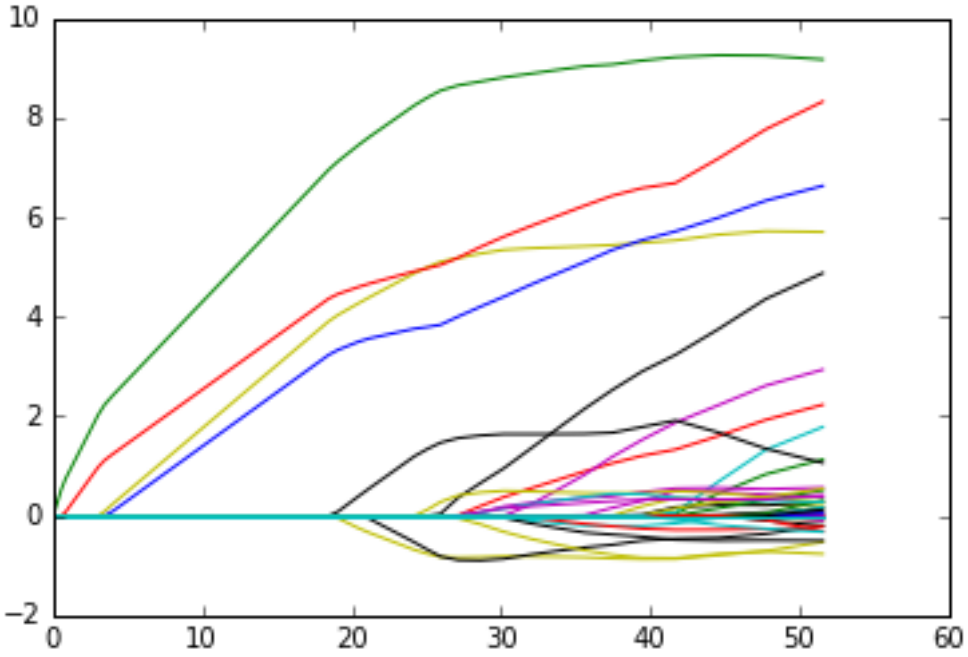
Anurag Pande
UID: 604749647

Lasso Regression

We plot the lasso solution path starting from a large value of lambda, gradually decreasing it. For the plotted points, Lambda was given an initial value of 1000, and decreased all the way to 10, in steps of 10. For every value of lambda, we implement co-ordinate descent via the for for-loop with j going from 1 to p. The single beta values are then stored column-wise for plotting.

As seen in the plots, the lasso solutions for sets of beta from large lambdas to small lambdas, we see large, convex curves going to more streamlined curves. Lasso regression prefers a set of sparse values of beta.

Python Plot



Python code to reproduce plot:

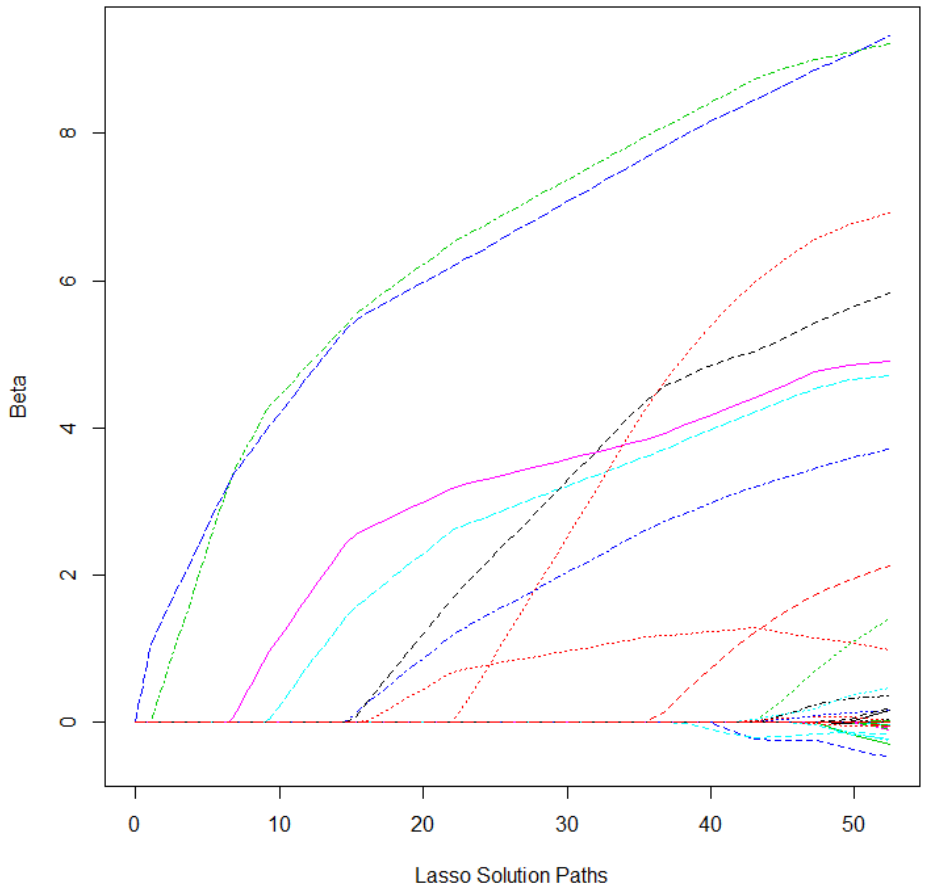
```
import matplotlib.pyplot as plt

u = np.transpose(np.dot(np.ones((1,p)),abs(beta_all)))
v = np.transpose(beta_all)
plt.figure()
plt.plot(u, v,  label='Spline Regression')
```

Input to python:

```
X = np.random.standard_normal((n,p))
beta_true = np.zeros(p)
s=10
beta_true[0:s] = range(1, s+1)
Y = np.add(np.dot(X,beta_true),np.random.standard_normal(n,))
lambda_all = np.arange(1000,0,-10)
myLasso(X,Y,lambda_all)
```

R Plot



R code to reproduce plot:

```
matplot(t(matrix(rep(1, p), nrow = 1)%*%abs(beta_all)), t(beta_all), xlab = "Lasso Solution Paths", ylab = "Beta", type = 'l')
```

Input to R:

```
n=50
p=200
s=10
X = matrix(rnorm(n*p), nrow=n)
beta_true = np.zeros(p)
beta_true[0:s] = range(1, s+1)
Y = X %*% beta_true + rnorm(n)
lambda_all = (100:1)*10
myLasso(X,Y,lambda_all)
```