LINEAR AND NONLINEAR SEPARABILITY

ABSTRACT

PURPOSE: Assess the relative strengths and weaknesses of varying methods for data separation of linearly and nonlinearly separable data.

METHODS: The perceptron algorithm, built-in *kmeans* function and provided *svm271* functions were used to separate the data in 2D space and only the provided *svm271* function was used for 3D space (nonlinearly separable data) separation.

RESULTS: kmeans misclassified 33 vectors, 7 of which are duplicates. The perceptron algorithm was unable to separate the data under a 1000 iteration restriction and SVM managed to separate the data well in both 2D and 3D space.

CONCLUSIONS: SVM is most well rounded hyperplane producing method. The perceptron algorithm is optimal with tightly plotted data and kmeans is optimal with no duplicates and perhaps less data.

INTRODUCTION

The objective was to assess the relative strengths and weaknesses of different data separating methods for two sets of data, linearly and nonlinearly separable. The methods used for the separable data are kmeans, the perceptron algorithm and SVM. For matlab, the built-in kmeans function operates on random number generation, meaning the results can be different for the same data and code, therefore we must implement *rng('default');* so it runs with the default randomized values each time. The perceptron algorithm must be implemented in the *linseplearn* skeleton provided by the instructor and for SVM the provided function *svm271* returns a hyperplane which can be easily plotted. For nonlinearly separable data, we use the provided *svm271* to produce a hyperplane from 3D space data computed from an embedding map. The relative strengths and weaknesses of these methods are assessed qualitatively through plots created in matlab. The functions *plotclass2d* and *plotclass3d* were provided by the instructor and used to plot the given data with proper classifications/separation. Furthermore to plot the hyperplanes produced by functions in the matlab code, the instructor-provided *plotline* and *plotplane* functions were used on 2D and 3D space data accordingly. There were no quantitative measures to assess the strength of the hyperplane/separation besides counting misclassifications for the kmeans method of classifying data.

METHODS

I began with assessing kmeans on linearly separable data, we pass the built-in *kmeans* function a Nx2 data set and the number of clusters to separate into, which would be 2. The *kmeans* function returns centroids as a 2x2 matrix and classifications of the data in a vector, the centroids are irrelevant for the purposes at hand. The classification vector is set to 1's and 2's rather than -1 and +1 like the classification data provided, we must create a loop to set these classification values appropriately so that the two vectors can be compared for misclassifications.

Classifications set to 1 are set to -1 and classifications set to 2 are set to +1. If the current data point is misclassified, both its data and index in the data set are recorded in two different arrays. The number of misclassified vectors and their indices are displayed in the command window. We can now create plots 2 and 3. Plot 2 is created using the *plotclass2d* function with the original data and kmeans classifications as parameters. Plot 3 is created by plotting each data point in the matrix of misclassified vectors through a for loop. Moving on to the perceptron algorithm of part b, we must implement the for loop in the *linsepiterate* function. This for loop compares the classification with the current weight vector to the proper classification (separate parameter) and adjusts according to the perceptron algorithm. The convergence/strength of the hyperplane is limited by the amount of iterations, it is set to 1000 in this case. We will pass this function a random augmented weight vector with each entry between 0 and 1 using the *rand* function, as well as the data and classifications of such. *Linsepiterate* will return a hyperplane, v_final and the number of iterations used, i_used. Plot 4 can now be created using *plotclass2d* to plot the data and the provided *plotline* to plot the hyperplane v_final. Proceeding to the SVM, we use the provided *svm271* function to return a hyperplane when giving the data and its classifications as parameters. We can now create plot 5 in the same fashion as plot 4, using *plotclass2d* and *plotline*. We now switch to nonlinearly separable data, beginning by plotting the data using *plotclass2d*, this is plot 6. Plot 7 requires that we create an embedding map to transform the data into 3D space so that its separable, the map is as such [x; y; x*y]. A loop is created to transform the data, with each iteration a vector of form [x; y; x*y] is created from the original data of form [x; y] and added to a new array containing the transformed data. Plotting this transformed data we use the provided *plotclass3d* function, this is plot 7. Now that the data is separable, we can compute and display a hyperplane for plot 8. We can get the hyperplane using the provided *svm271* function giving it the 3D space data and the associated classifications as parameters. Using *plotclass3d* to plot the data and the provided *plotplane* to plot the hyperplane, we come up with the final plot, plot 8.

RESULTS

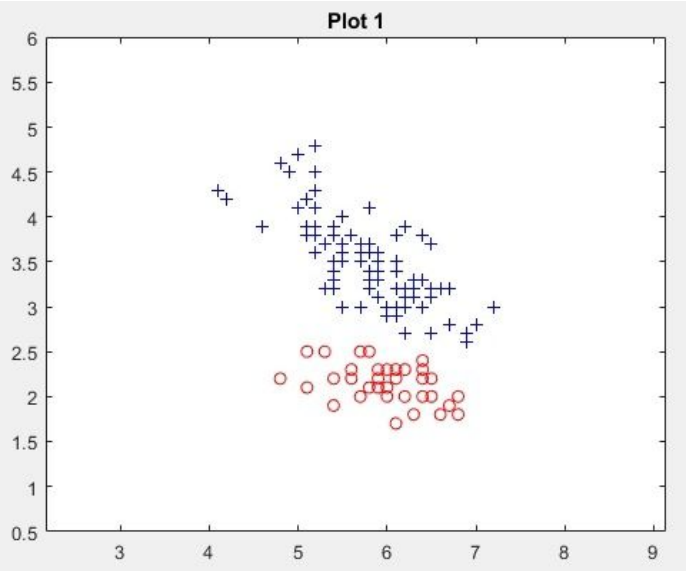

Figure 1: Linearly separable data

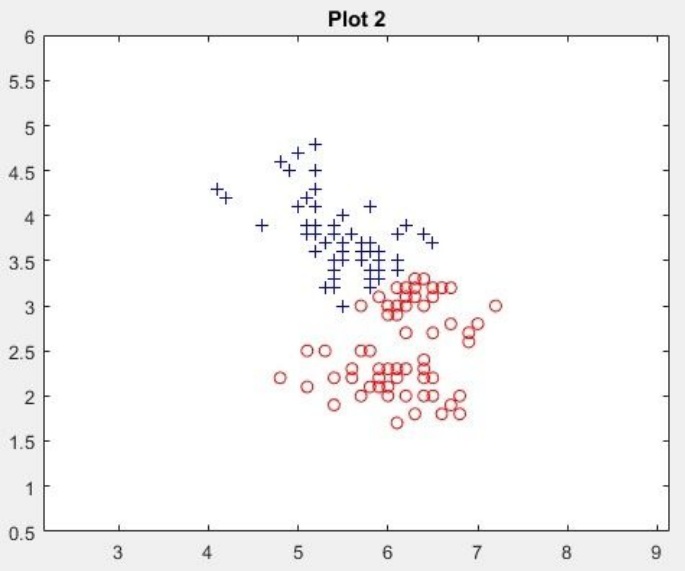

Figure 2: kmeans classification of vectors

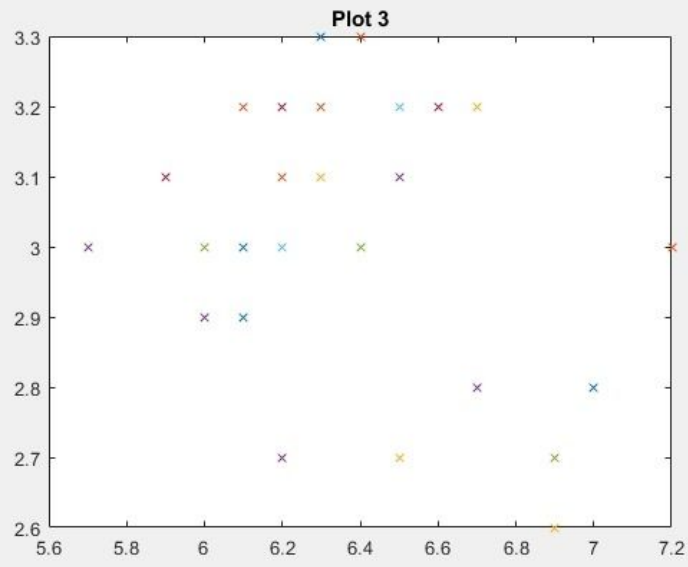

Figure 3: Misclassified vectors of kmeans

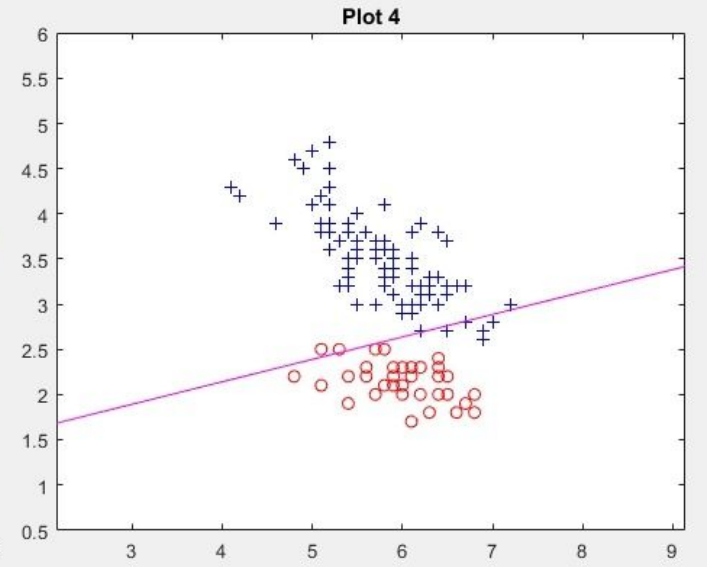

Figure 4: Perceptron algorithm hyperplane

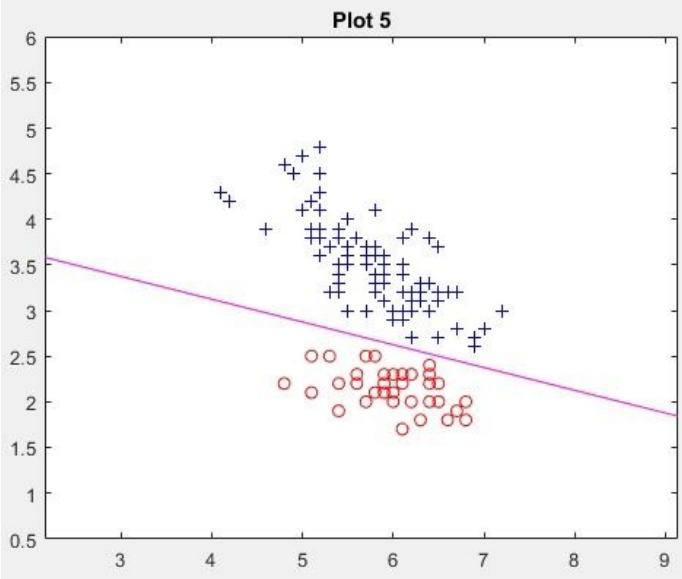

Figure 5: SVM hyperplane in 2D space

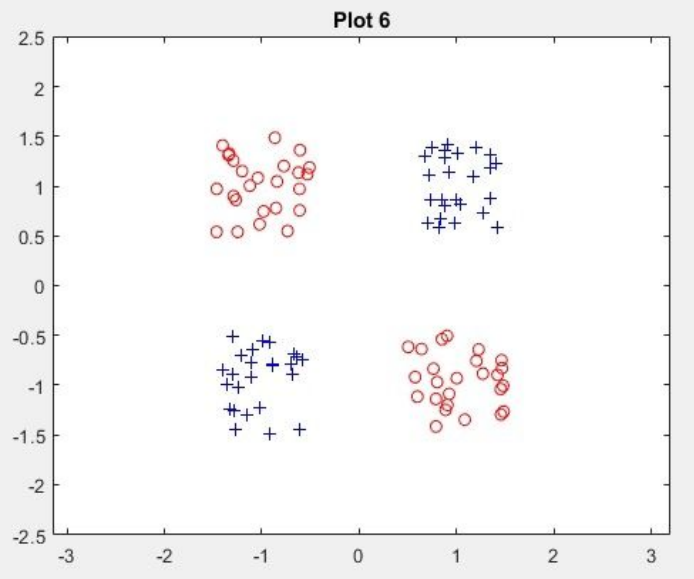

Figure 6: Nonlinearly separable data

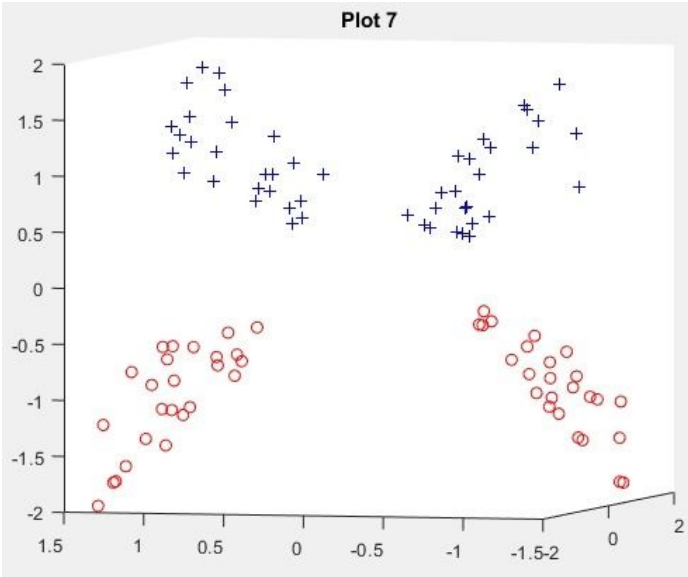

Figure 7: Transformed data using embedded map

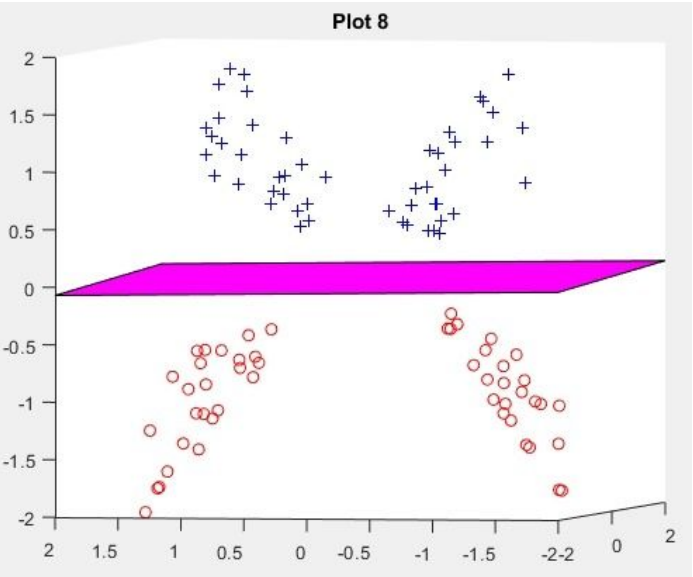

Figure 8: SVM hyperplane in 3D space

Problem 1.

(a) Misclassified vector indexes: [1, 5, 9, 14, 27, 29, 30, 33, 39, 40, 42, 48, 50, 51, 52, 58, 72, 91, 93, 102, 104, 107, 108, 111, 112, 116, 124, 126, 128, 138, 139, 143, 149]

(b) Augmented vector: 1.0e+03 * [-0.2220; 0.8919; -1.0204]

(c) Augmented vector: [0.8353; 3.3412; -13.7847]

Problem 2.

(b) Augmented vector: [-0.0620; 0.0343; 2.2183; -0.0367]

DISCUSSION

For linearly separable data, from *figures 1 and 2*, we can deduce that using kmeans with a fairly large, focused amount of data may not be optimal. This is due to duplicates of data and how it affects the algorithm. In the dataset xpetal there are duplicate vectors, you can see this in *figure 2* as only 26 points swapped labels but there are 33 indexes of misclassified vectors. Having duplicates acts as a weight on the means of data, a duplicate point would in theory be worth more than a non-duplicate and hence results in the classifications of *figure 2,* wrong. Furthermore from *figures 4 and 5*, we can see that the SVM hyperplane actually separates the data, whereas the perceptron algorithm hyperplane does not. We could suggest that for large amounts of data SVM computation of a hyperplane is recommended, this is arguable however, only 1000 iterations were used for the perceptron algorithm. If we increase this value to find the convergence, it stops at 2325 iterations and the hyperplane separates by the difference of the algorithms last operation, which is minute. If the data is tightly plotted, the perceptron algorithm would be optimal as the hyperplane only shifts by the difference of the data, so it can separate by the smallest margins. Furthermore, from *figure 8* we can see that once again SVM is able to separate the data well and one might argue that its optimal in any space as it computed hyperplanes in both 2D and 3D. It's a well rounded method for computing a hyperplane. Overall, each method has its strengths and weaknesses, but the most consistent of the bunch is SVM computation of a hyperplane as it was able to separate the data well in different spaces with different data.

REFERENCES

[1] Classification – K-Means Clustering [Internet]. queensu. Available from: https://onq.queensu.ca/content/enforced/260955-CISC271/Notes/Class28.pdf

[2] CISC 271, Winter 2019, Assignment #4 [Internet]. queensu. Available from: https://onq.queensu.ca/content/enforced/260955-CISC271/Homework/A4.pdf