medical_imaging_reconstruction

September 2, 2021

Medical imaging reconstruction In this problem, you will consider an example resembles medical imaging reconstruction in MRI. We begin with a true image image of dimension 50 x 50 (i.e., there are 2500 pixels in total). Data is cs.mat; you can plot it first. This image is truly sparse, in the sense that 2084 of its pixels have a value of 0, while 416 pixels have a value of 1. You can think of this image as a toy version of an MRI image that we are interested in collecting.

Because of the nature of the machine that collects the MRI image, it takes a long time to measure each pixel value individually, but it's faster to measure a linear combination of pixel values. We measure n = 1300 linear combinations, with the weights in the linear combination being random, in fact, independently distributed as N(0; 1). Because the machine is not perfect, we don't get to observe this directly, but we observe a noisy version. These measurements are given by the entries of the vector

y = Ax + n;

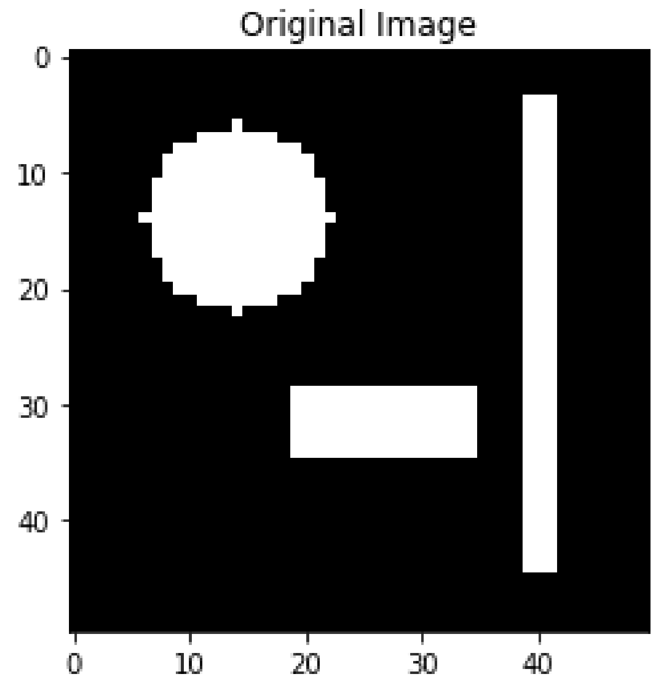
where y = R1300, A = R1300x2500, and n x N(0; 25 x I1300) where In denotes the identity matrix of size n x n. In this homework, you can generate the data y using this model.
Now the question is: can we model y as a linear combination of the columns of x to recover some coefficient vector that is close to the image? Roughly speaking, the answer is yes.

Key points here: although the number of measurements n = 1300 is smaller than the dimension p = 2500, the true image is sparse. Thus we can recover the sparse image using few measurements exploiting its structure. This is the idea behind the field of compressed sensing. The image recovery can be done using lasso

0.0.1 (a) Using lasso to recover the image and selecting lambda using 10-fold cross-validation.

Plotting the Original Image

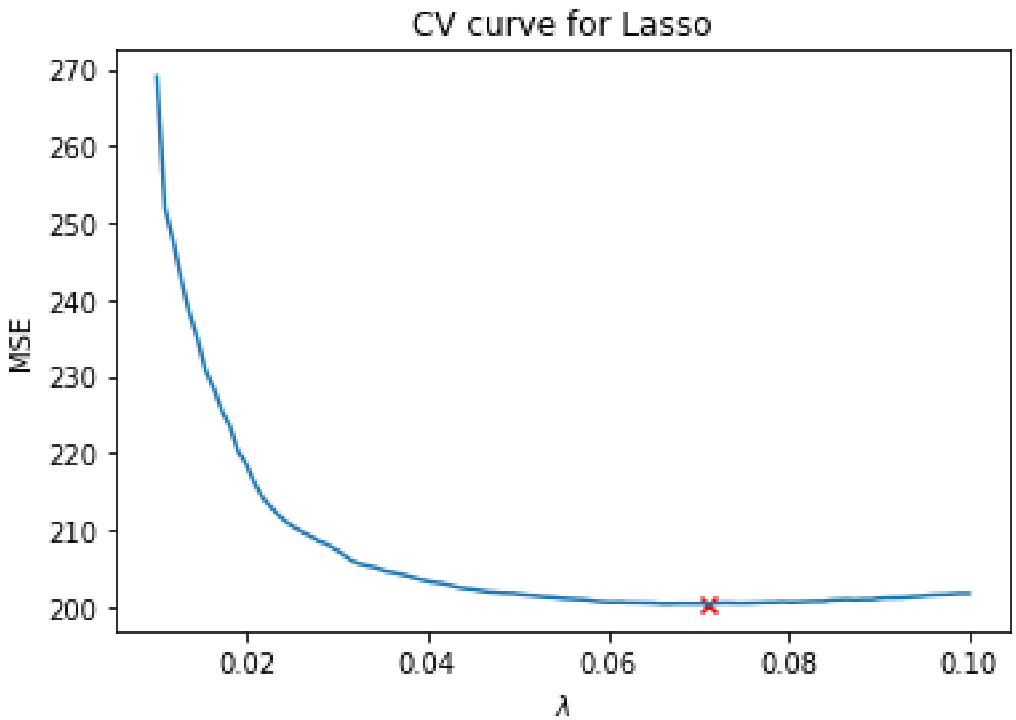
```
Text(0.5, 1.0, 'Original Image')
```



Following the steps to generate a normal gaussian matrix

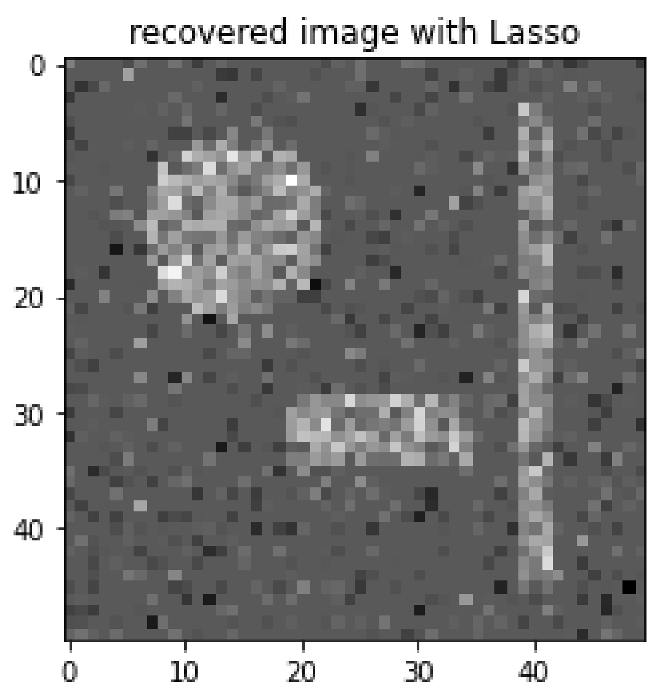
Getting the best lambda value for Lasso

Plotting the Cross-Validation curve for Lasso



Reconstructing the image with Lasso coefficients

```
Text(0.5, 1.0, 'recovered image with Lasso')
```

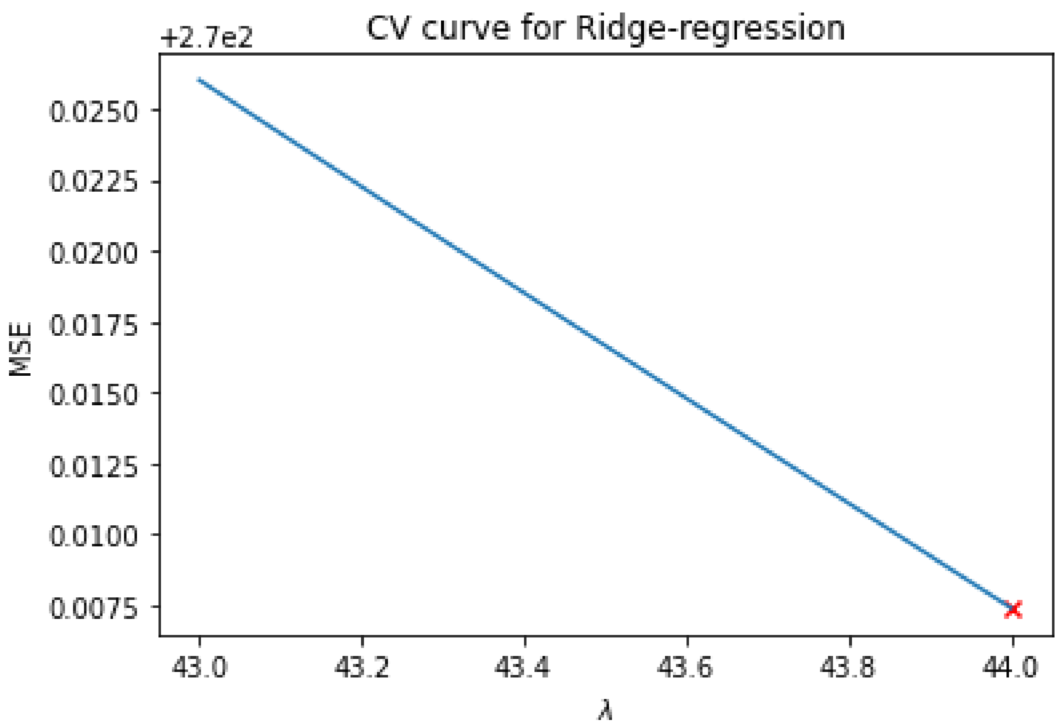


Identifying the reconstruction error

```
reconstruction error with Lasso :0.0631
```

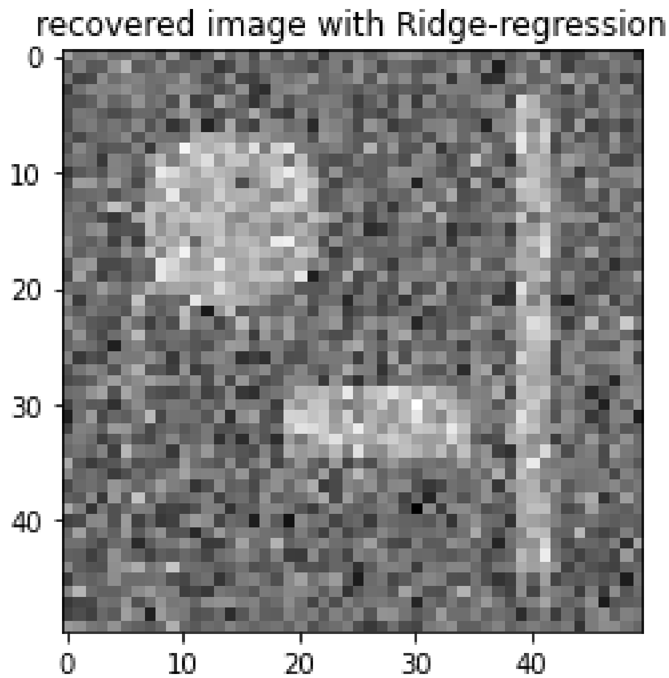
We do see that there is a decent reconstruction of the image with a low error rate of 6% after regularization.

0.0.2 (b) Using Ridge to recover the image and selecting lambda using 10-fold cross-validation.



Reconstructing image with Ridge coefficients

```
Text(0.5, 1.0, 'recovered image with Ridge-regression')
```



Reconstruction error calculation

```
reconstruction error with Ridge :0.0875
```

Ridge has a slightly higher reconstruction error rate than LASSO.