Combining l^1 Penalization with Higher Moment Feasible Sets in Regression Models: a STAT8053 Project

Austin David Brown

November 28, 2018

1 Introduction

In some sense we want a stable solution So we can control the hieight and widths by using l norms This is one way to enforce a stable solution
I want a "stable" solution If the solution sucks at predicting, we can interpret this as the solution is not stable.
TODO
I asked a question during class as to what happens if you add higher moments to elasticnet. This project seeks to explore that question.

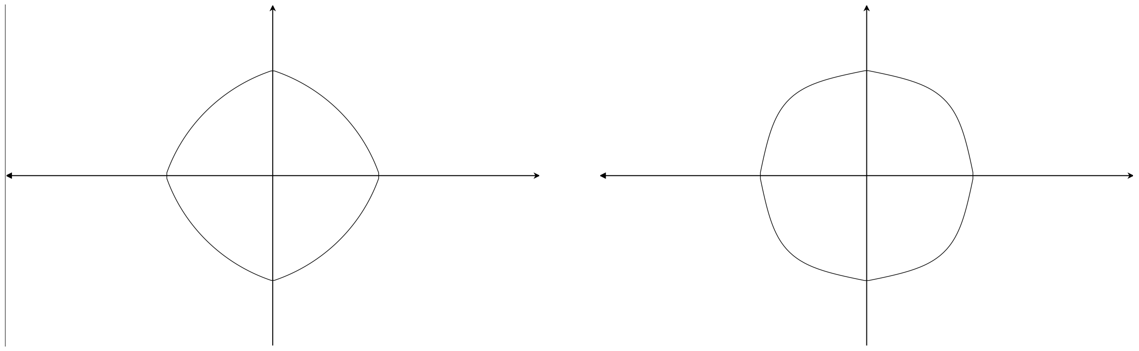

Figure 1: Elasticnet on the left and 4th moment with l1 on the right

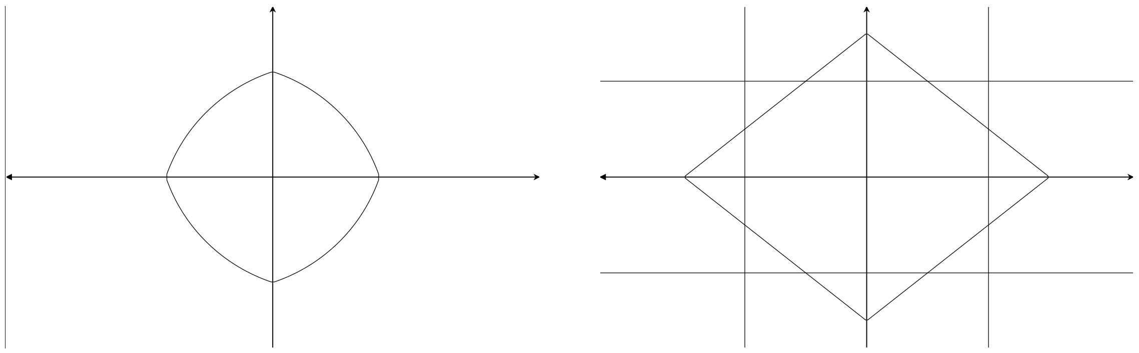

Figure 2: Elasticnet on the left and l1 and infinity norm on the right

2 Implementation

TODO currently library is implemented assuming user knows what they are doing so it will crash. This can easily be updated but it saves time.
TODO talk about step sizes
TODO talk about convergence guarantee
The package can be found here [1]. The goal was to make an package usable by scientists and researchers. For speed, $C++$ was used along with the Eigen library [4] for matrix computations. This is analogous to using Fortran with LAPACK. To interact with R, 2 interfacing layers need to be created: an R to C interface and an R script that the user calls functions from. The benefit is that any other language can then be interfaced with the library. The following diagram illustrates the idea.

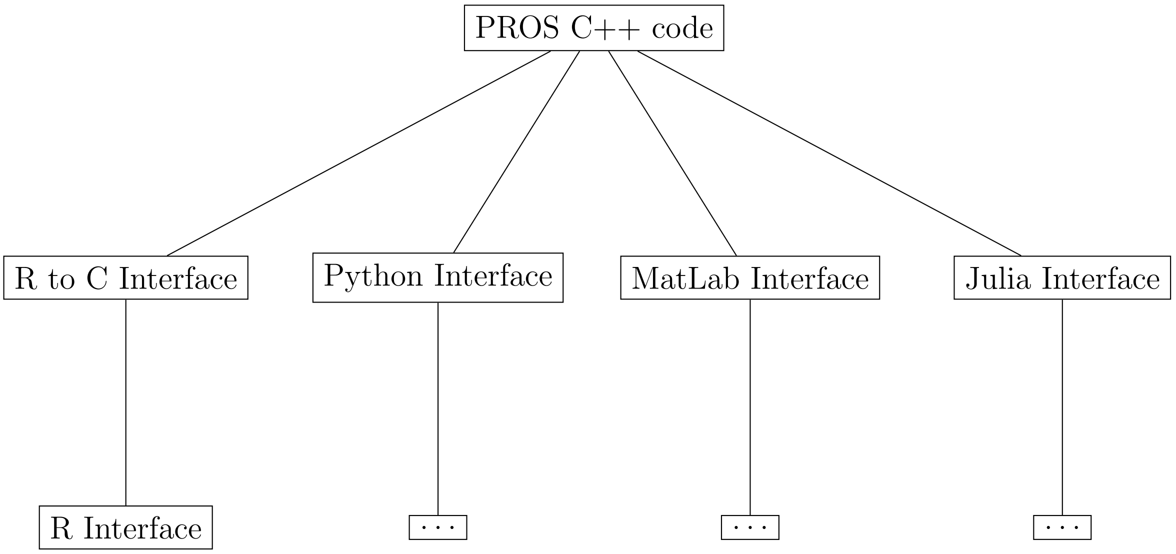

Figure 3: TODO

In the glmnet paper [5], they use coordinate descent. With the elasticnet [6] penalty, this results in a closed form coordiante-wise minimization soluton.
The l1 penalty is not differentiable, but it is separable meaning that it can be decomposed into a sum.
First, I wrote the entire program using Subgradient coordinate descent using guidelines from [3]. The algorithm is

$$L(\beta) = \frac{1}{2} \|y - X\beta\|_2^2 + \lambda P(\beta)$$

Algorithm 1: Subgradient Coordinate Method

```
Choose $\beta^0 \in \mathbb{R}^p$, Choose the tolerance $\delta > 0$;
Set $k \leftarrow 0$
repeat
    Set the step size $h^k > 0$
    Permute $I = \{1, \dots, p\}$
    for $i \in I$ do
        $\beta_i^{k+1} \leftarrow \beta_i^k - h^i g^i$ where $g^i \in (\partial L)_i$
    end
    $k \leftarrow k + 1$
until Until the loss difference $\Delta L$ is less than $\delta$;
```

The subgradient makes this algorithm difficult in general. The convergence is dependent on the step sizes, the algorithm is not a descent method. Another issue is that we cannot do line search due to the subgradient. The step sizes were chosen due to Nesterov [7] as diminishing with $\frac{R}{\sqrt{k+1}}$.
The better algorithm is proximal gradient descent. Because we constructed everything to be seperable the proximal mapping is done coordinate wise. The algorithm is

$$L(\beta) = \|y - X\beta\|_2^2 + \lambda P(\beta)$$

Algorithm 2: Proximal Gradient Coordinate Descent

```
Choose $\beta^0 \in \mathbb{R}^p$, Choose the tolerance $\delta > 0$;
Set $k \leftarrow 0$
repeat
    Set the step size $h^k > 0$
    Permute $I = \{1, \dots, p\}$
    for $i \in I$ do
        $\beta_i^{k+1} \leftarrow (\mathbf{prox}_{h^k L})_i(\beta_i^k - h^k \langle X_i, y - X\beta \rangle)$
    end
    $k \leftarrow k + 1$
until Until the loss difference $\Delta L$ is less than $\delta$;
```

The step size is chosen by diminishing step size, but I should do line search. This is easy to implement.

The cross validation was implemented with the warm starting algorithm.

Algorithm 3: Warm Start Cross-Validation

```
TODO
```

References

[1] PROS. github.com/austindavidbrown/pros

[2] Neal Parikh and Stephen Boyd. 2014. Proximal Algorithms. Found. Trends Optim. 1, 3 (January 2014), 127-239. DOI=10.1561/2400000003 http://dx.doi.org/10.1561/2400000003

[3] Stephen J. Wright. 2015. Coordinate descent algorithms. Math. Program. 151, 1 (June 2015), 3-34. DOI=10.1007/s10107-015-0892-3 http://dx.doi.org/10.1007/s10107-015-0892-3

[4] Guennebaud, Gaël (2013). Eigen: A C++ linear algebra library (PDF). Eurographics/CGLibs.

[5] Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33(1), 1-22. URL http://www.jstatsoft.org/v33/i01/.

[6] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B, 67, 301–320.

[7] Yurii Nesterov. 2014. Introductory Lectures on Convex Optimization: A Basic Course (1 ed.). Springer Publishing Company, Incorporated.