UNIVERSITÀ di VERONA

Machine Learning

Kernel-based Learning methods & Support Vector Machines

Cigdem Beyan

A. Y. 2024/2025

Binary Classification - recap

Given a training data $(\mathbf{x}_i, y_i)$ for *i=1,…N* with $\mathbf{x}_i \in \mathbb{R}^d$ and, $y_i \in \{-1, 1\}$ learn a classifier *f(x)* such that:

$$f(\mathbf{x}_i) \begin{cases} \geq 0 & y_i = +1 \\ < 0 & y_i = -1 \end{cases}$$

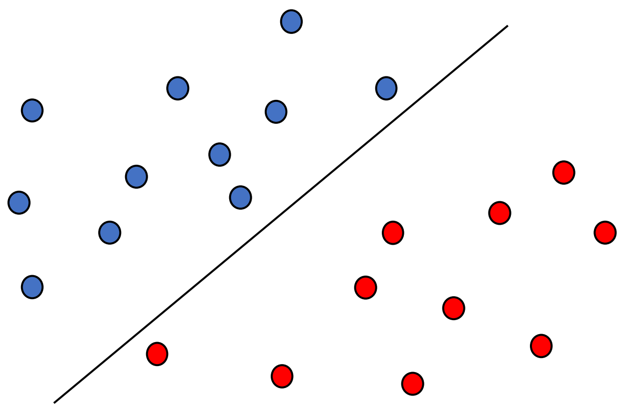
i.e. $y_i f(\mathbf{x}_i) > 0$ for a correct classification.

Linear SVM

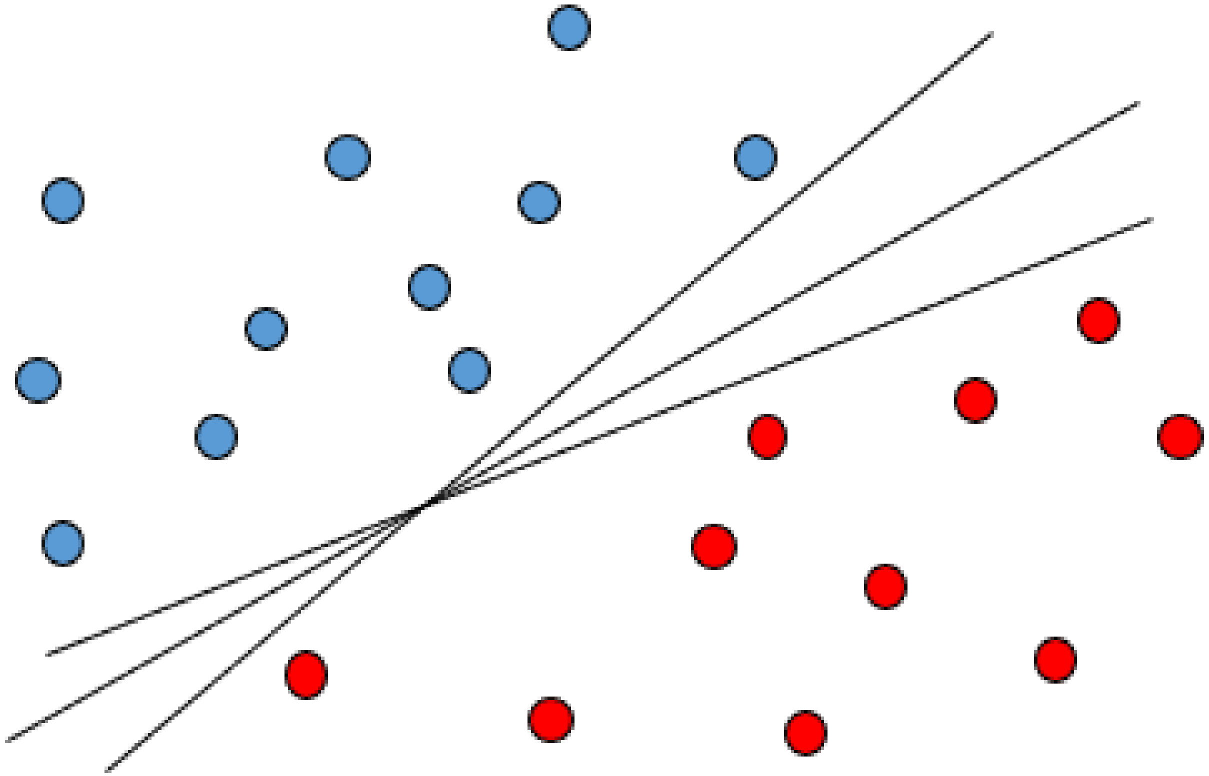
- Binary classification
- Assumes that data is **linear**
  - Exists a hyperplane that divides the classes as positives and negatives.

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i \in \{1,\ldots,m\}$$

$$\mathbf{w} \in R^n, \quad \mathbf{x}_i \in R^n, \quad b \in R, \quad y_i \in \{+1,-1\}$$

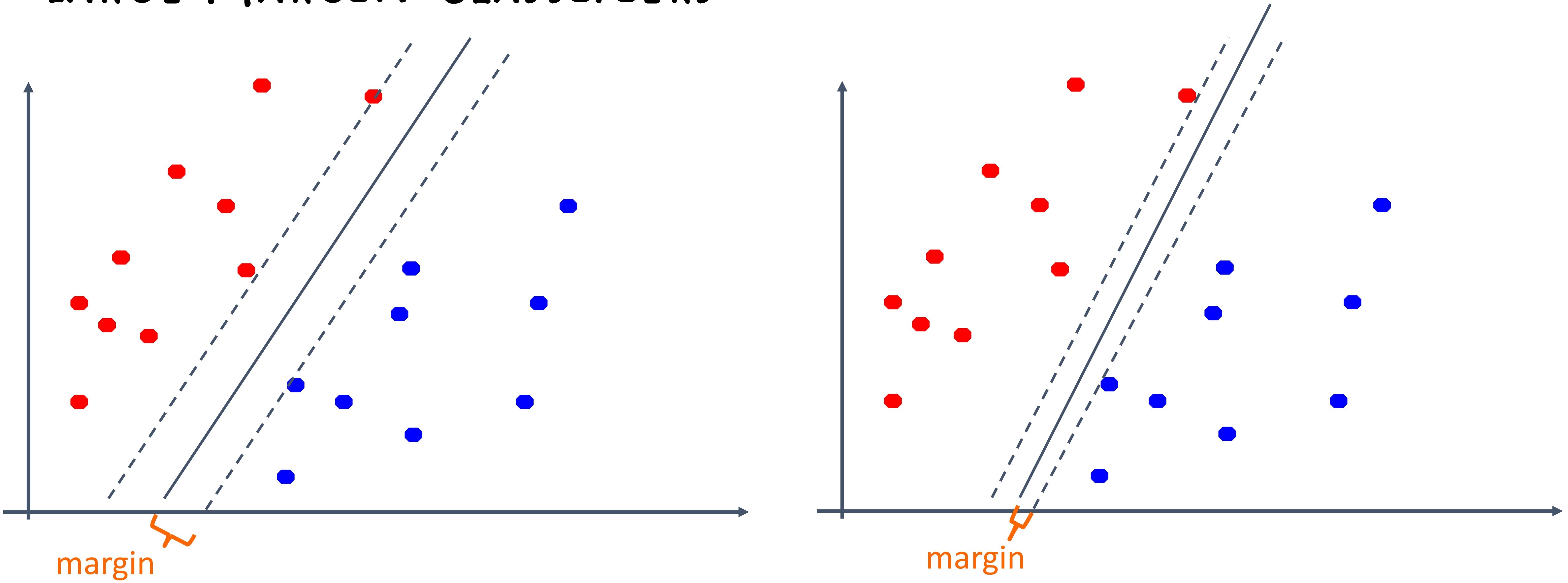


Which hyperplane?



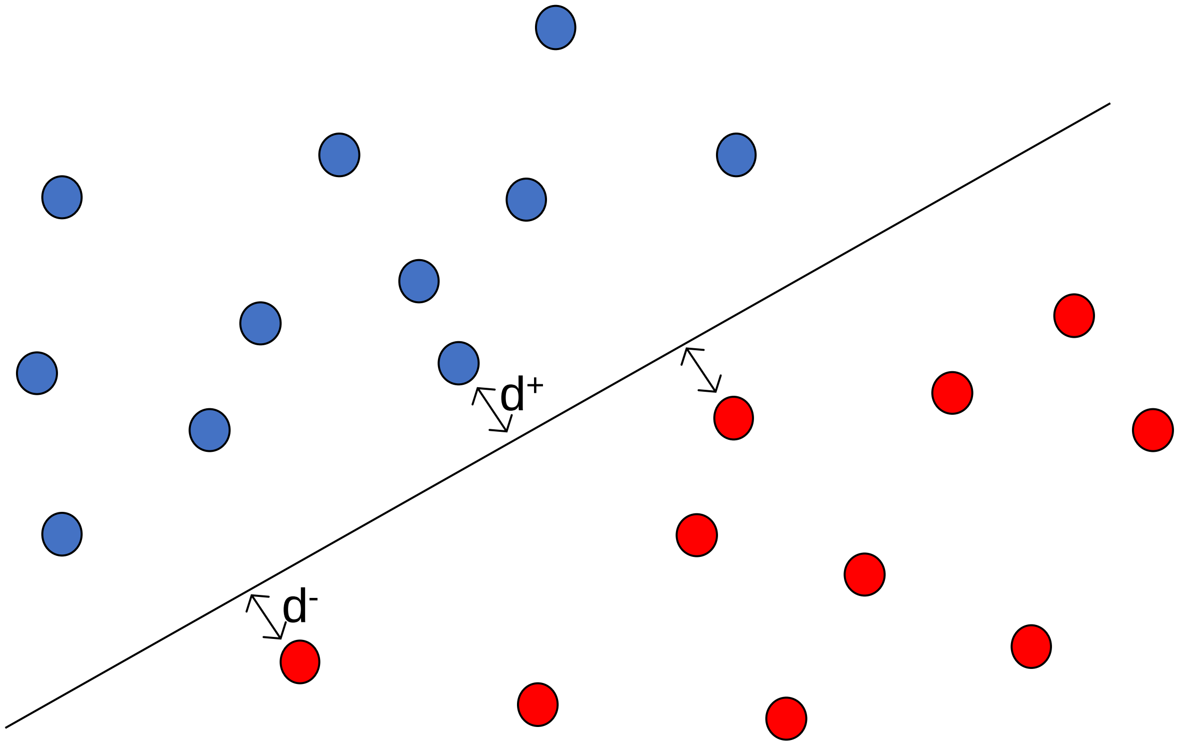
What is the best-separating hyperplane?

Large margin Classifiers



The **margin** of a classifier is the distance to the closest points of either class. **Large margin** classifiers attempt to maximize this.

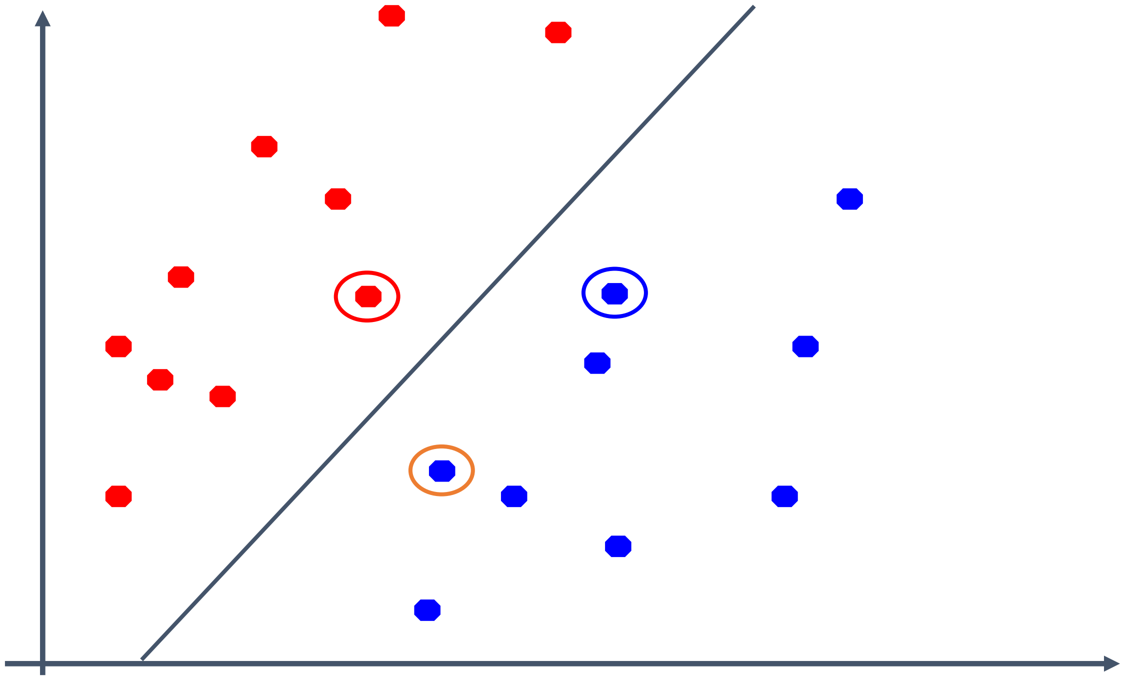
Maximizing the margin



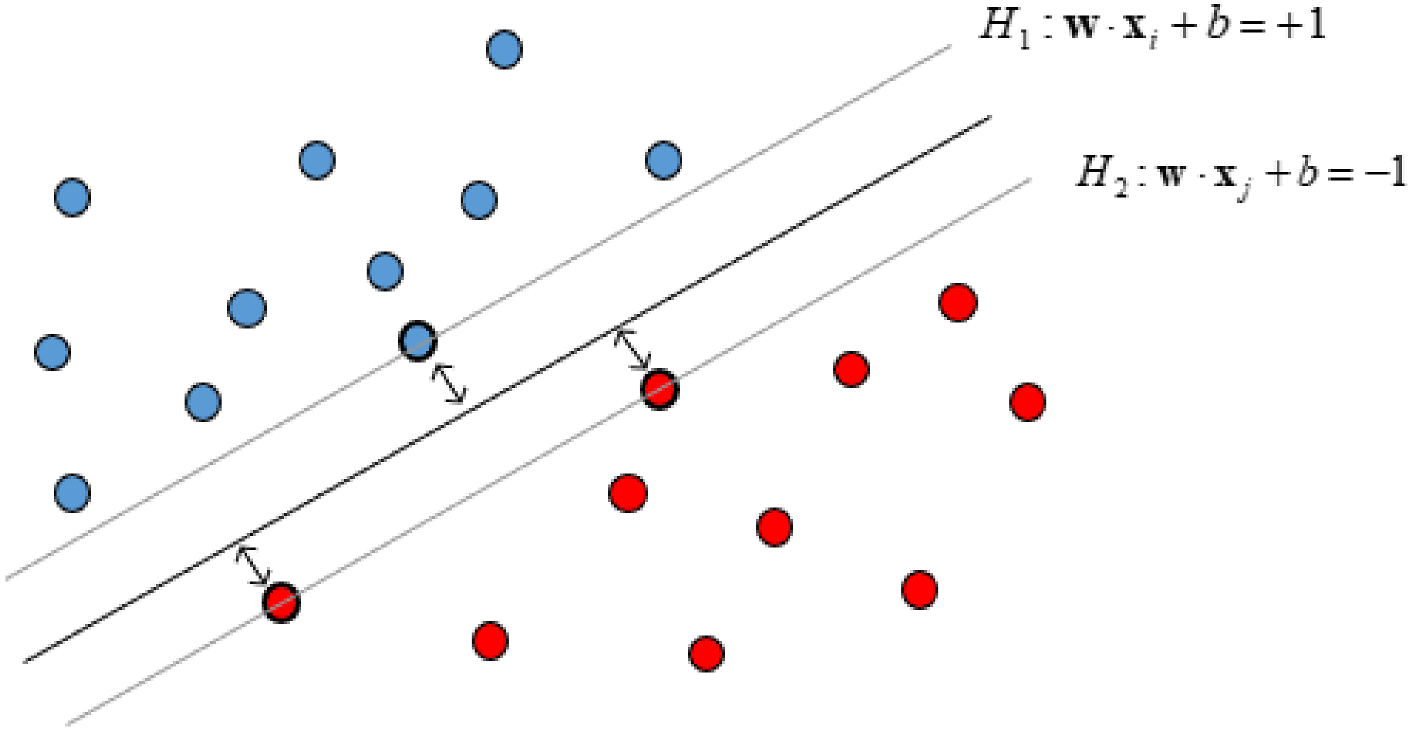
SVMs **maximize** the margin $d=d^{+}+d^{-}$
d^{+} (d^{-}) is the minimum distance to the nearest positive (negative) sample

Support Vectors

- For any separating hyperplane, there exist some set of “closest points”
  - These are called the support vectors.
- For n dimensions, there will be at least $n+1$ support vectors.



Support Vectors

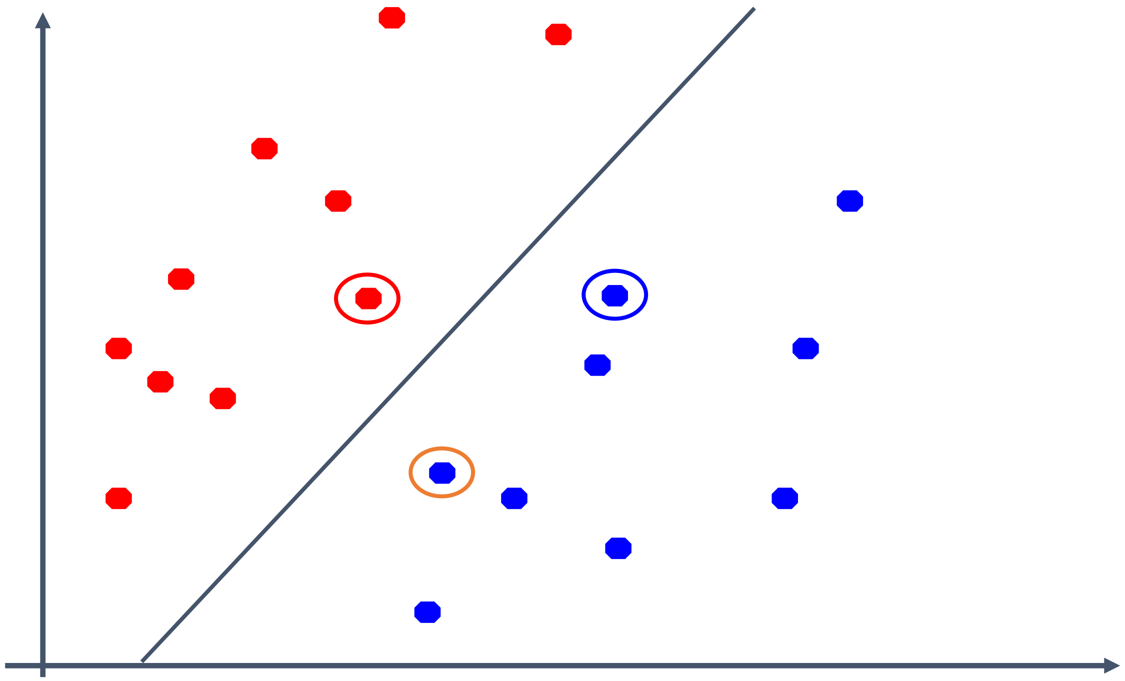


The samples that lie on the lines

H_1 and H_2 are called *Support Vectors.*

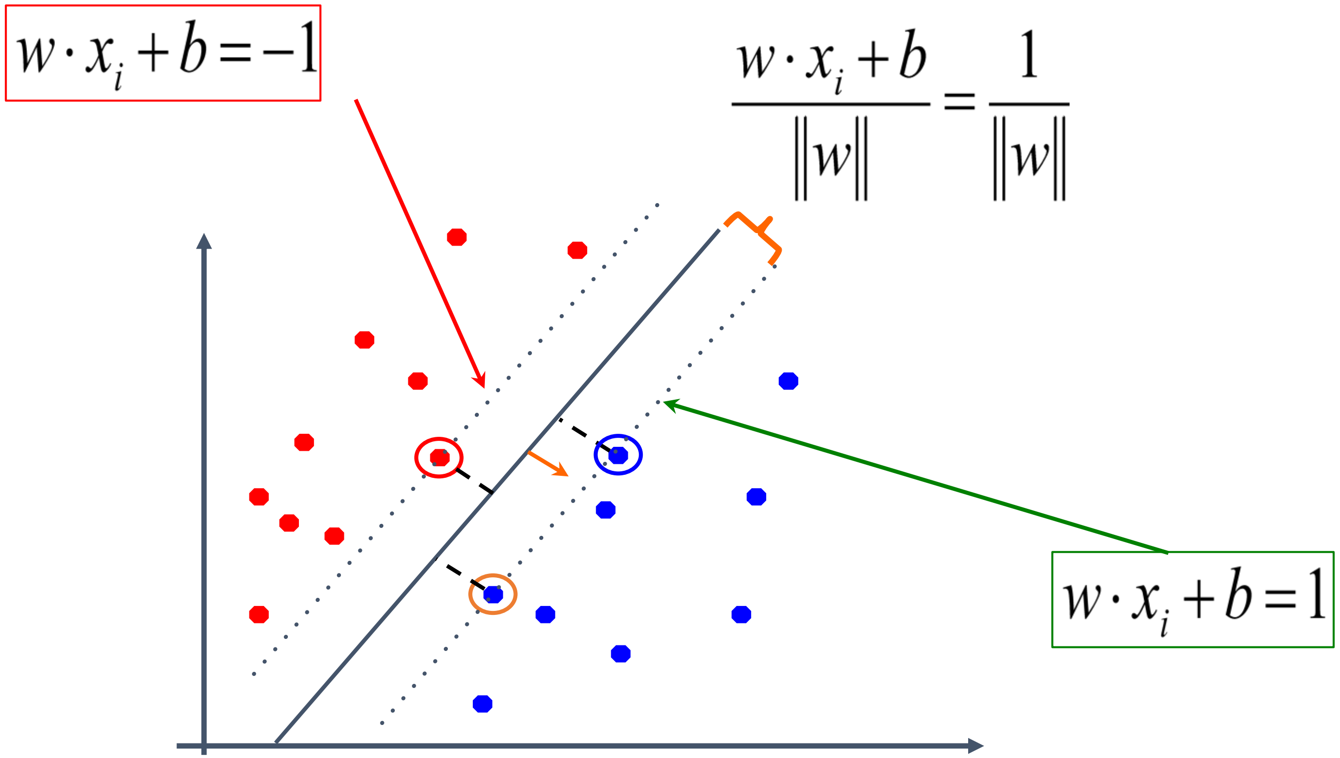
Large Margin Classifiers & Support Vectors

- Maximizing the margin is **good** since it implies that **only support vectors matter**, other training examples are ignorable.



Finding the Hyperplane

- How to measure the margin?
  - Notice that the **margin** is the distance to the support vectors, i.e., the "closest points", on either side of the hyperplane



Herein, we formulate only for one side of the margin but when we consider both sides then the formula becomes 2/ ǁwǁ, but constants do not affect the optimization.

Maximize the margin

- Select the hyperplane with the largest margin where the points are classified correctly and outside the margin!

- This is setting up as a **constrained optimization problem**:

$$\max_{w,b} \quad \text{margin}(w,b)$$
subject to: $y_i(w \cdot x_i + b) \geq 1 \quad \forall i$

$$\Rightarrow$$

$$\max_{w,b} \quad \frac{1}{\|w\|}$$
subject to: $y_i(w \cdot x_i + b) \geq 1 \quad \forall i$

Maximize the margin

- Maximizing the margin is equivalent to minimizing the norm of the weights (subject to separating constraints).

$$\max_{w,b} \quad \mathrm{margin}(w,b)$$
subject to: $y_i(w \cdot x_i + b) \geq 1 \quad \forall i$

$\Rightarrow$

$$\max_{w,b} \quad \frac{1}{\|w\|}$$
subject to: $y_i(w \cdot x_i + b) \geq 1 \quad \forall i$

$\Rightarrow$

$$\min_{w,b} \quad \|w\|$$
subject to: $y_i(w \cdot x_i + b) \geq 1 \quad \forall i$

Maximize the margin

$$\min_{w,b} \quad \|w\|$$

subject to: $y_i(w \cdot x_i + b) \geq 1 \quad \forall i$

- The minimization criterion wants ***w*** to be as small as possible!
- The constraint makes sure that the data is separable!

Support Vector Machine Formula

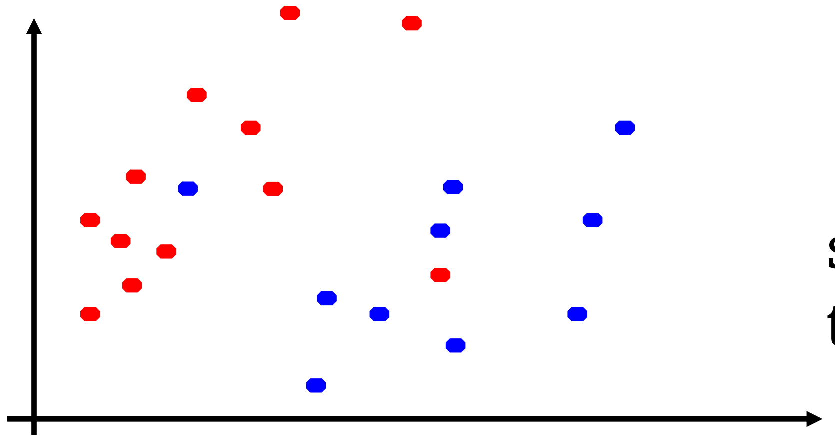
$$\min_{w,b} \quad \|w\|^2$$

subject to: $y_i(w \cdot x_i + b) \geq 1 \quad \forall i$

- Rather than directly minimizing the norm of the ***w***, the formulation of the SVM optimization minimizes the **square** of it.
  - Square is differentiable, thus easy to handle mathematically.
  - Square of the normal of the ***w*** still achieves the same results.

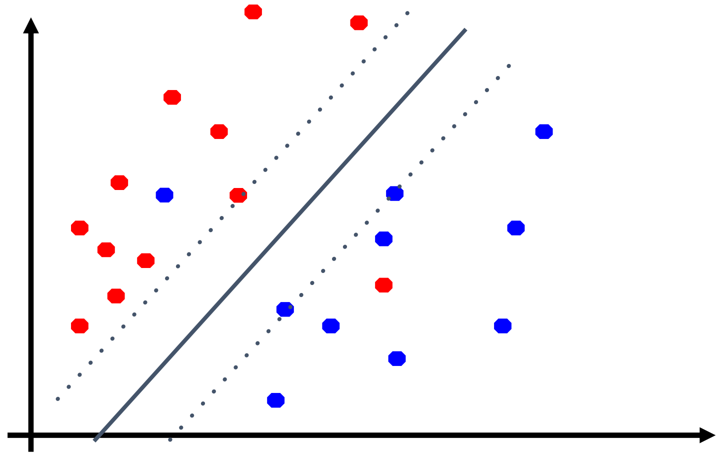
Soft margin classification

- What about this problem?
- What do we do if the dataset is not linearly separable?



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

subject to: $y_i(w \cdot x_i + b) \geq 1 \quad \forall i$



- This quadratic optimization does not converge for non-linearly separable datasets.
- Therefore, we need to do some modifications.

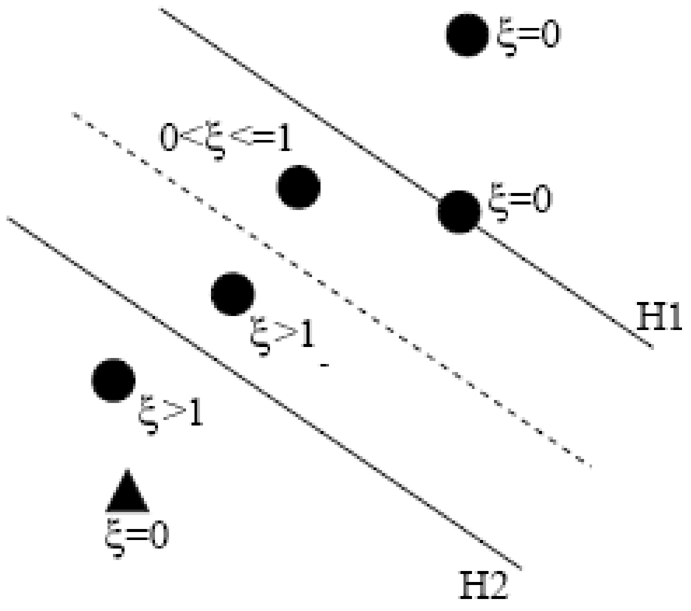
Slack variables

UNIVERSITÀ di VERONA

- We modify the constraints by adding "slack" variables, which enable vectors to cross the margin.

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i$$

- The value of the ξ_i indicates the position of the vector with respect to the hyperplane



Slack variables

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

subject to:

$$y_i(w \cdot x_i + b) \geq 1 \quad \forall i$$

$$\min_{w,b} \quad \|w\|^2 + C\sum_i \varsigma_i$$

subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

$$\varsigma_i \geq 0$$

slack variables

(one for each example)

What effect do they have?

Soft margin SVM

$$\min_{w,b} \quad \|w\|^2$$
subject to:
$$y_i(w \cdot x_i + b) \geq 1 \quad \forall i$$

$$\min_{w,b} \quad \|w\|^2 + C\sum_i \varsigma_i$$
subject to:
$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$
$$\varsigma_i \geq 0$$

penalized by how far from "correct"

allowed to make a mistake

- Every constraint can be satisfied if **slack variable** is sufficiently large.
- C is a regularization parameter.
- Small C (large margin)
  - Constraints to be easily ignored
- Large C (narrow margin)
  - Constraints hard to ignore

- $C = \infty$ enforces all constraints: hard margin

Soft margin SVM

$$\min_{w,b} \quad \|w\|^2 + C\sum_i \varsigma_i$$

subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

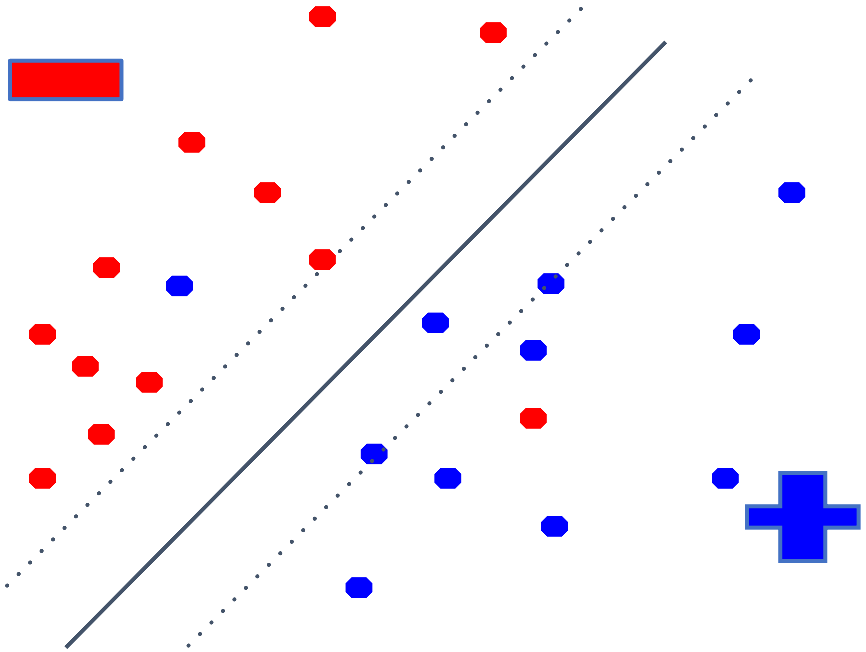
$$\varsigma_i \geq 0$$

In order words,

- C is a user-defined parameter that represents the cost for misclassified data.

- C determines the sensitivity of the classifier to the errors and its generalization performance.

Understanding the Soft Margin SVM



$$\min_{w,b} \; \|w\|^2 + C\sum_i \varsigma_i$$

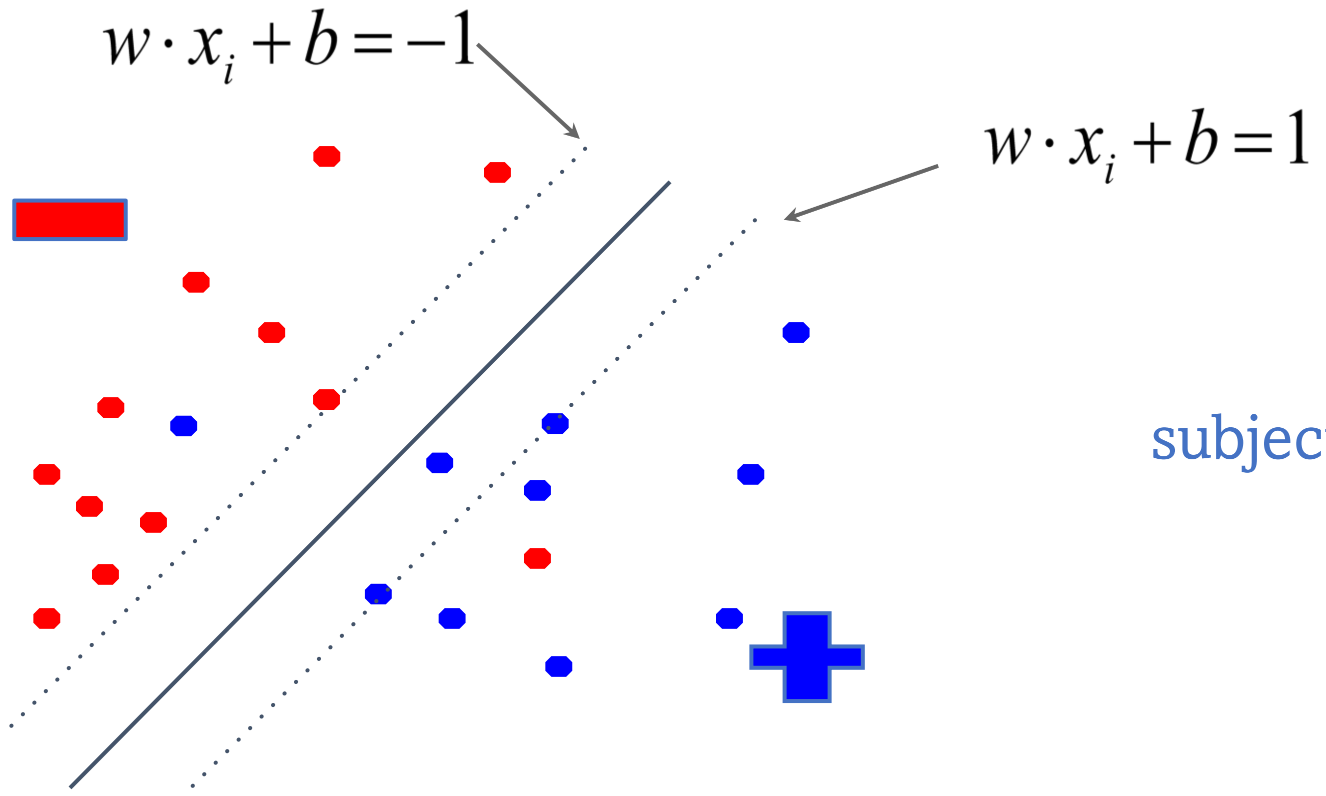
subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

$$\varsigma_i \geq 0$$

Given the optimal solution (w, b), can we figure out what the slack penalties are for each point?

Understanding the Soft Margin SVM



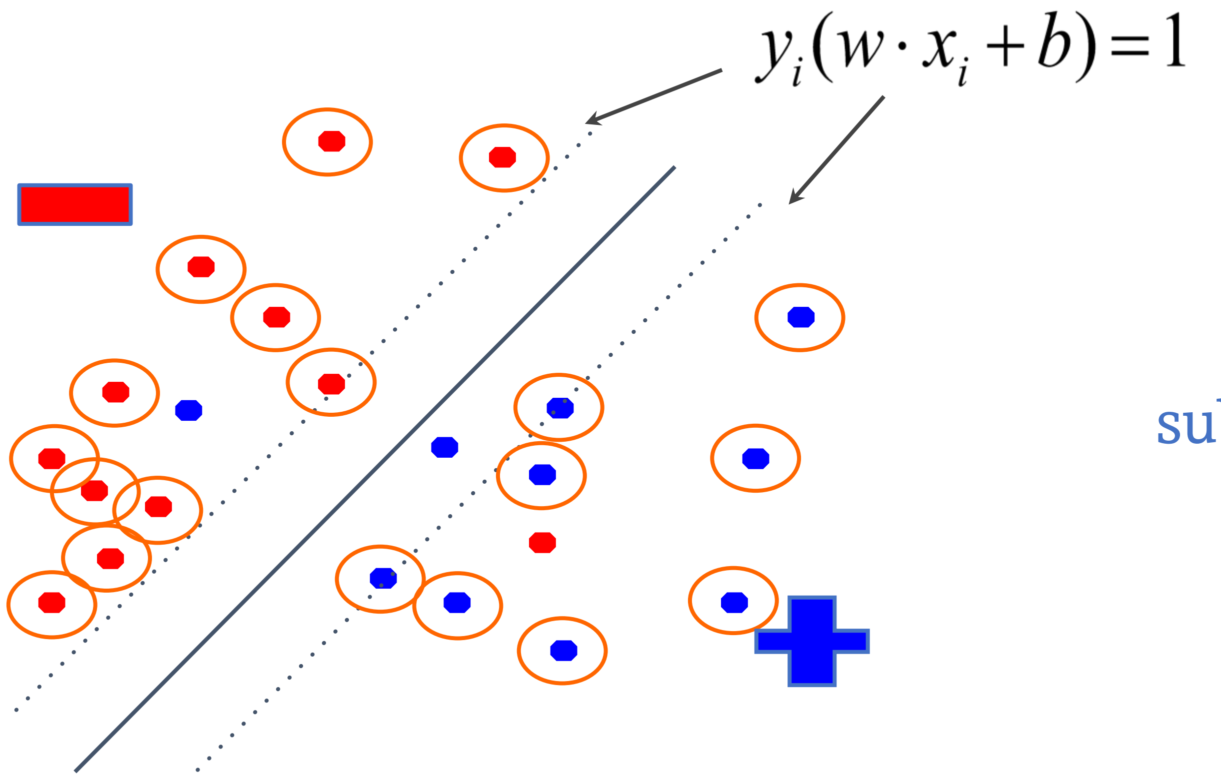
$$\min_{w,b} \quad \|w\|^2 + C \sum_i \varsigma_i$$

subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$
$$\varsigma_i \geq 0$$

or: $\boxed{y_i(w \cdot x_i + b) = 1}$

Understanding the Soft Margin SVM



$$\min_{w,b} \quad \|w\|^2 + C\sum_i \varsigma_i$$

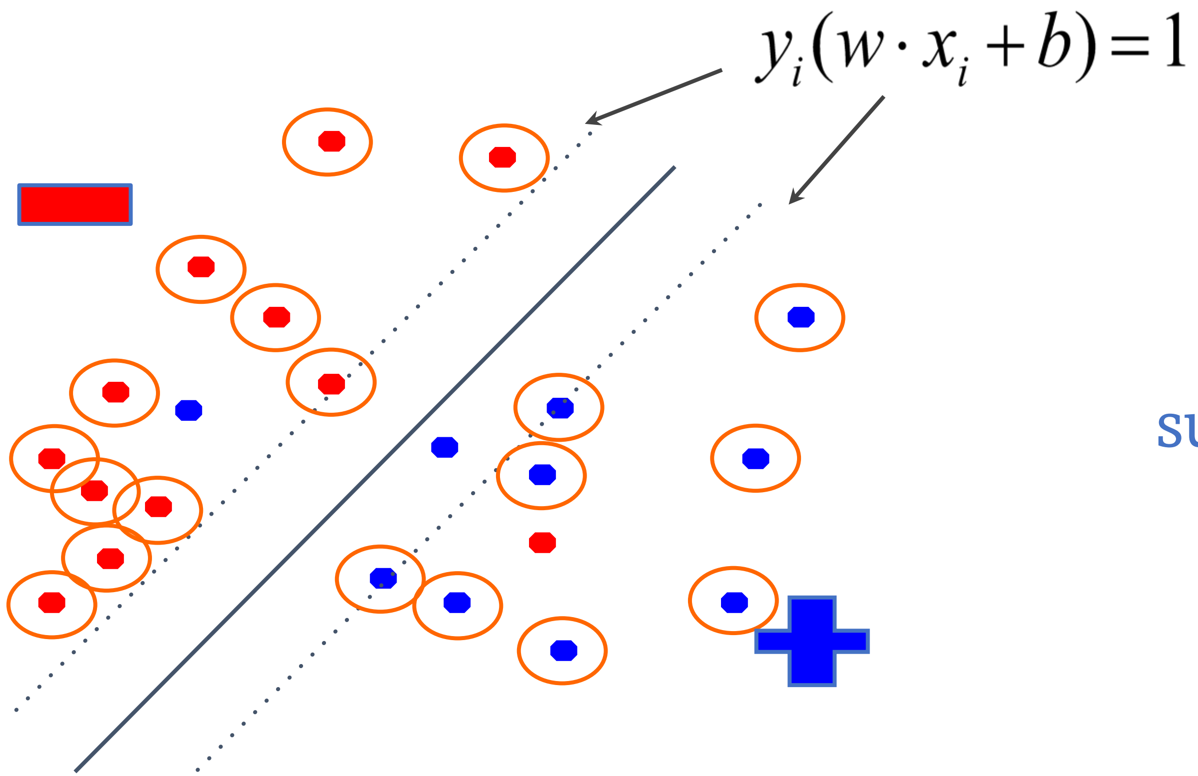
subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

$$\varsigma_i \geq 0$$

What are the slack values for points outside (or on) the margin AND correctly classified?

Understanding the Soft Margin SVM



$$\min_{w,b} \ \|w\|^2 + C\sum_i \varsigma_i$$

subject to:

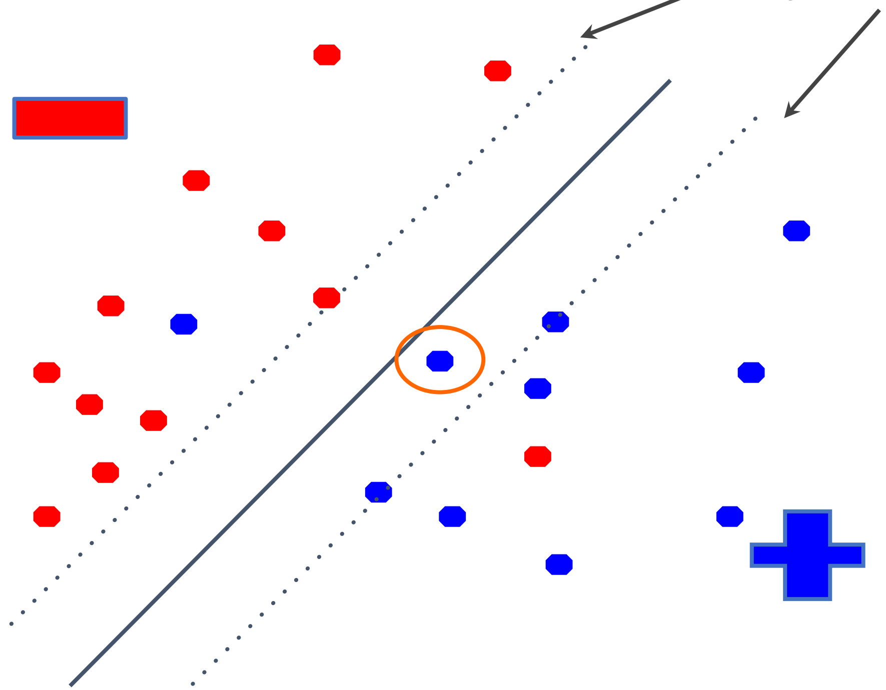
$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

$$\varsigma_i \geq 0$$

0! The slack variables have to be greater than or equal to zero and if they are on or beyond the margin then $y_i(wx_i+b) \geq 1$ already satisfied.

Understanding the Soft Margin SVM

$$y_i(w \cdot x_i + b) = 1$$



$$\min_{w,b} \quad \|w\|^2 + C\sum_i \varsigma_i$$

subject to:

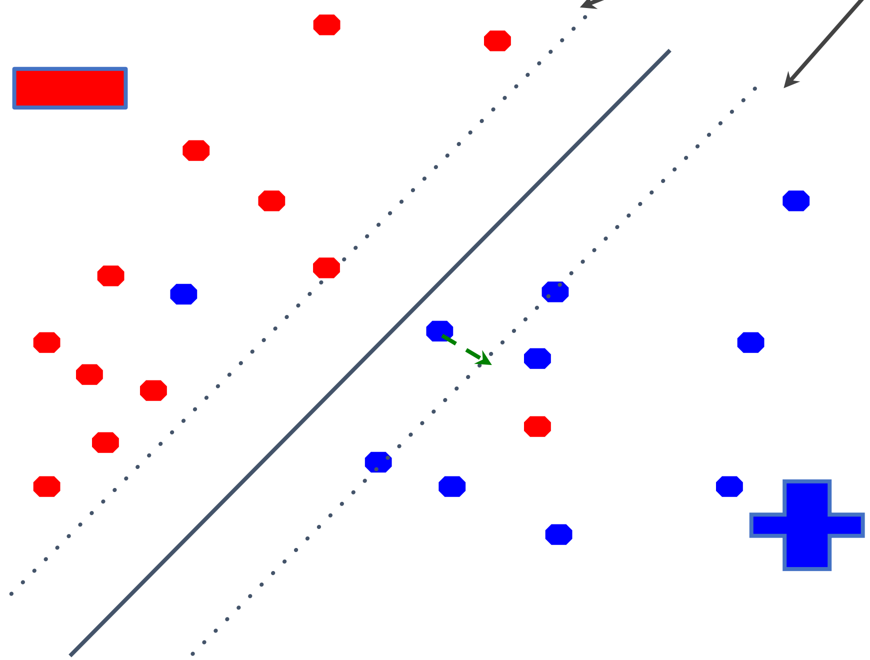
$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

$$\varsigma_i \geq 0$$

What are the slack values for points inside the margin AND classified correctly?

Understanding the Soft Margin SVM

$$y_i(w \cdot x_i + b) = 1$$



$$\min_{w,b} \quad \|w\|^2 + C\sum_i \varsigma_i$$

subject to:

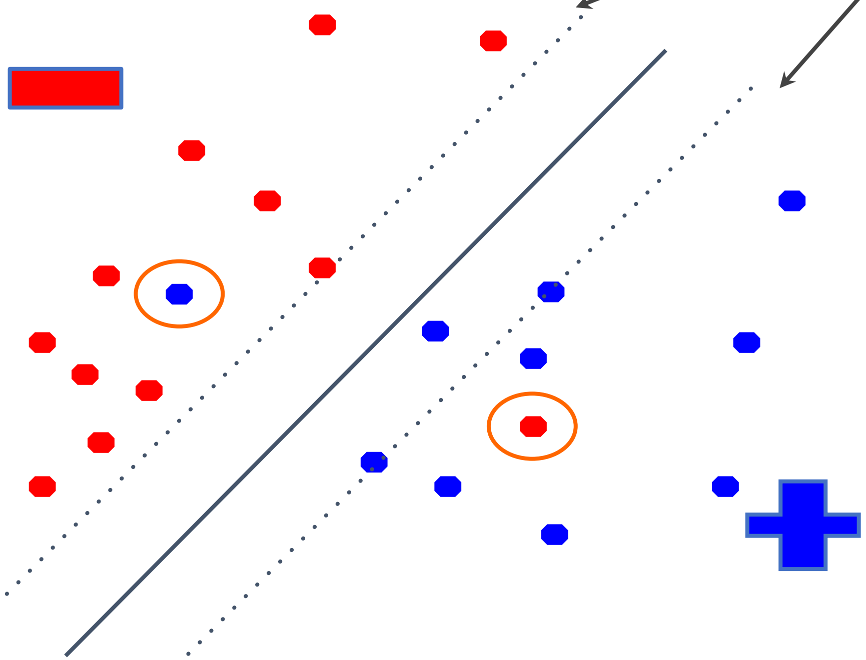
$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

$$\varsigma_i \geq 0$$

Difference from the point to the margin, i.e. $\varsigma_i = 1 - y_i(w \cdot x_i + b)$

Understanding the Soft Margin SVM

$$y_i(w \cdot x_i + b) = 1$$



$$\min_{w,b} \quad \|w\|^2 + C\sum_i \varsigma_i$$

subject to:

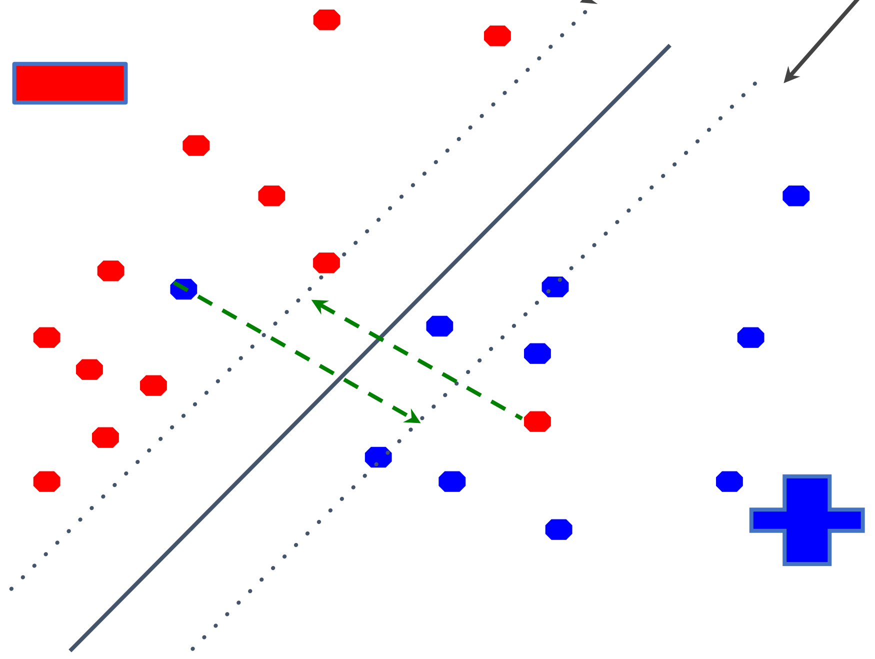
$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

$$\varsigma_i \geq 0$$

What are the slack values for points that are **incorrectly classified?**

Understanding the Soft Margin SVM

$$y_i(w \cdot x_i + b) = 1$$



$$\min_{w,b} \quad \|w\|^2 + C\sum_i \varsigma_i$$

subject to:

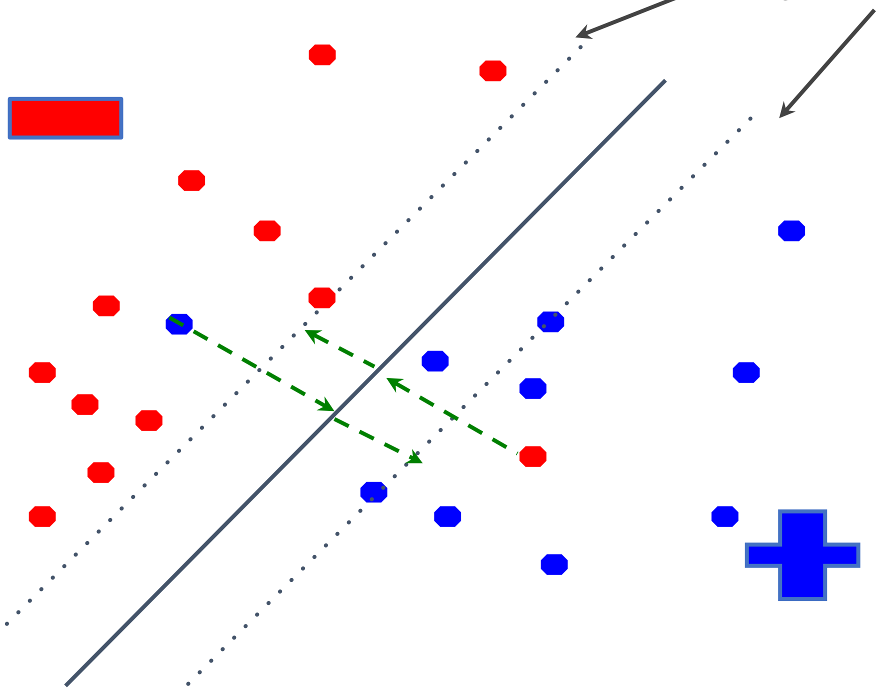
$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

$$\varsigma_i \geq 0$$

What are the slack values for points that are **incorrectly classified?**

Understanding the Soft Margin SVM

$$y_i(w \cdot x_i + b) = 1$$



$$\min_{w,b} \quad \|w\|^2 + C\sum_i \varsigma_i$$

subject to:

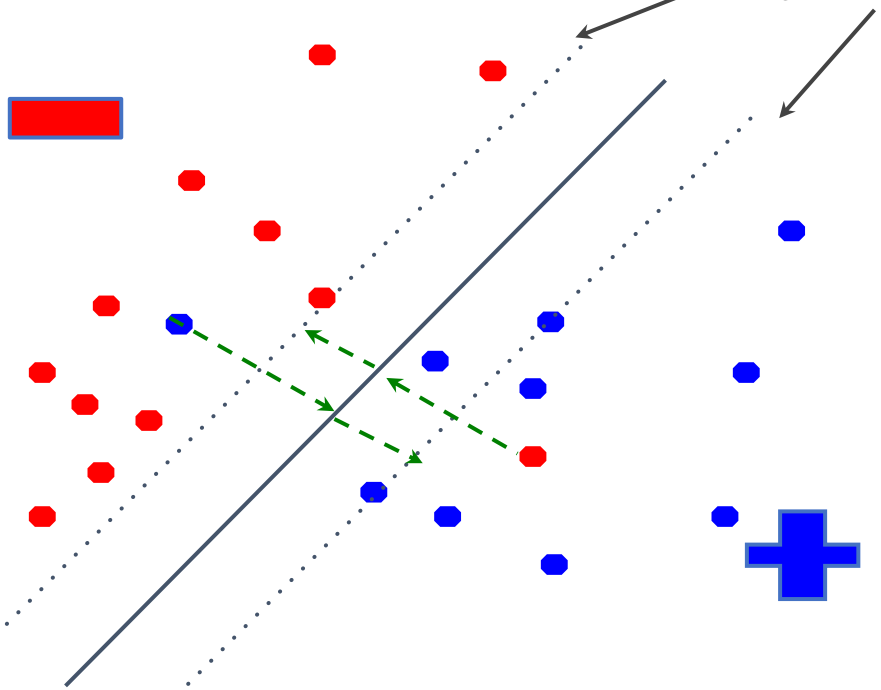
$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

$$\varsigma_i \geq 0$$

"distance" to the hyperplane *plus* the "distance" to the margin.
"distance" is the **unnormalized projection**, not to be confused with the true distance which would be with respect to w/||w||.

Understanding the Soft Margin SVM

$$y_i(w \cdot x_i + b) = 1$$



$$\min_{w,b} \quad \|w\|^2 + C\sum_i \varsigma_i$$

subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

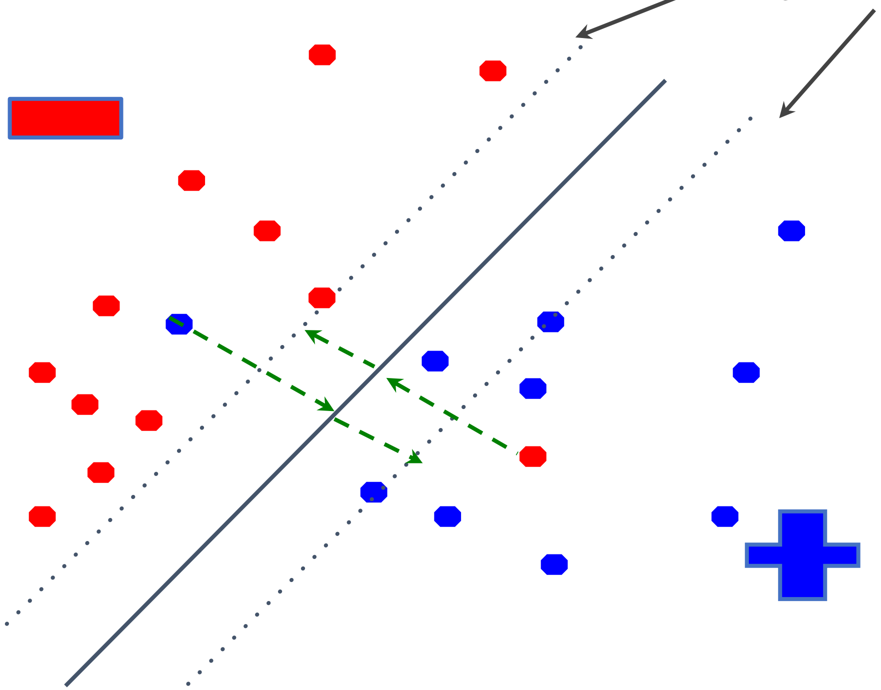
$$\varsigma_i \geq 0$$

"distance" to the hyperplane *plus* the "distance" to the margin

$$-y_i(w \cdot x_i + b) \qquad\qquad\qquad\qquad 1$$

Understanding the Soft Margin SVM

$$y_i(w \cdot x_i + b) = 1$$



$$\min_{w,b} \; \|w\|^2 + C\sum_i \varsigma_i$$

subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

$$\varsigma_i \geq 0$$

"distance" to the hyperplane *plus* the "distance" to the margin

$$\varsigma_i = 1 - y_i(w \cdot x_i + b)$$

Understanding the Soft Margin SVM

$$\min_{w,b} \; \|w\|^2 + C\sum_i \varsigma_i$$

subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$
$$\varsigma_i \geq 0$$

$$\varsigma_i = \begin{cases} 0 & if\ y_i(w \cdot x_i + b) \geq 1 \\ 1 - y_i(w \cdot x_i + b) & otherwise \end{cases}$$

Understanding the Soft Margin SVM

$$\varsigma_i = \begin{cases} 0 & if\ y_i(w \cdot x_i + b) \geq 1 \\ 1 - y_i(w \cdot x_i + b) & otherwise \end{cases}$$

$$\varsigma_i = \max(0, 1 - y_i(w \cdot x_i + b))$$

$$= \max(0, 1 - yy')$$

Hinge Loss

Understanding the Soft Margin SVM

$$\min_{w,b} \quad \|w\|^2 + C\sum_i \varsigma_i$$

subject to:

$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

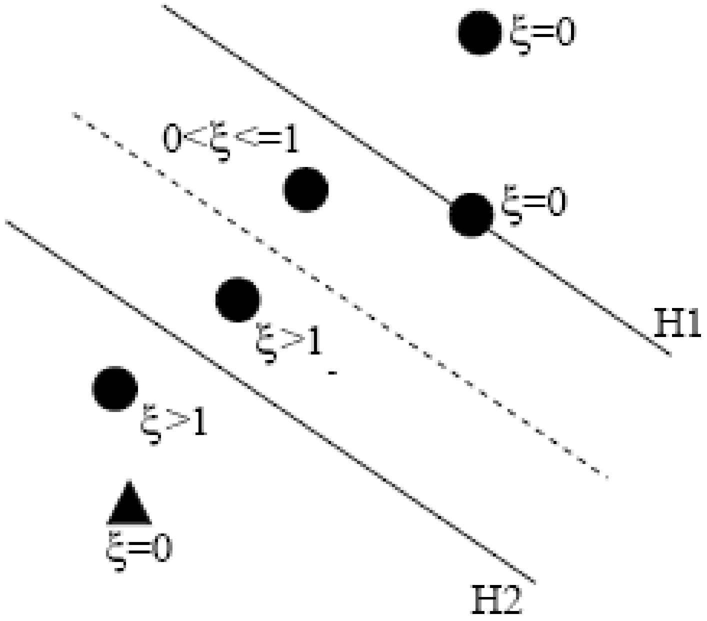
$$\varsigma_i \geq 0$$

$$\varsigma_i = \max(0, 1 - y_i(w \cdot x_i + b))$$

$$\min_{w,b} \quad \|w\|^2 + C\sum_i \max(0, 1 - y_i(w \cdot x_i + b))$$

Unconstrained problem!

Summary: Soft Margin SVM



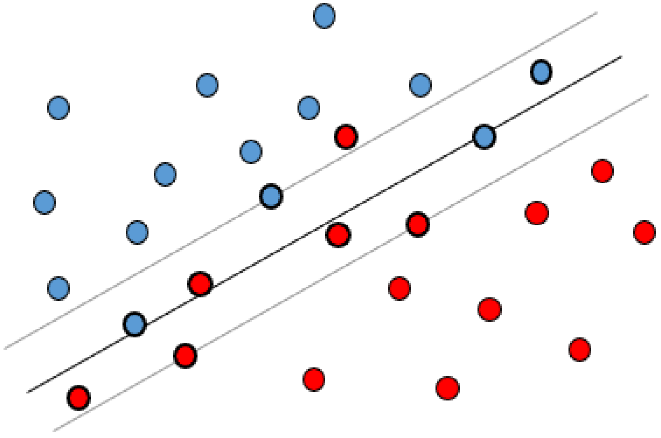
Values of ξ mean:

$\xi_i = 0$ the data point is correctly classified

$0 < \xi_i < 1$ correctly classified but beyond H_i

$\xi_i \geq 1$ incorrectly classified, error

Support vectors in Non-linear case



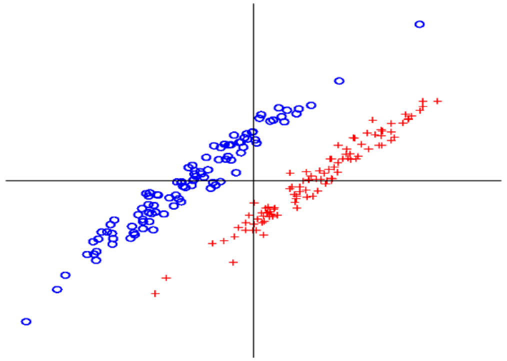
- Support vectors are all points for which a slack variable exists.

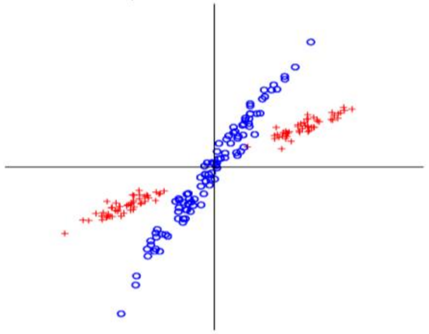
- The points where the slack variable is zero (i.e., the point is correctly classified and outside the margin) cannot be a support vector.
    - Thus, support vectors **lie on the margin** for **correctly classified points**.
    - And lie inside the margin for points that may be misclassified or violate the margin in soft-margin SVM.

- This means that the **misclassified data** or the data **within the margin** are the support vectors.

Complex cases

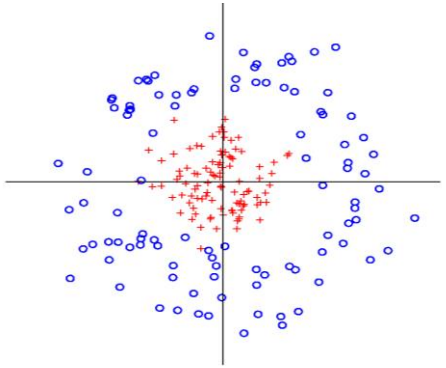
Parallel clouds

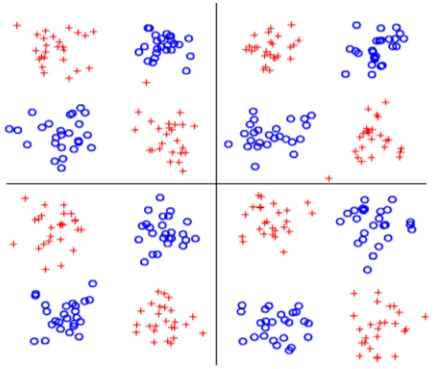




Split

Donut



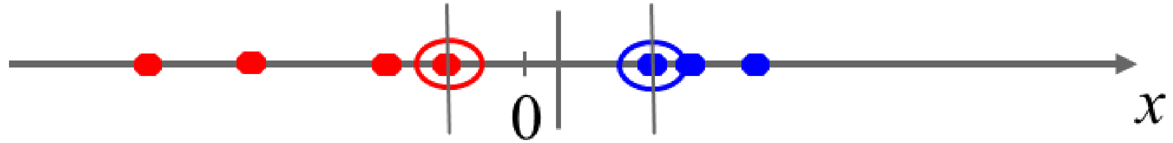


Checkerboard

What do we do when linear hyperplanes are not sufficient?

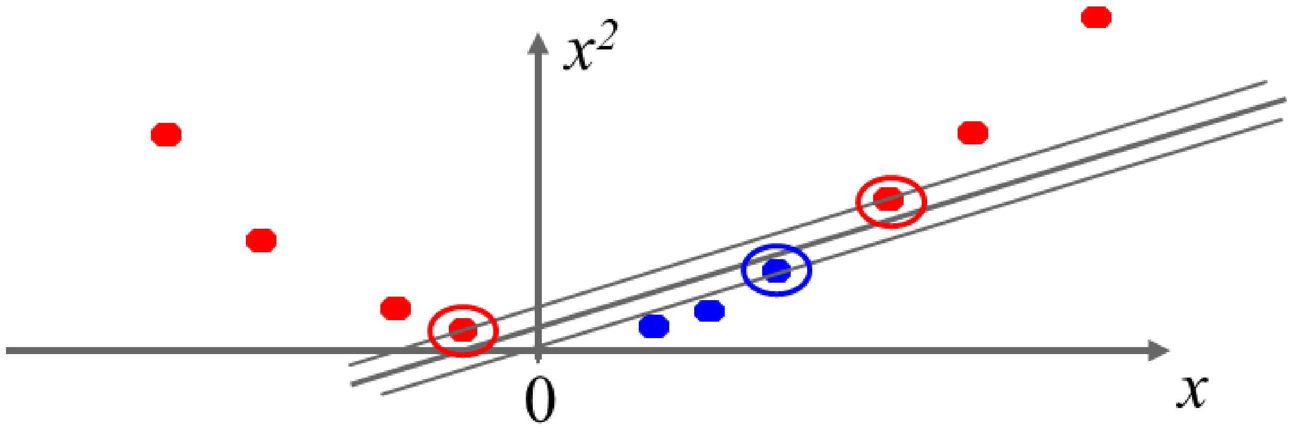
NON-LINEAR SVM

- Datasets that are linearly separable with some noise work out great:



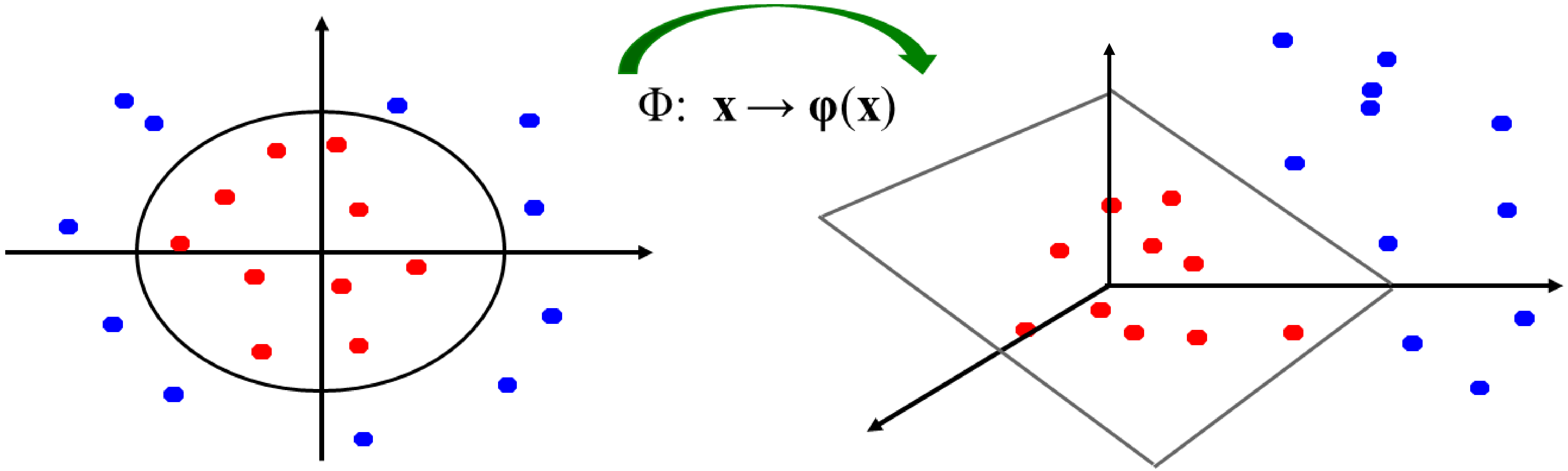
- However, what can we do if the dataset is just too hard?

- What about **mapping data to a higher-dimensional space**:

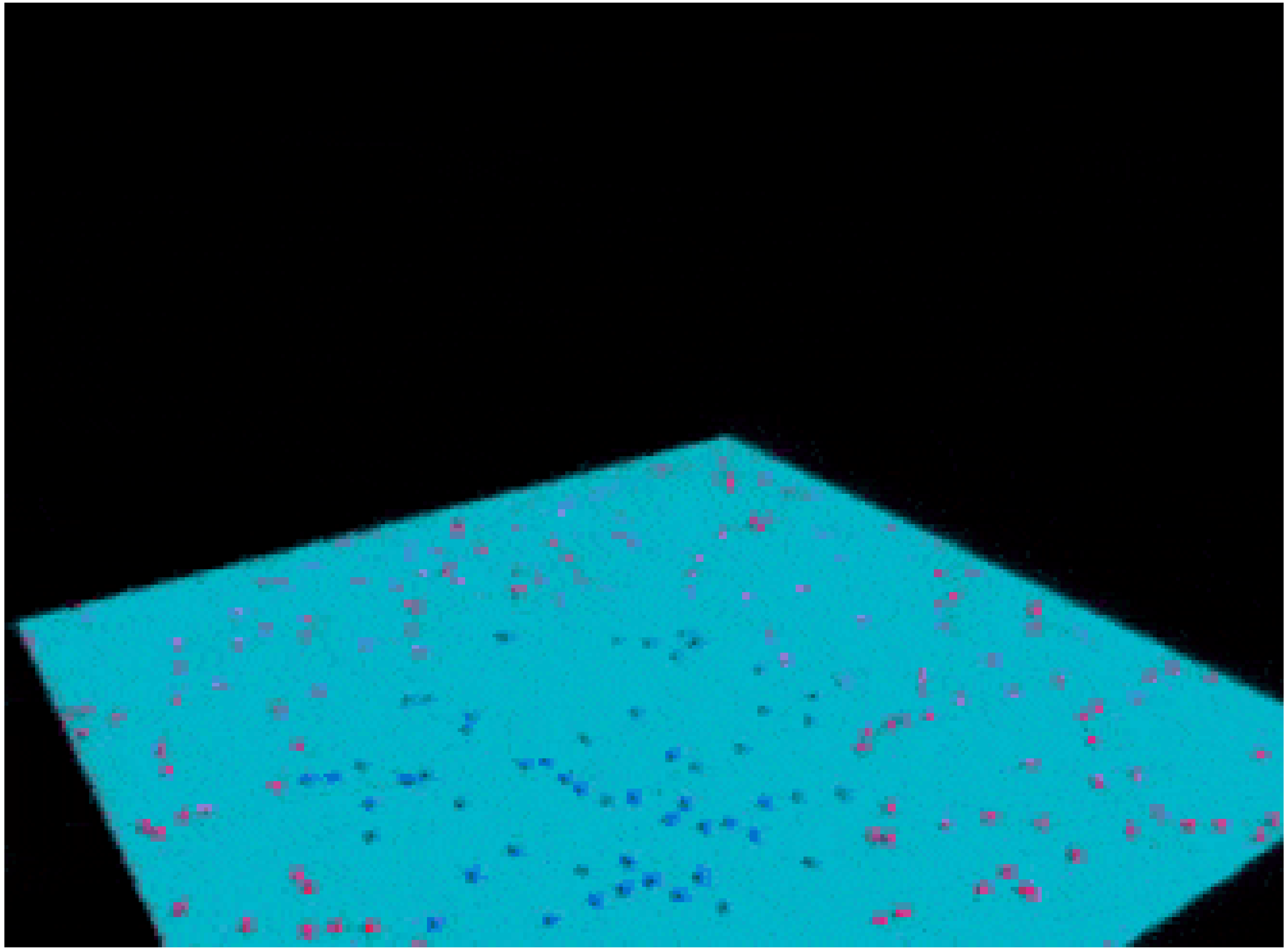


Non-Linear SVM

- General idea: The original feature space can always be mapped to some higher-dimensional feature space where the training set is linearly separable.



Non-Linear SVM



Example: Mapping

Let's map the space $X = \{x, y, z\}$ into the higher dimensional space Z:

$$\varphi_1(X) = x \quad \varphi_2(X) = y \quad \varphi_3(X) = z$$
$$\varphi_4(X) = x^2 \quad \varphi_5(X) = y^2 \quad \varphi_6(X) = z^2$$
$$\varphi_7(X) = xy \quad \varphi_8(X) = xz \quad \varphi_9(X) = yz$$

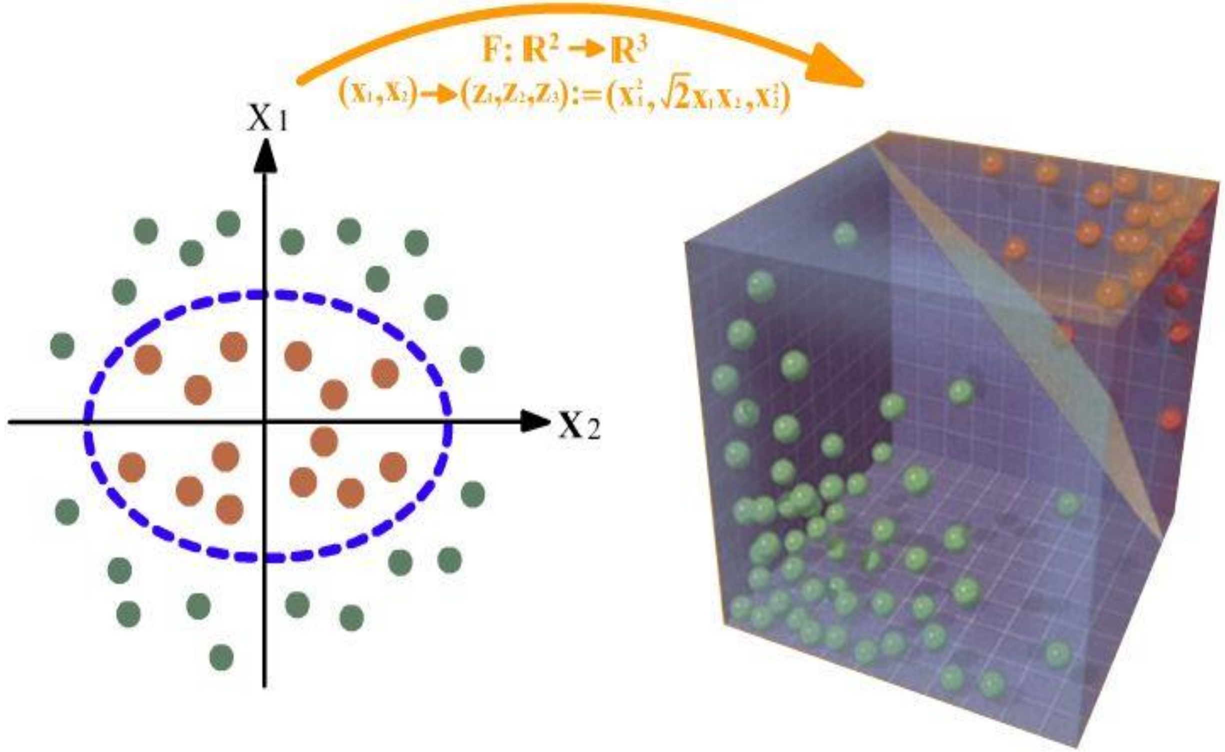
$$Z = \left(\varphi_1(X), \varphi_2(X), \ldots, \varphi_9(X)\right)$$

Notice that a linear classifier in the space Z corresponds to a polynomial one in the original space.

Example 2: Mapping

The below function $R^2 \rightarrow R^3$ maps the cartesian plane onto a 3D cone whose intersection with the x_1x_2 plane gives the elliptical boundary.

$$\varphi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$



Problems with Mapping

- Can have huge dimensionality (even infinite).
- Mappings are in general difficult to compute.
- It is not clear how to find the proper mapping that will separate the data.

Applying Mapping to SVMs

- Instead of computing the transformation Φ explicitly, we use a kernel trick.
- The **kernel trick** allows us to compute **dot product of two points** in the higher dimensional space.
  - dot products $(\mathbf{x}_i \cdot \mathbf{x}_j)$
- In other words, we define a kernel function ***K:*** $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$

Allowing us to train the classifier without even ever bothering to explicityly compute the mapping Φ.

- **IMPORTANT!** Since the resulting classification algorithm is independent from the size of the target space, we avoid curse of dimensionality.

Kernel Trick

- The linear classifier relies on inner product between vectors:

$$K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$$

- If every datapoint is mapped into high dimensional space via some transformation Φ: $\mathbf{x} \rightarrow \varphi(\mathbf{x})$, the inner product becomes:

$$K(\mathbf{x}_i,\mathbf{x}_j)= \boldsymbol{\varphi}(\mathbf{x}_i)^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x}_j)$$

- A kernel function is a function that is equivalent to an inner product in some feature space.

Kernel Trick

Example:

2-dimensional vectors $\mathbf{x}=[x_1 \ \ x_2]$; let $K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j)^2$,

Need to show that $K(\mathbf{x}_i,\mathbf{x}_\mathbf{j})= \boldsymbol{\varphi}(\mathbf{x}_i)^{\mathbf{T}}\boldsymbol{\varphi}(\mathbf{x}_j)$:

$$K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j)^2 = 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}=$$

$$= [1 \ \ x_{i1}^2 \ \ \sqrt{2}\, x_{i1}x_{i2} \ \ x_{i2}^2 \ \ \sqrt{2}x_{i1} \ \ \sqrt{2}x_{i2}]^{\mathbf{T}} [1 \ \ x_{j1}^2 \ \ \sqrt{2}\, x_{j1}x_{j2} \ \ x_{j2}^2 \ \ \sqrt{2}x_{j1} \ \ \sqrt{2}x_{j2}] =$$

$$= \boldsymbol{\varphi}(\mathbf{x}_i)^{\mathbf{T}}\boldsymbol{\varphi}(\mathbf{x}_j), \quad \text{where } \boldsymbol{\varphi}(\mathbf{x}) = [1 \ \ x_1^2 \ \ \sqrt{2}\, x_1x_2 \ \ x_2^2 \ \ \sqrt{2}x_1 \ \ \sqrt{2}x_2]$$

- A kernel function implicitly maps data to a high-dimensional space (without the need to compute each $\boldsymbol{\varphi}(\mathbf{x})$ explicitly).

Standard Kernels

UNIVERSITÀ di VERONA

Linear	$K(x,z)=\langle x,z\rangle$
Polynomial	$K(x,z)=(\langle x,z\rangle+1)^p$
Radial basis functions	$K(x,z)=e^{-\frac{\lVert x-z\rVert^2}{2\sigma^2}}$
Sigmoid	$K(x,z)=\tanh(a\langle x,z\rangle+b)$

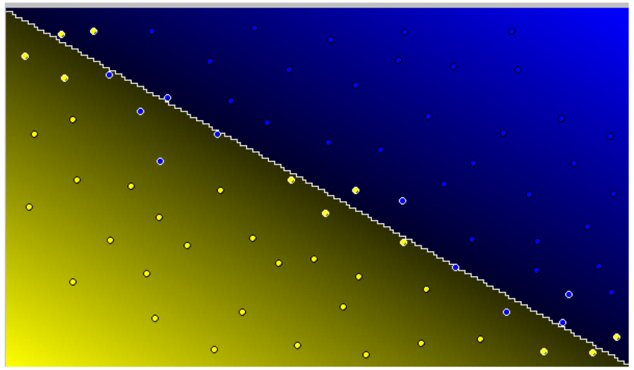
Hints about kernel

- Kernel selection and parameter tuning are critical.
- C has a huge impact on the generalization ability.
- Lowering the degree for polynomial kernels or larger signals for RBF kernels can avoid overfitting.
- The number of support vectors is a measure of generalization performance.

Kernel Selection

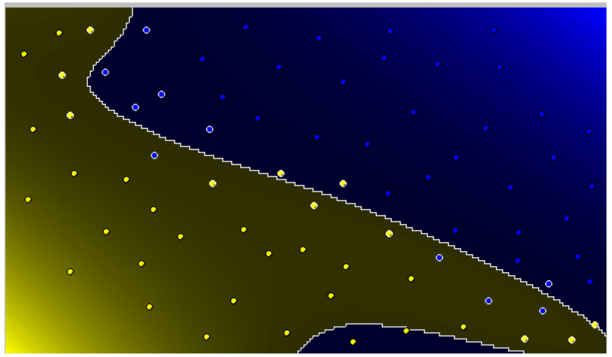
Linear kernel

- Used when the feature space is **huge** (for example in text classification, which uses individual word counts as features)
- Shown to be a special case of the RBF kernel
- No additional parameters



Polynomial

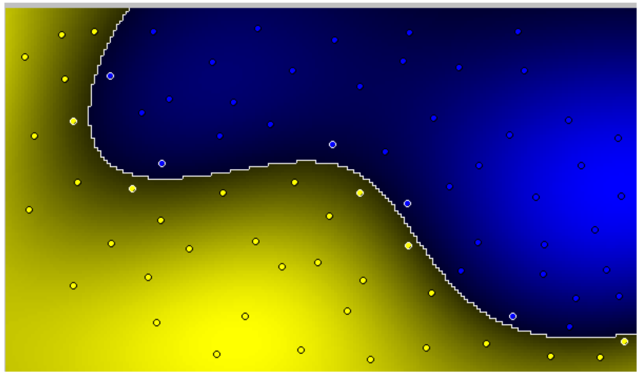
- Has numerical difficulties approaching 0 or infinity
- A good choice for well known and well conditioned tasks
- One additional parameter (degree *p*)



Kernel Selection

Radial basis functions

- Indicated in general as the best choice in the literature
- One additional parameter (sigma σ)



Sigmoid

- Two additional parameters (*a* and *b*)
- From neural networks
- Not much recommended in the literature

Typical Implementation

1. Apply scaling/normalization on the data
2. Consider RBF kernel
3. Use cross-validation to find the best parameters $\boldsymbol{C}$ and $\boldsymbol{\sigma}$
4. Use the best $\boldsymbol{C}$ and $\boldsymbol{\sigma}$ to train the model by using the whole training set
5. Apply the trained model on unseen data (test data)

Typical Implementation - Problems

- Parameter search can be very time consuming
  → Solution: conduct parameter search hierarchically
- Search ranges for C and σ are tricky to choose
  → Solution: literature suggests using exponentially growing values like $C = 2^{[-5..15]}$ and $\sigma = 2^{[-15..5]}$
- RBF kernels are sometimes subject to overfitting
  → Solution: use high degree polynomials kernels
- Parameter search must be repeated for every chosen features; there no reuse of computations
  → Solution: compare features on random subsets of the entire dataset to contain computational cost

SVM Facts

- Were originally proposed by Boser, Guyon and Vapnik in 1992 and gained increasing popularity in late 1990s.
- Are still among the best performers for several classification tasks ranging from text to genomic data.
- Can be applied to complex data types beyond feature vectors (e.g., graphs, sequences, relational data) by designing kernel functions for such data.
- Have been extended to several tasks such as regression [Vapnik et al. ’97], principal component analysis [Schölkopf et al. ’99], etc.

(Some) Applications

- Facial recognition
- Content–based image retrieval
- Facial expression classification
- Hand–written text interpretation
- 3D object recognition
- Texture Classification
- Text classification
- Traffic prediction
- Disease identification
- Gene sequencing
- Protein folding
- Weather forecasting
- Earthquake prediction
- Automated diagnosis
- Many more…

Face recognition demo as seen in Viola, Jones, "Robust Real-time Object Detection", IJCV 2001

Cigdem Beyan
cigdem.beyan@univr.it
https://cbeyan.github.io/