Chapter 6

Logistic Regression

Goal: Binary classification with linear decision boundary.

6.1 Generalized Binary Linear Classification Models

Generalized linear model: Generalized linear model is the model that applies some (non-linear) function $g(\cdot) : \mathscr{R} \to \mathscr{R}$ on $\mathbf{w}^T\mathbf{x}_i$,

$$\hat{y}_i = g(\mathbf{w}^T\mathbf{x}_i) \tag{6.1}$$

For classification, $y \in \{0, 1\}$.

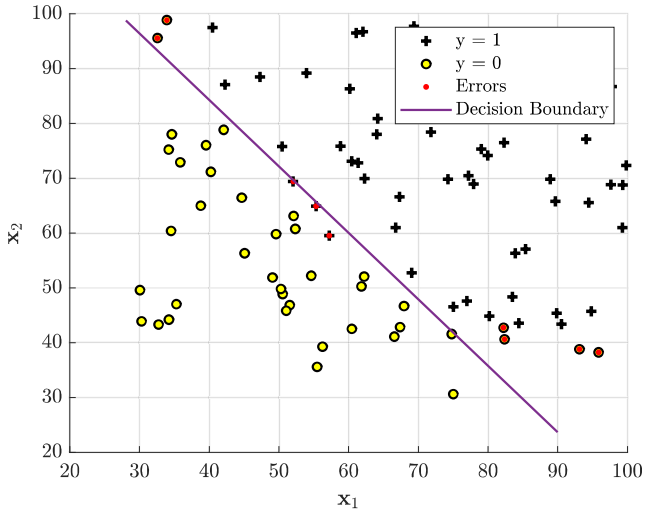

Figure 6.1: An example of linear classification boundary.

Example 6.1: Examples of a function $g(x)$ are,

$$g(x) = x \text{ basic linear model} \tag{6.2}$$

$$\sigma(x) = \frac{1}{1 - \exp(x)} \text{ sigmoid} \tag{6.3}$$

$$0 \leq \sigma(x) \leq 1$$

Goal: How to find weights $\mathbf{w}$?

Remainder for the line equation,

$$\mathbf{x}^T\mathbf{y} = \|\mathbf{x}\|\|\mathbf{y}\| \cos(\theta) \tag{6.4}$$

$\mathbf{x} \perp \mathbf{w} \Rightarrow \theta = 90° \Rightarrow \mathbf{x}^T\mathbf{w} = 0$

6.2 Basic Linear Model

$$\tilde{\mathbf{y}} = \mathbf{X}\mathbf{w} \tag{6.5}$$

$$\hat{y}_j = \begin{cases} 1 & \tilde{y}_j > \frac{1}{2} \\ 0 & \tilde{y}_j < \frac{1}{2} \end{cases} \tag{6.6}$$

Example

$$\mathbf{X}^T = \left[\begin{array}{cccc|ccc} 1 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ 10 & 11 & \cdots & 19 & 20 & \cdots & 29 \end{array}\right]$$

$$\mathbf{y}^T = \left[\begin{array}{cccc|ccc} 1 & 1 & \cdots & 1 & 0 & \cdots & 0 \end{array}\right]$$

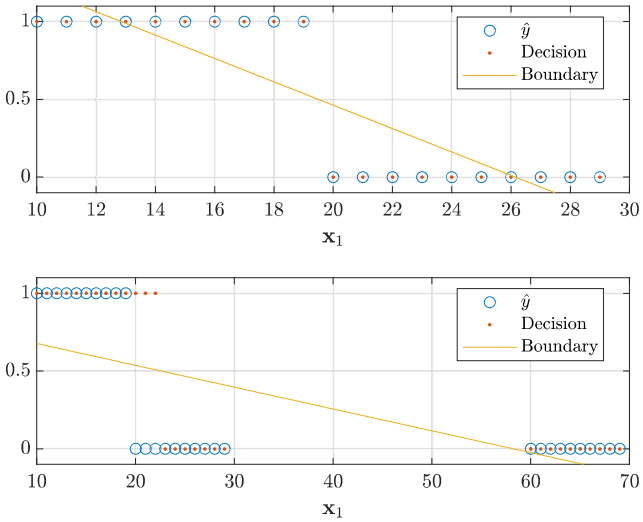

Figure 6.2: 1D synthetic example of classification by linear regression.

Summary

- $\tilde{y}$ may be higher than 1 and lower than 0.
- **Outlier** has a dramatic influence.

6.3 Logistic Model

Goal: Binary classification model with:

- Linear model
- Outliers handling
- Probabilistic interpretation

Logistic regression is one of the generalized linear classification models that is based on sigmoid function.
Sigmoid function:

$$\sigma(x) = \frac{\exp(x)}{1+\exp(x)} = \frac{1}{1+\exp(-x)} \tag{6.7}$$

The function is visualized in Fig. 6.3.

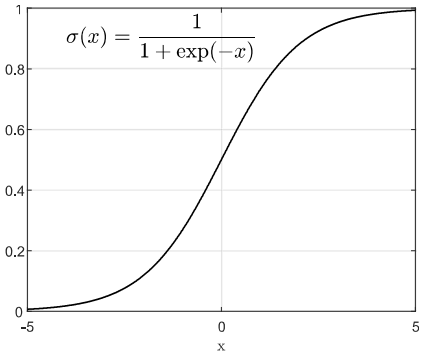

Figure 6.3: Plot of a sigmoid function.

Logistic regression:

$$\hat{y}_i = \sigma(\mathbf{w}^T\mathbf{x}_i) = \frac{1}{1+\exp\left(-\mathbf{w}^T\mathbf{x}_i\right)} \tag{6.8}$$

$$\hat{\mathbf{y}} = \sigma(\mathbf{X}\mathbf{w}) \tag{6.9}$$

Loss function: MSE loss has no closed-form solution and has local minimums⇒ not used.

$$\mathcal{L}(\cdot) = \frac{1}{2M}\|\hat{\mathbf{y}} - \mathbf{y}\|^2 \tag{6.10}$$

Loss function is not necessary metric.

6.4 Cross-entropy loss

Goal: Probabilistic loss.

The resulting value of $\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}(\mathbf{X})$ has **probabilistic** interpretation and $L(\hat{\mathbf{y}}, \mathbf{y})$ quantifies a distance between target and output distributions. Typically, the distance metrics between probability density functions (PDFs) are used.

6.4.1 Entropy

Entropy: For the discrete distribution $P = \{p_i = \Pr[X = x_i]\}$, the entropy is given by

$$H(P) = -\sum_i p_i \log(p_i) \tag{6.11}$$

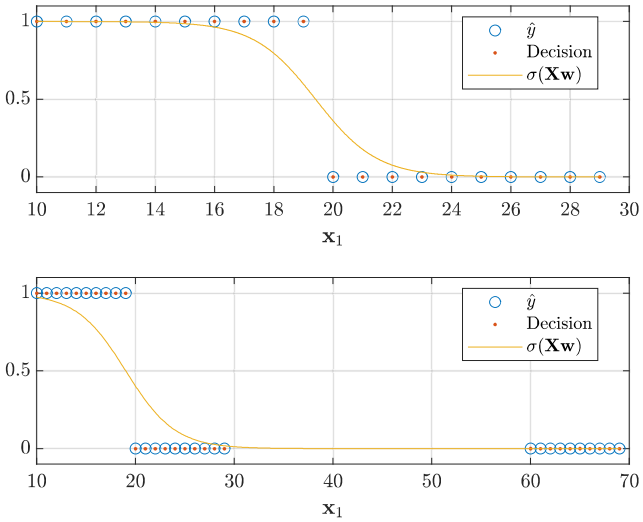

Figure 6.4: 1D synthetic example of classification by linear regression.

The sign depends on the context of the definition (sometimes + is used).
Entropy is a measure of the uncertainty associated with a given distribution, P. It has the maximum value for $p_i = p_j \; \forall i, j$ and drops down for any other combinations.
Coding interpretation: The entropy is an information theory measure. One of the interpretations of entropy (with base-2 logarithm and '-' sign) is the theoretical limit on the average number of bits needed to compress the outcomes of the distribution p. Numerical example:

$$p_1 = p_2 = \frac{1}{2} \Rightarrow H(p) = -2 \cdot \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$$

$$p_1 = \frac{1}{10}, p_2 = \frac{9}{10} \Rightarrow H(p) = -\frac{1}{10}\log_2\left(\frac{1}{10}\right) - \frac{9}{10}\log_2\left(\frac{9}{10}\right) \approx 0.4690$$

Numerical example:

- Equal probabilities:
  - Consider the transmission of $\{A, B, C, D\}$ sequences over a binary channel. If all 4 letters are equally likely (25%) probable, $p_i = 0.25$.
  - The possible code is $\{00, 01, 11, 10\}$. One can not do better than using two bits to encode each letter.
  - $H(P) = -4 * \frac{1}{4}\log_2\left(\frac{1}{4}\right) = 2$
  - The lowest possible coding rate is achieved.
- Unequal probabilities example in Table 6.1:
  - Average coding rate is

$$\sum_i \text{length}_i \cdot p_i = 1 * 0.7 + 2 * 0.26 + 3 * 0.02 + 3 * 0.02 = 1.34$$

  - The theoretical lowest coding rate.

$$H(P) = -0.7\log_2(0.7) - 0.26\log_2(0.26) - 0.02\log_2(0.02) - 0.02\log_2(0.02) \approx 1.0912$$

Table 6.1: Unequal probabilities example.

Word	Probability, p_i	Codeword C_i	Codeword length, length_i
A	0.7	0	1
B	0.26	10	2
C	0.02	110	3
D	0.02	111	3

$E[Y] = \sum_i y_i \Pr[Y = y_i]$

6.4.2 Cross-entropy

Cross-entropy: For two discrete distributions, p and q, the cross-entropy is given by

$$H(p, q) = \pm \sum_i p_i \log(q_i) \tag{6.12}$$

The minimum value of $P(p, q)$ is when $p = q$, and as a consequence $H(p, q) = H(p)$.
Coding interpretation: One of the interpretations of entropy (with base-2 logarithm and $'-'$ sign) is the theoretical limit on the average number of bits that are required to encode distribution p with the theoretically optimal code for q. Numerical example: Let's encode $\{A, B, C, D\}$ with probabilities $q_i = \left\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right\}$. In this case, one of the optimal coding options is to use two-bit binary code. Now, let's encode $p_i = \left\{\frac{1}{2}, \frac{1}{2}, 0, 0\right\}$ with this code. The resulting quantities are $H(p) = 1$ and $H(p, q) = 2$ which means that instead of optimal 1-bit code for p, the 2-bit code is required if the code optimal for q is used to encode p.
Notes:

- $\lim_{x \to 0} x \log(x) \to 0$
- For a loss function, typically e-base logarithm is used.
- Maximum likelihood estimation (MLE) has the same minimum for $\boldsymbol{\theta}$.

6.4.3 Binary Cross-Entropy (BCE)

The visualization of BCE of the form

$$H(p, q) = -p_0 \log(q_0) - p_1 \log(q_1) \tag{6.13}$$

is presented in Fig. 6.5 For example, when $p_0 = 0$ and $p_1 = 1 - p_0 = 1$, the expression reduces to $H(p, q) = \log(q_1), q_1 \in [0, 1]$.

6.4.4 Binary Cross-Entropy (BCE) Loss

Goal: Minimum cross entropy.

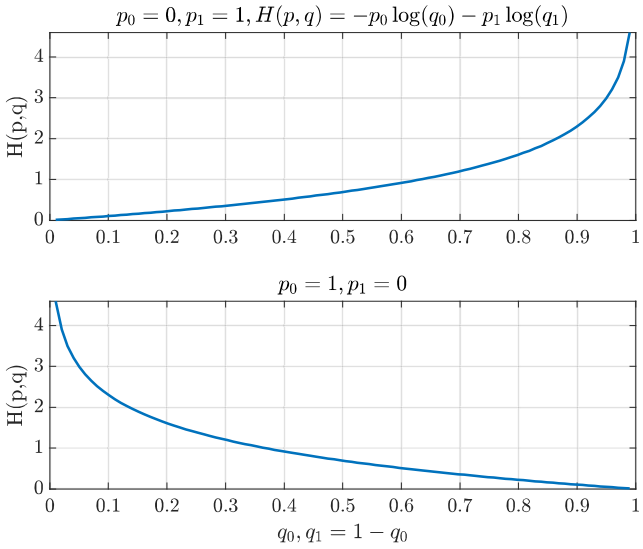

Figure 6.5: Illustration of BCE (Eq. (6.13).

For example, for a single decision for $y = 1$ in this example, we would like that $f_{\boldsymbol{\theta}}(\mathbf{x}) \to 1$,

$$\begin{aligned} p_0 &= \Pr(y = 0) = 1 - y \\ p_1 &= \Pr(y = 1) = y \\ q_0 &= \Pr(\hat{y} = 0) = 1 - f_{\boldsymbol{\theta}}(\mathbf{x}) \\ q_1 &= \Pr(\hat{y} = 1) = f_{\boldsymbol{\theta}}(\mathbf{x}) \\ H(p, q) &= -p_0 \log(q_0) - p_1 \log(q_1) \\ &= -(1 - y) \log\big(1 - f_{\boldsymbol{\theta}}(\mathbf{x})\big) - y \log\big(f_{\boldsymbol{\theta}}(\mathbf{x})\big) \end{aligned}$$

The discussion is symmetric for $y = 0, f_{\boldsymbol{\theta}}(\mathbf{x}) \to 0$.
BCE loss: Binary cross-entropy (BCE) loss function

$$\mathcal{L}(y, \hat{y}) = -(1 - y) \log(1 - \hat{y}) - y \log(\hat{y}) \tag{6.14}$$

For multi-valued vector $\mathbf{y}$ the loss is the average (or sum) over all y_i elements,

$$\begin{aligned} \mathcal{L} &= -\frac{1}{M} \sum_{j=1}^{M} (1 - y_i) \log(1 - \hat{y}_i) + y_i \log(\hat{y}_i) \\ &= -\frac{1}{M} \left[(1 - \mathbf{y}) \log(1 - \hat{\mathbf{y}}) + \mathbf{y} \log(\hat{\mathbf{y}})\right] \end{aligned} \tag{6.15}$$

Properties: The BCE loss is continuous, differentiable and convex.

6.5 BCE Loss for Logistic Regression

Probabilistic prediction:

$$\begin{aligned} p(y = 1 | \mathbf{x}, \mathbf{w}) &= \sigma(\tilde{\mathbf{x}}\mathbf{w}) \\ p(y = 0 | \mathbf{x}, \mathbf{w}) &= 1 - \sigma(\tilde{\mathbf{x}}\mathbf{w}) \end{aligned} \tag{6.16}$$

Classification decision: $\hat{y} \lessgtr \frac{1}{2}$

Another way:

$$\hat{\mathbf{y}} = \begin{cases} 1 & \mathbf{x}^T\mathbf{w} \geq 0 \\ 0 & \mathbf{x}^T\mathbf{w} < 0 \end{cases} \tag{6.17}$$

Vector notation

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}; \mathbf{w}) = \frac{1}{M}\Big[-\mathbf{y}^T \log\big(\sigma(\mathbf{X}\mathbf{w})\big) - (1-\mathbf{y})^T \log\big(1-\sigma(\mathbf{X}\mathbf{w}\big) \tag{6.18}$$

The first order gradient does not have a closed-form solution.

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}) = \frac{1}{M}\mathbf{X}^T\left(\sigma\left(\mathbf{X}\mathbf{w}\right) - \mathbf{y}\right) \tag{6.19}$$

However, it can easily found by GD minimization that involves only vector and matrix operations:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \alpha\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}) \tag{6.20}$$

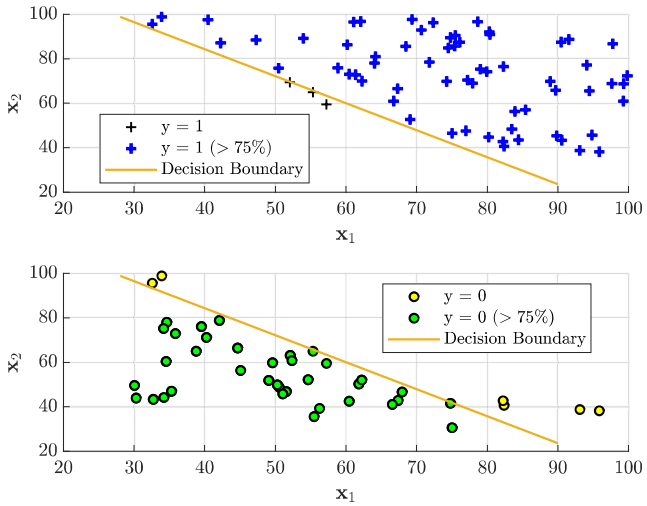

Figure 6.6: Example of $\sigma(\mathbf{X}\mathbf{w}) \geq 0.75$ and $\sigma(\mathbf{X}\mathbf{w}) \leq 0.25$.

Important properties of the logistic regression with BCE loss function:

- Global minimum.
- Continuous, differentiable and convex.
- Regularization can be applied,

$$\mathcal{L}_{new} = \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) + \frac{\lambda}{2M}\sum_{i=1}^{N} w_i^2 \tag{6.21}$$

- Mapping functions or kernels can be applied. For example, for polynomial mapping

$$\varphi(x_1, x_2) = \langle 1, x_1, x_1^2, x_2, x_2^2, x_1x_2, x_1^2x_2, x_1x_2^2, x_1^2x_2^2 \rangle$$

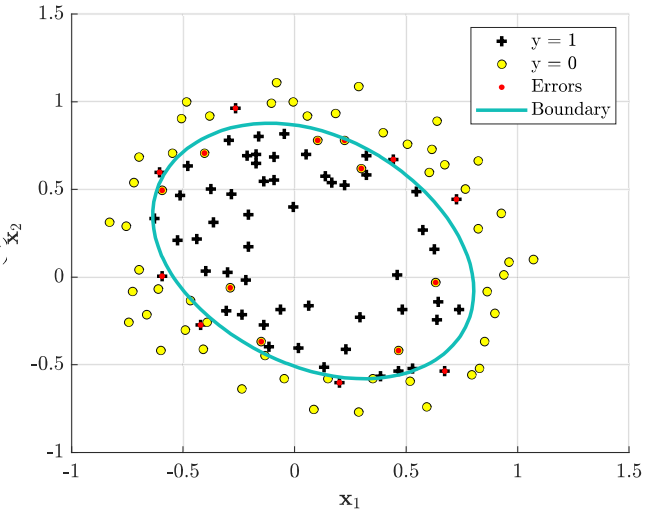

Figure 6.7: Example polynomial $\varphi(x_1, x_2)$ mapping.

6.6 k-NN

Uses k nearest neighbors for a decision.
Different distance metrics, some of them with additional hyper-parameter(s). For example:

- Euclidean distance metric,

$$\begin{aligned} d(\mathbf{a}, \mathbf{b}) &= \|\mathbf{a} - \mathbf{b}\| \\ &= \sqrt{(a_1-b_1)^2 + (a_2-b_2)^2 + \cdots + (a_M-b_M)^2} \end{aligned} \tag{6.22}$$

- City block (Manhattan) distance

$$\begin{aligned} d(\mathbf{a}, \mathbf{b}) &= \sum_{j=1}^{M} |a_j - b_j| \\ &= |a_1 - b_1| + \cdots + |a_M - b_M| \end{aligned} \tag{6.23}$$

- Minkowski distance with (hyper) parameter p,

$$d(\mathbf{a}, \mathbf{b}) = \sqrt[p]{\sum_{j=1}^{M} |a_j - b_j|^p} \tag{6.24}$$

  For the special case of $p = 1$ the Minkowski distance gives the city block distance. For $p = 2$, the Minkowski distance gives the Euclidean distance.

Tie-breaking (same number of neighbors) algorithm for $k > 1$ neighbors:

- Random selection.
- Use the class with the nearest neighbor.

Summary

- Fair baseline performance.
- High inference complexity. It requires M distance calculations for each new point (e.g., logistic regression uses $\mathbf{w}$ vector.)
- Can not handle outliers.
- 100% training performance.
- Normalization is required!
- Two hyper-parameters: k and distance metric.
- Can also be applied for regression.