Machine Learning

Support Vector Machine

Karol Przystalski

April 8, 2020

Department of Information Technologies, Jagiellonian University

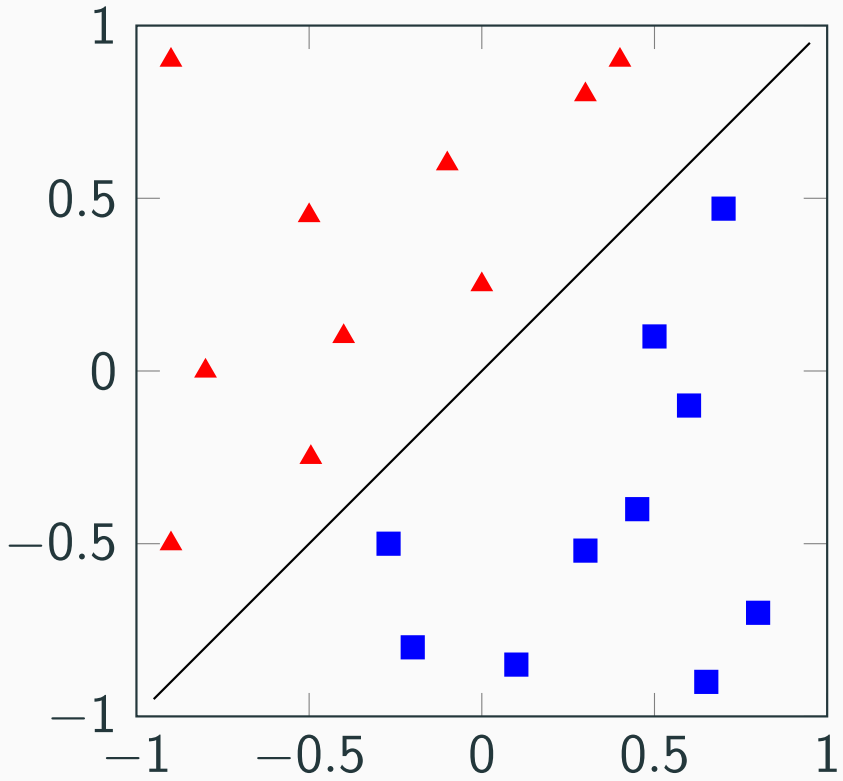
Agenda

1. Introduction
2. Lagrangian multipliers
3. Different types of SVM
4. Non-linear separation
5. Extensions

Introduction

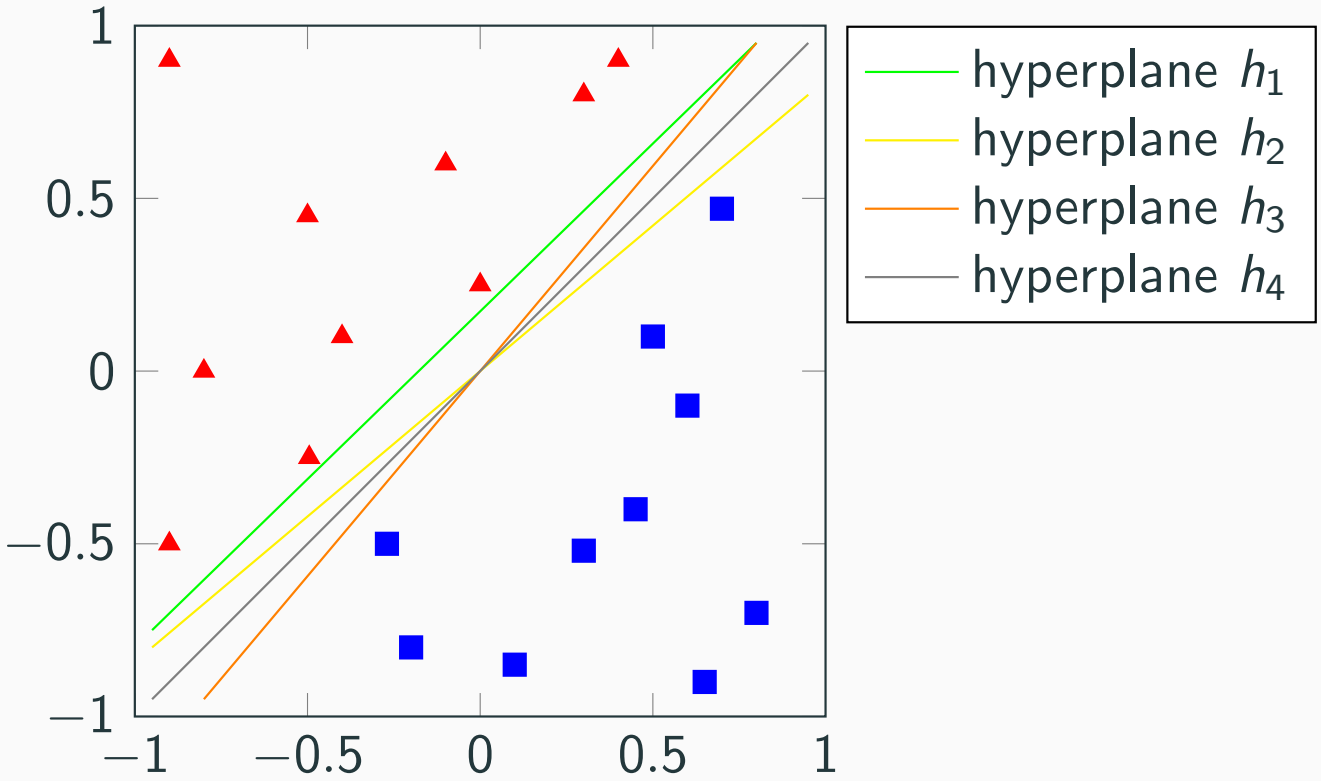
General overview

SVM is a binary classifier, so we consider two classes and to simplify the next examples, we use a two-dimensional feature space. For now let's assume that the problem is linearly separable as shown below.



SVM – separation hyperplanes

We have an infinite number of hyperplanes. A few possible separation options are shown below.



General overview

The line that distinguish both classes can be described by the following equation:

$$g(x) = w_0 + w_1 x_1 + w_2 x_2 = 0. \tag{1}$$

It means that if we find such a line then for all points in the feature space representing class 1 we have:

$$w_0 + w_1 x_1^1 + w_2 x_2^1 > 0 \tag{2}$$

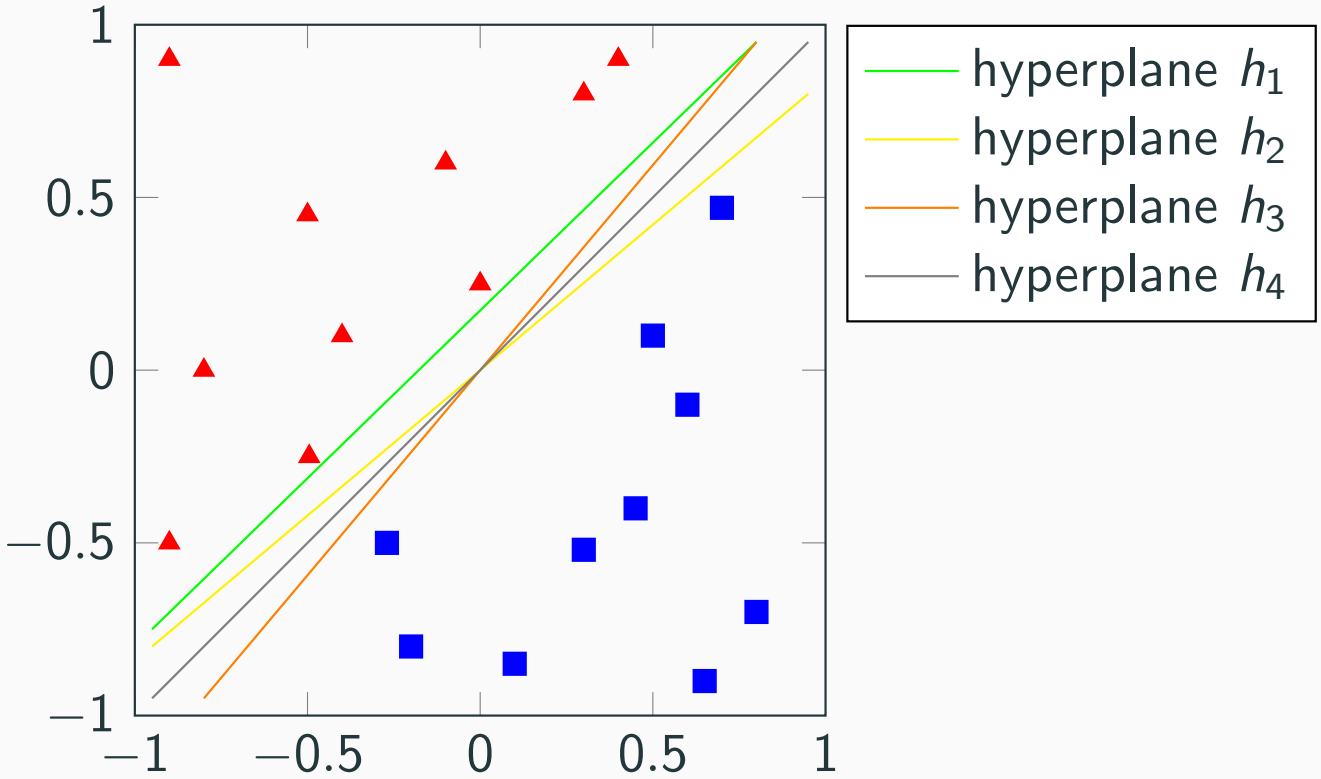
and

$$w_0 + w_1 x_1^{-1} + w_2 x_2^{-1} < 0 \tag{3}$$

for all points $x^{-1} = (x_1^{-1}, x_2^{-1})$ representing class -1.

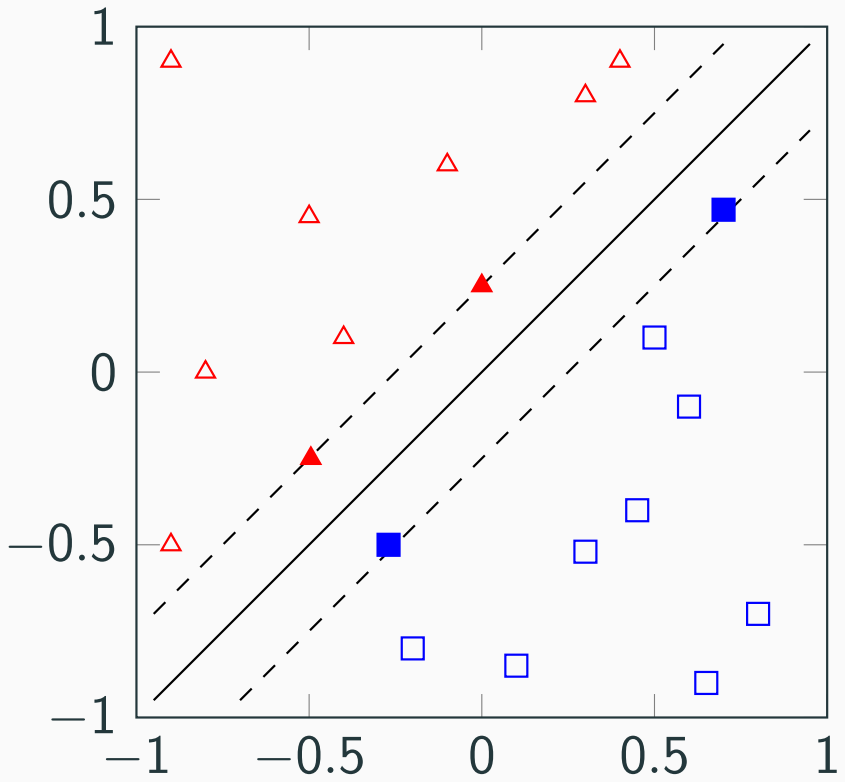
Maximum margin

Let's go back and consider which hyperplane is the best one.



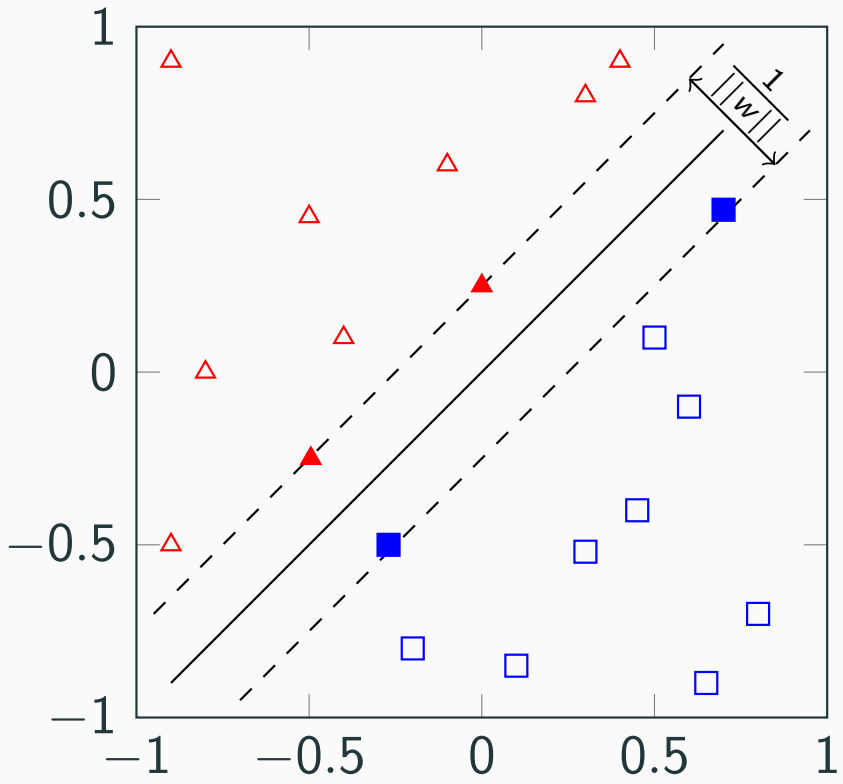
Maximum margin

Obviously, it's hyperplane h_4 as the margin from the red and blue objects is the biggest.



Maximum margin

The margin is set as $\frac{1}{||w||}$ and our goal is to minimize w.



Maximum margin – formulas

So far we know that have to maximize:

$$\frac{1}{||w||}, \tag{4}$$

what means that we need to:

$$\min \frac{1}{2} w^T w \tag{5}$$

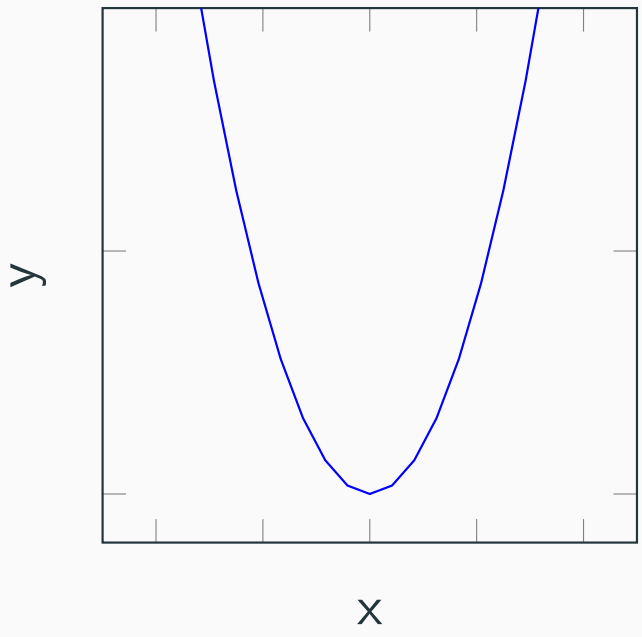
and take into consideration some constraints:

$$t_i(w^T x_i + w_0) \geq 1 \ \forall i = 1, \ldots, n. \tag{6}$$

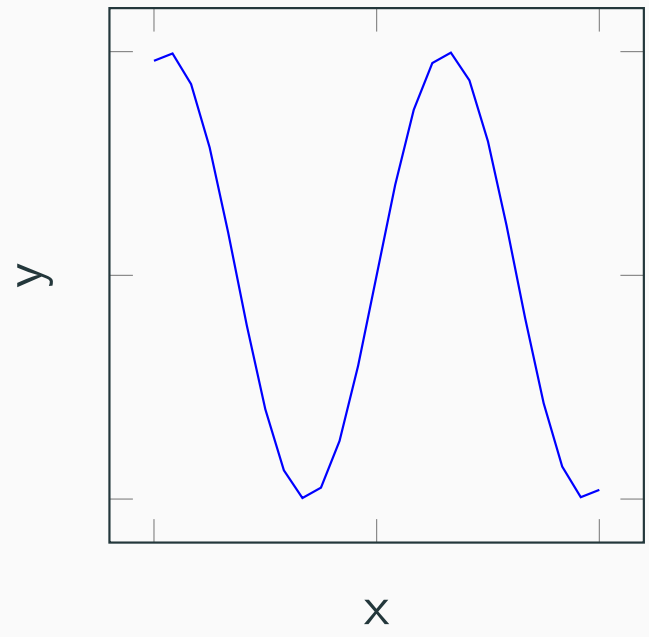
This problem is a convex problem.

Convex problem

A function is convex if from each point on the curve to another point on the curve, it does not intersect the curve anywhere else. In figure (a) we see a convex function, in figure (b) it's not.



(a)

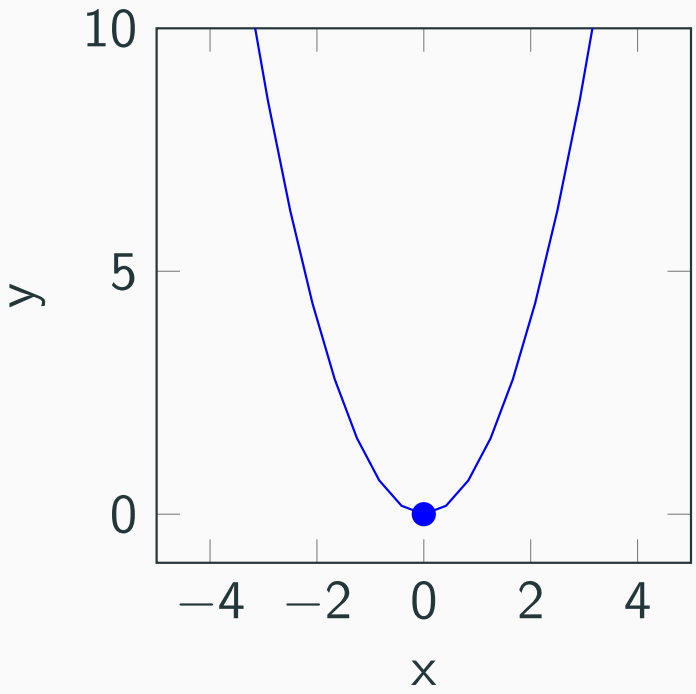


(b)

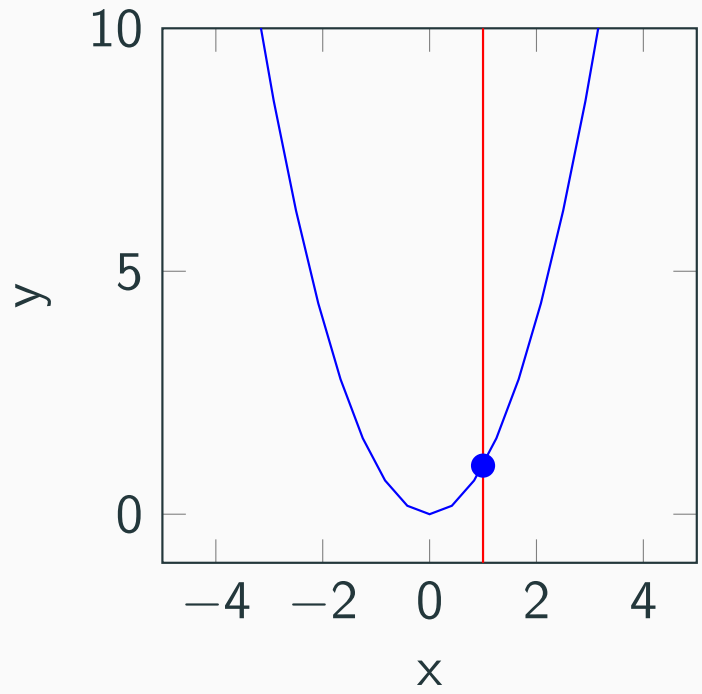
Lagrangian multipliers

Lagrangian multipliers

We have to solve a quadratic optimization problem with constraints to find an optimal separating hyperplane. What does it mean? Example of the minimization problem of x^2 function with a constraint function $x = 1$ (b) and without it (a).



(a)



(b)

Langrangean multipliers

This problem is very simple as it contain only a function with one variable. A function with multiple variables is much harder to solve. Let's assume we have to minimize the function $f(x)$ with the constraint $g(x) = 0$, where x might be a vector of variables $x = (x_1, \ldots, x_n)$. We can notice that the minimum of the $f(x)$ is found when the gradients of these two functions are parallel i.e.

$$\nabla f(x) = \alpha \nabla g(x), \tag{7}$$

where α is the scaling factor, we call it a Lagrange multiplier. To find the minimum of f under the constraint g, we just need to solve:

$$\nabla f(x) - \alpha \nabla g(x) = 0. \tag{8}$$

Langrangean – example

To solve that equation we can define a function

$$L(x, \alpha) = f(x) - \alpha g(x), \tag{9}$$

then its gradient is

$$\nabla L(x, \alpha) = \nabla f(x) - \alpha \nabla g(x). \tag{10}$$

Solving

$$\nabla L(x, \alpha) = 0 \tag{11}$$

allows us to find the minimum.

Langrangean – example

Let's take an example to understand how it's used to solve and find the minimum of the function

$$f(x, y) = x^2 + y^2 \tag{12}$$

under the constraint

$$g(x, y) = x + y - 1 = 0. \tag{13}$$

In our case the Lagrangian is defined as follows:

$$L(x, y, \alpha) = x^2 + y^2 - \alpha(x + y - 1). \tag{14}$$

Langrangean – example

Now, we have to calculate when the gradient of this function equals to zero, which means solving the following system of equations:

$$\frac{\partial}{\partial x} L(x, y, \alpha) = 2x - \alpha = 0 \tag{15}$$

$$\frac{\partial}{\partial y} L(x, y, \alpha) = 2y - \alpha = 0 \tag{16}$$

$$\frac{\partial}{\partial \alpha} L(x, y, \alpha) = -x - y + 1 = 0 \tag{17}$$

Langrangean – example

We calculate the derivative of each variable in each equation separately. Finally, we get the answers as:

$$x = \frac{1}{2},$$

$$y = \frac{1}{2},$$

and

$$\alpha = 1.$$

It mean that the function

$$f(x, y) = x^2 + y^2 \tag{18}$$

have the minimum in

$$f(\frac{1}{2}, \frac{1}{2}) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}.$$

Langrangean – multiple constraints

Lagrange multipliers work also with multiple constraints. We are just adding another boundary to the problem. When we deal with multiple constraints then our Lagrangian becomes:

$$L(x, \alpha) = f(x) - \sum_i \alpha_i g_i(x), \tag{19}$$

where $g_i(x) = 0$ for $i = 1, \ldots, n$ are the constraints. Notice that each constraint has its own Lagrange multiplier. The Lagrangian is equal to 0:

$$\nabla L(x, \alpha) = 0. \tag{20}$$

Solving this case is not much different compared to a single constraint case. However, when we are looking for optimal hyperplane then our constraints are inequalities. The constraints are handled by the Lagrange multipliers, but the following equations should be met when dealing the inequality constraints:

$$\begin{aligned} g(x) \geq 0, \ \textit{then } \alpha \geq 0 \\ g(x) \leq 0, \ \textit{then } \alpha \leq 0 \end{aligned} \tag{21}$$

Langrangean – multiple constraints

Let's take an example of a function with two constraints. The function is defined as

$$f(x, y) = 2x^2 - 3y^2 \tag{22}$$

and the constraints are defined as:

$$g_1(x, y) = x^2 - 4 \geq 0, \tag{23}$$

and

$$g_2(x, y) = y + 1 \geq 0. \tag{24}$$

Based on the previous example we can set our Langrangian as:

$$L(x, y, \alpha_1, \alpha_2) = 2x^2 - 3y^2 - \alpha_1(x^2 - 4) - \alpha_2(y - 1). \tag{25}$$

Langrangean – multiple constraints

All derivatives should be zero and we get a system of equations as:

$$\frac{\partial}{\partial x} L(x, y, \alpha_1, \alpha_2) = 4x - 2x\alpha_1 = 0, \tag{26}$$

$$\frac{\partial}{\partial y} L(x, y, \alpha_1, \alpha_2) = -6y - 2y\alpha_2 = 0, \tag{27}$$

$$\frac{\partial}{\partial \alpha_1} L(x, y, \alpha_1, \alpha_2) = x^2 - 4 = 0, \tag{28}$$

$$\frac{\partial}{\partial \alpha_2} L(x, y, \alpha_1, \alpha_2) = y + 1 = 0. \tag{29}$$

We have also additional constraints:

$$\alpha_1 \geq 0, \tag{30}$$

$$\alpha_2 \geq 0. \tag{31}$$

Langrangean – multiple constraints

Above equations give as the values of y, x, α_1 and α_2:

$$y = -1, \tag{32}$$

$$x = 2, \tag{33}$$

$$\alpha_1 = 2, \tag{34}$$

$$\alpha_2 = 3. \tag{35}$$

Finally, we have the minimum of the function $f(x, y)$ in $(-1, 2)$ equals to:

$$f(x, y) = 2 \cdot 2^2 - 3 \cdot (-1) = 11. \tag{36}$$

Multipliers and SVM

Let's go back to our problem of maximizing the margin between the hyperplanes. We concluded that it is enough to find the minimum for function:

$$f(w) = \frac{1}{2}||w||^2, \tag{37}$$

with the constraints:

$$g(w, w_0) = y_i(w^T x_i + w_0) - 1 \geq 0, \ i = 1, \ldots, n. \tag{38}$$

C-SVM – dual problem

There are different ways of solving the equations. The primal problem isn't easy to calculate, because we don't know much about α_i. It is easier if we transform the equations into a dual problem. Dual problem of C-SVM can be written as:

$$L(w, w_0, \alpha) = \sum \alpha_i - \frac{1}{2} \sum \alpha_i \alpha_j y_i y_j (x_i^T + x_j) \tag{39}$$

with constraints

$$C \geq \alpha_i \geq 0, \ i = 1, \ldots, n \quad \textit{and} \quad \sum \alpha_i y_i = 0. \tag{40}$$

It is be solved using quadratic programming, means a simpler approach.

Different types of SVM

C-SVM – soft margin

The presented solution is very intuitive and it performs well, but only if a separating hyperplane exists – the problem is linearly separable. In many cases no separating hyperplane exists and therefore the solution of the optimization problem has no solution with margin $\frac{1}{2}||w|| > 0$.

C-SVM – soft margin

The pure maximum margin classifier is very sensitive to single outlying observations. Every change of the support vector impacts the separating hyperplane. It can change drastically even if we add one outlying sample. It can even become unseparable. The problem can be solved by allowing to misclassify a few training samples in order to achieve a better accuracy in classifying the test samples.

C-SVM – soft margin

This approach is a soft margin approach. Rather than seeking the largest possible margin so that every observation is not only on the correct side of the hyperplane, but also on the correct side of the margin, we instead allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane. We call the margin soft because it can be violated by some of the training samples

C-SVM – formulas

In such case we can use C-SVM and it's defined as:

$$\frac{1}{2}||w||^2 + C\sum \xi_i \tag{41}$$

with constraints

$$\xi_i \geq 0 \quad \textit{and} \quad \sum \xi_i < C,\ C > 0. \tag{42}$$

The ξ_i are slack variables that allow some individual samples to be on the wrong side of the margin or the hyperplane. Ok, but what are these slack variables? Generally they tell us where the i-th samples is in the feature space relative to the hyperplane we are looking for. Even more it tells us where this sample is in relative to the margin.

C-SVM – slack variables

There are four possible cases:

- $\xi_i = 0$ the sample is on the correct side of the hyperplane and on the correct side of the margin,
- $\xi_i > 0$ and $\xi_i < 1$ the sample is on the correct side of the hyperplane, but it lies inside the margin,
- $\xi_i = 1$ the sample lies just on the separating hyperplane,
- $\xi_i > 1$ the sample lies on the wrong side of the hyperplane.

C-SVM – slack variables and C

The second case is is not causing a classification error when third and fourth lead to misclassification. We can differ the approach to them, but it is more practical not to.

Parameter C which is a sum of all ξ_i's so it will determine sum of the violations to the margin and to the hyperplane. It means that we will tolerate the total weight of the violations, but no more than C to find better hyperplane i.e. with better margin.

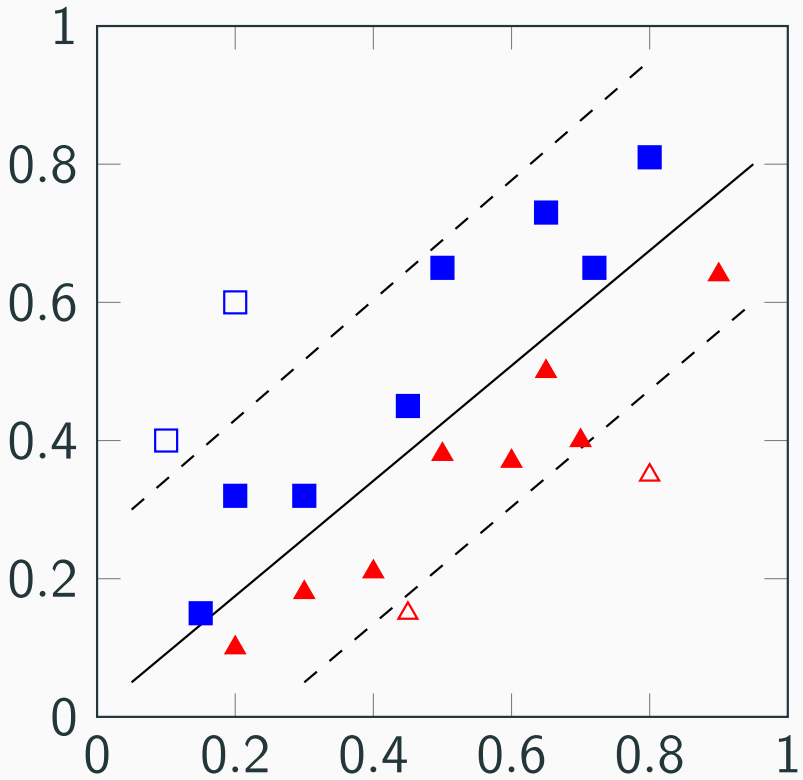
C-SVM – C

We can think of C as a trade-off between the width of the margin and the cost of all the violations cause by n samples. We can demand that $C = 0$ which means that we do not allow any sample from the training data set to be misclassified. However we risk that we find the solution with very small margin or even we cannot be able to separate our samples.

If we allow $C > 0$ the we allow some samples to lie in the margin or even to be misclassified which result with wider margin or even makes our problem to be separable. The bigger value of C then we get more general solution but the at the cost of increasing error rate. How to choose the value of C is not a trivial problem.

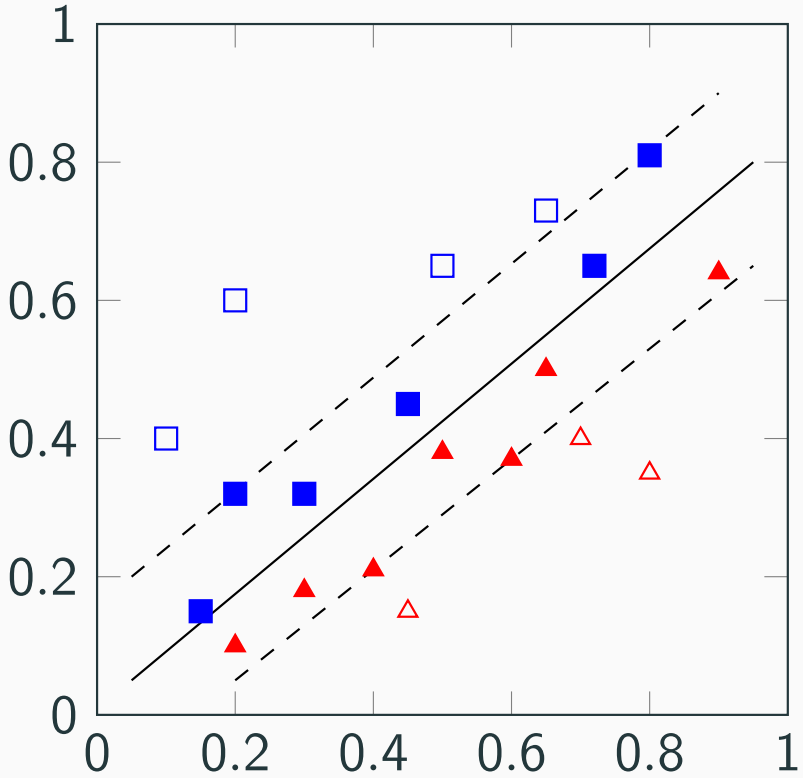
C-SVM

This how the margins looks with a low C value, like 0.5.



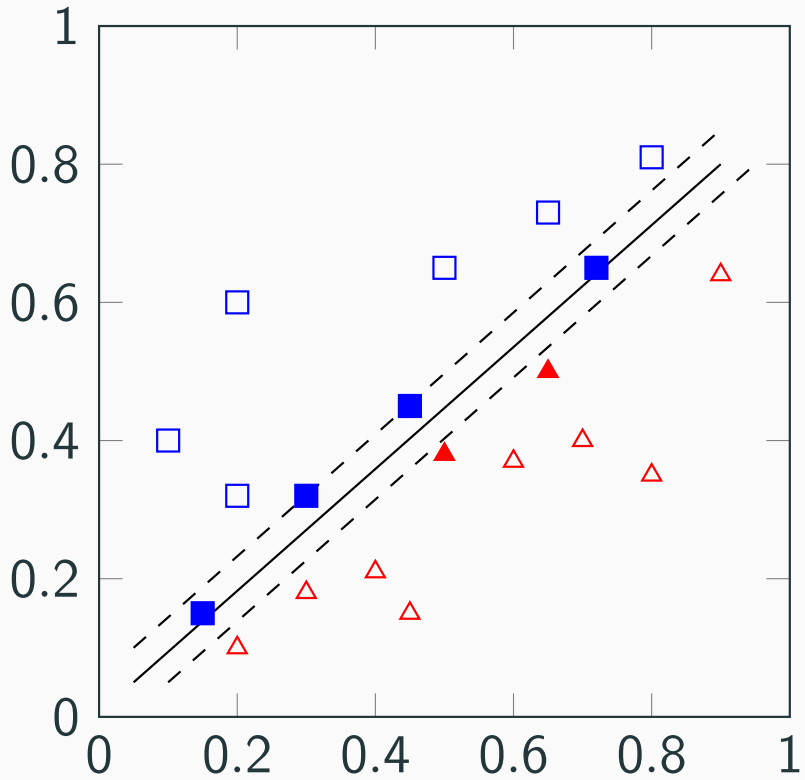
C-SVM

This how the margins looks with a medium C value, like 10.



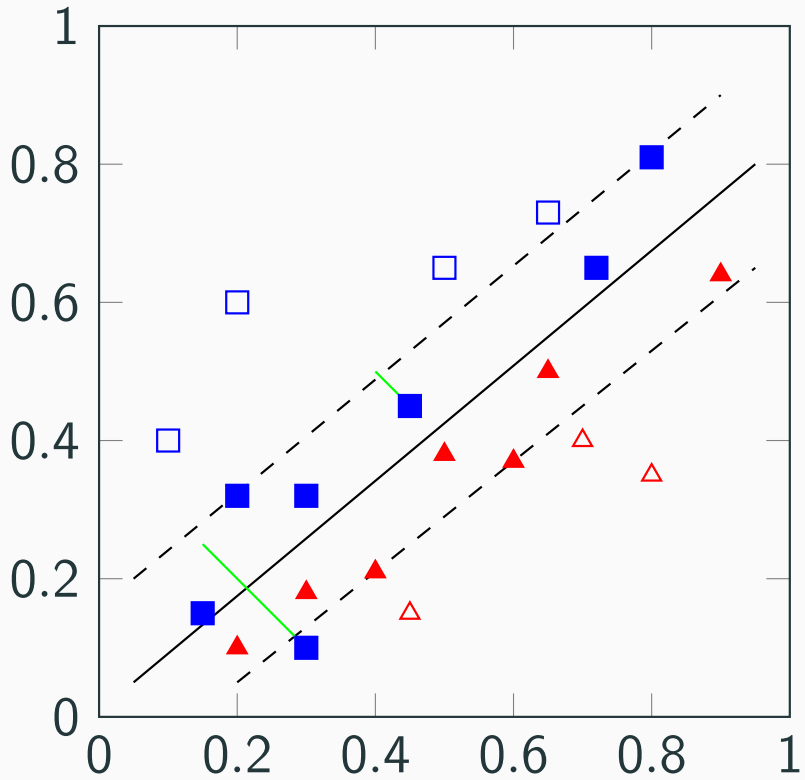
C-SVM

This how the margins looks with a medium C value, like 1000.



C-SVM – ξ

The value of ξ is higher when the object is farther away from the margin.



ν-SVM

The soft margin classifier is a very flexible approach allowing us to soften the criteria of separating hyperplane. We see that the bigger value of C more support vectors we have. However we do not have the direct influence of this number. So we can propose another realization of the soft margins called ν-parametrization.

The parameter C is replaced by a parameter $\nu \in [0, 1]$ which is the lower and upper bound on the number of examples that are support vectors and that lie on the wrong side of the hyperplane.

ν-SVM

The primal problem in this approach will be formulated as follows:

$$\frac{1}{2}||w||^2 - \nu\rho + \frac{1}{2}\sum \xi_i \tag{43}$$

subject to:

$$y_i(w^T x_i) + b \geq \rho - \xi_i, \quad i = 1, \ldots, n, \quad \textit{and} \quad \xi_i \geq 0, \quad \rho \geq 0 \tag{44}$$

ν-SVM – ν and ρ

Now, we have no constant C in the formula. It has been replaced by a parameter ν and an additional variable ρ to be optimized. Note that for $\xi_i = 0$ our constraint states that the two classes are separated by the margin equal to $\frac{2\rho}{||w||}$.

ν-SVM – ν and ρ

To explain what is the parameter ν, let us introduce the term margin error R. We denote that training points with $\xi_i > 0$ are the points which either are errors, or lie within the margin. So formally, the fraction of margin errors is:

$$R_\rho(w, b) = \frac{1}{n}|\{i|y_i(w^T x_i + b) < \rho\}| \tag{45}$$

Now, let us assume that we run ν-SVM with kernel function k; on some data with and we get some $\rho > 0$ Then:

- ν is an upper bound of the fraction of margin errors and hence also on the fraction of training errors,
- ν is a lower bound on the fraction of support vectors.

ν-SVM – dual problem

Let us look at the dual problem for ν-SVM algorithm. Our Lagrangian will be in the form:

$$L = \frac{1}{2}||w||^2 - \nu\rho + \frac{1}{n}\sum \xi_i - \sum(\alpha_i(y_i(w^T x_i + b) - \rho + \xi_i) + \beta_i\xi_i = \delta\rho) \tag{46}$$

where $\alpha_i, \beta_i, \delta$ are the multipliers. If we compute the partial derivatives from KKT conditions and set set them to 0 we obtain the following conditions.

$$w = \sum \alpha_i y_i x_i \tag{47}$$

$$\alpha_i + \beta_i = 1/n \tag{48}$$

$$\sum \alpha_i y_i = 0 \tag{49}$$

$$\sum \alpha_i - \delta = \nu \tag{50}$$

Non-linear separation

Kernels

In some cases the problem is that we are looking for a linear boundary when the boundary in our space is non-linear. However is does not mean that there is no space in which this problem have no linear boundary. If we find the mapping to such space the problem could be solved.

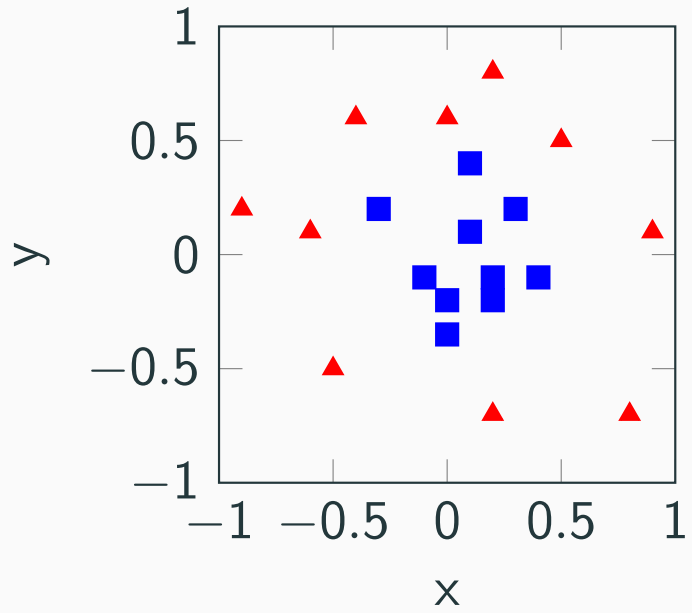
The question is if we can map the features space into more dimensions in such a way to make the problem linearly separable. Is is quite obvious that if we will use the function:

$$K(x) = (x_1^2, \sqrt{(2)}x_1x_2, x_2^2) \tag{51}$$

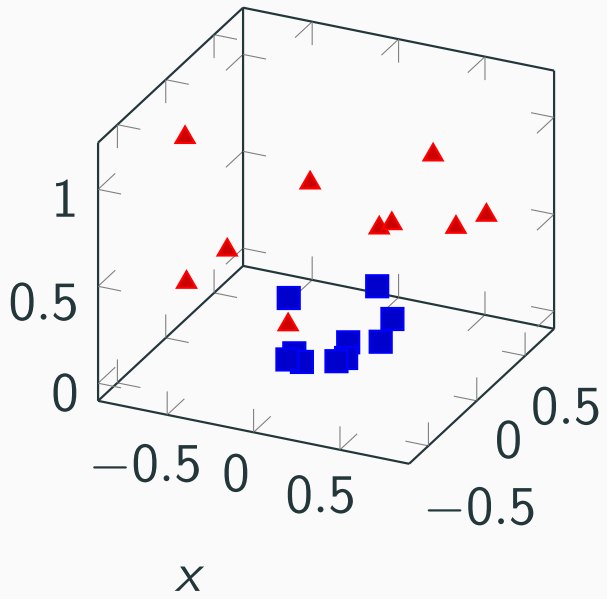
then we transfer our problem into R^3 space in which it is linearly separable.

Kernel trick

Linear separable in three-dimensional feature space Linear non-separable in two-dimensional feature space



(c)



(d)

Kernel function types

Kernel function can be formulated as:

$$K : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}. \tag{52}$$

We have a few commonly used like:

- linear $K(x_i, x_j) = \sum x_{ik} x_{jk}$,
- polynomial $K(x_i, x_j) = (\sum x_{ik} x_{jk} + a)^d$,
- radial $K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$,
- sigmoid $K(x_i, x_j) = \tanh(\langle (x_i, x_j \rangle + r)$.

When the support vector classifier is combined with a non-linear kernel the decision function is non-linear and has the form:

$$f(x) = \beta_0 + \sum \alpha_i K(x_i, x_j) \tag{53}$$

Kernels

We have two big advantages of using a kernel rather than simply enlarging the feature space using functions of the original features:

- one advantage is computational – this can be done without explicitly working in the enlarged feature space. This is important because in many applications of SVMs, the enlarged feature space is so large that computations are intractable,
- seconds advantage is that we go into much much more dimensional feature space in which our problem can be linearly separable.

Extensions

Multiclass SVM

SVM is a binary classifier, but there are two approaches how to deal with multiclass cases:

- one versus one,
- one versus all.

Multiclass SVM

In one versus one we build a classifier with two classes in each possible combination. It means that we get $\frac{N(N-1)}{2}$ combinations. Each combination is a vote for a class that a given is classified to. The object is assigned a label that gets most votes.

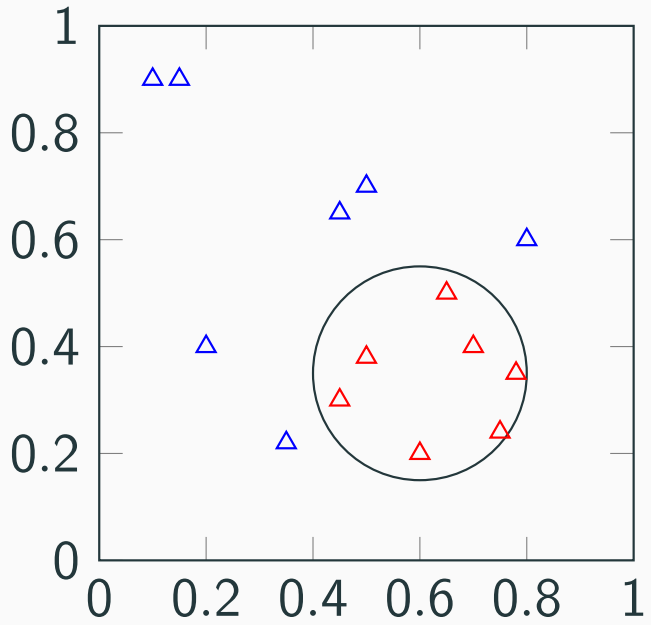
Multiclass SVM

One versus all is less complex as we build N classifiers for each class. Each combination divide the data set into two subsets, one that contains a given class, second that contains the rest.

Usually one versus one gives better results, but is more complex.

One-class SVM

One class example. The classified one class examples are marked with red and outfittery cases with blue. The one-class SVM is similar to one versus all, but just for one class. The basic idea is to enclose data with a hypersphere and classify new data as normal if it falls within the hypersphere and otherwise as anomalous data.



One-class SVM

Let r be the radius of the hypersphere and $c \in R$ the center of this hypersphere. To find the minimum enclosing hypersphere we have to minimize r^2 subject to:

$$||K(x_i) - c||^2 \leq r^2, \quad i = 1, \ldots, p. \tag{54}$$

Then we introduce Lagrangian multiplier for each constraint and obtain:

$$L(,r, \alpha) = r^2 + \sum \alpha_i (||K(x_i) - c||^2 - r^2), \quad \alpha_i \geq 0. \tag{55}$$

Questions?