

# Clustering & Classification

What are clustering and classification ?

Support vector machines

# Clustering vs classification

Clustering is partitioning an unlabelled dataset into groups of similar objects

Classification sorts data into specific categories using a labelled dataset

# Clustering

From Wikipedia

**Cluster analysis** or **clustering** *is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (clusters).*

There are a myriad of ways to do clustering, this is an extremely active field of research and application. See the Wikipedia page for leads

## We have done clustering already

We have seen some clustering, for example when we sought strongly connected components or cliques or, to some extent, when we did PCA

# Classification

From Wikipedia

*In statistics, **classification** is the problem of identifying which of a set of categories (sub-populations) an observation (or observations) belongs to. Examples are assigning a given email to the "spam" or "non-spam" class, and assigning a diagnosis to a given patient based on observed characteristics of the patient (sex, blood pressure, presence or absence of certain symptoms, etc.).*

What are clustering and classification ?

Support vector machines

# Support vector machines (SVM)

We are given a training dataset of  $n$  points of the form

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$$

where  $\mathbf{x}_i \in \mathbb{R}^P$  and  $y_i = \{-1, 1\}$ . The value of  $y_i$  indicates the class to which the point  $\mathbf{x}_i$  belongs

We want to find a **surface**  $\mathcal{S}$  in  $\mathbb{R}^P$  that divides the group of points into two subgroups

Once we have this surface  $\mathcal{S}$ , any additional point that is added to the set can then be *classified* as belonging to either one of the sets depending on where it is with respect to the surface  $\mathcal{S}$



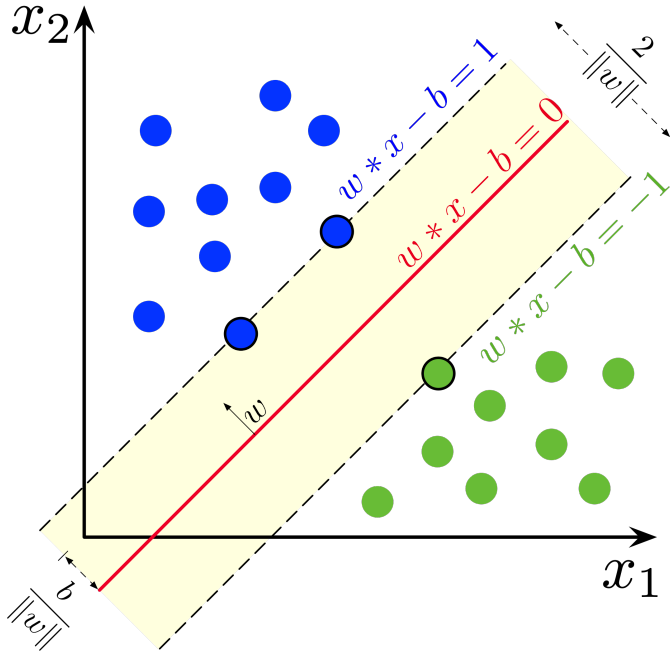
# Linear SVM

We are given a training dataset of  $n$  points of the form

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$$

where  $\mathbf{x}_i \in \mathbb{R}^p$  and  $y_i = \{-1, 1\}$ . The value of  $y_i$  indicates the class to which the point  $\mathbf{x}_i$  belongs

**Linear SVM** – Find the “maximum-margin hyperplane” that divides the group of points  $\mathbf{x}_i$  for which  $y_i = 1$  from the group of points for which  $y_i = -1$ , which is such that the distance between the hyperplane and the nearest point  $\mathbf{x}_i$  from either group is maximized.



Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are the **support vectors**

Any **hyperplane** can be written as the set of points  $\mathbf{x}$  satisfying

$$\mathbf{w}^T \mathbf{x} - b = 0$$

where  $\mathbf{w}$  is the (not necessarily normalized) **normal vector** to the hyperplane (if the hyperplane has equation  $a_1 z_1 + \dots + a_p z_p = c$ , then  $(a_1, \dots, a_n)$  is normal to the hyperplane)

The parameter  $b/\|\mathbf{w}\|$  determines the offset of the hyperplane from the origin along the normal vector  $\mathbf{w}$

Remark: a hyperplane defined thusly is not a subspace of  $\mathbb{R}^p$  unless  $b = 0$ . We can of course transform the data so that it is...

# Linearly separable points

Let  $X_1$  and  $X_2$  be two sets of points in  $\mathbb{R}^p$

Then  $X_1$  and  $X_2$  are **linearly separable** if there exist  $w_1, w_2, \dots, w_p, k \in \mathbb{R}$  such that

- ▶ every point  $x \in X_1$  satisfies  $\sum_{i=1}^p w_i x_i > k$
- ▶ every point  $x \in X_2$  satisfies  $\sum_{i=1}^p w_i x_i < k$

where  $x_i$  is the  $i$ th component of  $x$

# Hard-margin SVM

If the training data is **linearly separable**, we can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible

The region bounded by these two hyperplanes is called the “margin”, and the maximum-margin hyperplane is the hyperplane that lies halfway between them

With a normalized or standardized dataset, these hyperplanes can be described by the equations

- ▶  $\mathbf{w}^T \mathbf{x} - b = 1$  (anything on or above this boundary is of one class, with label 1)
- ▶  $\mathbf{w}^T \mathbf{x} - b = -1$  (anything on or below this boundary is of the other class, with label -1)

Distance between these two hyperplanes is  $2/\|\mathbf{w}\|$

$\Rightarrow$  to maximize the distance between the planes we want to minimize  $\|\mathbf{w}\|$

The distance is computed using the distance from a point to a plane equation

We must also prevent data points from falling into the margin, so we add the following constraint: for each  $i$  either

$$\mathbf{w}^T \mathbf{x}_i - b \geq 1, \text{ if } y_i = 1$$

or

$$\mathbf{w}^T \mathbf{x}_i - b \leq -1, \text{ if } y_i = -1$$

(Each data point must lie on the correct side of the margin)

This can be rewritten as

$$y_i(\mathbf{w}^T \mathbf{x}_i - b) \geq 1, \quad \text{for all } 1 \leq i \leq n$$

or

$$y_i(\mathbf{w}^T \mathbf{x}_i - b) - 1 \geq 0, \quad \text{for all } 1 \leq i \leq n$$

We get the optimization problem:

$$\text{Minimize } \|\mathbf{w}\| \text{ subject to } y_i(\mathbf{w}^T \mathbf{x}_i - b) - 1 \geq 0 \text{ for } i = 1, \dots, n$$

The  $\mathbf{w}$  and  $b$  that solve this problem determine the classifier,  $\mathbf{x} \mapsto \text{sgn}(\mathbf{w}^T \mathbf{x} - b)$  where  $\text{sgn}(\cdot)$  is the **sign function**.

The maximum-margin hyperplane is completely determined by those  $\mathbf{x}_i$  that lie nearest to it.

These  $\mathbf{x}_i$  are the **support vectors**



# The Lagrangian approach

Recall that our goal is to

*minimize  $\|\mathbf{w}\|$  subject to  $y_i(\mathbf{w}^\top \mathbf{x}_i - b) - 1 \geq 0$  for  $i = 1, \dots, n$*

Using Lagrange multipliers  $\lambda_1, \dots, \lambda_n$ , we have the function

$$L_P := F(\mathbf{w}, b, \lambda_1, \dots, \lambda_n) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \lambda_i y_i (\mathbf{x}_i \mathbf{w} + b) + \sum_{i=1}^n \lambda_i$$

Note that we have as many Lagrange multipliers as there are data points. Indeed, there are that many inequalities that must be satisfied

The aim is to minimise  $L_P$  with respect to  $\mathbf{w}$  and  $b$  while the derivatives of  $L_P$  w.r.t.  $\lambda_i$  vanish,  $i = 1, \dots, n$

## Soft-margin SVM

To extend SVM to cases in which the data are not linearly separable, the **hinge loss** function is helpful

$$\max \left( 0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i - b) \right)$$

$y_i$  is the  $i$ th target (i.e., in this case, 1 or -1), and  $\mathbf{w}^T \mathbf{x}_i - b$  is the  $i$ -th output

This function is zero if the constraint is satisfied, in other words, if  $\mathbf{x}_i$  lies on the correct side of the margin

For data on the wrong side of the margin, the function's value is proportional to the distance from the margin

The goal of the optimization then is to minimize

$$\lambda \|\mathbf{w}\|^2 + \left[ \frac{1}{n} \sum_{i=1}^n \max \left( 0, 1 - y_i (\mathbf{w}^T \mathbf{x}_i - b) \right) \right]$$

where the parameter  $\lambda > 0$  determines the trade-off between increasing the margin size and ensuring that the  $\mathbf{x}_i$  lie on the correct side of the margin. Thus, for sufficiently small values of  $\lambda$ , it will behave similar to the hard-margin SVM, if the input data are linearly classifiable, but will still learn if a classification rule is viable or not