UNIVERSITY OF LEEDS

Class: Machine Learning

Support Vector Machines

Instructor: Matteo Leonetti

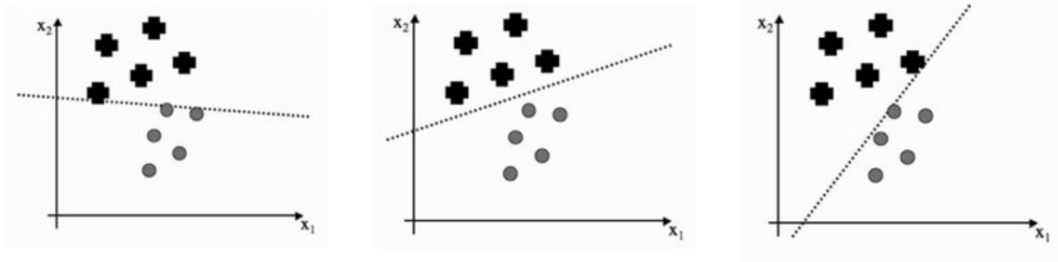
Learning outcomes

UNIVERSITY OF LEEDS

- Derive the formulation of support vector machines as a constrained optimisation problem

Multiple separation boundaries

UNIVERSITY OF LEEDS



Is any one better than the others?

A linearly separable dataset can be classified by many different linear surfaces.

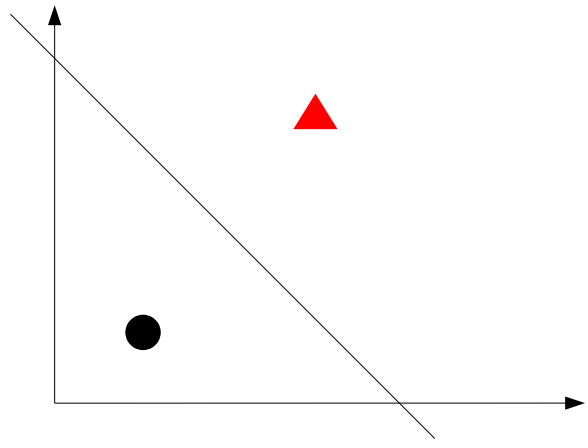
From the point of view of the perceptron algorithm they are all equivalent, because they all achieve 0 error (also known as loss).

Is any of these discriminating boundaries better than the others, though?

Both empirical and theoretical analysis suggests that the one that's farther form the the points on both sides is the best one. Computing this boundary is the goal of Support Vector Machines.

The Best Discriminating Boundary

UNIVERSITY OF LEEDS



What is special about this line?

The line we want has two characteristics:

- all the points are on the correct side
- the distance between the line and the points is maximum.

This will result into a new optimisation problem.

So far we have only seen *unconstrained* optimisation, which means that we tried to minimise a certain error (usually the mean squared error), but the weights were allowed to take any value (hence, unconstrained). Each weight of a neural network or of a linear regressor is a real number, and can take any value in the set of real numbers.

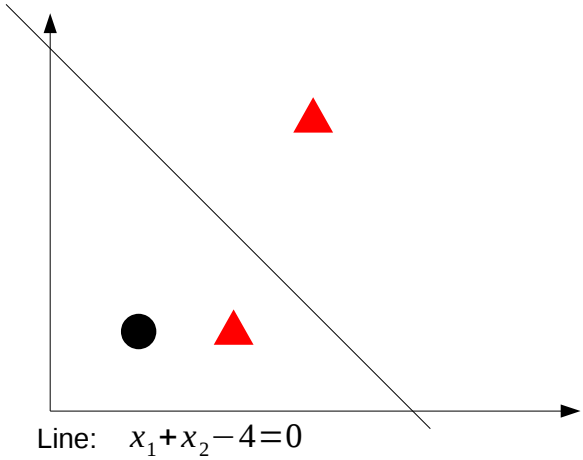
We are now going to consider *constrained* optimisation, where some solutions are explicitly forbidden. The solutions that are allowed form what's called the *feasible* region.

In our case, the only feasible solutions are the ones for which the corresponding discriminating boundary has all the points on the correct side. Among those, we want the boundary that is at the maximum distance from all the points.

We now proceed to define such constraints. We ask the question: how can we mathematically express that a point is on the correct side?

The Constraints

UNIVERSITY OF LEEDS



Classes: {-1,1}

Let's redefine the output of the classifier:

$$y(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{w}^T\boldsymbol{x} + w_0 \geq 0 \\ -1 & \text{if } \boldsymbol{w}^T\boldsymbol{x} + w_0 < 0 \end{cases}$$

Example

Line: $x_1 + x_2 - 4 = 0$

point: $\langle 1,1 \rangle$ desired class: $t = -1$ output: $1+1-4 = -2 \leq 0 \Rightarrow y = -1$

point: $\langle 3,3 \rangle$ desired class: $t = 1$ output: $3+3-4 = 2 > 0 \Rightarrow y = 1$

point: $\langle 2,1 \rangle$ desired class: $t = 1$ output: $2+1-4 = -1 \leq 0 \Rightarrow y = -1$

Instead of the usual 0 and 1, let's use -1 and 1 for the classes. You'll see that this simplifies the notation.

We can note that when a point is classified correctly, t and y are the same, while they are different otherwise.

The Constraints

UNIVERSITY OF LEEDS

Line: $x_1 + x_2 - 4 = 0$

point: $\langle 1,1 \rangle$	desired class:	$t = -1$	output:	$1+1-4 = -2 \leq 0 \Rightarrow y = -1$
point: $\langle 3,3 \rangle$	desired class:	$t = 1$	output:	$3+3-4 = 2 > 0 \Rightarrow y = 1$
point: $\langle 2,1 \rangle$	desired class:	$t = 1$	output:	$2+1-4 = -1 \leq 0 \Rightarrow y = -1$

When the point is classified correctly:

$$ty = 1$$

Since: $y(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{w}^T \boldsymbol{x} + w_0 \geq 0 \\ -1 & \text{if } \boldsymbol{w}^T \boldsymbol{x} + w_0 < 0 \end{cases}$ This is the same as:

$$t(\boldsymbol{w}^T \boldsymbol{x} + w_0) \geq 0$$

If x is correctly classified then t and ($\mathbf{w}^T\mathbf{x}$ + w_0) have the same sign, therefore t($\mathbf{w}^T\mathbf{x}$ + w_0) must be positive.

On the other hand, when t($\mathbf{w}^T\mathbf{x}$ + w_0) is positive it means that the point **x** has been classified correctly. However, if t($\mathbf{w}^T\mathbf{x}$ + w_0) = 0 the point is on the discriminating boundary, and we don't really know how to classify it. To avoid this situation, we must additionally impose that no point is allowed to be on the boundary.

Canonical form

UNIVERSITY OF LEEDS

$t(\boldsymbol{w}^T \boldsymbol{x} + w_0) = 0$ Means the classifier is undecided, and should be avoided.

We can do so by imposing that: $t(\boldsymbol{w}^T \boldsymbol{x} + w_0) \geq \epsilon$ with $\epsilon > 0$

By dividing both sides of the inequality by a constant, we can make ε any number (other than 0). We like 1:

$$t(\boldsymbol{w}^T \boldsymbol{x} + w_0) \geq 1$$

This is called the *canonical* form of the constraints.

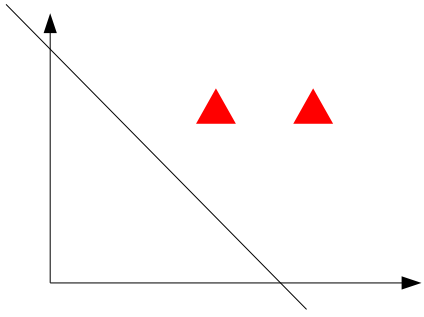
To prevent any point to by on the boundary, we need to separate $t(\mathbf{w}^T\mathbf{x} + w_0)$ from zero, with any small positive number.

Interestingly, similar to the bias input, the actual number does not matter, because we can always make it 1 by rescaling the vector **w** appropriately. Therefore, for notation simplicity, just like with the bias input, we use the constant 1.

I now want to convince you that the particular number ε does not matter, and therefore we may as well use 1.

Example of Constraint

UNIVERSITY OF LEEDS



Line: $x_1 + x_2 - 4 = 0$

$\boldsymbol{w} = \langle -4, 1, 1 \rangle$

The closest point is $\langle 3, 3 \rangle$

$t(\boldsymbol{w}^T \langle 3,3 \rangle + w_0) = 1(\langle 1,1 \rangle^T \cdot \langle 3,3 \rangle - 4) = 2$

$t(\boldsymbol{w}^T \langle 4,3 \rangle + w_0) = 1(\langle 1,1 \rangle^T \cdot \langle 4,3 \rangle - 4) = 3$

So right now, for any point x: $t(\boldsymbol{w}^T \boldsymbol{x} + w_0) \geq 2$

I can rescale the weights so that for the closest point: $t(\boldsymbol{w}^T \boldsymbol{x} + w_0) = 1$

$$\frac{t(\boldsymbol{w}^T \langle 3,3 \rangle + w_0)}{2} = 1\left(\frac{\langle 1,1 \rangle^T}{2} \cdot \langle 3,3 \rangle - \frac{4}{2}\right) = 1(\langle 0.5, 0.5 \rangle^T \langle 3,3 \rangle - 2) = 1$$

So, with the new vector: $\boldsymbol{w}' = \langle -2, 0.5, 0.5 \rangle$

$t(\boldsymbol{w}'^T \boldsymbol{x} + w'_0) \geq 1$ Which is the canonical form

Here is an example of how by rescaling the weights we can obtain the canonical form for any point that is correctly classified.

We begin with the same line we had before, whose initial weights are <-4, 1, 1 >.

We have two points <3,3> and <4,3>. If we take the closest point to the line, <3,3>, we can see that $\mathbf{w}^T\mathbf{x} + w_0$ evaluates to 2. For any other point farther away from the line, it evaluates to something bigger, for instance for <4,3> it is 3.

In order to get $\mathbf{w}^T\mathbf{x} + w_0 >= 1$ rather than 2, we can divide both sides of the inequality by 2. The only thing that can change on the left-hand side is the vector of weights, because the points and the class belong to the dataset and cannot be modified.

With the new vector of weights, we can verify that every point that is correctly classified fulfils the inequality constraint in canonical form (that is, with 1 as its constant).

Constraints

UNIVERSITY OF LEEDS

We consider a vector w valid only if:

$$t\left(\boldsymbol{w}^T \boldsymbol{x} + w_0\right) \geq 1$$

In the notes of slide 4 we asked the question: how can we mathematically express that a point is on the correct side? This is the answer! A point x is on the correct side and not on boundary when: $t(\mathbf{w}^T\mathbf{x} + w_0) >= 1$.

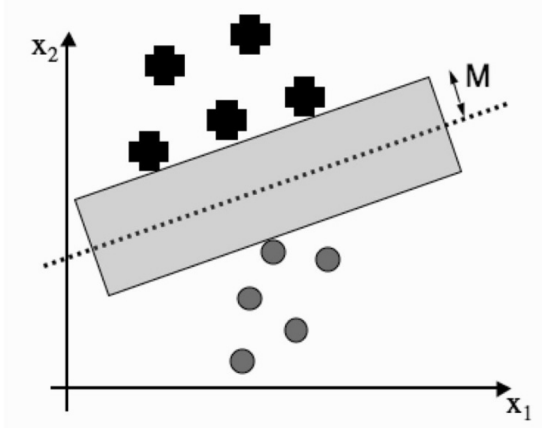
If we impose this for every point we have our feasibility region, that is, only the vectors **w** that fulfil these constraints (one per point!) are admissible solutions. We want to do more though, not only we want one of the vectors that achieve a correct classification, but among those we want the **best** one. We now set out to express this idea of best mathematically as well.

Note how **for the closest points to the boundary, the constrained is satisfied at the equality**. This is very important!

It has to be the case, because $t(\mathbf{w}^T\mathbf{x} + w_0)$ decreases as we get closer to the boundary (becoming 0 on the boundary), and we imposed $t(\mathbf{w}^T\mathbf{x} + w_0) >= 1$ so it must be $t(\mathbf{w}^T\mathbf{x} + w_0) = 1$ for the x that is the closest to the boundary.

The Margin

UNIVERSITY OF LEEDS



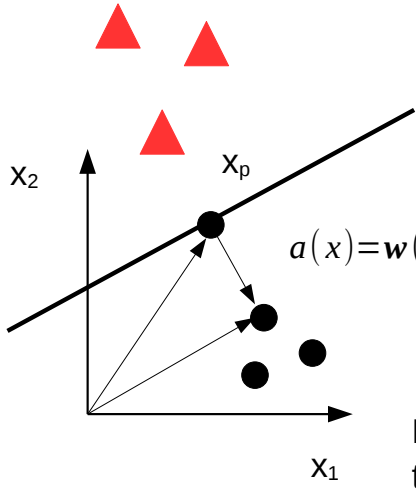
The margin is the distance between the closest point to the separation boundary, and the boundary itself.

We need to express the second requirement we wanted: “the discriminating boundary is as far as possible from the points”.

The distance between the boundary and the *closest* points is called the *margin*.

Recall from the perceptron...

UNIVERSITY OF LEEDS



Distance to the hyperplane

$$\boldsymbol{x} = \boldsymbol{x}_p + d\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$$

$$a(x) = \boldsymbol{w}\left(\boldsymbol{x}_p + d\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right) + w_0 = \underbrace{\boldsymbol{w}\,\boldsymbol{x}_p + w_0}_{=0} + d\frac{\boldsymbol{w}^T\boldsymbol{w}}{\|\boldsymbol{w}\|} = d\|\boldsymbol{w}\|$$

Because x_p is on the hyperplane

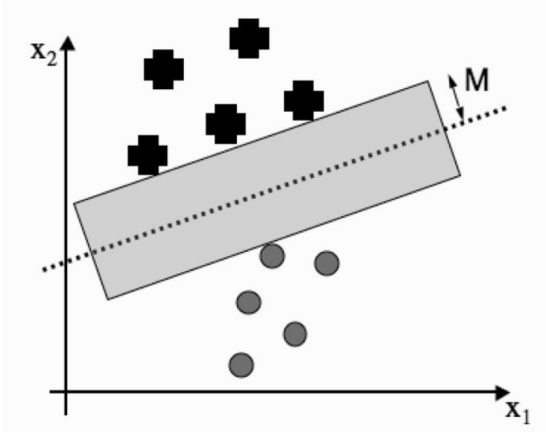
$$\boldsymbol{w}^T\boldsymbol{x} + w_0 = d\|\boldsymbol{w}\|$$

We saw for the neuron that if we evaluate a(x) (which for the neuron we called the "stimulus", that is, the input of the activation function) on any point we get the product of the distance of the point and the norm of the gradient of a(x) (note: without w_0!).

The variable "d" is actually the distance in magnitude, but its sign can be either positive or negative, depending on which side of the hyperplane the point lies on. If it is the side where **w** points to, than it is positive, if it is the other side, it is negative.

The Margin

UNIVERSITY OF LEEDS



The margin is the distance between the closest point to the separation boundary, and the boundary itself

$t = \{-1, +1\}$

$$\frac{t(\boldsymbol{w}^T \boldsymbol{x} + w_0)}{\|\boldsymbol{w}\|} = |d|$$

The margin is what we intend to maximize.

How can we get an analytical expression of the margin?

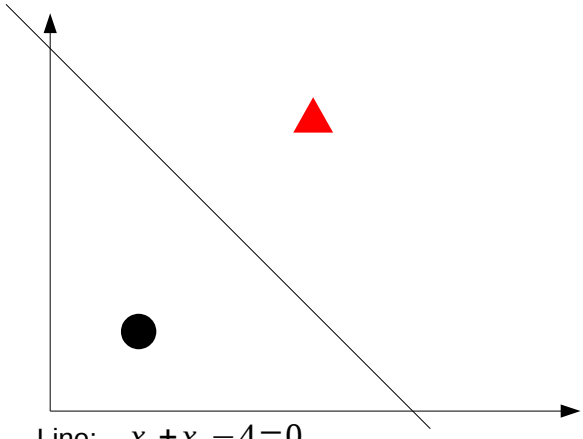
We make use of the fact that a equals d times the norm of **w**. If we then divide by the norm of **w**, we get d, which is the distance of the input point **x**.

As we saw before, d can be either positive or negative depending on which side of the line the input point lies on. To make sure that we get something that's just the magnitude of the distance, we multiply by t, which is negative if d is negative, ensuring that the resulting number is always positive. We can count on this because we have imposed that each point is correctly classified, therefore d and t must have the same sign.

You can verify this on the examples I showed before, which are repeated for convenience in the next slide.

Let's look at this again...

UNIVERSITY OF LEEDS



Classes: {-1,1}

$$y(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{w}^T \boldsymbol{x} + w_0 > 0 \\ -1 & \text{if } \boldsymbol{w}^T \boldsymbol{x} + w_0 \leq 0 \end{cases}$$

$d\|\boldsymbol{w}\|$

Line: $x_1 + x_2 - 4 = 0$

point: $\langle 1,1 \rangle$ desired class: $t = -1$ output: $1+1-4 = -2 \leq 0 \Rightarrow y = -1$

point: $\langle 3,3 \rangle$ desired class: $t = 1$ output: $3+3-4 = 2 > 0 \Rightarrow y = 1$

Here it is possible to see that when a(x) = $\mathbf{w}^T\mathbf{x} + w_0$ is negative, if the points is classified correctly then t=-1. So if we multiply them we can make the result always positive, and that corresponds to the distance of the input point from the line.

Maximum margin

UNIVERSITY OF LEEDS

We saw that for the closest points: $t(\boldsymbol{w}^T \boldsymbol{x} + w_0) = 1$

Therefore:

$$|d| = \frac{t(\boldsymbol{w}^T \boldsymbol{x} + w_0)}{\|\boldsymbol{w}\|} = \frac{1}{\|\boldsymbol{w}\|}$$

We can maximise it by minimising: $\|\boldsymbol{w}\|$

We saw before that the points that are closest to the line verify the constraint at the equality (see notes to slide 10).

If we substitute those points in the expression of the distance we found before, we can see that the distance of the closest points to the line is exactly 1 over the norm of **w**.

Since we want to maximise such a distance, we can do so by minimising the norm of **w**.

The SVM Formulation

UNIVERSITY OF LEEDS

Margin as large as possible

minimise: $\frac{1}{2}\|\boldsymbol{w}\|^2$

Subject to the constraints: $t_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) \geq 1$

Every point is on the correct side, no point is on the hyperplane

This is the entire formulation of the Support Vector Machines.

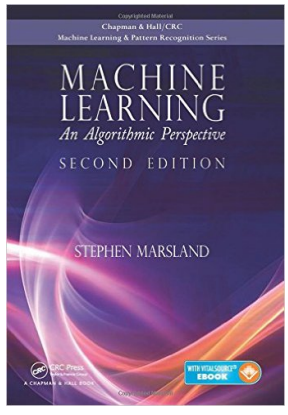
Since the norm has a square root, we actually prefer to minimise it's square, because it is easier to differentiate. Analogously, we can introduce a constant of 1/2 because it gets rid of the 2 in the square when derived, without changing the actual solution.

Quadratic problems have important properties that make them "easy" to solve. Furthermore, the solution, if it exists, is unique.

Therefore, differently from the perceptron and the MLP, if the SVM can find a solution it is the single optimal one. There is no problem with "local" minima, because there is either no minimum, and no solution exists that satisfies all the constraints, or there is exactly one.

Conclusion

UNIVERSITY OF LEEDS



Chapter 8.1