Analysis of Self Implemented Logistic Regression

1. Introduction

Logistic regression is a common machine learning technique used for both binary and multiclass classification problems. In this paper, I demonstrate how logistic regression can be used to identify the topic of a document. Specifically, I use a technique called stochastic gradient decent to build a fast and accurate model for classification. Analysis of the forulas used will show how large beta values are used heavily in predicting a class.

2. Formulas

2.1. The Sigmoid Fuction

The sigmoid function is used heavily in logistic regression because it sqeezes any real value into a number between 0 and 1 (like a probability) and is easily differentiable. To calculate the probability of a given feature vector, we feed the dot product of beta values and the training examples into the sigmoid function.

2.2. The Delta

On each beta update, we change each beta value by some amount delta. We know that delta needs to move beta in the right direction, so we take the actual label and subtract the calculated probability and then we multiply by the number of times that feature appears in the feature vector. After multiplying this value by the step function, we add it to the beta vector, allowing us to shift the beta vector is the right direection based on the training example.

2.3. Regularization

In order to ensure that our model generalizes well to unseen data, we need to prevent overfitting. The most common way to prevent overfitting in logistic regression is by using regularization, or penalizing large beta values. The way we do this is by shrinking large beta values. We define a shrinkage value that gets larger based on the iteration. We then take all of the features that we just updated and multiply them by our shrinkage number raised to the gap power (how long it has been since we last updated). This will effectively scale the beta value down if it is very large.

3. Questions

3.1. What is the learning rate?

The learning rate determines how quickly our model will converge. In this program, the learning rate is determined in our lambda function step and is set to 0.05.

3.2. How many passes over the data did we need?

In this program, we are able to converge to a reasonable answer fairly quickly, using 1 or 2 passes over the data. Heuristically, I would guess the sweetspot is somewhere between 5-10 passes with mu being less than 0.1. Any more and we would risk overfitting.

3.3. What are the best predictors?

The best predictors of each class, are the highest weighted beta values for that class. Hockey was denoted by negative beta values and when we sorted them we found that the highest predictors were:

Feature	Beta
hockey	-2.39701629657
playoffs	-1.44564218453
golchowy	-1.17516769744
ice	-1.03574704859
next	-1.02160900872
goals	-1.01665832771
pick	-1.01647140833
playoff	-0.976664795926
points	-0.954740265196
biggest	-0.934753162921

And for baseball, the beta values were positive:

Feature	Beta
runs	1.43294142746
hit	1.10998401766
pitching	1.03061549829
baseball	1.02607170122
catcher	0.854081642361
ball	0.793975224712
anyone	0.784188784459
saves	0.764849612496
run	0.697562386451
book	0.684478585989

Interestingly, the words that best predicted baseball were verbs associated with the activity, with the actual word baseball coming in at number 4.

3.4. What are the worst predictors?

The worst predictors for a given class are the best predictors for the other class. In this case the worst predictors for hockey are the best predictors for baseball and vice versa.

3.5. What happens if mu is 0?

If mu is zero, our regularization will have no effect. This means a lot of things including that we might risk overfitting. It also means it takes less passes for our model to converge.

4. Conclusion

In this lab, we used a logistic regression model to predict whether a document was talking about hockey or baseball. We used this model to predict which unigram features were most predictive of each class, and created a highly accurate classifier to go along with our testing and training data. We learned about regularization and stochastic gradient decent. I was most supprised at how accurate the model became even on a small dataset.