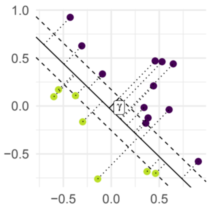
Introduction to Machine Learning

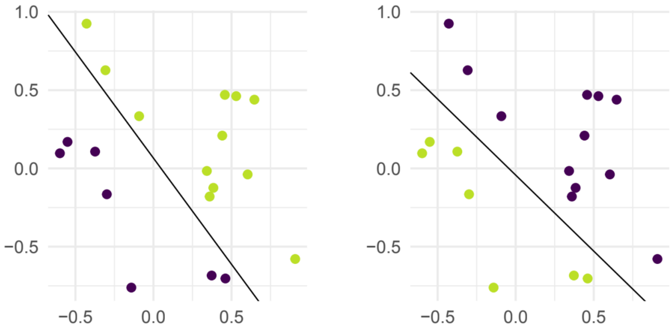
Linear Hard Margin SVM



Learning goals

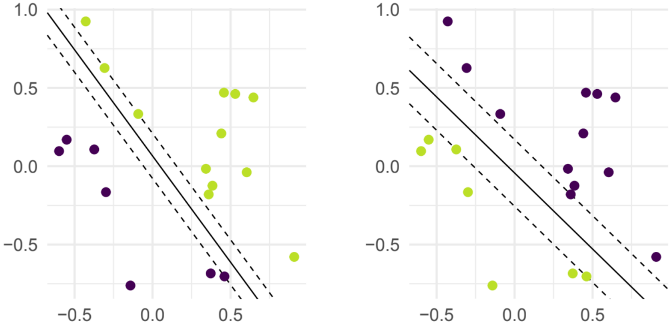
- Know that the hard-margin SVM maximizes the margin between data points and hyperplane
- Know that this is a quadratic program
- Know that support vectors are the data points closest to the separating hyperplane

LINEAR CLASSIFIERS



- We want study how to build a binary, linear classifier from solid geometrical principles.
- Which of these two classifiers is “better”?

LINEAR CLASSIFIERS



- We want study how to build a binary, linear classifier from solid geometrical principles.
- Which of these two classifiers is “better”?

$\rightarrow$ The decision boundary on the right has a larger **safety margin.**

SUPPORT VECTOR MACHINES: GEOMETRY

For labeled data $\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)}\right), \dots, \left(\mathbf{x}^{(n)}, y^{(n)}\right)\right)$, with $y^{(i)} \in \{-1, +1\}$:

- Assume linear separation by $f(\mathbf{x}) = \boldsymbol{\theta}^\top \mathbf{x} + \theta_0$, such that all $+$-observations are in the positive halfspace

$$\{\mathbf{x}^{(i)} \in \mathcal{X} : f(\mathbf{x}) > 0\}$$

  and all $-$-observations are in the negative halfspace

$$\{\mathbf{x} \in \mathcal{X} : f(\mathbf{x}) < 0\}.$$

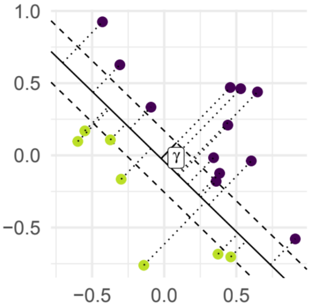
- For a linear separating hyperplane, we have

$$y^{(i)} \underbrace{\left(\boldsymbol{\theta}^\top \mathbf{x}^{(i)} + \theta_0\right)}_{=f\left(\mathbf{x}^{(i)}\right)} > 0 \quad \forall i \in \{1, 2, ..., n\}.$$

SUPPORT VECTOR MACHINES: GEOMETRY

- $$d\left(f, \mathbf{x}^{(i)}\right) = \frac{y^{(i)} f\left(\mathbf{x}^{(i)}\right)}{\|\boldsymbol{\theta}\|} = y^{(i)} \frac{\boldsymbol{\theta}^T \mathbf{x}^{(i)} + \theta_0}{\|\boldsymbol{\theta}\|}$$

  computes the (signed) distance to the separating hyperplane $f(\mathbf{x})$, positive for correct classifications, negative for incorrect.
- This expression becomes negative for misclassified points.

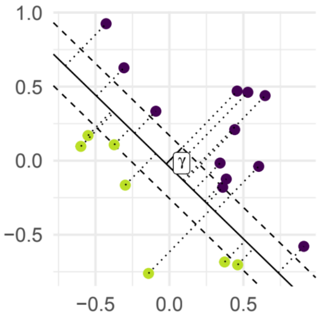


SUPPORT VECTOR MACHINES: GEOMETRY

- The distance of f to the whole dataset $\mathcal{D}$ is the smallest distance

$$\gamma = \min_i \left\{ d\left(f, \mathbf{x}^{(i)}\right) \right\}.$$

- This represents the “safety margin”, it is positive if f separates and we want to maximize it.



MAXIMUM MARGIN SEPARATION

We formulate the desired property of a large “safety margin” as an optimization problem:

$$\begin{aligned} \max_{\boldsymbol{\theta},\theta_0} \quad & \gamma \\ \text{s.t.} \quad & d\left(f, \mathbf{x}^{(i)}\right) \geq \gamma \quad \forall\, i \in \{1, \ldots, n\}. \end{aligned}$$

- The constraints mean: We require that any instance *i* should have a “safety” distance of at least γ from the decision boundary defined by $f = \boldsymbol{\theta}^T \mathbf{x} + \theta_0$.
- Our objective is to maximize the “safety” distance.

MAXIMUM MARGIN SEPARATION

We reformulate the problem:

$$
\begin{aligned}
\max_{\boldsymbol{\theta},\theta_0} \quad & \gamma \\
\text{s.t.} \quad & \frac{y^{(i)}\left(\left\langle\boldsymbol{\theta},\mathbf{x}^{(i)}\right\rangle+\theta_0\right)}{\|\boldsymbol{\theta}\|} \geq \gamma \quad \forall\, i \in \{1,\ldots,n\}.
\end{aligned}
$$

- The inequality is rearranged by multiplying both sides with $\|\boldsymbol{\theta}\|$:

$$
\begin{aligned}
\max_{\boldsymbol{\theta},\theta_0} \quad & \gamma \\
\text{s.t.} \quad & y^{(i)}\left(\left\langle\boldsymbol{\theta},\mathbf{x}^{(i)}\right\rangle+\theta_0\right) \geq \|\boldsymbol{\theta}\|\gamma \quad \forall\, i \in \{1,\ldots,n\}.
\end{aligned}
$$

MAXIMUM MARGIN SEPARATION

- Note that the same hyperplane does not have a unique representation:

$$\{\mathbf{x} \in \mathcal{X} \mid \boldsymbol{\theta}^\top \mathbf{x} = 0\} = \{\mathbf{x} \in \mathcal{X} \mid c \cdot \boldsymbol{\theta}^\top \mathbf{x} = 0\}$$

  for arbitrary $c \neq 0$.
- To ensure uniqueness of the solution, we make a reference choice – we only consider hyperplanes with $\|\boldsymbol{\theta}\| = 1/\gamma$:

$$\begin{aligned} \max_{\boldsymbol{\theta}, \theta_0} \quad & \gamma \\ \text{s.t.} \quad & y^{(i)} \left(\left\langle \boldsymbol{\theta}, \mathbf{x}^{(i)} \right\rangle + \theta_0 \right) \geq 1 \quad \forall\, i \in \{1, \dots, n\}. \end{aligned}$$

MAXIMUM MARGIN SEPARATION

- Substituting $\gamma = 1/\|\boldsymbol{\theta}\|$ in the objective yields:

$$\begin{aligned} \max_{\boldsymbol{\theta},\theta_0} \quad & \frac{1}{\|\boldsymbol{\theta}\|} \\ \text{s.t.} \quad & y^{(i)}\left(\left\langle \boldsymbol{\theta}, \mathbf{x}^{(i)} \right\rangle + \theta_0\right) \geq 1 \quad \forall\, i \in \{1, \ldots, n\}. \end{aligned}$$

- Maximizing $1/\|\boldsymbol{\theta}\|$ is the same as minimizing $\|\boldsymbol{\theta}\|$, which is the same as minimizing $\frac{1}{2}\|\boldsymbol{\theta}\|^2$:

$$\begin{aligned} \min_{\boldsymbol{\theta},\theta_0} \quad & \frac{1}{2}\|\boldsymbol{\theta}\|^2 \\ \text{s.t.} \quad & y^{(i)}\left(\left\langle \boldsymbol{\theta}, \mathbf{x}^{(i)} \right\rangle + \theta_0\right) \geq 1 \quad \forall\, i \in \{1, \ldots, n\}. \end{aligned}$$

QUADRATIC PROGRAM

We derived the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\theta},\theta_0} \quad & \frac{1}{2}\|\boldsymbol{\theta}\|^2 \\ \text{s.t.} \quad & y^{(i)}\left(\left\langle \boldsymbol{\theta}, \mathbf{x}^{(i)} \right\rangle + \theta_0\right) \geq 1 \quad \forall\, i \in \{1, \ldots, n\}. \end{aligned}$$

This turns out to be a **convex optimization problem** – particularly, a **quadratic program**: The objective function is quadratic, and the constraints are linear inequalities.

This is called the **primal** problem. We will later show that we can also derive a dual problem from it.

We will call this the **linear hard-margin SVM**.

SUPPORT VECTORS

- There exist instances $(\mathbf{x}^{(i)}, y^{(i)})$ with minimal margin $y^{(i)} f\left(\mathbf{x}^{(i)}\right) = 1$, fulfilling the inequality constraints with equality.
- They are called **support vectors (SVs)**. They are located exactly at a distance of $\gamma = 1/\|\boldsymbol{\theta}\|$ from the separating hyperplane.
- It is already geometrically obvious that the solution does not depend on the non-SVs! We could delete them from the data and would arrive at the same solution.