DIGITAL CULTURE
ITMO UNIVERSITY

Support Vector Machine

High School of Digital Culture
ITMO University
dc@itmo.ru

Contents

1 Support Vector Machine (SVM): Linearly Separable Sample

1.1 Introduction

Hello everyone! This module will discuss a support vector machine, or SVM, which is an effective technique used to solve classification problems. SVM has been developed in the computer science community in the 1990s and has grown in popularity since then. Before giving the strict description, let's try to grasp the idea of the technique at the intuitive and geometric levels.

1.2 Heuristic Arguments

From now on, we will assume that there are only two classes. Which is to say, we will talk about a binary classification. It has been well noted when we were discussing different classifiers that the introduced constraint is an encumbrance (because it narrows the range of problems that the technique can solve). Nevertheless, we still can examine a broad array of issues and problems that have only two possible solutions. They can be labeled as, for example, yes or no, plus or minus, and so on. Moreover, there are ways to generalize binary classifiers to multiclass ones.

For now, let's return to our goal of understanding the idea behind SVM using simple examples. So, let's look at fig. 1. Here we see 10 training data items x_1, x_2, ..., x_{10}, each having two attributes (let's say, $x_i = (x_{i1}, x_{i2})$) and one response (say, $+1$ and -1). To make the difference more clear, the data items with the response $+1$ are represented by the red circles, and the data items with the response -1 by blue. For example, the class $+1$ includes the data with the predictors $(1, 0)$, $(2, 1)$, $(1, 2)$, $(0.5, 4)$, while -1 includes $(4, 1)$, $(3.5, 3)$, $(2.5, 5)$, and so on. The data is given in the table for ease of understanding.

Object	Values of X_1	Values of X_2	Response
x_1	$x_{11} = 0.5$	$x_{12} = 4$	$+1$
x_2	$x_{21} = 1$	$x_{22} = 0$	$+1$
x_3	$x_{31} = 1$	$x_{32} = 2$	$+1$
x_4	$x_{41} = 2$	$x_{42} = 1$	$+1$
x_5	$x_{51} = 2.5$	$x_{52} = 5$	-1
x_6	$x_{61} = 3$	$x_{62} = 7$	-1
x_7	$x_{71} = 3.5$	$x_{72} = 3$	-1
x_8	$x_{81} = 4$	$x_{82} = 1$	-1
x_9	$x_{91} = 4$	$x_{92} = 6$	-1
x_{10}	$x_{10\ 1} = 5$	$x_{10\ 2} = 4$	-1

What do we see? The red items are distinctively separated from the blue ones so that they can be separated by a straight line (such samples are called linearly separable). But how to draw the straight line? Fig. 2 shows several ways of

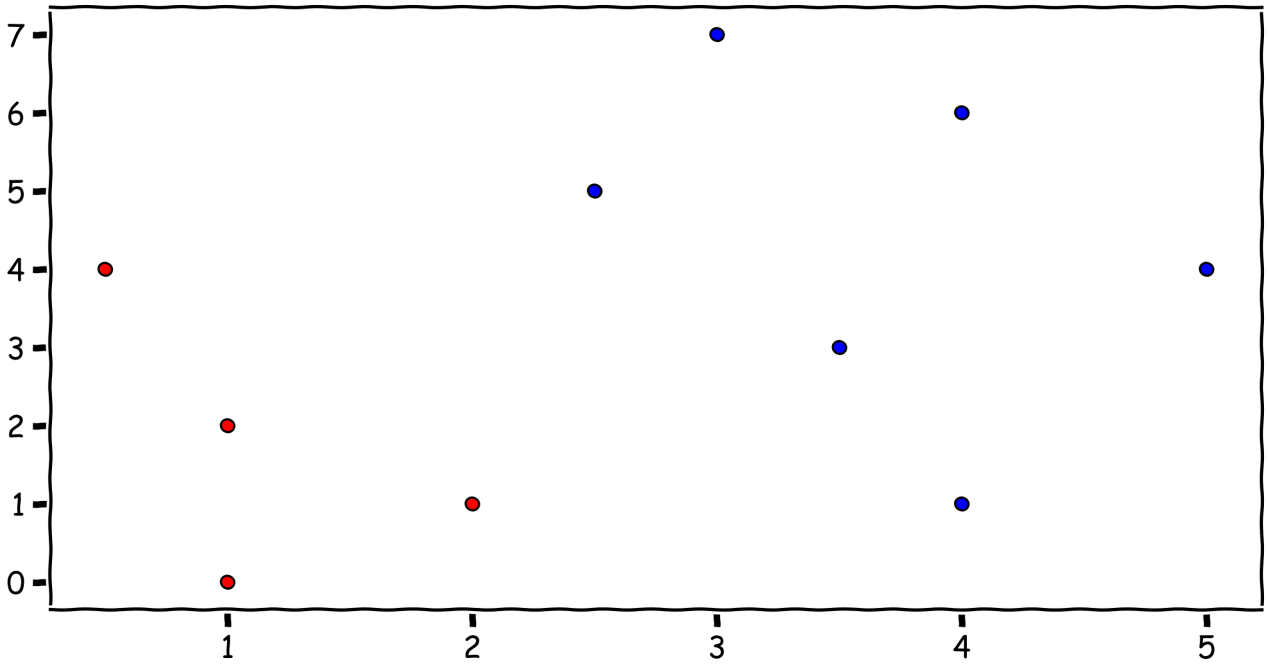


Figure 1: The linearly separable sample. The red items belong to the $+1$ class, and blue to -1.

constructing such a straight line. Which one seems to be better? Well, it's logical to make the following condition. The elements of different classes should be as far from the separating straight line as possible. It will allow us to differentiate the classes with ease. Thus, the orange straight line is not our candidate. Although it separates blues and reds, the distance between the line and the closest elements of the classes is extremely small (the closest element of blues is the object with the coordinates $(4, 1)$, and of reds is with the coordinates $(0.5, 4)$).

What about the blue straight line? It doesn't seem fine either. The distance between it and the closest element of reds is less than that to the closest element of blues. Why is it so hardly pushed down? How is it confirmed? Based on purely geometrical considerations, we formulate the second condition. The distance between the separating straight line and the closest element of one and another class should be the same (the straight line should be right in between the two classes). Thus, among the proposed variants, only two straight lines remain. They are the green and violet lines. Let's talk about them in more detail and introduce the third condition relating to the desired separating straight line.

Look at modified fig. 3. The violet and green dashed straight lines pass through the closest elements of the classes up to the violet and green separating straight lines, respectively.

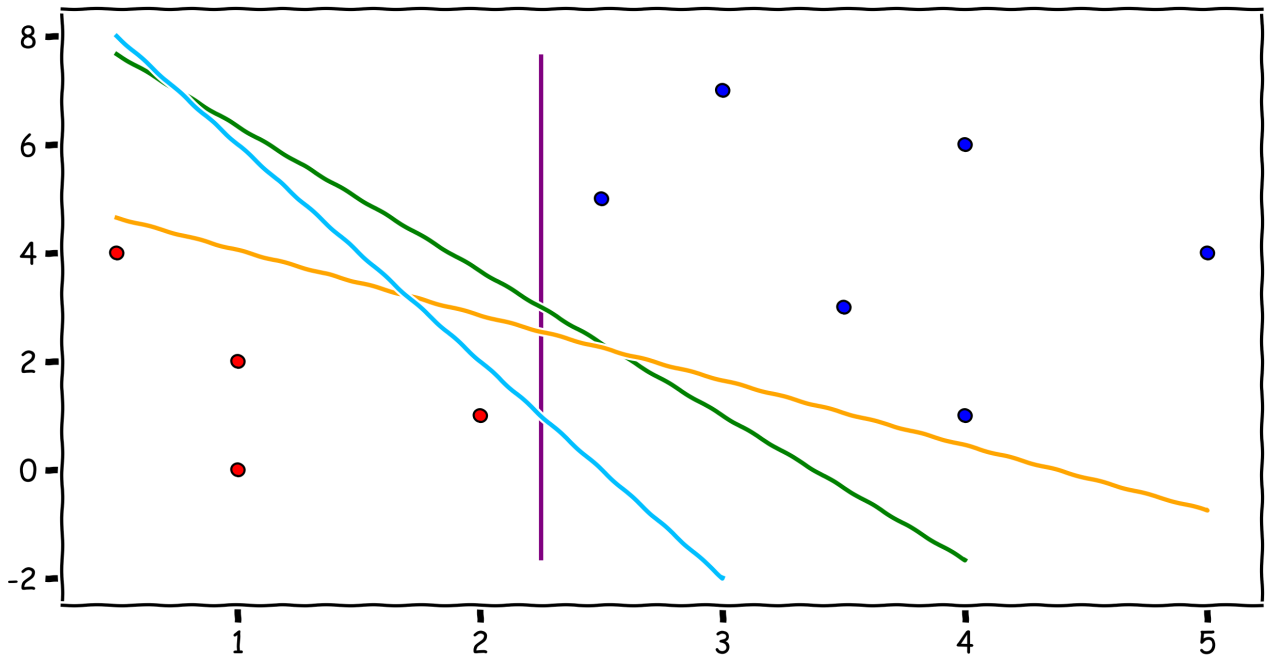


Figure 2: Possible straight lines that separate the two classes.

We inherently understand that the green straight line is better because the distance between the green dashed lines is greater than that between the violets. Thus, the classification is more confident. Hence, the third condition is that the separating strip should be the widest.

For ease of understanding, you can think of the separating straight line and dashed lines on its sides as a bit like a city road. The separating line is a solid line that divides the roadway into two equal parts, and the dashed line is a border between the roadway and the sidewalk. Clearly, the greater the distance between the sidewalks, the greater the road capacity, and, since 'drivers' have an equal right to both lanes of the road, the separating line is right in the middle. On the other hand, the distance between the dashed lines cannot be unlimitedly large because the 'pedestrians' (our training data) will be on the roadway.

These geometrical considerations lead us to the main idea of SVM in case of a linearly separable sample, that is, to construct a separating hyperplane that is equidistant and farthest from the closest training data items of the classes. But how to explain this task to a computer? How to deal with multidimensional spaces when there's no visualization? How to solve a classification problem when the separating hyperplane is constructed? Well, let's see. We will start with classification and multidimensionality.

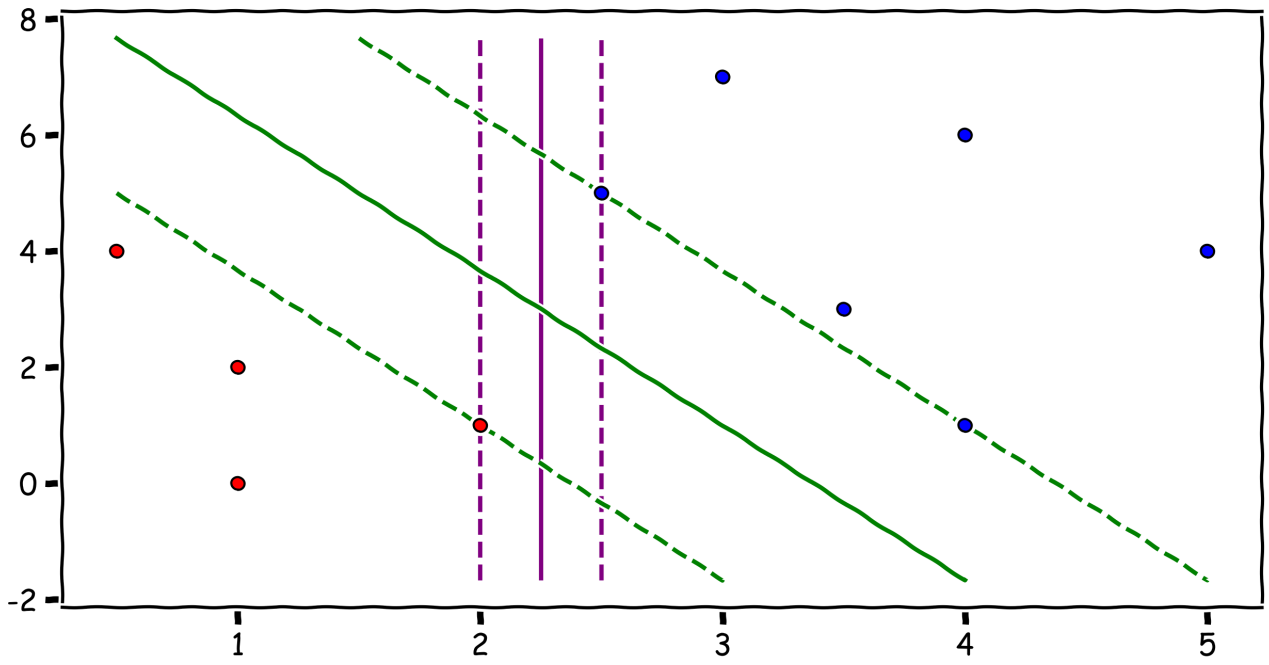


Figure 3: Separating strips.

1.3 Brief Overview of Hyperplane and Hyperplane-Based Classification

We have already used the concept of hyperplane while discussing the linear and logistic regression. In a two-dimensional space, the hyperplane l is a set of points satisfying an equation of the form

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 = 0,$$

where at least one coefficient of X_1 and X_2 is not zero. Thus, $\theta_1^2 + \theta_2^2 \neq 0$. This equation is nothing but the equation of a straight line in the plane, and the vector $n = (\theta_1, \theta_2)$ is a normal vector to the straight line. The normal vector to the straight line has one useful property. It's perpendicular to the straight line. It means the following. If the points A and B lie in l, the vectors AB and n are perpendicular (orthogonal). In terms of a dot product discussed earlier, it means that

$$(AB, n) = 0.$$

In a three-dimensional space, the hyperplane l is a set of points satisfying an equation of the form

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_3 = 0,$$

where at least one coefficient of X_1, X_2, and X_3 is not zero. Thus, $\theta_1^2 + \theta_2^2 + \theta_3^2 \neq 0$. Clearly, it's the equation of a plane in space, and the vector $n = (\theta_1, \theta_2, \theta_3)$ is

a normal vector to the plane. The normal vector is perpendicular to the plane, which is the same as in the case of the straight line. If the points A and B lie in l, the vectors AB and n are perpendicular (orthogonal). In terms of a dot product, we obtain that

$$(AB, n) = 0.$$

Perhaps, the logic is clear. Therefore, we can introduce a general definition.

Definition 1.3.1 *In a p-dimensional space, the hyperplane l is a set of points the coordinates of which satisfy an equation of the form*

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0,$$

where $\theta_1^2 + \theta_2^2 + ... + \theta_p^2 \neq 0$.

Moreover, it is logical to accept the following definition.

Definition 1.3.2 *Let the hyperplane l be defined by the equation of the form:*

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0.$$

The vector $n = (\theta_1, \theta_2, ..., \theta_p)$ is called a normal to the hyperplane l.

It can be proved that the normal to the hyperplane is perpendicular to the hyperplane, as discussed earlier. A vector formed by any two points in a hyperplane is perpendicular (orthogonal) to its normal.

Well, but how to perform classification using a hyperplane? Actually, it's quite easy. A hyperplane divides the space into two parts. Imagine a straight line that separates a plane and a plane that separates a three-dimensional space. All we've got left is to understand which of the two parts will contain the test observation. In cases of more than 3 space dimensions, visualization becomes more complicated, and an algebraic approach is used.

Assume that the hyperplane l is defined by the equation of the following form:

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0.$$

Let's consider the function

$$f(X) = f(X_1, X_2, ..., X_p) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p.$$

If the point X lies in the hyperplane, then, according to the definition of the latter, $f(X) = 0$. If the point X does not lie in the hyperplane, then either $f(X) > 0$, which means that the point lies from one side of the hyperplane (thus, falls to one class that we will designate by $+1$), either $f(X) < 0$, which means that the point lies from the other side of the hyperplane (thus, falls to another class that we will designate by -1). To determine from which side of the hyperplane a test observation lies (which class the test point belongs to), we can calculate the value of the function $f(X)$ on it.

Remark 1.3.1 *At this point, we should stop for a moment and think about conclusions we can draw from the facts. The sign of the expression $f(X)$ given the same X (as well as the class enumeration) depends on the directions of the normal. Let the hyperplane l be defined by*

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0.$$

Then, given any $k \neq 0$, the equation

$$k\left(\theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p\right) = 0$$

defines the same hyperplane. However, if $k < 0$, the normal to the hyperplane with the coordinates $(k\theta_1, k\theta_2, ..., k\theta_p)$ is in the opposite direction to the normal $n = (\theta_1, \theta_2, ..., \theta_p)$.

For this reason, we will consider a class positive if its points belong to a part of the separated space to which the normal is pointing (as in the logistic regression case).

Let's look at the example. Suppose we have the same training data as earlier and 3 ways of dividing the plane into two parts (each corresponds to the straight line of the respective color). We will assume that red points belong to the +1

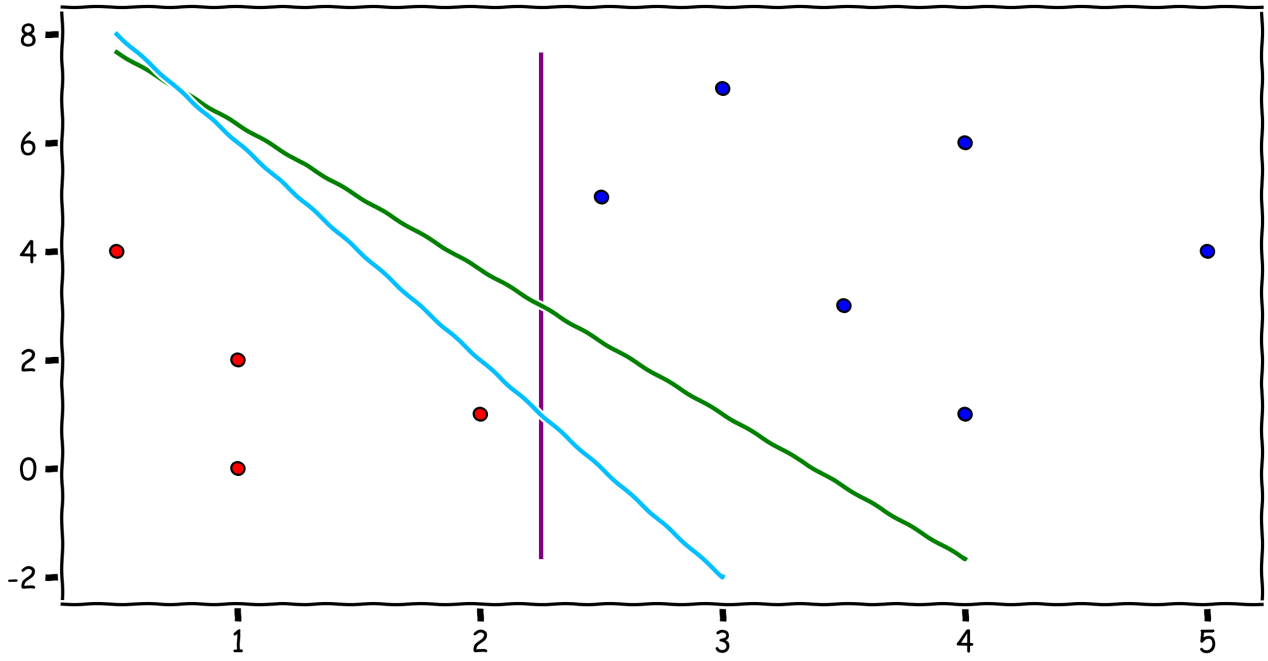


Figure 4: The linearly separable sample. The red items belong to the +1 class, and blue to −1.

class, and blue to −1. The straight lines are given by the equations. A blue line is given by $-4X_1 - X_2 + 10 = 0$, violet by $X_1 - 2.25 = 0$, and green by

$-2.667X_1 - X_2 + 8.999 = 0$. Is it correct, considering that we've chosen the class enumeration?

Well, the normal to the blue line with the coordinates $(-4, -1)$ is pointed towards reds. Thus, everything is correct. The normal to the violet line with the coordinates $(1, 0)$ is pointed towards blues. To preserve the class enumeration, we need to rewrite the equation of the violet line as $-X_1 + 2.25 = 0$ (we've multiplied the entire equation by -1 to change the direction of the normal). The normal to the green line with the coordinates $(-2.667, -1)$ is pointed towards reds. Thus, no problem with that. There are three functions (with respect to each line) to be classified:

$$f_{\text{blue}}(X_1, X_2) = -4X_1 - X_2 + 10,$$
$$f_{\text{violet}}(X_1, X_2) = -X_1 + 2.25,$$
$$f_{\text{green}}(X_1, X_2) = -2.667X_1 - X_2 + 8.999.$$

Let's classify the orange test object with the coordinates $(2, 3)$. The figure makes

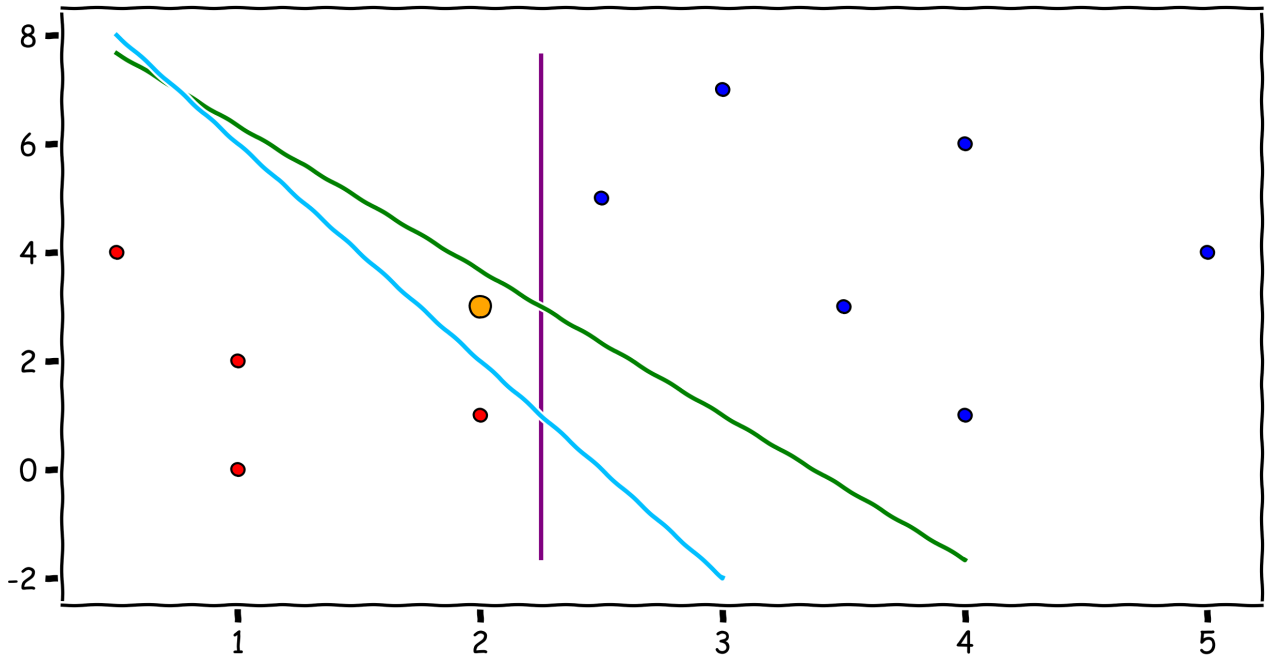


Figure 5: Classification of a new object with the coordinates $(2, 3)$.

it clear that the object will be assigned to the class -1 with respect to the blue line because the object is on the same side with blues, and the object will be assigned to the class $+1$ with respect to other lines. As has been well noted, it's not always possible to perform visualization, so let's conduct an analysis. For the blue line,

$$f_{\text{blue}}(2, 3) = -8 - 3 + 10 = -1 < 0,$$

we obtain a negative value. Thus, the assigned class is -1. As to the remaining lines,

$$f_{\text{violet}}(2,3) = -2 + 2.25 = 0.25 > 0,$$

$$f_{\text{green}}(X_1, X_2) = -2.667 \cdot 2 - 3 + 8.999 = 0.665 > 0,$$

the object is assigned to the class $+1$ (as we've assumed).

To sum up, we can make the following conclusions.

1. The hyperplane

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0$$

   separates the p-dimensional space into 2 parts.

2. The points from one side of the space belong to one class, and the points from the other side belong to another class.

3. The normal $n = (\theta_1, \theta_2, ..., \theta_p)$ to the hyperplane is pointed towards such a part of the space for which points the classifier gives the positive values. This part of the space is often called the class $+1$, and another -1.

4. The classifier is given by the following analytical expression:

$$f(X) = f(X_1, X_2, ..., X_p) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p.$$

5. To classify the test item x^* with the predictor values $(x_1^*, x_2^*, ..., x_p^*)$, we calculate the value of f:

$$f(x^*) = \theta_0 + \theta_1 x_1^* + \theta_2 x_2^* + ... + \theta_p x_p^*.$$

   If the value of $f(x^*)$ is positive, the object x^* is assigned to the class $+1$, and if negative, to the class -1. If the value is zero, the object lies in the hyperplane, and it can be assigned to any class.

6. Now the last one. The greater the value of $|f(x^*)|$ for the test object x^*, the more distant the object from the separating hyperplane, and the more confident the classification becomes, that is, the less the chance to introduce an error and assign the object to a wrong class.

Now we are ready to answer the last question posed. How to construct an optimal hyperplane we discussed in the very beginning? Let's find out.

1.4 Mathematical Construction of an Optimal Separating Hyperplane

Assume that we have n training data items, that is, n responses Y to p predictors $X_1, X_2, ..., X_p$, and the data is linearly separable. Each training data item x_i, $i \in \{1, 2, ..., n\}$ can be identified with $(p+1)$ numerical values. It's a set

$$x_i = (x_{i1}, x_{i2}, ..., x_{ip}), \quad i \in \{1, 2, ..., n\}$$

of p predictors and the response y_i that is equal to $+1$ or -1, depending on the class of a training object. Let's find the separating hyperplane l in the form:

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0$$

given that

$$\theta_1^2 + \theta_2^2 + ... + \theta_p^2 = \sum_{i=1}^{p} \theta_i^2 = 1.$$

The last condition ensures that the length $|n|$ of the normal $n = (\theta_1, \theta_2, ..., \theta_p)$ to the plane is equal to one. It's easy to achieve for any hyperplane by multiplying the equation

$$\theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0$$

by $\left(\theta_1^2 + \theta_2^2 + ... + \theta_p^2\right)^{-1/2} = |n|^{-1/2}$. Thus, the introduced constraint doesn't affect the overall integrity. To avoid additional notations, we will assume that the coefficients of the equation of a hyperplane satisfy the condition. Recall the following theorem from analytic geometry.

Theorem 1.4.1 *Assume that the hyperplane l is defined by the equation of the following form:*

$$l: \ \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0,$$

given that $|n| = 1$, $n = (\theta_1, \theta_2, ..., \theta_p)$. Then the distance $\rho(X, l)$ from an arbitrary point $X = (X_1, X_2, ..., X_p)$ to the hyperplane l can be calculated as

$$\rho(X, l) = |\theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p|.$$

Remark 1.4.1 *Note that the last expression is the absolute value of the classifier $f(X)$ defined by the hyperplane l:*

$$f(X) = f(X_1, X_2, ..., X_p) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p,$$

calculated for the object X.

We want to construct a hyperplane with the normal pointed towards that part of the space where the objects of class +1 (with the response +1) are located. Based on the training data, the expression for the distance to the hyperplane can be written as follows:

$$\rho(x_i, l) = y_i \left(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_p x_{ip}\right), \quad i \in \{1, 2, ..., n\}.$$

Let's write down three conditions relating to the hyperplane in the language of mathematics. We want each training data item to be correctly classified and the distance from it to the hyperplane to be as small as possible. Thus, we are looking for a such largest non-negative M that the distance from each training data item to the hyperplane satisfies the inequality

$$\rho(x_i, l) = y_i \left(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_p x_{ip}\right) \geq M,$$

by varying (or changing) the parameters $\theta_0, \theta_1, ..., \theta_p$. More precisely, we are looking for the hyperplane (or for the coefficients), considering that the value M is maximized.

Hence, all the conditions can be written as follows:

$$\max_{\theta_0, \theta_1, ..., \theta_p} M,$$

$$y_i \left(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_p x_{ip}\right) \geq M,$$

$$\theta_1^2 + \theta_2^2 + ... + \theta_p^2 = \sum_{i=1}^{p} \theta_i^2 = 1.$$

The optimization problem is solved numerically, which leads us to the following definition.

Definition 1.4.1 *The hyperplane l, which equation has the form:*

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0,$$

where the coefficients $\theta_0, \theta_1, ..., \theta_p$ are derived from the optimization problem, is called an optimal separating hyperplane.

To study the method described, let's rewrite the problem in less obvious, but more convenient, notations. Let $w = (\theta_1, \theta_2, ..., \theta_p)$ be a hyperplane normal vector, $w_0 = -\theta_0$, $X = (X_1, X_2, ..., X_p)$. The equation of the hyperplane is rewritten as follows:

$$(w, X) - w_0 = 0.$$

To ease the optimization problem, we'll divide the inequality by M

$$y_i \left(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_p x_{ip}\right) \geq M,$$

and release the constraint that $|w| = 1$. Since we maximize M, and since the multiplication by a positive constant doesn't affect the minimization, we can formulate the problem as follows:

$$\begin{cases} \frac{1}{2}(w, w) \to \min\limits_{w, w_0} \\ y_i((w, x_i) - w_0) \geq 1, \quad i \in \{1, 2, ..., n\} \end{cases}$$

Well, we minimize half of the square of the normal vector length by changing the normal to the hyperplane and free term, or (what's the same) parameters of the hyperplane $\theta_0, \theta_1, ..., \theta_p$ in such a way that n conditions of the following form are satisfied:

$$y_i((w, x_i) - w_0) \geq 1, \quad i \in \{1, 2, ..., n\},$$

each of them means that all the training data items are outside (or on the boundary) of the separating strip.

We could have formulated the problem this way based on the geometrical considerations mentioned earlier. Let's do this now. The second condition

$$y_i((w, x_i) - w_0) \geq 1 \Leftrightarrow -1 \leq (w, x_i) - w_0 \leq 1$$

defines a strip that separates two classes. Fig. 6 shows two such strips. No

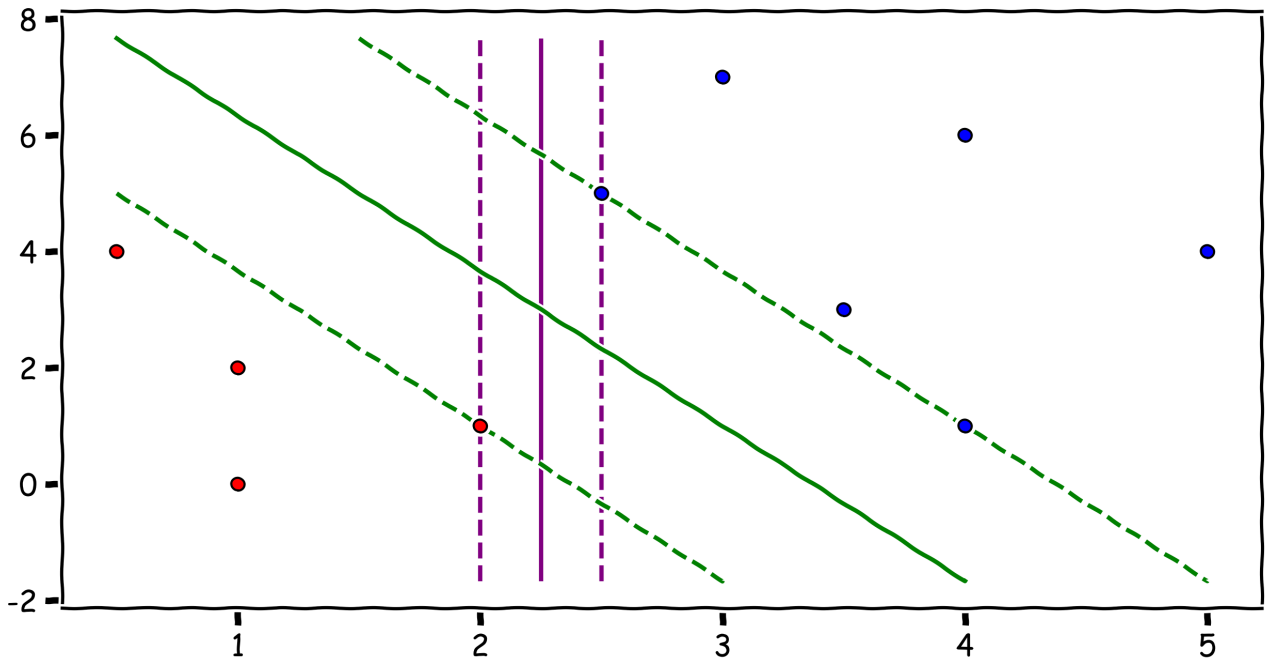


Figure 6: Two separating strips.

training data point can appear within the strip because the strip is separated by two hyperplanes with the normal w. The points closest to the hyperplane lie on the strip boundaries (two on the violet strip and three on the green), and the

separating hyperplane divides the strip into halves. It's quite obvious that the strip must be as wide as possible for the separating hyperplane to be as far as possible from the objects of each class. Let's calculate the width. Let x_- and x_+ be two elements of the classes $+1$ and -1 that lie on the strip boundary. Then the strip width can be calculated as

$$\left(x_+ - x_-, \frac{w}{|w|}\right) = \frac{(x_+, w) - (x_-, w)}{|w|} = \frac{(w_0 + 1) - (w_0 - 1)}{|w|} = \frac{2}{|w|}.$$

Thus, the strip width is maximal when the length $|w|$ of the normal w is minimal (or half of the square of the normal vector length is minimal). Thereby, we obtain the discussed problem:

$$\begin{cases} \frac{1}{2}(w, w) \to \min\limits_{w, w_0} \\ y_i((w, x_i) - w_0) \geq 1, \quad i \in \{1, 2, ..., n\} \end{cases}.$$

We hope the maximal width of the separating strip will provide us with the most accurate splitting of the training data into two classes since the data items are located as far from each other as possible in case of the linear separability.

1.5 Support Vectors and Kuhn – Tucker Conditions

What we described is a conditional extremum problem because we need to find the function minimum given some conditions. For such problems, we can use Kuhn – Tucker conditions (this approach generalizes the method of Lagrange multipliers in conditional extremum problems). We can use the Kuhn – Tucker theorem to reformulate the problem into the problem of finding a saddle point of the Lagrange function.

$$\begin{cases} L(w, w_0, \lambda) = \frac{1}{2}(w, w) - \sum\limits_{i=1}^{n} \lambda_i \left(y_i((w, x_i) - w_0) - 1\right) \to \min\limits_{w, w_0} \max\limits_{\lambda} \\ \lambda_i \geq 0, \quad i \in \{1, 2, ..., n\} \\ \lambda_i = 0 \text{ or } (w, x_i) - w_0 = y_i, \quad i \in \{1, 2, ..., n\} \end{cases}.$$

The first expression $L(w, w_0, \lambda)$ in the posed problem is often called a Lagrange function. The second and third lines describe the conditions established for the arguments of the Lagrange function. The third condition deserves special attention. According to it, the term corresponding to the training data item x_i of the Lagrange function is not zero if and only if $(w, x_i) - w_0 = y_i$, that is, if the object x_i lies on the boundary of the separating strip.

A necessary condition for a saddle point of the Lagrange function is the equality of the partial derivatives to zero, which leads us to the following relations:

$$\frac{\partial}{\partial w} L(w, w_0, \lambda) = w - \sum_{i=1}^{n} \lambda_i y_i x_i = 0,$$

$$\frac{\partial}{\partial w_0} L(w, w_0, \lambda) = \sum_{i=1}^{n} \lambda_i y_i = 0.$$

It follows from the first condition that

$$w = \sum_{i=1}^{n} \lambda_i y_i x_i,$$

that is, the desired vector w is a linear combination of the training data items x_i, in particular, of only those that lie on the boundary of the separating strip (λ_i are zero otherwise).

Definition 1.5.1 *The training data items, for which the condition*

$$(w, x_i) - w_0 = y_i,$$

is true, are called support vectors.

Thus, to construct a separating hyperplane, we only need support vectors. Unfortunately, it's not always easy to differentiate support vectors and thin down the training data.

Let's return to the problem and plug the relations

$$w = \sum_{i=1}^{n} \lambda_i y_i x_i, \quad \sum_{i=1}^{n} \lambda_i y_i = 0$$

into the Lagrange function. Let's consider the first term. It transforms as follows:

$$\frac{1}{2}(w, w) = \frac{1}{2}\left(\sum_{i=1}^{n} \lambda_i y_i x_i, \ \sum_{j=1}^{n} \lambda_j y_j x_j\right) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j (x_i, x_j).$$

Next, we consider the second term that transforms like this:

$$\sum_{i=1}^{n} \lambda_i (y_i((w, x_i) - w_0) - 1) = \sum_{i=1}^{n} \lambda_i y_i (w, x_i) - w_0 \sum_{i=1}^{n} \lambda_i y_i - \sum_{i=1}^{n} \lambda_i.$$

The second term is zero given that $\sum\limits_{i=1}^{n} \lambda_i y_i = 0$. We plug the expression for w into the first term and obtain:

$$\sum_{i=1}^{n} \lambda_i (y_i((w, x_i) - w_0) - 1) = \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j (x_i, x_j) - \sum_{i=1}^{n} \lambda_i.$$

Hence, the Lagrange function takes the following form:

$$L(\lambda) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j (x_i, x_j) + \sum_{i=1}^{n}\lambda_i.$$

As a result, we obtain an equivalent problem written as follows:

$$\begin{cases} -L(\lambda) = -\sum\limits_{i=1}^{n}\lambda_i + \frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}\lambda_i\lambda_j y_i y_j (x_i, x_j) \to \min\limits_{\lambda} \\ \lambda_i \geq 0, \quad i \in \{1, 2, ..., n\} \\ \sum\limits_{i=1}^{n}\lambda_i y_i = 0 \end{cases},$$

where we are looking for the minimum of the function $-L(\lambda)$ instead of the maximum of $L(\lambda)$ (which is the same). The rewritten problem is much better than before. Here the quadratic functional equation with a non-negative-definite quadratic form is minimized, which means it is convex. The area of constraints defined by the inequalities and equality is also convex, which means that the posed problem has one solution.

When the problem is solved, we can use the following algorithm to construct a separating hyperplane:

1. Define w from the relation

$$w = \sum_{i=1}^{n}\lambda_i y_i x_i.$$

2. Find w_0. To do so, we can take a support vector x_i and use $(w, x_i) - w_0 = y_i$ to find

$$w_0 = (w, x_i) - y_i.$$

3. The classifier is given by the analytical expression

$$f(X) = (w, X) - w_0.$$

The next classification steps comply with the general rules given in 1.3.

Definition 1.5.2 *The classifier constructed this way is often called an optimal margin classifier, or hard margin classifier.*

Let's go back to the example. It turns out that the green straight line constructed earlier is an optimal separating hyperplane. 3 training data items (in

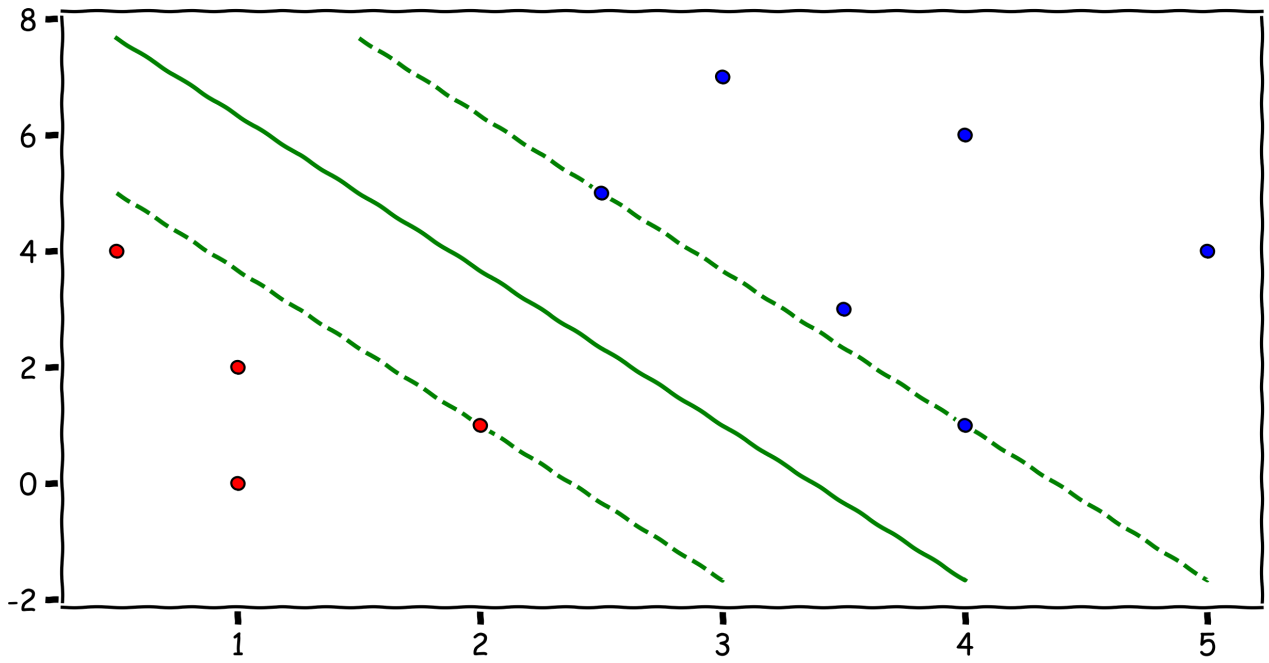


Figure 7: Optimal separating hyperplane.

particular, a red with the coordinates $(2, 1)$ and two blues with the coordinates $(4, 1)$, $(2.5, 5)$) are support vectors. The hyperplane is defined by the equation (with approximate coefficients)

$$-X_1 - 0.375X_2 + 3.375 = 0,$$

and the classifier is defined by the expression

$$f(X_1, X_2) = -X_1 - 0.375X_2 + 3.375.$$

Let's classify two new objects. The first is an orange one with the coordinates $(4, 0)$, and the second object is violet with the coordinates $(1.5, 1)$. It's a geometrical notion that the orange object belongs to blues (-1), and the violet to reds $(+1)$. Moreover, the classification of the red object is not quite confident because the object fell inside the separating strip.

These conclusions are supported analytically, because, for the orange object,

$$f(4, 0) = -4 + 3.375 = -0.625 < 0 \Rightarrow \text{class} - 1,$$

and, for the violet,

$$f(1.5, 1) = -1.5 - 0.375 + 3.375 = 1.5 > 0 \Rightarrow \text{class} + 1.$$

Now you know how to classify linearly separable data, but this approach has obvious drawbacks. First, it is useful only when the data can be separated by a hyperplane. Second, it is sensitive to outliers. We will explain later how to deal with these issues. As for now, let's consider an example to fix this in mind.

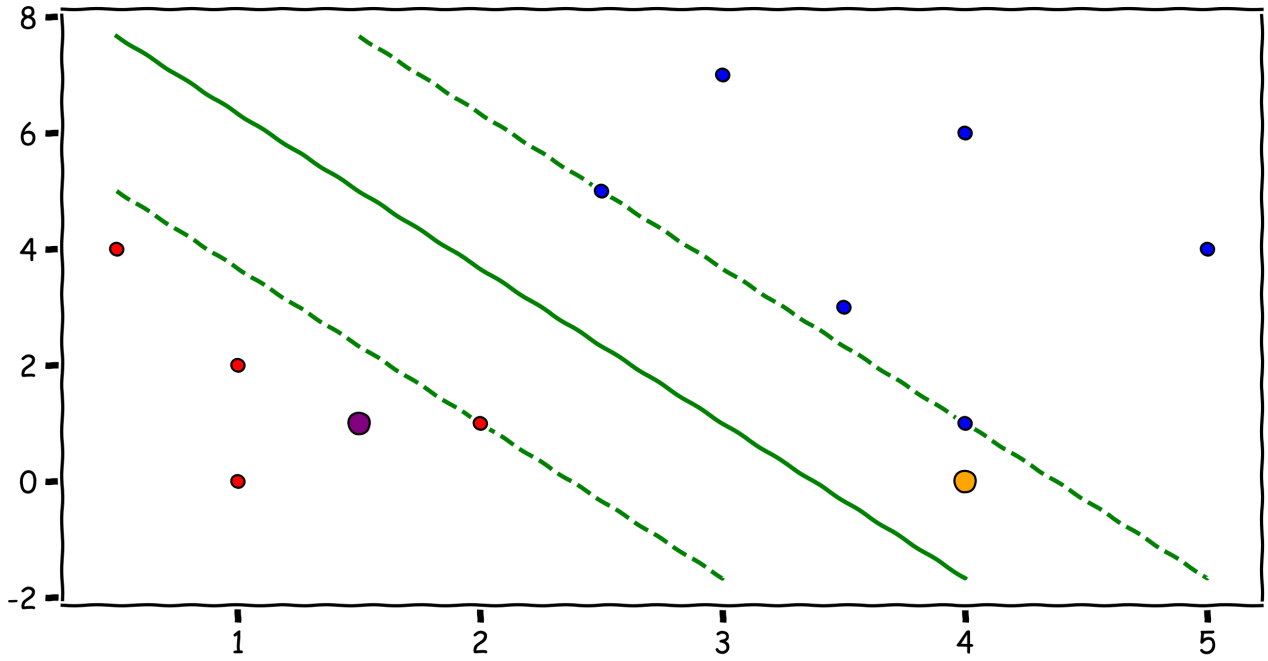


Figure 8: Classification of new objects.

1.6 Simple Detailed Example

To make the formulas seem less scary, we will look into a very synthetic example. Let the class -1 (blue) contain points $x_1 = (0, 0)$ and $x_2 = (1, 0)$, and the class $+1$ (red) contain points $x_3 = (2, 0)$ and $x_4 = (0, 2)$. For convenience, the data is given in the following table:

Object	X_1	X_2	Response
x_1	0	0	-1
x_2	1	0	-1
x_3	2	0	$+1$
x_4	0	2	$+1$

and visualized in the plane in fig. 9. Try to guess based on what you see, which data items will be support vectors, and where the straight line will lie. Perhaps, x_2, x_3, x_4 are the obvious choice for the future support vectors, but let's prove it.

Well, in the introduced notations,

$$x_1 = (0, 0),\ x_{11} = 0,\ x_{12} = 0,$$

$$x_2 = (1, 0),\ x_{21} = 1,\ x_{22} = 0,$$

$$x_3 = (2, 0),\ x_{31} = 2,\ x_{32} = 0,$$

$$x_4 = (0, 2),\ x_{41} = 0,\ x_{42} = 2$$

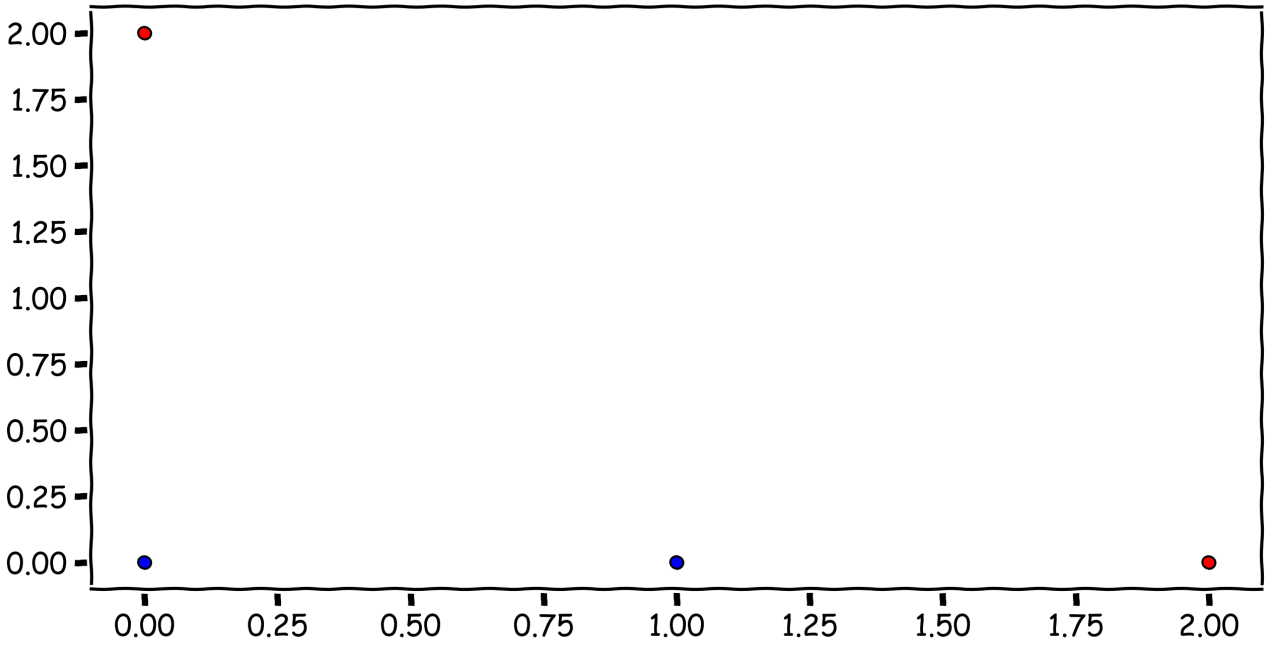


Figure 9: Training data.

and the dot product of the element x_i and the element x_j is calculated as

$$(x_i, x_j) = x_{i1}x_{j1} + x_{i2}x_{j2},\ \ i, j \in \{1, 2, 3, 4\}.$$

Since we have four training objects, that is, $n = 4$, given that

$$(x_1, x_1) = (x_1, x_2) = (x_1, x_3) = (x_1, x_4) = 0,$$

$$(x_2, x_2) = 1,\ \ (x_2, x_3) = 2,\ \ (x_2, x_4) = 0,$$

$$(x_3, x_3) = 4,\ \ (x_3, x_4) = 0,$$

$$(x_4, x_4) = 4,$$

the minimized expression

$$-L(\lambda) = -\sum_{i=1}^{n} \lambda_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j (x_i, x_j)$$

takes the following form:

$$-L(\lambda) = -\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4 + \frac{1}{2} \left(\lambda_2^2 + 4\lambda_3^2 + 4\lambda_4^2 - 4\lambda_2\lambda_3 \right)$$

and, besides that $\lambda_i \geq 0$, $i \in \{1, 2, 3, 4\}$, due to the last condition of the optimization problem

$$\sum_{i=1}^{n} \lambda_i y_i = 0,$$

one more condition is added

$$\lambda_1 + \lambda_2 - \lambda_3 - \lambda_4 = 0.$$

Thus, we obtain the following problem:

$$\begin{cases} -L(\lambda) = -\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4 + \frac{1}{2}\left(\lambda_2^2 + 4\lambda_3^2 + 4\lambda_4^2 - 4\lambda_2\lambda_3\right) \to \min\limits_{\lambda} \\ \lambda_i \geq 0, \quad i \in \{1,2,3,4\} \\ \lambda_1 + \lambda_2 - \lambda_3 - \lambda_4 = 0 \end{cases}.$$

This is a standard problem of the differential calculus for functions of several variables, that is, the problem of finding the smallest value of a given function in a given region under given conditions. This value can be reached for a continuous function either at the inner point of the region or on the boundary.

A necessary condition for the local extremum at the inner point of the region is the equality of the partial derivatives to zero, which leads us to the following system of equations:

$$\begin{cases} -\frac{\partial L(\lambda)}{\partial \lambda_1} = 0 \\ -\frac{\partial L(\lambda)}{\partial \lambda_2} = 0 \\ -\frac{\partial L(\lambda)}{\partial \lambda_3} = 0 \\ -\frac{\partial L(\lambda)}{\partial \lambda_4} = 0 \end{cases} \Leftrightarrow \begin{cases} -1 = 0 \\ -1 + \lambda_2 - 2\lambda_3 = 0 \\ -1 + 4\lambda_3 - 2\lambda_2 = 0 \\ -1 + 4\lambda_4 = 0. \end{cases}$$

The first equation has no solution. Hence, the entire system also has no solution. Thus, we should look for the smallest value of the function on the boundary of the region.

Assume that $\lambda_1 = 0$, then the third equation is rewritten as $\lambda_2 = \lambda_3 + \lambda_4$, and the function, which we study to find the smallest value, is rewritten as follows:

$$-L(\lambda_3, \lambda_4) = -2\lambda_3 - 2\lambda_4 + \frac{1}{2}\left((\lambda_3 + \lambda_4)^2 + 4\lambda_3^2 + 4\lambda_4^2 - 4(\lambda_3 + \lambda_4)\lambda_3\right),$$

and the region, where we are looking for the smallest value, will be defined by the relations $\lambda_3, \lambda_4 \geq 0$.

After calculating the partial derivatives for λ_3 and λ_4, we obtain the following system of equations:

$$\begin{cases} -2 + \lambda_3 + \lambda_4 + 4\lambda_3 - 4\lambda_3 - 2\lambda_4 = 0 \\ -2 + \lambda_3 + \lambda_4 + 4\lambda_4 - 2\lambda_3 = 0 \end{cases} \Leftrightarrow \begin{cases} \lambda_3 - \lambda_4 = 2 \\ -\lambda_3 + 5\lambda_4 = 2 \end{cases},$$

hence, $\lambda_3 = 3$, $\lambda_4 = 1$, $\lambda_2 = 4$, and $\lambda_1 = 0$, and $-L(3,1) = -4$. We can show that it will be the smallest value of the given function in the given region.

By checking the remaining boundaries in the same way, we find out that the smallest value of the function on these boundaries is more than -4. Thus, as has been assumed, the elements x_2, x_3, x_4 are support vectors because the corresponding coefficients λ are not zero.

To construct an optimal separating hyperplane, we find the normal vector w:

$$w = \sum_{i=1}^{4} \lambda_i y_i x_i = 0 \cdot (-1) \cdot x_1 + 4 \cdot (-1) \cdot x_2 + 3 \cdot 1 \cdot x_3 + 1 \cdot 1 \cdot x_4 =$$

$$= -(0,0) - 4(1,0) + 3(2,0) + (0,2) = (0,0) - (4,0) + (6,0) + (0,2) = (2,2).$$

The only thing left is to calculate w_0. We take some support vector, for example, $x_2 = (1,0)$, and calculate

$$w_0 = (w, x_2) - y_2 = 2 - (-1) = 3.$$

Hence, the optimal separating hyperplane is defined by the equation

$$2X_1 + 2X_2 - 3 = 0,$$

and the classifier is defined by the expression

$$f(X) = f(X_1, X_2) = 2X_1 + 2X_2 - 3.$$

Fig. 10 shows the constructed separating hyperplane, input data, as well as the separating strip. The results meet the expectations. Now we want to identify the class of the test observation $x^* = (1,2)$. To do so, we calculate $f(x^*)$ and obtain

$$f(x^*) = f(1,2) = 2 + 4 - 3 = 3 > 0,$$

thus, the test observation belongs to the class $+1$ (of reds). It is confirmed by fig. 11, where the orange point corresponding to the test data item lies in the same half-plane as reds.

2 Support Vector Machine (SVM): Linearly Inseparable Sample

2.1 Hyperplane for Linearly Inseparable Data

The classifier we've been constructing works well when the data can be separated by a hyperplane. However, it is not always like this. Note that a change in one class of the training data can destroy linear separability. To make things worse, this uncommon observation can be an error also called an outlier. Take a

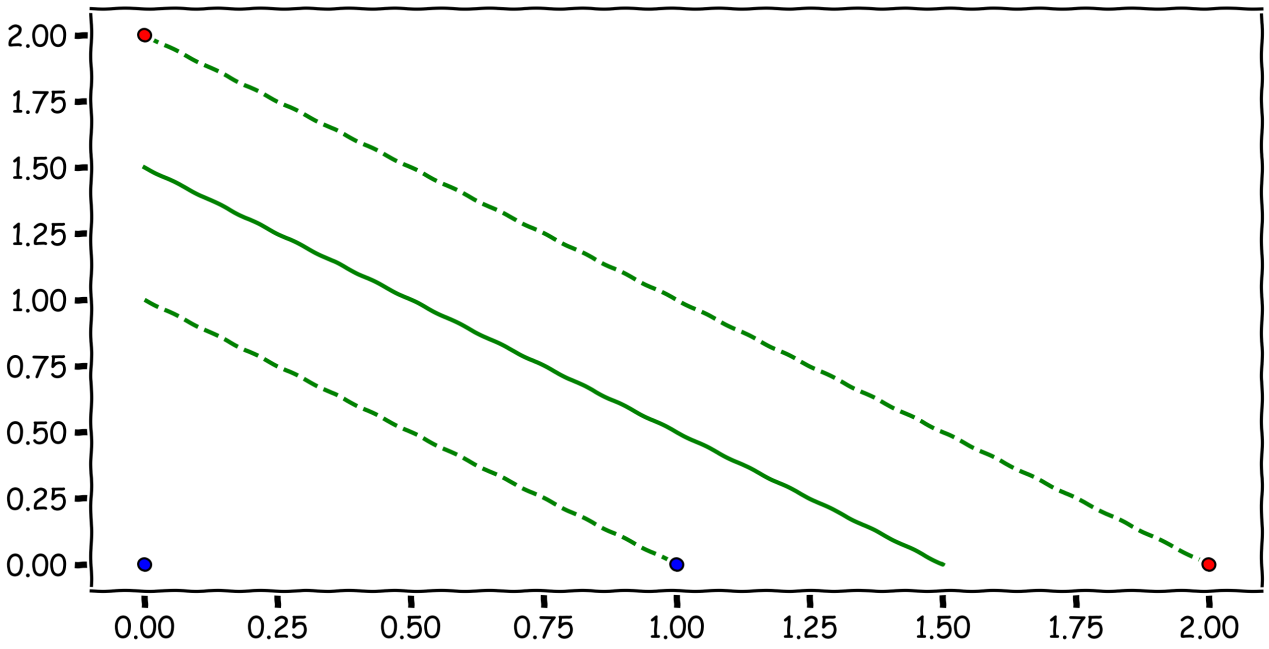


Figure 10: The optimal separating hyperplane.

look at fig. 12. Moreover, sometimes we can allow the errors on some training data for greater profit (for example, the width of the separating strip).

It turns out that the approach can be expanded. We can construct an optimal almost separating hyperplane (or a soft margin hyperplane). This hyperplane will support almost linear separation despite the classification errors on the training data (since the data is not linearly separable). Clearly, we will not solve all the problems because sometimes there's no way to separate the data linearly (see fig. 13). We'll talk about this a little bit later. Now our goal is to construct a classifier based on an almost separating hyperplane, although not ideally separating the data, but having the following advantages:

1. High resistance to specific observations, which means to outliers too.

2. Better classification of the majority of training observations.

3. Easy interpretation of the separation rule (of the classifier).

Hence, the classifier will make mistakes on training data, but not too many of them. At the same time, it should better classify the remaining observations.

Let's introduce an additional set of variables $\xi_i \geq 0$, $i \in \{1, 2, ..., n\}$, each characterizing the degree of error on the training data item x_i. We weaken the constraints of the problem set earlier and come up with the following optimization

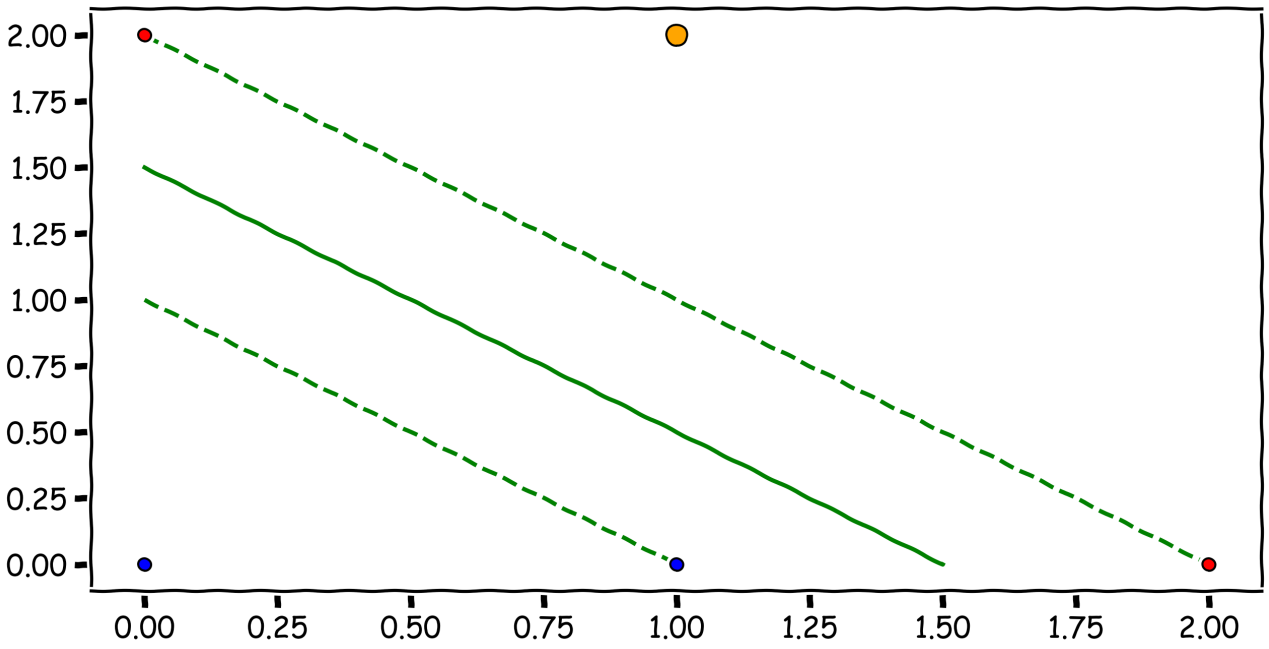


Figure 11: Classification of a test object.

problem:

$$\begin{cases} \frac{1}{2}(w,w) + C\sum\limits_{i=1}^{n}\xi_i \to \min\limits_{w,w_o,\xi} \\ y_i((w,x_i) - w_0) \geq 1 - \xi_i, \ i \in \{1,2,...,n\} \\ \xi_i \geq 0, \ i \in \{1,2,...,n\} \end{cases},$$

where C is a positive parameter.

Consider the second condition to understand what it means. It formally describes the position of each training data item x_i with respect to the almost separating hyperplane. Well, for the training data item x_i, 3 cases are possible:

1. $\xi_i = 0$, that is, $y_i((w,x_i) - w_0) \geq 1$. In this case, the classifier correctly classifies the observation x_i, and it lies outside the separating strip. If the inequality converts into equality, that is, if $y_i((w,x_i) - w_0) = 1$, the object lies on the boundary of the separating strip.

2. $0 < \xi_i \leq 1$, that is, $0 \leq y_i((w,x_i) - w_0) < 1$. In this case, the observation x_i is also correctly classified. However, it is either inside the separating strip if the left inequality is strict, or it lies on the separating hyperplane if $y_i((w,x_i) - w_0) = 0$

3. $\xi_i > 1$, that is, $y_i((w,x_i) - w_0) < 0$. In this case, the observation x_i is incorrectly classified.

The first condition of the optimization problem aims to maximize the width of the separating strip (by minimizing (w,w)), as well as the sum of the errors ξ_i with

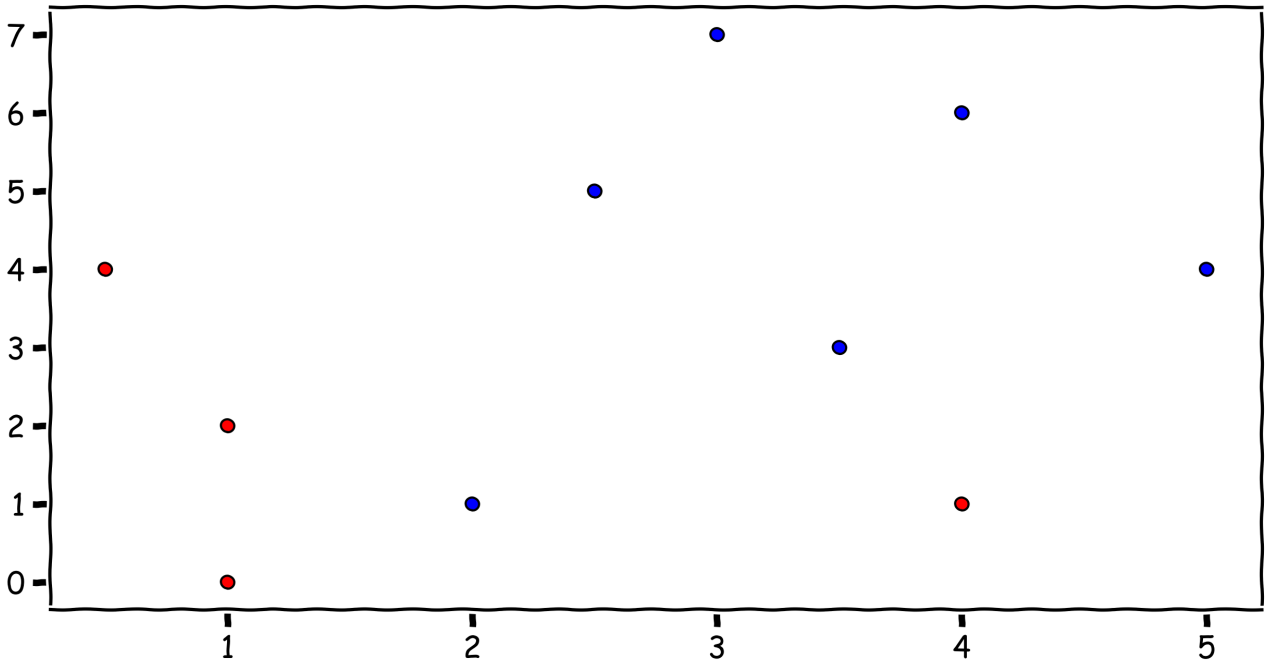


Figure 12: An example of the linearly inseparable data.

some positive weight C. The parameter C that is a controlling parameter of the model is put in charge of a researcher. It keeps the balance between maximization of the separating strip width and the total error of the training data classification.

As before, let's introduce the definition.

Definition 2.1.1 *The hyperplane l, which equation has the form:*

$$l: \quad \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_p X_p = 0,$$

where the coefficients $\theta_0, \theta_1, ..., \theta_p$ are derived from the optimization problem, is called a soft margin hyperplane, or almost separating hyperplane.

As before, the problem boils down to finding a saddle point of the Lagrange function. In this case, the Lagrange function is rewritten as follows:

$$L(w, w_0, \xi, \lambda, \eta) = \frac{1}{2}(w, w) - \sum_{i=1}^{n} \lambda_i (y_i((w, x_i) - w_0) - 1) - \sum_{i=1}^{n} \xi_i (\lambda_i + \eta_i - C),$$

where η_i, is a so-called dual variable for ξ_i. Using the Kuhn – Tucker conditions, we get the following problem:

$$\begin{cases} L(w, w_0, \xi, \lambda, \eta) \to \min\limits_{w, w_0, \xi} \max\limits_{\lambda, \eta} \\ \xi_i, \eta_i, \lambda_i \geq 0, \ i \in \{1, 2, ..., n\} \\ \lambda_i = 0 \text{ or } y_i(w, x_i) - w_0) = 1 - \xi_i, \ i \in \{1, 2, ..., n\} \\ \eta_i = 0 \text{ or } \xi_i = 0, \ i \in \{1, 2, ..., n\} \end{cases}.$$

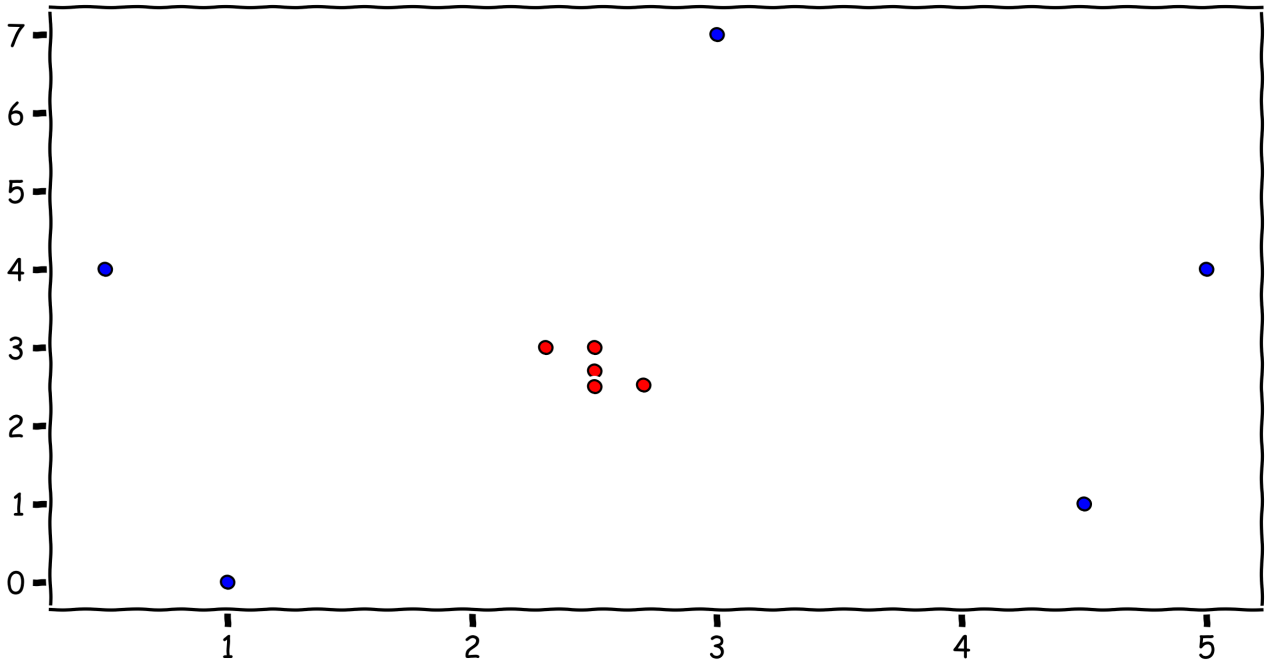


Figure 13: An example of linearly inseparable data.

The next steps are almost the same as the steps taken earlier. A necessary condition for a saddle point is the equality of the partial derivatives to zero. Thus,

$$\frac{\partial}{\partial w}L(w, w_0, \xi, \lambda, \eta) = w - \sum_{i=1}^{n} \lambda_i y_i x_i = 0,$$

$$\frac{\partial}{\partial w_0}L(w, w_0, \xi, \lambda, \eta) = \sum_{i=1}^{n} \lambda_i y_i = 0,$$

$$\frac{\partial}{\partial \xi_i}L(w, w_0, \xi, \lambda, \eta) = -\lambda_i - \eta_i + C = 0.$$

From these equations, we obtain important relations. Some of them (the first two), we already know:

$$w = \sum_{i=1}^{n} \lambda_i y_i x_i,$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0,$$

$$\lambda_i + \eta_i = C,\ i \in \{1, 2, ..., n\}.$$

After plugging it into the Lagrange function, we get the problem of finding the

minimum of the following form:

$$\begin{cases} -L(\lambda) = -\sum\limits_{i=1}^{n} \lambda_i + \frac{1}{2} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} \lambda_i \lambda_j y_i y_j (x_i, x_j) \to \min\limits_{\lambda} \\ 0 \le \lambda_i \le C, \ i \in \{1, 2, ..., n\} \\ \sum\limits_{i=1}^{n} \lambda_i y_i = 0 \end{cases}.$$

The previous problem differs from the current one in the second line with the additional constraint that $\lambda_i \le C$ for $i \in \{1, 2, ..., n\}$. Due to the reasons discussed before, the problem has one and only solution.

When the problem is solved (that is, when the parameters λ_i are found), the next step is to build a classifier. To do so, first, we need to understand what objects are classified and how they are classified, then introduce the necessary terms. Without going into details, let's note the following facts:

1. If $\lambda_i = 0$, then as before, the object x_i is correctly classified and sufficiently distanced from the almost separating hyperplane (that is, outside the separating strip).

2. If $0 < \lambda_i < C$, then the object x_i lies on the boundary of the separating strip (which means it is a support vector).

3. If $\lambda_i = C$, then we cannot make conclusions about the object x_i. The object x_i is either correctly classified, but lies inside the separating strip, or it lies in the almost separating hyperplane, or it is incorrectly classified.

The following definition is quite common.

Definition 2.1.2 *If $\lambda_i = C$, the object x_i is called an intruder.*

An intruder is either an object classified incorrectly or an object inside the almost separating strip.

Well, and how to build a classifier? We can use the following algorithm that is similar to that in case of a linearly separable sample.

1. Define w from the relation

$$w = \sum_{i=1}^{n} \lambda_i y_i x_i.$$

2. Find w_0. To do so, we can take a support vector x_i and use $(w, x_i) - w_0 = y_i$ to find

$$w_0 = (w, x_i) - y_i.$$

3. The classifier is given by the analytical expression

$$f(X) = (w, X) - w_0.$$

The next classification steps comply with the general rules given in 1.3.

Definition 2.1.3 *The classifier constructed this way is often called a soft margin classifier.*

Well, let's construct an almost separating hyperplane for the data given that the parameter $C = 3$. Fig. 14 shows that two blues and one red are support vectors, and classification makes errors on the data of both classes. One red is misplaced (just like one blue). The almost separating hyperplane is defined by

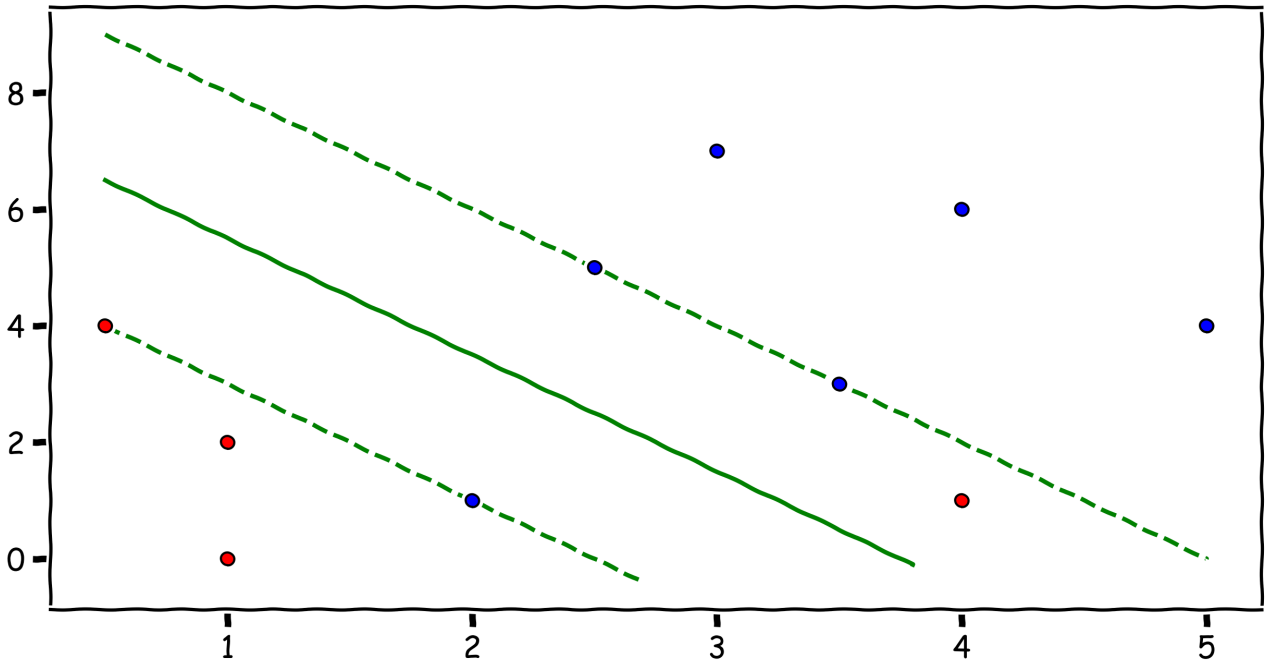


Figure 14: Soft margin classification.

the equation

$$-0.8X_1 - 0.4X_2 + 3 = 0,$$

and the classifier is defined by the expression

$$f(X) = f(X_1, X_2) = -0.8X_1 - 0.4X_2 + 3.$$

Given that $C = 1$, the situation changed somehow. It is represented in fig. 15 by the orange lines. The separating hyperplane became wider, which made the classification of the red training data less confident. One of the reds that was a support vector was correctly classified but became an intruder. The upper boundaries of both almost separating strips have merged in the figure because they are the same.

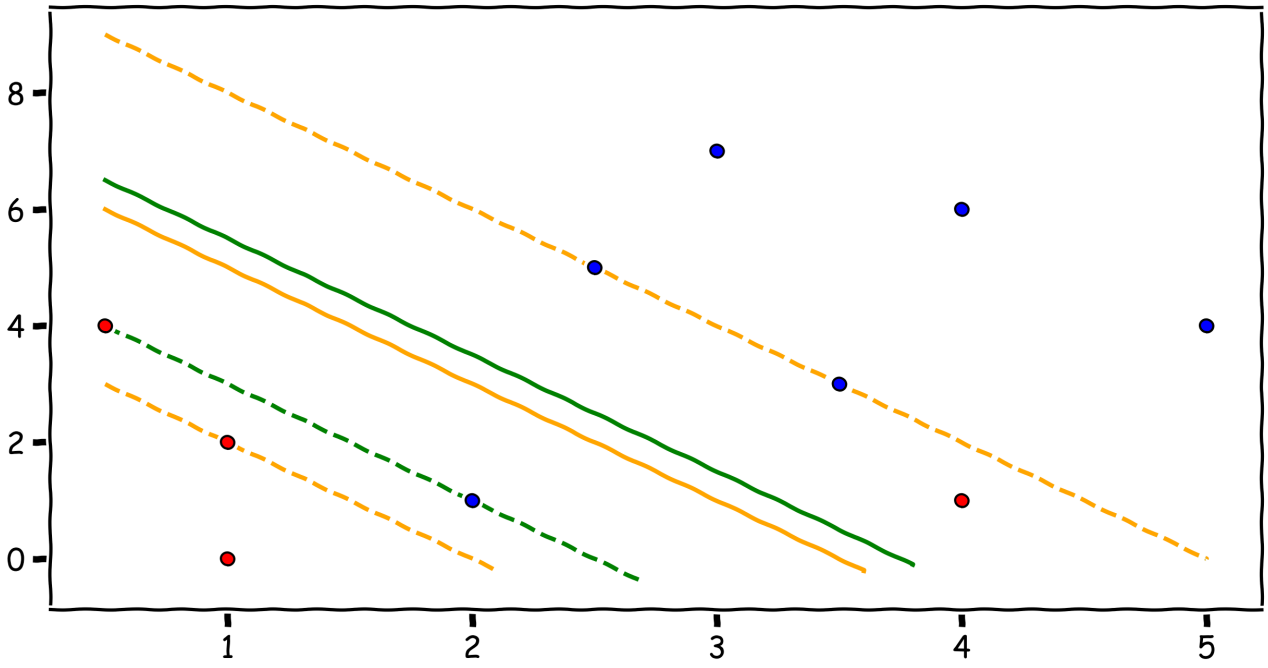


Figure 15: Soft margin classification.

Remark 2.1.1 *Using the classifier, we can separate the correctly classified intruders from those incorrectly classified. To do so, we can calculate* $f(x_i)$ *for the training objects* x_i*, for which* $\lambda_i = C$*, and compare the sign of the obtained value and the sign of the class. If the signs match, the classification is performed correctly. However, the object is inside the almost separating strip. Thus, the classification is not confident. If the signs differ, the classification was performed improperly. If* $f(x_i) = 0$*, the object lies in the almost separating hyperplane, and its class cannot be unambiguously defined.*

Perhaps, some of you think that it doesn't make sense to separate everything by a hyperplane. And those who think that are right. Let's dig into it.

2.2 Kernels and Space Transformations: General Theory

In conclusion, we would like to consider one more approach to linearly inseparable samples when even an almost separating hyperplane is no help. For example, it isn't right to split the dataset shown in fig. 16 by drawing a straight line.

The idea is simple but technically sophisticated. It's proposed to transform a feature space into a space of a higher dimension, where the data will be linearly separable. If ψ is such a transformation, the object $\psi(x_i)$ will correspond to the object x_i in a new space. The subsequent construction of the classifier can be performed according to the described scheme. The only difference is that the dot

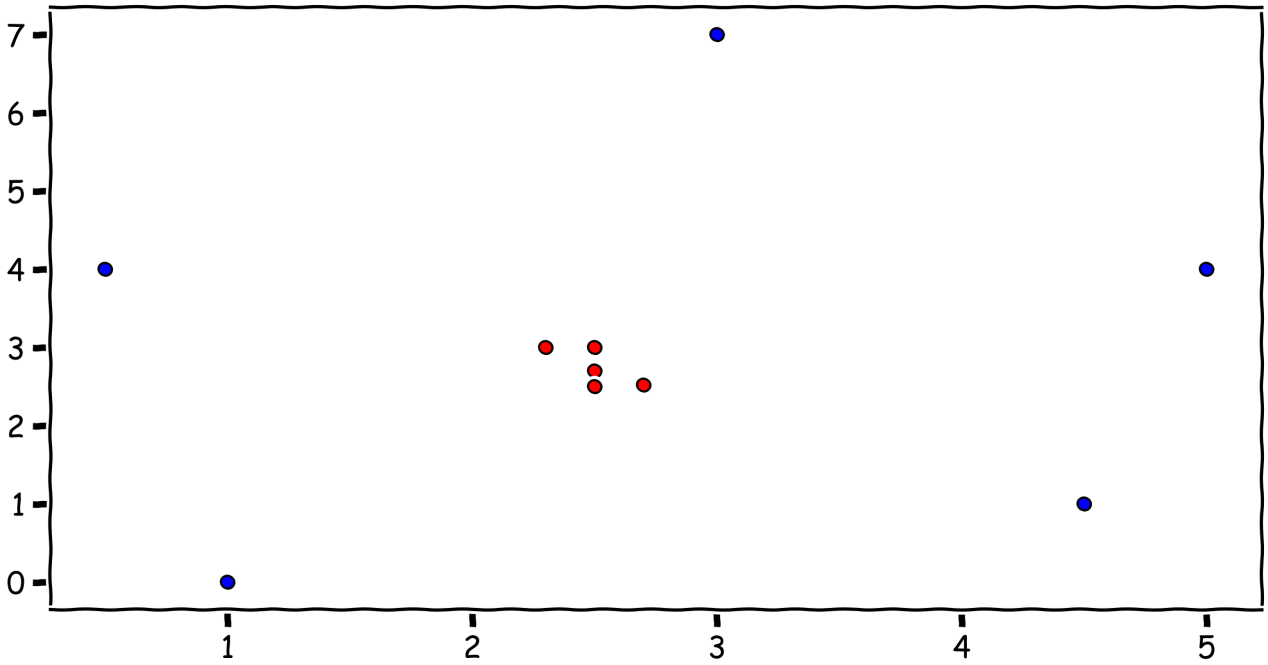


Figure 16: An example of linearly inseparable data.

product of (x_i, x_j) will change to some other dot product of the elements $\psi(x_i)$ and $\psi(x_j)$ in a space of a higher dimension. The transformation is performed using the function $\psi : \mathbb{R}^p \to H$, where H is a space with the dot product $(\cdot, \cdot)_H$. Here's the question. Where do we get ψ, and what is that space H? Let's begin with some easy definitions.

Definition 2.2.1 *A space H, where an initial dataset is linearly separable, is called rectifying.*

Remark 2.2.1 *As in the previous section, we can require almost linear separability instead of linear separability.*

It won't hurt to review once more the optimization problems we are dealing with. After that, it becomes clear that the formulations of these problems depend only on dot products of objects instead of their descriptions. This leads us to the concept of a kernel.

Definition 2.2.2 *Assume that $\psi : \mathbb{R}^p \to H$, where H is a space with the dot product $(\cdot, \cdot)_H$, is a function that maps the object $x_i \in \mathbb{R}^p$ into a space of a higher dimension. A function $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is called a kernel if it has the following form:*

$$K(x, x') = (\psi(x), \psi(x'))_H.$$

The definition makes it clear that the optimization problem to be solved using kernels in case of a linearly separable sample in H can be rewritten as follows:

$$\begin{cases} -L(\lambda) = -\sum\limits_{i=1}^{n} \lambda_i + \frac{1}{2} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} \lambda_i \lambda_j y_i y_j K(x_i, x_j) \to \min\limits_{\lambda} \\ \lambda_i \geq 0, \quad i \in \{1, 2, ..., n\} \\ \sum\limits_{i=1}^{n} \lambda_i y_i = 0 \end{cases}$$

In case of a linearly inseparable sample, it can be rewritten as:

$$\begin{cases} -L(\lambda) = -\sum\limits_{i=1}^{n} \lambda_i + \frac{1}{2} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} \lambda_i \lambda_j y_i y_j K(x_i, x_j) \to \min\limits_{\lambda} \\ 0 \leq \lambda_i \leq C, \ i \in \{1, 2, ..., n\} \\ \sum\limits_{i=1}^{n} \lambda_i y_i = 0 \end{cases} .$$

As usual, to define w, we can calculate

$$w = \sum_{i=1}^{n} \lambda_i y_i \psi(x_i),$$

and to define w_0, we will calculate

$$w_0 = (w, \psi(x_i))_H - y_i$$

on the support vector $\psi(x_i)$. The classifier is defined as the relation:

$$f(X) = (w, \psi(X)) - w_0.$$

It's important to note that the kernels leave a trace in the initial space. They just change a separating curve. Take a look. Using the kernels, the classifier expression can be rewritten as follows:

$$f(X) = (w, \psi(X)) - w_0 = \left(\sum_{i=1}^{n} \lambda_i y_i \psi(x_i), \psi(X) \right) - w_0 = \sum_{i=1}^{n} \lambda_i y_i K(x_i, X) - w_0.$$

By equating the expression to zero, we obtain the defined geometric place of points in the space separating the input data. Well, it's time to look at the concrete examples of kernels.

2.3 A Concrete Example of Separation With Kernels

We are going to find out what kernels exist and how the separation works.

Assume $p = 2$, which means we are dealing with a two-dimensional feature space. Let's consider the function $K(x, y) = (x, y)^2$ to show that it is a kernel. Assume $x = (x_1, x_2)$, $y = (y_1, y_2)$, then

$$K(x, y) = (x, y)^2 = (x_1 y_1 + x_2 y_2)^2 = x_1^2 y_1^2 + 2x_1 y_1 x_2 y_2 + x_2^2 y_2^2.$$

Let's also consider the space $H = \mathbb{R}^3$ and the function $\psi : \mathbb{R}^2 \to H$ based on the rule:

$$\psi(x) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2), \quad x = (x_1, x_2).$$

When the vectors $\psi(x)$ and $\psi(y)$ (as well as the vectors in H) are multiplied, we obtain that

$$(\psi(x), \psi(y))_H = x_1^2 y_1^2 + 2x_1 y_1 x_2 y_2 + x_2^2 y_2^2,$$

that is,

$$K(x, y) = (\psi(x), \psi(y))_H.$$

Hence, the introduced function is a kernel. Let's see how it works and how it may help us with the data. The described transformation maps the dataset into

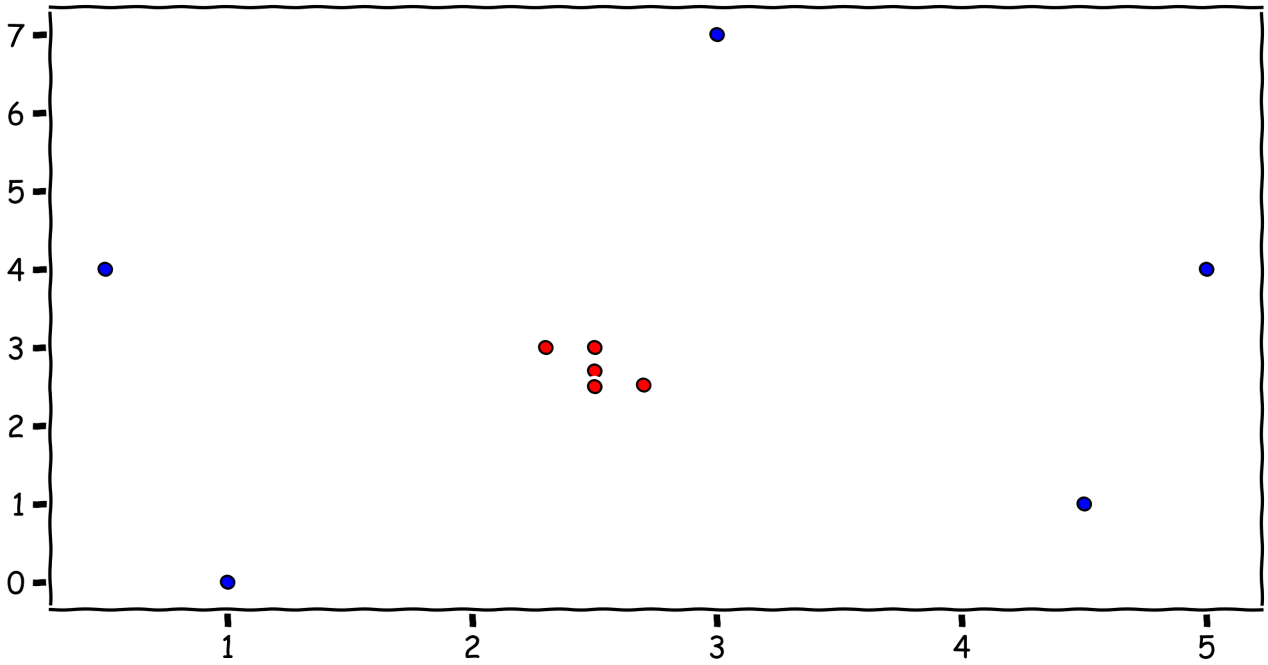


Figure 17: An example of linearly inseparable data.

a three-dimensional space. And we see that the data is now linearly separable. But what shape will separate the data in a plane? Take a look at the figure. As

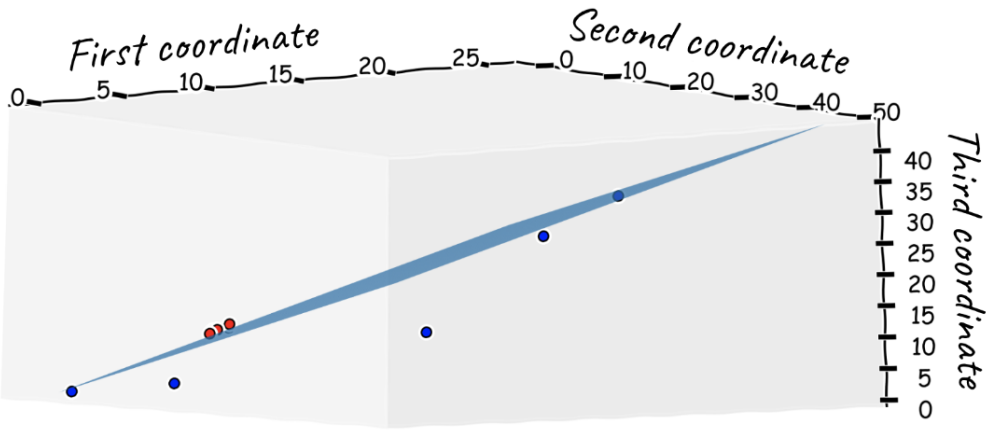
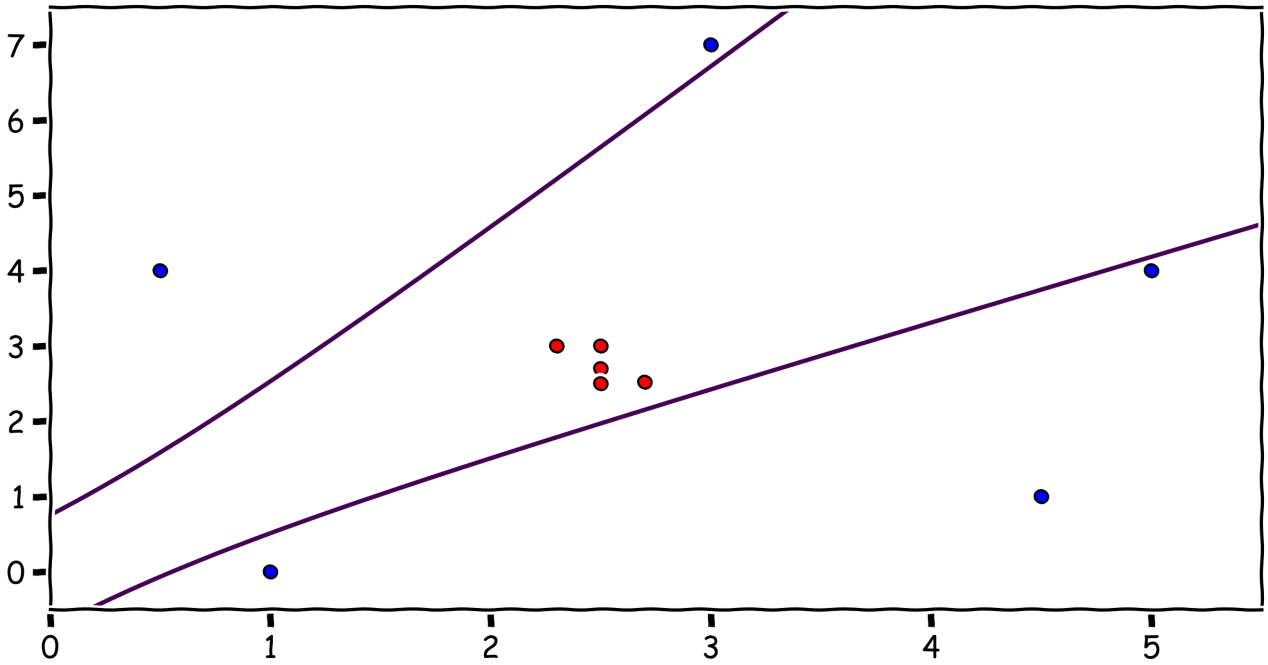


Figure 18: Separating the data in a space of a higher dimension.



you can see, the data is separated by a hyperbola. The geometric classification is also straightforward. Everything between the two branches of the hyperbola belongs to reds, the class $+1$, and everything inside the two branches belongs to blues, the class -1.

Not bad. However, one question remained unanswered. Where can we get these kernels and how to make them up?

2.4 More About Kernels and Their Properties

It's not easy and takes time to explain the theory of kernel creation, as well as to describe kernel properties and show how to differentiate the functions that can be kernels from those that cannot. Let's briefly review the well-known kernels available in almost all mathematical packages and libraries. To learn more, please

refer to additional materials.

Many standard kernels are related to the common algorithms, including polynomial separation, potential functions, two-layer neural networks, and so on. As a result, kernels look very promising. They describe (or substitute) a set of different known and used supervised learning techniques. Paradoxically, there's no general approach to their selection for concrete problems. Let's consider several standard kernels that are built-in many packages and libraries and describe their advantages.

1. We are going to generalize the given example. Let the feature space be $\mathbb{R}^p$. It is often convenient to consider the kernel

$$K(x, x') = (x, x')^d,\ d \in \mathbb{N}.$$

We can use it to separate what's inside the hyperellipsoid from what's outside.

2. Let the feature space be $\mathbb{R}^p$. It is often convenient to consider the kernel

$$K(x, x') = (1 + (x, x'))^d,\ d \in \mathbb{N}.$$

It's a so-called polynomial kernel. The separability of the data in the new space is equivalent to separating its polynomial separability in the initial space.

3. Let the feature space be $\mathbb{R}^p$. A so-called radial basis function (or RBF) kernel is often considered.

$$K(x, x') = e^{-\beta\|x-x'\|^2}, \beta > 0.$$

An RBF kernel exhibits a property of locality. If a test observation is far from a training observation, the value of the RBF kernel will be small. Thus, the kernel is sensitive only to the closest objects.

There are many different kernels, and some of them were developed for specific data types. It's an entire science. On the empirical level, we can do the following with available tools. First, it's worth trying linear separability or a linear kernel. If the training dataset is big, it makes sense to use an RBF kernel. Further investigations are put in charge of a researcher.

2.5 Application Examples

Let's see how to apply kernels using a well-known Fisher's iris dataset as an example. For visualization, it's better to use only the data with two predictors. Thus, we are going to keep only the first two attributes of sepal length and sepal width and only two types of flowers setosa and versicolor.

The data is shown in fig. 19. Note that first we've standardized the data. The figure shows that the classes are linearly separable. Using linear separation, we obtain fig. 20. We see 4 support vectors. They are outlined in black circles. What if we use non-linear separability? Let's find out. Fig. 21 shows the process

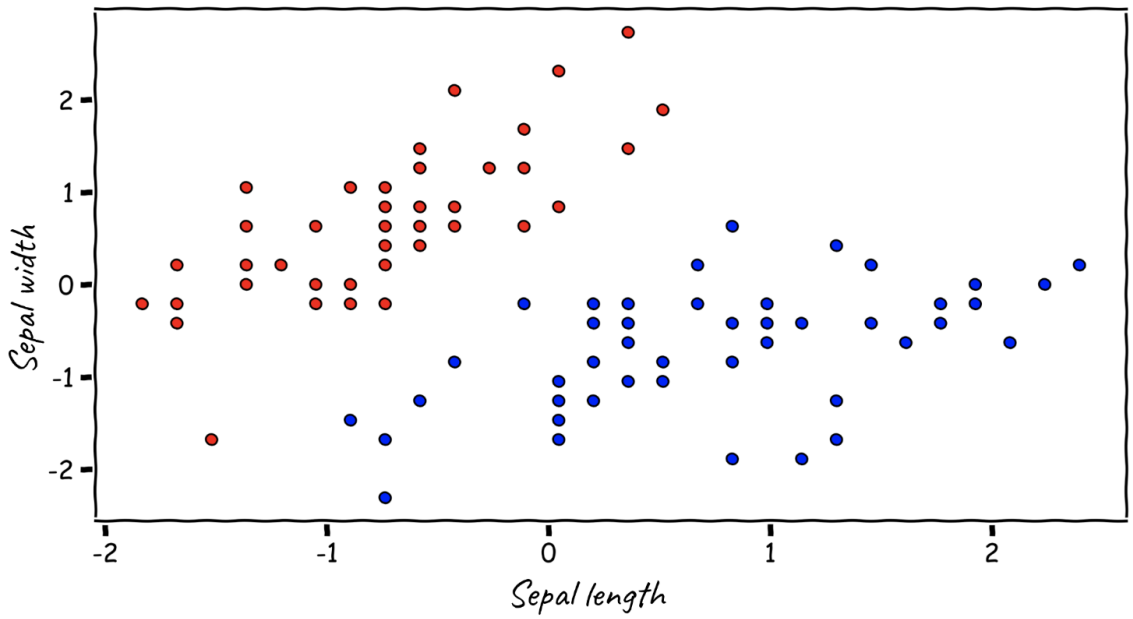


Figure 19: The first two attributes of flowers setosa and versicolor.

of separation using a third-degree polynomial kernel. You can see that the data remains separable. Support vectors have changed. They are also outlined. Fig. 26 shows what happens when an RBF kernel is used.

Now, let's take flowers versicolor and virginica and the same attributes. The data is no longer linearly separable as shown in fig. 23. It's even hard to imagine how to separate them. The classifier is also doing badly. The following three figures reflect the attempts to separate the data linearly (even with errors) using a third-degree polynomial kernel and RBF kernel. The intruders are outlined. As you can see, there's no modest separation.

3 Conclusion

This module covered one more classification technique. As you see, all the approaches are far from perfect. That's why an expert opinion and expert approach are most vital. Moreover, there's data that cannot be separated reasonably.

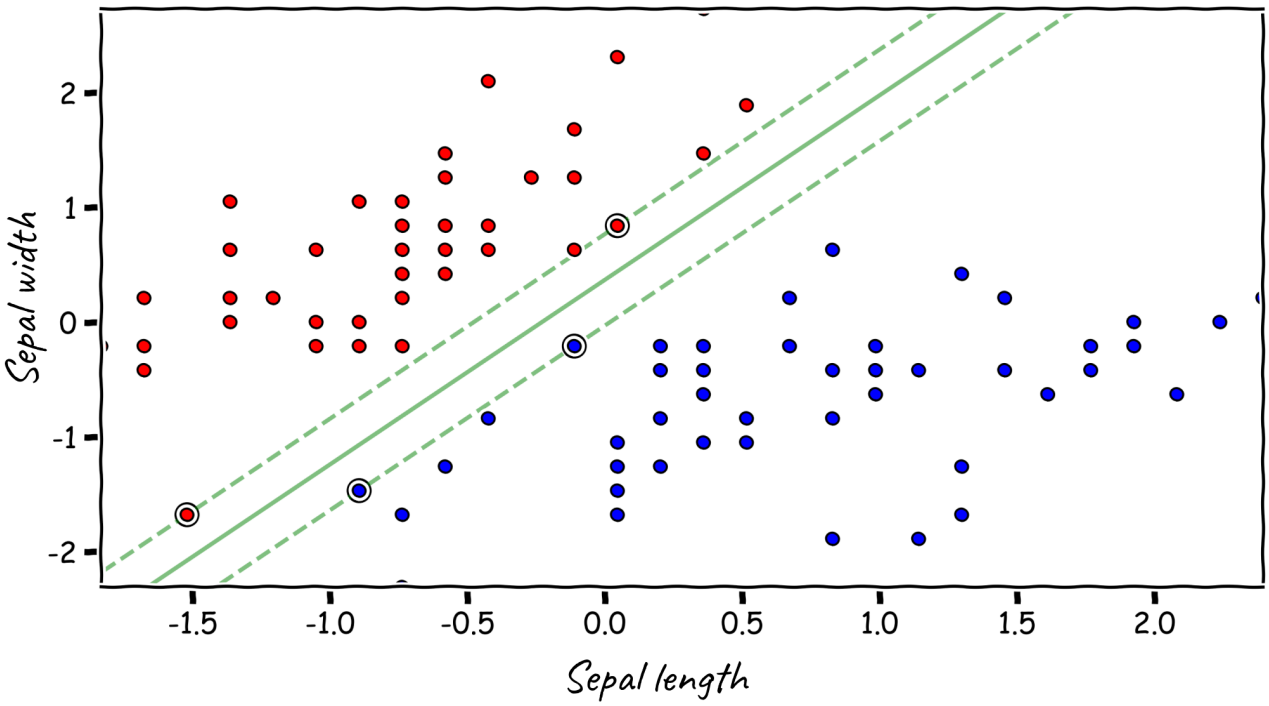


Figure 20: Linear separation.

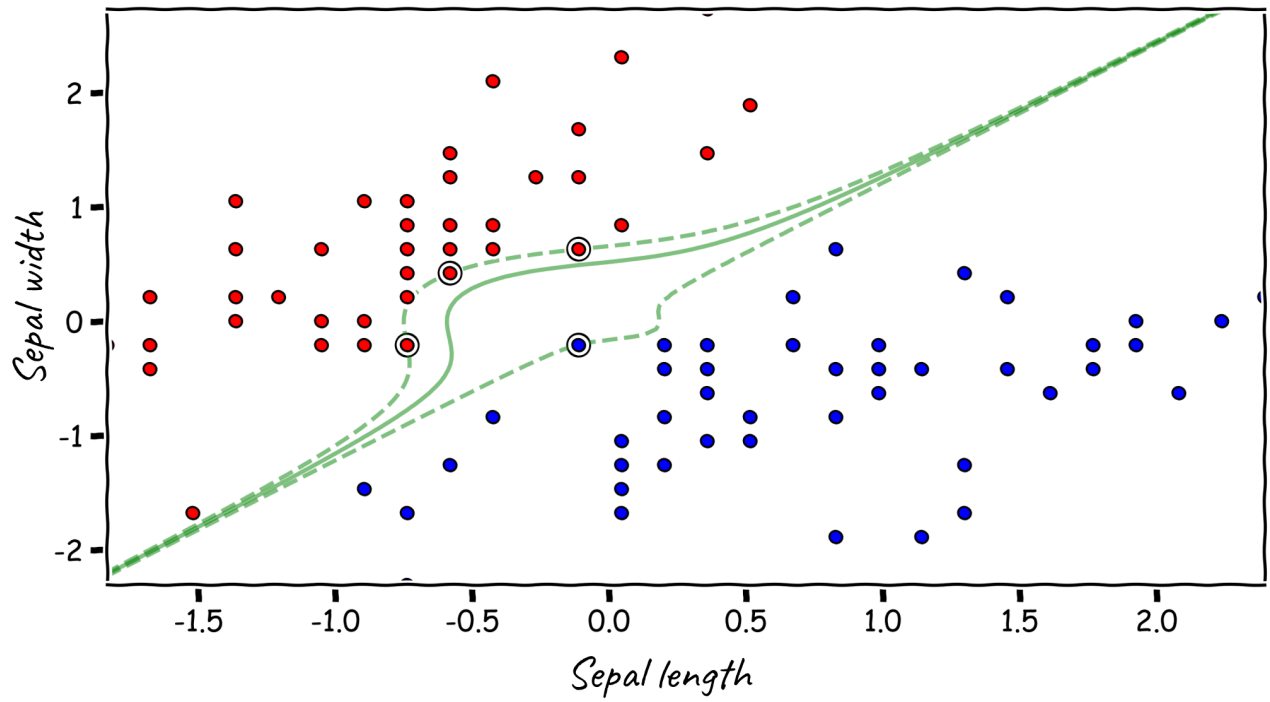


Figure 21: Separation with a third-degree polynomial kernel.

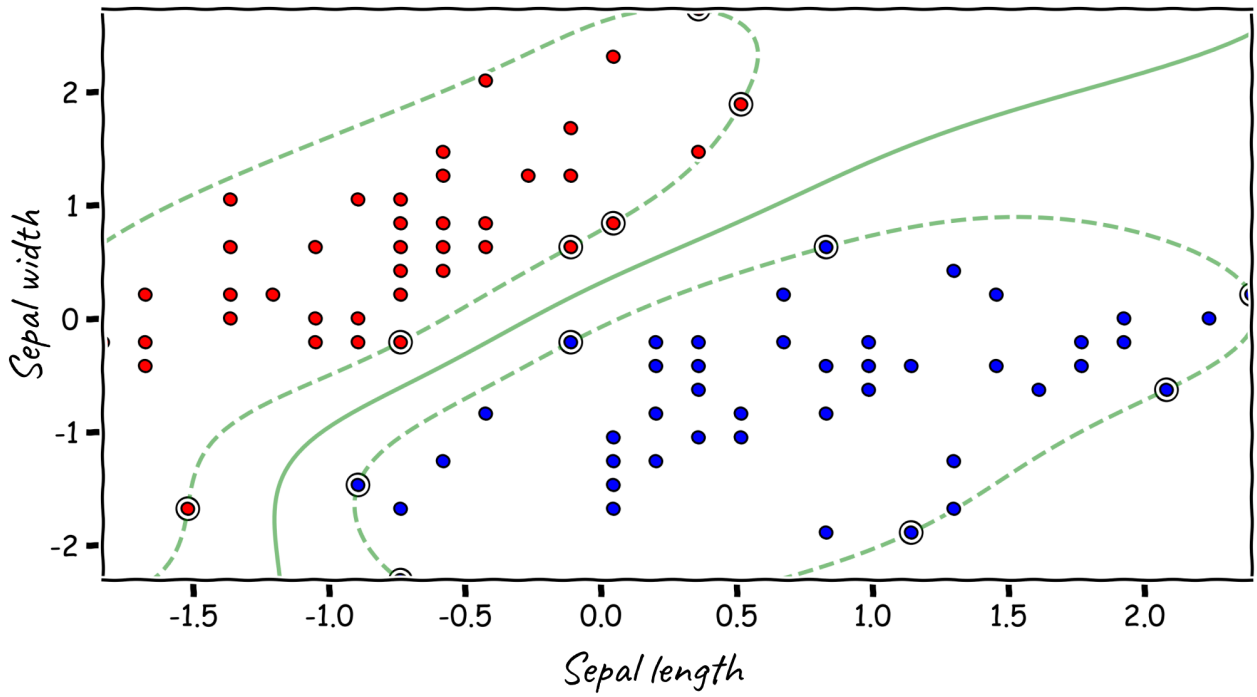


Figure 22: Separation with an RBF kernel.

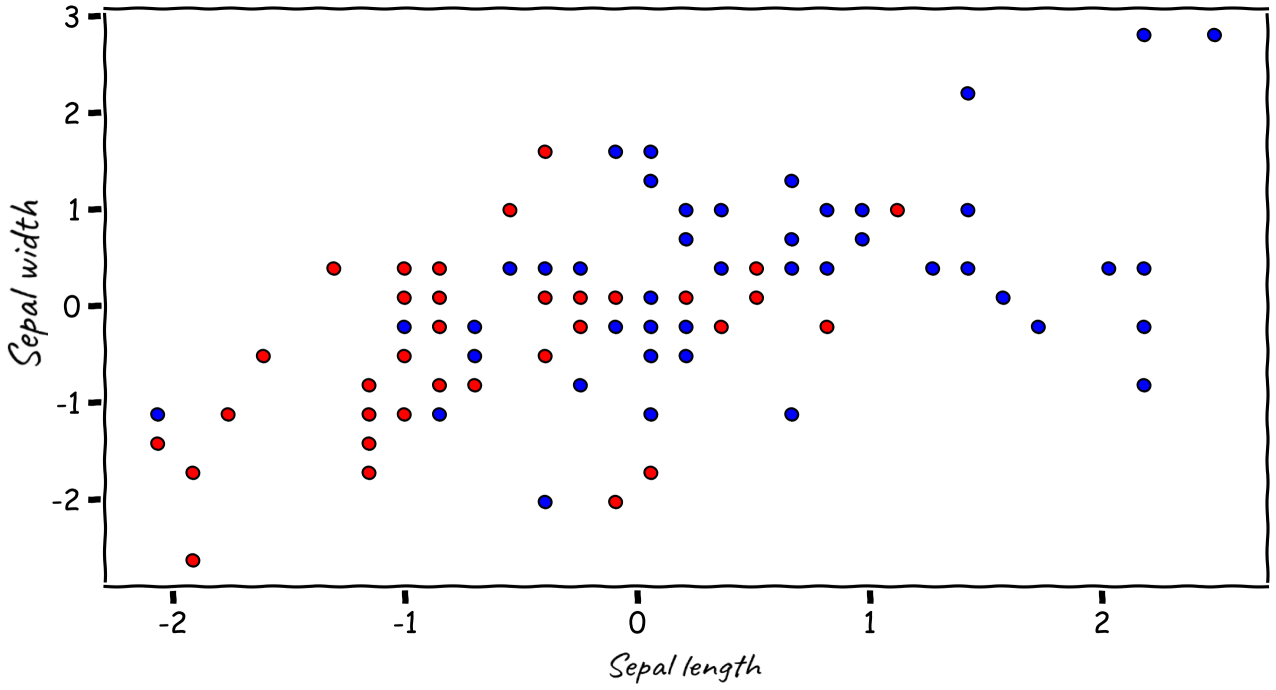


Figure 23: The first two attributes of flowers versicolor and virginica.

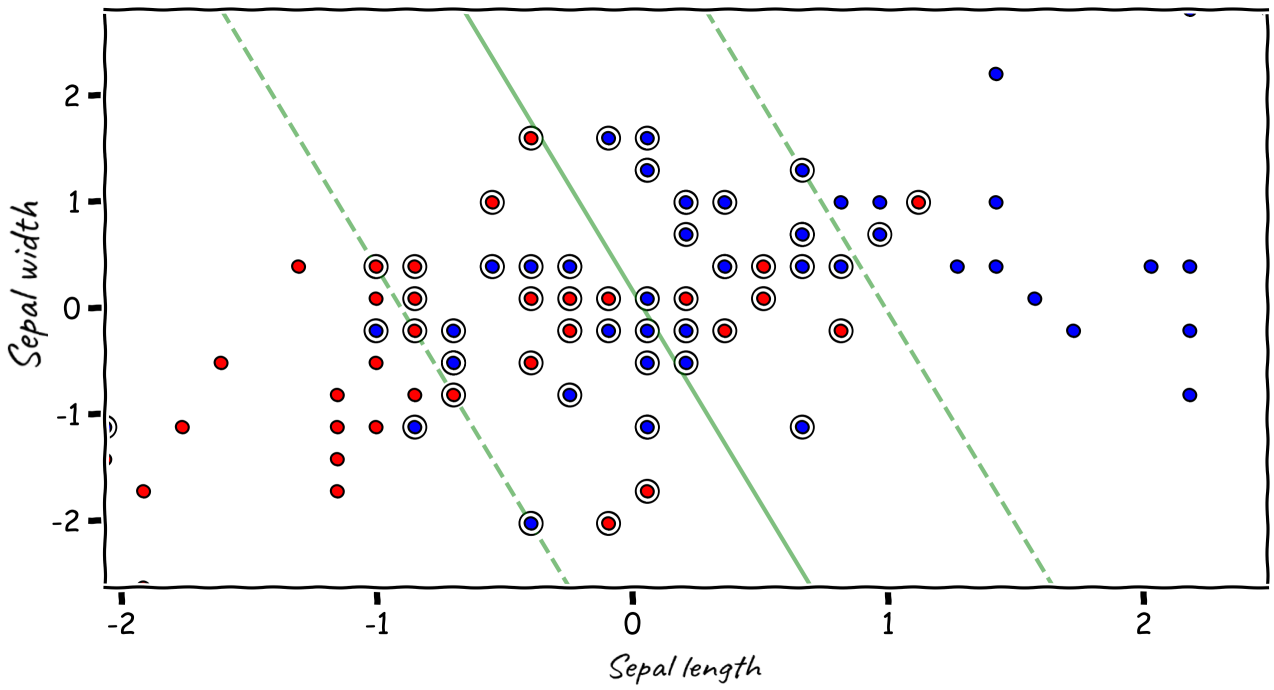


Figure 24: Separation by a hyperplane.

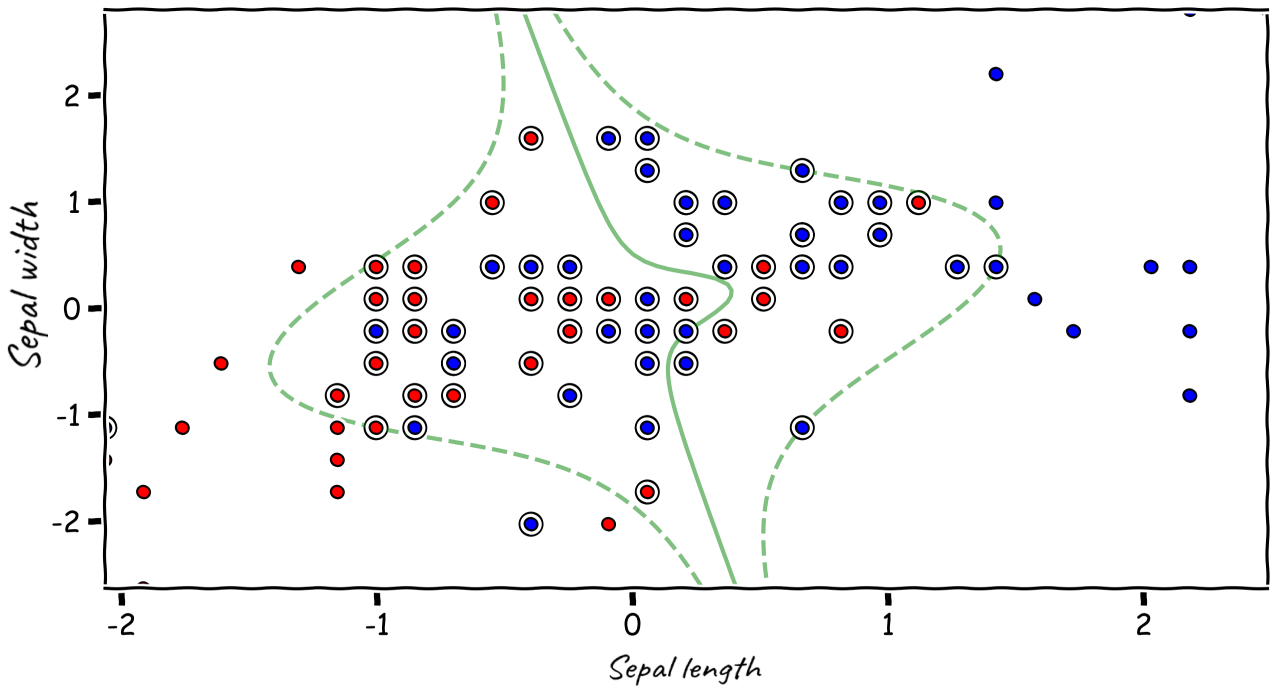


Figure 25: Separation with a third-degree polynomial kernel.

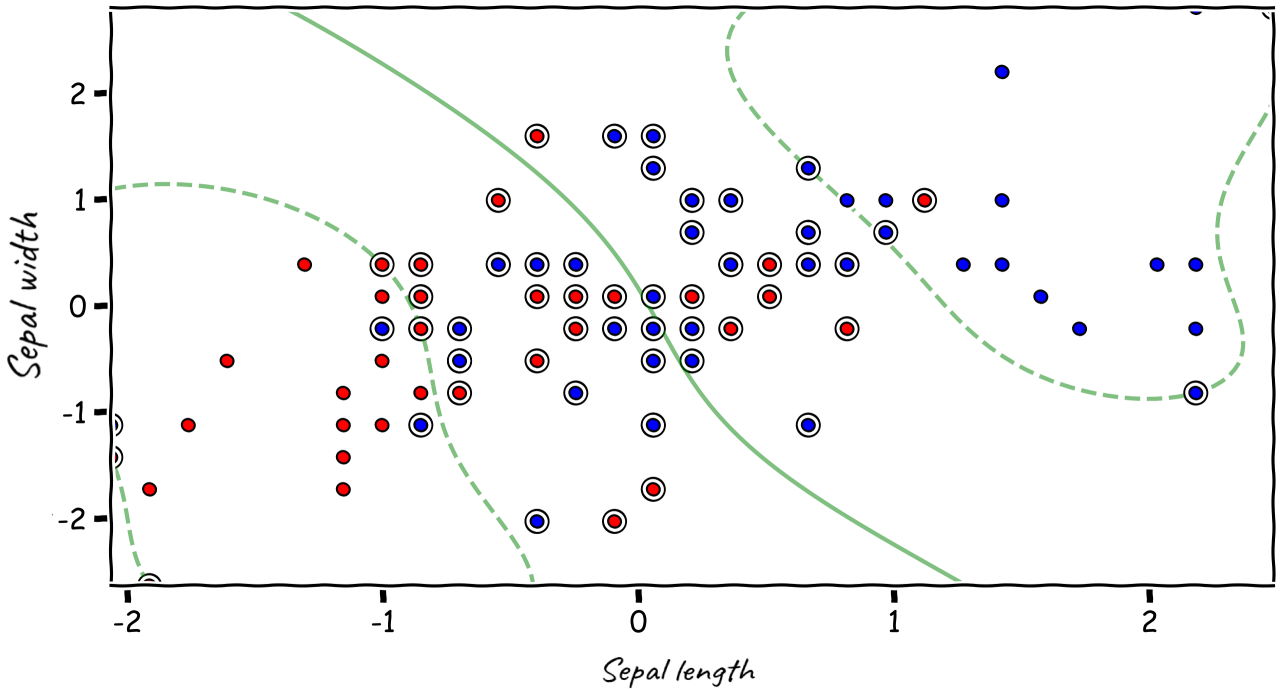


Figure 26: Separation with an RBF kernel.