Least Absolute Shrinkage and Selection Operator (LASSO) in Python

Tyler Oshiro

May 20, 2013

Contents

Let

$N = $ the number of datapoints we are processing.

$k = $ the number of features for the data.

$\lambda = $ the regularization constant.

$t(x_j) = $ the response variable given a datapoint x_j.

$w_0 = $ our intercept for the optimization function.

$w_i = $ our weight vector for the data.

$h_i(x_j) = $ the weight of feature i for datapoint x_j.

Then the objective function we are trying to maximize is given by:

$$\hat{w} = \min_{w} \sum_{j=1}^{N} \left(t\left(x_j\right) - (w_0 + \sum_{i=1}^{k} w_i h_i(x_j)) \right)^2 + \lambda \sum_{i=1}^{k} |w_i| \qquad (1)$$

1 About

Since we want to perform "feature selection" for our data (i.e. choose which statistics of a baseball player are important for winning games), we will use LASSO to reduce the number of possible dimensions down from hundreds to approximately ten. The regularization penality that we use for LASSO is that of the L_2 norm. Doing so guarantees sparse solutions.

2 Logistics

Unfortunately, we cannot differentiate our objective function (1) easily and there is no closed-form solution for the minimization so we resort to an iterative coordinate descent method. We can do so because the LASSO objective is convex. Thus, using the subgradient will result in an optimal result.

3 Regularization

Regularization is done to penalize "complex" solutions with alrge weights. To choose our regularization constant we will use k-folds cross-validation. That is, we will split the data into k evenly distributed subgroups and then train our algorithm on all but one of the k subgroups and then test our results with the remaining subgroup. We repeat this process k times and average the errors reported. The final λ value will be chosen based on producing the lowest error using this technique.