



UNIVERSIDADE  
CATÓLICA  
PORTUGUESA

BRAGA

# Machine Learning

Session 17 - T

## Support Vector Machines – Part 1

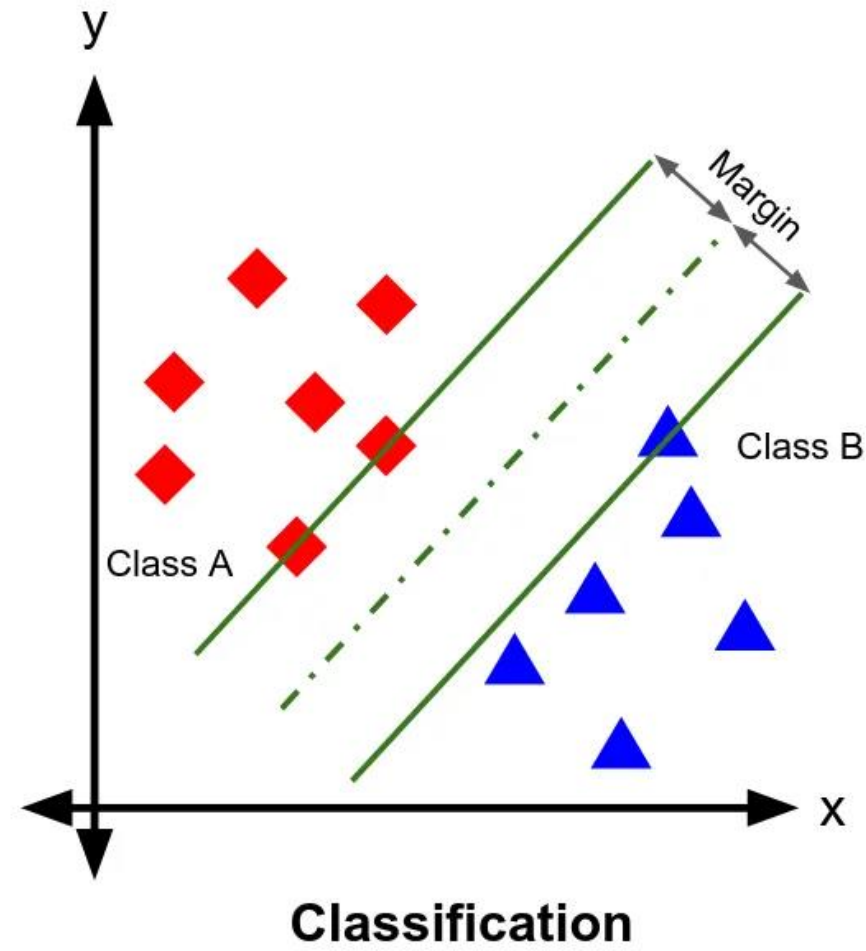
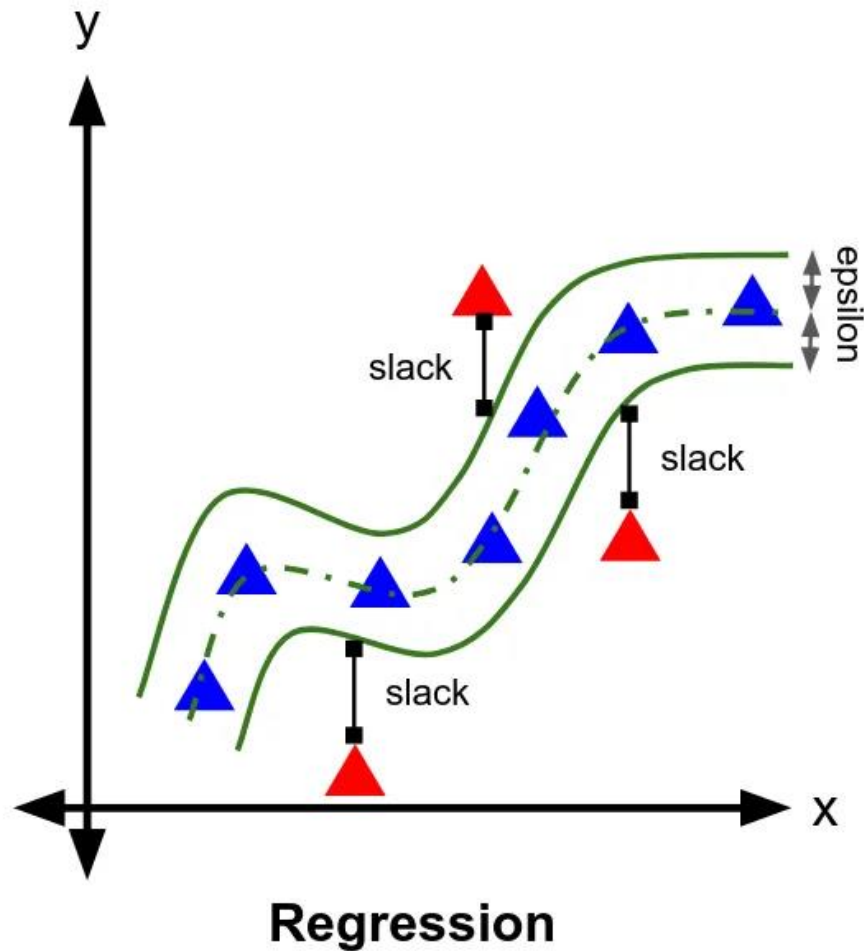
Ciência de Dados Aplicada

2023/2024

# Support Vector Machines (SVMs) - Basics

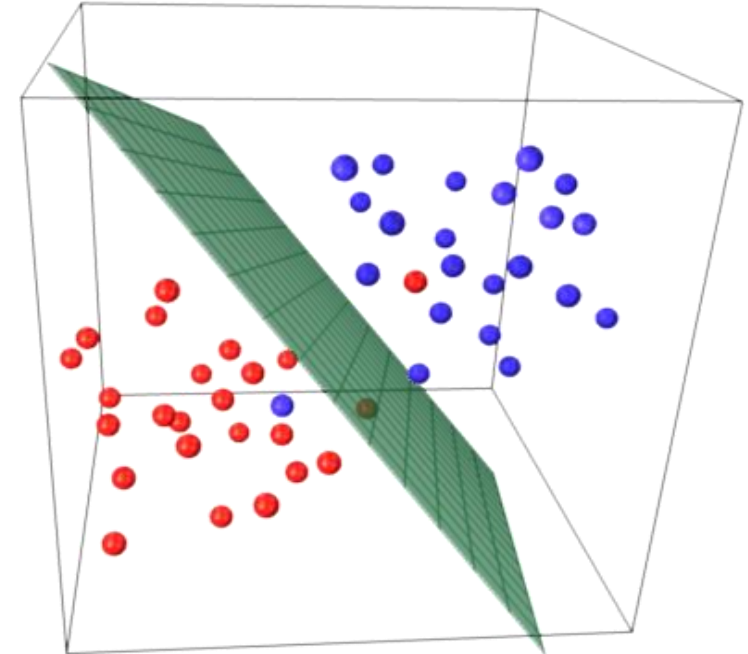
- Supervised machine learning algorithm suitable for both classification and regression.
- **Objective:** Find the optimal hyperplane or decision boundary in the feature space.
- **How?** By maximizing the margin or distance between data points of different classes or regression targets.

# SVMs - Basics



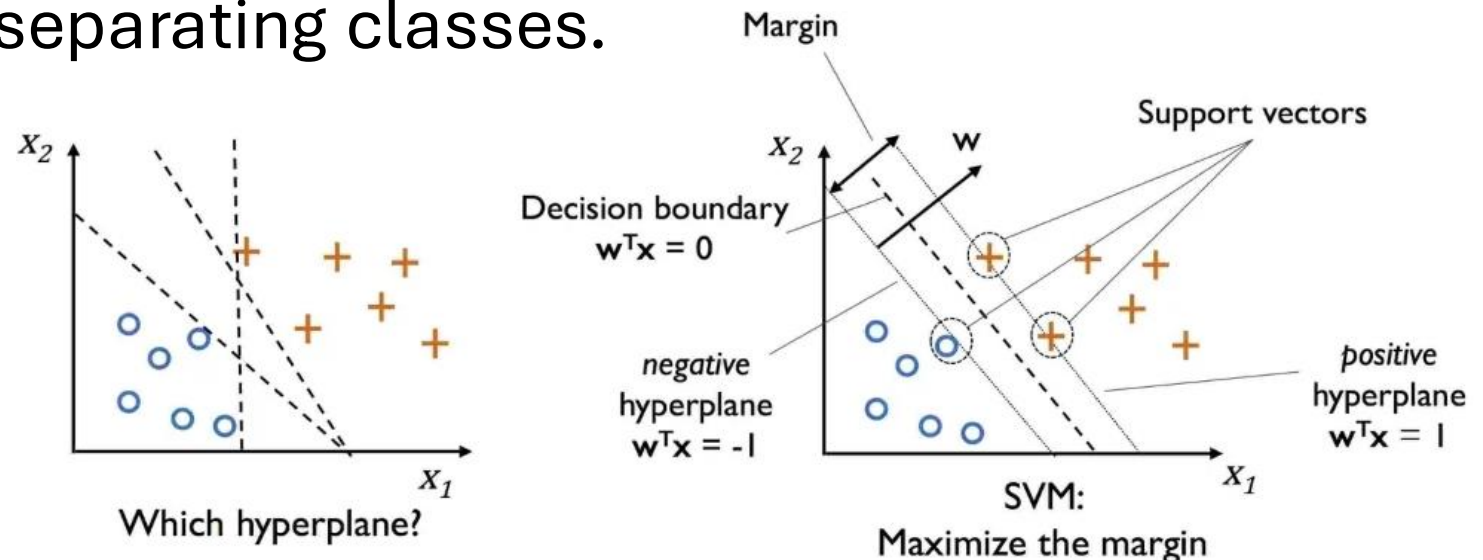
# SVMs - Hyperplanes

- A **hyperplane** is a subspace with **one dimension less** than the number of variables in the dataset ( $n-1$ ).
- For a 3-dimensional dataset, a 2-dimensional plane can be used to separate the data into two distinct groups.
- Multiple hyperplanes can separate the data. The goal is to find the **hyperplane with the maximum margin**.



# Maximum Margin Classifier

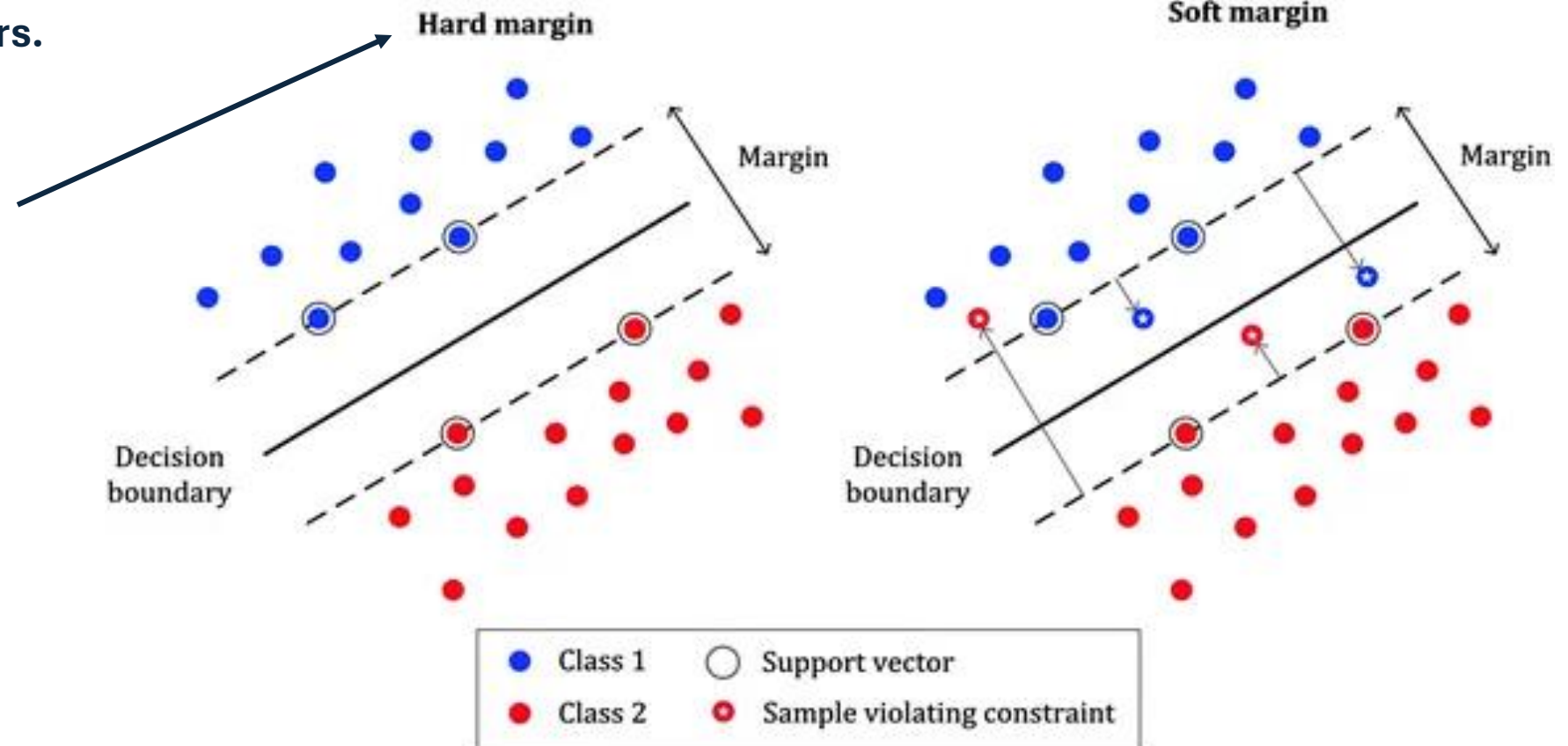
- A **Margin Classifier** is a type of classifier that provides a distance to the discriminant.
  - The hyperplane serves as the boundary separating classes in a linear classifier.
- A **Maximum Margin Classifier** seeks the discriminant with the maximum margin, maximizing the distance to the nearest points (**support vectors**) for separating classes.



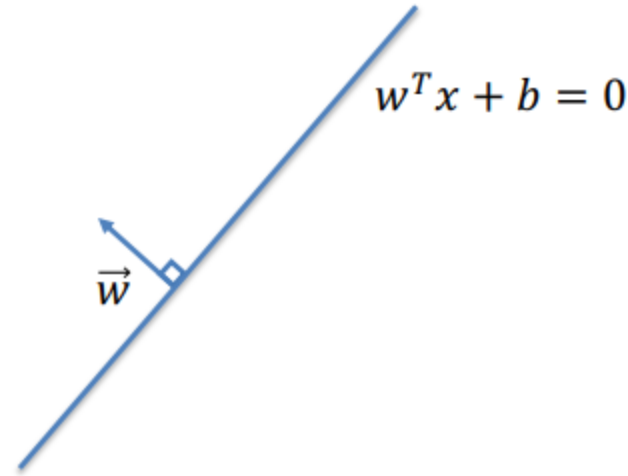
# Maximum Margin Classifier

- **Hard vs Soft Margin**
- Very sensitive to outliers.
- All training instances need to be correctly classified.
- Only for linearly separable data.

- Uses cross validation to find the best support vectors.



# Some Geometry



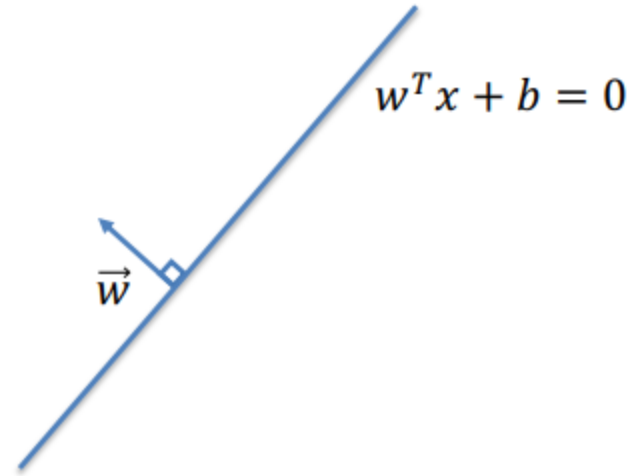
- In  $n$  dimensions, a hyperplane is a solution to the equation:

$$w^T x + b = 0$$

with  $w \in \mathbb{R}^n, b \in \mathbb{R}$

- The vector  $\vec{w}$  is called the **normal vector** of the hyperplane.

# Some Geometry



- In  $n$  dimensions, a hyperplane is a solution to the equation:

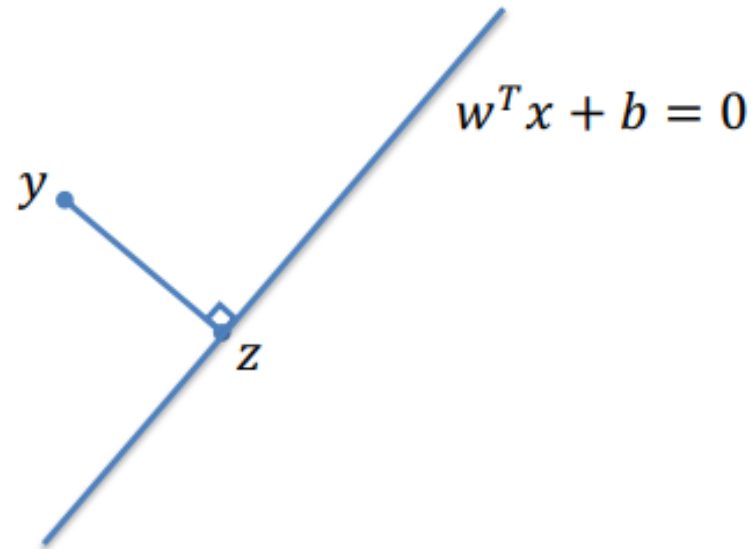
$$w^T x + b = 0$$

- Note that this equation is **scale invariante** for any scalar  $c$

$$c \cdot (w^T x + b) = 0$$



# Some Geometry



- The distance between a point  $y$  and a hyperplane  $w^T x + b = 0$  is the length of the segment **perpendicular** to the line to the point  $y$

- The vector from  $y$  to  $z$  is given by:

$$y - z = \|y - z\| \frac{w}{\|w\|}$$

Note that:

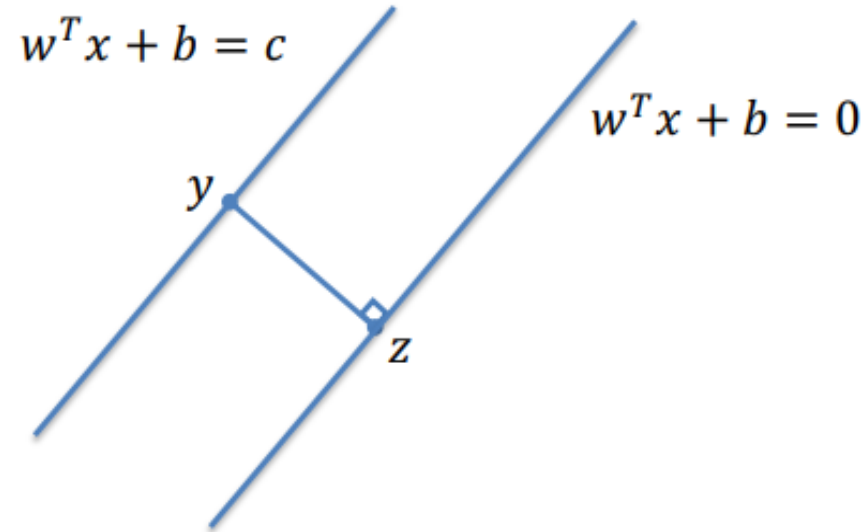
Length of vector  $x(x_1, x_2, x_3)$  is calculated as :

$$\|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2}$$

Direction of vector  $x(x_1, x_2, x_3)$  is calculated as:

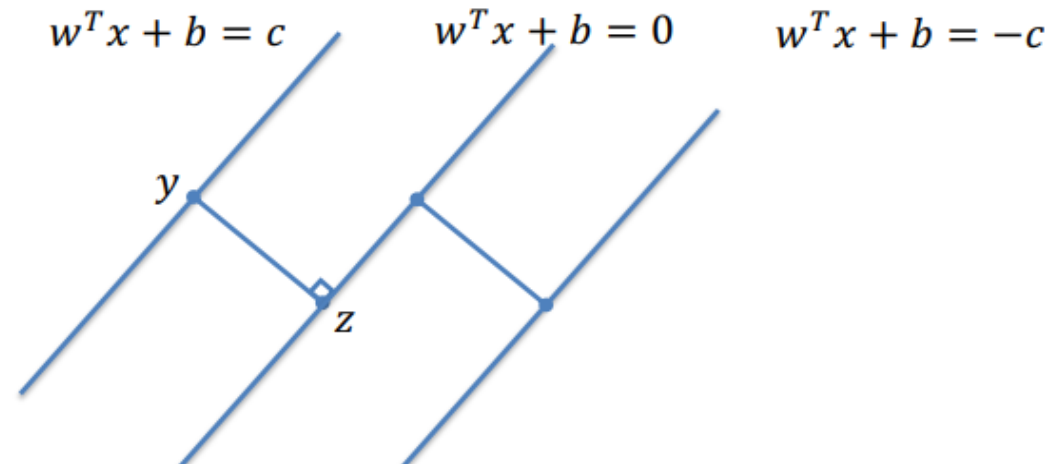
$$\left\{ \frac{x_1}{\|x\|}, \frac{x_2}{\|x\|}, \frac{x_3}{\|x\|} \right\}$$

# Scale Invariance



- By scale invariance, we can assume that  $c = 1$
- The maximum margin is always obtained by choosing  $w^T x + b = 0$  so that it is **equidistant** from the closest data point from each class.

# Scale Invariance

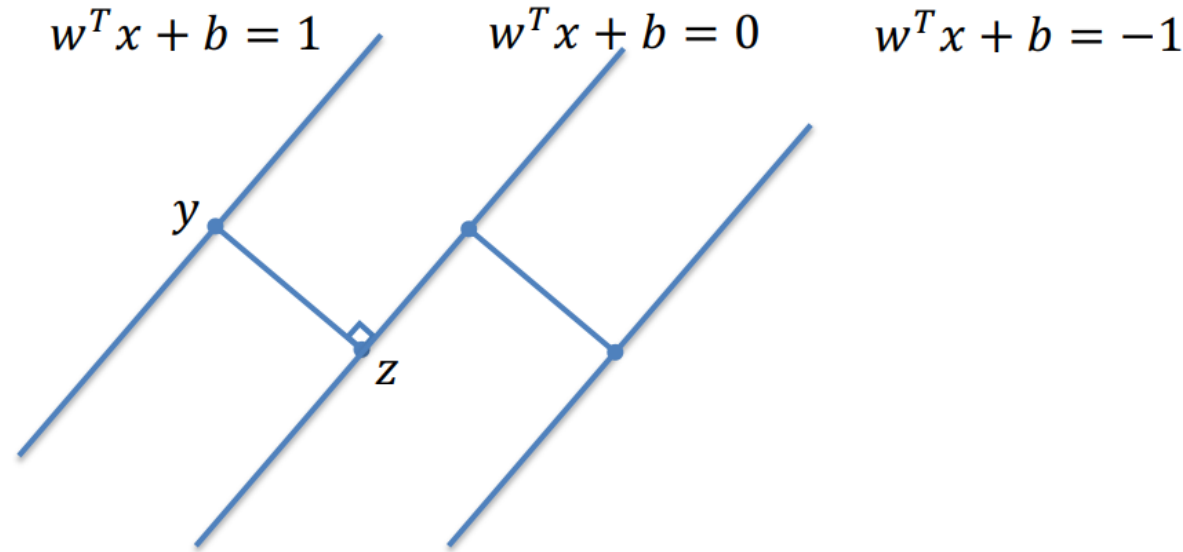


- We want to **maximize the margin** subject to the constraints that

$$y^{(i)}(w^T x^{(i)} + b) \geq 1$$

- But how do we compute the size of the margin?

# Some Geometry



- Putting it all together:

$$y - z = \|y - z\| \frac{w}{\|w\|}$$

and

$$\begin{aligned} w^T y + b &= 1 \\ w^T z + b &= 0 \end{aligned}$$



$$w^T (y - z) = 1$$

and

$$w^T (y - z) = \|y - z\| \|w\|$$

which gives:

$$\|y - z\| = 1 / \|w\|$$

- From the previous analysis we get the following optimization problem:

$$\max_{w,b} \frac{1}{\|w\|} \quad \text{OR} \quad \min_{w,b} \|w\|^2$$

such that:

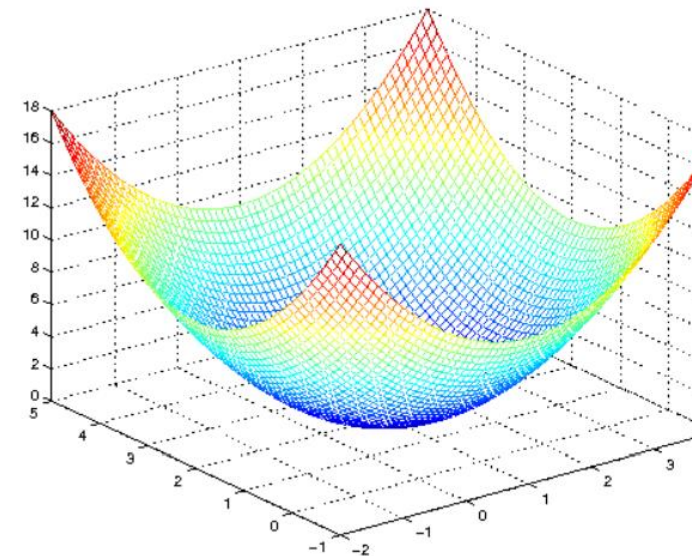
$$y^{(i)}(w^T x^{(i)} + b) \geq 1, \text{ for all } i$$

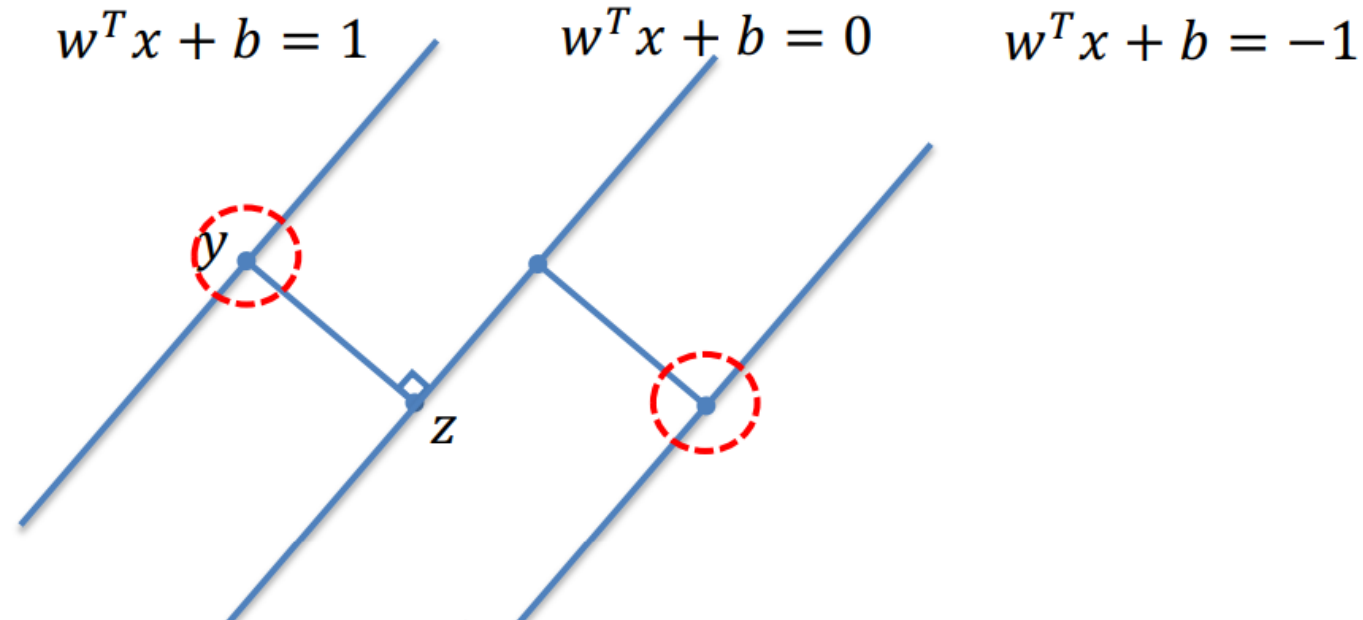
$$\min_{w,b} \|w\|^2$$

such that:

$$y^{(i)}(w^T x^{(i)} + b) \geq 1, \text{ for all } i$$

- This is a standard **quadratic programming problem!**
  - Convex optimization problems





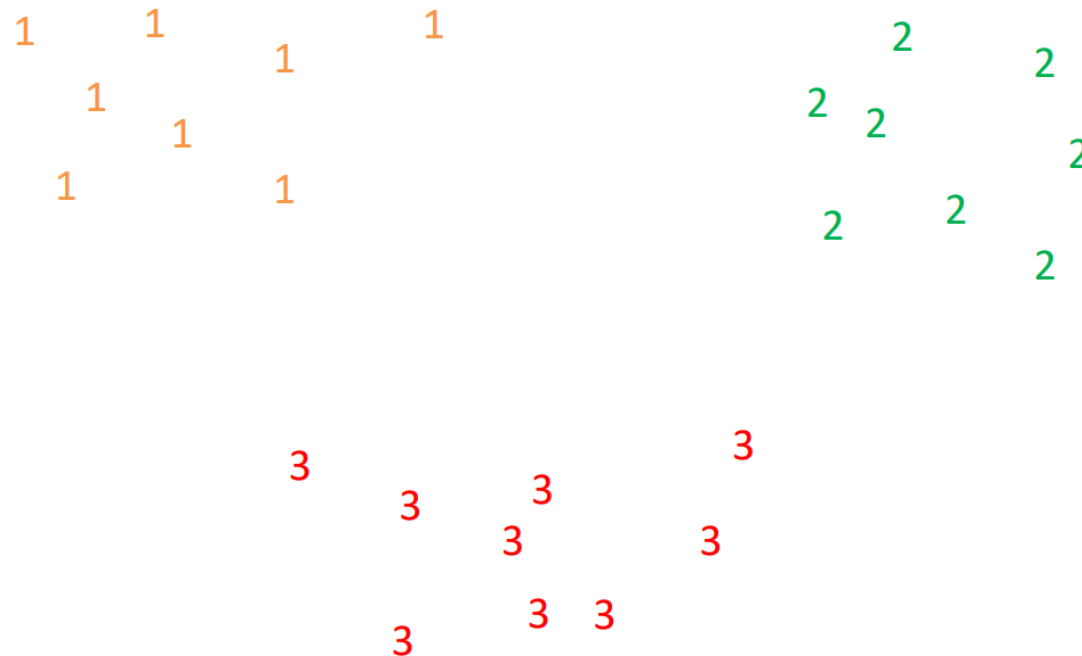
- Where does the name come from?
  - The set of data points such that:  $y^{(i)}(w^T x^{(i)} + b) = 1$  are called **support vectors**.
  - The SVM classifier is completely determined by the support vectors (the **other data points don't influence the algorithm**).

- **Assumptions made so far:**
  - The data is linearly separable!
  - We are dealing with binary classification tasks!



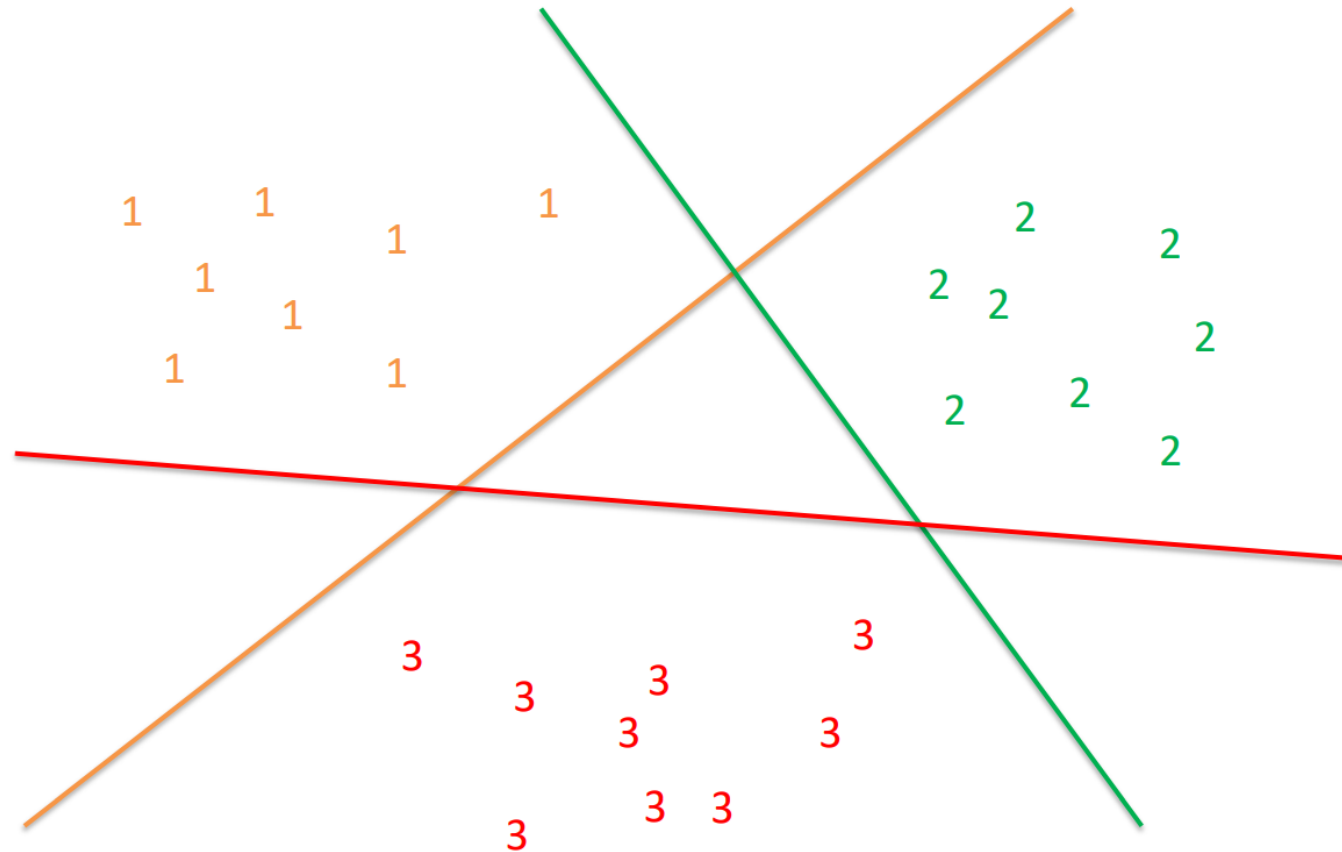
# SVMs – Multiclass Classification

- What if we want to do more than just binary classification
  - for instance if  $y \in \{1,2,3\}$



# One-Versus-All SVMs

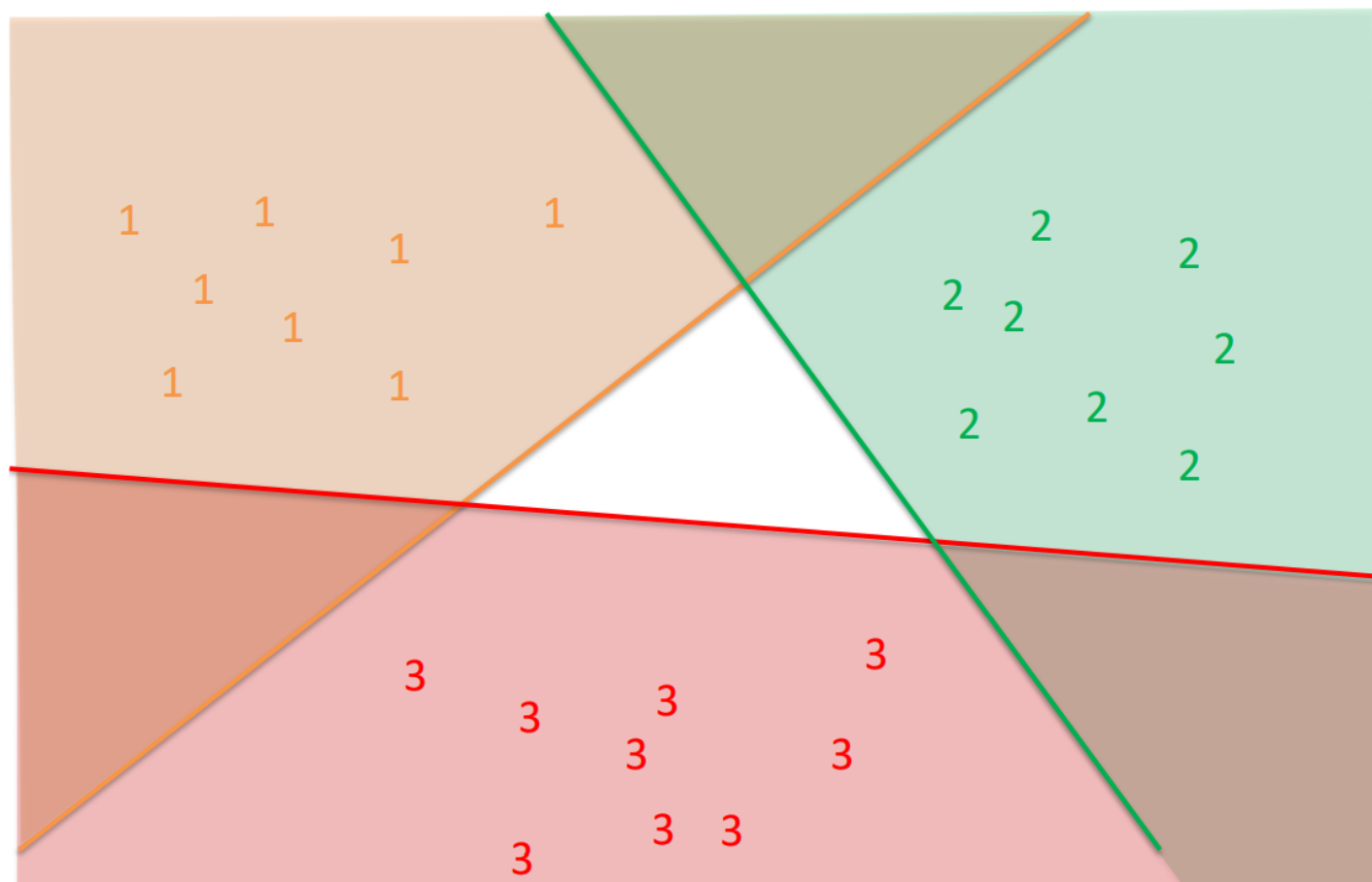
- What if we want to do more than just binary classification
  - for instance if  $y \in \{1,2,3\}$



# One-Versus-All SVMs

- What if we want to do more than just binary classification
  - for instance if  $y \in \{1,2,3\}$

**Regions correctly  
classified by exactly  
one classifier.**



# One-Versus-All SVMs

- **Compute a classifier for each label versus all other labels;**
- Let  $f^k(x) = w^{(k)T}x + b^{(k)}$  be the classifier for the  $k^{\text{th}}$  label
- For a new datapoint  $x$ , classify it as:

$$k' \in \operatorname{argmax}_k f^k(x)$$

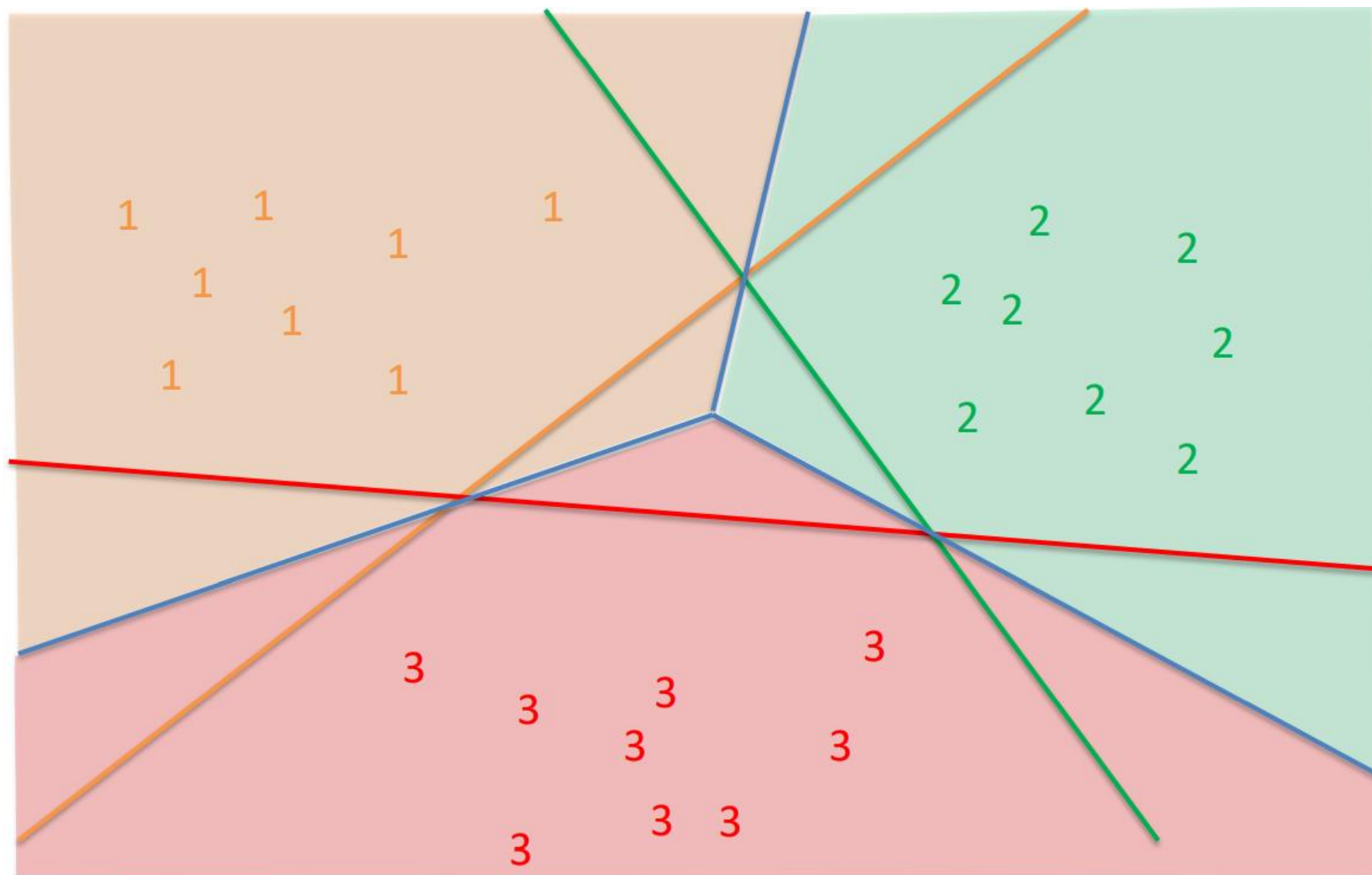
- Drawbacks:
  - If there are  $L$  possible labels, requires **learning  $L$  classifiers** over the entire dataset.

# One-Versus-All SVMs



Regions in  
which points are  
classified by the  
highest value of:

$$w^T x + b$$

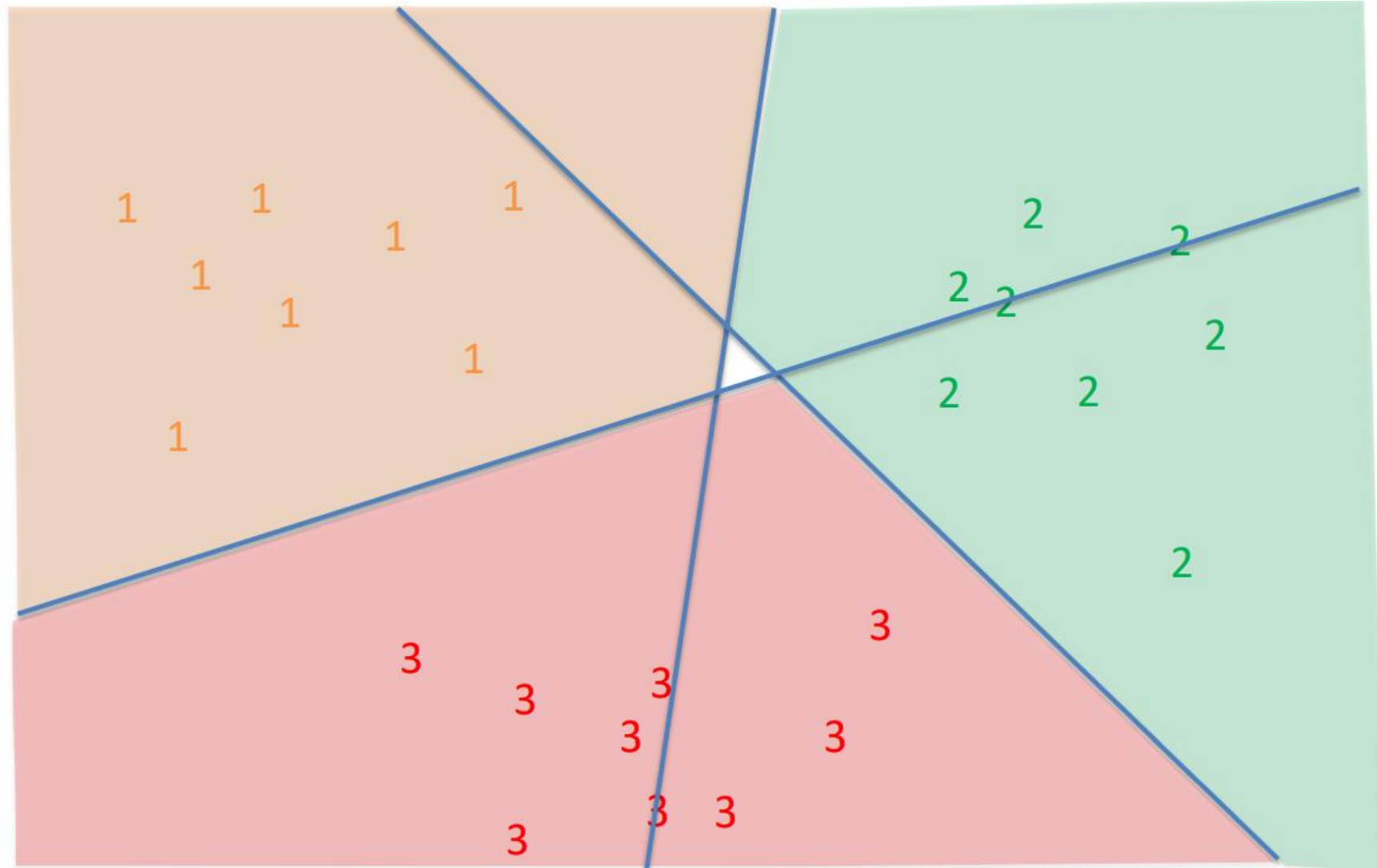


# One-Versus-One SVMs

- Alternatively, it is possible to learn a classifier for **all possible pairs of labels**;
- Given a new data point, it will be classified by **majority vote** (by finding the most common label among all possible classifiers);
- If there are  $L$  labels, requires learning  $\frac{L(L-1)}{2}$  different classifiers using different fractions of the data.
- **Drawbacks:**
  - Can overfit some pairs of labels;
  - Computationally expensive.

# One-Versus-All SVMs

Regions determined  
by majority vote  
over all classifiers



# Resources



- <https://www.youtube.com/watch?v=efR1C6CvhmE>