UNIVERSITY OF LEEDS

Class: Machine Learning

Support Vector Machines – part 2

Instructor: Matteo Leonetti

Learning outcomes

UNIVERSITY OF LEEDS

- Define Soft-Margin SVMs
- Project a given dataset to a higher-dimensional space

The SVM Formulation

UNIVERSITY OF LEEDS

Margin as large as possible

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

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

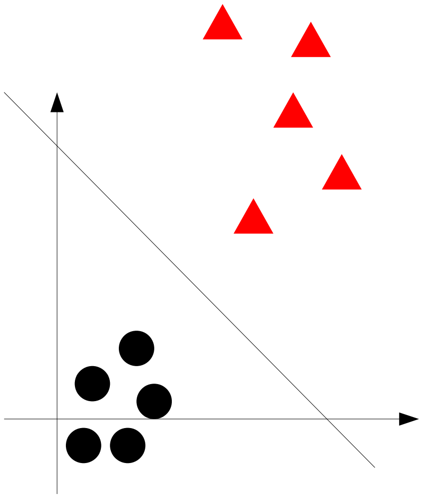
Every point is on the correct side,
no point is on the hyperplane

Let's start from here.

Why SUPPORT VECTOR Machine?

UNIVERSITY OF LEEDS

Consider this dataset:

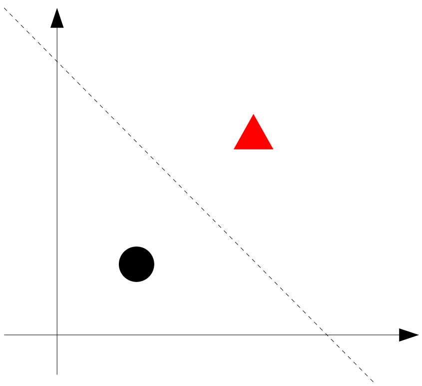


Why is this method called “support vector” machine?

Why SUPPORT VECTOR Machine?

UNIVERSITY OF LEEDS

Consider this dataset:

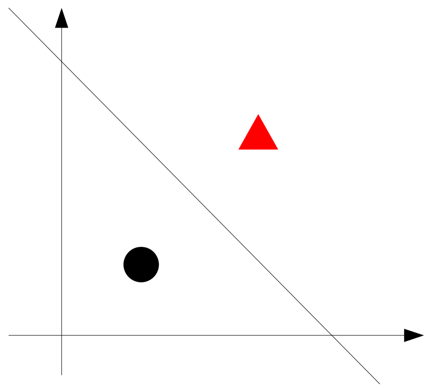


Does the optimal separating line change if I remove all but the closest points?

Why SUPPORT VECTOR Machine?

UNIVERSITY OF LEEDS

Consider this dataset:



No!

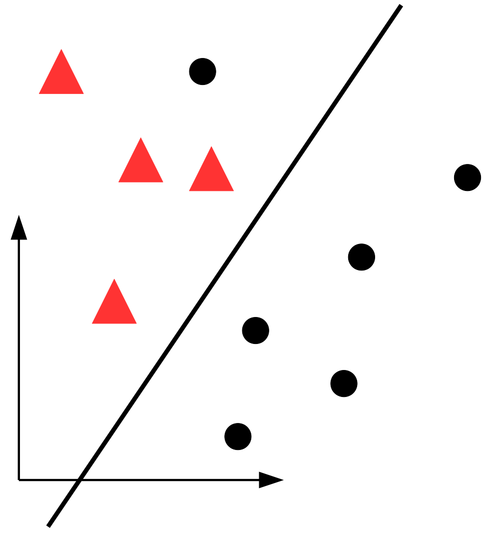
The separating boundary is determined entirely by the points that are closest to it, which for this reason are called *support vectors*.

Those are the points that satisfy their constraint at the equality.

If we knew beforehand which points in the dataset are going to be support vectors we could remove all of the others! However, we cannot know this before we solve the entire optimisation problem.

Back to linear separability

UNIVERSITY OF LEEDS



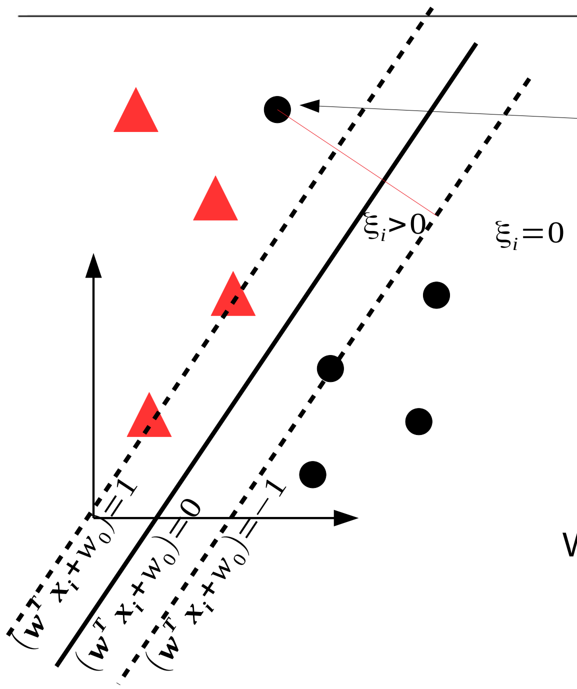
Is this classifier acceptable?

The SVM formulation we saw is called *hard-margin* SVM, since it imposes a hard constraint: either all the points are correctly classified, or no solution is returned.

However, we may find acceptable if a few points are misclassified, and prefer a "softer" solution.

Slack Variables

UNIVERSITY OF LEEDS



For this point we want to allow the corresponding constraint to be *softened*:

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

One way to do this is to introduce an additional, positive, variable:

$$t_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) \geq 1 - \xi_i$$

With the additional constraints:

$$\xi_i \geq 0$$

We can *soften* the constraints by adding additional variables, that represent how much the points are away from where they should be, that is, from being on the correct side of the margin.

These variables, called *slack* variables, are zero for the points that are either support vectors or are behind the margin, while take positive values for the points that are on the wrong side of the margin. Their value is as large as the distance from the point to its margin.

Since the value of the slack variables is a sort of error, we want to minimise this too.

Soft margin

UNIVERSITY OF LEEDS

$$\text{Min} \quad \|\boldsymbol{w}\|^2 + C\sum_i \xi_i$$

How much we are violating the constraints

Weight of violations

Subject to:

$$t_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) \geq 1 - \xi_i \qquad \xi_i \geq 0$$

Sensible to outliers!

$C = \infty$ Hard margin

10

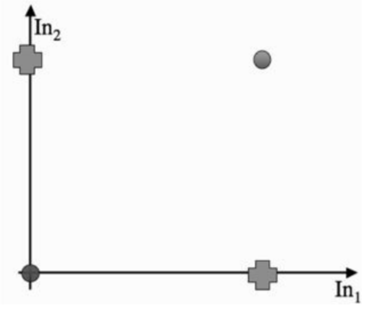
We can minimise the total value of the slack variables by adding them to the objective function.

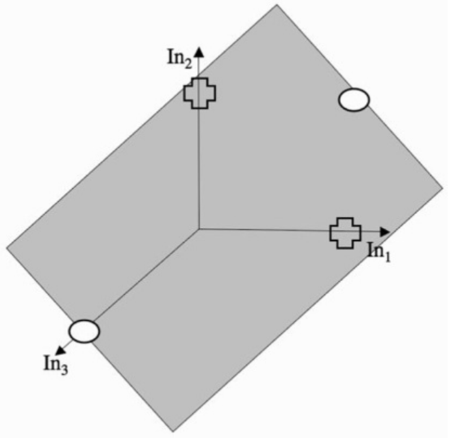
In this way, the optimisation will try to set as many of them as possible to 0 (since that is their minimal value) while minimising the error on the others.

A parameter, C, controls the relative importance of the maximisation of the margin with respect to the minimisation of the slack variables.

The hard-margin corresponds to an infinite importance of the slack variables, so that we either want them all zeros or no solution must be allowed.

Linear separability… revisited





Wait, what?!? More dimensions seem to help!

Where do we get the extra dimensions from?

A liner model, such as an SVM, can only classify a dataset if it is linearly separable. However, by adding dimensions to the datapoints, it is often possible (in fact, always, if adding enough many dimensions) to classify the dataset in the augmented space.

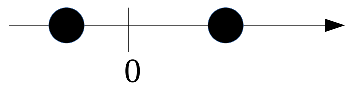
In the example above, the XOR function, which we used as an example of a dataset that is not linearly separable, can be separated by lifting two of the points along a third dimension.

Example: Polynomial features

UNIVERSITY OF LEEDS

Let's take 2 points in 1 dimension:

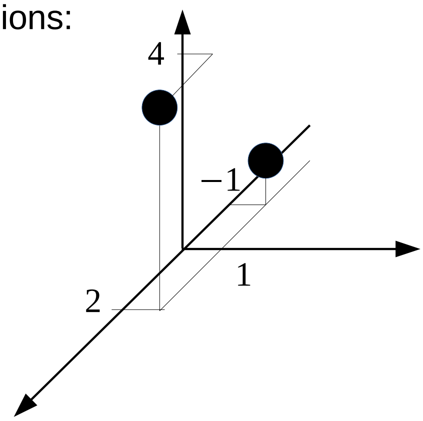
$\langle -1 \rangle, \langle 2 \rangle$



and project them in 3 dimensions:

In general: $\langle 1, x, x^2 \rangle$

Our points: $\langle 1, -1, 1 \rangle, \langle 1, 2, 4 \rangle$



An example of a way to add dimensions is the use of polynomial functions of the data points.

The input points are 1-dimensional, but we can project them in a 3D space by using the features: 1, x, and x^2. You can add as many features as you want, increasing the number of dimensions indefinitely.

Example: Polynomial features

UNIVERSITY OF LEEDS

The original dataset has 1 variable:

$$\langle x_{1,} t_1 \rangle, \langle x_{2,} t_2 \rangle, \ldots, \langle x_N, t_N \rangle$$

But we want a higher dimensional space...

Let's use polynomial features: $\Phi_i(x) = x^i$

Our points become:

$$\langle 1, x_{1,} x_{1,}^2 x_{1,}^3 \ldots, x_{1,}^d t_1 \rangle, \langle 1, x_{2,} x_{2,}^2 x_{2,}^3 \ldots, x_{2,}^d t_2 \rangle, \ldots, \langle 1, x_N, x_N^2, x_N^3, \ldots, x_N^d, t_N \rangle$$

In general, we want to project the dataset on a higher-dimensional space, by using a set of *basis functions* φ.

Features

How can we add extra dimensions?

Original point: x

Define a set of functions $\Phi_i(x)$

New point: $\Phi(x) = \langle \Phi_0(x), \Phi_1(x), \Phi_2(x), \Phi_3(x), \ldots, \Phi_n(x) \rangle$

Substitution

UNIVERSITY OF LEEDS

dataset:

$$\langle x_i, t_i \rangle = \langle -1, 1 \rangle, \langle 2, -1 \rangle \qquad \longrightarrow \qquad \langle \boldsymbol{x}_i, t_i \rangle = \langle 1, -1, 1, 1 \rangle, \langle 1, 2, 4, -1 \rangle$$

problem:

$$\min \quad \frac{1}{2} \| \langle w_1 \rangle \|^2 = \frac{1}{2} w_1^2$$

s.t.:

$$1 \cdot (-1 \cdot w_1 + w_0) \geq 1$$

$$-1 \cdot (2 \cdot w_1 + w_0) \geq 1$$

$$\min \quad \frac{1}{2} \| \langle w_{1,} w_{2,} w_3 \rangle \|^2$$

s.t.:

$$1 \cdot \left(\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + w_0 \right) \geq 1$$

$$-1 \cdot \left(\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} + w_0 \right) \geq 1$$

With the projection, the problem which originally had a single dimension becomes a problem with points in 3D.

Substitution

UNIVERSITY OF LEEDS

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

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

This way we would have a higher dimensional problem, which is also more difficult to solve.

Is there a better formulation?

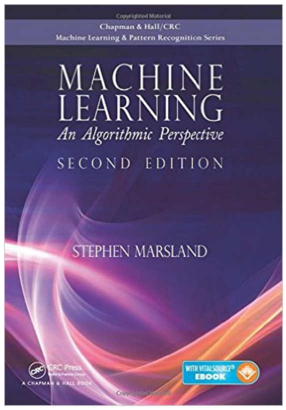
UNIVERSITY OF LEEDS

Conclusion

Learning outcomes

UNIVERSITY OF LEEDS

- Define Soft-Margin SVMs
- Project a given dataset to a higher-dimensional space


Copyrighted Material
Chapman & Hall/CRC
Machine Learning & Pattern Recognition Series
MACHINE LEARNING
An Algorithmic Perspective
SECOND EDITION
STEPHEN MARSLAND
CRC Press
A CHAPMAN & HALL BOOK
WITH VITALSOURCE® EBOOK

Chapter 8.2