DSC 255: Machine learning

Week 7 — Solutions

1. To classify a point x, we evaluate the three linear functions and pick the one with the highest value.

   The region where class 1 beats class 2 is:

   $$w_1 \cdot x + b_1 > w_2 \cdot x + b_2 \;\Leftrightarrow\; (w_1 - w_2) \cdot x + (b_1 - b_2) > 0 \;\Leftrightarrow\; x_2 > 1$$

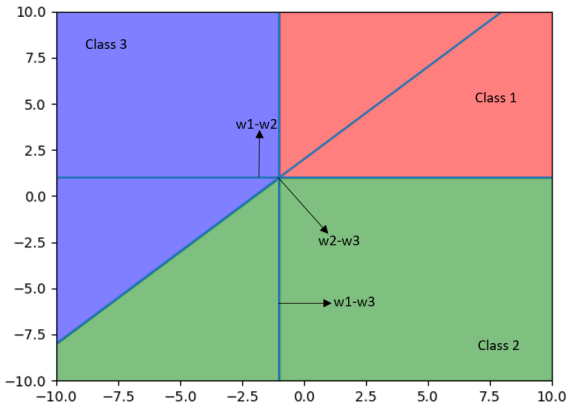
   The region where class 1 beats class 3 is:

   $$w_1 \cdot x + b_1 > w_3 \cdot x + b_3 \;\Leftrightarrow\; (w_1 - w_3) \cdot x + (b_1 - b_3) > 0 \;\Leftrightarrow\; x_1 > -1$$

   The region where class 2 beats class 3 is:

   $$w_2 \cdot x + b_2 > w_3 \cdot x + b_3 \;\Leftrightarrow\; (w_2 - w_3) \cdot x + (b_2 - b_3) > 0 \;\Leftrightarrow\; x_1 - x_2 > -2$$

   So class 1 is predicted in the intersection of the first two regions, etc. This is summarized in the figure below.



2. *Plant recognizer.* This is somewhat subjective. However, options (a) and (c) are clearly problematic. In option (a), you could end up with all sorts of plants that don't even appear in California. In option (c), you might end up with only a tiny subset of the plants that are present in the entire state. Thus (b) seems wisest.

3. *A shortage of data.* There are several ways in which this situation — good SVM generalization even though the number of data points is much smaller than the data dimension — might be explained. Here are two prominent candidates.

   - The SVM might have large margin. In this case, the model has "low complexity" in spite of the high apparent dimension.

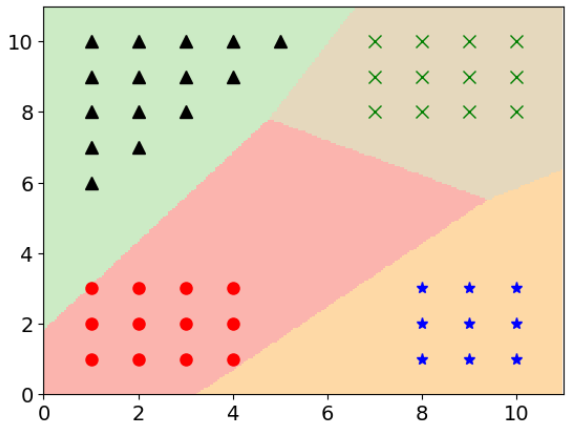
- The data distribution might be very simple: for instance, many of the features might have close-to-constant values, so that the effective dimension is much lower than it appears to be.

4. *Distribution shift.*

   (a) This is a clear case of *label shift*: the proportions of different labels has changed.

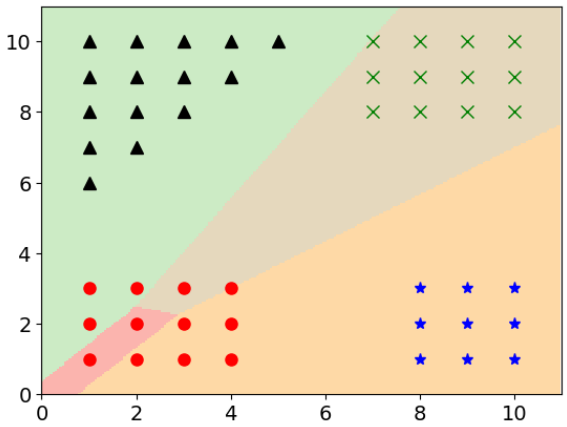
   (b) This is *covariate shift*: the marginal distribution on x has changed.

5. *Multiclass Perceptron.*

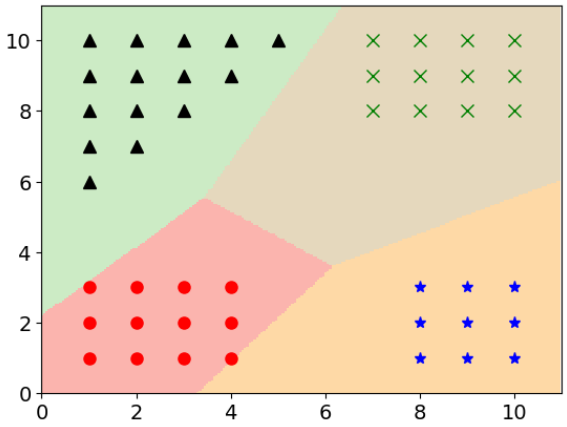


6. *Multiclass SVM.*

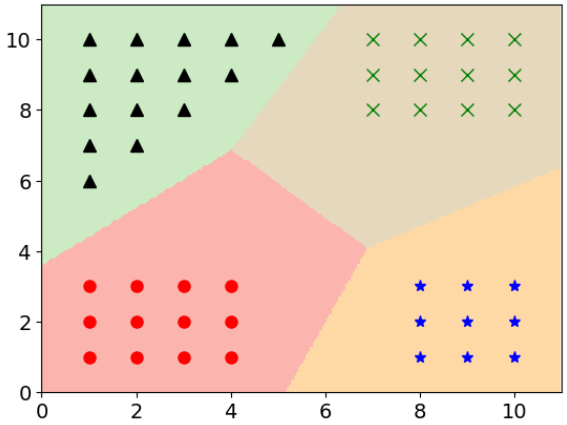
   (a) Here are the plots for the four settings of C.
   $C = 0.01$.



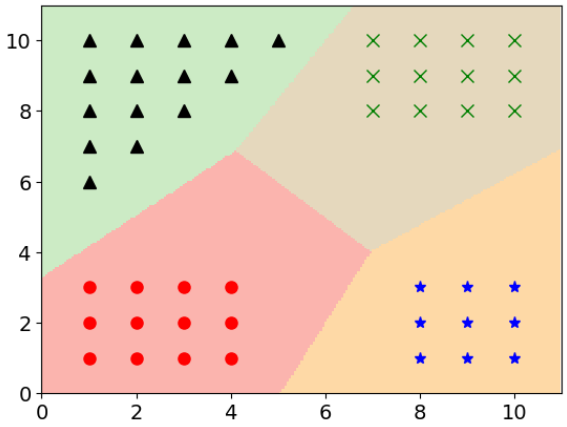
   $C = 0.1$.



$C = 1.0$.



$C = 10$.



(b) When C is tiny (that is, when slack is cheap), the decision boundary misclassifies some of the training points. As C grows, the training error goes to zero.