Classification Task

Carlos Valle

Departamento de Informática
Universidad Técnica Federico Santa María

cvalle@inf.utfsm.cl

October 30, 2015

Overview

1. Classification task

2. Logistic Regression

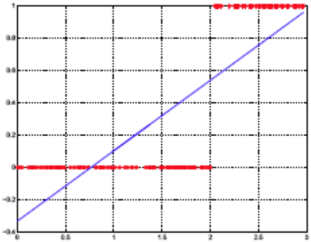
Motivation

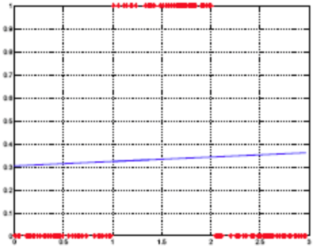
- As we stated before, in a classification problem the targets y can take K possible values, $k = 1, \ldots, K$
- For now, we will focus on binary classification where either $y \in \{0, 1\}$ or $y \in \{-1, 1\}$.
- Note that if $y \in \{0, 1\}$ can transformed to $y' \in \{-1, 1\}$ by doing $y' = y * 2 - 1$
- For example, a credit card company wants to classify its credit applications as "good credit" or "bad credit" given the annual salary, age, amount of previous debts:
-

Annual Salary (M$)	Age	previous debts (M$)	Credit
26	34	0	Good
28	28	203	Bad
6	55	7	Bad
32	42	10	Good
⋮	⋮	⋮	⋮

Why don't use a linear regression algorithm?

- Linear regression models might mask some classes.

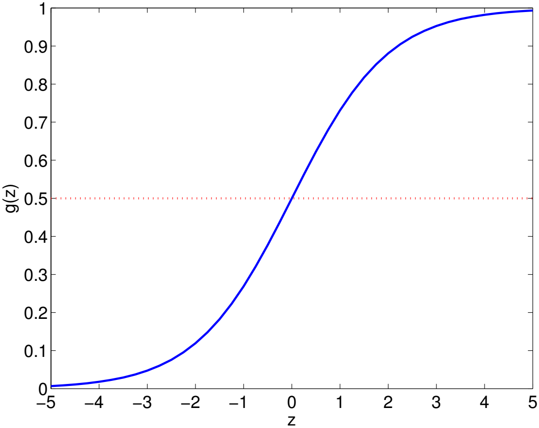




Logistic regression

- It measures the relationship between the class and the input vector by estimating probabilities using a logistic function:
- $$f_\beta(\mathbf{x}) = g(\beta^T \mathbf{x}) = \frac{1}{1 + e^{-\beta^T \mathbf{x}}},$$
- where
$$g(z) = \frac{1}{1 + e^{-z}} = \frac{e^z}{1 + e^z}$$
is the logistic or sigmoid function
- As before, $\beta^T \mathbf{x} = \beta_0 + \sum_{i=1}^{I} \beta_i x(i)$
- However, it is worth mentioning that $f_\beta(\mathbf{x})$ is not linear.

Logistic function



- When $z = 0 = \beta^T x$, $g(z)$ is on the decision threshold 0.5.
- When $z \to \infty$, $g(z) \to 1$.
- When $z \to -\infty$, $g(z) \to 0$.

Logistic function (2)

- The derivative of $g(z)$ is
- $$\begin{aligned} g'(z) &= \frac{d}{dz}\frac{1}{1+e^{-z}} \\ &= \frac{1}{(1+e^{-z})^2}\left(e^{-z}\right) \\ &= \frac{1}{(1+e^{-z})}\cdot\left(1-\frac{1}{(1+e^{-z})}\right) \\ &= g(z)(1-g(z)) \end{aligned}$$

How to obtain β?

- Let us assume that

$$\begin{aligned} P(y=1|\mathbf{x};\beta) &= f_\beta(\mathbf{x}) \\ P(y=0|\mathbf{x};\beta) &= 1-f_\beta(\mathbf{x}) \end{aligned}$$

- Thus $p(y|\mathbf{x};\beta) = (f_\beta(\mathbf{x}))^y(1-f_\beta(\mathbf{x}))^{1-y}$

How to obtain β? (2)

- Assuming that the M training example are independent, we can express the likelihood function as:

$$\begin{aligned} L(\beta) &= \prod_{m=1}^{M} p(y_m|\mathbf{x}_m;\beta) \\ &= \prod_{m=1}^{M} (f_\beta(\mathbf{x}_m))^{y_m}(1-f_\beta(\mathbf{x}_m))^{1-y_m} \end{aligned}$$

- As we did before, we maximize the log likelihood function:

$$\begin{aligned} \ell(\beta) &= \sum_{m=1}^{M} y_m \log f_\beta(\mathbf{x}_m) + (1-y_m)\log(1-f_\beta(\mathbf{x}_m)) \\ &= \sum_{m=1}^{M} y_m \beta^T \mathbf{x}_m - \log(1+e^{\beta^T \mathbf{x}_m}) \end{aligned} \tag{1}$$

Stochastic gradient ascent

- Now we can use gradient descent algorithm:

$$\beta^{p+1} = \beta^p + \alpha \nabla_\beta \ell(\beta)$$

- Let's obtain the derivative:

$$\begin{aligned}
\frac{\partial}{\partial \beta_i}\ell(\beta) &= \left(y\frac{1}{f_\beta(\mathbf{x})} - (1-y)\frac{1}{1-f_\beta(\mathbf{x})}\right)\frac{\partial}{\partial \beta_i} f_\beta(\mathbf{x}) \\
&= \left(y\frac{1}{f_\beta(\mathbf{x})} - (1-y)\frac{1}{1-f_\beta(\mathbf{x})}\right) f_\beta(\mathbf{x})(1-f_\beta(\mathbf{x}))\frac{\partial}{\partial \beta_i}\beta^T\mathbf{x} \\
&= \left(y(1-f_\beta(\mathbf{x})) - (1-y)f_\beta(\mathbf{x})\right)x^{(i)} \\
&= \left(y - f_\beta(\mathbf{x})\right)x^{(i)}
\end{aligned}$$

- Thus, the stochastic gradient ascent rule is

$$\beta_i^{p+1} = \beta_i^p + \alpha\left(y - f_\beta(\mathbf{x})\right)x^{(i)}$$

Newton-Raphson Method

- Another way for maximizing $\ell(\beta)$ is the Newton-Raphson method.
- Let $f : \mathbb{R} \to \mathbb{R}$ be a real valued function. From f_T the second order Taylor expansion we have

$$f_T(\beta) = f_T(\beta^p + \Delta\beta) \approx f(\beta^p) + f'(\beta^p)\Delta\beta + \frac{1}{2}f''(\beta^p)\Delta\beta^2.$$

- Setting to zero the derivative with respect to $\Delta\beta$

$$\frac{d}{d\Delta\beta}\left(f(\beta^p) + f'(\beta^p)\Delta\beta + \frac{1}{2}f''(\beta^p)\Delta\beta^2\right) = f'(\beta^p) + f''(\beta^p)\Delta\beta = 0.$$

Newton-Raphson Method (2)

- This implies that

$$\begin{aligned}
\Delta\beta &= -\frac{f'(\beta^p)}{f''(\beta^p)} \\
\beta^{p+1} - \beta^p &= -\frac{f'(\beta^p)}{f''(\beta^p)} \\
\beta^{p+1} &= \beta^p - \frac{f'(\beta^p)}{f''(\beta^p)}
\end{aligned}$$

Multidimensional Newton-Raphson Method

- Generalizing for a vector β
- $$\beta^{p+1} = \beta^{p} - H^{-1}\nabla_{\beta}(\ell(\beta)),$$
- where the Hessian H is computed by

$$H_{ij} = \frac{\partial^2 \ell(\beta)}{\partial \beta_i \partial \beta_j} = \frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T}$$

- Where, W is a diagonal matrix with ith element $W_{ii} = f(\mathbf{x}_i)(1 - f(\mathbf{x}_i))$

Multidimensional Newton-Raphson Method (2)

- $\nabla_\beta \ell(\beta) = X^T(Y - f_\beta(X))$
- $H = -X^T W X$
- $H^{-1}\nabla_\beta(\ell(\beta)) = -(X^T W X)^{-1} X^T (Y - f_\beta(X))$
- Thus, the update rule is:

$$\begin{aligned}
\beta^{p+1} &= \beta^p + (X^T W X)^{-1} X^T (Y - f_\beta(X)) \\
&= (X^T W X)^{-1} X^T W X \beta^p + (X^T W X)^{-1} X^T W W^{-1} (Y - f_\beta(X)) \\
&= (X^T W X)^{-1} X^T W (X\beta^p + W^{-1}(Y - f_\beta(X)) \\
&= (X^T W X)^{-1} X^T W z
\end{aligned}$$

- where $z = X\beta^p + W^{-1}(Y - f_\beta(X))$
- Sometimes z is so-called the adjusted response.
- Note that we need recompute $f_\beta(X)$.

What is logistic regression doing?

- z can be viewed as a target vector. Where X is the input matrix and β^{p+1} is the solution of the least square problem:
- $$\beta^{p+1} = \text{argmin}_{\beta}(z - XB)^T W(z - XB)$$

Algorithm 1 Iterative Reweighted Least Squares

1: Initialize β

2: **repeat**

3: Compute $f_\beta(\mathbf{x})$

4: Update $\beta \leftarrow \beta + (X^T W X)^{-1} X^T (Y - f_\beta(X))$

5: Compute objective function (Validation Set)

$$\text{OF} \leftarrow \sum_{m'=1}^{M'} p(y_{m'}|\mathbf{x}_{m'};\beta) = \sum_{m'=1}^{M'} (f_\beta(\mathbf{x}_{m'}))^{y_{m'}} (1 - f_\beta(\mathbf{x}_{m'}))^{1-y_{m'}}$$

6: **until** OF converges

Any questions?