Support-Vector Machines

Given a training set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, $\mathbf{x}_i \in \mathbb{R}^p$ are data points belonging to two different classes or groups. The class membership is indicated by $y_i \in \{-1, 1\}$.

The key idea is to select a particular hyperplane that separates the points in the two classes and maximizes the *margin*, i.e., the distance between the hyperplane and the closest points of the training set from each class.

The basic concept is drafted in Fig. 6.1.

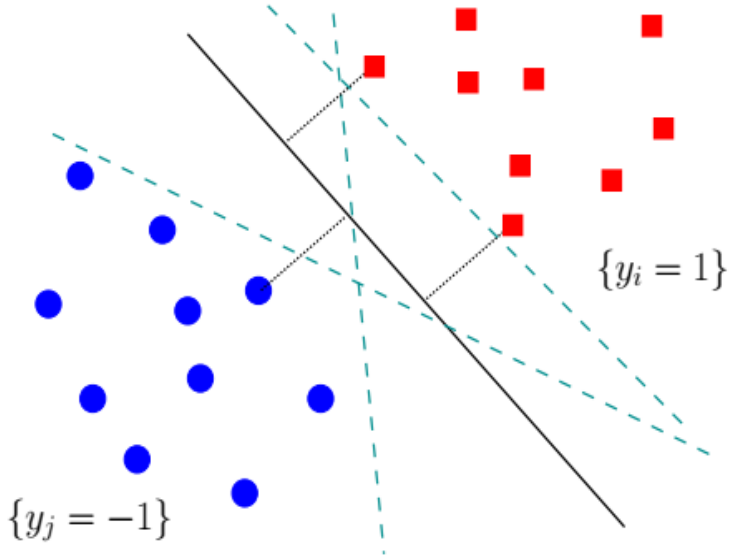


Figure 6.1: Potential separating hyperplanes (dashed, cyan) and the margin optimizing one (solid, black).

Support-Vector Machines are a set of learning methods used for classification, regression and outlier detection. They are among the best "off-the-shelf" supervised learning algorithms, if not even the best. Since only support-vectors are used for decisions SVMs are also memory efficient. They perform extremely effective if the number of dimensions is high, even in cases where the number of samples is smaller than the number of dimensions. The application of so called *kernel functions* is a way to generalize SVMs to nonlinear decision patterns, which makes SVMs very flexible and versatile.

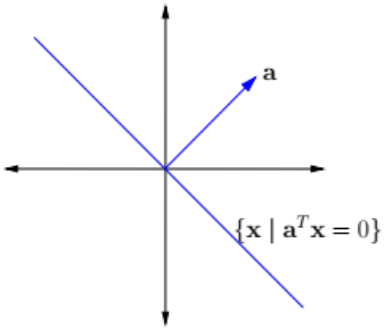
Hyperplanes and Margins

We commence by briefly considering the representation of hyperplanes and half-spaces in $\mathbb{R}^p$.
Suppose that $\mathbf{a} \in \mathbb{R}^p$ and $b \in \mathbb{R}$ are given.

a) Given $\mathbf{a} \in \mathbb{R}^p$

$$\{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{a}^T\mathbf{x} = 0\}$$

is the $(p-1)$-dimensional linear subspace orthogonal to $\mathbf{a}$.

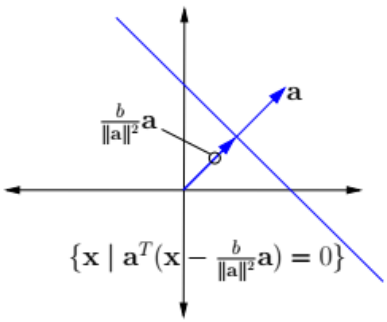


b) Given $\mathbf{a} \in \mathbb{R}^p$ and $b \in \mathbb{R}$

$$\{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{a}^T\mathbf{x} - b = 0\}$$

is the linear space shifted by the vector $\frac{\mathbf{b}}{\|\mathbf{a}\|^2}\mathbf{a}$. This holds since $\mathbf{a}^T\mathbf{x} - b = 0$ if and only if $\mathbf{a}^T\mathbf{x} - \frac{\mathbf{a}^T\mathbf{a}}{\|\mathbf{a}\|^2}b = 0$ if and only if $\mathbf{a}^T\left(\mathbf{x} - \frac{b}{\|\mathbf{a}\|^2}\mathbf{a}\right) = 0$.

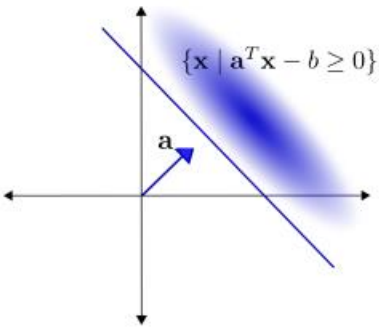
Hence, a linear space shifted by $\frac{b}{\|\mathbf{a}\|^2}\mathbf{a}$, is a hyperplane of distance $\frac{b}{\|\mathbf{a}\|}$ from $\{\mathbf{a}^T\mathbf{x} = 0\}$.



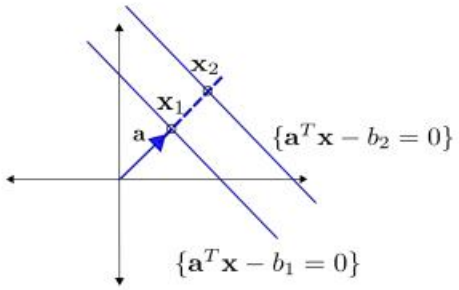
c) Given $\mathbf{a} \in \mathbb{R}^p$ and $b \in \mathbb{R}$

$$\{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{a}^T\mathbf{x} \geq b\}$$

represents a half-space of points lying on one side of the corresponding hyperplane.



d) Given $\mathbf{a} \in \mathbb{R}^p$, $b_1, b_2 \in \mathbb{R}$, the distance between two hyperplanes is required $H_1 = \{\mathbf{a}^T\mathbf{x} - b_1 = 0\}$ and $H_2 = \{\mathbf{a}^T\mathbf{x} - b_2 = 0\}$.



Both hyperplanes are parallel and orthogonal to $\mathbf{a}$. Pick $\mathbf{x}_1$ and $\mathbf{x}_2$ such that:

$$\begin{aligned} \mathbf{x}_1 = \lambda_1\mathbf{a}, \quad \mathbf{x}_2 = \lambda_2\mathbf{a} \\ \mathbf{a}^T\mathbf{x}_1 - b_1 = 0, \quad \mathbf{a}^T\mathbf{x}_2 - b_2 = 0. \end{aligned}$$

Then:

$$\begin{aligned} \lambda_1\mathbf{a}^T\mathbf{a} - b_1 = 0, \quad \lambda_2\mathbf{a}^T\mathbf{a} - b_2 = 0 \\ \lambda_1\|\mathbf{a}\|^2 - b_1 = 0, \quad \lambda_2\|\mathbf{a}\|^2 - b_2 = 0 \\ \lambda_1 = \frac{b_1}{\|\mathbf{a}\|^2}, \quad \lambda_2 = \frac{b_2}{\|\mathbf{a}\|^2}. \end{aligned}$$

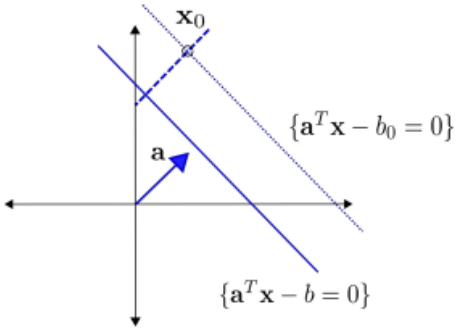
See that $\mathbf{x}_1 - \mathbf{x}_2$ is orthogonal to both hyperplanes and therefore the norm of this vector gives the distance between the two hyperplanes:

$$\begin{aligned} \|\mathbf{x}_1 - \mathbf{x}_2\| &= \|\lambda_2\mathbf{a} - \lambda_1\mathbf{a}\| = |\lambda_1 - \lambda_2|\|\mathbf{a}\| \\ &= \left(\frac{b_2}{\|\mathbf{a}\|^2} - \frac{b_1}{\|\mathbf{a}\|^2}\right)\|\mathbf{a}\| = \frac{|b_2 - b_1|}{\|\mathbf{a}\|}. \end{aligned}$$

Hence the distance between parallel H_1 and H_2 is:

$$\frac{1}{\|\mathbf{a}\|}|b_1 - b_2|.$$

e) Given $\mathbf{a} \in \mathbb{R}^p$, $b \in \mathbb{R}$ and $\mathbf{x}_0 \in \mathbb{R}^p$, the distance between the hyperplane $H_1 = \{\mathbf{x} \mid \mathbf{a}^T\mathbf{x} - b = 0\}$ and the point $\mathbf{x}_0$ is required.



Consider the auxiliary hyperplane containing $\mathbf{x}_0$:

$$H_0 = \{\mathbf{x} \mid \mathbf{a}^T\mathbf{x} - b_0 = 0\} = \{\mathbf{x} \mid \mathbf{a}^T\mathbf{x} - \mathbf{a}^T\mathbf{x}_0 = 0\}.$$

Note that $b_0 = \mathbf{a}^T\mathbf{x}_0$ since $\mathbf{a}^T\mathbf{x}_0 - b_0 = 0$. By the previous step, the distance between H and H_0 is:

$$\frac{1}{\|\mathbf{a}\|}|b - \mathbf{a}^T\mathbf{x}_0|.$$

The distance is called the margin of $\mathbf{x}_0$.

A hyperplane $\{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{a}^T\mathbf{x} = b\}$ is separating points of two classes, if one group is contained in the half-space $\{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{a}^T\mathbf{x} \geq b\}$ and the other group in $\{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{a}^T\mathbf{x} \leq b\}$, see Fig. 6.1. The minimum distance from points to the separating hyperplane is called *margin*. Our aim is to find a hyperplane which maximizes the margin.

The Optimal Margin Classifier

We now set out to formulate an optimization problem that will give us the separating hyperplane with maximum margins. Given a training set

$$\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}, \quad \mathbf{x}_i \in \mathbb{R}^p, \ y_i \in \{-1, 1\},$$

assume there exists a separating hyperplane

$$\{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{a}^T\mathbf{x} + b = 0\}.$$

Then for some $\gamma \geq 0$ and for all $i = 1, \ldots, n$, we have:

$$\begin{aligned} y_i = +1 \quad &\Longrightarrow \quad \mathbf{a}^T\mathbf{x}_k + b \geq \gamma \\ y_i = -1 \quad &\Longrightarrow \quad \mathbf{a}^T\mathbf{x}_k + b \leq -\gamma. \end{aligned}$$

Hence

$$y_i\big(\mathbf{a}^T\mathbf{x}_i + b\big) \geq \gamma \text{ for some } \gamma \geq 0, \text{ for all } i = 1, \ldots, n.$$

The minimum margin of the members of each class from the separating hyperplane is given by

$$\frac{1}{\|\mathbf{a}\|}\gamma.$$

The objective is to find a separating hyperplane such that this minimum margin is maximum.

$$\max_{\gamma,\mathbf{a},b} \frac{\gamma}{\|\mathbf{a}\|} \quad \text{s.t.} \quad y_i\big(\mathbf{a}^T\mathbf{x}_i + b\big) \geq \gamma \text{ for all } i = 1,\ldots,n.$$

This problem is scale invariant. In other words if $(\gamma, \mathbf{a}, b)$ is a solution so is $(2\gamma, 2\mathbf{a}, 2b)$. Therefore a normalization by γ leads to the following formulation

$$\begin{aligned}
&\min_{\gamma,\mathbf{a},b} \|\frac{\mathbf{a}}{\gamma}\| \quad \text{s.t.} \quad y_i\left((\frac{\mathbf{a}}{\gamma})^T\mathbf{x}_i + \frac{b}{\gamma}\right) \geq 1 \text{ for all } i = 1,\ldots,n \\
\iff &\min_{\mathbf{a},b} \|\mathbf{a}\| \quad \text{s.t.} \quad y_i\big(\mathbf{a}^T\mathbf{x}_i + b\big) \geq 1 \text{ for all } i = 1,\ldots,n \\
\iff &\min_{\mathbf{a},b} \frac{1}{2}\|\mathbf{a}\|^2 \quad \text{s.t.} \quad y_i\big(\mathbf{a}^T\mathbf{x}_i + b\big) \geq 1 \text{ for all } i = 1,\ldots,n.
\end{aligned}$$

This leads to the optimization problem for finding *Optimal Margin Classifier* (OMC):

$$\begin{aligned}
&\min_{\mathbf{a}\in\mathbb{R}^p,\, b\in\mathbb{R}} \frac{1}{2}\|\mathbf{a}\|^2 \\
&\text{such that } y_i\big(\mathbf{a}^T\mathbf{x}_i + b\big) \geq 1,\ i = 1,\ldots,n.
\end{aligned} \tag{OMC}$$

This is a quadratic optimization problem with linear inequality constraints, a special case of a convex general optimization problem.

- Assume $\mathbf{a}^*$ is an optimal solution of (OMC) and a nearest support point $\mathbf{x}_k$ and $\mathbf{x}_\ell$ from each class is known, as will become clear later from duality theory, then

  $$\begin{aligned}
  \mathbf{a}^{*T}\mathbf{x}_k + b &= 1, \\
  \mathbf{a}^{*T}\mathbf{x}_\ell + b &= -1.
  \end{aligned}$$

  Hence

  $$\mathbf{a}^{*T}\mathbf{x}_k + \mathbf{a}^{*T}\mathbf{x}_\ell + 2b = 0$$

  yielding

  $$b^* = -\frac{1}{2}\mathbf{a}^{*T}\big(\mathbf{x}_k + \mathbf{x}_\ell\big) \tag{6.1}$$

  as the optimum b-value.

  Even using a single point $\mathbf{x}_k$ satisfying $y_k(\mathbf{a}^{*T}\mathbf{x}_k + b^*) = 1$ is enough to obtain b^* as

  $$\begin{aligned}
  &y_k(\mathbf{a}^{*T}\mathbf{x}_k + b^*) = 1 \\
  \iff &\mathbf{a}^{*T}\mathbf{x}_k + b^* = y_k \quad (\text{since } y_k^2 = 1) \\
  \iff &b^* = y_k - \mathbf{a}^{*T}\mathbf{x}_k.
  \end{aligned}$$

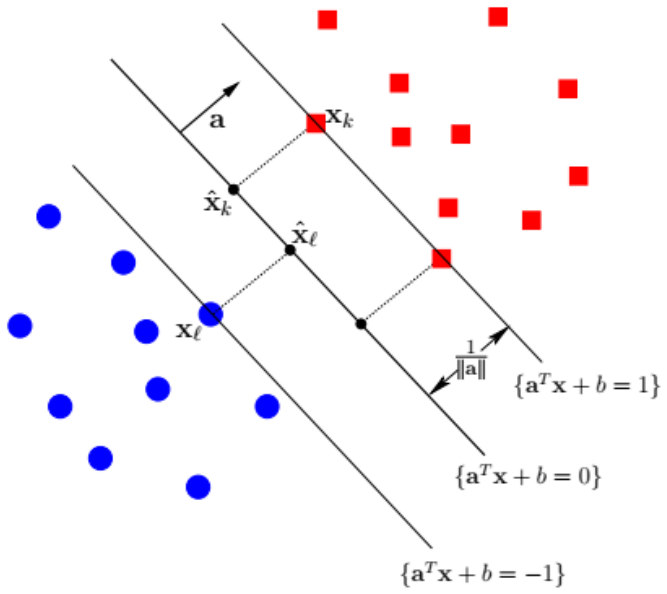


Figure 6.2: The margin maximizing hyperplane (solid, black) and its parallel sisters.

- The solution $(\mathbf{a}^*, b^*)$ is called the *optimal margin classifier*.
- It can be solved by commercial or public domain generic quadratic programming (QP) software.

Is the problem completely solved by now? Yes and no. We can do better by applying Lagrange duality theory. By this we not only get the optimum solution efficiently but also identify all support points. Moreover, a simple decision rule can be derived, which only depends on the support points. Moreover the case of non-separable data should also be considered.

SVM and Lagrange Duality

Let's start with the brief excursion on convex optimization.

- Normally a convex optimization problem can be formulated as,

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i,\ i = 1, \ldots, m. \\ & h_j(x) = 0,\ j = 1, \ldots, p \end{array}$$

  where $f_0(x)$ and $f_i(x)$ are convex and $h_j(x)$ are linear.

- Lagrangian function is derived from primal optimization problem:

$$L(x, \boldsymbol{\lambda}, \boldsymbol{\upsilon}) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{p} \upsilon_j h_j(x).$$

- Lagrangian dual function is defined as the infimum of Lagrangian function:

$$g(\boldsymbol{\lambda}, \boldsymbol{\upsilon}) = \inf_{x \in \mathcal{D}} L(x, \boldsymbol{\lambda}, \boldsymbol{\upsilon}) \tag{6.2}$$

$$\mathcal{D} = \bigcap_{i=0}^{m} \text{dom}(f_i) \cap \bigcap_{j=0}^{p} \text{dom}(h_j) \tag{6.3}$$

  The Lagrangian dual function is a concave function.

- The Lagrange dual problem is derived as follows,

$$\begin{aligned} &\text{maximize} \quad g(\boldsymbol{\lambda}, \boldsymbol{\upsilon}) \\ &\text{subject to} \quad \boldsymbol{\lambda} \geq 0. \end{aligned} \tag{6.4}$$

- Suppose that $\boldsymbol{\lambda}^*, \boldsymbol{\upsilon}^*$ are the optimal solutions of Lagrangian duality problem, and x^* is the optimal solution of primal optimization problem. The weak duality theorem states that

$$g(\boldsymbol{\lambda}^*, \boldsymbol{\upsilon}^*) \leq f_0(x^*).$$

  The strong duality holds if

$$g(\boldsymbol{\lambda}^*, \boldsymbol{\upsilon}^*) = f_0(x^*).$$

- If the constraints are linear then "the Slater's condition" holds, which implies that $g(\boldsymbol{\lambda}^*, \boldsymbol{\upsilon}^*) = f_0(x^*)$, i.e., strong duality holds, i.e., and therefore the duality gap is zero.

- If strong duality holds, then

$$\begin{aligned} f_0(x^*) &= g(\boldsymbol{\lambda}^*, \boldsymbol{\upsilon}^*) \\ &= \inf_x \left(f_0(x) + \sum_{i=1}^{m} \lambda_i^* f_i(x) + \sum_{i=1}^{p} \upsilon_i^* h_i(x) \right) \\ &\leq f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* f_i(x^*) + \sum_{i=1}^{p} \upsilon_i^* h_i(x^*) \\ &\leq f_0(x^*), \end{aligned}$$

  since $\lambda_i^* \geq 0$, $f_i(x^*) \leq 0$ and $h_i(x^*) = 0$. Hence, equality holds everywhere such that

  (i) x^* minimizes $L(x, \boldsymbol{\lambda}^*, \boldsymbol{\upsilon}^*)$,

(ii) $\lambda_i^* f_i(x^*) = 0, i = 1, \ldots, m$ (Complementary slackness), i.e.,

$$\begin{aligned} \lambda_i^* > 0 &\implies f_i(x^*) = 0 \\ f_i(x^*) < 0 &\implies \lambda_i^* = 0. \end{aligned}$$

- Karush-Kuhn-Tucher conditions (KKT) is defined by
  1. $f_i(x) \leq 0, i = 1, \ldots, m$, $h_j(x) = 0, j = 1, \ldots, p$ (primal constraints)
  2. $\lambda_i \geq 0, i = 1, \ldots, m$ (dual constraints)
  3. $\lambda_i f_i(x) = 0, i = 1, \ldots, m$ (complementary slackness)
  4. $\nabla_x L(x, \boldsymbol{\lambda}, \boldsymbol{\upsilon}) = 0$

Theorem 6.1. *If slater's condition is satisfied then the strong duality holds. If in addition f_i, h_j are differentiable, then for x^*, (λ^*, υ^*) to be primal and dual optimal, it is necessary and sufficient that the KKT condition holds.*

We can see the application to SVM. For example, given training set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \{-1, 1\}$. The primal optimization problem is given by

$$\begin{aligned} \min_{\mathbf{a} \in \mathbb{R}^p, b \in \mathbb{R}} \quad & \frac{1}{2}\|\mathbf{a}\|^2 \\ \text{s.t} \quad & y_i(\mathbf{a}^T \mathbf{x}_i + b) \geq 1, i = 1, \ldots, n \end{aligned} \tag{6.5}$$

The Lagrangian function writes as

$$L(\mathbf{a}, b, \boldsymbol{\lambda}) = \frac{1}{2}\|\mathbf{a}\|^2 - \sum_{i=1}^{n} \lambda_i (y_i(\mathbf{a}^T \mathbf{x}_i + b) - 1)$$

$$\frac{\partial(\mathbf{a}, b, \boldsymbol{\lambda})}{\partial \mathbf{a}} = \mathbf{a} - \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i = 0 \Longrightarrow \mathbf{a}^* = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i,$$

$$\frac{\partial(\mathbf{a}, b, \boldsymbol{\lambda})}{\partial b} = \sum_{i=1}^{n} \lambda_i y_i = 0 \Longrightarrow \sum_{i=1}^{n} \lambda_i y_i = 0$$

Dual function writes as

$$\begin{aligned} g(\boldsymbol{\lambda}) = L(\mathbf{a}^*, b^*, \boldsymbol{\lambda}) &= \frac{1}{2}\|\mathbf{a}^*\|^2 - \sum_{i=1}^{n} \lambda_i (y_i(\mathbf{a}^{*T} \mathbf{x}_i + b^*) - 1) \\ &= \sum_{i=1}^{n} \lambda_i + \frac{1}{2}(\sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i)^T (\sum_{j=1}^{n} \lambda_j y_j \mathbf{x}_j) - \sum_{i=1}^{n} \lambda_i y_i (\sum_{j=1}^{n} \lambda_j y_j \mathbf{x}_j)^T x_i - \sum_{i=1}^{n} \lambda_i y_i b^* \\ &= \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} y_i y_j \lambda_i \lambda_j \mathbf{x}_i^T \mathbf{x}_j \end{aligned}$$

Finally the dual problem can be obtained as

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & g(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} y_i y_j \lambda_i \lambda_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{s.t} \quad & \lambda_i \geq 0 \\ & \sum_{i=1}^{n} \lambda_i y_i = 0 \end{aligned} \tag{(DP)}$$

If λ_i^*'s are the solution of the dual problem, then $\mathbf{a}^* = \sum_{i=1}^{n} \lambda_i^* y_i \mathbf{x}_i$ and $b^* = y_k - \mathbf{a}^{*T} \mathbf{x}_k$ with $\mathbf{x}_k$ are some support vector. The Slater's condition is also satisfied, so strong duality holds. From KKT for optimal $\boldsymbol{\lambda}^*$ complementary slackness follows

$$\lambda_i^* (y_i (\mathbf{a}^{*T} \mathbf{x}_i + b^*) - 1) = 0, i = 1, \ldots, n$$

Hence

$$\lambda_i^* > 0 \Longrightarrow y_i (\mathbf{a}^{*T} \mathbf{x}_i + b^*) = 1 \tag{6.6}$$

$$\lambda_i^* = 0 \Longrightarrow y_i (\mathbf{a}^{*T} \mathbf{x}_i + b^*) \geq 1 \tag{6.7}$$

$\lambda_i^* > 0$ indicates supporting vectors, those which have smallest distance to the separating hyperplane. After solving (DP) the support vectors are determined by $\lambda_i^* > 0$.
Let $S = \{i | \lambda_i^* > 0\}$, $S_+ = \{i \in S | y_i = +1\}$ and $S_- = \{i \in S | y_i = -1\}$. Then

$$\mathbf{a}^* = \sum_{i \in S} \lambda_i^* y_i \mathbf{x}_i$$

$$b^* = -\frac{1}{2} \mathbf{a}^{*T} (\mathbf{x}_k + \mathbf{x}_l), \quad k \in S_+, l \in S_-$$

SVM can be then simply stated as follows

- Training set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$.
- Determine λ^* and $\mathbf{a}^*, b^*$
- For each new point $\mathbf{x}$, to find class label $y \in \{-1, 1\}$, first compute

$$d(x) = \mathbf{a}^{*T} \mathbf{x} + b^* = (\sum_{i \in S} \lambda_i^* y_i \mathbf{x}_i)^T \mathbf{x} + b^* = \sum_{i \in S} \lambda_i^* y_i \mathbf{x}_i^T \mathbf{x} + b^*$$

  and then predict $y = 1$, if $d(x) \geq 0$, otherwise $y = -1$.

Remarks:

- $|S|$ is normally much less than n.
- The decision only depends on the inner products $\mathbf{x}_i^T \mathbf{x}$ for support-vectors $\mathbf{x}_i, i \in S$.

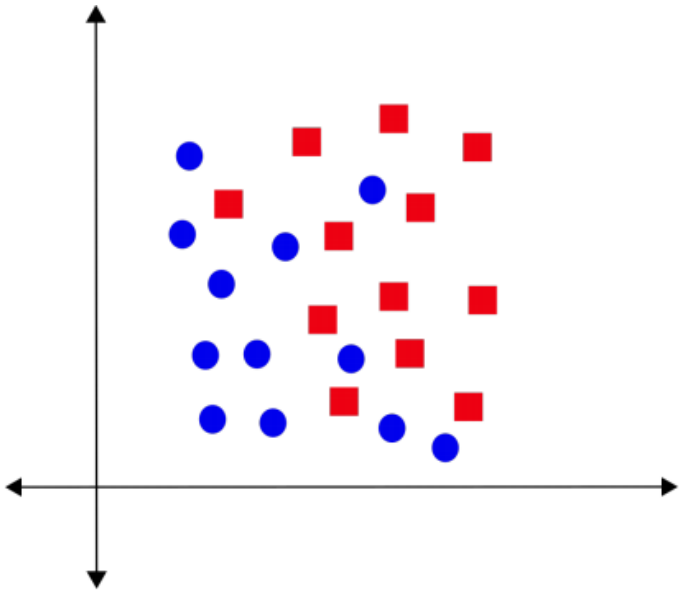
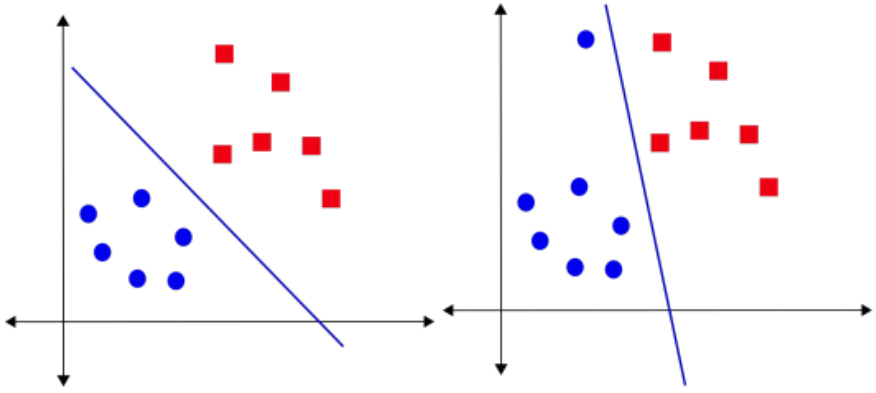


Figure 6.3: A linearly non-separable dataset

Robustness and Non-separability

So far, the assumption that there exists a separating hyperplane between two classes. What happens if not? For example, the points in Fig. 6.3 are not linearly separable. Moreover the optimum margin classifier is sensitive to outliers. Outliers cause drastic swing of the optimal margin classifier.



Both problems are addressed by the following approach: ℓ_1-regularization.

$$\begin{aligned} \min_{\mathbf{a}\in\mathbb{R}^p, b, \xi} \quad & \frac{1}{2}\|\mathbf{a}\|^2 - c\sum_{i=1}^{n}\xi_i \\ \text{s.t} \quad & y_i(\mathbf{a}^T\mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, \ldots, n \\ & \xi_i \geq 0, i = 1, \ldots, n \end{aligned} \tag{6.8}$$

For the optimal solution $\mathbf{a}^*, b^*$, it is allowed that margins are less than $\frac{1}{\|\mathbf{a}^*\|}$, i.e.,

$$y_i(\mathbf{a}^{*T}\mathbf{x}_i + b^*) \leq 1.$$

If $y_i(\mathbf{a}^{*T}\mathbf{x}_i + b^*) = 1 - \xi_i, \xi_i > 0$, then a cost of $c\xi_i$ is paid. Parameter c controls the balance between the two goals in (6.8).
Lagrangian for (6.8) is given by:

$$L(\mathbf{a}, b, \xi, \boldsymbol{\lambda}, \boldsymbol{\gamma}) = \frac{1}{2}\|\mathbf{a}\|^2 + c\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\lambda_i(y_i(\mathbf{a}^T\mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{n}\gamma_i\xi_i$$

where γ_i, λ_i's are Lagrangian multipliers. Analogously to the above, obtain the dual problem as

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i,j} y_i y_j \lambda_i \lambda_j \mathbf{x}_i^T\mathbf{x}_j \\ \text{s.t} \quad & 0 \leq \lambda_i \leq c \\ & \sum_{i=1}^{n}\lambda_i y_i = 0 \end{aligned} \tag{6.9}$$

Let λ_i^* be the optimum solution of (6.9). As before, let $S = \{i|\lambda_i^* > 0\}$. Corresponding $\mathbf{x}_i$'s are called support vectors. Then $\mathbf{a}^* = \sum_{i\in S}\lambda_i^* y_i \mathbf{x}_i$ is the optimum $\mathbf{a}$.
Complementary slackness conditions are

$$\lambda_i = 0 \Longrightarrow y_i(\mathbf{a}^{*T}\mathbf{x}_i + b_i) \geq 1 \tag{6.10}$$

$$\lambda_i = c \Longrightarrow y_i(\mathbf{a}^{*T}\mathbf{x}_i + b_i) \leq 1 \tag{6.11}$$

$$0 < \lambda_i < c \Longrightarrow y_i(\mathbf{a}^{*T}\mathbf{x}_i + b_i) = 1 \tag{6.12}$$

$$\tag{6.13}$$

If $0 < \lambda_k < c$ for some k ($\mathbf{x}_k$ support vector), then $b^* = y_k - \mathbf{a}^{*T}\mathbf{x}_k$ providing the optimum b.
Also it is possible to pick two support vectors $\mathbf{x}_k$ and $\mathbf{x}_\ell$ with $y_k = +1$ and $y_\ell = -1$, then

$$\left.\begin{aligned} b^* &= y_k - \mathbf{a}^{*T}\mathbf{x}_k \\ b^* &= y_\ell - \mathbf{a}^{*T}\mathbf{x}_\ell \end{aligned}\right\} \Longrightarrow 2b^* = -\mathbf{a}^{*T}(\mathbf{x}_k + \mathbf{x}_\ell),$$

hence $b^* = -\frac{1}{2}\mathbf{a}^{*T}(\mathbf{x}_k + \mathbf{x}_\ell)$ is the optimum b.
To classify a new point $\mathbf{x} \in R^p$, there are two classifiers.

- Hard classifier: first compute:

$$\mathbf{a}^{*T}\mathbf{x} + b^* = (\sum_{i\in S}\lambda_i^* y_i \mathbf{x}_i)^T\mathbf{x} + b^* = \sum_{i\in S}\lambda_i^* y_i \mathbf{x}_i^T\mathbf{x} + b^* = d(\mathbf{x}).$$

  Decide $y = 1$ if $d(\mathbf{x}) \geq 0$, otherwise $y = -1$.

- Soft classifier: Compute $d(\mathbf{x}) = h(\mathbf{a}^{*T}\mathbf{x} + b^*)$ where

$$h(t) = \begin{cases} -1, & t < -1 \\ t, & -1 \le t \le +1 \\ +1, & t > 1 \end{cases} \tag{6.14}$$

  $d(\mathbf{x})$ is a real number in $[-1, +1]$ if $\mathbf{a}^{*T}\mathbf{x} + b^* \in [-1, +1]$, i.e., if x is residing in the overlapping area.

Both classifiers only depend on the inner products $\mathbf{x}_i^T\mathbf{x} = \langle \mathbf{x}_i, \mathbf{x} \rangle$ with support vectors $\mathbf{x}_i, i \in S$.

The SMO Algorithm

The Sequential Minimal Optimization (SMO) algorithm is an algorithm to solve the dual problem, which is given as

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & W(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} y_i y_j \lambda_i \lambda_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{s.t.} \quad & 0 \le \lambda_i \le c \\ & \sum_{i=1}^{n} \lambda_i y_i = 0. \end{aligned} \tag{6.15}$$

Assume $\boldsymbol{\lambda}$ is a feasible point, i.e., $\boldsymbol{\lambda}$ satisfies the constraints. Note that *cyclic coordinate optimization* does not work, since, e.g.,

$$\lambda_1 y_1 = -\sum_{i=2}^{n} \lambda_i y_i \quad \text{or} \quad \lambda_1 = -y_1 \sum_{i=2}^{n} \lambda_i y_i.$$

Hence each λ_j is determined by fixing λ_i, $i \neq j$. The idea is to update at least two λ_j simultaneously, which is SMO algorithm.

Algorithm 4 SMO algorithm

1: **procedure** SMO
2: **repeat**
3: 1. Select a pair (i, j) to be updated next, the one which promises the most progress
4: 2. Optimize $W(\boldsymbol{\lambda})$ w.r.t. λ_i and λ_j while keeping λ_k, $k \neq i, j$, fixed.
5: **until** Convergence

One can check KKT within a tolerance limit $\epsilon = 0.01$ or 0.001, to verify Convergence. Optimize $W(\boldsymbol{\lambda})$ w.r.t. λ_1, λ_2 with $\lambda_3, \ldots, \lambda_n$ fixed and $\boldsymbol{\lambda}$ feasible. It holds

$$\lambda_1 y_1 + \lambda_2 y_2 = -\sum_{i=3}^{n} \lambda_i y_i = \zeta, \quad \zeta \text{ fixed.}$$

Derive the following:

$$\begin{aligned} 0 \leq \lambda_1, \lambda_2 \leq c \\ L \leq \lambda_2 \leq H. \end{aligned} \tag{6.16}$$

Moreover:

$$\lambda_1 = y_1(\zeta - \lambda_2 y_2). \tag{6.17}$$

Hence $W(\lambda_1, \ldots, \lambda_n) = W(y_1(\zeta - \lambda_2 y_2), \lambda_2, \underbrace{\lambda_3, \ldots, \lambda_n}_{\text{fixed}})$ and therefore the objective function turns out to be a quadratic function of λ_2 and it can be written as :

$$\gamma_2 \lambda_2^2 + \gamma_1 \lambda_2 + \gamma_0.$$

Determine the maximum by differentiation:

$$2\gamma_2 \lambda_2 + \gamma_1 = 0 \implies \lambda_2 = -\frac{\gamma_1}{2\gamma_2}$$

with optimum solution $\lambda_2^{(r)}$. The final solution, following (6.16), is

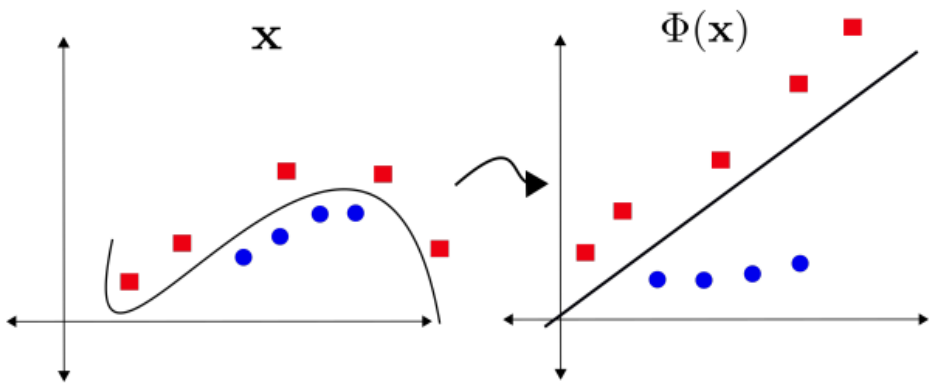
$$\lambda_2^{(c)} = \begin{cases} H & \text{if } \lambda_2^{(r)} > H \\ \lambda_2^{(r)} & \text{if } L \leq \lambda_2^{(r)} \leq H \\ L & \text{if } \lambda_2^{(r)} < L \end{cases} \tag{6.18}$$

λ_1 is computed using (6.17).
It still remains to clarify:

- What is the best choice of the next pair (i, j) to update?
- How to update the coefficients $\gamma_0, \gamma_1, \gamma_2$ in the run of SMO.
- The algorithm converges, however, the right choice of (i, j) in each step accelerates the rate of Convergence.
- Generalization of SMO algorithm [OFG97]

Kernels

Instead of applying SVM to the raw data ("attributes") $\mathbf{x}_i$, one can apply it to transformed data ("features") $\Phi(\mathbf{x}_i)$. The function Φ is called feature mapping. The aim of kernels is to achieve better separability.



Consider the dual SVM problem.

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & g(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} y_i y_j \lambda_i \lambda_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{s.t.} \quad & 0 \leq \lambda_i \leq c \\ & \sum_{i=1}^{n} \lambda_i y_i = 0. \end{aligned} \tag{6.19}$$

$g(\boldsymbol{\lambda})$ only depends on the inner products $\mathbf{x}_i^T \mathbf{x}_j$. Substitute $\mathbf{x}_i$ by $\Phi(\mathbf{x}_i)$ and use the same inner product $\langle , \rangle$. Replace $\mathbf{x}_i^T \mathbf{x}_j$ by,

$$\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_j).$$

In other words, the inner product of the features are given by the function $K(\mathbf{x}_i, \mathbf{x}_j)$. Note that $K(\mathbf{x}_i, \mathbf{x}_j)$ is often easier to compute than $\Phi(\mathbf{x})$ itself.
The intuition why we need kernels is that if $\Phi(\mathbf{x}), \Phi(\mathbf{y})$ are close, $\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$ is large. If $\Phi(\mathbf{x}) \perp \Phi(\mathbf{y})$ then $\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle = 0$. Hence, $K(\mathbf{x}_i, \mathbf{x}_j)$ measures how similar $\mathbf{x}$ and $\mathbf{y}$ are. What is needed for Kernel based methods is an inner product in some feature space $\{\Phi(\mathbf{x}) | \mathbf{x} \in \mathbb{R}^p\}$.

Example. Given $\mathbf{x}, \mathbf{z} \in \mathbb{R}^p$ define the Kernel functions as

$$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^2 = \left(\sum_{i=1}^{p} x_i z_i \right)^2$$

The question is whether there is some function Φ such that $\langle \mathbf{x}, \mathbf{z} \rangle^2$ is an inner product in the feature space. Let $p = 2$ and $\mathbf{x} = (x_1, x_2)^T$ and $\mathbf{z} = (z_1, z_2)^T$. Use $\Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2} x_1 x_2) : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$. Then

$$\begin{aligned} \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2 x_1 x_2 z_1 z_2 \\ &= (x_1 z_1 + x_2 z_2)^2 = \langle \mathbf{x}, \mathbf{z} \rangle^2. \end{aligned}$$

Example. (Gaussian Kernel) Given $\mathbf{x}, \mathbf{z} \in \mathbb{R}^p$, the kernel is defined as

$$K(\mathbf{x}, \mathbf{z}) = \exp(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2})$$

Question: is there a feature mapping Φ and a feature space with inner product specified as above?

Definition 6.2. Kernel $K(\mathbf{x}, \mathbf{z})$ is called valid, if there exists a feature Φ such that $K(\mathbf{x}, \mathbf{z}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle$ for all $\mathbf{x}, \mathbf{z} \in \mathbb{R}^p$

Theorem 6.3 (Mercer's theorem)**.** *Given $K : \mathbb{R}^p \times \mathbb{R}^p \longrightarrow \mathbb{R}$, K is a valid kernel if and only if for any $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ the kernel matrix $(K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1,\ldots,n}$ is nonnegative definite.*

Proof. (Only $\Longrightarrow$) If K is valid then there exists a function Φ such that:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)) \rangle = \langle \Phi(\mathbf{x}_j), \Phi(\mathbf{x}_i) \rangle = K(\mathbf{x}_j, \mathbf{x}_i)$$

Moreover,

$$\mathbf{z}^T (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j} \mathbf{z} = \sum_{k,l} z_k z_l \langle \Phi(\mathbf{x}_k), \Phi(\mathbf{x}_l) \rangle = \langle \sum_k z_k \Phi(\mathbf{x}_k), \sum_l z_l \Phi(\mathbf{x}_l) \rangle \geq 0.$$

□

Example. (Polynomial Kernel) Consider the following kernel:

$$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z} + c)^d, \mathbf{x}, \mathbf{z} \in \mathbb{R}^p, c \in \mathbb{R}, d \in \mathbf{N}, d \geq 2.$$

Feature space of this kernel is of dimension $\binom{p+d}{d}$, containing all monomials of degree less that or equal to d.
(Exercise: Determine $\Phi(x)$.)

Kernels can also be constructed over infinite dimensional spaces,e.g, function space or probability distributions, providing a lot of modeling power. The solution of the optimal problem is still a convex problem with linear constraints.