Homework 2 - LOGREG
2015 Spring, Machine Learning
Choong-Wan Woo
February 2, 2015

Analysis 1.

What is the role of the learning rate?

As shown in **Figure 1**, different step sizes (i.e., learning rates) have substantial influences on training and testing accuracy in stochastic gradient ascent of logistic regression models. The plots show large (e.g., 1.0) or small (e.g., 0.001) learning rates have negative effects on both training and testing accuracy. In our analysis, 0.1 or 0.01 showed high accuracy in both training and testing data.

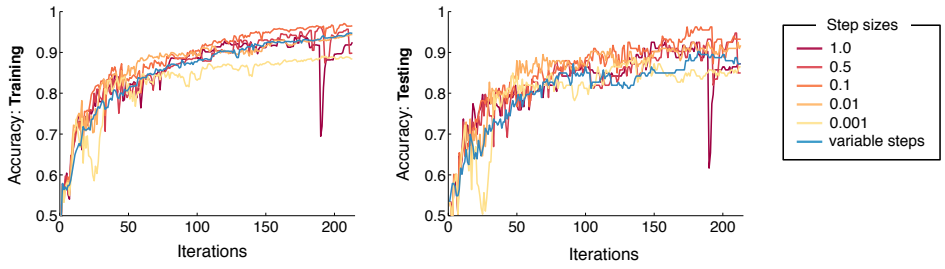

Figure 1. The effects of learning rate (i.e., step size of gradient ascent) on accuracy.

Analysis 2.

How many passes over the data do you need to complete?

As **Figure 2** demonstrate, I tested one to five passes, and all of them seem okay given their high accuracy results. However, **Figure 2** shows an interesting pattern of results: The accuracy in training data kept increasing, whereas the accuracy in testing data decreased with five passes. Because this pattern could be due to an overfitting when five passes were used, four passes might be ideal.

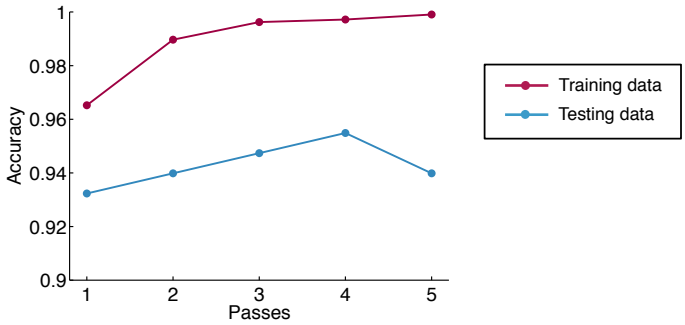

Figure 2. The effects of passes on accuracy.

Analysis 3.

What words are the best predictors of each class? How did you find them?

From my analysis, the best predictor for the documents about baseball was **"hit"**, and the best predictor for hockey documents was **"hockey"**. To identify the best words, I found the words that have the maximum positive (for baseball) and negative (for hockey) beta values. What I mean by the maximum negative value is the biggest absolute value among the negative values.

Analysis 4.

What words are the poorest predictors of classes? How did you find them?

The poorest predictor for the documents about baseball was **"tandem"**, and the poorest predictor for hockey documents was **"kicked"**. To identify the poorest words, I searched the words that have the minimum positive (for baseball) and negative (for hockey) beta values. What I mean by the minimum negative value is the smallest absolute value among negative values.

Extra.

Use a schedule to update the learning rate, and show the effect in your analysis document

In my submission of **logreg.py**, I used $\frac{2}{\sqrt{iteration}}$ to update the learning rate over iterations (which is activated by adding the argument, **--var_step True**). The results of using the argument are shown in **Figure 1** using a blue line. In The results were not better than a fixed learning rate (e.g., 0.1).