Data may not be linearly separable. Main idea: model this probabilistically.

Model:

\[ p(y=1|x) = F(w^T x) \quad \text{where} \quad w = \text{a parameter vector} \]

Geometrically

\[ w^T x = 0 \quad p(y=1|x) = \frac{1}{2} \]

F should "squash" \( w^T x \) into a probability value in [0, 1]

For logistic regression,

\[ F(x) = \frac{1}{1 + e^{-2x}} \]

Main Idea:

1. Fit the model \( p(y=1|x) = \frac{1}{1 + e^{-w^T x}} \)
2. Find the vector \( w \) based on training data

How to find \( w \)?

Maximum likelihood principle.

Find \( w \) so that

\[ p_{\hat{w}}(y_1, \ldots, y_n | z_1, \ldots, z_n) \] is Maximum

training labels

training feature vectors
\[
\arg\max_{\omega} \ p_\omega(y_1, \ldots, y_n | x_1, \ldots, x_n) \\
= \arg\max_{\omega} \ \prod_{i=1}^{n} p_\omega(y_i = y_i | x_i) \\
= \arg\max_{\omega} \ \log \left( \prod_{i=1}^{n} p_\omega(y_i = y_i | x_i) \right) \\
= \arg\max_{\omega} \ \sum_{i=1}^{n} \log p(y_i = y_i | x_i) \\
= \arg\max_{\omega} \ \sum_{i, y_i = 1} \log \frac{1}{1 + e^{-\omega^T x_i}} + \sum_{i, y_i = -1} \log \frac{1}{1 + e^{\omega^T x_i}} \\
= \arg\max_{\omega} \ \sum_{i=1}^{n} - \log (1 + e^{-y_i \omega^T x_i}) \\
= \arg\min_{\omega} \ \sum_{i=1}^{n} \log (1 + e^{y_i \omega^T x_i})
\]

So finding the best \( \omega \) reduces to finding the \( \omega \) that minimizes the function:

\[
L(\omega) = \sum_{i=1}^{n} \log (1 + e^{-y_i \omega^T x_i})
\]

Called a **loss function**.

Generally used beyond logistic regression.
How to minimize the loss function?

Through **gradient descent**

From an arbitrary starting point, take small steps along \(-\nabla L(w)\) until we get close to the minimum.

At minimum, \(\nabla L(w) = 0\).

\[
L(w) = \sum_{i=1}^{n} \log (1 + e^{-y_i w^T x_i})
\]

\[
\nabla L(w) = \sum_{i=1}^{n} \frac{1}{1 + e^{y_i w^T x_i}} e^{-y_i w^T x_i} (-y_i x_i)
\]

\[
= - \sum_{i=1}^{n} \frac{y_i x_i}{1 + e^{y_i w^T x_i}}
\]

**Algorithm:**

1. Initialize \(w_0\)

2. For \(t=1,2, \ldots, T\)

\[
W_{t+1} = W_t + \eta_t \sum_{i=1}^{n} \frac{y_i x_i}{1 + e^{y_i w^T x_i}}
\]

Learning rate usually decreases with \(t\).

\(\eta_t\) measures how big the steps are.

Too big \(\rightarrow\) too much bouncing around

Too small \(\rightarrow\) get stuck, too little progress.
Gradient descent does not always converge to the optimal solution. It will not converge for I but will for II. The logistic regression loss function looks like II.

**Summary:** Important concepts.
- Logistic regression fits a model described by a vector $\mathbf{w}$ to data.
- The best $\mathbf{w}$ is one that minimizes a particular loss function.
- $\mathbf{w}$ is computed through gradient descent.