Centre de Visió per Computador

Universitat Autònoma de Barcelona

# Master in Computer Vision Barcelona

**Module:** M3. Machine Learning for Computer Vision

**Lecture:** The Support Vector Machine (SVM): Basic Concepts

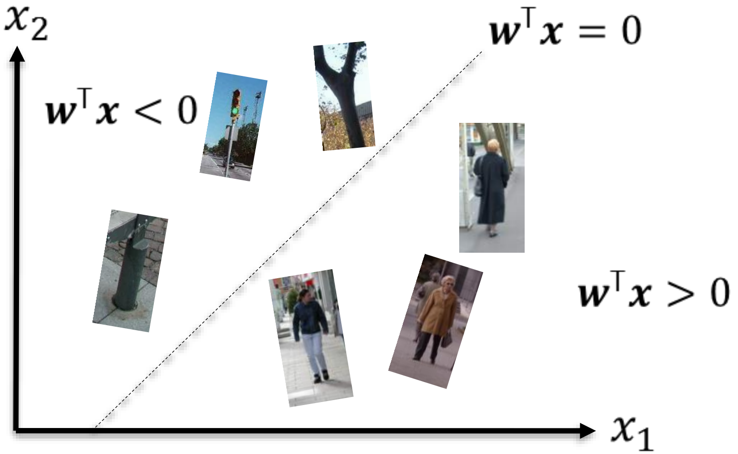
**Lecturer: Ramon Baldrich / Fernando Vilariño**

# The Support Vector Machine

Master in Computer Vision Barcelona

1. Understanding linear classifiers
2. The support vector machine as a linear classifier
    1. Maximal margin solution
    2. Support vectors
3. Tolerance to errors
    1. Soft Margins
    2. Slack variables and the regularization parameter *C*
4. Non linearly-separable data sets
    1. Higher dimension projection
    2. The kernel trick
5. Mathematical development
    1. Hyperplane solution
    2. Maximal margin condition
    3. The Quadratic Optimisation Problem

We already know examples of binary classifiers in a 2D space:




$$\boldsymbol{w} = (w_0, w_1, w_2)^{\mathsf{T}}$$

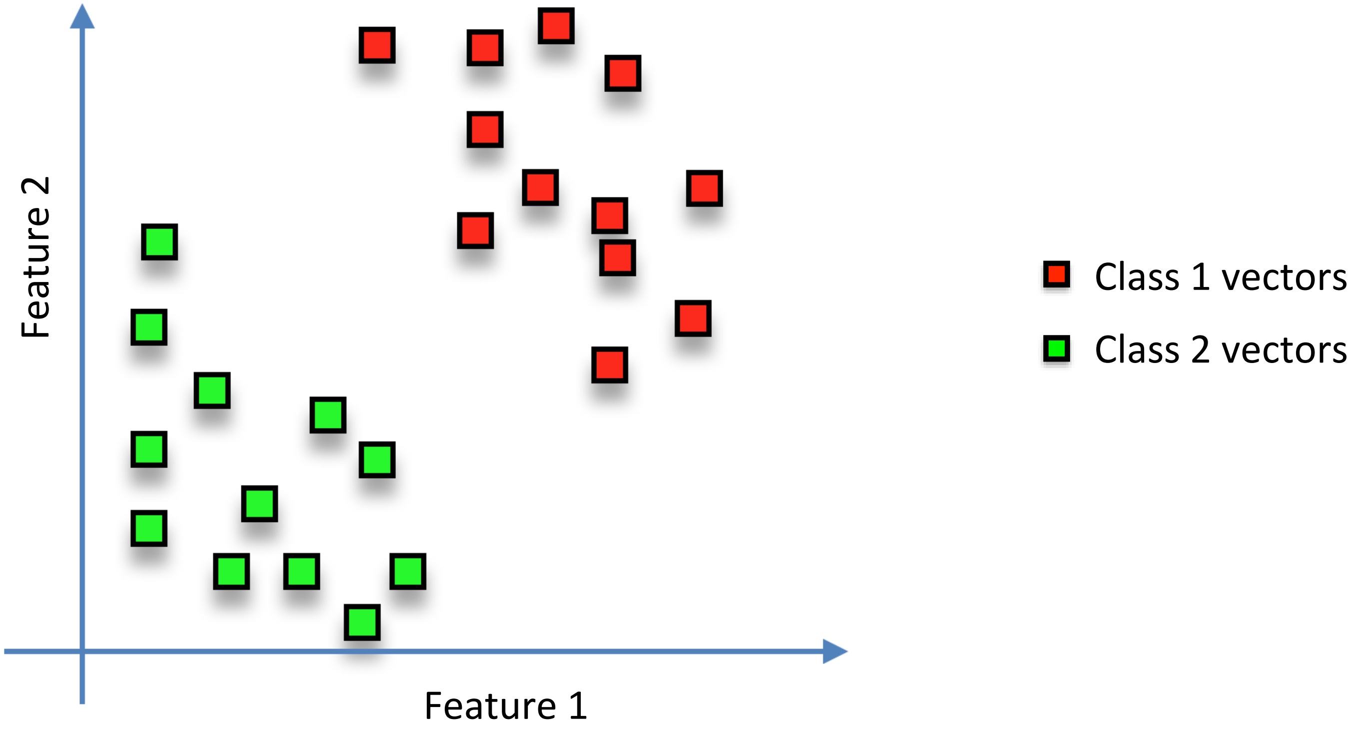
$$\boldsymbol{x} = (1, x_1, x_2)^{\mathsf{T}}$$

$\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} > 0 \longrightarrow$ pedestrian

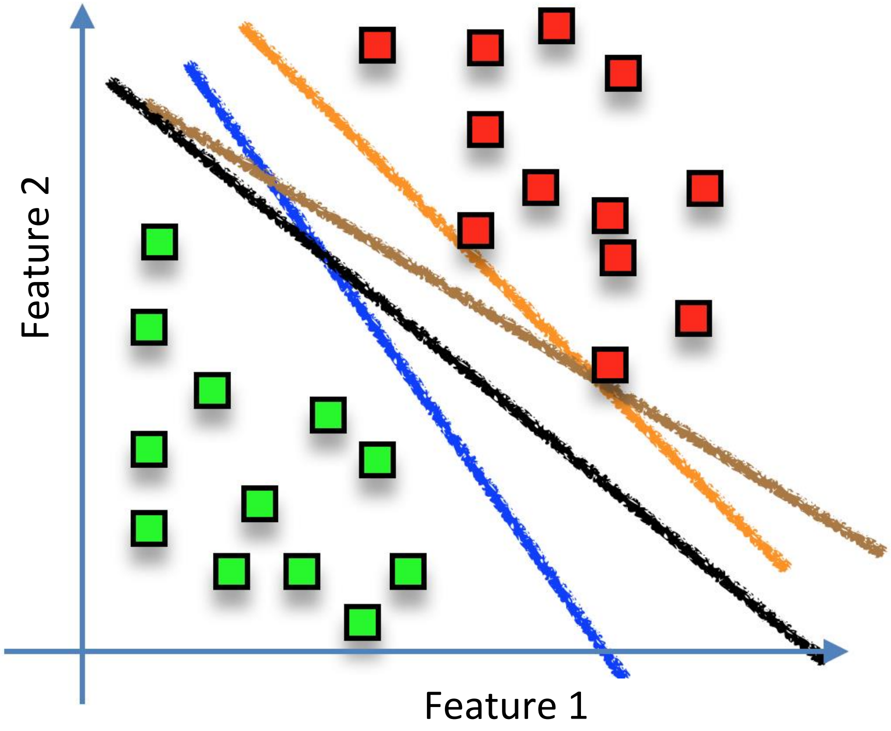
$\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} < 0 \longrightarrow$ Non-pedestrian

$\boldsymbol{w} = (w_0, w_1, w_2)^{\mathsf{T}}$ Design decision

There is no unique way of drawing lines (**solution hyperplanes**) dividing the feature space for the 2-class problem.



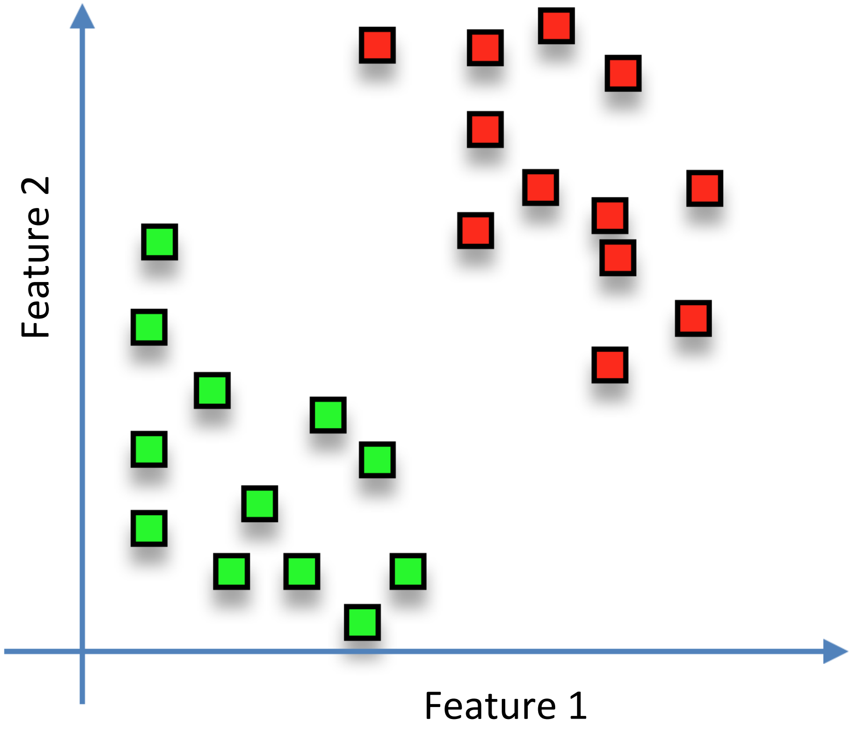
There is no unique way of drawing lines (**solution hyperplanes**) dividing the feature space for the 2-class problem.




- **Generative Models**: Estimation of the probability density (*pdf*): *naïve Bayes*, *Fisher discriminant anaylisis*.
- **Discriminative Models**: Use of the training dataset, no need of the *pdf*: logistic regression, neural networks and ***Support Vector Machines (SVM)***.

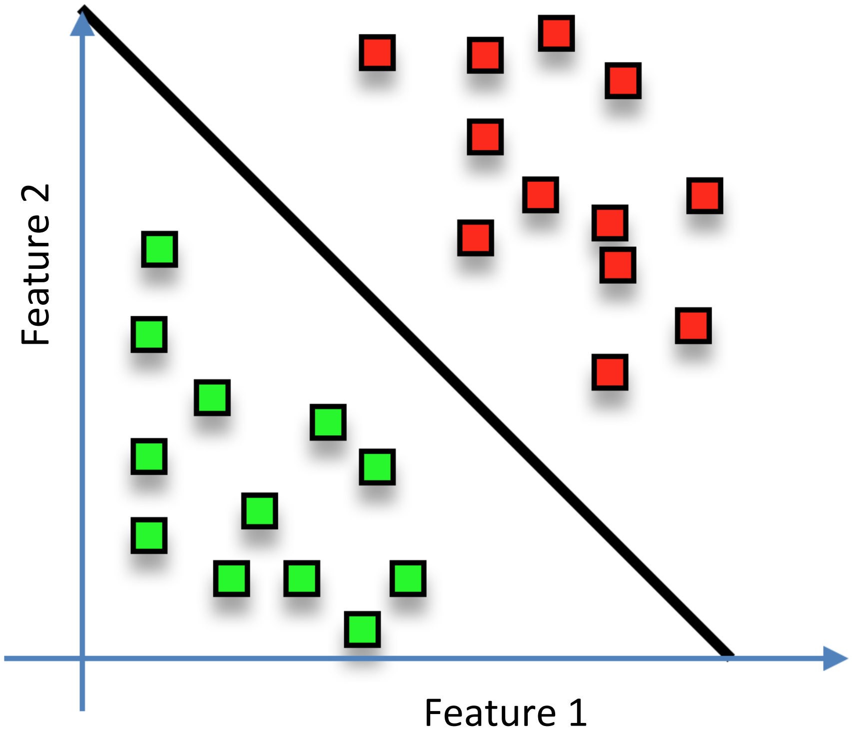
Support Vector Machines (SVM):

***Linear*** *classifier based on* ***maximal margin*** *from* ***the support vectors.***



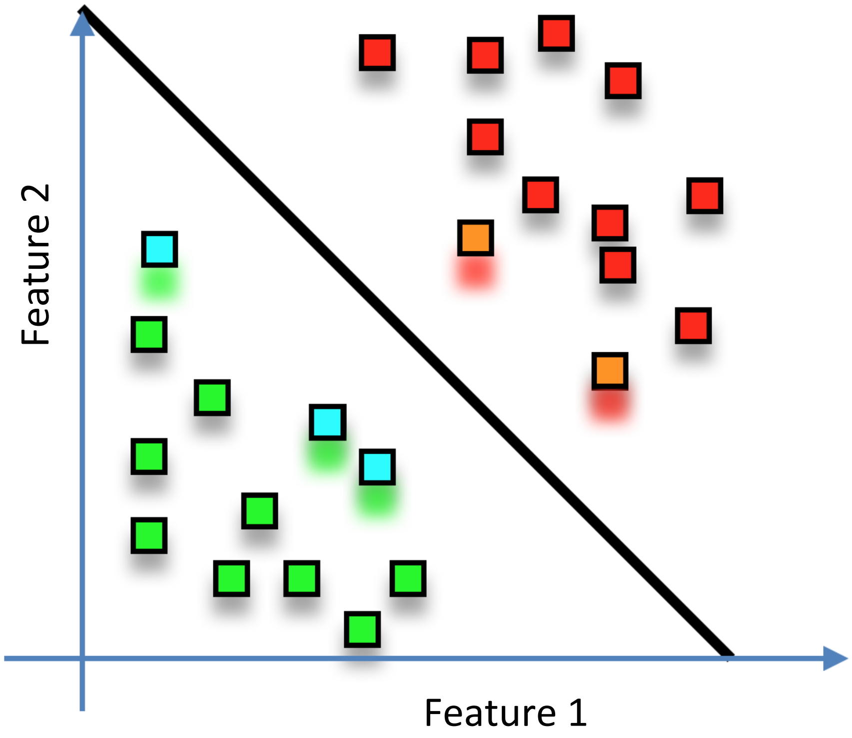
Support Vector Machines (SVM):

***Linear*** *classifier based on* ***maximal margin*** *from* ***the support vectors.***




The decision border of a SVM is a **hyperplane**.

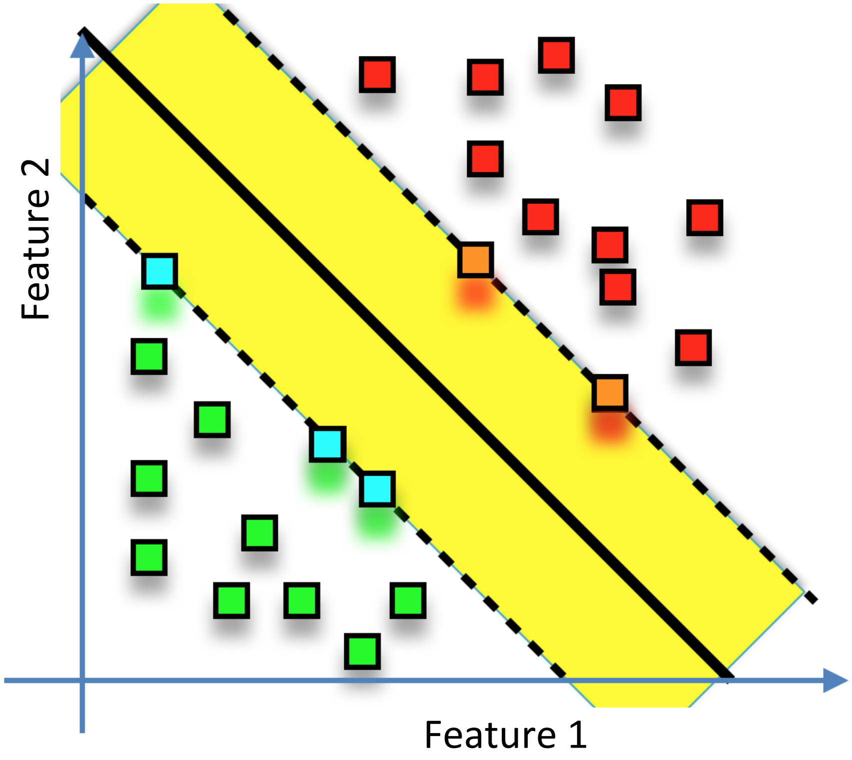
SVM:***Linear*** *classifier based on* ***maximal margin*** *from* ***the support vectors.***




Only a few samples from the training set (**the support vectors**) are used to calculate the solution.

- Class 1 vectors
- Class 1 support vectors
- Class 2 vectors
- Class 2 support vectors

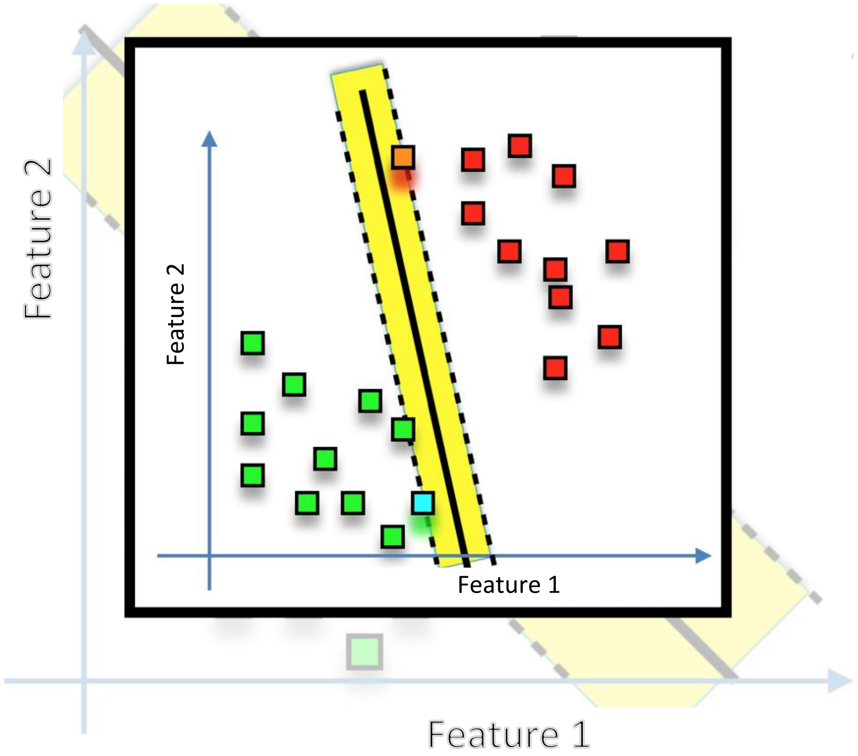
SVM: ***Linear*** *classifier based on* ***maximal margin*** *from* ***the support vectors.***




The separating hyperplane is obtained from the solution of an optimization problem: maximal distance between the 2 hyperplanes containing the support vectors of both classes (**maximal margin**).

- Class 1 vectors
- Class 1 support vectors
- Class 2 vectors
- Class 2 support vectors
- Margin
- Solution hyperplane
- Support vectors hyperplanes

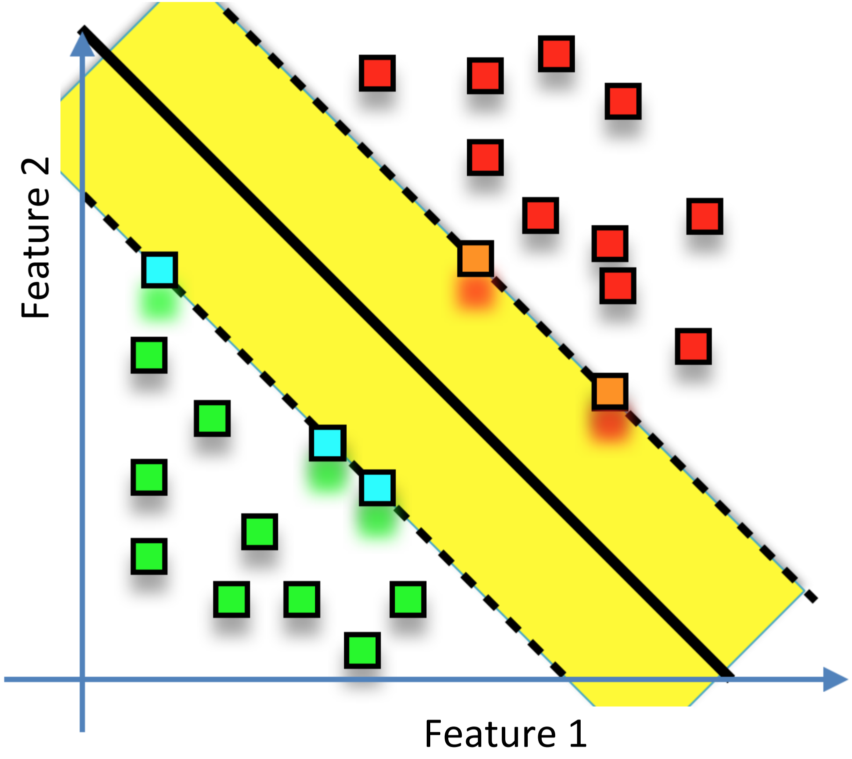
SVM:***Linear*** *classifier based on* ***maximal margin*** *from* ***the support vectors.***




Any other choice of support vectors generate a lower margin.

- Class 1 vectors
- Class 1 support vectors
- Class 2 vectors
- Class 2 support vectors
- Margin
- Solution hyperplane
- Support vectors hyperplanes

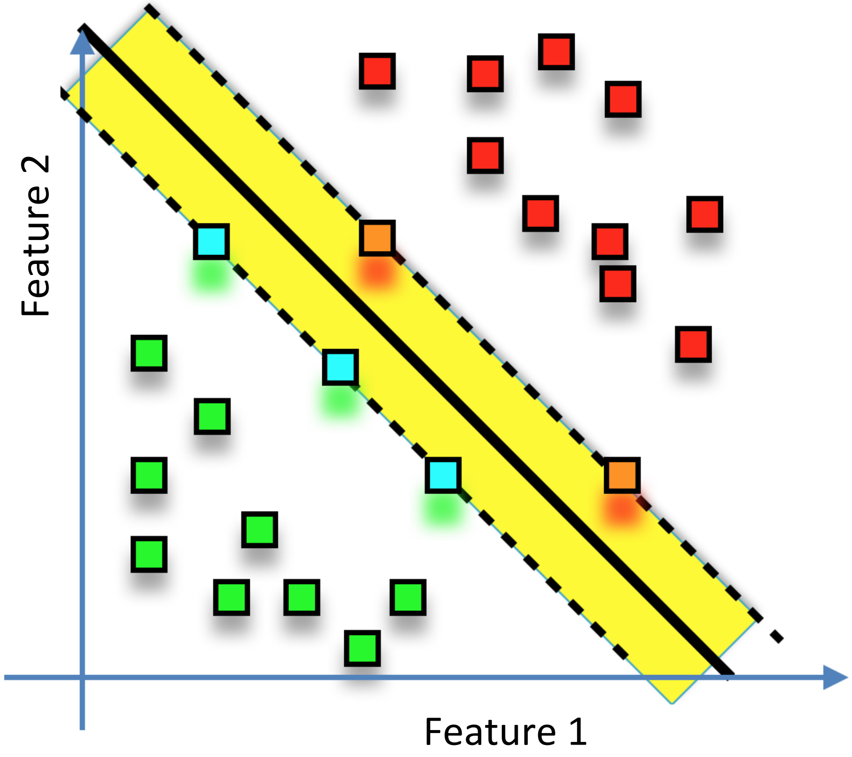
The **support vectors**:




The support vector approach reduces the **overfitting**.

If the samples associated to the support vectors are moved, the SVM solution is **changed**.

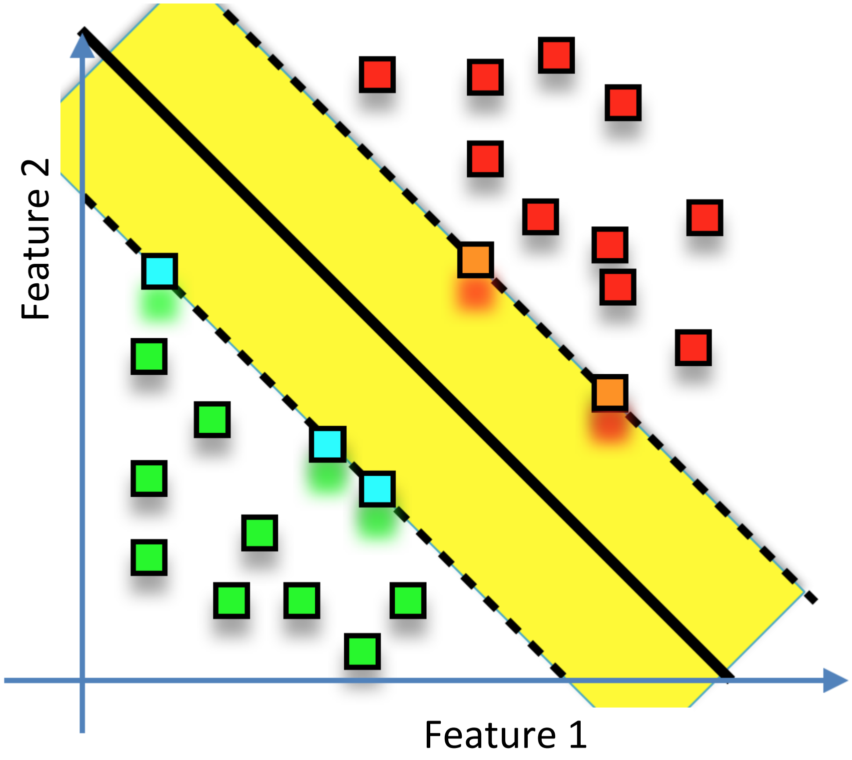
The **support vectors**:




The support vector approach reduces the **overfitting**.

If the samples associated to the support vectors are moved, the SVM solution is **changed**.

The **support vectors**:

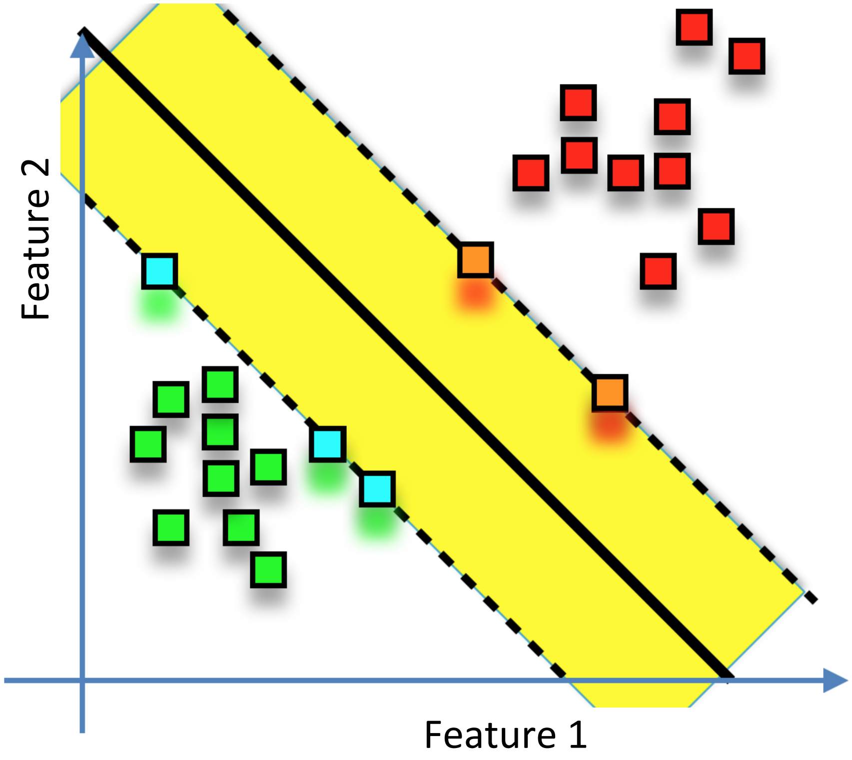



The support vector approach reduces the **overfitting**.

If the samples associated to the support vectors are moved, the SVM solution is **changed**.

On the contrary, if the rest of the samples are moved, the SVM solution is not changed.

The **support vectors**:

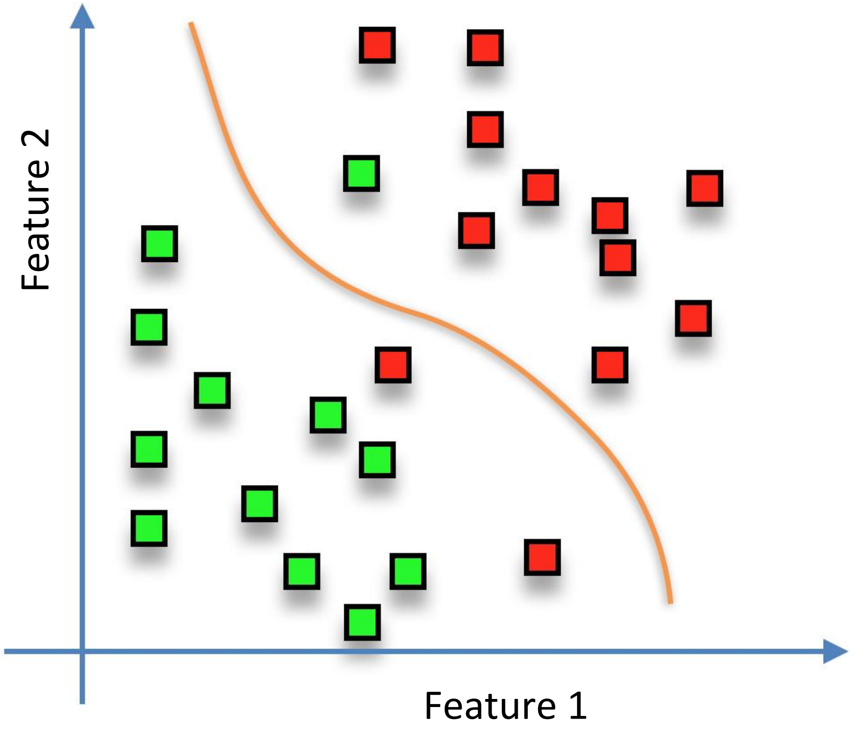



The support vector approach reduces the **overfitting**.

If the samples associated to the support vectors are moved, the SVM solution is **changed**.

On the contrary, if the rest of the samples are moved, the SVM solution is not changed.

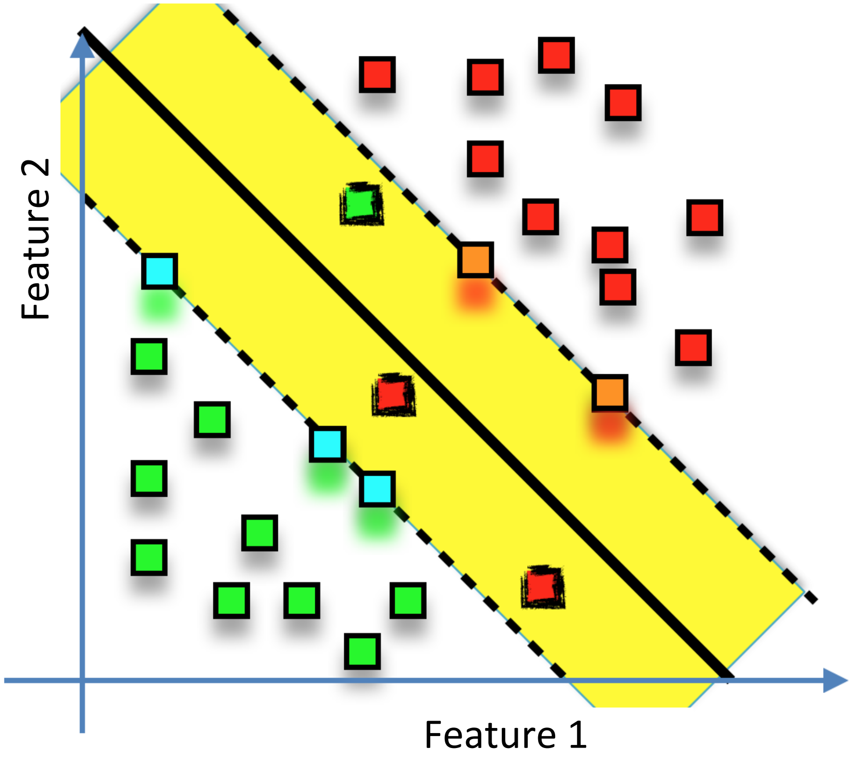
## Non linearly separable datasets: **Soft margin**




- The approach shown is only working for **linearly separable** classes.

  In order make the SVM efficient for non linearly separable datasets:
  - The maximal margin condition is relaxed (**soft margin**).
  - The feature space is mapped into a higher dimensional space in which the dataset is linearly separable (**kernel trick**).

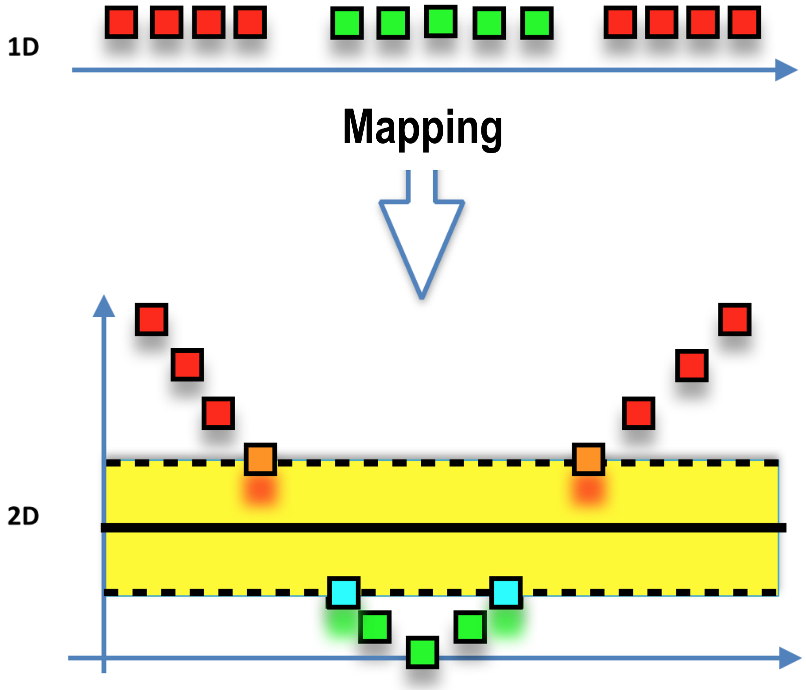
Non linearly separable datasets: **Soft margin**.




- The soft margin solution implies to implement a **tolerance to errors**. (vectors violating the margin condition).
- This tolerance is controlled by the ***slack variables.***

Vectors violating the margin condition

Non linearly separable datasets: **Kernel trick**.




- Non linearly separable datasets can be transformed into linearly separable ones in a **higher dimension space**.
- **It is not necessary to define the mapping function** between the original space and the higher dimension space.
- It is only needed to define the scalar product (**kernel**).

## Key concepts:

- The SVM as a ***linear classifier***.
- The SVM solution as an optimization problem***: maximal margin***.
- The ***support vectors*** as the samples defining the maximal margin.
- The ***slack variables****,* allowing to relax the maximal margin condition.
- The ***kernel trick****,* allowing to work with non linearly separable datasets.