Quiz Submissions - Quiz 5 - Attempt 1

×

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

Attempt 1

Written: Mar 13, 2019 6:50 PM - Mar 13, 2019 7:43 PM

Submission View

Released: Feb 21, 2019 11:59 AM

View the quiz answers.

Question 1 **0 / 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 an AND 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 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

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

   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.

➡ ◯ w = [1,1]

✖ ◯ w = [-1,1]

◯ w = [-1,-1]

◯ w = [1,0]

▼ Hide Feedback

By drawing out the dataset, one can observe that the line defined by **w**=[1,1] with b=-1 (i.e., the line that goes through [0,1] and [1,0]) correctly splits the data with the positive point ($\mathbf{x}_4$=[1, 1]) on one side and the three other negative points on the other.

Moreover, the line defined by **w**=[1,1] and b=-1 is equidistant from the positive point $\mathbf{x}_4$=[1, 1] and the two negative 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, which is**

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

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 AND 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 by the size of the margin, we mean **twice** the distance between the nearest point and the decision boundary).

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.

$$M = 1$$

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

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

✓ $$M = 2\sqrt{\frac{1}{2}}$$

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 AND 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

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

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.

x_1, x_2, x_3

✓ x_2, x_3, x_4

- x_1, x_3, x_4
- x_1, x_2, x_4

View Feedback

Question 4 **0 / 1 point**

True or false: If a hard linear SVM is able to achieve 100% accuracy on a training data set, then the perceptron algorithm is guaranteed to achieve 100% training accuracy on that same dataset.

- True
- False

Hide Feedback

The linear SVM will only achieve 100% accuracy if there is a linear decision boundary that perfectly separates the data. And the perceptron convergence theorem guarantees that the Perceptron will be able to successfully separate linearly separable data, so since we know that there is a linear decision boundary, the Perceptron algorithm will find it.

Question 5 **0 / 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.

Hide Feedback

Increasing C increases the penalty for misclassifying training points, which means that increasing C will generally lead to higher accuracy on the training set but also a risk of overfitting (i.e., increasing C will generally lead to more variance and less bias in the learned solution).

Attempt Score:	2 / 5 - 40 %
Overall Grade (highest attempt)**:**	2 / 5 - 40 %

Done