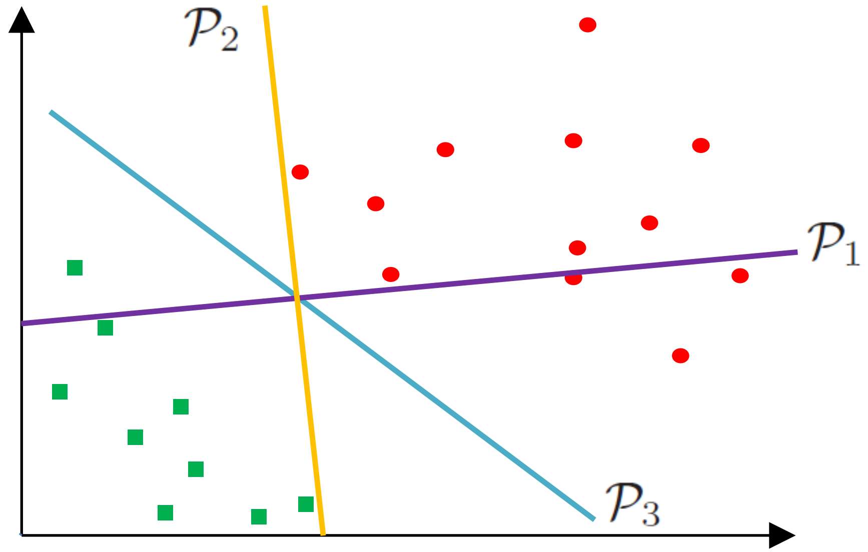
Support Vector Machines

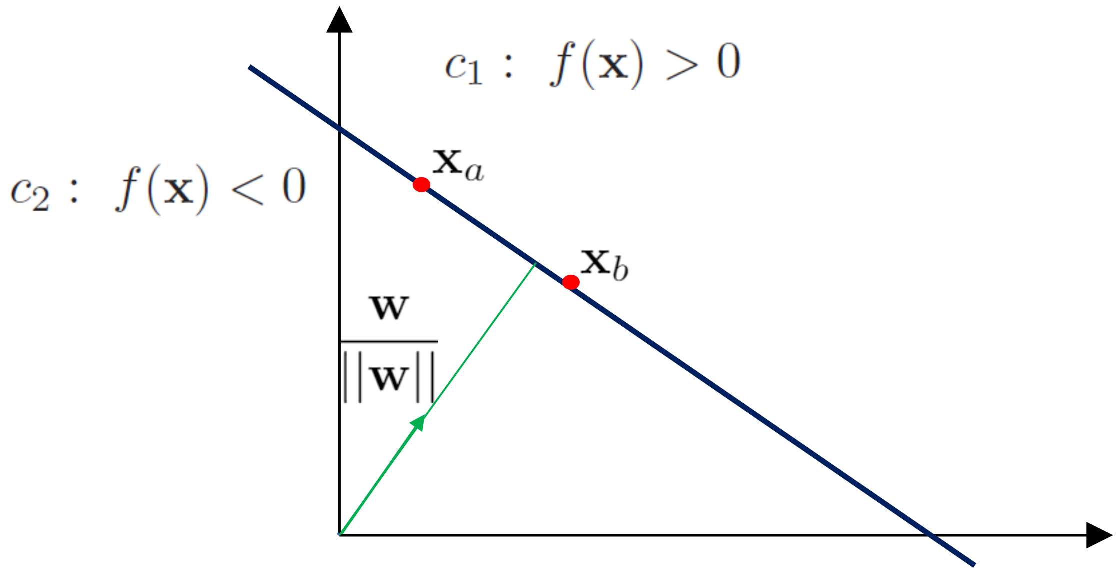
Dripta Mj
Department of Mathematics
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur Math, India

Hyperplanes



- Find a hyperplane that separates the classes.
  - $\mathcal{P}_1$ does not separate the classes.
- Many hyperplanes are possible that separates the classes.
  - $\mathcal{P}_2$ separates the classes but with small separation between them.
  - $\mathcal{P}_3$ also separates the classes with large separation.

Two classes – linear discriminant



- Linear discriminant function can written in the form:

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

- Consider two points – $\mathbf{x}_a$ and $\mathbf{x}_b$ – on the decision surface $f(\mathbf{x}) = 0$.

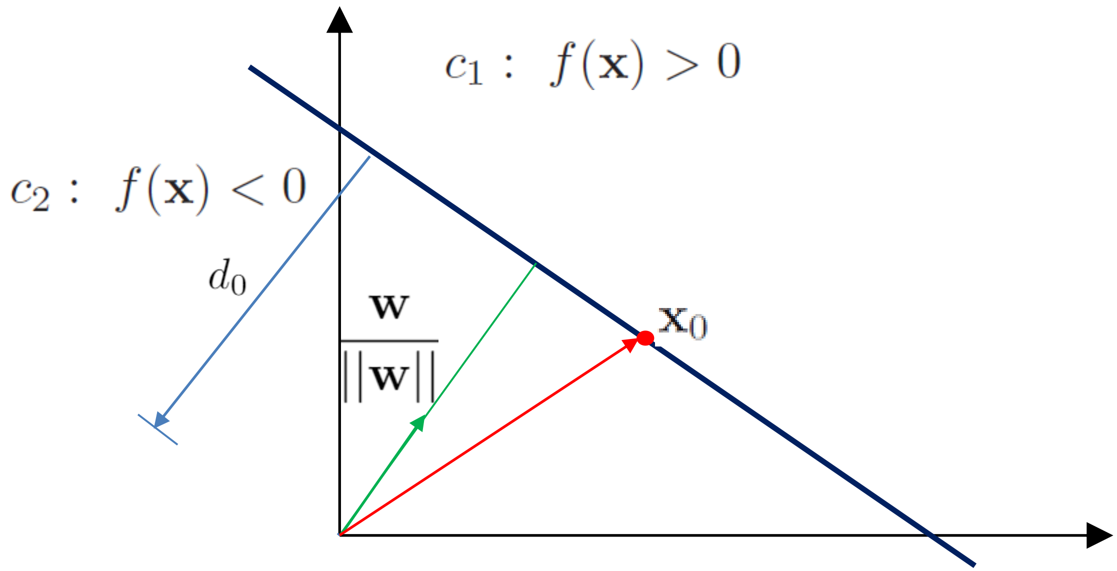
$$f(\mathbf{x}_a) = 0 \ \Rightarrow \mathbf{w}^{\mathrm{T}}\mathbf{x}_a + w_0 = 0$$

$$f(\mathbf{x}_b) = 0 \ \Rightarrow \mathbf{w}^{\mathrm{T}}\mathbf{x}_b + w_0 = 0$$

$$\mathbf{w}^{\mathrm{T}}(\mathbf{x}_a - \mathbf{x}_b) = 0$$

- Therefore the vector $\mathbf{w}$ is orthogonal to all vectors lying on the decision surface.

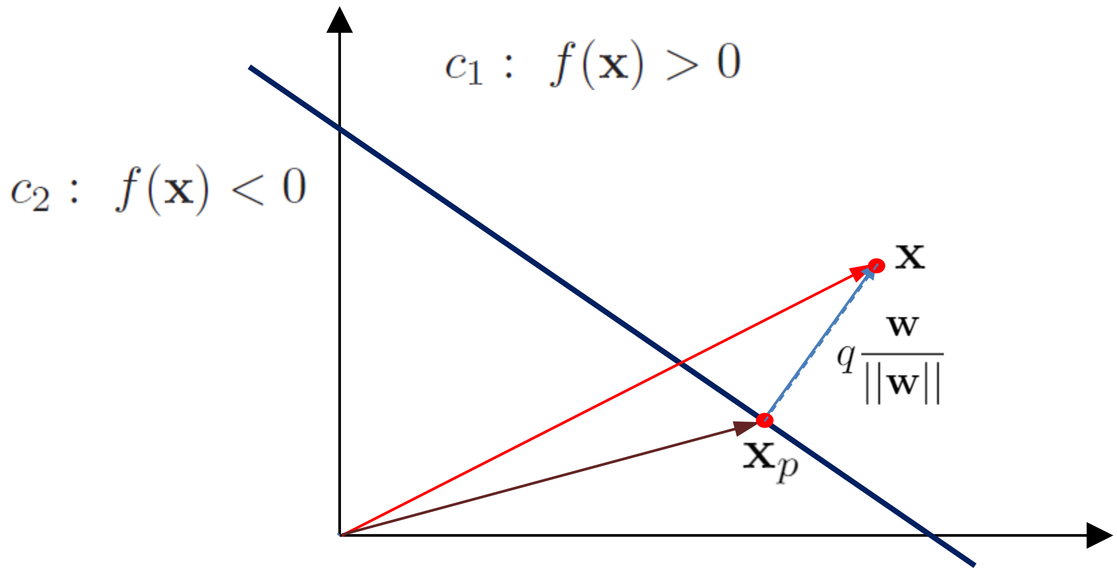
Distance from origin



- Want to compute the distance d_0 between the decision surface and the origin.
- Consider a point (say $\mathbf{x}_0$) on the decision surface, then d_0 can be computed as

$$\begin{aligned} d_0 &= \frac{\mathbf{w}^{\mathrm{T}}}{||\mathbf{w}||}(\mathbf{x}_0 - \mathbf{0}) \\ &= -\frac{w_0}{||\mathbf{w}||} \qquad\qquad (\text{since } f(\mathbf{x}_0) = 0) \end{aligned}$$

Modelling distance from an arbitrary point



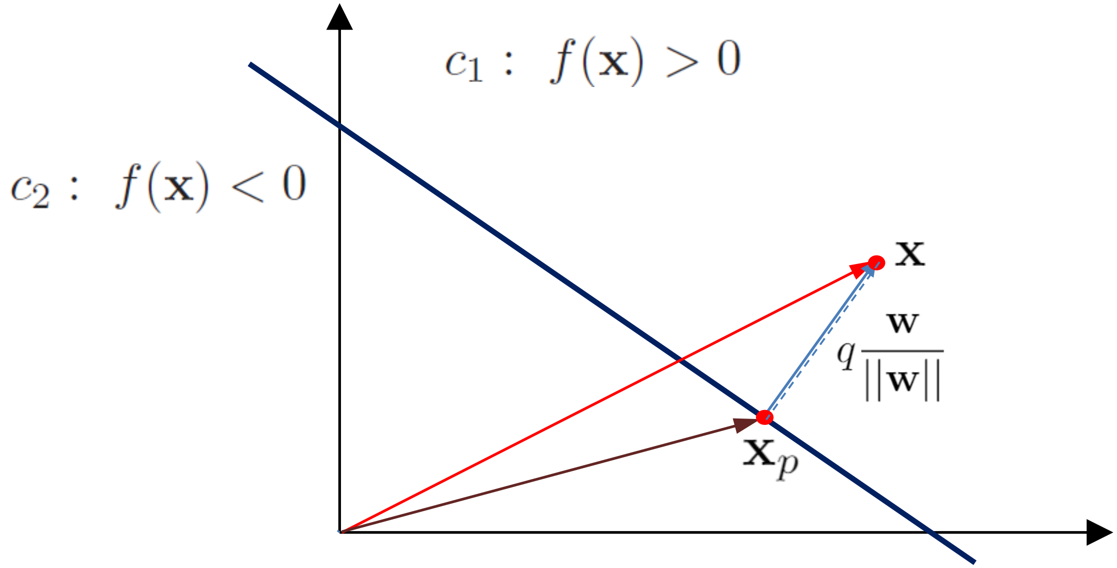
- Consider an arbitrary point $\mathbf{x}$ in the feature space.
- Suppose $\mathbf{x}_p$ is the orthogonal projection of the point $\mathbf{x}$ on the decision surface, which means

$$f(\mathbf{x}_p) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_p + w_0 = 0$$

- Let q be the distance between $\mathbf{x}$ and $\mathbf{x}_p$, then can write

$$\mathbf{x} = \mathbf{x}_p + q\frac{\mathbf{w}}{||\mathbf{w}||}$$

Signed orthogonal distance



- Multiplying both sides of the equation by $\mathbf{w}^{\mathrm{T}}$, we have

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} = \mathbf{w}^{\mathrm{T}}\mathbf{x}_p + q\frac{\mathbf{w}^{\mathrm{T}}\mathbf{w}}{||\mathbf{w}||}$$

$$f(\mathbf{x}) - w_0 = -w_0 + q\frac{||\mathbf{w}||^2}{||\mathbf{w}||}$$

$$\Rightarrow \qquad q = \frac{f(\mathbf{x})}{||\mathbf{w}||}$$

Margin

- Geometric margin γ_n is the perpendicular distance from the point $\mathbf{x}^{(n)}$ to the hyperplane

$$\gamma_n = y^{(n)} \left(\frac{\mathbf{w}^{\mathrm{T}} \mathbf{x}^{(n)} + w_0}{||\mathbf{w}||} \right)$$

- Margin is defined as the minimum of the geometric margin.

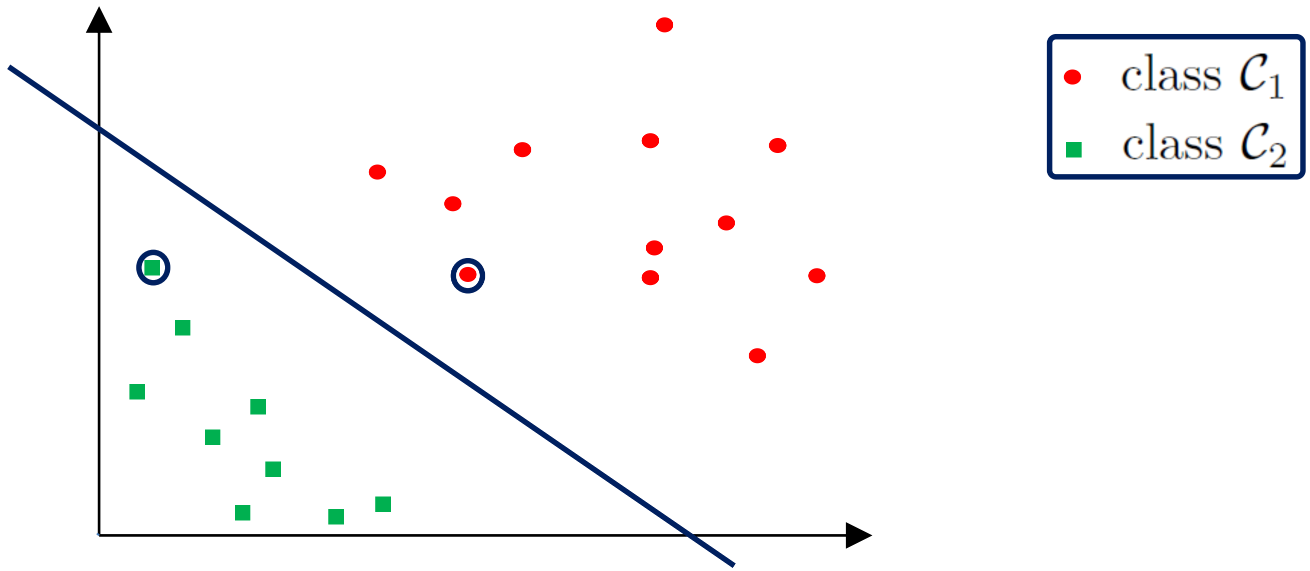
$$\gamma = \min_{\mathcal{D}} \gamma_n$$

- Functional margin $\widehat{\gamma}_n$ of an example $(\mathbf{x}^{(n)}, y^{(n)})$ with respect to the hyperplane is

$$\widehat{\gamma}_n = y^{(n)} (\mathbf{w}^{\mathrm{T}} \mathbf{x}^{(n)} + w_0)$$

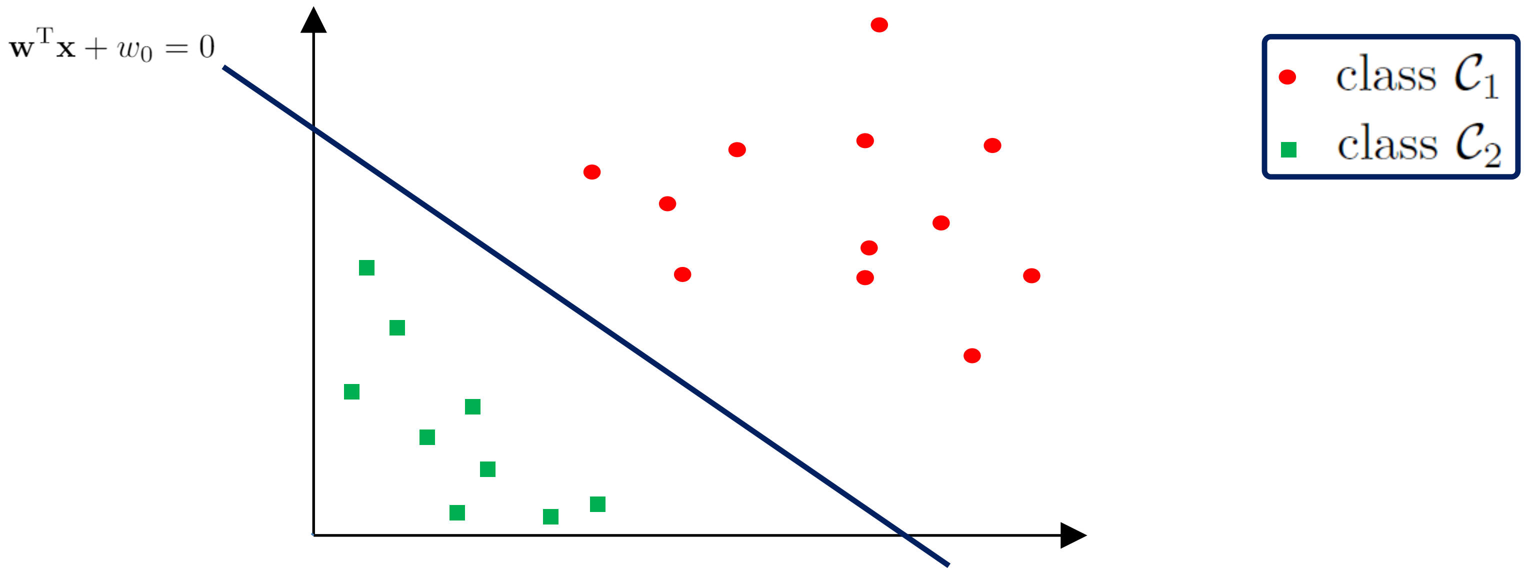
- $+$ve $\widehat{\gamma}_n$ means the example is correctly classified.
- $-$ve $\widehat{\gamma}_n$ means the example is incorrectly classified.

Maximum margin hyperplane



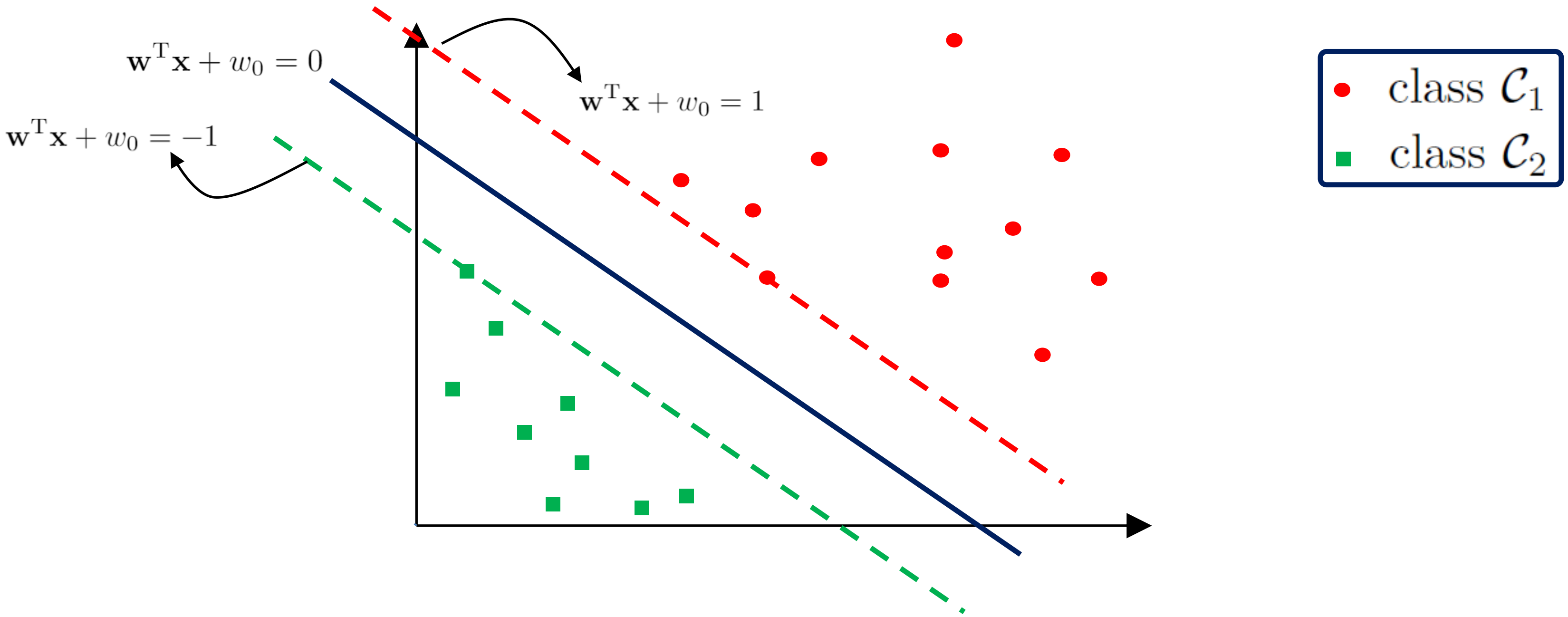
- Learn the hyperplane with the maximum separation.
- Support Vector Machine provide a framework for the learning the maximum margin hyperplane.
- SVM find the most important examples in the training dataset that define the separating hyperplane. These examples are called the “support vectors”.

Intuition



- Separating hyperplane: $f(\mathbf{x}) = 0$ i.e. $\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0$.
- If $f(\mathbf{x}^{(n)}) \geq 0$, then $y^{(n)} = 1$, i.e. $\mathbf{x}^{(n)}$ belongs to class $\mathcal{C}_1$.
  - If $f(\mathbf{x}^{(n)}) >> 0$, then higher is the confidence of $\mathbf{x}^{(n)}$ belonging to class $\mathcal{C}_1$.
- If $f(\mathbf{x}^{(n)}) < 0$, then $y^{(n)} = -1$, i.e. $\mathbf{x}^{(n)}$ belongs to class $\mathcal{C}_2$.
  - If $f(\mathbf{x}^{(n)}) << 0$, then higher is the confidence of $\mathbf{x}^{(n)}$ belonging to class $\mathcal{C}_2$.

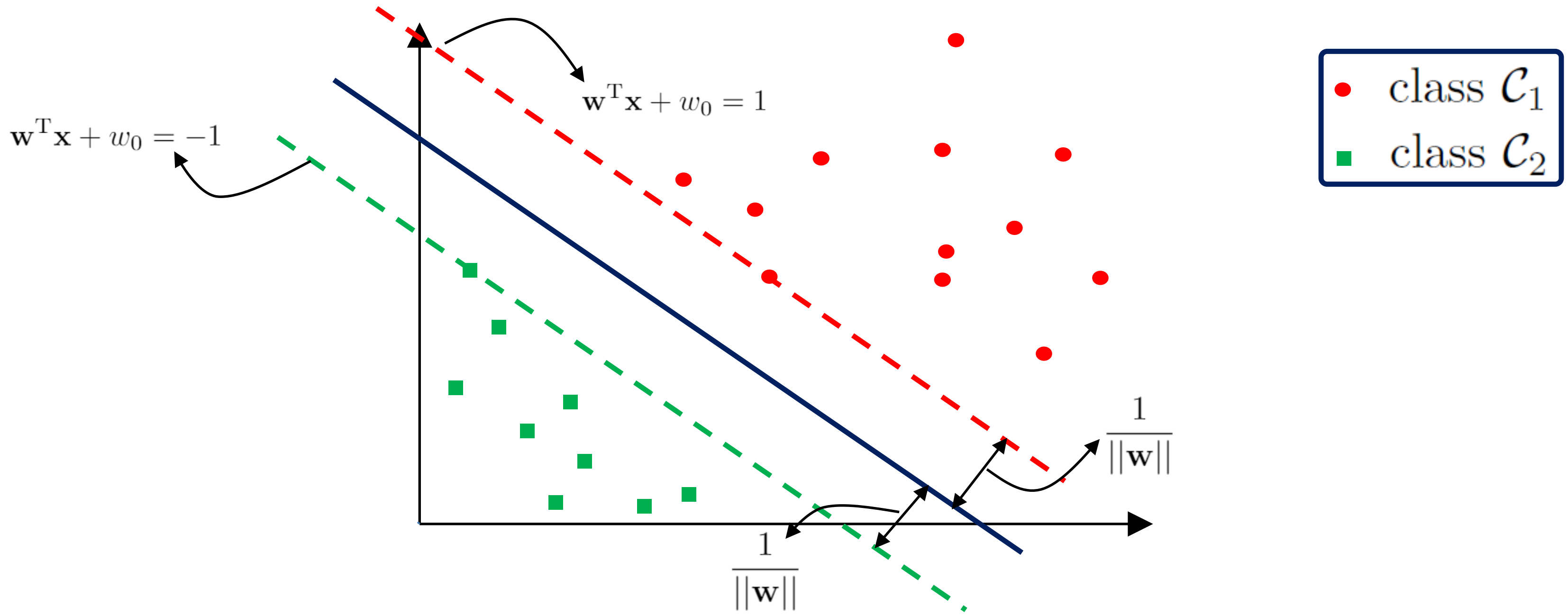
Margin boundaries



- Decision boundary (hyperplane) $\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0$ is to be chosen such that
  - If $\mathbf{x}^{(n)}$ is in $\mathcal{C}_1$ $(y^{(n)} = 1)$: $\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + w_0 \geq 1$
  - If $\mathbf{x}^{(n)}$ is in $\mathcal{C}_2$ $(y^{(n)} = -1)$: $\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + w_0 \leq -1$
- So we have $\min_{n=(1,\ldots,N)} |\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + w_0| = 1$
- Margin condition:

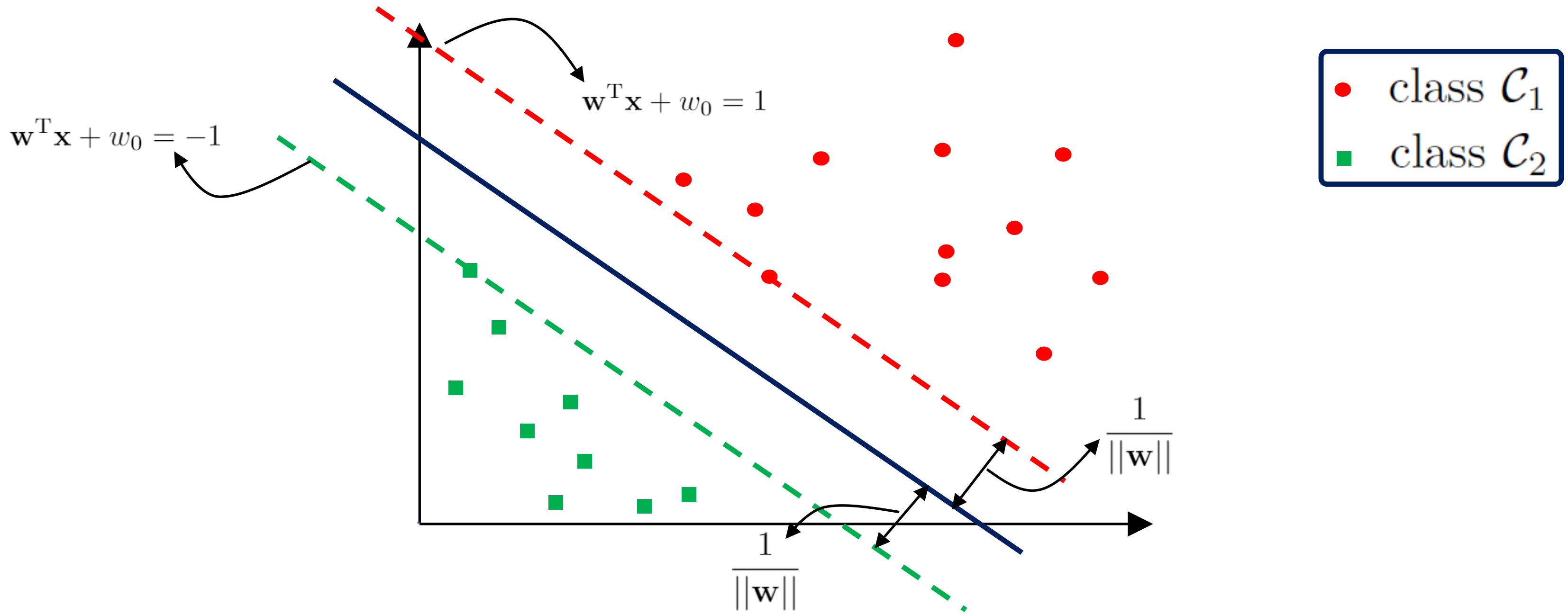
$$y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + w_0) \geq 1, \qquad n = 1, 2, \ldots N$$

Support Vector Machines



- The goal is to find the optimal hyperplane separating the classes that has the maximum margin.
- Recall, the signed distance of a point $\mathbf{x}$ from the decision boundary is given as $\frac{f(\mathbf{x})}{||\mathbf{w}||}$.
- The distance between the two margins is then $\frac{2}{||\mathbf{w}||}$.
- Obtain a decision boundary (hyperplane) with the maximum possible margin.

Hard-margin SVM



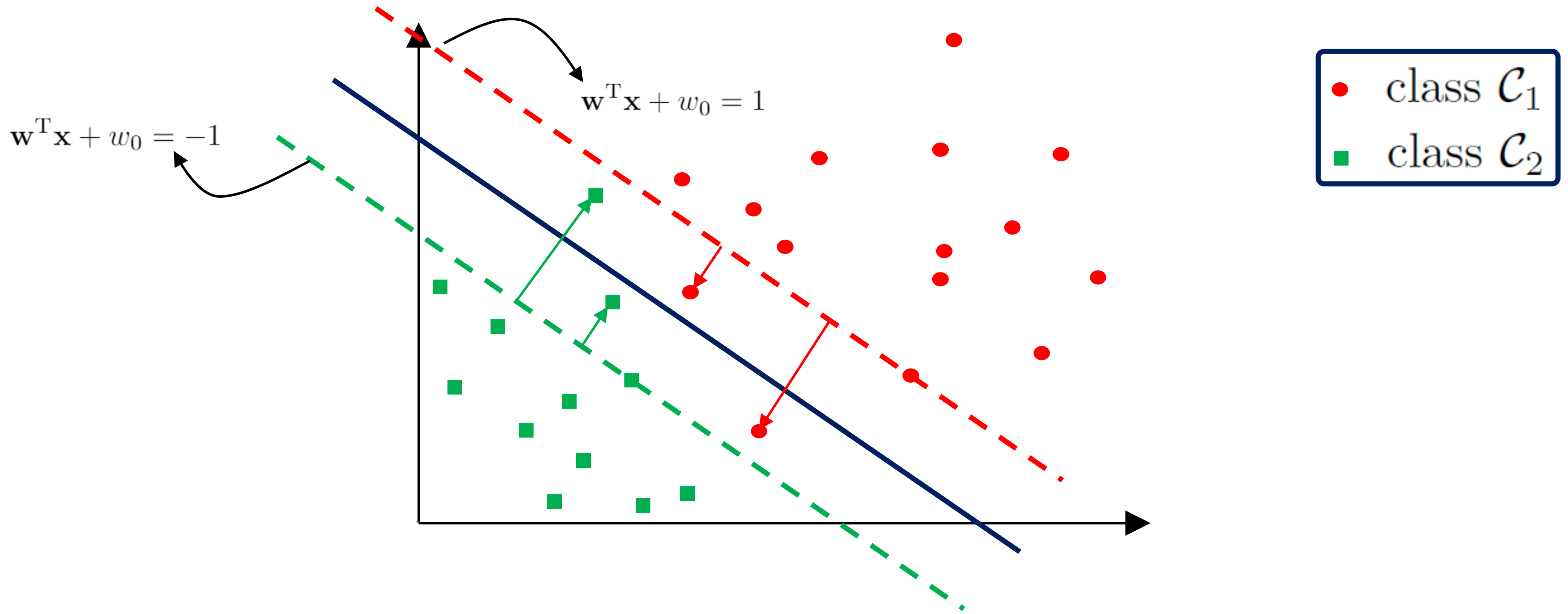
$$\text{Maximize } \frac{1}{||\mathbf{w}||} \longleftrightarrow \text{Minimize } ||\mathbf{w}||^2 \text{ or } \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\min_{\mathbf{w}, w_0} \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\text{subject to} \quad y^{(n)}[\mathbf{w}^T\mathbf{x}^{(n)} + w_0] \geq 1, \qquad n = 1, \dots, N$$

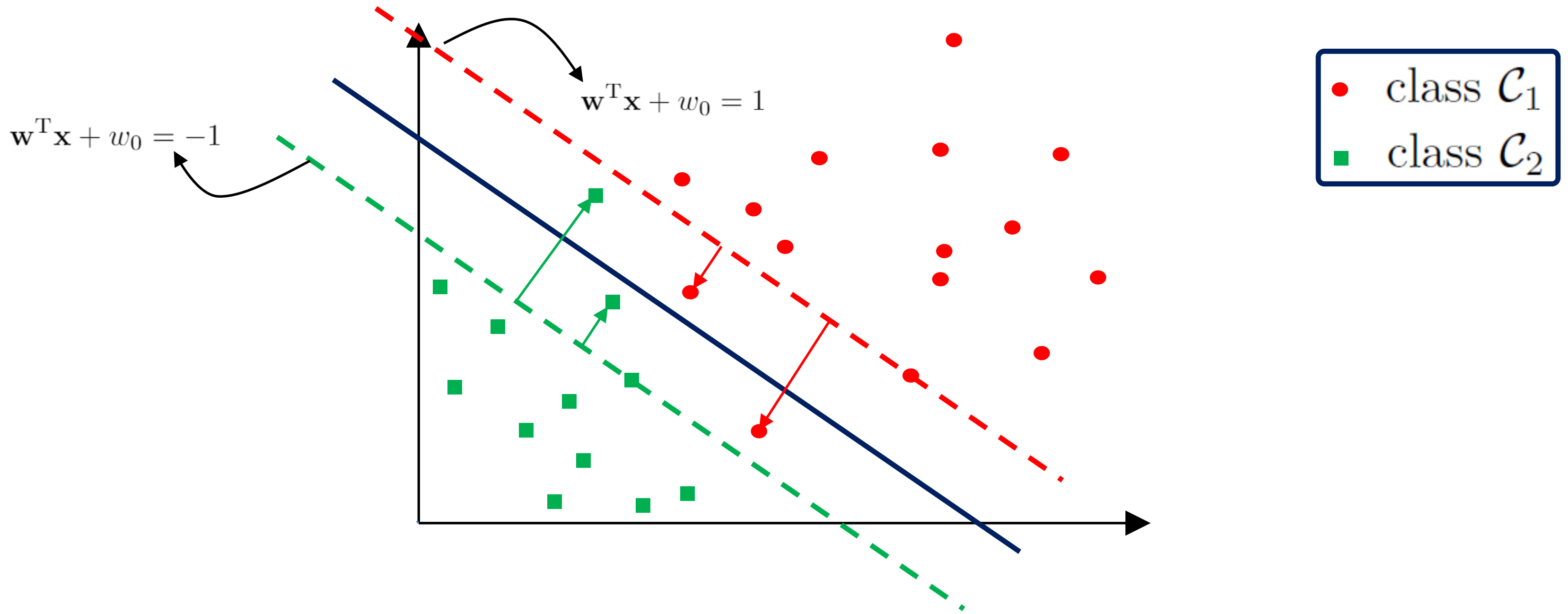
Hard-margin SVM objective

Slack variables



- Data not linearly separable in input space (due to noise).
- For nonlinear boundary, perfect separation of training data in the feature space can lead to poor generalization.
- Method modified to permit a few points to lie on the wrong side of the separating hyperplane.
- Approach: Use slack variables ξ_n, where $n = 1, .., N$, for every data point.

Soft-margin SVM

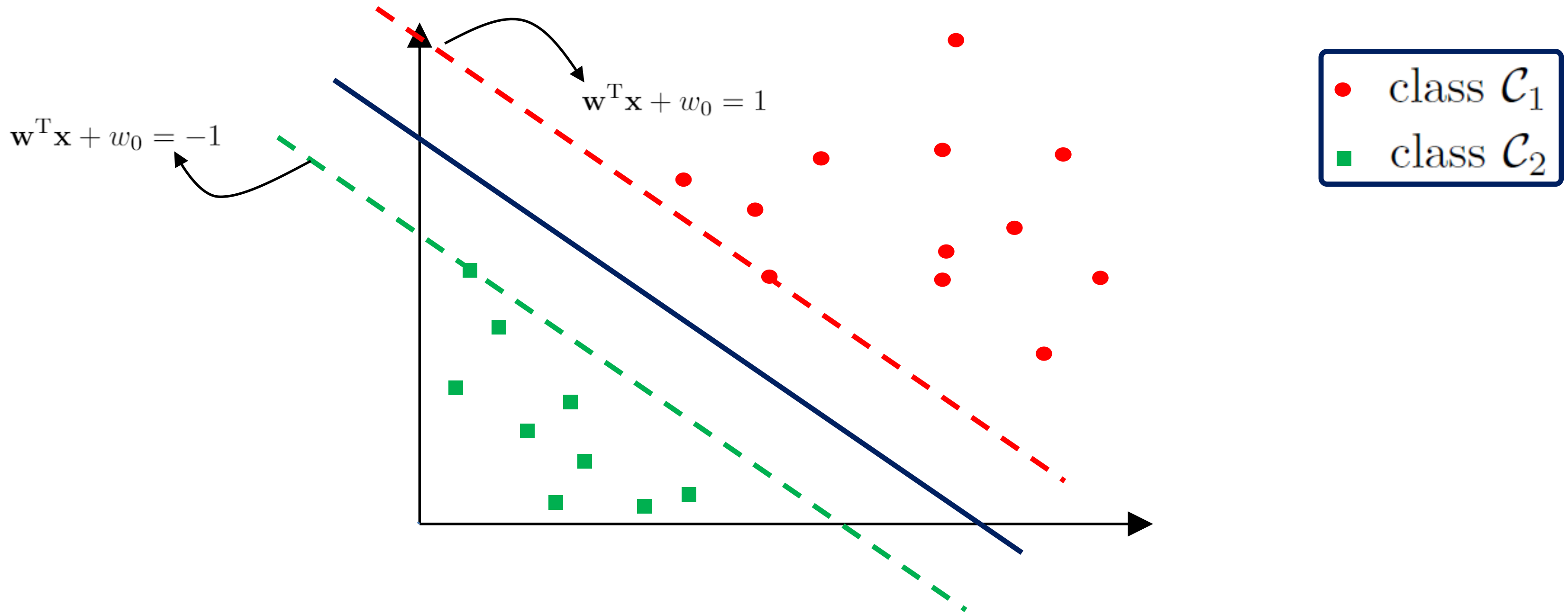


- Each example (say the nth) is associated with a variable $\xi_n \geq 0$ which indicates the degree to which the margin constraint is violated.
- ξ_ns are known as the "slack" variables.
- Soft-margin constraint: $y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + w_0) \geq 1 - \xi_n$.

$$\min_{\mathbf{w}, w_0, \boldsymbol{\xi}} \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{n=1}^{N} \xi_n$$

$$\text{subject to} \quad y^{(n)}[\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + w_0] \geq 1 - \xi_n, \quad \text{and} \quad \xi_n \geq 0, \qquad n = 1, \ldots, N$$

Solution to hard-margin SVM



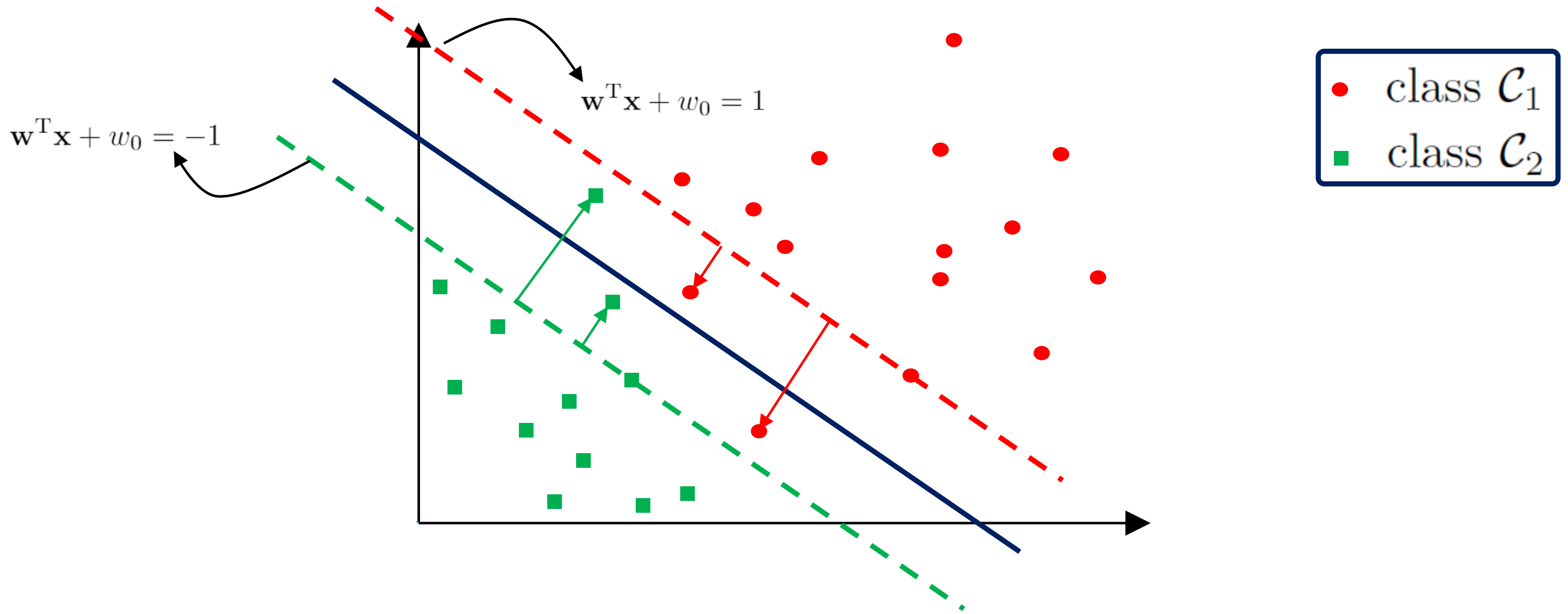
- The solution to $\mathbf{w}$ can be found as

$$\mathbf{w} = \sum_{n=1}^{N} \lambda_n y^{(n)} \mathbf{x}^{(n)}$$

- The intercept of the separating hyperplane is the mean of the two intercepts:

$$w_0 = -\frac{1}{2}\left(\min_{\mathbf{x} \in \mathcal{C}_1} \mathbf{w}^{\mathrm{T}}\mathbf{x} + \max_{\mathbf{x} \in \mathcal{C}_2} \mathbf{w}^{\mathrm{T}}\mathbf{x} \right)$$

Soft-margin support vectors



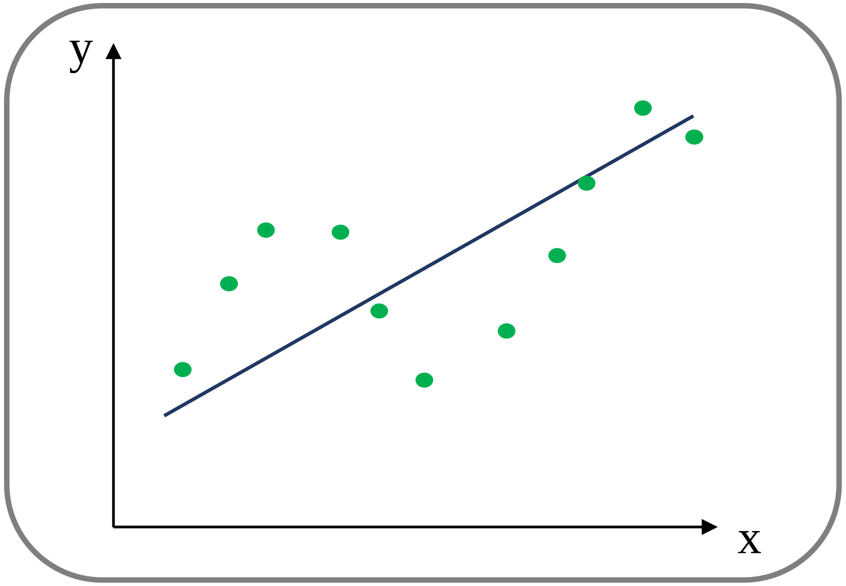
- Three types of support vectors:
  - $\xi_n = 0$: Examples lying on the margin boundaries.
  - $0 < \xi_n < 1$: Examples lying in the margin region and on the correct side of the separating hyperplane.
  - $\xi_n \geq 1$: Examples lying on the wrong side of the separating hyperplane.

Kernel-SVM

THE INTUITION

Using Kernels

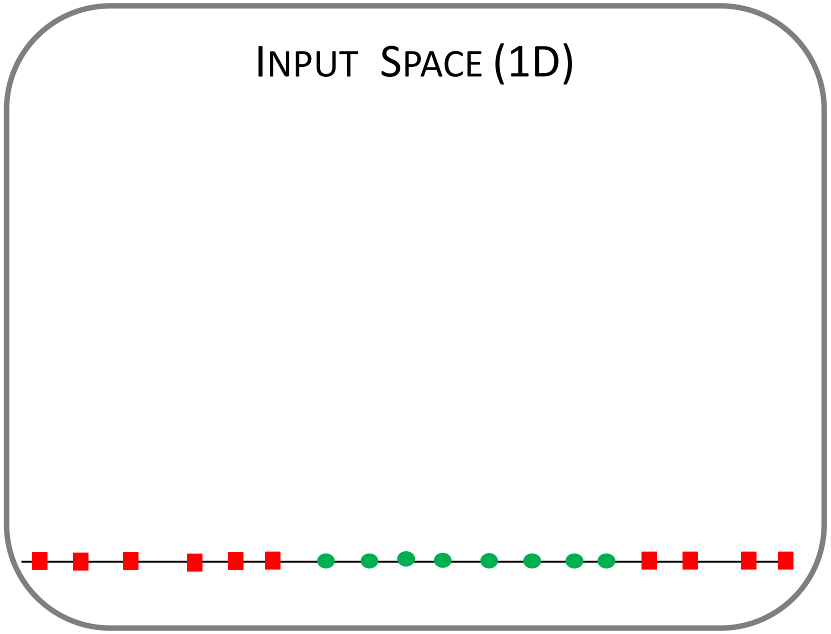
- Structures in real-world data are often non-linear.
  - Linear models are not suitable in such cases.

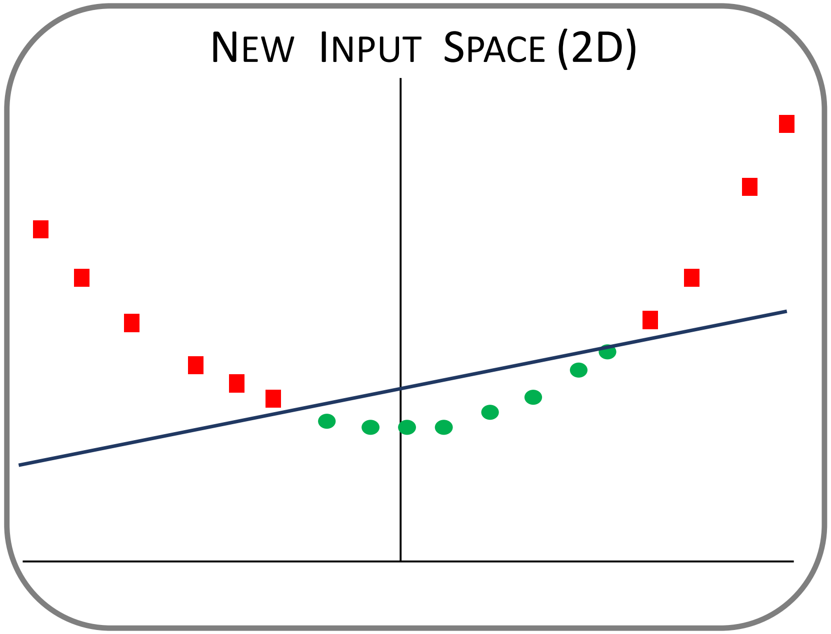


- Kernels project data to a higher dimensional space where the structures are linear.
  - The transformation facilitates application of linear models in the new space.

- Explicit evaluation of feature mappings can be computationally expensive, but kernel methods overcome the issue....

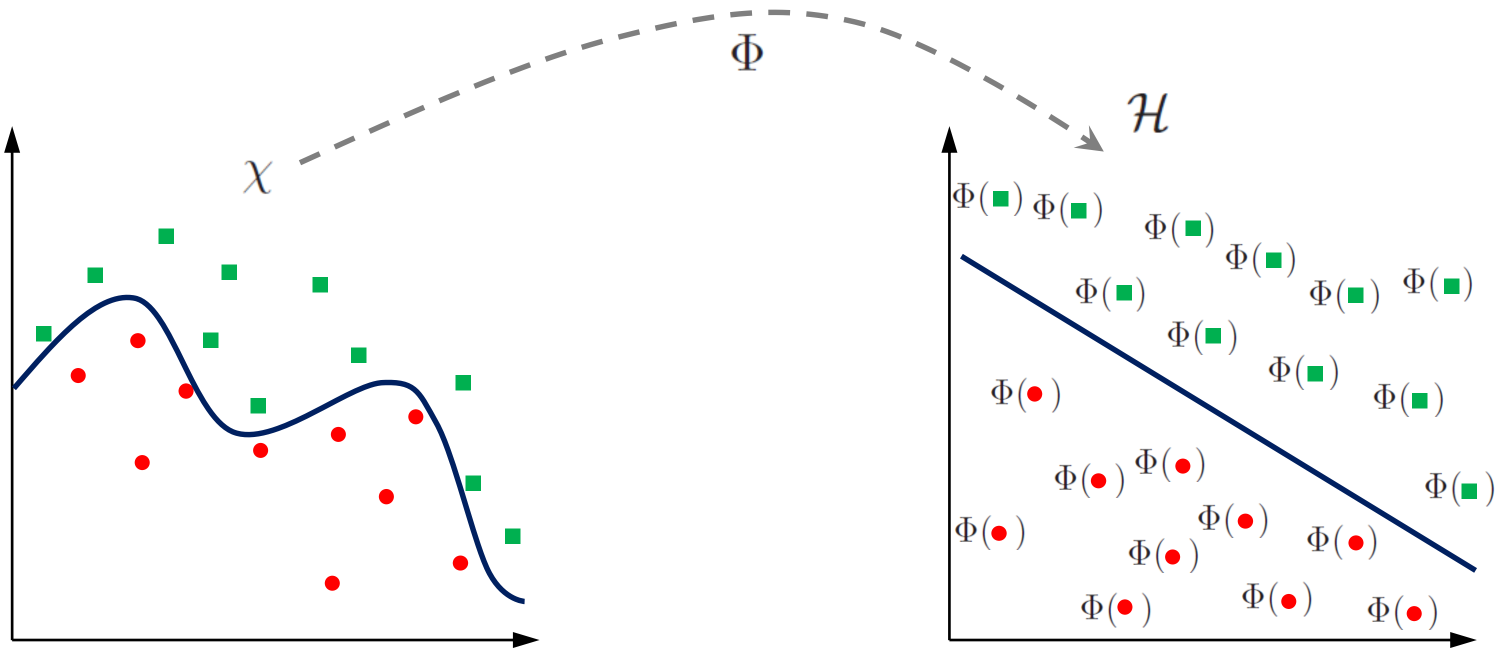
Binary classification problem



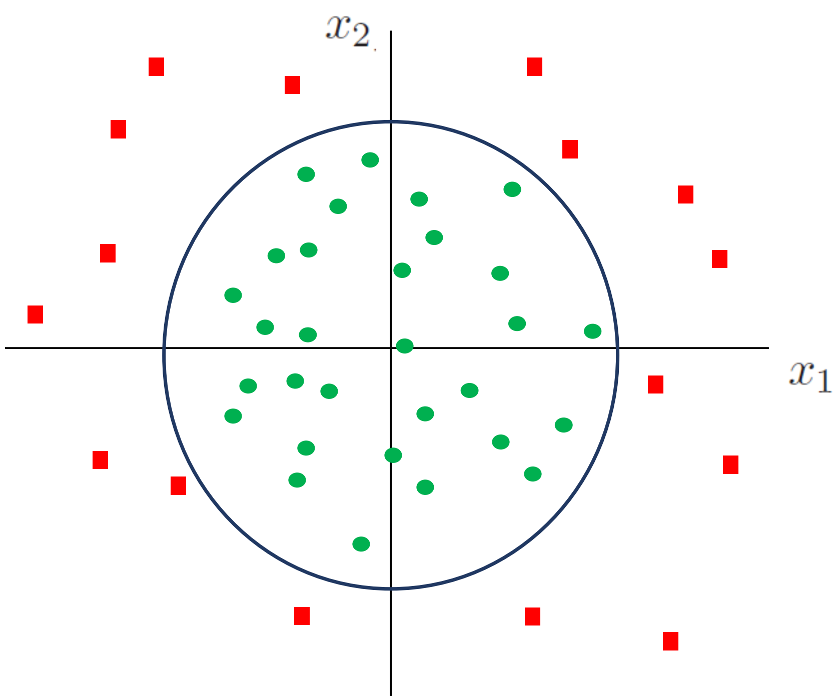


- Linear separation of data is not possible.
- Consider the following mapping: $\Phi(x) : \ x \rightarrow [x, \ x^2]$
- The dimension of the new input space is 2 as there are two features.
- Data linearly separable in the new input space.

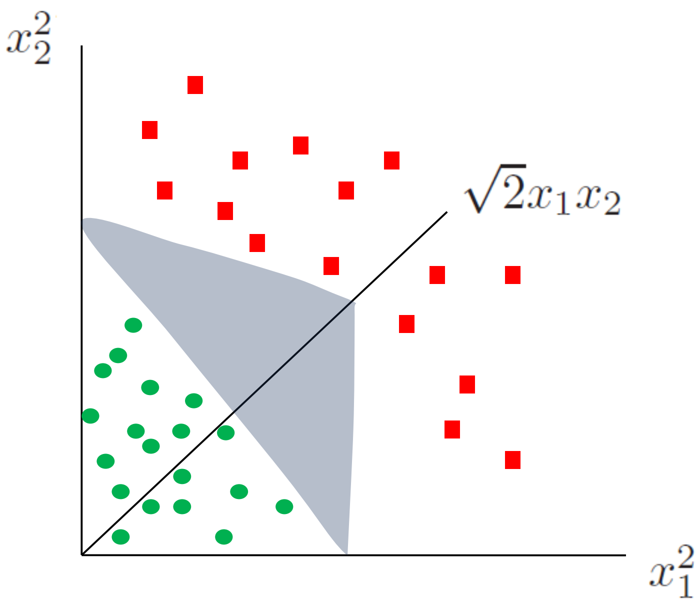
Mapping



Example



- Input space: $\mathbf{x} = [x_1 \;\; x_2]$.
- Data **not** linearly separable in input space.



- Feature space: $\Phi(\mathbf{x}) = [x_1^2 \;\; \sqrt{2}x_1x_2 \;\; x_2^2]$.
- Data linearly separable in feature space.

Multi-class Classification

One-against-all

- Suppose the number of classes is J.
- Approach: Construct J SVM models
  - The jth SVM model is trained such that
    - examples in the jth class are labelled positive
    - examples in all other classes are labelled negative
- Finally we have J decision functions

$$\left(\mathbf{w}^{(1)}\right)^{\mathrm{T}}\mathbf{x} + w_0^{(1)} = 0$$
$$\left(\mathbf{w}^{(2)}\right)^{\mathrm{T}}\mathbf{x} + w_0^{(2)} = 0$$
$$\vdots$$
$$\left(\mathbf{w}^{(J)}\right)^{\mathrm{T}}\mathbf{x} + w_0^{(J)} = 0$$

- Prediction:

$$y^* = \arg \max_{j=[1,2,\ldots,J]} \left(\left(\mathbf{w}^{(j)}\right)^{\mathrm{T}}\mathbf{x}^* + w_0^{(j)}\right)$$

One-against-one

- Construct a classifier using data from two classes.
  - Say the jth classifier comprise mth and nth class.
- Training: In total construct $J(J-1)/2$ classifiers.
- Prediction:
  - Can use a voting strategy
    * If the jth classifier predicts the point to be in class m, then increase vote of class m by one
    * otherwise increase vote of class n by one
  - Repeat the process for all the $J(J-1)/2$ classifiers.
  - Assign example to the class which receives the highest number of votes.