Computational Statistics

LASSO

The LASSO (Least Absolute Shrinkage and Selection Operator) imposes an ℓ_1 (absolute value) penalty on model coefficients,

$$\hat{B}^{\text{LASSO}} = \underset{B}{\operatorname{argmin}} (\mathrm{y} - \mathrm{X}\,B)'(\mathrm{y} - \mathrm{X}\,B) + \lambda \sum_{j=1}^{p} |\beta_j|,$$

In LASSO the hyperparameter λ controls the amount of shrinkage. The larger the value of λ, the more shrinkage (coefficients are shrunk towards zero). Unlike ridge regression, the LASSO is not differentiable and cannot be solved directly.

Dual Formulation

The optimization problem for the LASSO (and ridge regression) has a dual formulation,

$$\hat{B}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} (\mathrm{y} - \mathrm{X}\, B)'(\mathrm{y} - \mathrm{X}\, B),$$

subject to

$$\sum_{j=1}^{p} |\hat{\beta}^{\text{lasso}}| \leq t.$$

For the dual formulation of the problem, there is one-to-one relationship between λ and t.

Least Angle Regression

Least Angle Regression (LAR) is an algorithm for computing the LASSO estimates for a linear regression problem.

Note: LAR was originally developed as a 'continuous' version of subset selection, but with only a small modification it turned into an efficient algorithm for computing the LASSO solution.

Sparsity

Consider increasing a parameter $\hat{\beta}$ by a small amount $d\beta$:

- the penalty will be the same regardless of the value of β
- the LASSO will always increase whichever parameters are most correlated with the residuals

The LAR algorithm exploits this property of the LASSO.

Least Angle Regression

intuition

The intuition for the LAR algorithm is as follows: starting from an initial estimate of $\hat{\beta}_j = 0$ for all j:

- Consider the model residuals, $\hat{\epsilon} = y - \hat{y} = y - \mathrm{X}\hat{\beta}$.
- Find the input x_j with the highest correlation with $\hat{\epsilon}$
- Increase the corresponding $\hat{\beta}_j$ (and thus decrease the correlation between x_j and $\hat{\epsilon}$)
- Continue to increase $\hat{\beta}_j$ (and decrease the correlation) until two inputs x_j and x_k are 'tied' for the highest correlation
- Increase $\hat{\beta}_j$ and $\hat{\beta}_k$ together
  - make sure the correlations with ϵ decrease together and remain equal, until x_j and x_k are tied with a third input
- Repeat the process until all correlations are zero (at which point, we have arrived at the least-squares solution)

Least Angle Regression

algorithm

- Start by scaling the inputs

$$y_i = y_i - \bar{y}, \quad x_{ij} = \frac{x_{ij} - \bar{x}_j}{\sqrt{\sum_{i=1}^{n}(x_{ij} - \bar{x})^2}}.$$

- Starting with $\hat{\beta}^{(1)} = \mathbf{0}$, repeat the following steps:
  - Compute correlation between x_j and $\hat{\epsilon}$ (this is only proportional to correlation)

$$c = \mathrm{X}'(\mathrm{y} - \hat{\mathrm{y}}), \quad C = \max_j |c_j|$$

  - Identify the input (or inputs) that are maximally correlated with $\hat{\epsilon}$

$$J = \{j : |c_j| = C\}$$

Least Angle Regression

algorithm

- Starting with $\hat{\beta} = \mathbf{0}$, repeat the following steps (continued):
  - Compute how much we should increase β_j for all $j \in J$
    - Create a reduced input matrix with only the data from J multiplied by the sign of c

$$\mathrm{X}_J = [\mathrm{sign}(c_j)x_j]_{j \in J}$$

    - Compute several intermediate quantities ($\mathbf{1}$ is a row vector of $|J|$ ones)

$$G_J = \mathrm{X}_J' \mathrm{X}_J$$
$$A_J = (\mathbf{1}' G_J^{-1} \mathbf{1})^{-1/2}$$
$$w_J = A_J G_J^{-1} \mathbf{1}$$
$$u_J = X_J w_J$$
$$a = X' u_J$$

Least Angle Regression

algorithm

- Starting with $\hat{\beta} = \mathbf{0}$, repeat the following steps (continued):
  - Compute how much we should increase β_j for all $j \in J$ (continued):
    - Identify the 'next most correlated variable' and how far to update $\hat{\beta}_j$'s
      $$\hat{\gamma} = \min_j^{+} \left\{ \frac{C - c_j}{A_J - a_j}, \frac{C + c_j}{A_J + a_j} \right\}$$
      where $\min^{+}$ is the smallest strictly positive value in the set
  - Update $\hat{\beta}_j$ for all $j \in J$
    $$\hat{\beta}_j^{(i+1)} = \hat{\beta}_j^{(i)} + \text{sign}(c_j)\hat{\gamma} w_j$$
- Repeat until $\hat{\beta}$ becomes the least-squares estimates

LASSO Path

The above algorithm results in a sequence $\hat{\beta}$ estimates, $\hat{\beta}_j^{(i)}$ $(i = 1, \ldots, (p+1))$, where p is the number of inputs. If $j \notin J$, then $\hat{\beta}_j$ stays at zero for the update. For each k, we can compute the LASSO penalty,

$$t_i = \sum_{j=1}^{p} |\hat{\beta}_j^{(i)}|$$

Plotting each sequence of updates $\hat{\beta}_j^{(i)}$ against t_i is called the 'LASSO Path', providing the solution to the LASSO problem for each possible t (and thus each possible λ because of the dual formulation).

LASSO Path

example

The LASSO path for a regression model with `lpsa` as output and `lcavol`, `lweight`, and `age` as inputs,