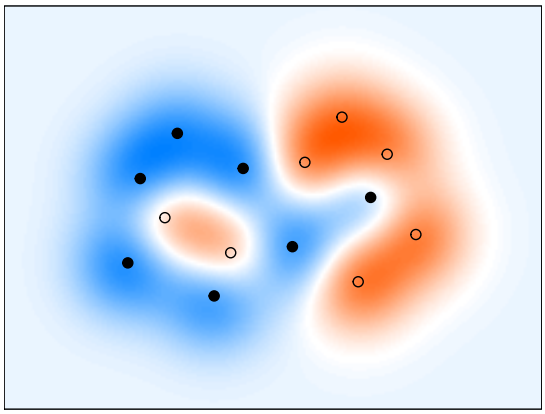
Support Vector Machines: A Simple Tutorial

Alexey Nefedov
svmtutorial@gmail.com



Contents

1. Introduction

In this section we review several basic concepts that are used to define support vector machines (SVMs) and which are essential for their understanding. We assume that the reader is familiar with real coordinate space, inner product of vectors, and vector norm (a brief review of these concepts is given in Appendix).

1.1. Classification problem

We consider a pattern classification problem which is formulated in the following way. There is a large, perhaps infinite, set of objects (observations, patterns, etc.) which can be classified into two classes (that is, assigned to two sets). We do not have an algorithm that does this classification, but we have a sample of objects with known class labels. Using these classification examples, we want to define an algorithm that will classify objects from the entire set with the minimum error.

Objects in a classification problem are represented by vectors from some vector space V. Although SVMs can be used in arbitrary vector spaces supplied with the inner product or kernel function, in most practical applications vector space V is simply the n-dimensional real coordinate space $\mathbf{R}^n$. In this space, vector $\boldsymbol{x}$ is a set of n real numbers x_i called the components of the vector: $\boldsymbol{x} = (x_1, x_2, ..., x_n)$.

Training set

A sample of objects with known class labels is called a *training set* and is written as

$$(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_l, y_l),$$

where $y_i \in \{-1, 1\}$ is the class label of vector $\boldsymbol{x}_i$, and l is the size of the training set.

Decision function

A classification algorithm (classifier) is represented by a *decision function*

$$f(\boldsymbol{x}) : V \to \{-1, 1\}$$

such that $f(\boldsymbol{x}) = 1$ if the classifier assigns $\boldsymbol{x}$ to the first class, and $f(\boldsymbol{x}) = -1$ if the classifier assigns $\boldsymbol{x}$ to the second class.

1.2. Equation of a hyperplane

In coordinate space $\mathbf{R}^n$ equation

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0 \tag{1.1}$$

$$\sum_{k=1}^{n} w_k x_k + b = 0$$

defines a $(n-1)$-dimensional set of vectors called *hyperplane*. That is, for a given non-zero vector $\boldsymbol{w} = (w_1, w_2, ..., w_n) \in \mathbf{R}^n$ and a scalar $b \in \mathbf{R}$, the set of all vectors $\boldsymbol{x} = (x_1, x_2, ..., x_n) \in \mathbf{R}^n$ satisfying equation (1.1) forms a hyperplane (Figure 1a). We will denote this hyperplane by letter π or by $\pi(\boldsymbol{w}, b)$.

The term "hyperplane" means that the dimensionality of the plane is by one less than the dimensionality of the entire space $\mathbf{R}^n$. For example, a point is a hyperplane in $\mathbf{R}$; a line is a hyperplane in $\mathbf{R}^2$; a plane is a hyperplane in $\mathbf{R}^3$; a three-dimensional space is a hyperplane in $\mathbf{R}^4$, and so on.

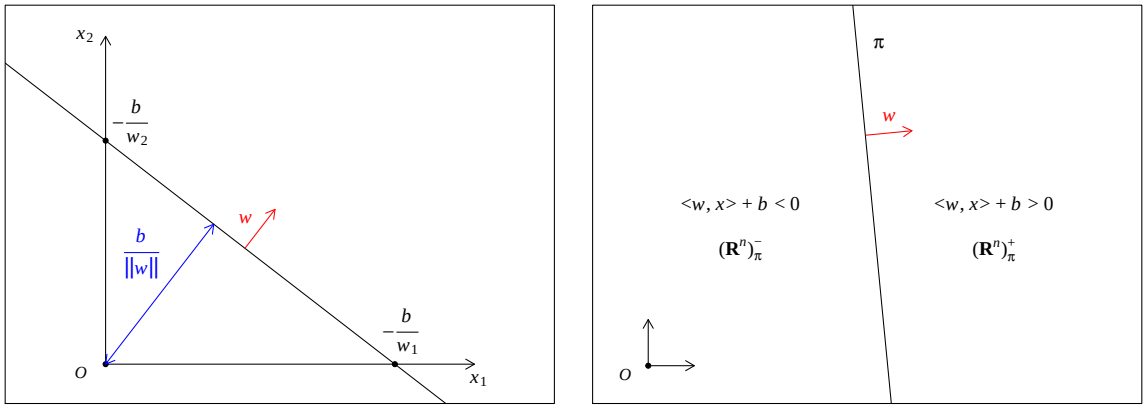
$\boldsymbol{w}$ – normal vector

b – intercept

Vector $\boldsymbol{w}$ is called the *normal vector* of the hyperplane, and number b is called the *intercept* of the hyperplane. The normal vector defines the orientation of the hyperplane in space, while the ratio between $||\boldsymbol{w}||$ and b (not the intercept alone) defines the distance between the hyperplane and the origin of space[1]. The normal vector is perpendicular to all vectors parallel to the hyperplane. That is, if $\boldsymbol{z} = \boldsymbol{x}_1 - \boldsymbol{x}_2$ such that $\langle \boldsymbol{w}, \boldsymbol{x}_1 \rangle + b' = 0$ and $\langle \boldsymbol{w}, \boldsymbol{x}_2 \rangle + b' = 0$ for some b' then $\langle \boldsymbol{w}, \boldsymbol{z} \rangle = 0$.

Half-spaces

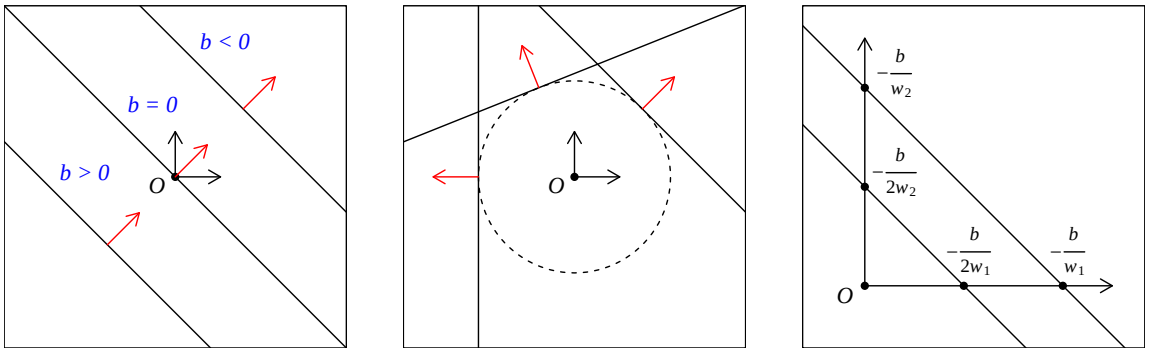
Hyperplane π divides coordinate space $\mathbf{R}^n$ into two parts located sidewise of the hyperplane, called positive and negative *half-spaces*: $(\mathbf{R}^n)^+_\pi$ and $(\mathbf{R}^n)^-_\pi$. The positive half-space is pointed by the normal vector of the hyperplane (Figure 1b). For any vector $\boldsymbol{x} \in (\mathbf{R}^n)^+_\pi$ we have $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b > 0$, while for any $\boldsymbol{x} \in (\mathbf{R}^n)^-_\pi$ we have $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b < 0$.



(a) (b)

Figure 1: (a) Hyperplane in two-dimensional space $\mathbf{R}^2$ is a line. For the depicted line we can infer that $b < 0$, $w_1 > 0$, $w_2 > 0$. (b) Negative and positive half-spaces defined by hyperplane $\pi(\boldsymbol{w}, b)$.

[1] The origin of space is zero vector $\mathbf{0} = (0, 0, ..., 0)$; in figures, we denote it by capital letter O.



(a) (b) (c)

Figure 2: Understanding the meaning of hyperplane parameters $\boldsymbol{w}$ and b (see text).

Ambiguity of hyperplane parameters

A single hyperplane is defined by infinite number of parameters $\boldsymbol{w}, b$. Indeed, multiplying equation (1.1) by arbitrary constant $c \neq 0$ we see that parameters $c\boldsymbol{w}, cb$ define the same hyperplane[2]. In other words, if two hyperplanes π_1 and π_2 are defined by parameters $\boldsymbol{w}_1, b_1$ and $\boldsymbol{w}_2, b_2$, and $\boldsymbol{w}_2 = c\boldsymbol{w}_1$, $b_2 = cb_1$, then π_1 and π_2 are the same hyperplanes. Since we can arbitrary scale parameters $\boldsymbol{w}, b$ defining fixed hyperplane π, we can choose $\boldsymbol{w}, b$ such that $||\boldsymbol{w}|| = 1$. Note that this pair of parameters is unique for any hyperplane[3].

Distance between vector and hyperplane

The distance $\rho(\boldsymbol{x}, \pi)$ between a vector $\boldsymbol{x}$ and a hyperplane $\pi(\boldsymbol{w}, b)$ can be calculated according to the following equation:

$$\rho(\boldsymbol{x}, \pi) = \frac{\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b}{||\boldsymbol{w}||}. \tag{1.2}$$

Note that this is a signed distance: $\rho(\boldsymbol{x}, \pi) > 0$ when $\boldsymbol{x} \in (\mathbf{R}^n)^+_\pi$, and $\rho(\boldsymbol{x}, \pi) < 0$ when $\boldsymbol{x} \in (\mathbf{R}^n)^-_\pi$. Obviously, $\rho(\boldsymbol{x}, \pi) = 0$ when $\boldsymbol{x} \in \pi$. If $||\boldsymbol{w}|| = 1$, then the equation for the distance is simply $\rho(\boldsymbol{x}, \pi) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$.

It follows from equation (1.2) that the distance between the origin of space and hyperplane π is equal to $\frac{b}{||\boldsymbol{w}||}$. This simple fact allows us to make several useful observations regarding the position and orientation of the hyperplane in space, and how parameters b and $\boldsymbol{w}$ affect them. These observations will help us later in considerations related to the maximum margin hyperplane (subsection 2.1).

How $\boldsymbol{w}$ and b define position of hyperplane

1. The origin of space is in the positive half-space of the hyperplane $\pi(\boldsymbol{w}, b)$ if $b > 0$, and in the negative half-space if $b < 0$. If $b = 0$ then the hyperplane passes through the origin (Figure 2a).

2. By increasing the absolute value $|b|$ of the intercept, we move the hyperplane parallel to itself in the direction from the origin. By decreasing $|b|$, we move the hyperplane towards the origin (Figure 2a).

3. By changing the normal vector $\boldsymbol{w}$ in a way that preserves its norm, we move the hyperplane in a circle around the origin; the radius of the circle is $\frac{|b|}{||\boldsymbol{w}||}$ (Figure 2b).

[2] Note that if $c < 0$ then the positive and negative half-spaces will swap around.

[3] We can also consider parameters $-\boldsymbol{w}, -b$ which define the same hyperplane (and of course $|| - \boldsymbol{w}|| = ||\boldsymbol{w}|| = 1$), but the positive and negative half-spaces will be swapped around.

4. By reducing the length of the normal vector $\boldsymbol{w}$ in a way that preserves its direction, we move the hyperplane parallel to itself from the origin. By increasing the length of the normal vector $\boldsymbol{w}$ in a way that preserves its direction, we move the hyperplane towards the origin (Figure 2c). Thus, a hyperplane can be moved parallel to itself not only by changing intercept b, but also by scaling normal vector $\boldsymbol{w}$.

1.3. Hyperplane separating two classes. Margin

We say that a hyperplane $\pi(\boldsymbol{w}, b)$ *separates* two classes (sets) of vectors C_1 and C_2 if either

$$\begin{aligned} \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b > 0, \ \forall \boldsymbol{x} \in C_1 \\ \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b < 0, \ \forall \boldsymbol{x} \in C_2 \end{aligned} \tag{1.3}$$

or

$$\begin{aligned} \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b < 0, \ \forall \boldsymbol{x} \in C_1 \\ \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b > 0, \ \forall \boldsymbol{x} \in C_2. \end{aligned}$$

Linearly separable classes

Two classes are called *linearly separable* if there exists at least one hyperplane that separates them. If hyperplane $\pi(\boldsymbol{w}, b)$ separates classes C_1 and C_2 according to (1.3) then decision function

$$f(\boldsymbol{x}) = \operatorname{sgn}\{\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b\} = \begin{cases} 1, \text{ if } \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b \geq 0 \\ -1, \text{ if } \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b < 0 \end{cases} \tag{1.4}$$

gives us a classifier that correctly classifies all vectors from C_1 and C_2.

It is clear that for two linearly separable classes that are finite there always exist an infinite number of hyperplanes (with differently oriented $\boldsymbol{w}$ and different b) that separate them. Which one should be used to define a classifier? Support vector machine chooses the one with the maximum margin. For a hyperplane π separating classes C_1 and C_2, its

Margin

margin $m(\pi, C_1, C_2)$ is defined as the distance between π and class C_1, plus the distance between π and class C_2 (Figure 3a):

$$m(\pi, C_1, C_2) = \rho(\pi, C_1) + \rho(\pi, C_2).$$

Here, the distance between hyperplane π and a set of vectors C is defined as the minimal distance between π and vectors from C:

$$\rho(\pi, C) = \min_{\boldsymbol{x} \in C} |\rho(\pi, \boldsymbol{x})|.$$

Note that in this definition we are using the absolute value of the signed distance $\rho(\pi, \boldsymbol{x})$ defined by equation (1.2), in order for this definition to make sense when all, or some, vectors from C lie in the negative half-space of π.

Equivalently, the margin can be defined as the distance between classes C_1 and C_2 measured along the normal vector $\boldsymbol{w}$ (Figure 3b). If C_1^w is the set containing projections of all vectors from C_1 onto the line parallel to vector $\boldsymbol{w}$, and C_2^w is the set containing similar projections of all vectors from C_2, then

$$m(\pi, C_1, C_2) = \rho(C_1^w, C_2^w),$$

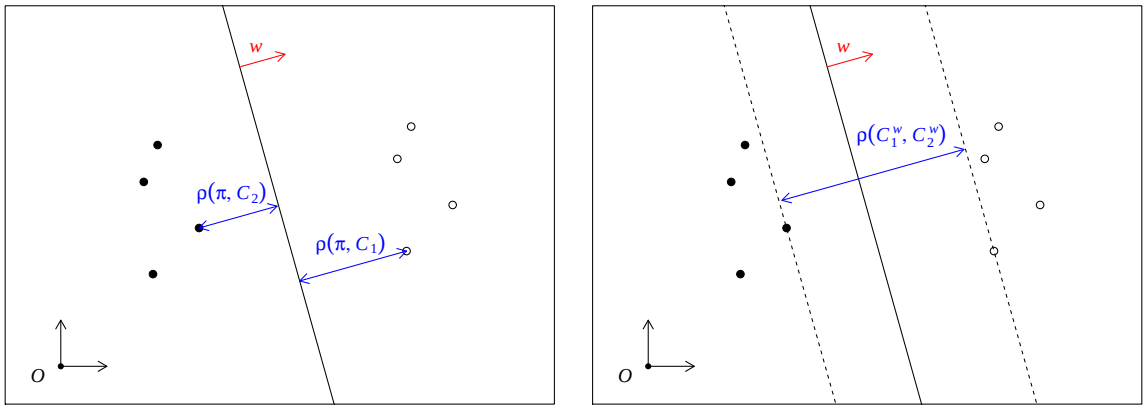


Figure 3: (a) Margin of hyperplane π is the distance from π to the first class (white points) plus the distance from π to the second class (black points). (b) Equivalently, it can be defined as the distance between two classes measured along the normal vector $\boldsymbol{w}$ of the hyperplane.

where

$$\rho(C_1^w, C_2^w) = \min_{\substack{\boldsymbol{x}_1 \in C_1^w \\ \boldsymbol{x}_2 \in C_2^w}} \rho(\boldsymbol{x}_1, \boldsymbol{x}_2).$$

Note the following two properties of margin. First, as long as the hyperplane lies between classes C_1 and C_2, its margin only depends on the normal vector $\boldsymbol{w}$, and does not depend on the intercept b. Second, for any hyperplane separating two classes, its margin can not be larger than the distance between the classes:

$$m(\pi, C_1, C_2) \leq \rho(C_1, C_2),$$

where

$$\rho(C_1, C_2) = \min_{\substack{\boldsymbol{x}_1 \in C_1 \\ \boldsymbol{x}_2 \in C_2}} \rho(\boldsymbol{x}_1, \boldsymbol{x}_2).$$

Clearly, there exist infinitely many hyperplanes with the maximum margin equal to $\rho(C_1, C_2)$, all of which are perpendicular to the shortest line segment connecting C_1 and C_2 (Figure 4).

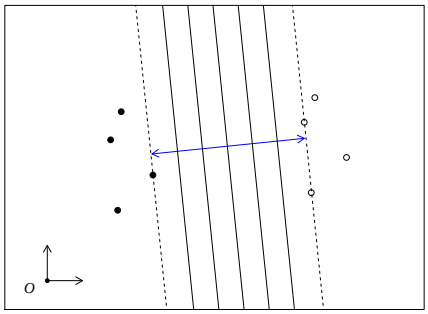


Figure 4: All five hyperplanes shown here have the same margin equal to $\rho(C_1, C_2)$.

2. Maximum margin hyperplane for linearly separable classes

Suppose we have two linearly separable classes of training vectors. Support vector machine defined on such a training set is a classifier with decision function

$$f(\boldsymbol{x}) = \operatorname{sgn}\{\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b\}, \tag{2.1}$$

where $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$ is an equation of hyperplane that separates the two classes (see (1.3)), has maximum margin, and is equidistant from both classes (Figure 5). In this section we consider an optimization problem which is being solved in order to obtain parameters $\boldsymbol{w}, b$ of this hyperplane, and explain where this problem comes from. We also consider important properties of the maximum margin hyperplane. Some concepts from calculus and optimization theory that are used in this section are briefly reviewed in Appendix.

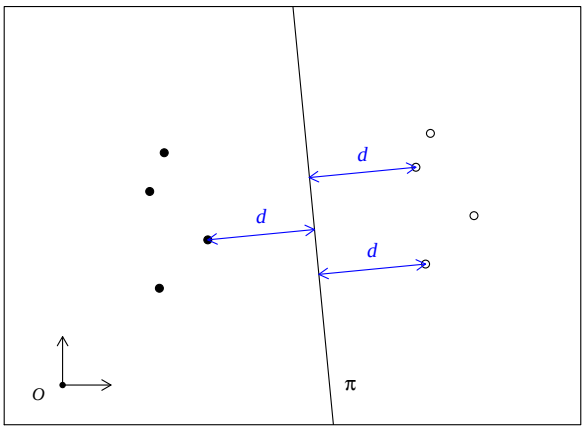


Figure 5: Maximum margin hyperplane for two linearly separable classes: $d = \rho(\pi, C_1) = \rho(\pi, C_2)$ is maximized.

2.1. Primary optimization problem

Parameters $\boldsymbol{w}, b$ of the SVM hyperplane can be found as a solution to the following optimization problem:

$$\frac{1}{2}||\boldsymbol{w}||^2 \to \min_{\boldsymbol{w},b} \tag{2.2}$$

subject to

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b \geq 1, \ \forall \boldsymbol{x} \in C_1 \tag{2.3}$$

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b \leq -1, \ \forall \boldsymbol{x} \in C_2, \tag{2.4}$$

where C_1 and C_2 are two classes of training examples.

In coordinate form, this problem is written as

$$\frac{1}{2}\sum_{k=1}^{n} w_k^2 \to \min_{\boldsymbol{w},b} \tag{2.5}$$

s.t.

$$\sum_{k=1}^{n} w_k x_k + b \geq 1,\ \forall \boldsymbol{x} \in C_1 \tag{2.6}$$

$$\sum_{k=1}^{n} w_k x_k + b \leq -1,\ \forall \boldsymbol{x} \in C_2. \tag{2.7}$$

Note that parameter b is one of the optimization variables, although it is not present in the objective function (2.5). Two sets of constraints (2.6), (2.7) can be written in a unified way as

$$y_i \left(\sum_{k=1}^{n} w_k x_{ik} + b \right) \geq 1, \quad i = 1, 2, ..., l. \tag{2.8}$$

where $y_i = 1$ if $\boldsymbol{x}_i \in C_1$, and $y_i = -1$ if $\boldsymbol{x}_i \in C_2$, l is the total number of training vectors $\boldsymbol{x}_i$, and by x_{ik} we denote the k-th component of vector $\boldsymbol{x}_i$.

Optimization problem (2.5)-(2.7) has quadratic objective function (2.5) and linear constraints (2.6), (2.7). Such problems are called *quadratic programming* problems. Their properties are well known and there are quite efficient algorithms for solving these problems.

Primary problem has unique solution

Note that objective function (2.5) is strictly convex (since the matrix of its second-order derivatives – the Hessian – is positive definite), and the feasible region defined by linear inequalities (2.6)-(2.7) is also convex. Therefore, this problem will have a unique solution (global minimum) $\boldsymbol{w}^*, b^*$[4]. In case when two classes are not linearly separable, the feasible region defined by constraints (2.6)-(2.7) will be empty, and the problem will have no solution. It will also have no solution when the training set contains only one class.

Why parameters of the maximum margin hyperplane can be found by solving problem (2.2)-(2.4)? To answer this question, let us transform this problem into an equivalent one that has more apparent geometrical interpretation. First, minimizing $\frac{1}{2}||\boldsymbol{w}||^2$ is equivalent to minimizing $||\boldsymbol{w}||$, which in turn is equivalent to maximizing $1/||\boldsymbol{w}||$, so we can rewrite problem (2.2)-(2.4) as follows:

$$\frac{1}{||\boldsymbol{w}||} \to \max_{\boldsymbol{w},b}$$

s.t.

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b \geq 1,\ \forall \boldsymbol{x} \in C_1 \tag{2.9}$$

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b \leq -1,\ \forall \boldsymbol{x} \in C_2. \tag{2.10}$$

[4]Note that the maximum margin hyperplane can be defined by infinite number of parameters $\boldsymbol{w}, b$; it is the solution of problem (2.5)-(2.7) which is unique.

Second, we can divide constraints (2.9), (2.10) by a positive number $||\boldsymbol{w}||$:

$$\frac{1}{||\boldsymbol{w}||} \to \max_{\boldsymbol{w},b}$$

s.t.

$$\frac{\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b}{||\boldsymbol{w}||} \geq \frac{1}{||\boldsymbol{w}||},\ \forall \boldsymbol{x} \in C_1$$

$$\frac{\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b}{||\boldsymbol{w}||} \leq -\frac{1}{||\boldsymbol{w}||},\ \forall \boldsymbol{x} \in C_2.$$

Recalling that $\frac{\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b}{||w||}$ is the distance $\rho(\pi, \boldsymbol{x})$ between hyperplane $\pi(\boldsymbol{w}, b)$ and point $\boldsymbol{x}$ (see equation (1.2)), and introducing new variable $d = 1/||\boldsymbol{w}||$, we get the following problem which is equivalent to (2.2)-(2.4):

Primary problem in geometrical terms

$$d \to \max_{\boldsymbol{w},b}$$

s.t.

$$\rho(\pi, \boldsymbol{x}) \geq d,\ \forall \boldsymbol{x} \in C_1 \tag{2.11}$$

$$\rho(\pi, \boldsymbol{x}) \leq -d,\ \forall \boldsymbol{x} \in C_2. \tag{2.12}$$

That is, find parameters $\boldsymbol{w}, b$ that maximize margin $m = 2d$ between π, C_1 and C_2.

Solving primary problem: geometrical insight

Geometrically, connection between parameters $\boldsymbol{w}$, b and d can be illustrated in the following way. Let us draw a spherical hull of radius $d = 1/||\boldsymbol{w}||$ around each training point $\boldsymbol{x}$. Consider some feasible hyperplane $\pi(\boldsymbol{w}, b)$. According to constraints (2.11), (2.12), this hyperplane must separate our points together with their hulls (Figure 6a). Now suppose we want to increase d twofold. Since d is a function of $||\boldsymbol{w}||$, we have to decrease $||\boldsymbol{w}||$ twofold. If we divide vector $\boldsymbol{w}$ by two, we move our hyperplane parallel to itself further from the origin. However, if we divide by two vector $\boldsymbol{w}$ and intercept b, we do not move the hyperplane. This way, downscaling $\boldsymbol{w}$ and b, we increase the radius of the hulls while keeping hyperplane π in the same position and orientation, until at least one hull touches it (Figure 6b). If we have space to move π parallel to itself away from the hull that touches it, we can do it by changing b only, and let the hulls grow further. At some point, our hulls will reach maximum size d achievable for hyperplanes with normal vectors collinear to $\boldsymbol{w}$ (Figure 6c). If we have space for the hulls to grow further, we can change orientation of π by changing components of vector $\boldsymbol{w}$, while keeping $||\boldsymbol{w}||$ equal to the current value of $1/d$ (see the end of subsection 1.2). Doing so and adjusting b, we keep π feasible and increase d until we arrive to the optimal configuration (Figure 6d).

2.2. Dual problem

The concept of duality plays an important role in the optimization theory. It turns out that for many optimization problems we can consider associated optimization problems, called dual, such that their solutions are related to the solutions of the original (primal) problems. In particular, for a broad class of problems the primal solutions can be easily calculated from the dual ones.

In general, solving the primal problem and solving the dual will have its own computational advantages and disadvantages, so it is our choice which one we prefer to deal with.

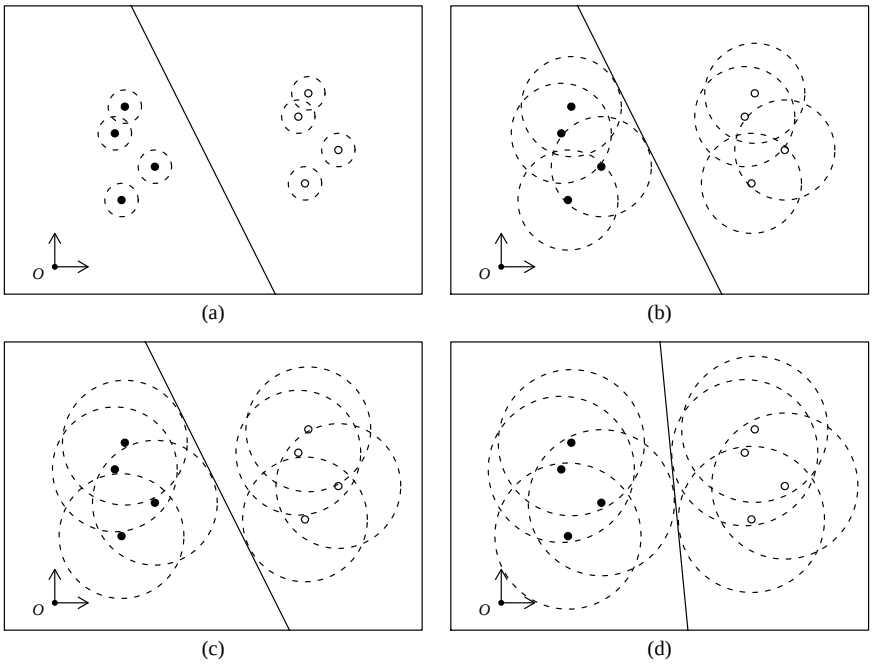


Figure 6: Finding maximum margin hyperplane: geometrical insight.

In case of maximum margin hyperplane, the dual formulation has two major benefits: its constraints are easier to handle than constraints of the primal problem, and it is better suited to deal with kernel functions (see section 4). This is why it is the dual problem which is actually solved by most SVM packages[5]. Besides computational considerations, the dual formulation allows us to establish the concept of support vectors – the training points that define orientation and intercept of the maximum margin hyperplane.

So let us recall the primary optimization problem for finding maximum margin hyperplane $\pi(\boldsymbol{w}, b)$:

$$\frac{1}{2}\sum_{k=1}^{n} w_k^2 \to \min_{\boldsymbol{w},b} \tag{2.13}$$

s.t.

$$y_i\left(\sum_{k=1}^{n} w_k x_{ik} + b\right) \geq 1, \quad i = 1, 2, ..., l. \tag{2.14}$$

Dual problem in general form

The *dual problem* for (2.13)-(2.14) in its general form is written as

$$L_d(\boldsymbol{\alpha}) \to \max_{\boldsymbol{\alpha}} \tag{2.15}$$

s.t.

$$\alpha_i \geq 0, \quad i = 1, 2, ..., l, \tag{2.16}$$

[5]For some interesting discussion on primary and dual problems for SVM, see paper by Olivier Chapelle "Training a support vector machine in the primal".

Dual function

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_l)$, $L_d(\boldsymbol{\alpha})$ is the *dual function* defined as

$$L_d(\boldsymbol{\alpha}) = \min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \boldsymbol{\alpha}), \tag{2.17}$$

Lagrangian

and $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ is the *Lagrangian* defined as

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\sum_{k=1}^{n} w_k^2 - \sum_{i=1}^{l} \alpha_i \left(y_i \left(\sum_{k=1}^{n} w_k x_{ik} + b \right) - 1 \right). \tag{2.18}$$

The dual can be written in a more explicit form. Since the Lagrangian is a convex function in our case[6], for any $\boldsymbol{\alpha}$ pair $(\boldsymbol{w}^*, b^*)$ is the global minimum of $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ if and only if

$$\nabla_{\boldsymbol{w},b} L(\boldsymbol{w}^*, b^*, \boldsymbol{\alpha}) = \mathbf{0}, \tag{2.19}$$

where $\nabla_{\boldsymbol{w},b} L$ denotes the gradient of function L (vector of its first derivatives with respect to w_i and b), and $\mathbf{0}$ denotes the null vector from $\mathbf{R}^n$. Thus, $L_d(\boldsymbol{\alpha}) = L(\boldsymbol{w}^*, b^*, \boldsymbol{\alpha})$ given that (2.19) is satisfied, and therefore the dual problem (2.15)-(2.16) can be stated as[7]

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) \to \max_{\boldsymbol{w},b,\boldsymbol{\alpha}} \tag{2.20}$$

s.t.

$$\nabla_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \mathbf{0} \tag{2.21}$$

$$\alpha_i \geq 0, \;\; i = 1, 2, ..., l. \tag{2.22}$$

The constraint (2.21) states that

$$\frac{\partial L}{\partial w_k} = 0, \; k = 1, 2, ..., n$$
$$\frac{\partial L}{\partial b} = 0,$$

which is equivalent to

$$w_k = \sum_{i=1}^{l} \alpha_i y_i x_{ik}, \; k = 1, 2, ..., n \tag{2.23}$$
$$\sum_{i=1}^{l} \alpha_i y_i = 0.$$

Finally, substituting (2.23) into (2.18), we rewrite the dual problem (2.20)-(2.22) as

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \to \max_{\boldsymbol{\alpha}} \tag{2.24}$$

s.t.

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.25}$$

$$\alpha_i \geq 0, \;\; i = 1, 2, ..., l. \tag{2.26}$$

[6] Because it is a convex function minus linear functions.

[7] This form of the dual problem is also know as the Wolfe dual.

Like the primary problem, the dual is also a quadratic programming problem, but constraints (2.25)-(2.26) are easier to handle than constraints (2.14). This is one of the reasons why many SVM packages solve the dual problem instead of the primal.

The Hessian of objective function (2.24) (matrix of its second derivatives) with respect to variables α_i has the form

$$-1 \times [y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle]_{l \times l}. \tag{2.27}$$

Matrix $[y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle]_{l \times l}$ is congruent[8] to $[\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle]_{l \times l}$, which is the Gramian matrix for vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_l$. The Gramian matrix is always positive semidefinite[9], therefore matrix (2.27) is negative semidefinite, which means that problem (2.24)-(2.26) is concave (but not strictly concave). A concave problem may have a single global maximum or many global maxima. Therefore, generally speaking, the solution of problem (2.24)-(2.26) is not unique, while the solution of the primal problem (2.13)-(2.14) is. We will give an example of problem with several optimal dual solutions in subsection 2.3 devoted to support vectors.

Connection between primary and dual problems

What is the connection between the dual problem (2.24)-(2.26) and the primal problem (2.13)-(2.14)? If the primal has solution, so does the dual, and vice versa. If the primal is unbounded (which happens if all training vectors belong to a single class), then the dual is infeasible; if the primal is infeasible (which happens if two classes are not linearly separable), then the dual is either infeasible or unbounded.

What is the connection between the dual solution $\boldsymbol{\alpha}^*$ and the primal solution $(\boldsymbol{w}^*, b^*)$? Rewriting equation (2.23) in vector form, we get:

$$\boldsymbol{w}^* = \sum_{i=1}^{l} \alpha_i^* y_i \boldsymbol{x}_i. \tag{2.28}$$

Intercept b^* can be found from any of the constraints (2.14) that holds as an equality[10]. For example, if

$$\sum_{k=1}^{n} w_k^* x_{1k} + b^* = 1$$

then

$$b^* = 1 - \sum_{k=1}^{n} w_k^* x_{1k}.$$

Let us recall that $(\boldsymbol{w}^*, b^*)$ is the unique global solution to the primal problem, while $\boldsymbol{\alpha}^*$ may be any of numerous global solutions to the dual one. It is also worth to note that in our case there is no duality gap between the primary and the dual objective functions, which means that

$$\frac{1}{2} \sum_{k=1}^{n} (w_k^*)^2 = \sum_{i=1}^{l} \alpha_i^* - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i^* \alpha_j^* y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \ .$$

[8] Matrices A and B are called congruent if there is nonsingular matrix P such that $A = P^T B P$. If A and B are congruent and A is positive semidefinite, then B is also positive semidefinite.

[9] Matrix A is called positive semidefinite if $\boldsymbol{x} A \boldsymbol{x}^T = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j \geq 0$ for any $\boldsymbol{x}$.

[10] Constraints that hold as equalities correspond to training vectors called support vectors; there always will be at least two such constraints (see subsection 2.3).

2.3. Support vectors

We saw in the previous subsection that if $(\boldsymbol{w}, b)$ is the primal solution of (2.13)-(2.14), and $\boldsymbol{\alpha}$ is the dual solution, then

$$\boldsymbol{w} = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x}_i. \tag{2.29}$$

This means that the normal vector $\boldsymbol{w}$ of the maximum margin hyperplane expands into training vectors $\boldsymbol{x}_i$ with coefficients $\alpha_i y_i$, where

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \tag{2.30}$$

Support vectors

$\alpha_i \geq 0$, and $y_i = 1$ if $\boldsymbol{x}_i \in C_1$, and $y_i = -1$ if $\boldsymbol{x}_i \in C_2$. Training vectors $\boldsymbol{x}_i$ such that $\alpha_i > 0$ are called *support vectors* of the maximum margin hyperplane.

We see that support vectors are the only vectors from the training set that determine the position of the maximum margin hyperplane. Where are these vectors located? To answer this question, note that vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_l)$ used in expansion (2.29) is a minimum of the Lagrangian (2.18) for given $\boldsymbol{w}$ and b. Therefore, α_i can be nonzero only if

$$y_i \left(\sum_{k=1}^{n} w_k x_{ik} + b \right) - 1 = 0, \tag{2.31}$$

because if

$$y_i \left(\sum_{k=1}^{n} w_k x_{ik} + b \right) - 1 > 0 \tag{2.32}$$

then α_i must be zero[11]. Equation (2.31) is equivalent to

$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b = 1, \text{ if } x_i \in C_1, \tag{2.33}$$

and to

$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b = -1, \text{ if } x_i \in C_2. \tag{2.34}$$

Dividing these equations by a positive number $||\boldsymbol{w}||$, we get:

$$\frac{\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b}{||\boldsymbol{w}||} = \frac{1}{||\boldsymbol{w}||}, \text{ if } x_i \in C_1$$
$$\frac{\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b}{||\boldsymbol{w}||} = -\frac{1}{||\boldsymbol{w}||}, \text{ if } x_i \in C_2.$$

Support vectors lie on the margin between two classes

On the left-hand side of these equation we now have the signed distance $\rho(\pi, \boldsymbol{x}_i)$ between hyperplane $\pi(\boldsymbol{w}, b)$ and vector $\boldsymbol{x}_i$. Comparing them with constraints (2.11)-(2.12) of the primary optimization problem we conclude that any support vector $\boldsymbol{x}_i$ is a vector closest to the optimal hyperplane, and the distance between the two is $d = 1/||\boldsymbol{w}||$. Another way to put it is to say that support vectors lie on the margin between two classes (Figure 7).

[11] Assume (2.32) holds and α_i is not zero. Then $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_i, ..., \alpha_l)$ cannot be a minimizer of the Lagrangian, since vector $\boldsymbol{\alpha}' = (\alpha_1, ..., 2\alpha_i, ..., \alpha_l)$ will render smaller value of the Lagrangian than vector $\boldsymbol{\alpha}$.

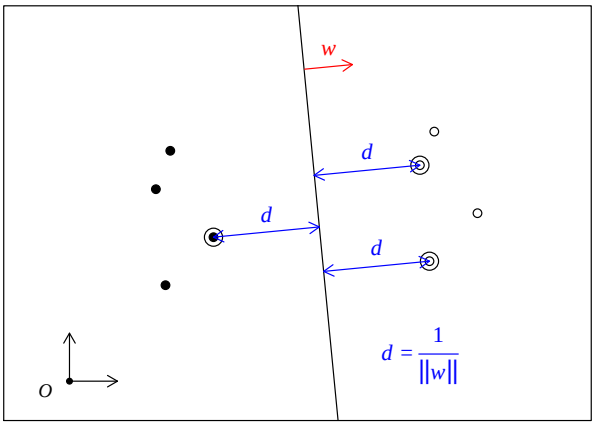


Figure 7: Support vectors (circled) are those training vectors that are closest to the separating plane. They are said to lie on the margin between two classes.

Note that in section 1 the margin was defined as certain distance, that is – as a number. However, sometimes it is convenient to use term "margin" to refer to a part of space – the gap between the hyperplane and the classes. For the maximum margin hyperplane, this gap is the space between two hyperplanes: $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = -1$ and $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 1$ (and the width of this gap is $\frac{2}{||\boldsymbol{w}||}$). Thus, we say "vector $\boldsymbol{x}$ lies on the margin" meaning that either $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = -1$ or $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 1$. We can also say "vector $\boldsymbol{x}$ lies in the margin" meaning that $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b > -1$ and $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b < 1$.

Note that among all training examples of one class, those located closer to another class may be considered more difficult to learn since it is easier to confuse them with examples from another class. Therefore, we can establish the following property of SVM classifier: of all the given training examples, the most difficult ones, located on the margin between two classes, have the strongest effect on SVM classifier, while the typical or average examples, located in the center of classes, as well as the easiest examples, located on remote boundaries of the classes, have weaker effect on the classifier. A consequence of this property is that SVM is not sensitive to outliers located far away from the margin.

Number of support vectors

How many support vectors can a classifier have? The minimum number is two – at least one in each class. This follows from the constraint (2.30), which can be rewritten as

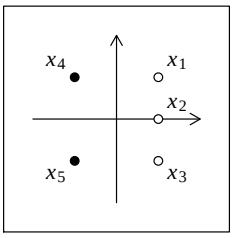
$$\sum_{i:\, y_i=1} \alpha_i = \sum_{j:\, y_j=-1} \alpha_j.$$

The maximum number is, of course, l – the size of the training set. The fraction of support vectors in the training set is an important property of the classifier related to its generalization performance and overfitting. The classifier is said to overfit the training data when it makes few errors while classifying training examples and makes a lot of errors while classifying new data. The classifier is said to have good generalization performance when it classifies new data with few errors. Thus, overfitting implies bad generalization

performance, and a high fraction of support vectors in the training set is a good indication that SVM may have overfitted the training data[12].

Expansion of $\boldsymbol{w}$ into support vectors is not unique

Finally, let us note that the expansion of $\boldsymbol{w}$ into support vectors is not unique, and therefore the same optimal hyperplane can be defined by different sets of support vectors. This follows from the fact that solution $\boldsymbol{\alpha}$ of the problem (2.24)-(2.26) is not necessarily unique, so that expansion (2.29) is not unique, either. The following simple configuration gives an example of this situation. Consider points $\boldsymbol{x}_1 = (1,1)$, $\boldsymbol{x}_2 = (1,0)$, $\boldsymbol{x}_3 = (1,-1)$ from class C_1, and points $\boldsymbol{x}_4 = (-1,1)$, $\boldsymbol{x}_5 = (-1,-1)$ from class C_2 (see the figure on the right). Parameters of the maximum margin hyperplane separating these two classes are: $\boldsymbol{w} = (1,0)$, $b = 0$. Consider three different dual solutions for this SVM: $\boldsymbol{\alpha}^1 = (0.25, 0, 0.25, 0.25, 0.25)$, $\boldsymbol{\alpha}^2 = (0.2, 0.1, 0.2, 0.25, 0.25)$, and $\boldsymbol{\alpha}^3 = (0.1, 0.3, 0.1, 0.25, 0.25)$. Clearly, $\boldsymbol{\alpha}^1$ and $\boldsymbol{\alpha}^2$ lead to different sets of support vectors, while $\boldsymbol{\alpha}^2$ and $\boldsymbol{\alpha}^3$ lead to the same sets of support vectors but different expansions of $\boldsymbol{w}$ into support vectors.



3. Maximum margin hyperplane for linearly nonseparable classes

In real-life classification problems we rarely deal with linearly separable classes. Most of the time our observations will form classes that no hyperplane can separate without errors. Here, we will call these classes overlapping[13]. For overlapping classes, problem (2.2)-(2.4) becomes infeasible (and the dual problem – unbounded), since there exist no $\boldsymbol{w}, b$ that could satisfy all constraints (2.3), (2.4) at the same time. In this section the idea of maximum margin hyperplane is generalized for the case of linearly nonseparable classes.

3.1. Primary problem

Slack variables

For overlapping classes, constraints (2.8) cannot be simultaneously satisfied for all training vectors. We can relax these constraints by introducing error terms ξ_i, also called *slack variables*, in the following way:

$$y_i \left(\sum_{k=1}^{n} w_k x_{ik} + b \right) \geq 1 - \xi_i, \quad i = 1, 2, ..., l \tag{3.1}$$

$$\xi_i \geq 0, \quad i = 1, 2, ..., l. \tag{3.2}$$

[12] Overfitting is not likely to happen when classes are linearly separable, but it may become a serious problem when two classes are heavily mixed or when SVM uses a kernel function instead of the inner product.

[13] Note that for classes C_1 and C_2 it is always assumed in this tutorial that $C_1 \cap C_2 = \varnothing$. Our definition of overlapping classes does not imply the opposite. However, it implies that $conv(C_1) \cap conv(C_2) \neq \varnothing$, where $conv(C_1)$ and $conv(C_2)$ denote convex hulls of C_1 and C_2.

If we consider minimizing objective function

$$\frac{1}{2}\sum_{k=1}^{n} w_k^2 \to \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \tag{3.3}$$

with constraints (3.1)-(3.2), we will find that it can be made arbitrarily small because any vector $\boldsymbol{w}$ can be made feasible by using freedom in the choice of slack variables ξ_i. To make this problem meaningful, we should also seek minimization of error terms, which is usually achieved by adding their sum to the objective function (3.3):

Soft margin SVM

$$\frac{1}{2}\sum_{k=1}^{n} w_k^2 + C\sum_{i=1}^{l} \xi_i \to \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \tag{3.4}$$

s.t.

$$y_i\left(\sum_{k=1}^{n} w_k x_{ik} + b\right) \geq 1 - \xi_i, \quad i = 1, 2, ..., l \tag{3.5}$$

$$\xi_i \geq 0, \quad i = 1, 2, ..., l. \tag{3.6}$$

Here C is a positive constant balancing two different goals: maximizing the margin and minimizing the number of errors on the training data. We will consider this parameter in more detail in the next subsection.

Optimization problem (3.4)-(3.6) defines so called *soft margin* SVM, as opposed to the *hard margin* SVM which we considered in section 2. For a soft margin SVM, we want to find a separating hyperplane with the maximum margin, we allow training vectors to lie inside the margin or to be missclassified, and we want the overall error measured by the sum of slack variables to be minimized. Note that when two classes are linearly separable, problem (3.4)-(3.6) will have the same solution as problem (2.2)-(2.4). Therefore, optimization problem (3.4)-(3.6) can be used as a general formulation that defines SVM on arbitrary training set, regardless of linear separability of two classes.

Margin for hyperplane and linearly nonseparable classes

If $\boldsymbol{w}^*$, b^* is a solution of (3.4)-(3.6) then hyperplane $\langle \boldsymbol{w}^*, \boldsymbol{x}\rangle + b^* = 0$ is called optimal or maximum margin hyperplane. What exactly do we mean by hyperplane's margin when we consider two overlapping classes? In section 1, margin was defined for a hyperplane and two linearly separable classes. In section 2, it was shown that for the maximum margin hyperplane its margin is

- as a number: $\frac{2}{||\boldsymbol{w}^*||}$
- as a part of space: the gap between $\langle \boldsymbol{w}^*, \boldsymbol{x}\rangle + b^* = 1$ and $\langle \boldsymbol{w}^*, \boldsymbol{x}\rangle + b^* = -1$

The same is assumed for a soft margin hyperplane: its margin is defined as the region between hyperplanes $\langle \boldsymbol{w}^*, \boldsymbol{x}\rangle + b^* = 1$ and $\langle \boldsymbol{w}^*, \boldsymbol{x}\rangle + b^* = -1$, although this region is not anymore the gap between the optimal hyperplane and two classes.

Minimization of quadratic objective function (3.4) subject to linear constraints (3.5)-(3.6) is a problem of quadratic programming. This function is convex, but in contrast to function (2.5) is not strictly convex (since its Hessian is positive semidefinite). This implies that solution $\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*$ of problem (3.4)-(3.6) is not unique. Problem (3.4)-(3.6) will always have solution, unless the training set contains only one class, in which case the problem will be unbounded.

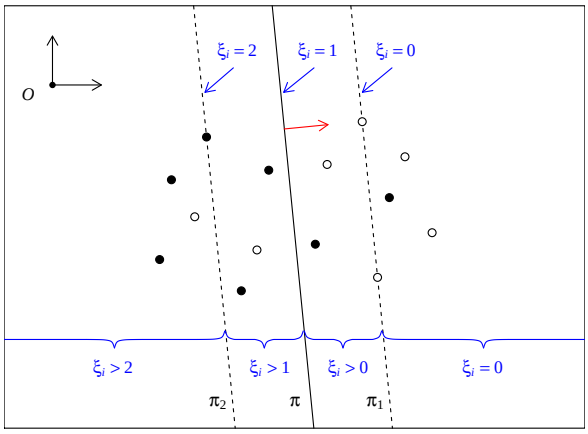


Figure 8: Values of slack variables ξ_i shown for training points from C_1 (white points). Optimal hyperplane π is defined by $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0$, π_1 is defined by $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 1$, and π_2 is defined by $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = -1$. The margin is the region between π_1 and π_2. Margin expansion may increase the overall error. Note that points located in the margin will always have $\xi_i > 0$, even when they are correctly classified by the hyperplane.

3.2. Parameter C

As we noted before, soft margin SVM has one parameter which should be adjusted by the user: a positive constant C in objective function (3.4). This parameter balances two different goals: maximizing the margin and minimizing the number of errors on the training data. These goals may be conflicting, since margin expansion may increase the overall error (Figure 8). By changing parameter C we can choose to favor one goal over another. This is illustrated on Figure 9, which shows four separating hyperplanes and their margins, obtained for the same training set using increasing values of parameter C. Circled are points with non-zero error terms ξ_i – they either lie on the wrong side of the hyperplane, or in the margin. When C is very small, the sum of error terms becomes negligible in objective function (3.4), so that the goal of optimization is to maximize the margin. As a result, the margin can be large enough to contain all the points. At another extreme, when C is very large, the sum of error terms dominates the margin term in objective function (3.4), so that the goal of optimization is to minimize the sum of error terms. As a result, the margin can be so small that it does not contain any points. Note that despite differences in the value of parameter C and the size of the margin, all four hyperplanes shown in Figure 9 are fairly similar, and they all correctly classify the same training points.

Clearly, values of parameter C do not have absolute meaning. They are related to the number of training points and the range of data. In the example shown on Figure 9, we have a training set consisting of 14 points whose abscissa (horizontal coordinate) varies between 20 and 80, and the ordinate (vertical coordinate) – between 15 and 55. Note that it is possible to make C independent of the number of training points if $C\frac{1}{l}\sum_{i=1}^{l}\xi_i$ term is used instead of $C\sum_{i=1}^{l}\xi_i$ in objective function (3.4). Note also that Figure 9 suggests that the tuning of parameter C should be done on a logarithmic scale. This is indeed a good approach for many practical applications.

Tuning of parameter C

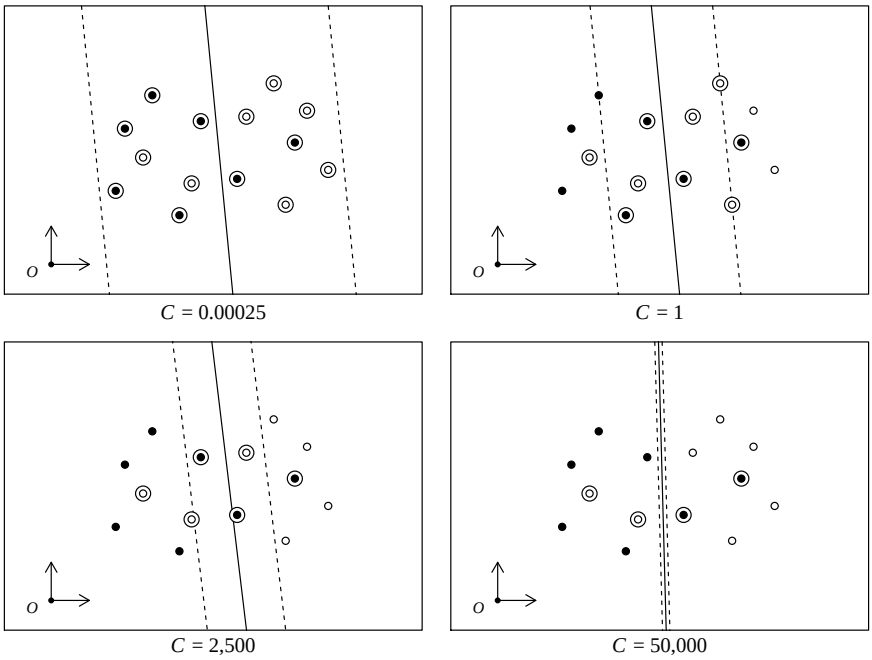


Figure 9: Optimal separating hyperplanes and their margins obtained for the same training set by using different values of parameter C. Circled are points with $\xi_i > 0$.

Classes with different cost of misclassification

In addition to balancing the goals of margin maximization and error minimization, parameter C also provides a convenient way to deal with situations when the cost of misclassification is different for points from C_1 and C_2. For example, if the cost of misclassification a point from C_1 is two times higher then the cost of misclassification a point from C_2, then instead of objective function (3.4) we can consider objective function

$$\frac{1}{2}\sum_{k=1}^{n} w_k^2 + \tilde{C}_1 \sum_{i:\ y_i=1} \xi_i + \tilde{C}_2 \sum_{i:\ y_i=-1} \xi_i \to \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \tag{3.7}$$

Classes unbalanced in size

where $\tilde{C}_1/\tilde{C}_2 = 2$. This trick also allows to handle situations with unbalanced classes, when the number of training points from one class significantly exceeds the number of training points from another class. Given unbalanced training data, SVM classifier will tend to have higher accuracy on larger class, and lower accuracy on smaller class. To level these accuracies, objective function (3.7) can be used, where $\tilde{C}_1 > \tilde{C}_2$ if C_1 is smaller than C_2.

3.3. Dual problem

The derivation of the dual for problem (3.4)-(3.6) exactly follows the derivation of the dual for problem (2.13)-(2.14), considered in subsection 2.2. We start from the general definitions of the dual problem (2.15)-(2.16) and dual function (2.17), which in our case will include two extra sets of variables: $\boldsymbol{\xi} = (\xi_1, \xi_2, ..., \xi_l)$, and $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_l)$ (together

with constraints $\beta_i \geq 0$), because for problem (3.4)-(3.6) the Lagrangian is defined as

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\sum_{k=1}^{n} w_k^2 + C\sum_{i=1}^{l}\xi_i \\ - \sum_{i=1}^{l}\alpha_i\left(y_i\left(\sum_{k=1}^{n} w_k x_{ik} + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{l}\beta_i\xi_i\ . \tag{3.8}$$

Since the Lagrangian is a convex function in our case[14], for any $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ point $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*)$ is the global minimum of $L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ if and only if

$$\nabla_{\boldsymbol{w},b,\boldsymbol{\xi}} L(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathbf{0}. \tag{3.9}$$

Thus, $L_d(\boldsymbol{\alpha}, \boldsymbol{\beta}) = L(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\beta})$ given that (3.9) is satisfied, and therefore the dual problem can be stated as

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \rightarrow \max_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}}$$

s.t.

$$\nabla_{\boldsymbol{w},b,\boldsymbol{\xi}} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \mathbf{0} \tag{3.10}$$
$$\alpha_i \geq 0,\ \ i = 1, 2, ..., l$$
$$\beta_i \geq 0,\ \ i = 1, 2, ..., l.$$

The constraint (3.10) implies that

$$\frac{\partial L}{\partial w_k} = 0,\ k = 1, 2, ..., n$$
$$\frac{\partial L}{\partial b} = 0$$
$$\frac{\partial L}{\partial \xi_i} = 0,\ i = 1, 2, ..., l,$$

which is equivalent to

$$w_k = \sum_{i=1}^{l}\alpha_i y_i x_{ik},\ k = 1, 2, ..., n \tag{3.11}$$
$$\sum_{i=1}^{l}\alpha_i y_i = 0$$
$$\alpha_i + \beta_i = C,\ i = 1, 2, ..., l. \tag{3.12}$$

From (3.12) we get $\beta_i = C - \alpha_i$, and after plugging this into the Lagrangian (3.8) slack variables ξ_i cancel out. Thus, we arrive to the Lagrangian of the form (2.18) defined previously for a separable case. Condition (3.12) also has another implication: since $\beta_i \geq 0$, we must have $\alpha_i \leq C$.

[14] Because it is a convex function plus/minus linear functions.

Substituting (3.11) into (2.18), we get the final form of the dual problem for the soft-margin SVM:

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \rightarrow \max_{\boldsymbol{\alpha}} \tag{3.13}$$

s.t.

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{3.14}$$

$$0 \leq \alpha_i \leq C, \;\; i = 1, 2, ..., l. \tag{3.15}$$

Note that this dual is almost identical to the dual (2.24)-(2.26) derived for the case of linearly separable classes; the only difference is that here dual variables α_i have additional constraints – they are upper bounded by C. Thus, everything that we said in subsection 2.2 about the connection between the dual and the primal problems, as well as between the dual and primal solutions, is also true for the case of soft margin SVM[15].

Note also that the fact that the dual variables α_i are upper bounded by C implies that in soft margin SVM the influence of each training point on the separating hyperplane is limited, which makes it less sensitive to the outliers located close to the margin.

3.4. Support vectors

Just like in a separable case, equation (3.11) implies that if $(\boldsymbol{w}, b)$ is the primal solution of (3.4)-(3.6), and $\boldsymbol{\alpha}$ is the dual solution, then

$$\boldsymbol{w} = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x}_i. \tag{3.16}$$

Training vectors $\boldsymbol{x}_i$ for which $\alpha_i > 0$ are called *support vectors* of the hyperplane. Where are these vectors located? Using reasoning similar to what we used in subsection 2.3, it is easy to show that for the training vectors from the first class coefficients α_i are equal to zero when $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b > 1$; range between 0 and C when $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 1$; and are equal to C when $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b < 1$ (Figure 10). Thus, in contrast to a hard margin SVM, whose support vectors are located on the margin between two classes (Figure 7), support vectors of a soft margin SVM may be located either on the margin between two classes, in the margin, or outside the margin and on the wrong side of the hyperplane. Also, the number of support vectors for a soft margin classifier is affected by parameter C: when we decrease C, the margin expands and the number of support vectors may grow.

[15] Except that the primal problem for the soft margin SVM can never be infeasible.

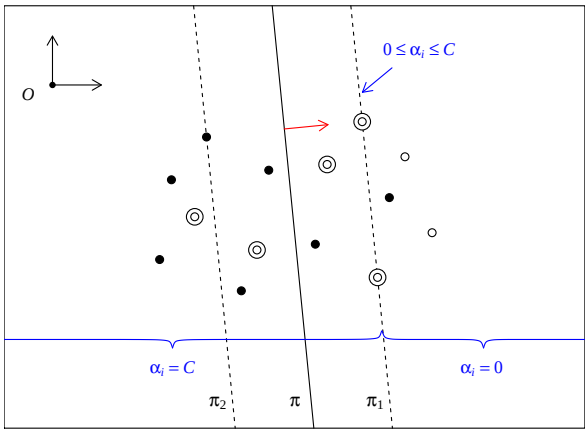


Figure 10: Regions of values for coefficients α_i, for training points from C_1 (white points). Optimal hyperplane π is defined by $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0$, π_1 is defined by $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 1$, and π_2 is defined by $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = -1$. Support vectors are those points that have $\alpha_i > 0$. Support vectors from C_1 are circled.

4. SVM with kernels

In the context of SVM, a kernel $k(\boldsymbol{x}, \boldsymbol{z})$ is a special function which is used in dual problem (3.13)-(3.15) and in decision function (2.1) instead of the inner product $\langle \boldsymbol{x}, \boldsymbol{z} \rangle$. The use of kernels allows us to define SVMs which instead of hyperplanes utilize a much wider class of separating surfaces. When we use a kernel k, we implicitly define (not necessarily in a unique way) some new space H (called feature space) and a mapping Φ (called feature mapping) which transforms our original space $\mathbf{R}^n$ to H. In this new space H we seek the usual maximum margin hyperplane separating our transformed data by solving problem (3.4)-(3.6). For the majority of kernels mapping Φ is non-linear, so that the maximum margin hyperplane in H corresponds to a non-linear (and sometimes quite complex) separating surface in $\mathbf{R}^n$. By using non-linear surfaces we can expect to get better separation of given classes than by using a hyperplane. Put differently, given classes may be better separated with a hyperplane when they are non-linearly transformed into another space, especially when this new space has more dimensions than the original one.

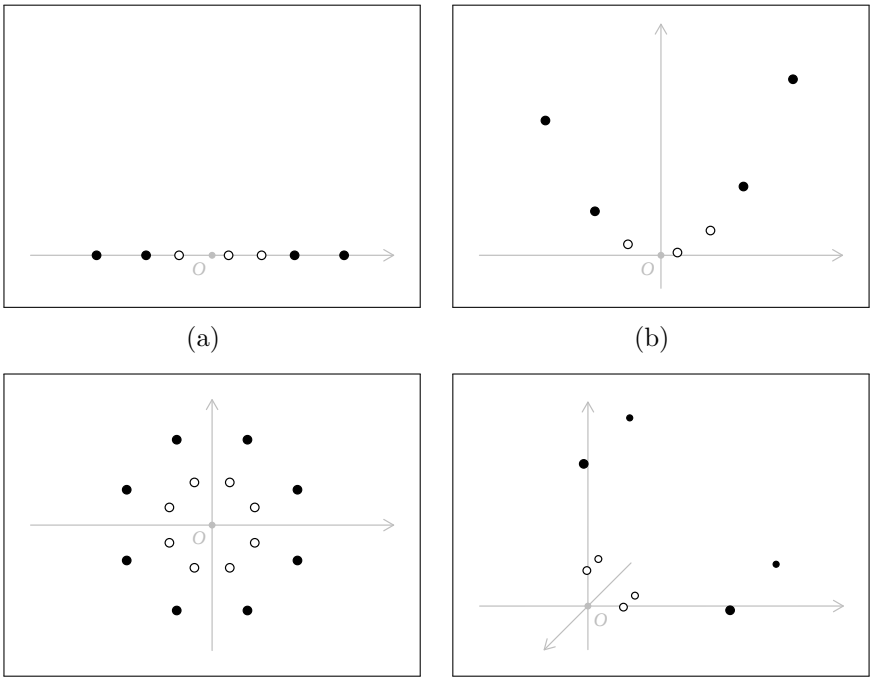
4.1. Kernel trick

Consider white and black points shown in Figures 11a and 11c. Clearly, there is no way to achieve good separation of these points using a hyperplane (that is, using a point in case of one-dimensional space, Figure 11a, and a line in case of two-dimensional space, Figure 11c). Let us define the following mappings:

$$\Phi_1 : \ \boldsymbol{x} = (x_1) \ \rightarrow \ \Phi_1(\boldsymbol{x}) = (x_1, x_1^2) \tag{4.1}$$

$$\Phi_2 : \ \boldsymbol{x} = (x_1, x_2) \ \rightarrow \ \Phi_2(\boldsymbol{x}) = (x_1^2, x_2^2, \sqrt{2}\, x_1 x_2). \tag{4.2}$$

These mappings transform linearly nonseparable classes shown in Figures 11a, 11c into linearly separable classes shown in Figures 11b, 11d, respectively. Note that these mappings are non-linear, and they increase the dimensionality of data.



(a) (b)

(c) (d)

Figure 11: (a), (c) Examples of linearly nonseparable classes (white and black points)) in one- and two-dimensional spaces. (b), (d) Mappings Φ_1 and Φ_2 transform classes into spaces of higher dimensionality where they become linearly separable.

Assume we have a mapping Φ from our original space $\mathbf{R}^n$ to a new space H, and we want to find the optimal separating hyperplane in the space H. To do this, we should transform our training set $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_l$ from $\mathbf{R}^n$ into $\Phi(\boldsymbol{x}_1), \Phi(\boldsymbol{x}_2), ..., \Phi(\boldsymbol{x}_l)$ from H, and solve the following problem:

$$\frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{l}\xi_i \to \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \tag{4.3}$$

s.t.

$$y_i\Big(\langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle + b\Big) \geq 1 - \xi_i, \quad i = 1, 2, ..., l \tag{4.4}$$

$$\xi_i \geq 0, \quad i = 1, 2, ..., l, \tag{4.5}$$

where $\boldsymbol{w} \in H$, and the norm and the inner product used in this problem are defined in H. In order to classify a new point $\boldsymbol{x}$ from $\mathbf{R}^n$, we need to find its image $\Phi(\boldsymbol{x})$ in the space H, and then use the decision function

$$f(\boldsymbol{x}) = \operatorname{sgn}\{\langle \boldsymbol{w}, \Phi(\boldsymbol{x})\rangle + b\}. \tag{4.6}$$

Kernel trick

It turns out that there is another, more convenient approach to define an SVM that uses given data transformation Φ. It is called the *kernel trick*. Let us write the dual problem for (4.3)-(4.5):

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle \to \max_{\boldsymbol{\alpha}} \tag{4.7}$$

s.t.

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{4.8}$$

$$0 \le \alpha_i \le C, \;\; i = 1, 2, ..., l. \tag{4.9}$$

Using (3.16), decision function (4.6) can be rewritten as

$$f(\boldsymbol{x}) = \text{sgn}\{\sum_{i=1}^{l} \alpha_i y_i \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}) \rangle + b\}. \tag{4.10}$$

It is easy to note now that the knowledge of $\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{z}) \rangle$ for every $\boldsymbol{x}$ and $\boldsymbol{z}$ from $\mathbf{R}^n$ is enough to solve problem (4.7)-(4.9) and to use decision function (4.10)[16]. Therefore, if we denote $\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{z}) \rangle$ by $k(\boldsymbol{x}, \boldsymbol{z})$, we can seek separating surface in the original space $\mathbf{R}^n$ simply by using $k(\boldsymbol{x}, \boldsymbol{z})$ instead of $\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{z}) \rangle$, and avoid the need to explicitly map our data from $\mathbf{R}^n$ to H. Consequently, the dual (4.7)-(4.9) can be rewritten as

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \, k(\boldsymbol{x}_i, \boldsymbol{x}_j) \to \max_{\boldsymbol{\alpha}} \tag{4.11}$$

s.t.

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{4.12}$$

$$0 \le \alpha_i \le C, \;\; i = 1, 2, ..., l, \tag{4.13}$$

while decision function (4.10) becomes

$$f(\boldsymbol{x}) = \text{sgn}\{\sum_{i=1}^{l} \alpha_i y_i \, k(\boldsymbol{x}_i, \boldsymbol{x}) + b\}. \tag{4.14}$$

The function $k(\boldsymbol{x}, \boldsymbol{z})$ used instead of $\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{z}) \rangle$ is called a kernel defined by mapping Φ.

Going back to our example, consider mapping Φ_2 defined by equation (4.2). Note that

$$\begin{aligned} \langle \Phi_2(\boldsymbol{x}), \Phi_2(\boldsymbol{z}) \rangle &= \langle (x_1^2, x_2^2, \sqrt{2}\, x_1 x_2), (z_1^2, z_2^2, \sqrt{2}\, z_1 z_2) \rangle \\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2 x_1 x_2 z_1 z_2 \\ &= (x_1 z_1 + x_2 z_2)^2 \\ &= \langle \boldsymbol{x}, \boldsymbol{z} \rangle^2. \end{aligned} \tag{4.15}$$

[16] In fact, to solve problem (4.7)-(4.9) we only need to know $\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{z}) \rangle$ for every pair of training points.

Therefore, $k(\boldsymbol{x}, \boldsymbol{z}) = \langle \boldsymbol{x}, \boldsymbol{z} \rangle^2$ is the kernel function defined by mapping Φ_2, and if we want to build an SVM in the new space $\mathbf{R}^3$ obtained from $\mathbf{R}^2$ with the help of mapping (4.2), we can still use training points and inner product from $\mathbf{R}^2$ – we should simply substitute $\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{z}) \rangle$ in (4.7) and (4.10) by $\langle \boldsymbol{x}, \boldsymbol{z} \rangle^2$.

The natural extension of this approach will be the following. If we have some "suitable" function $k(\boldsymbol{x}, \boldsymbol{z})$ to use instead of $\langle \boldsymbol{x}, \boldsymbol{z} \rangle$, we can define an SVM without even knowing mapping $\Phi(\boldsymbol{x})$. And "suitable" would be any function that could be represented through the inner product in some space H. Now we are ready to give a formal definition of kernels.

4.2. Kernels

Let V be the n-dimensional real coordinate space $\mathbf{R}^n$ or any other vector space. A *kernel* k is a mapping from $V \times V$ to $\mathbf{R}$ such that there exist space H supplied with the inner product $\langle ., . \rangle$, and a mapping Φ from V to H, so that for arbitrary $\boldsymbol{x}$, $\boldsymbol{z}$ from V

$$k(\boldsymbol{x}, \boldsymbol{z}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{z}) \rangle. \tag{4.16}$$

Space H is often called *feature space*, and mapping Φ is called *feature mapping*. Two properties of kernels immediately follow from the definition.

1. Since the inner product is a symmetric function, so must be a kernel: $k(\boldsymbol{x}, \boldsymbol{z}) = k(\boldsymbol{z}, \boldsymbol{x})$ for any $\boldsymbol{x}$ and $\boldsymbol{z}$.

2. Consider a set of m arbitrary vectors $\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_m$ and form a square $m \times m$ matrix $K(\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_m) = (k(\boldsymbol{z}_i, \boldsymbol{z}_j))$. Since k is a kernel, there exist H and Φ such that $k(\boldsymbol{z}_i, \boldsymbol{z}_j) = \langle \Phi(\boldsymbol{z}_i), \Phi(\boldsymbol{z}_j) \rangle$. Thus, matrix $K(\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_m)$ is the Gramian matrix of the set of vectors $\Phi(\boldsymbol{z}_1), \Phi(\boldsymbol{z}_2), ..., \Phi(\boldsymbol{z}_m)$ and therefore it must be positive semidefinite. Function $k(\boldsymbol{x}, \boldsymbol{z})$ that defines a positive semidefinite matrix $K(\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_m)$ for any $\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_m$ is called *positive semidefinite*[17].

Thus, any kernel is a symmetric and positive semidefinite function. Is the converse true? Namely, is it true that any symmetric and positive semidefinite function is a kernel? It is proved in Mercer's theorem that this indeed is true: for any such function we can always find an infinite-dimensional space H and mapping Φ such that (4.16) holds. Thus, the symmetry and positive semidefiniteness of a function are sometimes called Mercer's conditions (for a function to be a kernel).

Mercer's conditions

For SVMs, kernels can be regarded as a generalization of the inner product. As we showed in the previous subsection, we do not have to know H and Φ to use k. Also, for a given kernel k neither the mapping Φ nor the space H are uniquely defined (see an example with the polynomial kernel below).

What happens if a function that does not satisfy Mercer's conditions is used as a kernel? Specifically, assume that the more restrictive of the two conditions is violated – that of a function being positive semidefinite. This means that for some set of vectors $\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_m$ matrix $K(\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_m)$ is not positive semidefinite, i.e., it is either indefinite or negative semidefinite. If $K(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_l)$ is yet positive semidefinite for our training set $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_l$ then optimization problem (4.11)-(4.13) remains concave and has either a

[17] Sometimes instead of positive *semidefinite* such function is called positive *definite*.

single global maximum or many global maxima. In this case we will be able to solve it and find the optimal separating hyperplane, although geometrical properties of this hyperplane (maximum margin subject to given penalty C for errors) may not be valid. If, however, $K(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_l)$ is indefinite or negative semidefinite, objective function (4.11) may become unbounded in the feasible region defined by constraints (4.12), (4.13), which means that we will not be able to solve the problem (4.11)-(4.13).

Constructing new kernels

It is easy to prove the following statements which may be helpful for constructing kernels. Suppose $k_1(\boldsymbol{x}, \boldsymbol{z})$ and $k_2(\boldsymbol{x}, \boldsymbol{z})$ are two kernels defined on $V \times V$. Then the following functions are also kernels:

- $\alpha k_1(\boldsymbol{x}, \boldsymbol{z}) + \beta k_2(\boldsymbol{x}, \boldsymbol{z}),\ \alpha \geq 0,\ \beta \geq 0$
- $k_1(\boldsymbol{x}, \boldsymbol{z})\, k_2(\boldsymbol{x}, \boldsymbol{z})$

If $f(\boldsymbol{x})$ is a real-valued function defined on V then the following functions are kernels:

- $k(\boldsymbol{x}, \boldsymbol{z}) = f(\boldsymbol{x}) f(\boldsymbol{z})$
- $k_1(f(\boldsymbol{x}), f(\boldsymbol{z}))$

In the next two subsections we consider in detail two examples of kernel functions: polynomial and Gaussian radial basis functions.

4.3. Polynomial kernel

Polynomial kernel is defined as

$$k(\boldsymbol{x}, \boldsymbol{z}) = \langle \boldsymbol{x}, \boldsymbol{z} \rangle^d, \tag{4.17}$$

where $d \geq 2$ is an integer number. We already know kernel $\langle \boldsymbol{x}, \boldsymbol{z} \rangle^2$ from subsection 4.1, where it was defined for $\boldsymbol{x}, \boldsymbol{z} \in \mathbf{R}^2$. We showed that this kernel can be represented through the inner product in the feature space $\mathbf{R}^3$ given the feature mapping

$$\Phi_2 :\ (x_1, x_2)\ \rightarrow (x_1^2, x_2^2, \sqrt{2}\, x_1 x_2).$$

As was noted above, for a given kernel the corresponding feature space H and feature mapping Φ are not uniquely defined. For example, for kernel $\langle \boldsymbol{x}, \boldsymbol{z} \rangle^2$ we can consider an alternative feature space $\mathbf{R}^4$ with the feature mapping

$$\Phi_3 :\ (x_1, x_2)\ \rightarrow (x_1^2, x_2^2, x_1 x_2, x_2 x_1).$$

For polynomial kernel of degree d defined for $\boldsymbol{x}, \boldsymbol{z} \in \mathbf{R}^n$, it is convenient to consider feature space H with coordinates corresponding to all ordered monomials of variables $x_1, x_2, ..., x_n$ of degree d, since

$$\begin{aligned} \langle \boldsymbol{x}, \boldsymbol{z} \rangle^d &= \left(\sum_{i=1}^{n} x_i z_i \right)^d \\ &= \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} ... \sum_{j_d=1}^{n} x_{j_1} x_{j_2} ... x_{j_d} z_{j_1} z_{j_2} ... z_{j_d}. \end{aligned} \tag{4.18}$$

That is, mapping Φ is defined as a transformation of vector $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ into a vector containing all possible ordered terms of the form $x_{j_1} \times x_{j_2} \times ... \times x_{j_d}$, where $j_1, j_2, ..., j_d$ are independent indices running from 1 to n. The number of such monomials is equal to the number of ways to choose d elements from n if repetitions are allowed, so the dimensionality of space H is

$$\dim H = \left(\begin{array}{c} n+d-1 \\ d \end{array} \right).$$

Polynomial kernel may be defined in a more general form as

$$k(\boldsymbol{x}, \boldsymbol{z}) = (\langle \boldsymbol{x}, \boldsymbol{z} \rangle + p)^d, \tag{4.19}$$

where $p \geq 0$. Using the binomial formula to expand (4.19) and transformations similar to (4.18), it is possible to show that for this kernel one of the possible mappings Φ maps $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ into a vector of all possible monomials of degree no larger than d.

Finally, the most general form of the polynomial kernel is

$$k(\boldsymbol{x}, \boldsymbol{z}) = (q\langle \boldsymbol{x}, \boldsymbol{x}_i \rangle + p)^d, \tag{4.20}$$

where $q > 0$, $p \geq 0$. Note that $(q\langle \boldsymbol{x}, \boldsymbol{z} \rangle + p)^d = \sqrt[d]{q}(\langle \boldsymbol{x}, \boldsymbol{z} \rangle + p/q)^d$, which is essentially kernel (4.19) multiplied by constant $\sqrt[d]{q}$.

Heatmaps on Figure 12 show separation of two classes of points (first class – white points, second class – black points) from $\mathbf{R}^2$ produced by SVMs with various polynomial

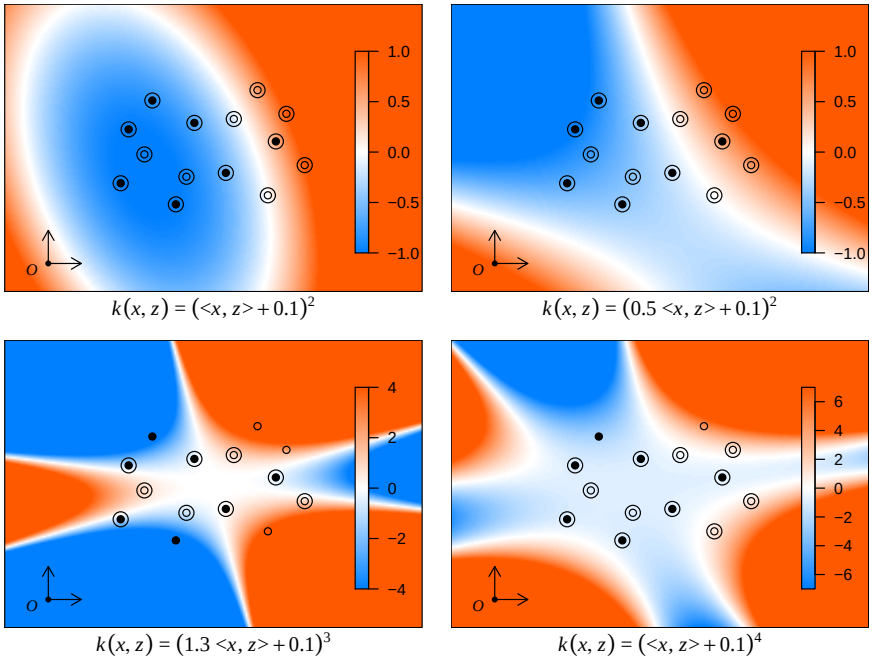


Figure 12: Heatmaps showing separation of two classes of points by using SVM with polynomial kernel $k(\boldsymbol{x}, \boldsymbol{z})$. $C = 1$ in all four examples; b varies between 0.01 and 0.88. Support vectors are circled.

kernels. On these heatmaps each point $\boldsymbol{x}$ is colored according to the corresponding value of function

$$\sum_{i=1}^{l} \alpha_i y_i \left(q\langle \boldsymbol{x}, \boldsymbol{x}_i \rangle + p\right)^d + b, \tag{4.21}$$

which is the core part of decision function (4.14). Positive values of function (4.21) are colored by orange colors and correspond to the classification decision "assign to the first class"; negative values of function (4.21) are colored by blue colors and correspond to the classification decision "assign to the second class". Separating surface is depicted by white color corresponding to the decision boundaries where (4.21) is equal to zero[18]. Notice how small changes in the parameters of the polynomial kernel substantially change the separating surface.

4.4. Gaussian radial basis function kernel

Gaussian radial basis function (GRBF) kernel, or simply Gaussian kernel, is defined by the following equation:

$$k(\boldsymbol{x}, \boldsymbol{z}) = e^{-\frac{||\boldsymbol{x}-\boldsymbol{z}||^2}{2\sigma^2}}, \tag{4.22}$$

where $\sigma > 0$. Sometimes it is also written as

$$k(\boldsymbol{x}, \boldsymbol{z}) = e^{-\gamma||\boldsymbol{x}-\boldsymbol{z}||^2},$$

where $\gamma = \frac{1}{2\sigma^2}$. Note the similarity between GRBF and Gaussian probability density function. When x and z are real numbers and z is fixed, function $k(x, z)$ has a bell-shaped graph centered at point z, whose width is directly proportional to σ (and inversely proportional to γ) (Figure 13).

Any feature space H corresponding to Gaussian kernel is infinite-dimensional. This can be proved by showing that for any set of distinct vectors $\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_m \in V$ matrix $(k(\boldsymbol{z}_i, \boldsymbol{z}_j))_{m\times m}$ is not singular, which means that vectors $\Phi(\boldsymbol{z}_1), \Phi(\boldsymbol{z}_2), ..., \Phi(\boldsymbol{z}_m)$ are linearly independent. For Gaussian kernel, in contrast to the polynomial kernel, interpretation of the corresponding feature space H is more complicated. We will only notice that H is an infinite-dimensional Hilbert space which can be exemplified by a space of infinite numerical sequences or by a space of functions with certain properties.

What does the separating surface of an SVM with Gaussian kernel look like? Consider the decision function

$$f(\boldsymbol{x}) = \operatorname{sgn}\{\sum_{i=1}^{l} \alpha_i y_i e^{-\frac{1}{2\sigma^2}||\boldsymbol{x}-\boldsymbol{x}_i||^2} + b\}, \tag{4.23}$$

where non-zero α_i correspond to support vectors $\boldsymbol{x}_i$. For each support vector, we have a "local" bell-shaped surface centered at this vector. For vectors from positive class ($y_i > 0$) this surface is "positive" ($\alpha_i y_i e^{-\frac{1}{2\sigma^2}||\boldsymbol{x}-\boldsymbol{x}_i||^2} > 0$), while for vectors from negative class

[18] Note that the color legend on each heatmap on Figure 12 shows artificially compressed range of values of the corresponding function (4.21). For example, for kernel $k(\boldsymbol{x}, \boldsymbol{z}) = (\langle \boldsymbol{x}, \boldsymbol{z} \rangle + 0.1)^2$ function (4.21) actually takes values higher than 1 and lower than -1 on the domain shown on Figure 12, but on the heatmap those values are depicted by the same color as 1 or -1.

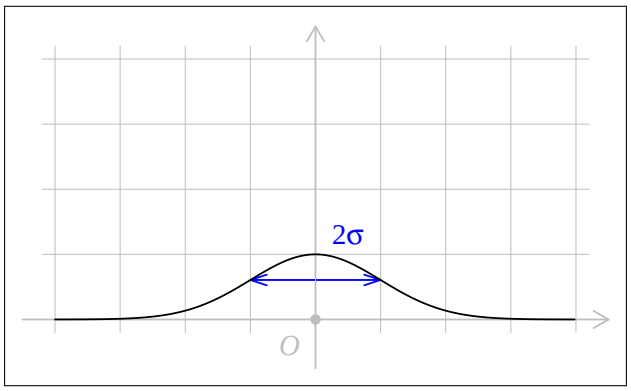


Figure 13: Graph of function $e^{-\frac{x^2}{2}}$ ($z = 0$, $\sigma = 1$). The length of the blue line segment with arrows is equal to $2\sigma = 2$ (this number is often called the width of the bell shape); the segment connects two inflection points of the function.

($y_i < 0$) this surface is "negative" ($\alpha_i y_i e^{-\frac{1}{2\sigma^2}||\boldsymbol{x}-\boldsymbol{x}_i||^2} < 0$). The resulting surface is the superposition (sum) of these local "positive" and "negative" bell-shaped surfaces.

Heatmaps on Figure 14 show separation of two classes of points (first class – white points, second class – black points) from $\mathbf{R}^2$ produced by SVMs with various Gaussian kernels. On these heatmaps each point $\boldsymbol{x}$ is colored according to the corresponding value

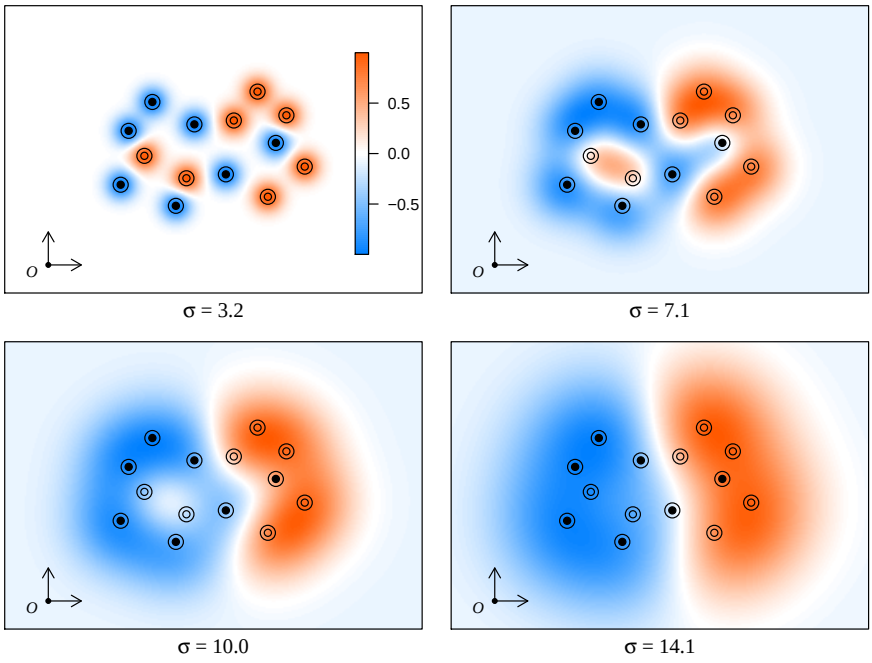


Figure 14: Heatmaps showing separation of two classes of points by using SVM with Gaussian kernel with different values of σ. $C = 1$ in all four examples; b varies between 0 and 0.08. Support vectors are circled.

of function

$$\sum_{i=1}^{l} \alpha_i y_i e^{-\frac{1}{2\sigma^2}||\boldsymbol{x}-\boldsymbol{x}_i||^2} + b, \tag{4.24}$$

which is the core part of decision function (4.23). Positive values of function (4.24) are colored by orange colors and correspond to the classification decision "assign to the first class"; negative values of function (4.24) are colored by blue colors and correspond to the classification decision "assign to the second class". Separating surface is depicted by white color corresponding to the decision boundaries where (4.24) is equal to zero.

Note that on Figure 14a the width of each "local" bell-shaped surface is smaller than the minimum distance between the points, so that these surfaces do not affect each other. This results in the separating surface that perfectly divides two classes of points but, being too detailed, overfits the data. On Figures 14b-14d, the width of each "local" surface is increasing, and they begin to affect each other. This results in the separating surface that defines more generalized shapes of the classes. This example shows that the practical range of variation for 2σ should be between the minimum and maximum distance between points in given training data.

Bibliography

For those who would like to study SVM further and those interested in statistical properties of SVM or efficient implementation of SVM for large training sets, we recommend the following references.

[1] C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 1998.

[2] V. Vapnik. *Statistical Learning Theory*, John Wiley & Sons, 1998.

[3] B. Schölkopf, C.Burges, A. Smola. *Advances in Kernel Methods: Support Vector Learning*, MIT Press, 1999.

[4] B. Schölkopf, A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002.

[5] C.-C. Chang, C.-J. Lin. LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology* 2 (3), 2011.

Appendix

Here we provide some definitions and results related to vector spaces and optimization that are used in the tutorial. Detailed treatment of these topics can be found, for example, in the following books:

[1] D. G. Luenberger. *Optimization by Vector Space Methods.* John Wiley & Sons, 1997.

[2] D. G. Luenberger. *Linear and Nonlinear Programming.* Springer, 2nd edition, 2003.

[3] I. Griva, S. G. Nash, and A. Sofer. *Linear and Nonlinear Optimization.* SIAM, 2nd edition, 2009.

A1. Real coordinate space, inner product, norm, distance

The *n-dimensional real coordinate space* $\mathbf{R}^n$ is a set of all vectors $\boldsymbol{x} = (x_1, x_2, ..., x_n)$, where x_i is a real number. Two operations are defined in $\mathbf{R}^n$: componentwise addition of two vectors, and componentwise multiplication of a vector by a real number. It is easy to prove that these operations possess a set of natural properties which make them similar to the addition and multiplication of real numbers in $\mathbf{R}$ (and make $\mathbf{R}^n$ an instance of a vector space). In particular, in $\mathbf{R}^n$ there is a null vector $(0, 0, ..., 0)$ which, when added to any vector $\boldsymbol{x}$, will not change it. To make explicit distinction between the real number zero from $\mathbf{R}$ and the null vector from $\mathbf{R}^n$, we will denote the latter by bold zero symbol $\mathbf{0}$: $\mathbf{0} = (0, 0, ..., 0)$. Terms "vector" and "point" are used interchangeably to call elements of $\mathbf{R}^n$.

Inner product $\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle$

For arbitrary vector space V, the *inner product* (also called the *dot product* or *scalar product*) is defined as a mapping from $V \times V$ to $\mathbf{R}$ that satisfies the following axioms for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in V$ and $\alpha \in \mathbf{R}$:

1. $\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle = \langle \boldsymbol{x}_2, \boldsymbol{x}_1 \rangle$ — *symmetry*
2. $\langle \alpha \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle = \alpha \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle$ — *linearity*
3. $\langle \boldsymbol{x}_1 + \boldsymbol{x}_3, \boldsymbol{x}_2 \rangle = \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle + \langle \boldsymbol{x}_3, \boldsymbol{x}_2 \rangle$ — *linearity*
4. $\langle \boldsymbol{x}, \boldsymbol{x} \rangle \geq 0$; $\langle \boldsymbol{x}, \boldsymbol{x} \rangle = 0$ if and only if $\boldsymbol{x} = \mathbf{0}$.

This definition generalizes the concept of dot product defined in geometry as the length of vector $\boldsymbol{x}_1$ multiplied by the length of vector $\boldsymbol{x}_2$ multiplied by the cosine of the angle between the two vectors:

$$\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle = |\boldsymbol{x}_1||\boldsymbol{x}_2| \cos(\boldsymbol{x}_1, \boldsymbol{x}_2),$$

which is nothing but the length of projection of vector $\boldsymbol{x}_1$ onto vector $\boldsymbol{x}_2$, multiplied by the length of vector $\boldsymbol{x}_2$ (or, equivalently, the length of projection of vector $\boldsymbol{x}_2$ onto vector $\boldsymbol{x}_1$, multiplied by the length of vector $\boldsymbol{x}_1$).

Norm $||\boldsymbol{x}||$

In an abstract vector space where an inner product has been defined following above axioms, one can introduce a measure of vector length called the *norm*:

$$||\boldsymbol{x}|| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}.$$

Using the norm, the distance between two vectors can be introduced as

$$\rho(\boldsymbol{x}_1, \boldsymbol{x}_2) = ||\boldsymbol{x}_1 - \boldsymbol{x}_2||.$$

The distance between vector $\boldsymbol{x}_1$ and set of vectors S is defined as

$$\rho(\boldsymbol{x}_1, S) = \min_{\boldsymbol{x}_2 \in S} \rho(\boldsymbol{x}_1, \boldsymbol{x}_2).$$

In $\mathbf{R}^n$, all these concepts can be defined as follows:

1. Inner product:

$$\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle = \sum_{k=1}^{n} x_{1k} x_{2k},$$

   where x_{ik} is the k-th component of vector $\boldsymbol{x}_i$.

2. Norm:

$$||\boldsymbol{x}|| = \sqrt{\sum_{k=1}^{n} x_k^2}.$$

3. Distance:

$$\rho(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{\sum_{k=1}^{n} (x_{1k} - x_{2k})^2}.$$

A2. Optimization problems: basic terminology

Consider the following constrained optimization problem:

$$f(\boldsymbol{x}) \to \min_{\boldsymbol{x}} \tag{A.1}$$

s.t.

$$h_1(\boldsymbol{x}) = 0, \ \ldots, \ h_m(\boldsymbol{x}) = 0 \tag{A.2}$$

$$g_1(\boldsymbol{x}) \le 0, \ \ldots, \ g_p(\boldsymbol{x}) \le 0. \tag{A.3}$$

Note that maximization of function f can be equivalently rewritten as minimization of function $\tilde{f} = -f$, and inequality of the form $g_i(\boldsymbol{x}) \ge 0$ can be equivalently rewritten as $\tilde{g}_i(\boldsymbol{x}) \le 0$, where $\tilde{g}_i = -g_i$.

Feasible solution

Point $\boldsymbol{x}$ that satisfies constraints (A.2)-(A.3) is called a *feasible solution* of optimization problem (A.1)-(A.3) or simply *feasible point*. A set of all feasible points is called *feasible region* (or *feasible set*) defined by constraints. Optimization problem is called *infeasible* if given constraints define empty feasible region.

Local and global solutions

Point $\boldsymbol{x}^*$ is called a *local solution* of optimization problem (A.1)-(A.3) if it is feasible and $f(\boldsymbol{x}^*) \le f(\boldsymbol{x})$ for all feasible $\boldsymbol{x}$ from some neighborhood of $\boldsymbol{x}^*$, that is, for all feasible $\boldsymbol{x}$ such that $||\boldsymbol{x}^* - \boldsymbol{x}|| \le \epsilon$ for some $\epsilon > 0$.

Point $\boldsymbol{x}^*$ is called a *global solution* of optimization problem (A.1)-(A.3) if it is feasible and $f(\boldsymbol{x}^*) \le f(\boldsymbol{x})$ for all feasible $\boldsymbol{x}$.

Unbounded problem

If for any feasible $\boldsymbol{x}$ there exist feasible $\boldsymbol{x}'$ such that $f(\boldsymbol{x}') < f(\boldsymbol{x})$ then optimization problem (A.1)-(A.3) is called *unbounded*.

An optimization problem may have a single solution, many solutions, or no solutions at all. Unbounded problem cannot have a global solution but can have local solutions.

A3. Necessary and sufficient conditions for local solutions of optimization problems

Table 1 summarizes necessary and sufficient conditions for a local solution of unconstrained and constrained optimization problems. We consider minimization of a function f of one or many variables. First-order conditions for problems (A), (B), (C), and (D) are written under assumption that functions f, h_i, g_i are differentiable. Second-order conditions for problems (A) and (B) assume that f is twice differentiable, while second-order conditions for problems (C) and (D) assume that f, h_i, g_i are twice differentiable, their second derivatives are continuous functions, and $\boldsymbol{x}^*$ is a regular point of the constraints (see the end of this subsection).

For a function of one or many variables, in case when no constraints are given (problems (A) and (B)), a local minimum must be a stationary point, that is, a point where the first derivative of the function is zero, or all first partial derivatives are zero (i.e., the gradient of the function is zero). If the second derivative is positive in a stationary point x^*, or, for a function of many variables, the matrix of second-order derivatives (the Hessian) is a positive definite matrix, then f is strictly convex in some neighborhood of x^* and therefore x^* is a strict local minimum[19].

For optimization problems with constraints (problems (C) and (D)), conditions for a local solution can be written very similar to the case without constraints by using *the Lagrange function* L (also called *the Lagrangian*). Variables λ_i and μ_i used to define the Lagrange function L are called *Lagrange multipliers*. Note that in Karush-Kuhn-Tucker conditions values λ_i^* of the variables λ_i associated with equality constraints may have arbitrary sign, while values μ_i^* of the variables μ_i associated with inequality constraints $g_i(\boldsymbol{x}) \leq 0$ must be nonnegative.

Let us reformulate the necessary condition for a local solution of problem (C) written by using the notation of the Lagrangian. Equation $\nabla L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}$ is equivalent to

$$\nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}$$
$$\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}.$$

The first equation expands to

$$\nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(\boldsymbol{x}^*) = \mathbf{0},$$

which means that in the point of the local constrained minimum the gradient of the objective function is a linear combination of the gradients of the constraints. The second equation expands to

$$h_1(\boldsymbol{x}^*) = 0,\ h_2(\boldsymbol{x}^*) = 0,\ \ldots,\ h_m(\boldsymbol{x}^*) = 0,$$

which simply restates that $\boldsymbol{x}^*$ is feasible. From this we can see that requirements $\nabla L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}$ and $\nabla L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \mathbf{0}$ used in the sufficient conditions for problems (C) and (D) imply that $\boldsymbol{x}^*$ is feasible.

[19] Matrix A is called positive definite if $\boldsymbol{x} A \boldsymbol{x}^T = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j > 0$ for any $\boldsymbol{x} \neq \mathbf{0}$. It is called positive semidefinite if $\boldsymbol{x} A \boldsymbol{x}^T \geq 0$ for any $\boldsymbol{x}$.

Table 1: Necessary and sufficient conditions for a local solution of unconstrained and constrained optimization problems *(continued on the next page)*.

	Optimization problem	Necessary conditions for a local solution x^* (first-order conditions)	Sufficient conditions for a local solution x^* (second-order conditions)
(A)	$f(x) \to \min$	$f'(x^*) = 0$	$f'(x^*) = 0, \;\; f''(x^*) > 0$
	Function of one variable	*The first derivative is zero*	*The first derivative is zero, and the second derivative is positive*
(B)	$\boldsymbol{x} = (x_1, x_2, ..., x_n)$ $f(\boldsymbol{x}) \to \min$	$\nabla f(\boldsymbol{x}^*) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ ... \\ \frac{\partial f}{\partial x_n} \end{pmatrix} = \mathbf{0}$	$\nabla f(\boldsymbol{x}^*) = \mathbf{0}, \; \mathcal{F}(\boldsymbol{x}^*) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & ... & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ ... & & ... \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & ... & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix} > 0$
	Function of many variables	*The gradient of the function is zero*	*The gradient of the function is zero, and the Hessian matrix is positive definite*
(C)	$f(\boldsymbol{x}) \to \min$ s.t. $h_1(\boldsymbol{x}) = 0, \; ... \; , \; h_m(\boldsymbol{x}) = 0$	There exist numbers $\lambda_1^*, \lambda_2^*, ..., \lambda_m^*$ such that $\nabla L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}$ where $L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i h_i(\boldsymbol{x})$	There exist numbers $\lambda_1^*, \lambda_2^*, ..., \lambda_m^*$ such that $\nabla L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}, \;\; \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) > 0$
	Function of many variables and equality constraints	*The gradient of the Lagrangian is zero*	*The gradient of the Lagrangian is zero, and the Hessian matrix of the Lagrangian is positive definite*

Table 1: Necessary and sufficient conditions for a local solution of unconstrained and constrained optimization problems *(continued from the previous page)*.

	Optimization problem	Necessary conditions for a local solution x^* (first-order conditions)	Sufficient conditions for a local solution x^* (second-order conditions)
(D)	$f(\boldsymbol{x}) \to \min$ s.t. $h_1(\boldsymbol{x}) = 0,\ \ldots,\ h_m(\boldsymbol{x}) = 0$ $g_1(\boldsymbol{x}) \le 0,\ \ldots,\ g_p(\boldsymbol{x}) \le 0$	There exist numbers $\lambda_1^*, \lambda_2^*, ..., \lambda_m^*$, and $\mu_1^* \ge 0, \mu_2^* \ge 0, ..., \mu_p^* \ge 0$ such that $\nabla L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \boldsymbol{0},\ \sum_{i=1}^{p} \mu_i^* g_i(\boldsymbol{x}^*) = 0,$ where $L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i h_i(\boldsymbol{x}) + \sum_{i=1}^{p} \mu_i g_i(\boldsymbol{x})$	There exist numbers $\lambda_1^*, \lambda_2^*, ..., \lambda_m^*$, and $\mu_1^* \ge 0, \mu_2^* \ge 0, ..., \mu_p^* \ge 0$ such that $\nabla L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \boldsymbol{0},\ \sum_{i=1}^{p} \mu_i^* g_i(\boldsymbol{x}^*) = 0$ $\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda^*}, \boldsymbol{\mu^*}) > 0$
	Function of many variables, equality and inequality constraints	*Karush-Kuhn-Tucker conditions*	*The gradient of the Lagrangian is zero, and the Hessian matrix of the Lagrangian is positive definite*

As we noted earlier, the necessary conditions for a local solution $\boldsymbol{x}^*$ of problems (C) and (D) are valid under assumption that $\boldsymbol{x}^*$ is a *regular point* of the constraints. For problem (C), which has equality constraints only, point $\boldsymbol{x}^*$ is called regular if vectors $\nabla h_1(\boldsymbol{x}^*), \nabla h_2(\boldsymbol{x}^*)$, ..., $\nabla h_m(\boldsymbol{x}^*)$ are linearly independent. For problem (D), which also has inequality constraints, point $\boldsymbol{x}^*$ is called regular if vectors $\nabla h_1(\boldsymbol{x}^*), \nabla h_2(\boldsymbol{x}^*), ..., \nabla h_m(\boldsymbol{x}^*)$, $\nabla g_{j_1}(\boldsymbol{x}^*), \nabla g_{j_2}(\boldsymbol{x}^*), ..., \nabla g_{j_t}(\boldsymbol{x}^*)$ are linearly independent, where $j_1, j_2, ..., j_t$ are all indices such that $g_{j_1}(\boldsymbol{x}^*) = 0$, $g_{j_2}(\boldsymbol{x}^*) = 0$, ..., $g_{j_t}(\boldsymbol{x}^*) = 0$.

A4. Convexity

Convex set

Set S is called *convex* if for any two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ from S a point $t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_2$ also belongs to S for any $t : 0 \le t \le 1$. This means that for any two points from a convex set, the segment connecting these points also belongs to the set.

Convex function

Function $f(\boldsymbol{x})$ defined on a convex set S is called *convex* if

$$f(t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_2) \le tf(\boldsymbol{x}_1) + (1-t)f(\boldsymbol{x}_2) \tag{A.4}$$

for any $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ from S, and any number $t : 0 \le t \le 1$. Note that if S is not a convex set then function f cannot be convex on S. If instead of (A.4) function f satisfies analogous strict inequality for any $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ from S, $\boldsymbol{x}_1 \ne \boldsymbol{x}_2$, and any $t : 0 < t < 1$, then f is called *strictly convex*.

Concave function

Function f defined on a convex set S is called *concave* (*strictly concave*) if function $-f$ is convex (strictly convex) on S.

A simplest example of convex function is the square function $f(x) = x^2$ or $f(\boldsymbol{x}) = ||\boldsymbol{x}||^2 = \sum_{i=1}^n x_i^2$. Function $f(x) = x^3$ is convex on the set $S_1 = \{x : x \ge 0\}$ and concave on the set $S_2 = \{x : x \le 0\}$. Linear functions – that is, functions of the form $f(x) = ax + b$ or $f(\boldsymbol{x}) = \sum_{k=1}^n a_k x_k + b$ – are convex and concave simultaneously on their entire domain.

A function $f(\boldsymbol{x})$ defined on a convex set S, having two continuous derivatives, is convex if and only if the Hessian $\mathcal{F}(\boldsymbol{x})$ of $f(\boldsymbol{x})$ is positive semidefinite for all $\boldsymbol{x} \in S$. If $\mathcal{F}(\boldsymbol{x})$ is positive definite for all $\boldsymbol{x} \in S$, then $f(\boldsymbol{x})$ is strictly convex on S. However, the converse is not true.

Minimization of convex function

For a convex function f, first-order necessary conditions for a local minimizer listed in the second column of Table 1 for problems (A) and (B) are sufficient conditions. They are also sufficient conditions for a local minimizer for problems (C) and (D) when these problems are convex, that is, when in addition to convexity of f, each h_i is linear and each g_i is convex.

A local minimum of a convex function is also its global minimum. If function has global minimum and is strictly convex, then the global minimum is unique.

It was mentioned in subsection A3 that the necessary conditions for a local solution $\boldsymbol{x}^*$ of problems (C) and (D) are valid under assumption that $\boldsymbol{x}^*$ is a regular point of the constraints. For convex optimization problems, this regularity requirement may be replaced with either one of the following two conditions:

1. All functions $h_i,\ i = 1, 2, ..., m$, and $g_i,\ i = 1, 2, ..., p$, are linear.

2. Point $\boldsymbol{x}^*$ satisfies constraints $h_i(\boldsymbol{x}^*) = 0,\ i = 1, 2, ..., m$, and $g_i(\boldsymbol{x}^*) < 0,\ i = 1, 2, ..., p$ (this is called *Slater condition*).