Lecture 9. Support Vector Machines

COMP90051 Statistical Machine Learning

Semester 2, 2017
Lecturer: Andrey Kan

THE UNIVERSITY OF MELBOURNE

This lecture

- Support vector machines (SVMs) as maximum margin classifiers
- Deriving hard margin SVM objective
- SVM objective as regularised loss function

Maximum Margin Classifier

A new twist to binary classification problem

Beginning: linear SVMs

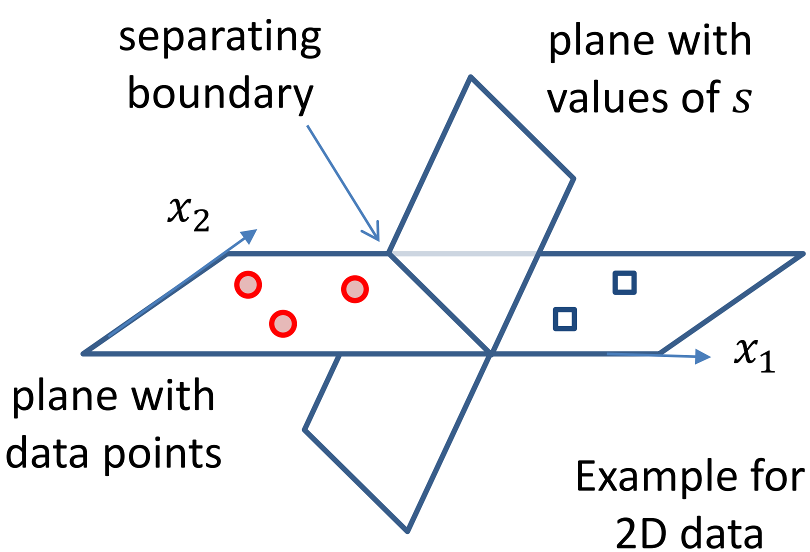
- In the first part, we will consider a basic setup of SVMs, something called linear *hard margin SVM*. These definitions will not make much sense now, but we will come back to this later today
- The point is to keep in mind that SVMs are more powerful than they may initially appear
- Also for this part, we will assume that the data is linearly separable, i.e., the exists a hyperplane that perfectly separates the classes
- We will consider training using all data at once

SVM is a linear binary classifier

SVM is a binary classifier:

Predict class A if $s \geq 0$
Predict class B if $s < 0$
where $s = b + \sum_{i=1}^{m} x_i w_i$

SVM is a linear classifier: s is a linear function of inputs, and the separating boundary is linear



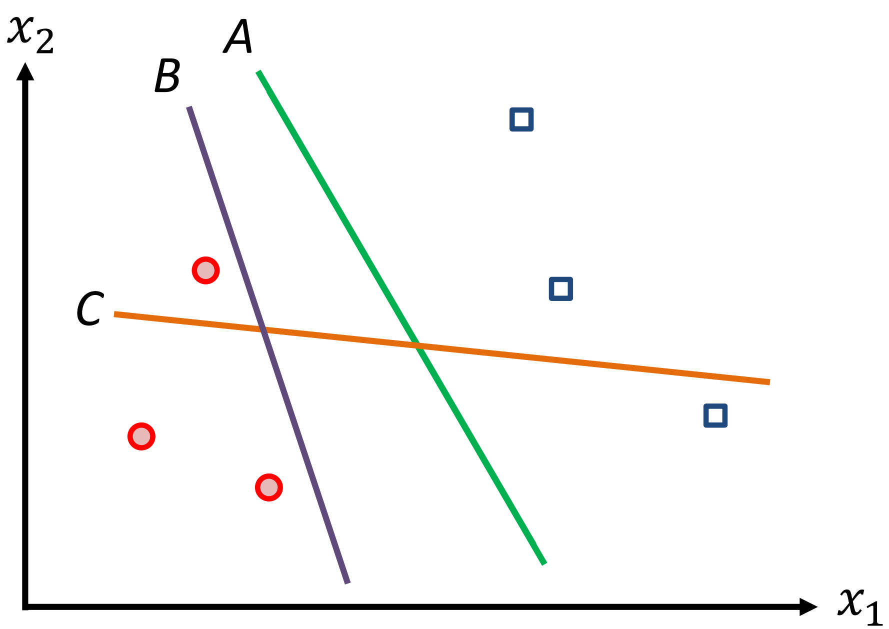
Example for 2D data

SVM and the perceptron

- In fact, the previous slide was taken from the perceptron lecture
- Given learned parameter values, an SVM makes predictions exactly like a perceptron. That is not particularly ground breaking. Are we done here?
- What makes SVMs different is the way the parameters are learned. Remember that here learning means choosing parameters that optimise a predefined criterion
  - E.g., the perceptron minimises perceptron loss that we studied earlier
- SVMs use a different objective/criterion for choosing parameters

Choosing separation boundary

- An SVM is a linear binary classifier, so choosing parameters essentially means choosing how to draw a separation boundary (hyperplane)
- In 2D, the problem can be visualised as follows

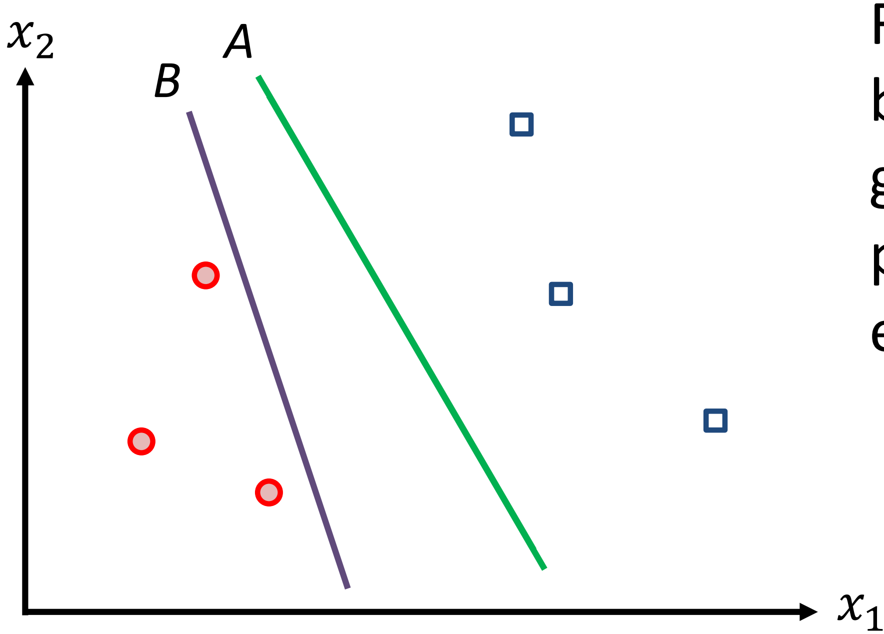


Which boundary should we use?

Line C is a clear “no”, but A and B both perfectly separate the classes

Which boundary should we use?

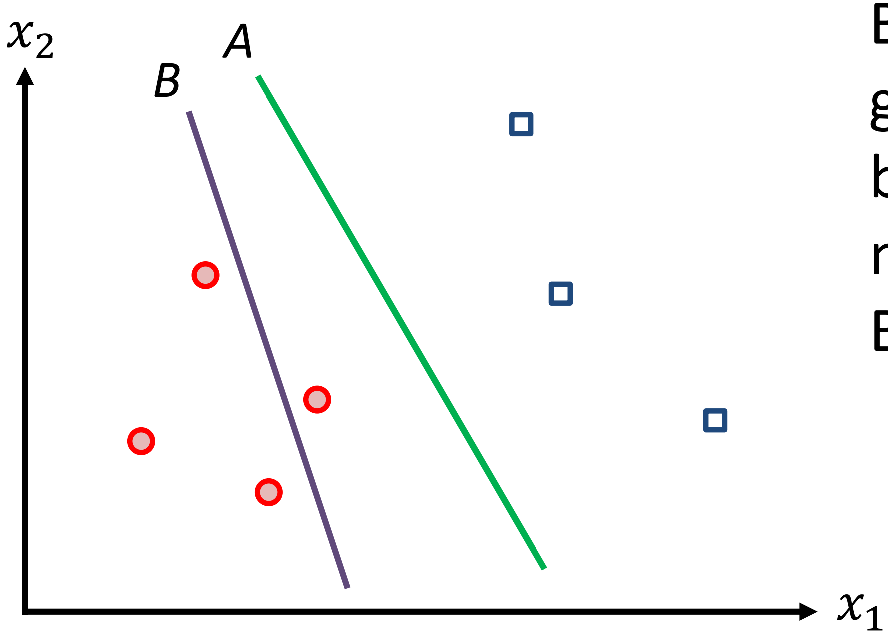
- Provided the dataset is linearly separable, the perceptron will find a boundary that separates classes perfectly. This can be any such boundary, e.g., A or B



For the perceptron, all such boundaries are equally good, because the perceptron loss is zero for each of them.

Which boundary should we use?

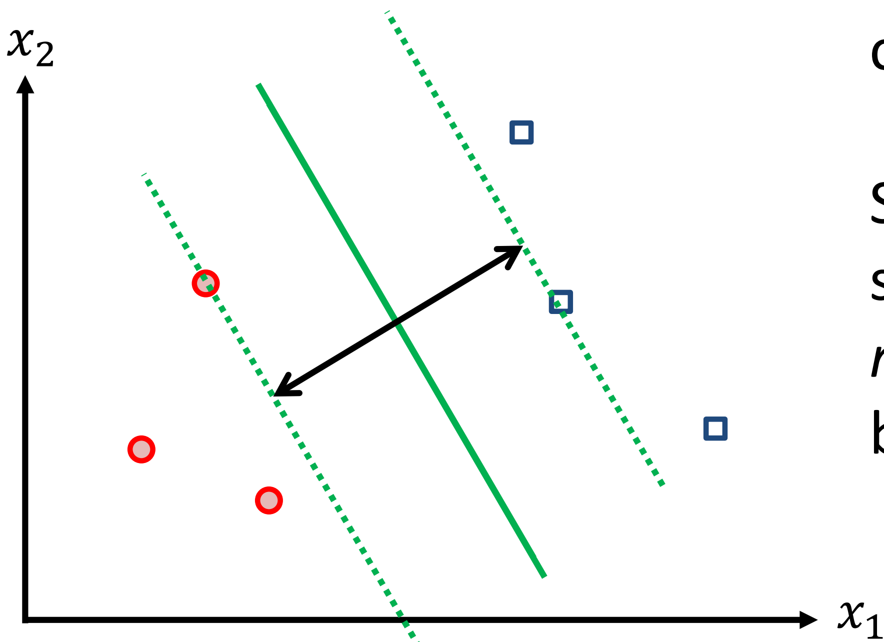
- Provided the dataset is linearly separable, the perceptron will find a boundary that separates classes perfectly. This can be any such boundary, e.g., A or B



But they don't look equally good to us. Line A seems to be more reliable. When new data point arrives, line B is likely to misclassify it

Aiming for the safest boundary

- Intuitively, the most reliable boundary would be the one that is between the classes and as far away from both classes as possible



SVM objective captures this observation

SVMs aim to find the separation boundary that *maximises the margin* between the classes

Maximum margin classifier

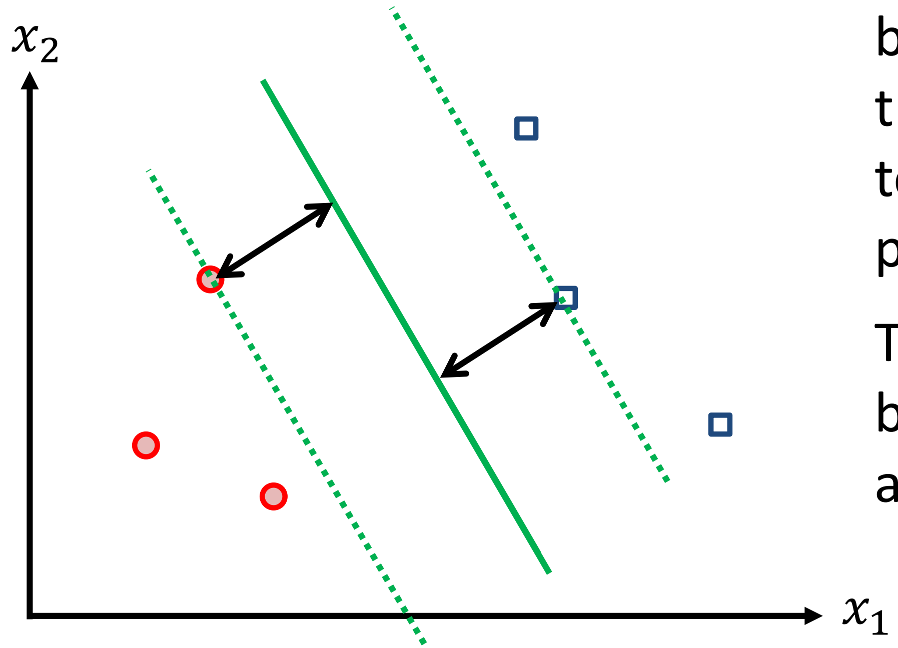
- An SVM is a linear binary classifier. During training, the SVM aims to find the separating boundary that maximises margin
- For this reason, SVMs are also called *maximum margin classifiers*
- The training data is fixed, so the margin is defined by the location and orientation of the separating boundary which, of course, are defined by SVM parameters
- Our next step is therefore to formalise our objective by expressing *margin width* as a function of parameters (and data)

The Mathy Part

In which we derive the SVM objective using geometry

Margin width

- While the margin can be thought as the space between two dashed lines, it is more convenient to define margin width as the distance between the separating boundary and the nearest data point(s)

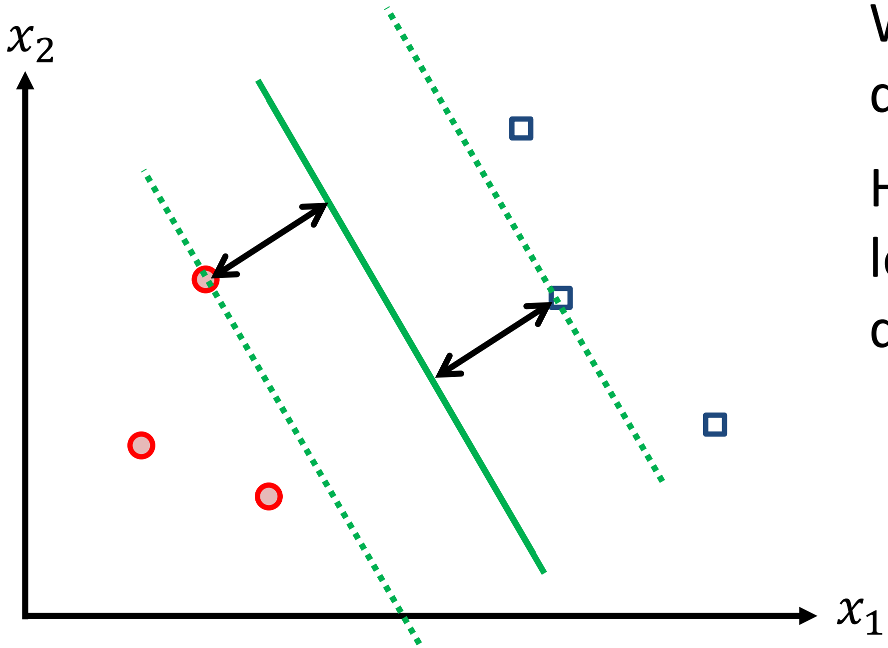


In the figure, the separating boundary is exactly "between the classes", so the distances to the nearest red and blue points are the same

The point(s) on margin boundaries from either side are called *support vectors*

Margin width

- While the margin can be thought as the space between two dashed lines, it is more convenient to define margin width as the distance between the separating boundary and the nearest data point(s)

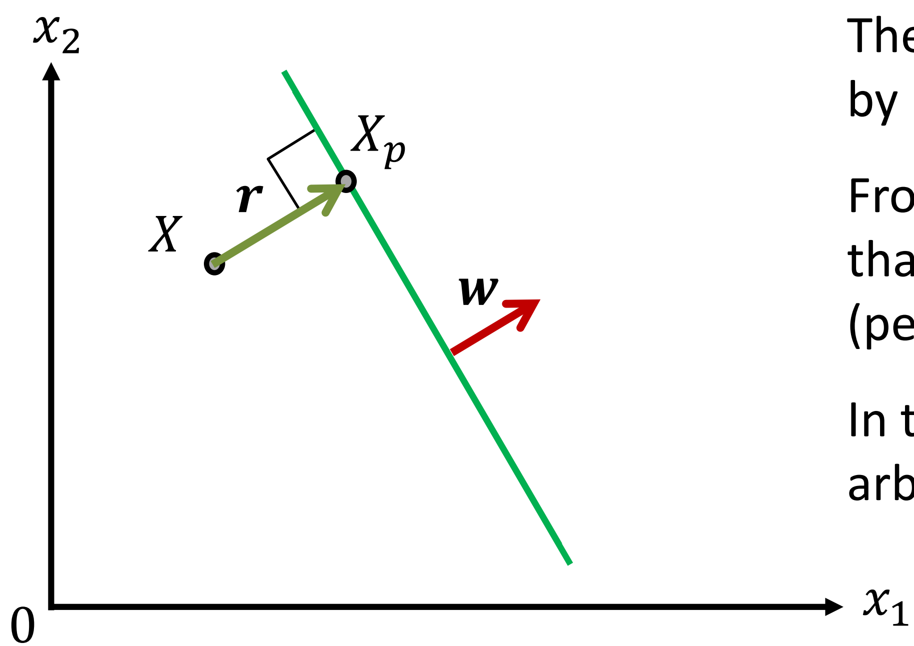


We want to maximise the distance to support vectors

However, before doing this, let's derive the expression for distance to an arbitrary point

Distance from point to hyperplane 1/3

- Consider an arbitrary point X (from either of the classes, and not necessarily the closest one to the boundary), and let X_p denote the projection of X onto the separating boundary
- Now, let $\boldsymbol{r}$ be a vector $\overline{XX_p}$. Note that $\boldsymbol{r}$ is perpendicular to the boundary, and also that $\|\boldsymbol{r}\|$ is the required distance



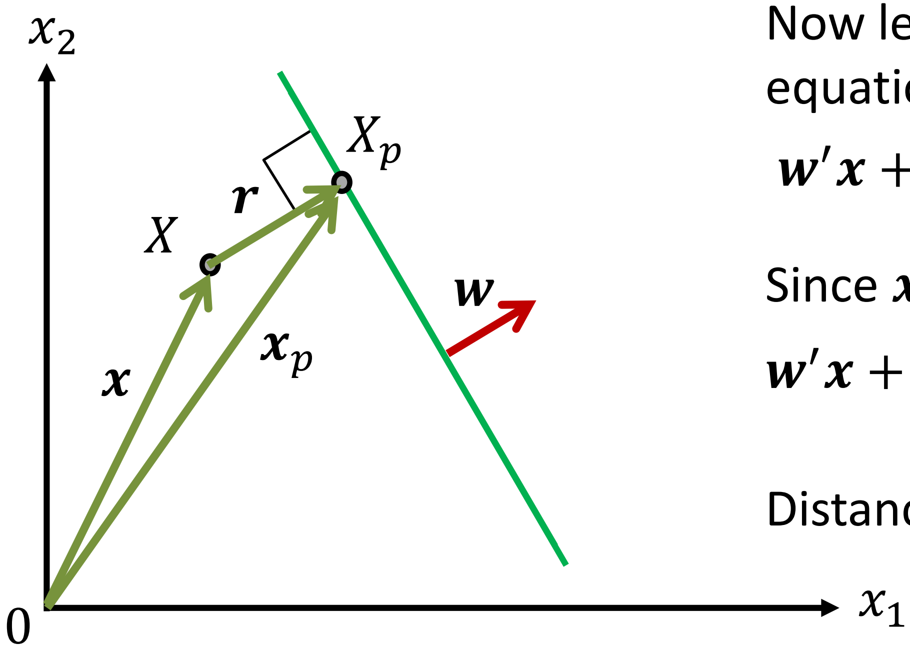
The separation boundary is defined by parameters $\boldsymbol{w}$ and b.

From our previous lecture, recall that $\boldsymbol{w}$ is a vector normal (perpendicular) to the boundary

In the figure, $\boldsymbol{w}$ is drawn from an arbitrary starting point

Distance from point to hyperplane 2/3

- Vectors $\boldsymbol{r}$ and $\boldsymbol{w}$ are parallel, but not necessarily of the same length. Thus $\boldsymbol{r} = \boldsymbol{w}\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{w}\|}$
- Next, points X and X_p can be viewed as vectors $\boldsymbol{x}$ and $\boldsymbol{x}_p$. From vector addition rule, we have that $\boldsymbol{x} + \boldsymbol{r} = \boldsymbol{x}_p$ or $\boldsymbol{x} + \boldsymbol{w}\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{w}\|} = \boldsymbol{x}_p$



Now let's multiply both sides of this equation by $\boldsymbol{w}$ and also add b:

$$\boldsymbol{w}'\boldsymbol{x} + b + \boldsymbol{w}'\boldsymbol{w}\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{w}\|} = \boldsymbol{w}'\boldsymbol{x}_p + b$$

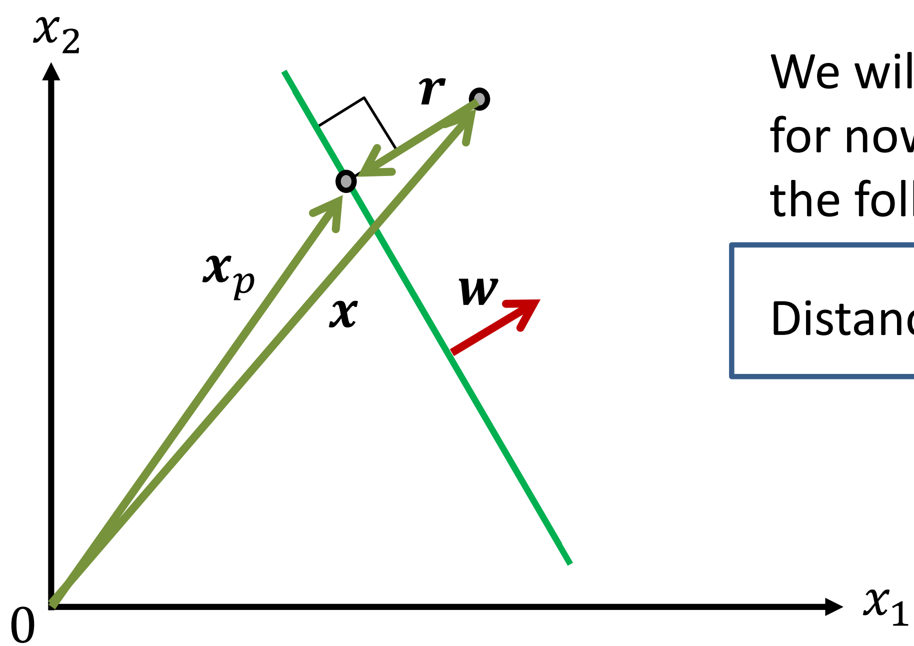
Since $\boldsymbol{x}_p$ lies on the boundary, we have

$$\boldsymbol{w}'\boldsymbol{x} + b + \|\boldsymbol{w}\|^2\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{w}\|} = 0$$

Distance is $\|\boldsymbol{r}\| = -\frac{\boldsymbol{w}'\boldsymbol{x}+b}{\|\boldsymbol{w}\|}$

Distance from point to hyperplane 3/3

- However, if we took our point from the other side of the boundary, vectors $\boldsymbol{r}$ and $\boldsymbol{w}$ would be anti-parallel, giving us $\boldsymbol{r} = -\boldsymbol{w}\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{w}\|}$
- In this case, distance is $\|\boldsymbol{r}\| = \frac{\boldsymbol{w}'\boldsymbol{x}+b}{\|\boldsymbol{w}\|}$



We will return to this fact shortly, and for now we combine the two cases in the following result:

$$\text{Distance is } \|\boldsymbol{r}\| = \pm\frac{\boldsymbol{w}'\boldsymbol{x}+b}{\|\boldsymbol{w}\|}$$

Encoding the side using labels

- Training data is a collection $\{\boldsymbol{x}_i, y_i\}$, $i = 1, \dots, n$, where each $\boldsymbol{x}_i$ is an m-dimensional instance and y_i is the corresponding binary label encoded as -1 or 1
- Given a perfect separation boundary, y_i encode the side of the boundary each $\boldsymbol{x}_i$ is on
- Thus the distance from the i-th point to a perfect boundary can be encoded as $\|\boldsymbol{r}_i\| = \frac{y_i(\boldsymbol{w}'\boldsymbol{x}_i + b)}{\|\boldsymbol{w}\|}$

Maximum margin objective

- The distance from the i-th point to a perfect boundary can be encoded as $\|\boldsymbol{r}_i\| = \frac{y_i(\boldsymbol{w}'\boldsymbol{x}_i+b)}{\|\boldsymbol{w}\|}$
- The margin width is the distance to the closest point
- Thus SVMs aim to maximise $\left(\min_{i=1,\dots,n} \frac{y_i(\boldsymbol{w}'\boldsymbol{x}_i+b)}{\|\boldsymbol{w}\|}\right)$ as a function of $\boldsymbol{w}$ and b

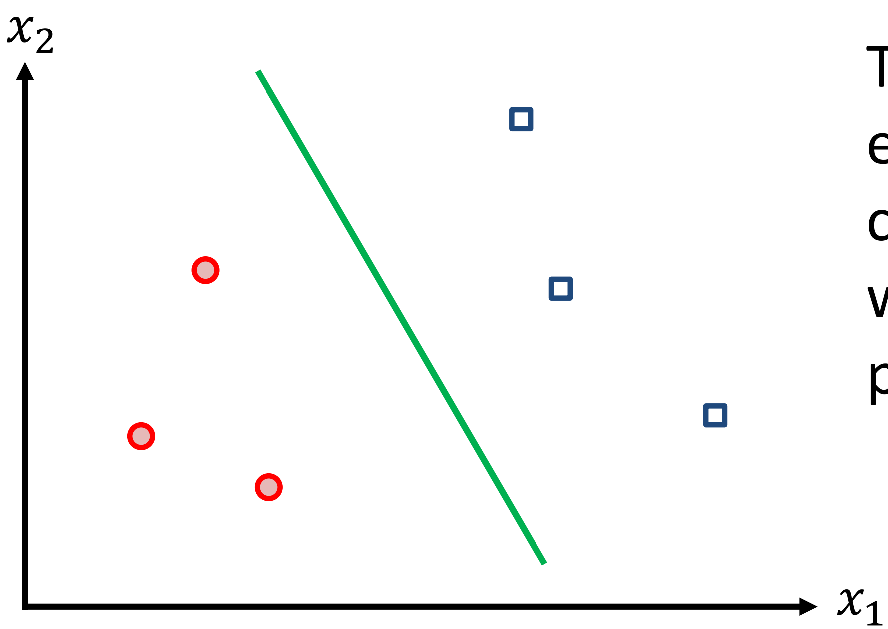
Do you see any problems with this objective?

art: OpenClipartVectors at pixabay.com (CC0)

Remember that $\|\boldsymbol{w}\| = \sqrt{w_1^2 + \cdots + w_m^2}$

Non-unique representation

- A separating boundary (e.g., a line in 2D) is a set of points that satisfy $\boldsymbol{w}'\boldsymbol{x} + b = 0$ for some given $\boldsymbol{w}$ and b
- However, the same set of points will also satisfy $\tilde{\boldsymbol{w}}'\boldsymbol{x} + \tilde{b} = 0$, with $\tilde{\boldsymbol{w}} = \alpha\boldsymbol{w}$ and $\tilde{b} = \alpha b$, where α is an arbitrary constant



The same boundary, and essentially the same classifier can be expressed with infinitely many parameter combinations

Resolving ambiguity

- Consider a “candidate” separating line. Which parameter combinations should we choose to represent it?
  - As humans, we do not really care
  - Math/Machines require a precise answer

- A possible way to resolve ambiguity: *measure the distance to the closest point (i^*), and rescale parameters such that*

$$\frac{y_{i^*}(\boldsymbol{w}'\boldsymbol{x}_{i^*} + b)}{\|\boldsymbol{w}\|} = \frac{1}{\|\boldsymbol{w}\|}$$

- For a given “candidate” boundary, and fixed training points there will be only one way of scaling $\boldsymbol{w}$ and b in order to satisfy this requirement

Constraining the objective

- SVMs aim to maximise $\left(\min_{i=1,\ldots,n} \frac{y_i(\boldsymbol{w}'\boldsymbol{x}_i+b)}{\|\boldsymbol{w}\|}\right)$
- Introduce (arbitrary) extra requirement $\frac{y_{i^*}(\boldsymbol{w}'\boldsymbol{x}_{i^*}+b)}{\|\boldsymbol{w}\|} = \frac{1}{\|\boldsymbol{w}\|}$
  * Here i^* denotes the distance to the closest point
- We now have that SVMs aim to find

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|$$

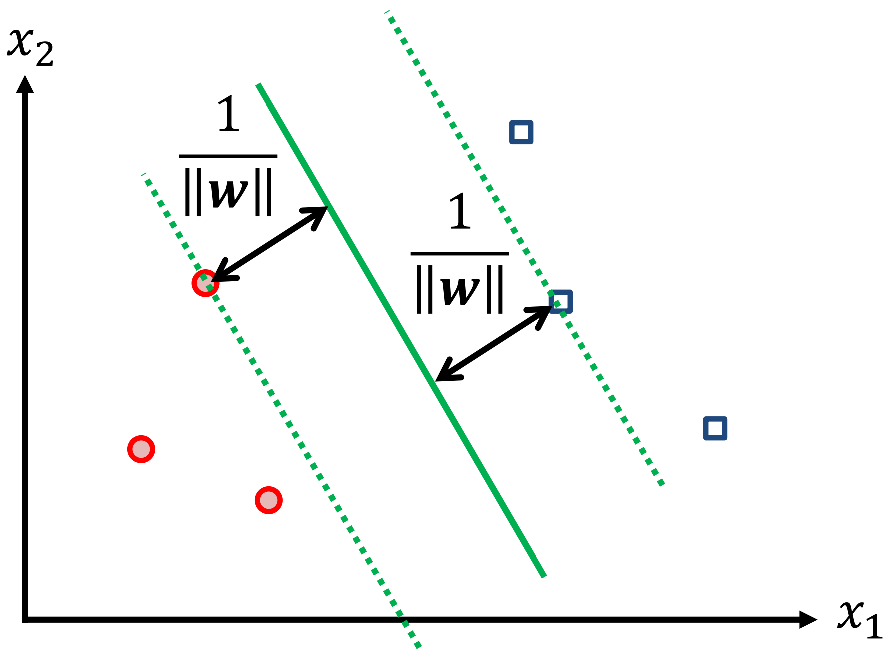
s.t. $y_i(\boldsymbol{w}'\boldsymbol{x}_i + b) \geq 1$ for $i = 1, \ldots, n$

Hard margin SVM objective

We now have a major result: SVMs aim to find

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|$$

s.t. $y_i(\boldsymbol{w}'\boldsymbol{x}_i + b) \geq 1$ for $i = 1, \dots, n$



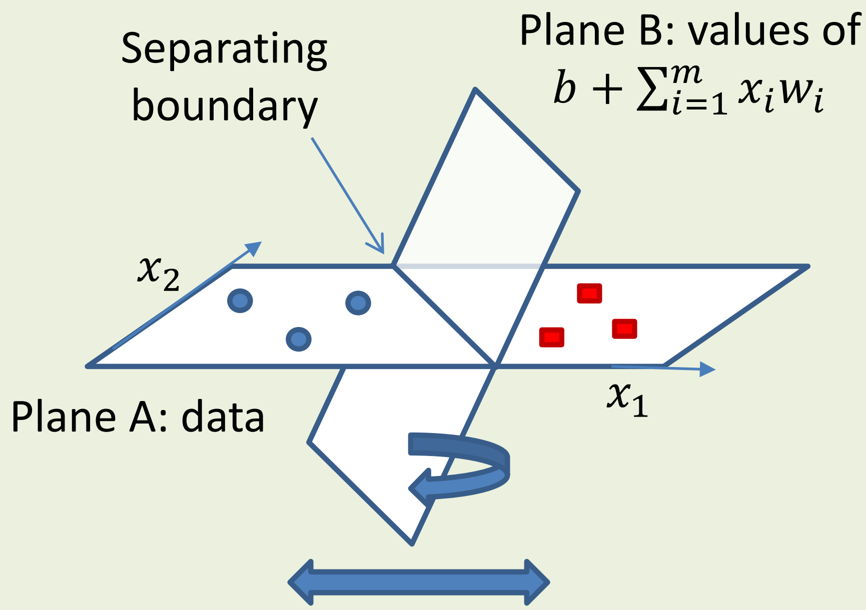
Note 1: parameter b is optimised indirectly by influencing constraints

Note 2: all points are enforced to be on or outside the margin

Therefore, this version of SVM is called *hard-margin SVM*

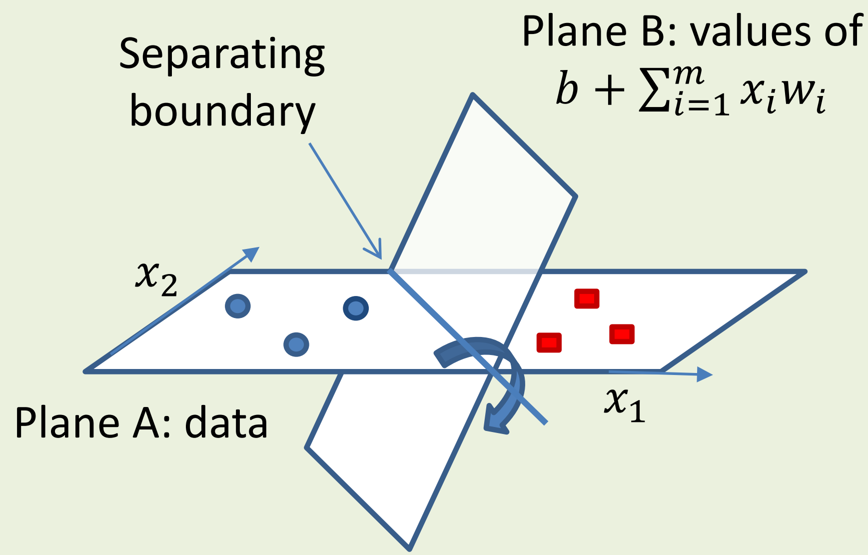
Geometry of SVM training 1/2

- Training a linear SVM essentially means moving/rotating plane B so that separating boundary changes
- This is achieved by changing w_i and b



Geometry of SVM training 2/2

- However, we can also rotate Plane B along the separating boundary
- In this case, the boundary does not change
    - Same classifier!
- The additional requirement fixes the angle between planes A and B to a particular constant



SVM Objective as Regularised Loss

Relating the resulting objective function to that of other machine learning methods

Previously on COMP90051 …

1. Choose/design a model
2. Choose/design discrepancy function
3. Find parameter values that minimise discrepancy with training data

A *loss function* measures discrepancy between prediction and true value for a single example

Training error is the average (or sum) of losses for all examples in the dataset

So step 3 essentially means minimising training error

We defined loss functions for perceptron and ANN, and aimed to minimised the loss during training

But how do SVMs fit this pattern?

Regularised training error as objective

- Recall ridge regression objective

$$\text{minimise } \left(\boxed{\sum_{i=1}^{n}(y_i - \boldsymbol{w}'\boldsymbol{x}_i)^2} + \boxed{\lambda\|\boldsymbol{w}\|^2}\right)$$

- Hard margin SVM objective

data-dependent training error

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \boxed{\|\boldsymbol{w}\|}$$

data-independent regularisation term

$$\text{s.t. } \boxed{y_i(\boldsymbol{w}'\boldsymbol{x}_i + b) \geq 1 \text{ for } i = 1, \ldots, n}$$

- The constraints can be interpreted as loss

$$l_\infty = \begin{cases} 0 & 1 - y_i(\boldsymbol{w}'\boldsymbol{x}_i + b) \leq 0 \\ \infty & 1 - y_i(\boldsymbol{w}'\boldsymbol{x}_i + b) > 0 \end{cases}$$

Hard margin SVM loss

- The constraints can be interpreted as loss

$$l_\infty = \begin{cases} 0 & 1 - y_i(\boldsymbol{w}'\boldsymbol{x}_i + b) \leq 0 \\ \infty & 1 - y_i(\boldsymbol{w}'\boldsymbol{x}_i + b) > 0 \end{cases}$$

- In other words, for each point:
  - If it's on the right side of the boundary and at least $\frac{1}{\|\boldsymbol{w}\|}$ units away from the boundary, we're OK, the loss is 0
  - If the point is on the wrong side, or too close to the boundary, we immediately give infinite loss thus prohibiting such a solution altogether

This lecture

- Support vector machines (SVMs) as maximum margin classifiers
- Deriving hard margin SVM objective
- SVM objective as regularised loss function