HMMA 308 : Machine learning

Lasso vs FoBa

Ophélie Coiffier

https://github.com/opheliecoiffier/LASSOvsFoBa

Université de Montpellier

Table of Contents

Table of Contents

Introduction

To measure the quality of prediction, we use linear prediction model :

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

With

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

The Lasso method

Method uses L_1 regularization.
We have

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

2 issues :

- strong assumptions
- large regularization parameter (λ)

The OMP method

"Orthogonal Match Pursuit" : forward greedy algorithm

The OMP method

"Orthogonal Match Pursuit" : forward greedy algorithm

Principle : Add feature at every step to reduce the squared error
& Calculate its error

The OMP method

"Orthogonal Match Pursuit" : forward greedy algorithm

Principle : Add feature at every step to reduce the squared error
& Calculate its error
Ability : Select relevant features

The OMP method

"Orthogonal Match Pursuit" : forward greedy algorithm

Principle : Add feature at every step to reduce the squared error
& Calculate its error
Ability : Select relevant features
Main issue : Never correct mistakes made earlier steps

The OMP method

"Orthogonal Match Pursuit" : forward greedy algorithm

Principle : Add feature at every step to reduce the squared error
& Calculate its error
Ability : Select relevant features
Main issue : Never correct mistakes made earlier steps
Solution : Backward greedy algorithm ?

The FoBa algorithm (Lars method)

"Forward-Backward" greedy algorithm

Principle :

The FoBa algorithm (Lars method)

"Forward-Backward" greedy algorithm

Principle :
1) Use Forward greedy step

The FoBa algorithm (Lars method)

"Forward-Backward" greedy algorithm

Principle :
1) Use Forward greedy step
2) Until squared error increase is no more than half of the squared error decrease in the earlier forward step

The FoBa algorithm (Lars method)

"Forward-Backward" greedy algorithm

Principle :
1) Use Forward greedy step
2) Until squared error increase is no more than half of the squared error decrease in the earlier forward step
3) Use Backward greedy step & adaptive Backward step

The FoBa algorithm (Lars method)

"Forward-Backward" greedy algorithm

Principle :
1) Use Forward greedy step
2) Until squared error increase is no more than half of the squared error decrease in the earlier forward step
3) Use Backward greedy step $\&$ adaptive Backward step
*) **Adaptive backward functions :**

- Make sure that we progress

Comparison between Lasso and FoBa

Similarity	Difference
Path-following algorithm	Conditions to find threshold
Large regularization parameter	Bias
Mistakes in earlier step	

Table of Contents

Introduction

Loss function used :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

Principle :

- Randomly partition into 50 training points and *nb* test points
- Predict variables
- Compare MSE for each methods and for each groups

***Boston Housing* data**

Data : 506 census tracts from 1970 census
14 features

Y : housing price

Principle : 50 training points / 456 test points
We repeat 50 times MSE calculations and compare MSE obtained
With a sparsity between 1 and 10.

Boston Housing data

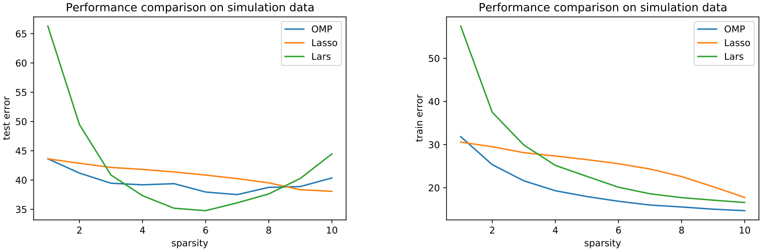


Figure: MSE depending on sparsity for the 3 methods and for the 2 groups.

Ionosphere data

Data : 351 data points
35 features

Y : binary value (Y$\in \{0, 1\}$)

Principle : 50 training points / 301 test points
We repeat 50 times MSE calculations and compare MSE obtained
With a sparsity between 1 and 10.

Ionosphere data

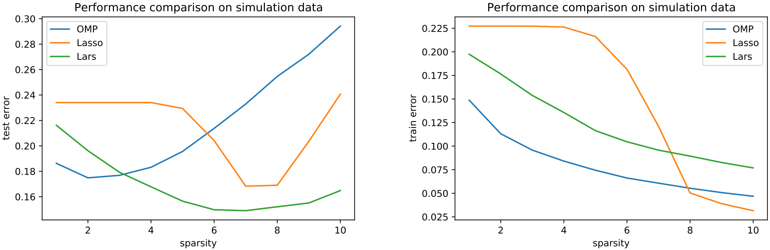
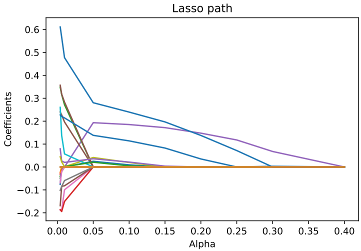


Figure: MSE depending on sparsity for the 3 methods and for the 2 groups.

Issues

- Representation of Lasso



alpha	Nb non-zero coefs
0.400	1
0.300	2
0.250	2
0.200	4
0.150	4
0.100	4
0.050	8
0.010	13
0.006	15
0.004	18

Figure: Lasso path and non-zero coefficients for Ionosphere data

- Differences between our algorithms and these in the article

Table of Contents

Conclusion

The main question is about the learning sparse representations using greedy algorithms.

FoBa is useful with training points.

We need to use a novel combination of the forward-greedy and backward-greedy algorithms to obtain a smaller error.

But, it depends on the data : with Ionosphere data (test points), with the L1 regularization, the "bias" is an advantageous.

Finally, we must choose the best algorithm and the best method depending on data, sparsity and mathematics conditions that we can check.

Bibliography

- (BSD License), scikit-learn developers. *scikit-learn*.
  https://scikit-learn.org/stable/modules/classes.html/module-sklearn.linear_model.
  2020.
- Coiffier, Ophélie. *Github LASSOvsFoBa*.
  https://github.com/opheliecoiffier/LASSOvsFoBa. 2020.
- Salmon, Joseph. *HMMA 308 - Apprentissage statistique*. 2019.
- Sigillito, Vince. *Ionosphere Data Set*.
  https://archive.ics.uci.edu/ml/datasets/ionosphere.
- University, Johns Hopkins. *Index of /ml/machine-learning-databases/housing*.
  https://archive.ics.uci.edu/ml/machine-learning-databases/housing/.
- Zhang, Tong. *Adaptive Forward-Backward Greedy Algorithm for Learning Sparse Representations*.
  http://tongzhang-ml.org/papers/it11-foba.pdf.