

# Machine learning

## Logistic Regression

Hamid Beigy

Sharif University of Technology

April 8, 2023





1. Introduction
2. Logistic regression
3. MLE formulation of Logistic regression
4. MAP formulation of Logistic regression
5. Connection with Bayes
6. Multiclass classification
7. Reading

## Introduction

---

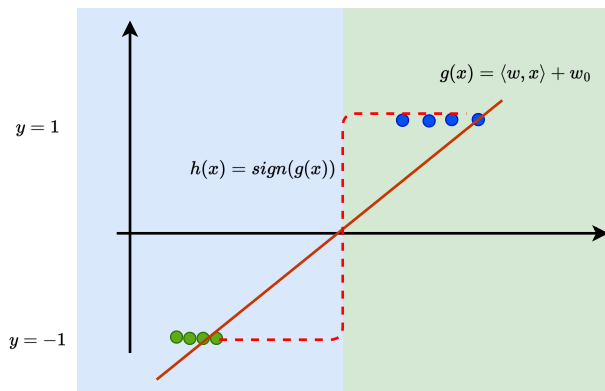


1. In the case of

- Gaussian class conditional densities
- same covariance matrix
- equal prior

the separating hyperplane is linear.

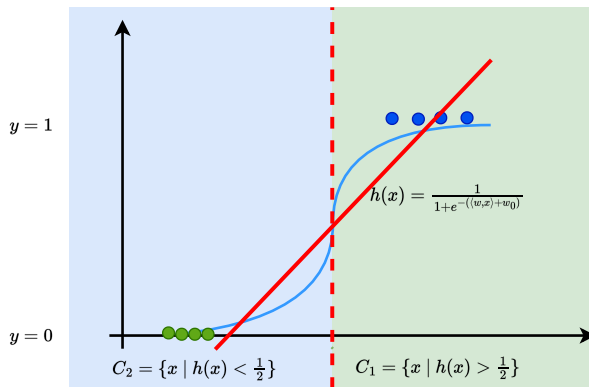
2. In this case, the separating hyperplane returned by **linear regression** coincides with the separating hyperplane returned by **Bayes classifier**.



## Logistic regression

---

1. If we replace *sign* with its *soft-version*, we obtain *logistic regression*.

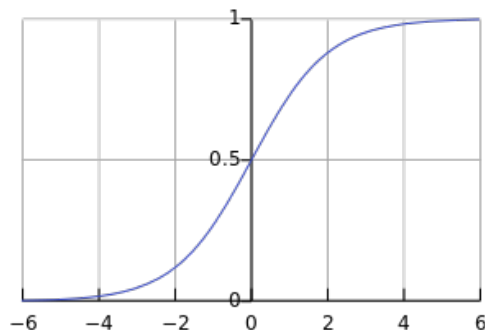




## 1. Function

$$h(x) = \frac{1}{1 + e^{-\langle w, x \rangle + w_0}}$$

is called a **Sigmoid function**.



## 2. Properties of Sigmoid function

- Limits of Sigmoid function

$$h(z) \rightarrow 1 \quad \text{as} \quad z \rightarrow \infty$$

$$h(z) \rightarrow 0 \quad \text{as} \quad z \rightarrow -\infty$$

Hence,  $h(z)$  can be regarded as a **probability**.

- Derivative of  $h(z)$  equals to

$$\frac{\partial h(z)}{\partial z} = h(z) [1 - h(z)]$$

The derivative is **always positive**, and  $h$  is always an **increasing function**. Hence,  $h$  can be considered as a **CDF**.



1. One loss function may be

$$\ell(t_n, h(\mathbf{x}_n)) = (t_n - h(\mathbf{x}_n))^2$$

This loss function is not a **convex** function and is not easy to optimize.

2. The likelihood function can be written

$$p(t|\mathbf{w}) = \begin{cases} y_n & t_n = 1 \\ (1 - y_n) & t_n = 0 \end{cases}$$

3. If  $t_n = 1$  but  $y_n$  is close to 0 then loss will be high.
4. If  $t_n = 0$  but  $y_n$  is close to 1 then loss will be high.
5. The likelihood function can also be written

$$p(t|\mathbf{w}) = y_n^{t_n} (1 - y_n)^{(1-t_n)}$$

6. We can define a loss function by taking the negative logarithm of the likelihood.

$$\mathcal{L}(\mathbf{w}) = -\ln \prod_{n=1}^N \ell(t_n, h(\mathbf{x}_n)) = -\sum_{n=1}^N [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)]$$

7. This loss function is called the **cross-entropy loss** and is **convex**.





1. Let  $t_n \in \{-1, +1\}$ . Another way to write the log-likelihood of data is.

$$\begin{aligned} p(+1|\mathbf{x}) &= \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x})} \\ p(-1|\mathbf{x}) &= \frac{1}{1 + \exp(+\mathbf{w}^\top \mathbf{x})} \end{aligned}$$

2. By combining the above equations and computing negative log-likelihood of data, we obtain

$$\begin{aligned} \mathcal{L}(\mathbf{w}) &= -\sum_{n=1}^N \ln \frac{1}{1 + \exp(-t_n \mathbf{w}^\top \mathbf{x}_n)} \\ &= \sum_{n=1}^N \ln [1 + \exp(-t_n \mathbf{w}^\top \mathbf{x}_n)] \end{aligned}$$

3. Unlike linear regression, we can no longer write down the minimum of negative log-likelihood in the **closed form**. Instead, we need to use an optimization algorithm for computing it.



1. Computing the gradients of  $\mathcal{L}(\mathbf{w})$  with respect to  $\mathbf{w}$ , we obtain

$$\nabla \mathcal{L}(\mathbf{w}) = \sum_{n=1}^N t_n \mathbf{x}_n (y_n - t_n)$$

2. Updating the weight vector using the gradient descent rule will result in

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta \sum_{n=1}^N t_n \mathbf{x}_n (y_n - t_n)$$

$\eta$  is the learning rate.

3. In order to have a good trade-off between the training error and the generalization error, we can add the regularization term.

$$\mathcal{L}(\mathbf{w}) = \sum_{n=1}^N \log [1 + \exp(-t_n \mathbf{w}^\top \mathbf{x}_n)] + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

4. Using the gradient descent rule, will result in the following updating rule.

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta \sum_{n=1}^N t_n \mathbf{x}_n (y_n - t_n) - \lambda \mathbf{w}^{(k)}$$

## MLE formulation of Logistic regression

---



1. In linear regression, we often assume that the noise has a Gaussian distribution.

$$p(t|\mathbf{x}, \mathbf{w}) = \mathcal{N}(t|\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$$

2. We can generalize the linear regression to binary classification by making two changes:

- First, replacing the Gaussian distribution for  $t$  with Bernoulli distribution, which is more appropriate for classification.

$$p(t_n|\mathbf{x}_n, \mathbf{w}) = \text{Ber}(t_n|y_n) = \begin{cases} y_n & \text{if } t_n = 1 \\ 1 - y_n & \text{if } t_n = 0 \end{cases}$$

where  $\mu(\mathbf{x}_n) = \mathbb{E}[t_n|\mathbf{x}_n] = p(t_n = 1|\mathbf{x}_n)$ .

- This is equivalent to

$$p(t_n|\mathbf{x}_n, \mathbf{w}) = \text{Ber}(t_n|\mu(\mathbf{x}_n)) = \mu(\mathbf{x}_n)^{t_n} (1 - \mu(\mathbf{x}_n))^{(1-t_n)}$$

- Second, compute a linear combination of the inputs and then we pass this through a function that ensures  $0 \leq \mu(\mathbf{x}) \leq 1$  by defining

$$\mu(\mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x})$$



1. Putting these two steps together and dropping index  $n$ , we obtain

$$p(t|\mathbf{x}, \mathbf{w}) = \text{Ber}(t|\sigma(\mathbf{w}^\top \mathbf{x})).$$

2. This is called **logistic regression** due to its similarity to linear regression.
3. If we threshold the output probability at  $\frac{1}{2}$ , we can introduce a decision rule of the form

$$\text{if } p(t = 1|\mathbf{x}) > 0.5 \iff h(\mathbf{x}) = 1.$$

4. Logistic regression learns weights so as to maximize the (log-)likelihood of the data.
5. Let  $S = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_N, t_N)\}$  be the training set. The negative log-likelihood of data equals

$$\begin{aligned} \mathcal{L}(\mathbf{w}) &= -\ln \prod_{n=1}^N y_n^{t_n} (1 - y_n)^{(1-t_n)} \\ &= -\sum_{n=1}^N t_n \ln y_n + (1 - t_n) \ln(1 - y_n) \end{aligned}$$

This is called the **cross-entropy** error function.

## MAP formulation of Logistic regression

---



1. Maximum likelihood estimate of  $\mathbf{w}$  can lead to **overfitting** when data set is linearly separable. A solution is to use a prior on  $\mathbf{w}$ .
2. This can be avoided by inclusion of a prior and finding a MAP solution or equivalently by adding a regularization term to the error function.
3. Same as linear regression, we consider a Gaussian prior on  $w$

$$p(\mathbf{w}) = \mathcal{N}(0, \sigma_0^2 I_D).$$

4.  $I_D$  denotes the  $D \times D$  identity matrix. This is equivalent to assume that the prior selects each component of  $W$  independently from a  $\mathcal{N}(0, \sigma_0^2)$ . This prior can be written as

$$p(\mathbf{w}) = \frac{1}{(2\pi)^{D/2} \sigma_0^D} \exp \left\{ -\frac{1}{2\sigma_0^2} \|\mathbf{w}\|_2^2 \right\}.$$



1. Assume that noise precision is known, The posterior density of  $\mathbf{w}$  given set  $S$  and solving the equation gives the form

$$\mathcal{L}(\mathbf{w}) = \sum_{n=1}^N \log \left[ 1 + \exp(-t_n \mathbf{w}^\top \mathbf{x}_n) \right] + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

2. Thus MAP estimation is equivalent to regularized logistic regression.
3. Using the gradient descent rule, will result in the following updating rule.

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta \sum_{n=1}^N t_n \mathbf{x}_n (y_n - t_n) - \lambda \mathbf{w}^{(k)}$$



## Connection with Bayes

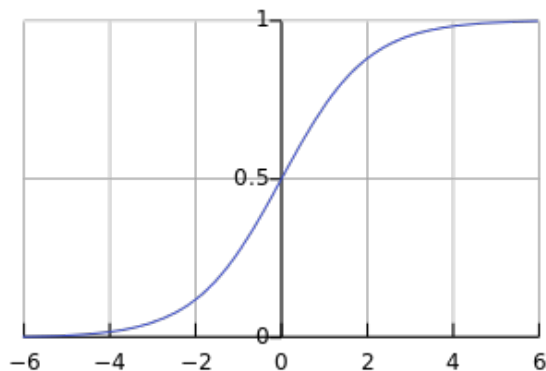
---



## 1. Bayes classifier for two classes +1 and -1

$$\begin{aligned} p(+1|x) &= \frac{P(x|+1)P(+1)}{P(x)} = \frac{P(x|+1)P(+1)}{p(x|+1)p(+1) + p(x|-1)p(-1)} \\ &= \frac{1}{1 + \frac{p(x|-1)p(-1)}{P(x|+1)P(+1)}} = \frac{1}{1 + \exp(-a)} = \sigma(a) \\ a &= \ln \frac{P(x|+1)P(+1)}{P(x|-1)P(-1)} \end{aligned}$$

where  $\sigma(z)$  refers to Sigmoid function.





1. Let the class conditional densities be  $D$ -dimensional Gaussian (for  $k \in \{-1, +1\}$ )

$$p(\mathbf{x}|k) = \mathcal{N}(\mu, \Sigma) = \frac{1}{|\Sigma|^{D/2}(2\pi)^{D/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_k)^\top \Sigma^{-1}(\mathbf{x} - \mu_k)\right)$$

2. Hence  $a$  equals to

$$\begin{aligned} a &= \ln \frac{P(\mathbf{x}|+1)P(+1)}{P(\mathbf{x}|-1)P(-1)} \\ &= \ln \frac{\exp\left(-\frac{1}{2}(\mathbf{x} - \mu_1)^\top \Sigma^{-1}(\mathbf{x} - \mu_1)\right) P(+1)}{\exp\left(-\frac{1}{2}(\mathbf{x} - \mu_2)^\top \Sigma^{-1}(\mathbf{x} - \mu_2)\right) P(-1)}. \end{aligned}$$

3. Hence, we have

$$P(+1|\mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + w_0)$$

where

$$\begin{aligned} \mathbf{w} &= \Sigma^{-1}(\mu_1 - \mu_2) \\ w_0 &= -\frac{1}{2}\mu_1^\top \Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^\top \Sigma^{-1}\mu_2 + \ln \frac{P(+1)}{P(-1)} \end{aligned}$$

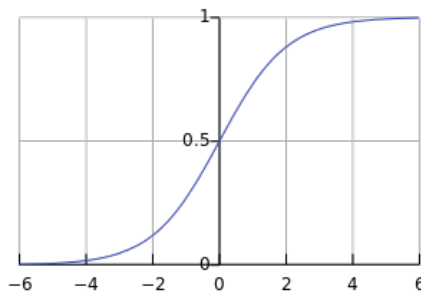
or simply

$$P(+1|\mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x})$$



1. We compute a linear combination of the inputs but then we pass through a function that ensures  $0 \leq y_n \leq 1$  by defining.

$$y_n = \sigma(\mathbf{w}^\top \mathbf{x}) \triangleq \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x})}.$$



2. Finding  $\mathbf{w}$  directly, we need to find  $D$  parameters.
3. Finding  $P(k|\mathbf{x})$  via probabilistic modeling of data using Gaussian distribution and MLE, we need
  - $2D$  parameters for mean
  - $\frac{D(D+1)}{2}$  parameters for shared covariance matrix
  - One parameter for  $P(+1)$resulting  $\frac{D(D+5)}{2} + 1$  parameters.



1. Logistic regression is a model for **probabilistic classification**.
2. It predicts **label probabilities** rather than a **hard value of the label**.
3. Let

$$y_n = P(+1|\mathbf{x}_n)$$

$$1 - y_n = P(-1|\mathbf{x}_n)$$

4. The output of Logistic regression is a probability defined using the Sigmoid function

$$\begin{aligned} P(+1|\mathbf{x}_n) = y_n &= \sigma(\mathbf{w}^\top \mathbf{x}_n) \\ &= \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x}_n)} \end{aligned}$$

5. The log of the ratio of probabilities  $\ln \frac{P(+1|\mathbf{x}_n)}{P(-1|\mathbf{x}_n)}$  for the two classes, also known as the **log odds** equals to

$$\begin{aligned} \ln \frac{P(+1|\mathbf{x}_n)}{P(-1|\mathbf{x}_n)} &= \ln \exp(\mathbf{w}^\top \mathbf{x}_n) \\ &= \mathbf{w}^\top \mathbf{x}_n \end{aligned}$$

6. Thus if  $\mathbf{w}^\top \mathbf{x}_n > 0$ , the probable class is **+1**.

## Multiclass classification

---



1. Targets form a discrete set  $\{1, \dots, K\}$ .
2. We represent them as **one-hot vectors**  $\mathbf{t} = \{0, \dots, 0, 1, 0, \dots, 0\}$ .  
entry  $k$  is 1
3. There are  $D$  input dimensions and  $K$  output dimensions.
4. We need  $K \times D$  weights, arranged as a weight matrix  $\mathbf{W}$  and a  $K$ -dimensional vector  $\mathbf{w}_0$ .
5. Linear predictions

$$z_k = \sum_j w_{kj} x_j + w_{0k}$$

and vectorized as

$$\mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{w}_0$$

6. The **probability that the sample belong to class  $k$**  equals to

$$p(k|\mathbf{x}) = \frac{e^{z_k}}{\sum_j e^{z_j}}$$

7. If one of the  $z_k$ 's is much larger than the others, then **softmax( $\mathbf{z}$ )** is approximately the **argmax**. So really it's more like **soft-argmax**.

## Reading




---





1. Sections 4.3.2 of [Pattern Recognition and Machine Learning Book](#) (Bishop 2006).
2. Chapter 8 of [Machine Learning: A probabilistic perspective](#) (Murphy 2012).
3. Chapter 10 of [Probabilistic Machine Learning: An introduction](#) (Murphy 2022).



-  Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag.
-  Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.
-  — (2022). *Probabilistic Machine Learning: An introduction*. The MIT Press.

Questions?