Logistic Regression Tutorial

Contents

1 Introduction **1**

2 Parameter Learning in Logistic Regression **2**
2.1 Algorithm - SGD 3

3 Logistic Regression with Regularizer **3**
3.1 L1 Regularizer 3
3.2 L2 Regularizer 3

4 Bayesian interpretation of Regularizer **4**
4.1 Gaussian Prior 4

5 Multinomial Logistic Regression **4**

1 Introduction

Logistic regression is a statistical model that uses logistic function to model a binary classifier [1]. It outputs score from 0 to 1. If we have input data $x \in \mathcal{R}^n$ and output class $y \in \{0, 1\}$ then probability of each class is defined as below

$$\begin{aligned} P(y=1) &:= \sigma(-(w^T x + b)) \\ &= log\left(\frac{1}{1+e^{-(w^T x+b)}}\right) \\ P(y=0) &= 1 - P(y=1) \\ &= log\left(\frac{1}{1+e^{w^T x+b}}\right) \end{aligned}$$

where $w \in \mathcal{R}^n$ is parameter which is learnt from training data. During prediction :

$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

[1] https://en.wikipedia.org/wiki/Logistic_regression

2 Parameter Learning in Logistic Regression

Let $p(x_i) = P(Y = 1|X = x_i)$, and we have m training data points, then (conditional) likelihood function is

$$\begin{aligned}\mathcal{L}(w) &= \prod_{i=1}^{m} P(Y = y_i|X = x_i) \\ &= \prod_{i=1}^{m} p(x_i)^{y_i} * (1 - p(x_i))^{1-y_i}\end{aligned}$$

Taking negative log likelihood

$$l(w) = -\sum_{i=1}^{m} y_i log(p(x_i)) + (1 - y_i) log(1 - p(x_i)) \tag{1}$$

$$= -\sum_{i=1}^{m} y_i log(\sigma(w.x_i)) + (1 - y_i) log(1 - \sigma(w.x_i)) \tag{2}$$

We need to minimize the negative log likelihood function which is error function. This is unconstrained convex optimization problem and we can use gradient descent to learn the parameters which minimize the error function. Lets compute the gradient of sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\frac{d}{dx}\sigma(x) = \sigma(x) * (1 - \sigma(x))$$

Suppose $h = \sigma(w.x)$ then the loss function with respect to particular training example

$$f(w) = y * log(h) * (1 - y) * log(1 - h)$$

$$\frac{d}{dx}f(x) = (y - h)x$$

Then gradient of cost function mentioned in eq-2 is

$$\frac{d}{dw}l(w) = -\sum_{i=1}^{m} \frac{d}{dw}(y_i log(\sigma(w.x_i)) + (1 - y_i) log(1 - \sigma(w.x_i))) \tag{3}$$

$$= \sum_{i=1}^{m} (\sigma(w.x_i) - y_i)x_i \tag{4}$$

2.1 Algorithm - SGD

Data: Input data $\{x_i, y_i\}$ $i \in \{1 \dots m\}$ and $x_i \in R^n$, $y_i \in \{0, 1\}$
Result: $w \in R^n$
$w = 0$;
while *error is greater than* ϵ **do**
  for (x_i, y_i) *in training data* **do**
    1. Compute $grad = (\sigma(w.x_i) - y_i)x_i$;
    2. Update $w = w - \eta * grad$
  end
end

3 Logistic Regression with Regularizer

While training model using data, important features will be assigned high weight and there may be a chance that model can learn noise from data using these high weights. This is called overfitting. To avoid overfitting, we add regularizer term in objective function. Regularizer penalize large weight or basically control free parameters.

$$l(w) = -\sum_{i=1}^{n} y_i log(p(x_i)) + (1 - y_i) log(1 - p(x_i)) + \alpha \mathcal{R}(w)$$

After adding regularizer in loss function, our update step will be the following

$$w = w - \eta * \frac{d}{dx} l(w) - \eta * \alpha * \frac{d}{dx} \mathcal{R}(w)$$

$-\eta * \alpha * \frac{d}{dx} \mathcal{R}(w)$ is called weight decay. Weight decay is an example of regularization method.

3.1 L1 Regularizer

$$\mathcal{R}(w) = ||w||_1 = \sum_{i=1}^{n} |w_i|$$

L1 is difficult of optimize because L1 is not differentiable function. It provided sparse solution.

3.2 L2 Regularizer

$$\mathcal{R}(w) = ||w||_2 = \sum_{i=1}^{n} w_i^2$$

L2 is easier to optimize because of differentiability of function.

4 Bayesian interpretation of Regularizer

Both L1 and L2 regularization have Bayesian interpretations as constraints on the prior of how weights should look. L1 regularization can be viewed as a Laplace prior on the weights. L2 regularization corresponds to assuming that weights are distributed according to a gaussian distribution with mean 0.

4.1 Gaussian Prior

In a gaussian prior, the further away a value is from the mean, the lower its probability. By using a gaussian prior on the weights, we are saying that weights prefer to have the value 0. Prior $p(w_i) \in \mathcal{N}(0, \sigma^2_{w_i})$. Using Bayes formula $\hat{\theta} = argmin_\theta p(x|\theta) * p(\theta)$, new cost function

$$\mathcal{L}(w) = \prod_{i=1}^{n} p(x_i)^{y_i} * (1 - p(x_i))^{1-y_i}$$

$$p(w) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_{w_i}}} exp\left(-\frac{w_i^2}{2\sigma^2_{w_i}}\right)$$

Taking negative log likelihood

$$\begin{aligned} l(w) &= \mathcal{L}(w) * p(w) \\ l(w) &= -\sum_{i=1}^{n} y_i log(p(x_i)) + (1 - y_i) log(1 - p(x_i)) \\ &\quad - \sum_{i=1}^{n} log\left(\frac{1}{\sqrt{2\pi\sigma_{w_i}}} exp\left(-\frac{w_i^2}{2\sigma^2_{w_i}}\right)\right) \\ &= -\sum_{i=1}^{n} y_i log(\sigma(w.x_i)) + (1 - y_i) log(1 - \sigma(w.x_i)) + \alpha \sum_{i=1}^{n} w_i^2 \end{aligned}$$

As we can see that objective function achieved from gaussian prior is same as L2 regularizer.

5 Multinomial Logistic Regression

Multinomial Logistic Regression is also called softmax regression. The prediction for multinomial logistic regression is given as

$$p(y = c|x) = \frac{e^{w_c.x+b_c}}{\sum_{j=1}^{C} e^{w_j.x+b_j}}$$

Loss Function

$$
\begin{aligned}
L(y, \hat{y}) &= -\sum_{k=1}^{K} 1\{y = k\} log(p(y = k|x)) \\
&= -\sum_{k=1}^{K} 1\{y = k\} log \frac{e^{w_k.x+b_k}}{\sum_{j=1}^{C} e^{w_j.x+b_j}}
\end{aligned}
$$

Gradient

$$
\frac{d}{dw_k} L(y, \hat{y}) = -\left(1\{y = k\} - \frac{e^{w_k.x+b_k}}{\sum_{j=1}^{C} e^{w_j.x+b_j}}\right) x
$$