Homework 6
CSCI4100

Han Hai
Rin:661534083
haih2@rpi.edu

October 14th 2018

1. **[200 points Exercise 3.4]** Consider a noisy target y=${w^*}^T x + \epsilon$ for generating the data, where ϵ is a noise term with zero mean and σ^2 variance, independently generated for every example(x,y). The expected error of the best possible linear fit to this target is thus σ^2.
For the data $D = \{(x1, y1), ..., (xN, yN)\}$ denote the noise in y_n as ϵ_n and let $\epsilon = [\epsilon_1, \epsilon_2, ..., \epsilon_N]^T$, assume that $X^T X$ is invertible. By following the steps below, show that the expected in-sample error of linear regression with respect to D is given by
$E_D[E_{in}(w_{lin})] = \sigma^2(1 - \frac{d+1}{N})$.

   (a) Show that the in-sample estimate of y is given by $\hat{y} = Xw^* + H\epsilon$.
   $\hat{y} = Xw_{lin}$
   $= X(X^T X)^{-1} X^T y$
   $= X(X^T X)^{-1} X^T ({w^*}^T x + \epsilon)$
   $= Xw^* + H\epsilon$

   (b) Show that the in-sample error vector $\hat{y} - y$ can be expressed by matrix times ϵ What is the matrix?
   $\hat{y} - y = Xw^* + H\epsilon - ({w^*}^T x + \epsilon)$
   $= (H - I)\epsilon$
   the matrix is (H-I)

   (c) Express $E_{in}(w_{lin})$ in terms of ϵ using b, and simplify the expression using Exercise 3.3c.
   $E_{in} = \frac{1}{N}||\hat{y} - y||^2$

   $= \frac{1}{N}||(H - I)\epsilon||^2$

   $= \frac{1}{N}(H - I)\epsilon^T (H - I)\epsilon$

   $= \frac{1}{N}\epsilon^T (I - H)\epsilon$
   using 3.3 c

   (d) Prove the statement using c and the Independence of epsilons
   $= E_D[\frac{1}{N}\epsilon^T (I - H)\epsilon]$

   $= \frac{1}{N} E_D trace(\epsilon^T (I - H)\epsilon)$

   $= \frac{1}{N}(ED(\sum_{i=1}^N \epsilon_i^2) - ED(\sum_{i=1}^N \epsilon_i^2 H_i)$

   $=\frac{1}{N}(N\sigma^2 - trace(H)\sigma^2)$

$= \frac{1}{N}N\sigma^2 - (d+1)\sigma^2$ from 3.3d

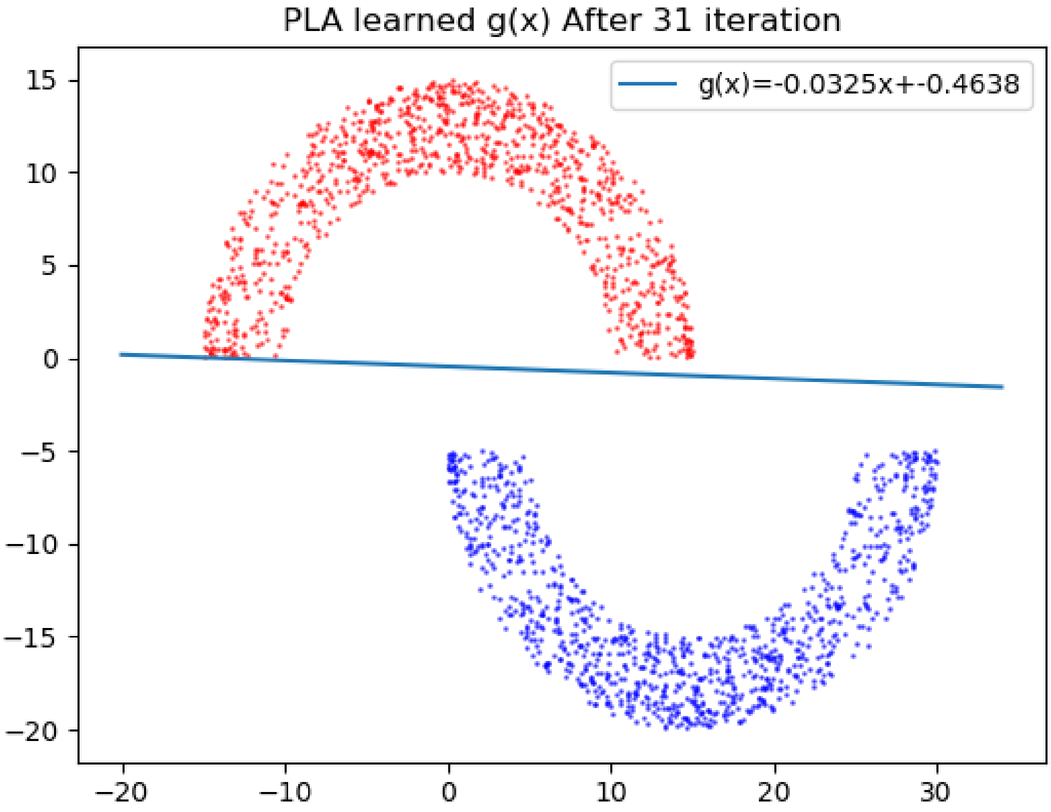
$= \sigma^2(1 - \frac{d+1}{N})$ For the expected out-of-sample error, we take a special case which is easy to analyze. Consider a test data set D_{test}... which shares the same input vectors x_n with D but with a different realization of the noise terms. Denote the noise in y'_n as ϵ'_n and let $\epsilon' = [\epsilon'_1.....\epsilon'_n]$. Define $E_{test}(w_{lin})$ to be the average squared error on D_{test}

(e) Prove that $E_{D,\epsilon'}[E_{test}(w_{lin})] = \sigma^2(1 + \frac{d+1}{N})$ $y' = Xw^* + \epsilon'$
from a
$\hat{y} = Xwlin = Xw^* + H\epsilon$
$E_{test}(w_{lin}) = \frac{1}{N}||\hat{y} - y'||^2$

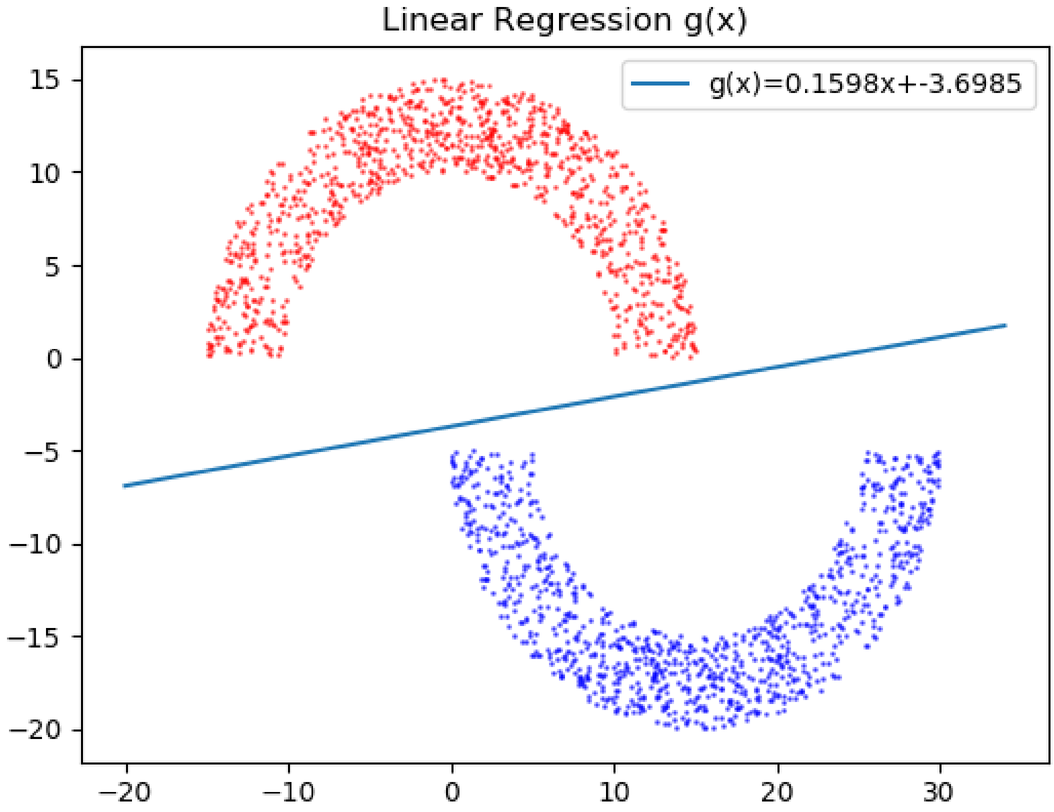
$= \frac{1}{N}||Xw^* + H\epsilon - (Xw^* + \epsilon')||^2$
$= \frac{1}{N}(\epsilon^T H\epsilon - 2\epsilon'^T H\epsilon + \epsilon'^T\epsilon')$
and $E_D(\epsilon^T\epsilon) = N\sigma^2$
$E_D(\epsilon^T H\epsilon)) = (d+1)\sigma^2$
So $E_{D,\epsilon'}[E_{in}(W_{lin})] = \sigma^2(1 + \frac{d+1}{N}) - \frac{2}{N}E_{D,\epsilon'}(\epsilon'^T H\epsilon)$,
the later part is zero because it's equal to the trace of inner $E(\epsilon) = E(\epsilon') = 0$
thus we proved the statement

2. **[200 POINTS] Problem 3.1**

(a) I generated 2000 data in python, and ran the PLA on the data, the below graph plot the learned g(x) of the alogirthm along with data

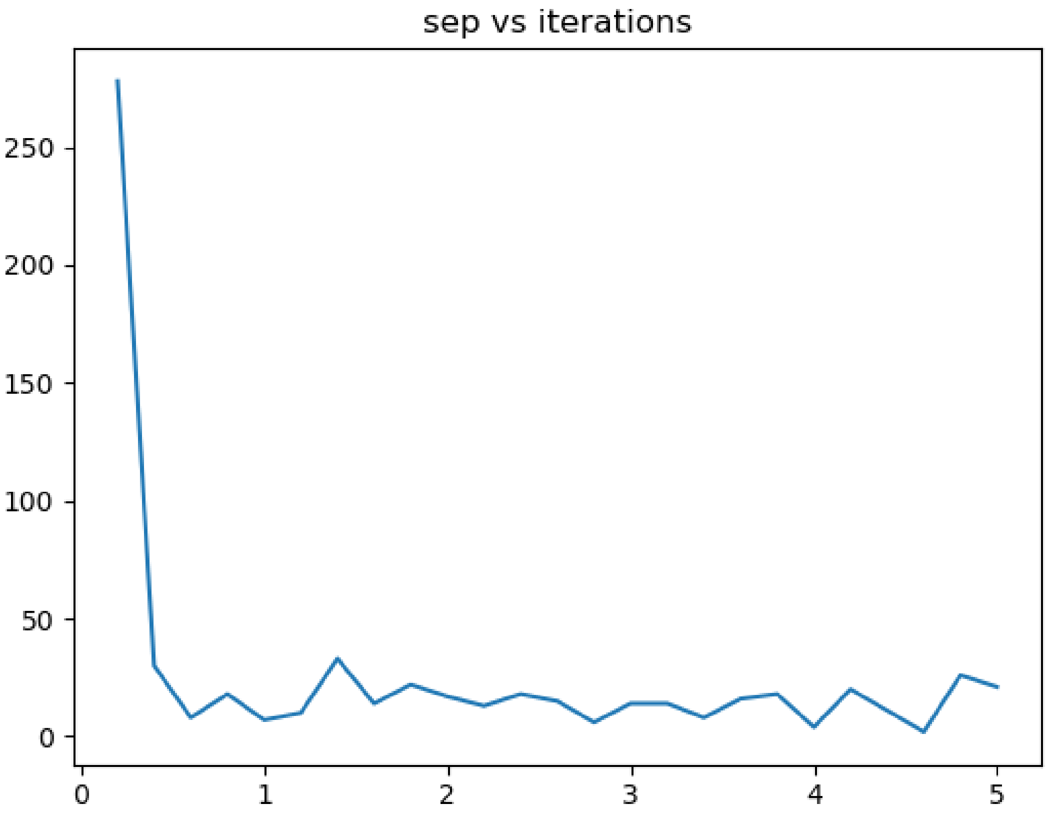


(b) I implemented linear regression algorithm using python numpy linalg module, the result graph along with data is shown below



As we can see, linear regression algorithm is more centered between two classes comparing to PLA, while PLA lean towards one set/class of data. That is because PLA stops updating as long as it finds the edge of one class, while linear regression is found analytically through math by the purpose of finding the center.

3. [**200 points**] **Problem 3.2** For the double-semi-circle task in 3.1, vary sep in the range$\{0.2, 0.4, ..., 5\}$. Generate 2000 examples and run the PLA starting with w=0, Record the number of iterations PLA takes to converge.
Here is the full data of sep vs iteration, spe:0.20 iteration:278 spe:0.40 iteration:30 spe:0.60 iteration:8 spe:0.80 iteration:18 spe:1.00 iteration:7 spe:1.20 iteration:10 spe:1.40 iteration:33 spe:1.60 iteration:14 spe:1.80 iteration:22 spe:2.00 iteration:17 spe:2.20 iteration:13 spe:2.40 iteration:18 spe:2.60 iteration:15 spe:2.80 iteration:6 spe:3.00 iteration:14 spe:3.20 iteration:14 spe:3.40 iteration:8 spe:3.60 iteration:16 spe:3.80 iteration:18 spe:4.00 iteration:4 spe:4.20 iteration:20 spe:4.40 iteration:11 spe:4.60 iteration:2 spe:4.80 iteration:26 spe:5.00 iteration:21
and the plot below

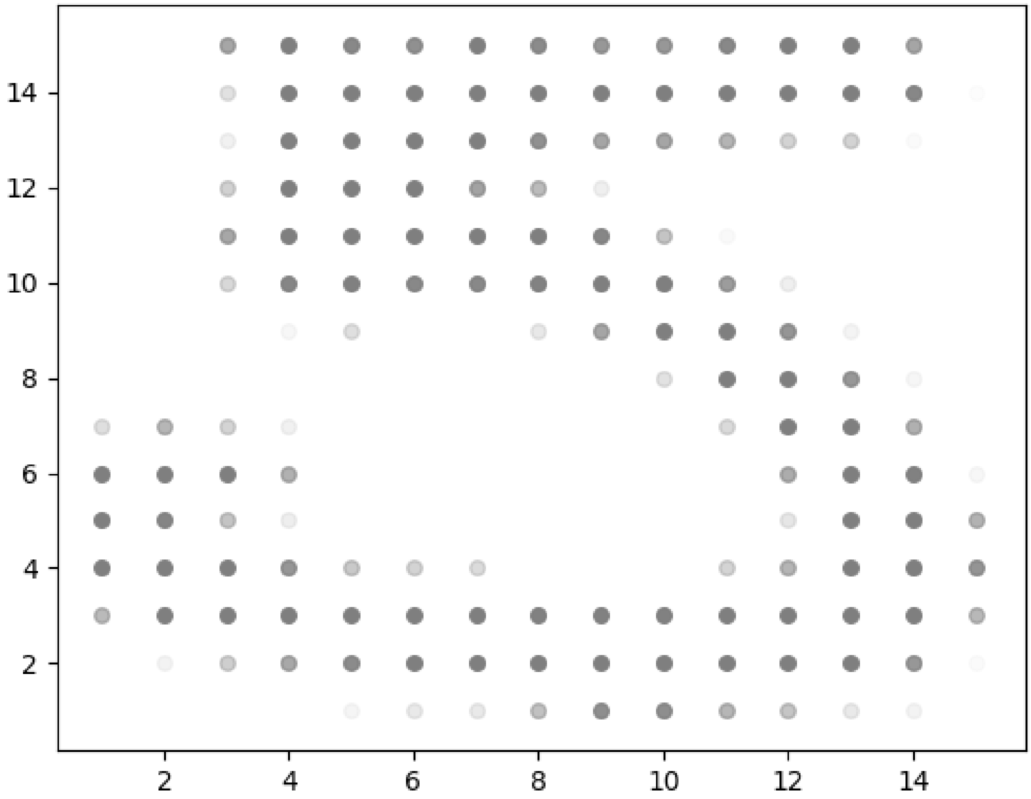


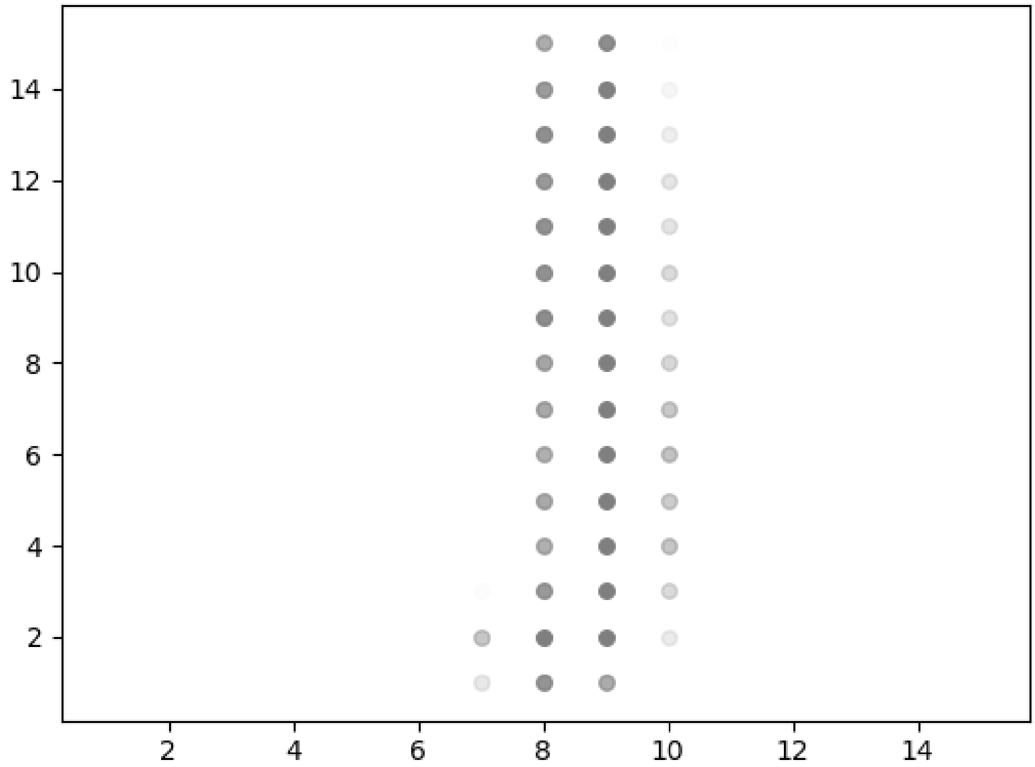
As we can see , there is an over all downward trend between separation of two semi circles and the number of iteration PLA it takes. It makes sense intuitively because as two semi-circles are closer, it takes longer for PLA to updates till the data are separated correctly. It also makes sense analytically, from problem 1.3 we concluded that $t \leq \frac{R^2||w^*||^2}{p^2}$. In this question, as distance sep get larger, the minimum distance get larger, therefore$\frac{||w^*||^2}{p^2}$ get smaller, and while R remains the same, t is bounded by a increasingly smaller number.

4. **[200 points] Problem 3.8** For linear regression, the out of sample error is $E_{out}(h) = E[(h(x) - y)^2]$. Show that among all hypotheses, the one that minimizes E_{out} is given by $h^*(x) = E[y|x]$ The function h^* can be treated as deterministic target function, in which case we can write $y = h^*(x) + \epsilon(x)$ where $\epsilon(x)$ is an input dependent noise variable. Show that $\epsilon(x)$ has expected value zero.
$E_{out}(h) = E([h(x) - y)^2]$
$= E_{out}(h) = E([h(x) + E[y|x] - E[y|x] - y)^2]$
$= E[(h(x) - h^*(x))^2] + E[(h^*(x) - y)^2] + 2E[(h(x) - h*(x))(h^*(x) - y)]$
Since $E[E(y|x)] = E(y)$, the last term $=E[h(x) - h*(x))E[(h*(x) - y)|x]]$
then $E[(h^*(x) - y)|x] = h*(x) - h*(x) = 0$ the rightmost term is zero, therefore the whole expression is zero when $h(x) = h*(x)$ therefore, $E_{out}(h) = E[h(x) - h^*(x)]^2 + E[(h^*(x) - y)^2] \geq E[(h^*(x) - y)^2]$ thus $h^*(x)$ is the one that minimize Eout
then $E(\epsilon(x)) = E[E(y|x) - E(h*(x)|x)] = h*(x) - h*(x) = 0$

5. **[200 points] Question 6** Handwritten digits Data -Obtainging Features

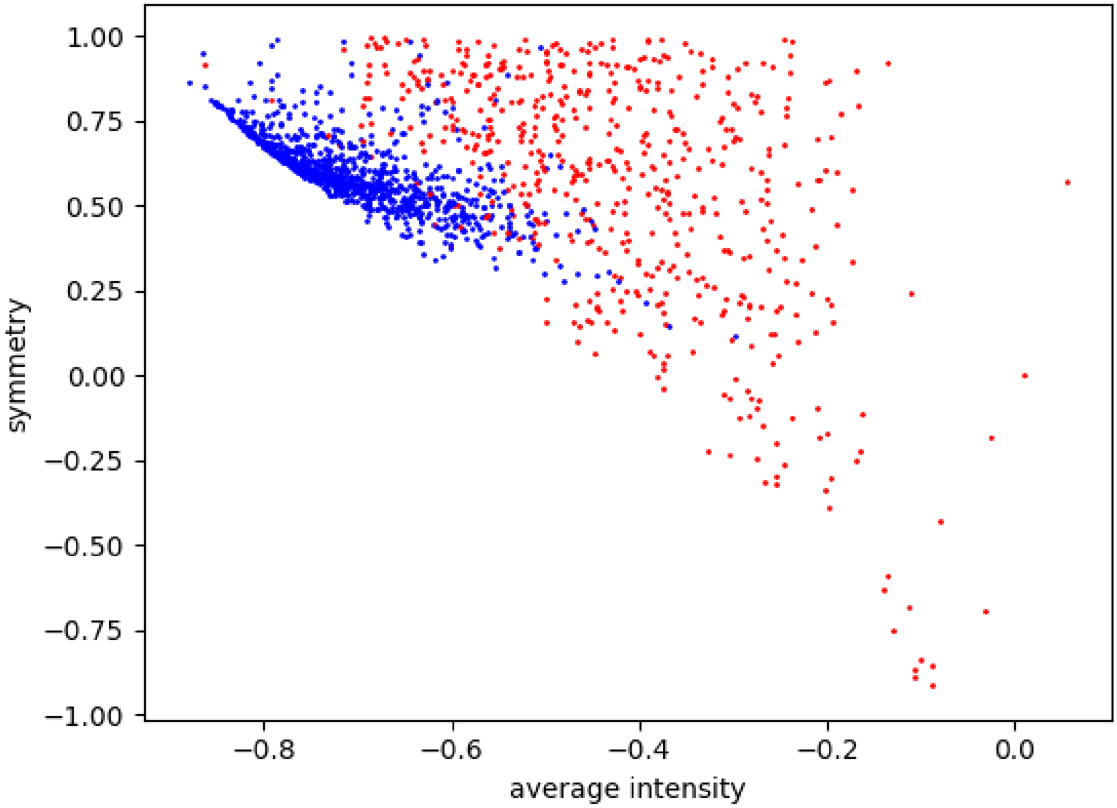
(a) Familiarize yourself with the data by giving a plot of two of the digit images.
I used matplotlib to plot the below images, it's probably a 1 and a 5





(b) Develop two features to measure properties of the image that would be useful in distinguishing between 1 and 5. You may use symmetry and average intensity. Give the mathematical definition of your two features.
Fist,define the length of a image as l here l=16 and grayscale value of a pixel as gr
I will use average intensity which I define as
$AI = \frac{1}{l^2} \sum_{i,j=0}^{i,j=l} gr_{ij}$
the sum of the grayscale value of all pixels divided by the number of pixels.
and symmetry:
$SI = \frac{AI_{i<=0.5l}}{AI_{i>0.5l}}$ if $AI_{i<0.5l} \leq AI_{i>0.5l}$
$\frac{AI_{i>=0.5l}}{AI_{i<0.5l}}$ otherwise The average intensity of the left half of the image divided by average intensity of the right half, or the ohter way around if right half is more intense. The closer to one the more symmetry.

(c) As in the text, give a 2-D scatter plot of your features: for each data example, plot the two features with a red if it is a 5 and a blue if it is a 1.
The below graph is a graph of two features I chose in part b on each axis and the scatter plot of data. We can see the data can be roughly seperated by a line with some error.


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
symmetry
−0.8
−0.6
−0.4
−0.2
0.0
average intensity