Weekly Report 1-Logistic Regression

July 12, 2022

Ganji Varshitha
AI20BTECH11009

Introduction

Logistic regression is a misnomer since it is a classification algorithm. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

Algorithm

We try to fit the data into a curved 'S' like graph unlike the straight line graph in linear regression.
The name of the algorithm comes from the function which is used to transform its output to return a probability value which can then be mapped to two or more discrete classes.

We assume functional form of $P(Y|X)$ and estimate parameters of $P(Y|X)$ directly from training data.

Let $p_y(x;w)$ be our estimate of $P(Y|X)$, where $\mathbf{w}$ is a vector of adjustable parameters.

$$P(Y=1|X,\mathbf{w}) = \frac{1}{1+\exp\left(-\mathbf{w}^\top X\right)} \tag{1}$$

In case of binary class problem, log odds of y or logit transformation of y is a linear function of x.
Inverse of logit is the Logistic function which is sigmoid function for binary class problem.
We estimate $\mathbf{w}$ using maximum likelihood estimation.

$$\mathbf{w} = \arg\max_{\mathbf{w}} \sum_{l} \ln P(y^l|\mathbf{x}^l,\mathbf{w}) \tag{2}$$

This involves minimising negative likelihood which is convex.
We use gradient descent for the minimisation problem by updating the weights in the direction of gradient.

Key points

- It does not assume on $P(X|Y)$ and learns the parameters that maximises the conditional probability $P(Y|X)$.
- It is a discriminative model which draws boundaries in the data space and focusses on predicting the labels.
- After predicting the probabilities of the output belong to a class, we need to map the probabilities to classes by using the decision threshold.
- It is a linear classifier as the decision rule is hyperplane.
- Softmax activation is used for multi class logistic regression.

Figure 1: Algorithm for binary class gradient descent(Logistic Regression)

$$
\begin{array}{l}
\text{For } j = 0, \ldots, d \\
\quad w_j \leftarrow \text{rand}(-0.01, 0.01) \\
\text{Repeat} \\
\quad \text{For } j = 0, \ldots, d \\
\quad\quad \Delta w_j \leftarrow 0 \\
\quad \text{For } t = 1, \ldots, N \\
\quad\quad o \leftarrow 0 \\
\quad\quad \text{For } j = 0, \ldots, d \\
\quad\quad\quad o \leftarrow o + w_j x_j^t \\
\quad\quad y \leftarrow \text{sigmoid}(o) \\
\quad\quad \text{For } j = 0, \ldots, d \\
\quad\quad\quad \Delta w_j \leftarrow \Delta w_j + (r^t - y) x_j^t \\
\quad \text{For } j = 0, \ldots, d \\
\quad\quad w_j \leftarrow w_j + \eta \Delta w_j \\
\text{Until convergence}
\end{array}
$$

Figure 2: Algorithm for multi class gradient descent(Logistic Regression)

```
For i = 1,...,K
    For j = 0,...,d
        w_ij ← rand(−0.01,0.01)
Repeat
    For i = 1,...,K
        For j = 0,...,d
            Δw_ij ← 0
    For t = 1,...,N
        For i = 1,...,K
            o_i ← 0
            For j = 0,...,d
                o_i ← o_i + w_ij x_j^t
        For i = 1,...,K
            y_i ← exp(o_i) / Σ_k exp(o_k)
        For i = 1,...,K
            For j = 0,...,d
                Δw_ij ← Δw_ij + (r_i^t − y_i) x_j^t
    For i = 1,...,K
        For j = 0,...,d
            w_ij ← w_ij + ηΔw_ij
Until convergence
```

```python

class LogisticRegression():
  def __init__(self,w, learning_rate,num_epochs):
    self.w = w
    self.learning_rate = learning_rate
    self.num_epochs = num_epochs

  def sigmoid(self,x):
    return 1/(1 + np.exp(-x))

  def train(self,X, y):
    n, m = X.shape


    for epoch in range(self.num_epochs):
      y_hat = [self.sigmoid(w@X[i]) for i in range(n)]
      error = -(np.sum([y[i]*np.log(y_hat[i]) + (1-y[i])*np.log(1-y_hat[i]) for i in range(n)]))
      dw = X.T@(y-y_hat)
      self.w = self.w - self.learning_rate*dw
    return y_hat, error, self.w

  def predict(self,X, w):
    y_pred = self.sigmoid(X@w)
    y_class = [1 if i > 0.5 else 0 for i in y_pred]
    return y_class

  def accuracy(self,y_true, y_pred):
```

```
    return np.mean([1 if (y_true[i]==y_pred[i]) else 0 for i in range(
  len(y_pred))])
```

Listing 1: Logistic Regression Code

Questions

1. Parameters of Logistic regression model are estimated using maximum likelihood estimation(MLE). State true or false.
2. Logit function gives __________of output variable.
3. Does the probability sum equal to 1 for sigmoid function in Logistic regression?
   - A. Yes, always.
   - B. Not always.
4. What can be done to avoid over-fitting in the model?
5. Which metric should be used to compare the logistic regression model output to the target variable?
   A. AUC-ROC curve B. Logloss C. MSE