2. Hinge Loss

In this project you will be implementing linear classifiers beginning with the Perceptron algorithm. You will begin by writing your loss function, a hinge-loss function. For this function you are given the parameters of your model θ and θ_0. Additionally, you are given a feature matrix in which the rows are feature vectors and the columns are individual features, and a vector of labels representing the actual sentiment of the corresponding feature vector.

Hinge Loss on One Data Sample

1.0/1 point (graded)
First, implement the basic hinge loss calculation on a single data-point. Instead of the entire feature matrix, you are given one row, representing the feature vector of a single data sample, and its label of +1 or -1 representing the ground truth sentiment of the data sample.

Reminder: You can implement this function locally first, and run `python test.py` in your `sentiment_analysis` directory to validate basic functionality before checking against the online grader here.

Available Functions: You have access to the NumPy python library as np; No need to import anything.

```
def hinge_loss_single(feature_vector, label, theta, theta_0):
    """
    Finds the hinge loss on a single data point given specific classifi
    parameters.

    Args:
        feature_vector - A numpy array describing the given data point.
        label - A real valued number, the correct classification of the
            point.
        theta - A numpy array describing the linear classifier.
        theta_0 - A real valued number representing the offset paramete


    Returns: A real number representing the hinge loss associated with
    given data point and parameters.
```

Press ESC then TAB or click outside of the code editor to exit

Correct

```
def hinge_loss_single(feature_vector, label, theta, theta_0):
    """
    Finds the hinge loss on a single data point given specific classification
    parameters.

    Args:
        feature_vector - A numpy array describing the given data point.
        label - A real valued number, the correct classification of the data
            point.
        theta - A numpy array describing the linear classifier.
        theta_0 - A real valued number representing the offset parameter.


    Returns: A real number representing the hinge loss associated with the
    given data point and parameters.
    """
    y = np.dot(theta, feature_vector) + theta_0
    loss = max(0.0, 1 - y * label)
    return loss
```

Test results

See full output

CORRECT

See full output

Solution:

See above for expected answer.

Another possible solution is:

```
def hinge_loss_single(feature_vector, label, theta, theta_0):
    y = theta @ feature_vector + theta_0
    return max(0, 1 - y * label)
```

Here, the `@` operator is shorthand for dot product.

Submit You have used 4 of 25 attempts

Answers are displayed within the problem

The Complete Hinge Loss

1.0/1 point (graded)

Now it's time to implement the complete hinge loss for a full set of data. Your input will be a full feature matrix this time, and you will have a vector of corresponding labels. The k^{th} row of the feature matrix corresponds to the k^{th} element of the labels vector. This function should return the appropriate loss of the classifier on the given dataset.

Available Functions: You have access to the NumPy python library as np, and your previous function as `hinge_loss_single`

```
def hinge_loss_full(feature_matrix, labels, theta, theta_0):
    """
    Finds the total hinge loss on a set of data given specific classifi
    parameters.

    Args:
        feature_matrix - A numpy matrix describing the given data. Each
            represents a single data point.
        labels - A numpy array where the kth element of the array is th
            correct classification of the kth row of the feature matrix
        theta - A numpy array describing the linear classifier.
        theta_0 - A real valued number representing the offset paramete


    Returns: A real number representing the hinge loss associated with
```

Press ESC then TAB or click outside of the code editor to exit

Correct

```
def hinge_loss_full(feature_matrix, labels, theta, theta_0):
    """
    Finds the total hinge loss on a set of data given specific classification
    parameters.

    Args:
        feature_matrix - A numpy matrix describing the given data. Each row
            represents a single data point.
        labels - A numpy array where the kth element of the array is the
            correct classification of the kth row of the feature matrix.
        theta - A numpy array describing the linear classifier.
        theta_0 - A real valued number representing the offset parameter.


    Returns: A real number representing the hinge loss associated with the
    given dataset and parameters. This number should be the average hinge
    loss across all of the points in the feature matrix.
    """
    loss = 0
    for i in range(len(feature_matrix)):
        loss += hinge_loss_single(feature_matrix[i], labels[i], theta, theta_0)
    return loss / len(labels)
```

Test results

See full output

CORRECT

See full output

Solution:

See above for expected answer: we simply sum and take the average.

Another possible solution is:

```
def hinge_loss_full(feature_matrix, labels, theta, theta_0):
    ys = feature_matrix @ theta + theta_0
    loss = np.maximum(1 - ys * labels, np.zeros(len(labels)))
    return np.mean(loss)
```

Here, we use the fact that matrix multiplication is equivalent to stacking the result of dot products to from `ys`, an array of outputs. We then use `np.maximum` to take the element-wise maximum of two arrays.

This solution is more efficient because it makes use of NumPy's fast matrix multiplication capability.

Submit You have used 4 of 25 attempts

Answers are displayed within the problem

Discussion

Hide Discussion

Topic: Unit 1 Linear Classifiers and Generalizations (2 weeks):Project 1: Automatic Review Analyzer / 2. Hinge Loss

Add a Post

Show all posts by recent activity

☑	Unable to understand where to start from Hi TA, I am learning machine learning for the first time. Also, the questions that are being ask...	3
?	Aslo test passing check_hinge_loss_full on loacal but not in grader i have been stuck in this for hours (almost all the day actually) it is driving me nuts. I am using...	3
?	Staff— shouldn't the full_hinge_loss test give a value for feature_matrix not feature_vector? As above... the parameter for hinge_loss_full is feature_matrix ... but the test defines feature_...	6
💬	Doubt in hinge loss Mine two test cases failed 1. expected 308 instead of 306 2. expected 2 intead of 1 Not able t...	2
?	Passed test.py but not the grader? My code for the Hinge Loss Full passed the test.py run however it was marked as incorrect fr...	6
💬	[STAFF] Problems with test.py Hi, I've done the first function: Hinge Lost in One Sample Data, and in the grader it works well...	6
💬	How to use test.py hi i am confused. I am working on my codes and project by manually creating my codes in sh...	13
💬	label - A real valued number Label is said to be a real valued number. My answer is correct but I'm just wondering, at this ...	3
☑	Where can I find the feature_vector? The question says you are given one row representing the feature vector. Where can I find th...	2
💬	Doubts about Complete Hinge Loss For the complete hinge loss, Do I have to update theta each time there is an error ?? ie, when ...	3
?	[STUFF] My code passes tester but fails grader Dear stuff, my code passes the tester but fails the grader for the full hinge loss... This is also ...	3
💬	Confused by the results @staff	3
💬	Do I need to use for loop for the first question?	2

Learn About Verified Certificates