8. Parameter Tuning

Extension Note: Project 1 due date has been extended by 2 days to **July 4 23:59UTC** (Note the UTC time zone).

You finally have your algorithms up and running, and a way to measure performance! But, it's still unclear what values 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 perform a grid search over all the possible combinations of values. If your hyperparameters can be any real number, you will need to limit the search to some finite set of possible values for each hyperparameter. For efficiency reasons, often you might want to tune one individual parameter, keeping all others constant, and then move onto the next one; Compared to a full grid search there are many fewer possible combinations to check, and this is what you'll be doing for the questions below.

In **main.py** uncomment Problem 8 to run the staff-provided tuning algorithm from **utils.py**. For the purposes of this assignment, please try the following values for T: [1, 5, 10, 15, 25, 50] and the following values for λ [0.001, 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

7/7 points (graded)

After tuning, please enter the best T value for each of the perceptron and average percepton algorithms, and 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 result does not reflect the best combination of T and λ.

For the **perceptron** algorithm:

$T =$ 25 ✓ **Answer:** 25

With validation accuracy =

0.794 ✓ **Answer:** 0.7940

For the **average perceptron** algorithm:

$T =$ 25 ✓ **Answer:** 25

With validation accuracy =

0.8 ✓ **Answer:** 0.8000

For the **pegasos** algorithm:

$T =$ 25 ✓ **Answer:** 25

$\lambda =$ 0.01 ✓ **Answer:** 0.01

With validation accuracy =

0.806 ✓ **Answer:** 0.8060

Solution:

- The best Perceptron T should be 25 with validation accuracy = 0.7940
- The best Average Perceptron T should be 25 with validation accuracy = 0.8000
- The best Pegasos T should be 25 with validation accuracy = 0.8060
- The best Pegasos λ should be 0.01 with validation accuracy = 0.8060
- You may notice that all the algorithms result in similar performance. It is due to the fact that they are all linear models.
- In project 2, you will experience how to properly choose model family can lead to betterperformance.

Submit You have used 2 of 20 attempts

Answers are displayed within the problem

Accuracy on the test set

1/1 point (graded)
After you have chosen your best method (perceptron, average perceptron or Pegasos) and parameters, use 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` and `test_labels`.

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

Accuracy on the test set :

0.802 ✓ **Answer:** 0.8020

Submit You have used 1 of 20 attempts

Answers are displayed within the problem

The most explanatory unigrams

10/10 points (graded)
According to the largest weights (i.e. individual i values in your vector), you can find out which unigrams were the most impactful ones in predicting **positive** labels. Uncomment the relevant part in `main.py` to call `utils.most_explanatory_word`.

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

Top 1 : delicious ✓ **Answer:** delicious

Top 2: great ✓ **Answer:** great

Top 3: ! ✓ **Answer:** !

Top 4: best ✓ **Answer:** best

Top 5: perfect ✓ **Answer:** perfect

Top 6: loves ✓ **Answer:** loves

Top 7: wonderful ✓ **Answer:** wonderful

Top 8: glad ✓ **Answer:** glad

Top 9: love ✓ **Answer:** love

Top 10: quickly ✓ **Answer:** quickly

Also experiment with finding unigrams that were the most impactful in predicting negative labels.

Submit You have used 1 of 20 attempts

ℹ Answers are displayed within the problem

Discussion

Show Discussion

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