

MITx 6.86x

Machine Learning with Python-From Linear Models to Deep Learning

Discussion Course **Progress** Dates Resources

★ Course / Unit 1. Linear Classifiers and Generalizatio... / Project 1: Automatic R



4. Pegasos Algorithm

☐ Bookmark this page

Project due Mar 1, 2023 08:59 -03 Completed

Now you will implement the Pegasos algorithm. For more information, refer to the origin paper.

The following pseudo-code describes the Pegasos update rule.

Pegasos update rule
$$\left(x^{(i)},y^{(i)},\lambda,\eta,\theta
ight)$$
 : if $y^{(i)}\left(heta\cdot x^{(i)}
ight)\leq 1$ then update $heta=(1-\eta\lambda)\, heta+\eta y^{(i)}x^{(i)}$ else: update $heta=(1-\eta\lambda)\, heta$

The η parameter is a decaying factor that will decrease over time. The λ parameter is a

In this problem, you will need to adapt this update rule to add a bias term (θ_0) to the hynot to penalize the magnitude of θ_0 .

Pegasos Single Step Update

1.0/1 point (graded)

Next you will implement the single step update for the Pegasos algorithm. This function function that you implemented in **Perceptron Single Step Update**, except that it should parameter update rules instead of those for perceptron. The function will also be passe 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
                   theta,
 7
                   theta_0):
 8
 9
           Updates the classification parameters `theta` and `theta_0` via a
10
           step of the Pegasos algorithm.
                                            Returns new parameters rather than
11
           modifying in-place.
12
13
          Args:
14
                   `feature_vector` - A numpy array describing a single data
15
                   `label` - The correct classification of the feature vector
```

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

1.0/1 point (graded)

Finally you will implement the full Pegasos algorithm. You will be given the same feature as you were given in **Full Perceptron Algorithm**. You will also be given the maximum you should iterate through the feature matrix before terminating the algorithm. Initialize each update, set where is a counter for the number of updates performed so

inclusive). This function should return a tuple in which the first element is the final value element is the value of .

Note: Please call <code>[get_order(feature_matrix.shape[0])]</code>, and use the ordering to iter each iteration. The ordering is specified due to grading purpose. In practice, people type shuffle indices to do stochastic optimization.

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

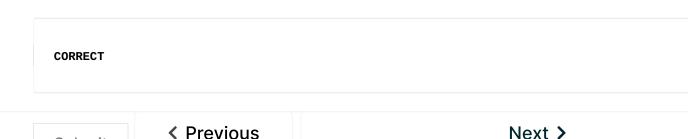
```
1 def pegasos(feature_matrix, labels, T, L):
 2
 3
          Runs the Pegasos algorithm on a given set of data. Runs T iteration
 4
          through the data set, there is no need to worry about stopping ear.
 5
          each update, set learning rate = 1/sqrt(t), where t is a counter for
 6
          number of updates performed so far (between 1 and nT inclusive).
 7
 8
          NOTE: Please use the previously implemented functions when applica
 9
          not copy paste code from previous parts.
10
11
          Args:
12
                   `feature_matrix` - A numpy matrix describing the given data
13
                           represents a single data point.
14
                   `labels` - A numpy array where the kth element of the arra
15
                           correct classification of the kth row of the featu
```

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

Correct

Test results

Submit





- Test.py shows FAIL despite having correct answer
- For full Pegasos, don't wanna use i to calculate the times used Because i is the index of the samples after shuffling, doesn't represent the numeric order T_T
- pegasos algorithm | please read this first
 - Community TA
- update of theta_0 I need some hints for the theta_0's update rule. How should I think this?
- Basic explanation of "take care not to penalize the magnitude of theta_0" I was able find the necessary formula for the algorithm using intuition (and, I must admit, a bit of trial-and-er
- Stuck at Test for prediction * label > 1 for first part of question I keep getting the wrong answer for that part of the question. Not sure what the error is? Is it due to penalisa
- Pegasus values for > 1 incorrect I can't see what is wrong ⇒ in the else argument, I am returning the current_theta_0 value and the current_th
- pegasos single step update, TypeError?

Legal

Terms of Service & Honor Code

Privacy Policy

Accessibility Policy

Trademark Policy

Sitemap.

Cookie Policy

Do Not Sell My Personal Information

Connect

Blog

Contact Us

Help Center

Security

Media Kit













