Solution 1

Step 1

The general form for a linear predictor is the following:

$$f(x) = w^T x + b$$

We are given three linear functions for each class:

$$\text{Class 1: } w_1 = (1,1), b_1 = 0 \rightarrow f_1(x) = x_1 \cdot (w_1^1) + x_2 \cdot (w_2^1) + b_1 = x_1 \cdot (1) + x_2 \cdot (1) + 0 = x_1 + x_2$$

$$\text{Class 2: } w_2 = (1,0), b_2 = 1 \rightarrow f_2(x) = x_1 \cdot (w_1^2) + x_2 \cdot (w_2^2) + b_2 = x_1 \cdot (1) + x_2 \cdot (0) + 1 = x_1 + 1$$

$$\text{Class 3: } w_3 = (0,1), b_3 = -1 \rightarrow f_3(x) = x_1 \cdot (w_1^3) + x_2 \cdot (w_2^3) + b_3 = x_2 \cdot (0) + x_1 \cdot (1) - 1 = x_2 - 1$$

Hence, for each class, we have the following:

$$\begin{aligned} f_1(x) &= x_1 + x_2, \\ f_2(x) &= x_1 + 1, \\ f_3(x) &= x_2 - 1. \end{aligned}$$

Step 2

Compare f_1, f_2, and f_3 at the intersection points of the decison boundaries:

$$\begin{aligned} &(1 \text{ vs } 2): \ f_1 = f_2 \ \rightarrow \ x_1 + x_2 = x_1 + 1 \ \rightarrow \ x_2 = 1 \ \rightarrow \ (0,1) \\ &(1 \text{ vs } 3): \ f_1 = f_2 \ \rightarrow \ x_1 + x_2 = x_2 - 1 \ \rightarrow \ x_1 = -1 \ \rightarrow \ (-1,0) \\ &(2 \text{ vs } 3): \ f_2 = f_3 \ \rightarrow \ x_1 + 1 = x_2 - 1 \ \rightarrow \ x_2 = x_1 + 2 \ \rightarrow \ (0,2) \text{ and } (-2,0) \end{aligned}$$

Hence, we have the following decision boundaries:

$$\begin{aligned} &\text{Decision Boundary between Class 1 and Class 2:} \quad x_2 = 1 \\ &\text{Decision Boundary between Class 1 and Class 3:} \quad x_1 = -1 \\ &\text{Decision Boundary between Class 2 and Class 3:} \quad x_2 = x_1 + 2 \end{aligned}$$

Step 3

Note: $x_2 = 1$, $x_1 = -1$, and $x_2 = x_1 + 2$ all pass through the common point $(-1, 1)$,and slice the plane into 6 wedges.

Now, select six test points for each wedge, and evaluate f_1, f_2, and f_3 to determine the classification.

Wedge label	Test point	$f_1(x)$	$f_2(x)$	$f_3(x)$	Classification
A	$(1, 4)$	5	2	3	Class 1
B	$(1, 2)$	3	2	1	Class 1
C	$(1, -1)$	0	2	-2	Class 2
D	$(-2, -1)$	-4	-1	-3	Class 2
E	$(-3, 0)$	-3	-2	-1	Class 3
F	$(-3, 3)$	0	-2	2	Class 3

Step 4

Visualize points and classify each wedge.

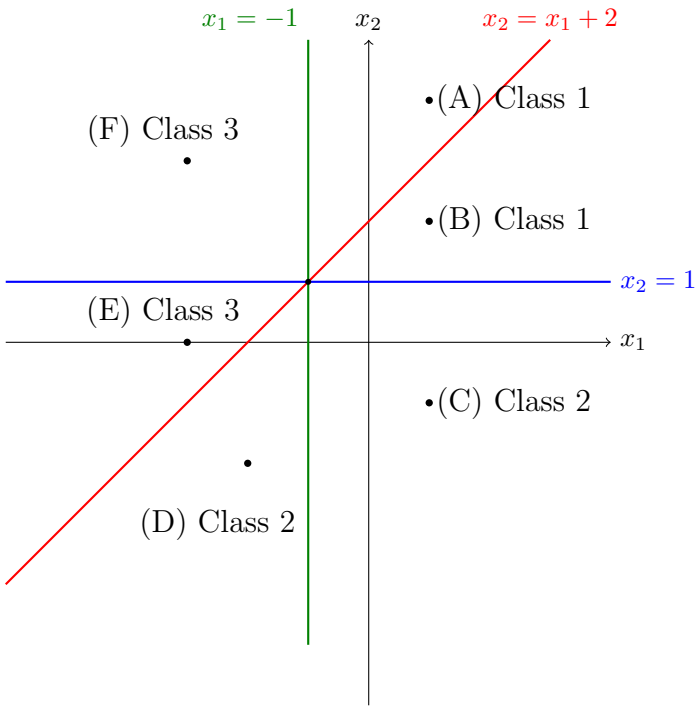

Figure 1: All six wedges produced by the three pairwise boundaries, with the winning class shown in each.

$\therefore$ the decision boundaries and the regions that belong to each class can be seen in *Figure 1.*

Solution 2

Step 1

In order to determine which option will best satisfy the statistical learning framework, we will evaluate each option.

- Option A: A web search on *American plants* will yeild subset of plant images outside of California.
- Option B: Plant photos from a similar region to that of California, will yeild similar flora but not identiencal to Californian flora.
- Option C: Plant photos from the favorite Californian city will yeild photos of plants will have a higher bias towards urban Californian flora.

$\therefore$ Option B will best satisfy the statistical learning framework, since it will introduce the least amount of bias towards Californian plant identification.

Solution 3

Step 1

We will explore a few potential reasons why the SVM classifier performs well despite the large number of features.

- If the classes the SVM is predicting are sufficently different, the SVM will be able to find a good decision boundary that separates the data. As a consquence of the difference between the classes, the SVM will have a large margin. This means that the SVM can find a good decision boundary even with a large number of features and minimal training points.
- The generalizaton error can be expressed as:

$$\text{generalisation error } \approx \sqrt{\frac{R^2/\gamma^2}{n}}$$

  So we can see that as the margin grows largers, the generalization error decreases. This means that the SVM can find a good decision boundary even with a large number of features and minimal training points.

$\therefore$ a wide margin for the SVM is a key reason the 1000 traing points are sufficent reguardless of the one million features in the data.

Solution 4 (a)

The class priors changed: fewer *sports* articles, more *politics* Within each topic (class), the way the text is written is assumed unchanged, so only the frequency of each label moves.

$\therefore$ this would be an example of a label shift

Solution 4 (b)

The vocabulary has drifted: new proper nouns appear and old ones vanish. This alters the distribution of the features within each topic, while the overall mix of topics (sports, politics, business,etc.) stays the same.

$\therefore$ this is an example of a covariate shift

Solution 5

Python Code

```python
## import libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.utils import shuffle

## load data0.txt
data = np.loadtxt('data0.txt', dtype=int)
x = data[:, :2]
y = data[:, 2]

## get number of unique classes
k = len(np.unique(y))

## define multi_perceptron function
def multi_perceptron(w, b, x):
    predictions = np.dot(w, x) + b
    best_prediction = predictions.argmax()
    return(best_prediction)

## define fit_multi_perceptron function
def fit_multi_perceptron(x, y, k,track_updates, set_seed):
    if set_seed:
        x, y = shuffle(x, y,random_state=42)
    else:
        x, y = shuffle(x, y)
    w = np.zeros((k, x.shape[1]))
    b = np.zeros(k)
    updates = 0
    max_updates = 1000

    def make_prediction(w, b, x, y, updates):
        error = False
        for xi, yi in zip(x, y):
            predicted_y = multi_perceptron(w, b, xi)
            if predicted_y != yi:
                w[yi] += xi
                b[yi] += 1
                w[predicted_y] -= xi
                b[predicted_y] -= 1
                updates += 1
                error = True
        if updates <= max_updates:
            if error:
                return make_prediction(w, b, x, y, updates)
            else:
                return (w, b, updates)
        else:
            print("Did not converge after {max_iterations} iterations")
    if track_updates:
        w, b, updates = make_prediction(w, b, x, y, updates)
        return (w, b, updates)
    else:
        w, b, updates = make_prediction(w, b, x, y, updates)
        return (w, b)
```

Plots

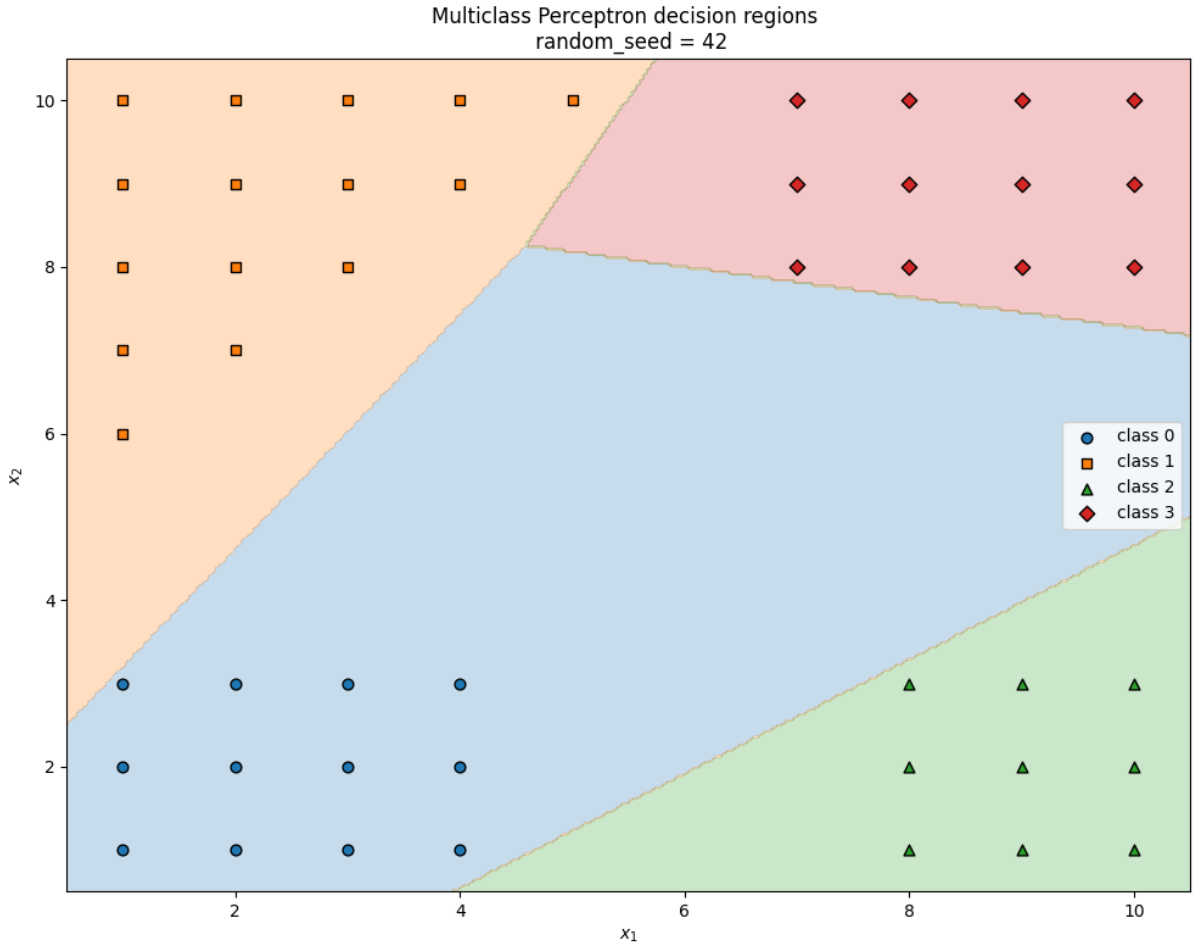

Figure 2: Multiclass Perceptron decision boundaries.

Solution 6

Plots

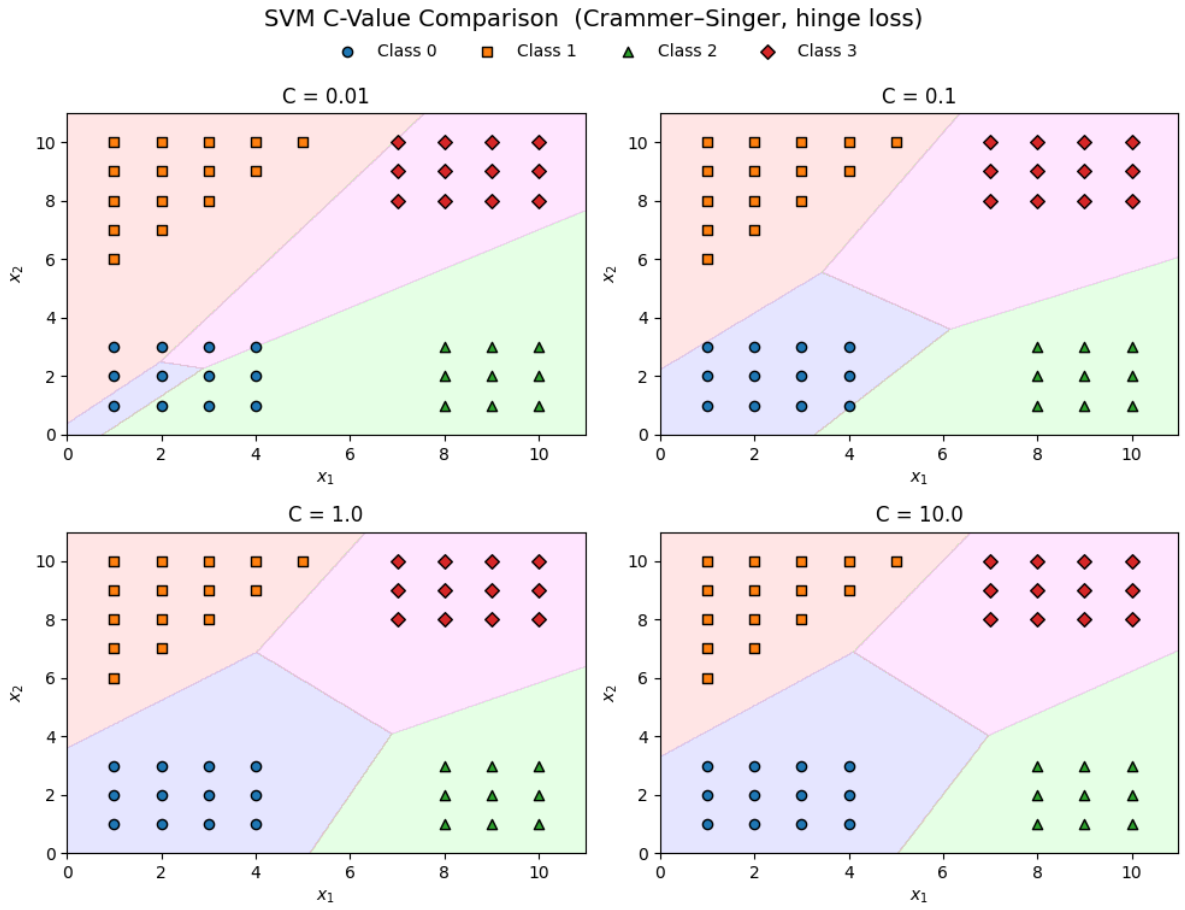

Figure 3: SVM C value comparison.

C Value Comparison Comments

Observations across C values:

- As C increases the model partitions the data more effectively
- Since there are a limited amount of features in this dataset (*data0.txt*), we don't have to worry about minimizing the margin, since the computational cost won't skyrocket with this dataset.