Quiz Submissions - Quiz 5 - Attempt 2

×

Nhat Le (username: hung.le@mail.mcgill.ca)

Attempt 1

Written: Mar 13, 2019 9:00 PM - Mar 13, 2019 9:04 PM

Submission View

Released: Jan 25, 2019 11:59 PM

View the quiz answers.

Question 1 **1 / 1 point**

[This is a three-part question, so save your work, since it will be relevant to the next two questions on the quiz]

You are training a hard SVM to fit data that follows a logical OR function. I.e., your goal is to fit the following four data points using a hard SVM:

- $\mathbf{x}_1$=[-1, -1], y_1=-1
- $\mathbf{x}_2$=[-1, 1], y_2=1
- $\mathbf{x}_3$=[1, -1], y_3=1
- $\mathbf{x}_4$=[1, 1], y_4=1

Suppose you train a hard linear SVM on this data. What would be the learned weights, ***w***?

You are to assume the following:

1. There is no need to add a bias term to all the data points. Instead you should assume that we are learning the bias term *b*. I.e., you should assume that decision boundary is specified by

$$\text{sign}(\mathbf{w}^\top \mathbf{x} + b)$$

 where ***w*** is a two-dimensional parameter vector and *b* is the bias offset. As a hint, note that the correct solution has *b=1.*
2. We are using the standard Euclidean distance function.
3. We are fitting a basic hard linear SVM.

Hints: Draw or plot the data set. There is no need to actually run a convex optimization solver; you should be able to determine and verify the correct solution by drawing things out and using basic linear algebra.

- ○ w = [1,0]
- ○ w = [-1,-1]
- ○ w = [-1,1]
- ✔ ○ w = [1,1]

▼ Hide Feedback

By drawing out the dataset, one can observe that the line defined by **w**=[1,1] and b=1 (i.e., the line that goes through [0,-1] and [-1,0]) correctly splits the data with the negative point ($\mathbf{x_1}$=[-1, -1]) on one side and the three other positive points on the other.

Moreover, the line **w**=[1,1] with *b=1* is equidistant from the negative point $\mathbf{x_1}$=[-1, -1] and the two positive points, $\mathbf{x_2}$=[-1, 1] and $\mathbf{x_3}$=[1, -1], with the line being a distance of

$$\sqrt{\frac{1}{2}}$$

from each of these three points, which thus must be the **support vectors**. This gives us the **size of the margin** as

$$M = 2\sqrt{\frac{1}{2}}$$

and also verifies that this is the correct decision boundary, since if we were to alter the line by shifting it in any direction it would be closer to one of these three point and thus not optimal (since the margin would be smaller).

Question 2 **1 / 1 point**

[This part 2 of a three-part question, so save your work, since it will be relevant to the question on the quiz]

You are training a hard SVM to fit data that follows an OR function. I.e., your goal is to fit the following four data points using a hard SVM:

- $\mathbf{x_1}$=[-1, -1], y_1=-1
- $\mathbf{x_2}$=[-1, 1], y_2=1
- $\mathbf{x_3}$=[1, -1], y_3=1
- $\mathbf{x_4}$=[1, 1], y_4=1

Suppose you train a hard linear SVM on this data. What would be the size of the margin, *M*? (Note that margin is twice the distance between the separating boundary and the nearest point.)

You are to assume the following:

1. There is no need to add a bias term to all the data points. Instead you should assume that we are learning the bias term *b.* I.e., you should assume that decision boundary is specified by

   $$\mathrm{sign}(\mathbf{w}^\top \mathbf{x} + b)$$

   where ***w*** is a two-dimensional parameter vector and *b* is the bias offset. As a hint, note that the correct solution has *b=1.*
2. We are using the standard Euclidean distance function.
3. We are fitting a basic hard linear SVM.

Hints: Draw or plot the data set. There is no need to actually run a convex optimization solver; you should be able to determine and verify the correct solution by drawing things out and using basic linear algebra.

$$M = 1$$

✔ $$\boldsymbol{M = 2\sqrt{\frac{1}{2}}}$$

$$M = \frac{1}{2}\sqrt{\frac{1}{2}}$$

$$M = \frac{1}{2}$$

View Feedback

Question 3 **1 / 1 point**

[This part 3 of a three-part question]

You are training a hard SVM to fit data that follows an OR function. I.e., your goal is to fit the following four data points using a hard SVM:

- x_1=[-1, -1], y_1=-1
- x_2=[-1, 1], y_2=1
- x_3=[1, -1], y_3=1
- x_4=[1, 1], y_4=1

Suppose you train a hard linear SVM on this data. What points would be the support vectors?

You are to assume the following:

1. There is no need to add a bias term to all the data points. Instead you should assume that we are learning the bias term *b.* I.e., you should assume that decision boundary is specified by
$$\mathrm{sign}(\mathbf{w}^\top \mathbf{x} + b)$$
where ***w*** is a two-dimensional parameter vector and *b* is the bias offset. As a hint, note that the correct solution has *b=1.*
2. We are using the standard Euclidean distance function.
3. We are fitting a basic hard linear SVM.

Hints: Draw or plot the data set. There is no need to actually run a convex optimization solver; you should be able to determine and verify the correct solution by drawing things out and using basic linear algebra.

x_1, x_3, x_4

x_2, x_3, x_4

x_1, x_2, x_3

x_1, x_2, x_4

View Feedback

Question 4 **0 / 1 point**

True or false: Suppose we train a soft linear SVM with *C=20* and we are able to achieve 100% accuracy on a **validation set;** then we can guarantee that a perceptron trained to convergence using the perceptron learning rule and a decaying learning rate will also achieve 100% accuracy on this **validation set.**

True

False

Hide Feedback

Since we only know that the soft SVM has 100% **validation accuracy**, there is no way to know if the **training data** is linearly separable; thus, it is possible that the training data is not linearly separable and we cannot make any assumptions about the solution that the perceptron converges to. So, even though validation data may be linearly separable, there is no way to guarantee that the perceptron will learn a solution from the training set that can achieve 100% accuracy on this validation data.

Question 5 **1 / 1 point**

Consider the primal representation of the soft SVM optimization problem (see, e.g., slide 17 in Lecture 11). Suppose we increase the value of C, which means that we increase the weight of the SVM hinge loss in the optimization (i.e., increasing C means we pay a larger cost for misclassifying training points). Which of the following statements is most applicable in this setting:

Increasing C will tend to improve the model's accuracy on the training set.

Increasing C will tend to decrease the variance of the model and decrease the bias.

Increasing C will tend to improve the model's accuracy on the development/test set.

Increasing C will tend to decrease the variance of the model and increase the bias.

View Feedback

Attempt Score: 4 / 5 - 80 %

Overall Grade (highest attempt)**:** 4 / 5 - 80 %

Done