DS222: Assignment-2

Rahul Chittimalla

1. Local Logistic regression

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. We need the output of the algorithm to be class variable, i.e 0-no, 1-yes. Therefore, we are squashing the output of the linear equation into a range of [0,1]. To squash the predicted value between 0 and 1, we use the sigmoid function.

$$z = \theta_0 + x_1 \cdot \theta_1 + x_2 \cdot \theta_2 ...$$
$$h = g(z) = \frac{1}{1 + e^{-z}}$$

The cost function used for logistic regression is

$$J = \frac{-1}{m}[\sum y^{(i)} log h_\theta^{(i)} + (1-y)^{(i)} log(1 - h_\theta^{(i)})]$$

If w is the weight parameter then the gradient of the loss function with respect to w is given by

$$J = \frac{-1}{m} X^T (h - y)$$

where X an mxd matrix, m are the number of instances and d is the dimension of features. Classification on DB-Pedia dataset is a multi-class, multi-label problem. So, we need to adapt the logistic regression to this variant. One way to do this is a one-vs-rest model where we build 50 classifiers. Each of the 50 classifiers give the probability of the document belonging to that class, we take the the higher probable value to classify. Data preprocessing is an important step to get better results using the model. Removing the useless symbols, stop words converting the text to lower case improves the model performance. The next step after data cleaning is feature extraction. Tf-idf can be used to extract features from raw text data where tf is *term frequency* and idf is *inverse document frequency*. Term frequency in multiplied by idf as to add more importance to the words which appear in less number of documents and penalize those words which occur in more documents (as those words dont play much role is differentiating the documents).

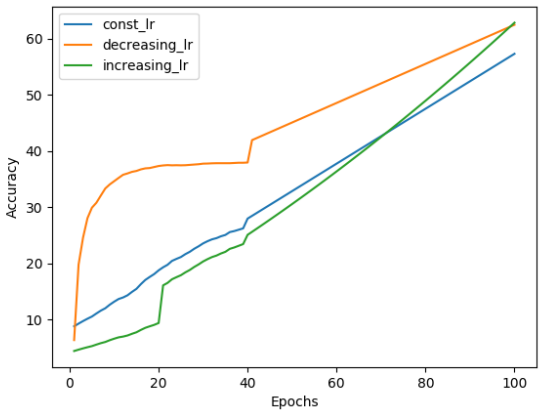

Figure 1. Accuracy using local logistic regression

$$tf_{t,d} = \begin{cases} 1 + log_{10} count(t,d) & \text{if count(t,d)} < 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$idf_t = log_{10}(\frac{N}{df_t})$$

where N is the number of total documents and df_t is the number of documents in which the term t appears.

$$w_{t,d} = tf_{t,d} \cdot idf_t$$

If the vocabulary size is $|V|$ then the feature vector of each document has the size of $|V|$. The model is trained in three modes:

- constant learning rate
- increasing learning rate
- decreasing learning rate

The accuracy for all the three different models are plotted in figure 1. For the constant learning rate model, the learning rate is set to 0.1. For the decreasing learning rate model, the

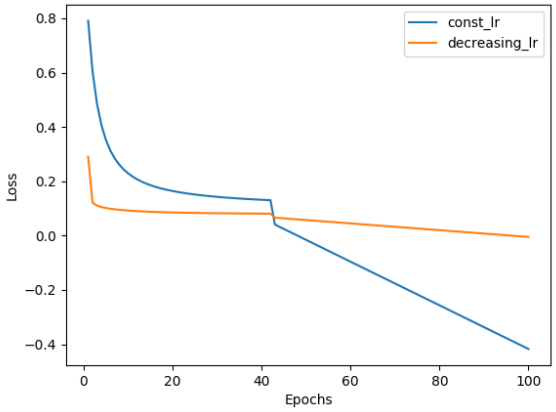

Figure 2. Loss using local logistic regression

initial learning rate is set to 5 and will start to decrease after every epoch by 5% until it reaches 0.1. For the increasing learning rate model, the initial learning rate is 0.0001 and it increases by 5% after every epoch. We can see from figure 1 that the accuracy of decreasing learning rate model is pretty high compared to other models, this is due to the fact that with larger step size in the initial stages the optimizer takes longer steps towards the minimum. As it goes towards the minimum, the learning rate reduces and takes smaller steps. The loss for local logistic regression is plotted in figure 2