

Homework Assignment #7

I found this to be the hardest homework assignment I have had in the class. I used the code provided on Moodle as a starting point, and attempted (with not much success) to complete the gradient and value functions. Although my code does work, I did not have enough time to actually evaluate to accuracy of the classifier.

The gradient function first calls another function that calculates the expectation of each feature. The constraint of each feature has been obtained in the `train_maxent` function and can be accessed from gradient, since it is declared as a global variable. The function then calculates the gradient, added the Gaussian prior, and negate the obtained value for each feature, since the function `fminNCG` minimizes instead of maximizing. It returns a Numeric array of the gradients of each feature. I modified the function value by adding Gaussian prior calculation. Again, since `fminNCG` minimizes instead of maximizing, value returns the negation of its actual return value.

I ran into multiple issues trying to implement the maximum entropy classifier. First, I had an issue getting Numeric to work. Even after Numeric was successfully installed, I still had no clue how to calculate gradient and Gaussian prior. I think the most confusing part was trying to make sense of optimize. Later, I found out that I should not have bothered to even change anything in optimize. It also took me a while before I figured out what return type is expected from gradient. I'm still not sure I understand when different functions in the program gets called. I also did the expectation calculation incorrectly and kept getting `OverflowError: math range error` until I talked to a TA today and figured out what I did wrong.

Even though I have a working program, I am still not sure how to handle a word that never occurred in the training data. My program currently just ignores the word (and skips to the next word). I also never trained the classifier using all the training data provided on Moodle because when I tried doing so, it took too long. Instead, I used partial training data and tested the classifier only on a single document, each time the program runs.

If I had more time to finish the assignment, I would evaluate the accuracy of the classifier and compare it to the accuracy of a naive Bayes classifier. I have manually run the program multiple times to test on several data (less than five, since I ran out of time, so it doesn't really say anything) and the classifier was able to accurately classify all the test data.