

STEM

Young

October 21, 2016

Contents

1	Linear Regression	1
1.1	Simple Linear Regression 10/15/16	1
2	Data	2
2.0.1	Convolution	2
2.0.2	Multislice	2

1 Linear Regression

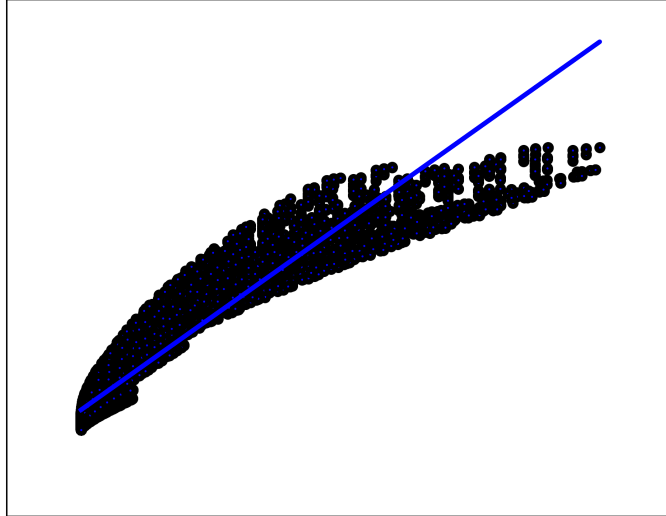
1.1 Simple Linear Regression 10/15/16

$$I.mult = \beta_1 * (I.conv) + \beta_0$$

For the initial model, I reduced the two data sets from 3D to 1D and performed linear regression with a bias term. The least squares estimates for intercept is 0 and slope is 0.0011319.

It is clear that relationship between the two image sets is not linear (see scatter plot with regression line). Also, there seems to be discrete "jumps" in the response variable that the linear model does not capture. I did not try cross-validation with the initial model.

There are many ways to improve the model. First, I would build a model that imposes a restriction on the response variable. One example would be logistic transformation which will output in the range of [0,1]. We could then rescale the output to match the range of the Multislice image intensities.



Secondly, I would develop a feature set that has more explanatory power. One natural way to start would be to include the dimensional indices as features to account for spatial correlation. However, this could be complicated as there seems to be clusters of high intensity pixels in the center and corners of the image. Either a kernel method or K-means method might be useful to account for spatially-clustered pixel intensities. Alternatively, we could perform segmentation on the different regions.

Third, some form of piecewise model (i.e. Decision Trees) or locally-weighted models might better account for "jumps".

- Python files: `olr.py`

2 Data

2.0.1 Convolution

- 50 by 36 by 20
- intensities = $[0, 139.7768]$

2.0.2 Multislice

- 50 by 36 by 20

- intensities = $[0, 0.1641]$