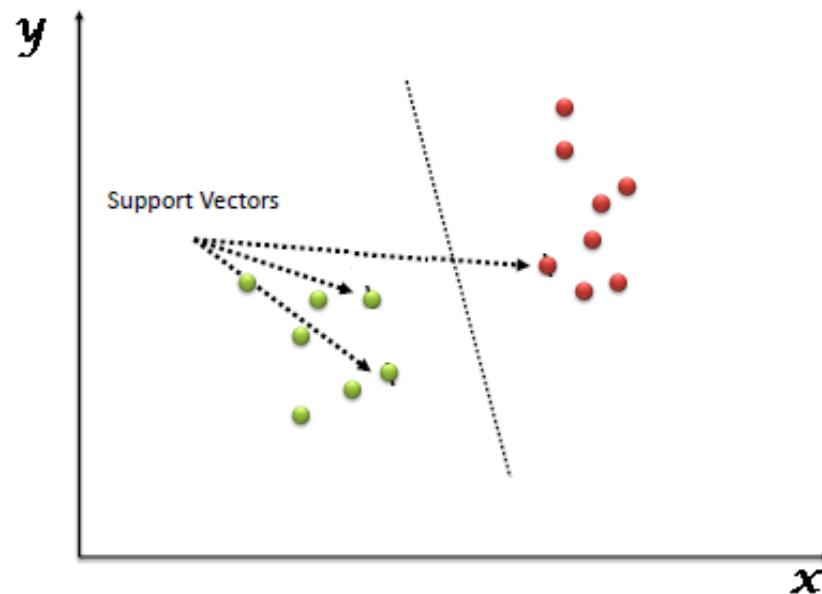


Support Vector Machines and Kernel Method

Lê Anh Cường

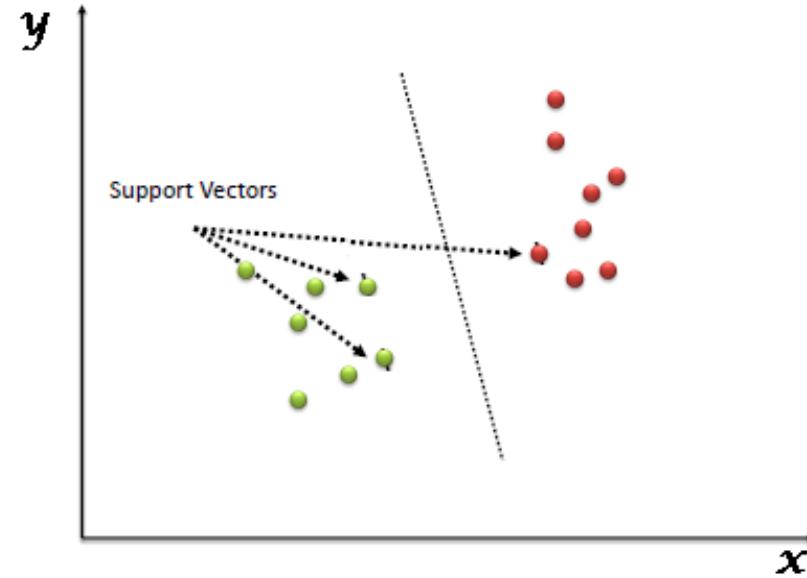
What is SVM?

- “Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges.
- However, it is mostly used in classification problems.



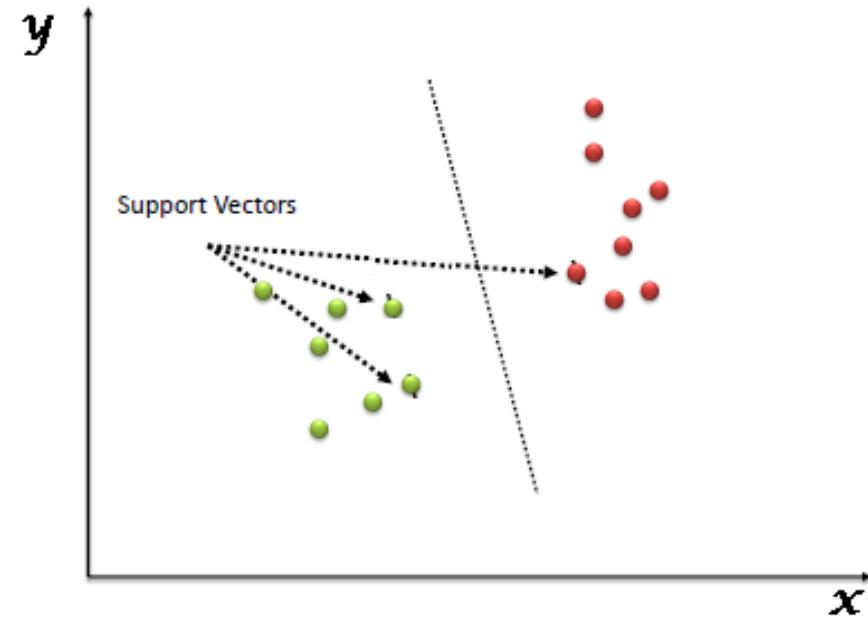
What is SVM?

- In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).



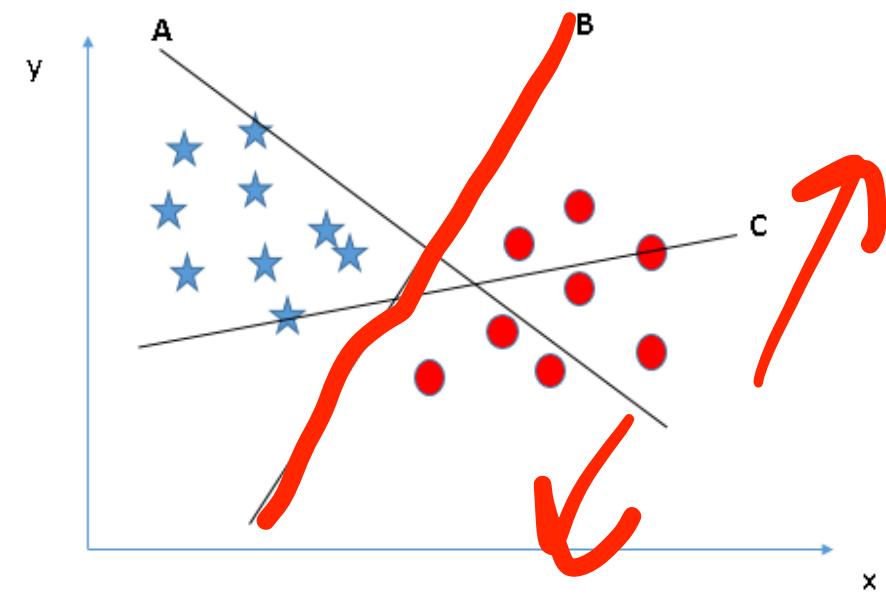
What is SVM?

- Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).



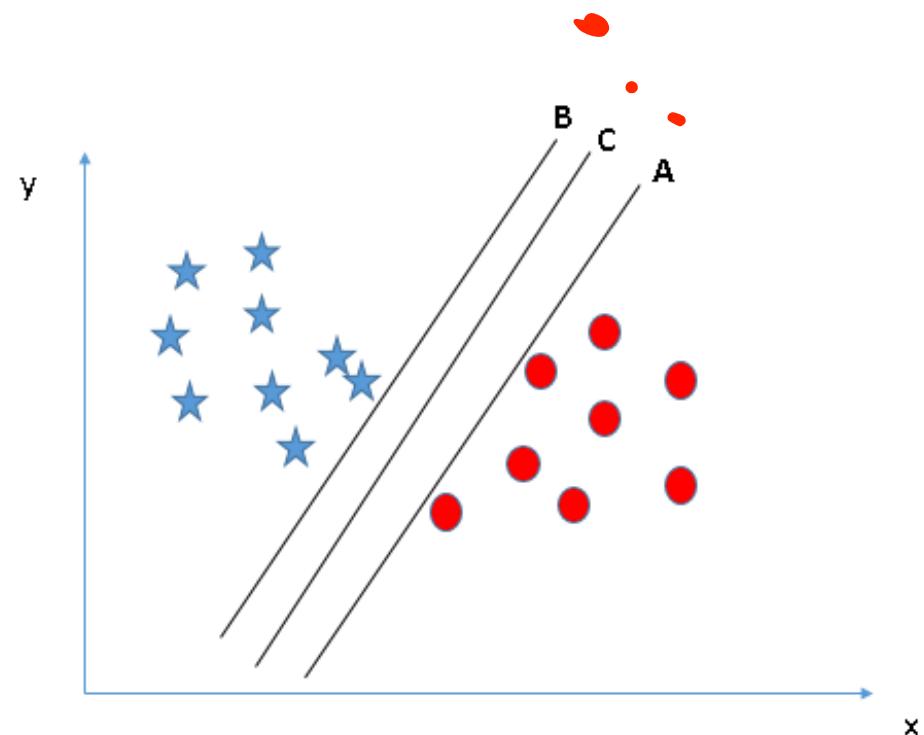
How does it work?

- we got accustomed to the process of segregating the two classes with a hyper-plane. Now the burning question is “How can we identify the right hyper-plane?”.



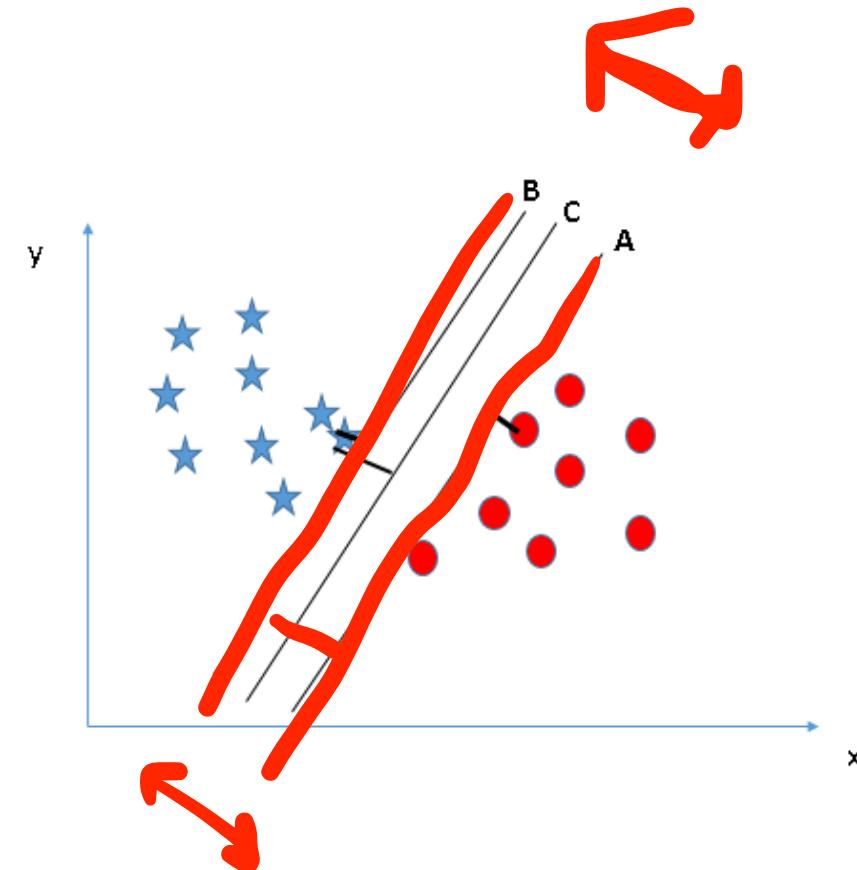
How does it work?

- Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?



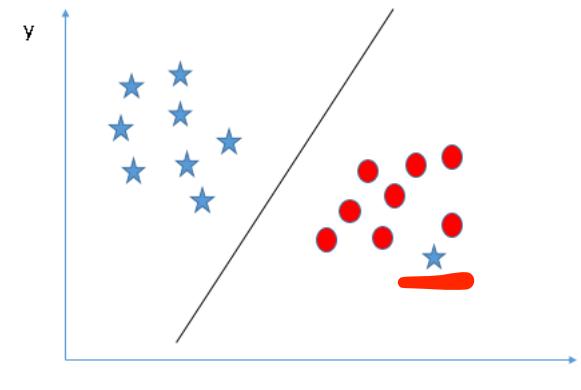
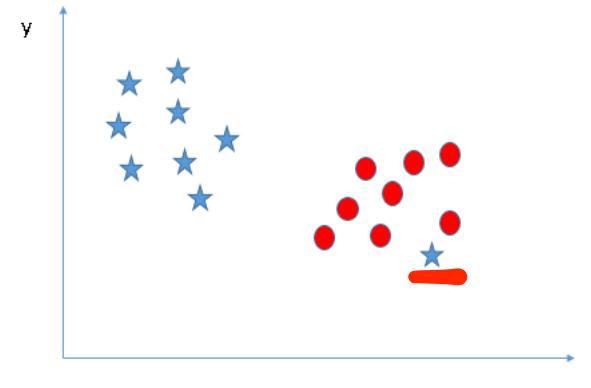
How does it work?

- maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**.



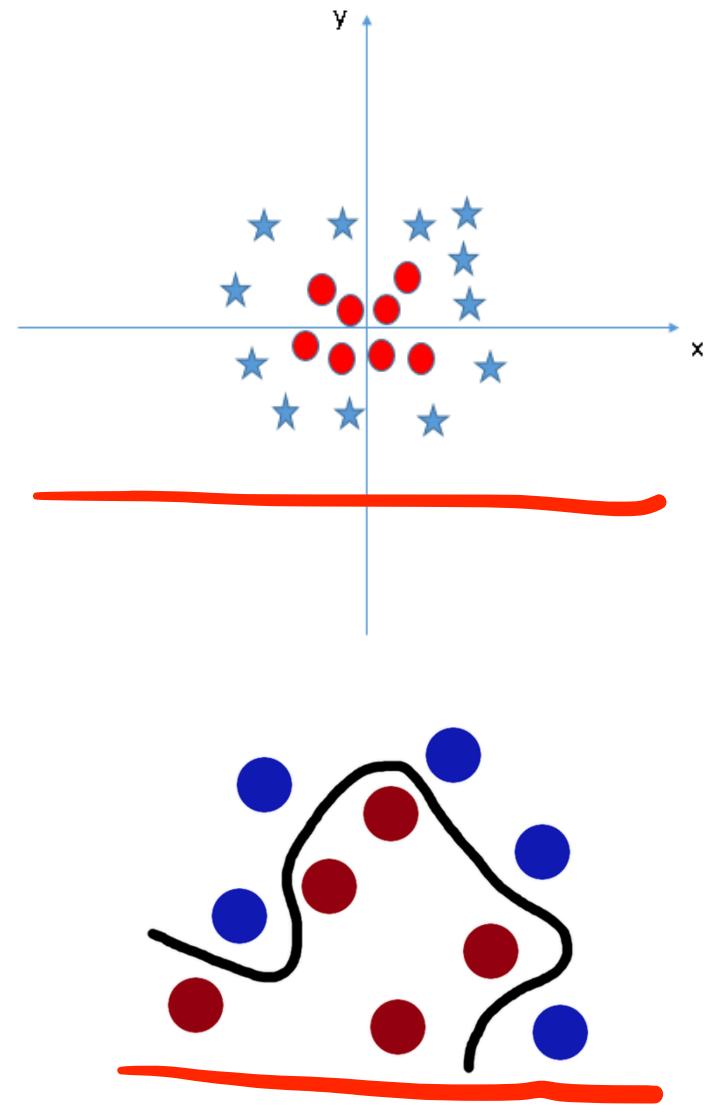
Issues: Outliers

- Below, I am unable to segregate the two classes using a straight line, as one of star lies in the territory of other(circle) class as an outlier.
- one star at other end is like an outlier for star class. SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin. Hence, we can say, SVM is robust to outliers.



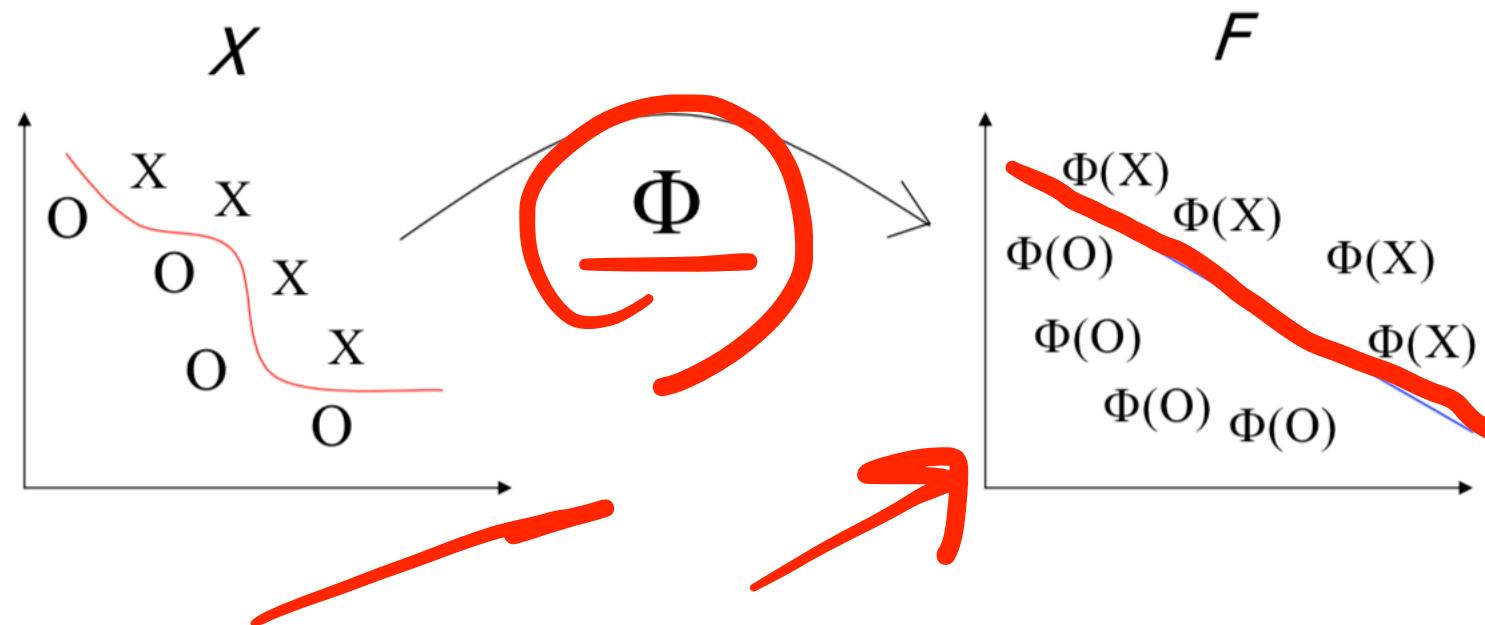
Limitation of linear hyper-plane

- In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.

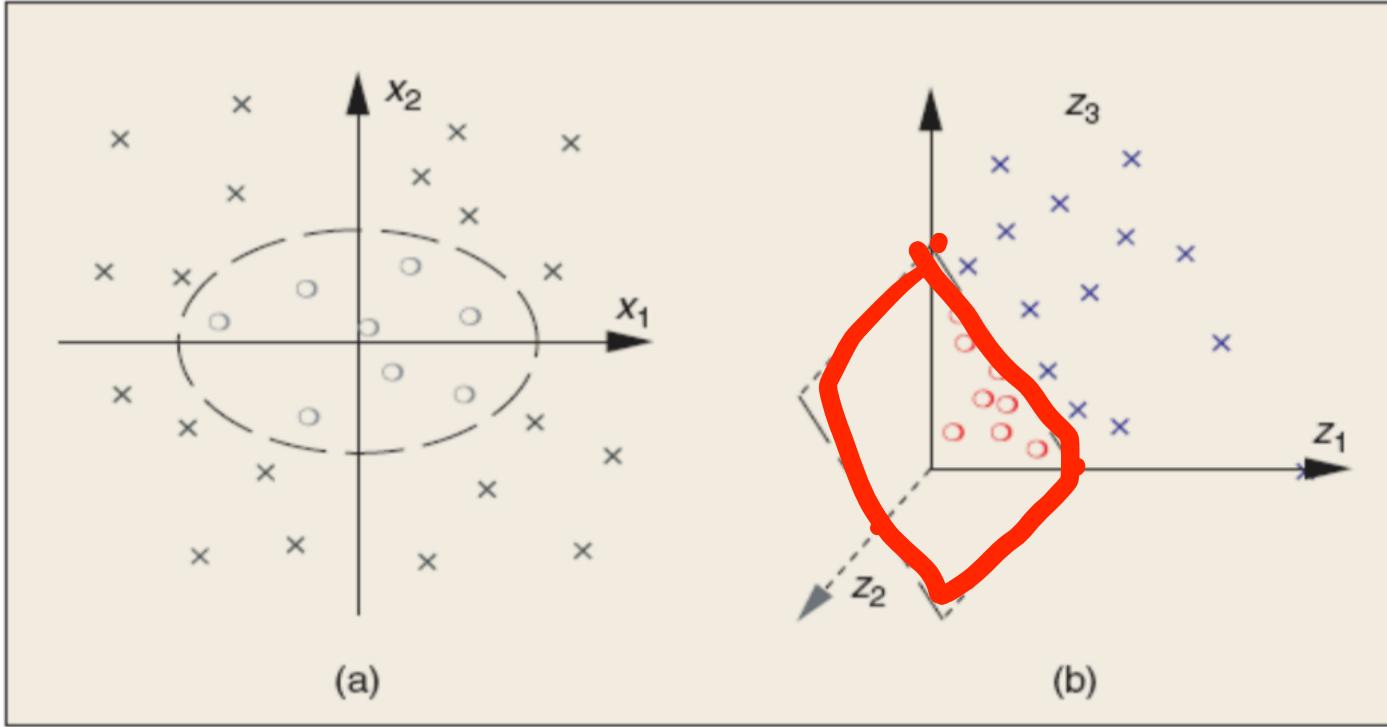


Data Transform from the original space to higher dimension space

- Solution: Map the data into a (possibly high-dimensional) vector space where linear relations exist among the data, then apply a linear algorithm in this space



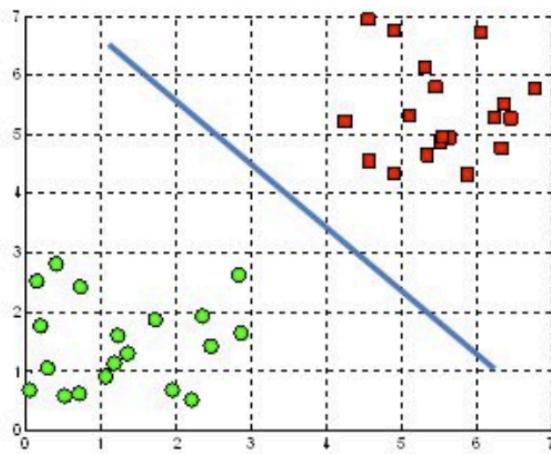
Data Transform from the original space to higher dimension space



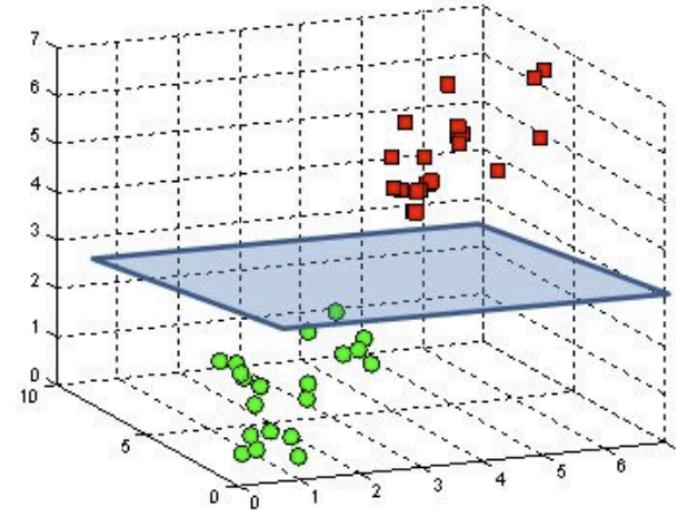
▲ 1. Effect of the map $\phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$ (a) Input space \mathcal{X} and (b) feature space \mathcal{H} .

Data Transform from the original space to higher dimension space

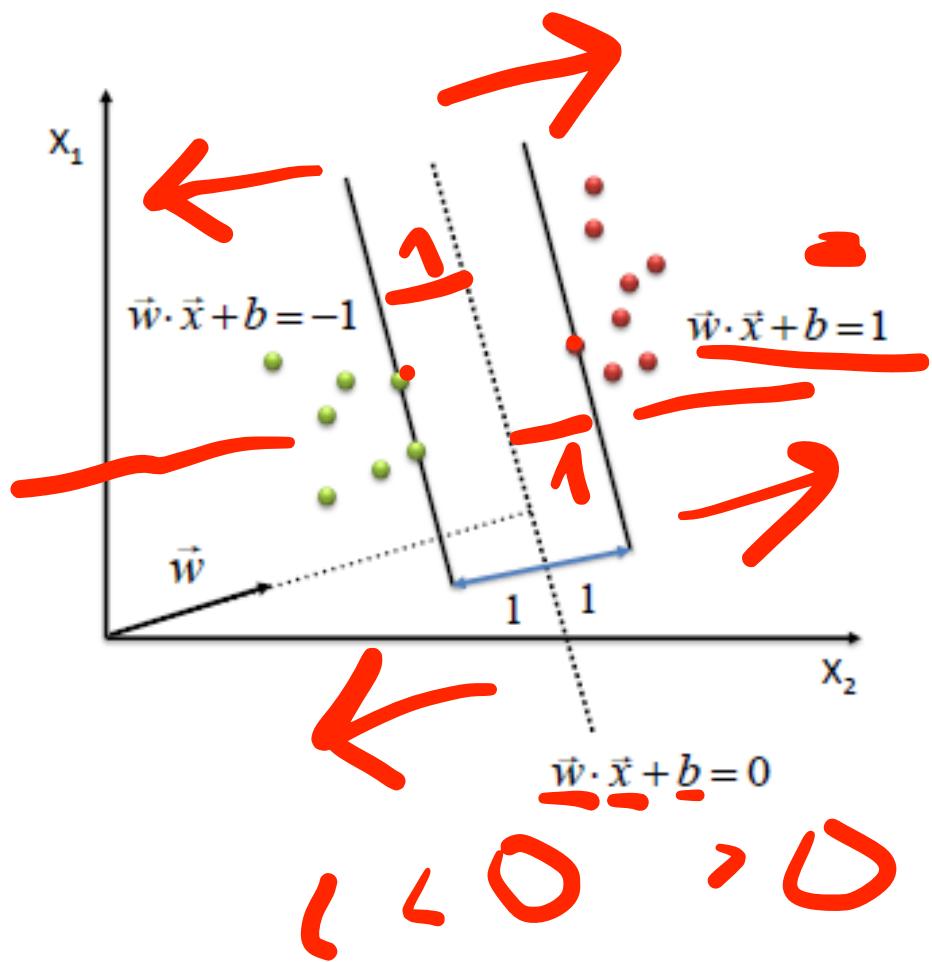
A hyperplane in \mathbb{R}^2 is a line



A hyperplane in \mathbb{R}^3 is a plane



Learning: Maximize Margin

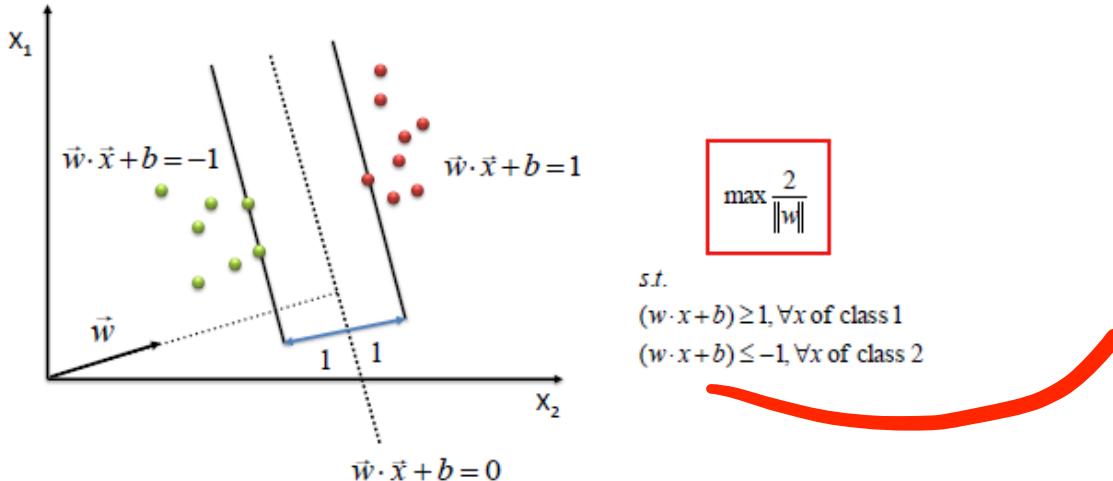


$$\max \frac{2}{\|\vec{w}\|}$$

s.t.

$$(\vec{w} \cdot \vec{x} + b) \geq 1, \forall x \text{ of class 1}$$

$$(\vec{w} \cdot \vec{x} + b) \leq -1, \forall x \text{ of class 2}$$



$$\boxed{\max \frac{2}{\|w\|}}$$

s.t.

$$(w \cdot x + b) \geq 1, \forall x \text{ of class 1}$$

$$(w \cdot x + b) \leq -1, \forall x \text{ of class 2}$$

Goal: we want to find the hyperplane (i.e. decision boundary) linearly separating our classes. Our boundary will have equation: $\mathbf{w}^T \mathbf{x} + b = 0$.

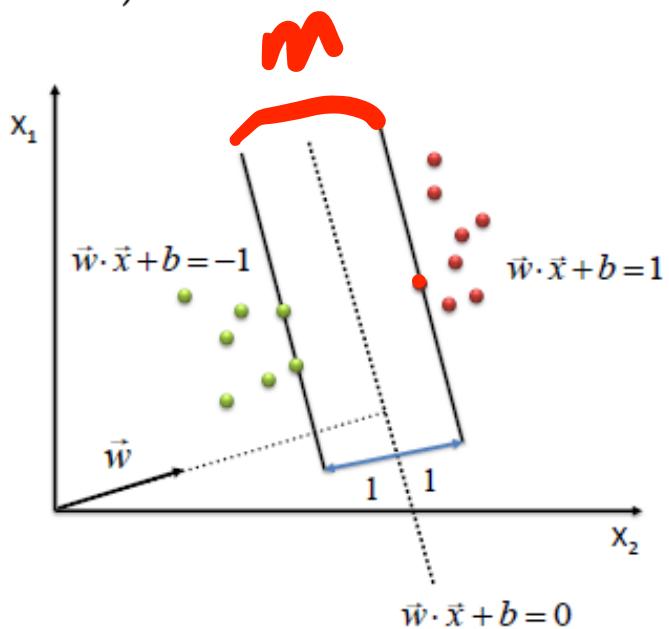
Anything above the decision boundary should have label 1.

i.e., \mathbf{x}_i s.t. $\mathbf{w}^T \mathbf{x}_i + b > 0$ will have corresponding $y_i = 1$.

Similarly, anything below the decision boundary should have label -1 .

i.e., \mathbf{x}_i s.t. $\mathbf{w}^T \mathbf{x}_i + b < 0$ will have corresponding $y_i = -1$.

You'll notice that we will now have some space between our decision boundary and the nearest data points of either class. Thus, let's rescale the data such that anything on or above the boundary $\mathbf{w}^T \mathbf{x} + b = 1$ is of one class (with label 1), and anything on or below the boundary $\mathbf{w}^T \mathbf{x} + b = -1$ is of the other class (with label -1).



$$\underline{\mathbf{w}^T \mathbf{x} + b = c}$$

$$\boxed{\max \frac{2}{\|\mathbf{w}\|}}$$

$$\begin{aligned} s.t. \\ (\mathbf{w}^T \mathbf{x} + b) &\geq 1, \forall \mathbf{x} \text{ of class 1} \\ (\mathbf{w}^T \mathbf{x} + b) &\leq -1, \forall \mathbf{x} \text{ of class 2} \end{aligned}$$

$$\underline{\frac{\mathbf{w}^T \mathbf{x} + b}{c} = 1}$$

First note that the two lines are parallel, and thus share their parameters \mathbf{w}, b .

Pick an arbitrary point $\underline{\mathbf{x}_1}$ to lie on line $\mathbf{w}^T \mathbf{x} + b = -1$. Then, the closest point on line $\mathbf{w}^T \mathbf{x} + b = 1$ to \mathbf{x}_1 is the point $\mathbf{x}_2 = \mathbf{x}_1 + \underline{\lambda \mathbf{w}}$ (since the closest point will always lie on the perpendicular; recall that the vector \mathbf{w} is perpendicular to both lines). Using this formulation, $\lambda \mathbf{w}$ will be the line segment connecting \mathbf{x}_1 and \mathbf{x}_2 , and thus, $\underline{\lambda \|\mathbf{w}\|}$, the distance between \mathbf{x}_1 and \mathbf{x}_2 , is the shortest distance between the two lines/boundaries. Solving for λ :

$$\Rightarrow \mathbf{w}^T \mathbf{x}_2 + b = 1 \text{ where } \mathbf{x}_2 = \mathbf{x}_1 + \lambda \mathbf{w}$$

$$\Rightarrow \mathbf{w}^T (\underline{\mathbf{x}_1} + \lambda \mathbf{w}) + b = 1$$

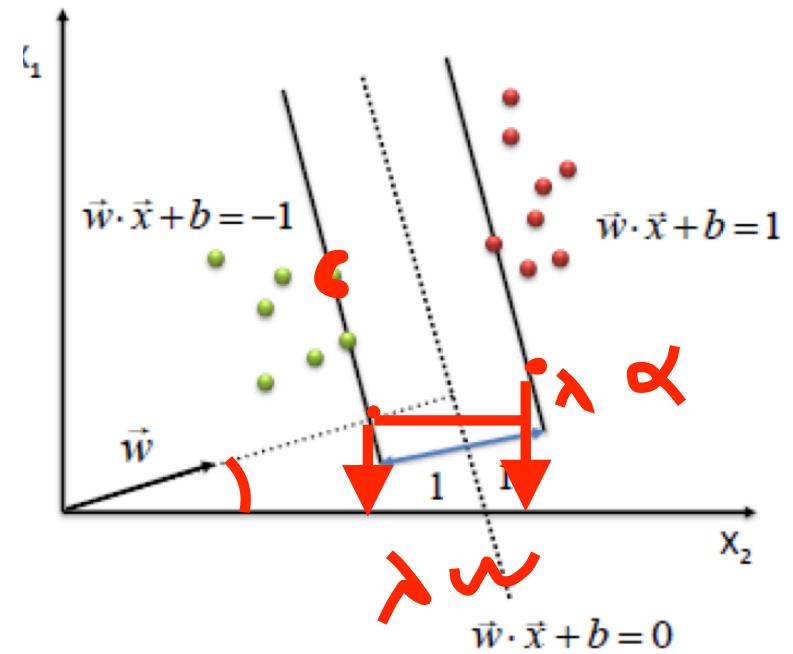
$$\Rightarrow \underline{\mathbf{w}^T \mathbf{x}_1 + b} + \lambda \mathbf{w}^T \mathbf{w} = 1 \text{ where } \mathbf{w}^T \mathbf{x}_1 + b = -1$$

$$\Rightarrow -1 + \lambda \mathbf{w}^T \mathbf{w} = 1$$

$$\Rightarrow \underline{\lambda \mathbf{w}^T \mathbf{w}} = 2$$

$$\Rightarrow \underline{\lambda} = \frac{2}{\mathbf{w}^T \mathbf{w}} = \frac{2}{\|\mathbf{w}\|^2}$$

$$\frac{2}{\|\mathbf{w}\|}$$



$$\frac{w}{t} = \frac{2}{m}$$

$$t = \frac{2}{m}$$

The reason for this labelling scheme is that it lets us condense the formulation for the decision function to $f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$ since $f(\mathbf{x}) = +1$ for all \mathbf{x} above the boundary, and $f(\mathbf{x}) = -1$ for all \mathbf{x} below the boundary.

Thus, we can figure out if an instance has been classified properly by checking that $y(\mathbf{w}^T \mathbf{x} + b) \geq 1$ (which will be the case as long as either both $y, \mathbf{w}^T \mathbf{x} + b > 0$ or else $y, \mathbf{w}^T \mathbf{x} + b < 0$).

This *quadratic programming* problem is expressed as:

$$\min_{\mathbf{w}, b} \frac{\mathbf{w}^T \mathbf{w}}{2}$$

subject to: $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$ (\forall data points \mathbf{x}_i).



This *quadratic programming* problem is expressed as:

$$\min_{\mathbf{w}, b} \frac{\mathbf{w}^T \mathbf{w}}{2}$$

subject to: $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$ (\forall data points \mathbf{x}_i).

2 Soft-margin extention

Consider the case that your data isn't perfectly linearly separable. For instance, maybe you aren't guaranteed that all your data points are correctly labelled, so you want to allow some data points of one class to appear on the other side of the boundary.

We can introduce *slack variables* - an $\epsilon_i \geq 0$ for each \mathbf{x}_i . Our quadratic programming problem becomes:

$$\min_{\mathbf{w}, b, \epsilon} \dots + C \sum_i \epsilon_i$$

subject to: $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \epsilon_i$ and $\epsilon_i \geq 0$ (\forall data points \mathbf{x}_i).

Kernel definition

- A function that takes as its inputs vectors in the original space and returns the dot product of the vectors in the feature space is called a *kernel function*
- More formally, if we have data $\mathbf{X}, \mathbf{Z} \in X$ and a map $\underline{\phi}: \underline{X} \rightarrow \underline{\mathcal{R}}^N$ then

$$\underline{k}(\mathbf{x}, \mathbf{z}) = \langle \underline{\phi}(\mathbf{x}), \underline{\phi}(\mathbf{z}) \rangle$$

is a kernel function

Kernel definition: an important point

- Using kernels, we do not need to embed the data into the space \Re^N explicitly, because a number of algorithms only require the inner products between image vectors!
 - We never need the coordinates of the data in the feature space!
-

Kernel Example

- Consider a two-dimensional input space $X \subseteq \mathbb{R}^2$ with the feature map:

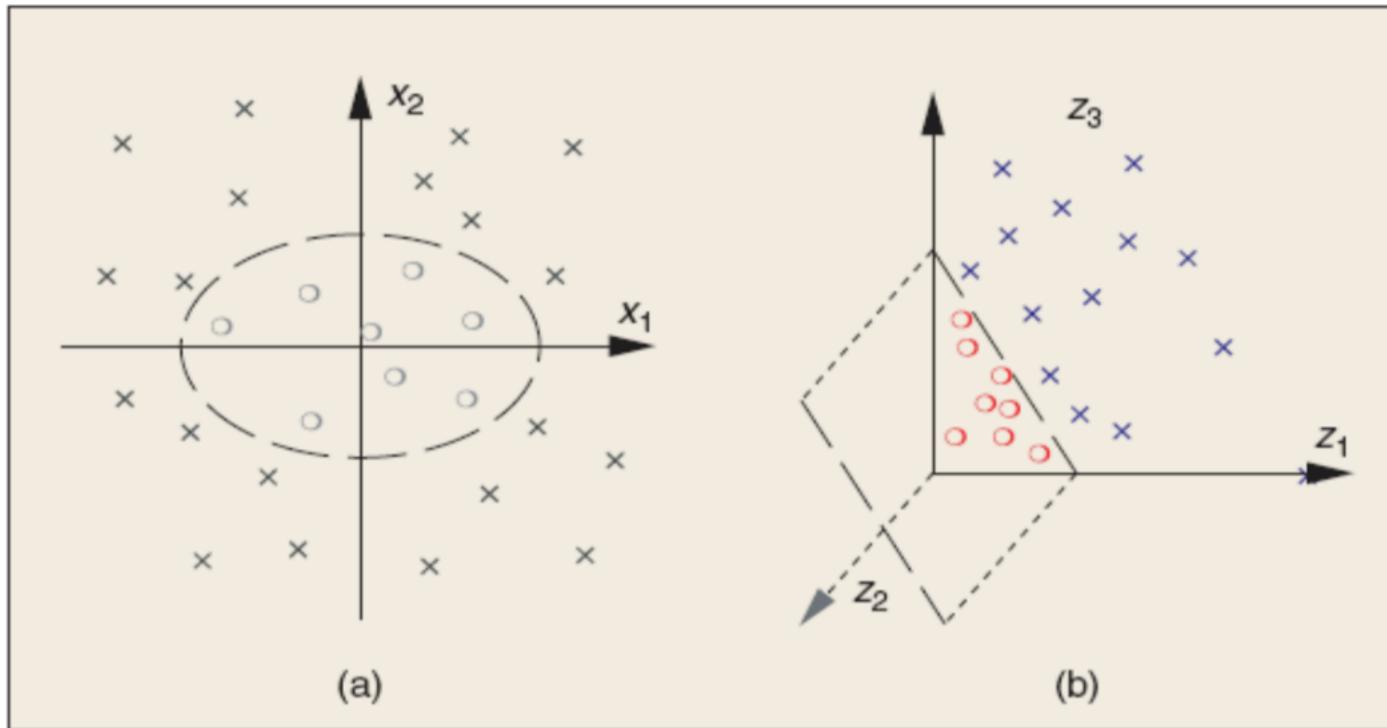
$$\phi: \underline{\mathbf{x}} = (x_1, x_2) \mapsto \phi(\mathbf{x}) = (\underline{x_1^2}, \underline{x_2^2}, \sqrt{2}x_1x_2) \in F = \underline{\mathbb{R}^3}$$

Now consider the inner product in the feature space:

$$\begin{aligned}\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle &= \left\langle (\underline{x_1^2}, \underline{x_2^2}, \sqrt{2}x_1x_2), (\underline{z_1^2}, \underline{z_2^2}, \sqrt{2}z_1z_2) \right\rangle \\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 x_2 z_1 z_2 = (\underline{x_1 z_1 + x_2 z_2})^2 \\ &= \langle \underline{\mathbf{x}}, \underline{\mathbf{z}} \rangle^2\end{aligned}$$

$k =$

Kernel example



▲ 1. Effect of the map $\phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$ (a) Input space \mathcal{X} and (b) feature space \mathcal{H} .

Kernel example

- Then $k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^2$ 
- But $k(\mathbf{x}, \mathbf{z})$ is also the kernel that computes the inner product of the map

$$\psi(\mathbf{x}) = (x_1^2, x_2^2, x_1 x_2, x_2 x_1) \in F = \mathbb{R}^4$$



Prove it?

- This shows that a given feature space is not unique to a given kernel function

SVM: optimization problem

■ Constrained optimization:

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0, i \in [1, m].$

■ Properties:

- $C \geq 0$ trade-off parameter.
- Convex optimization (strictly convex).
- Unique solution.

Dual optimization problem

■ Constrained optimization:

$$\max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (\underline{\mathbf{x}_i \cdot \mathbf{x}_j}) \quad \rightarrow$$

subject to: $\alpha_i \geq 0 \wedge \sum_{i=1}^m \alpha_i y_i = 0, i \in [1, m]$.

■ Solution:

$$h(x) = \text{sgn}\left(\sum_{i=1}^m \underline{\alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x})} + b\right),$$

with $b = y_i - \sum_{j=1}^m \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}_i)$ for any SV \mathbf{x}_i .

SVM with Kernel

■ Constrained optimization:

$$\max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j \underline{K(x_i, x_j)}$$

subject to: $0 \leq \alpha_i \leq C \wedge \sum_{i=1}^m \alpha_i y_i = 0, i \in [1, m]$.

■ Solution:

$$h(x) = \text{sgn}\left(\sum_{i=1}^m \alpha_i y_i \underline{K(x_i, x)} + b\right),$$

with $b = y_i - \sum_{j=1}^m \alpha_j y_j \underline{K(x_j, x_i)}$ for any x_i with $0 < \alpha_i < C$.

Optimization

■ Constrained optimization:

$$\begin{aligned} & \max_{\alpha} 2\alpha^\top \mathbf{1} - \alpha^\top \mathbf{Y}^\top \mathbf{K} \mathbf{Y} \alpha \\ \text{subject to: } & \mathbf{0} \leq \alpha \leq \mathbf{C} \wedge \alpha^\top \mathbf{y} = 0. \end{aligned}$$

■ Solution:

$$h = \operatorname{sgn}\left(\sum_{i=1}^m \alpha_i y_i K(x_i, \cdot) + b\right),$$

with $b = y_i - (\alpha \circ \mathbf{y})^\top \mathbf{K} \mathbf{e}_i$ for any x_i with

 $0 < \alpha_i < C$.

Regression problem

- **Training data:** sample drawn i.i.d. from set X according to some distribution D ,

$$S = ((x_1, y_1), \dots, (x_m, y_m)) \in X \times Y,$$

with $Y \subseteq \mathbb{R}$ is a measurable subset.

- **Loss function:** $L: Y \times Y \rightarrow \mathbb{R}_+$ a measure of closeness, typically $L(y, y') = (y' - y)^2$ or $L(y, y') = |y' - y|^p$ for some $p \geq 1$.
- **Problem:** find hypothesis $h: X \rightarrow \mathbb{R}$ in H with small generalization error with respect to target f

$$R_D(h) = \mathbb{E}_{x \sim D} [L(h(x), f(x))].$$

Kernel Ridge Regression

(Saunders et al., 1998)

■ Optimization problem:

$$\max_{\alpha \in \mathbb{R}^m} -\lambda \alpha^\top \alpha + 2\alpha^\top \mathbf{y} - \alpha^\top \mathbf{K} \alpha$$

or $\max_{\alpha \in \mathbb{R}^m} -\alpha^\top (\mathbf{K} + \lambda \mathbf{I}) \alpha + 2\alpha^\top \mathbf{y}.$

■ Solution:

$$h(x) = \sum_{i=1}^m \alpha_i K(x_i, x),$$

with $\alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}.$

Different kernel

Linear kernels $k(\mathbf{x}, \mathbf{x}') = \underline{\mathbf{x}^\top \mathbf{x}'}$

Polynomial kernels $k(\mathbf{x}, \mathbf{x}') = \underline{(1 + \mathbf{x}^\top \mathbf{x}')^d}$ for any $d > 0$

- Contains all polynomials terms up to degree d

Gaussian kernels $k(\mathbf{x}, \mathbf{x}') = \exp \left(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2 \right)$ for $\sigma > 0$

- Infinite dimensional feature space

Radial Basis Function (RBF) SVM

$$f(\mathbf{x}) = \sum_i^N \alpha_i y_i \exp \left(-\|\mathbf{x} - \mathbf{x}_i\|^2 / 2\sigma^2 \right) + b$$

Kernel Method

■ Idea:

- Define $K: X \times X \rightarrow \mathbb{R}$, called **kernel**, such that:

$$\Phi(x) \cdot \Phi(y) = K(x, y).$$

- K often interpreted as a similarity measure.

■ Benefits:

- **Efficiency:** K is often more efficient to compute than Φ and the dot product.
- **Flexibility:** K can be chosen arbitrarily so long as the existence of Φ is guaranteed (Mercer's condition).

Example - Polynomial Kernels

■ Definition:

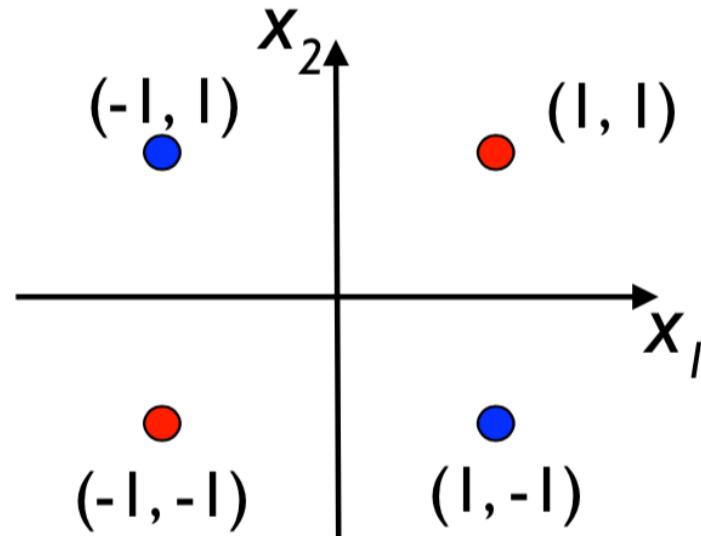
$$\forall x, y \in \mathbb{R}^N, K(x, y) = (x \cdot y + c)^d, \quad c > 0.$$

■ Example: for $N=2$ and $d=2$,

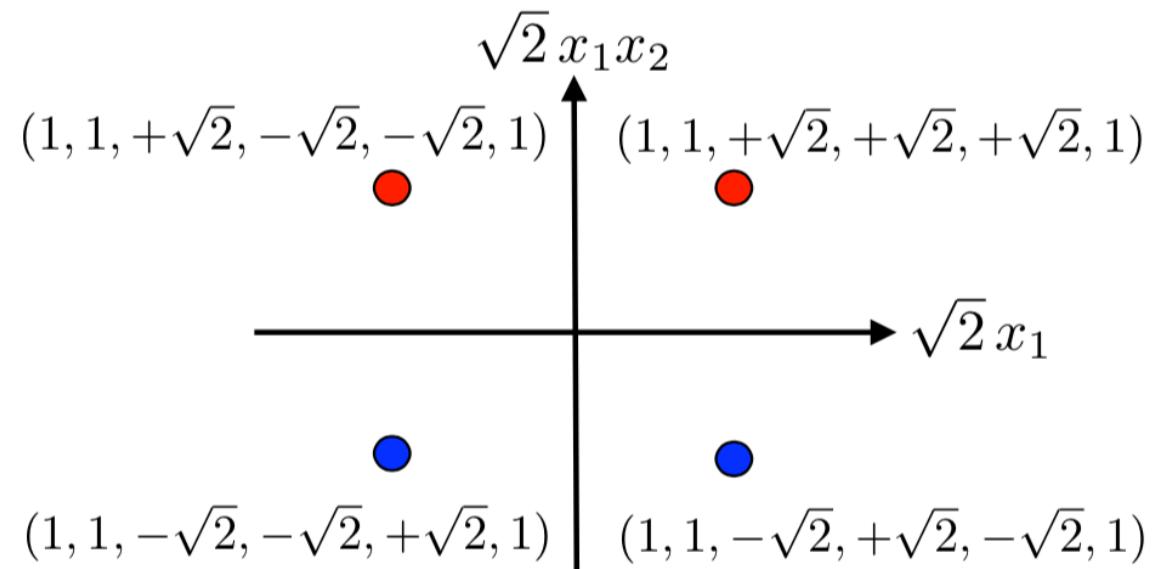
$$K(x, y) = (x_1 y_1 + x_2 y_2 + c)^2$$
$$= \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \\ \sqrt{2c} x_1 \\ \sqrt{2c} x_2 \\ c \end{bmatrix} \cdot \begin{bmatrix} y_1^2 \\ y_2^2 \\ \sqrt{2} y_1 y_2 \\ \sqrt{2c} y_1 \\ \sqrt{2c} y_2 \\ c \end{bmatrix}.$$

XOR Problem

- Use second-degree polynomial kernel with $c = 1$:



Linearly non-separable



Linearly separable by
 $x_1 x_2 = 0$.

Other Standard PDS Kernels

positive definite symmetric (PDS)

■ Gaussian kernels:

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right), \quad \sigma \neq 0.$$

■ Sigmoid Kernels:

$$K(x, y) = \tanh(a(x \cdot y) + b), \quad a, b \geq 0.$$

Thank you