

1 Classification

1.1 Introduction

Classification is a supervised machine learning approach to categorizing data into **distinct number of classes** where we can assign label to each class. Given a set of data $\{x^{(i)}, y^{(i)}\}$ where x is the feature space in $m \times (n + 1)$ dimension, y is the classification output such $y \in \{0, 1\}$ for binary output or $\{1, 2, \dots, n\}$ for multiclass output. Classification algorithms are most used for Spam detection, Voice and image recognition, sentiment analysis, fraud detection and many more.

Machine learning classification algorithms come in different types and they include:

- Logistic regression
- Naive Bayes
- K-Nearest Neighbor
- Support vector Classifier
- Decision trees
- Boosted trees (Adaboost)
- Random Forest
- Perceptron

1.2 Logistic Regression

Logistic regression is a discriminative model since it focuses only on the posterior probability of each class $Pr(Y|x; \beta)$. It is also a generalized linear model, mapping output of linear multiple regression to posterior probability of each class $Pr(Y|x; \beta) \in \{0, 1\}$. probability a data-sample belongs to class 1 is given by

$$Pr(Y = 1|X = x; \beta) = \sigma(z), \text{ where } z = \beta^T x \quad (1)$$

$$P(Y = 1|X = x; \beta) = \sigma(\beta^T x) \quad (2)$$

where

$$Pr(Y = 1|X = x; \beta) + Pr(Y = 0|X = x; \beta) = 1 \quad (3)$$

$$Pr(Y = 0|X = x; \beta) = 1 - Pr(Y = 1|X = x; \beta) \quad (4)$$

Hence, probability that a data-sample belongs to class 0 is give by:

$$Pr(Y = 0|X = x; \beta) = 1 - \sigma(z) \quad (5)$$

$\sigma(z)$ is called the **logistic sigmoid function** and is give by

$$\sigma(z) = \frac{1}{1 + \exp^{-z}} \quad (6)$$

The uniqueness of this function is that it maps all real numbers \mathbb{R} to range $\{0, 1\}$.

Again, we know

$$\log(odds(Pr(Y = 1|X = x; \beta))) = \frac{Pr(Y = 1|X = x; \beta)}{Pr(Y = 0|X = x; \beta)} = \frac{Pr(Y = 1|X = x; \beta)}{1 - Pr(Y = 1|X = x; \beta)} \quad (7)$$

Assuming $P(Y = 1|X = x; \beta) = p(x)$, the next most obvious idea is to let $\log p(x)$ be a linear function of x , so that changing an input variable multiplies the probability by a fixed amount. This is done by taking a \log transformation of $p(x)$.

Formally, $\text{logit}(p(x)) = \beta_0 + \beta^T x$ making

$$\text{logit}(p(x)) = \log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta^T x \quad (8)$$

Simplifying for $p(x)$ and $1 - p(x)$ we have

$$\frac{p(x)}{1 - p(x)} = \exp(\beta_0 + \beta^T x) \quad (9)$$

$$p(x) = (1 - p(x)) \exp(\beta_0 + \beta^T x) \quad (10)$$

$$p(x) = \exp(\beta_0 + \beta^T x) - p(x) \cdot \exp(\beta_0 + \beta^T x) \quad (11)$$

$$p(x) + p(x) \cdot \exp(\beta_0 + \beta^T x) = \exp(\beta_0 + \beta^T x) \quad (12)$$

$$p(x)(1 + \exp(\beta_0 + \beta^T x)) = \exp(\beta_0 + \beta^T x) \quad (13)$$

$$p(x) = \frac{\exp(\beta_0 + \beta^T x)}{1 + \exp(\beta_0 + \beta^T x)} = \frac{\frac{1}{\exp(\beta_0 + \beta^T x)} \cdot \exp(\beta_0 + \beta^T x)}{\frac{1}{\exp(\beta_0 + \beta^T x)} + \frac{\exp(\beta_0 + \beta^T x)}{\exp(\beta_0 + \beta^T x)}} = \frac{1}{1 + \exp(-(\beta_0 + \beta^T x))} \quad (14)$$

$$1 - p(x) = \frac{1}{1 + \exp(\beta_0 + \beta^T x)} \quad (15)$$

1.2.1 Learning Logistic regression by maximum likelihood

Assume that $P(Y = 1|X = x; \beta) = P(x; \beta)$, for some function p parameterized by β . Further assume that observations are independent of each other. The conditional likelihood function is given by Bernoulli sequence:

$$\prod_{i=1}^n \Pr(Y = y_i | X = x_i; \beta) = \prod_{i=1}^n p(x_i; \beta)^{y_i} (1 - p(x_i; \beta))^{(1-y_i)} \quad (16)$$

The probability of a class is p , if $y_i = 1$, or $1 - p$, if $y_i = 0$. The likelihood is then

$$L(\beta_0, \beta) = \prod_{i=1}^n p(x_i)^{y_i} (1 - p(x_i))^{(1-y_i)} \quad (17)$$

taking the log of this likelihood we have

$$l(\beta_0, \beta) = \sum_{i=1}^n y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)) \quad (18)$$

$$= \sum_{i=1}^n y_i \log p(x_i) - y_i \log(1 - p(x_i)) + \log(1 - p(x_i)) \quad (19)$$

$$= \sum_{i=1}^n \log(1 - p(x_i)) + y_i \log \left(\frac{p(x_i)}{1 - p(x_i)} \right) \quad (20)$$

replace $\log \left(\frac{p(x)}{1 - p(x)} \right)$ with $\beta_0 + x \cdot \beta$ as seen in equation (8) and $(1 - p(x))$ with $\frac{1}{1 + \exp(\beta_0 + x \cdot \beta)}$. Hence,

$$l(\beta_0, \beta) = \sum_{i=1}^n \log \left(\frac{1}{\exp(\beta_0 + x \cdot \beta)} \right) + y(\beta_0 + x \cdot \beta) \quad (21)$$

$$= \sum_{i=1}^n -\log(1 + \exp(\beta_0 + x \cdot \beta)) + y(\beta_0 + x \cdot \beta) \quad (22)$$

$$\nabla l(\beta_0, \beta) = - \sum_{i=1}^n \frac{1}{1 + \exp(\beta_0 + x \cdot \beta)} x_{ij} \exp(\beta_0 + x \cdot \beta) + \sum_{i=1}^n y_i x_i \quad (23)$$

$$= \sum_{i=1}^n (y_i - p(x_i; \beta_0, \beta)) x_{ij} \quad (24)$$

Since this is a transcendental equation and there is no closed-form solution, we would apply the gradient ascent optimization algorithm.

1.2.2 Gradient ascent

Algorithm:

Using the cost function equal to the negative average of log-likelihood. eq(18)

$$\max_{\beta} J(\beta) = -\frac{1}{m} \left[\sum_{i=1}^m y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)) \right] \quad (25)$$

Repeat {

$$\beta_{j+1} = \beta_j + \alpha \cdot \nabla J(\beta) \quad (26)$$

}simultaneously updating β_j

Where $\nabla J(\theta) = \frac{1}{m} \sum_{i=1}^m (y_i - p(x_i; \beta_0, \beta)) x_{ij}$ and $p(x)$ remains logistic function given by $\frac{1}{1 + \exp(-(x \cdot \beta))}$. Newtons numerical optimization method is another approach to solving this problem- a case where the second order differential is required.

1.2.3 Learning Logistic regression by least-squares

We define a least-square optimization problem

$$\min_{\beta} J(\beta) = -\frac{1}{2} \sum_{i=1}^N (y_i - p(x_i; \beta_0, \beta))^2 \quad (27)$$

- $y_n \in [0, 1]$ is the desired posterior probability for the points x_i .
- $p(x_i; \beta_0, \beta)$ is the parametric function to be learned and it defined by equation (14).
- learning the classifier by regression equally means estimating the desired posterior probabilities with the function $p(x_i; \beta_0, \beta)$.

We can rewrite the objective function as

$$\min_{\beta} J(\beta) = \frac{1}{2} \|(y_n - p(x_n; \beta_0, \beta))\|_2^2 \quad (28)$$

$$= \frac{1}{2} (y_n - p(x_n; \beta_0, \beta))^T (y_n - p(x_n; \beta_0, \beta)) \quad (29)$$

$$= \frac{1}{2} (y^T y - y^T p(x_n; \beta_0, \beta) - y_n p(x_n; \beta_0, \beta)^T - p(x_n; \beta_0, \beta)^2) \quad (30)$$

Using the trace property from linear algebra $tr(A^T B) = tr(AB^T)$, we can deduce that $y_n p(x_n; \beta_0, \beta)^T = y_n^T p(x_n; \beta_0, \beta)$

$$= \frac{1}{2} (y^T y - 2y^T p(x_n; \beta_0, \beta) + p(x_n; \beta_0, \beta)^2) \quad (31)$$

$$\nabla J(\beta) = \frac{1}{2} \left[\frac{\partial y^T y}{\partial \beta} - 2y^T \frac{\partial p(x_n; \beta_0, \beta)}{\partial \beta} + \frac{\partial p(x_n; \beta_0, \beta)^2}{\partial \beta} \right] = 0 \quad (32)$$

Where $\frac{\partial p(x_n; \beta_0, \beta)}{\partial \beta} = \frac{-x \exp(-x\beta)}{(1 + \exp(-x\beta))^2}$ and $\frac{\partial p(x_n; \beta_0, \beta)^2}{\partial \beta} = \frac{-2x \exp(-x\beta)}{(1 + \exp(-x\beta))^3}$

$$\nabla J(\beta) = \frac{1}{2} \left[\frac{2y x \exp(-x\beta)}{(1 + \exp(-x\beta))^2} - \frac{2x \exp(-x\beta)}{(1 + \exp(-x\beta))^3} \right] \quad (33)$$

$$= \left[\frac{y x \exp(-x\beta)}{(1 + \exp(-x\beta))^2} - \frac{x \exp(-x\beta)}{(1 + \exp(-x\beta))^3} \right] \quad (34)$$

$$= \frac{x \exp(-x\beta)}{1 + \exp(-x\beta)} \left[y - \frac{1}{1 + \exp(-x\beta)} \right] \quad (35)$$

$$= [y - p(x_n; \beta_0, \beta)] (1 - p(x_n; \beta_0, \beta)) x \quad (36)$$

The β parameters can be evaluated using the **Gradient ascent algorithm** from section **1.2.2**

1.2.4 Regularized Logistic regression

Regularized logistic regression is much like regularized linear regression (Ridge regression) except with the addition of a division term $2m$. The cost function for logistic regression can be altered with the addition of a regularization term $\frac{\lambda}{2m}$. The cost function for the optimization problem is described using the equation below.

$$\max_{\beta} J(\beta) = -\frac{1}{m} \left[\sum_{i=1}^m y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)) + \frac{\lambda}{2} \sum_{i=1}^n \beta_j^2 \right] \quad (37)$$

which can be solved using the gradient ascent algorithm above with an addition of $(\lambda/m)\beta$

Repeat {

$$\beta_{j+1} = \beta_j + \alpha \cdot \nabla J(\beta) \quad (38)$$

}simultaneously updating β_j

$$\text{where } \nabla J(\beta) = -\frac{1}{m} \left[\sum_{i=1}^n (y_i - p(x_i; \beta_0, \beta)) x_{ij} + \lambda \beta_j \right] \quad (39)$$