Journal of Statistical Software
MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

c-lasso - a package for constrained sparse and robust regression and classification in Python

Leo Simpson
TUM

Patrick L. Combettes
NC State

Christian L. Müller
LMU

Abstract

This article illustrates c-lasso, a Python package that enables sparse and robust linear regression and classification with linear equality constraints. The package handles several estimators for inferring location and scale, including the constrained Lasso, the constrained scaled Lasso, and sparse Huber M-estimation with linear equality constraints. Several algorithmic strategies, including path and proximal splitting algorithms, are implemented to solve the underlying convex optimization problems. We also include two model selection strategies for determining the sparsity of the model parameters: k-fold cross-validation and stability selection. This package is intended to fill the gap between popular python tools such as **scikit-learn** which *cannot* solve sparse constrained problems and general-purpose optimization solvers such as **cvx** that do not scale well for the considered problems or are inaccurate. We show several use cases of the package, including an application of sparse log-contrast regression tasks for compositional microbiome data. We also highlight the seamless integration of the solver into R via the **reticulate** package.

Keywords: constrained lasso, log-contrast model, sparse classification, Python.

1. Introduction

1.1. Forward model

The forward model is assumed to be:

$$y = X\beta + \sigma\epsilon \qquad \text{s.t.} \qquad C\beta = 0 \tag{1}$$

Here, X and y are given outcome and predictor data. The vector y can be continuous (for regression) or binary (for classification). C is a general constraint matrix. The vector β comprises the unknown coefficients and σ an unknown scale ().

1.2. Formulations

Depending on the prior on the solution sβ, σ and on the noise ϵ, the previous forward model can lead to different types of estimation problems.

Our package can solve six of those : four regression-type and two classification-type formulations.

[R1] Standard constrained Lasso regression

$$\arg\min_{\beta\in\mathbb{R}^d} ||X\beta - y||^2 + \lambda \left\|\beta\right\|_1 \qquad \text{s.t.} \qquad C\beta = 0$$

This is the standard Lasso problem with linear equality constraints on the β vector. The objective function combines Least-Squares for model fitting with l1 penalty for sparsity.

[R2] Contrained sparse Huber regression

$$\arg\min_{\beta\in\mathbb{R}^d} h_\rho(X\beta - y) + \lambda \left\|\beta\right\|_1 \qquad \text{s.t.} \qquad C\beta = 0$$

This regression problem uses the Huber loss as objective function for robust model fitting with l1 and linear equality constraints on the β vector. The parameter $\rho = 1.345$.

[R3] Contrained scaled Lasso regression

$$\arg\min_{\beta\in\mathbb{R}^d, \sigma>0} \left(\left\| \frac{X\beta - y}{\sigma} \right\|^2 + \frac{n}{2} \right) \sigma + \lambda \left\|\beta\right\|_1 \qquad \text{s.t.} \qquad C\beta = 0$$

This formulation is similar to [R1] but allows for joint estimation of the (constrained) β vector and the standard deviation σ in a concomitant fashion (see **?** or **?** for further info). This is the default problem formulation in c-lasso.

[R4] Contrained sparse Huber regression with concomitant scale estimation

$$\arg \min_{\beta \in \mathbb{R}^d, \sigma > 0} \left(h_\rho \left(\frac{X\beta - y}{\sigma} \right) + n \right) \sigma + \lambda \left\| \beta \right\|_1 \qquad \text{s.t.} \qquad C\beta = 0$$

This formulation combines [R2] and [R3] to allow robust joint estimation of the (constrained) β vector and the scale σ in a concomitant fashion (see **?** or **?** for further info).

[C1] Contrained sparse classification with Square Hinge loss

$$\arg \min_{\beta \in \mathbb{R}^d} L(y^T X\beta - y) + \lambda \left\| \beta \right\|_1 \qquad \text{s.t.} \qquad C\beta = 0$$

where $L\left((r_1, ..., r_n)^T\right) := \sum_{i=1}^n l(r_i)$ and l is defined as :

$$l(r) = \begin{cases} (1-r)^2 & \text{if } \leq r \leq 1 \\ 0 & \text{if } r \geq 1 \end{cases}$$

This formulation is similar to [R1] but adapted for classification tasks using the Square Hinge loss with (constrained) sparse β vector estimation.

[C2] Contrained sparse classification with Huberized Square Hinge loss

$$\arg \min_{\beta \in \mathbb{R}^d} L_\rho(y^T X\beta - y) + \lambda \left\| \beta \right\|_1 \qquad \text{s.t.} \qquad C\beta = 0$$

where $L_\rho\left((r_1, ..., r_n)^T\right) := \sum_{i=1}^n l_\rho(r_i)$ and l_ρ is defined as :

$$l_\rho(r) = \begin{cases} (1-r)^2 & \text{if } \rho \leq r \leq 1 \\ (1-\rho)(1+\rho-2r) & \text{if } r \leq \rho \\ 0 & \text{if } r \geq 1 \end{cases}$$

This formulation is similar to [C1] but uses the Huberized Square Hinge loss for robust classification with (constrained) sparse β vector estimation.

1.3. Model selections

2. Optimization algorithms

Alternative reference : **?**

Algorithm	R1	R2	R3	R4	C1	C2	Description
Path algorithms (Path-Alg)	✓	✓ ??	✓ ??	✓ ??	✓	✓	?
Douglas-Rachford-type splitting method (DR)	✓	✓	✓	✓	✗	✗	?
Projected primal-dual splitting method (P-PDS)	✓	✓	✗	✗	✗	✗	?
Projection-free primal-dual splitting method (PF-PDS)	✓	✓	✗	✗	✗	✗	?

Table 1: Overview of optimization schemes implemented

3. The python package

3.1. Installation

3.2. Dependencies

3.3. Defining the problem

3.4. Solving the problem

3.5. Visualisations

A. More technical details

A.1. Adaption of Path-Alg for R2

A.2. Adaptation of Path-Alg for R3

A.3. Adaptation of Path-Alg for R4

Affiliation:

Christian L. Müller
Department of Statistics
Ludwig-Maximilians-Universität München
Ludwigstrasse
München, Germany
E-mail: christian.mueller@stat.uni-muenchen.de
URL:

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd