial mapping defined as

$$\text{prox}_t(x) = \underset{\beta}{\text{argmin}} \left\{ \frac{1}{2t} \|\beta - x\|_2^2 + h(\beta) \right\} \tag{7}$$

This leaves us with the *proximal gradient descent* algorithm

Algorithm 1: Proximal Gradient Descent

for k = 0, 1, 2, ...:

$$\beta^{(k+1)} = \text{prox}_{t_k}\big(\beta^{(k)} - t_k \nabla g(\beta^{(k)})\big)$$

where t_k is the learning rate

1.1 Proximal gradient methods for lasso

For the lasso we have $g(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$ and $h(\beta) = \lambda\|\beta\|_1$. We now have the following proximal mapping

$$\begin{aligned} \text{prox}_t(\beta) &= \underset{z}{\text{argmin}} \left\{ \frac{1}{2t}\|\beta - z\|_2^2 + \lambda\|z\|_1 \right\} \\ &= S_{\lambda t}(\beta) \end{aligned} \tag{8}$$

where $S_{\lambda t}(\beta)$ is the soft-thresholding operator. Hence the proximal gradient update for the lasso is

$$\beta^{(k+1)} = S_{\lambda t}(\beta^{(k)} + tX^T(y - X\beta^{(k)})) \tag{9}$$

1.2 Proximal gradient methods for group lasso

For the lasso we have $g(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$ and $h(\beta) = \lambda \sum_j w_j \|\beta_{(j)}\|_2$ where the predictors in the regression problem are split into J groups

$$X = \begin{bmatrix} \mathbf{1} & X_{(1)} & X_{(2)} & ... & X_{(J)} \end{bmatrix}$$

where $\mathbf{1} = (1\ 1\ ...\ 1) \in \mathbb{R}^n$.

We now have the following proximal mapping

$$\text{prox}_t(\beta) = \psi_{bst}(\beta; \lambda) := \left[\left(1 - \frac{\lambda}{\|\beta_{(j)}\|_2} \right)_+ \beta_{(j)} \right]_{1 \le j \le J} \tag{10}$$

where $\psi_{bst}(\beta; \lambda)$ is the block soft-thresholding operator. Hence the proximal gradient update for the lasso is

$$\beta^{(k+1)} = \psi_{bst}(\beta^{(k)} + tX^T(y - X\beta^{(k)});\ t_k\lambda) \tag{11}$$

2 Predicting Parkinson's Disease (PD) score

In the following problem I will use linear regression to predict the Parkinson's Disease score on the Parkinson's dataset. The PD symptom score is measured on the unified Parkinson's disease rating scale (UPDRS). This data contains 5,785 observations, 18 predictors, and an outcome—the total UPDRS. The data were collected at the University of Oxford, in collaboration with 10 medical centers in the US and Intel Corporation. The 18 columns in the predictor matrix have the following groupings (in column ordering):

- `age`: Subject age in years
- `sex`: Subject gender, '0'–male, '1'–female
- `Jitter(%), Jitter(Abs), Jitter:RAP, Jitter:PPQ5, Jitter:DDP`: Several measures of variation in fundamental frequency of voice
- `Shimmer, Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, Shimmer:APQ11, Shimmer:DDA`: Several measures of variation in amplitude of voice
- `NHR, HNR`: Two measures of ratio of noise to tonal components in the voice
- `RPDE`: A nonlinear dynamical complexity measure
- `DFA`: Signal fractal scaling exponent
- `PPE`: A nonlinear measure of fundamental frequency variation

2.1 Ridge Penalization - L2

I first consider the Ridge Regression problem

$$\underset{\beta \in \mathbb{R}^{p+1}}{\text{minimize}} \left\{ \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \frac{\lambda}{2} \|\beta\|_2^2 \right\} \tag{12}$$

where N is the number of samples. To solve this problem I use stochastic gradient descent (SGD) with different batch sizes B and step sizes t. The stochastic gradient update w.r.t B and t can be derived as

$$\beta := \beta - t * \nabla f(\beta) \tag{13}$$

$$\beta_j^{(k+1)} = \beta_j^{(k)} - t \sum_{i=1}^{B} X_{ij}(y_i - \hat{y}_i) - t\lambda\beta_j^{(k)} \tag{14}$$

where $\hat{y}_i = \sum_j X_{ij}\beta_j^{(k)}$.

2.2 Lasso ($\ell 1$) and Group Lasso ($\ell 1 \backslash \ell 2$) penalization

Next, I consider the least squares lasso and group lasso problems.

The group lasso problem

$$\underset{\beta \in \mathbb{R}^{p+1}}{\text{minimize}} \left\{ \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \sum_j w_j \|\beta_{(j)}\|_2 \right\} \tag{15}$$

forces all the variables withing one group to be zero or nonzero.

To increase the convergence rate we can add Nesterov acceleration to the proximal gradient descent. When calculating the update for $\beta^{(k+1)}$, instead of just taking $\beta^{(k)}$ into account we also consider a momentum term $\beta^{(k-1)}$ which allows for some of the history to push us a little further.

Algorithm 2: Fast Iterative Block Thresholding

for k = 0, 1, 2, ...:

$$z = \beta^{(k)} + \frac{k}{k+3}(\beta^{(k)} - \beta^{(k-1)})$$
$$\beta^{(k+1)} = \text{prox}_{t_k}\big(z - t_k \nabla g(z)\big)$$

Clearly, acceleration here is a very useful speedup tool. It should be noted however that IBTA can perform almost identically to FIBTA if we use warm starts.

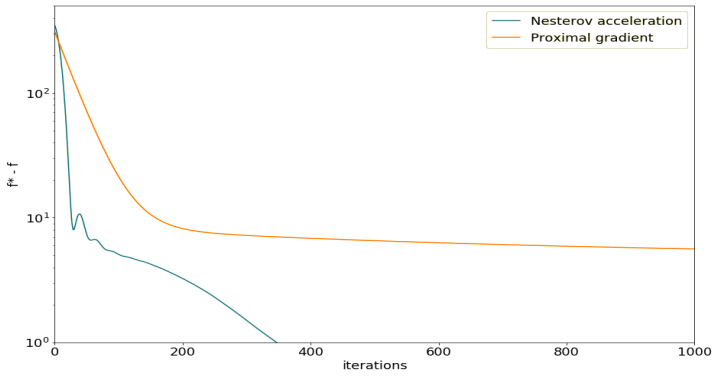
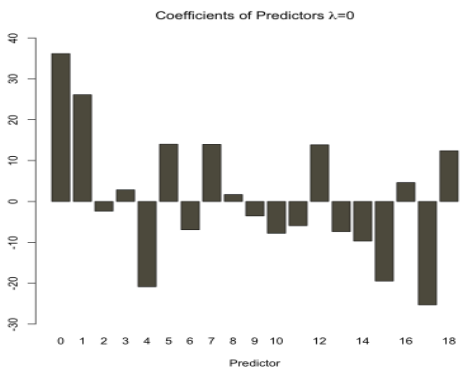


Figure 1: IBTA vs FIBTA

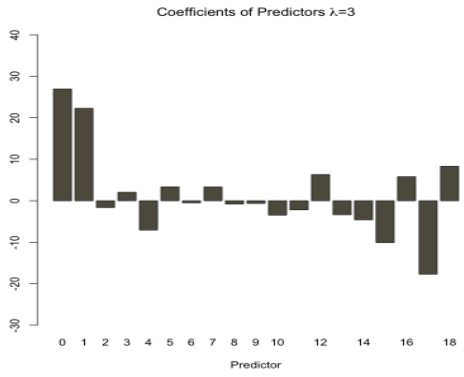
2.3 Results

2.3.1 Ridge

Adding ridge penalization to least squares regression shrinks the values of the coefficients and thus prevents overfitting to the data the model was trained on. However, none of the coefficients shrink to zero leaving us with a dense model. In other words, the number of predictors in our model remains the same as if we fit a standard least squares regression to the data.



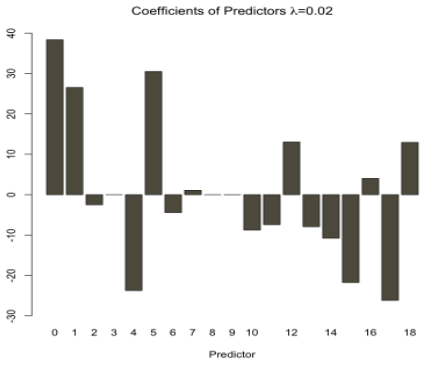
(a) $\lambda = 0$



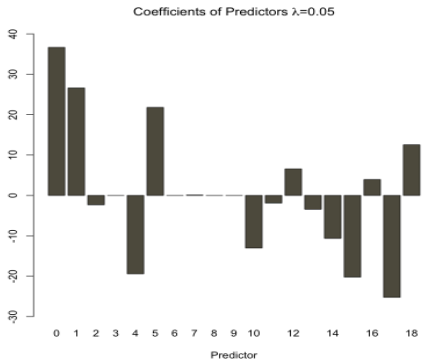
(b) $\lambda = 3$

2.3.2 Lasso

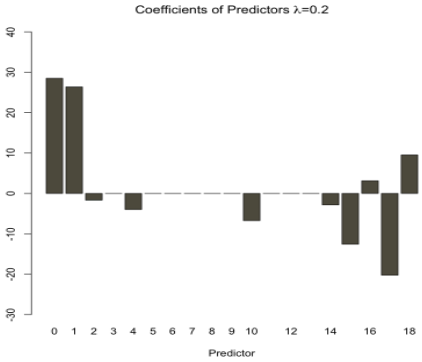
As we can see below, a larger penalization constant forces more predictors to be zero. This leaves us with a sparser model. From the last figure we can observe how all coefficients eventually go to zero as lambda increases. By adjusting lambda accordingly we can control the sparsity of our model to make it more interpretable while keeping it complex enough to capture all the important information.



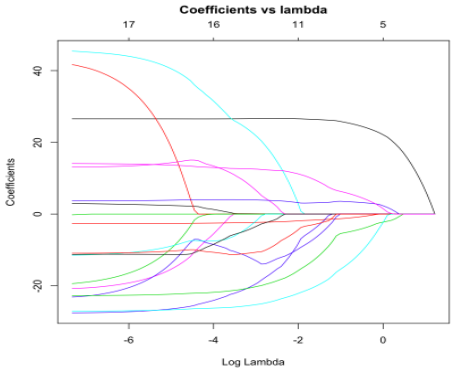
(a) $\lambda = 0.02$



(b) $\lambda = 0.05$



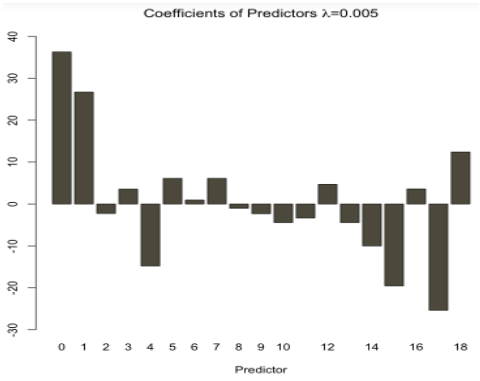
(a) $\lambda = 0.2$



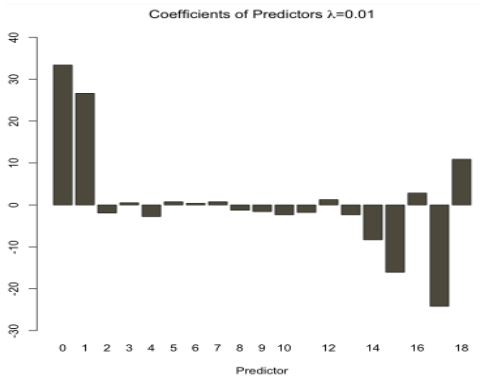
(b) Coefficients shrinking with increased λ

2.3.3 Group Lasso

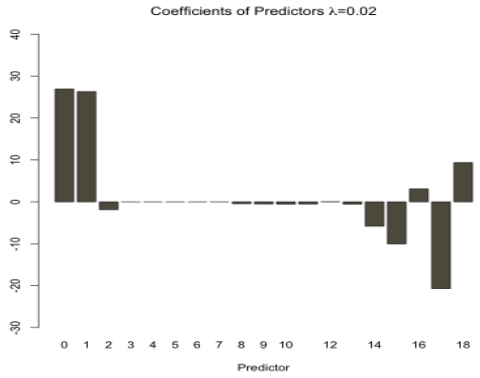
The group lasso forces entire groups of predictors to be zero or non-zero. From the figures below it is clear that the predictors using measures for variation in fundamental frequency of voice and measures of variation in amplitude of voice do not have predictive power. The coefficients $\beta_{(3)} = [\beta_3, \beta_4, \beta_5, \beta_6, \beta_7,]$ and $\beta_{(4)} = [\beta_8, \beta_9, \beta_{10}, \beta_{11}, \beta_{12}, \beta_{13}]$ are set to zero leaving us with a sparse model where only 6 out of the 8 predictor groups are nonzero. The behaviour described above is also reflected in figure 4 where all coefficients within a group tend to go to zero at the same value of λ.



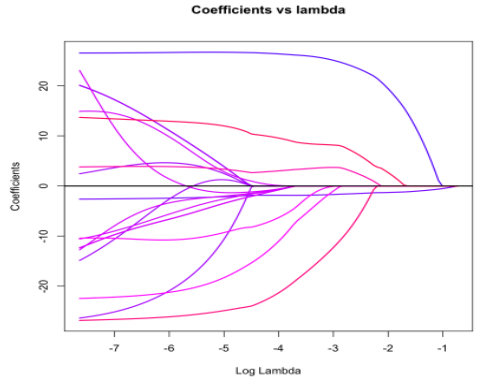
(a) $\lambda = 0.005$



(b) $\lambda = 0.01$.



(a) $\lambda = 0.02$



(b) Coefficients shrinking groupwise with increased λ