HMMA307

Lasso vs FoBa

Cours: Tong Zhang *Scribes: Ophélie Coiffier*

In this report, we speak about three methods which predicted variables and with these predicted variables, we can compare the mean squared error on test and train values. For that, we use Lasso (using L_1 regularization), OMP ("Orthogonal Matching Pursuit", using forward greedy algorithm) and LARS (using forward and backward steps, such as FoBa (Forward-Backward) algorithm). Given that the FoBa function is available with the software R, we replace FoBa by LARS. But there is one significant difference : LARS tracks the path through L_1 penalized least squared error by gradually decreasing the regularization parameter and FoBa uses the insertion and deletion with unbiased least squared error. So, examples will not have the same results.

Which methods will be chosen to predict variables depending on the sparsity ?

First, we will describe our three methods then we apply them with real examples.

1 Mathematic approach

In this part, we will describe three methods to predict variables :

- The Lasso
- The OMP (Orthogoanl Matching Pursuit)
- The LARS (the closest equivalent for FoBa algorithm)

First, to measure the quality of prediction, we use a linear prediction model :

$$f(x) = \beta^\intercal x$$

Where β is estimated with the empirical risk minimization :

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} \Phi(\beta^\intercal x_i, y_i)$$

and Φ is loss function.
But $\hat{\beta}$ overfits the data so we need to adjust β. That's why, we obtain :

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} \Phi(\beta^\intercal x_i, y_i) + \lambda g(\beta)$$

With $\lambda > 0$ is an adjustment parameter and $g(\beta)$ is a regularization condition.

1.1 The LASSO method

The Lasso method uses the L_1 regularization. That means :

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} \Phi(\beta^\intercal x_i, y_i) + \lambda ||\beta||_1$$

Where $||\beta||_1 < k$ and k is the sparsity parameter.
There are 2 main issues with this method :

- Desirable feature selection property can only be achieve under relatively strong assumption.
- We need to use a large regularization parameter (λ) but it leads to sub-optimal prediction accuracy. To counter this problem, we need a complex and less robust procedure.

1.2 The OMP method

The second method that we will use in examples is the OMP method. It's a forward greedy algorithm. We use to improve an arbitrary prediction method or select relevant features. The principle is : add a feature at every step to aggressively reduce the squared error. But, like the previous method, there are issues and one of the most important is :

- It never corrects mistakes made earlier steps. So it's inadequate for feature selection because it only works when small subsets of the basis functions are near orthogonal.

A solution is the backward greedy algorithm. The principle is to train a model with all the features and greedily remove one feature at a time. To choose the feature that we remove, we take these which have the smallest increase of squared error. Yet, if the number of features is higher than the number of samples, during the first step, β overfits the data.
To counter these problems, we can use a combination of both algorithms : forward and backward. It's the principle of FoBa method (and LARS method).

1.3 The FoBa algorithm (method)

This algorithm uses the Forward and Backward greedy algorithm. The approach that we use is : we want to use backward steps to remove any errors caused by earlier forward steps and to keep a reasonable number of basis functions. And we want to take backward step adaptively and make gain made in the forward steps. Actually, we use backward step when the squared error increase is no more than half of the squared error decrease in the earlier corresponding forward step. Moreover, it has a limited number of iterations so it is computationally efficient.

Now, we can compare these two principles methods : LASSO and FoBa.
The Lasso method allows to select features under some conditions. But we need to choose an appropriate non-zero threshold. And there are 2 problems : the L_1 regularization parameter and the non-zero threshold (there are difficult to determine). Besides, we have a 'bias' with the L_1 regularization. While the FoBa method hasn't the same conditions to find the threshold parameter.
On the contrary, the FoBa greedy algorithm is closely related to the path-following algorithm for solving Lasso. The LARS algorithm (a closer algorithm of FoBa) starts with a large regularization parameter such as Lasso, then it is gradually decreased. We can remind us that LARS tracks the path through L_1 penalised least squared error by gradually decreasing the regularization parameter. But this way to find regularization parameter produces a significant bias. That is a difference between LARS and FoBa because FoBa which tracks the insertion and deletion with unbiased least squared error. It will be needed to wait a few iterations to reduce the regularization parameter and finally reduce the bias to look like the FoBa parameter. Last, Lasso and FoBa has an other similarity : mistakes made in earlier steps can be removed later. That is not possible with a forward greedy algorithm (an OMP method).

In the next part, we will compare these three methods with two examples : one with *Ionosphere* data, one with *Boston Housing* data. Data is available on websites : https://archive.ics.uci.edu/ml/machine-learning-databases/housing/[Uni] and https://archive.ics.uci.edu/ml/datasets/ionosphere[Sig].

2 Examples with comparison of three methods

2.1 First example : Boston Housing data

2.2 Second example : Ionosphere data

3 Deal with linear models in depth

We can read the article and the lesson to learn more about the Machine Learning and the Adaptive FoBa algorithm : [Zha], [Sal19]
To visualize the Python code, we can go to the Github : [Coi20]

Références

[Coi20] Ophélie Coiffier. Github lassovsfoba. https ://github.com/opheliecoiffier/LASSOvsFoBa, 2020.

[Sal19] Joseph Salmon. HMMA 308 - apprentissage statistique, 2019.

[Sig] Vince Sigillito. Ionosphere data set. https ://archive.ics.uci.edu/ml/datasets/ionosphere.

[Uni] Johns Hopkins University. Index of /ml/machine-learning-databases/housing. https ://archive.ics.uci.edu/ml/machine-learning-databases/housing/.

[Zha] Tong Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. http ://tongzhang-ml.org/papers/it11-foba.pdf.