CS 5350/6350: Machine Learning Fall 2024

Homework 3

Handed out: 22 Oct, 2024
Due date: 11:59pm, 8 Nov, 2024

1 Paper Problems [36 points + 15 bonus]

1. [8 points] Suppose we have a linear classifier for 2 dimensional features. The classification boundary, i.e., the hyperplane is $2x_1 + 3x_2 - 4 = 0$ (x_1 and x_2 are the two input features).

x_1	x_2	label
1	1	1
1	-1	-1
0	0	-1
-1	3	1

Table 1: Dataset 1

(a) [4 points] Now we have a dataset in Table 1. Does the hyperplane have a margin for the dataset? If yes, what is the margin? Please use the formula we discussed in the class to compute. If no, why? (Hint: when can a hyperplane have a margin?)

The hyperplane has a margin if it correctly classifies all the training examples. Let's start by calculating the predictions for each example:

$$W = [2, 3] \quad B = -4$$

Ex 1: $2 \cdot 1 + 3 \cdot 1 - 4 = 1 \geq 0$ **so prediction is 1.**
Ex 2: $2 \cdot 1 + 3 \cdot -1 - 4 = -5 < 0$ **so prediction is -1.**
Ex 3: $2 \cdot 0 + 3 \cdot 0 - 4 = -4 < 0$ **so prediction is -1.**
Ex 4: $2 \cdot -1 + 3 \cdot 3 - 4 = 3 > 0$ **so prediction is 1.**

This hyperplace correctly classifies the data so it has a margin. Now we calculate the distance from each example to the hyperplane using the formula $\frac{|\mathbf{w}^\top \mathbf{x}_0 + b|}{||\mathbf{w}||}$**.**

The numerator term was calculated above so we just need ||w|| which is $\frac{1}{\sqrt{13}}$**. The distances for the 4 examples are** $\frac{1}{\sqrt{13}}$, $\frac{5}{\sqrt{13}}$, $\frac{4}{\sqrt{13}}$, **and** $\frac{3}{\sqrt{13}}$, **respectively. The closest example is** $\frac{1}{\sqrt{13}}$ **away from the hyperplane so the margin is** $\frac{1}{\sqrt{13}}$**.**

x_1	x_2	label
1	1	1
1	-1	-1
0	0	-1
-1	3	1
-1	-1	1

Table 2: Dataset 2

(b) [4 points] We have a second dataset in Table 2. Does the hyperplane have a margin for the dataset? If yes, what is the margin? If no, why?

The first four examples are the same as the first part. Let's calculate the prediction for the new example:

$$W = [2, 3] \quad B = -4$$

Ex 5: $2 \cdot -1 + 3 \cdot 3 - 4 = -9 < 0$ **so prediction is -1.**

This new example is misclassified so there is no margin.

2. [8 points] Now, let us look at margins for datasets. Please review what we have discussed in the lecture and slides. A margin for a dataset is not a margin of a hyperplane!

x_1	x_2	label
-1	0	-1
0	-1	-1
1	0	1
0	1	1

Table 3: Dataset 3

(a) [4 points] Given the dataset in Table 3, can you calculate its margin? If you cannot, please explain why.

The margin of the dataset is the smallest distance between any pair of points that have different labels. First, I will calculate the distances between all points with differing labels.

Ex 1 - Ex 3: Dist=$\sqrt{(1 - -1)^2 + (0 - 0)^2} = 2$

Ex 1 - Ex 4: Dist=$\sqrt{(0 - -1)^2 + (1 - 0)^2} = \sqrt{2}$

Ex 2 - Ex 3: Dist=$\sqrt{(1 - 0)^2 + (0 - -1)^2} = \sqrt{2}$

Ex 2 - Ex 4: Dist=$\sqrt{(0 - 0)^2 + (1 - -1)^2} = 2$

The margin is $\sqrt{2}$**.**

(b) [4 points] Given the dataset in Table 4, can you calculate its margin? If you cannot, please explain why.

x_1	x_2	label
-1	0	-1
0	-1	1
1	0	-1
0	1	1

Table 4: Dataset 4

Again, I will calculate the distances between all points with differing labels.
Ex 1 - Ex 2: Dist=$\sqrt{(0--1)^2+(-1-0)^2}=\sqrt{2}$
Ex 1 - Ex 4: Dist=$\sqrt{(0--1)^2+(1-0)^2}=\sqrt{2}$
Ex 3 - Ex 2: Dist=$\sqrt{(0-1)^2+(-1-0)^2}=\sqrt{2}$
Ex 3 - Ex 4: Dist=$\sqrt{(0-1)^2+(1-0)^2}=2$
The smallest distance is $\sqrt{2}$**. However, this data is similar to XOR and is not linearly separable so it does not have a margin.**

3. [**Bonus**] [5 points] Let us review the Mistake Bound Theorem for Perceptron discussed in our lecture. If we change the second assumption to be as follows: Suppose there exists a vector $\mathbf{u} \in \mathbb{R}^n$, and a positive γ, we have for each $(\mathbf{x}_i, y_i)$ in the training data, $y_i(\mathbf{u}^\top \mathbf{x}_i) \geq \gamma$. What is the upper bound for the number of mistakes made by the Perceptron algorithm? Note that $\mathbf{u}$ is unnecessary to be a unit vector.

   According to the slides, the magnitude of μ **simply scales the mistake bound. The upper bound for number of mistakes is** $\left(\frac{||\mu||\cdot R}{\gamma}\right)^2$**.**

4. [10 points] We want to use Perceptron to learn a disjunction as follows,

$$f(x_1, x_2, \ldots, x_n) = \neg x_1 \vee \neg \ldots \neg x_k \vee x_{k+1} \vee \ldots \vee x_{2k} \quad \text{(note that } 2k < n).$$

   The training set are all 2^n Boolean input vectors in the instance space. Please derive an upper bound of the number of mistakes made by Perceptron in learning this disjunction.

   The Perceptron algorithm will make at most $\left(\frac{R}{\gamma}\right)^2$ **mistakes. We just need to find** R **and** γ**.**

   $R = \sqrt{1+n}$ **in an augmented feature space.**

   The separating hyperplane is $-x_1 - x_2 \cdots - x_k + x_{k+1} + \cdots + x_{2k} = 0$**. This has zero margin because it passes through some of the data points so we need to shift by** $\frac{1}{2}$**. The resulting hyperplane is** $-x_1 - x_2 \cdots - x_k + x_{k+1} + \cdots + x_{2k} - \frac{1}{2} = 0$**. The closest positive example to this shifted hyperplane is** $[0, 0, 0, \ldots, 0, 1]$ **and the closest negative example is** $[1, 1, 1, \ldots, 0, 0, 0]$**. Both examples have a distance of** $\frac{1}{2}$ **to the hyperplane. The weight vector can be written as:**

$$\frac{1}{\sqrt{2k+\frac{1}{4}}}\left[-1, -1, \ldots, -1, 1, \ldots, 1, 1, \frac{1}{2}\right]$$

γ is the margin divided by the norm of the weight vector:

$$\gamma = \frac{\frac{1}{2}}{\sqrt{2k + \frac{1}{4}}}$$

Now we can calculate mistake bound as

$$\left(\frac{R}{\gamma}\right)^2 = \frac{\sqrt{n+1}}{\frac{\frac{1}{2}}{\sqrt{2k+\frac{1}{4}}}}$$

5. [10 points] Prove that linear classifiers in a plane cannot shatter any 4 distinct points.

   First, notice that any 4 non-collinear points in a plane can be represented by a quadrilateral whose corners are those points. If this quadrilateral were non-convex, then it could not be shattered because there would be one point inside a triangle made by the other 3 and it is impossible to include all 3 exterior points without including the interior point. Therefore, we know this quadrilateral must be convex.

   Now, consider the case where opposite corners have the same label (the parity case). It is impossible to draw a line through the convex quadrilateral that includes 2 opposite corners while excluding the other 2. Therefore, linear classifiers in a plane cannot shatter any 4 distinct points.

6. [**Bonus**] [10 points] Consider our infinite hypothesis space $\mathcal{H}$ are all rectangles in a plain. Each rectangle corresponds to a classifier — all the points inside the rectangle are classified as positive, and otherwise classified as negative. What is VC($\mathcal{H}$)?

   We need to determine the size of the largest subset that can be shattered by rectangles in the plane.

   It is trivial to shatter 1 or 2 points with a rectangle. Just place the rectangle only around the points you want to include.

   It is easy to shatter 3 points if they are placed as the points of an equilateral triangle: the rectangle just has to include no points, 1 of the corners, 1 of the sides (2 points), or the whole triangle.

   We can easily shatter 4 points that are arranged as the corners of a diamond. Similarly to the triangle, it is trivial to include 0, 1, 3, or 4 points. The 2 point case is also trivial unless the points across from each other have matching labels (the parity case). However, in this case, you can place a thin rectangle that crosses from corner to corner but that is too thin to include the other 2 corners.

   For 5 points, the best arrangement is as the corners of a regular pentagon. In most cases, the points can be correctly separated as in the 4 point case. However, the parity case causes some problems. Consider where 2 adjacent points have the same label as the one across from those 2 points while the

remaining 2 have the opposite label. This arrangement can be separated by a thin rectangle (as in the 4 point parity case) but there are 5 instances of this case each with different rotations so the rectangles must be able to rotate. Therefore, axis-aligned rectangles cannot shatter 5 points but we are considering all rectangles so it can shatter 5 points.

Finally, we consider the 6 point case. The best arrangement is as the corners of a regular hexagon. It is impossible to shatter these 6 points because of the parity case where labels alternate as you go around the hexagon. You cannot place a rectangle to include those 3 points without including at least 1 of the other points. Therefore, the VC for rectangles in a plane is 5.

2 Practice [64 points]

1. [2 Points] Update your machine learning library. **DONE**

2. We will implement Perceptron for a binary classification task — bank-note authentication.

   (a) [16 points] Implement the standard Perceptron. Set the maximum number of epochs T to 10. Report your learned weight vector, and the average prediction error on the test dataset.

   The final weight vector is $[10.6, -11.840, -7.585, -8.734, -0.155]$**. The test error is 0.0160.**

   (b) [16 points] Implement the voted Perceptron. Set the maximum number of epochs T to 10. Report the list of the distinct weight vectors and their counts — the number of correctly predicted training examples. Using this set of weight vectors to predict each test example. Report the average test error.

   Below are the counts for each weight vector.
   Count: 1 Weights: [0.00, 0.00, 0.00, 0.00, 0.00]
   Count: 3 Weights: [-0.20, -0.44, -1.20, -0.11, -0.17]
   Count: 3 Weights: [0.00, -0.45, -0.45, -1.01, -1.02]
   Count: 3 Weights: [0.20, -0.86, -1.79, 0.79, -1.00]
   Count: 4 Weights: [0.00, -1.79, -1.12, 0.27, -1.11]
   Count: 3 Weights: [0.20, -1.66, -0.34, -0.68, -1.99]
   Count: 3 Weights: [0.00, -1.22, -2.85, -1.26, -1.28]
   Count: 12 Weights: [0.20, -1.94, -2.05, -1.34, -2.07]
   Count: 49 Weights: [0.40, -2.05, -2.07, -1.38, -1.86]
   Count: 3 Weights: [0.20, -2.15, -1.42, -2.00, -1.66]
   Count: 2 Weights: [0.40, -2.31, -1.79, -1.52, -1.43]
   Count: 5 Weights: [0.20, -2.15, -3.71, -0.76, 0.07]
   Count: 12 Weights: [0.40, -2.09, -2.73, -1.79, -1.18]
   Count: 9 Weights: [0.60, -1.79, -2.34, -2.40, -1.20]
   Count: 2 Weights: [0.80, -2.73, -3.57, -0.12, -1.42]
   Count: 29 Weights: [0.60, -3.06, -2.61, -1.83, -0.99]

Count: 54 Weights: [0.80, -3.14, -2.03, -2.19, -1.44]
Count: 4 Weights: [1.00, -3.12, -1.99, -2.28, -1.24]
Count: 2 Weights: [1.20, -2.71, -1.62, -2.88, -1.24]
Count: 2 Weights: [1.40, -3.34, -4.23, 0.25, -1.37]
Count: 3 Weights: [1.60, -2.99, -3.43, -0.70, -1.87]
Count: 2 Weights: [1.80, -2.84, -2.46, -1.65, -3.01]
Count: 2 Weights: [1.60, -2.81, -1.17, -3.33, -2.70]
Count: 5 Weights: [1.80, -3.23, -1.27, -2.99, -2.35]
Count: 2 Weights: [2.00, -3.95, -4.00, 0.54, -2.85]
Count: 21 Weights: [2.20, -4.07, -3.50, -0.04, -3.03]
Count: 3 Weights: [2.40, -4.06, -2.92, -0.75, -3.57]
Count: 3 Weights: [2.20, -3.66, -5.28, -0.67, -1.99]
Count: 4 Weights: [2.40, -3.45, -5.04, -1.19, -1.97]
Count: 16 Weights: [2.20, -3.85, -3.67, -2.82, -1.92]
Count: 9 Weights: [2.00, -3.96, -3.02, -3.44, -1.73]
Count: 3 Weights: [1.80, -4.06, -2.36, -4.06, -1.53]
Count: 12 Weights: [2.00, -4.99, -3.50, -1.86, -1.60]
Count: 44 Weights: [1.80, -5.10, -2.84, -2.48, -1.40]
Count: 7 Weights: [2.00, -4.85, -2.42, -3.12, -1.37]
Count: 23 Weights: [2.20, -4.81, -2.56, -2.94, -1.31]
Count: 3 Weights: [2.40, -4.50, -2.39, -3.41, -1.15]
Count: 13 Weights: [2.60, -4.84, -3.34, -2.16, -1.07]
Count: 48 Weights: [2.80, -4.43, -2.99, -2.76, -1.03]
Count: 59 Weights: [3.00, -4.16, -2.77, -3.23, -0.95]
Count: 8 Weights: [2.80, -4.23, -1.88, -4.14, -0.75]
Count: 3 Weights: [3.00, -5.09, -3.45, -1.77, -1.13]
Count: 10 Weights: [3.20, -4.65, -2.54, -2.77, -1.67]
Count: 3 Weights: [3.00, -4.39, -4.60, -2.18, -0.50]
Count: 41 Weights: [2.80, -4.70, -3.66, -3.76, -0.19]
Count: 4 Weights: [3.00, -4.23, -2.75, -4.76, -0.77]
Count: 1 Weights: [3.20, -4.26, -2.98, -4.48, -0.63]
Count: 10 Weights: [3.40, -4.98, -4.18, -2.46, -0.79]
Count: 59 Weights: [3.20, -5.17, -3.42, -3.39, -0.73]
Count: 14 Weights: [3.40, -4.81, -2.46, -4.42, -1.38]
Count: 24 Weights: [3.60, -5.38, -3.79, -2.32, -1.46]
Count: 43 Weights: [3.40, -5.44, -2.90, -3.24, -1.27]
Count: 9 Weights: [3.60, -5.11, -2.05, -4.14, -1.74]
Count: 73 Weights: [3.80, -5.98, -3.16, -1.96, -1.83]
Count: 2 Weights: [4.00, -5.67, -2.72, -2.58, -1.85]
Count: 20 Weights: [3.80, -5.29, -4.29, -2.25, -1.48]
Count: 128 Weights: [4.00, -5.00, -3.61, -3.05, -1.77]
Count: 6 Weights: [3.80, -4.87, -2.69, -4.72, -1.22]
Count: 15 Weights: [4.00, -5.67, -4.35, -2.21, -1.53]
Count: 17 Weights: [3.80, -5.78, -3.69, -2.83, -1.33]
Count: 5 Weights: [3.60, -5.88, -3.04, -3.45, -1.13]

Count: 6 Weights: [3.80, -5.40, -2.13, -4.45, -1.71]
Count: 11 Weights: [4.00, -6.37, -3.31, -2.25, -1.88]
Count: 27 Weights: [3.80, -6.51, -2.21, -3.92, -1.30]
Count: 52 Weights: [4.00, -6.82, -3.23, -2.58, -1.27]
Count: 16 Weights: [4.20, -6.38, -2.32, -3.58, -1.81]
Count: 9 Weights: [4.40, -6.23, -3.07, -3.25, -1.50]
Count: 1 Weights: [4.60, -5.94, -2.35, -4.06, -1.82]
Count: 15 Weights: [4.80, -6.84, -4.96, -0.65, -2.42]
Count: 9 Weights: [5.00, -6.73, -4.91, -0.82, -2.23]
Count: 3 Weights: [5.20, -6.44, -4.22, -1.62, -2.52]
Count: 12 Weights: [5.00, -5.88, -4.58, -2.98, -2.44]
Count: 142 Weights: [5.20, -5.53, -3.79, -3.93, -2.94]
Count: 21 Weights: [5.40, -5.13, -3.43, -4.52, -2.90]
Count: 19 Weights: [5.20, -4.76, -5.00, -4.18, -2.53]
Count: 47 Weights: [5.00, -4.86, -4.35, -4.80, -2.33]
Count: 6 Weights: [5.20, -5.73, -5.45, -2.61, -2.41]
Count: 3 Weights: [5.00, -6.08, -4.56, -4.26, -2.05]
Count: 17 Weights: [4.80, -6.15, -3.67, -5.17, -1.85]
Count: 98 Weights: [5.00, -6.49, -4.62, -3.93, -1.77]
Count: 3 Weights: [5.20, -6.17, -4.18, -4.55, -1.80]
Count: 7 Weights: [5.00, -6.01, -6.10, -3.80, -0.30]
Count: 55 Weights: [5.20, -5.99, -5.35, -4.69, -1.17]
Count: 26 Weights: [5.00, -5.96, -4.06, -6.36, -0.87]
Count: 1 Weights: [5.20, -6.46, -5.45, -4.60, -0.56]
Count: 41 Weights: [5.40, -7.29, -4.02, -4.61, -1.84]
Count: 3 Weights: [5.20, -7.36, -3.13, -5.53, -1.65]
Count: 1 Weights: [5.00, -7.20, -5.05, -4.77, -0.15]
Count: 56 Weights: [5.20, -6.99, -4.31, -5.60, -0.53]
Count: 22 Weights: [5.40, -7.56, -5.63, -3.50, -0.62]
Count: 66 Weights: [5.20, -7.66, -4.98, -4.12, -0.42]
Count: 84 Weights: [5.40, -7.45, -4.24, -4.95, -0.81]
Count: 11 Weights: [5.20, -7.55, -3.59, -5.57, -0.61]
Count: 26 Weights: [5.40, -7.69, -4.26, -4.75, -0.31]
Count: 16 Weights: [5.60, -7.37, -3.82, -5.37, -0.33]
Count: 26 Weights: [5.80, -7.99, -5.15, -3.26, -0.51]
Count: 31 Weights: [5.60, -8.10, -4.50, -3.88, -0.32]
Count: 2 Weights: [5.80, -7.92, -3.54, -4.85, -1.43]
Count: 7 Weights: [5.60, -7.39, -5.57, -4.58, -0.34]
Count: 193 Weights: [5.40, -7.45, -4.68, -5.50, -0.14]
Count: 2 Weights: [5.60, -8.02, -6.00, -3.40, -0.23]
Count: 10 Weights: [5.80, -8.24, -5.34, -3.67, -0.62]
Count: 42 Weights: [5.60, -8.35, -4.68, -4.29, -0.42]
Count: 4 Weights: [5.40, -8.32, -3.39, -5.96, -0.12]
Count: 58 Weights: [5.60, -8.62, -4.41, -4.62, -0.08]
Count: 147 Weights: [5.80, -8.22, -4.04, -5.23, -0.08]

Count: 3 Weights: [6.00, -8.97, -6.73, -1.71, -0.64]
Count: 6 Weights: [6.20, -8.65, -6.56, -2.17, -0.47]
Count: 1 Weights: [6.40, -8.17, -5.65, -3.17, -1.05]
Count: 20 Weights: [6.60, -7.73, -4.74, -4.16, -1.60]
Count: 48 Weights: [6.80, -7.43, -4.05, -4.97, -1.88]
Count: 6 Weights: [6.60, -7.06, -5.63, -4.64, -1.51]
Count: 2 Weights: [6.40, -6.93, -4.71, -6.31, -0.97]
Count: 6 Weights: [6.60, -7.65, -7.44, -2.78, -1.47]
Count: 20 Weights: [6.40, -8.16, -6.45, -4.06, -1.39]
Count: 5 Weights: [6.60, -7.87, -5.73, -4.87, -1.71]
Count: 118 Weights: [6.80, -7.52, -4.77, -5.90, -2.35]
Count: 14 Weights: [6.60, -7.62, -4.12, -6.52, -2.16]
Count: 12 Weights: [6.80, -8.27, -6.67, -3.40, -2.18]
Count: 111 Weights: [6.60, -8.34, -5.78, -4.32, -1.99]
Count: 8 Weights: [6.40, -8.44, -5.12, -4.94, -1.79]
Count: 185 Weights: [6.60, -8.13, -4.68, -5.56, -1.81]
Count: 50 Weights: [6.80, -7.72, -4.32, -6.15, -1.77]
Count: 5 Weights: [7.00, -8.44, -7.05, -2.63, -2.27]
Count: 7 Weights: [6.80, -8.89, -6.22, -3.68, -2.19]
Count: 4 Weights: [7.00, -8.71, -5.27, -4.65, -3.31]
Count: 70 Weights: [7.20, -8.41, -4.88, -5.26, -3.33]
Count: 15 Weights: [7.00, -8.38, -3.58, -6.93, -3.03]
Count: 34 Weights: [7.20, -9.10, -4.83, -4.88, -3.26]
Count: 47 Weights: [7.00, -8.72, -6.41, -4.55, -2.89]
Count: 26 Weights: [7.20, -8.45, -6.20, -5.02, -2.81]
Count: 133 Weights: [7.40, -8.05, -5.83, -5.62, -2.81]
Count: 113 Weights: [7.20, -8.15, -5.17, -6.24, -2.61]
Count: 30 Weights: [7.40, -7.75, -4.81, -6.83, -2.57]
Count: 10 Weights: [7.20, -7.85, -4.16, -7.45, -2.37]
Count: 8 Weights: [7.00, -7.32, -6.19, -7.18, -1.28]
Count: 6 Weights: [6.80, -7.39, -5.30, -8.10, -1.08]
Count: 72 Weights: [7.00, -8.25, -6.87, -5.73, -1.46]
Count: 106 Weights: [7.20, -7.77, -5.96, -6.73, -2.04]
Count: 50 Weights: [7.00, -7.87, -5.31, -7.34, -1.84]
Count: 6 Weights: [6.80, -7.71, -7.22, -6.59, -0.34]
Count: 3 Weights: [7.00, -7.42, -6.50, -7.40, -0.66]
Count: 2 Weights: [7.20, -8.52, -4.86, -7.34, -1.67]
Count: 6 Weights: [7.40, -9.40, -6.41, -4.95, -1.96]
Count: 19 Weights: [7.60, -9.05, -5.62, -5.90, -2.46]
Count: 13 Weights: [7.80, -8.61, -4.71, -6.89, -3.01]
Count: 22 Weights: [7.60, -8.08, -6.73, -6.63, -1.91]
Count: 12 Weights: [7.40, -8.19, -6.08, -7.24, -1.72]
Count: 6 Weights: [7.60, -8.59, -7.42, -5.44, -1.70]
Count: 57 Weights: [7.80, -8.27, -7.25, -5.90, -1.53]
Count: 60 Weights: [8.00, -7.95, -6.81, -6.53, -1.55]

Count: 14 Weights: [8.20, -7.62, -5.97, -7.43, -2.03]
Count: 52 Weights: [8.40, -8.24, -7.29, -5.33, -2.21]
Count: 122 Weights: [8.60, -8.02, -6.55, -6.16, -2.60]
Count: 133 Weights: [8.40, -8.13, -5.90, -6.78, -2.40]
Count: 76 Weights: [8.20, -7.97, -7.82, -6.02, -0.90]
Count: 50 Weights: [8.40, -9.25, -5.91, -6.01, -2.27]
Count: 15 Weights: [8.20, -9.32, -5.02, -6.93, -2.07]
Count: 10 Weights: [8.40, -10.08, -7.63, -3.54, -2.53]
Count: 35 Weights: [8.20, -10.05, -6.34, -5.21, -2.23]
Count: 32 Weights: [8.40, -9.69, -5.38, -6.24, -2.88]
Count: 43 Weights: [8.20, -9.80, -4.73, -6.85, -2.68]
Count: 1 Weights: [8.00, -9.90, -4.08, -7.47, -2.48]
Count: 84 Weights: [8.20, -10.24, -5.51, -5.89, -2.29]
Count: 71 Weights: [8.40, -9.83, -5.14, -6.50, -2.29]
Count: 108 Weights: [8.20, -9.67, -7.05, -5.74, -0.79]
Count: 178 Weights: [8.40, -9.37, -6.66, -6.35, -0.81]
Count: 9 Weights: [8.20, -9.44, -5.77, -7.26, -0.62]
Count: 11 Weights: [8.40, -10.05, -7.09, -5.16, -0.79]
Count: 25 Weights: [8.20, -10.16, -6.44, -5.77, -0.60]
Count: 104 Weights: [8.40, -9.91, -5.47, -6.83, -1.77]
Count: 34 Weights: [8.20, -10.01, -4.81, -7.45, -1.57]
Count: 63 Weights: [8.00, -9.64, -6.39, -7.12, -1.21]
Count: 31 Weights: [7.80, -9.74, -5.74, -7.73, -1.01]
Count: 29 Weights: [8.00, -10.44, -8.25, -4.70, -1.16]
Count: 18 Weights: [8.20, -10.53, -7.67, -5.06, -1.60]
Count: 8 Weights: [8.40, -10.26, -7.46, -5.53, -1.52]
Count: 1 Weights: [8.60, -9.94, -7.28, -5.99, -1.36]
Count: 51 Weights: [8.80, -9.64, -6.89, -6.60, -1.38]
Count: 37 Weights: [9.00, -9.29, -6.10, -7.55, -1.88]
Count: 2 Weights: [8.80, -9.40, -5.45, -8.17, -1.68]
Count: 44 Weights: [9.00, -9.78, -7.18, -6.34, -1.50]
Count: 74 Weights: [9.20, -9.38, -6.83, -6.93, -1.45]
Count: 39 Weights: [9.00, -9.48, -6.17, -7.55, -1.26]
Count: 30 Weights: [9.20, -10.16, -7.62, -5.26, -1.37]
Count: 152 Weights: [9.40, -9.72, -6.71, -6.25, -1.92]
Count: 79 Weights: [9.60, -9.42, -6.02, -7.06, -2.20]
Count: 10 Weights: [9.40, -9.52, -5.37, -7.68, -2.00]
Count: 30 Weights: [9.60, -10.33, -7.03, -5.17, -2.31]
Count: 14 Weights: [9.80, -9.85, -6.12, -6.16, -2.89]
Count: 22 Weights: [9.60, -9.92, -5.23, -7.08, -2.69]
Count: 39 Weights: [9.40, -9.97, -7.10, -6.34, -1.44]
Count: 221 Weights: [9.20, -10.03, -6.20, -7.26, -1.24]
Count: 5 Weights: [9.40, -9.59, -5.29, -8.25, -1.79]
Count: 18 Weights: [9.60, -9.99, -7.19, -6.31, -1.81]
Count: 12 Weights: [9.80, -9.65, -6.40, -7.26, -2.31]

Count: 2 Weights: [9.60, -9.75, -5.75, -7.88, -2.11]
Count: 44 Weights: [9.80, -10.15, -7.10, -6.08, -2.09]
Count: 58 Weights: [9.60, -10.22, -6.20, -6.99, -1.90]
Count: 39 Weights: [9.40, -10.32, -5.55, -7.61, -1.70]
Count: 25 Weights: [9.20, -10.16, -7.46, -6.85, -0.20]
Count: 96 Weights: [9.40, -9.87, -6.74, -7.66, -0.52]
Count: 16 Weights: [9.20, -9.94, -5.85, -8.58, -0.32]
Count: 117 Weights: [9.40, -10.87, -8.40, -5.23, -0.96]
Count: 11 Weights: [9.60, -10.55, -8.23, -5.70, -0.80]
Count: 10 Weights: [9.40, -10.65, -7.58, -6.32, -0.60]
Count: 13 Weights: [9.20, -10.62, -6.28, -7.99, -0.30]
Count: 7 Weights: [9.40, -11.29, -7.73, -5.70, -0.41]
Count: 40 Weights: [9.60, -11.10, -6.96, -6.68, -1.24]
Count: 81 Weights: [9.80, -10.80, -6.57, -7.30, -1.26]
Count: 46 Weights: [10.00, -10.39, -6.21, -7.89, -1.22]
Count: 69 Weights: [9.80, -10.50, -5.56, -8.51, -1.02]
Count: 25 Weights: [10.00, -11.44, -6.79, -6.23, -1.24]
Count: 150 Weights: [10.20, -10.97, -5.88, -7.23, -1.82]
Count: 30 Weights: [10.00, -11.07, -5.23, -7.84, -1.62]
Count: 77 Weights: [10.20, -11.35, -6.21, -6.55, -1.55]
Count: 30 Weights: [10.00, -11.45, -5.55, -7.17, -1.36]
Count: 72 Weights: [9.80, -11.56, -4.90, -7.78, -1.16]
Count: 22 Weights: [10.00, -12.48, -7.45, -4.44, -1.80]
Count: 7 Weights: [10.20, -12.08, -7.09, -5.03, -1.76]
Count: 67 Weights: [10.40, -11.79, -6.37, -5.84, -2.08]
Count: 35 Weights: [10.60, -11.45, -5.53, -6.75, -2.56]
Count: 75 Weights: [10.40, -11.08, -7.11, -6.42, -2.19]
Count: 12 Weights: [10.60, -10.78, -6.42, -7.22, -2.48]
Count: 51 Weights: [10.40, -10.85, -5.53, -8.14, -2.28]
Count: 123 Weights: [10.60, -11.52, -6.98, -5.85, -2.39]
Count: 97 Weights: [10.80, -11.04, -6.07, -6.85, -2.97]
Count: 7 Weights: [10.60, -10.78, -8.12, -6.26, -1.80]
Count: 160 Weights: [10.80, -10.46, -7.68, -6.88, -1.82]
Count: 10 Weights: [10.60, -10.53, -6.79, -7.80, -1.63]
Count: 15 Weights: [10.80, -10.09, -5.88, -8.79, -2.17]
Count: 5 Weights: [11.00, -10.86, -8.49, -5.40, -2.63]
Count: 137 Weights: [11.20, -10.61, -7.68, -6.33, -3.41]
Count: 61 Weights: [11.00, -10.71, -7.03, -6.95, -3.22]
Count: 4 Weights: [10.80, -10.82, -6.38, -7.56, -3.02]
Count: 13 Weights: [10.60, -10.88, -5.48, -8.48, -2.82]
Count: 68 Weights: [10.40, -10.35, -7.51, -8.21, -1.73]
Count: 217 Weights: [10.20, -10.46, -6.86, -8.83, -1.53]
Count: 30 Weights: [10.00, -10.56, -6.21, -9.45, -1.33]
Count: 41 Weights: [10.20, -10.90, -7.16, -8.21, -1.25]
Count: 89 Weights: [10.40, -10.50, -6.79, -8.81, -1.25]

Count: 15 Weights: [10.60, -11.39, -9.40, -5.39, -1.86]
Test error is 0.0140

(c) [16 points] Implement the average Perceptron. Set the maximum number of epochs T to 10. Report your learned weight vector. Comparing with the list of weight vectors from (b), what can you observe? Report the average prediction error on the test data.

The final weight vector is $[67981.8, -81271.9, -52508.9, -53041.3, -12207.9]$**. The test error is 0.0140. Compared to the list of weight vectors, I can see that the averaged vector is roughly the weighted sum of all the vectors from the voted Perceptron. The final vector is very close to the final weight vector from the standard Perceptron.**

(d) [14 points] Compare the average prediction errors for the three methods. What do you conclude?

The test error for all three methods is roughly the same. However, the test error is slightly lower for the voted and averaged variants. These variants perform better than the standard version when the data is not completely separable.