c-lasso

Community Analysis

Connor Horn

Lasso Regression

In class we have seen regression of the form (with the 2 norm)

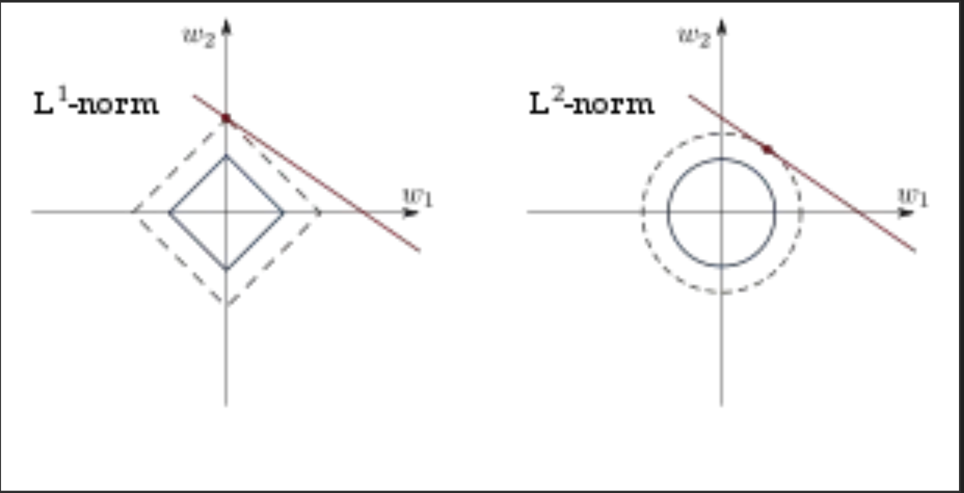
$$argmin_x \frac{1}{2}||Ax - b||^2$$

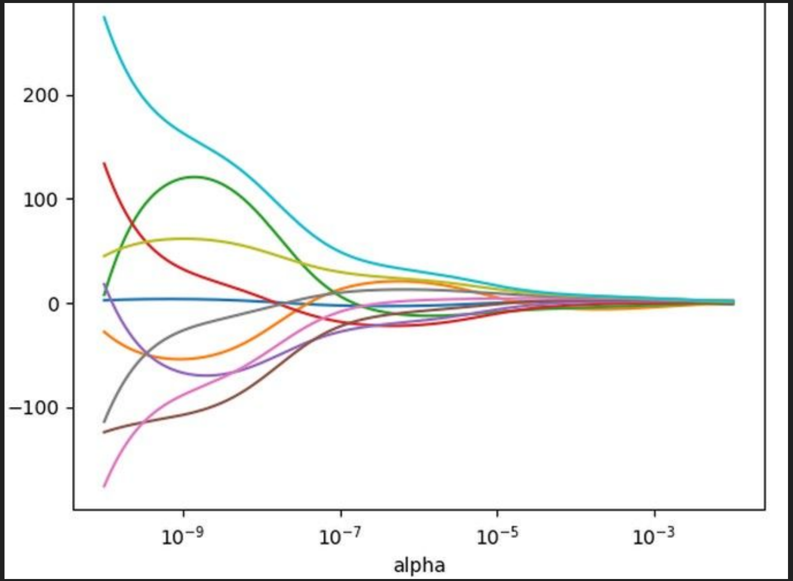
Looking at the 1 norm of this problem and adding a *regularization* term with a hyperparameter λ to minimize coefficients is known as *lasso regression.*

$$argmin_x(\frac{1}{2}||Ax - b||_1^2 + \lambda||x||_1)$$

(Regularization with the two-norm is known as ridge regression)

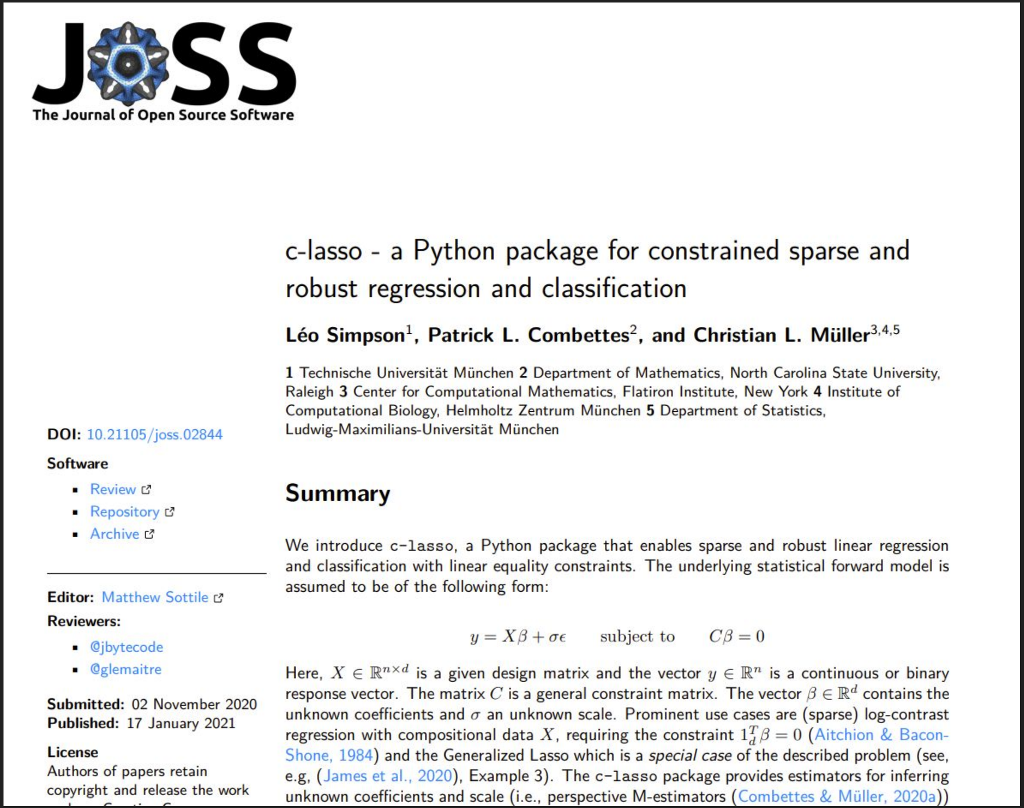
Why use L1 norm over L2?





Using L1 as opposed to L2 can be used as a form of dimension reduction!

c-lasso: a Python implementation of lasso regression


The Journal of Open Source Software

DOI: 10.21105/joss.02844

Software

- Review
- Repository
- Archive

Editor: Matthew Sottile

Reviewers:

- @jbytecode
- @glemaitre

Submitted: 02 November 2020
Published: 17 January 2021

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

Léo Simpson[1], **Patrick L. Combettes**[2], **and Christian L. Müller**[3,4,5]

1 Technische Universität München **2** Department of Mathematics, North Carolina State University, Raleigh **3** Center for Computational Mathematics, Flatiron Institute, New York **4** Institute of Computational Biology, Helmholtz Zentrum München **5** Department of Statistics, Ludwig-Maximilians-Universität München

Summary

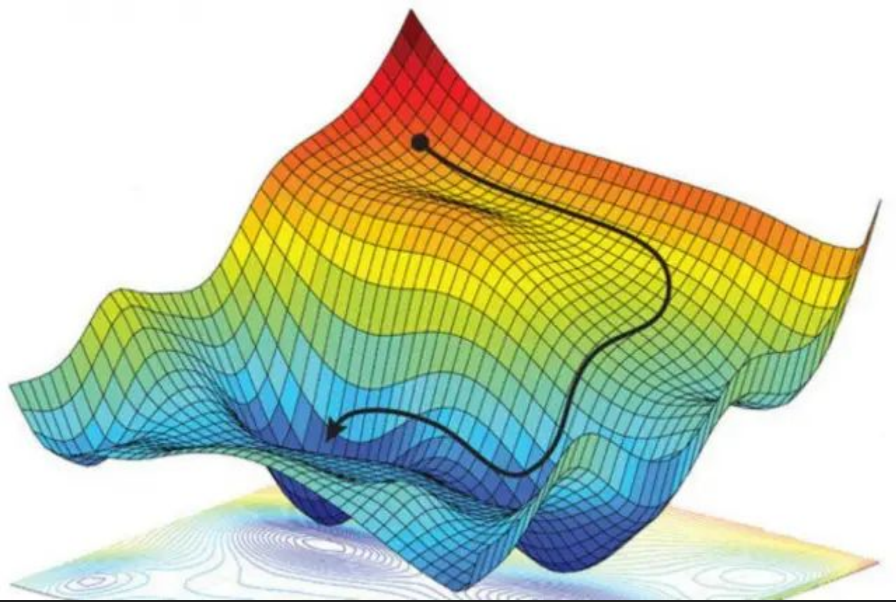
We introduce `c-lasso`, a Python package that enables sparse and robust linear regression and classification with linear equality constraints. The underlying statistical forward model is assumed to be of the following form:

$$y = X\beta + \sigma\epsilon \qquad \text{subject to} \qquad C\beta = 0$$

Here, $X \in \mathbb{R}^{n\times d}$ is a given design matrix and the vector $y \in \mathbb{R}^n$ is a continuous or binary response vector. The matrix C is a general constraint matrix. The vector $\beta \in \mathbb{R}^d$ contains the unknown coefficients and σ an unknown scale. Prominent use cases are (sparse) log-contrast regression with compositional data X, requiring the constraint $1_d^T\beta = 0$ (Aitchison & Bacon-Shone, 1984) and the Generalized Lasso which is a *special case* of the described problem (see, e.g, (James et al., 2020), Example 3). The `c-lasso` package provides estimators for inferring unknown coefficients and scale (i.e., perspective M-estimators (Combettes & Müller, 2020a))

- Made by only 3 people in 2020
  - Leo Simpson, Patrick Combettes, Christian Muller
- Not much of a community
- Only 3 total contributions (none in the last year)
- Fairly small commit size (20kb)
- Both automated test suite and continuous integration

cvxopt: Python software for Convex Optimization



- Package with much larger community than c-lasso
- More active with commits and maintaining a community
- Contains lasso regression, but also other types of convex optimization solutions

CVXOPT

PYTHON SOFTWARE FOR CONVEX OPTIMIZATION

One Question About the Software?

For c-lasso, the default constraint matrix is defined as $Cx = 0$, how can this be extended to include problems where $Cx \neq 0$? Would it be as simple as adding a constant column to C? Would that potentially cause problems elsewhere?

For cvxopt, it has the ability to have the same implementation as c-lasso, but there is no well-defined documentation on how easily this can be done? How are the actual problems set up to be solved for lasso regression?

Possible Project Ideas

- Comparison between the two packages? (computation time on the same problem)
- cvxopt has 45 open issues (none look too good for beginners / feasible in a few weeks)
- c-lasso has no open issues
- Both c-lasso and cvxopt have robust documentation
- c-lasso has 6 examples
- cvxopt has ~30 examples