Logistic regression (classification)

Example :

Suppose you're an actuary and want to predict that a given patient may have some major health issue in the next 5 years.

i.e. $p(\underbrace{\text{major health issue}} \mid x)$, $x = (x_1, x_2, x_3)$

$x_1 = \text{age}$, $x_2 = M/F$, $x_3 = $ cholesterol level,

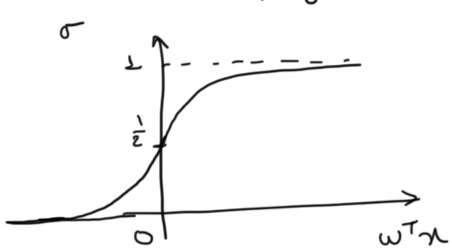
The simplest model would be to consider a linear combination of the input variables :

$$\left\{ w_0 + w_1 x_1 + w_2 x_2 + \dots + w_d x_d = w^T x \right. , \quad x = (1, x_1, x_2, \dots$$

→ this will not yield a probability

We can fix this by introducing a simple warping transformati

$$p(y|x) = \underbrace{\sigma}_{\text{sigmoid function}}(w^T x) \quad , \quad \sigma(x) = \frac{1}{1+e^{-x}} \ : \ \text{logistic sigmoid function}$$



Workflow :
$\left\{ \begin{array}{l} 1. \text{ Prior / model specification} \longrightarrow \text{Likelihood} \\ 2. \text{ Training} \\ 3. \text{ Prediction} \end{array} \right.$

Setup : Given $\mathcal{D} := \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, $x_i \in \mathbb{R}^d$, $y_i \in \{$
$i = 1, \dots, n$

Model : $y_i \sim Ber(\sigma(w^T x))$ y_i are i.i.d.

Pros :

- interpetable : the model parameters w have a meaning.
  e.g. $\underline{\underline{w_1}} \gg 0$ then probability of disease increases with age
- it reveals which input features are most influention.
- small number of parameters $(d+1)$
- computationally efficient ways to estimate w.
- easy extension to multi-class classification

Cons :

Being a simple model, its performance is often inferior to more sophisticated models.

Maximum Likehood Estimation :

$$w_{MLE} = \arg\max_{w} p(\mathcal{D}|w) \quad , \quad p(\mathcal{D}|w) =$$

$$= p(y_1, y_2, \dots, y_n | x_1, x_2, \dots, x_n, w) =$$

$$= \prod_{i=1}^{n} p(y_i | x_i, w)$$

predicted class probability

Let $a_i = \sigma(w^T x_i)$

then, $$p(\mathcal{D}|w) = \prod_{i=1}^{n} \underbrace{a_i^{y_i} (1-a_i)^{1-y_i}}_{\text{Bernoulli pmf}}$$

$$w_{MLE} = \arg\min_{w} - \log p(\mathcal{D}|w) := \mathcal{L}(w)$$

$$\mathcal{L}(w) = -\log p(\mathcal{D}|w) = -\sum_{i=1}^{n} y_i \log a_i + (1-y_i)\log(1-a_i)$$

Binary cross-entropy loss

Before we compute $\nabla_w \mathcal{L}(w)$, let's derive the following:

- $$\log a = \log \sigma(w^T x) = \log \frac{1}{1+e^{-w^T x}} = -\log(1+e^{-w^T x})$$

- $\log(1-a) = \log(1-\sigma(w^T x)) = \dots = -w^T x - \log(1+e^{-w^T x}$

- $\frac{\partial}{\partial w} \log a = - \frac{-x e^{-w^T x}}{1+e^{-w^T x}} = x(1-a)$

- $\frac{\partial}{\partial w} \log(1-a) = -x + x(1-a) = -ax$

Therefore,

$$\frac{\partial}{\partial w_j} L(w) = - \sum_{i=1}^{n} y_i x_{ij}(1-a_i) - (1-y_i) x_{ij} a_i$$

$$= \dots = \sum_{i=1}^{n} (a_i - y_i) x_{ij}$$

Notice that : $\underset{(d+1)\times 1}{\nabla_w L(w)} = \underset{(d+1)\times n}{X^T} \underset{n\times 1}{(a-y)}$, $\underset{n\times(d+1)}{X} = \begin{array}{c} w_0 \\ \begin{bmatrix} 1 & x_{11} & \dots & x_{1d} \\ 1 & \vdots & & \\ \vdots & \vdots & & \\ 1 & x_{n1} & \dots & x_{nd} \end{bmatrix} \end{array}$

$\underset{n\times 1}{a} = \begin{bmatrix} \sigma(w^T x_1) \\ \vdots \\ \sigma(w^T x_n) \end{bmatrix}$, $\underset{n\times 1}{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$

⊛ We cannot solve for w_{MLE} analytically since w shows up in a non-linear fashion in $\nabla_w L(w) = 0$.

<u>Hession</u> : $\nabla_w^2 L(w)$

$$\left\{ \frac{\partial^2}{\partial w_j \partial w_k} L(w) = \sum_{i=1}^{n} x_{ij} \left(\frac{\partial}{\partial w_k} a_i \right) = \overbrace{\sum_{i=1}^{n} x_{ij} x_{ik} a_i (1-a_i)}^{\text{quadratic form}} \right.$$

$$= z_j^T A z_k \quad , \quad z_j := (x_{1j}, \dots, x_{nj})^T \rightarrow \text{j-th of Col design ma}$$

where, $\underset{n\times n}{A} := \begin{bmatrix} a_1(1-a_1) & & 0 \\ & \ddots & \\ 0 & & a_n(1-a_n) \end{bmatrix}$

- $\nabla^2 L(w) = X^T A X$ one can show that this is

$$\Longrightarrow \underset{(d+1)\times(d+1)}{\nabla_w L(w)} = \underset{(d+1)\times n}{X}\ \underset{n\times n}{A}\ \underset{n\times(d+1)}{X} \longrightarrow$$

a positive-semidefinite matri

$\Longrightarrow L(w)$ is convex in w.

Iterative re-weighted least squares :

Recall Newton : $w_{t+1} = w_t - H_t^{-1} g_t$, $\boxed{\begin{aligned} H_t &= X^T A_t X \\ g_t &= X^T(a_t - y)\end{aligned}}$

$$\Longrightarrow w_{t+1} = w_t - (X^T A_t X)^{-1} X^T (a_t - y)$$

we re-write this as :

$$w_{t+1} = (X^T A_t X)^{-1} X^T A_t \overbrace{\left[X w_t - A_t^{-1}(a-y)\right]}^{V_t}$$

$$= \underbrace{(X^T A_t X)}_{\text{Hessian}}{}^{-1} X^T A_t V_t \longrightarrow$$

is the solution to a weighte least squares problems

Recall the MLE solution in linear regression:

$$\rightarrow w_{MLE} = (X^T X)^{-1} X^T y = (X^T A X)^{-1} X^T A y$$

where A is the identity matrix.

Multi-class logistic regression :

Model :

$$p(y = c \mid x, W) = \frac{\exp(W_c^T x)}{\sum_{c'=1}^{C} \exp(W_{c'}^T x)}$$

soft-max function, a generalization of t logistic sigmoic the multi-class setting

where w_c is the c-th column of $\underset{(d+1)\times C}{W}$

an y is a one-hot encoding matrix :

$$\underset{n\times C}{y} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

i.e. $y_{ic} = \mathbb{1}_{\{y_i = c\}}$

Likelihood :

$$p(\mathcal{D}|w) = \prod_{i=1}^{n} \prod_{c=1}^{C} p(y_i = c | x_i, W)$$

$$\Rightarrow \quad -\log p(\mathcal{D}|w) = -\sum_{i=1}^{n} \left[\left(\sum_{c=1}^{C} y_{ic} W_c^T x_i \right) - \log \left(\sum_{c'=1}^{C} \exp(W_{c'}^T \right.\right.$$

↳ multi-class cross-entropy loss