Homework 2
Kernel SVM and Perceptron

Dana Van Aken

Problem 1: SVM decision boundaries

1. (a) ⟶ 4
   Since this a linear SVM, it has to be either graph 3 or 4. The correct graph is 4 because there are 1-2 slight margin errors, so C must be small since the data is linearly separable and would otherwise try to fit the training data exactly if the cost was high.

   (b) ⟶ 3
   This corresponds to graph 3 because it's linear and is trying to fit the data more precisely than graph 4 which means that the cost is higher. The decision boundary in graph 3 is less general than the one in graph 4.

   (c) ⟶ 5
   This equation is quadratic and it's actual graph should look similar to graph 5. I do *not* think that this equation corresponds to graph 2 because graph 2 is noisy and I would expect it to be something like a Laplace kernel.

   (d) ⟶ 1
   Graphs 1 and 6 are similar in shape and appear to be RBF kernels. This equation specifically corresponds to graph 1 because the variance is larger. A larger variance helps generalize the decision boundary since the data points deviate further from the mean.

   (e) ⟶ 6
   This equation corresponds to graph 6 since the variance is small. This small variance results in overfitting the data points (in comparison to the more general fit in graph 1).

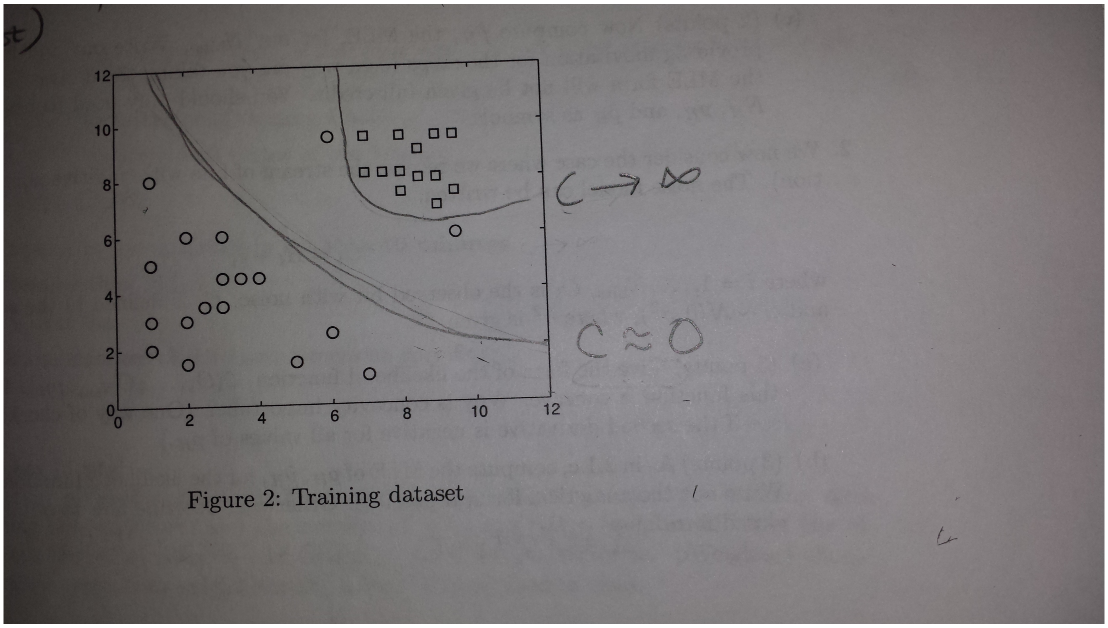

Figure 2: Training dataset

2. (a) As C approaches infinity, the cost of misclassification is extremely high so the model will try to separate the data exactly. In this case, it is able to do this since the training data is linearly separable, but the resulting boundary overfits the data.

(b) If C is near 0, then there is no cost to overfitting so 1 or more outlier data points do not determine the decision boundary. This means that we can permit the outliers to be margin errors which results in a decision boundary that is much more general in comparison to the decision boundary produced when the cost approaches infinity.

(c) Case (b) would work a lot better in the classification task. Given that the training data is error-prone, using case (a) would be highly inappropriate because it assumes that the training data is noise-free. Case (b) will produce a more general decision boundary that will likely classify the data more successfully than case (a).