Classification and Regularization

Abstract

The project includes three subtasks. The first one is kfold cross-validation linear SVM classification evaluating by a confusion matrix and displaying by ROC curve. The second one is nonlinear SVM classification with gaussian kernel function. We also check predictions of a Gaussian SVM against the user's labels. The last one performs Tikhonov Regularization to an audio with noise in order to de-noise it.

Method

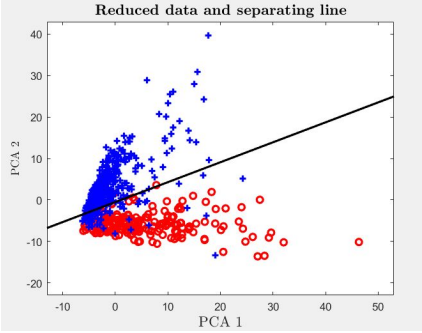
A description of how I implemented my code and tested it.
For q1, I used PCA for dimensionality reduction; performed dimensionality reduction and, as needed, standardize data; classify data using a linear SVM and interpret the results; and evaluate the performance of a binary classifier.

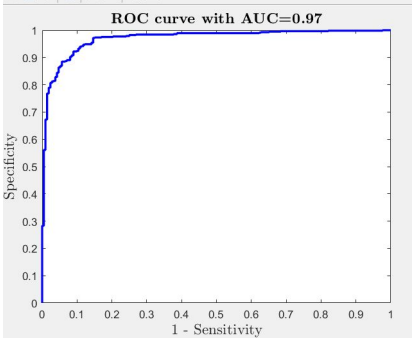
For q2, I used SVD for dimensionality reduction. Then I used gaussian SVM with kernel scalar 1 to classify each data vector. In function svmgausscheck, I computed the score of each data vector as the sum of the weighted and biased kernel responses of each support vector by score += alpha*ylabel*exp(-1*(norm(thisX - svec)^2)) and the correctness ratio.

For q3, I created the second diagonal matrix based on the provided code by changing ones to ones - 2, [0] to [1] (increase index of column by 1). To test it, I assigned 5 to n, and check whether R= R + q is a sparse bi-diagonal.

Result

Q1

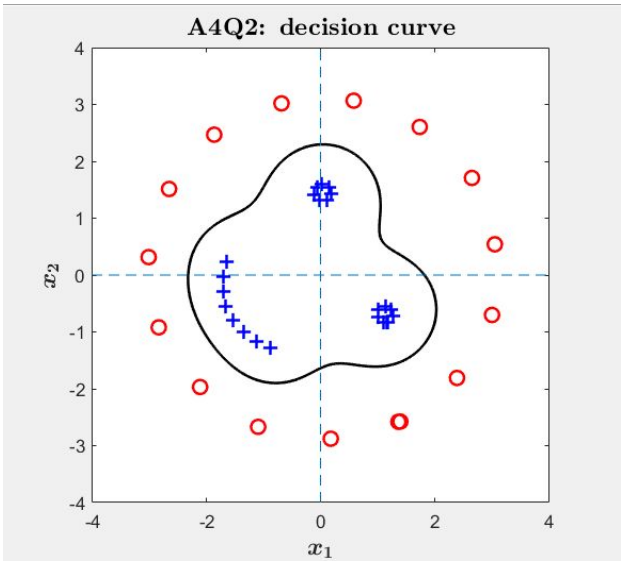




Accuracy = 0.93, confusion matrix is:

550 15
32 180

Q2



A4Q2: classified 100.00% correctly

Q3

q4a3out.wav with λ=10 was submitted.

Discussion

For q1, I chose to perform data standardization after comparing the graph that is with standardized data and one with non standardized data. Both accuracies are around 93%, so it doesn't matter whether it use standardization.
Dimensionality reduction can help SVM model classify the data vectors.
AUC measures the entire two-dimensional area underneath the entire ROC curve. We got 0.97 auc which means the predicitons are 97% correct.

For q2, since the function svmgausscheck has support score calculation, so I didn't use matlab built-in function predict to obtain the score.

For q3, Tikhonov Regularization is also a way to minimize error as CLS we learned before. The argument λ is a smoothing hyper-parameter, and the results vary from less smooth to more smooth as λ increases. Based on the class notes 26, the ideal λ is 100, so I set it to 100 first and found it over modified. Then, I checked the quality of audio by trying λ=50, λ=20, λ=10, and λ=5. Over modified audio makes the volume of the voice smaller and unclear. The audio that is not modified enough contains a lot of background noise. After several experiments, the one with λ=10 performs best.

Reference

[1] Tikhonov regularization

https://towardsdatascience.com/tikhonov-regularization-an-example-other-than-l2-8922ba51253d
[2] Class notes
https://onq.queensu.ca/d2l/home/329280