Evaristo Koyama

ek4ks

For this assignment, I have implemented the logistic regression. First, it looks at the names file. Every discrete value and each continuous variable is a feature. The program keeps track of its index and whether a feature is numeric. The program then trains using the dataset. It uses gradient descent to minimize the cost function. The number of iterations and the learning rate can be set by the user. Increasing the iterations makes the predictions more accurate, but it also increases the runtime. Increasing the learning rate usually allows the program to make better predictions, but if it is too high, then it ends up ruining the hypothesis functions. The theta value starts out with assuming no features has an effect on the result. For each example, it interprets the value of each variable and calculates its prediction and error. With these values, we calculate the new theta. To make a prediction, we interpret the values given by the testing example and then calculate our prediction. We guess depending on whether our prediction value is over some threshold value.

My program has an accuracy of about 70%, which is better than the classifier that makes prediction based on a coin flip which has an accuracy of about 50%. The program has some idea on the effect of each variable on the classification.

I tweaked with the learning rate and the iterations. First I set the learning rate as high as possible without it breaking. I changed the number of iterations to get the weighting. There is a very clear rise in accuracy as you increase the number of iterations. There is a larger increase in accuracy as you go from very little iterations to little iterations than from many iterations to more iterations. Next, I kept iterations constant and tweaked the learning rate. This is similar to above, but there is a point where it breaks the program. It started getting accuracies less than 40%.

Given more time and resources, there would have been some optimizations that I wanted to make. One is to make graphs to find a better hypothesis function. I assumed a very simple model. Another is to preemptively fill in some theta values. The names files contained variables that do not even appear once. Sometimes you can assume some things about certain variable. One was not-working. You can assume that the person would be making less than 50K. Finally, there were some variables that could have been expressed differently. Capital gain is usually 0, but if it is greater than 0, then the person more than likely makes more than 50K.