

Foundation of Machine Learning (IT 582)

Autumn 2022

Pritam Anand

Given the training set $\{(x_i, y_i) : x_i \in \mathbf{R}^n, y_i \in \mathbf{R} \text{ for } i = 1, 2, \dots, l\}$, the regularized Least Squares Regression model solves the optimization problem

$$\min_{(w)} \frac{\lambda}{2} w^T w + \frac{1}{2} \sum_{i=1}^l (y_i - (w^T \phi(x_i)))^2 \quad (1)$$

where $\phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \dots \\ \dots \\ \phi_m(x) \end{bmatrix}$ and $\phi_j(x)$ for $j = 1, 2, \dots, m$, are m different basis functions.

Though the solution of the regularized Least Squares Regression models is computationally handul, they often obtain poor prediction if $m \gg l$ i.e., the number of basis functions used is much more than the number of available training data points. It is often the case when we are working in high dimensions with small sample size data.

LASSO Regression model

This problem with the Least Square Regression model can be solved if we can obtain a solution vector $w \in \mathbb{R}^m$, such that its several components are equal to zero. Obtaining a sparse solution vector w means that we are working with fewer basis functions which can reduce the risk of over-fitting. It allows us to work with a simpler model and decrease the variance of the estimated functions.

The Least Absolute Shrinkage and Selection Operator (LASSO) regression model minimizes the L_1 -norm of regularization along with the Least Squares loss function in its optimization problem. The minimization of L_1 -norm of regularization helps it to obtain a sparse solution vector. The LASSO regression model solves the problem

$$\min_{(w)} \frac{\lambda}{2} \|w\|_1 + \frac{1}{2} \sum_{i=1}^l (y_i - (w^T \phi(x_i)))^2, \quad (2)$$

where $w \in \mathbb{R}^m$ and $\|w\|_1$ is obtained by $\sum_{i=1}^m |w_i|$.

We note that objective function of the LASSO problem is convex but, no more differentiable as the $\|w\|_1$ is not differentiable at $\mathbf{0} \in \mathbb{R}^m$. But, we can still compute the subgradient for it. Before we make use of the sub-gradients

concept, we need to briefly define it.

Definition (Sub-gradients): A sub-gradient of a convex function f (f may not be differentiable) at $x \in \mathbb{R}^m$ is any $g \in \mathbb{R}^m$ such that the following holds

$$f(y) \geq f(x) + g^T(y - x), \forall y \in \mathbb{R}^m.$$

For example, if we consider a real number t , then $|t|$ is not differentiable at $t = 0$. But, the subgradient g can be obtained as $g = +1$ for $t > 0$, $g = -1$ for $t < 0$. and g is any element in $[-1, 1]$ for $t = 0$.

Subgradient descent method for LASSO regression model

For $w \in \mathbb{R}^m$, we can obtain the subgradients $g = \delta(\|w\|_1)$ using

$$g[i] = \begin{cases} = \text{sign}(w(i)), & \text{if } w[i] \neq 0. \\ \text{any element in } [-1, 1], & \text{if } w[i] = 0. \end{cases} \quad (3)$$

If we deonte the objective functions of the problem (2) using $J(w)$ then we can uniquely obtain the $m \times 1$ subgradient vector $\delta(J(w))$, if $w \neq \mathbf{0}$, using

$$\delta(J(w)) = g - \sum_{i=1}^l (y_i - (w^T \phi(x_i))) \phi(x_i). \quad (4)$$

Now, we list the steps of the subgradient descent algorithm which can obtain the solution to the LASSO regression problem (2).

Algorithm 1 Subgradient method for LASSO regression model

Input :- Training set T and toelarence tol
 Intailize $w^0 \in \mathbb{R}^m$
 Repeat
 $w^{k+1} = w^k - \eta \delta(J(w^k))$
 until $\|\delta(J(w^k))\| \leq tol$.

Also, we list the steps of the stochastic subgradient descent algorithm which can obtain the solution of the LASSO regression problem (2) in Algorithm 2.

Algorithm 2 Stochastic Subgradient method for LASSO regression model

Input :- Training set T and max_iter
 Intailize $w^0 \in \mathbb{R}^m$
for $i = 1$ to max_iter **do**
 Draw a random subset B of size k from Training set T .
 $w^{k+1} = w^k - \eta_i \delta(J_B(w^k))$
end for.

The $\delta(J_B(w^k))$ used in Algorithm 2 computes the subgradient for only k data points of set B as

$$\delta(J(w)) = g - \sum_{(x_i, y_i) \in B} (y_i - (w^T \phi(x_i))) \phi(x_i).$$

Beyond the Least Squares loss function

A major problem with the Least Squares based Regression models is that the estimates obtained by them can deviate significantly if we change a few of the y_i values with outliers (You can empirically realize this while doing Assignment 3). It is because of that the Least Squares loss function is not a robust loss function as it considers the square of errors.

In literature, you can find several ideas that attempt to bring robustness back to the Least Square Regression models. Most of them consider different weighting mechanisms, which assign higher weights to dense data points and assign lower weights to outliers. Can you think of a simple and effective version of the Weighted Least Square Regression model?

Thanks
Pritam.