ENEE 436 Foundations of Machine Learning - Project 2

Fall 2020

Hongyu Tu

Task 1 - SVM classifier

A. Model Selection

To find the optimal value of the kernel parameter σ, I used the 'cross_val_score' function from sklearn to basically do 5-fold validation on training data for each σ we have(2^-2 to 2^8). I saved all the score to a 3 by 8 list, and the max in each of the 3 lists will be the most optimized σ.

From calculation:

Best classifier for dataset [banana] is: SVC(sig = 2^1)

Best classifier for dataset [twonorm] is: SVC(sig = 2^-2)

Best classifier for dataset [waveform] is: SVC(sig = 2^-2)

B. Training

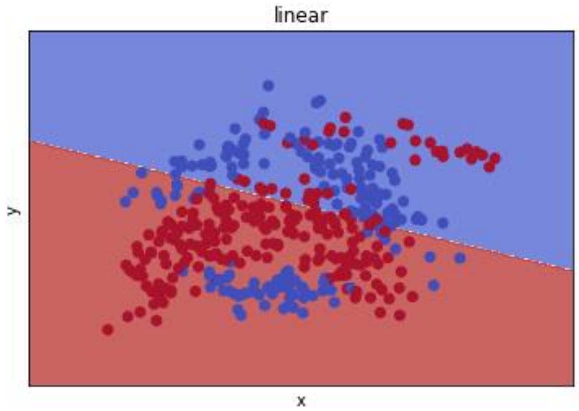
Dataset	Classifier	training accuracy	testing accuracy	Number of support vectors	
				Category 1	Category 2
banana	linear	0.695	0.60673	173	173
	RBF	0.935	0.88735	68	70
twonorm	linear	0.9875	0.96671	16	17
	RBF	1	0.95171	192	206
waveform	linear	0.93	0.86261	42	43
	RBF	1	0.8	261	129

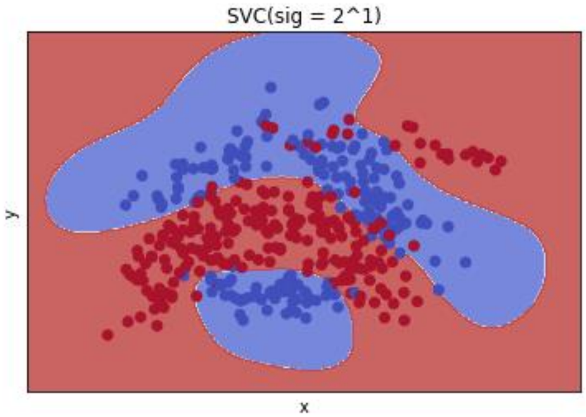
- For dataset banana, the RBF kernel performed a lot better than linear kernel in terms of accuracy, and RBF kernel has less support vectors for the two categories comparing to linear kernel.

- For dataset twonorm, the RBF kernel performed better than linear kernel just by a tiny amount in terms of training accuracy, but slightly worse in terms of testing accuracy. RBF kernel has a lot more support vectors for the two categories comparing to linear kernel.

- For dataset waveform, the RBF kernel performed better than linear kernel just by a fair amount in terms of training accuracy, but the RBF testing accuracy is worse than linear model. RBF kernel has a lot more support vectors for the two categories comparing to linear kernel.

C. Plot Banana with decision boundary

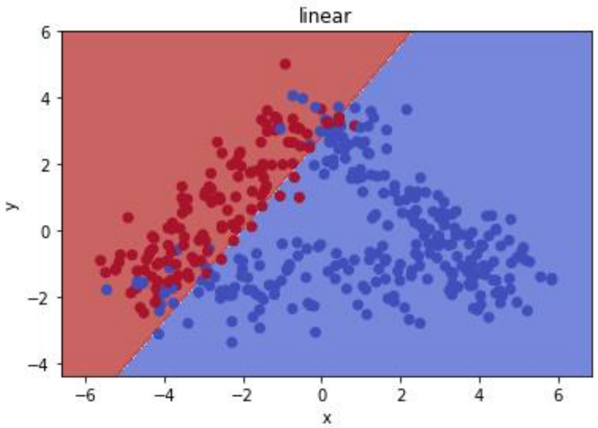


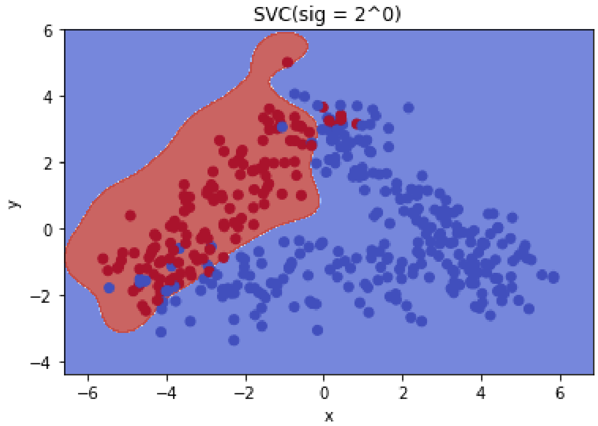


As shown above in the images, the linear kernel has a lot of mis-classification, as it can only have one straight line as boundary, whereas the RBF kernel showed its strength with multiple curvy lines of boundary, successfully classifying both the middle and the upper right to red.

D. Dimensionality Reduction on dataset 'Waveform'

Dataset	Classifier	training accuracy	testing accuracy	Number of support vectors	
				Category 1	Category 2
waveform	linear	0.93	0.86261	42	43
	RBF	1	0.8	261	129
Waveform - Reduced	linear	0.91	0.879	46	47
	RBF	0.945	0.903	83	55





After dimension reduction, for both kernel, training accuracy dropped a little, but the test accuracy increased. From the graph, it's quite obvious that most red dots are compressed to the left where most blue dots are on the right and bottom. Although there are some parts that both red and blue points exist, it's pretty separable. Here the RBF also does a better job than linear model in both train and test accuracy, with its curve-like boundary.

Task 2 - 3-layer Neural Network with Sigmoid activation

A. Model Selection

To find the number of neurons that will potentially produce the best output, I did what I did in task one and just run the k-fold algorithm to find what works best for each dataset with number of neurons as variable, range from 1 to 1000. And my result is below:

Best classifier for dataset [banana] is: SVC(#_neurons = 368)

Best classifier for dataset [twonorm] is: SVC(#_neurons = 1)

Best classifier for dataset [waveform] is: SVC(#_neurons = 14)

B. Training

Dataset	Classifier	training accuracy	testing accuracy
banana	Neural Network	0.685	0.59776
	RBF	0.935	0.88735
twonorm	Neural Network	0.995	0.96657
	RBF	1	0.95171
waveform	Neural Network	0.955	0.89804
	RBF	1	0.8

For banana, the neural network did a terrible job and for testing is almost close as a random guess.

For twonorm, the neural network had better accuracy on training and only tiny difference on testing data.

For waveform, the neural network did almost as good as the SVM but just a little bit worse.

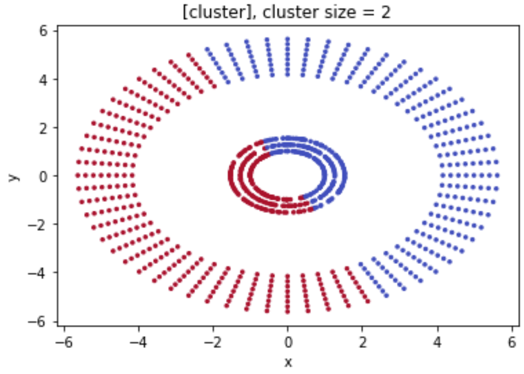
I think what's causing the problem is that just having one layer of hidden layer is not enough to do well for banana, and simply increasing the number of neurons in the same layer won't sufficient. Had there be more layers, I believe all three datasets will do a much better job.

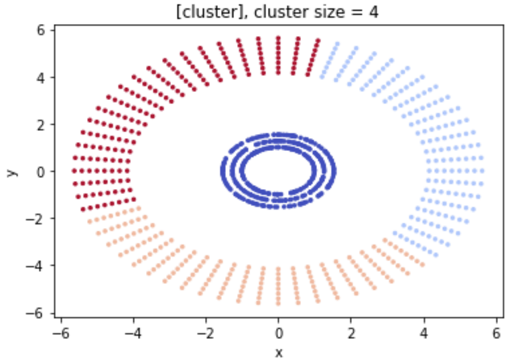
Another problem I noticed is that there are a lot more testing data than training data: for banana, the training size to testing size is 400 to 4900; for twonorm, the ratio is 400:7000 and for waveform, the ratio is 400:4600. Unlike SVM, neural networks rely a lot on the training data so that all the weights can be perfectly tuned by Backpropagation. Usually the training data to testing data ratio is about 70:30 or 75:15, but here we have 1:10. I believe this definitely another problem.

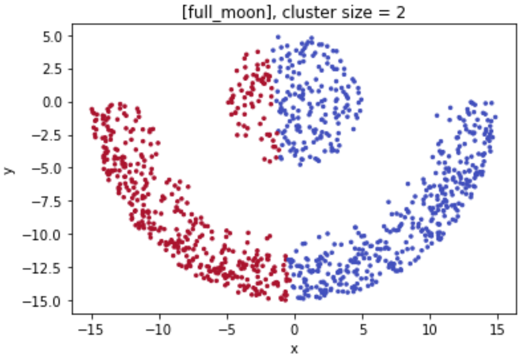
Task 3 - K-means, Spectral clustering and Gaussian mixture model

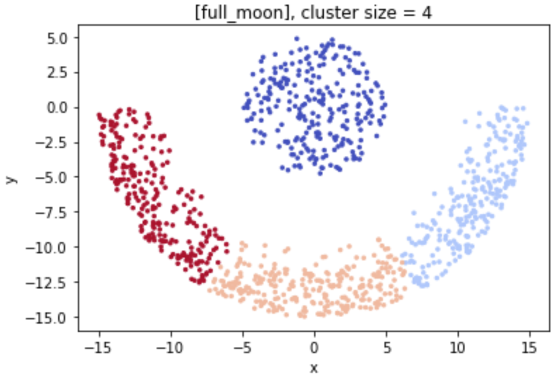
A. Two spirals, crescent and the full moon and cluster within cluster

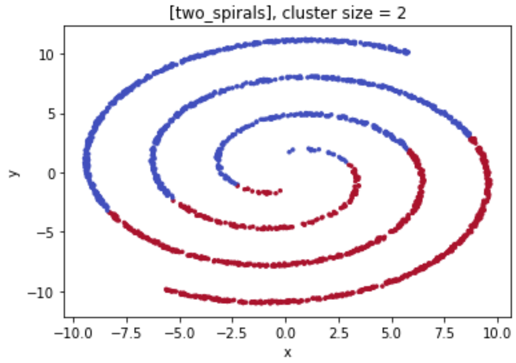
a) <u>K-means</u>

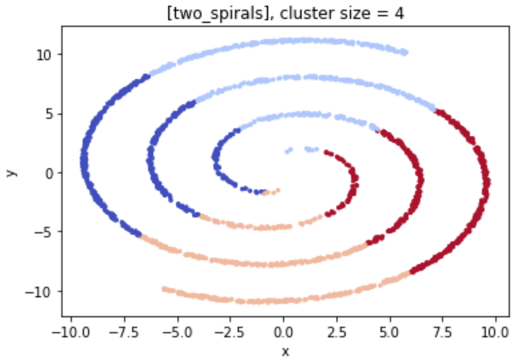






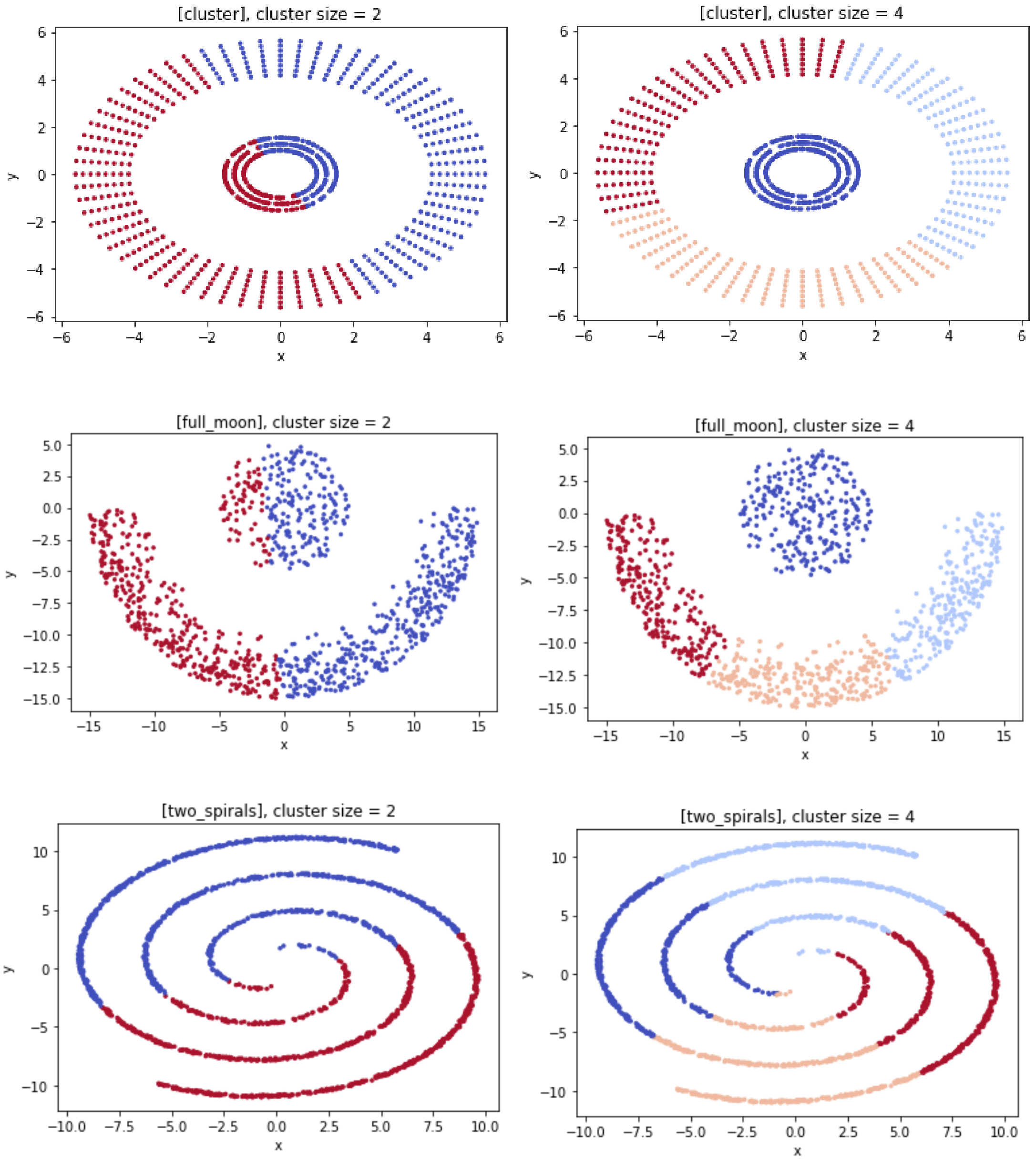






As the graph presents, k-Means is having a hard time trying to figure out the correct clustering when we do 2 clusters but when we do 4 clusters and groups them so we have 2 pairs each with 2 cluster, the correct clustering seems to be obtained.

b) Spectral clustering algorithm



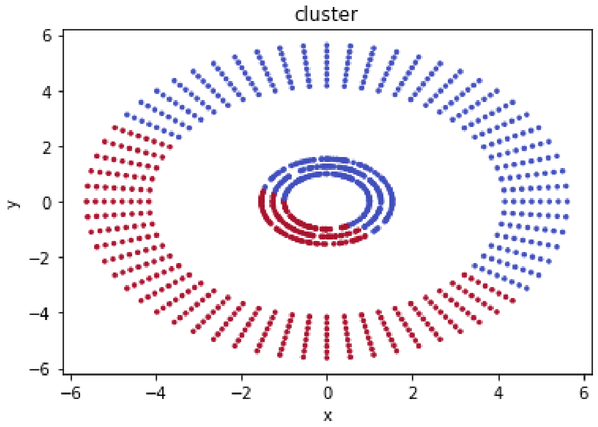
As the graph presents, k-Means is having a hard time trying to figure out the correct clustering when we do 2 clusters but when we do 4 clusters and groups them so we have 2 pairs each with 2 cluster, the correct clustering seems to be obtained.

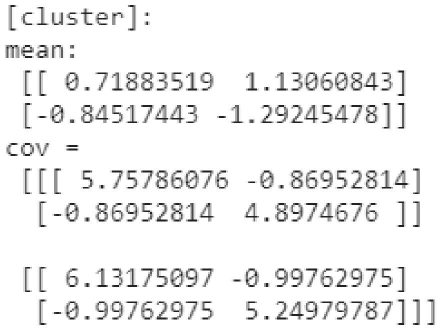
B. K-means and Spectral clustering algorithm on Waveform dataset

For K-means, the training error is 0.19250 and the test error is 0.20391.

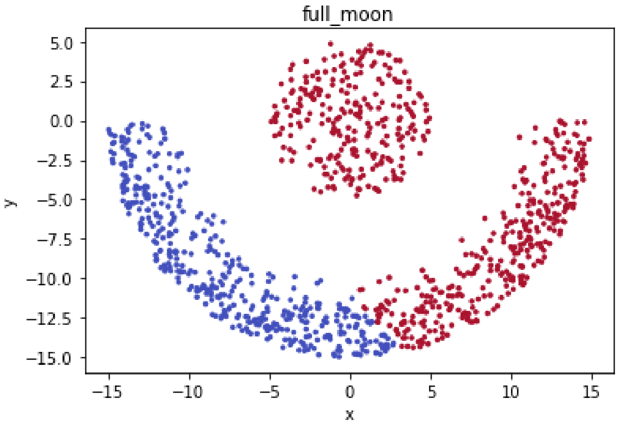
For Spectral clustering algorithm, the training error is 0.23250 and the test error is 0.19652.

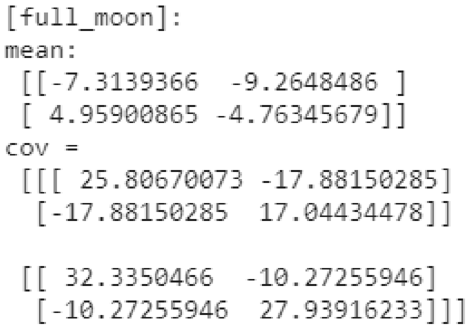
C. Gaussian mixture model algorithm



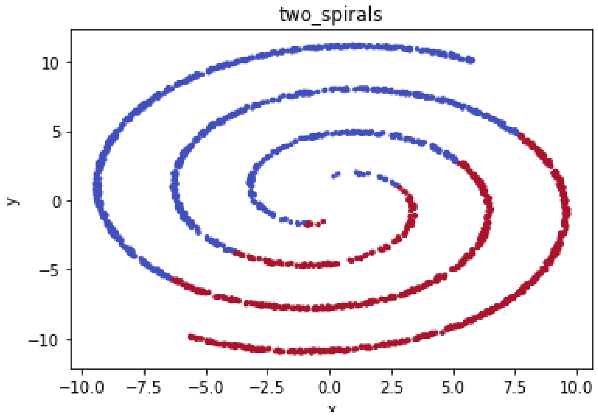


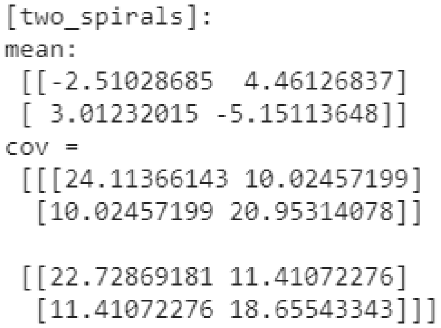
```
[cluster]:
mean:
 [[ 0.71883519  1.13060843]
 [-0.84517443 -1.29245478]]
cov =
 [[[ 5.75786076 -0.86952814]
  [-0.86952814  4.8974676 ]]

 [[ 6.13175097 -0.99762975]
  [-0.99762975  5.24979787]]]
```





```
[full_moon]:
mean:
 [[-7.3139366  -9.2648486 ]
 [ 4.95900865 -4.76345679]]
cov =
 [[[ 25.80670073 -17.88150285]
  [-17.88150285  17.04434478]]

 [[ 32.3350466  -10.27255946]
  [-10.27255946  27.93916233]]]
```





```
[two_spirals]:
mean:
 [[-2.51028685  4.46126837]
 [ 3.01232015 -5.15113648]]
cov =
 [[[24.11366143 10.02457199]
  [10.02457199 20.95314078]]

 [[22.72869181 11.41072276]
  [11.41072276 18.65543343]]]
```