Luke Jaffe
CS6140 Machine Learning
12/19/15

HW7 – Local Methods and Clustering

Problem 1. k-Nearest Neighbors

A) Spambase
python p1a.py

k=1: 0.930585683297
k=2: 0.908893709328
k=3: 0.9284164859

B) Digits
python p1b.py

Cosine:
k=1: 0.905
k=3: 0.905
k=7: 0.904

Gaussian:
k=1: 0.91
k=3: 0.912
k=7: 0.905

Polynomial:
k=1: 0.575
k=3: 0.588
k=7: 0.627

Problem 2. Neighbors in a range/window

A)

i. Spam + Euclidean
Usage: python p2a.py -d s

r=4.600000: 0.82863340564

ii. Digits + Cosine
Usage: python p2a.py -d d

r=0.325000: 0.881

B)

i. Spam + Gaussian (sigma=1.0)
Usage: python p2b.py -d s

Accuracy: 0.921908893709

ii. Digits + Gaussian (sigma=1.0)
Usage: python p2b.py -d d

Accuracy: 0.909

iii. Digits + Poly (degrees=2)
Usage: python p2b.py -d d

Accuracy: 0.682

Problem 3. Dual Perceptron with kernels

A)

i. primal
Usage: python p3a.py p
Iteration 1, total_mistake 523
Iteration 2, total_mistake 477
Iteration 3, total_mistake 513
...
Iteration 77, total_mistake 1
Iteration 78, total_mistake 1
Iteration 79, total_mistake 0

Classifier weights: [-1.08599 0.17555273 0.4362371 0.65881996 0.89480011]

Normalized with threshold: [-1. 0.16165225 0.40169532 0.6066538 0.82394876]

ii. dual
Usage: python p3a.py d
Iteration 1, total_mistake 136
Iteration 2, total_mistake 68
Iteration 3, total_mistake 50
...
Iteration 6, total_mistake 34
Iteration 7, total_mistake 25
Iteration 8, total_mistake 0

Classifier weights: [-14. 2.52873259 5.70717051 8.52231457 11.32560723]

Normalized with threshold: [-1. 0.18062376 0.40765504 0.60873676 0.80897195]

B)

i.
Usage: python p3b.py l
(doesn't converge)

Iteration 1, total_mistake 412
Iteration 2, total_mistake 415
Iteration 3, total_mistake 406
...
Iteration 28, total_mistake 410
Iteration 29, total_mistake 406
Iteration 30, total_mistake 422

ii.
Usage: python p3b.py g

Iteration 1, total_mistake 5
Iteration 2, total_mistake 18
Iteration 3, total_mistake 13
Iteration 4, total_mistake 0

Classifier weights: [-28.9255 59.6493]

Normalized with threshold: [-1. 2.06217006]

Problem 5. Feature Selection with kNN
Usage: python p5.py

Features selected: [23 6 15 26 51]
k=1: 0.863340563991