ML:hw6

sblanco2465

April 2024

1 Logistic Regression

1. we just apply the rules of calculus (chain rule) to get the following answer. Each step line indicates a step in calculation.
   $log(1 + exp(-y_i w^T x_i))$
   $\frac{1}{1+exp(-y_i w^T x_i)} * \frac{d}{dw}(1 + exp(-y_i w^T x_i))$
   $\frac{1}{1+exp(-y_i w^T x_i)} * exp(-y_i w^T x_i) * -y_i x_i$
   $\frac{exp(-y_i w^T x_i)}{1+exp(-y_i w^T x_i)} * -y_i x_i$

2. The objective function would be to find the weight vector that minimizes the loss function. we are given the loss function with a set of examples. In the given loss function we add up the loss for each example. Since this question asks for the loss of a single example we can get rid of the summation.
   $\arg\min_w [log(1 + exp(-y_i w^T x_i)) + \frac{1}{\sigma^2} w^T w]$

3. we start with the derivative that we calculated in question 1
   $\frac{exp(-y_i w^T x_i)}{1+exp(-y_i w^T x_i)} * -y_i x_i$
   we then calculate the derivative of $\frac{1}{\sigma^2} w^T w$
   $w^T w$ is simply the sum of each of its terms squared which means its derivative is $2w$. Multiplying by our constant we get $\frac{2}{\sigma^2} w$
   for a final answer of:
   $\frac{exp(-y_i w^T x_i)}{1+exp(-y_i w^T x_i)} * -y_i x_i + \frac{2}{\sigma^2} w$
   $w = w - learningRate * (\frac{exp(-y_i w^T x_i)}{1+exp(-y_i w^T x_i)} * -y_i x_i + \frac{2}{\sigma^2} w)$

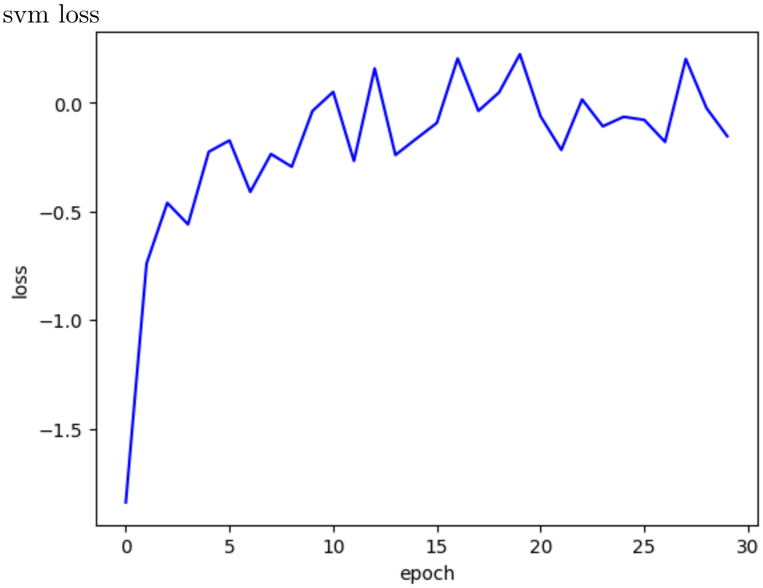
4. ```
   given a set of examples (x_i, y_i)
   w= random initial vector
   for i in T: # T is total number of rounds, i is current round
       pick a random example out of the set of examples (x_i, y_i)
       gradient = (exp(-y_iw^Tx_i)/1+exp(-y_iw^Tx_i)) * -y_ix_i + (2/sigma^2)w
       # this is the gradient calculated in step 3
       w = w - gradient * learning_rate
   ```
   ```

2 Experiments

2.1 SVM

- design decisions: I chose to implement svm in python using the pandas library. I utilized dataframes and numpy for data processing and representation.
- data representation: I chose to represent the data as numpy.array((x0, y0), (x1, y1) ... (xn, yn)). I did this so that I could make my algorithm in code look like the one in the slides.
- I also chose 30 epochs. This decision was made based on the f1 score not going up much beyond 30 epoch and even going down around 60 epoch.

svm loss



2.2 Logistic regression

- design decisions: I did everything that I did with SVM above. This includes decaying the learning rate. I found that this improved performance.
- data representation: I use the same code for data representation for Logistic regression as I did for SVM.
- epoch: I chose 30 epoch on logistic regression for a similar reason as I did for SVM

logistic loss

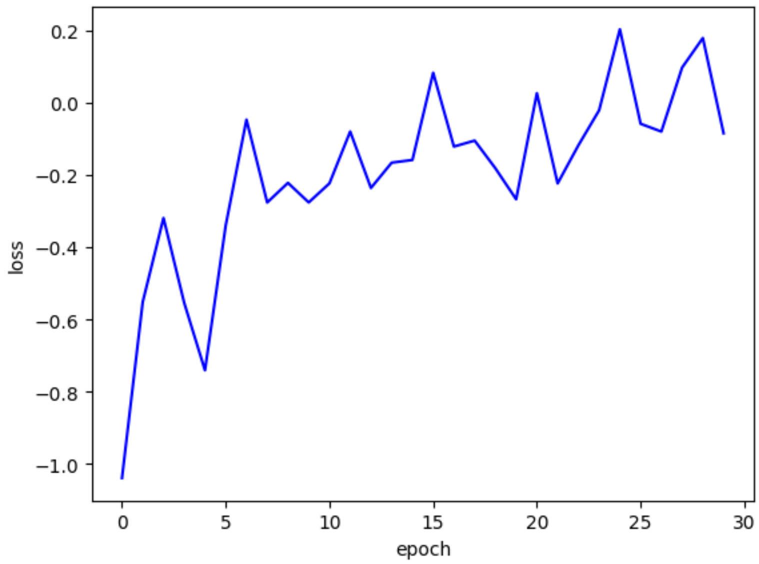


Table 1: Results Table

	Best hyper-parameters	avg Cross-validation P/R/F1	Test P/R/F1
SVM	C=10 $\gamma = .1$	F1=0.31 P=0.65 R=0.24	F1=0.44 P=0.39 R=0.53
Logistic Regression	C=1000 $\gamma = .01$	F1=0.4 P=0.69 R=0.29	F1=0.39 P=0.73 R=0.27