STEM

Young

November 4, 2016

0.1 Multiple Regression 10/30/16

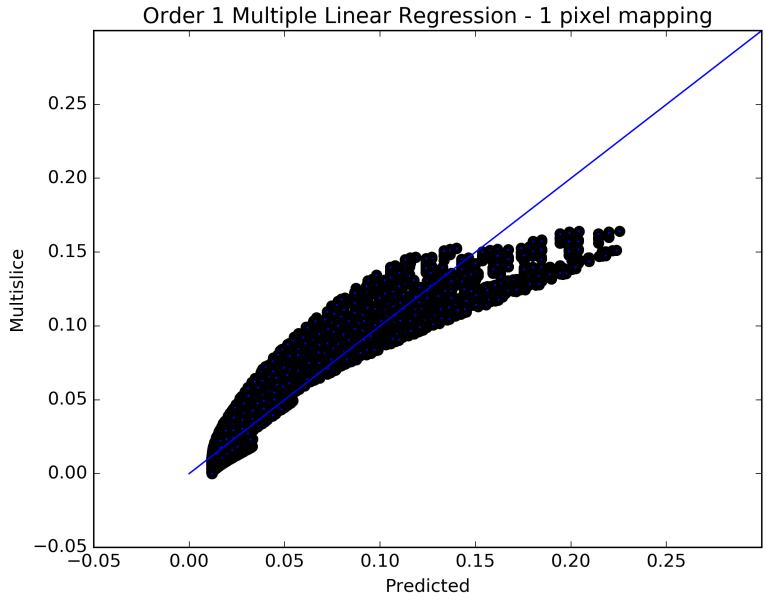
For each layer, I generated a .png file with side by side images of convolution, multislice, and their difference (after normalization). These files are in data/images/ directory. It looks like multislice images tend to be smoother than convolution images.

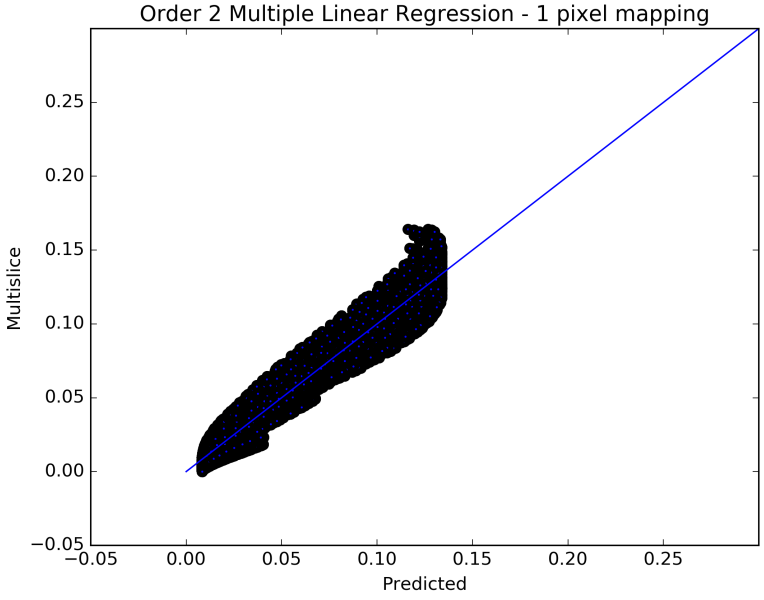
I fit various multiple linear regression models using

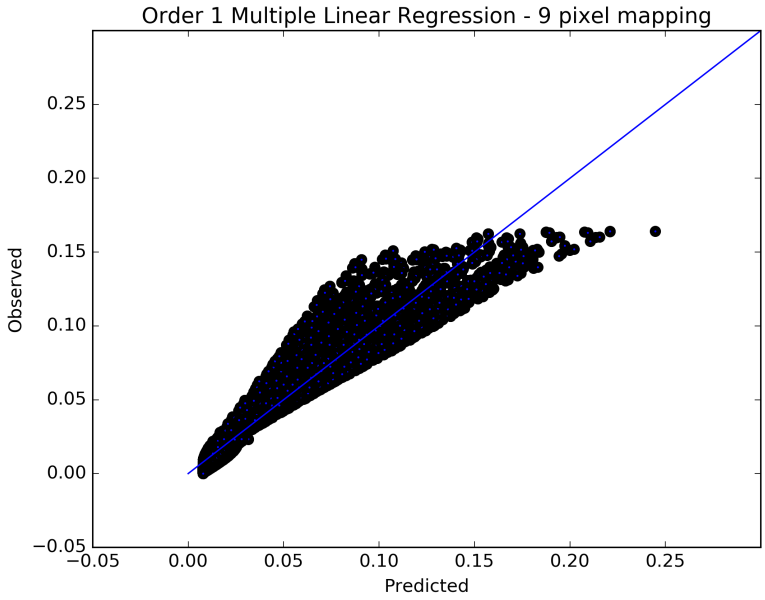
- 1 pixel or 9 pixels (pixel and 8 neighbors)
- gradient direction/magnitude
- linear and quadratic predictors

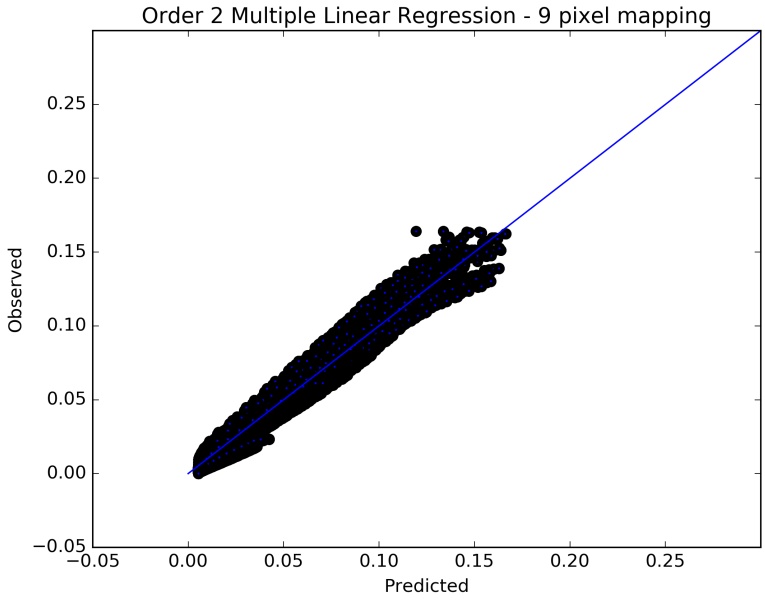
The graphs below show the predicted outcome vs. observed (multiscale) intensity as a measure of model fit. It's clear that adding the quadratic term improves the model fit significantly. Gradient information doesn't seem to improve the model beyond the quadratic terms of the neighbor intensities.

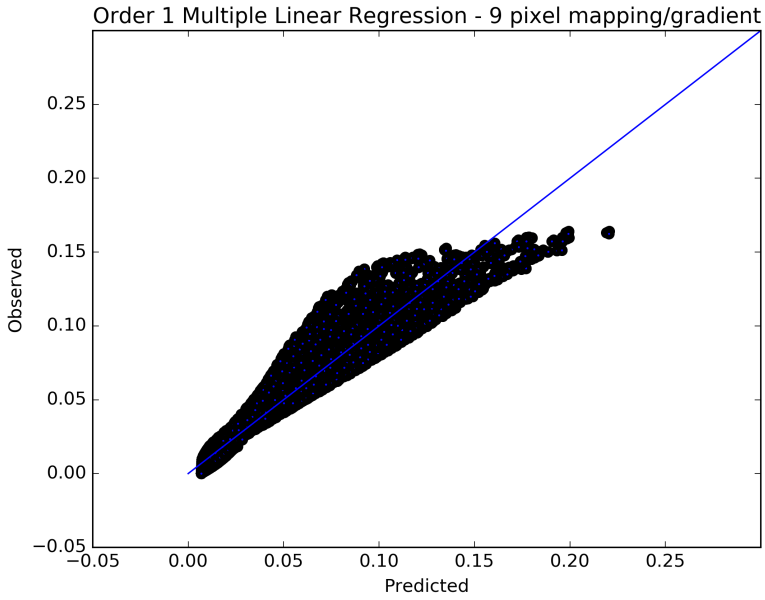
I plan to perform cross-validation to compare these models. I also would like to experiment with applying smoothing filters to the convolution image.

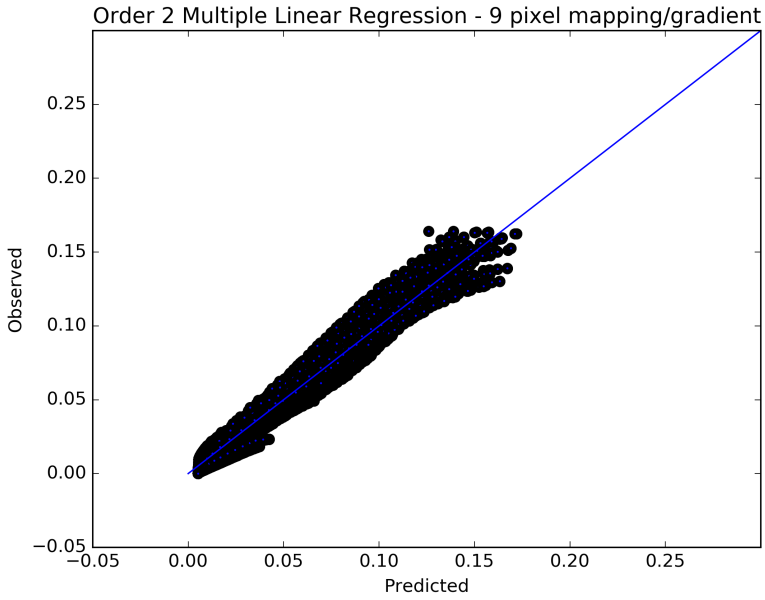

Order 1 Multiple Linear Regression - 1 pixel mapping
Multislice
Predicted
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.05
0.00
0.05
0.10
0.15
0.20
0.25


Order 2 Multiple Linear Regression - 1 pixel mapping
Multislice
Predicted
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.05
0.00
0.05
0.10
0.15
0.20
0.25


Order 1 Multiple Linear Regression - 9 pixel mapping
Observed
Predicted
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.05
0.00
0.05
0.10
0.15
0.20
0.25


Order 2 Multiple Linear Regression - 9 pixel mapping
Observed
Predicted
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.05
0.00
0.05
0.10
0.15
0.20
0.25





- Python files: olr2.py

1 Data

1.0.1 Convolution

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

1.0.2 Multislice

- 50 by 36 by 20
- intensities = [0, 0.1641]