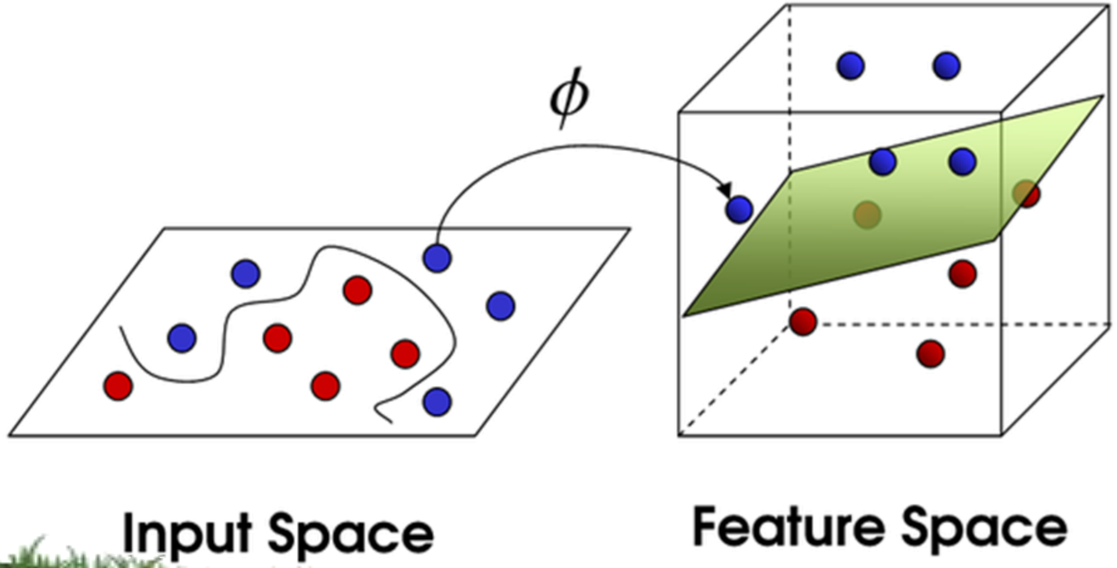
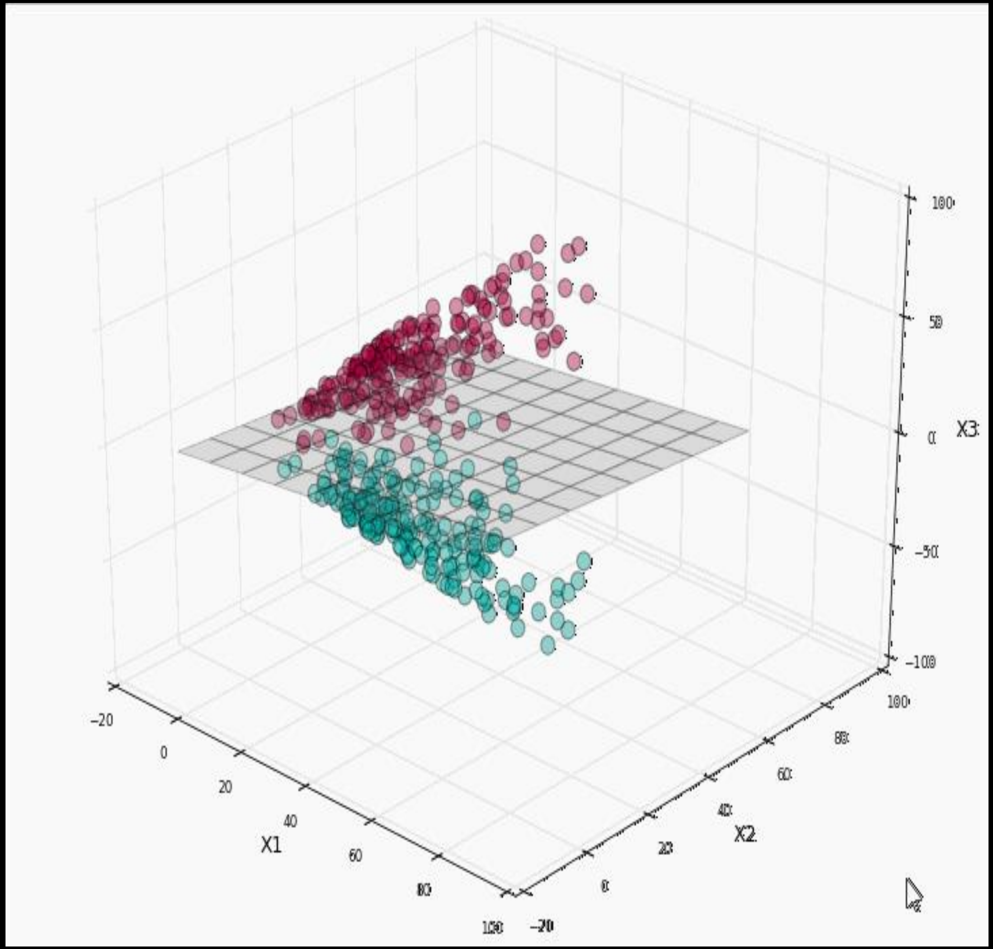
Module – 5:

Support Vector Machines:



Prepared by: Gloriya Mathew, Asst. Professor, Amal Jyothi College of Engineering

Support Vector Machines:



Support Vector Machines:

- ✓ Review of finite dimensional vector spaces
- ✓ Hyper planes
- ✓ Support Vector Classifier.
- ✓ Kernel methods
  - ➢ Gaussian kernel
  - ➢ Multi class SVM.

Support Vector Machine(SVM)

- A supervised machine learning algorithm
- Used for both classification & regression .

Support Vector Machines

- A Support Vector Machine (SVM) can be imagined as a surface that creates a boundary between points of data plotted in multidimensional that represent examples and their feature values.
- Goal of SVM:
  - To create a flat boundary called a hyperplane, which divides the space to create fairly homogeneous partitions on either side.

Support Vector Machines(cntd..)

- SVM learning combines :
  - Instance-based nearest neighbor learning &
  - Linear regression modeling .
- Extremely powerful.
- model highly complex relationships.

Support Vector Machines(cntd..)

- Adapted for use with any type of learning task:
  - Classification
  - Numeric prediction.
  - Pattern recognition.

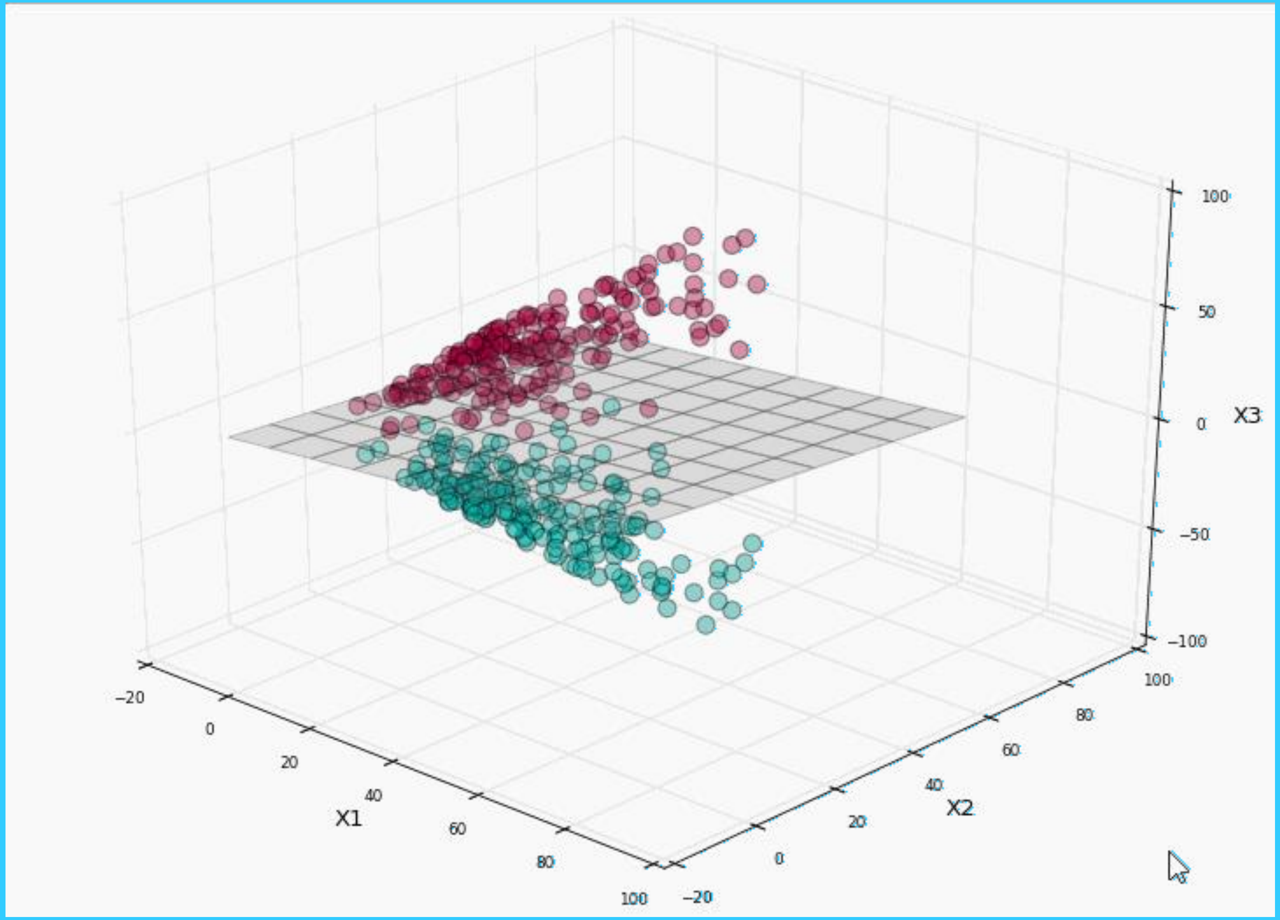
Support Vector Machines - Applications :

- **Gene Expression Data Classification:**
  - In the field of bioinformatics to identify cancer or other genetic diseases.
- **Text categorization:**
  - Identification of the language used in a document or the classification of documents by subject matter.
- **The detection of rare yet important events:**
  - Earthquakes, combustion engine failure, or security breaches.

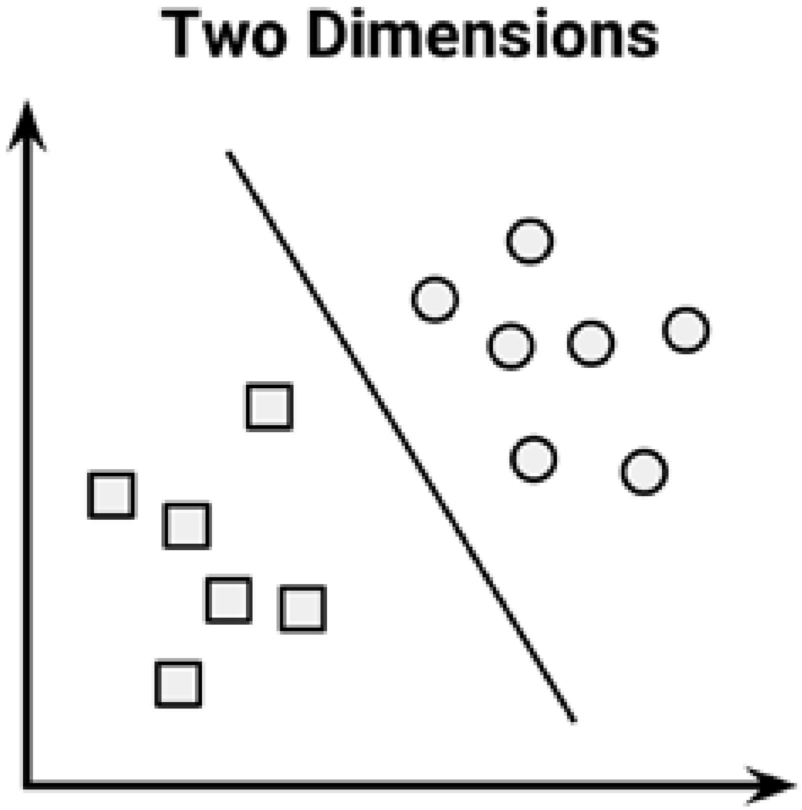
Hyperplanes

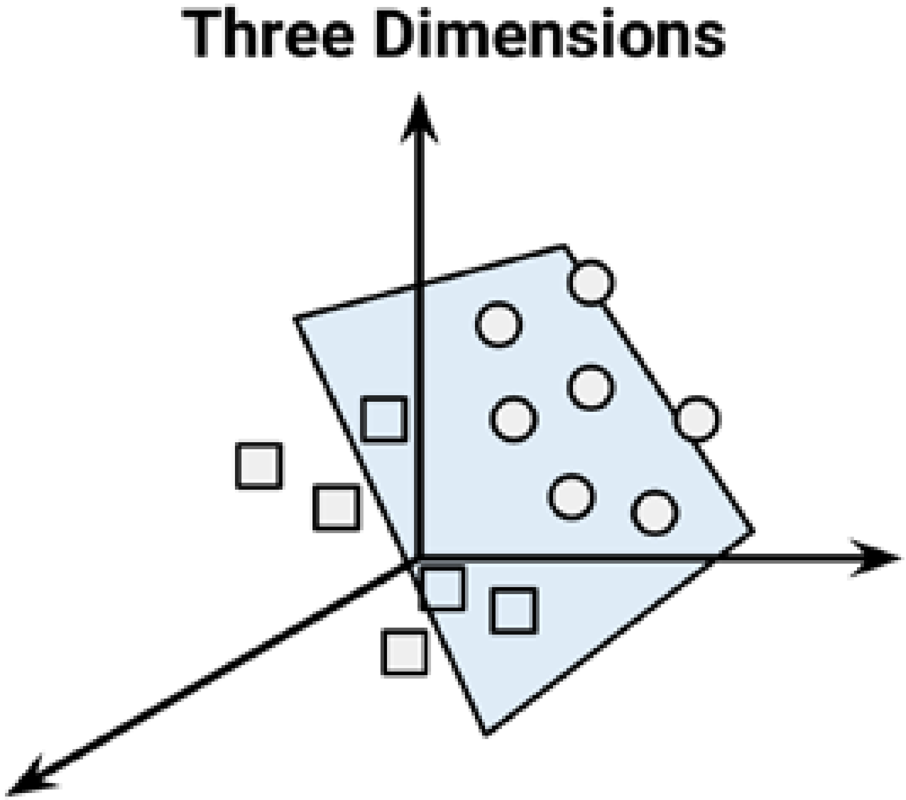
- ***A boundary which partitions the data into groups of similar class values.***
- **SVMs uses hyperplanes to partition data into groups of similar class values.**

Hyperplanes,..



Eg: Hyperplanes - Separate groups of circles and squares in two and three dimensions.





Linearly Separable:

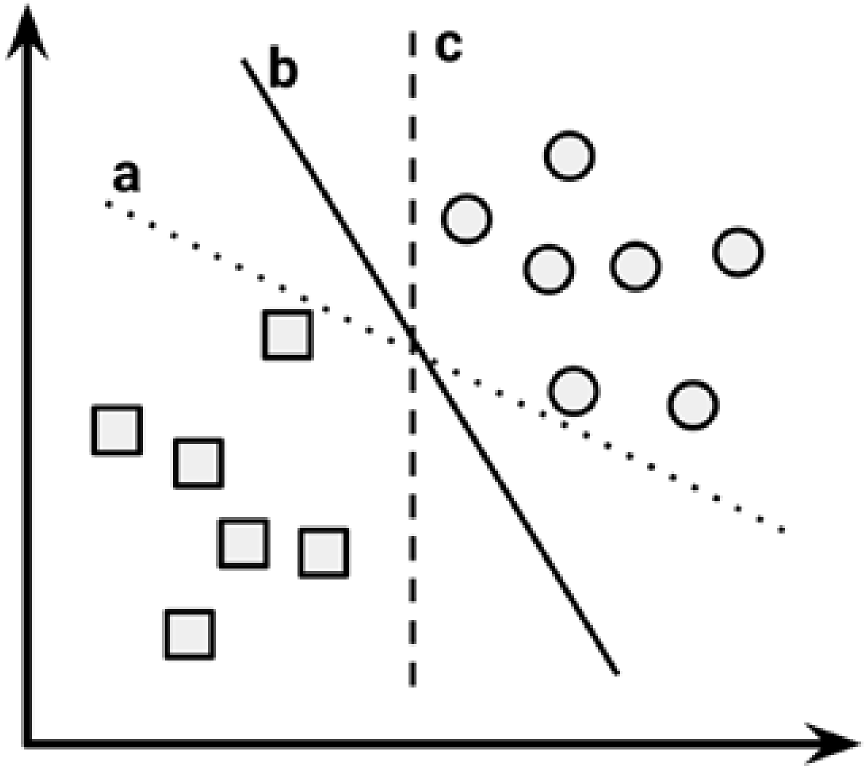
- **Circles and squares can be separated perfectly by the straight line or flat surface, they are said to be Linearly Separable.**

Hyperplanes(cntd..)

- **In Two Dimensions:**
  - **The task of the SVM algorithm is to identify a line that separates the two classes.**

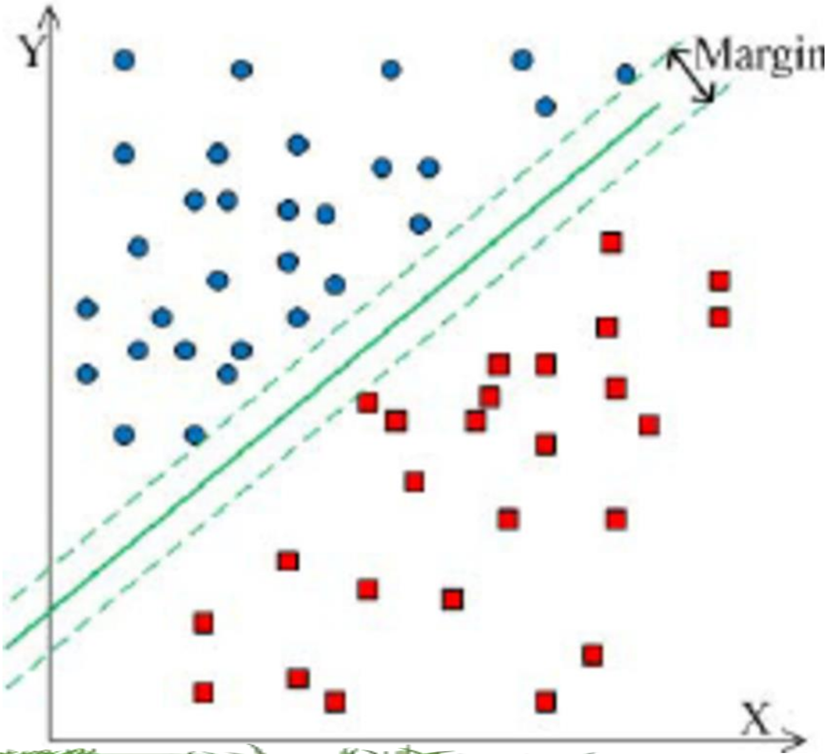
Hyperplanes(cntd..)

- More than one choice of dividing line between the groups of circles and squares.
- Three such possibilities are labeled a, b, and c.
- How does the algorithm choose?
  - Maximum Margin Hyperplane

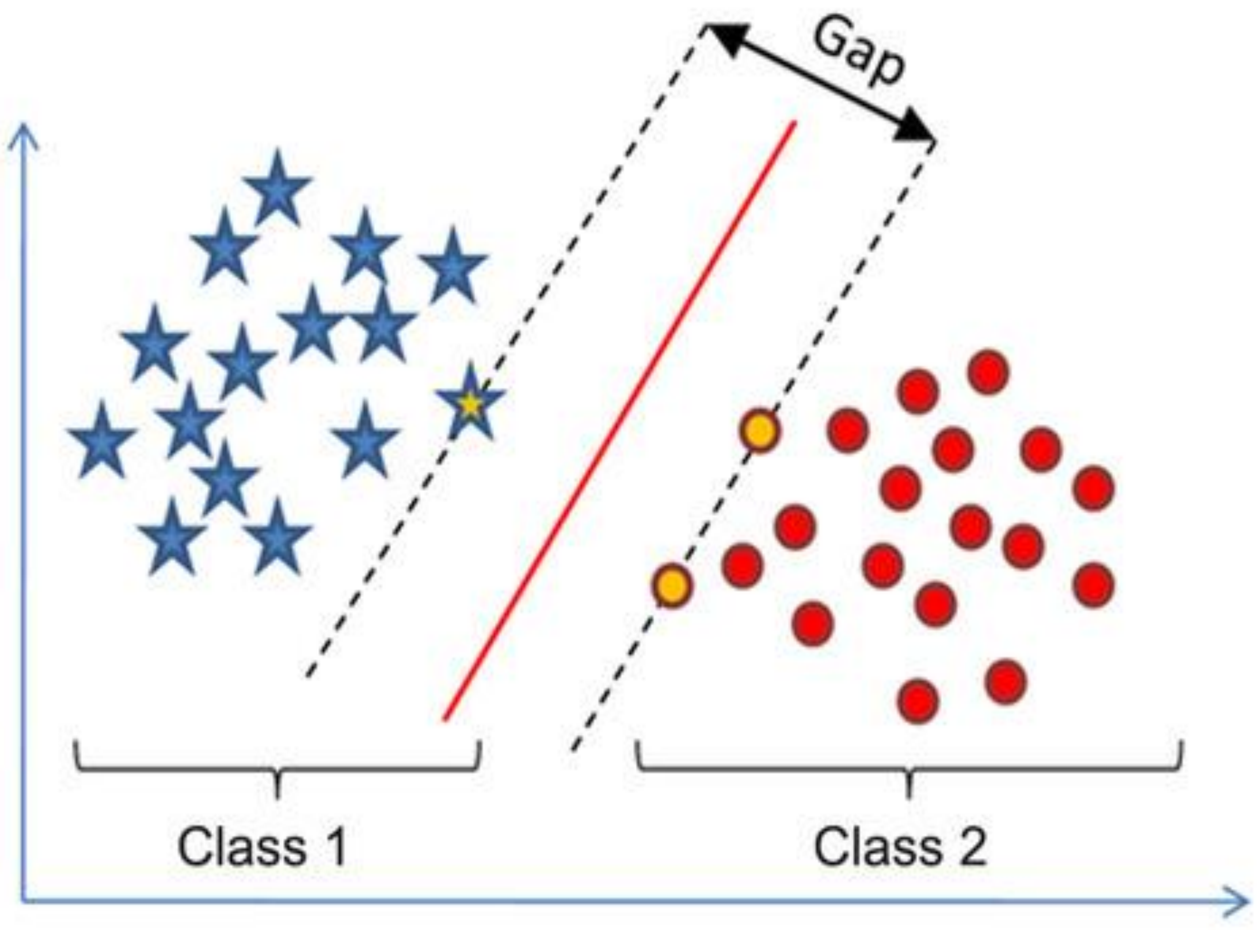


Maximum Margin Hyperplane(MMH)

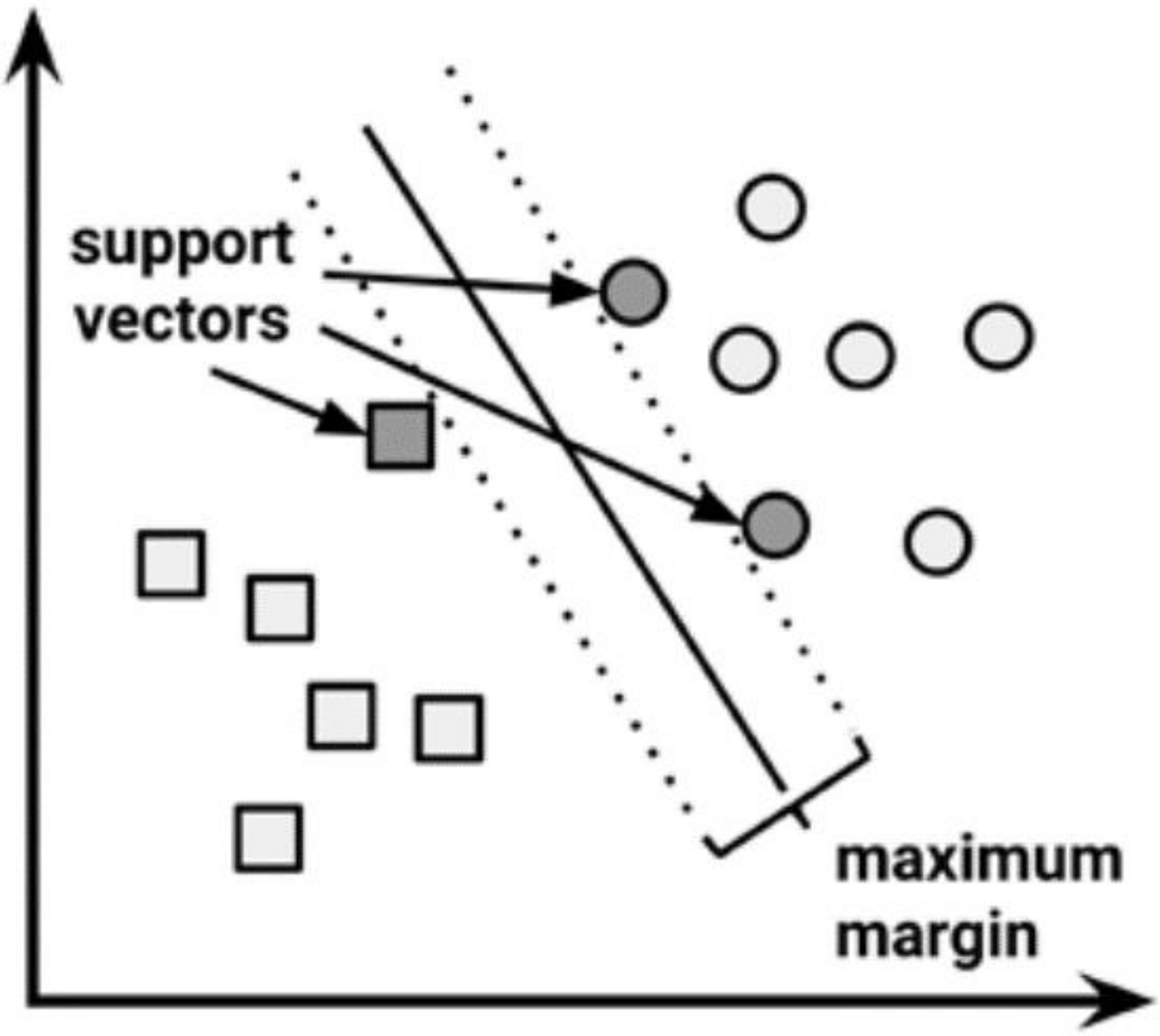
- Creates the greatest separation between the two classes.
- Generalize the best to the future data.
- Improve the chance that, incase of random noise, the points will remain on the correct side of the boundary.



Maximum Margin Hyperplane(MMH)



Maximum Margin Hyperplane(MMH)

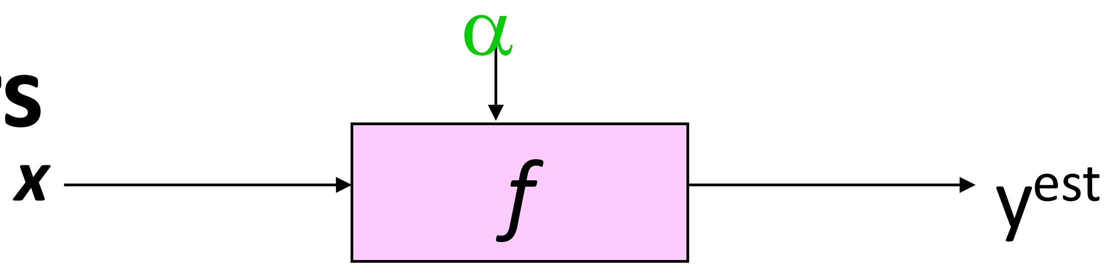


Maximum Margin Hyperplane(MMH) (cntd..)

Support Vectors **:**

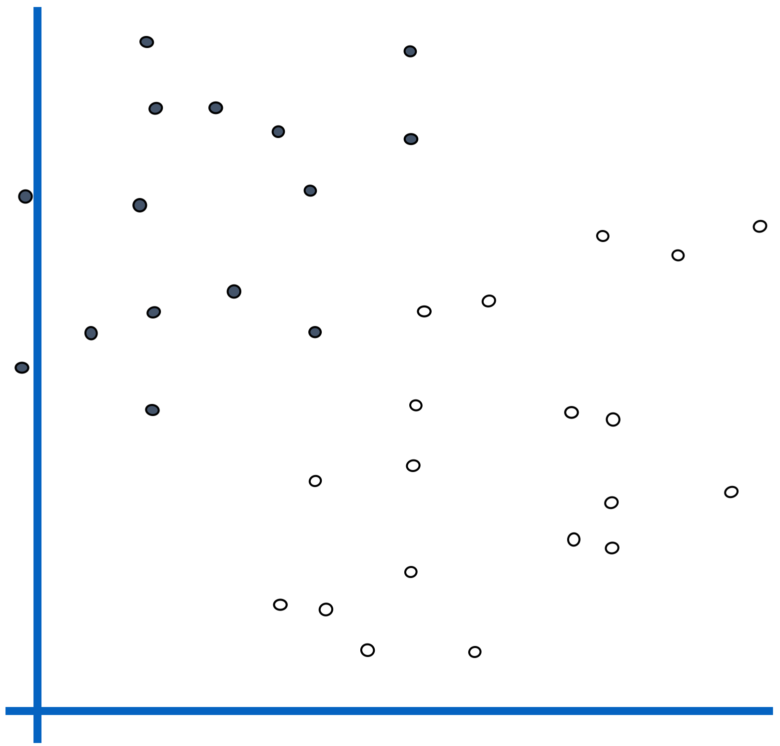
- **Points from each class that are the closest to the MMH;**
- **Each class must have at least one support vector, but it is possible to have more than one.**
- **Using the support vectors alone, it is possible to define the MMH.**
  - **This is a key feature of SVMs;**

Linear Classifiers



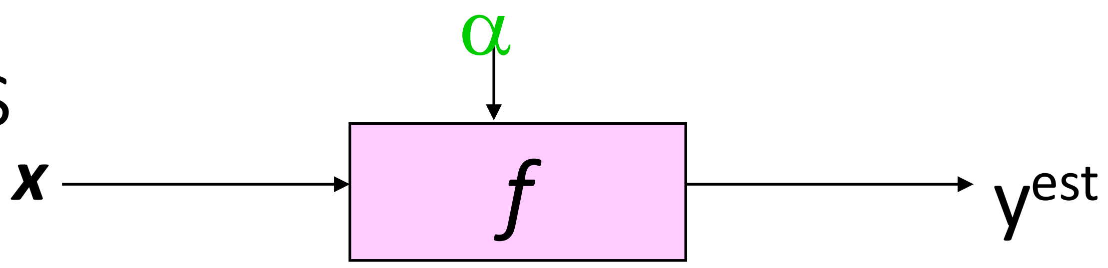
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

- denotes +1
- denotes -1

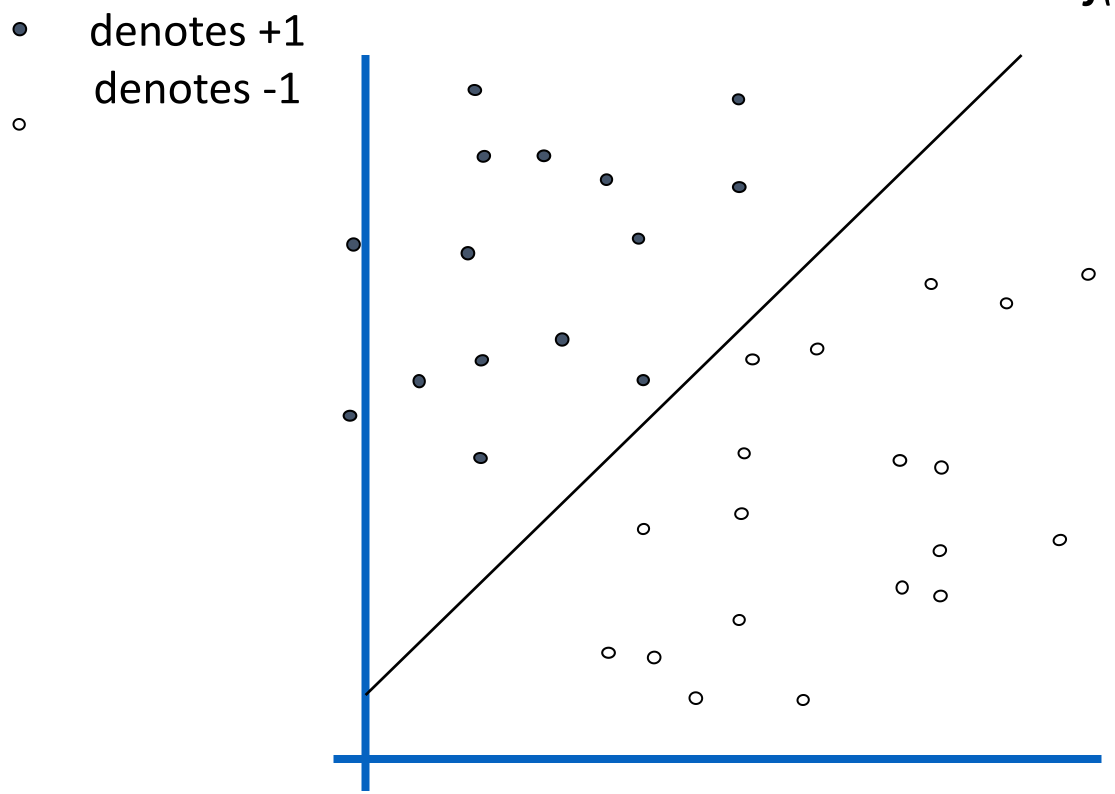


How would you classify this data?

Linear Classifiers

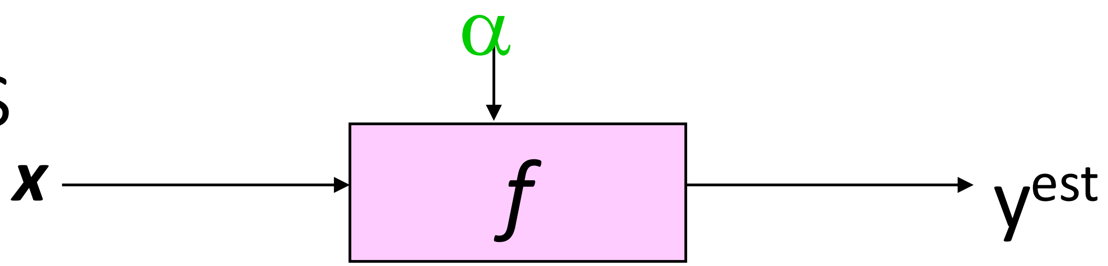


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$



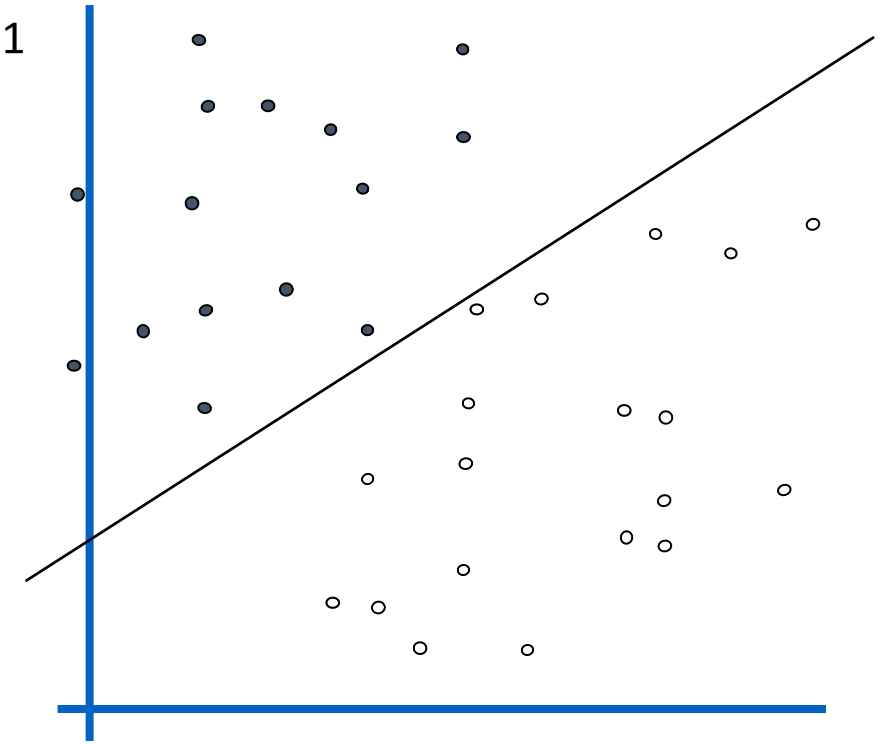
How would you classify this data?

Linear Classifiers



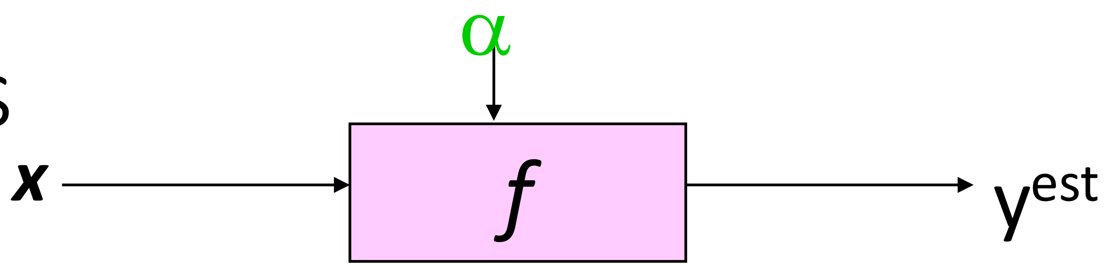
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

- denotes +1
- denotes -1



How would you classify this data?

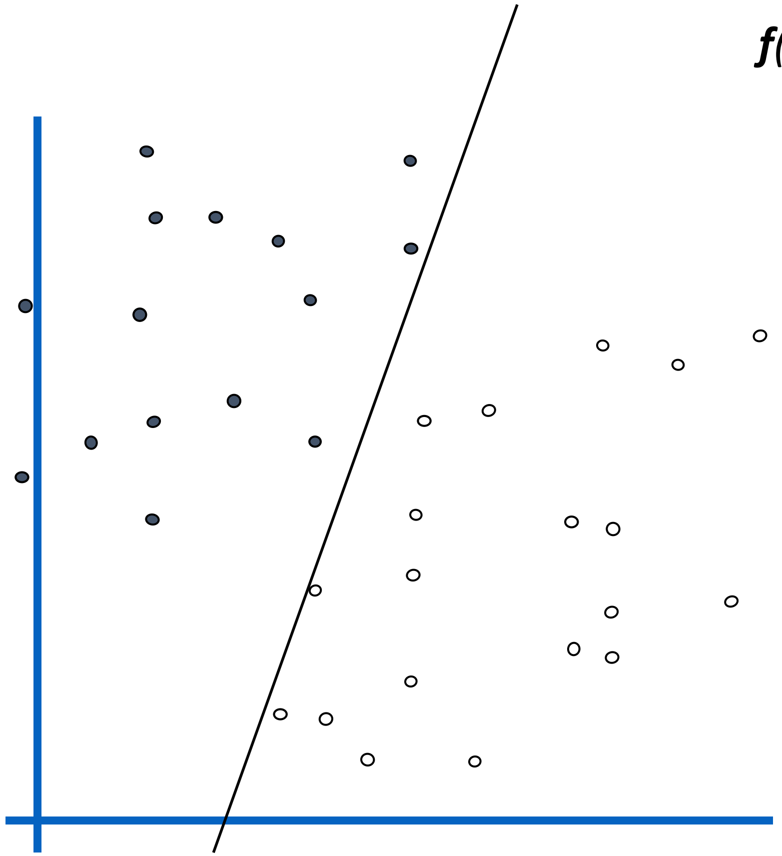
Linear Classifiers



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

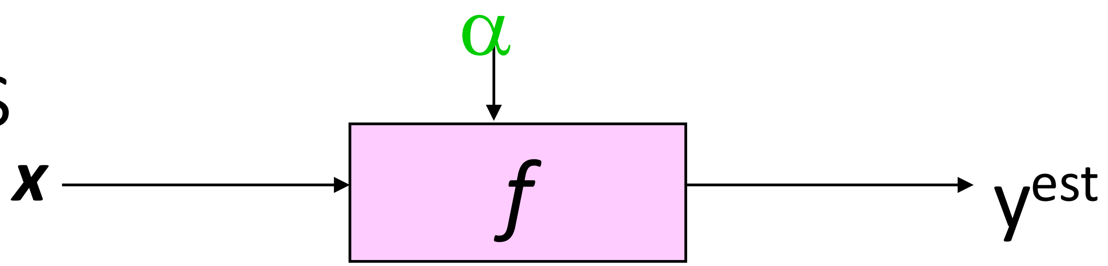
● denotes +1

○ denotes -1



How would you classify this data?

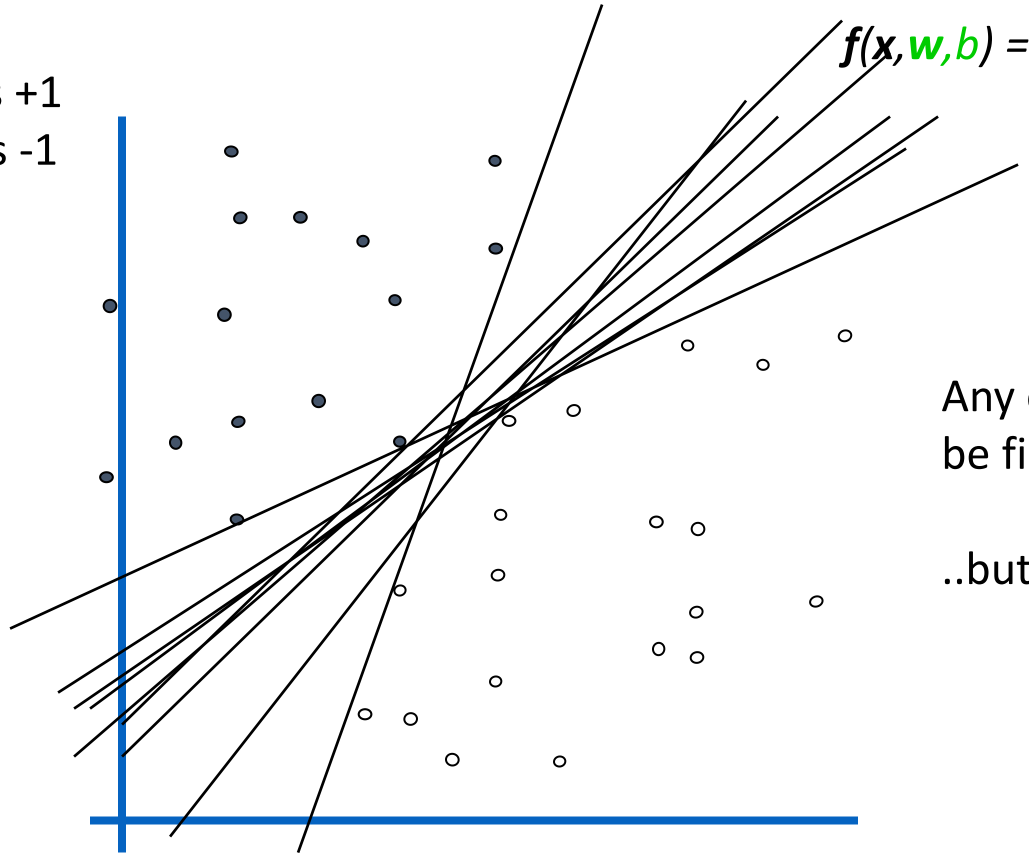
Linear Classifiers



$f(\boldsymbol{x}, \boldsymbol{w}, b) = sign(\boldsymbol{w} \cdot \boldsymbol{x} - b)$

● denotes +1

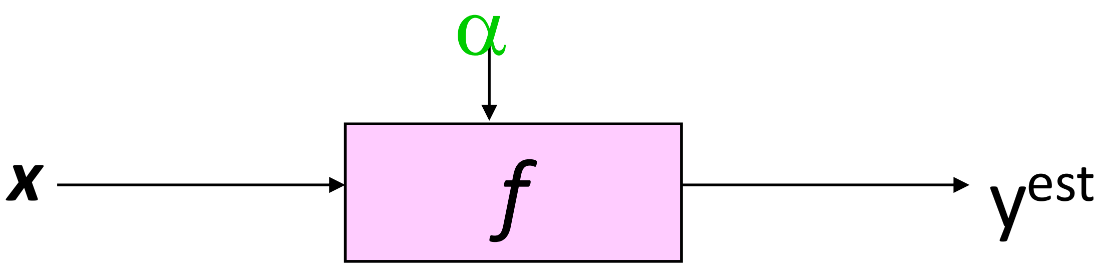
○ denotes -1



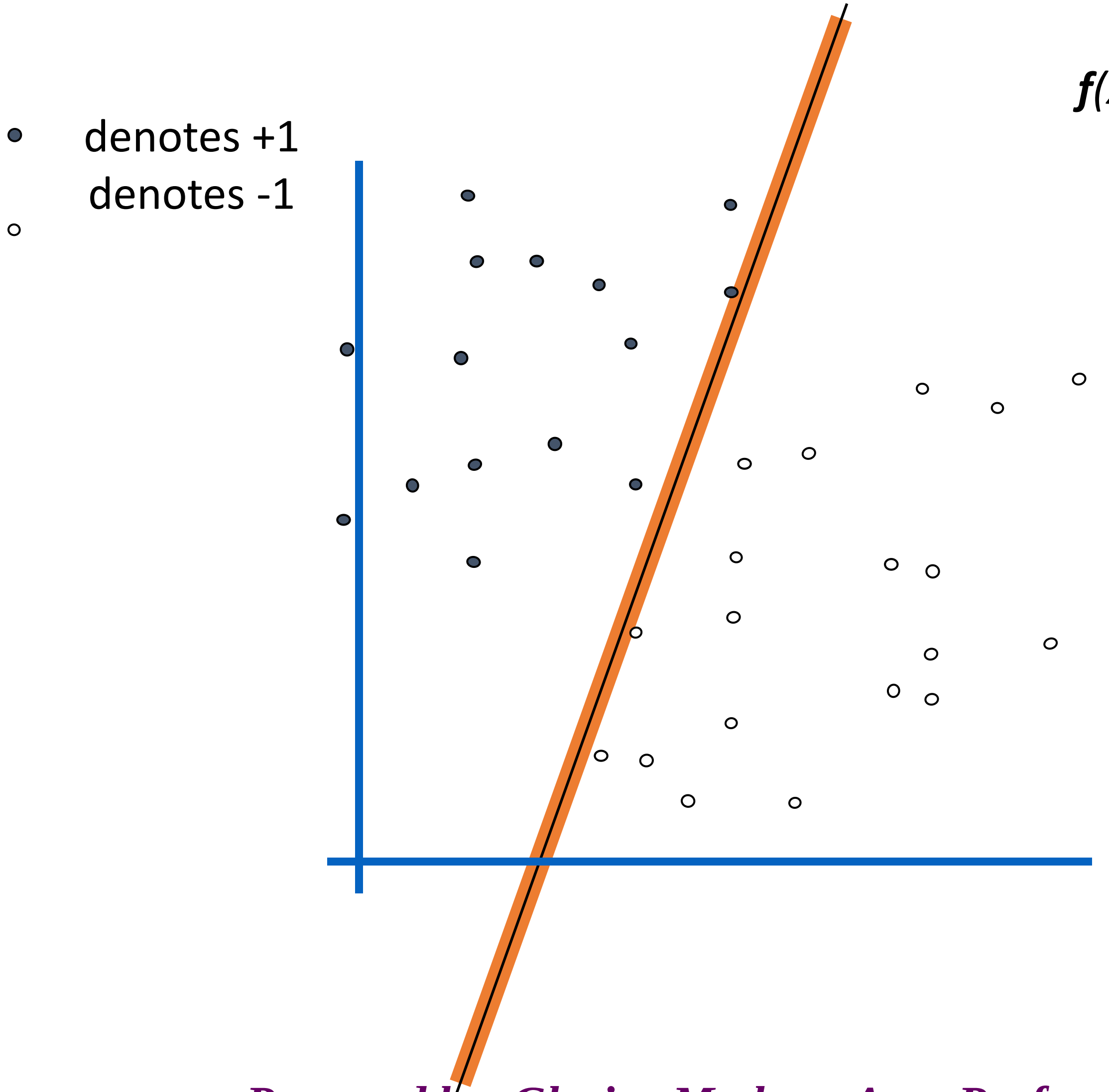
Any of these would be fine..

..but which is best?

Classifier Margin

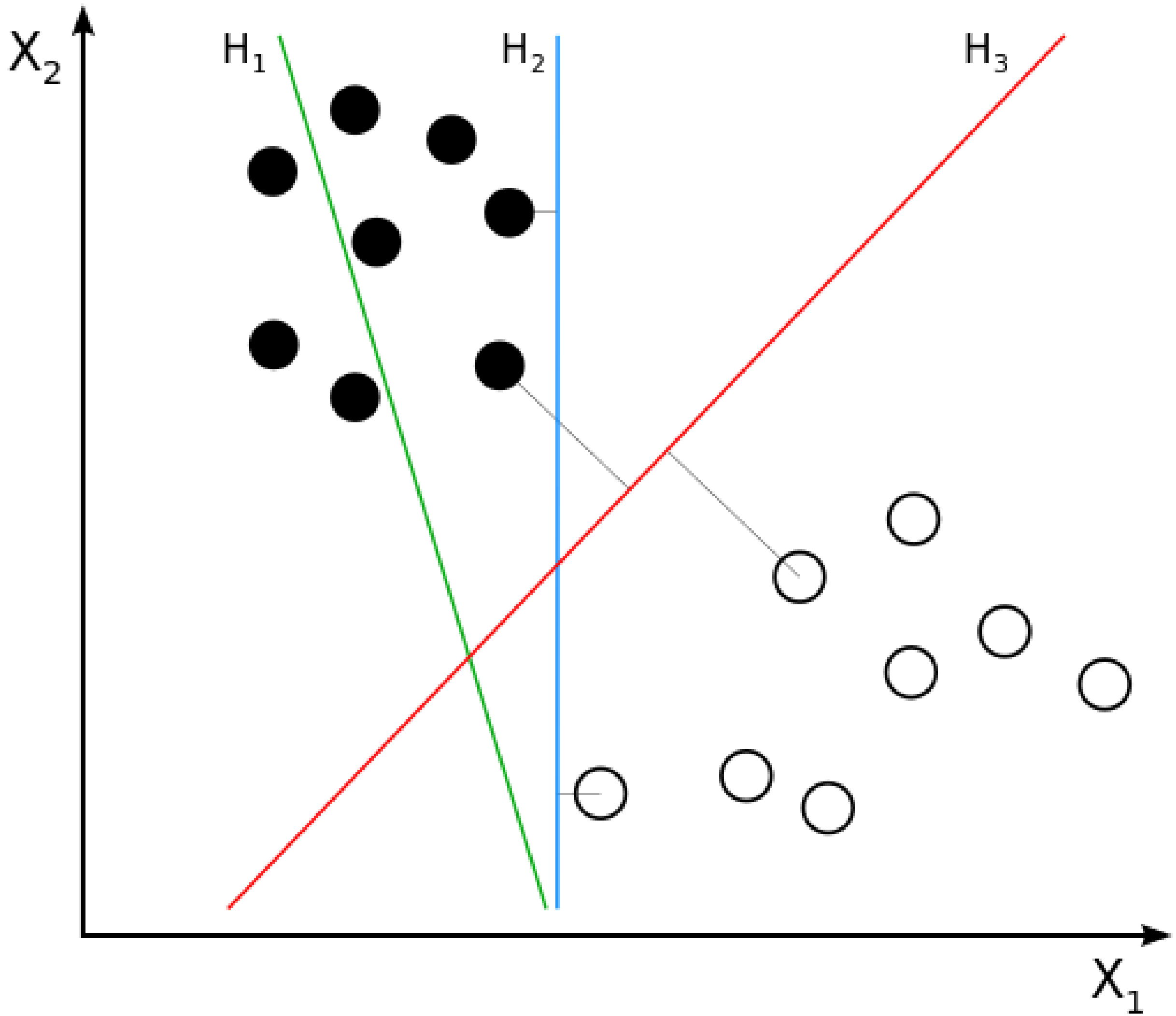


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$



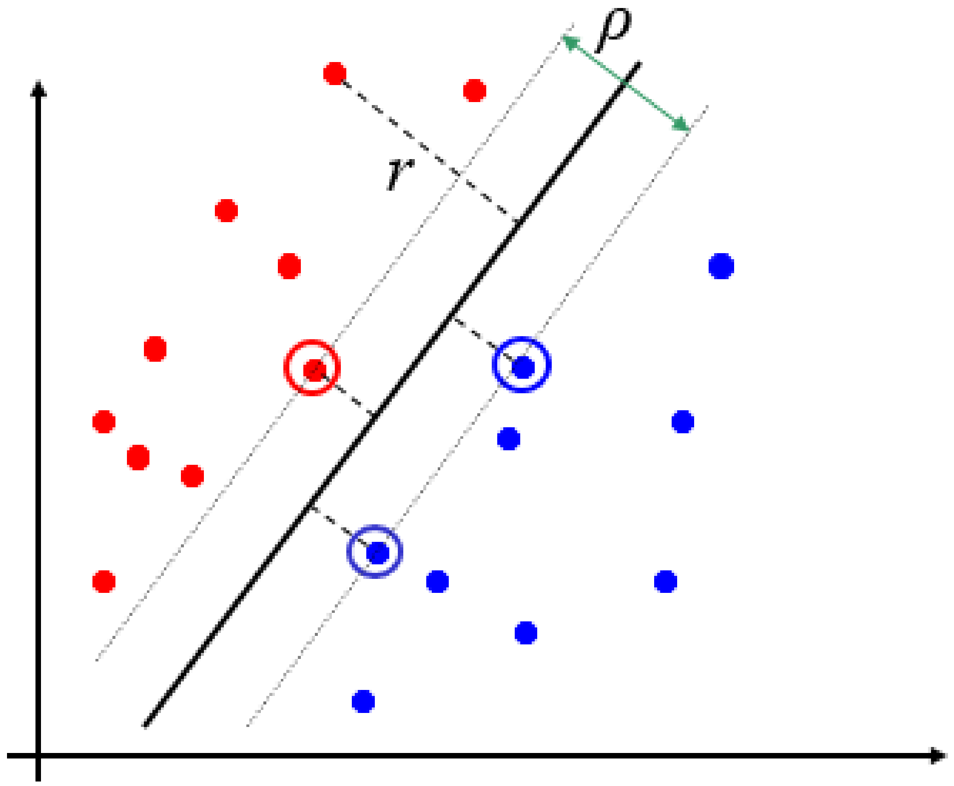
Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Maximum Margin Hyperplane(MMH) (cntd..)



- **H1** does **not separate** the classes.
- **H2** does, but **only with a small margin.**
- **H3 separates** them **with** the **maximum margin.**

- **Examples closest** **to the** **hyperplane** **are** ***support vectors*****.**
- ***Margin ρ*** **of the separator is the** **distance between support vectors.**



MMH-In The case of linearly separable data

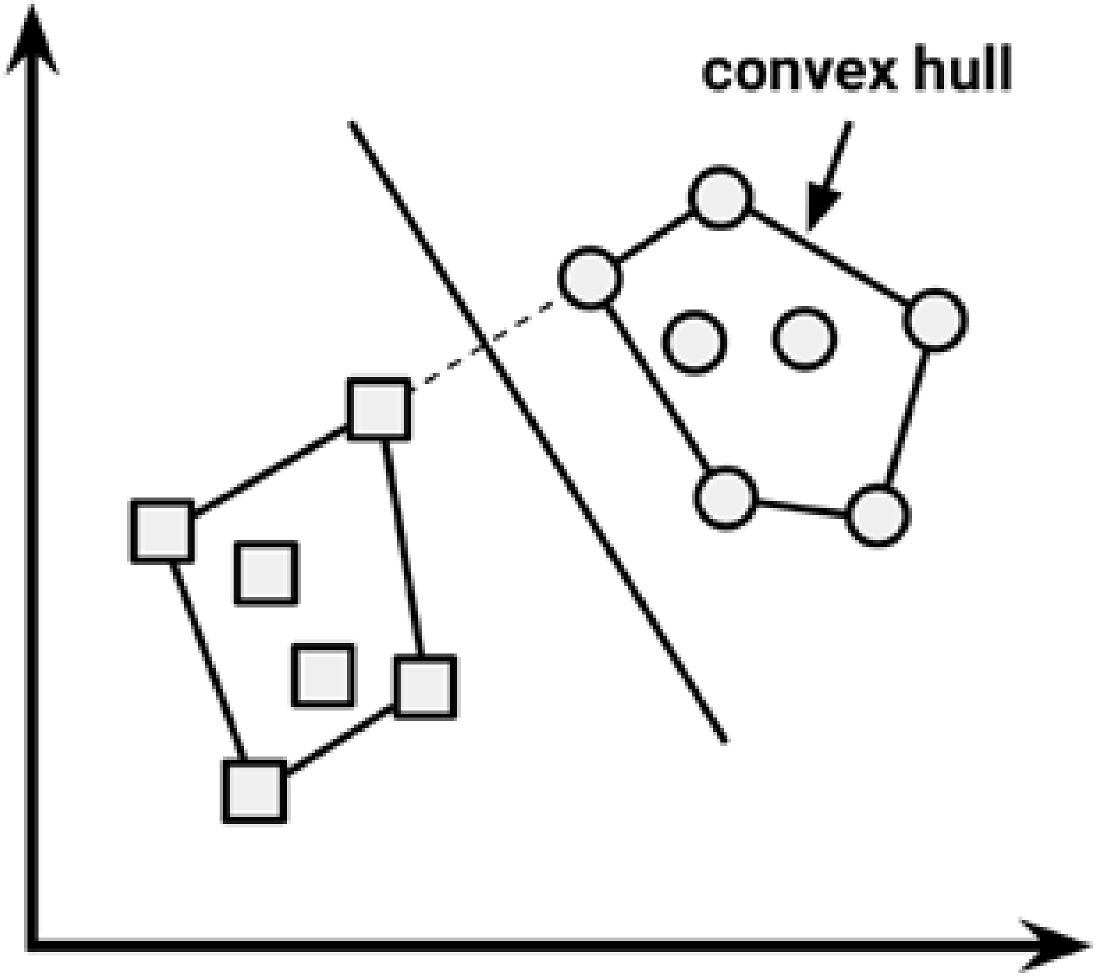
Convex Hull:

- **outer boundaries** **of the two groups of data points are known as the Convex Hull.**

1) The MMH is the perpendicular bisector of the shortest line between the two convex hulls.

- Sophisticated computer algorithms that use a technique known as quadratic optimization are capable of finding the maximum margin in this way.

MMH-In The case of linearly separable data(cntd..)



MMH-In The case of linearly separable data(cntd..)

2) An alternative (but equivalent) approach involves:

- A search through the space of every possible hyperplane in order to find a set of 2- parallel planes that divide the points into homogeneous groups yet themselves are as far apart as possible.

MMH-In The case of linearly separable data(cntd..)

- To understand this search process, we'll need to define exactly what we mean by a hyperplane.
- In *n*-dimensional space, the following equation is used:

$$\vec{w} \cdot \vec{x} + b = 0$$

(Maximum margin Decision Hyperplane)

- Arrows above the letters ➔vectors rather than single numbers.
- *Eg:-* *w* is a vector of *n* weights, that is, $\{w_1, w_2, ..., w_n\}$, &
- *b* ➔ *a* single number (bias).
  - Bias is conceptually equivalent to the intercept term in the slope-intercept form.

MMH-In The case of linearly separable data(cntd..)

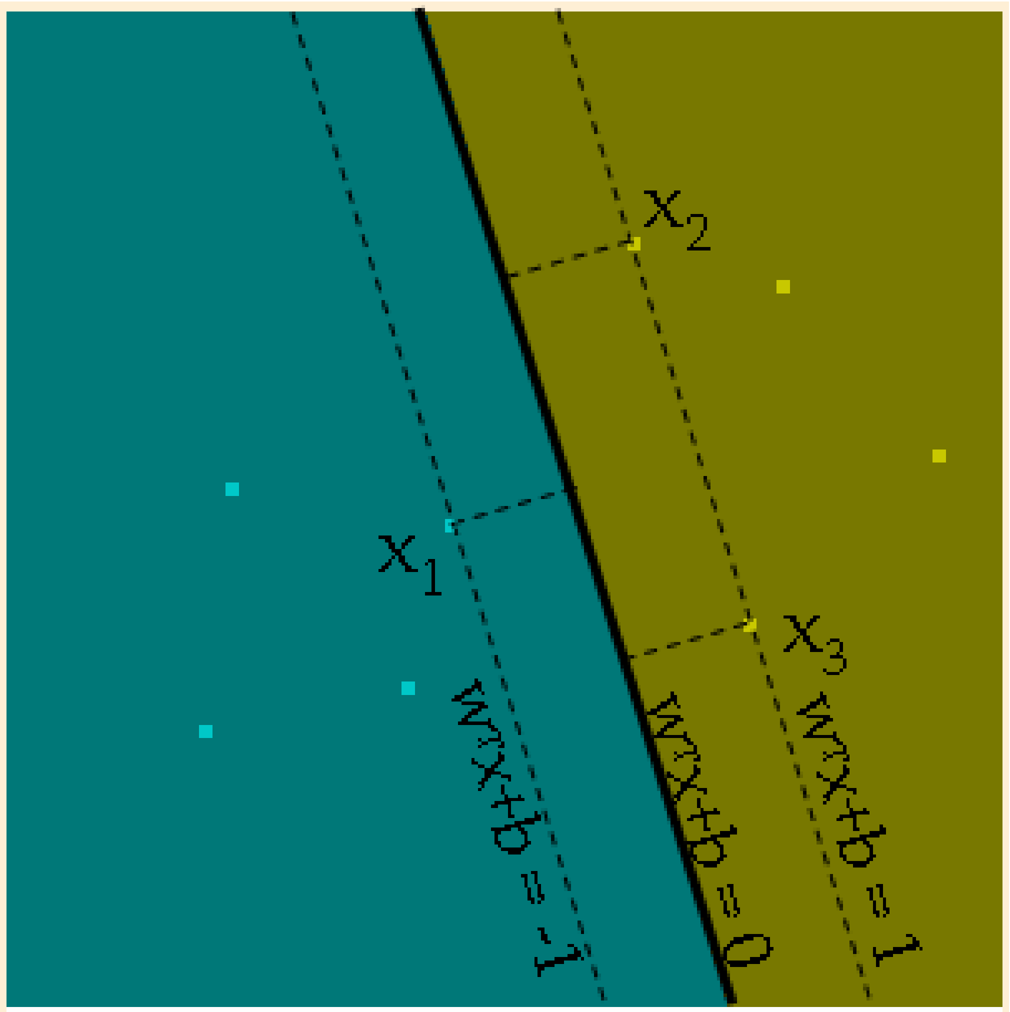
- Using this formula, the goal of the process is to find a set of weights that specify two hyperplanes, as follows:

$$\vec{w} \cdot \vec{x} + b \geq +1$$

$$\vec{w} \cdot \vec{x} + b \leq -1$$

- w → weight vector
- x → input vector
- b → bias

MMH-In The case of linearly separable data(cntd..)



MMH-In The case of linearly separable data(cntd..)

- Hyperplanes are specified such that:
  - All the points of one class fall above the first hyperplane &
  - All the points of the other class fall beneath the second hyperplane.
  - This is possible so long as the data are linearly separable.

MMH-In The case of linearly separable data(cntd..)

- Distance between these two planes as:
- D $= \dfrac{2}{||\vec{w}||}$
- ||w|| →Euclidean norm (the distance from the origin to vector *w*).
- To maximize distance, we need to minimize ||w||.

MMH-In The case of linearly separable data(cntd..)

- **The task is typically reexpressed as a set of constraints, as follows:**

$$\min \frac{1}{2} \|\vec{w}\|^2$$
$$s.t. \quad y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1, \forall \vec{x}_i$$

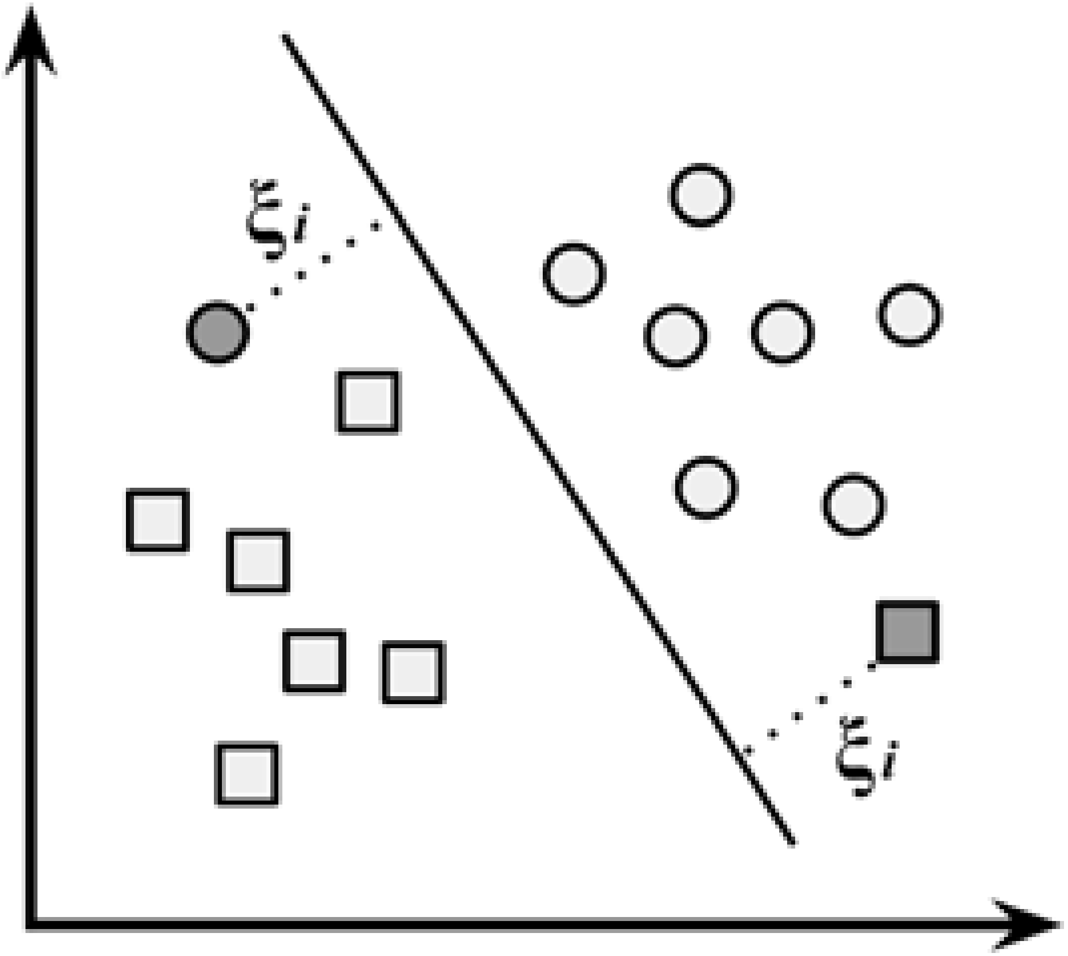
- First line→ Minimize the Euclidean norm (squared and divided by two to make the calculation easier).
- Second line → this is subject to (*s.t.*), the condition that each of the y_i data points is correctly classified.
- *y* → class value (transformed to either +1 or -1) and
- upside down "A" → "for all."

MMH-The case of nonlinearly separable data

- Use of a Slack Variable;
  - It creates a soft margin that allows some points to fall on the incorrect side of the margin.
  - Slack variables ξi can be added to allow misclassification of difficult or noisy examples, resulting margin called soft.

MMH-The case of nonlinearly separable data(cntd)

- The figure that follows illustrates two points falling on the wrong side of the line with the corresponding slack terms (denoted with the Greek letter Xi):



MMH-The case of nonlinearly separable data(cntd).

- A **cost value** (denoted as C) is applied **to all points** that **violate the constraints**, &
- Rather than finding the maximum margin, the algorithm attempts **to minimize the total cost.**
- Now, **optimization problem is** to:

$$\min \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^{n} \xi_i$$

$$s.t.\ \ y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1 - \xi_i, \forall \vec{x}_i, \xi_i \geq 0$$

MMH-The case of nonlinearly separable data(cntd).

- C → cost parameter.
- Greater the cost parameter → harder the optimization will try to achieve 100 percent separation.
- Lower cost parameter → will place the emphasis on a wider overall margin.
- It is important to strike a balance between these two in order to create a model that generalizes well to future data.

MMH-The case of nonlinearly separable data(cntd)

- Real-world applications, the relationships between variables are nonlinear.

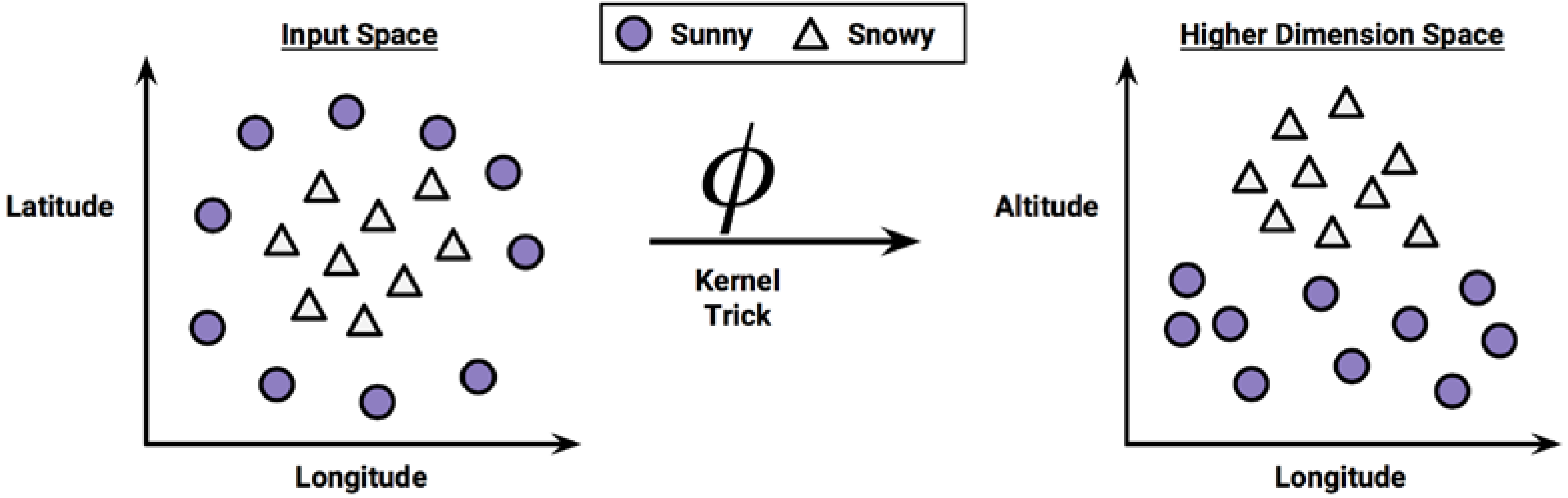
- <u>**Kernel Trick:**</u>

➢ A key feature of SVMs is their ability to map the problem into a higher dimension space.

➢ This is done using a process known as the kernel trick.

  ➢ In doing so, a nonlinear relationship may suddenly appear to be quite linear.

MMH-The case of nonlinearly separable data(cntd)



- Scatterplot on the left depicts a nonlinear relationship between a weather class (sunny or snowy) and two features: latitude and longitude.
- The points at the center of the plot are members of the snowy class, while the points at the margins are all sunny.
- Such data could have been generated from a set of weather reports, some of which were obtained from stations near the top of a mountain, while others were obtained from stations around the base of the mountain.

MMH-The case of nonlinearly separable data(cntd.)

- SVMs with nonlinear kernels, add additional dimensions to the data in order to create Separation.

Kernel Trick:

- **A process of constructing New features that express mathematical relationships between measured Characteristics.**
- A mapping function

- **Eg:- the altitude feature can be expressed mathematically as An interaction between latitude and longitude:**
  - The closer the point is to the center of Each of these scales, the greater the altitude.
  - This allows SVM to learn concepts that Were not explicitly measured in the original data.

Strengths & Weaknesses :- SVMs with nonlinear kernels

Strengths	Weaknesses
• Can be used for classification or numeric prediction problems • Not overly influenced by noisy data and not very prone to overfitting • May be easier to use than neural networks, particularly due to the existence of several well-supported SVM algorithms • Gaining popularity due to its high accuracy and high-profile wins in data mining competitions	• Finding the best model requires testing of various combinations of kernels and model parameters • Can be slow to train, particularly if the input dataset has a large number of features or examples • Results in a complex black box model that is difficult, if not impossible, to interpret

Kernel functions – general form.

- denoted by the Greek letter phi (φ(x))→ a mapping of the data into another space.
- General kernel function applies some transformation to the feature vectors x_i and x_j &
- Combines them using the dot product, which takes two vectors and returns a single number.

$$K(\vec{x_i}, \vec{x_j}) = \phi(\vec{x_i}) \cdot \phi(\vec{x_j})$$

Most Commonly Used Kernel Functions:

- Linear kernel
- Polynomial kernel
- Sigmoid kernel
- Gaussian RBF kernel

➢ Almost all SVM software packages will include these kernels.

1. Linear kernel

- Simplest kernel function
- Does not transform the data at all.
- expressed simply as the dot product of the features.

$$K(\vec{x_i}, \vec{x_j}) = \vec{x_i} \cdot \vec{x_j}$$

2. Polynomial Kernel

- Polynomial kernel of degree *d* adds a simple nonlinear transformation of the data:

$$K(\vec{x_i}, \vec{x_j}) = (\vec{x_i} \cdot \vec{x_j} + 1)^d$$

d → degree of polynomial

3. Sigmoid Kernel

- **Results in an SVM model, somewhat analogous to a neural network using a sigmoid activation function.**
- **The Greek letters kappa and delta are used as kernel parameters:**

$$K(\vec{x_i}, \vec{x_j}) = \tanh(\kappa\, \vec{x_i} \cdot \vec{x_j} - \delta)$$

- "tanh” → **hyperbolic tangent function**

Gaussian RBF kernel

- General-purpose kernel;
- Similar to a RBF neural network.
- The RBF kernel performs well on many types of data &
- a reasonable starting point for many learning tasks:

$$K(\vec{x}_i, \vec{x}_j) = e^{\frac{-||\vec{x}_i - \vec{x}_j||^2}{2\sigma^2}}$$

- Sigma →adjustable parameter (plays a major role in the performance of the kernel)

How to choose kernel?

- No reliable rule to match a kernel to a particular learning task.
- The fit depends on:
  - The concept to be learned
  - The amount of training data and
  - The relationships among the features.
- Choice of kernel is arbitrary
  - Performance may vary slightly.

Multiclass SVM

- **Classification** with **more than two classes**.
- **Extension of two-class linear classifiers to 'J>2' classes.**
- **The method depends on:**
  - **whether the classes are mutually exclusive or not.**
- **2-methods:**
  1. ***Any-of Classification***
  2. ***One-of Classification***
- Text Classification

Multiclass SVM(cntd..)

1. Any-of Classification (Multilabel / Multivalue):

- **Classification for classes that are not mutually exclusive.**
- ***a document can belong to several classes simultaneously, or to a single class, or to none of the classes.***
- **The decision of one classifier has no influence on the decisions of the other classifiers.**

- **Eg:** Text Classification

Any-of Classification (Multilabel / Multivalue) (cntd..)

- Formal definition of the classification problem, we learn J different classifiers - Y_j in any-of classification, each returning either C_j or $\bar{C}_j$:

$$\gamma_j(d) \in \{c_j, \bar{c}_j\}$$

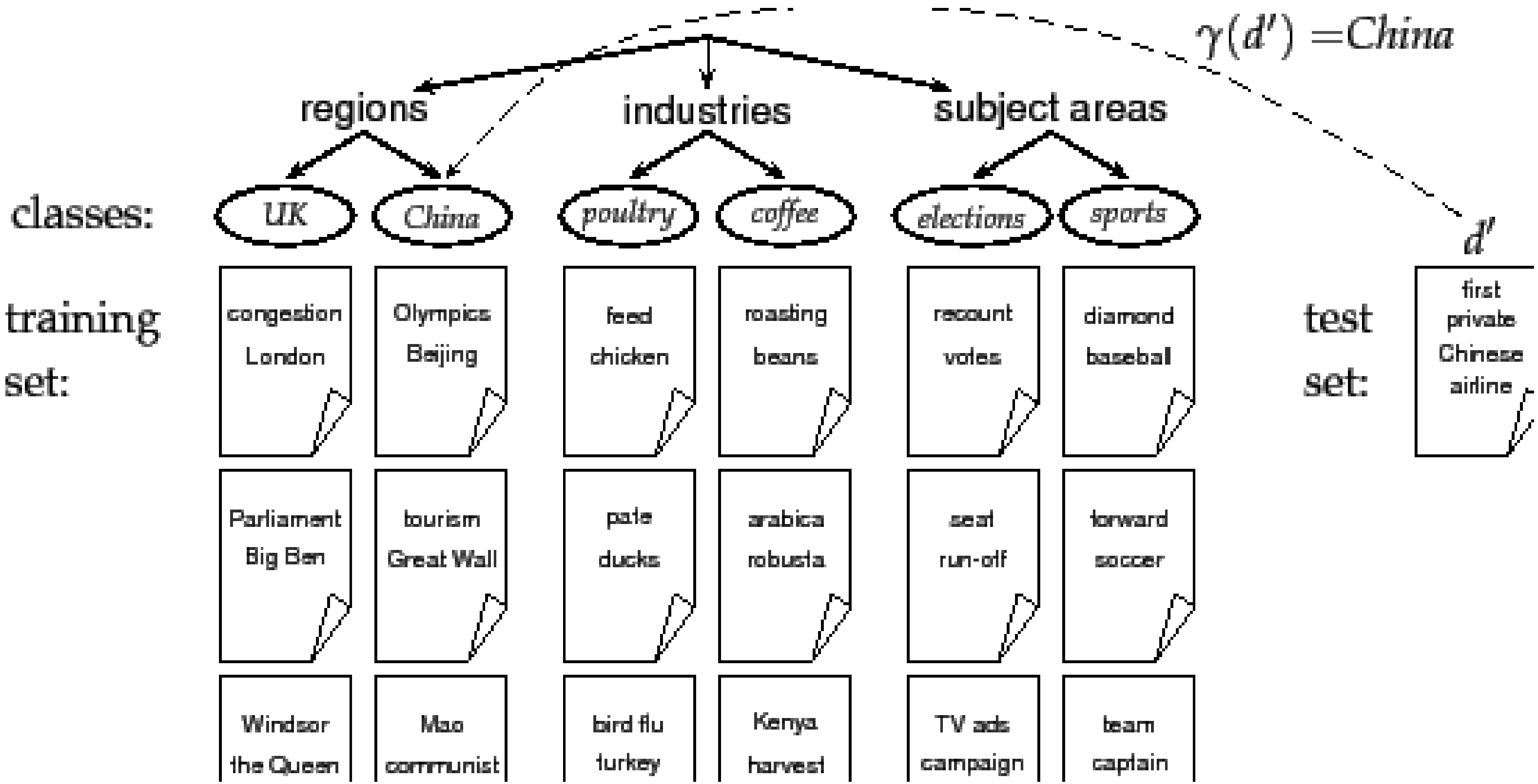


Figure 13.1: Classes, training set, and test set in text classification .

- Doc: 2008 Olympics : → China class and sports class

Any-of Classification

- Eg:-
- a document about the '2008 Olympics' should be a member of 2-classes:
  - China class and
  - sports class.
  - This type of classification problem is referred to as an *any-of* problem

Multiclass SVM(cntd..)

- **Any-of classification - steps:**

1. **Build a classifier for each class:**
   - **Where the training set consists of the set of documents in the class (positive labels) and its complement (negative labels).**

2. **apply each classifier separately – for the Given the test document.**

❖ **The decision of one classifier has no influence on the decisions of the other classifiers.**

Multiclass SVM(cntd..)

2. *one-of classification*:

- **Classes are mutually exclusive.**
- **Each document must belong to exactly one of the classes.**
- **Also called - *multinomial* , *polytomous* , *multiclass* , or *single-label* *classification*.**
- **Formally, there is a single classification function γ in one-of classification whose range is C. i.e., .** $\gamma(d) \in \{c_1, \ldots, c_J\}$ **;**
- **KNN is a (nonlinear) one-of classifier.**
- eg:- a document is a member of exactly one class.

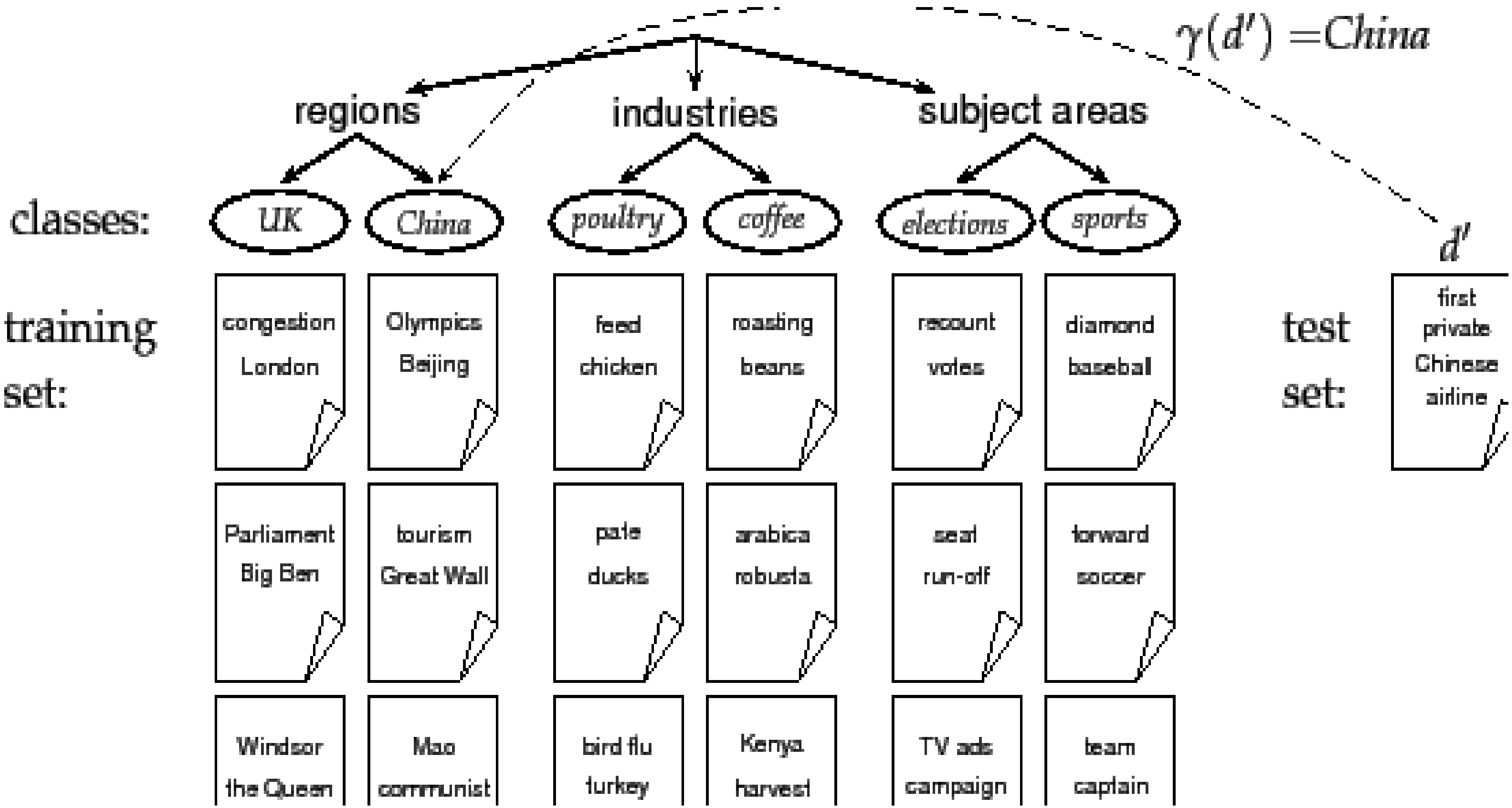


Figure 13.1: Classes, training set, and test set in text classification .

One-of Classification - Steps

1. Build a classifier for each class, where the training set consists of the set of documents in the class (positive labels) and its complement (negative labels).
2. apply each classifier separately - for the Given the test document.
3. Assign the document to the class with:
   - The maximum score
   - The maximum confidence value ,or
   - The maximum probability.

- ***Important Questions: Module-5***

1. **What is meant by a Support Vector?**
2. **How machine learning using Support Vector Machines possible.**
3. **What are the applications of SVM.**
4. **How Classification using hyperplanes is possible?**
5. **What is meant by Maximum Margin Hyperplane?**
6. **What do you meant by a kernel function? Explain the strengths and weaknesses of classification using kernel.**
7. **What are the different types of kernel functions.**
8. **Explain in detail about Multiclass SVM.**