



[Unit 1 Linear Classifiers and Course](#) > [Generalizations \(2 weeks\)](#)  
4. Pegasos Algorithm

[Project 1: Automatic Review](#)  
> [Analyzer](#)

&gt;

### Audit Access Expires May 11, 2020

You lose all access to this course, including your progress, on May 11, 2020. Upgrade by Mar 25, 2020 to get unlimited access to the course as long as it exists on the site. [Upgrade now](#)

## 4. Pegasos Algorithm

Now you will implement the Pegasos algorithm. For more information, refer to the original paper at [original paper](#).

The following pseudo-code describes the Pegasos update rule.

**Pegasos update rule**  $(x^{(i)}, y^{(i)}, \lambda, \eta, \theta)$  :

if  $y^{(i)} (\theta \cdot x^{(i)}) \leq 1$  then

    update  $\theta = (1 - \eta\lambda) \theta + \eta y^{(i)} x^{(i)}$

else:

    update  $\theta = (1 - \eta\lambda) \theta$

The  $\eta$  parameter is a decaying factor that will decrease over time. The  $\lambda$  parameter is a regularizing parameter.

In this problem, you will need to adapt this update rule to add a bias term  $(\theta_0)$  to the hypothesis, but take care not to penalize the magnitude of  $\theta_0$ .

## Pegasos Single Step Update

1.0/1 point (graded)

Next you will implement the single step update for the Pegasos algorithm. This function is very similar to the function that you implemented in **Perceptron Single Step Update**, except that it should utilize the Pegasos parameter update rules instead of those for perceptron. The function will also be passed a  $\lambda$  and  $\eta$  value to use for updates.

**Available Functions:** You have access to the NumPy python library as `np`.

```
1 def pegasos_single_step_update(  
2     feature_vector,  
3     label,  
4     L,  
5     eta,  
6     current_theta,  
7     current_theta_0):  
8     """  
9     Properly updates the classification parameter, theta and theta_0, o  
10    single step of the Pegasos algorithm  
11  
12    Args:  
13        feature_vector - A numpy array describing a single data point.  
14        label - The correct classification of the feature vector.  
15        L - The lambda value being used to update the parameters.
```

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

Correct

```
def pegasos_single_step_update(
    feature_vector,
    label,
    L,
    eta,
    current_theta,
    current_theta_0):
    """
    Properly updates the classification parameter, theta and theta_0, on a
    single step of the Pegasos algorithm

    Args:
        feature_vector - A numpy array describing a single data point.
        label - The correct classification of the feature vector.
        L - The lambda value being used to update the parameters.
        eta - Learning rate to update parameters.
        current_theta - The current theta being used by the Pegasos
            algorithm before this update.
        current_theta_0 - The current theta_0 being used by the
            Pegasos algorithm before this update.

    Returns: A tuple where the first element is a numpy array with the value of
    theta after the current update has completed and the second element is a
    real valued number with the value of theta_0 after the current updated has
    completed.
    """
    mult = 1 - (eta * L)
    if label * (np.dot(feature_vector, current_theta) + current_theta_0) <= 1:
        return ((mult * current_theta) + (eta * label * feature_vector),
                (current_theta_0) + (eta * label))
    return (mult * current_theta, current_theta_0)
```

## Test results

**CORRECT**

[See full output](#)

[See full output](#)

## Solution:

See above for expected answer.

The Pegasos algorithm mixes together a few good ideas: regularization, hinge loss, subgradient updates, and decaying learning rate.

When using a bias, the bias update for a mistake becomes:

$$\theta_0 = \theta_0 + \eta y^{(i)}$$

Submit

You have used 4 of 25 attempts

---

**i** Answers are displayed within the problem

---

## Full Pegasos Algorithm

1.0/1 point (graded)

Finally you will implement the full Pegasos algorithm. You will be given the same feature matrix and labels array as you were given in **Full Perceptron Algorithm**. You will also be given  $T$ , the maximum number of times that you should iterate through the feature matrix before terminating the algorithm. Initialize  $\theta$  and  $\theta_0$  to zero. For each update, set  $\eta = \frac{1}{\sqrt{t}}$  where  $t$  is a counter for the number of updates performed so far (between 1 and  $nT$  inclusive). This function should return a tuple in which the first element is the final value of  $\theta$  and the second element is the value of  $\theta_0$ .

**Note:** Please call `get_order(feature_matrix.shape[0])`, and use the ordering to iterate the feature matrix in each iteration. The ordering is specified due to grading purpose. In practice, people typically just randomly shuffle indices to do stochastic optimization.

**Available Functions:** You have access to the NumPy python library as `np` and `pegasos_single_step_update` which you have already implemented.

```
1 def pegasos(feature_matrix, labels, T, L):
2     """
3     Runs the Pegasos algorithm on a given set of data. Runs T
4     iterations through the data set, there is no need to worry about
5     stopping early.
6
7     For each update, set learning rate = 1/sqrt(t),
8     where t is a counter for the number of updates performed so far (be
9     and nT inclusive).
10
11     NOTE: Please use the previously implemented functions when applicab
12     Do not copy paste code from previous parts.
13
14     Args:
15         feature_matrix - A numpy matrix describing the given data. Each
```

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

Correct

```
def pegasos(feature_matrix, labels, T, L):
    """
    Runs the Pegasos algorithm on a given set of data. Runs T
    iterations through the data set, there is no need to worry about
    stopping early.

    For each update, set learning rate = 1/sqrt(t),
    where t is a counter for the number of updates performed so far (between 1
    and nT inclusive).

    NOTE: Please use the previously implemented functions when applicable.
    Do not copy paste code from previous parts.

    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.
        T - An integer indicating how many times the algorithm
            should iterate through the feature matrix.
        L - The lambda value being used to update the Pegasos
            algorithm parameters.

    Returns: A tuple where the first element is a numpy array with the value of
    the theta, the linear classification parameter, found after T
    iterations through the feature matrix and the second element is a real
    number with the value of the theta_0, the offset classification
    parameter, found after T iterations through the feature matrix.
    """
    (nsamples, nfeatures) = feature_matrix.shape
    theta = np.zeros(nfeatures)
    theta_0 = 0
    count = 0
    for t in range(T):
        for i in get_order(nsamples):
            count += 1
            eta = 1.0 / np.sqrt(count)
            (theta, theta_0) = pegasos_single_step_update(
                feature_matrix[i], labels[i], L, eta, theta, theta_0)
    return (theta, theta_0)
```

## Test results

CORRECT

See full output

See full output

Submit

You have used 1 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 / 4. Pegasos Algorithm

Add a Post

Show all posts	by recent activity
 <u>[Staff] Pegasus full runs locally but only 0.2 on the grader :(</u> <u>Could you look at my code? I am using the debugger, but since locally works fine I fail to see ...</u>	1
 <u>Full pegasos - test.py PASS, not online grader</u> <u>As other have said already, my Full pegasos implementation code works just fine locally, how...</u>	3
 <u>Is there a problem with my grader?</u>	8
 <u>[STAFF] Pegassos working on test.py but failing online test - Rounding issues ?</u> <u>Hi there, I manage to implement the Pegasos single update and the test.py working however ...</u>	6
 <u>[STAFF] message: Problem running your solution</u> <u>My code is running in my local python, but the grader is giving me the following two message...</u>	2
 <u>crashed.</u> <u>I found the right answer for Q-4.1 "Pegasos Single Step Update" but something wrong happe...</u>	2
 <u>single step and &gt;1 case</u> <u>Help! I can't seem to get the &gt;1 case to work. I am hand calculating the answer, but my outpu...</u>	6

 <u>what is n in "between 1 and <math>nT</math> inclusive"?</u>	6
<u>it's a silly question but I think it's where I'm hung up. my function passes the test.py tests but ...</u>	
 <u>[staff help] label&gt;1 can not pass,i tried 25 times still can not work it out</u>	4
 <u>[STAFF] SQRT problem in Full Pegasos Algorithm</u>	4
<u>Hj, I use math.sqrt(eta) in my code, the result is incorrect, the message is "NameError: name '..."</u>	
 <u>How to understand the update of theta_0</u>	12
<u>"In this problem,.you will need to adapt this update rule to add a bias term( <math>\theta_0</math>.) to the hypot...</u>	
 <u>Pegasus Single Step Error with Test for prediction * label &gt; 1</u>	12
<u>Haven't seen anyone else post this but for some reason I'm just getting an issue with Test for...</u>	
 <u>Small error in the "Pegasus" algorithm: should be <math>\theta_0</math> instead of <math>\theta_1</math> in the "Pegasus"...</u>	

Learn About Verified Certificates

© All Rights Reserved