Project 4: Support Vector Machines

Isaiah Chen

May 1, 2019

Question 1

(1) *kernel1Train.csv* contains training data with 1,000 samples, where each sample has two coordinates and the corresponding class. *kernel1Test.csv* contains testing data with 10,000 samples and the same feature space as the training data. The testing data has been reshaped to a 100×100 grid and a visualization of this grid is shown below in Figure 1.

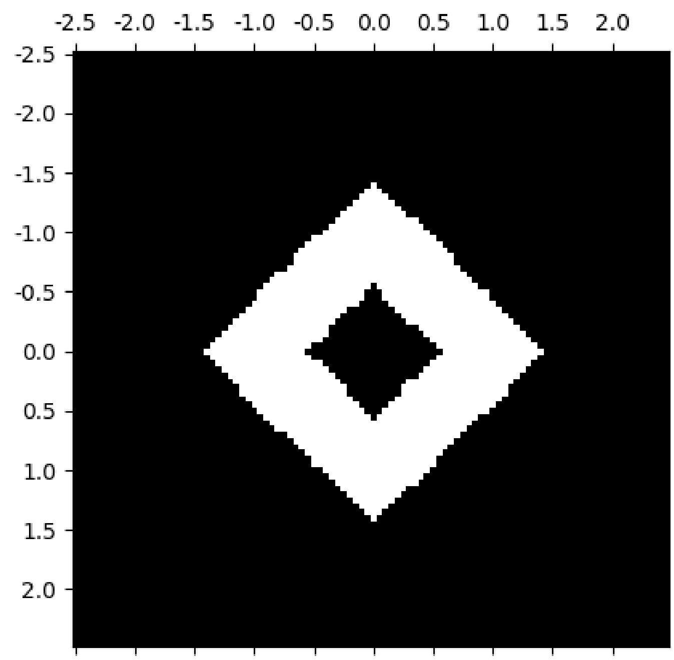

Figure 1: The test set visualized as a 100×100 grid ranging from -2.5 to 2.45 on both axes. The black cells correspond to a class of 0 and the white cells correspond to a class of 1.

(2) Using the SVC class in SciKit-learn, we can train an SVM using a pre-defined kernel or

a custom kernel that is specified by the user. For this problem, we use the Cauchy kernel, which is defined as:

$$K_\sigma(x, x') = \frac{1}{1+\frac{|x-x'|^2}{\sigma^2}}$$

We use the provided training data to train an SVM for different values of σ. The classifer's error for each value of σ is shown below in Table 1. The error is defined as the fraction of incorrectly-predicted classes over the total number of samples tested. The predicted classes for each case are shown below in Figures 2 - 8, all with the same style of visualization that is shown in Figure 1. A σ value of 0.1 yields the best classifier that is able to most accurately predict the classes of the test data.

σ	Error
0.01	0.1243
0.025	0.0277
0.05	0.0157
0.1	0.0144
0.5	0.0161
1.0	0.0217
2.5	0.0511

Table 1: Classification error for different values of σ.

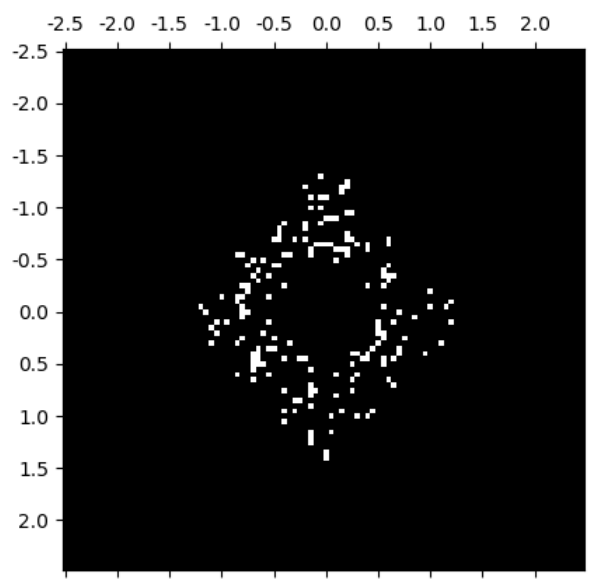

Figure 2: Predicted classes for $\sigma = 0.01$.

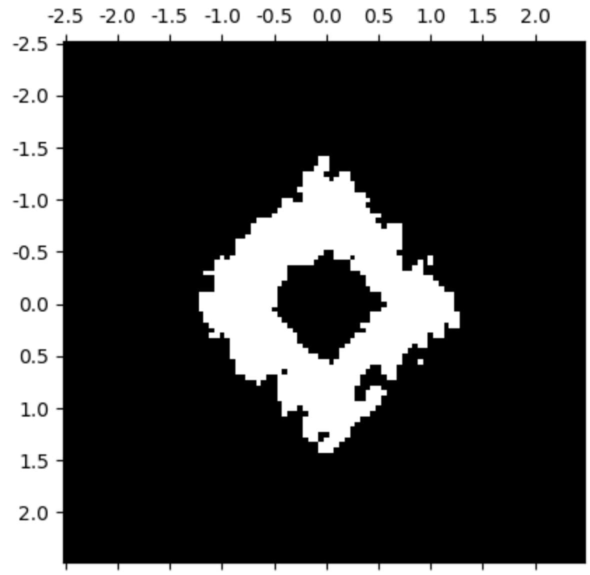

Figure 3: Predicted classes for $\sigma = 0.025$.

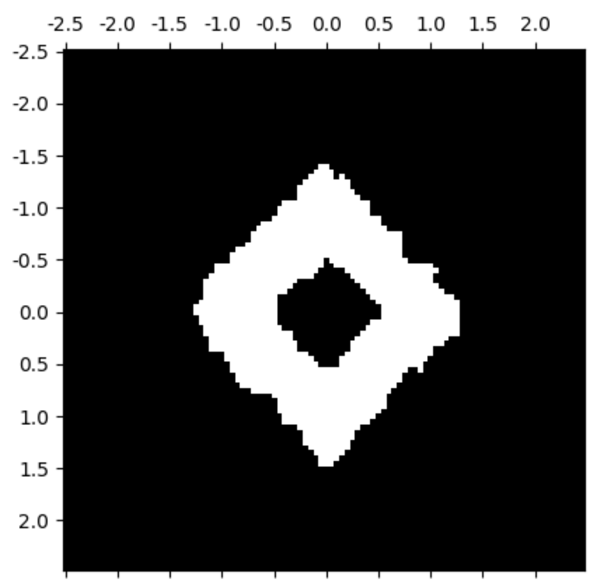

Figure 4: Predicted classes for $\sigma = 0.05$.

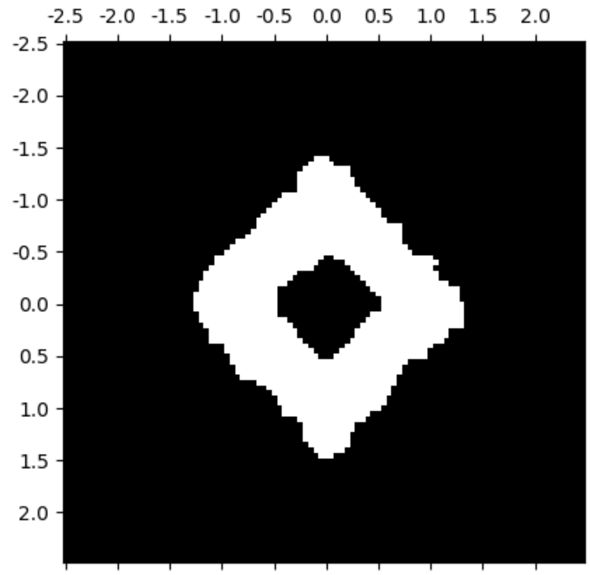

Figure 5: Predicted classes for $\sigma = 0.1$.

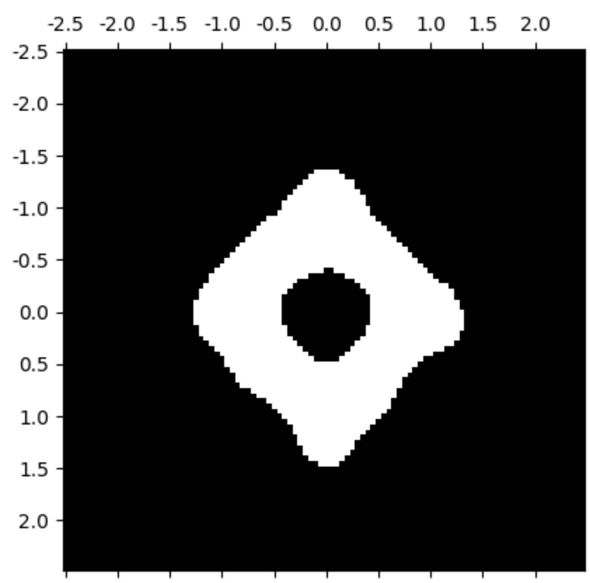

Figure 6: Predicted classes for $\sigma = 0.5$.

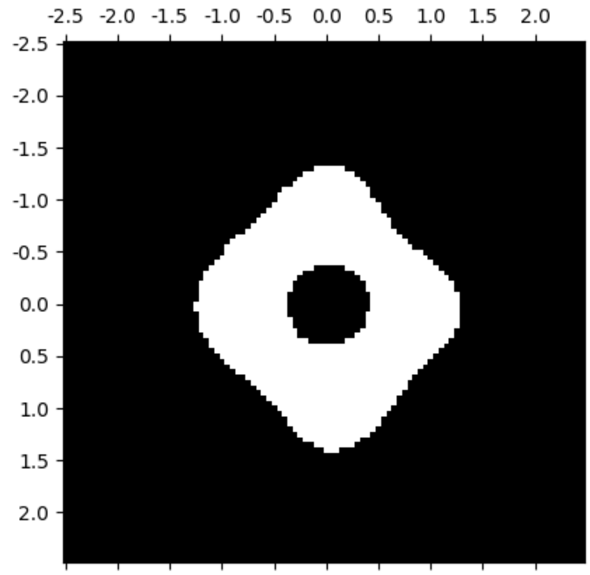

Figure 7: Predicted classes for $\sigma = 1.0$.

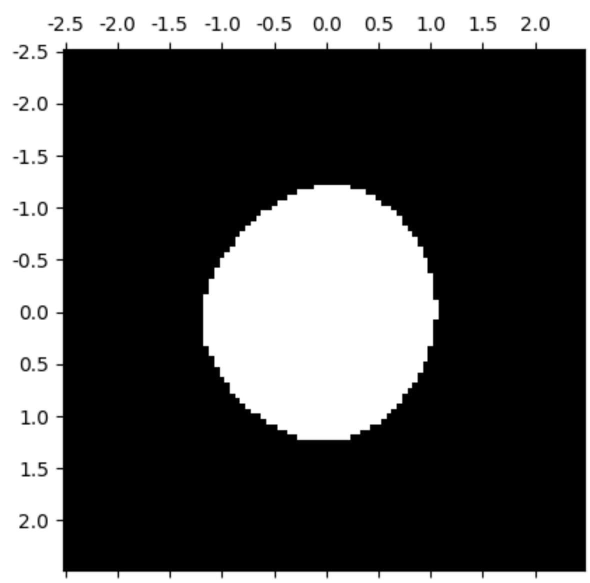

Figure 8: Predicted classes for $\sigma = 2.5$.

Question 2

(1) Using the procedure described in the question, an estimator is calculated using the following kernel:

$$K_\sigma(x, x') = (1 + \frac{|x-x'|}{\sigma} + \frac{|x-x'|^2}{3\sigma^2})e^{-\frac{|x-x'|}{\sigma}}$$

For this problem, $m = 10$ and $v_1, ..., v_{10}$ correspond to the m columns of the ten-dimensional identity matrix. Using the methodology described in the problem description, an SVM has been trained and the resulting confusion matrix for the test data is shown below:

$$\begin{bmatrix} 0.9857 & 0 & 0 & 0 & 0.0143 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9592 & 0.0204 & 0 & 0 & 0 & 0 & 0.0204 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9783 & 0 & 0 & 0 & 0.0217 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9836 & 0 & 0 & 0.0164 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0189 & 0.9811 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where an element $a(g, g_0), g, g_0 \in \mathcal{G}$ provides the fraction of testing samples whose true class is g_0 that are predicted to belong in class g. This syntax for all following confusion matrices applies.

(2) For this problem, $m = 20$ and $v_1, ..., v_m$ are given by random permutations of v_0 containing five zeros and five ones. Using the same kernel from Question 2.1 for training, the resulting confusion matrix is shown below:

$$\begin{bmatrix} 0.9714 & 0 & 0 & 0 & 0.0143 & 0 & 0 & 0 & 0.0143 & 0 \\ 0 & 0.9592 & 0.0204 & 0 & 0 & 0 & 0 & 0.0204 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9783 & 0 & 0 & 0 & 0.0217 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9836 & 0 & 0 & 0.0164 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0189 & 0.9811 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

This confusion matrix is nearly equivalent to the confusion matrix from the previous question. This must be due to the classifier that is used to determine which class each sample belongs to. The classifier must be nearly similar, if not equivalent for both problems.

(3) We repeat the analysis for the two values of m using the Cauchy kernel defined in Question 1. The resulting confusion matrix for $m = 10$ is shown below:

$$\begin{bmatrix} 0.9714 & 0 & 0 & 0 & 0.0143 & 0 & 0 & 0 & 0.0143 & 0 \\ 0 & 0.9592 & 0.0204 & 0 & 0 & 0 & 0 & 0.0204 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9783 & 0 & 0 & 0 & 0.0217 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9836 & 0 & 0 & 0.0164 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0189 & 0.9811 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The resulting confusion matrix for $m = 20$ is shown below:

$$\begin{bmatrix} 0.9714 & 0 & 0 & 0 & 0.0143 & 0 & 0 & 0 & 0.0143 & 0 \\ 0 & 0.9592 & 0.0204 & 0 & 0 & 0 & 0 & 0.0204 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9783 & 0 & 0 & 0 & 0.0217 & 0 & 0 \\ 0 & 0.0164 & 0 & 0 & 0.9836 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0189 & 0.9811 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For all 4 cases in Question 2, the SVM is able to consistently predict what the correct class is for the large majority of the samples in the test data. Again, this consistency may indicate that the classifier is nearly equivalent for all these cases. Because the methodology for computing the classifier is different for different values of m, the confusion matrices should be more varied. This may be due to possible programming errors.