



MITx 6.86x

Machine Learning with Python-From Linear Models to Deep Learning

[Course](#)

[Progress](#)

[Dates](#)

[Discussion](#)

[Resources](#)



[Course](#) / [Unit 1. Linear Classifiers and Generalizatio...](#) / [Project 1: Automatic R](#)

< Previous



8. Parameter Tuning

Bookmark this page

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

You finally have your algorithms up and running, and a way to measure performance! By choosing the right values for the hyperparameters like T and λ should have. In this section, you'll tune these hyperparameters to maximize the performance of each model.

One way to tune your hyperparameters for any given Machine Learning algorithm is to try out all the possible combinations of values. If your hyperparameters can be any real number, this is an infinite search space. To limit the search to some finite set of possible values for each hyperparameter. For efficiency, you might want to tune one individual parameter, keeping all others constant, and then move on to the next. Compared to a full grid search there are many fewer possible combinations to check, and you're doing for the questions below.

In **main.py** uncomment Problem 8 to run the staff-provided tuning algorithm from **utils.py**. For this assignment, please try the following values for T : [1, 5, 10, 15, 25, 50] and the following values for λ : [0.01, 0.1, 1, 10]. For pegasos algorithm, first fix $\lambda = 0.01$ to tune T , and then use the best T to tune λ .

Performance After Tuning

6/7 points (graded)

After tuning, please enter the best T value for each of the perceptron and average perceptron algorithms. Also enter both the best T and λ for the Pegasos algorithm.

Note: Just enter the values printed in your main.py. Note that for the Pegasos algorithm, the values should reflect the best combination of T and λ .

For the **perceptron** algorithm:

$T =$

25.0000



With validation accuracy =

0.7960



For the **average perceptron** algorithm:

$T =$

25.000



With validation accuracy =

[Submit](#)

You have used 1 of 20 attempts

Accuracy on the test set

1/1 point (graded)

After you have chosen your best method (perceptron, average perceptron or Pegasos) this classifier to compute testing accuracy on the test set.

We have supplied the feature matrix and labels in `main.py` as `test_bow_features` a

Note: In practice the validation set is used for tuning hyperparameters while a heldout benchmark used to compare disparate models that have already been tuned. You may using a validation set don't always align with those of the test set, and this is to be exp

Accuracy on the test set :

[Submit](#)

You have used 3 of 20 attempts

The most explanatory unigrams

10/10 points (graded)

According to the largest weights (i.e. individual values in your vector), you can find out the most impactful ones in predicting **positive** labels. Uncomment the relevant part in `utils.most_explanatory_word`.

Report the top ten most explanatory word features for positive classification below:

Top 1 : Top 2: Top 3: Top 4: Top 5: Top 6: Top 7:

Discussion

Topic: Unit 1. Linear Classifiers and Generalizations (2 weeks):Project 1: Automatic Review Analyzer / 8. Parameter Tuning

Show all posts ▾

💬 Memory error

I got a memory error on Pycharm, my laptop have only 4GB RAM and no GPU numpy.core._exceptions.Memo

? which theta to choose for the negative labels

Hello All, From the code, we are able to select the optimized theta to check the most positive labels. I wonder

? Are we supposed to guess the correct order of the words??

Are we supposed to guess the correct order of the words??

💬 For the last question

- remember you get the theta from the training set, not the test one - The classifier function returns a tuple (

💬 I guess epsilon is somewhat important

It seems the result is not very robust. I get some incorrect and some correct for the accuracy. the correct a

💬 finding unigrams that were the most impactful in predicting negative labels

How should I approach this?

💬 Predicting positive labels

I'm getting 1 out of 10 here. There is no reference to "positive" labels in the most explanatory word features in

💬 Finding best T and L for pegasos

I experimented with loads of Ts and Ls, the note said for pegasos the best might not be the combinatio print

? Exlanatory unigrams

In the main.py we will pass the parameters for our best classifier to get theta. The feature_matrix (parameter

💬 Just a hint for last question ;)

Just in case any of you are stuck like me in the last question, read the comments in line 120 !

? Accuracy on the test set

Hello, in the "Accuracy on the test set" problem, we are prompted to write code to the main file right (as show

? Is it necessary to change "feature_matrix[i, indices_by_word[word]] += 1" into "feature_matrix[indices_by_word[word]] = 1"?

This change enabled me to solve the Baseline Accuracy. So, should I change it back to "feature_matrix[i, indi

? Last question - utils.py Value Error

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](#)



© 2023 edX LLC. All rights reserved.

深圳市恒宇博科技有限公司 [粤ICP备17044299号-2](#)