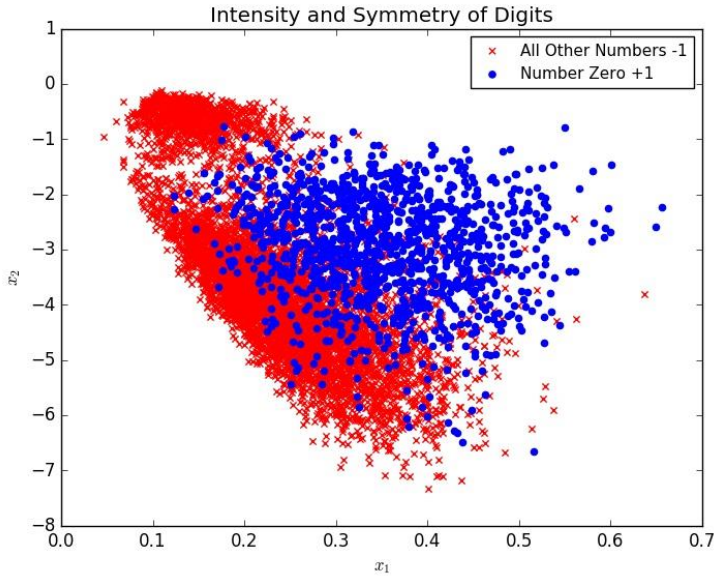
Midterm

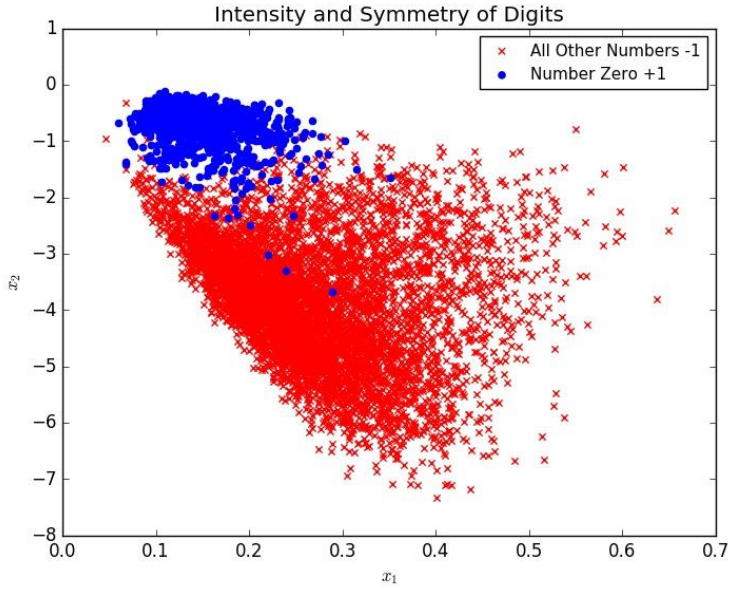
John Randis

10/12/2016

1. The original data set (classification of zero and nonzero):



The modified data set (classification of one and every other number):



a) Pocket Algorithm – The error returned when using the pocket algorithm was 0.040597997531202856. The algorithm was stopped after 200 iterations.

b) Linear Regression – After running the pocket algorithm it gave us a *w* of [0.54592846 -0.96745805 0.30085587]. Starting from this *w* and using linear regression, the error given is 0.020847620353860924.
c) Pocket Algorithm – Starting from the solution given by linear regression rather than 0, and once again stopped at 200 iterations, the error we end up with is 0.013989850500617199. We can conclude that using linear regression to find the best starting *w*, and then using the pocket algorithm is the best method in terms of finding the smallest $E_{out.}$

2.

```
import numpy as np
import math

class complexity:
    def findN(delta, N, dvc):
        out = 8/(delta*delta)
        ins = ((math.pow((2*N), dvc) + 1) * 4)/delta
        log = math.log(ins)
        newN = out * log
        print("N >= ", newN)


def main():
    c = complexity
    c.findN(.1, 1000, 3)

main()
```

output: N >= 21193.269466292146