

Support Vector Machines

Alymzhan Toleu
alymzhan.toleu@gmail.com

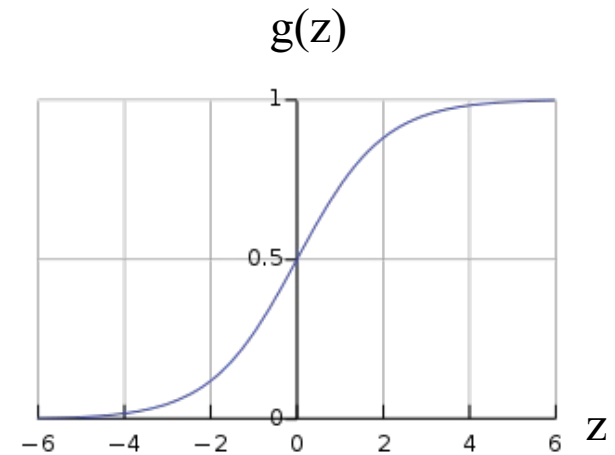
Decision Boundary

Logistic Regression

Hypothesis: $f_{\theta}(x) = g(\theta^T x)$

where, $\theta^T x = w_0 x_0 + w_1 x_1 + \dots + w_n x_n$

$$g(z) = \frac{1}{1+e^{-z}}$$



$$g(z) \geq 0.5, \quad z \geq 0$$

$$g(z) < 0.5, \quad z < 0$$



$$g(\theta^T x) \geq 0.5, \quad z = \theta^T x \geq 0$$

$$g(\theta^T x) < 0.5, \quad z = \theta^T x < 0$$

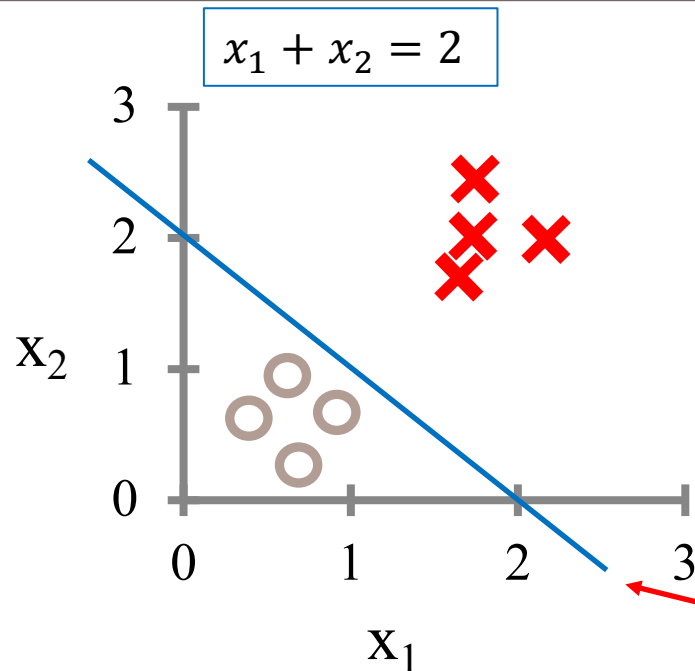
Set a threshold = 0.5

If $f_{\theta}(x) \geq 0.5$, predict $y = 1$

If $f_{\theta}(x) < 0.5$, predict $y = 0$

How hypothesis of logistic
regression makes predictions?

Decision Boundary



$$f_{\theta}(x) = g(w_0 + w_1x_1 + w_2x_2)$$

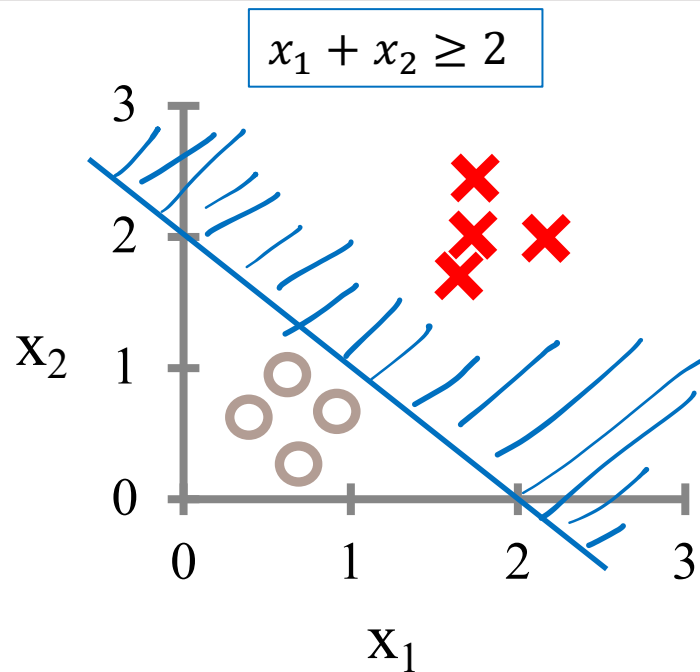
Suppose we found values for:

$$w_0 = -2, w_1 = 1, w_2 = 1$$

$$\text{Predict } y = 1, g(\theta^T x) \geq 0.5, \theta^T x \geq 0 \implies -2 + x_1 + x_2 \geq 0$$

$$\text{Predict } y = 0, g(\theta^T x) < 0.5, \theta^T x < 0 \implies -2 + x_1 + x_2 < 0$$

Decision Boundary



$$f_{\theta}(x) = g(w_0 + w_1x_1 + w_2x_2)$$

Suppose we found values for:

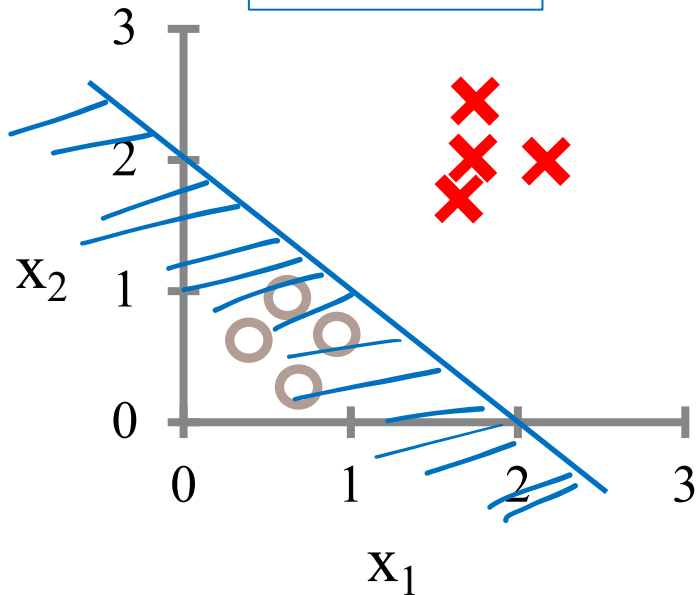
$$w_0 = -2, w_1 = 1, w_2 = 1$$

$$\text{Predict } y = 1, g(\theta^T x) \geq 0.5, \theta^T x \geq 0 \implies -2 + x_1 + x_2 \geq 0$$

$$\text{Predict } y = 0, g(\theta^T x) < 0.5, \theta^T x < 0 \implies -2 + x_1 + x_2 < 0$$

Decision Boundary

$$x_1 + x_2 < 2$$



$$f_{\theta}(x) = g(w_0 + w_1x_1 + w_2x_2)$$

Suppose we found values for:

$$w_0 = -2, w_1 = 1, w_2 = 1$$

$$\text{Predict } y = 1, g(\theta^T x) \geq 0.5, \theta^T x \geq 0 \implies -2 + x_1 + x_2 \geq 0$$

$$\text{Predict } y = 0, g(\theta^T x) < 0.5, \theta^T x < 0 \implies -2 + x_1 + x_2 < 0$$

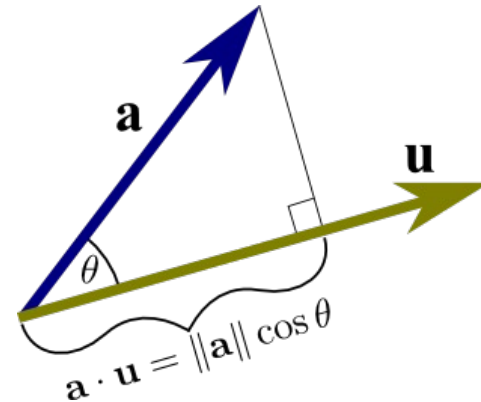
Support Vector Classifiers

Dot Product

The dot product between two vectors is based on the projection of one vector onto another

Calculate how much of \mathbf{a} is pointing in the same direction as the vector \mathbf{b} .

Unit vector: $\mathbf{u} = \frac{\mathbf{b}}{\|\mathbf{b}\|}$.

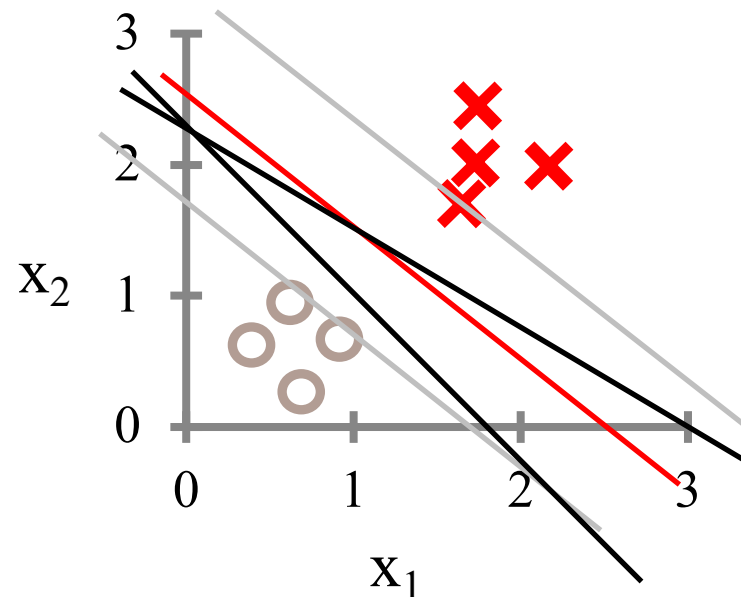


$$\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{b}\|} = \|\mathbf{a}\| \cos \theta.$$

The dot product of \mathbf{a} with unit vector \mathbf{u} , denoted $\mathbf{a} \cdot \mathbf{u}$, it is defined to be the projection of \mathbf{a} in the direction of \mathbf{u} , or the amount that \mathbf{a} is pointing in the same direction as unit vector \mathbf{u} .

Hyperplanes

There may be many hyperplanes separating the training samples, which one is better?



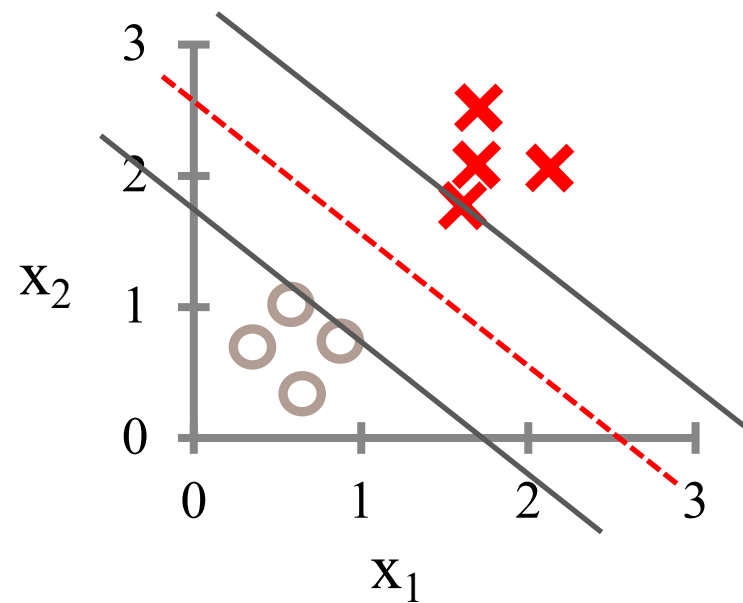
We should choose **the one in the middle**, which has **good tolerance**, **high robustness**, and the **good generalization ability**.

Hyperplanes

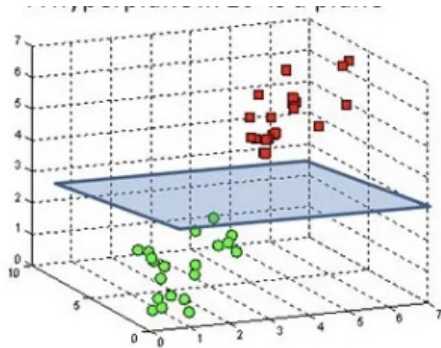
1-Dimension



2-Dimensions

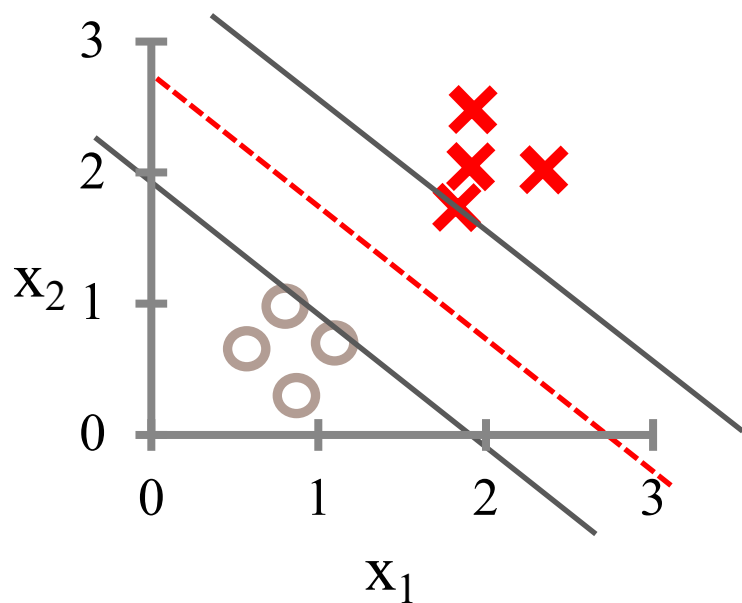


3-Dimensions



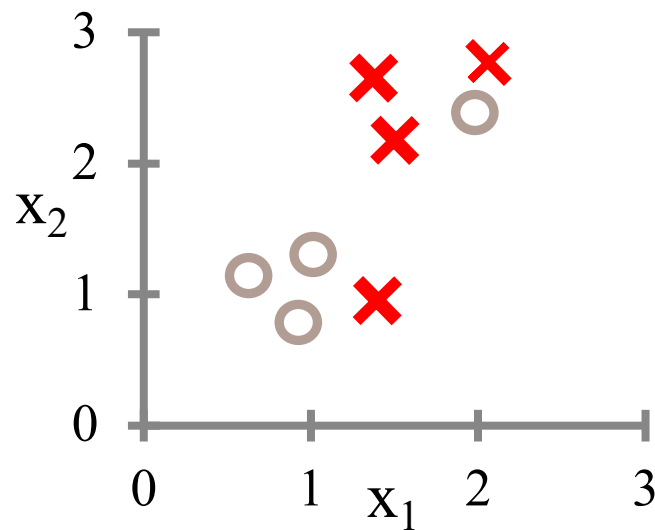
Two Cases of SVM

Case1: Linearly Separable



Hard margin SVM

Case2: None-Linearly Separable



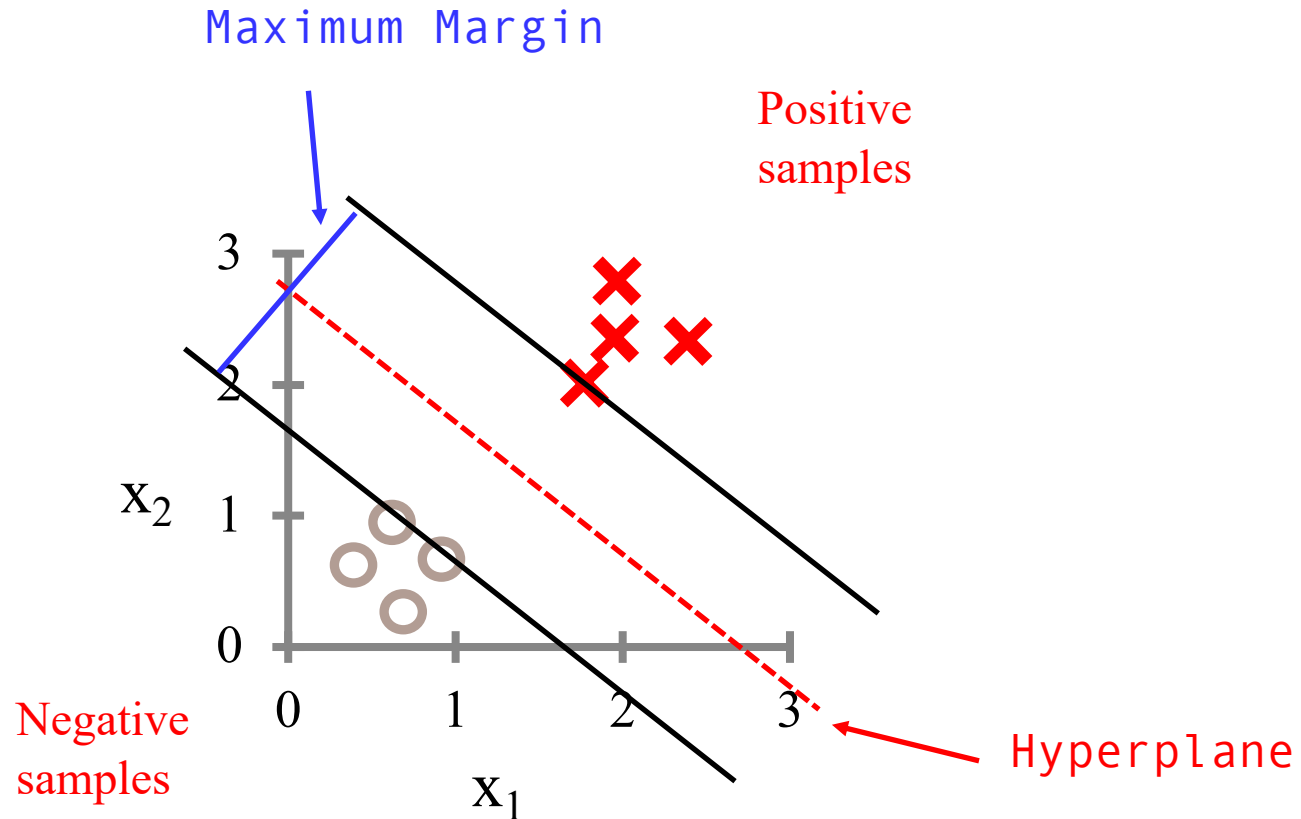
Soft margin SVM

Margin and Support Vectors

Maximum
Margin
Classifier

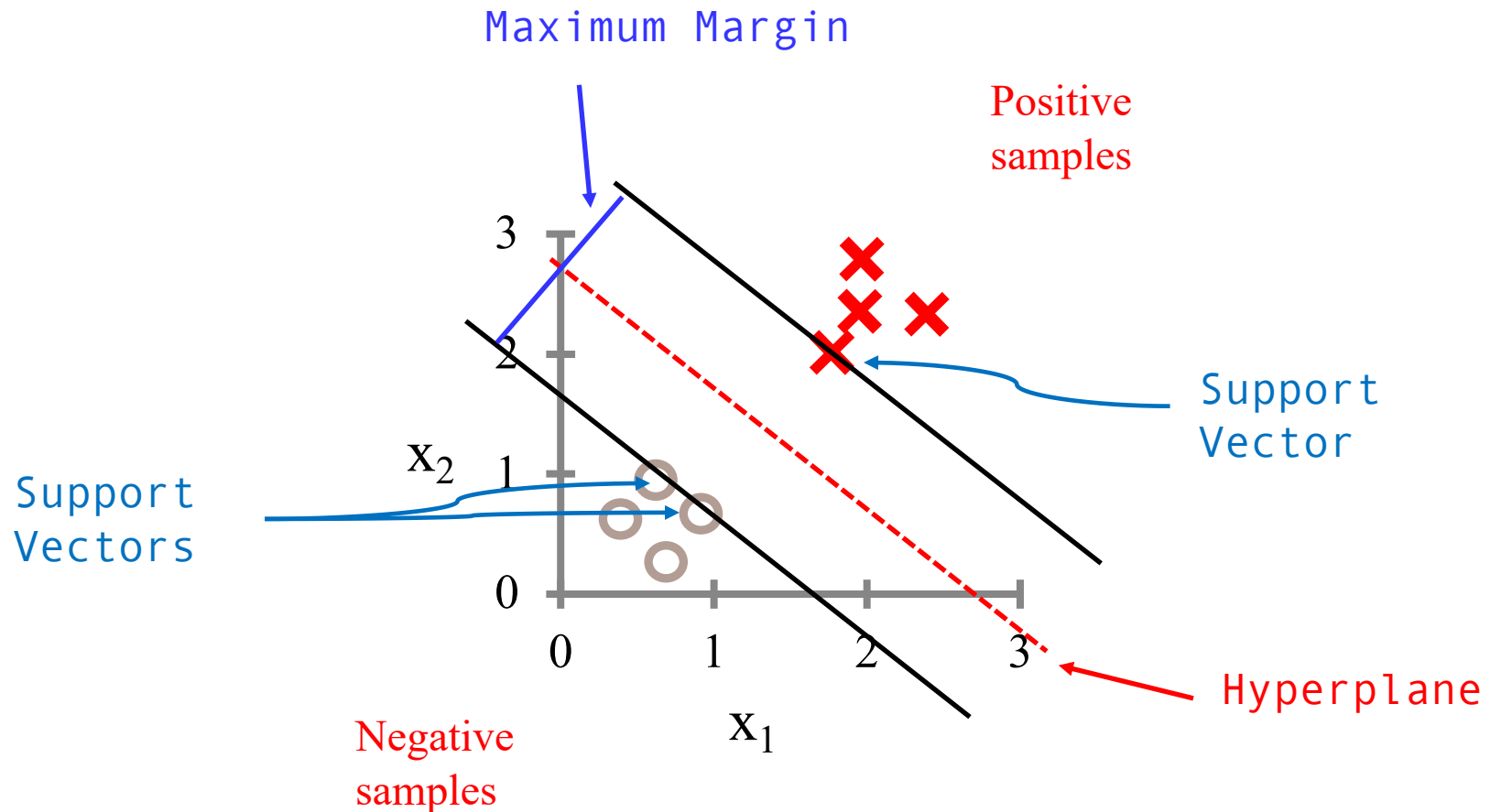
Or

Widest
Street
Approach



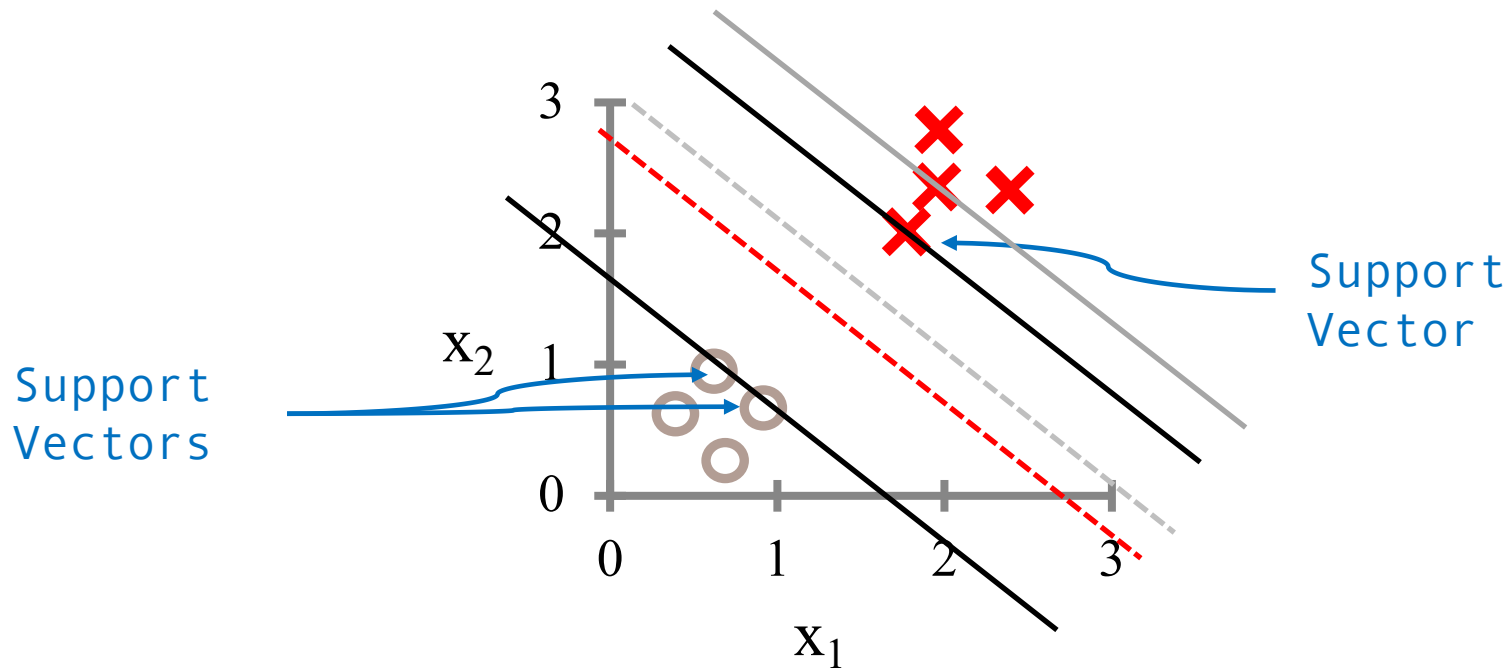
The straight line is different from the logistic regression.
It tries to position the line in a way that maximizes the separation between the positive and negative samples.
Make the street as wide as possible.

Margin and Support Vectors



Support Vectors are the data points that closest to the hyperplane.
They are the points the most difficult to classify.
They also decide which hyperplane we should choose.

Margin and Support Vectors



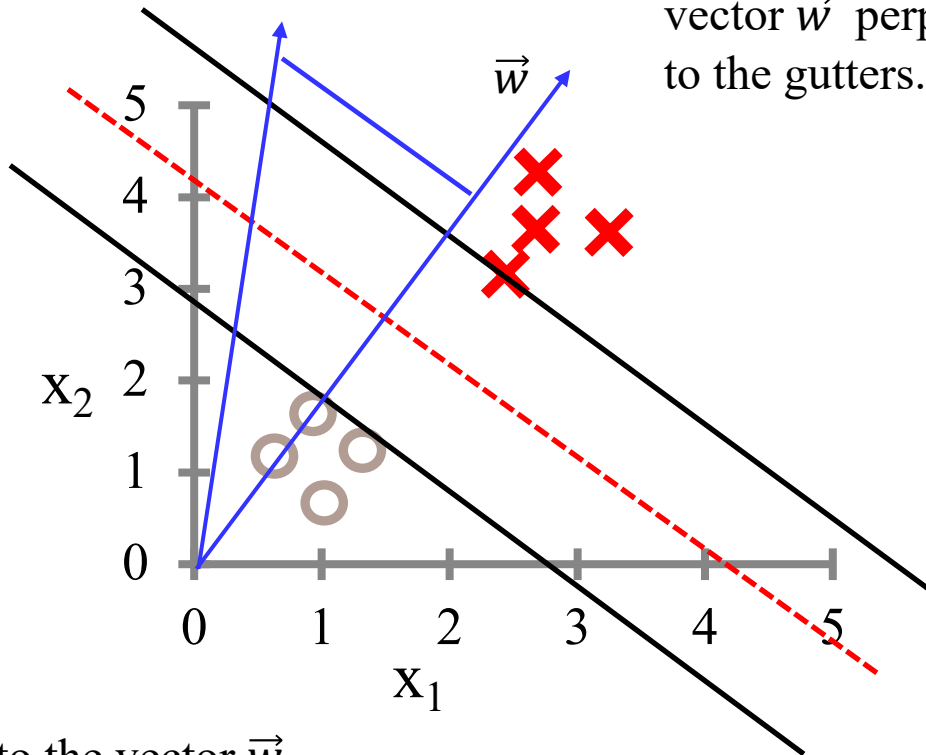
Unlike other machine learning approaches like logistic regression or others, which **all data points influence** the final optimization.

Form SVM, **only the support vectors** have an **impact** on the final decision boundary. Changing the support vectors that move the decision boundary.

Decision Rule

A unknown vector \vec{u}

Assume that there is a vector \vec{w} perpendicular to the gutters.



Project the vector \vec{u} to the vector \vec{w} .



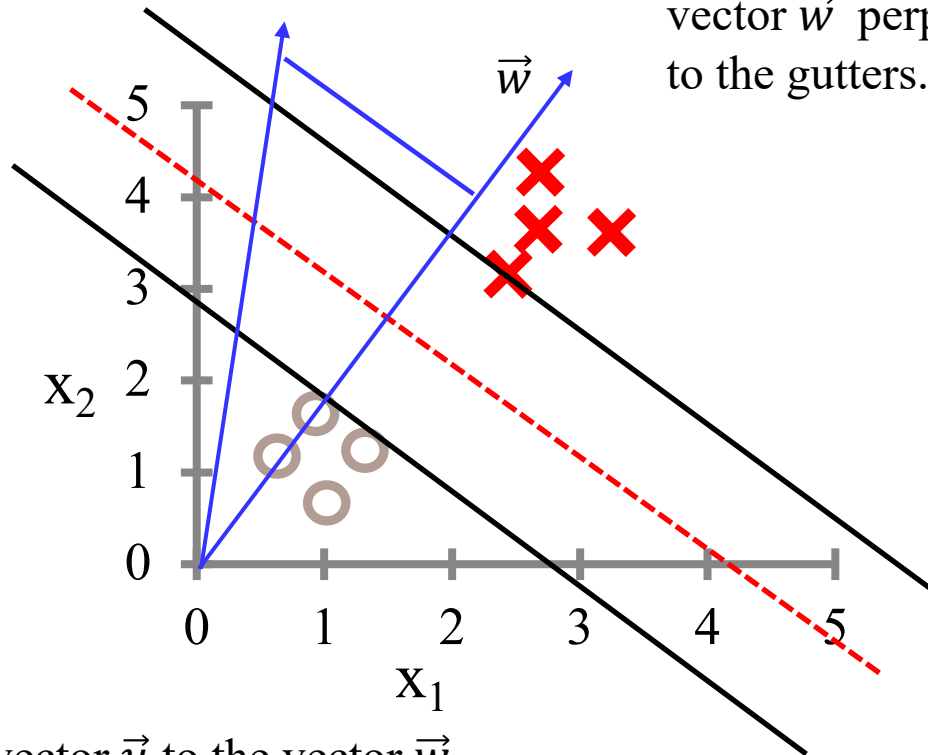
Dot product: $\vec{w} \cdot \vec{u} \geq c$

If projection cross the median line (c is large), positive (+);
If projection not cross the median line (c is small), negative (-);

Decision Rule

A unknown vector \vec{u}

Assume that there is a vector \vec{w} perpendicular to the gutters.



What we have:

- Know that w should be perpendicular to the median line.
- There are many w , they can be any length.
- Do not know the value for w and b .

Project the vector \vec{u} to the vector \vec{w} .



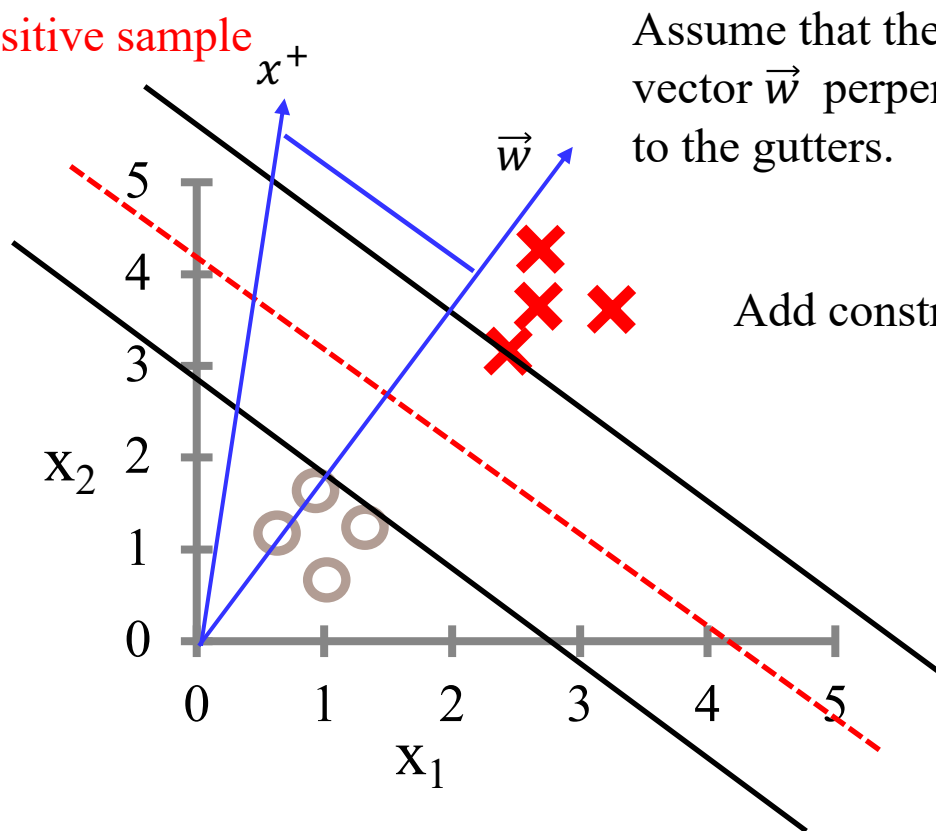
Dot product: $\vec{w} \cdot \vec{u} \geq c$ \Rightarrow

If $\vec{w} \cdot \vec{u} + b \geq 0$ then positive (+) samples
This is the decision rule.

What should we do next?

Constraints

A positive sample

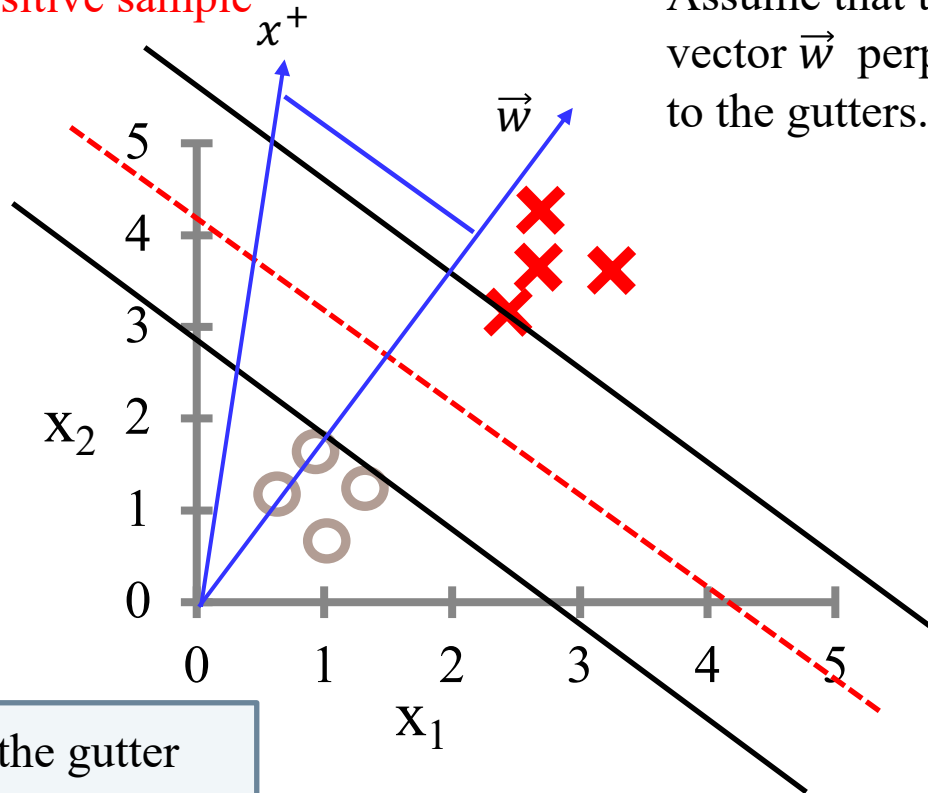


A positive sample

A negative sample

Constraints

A positive sample



Assume that there is a vector \vec{w} perpendicular to the gutters.

Add constraints:

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

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



introduce y_i , make it convenient for math :

$y_i = +1$ for (+) samples

$y_i = -1$ for (-) samples

Samples in the gutter
 $y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0$

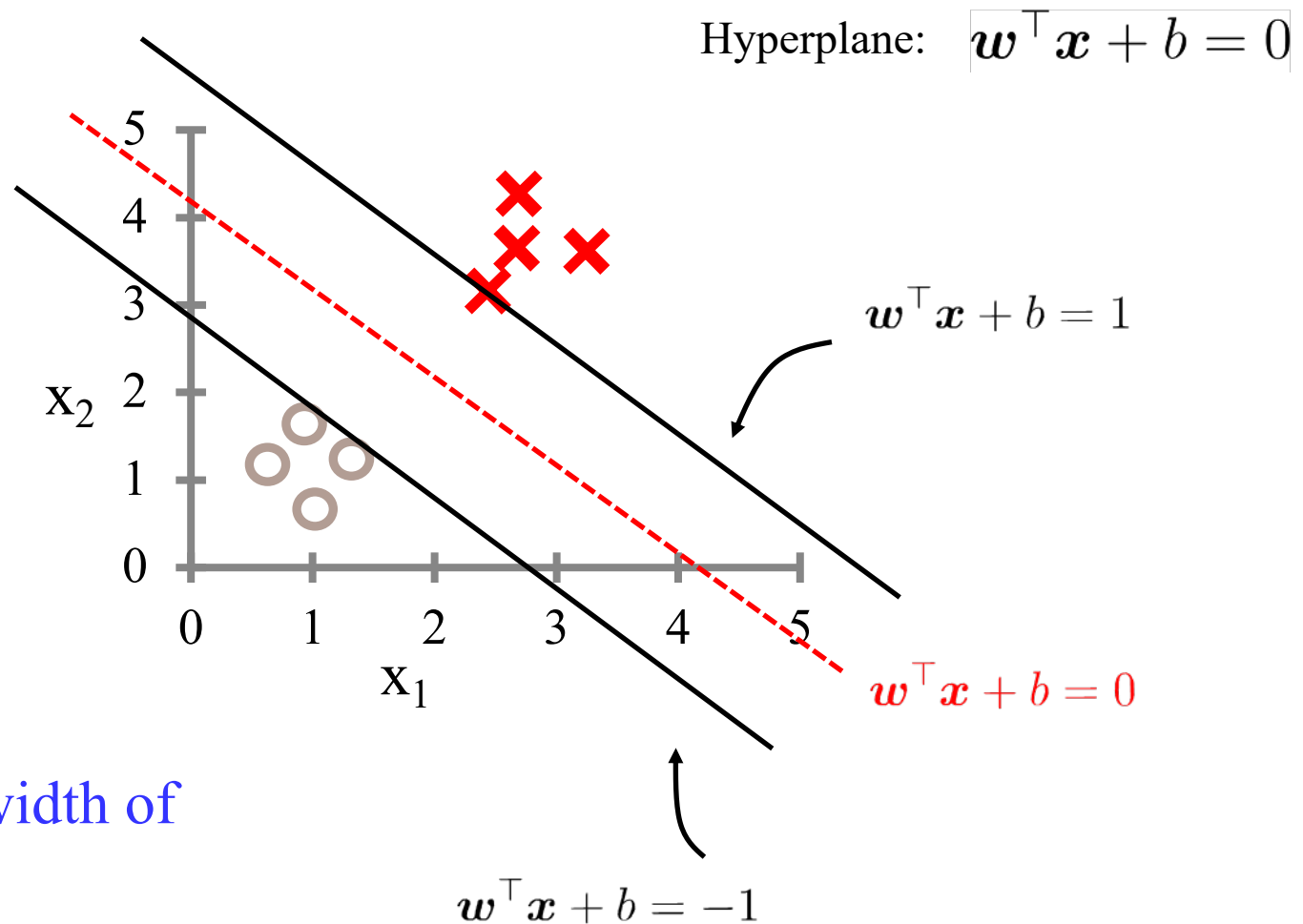
Multiply y_i to both equations:

$$y_i(\vec{w} \cdot \vec{x}^+ + b) \geq 1$$

$$y_i(\vec{w} \cdot \vec{x}^- + b) \geq 1$$

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0$$

Find Max Margin



Find Max Margin

Margin(width of street):

$$(x^+ \cdot \frac{\vec{w}}{\|\vec{w}\|} - x^- \cdot \frac{\vec{w}}{\|\vec{w}\|})$$

Unit vector

$$(x^+ - x^-) \cdot \frac{\vec{w}}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|}$$

Samples in the gutter

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0$$

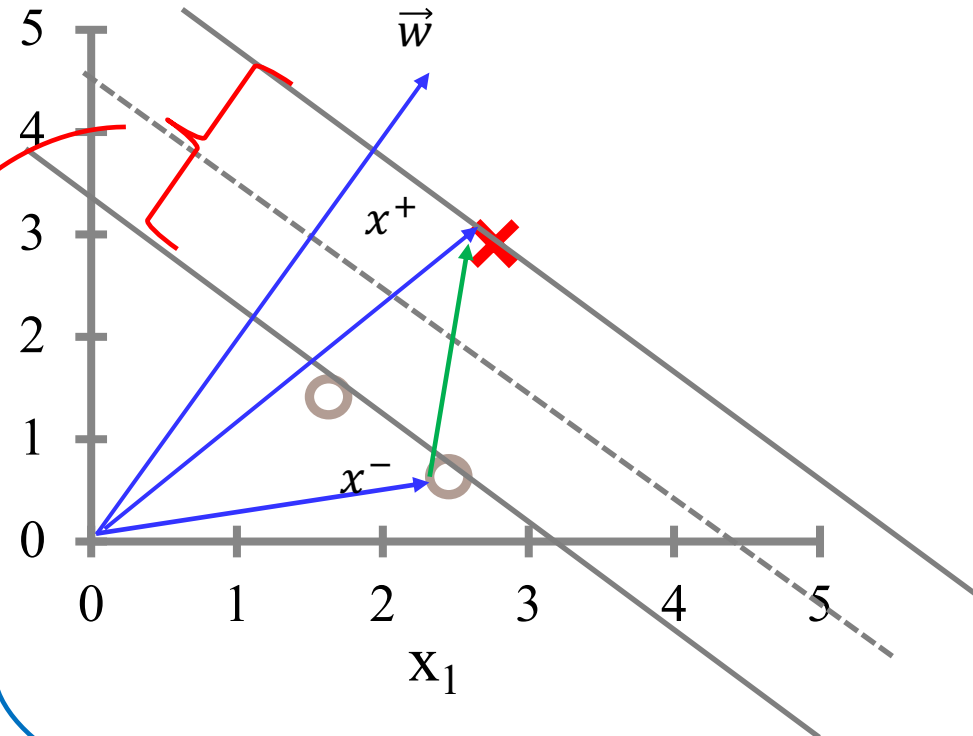
For (+) samples, $y_i=1$

For (-) samples, $y_i=-1$



$$\vec{w}x^+ = 1 - b$$

$$\vec{w}x^- = -1 - b$$



If we want to get a **widest** street, we need to **maximize** it.

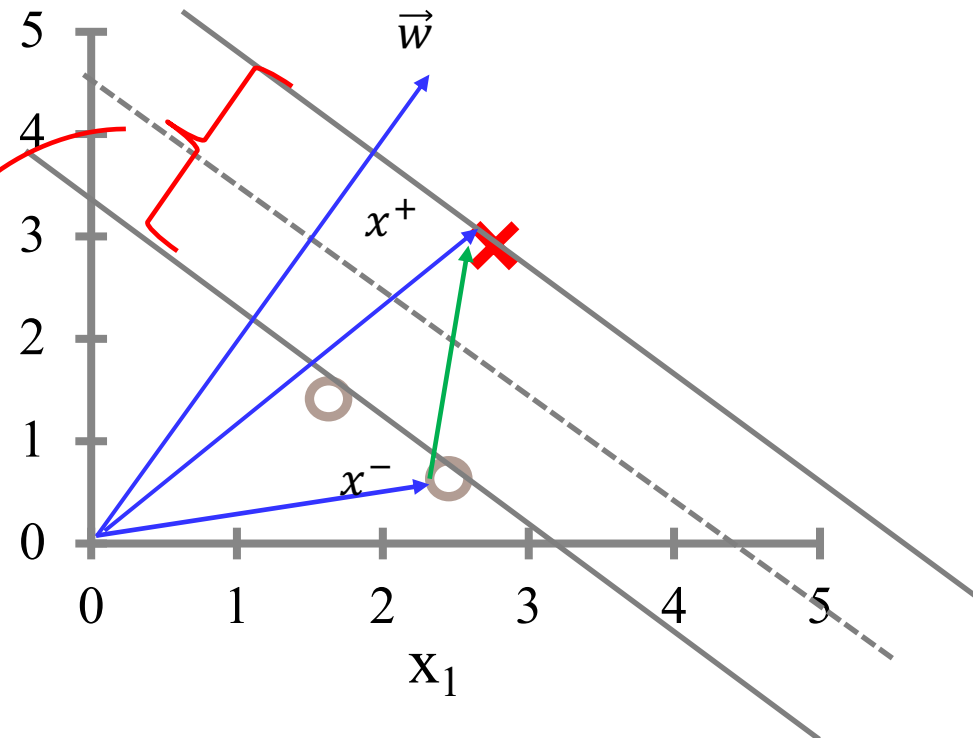
Find Max Margin

Margin(width of street):

$$(x^+ \cdot \frac{\vec{w}}{\|\vec{w}\|} - x^- \cdot \frac{\vec{w}}{\|\vec{w}\|})$$

Unit vector

$$(x^+ - x^-) \cdot \frac{\vec{w}}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|}$$



Maximizing this equals to

$$\text{Max } \frac{2}{\|\vec{w}\|}$$



Minimizing this

$$\text{Min } \|\vec{w}\|$$



Minimizing this

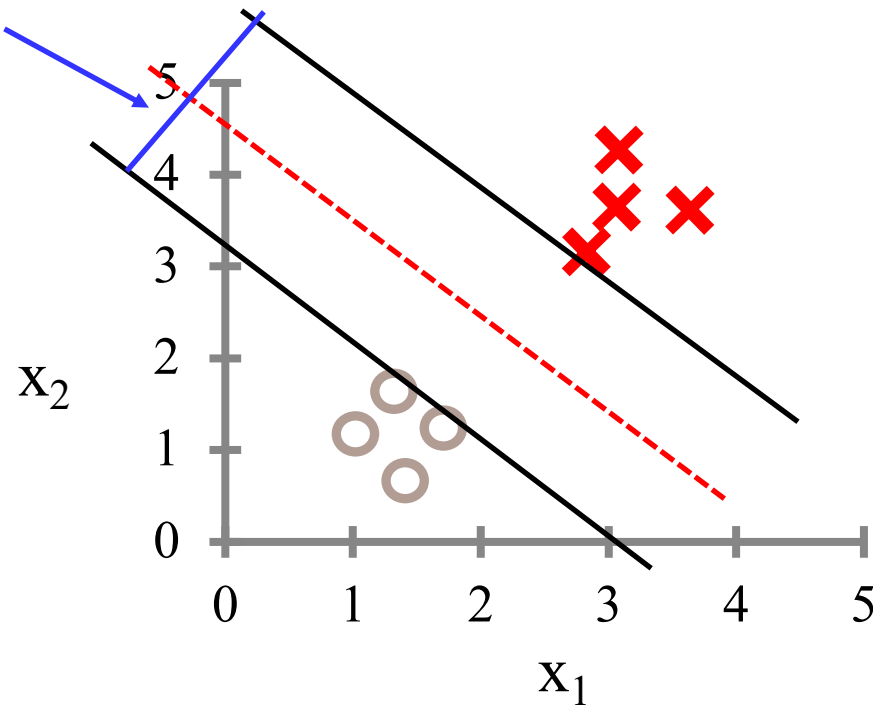
$$\text{Min } \frac{1}{2} \|\vec{w}\|^2$$

Margin with Constrains

Margin: $\frac{2}{\|\vec{w}\|}$

We cannot make this margin arbitrary large.

The condition is that there should be no samples in the margin.



$$\text{Max } \frac{2}{\|\vec{w}\|} \Rightarrow \text{Min } \frac{1}{2} \|\vec{w}\|^2$$

Constrains:

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0$$

$$\begin{aligned} y_i &= +1 \text{ for (+) samples} \\ y_i &= -1 \text{ for (-) samples} \end{aligned}$$

Turn it to Convex Optimization

Find the **minimum value** of the following function:

$$\text{Min } \frac{1}{2} \|\vec{w}\|^2$$

Constrains:

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0$$

$$y_i = +1 \text{ for (+) samples}$$

$$y_i = -1 \text{ for (-) samples}$$

Convex Optimization

1. Unconstrained optimization problem $\text{Min } f(x)$
2. Equality constrained optimization problem

$$\text{Min } f(x) \quad \text{s.t.} \quad h_i(x) = 0, \quad i = 0 \dots n$$

Lagrange multiplier method

$$L(x, \lambda) = f(x) + \sum_{i=1}^n \lambda_i h_i(x)$$

3. Inequality constrained optimization problem

$$\text{Min } f(x) \quad \text{s.t.} \quad \begin{aligned} h_i(x) &= 0, \quad i = 0 \dots n \\ g_i(x) &\leq 0, \quad i = 0 \dots k \end{aligned}$$

$$L(x, \lambda) = f(x) + \sum_{i=1}^n \lambda_i g_i(x) + \sum_{i=1}^n v_i h_i(x)$$

Lagrange multiplier

- In mathematical optimization, the method of **Lagrange multipliers** is a strategy for finding the local maxima and minima of a function subject to equality constraints.

$$\begin{aligned} &\text{maximize } f(x, y) \\ &\text{subject to: } g(x, y) = 0 \end{aligned}$$

Lagrange multipliers give us a new expression, which allow us to find the extremum of a function without thinking about constraints any more.

Define a Lagrange function

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda g(x, y) \quad \text{where } \lambda \text{ called a Lagrange multiplier.}$$

Calculate the gradient, and set to 0, to find extremum value.

$$\nabla_{x,y,\lambda} \mathcal{L}(x, y, \lambda) = \left(\frac{\partial \mathcal{L}}{\partial x}, \frac{\partial \mathcal{L}}{\partial y}, \frac{\partial \mathcal{L}}{\partial \lambda} \right)$$

Apply Lagrange multiplier

Objective: $\text{Min } \frac{1}{2} \|\vec{w}\|^2$

Subject to: $y_i(\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0$



$$L = \frac{1}{2} \|\vec{w}\|^2 - \sum \lambda_i [y_i(\vec{w} \cdot \vec{x}_i + b) - 1]$$

Lagrange
multipliers

We need find the
derivatives and set
them to 0!

The dual problem

$$L(\vec{w}, b, \lambda) = \frac{1}{2} \|\vec{w}\|^2 - \sum \lambda_i [y_i(\vec{w} \cdot \vec{x}_i + b) - 1]$$



$$\min_{\vec{w}, b} \max_{\lambda_i > 0} L(\vec{w}, b, \lambda)$$



dual problem

$$\max_{\lambda_i > 0} \min_{\vec{w}, b} L(\vec{w}, b, \lambda)$$

Partial Derivatives

Find minimum:
$$L = \frac{1}{2} \|\vec{w}\|^2 - \sum \lambda_i [y_i (\vec{w} \cdot \vec{x}_i + b) - 1]$$

$$\frac{1}{2} \|\vec{w}\|^2 = \frac{1}{2} \vec{w} \cdot \vec{w}$$

It tells us that the \vec{w} is the linear sum of some samples.

Use “some”, because some λ_i will be 0.

$$\frac{\partial L}{\partial \vec{w}} = \vec{w} - \sum \lambda_i y_i \vec{x}_i = 0 \Rightarrow \boxed{\vec{w} = \sum \lambda_i y_i \vec{x}_i}$$

$$\frac{\partial L}{\partial b} = \sum \lambda_i y_i = 0 \Rightarrow \boxed{\sum \lambda_i y_i = 0}$$

Apply Lagrange multiplier (cont.)

Find minimum: $L = \frac{1}{2} \|\vec{w}\|^2 - \sum \lambda_i [y_i (\vec{w} \cdot \vec{x}_i + b) - 1]$

We got this:

$$\vec{w} = \sum \lambda_i y_i \vec{x}_i$$

$$\sum \lambda_i y_i = 0$$

Let's plug it back to the L, see what happens...

$$L = \frac{1}{2} (\sum \lambda_i y_i \vec{x}_i) \cdot (\sum \lambda_j y_j \vec{x}_j) - (\sum \lambda_i y_i \vec{x}_i) \cdot (\sum \lambda_j y_j \vec{x}_j) - \sum \lambda_i y_i b + \sum \lambda_i$$

$$\max_{\lambda_i} L = \sum \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j \boxed{\vec{x}_i \cdot \vec{x}_j}$$

Need to find the maximum λ_i of this.

We discovered that this optimization depends only on the dot product of pairs of samples.

Rewrite Decision Rule

If $\vec{w} \cdot \vec{u} + b \geq 0$, then the prediction is positive (+) samples
This is the decision rule.

$$\vec{w} = \sum \lambda_i y_i \vec{x}_i$$

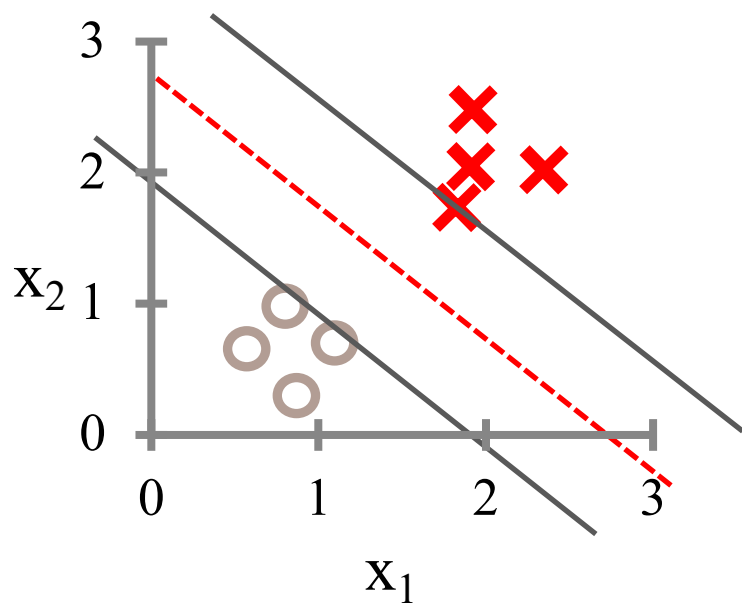
$$b = \sum \left(\frac{1}{y_j} - \sum \lambda_i y_i x_i \cdot x_j \right)$$

We discovered that the **decision rule** also depends only on the dot product of pairs of samples.

$$\sum \lambda_i y_i \boxed{\vec{x}_i \cdot \vec{u}} + b \geq 0$$

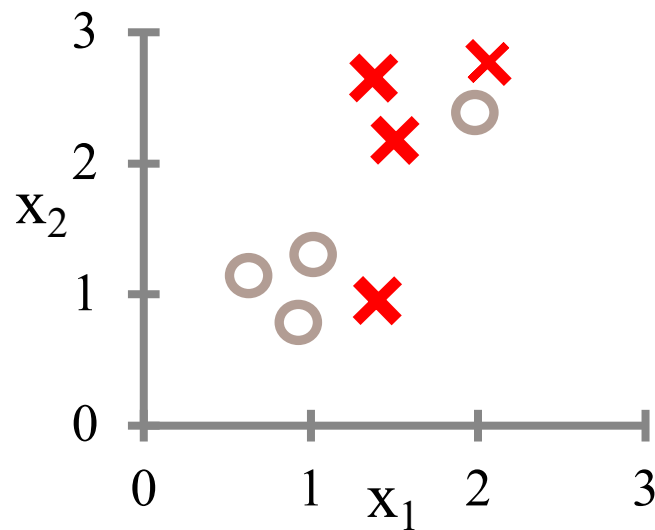
Two Cases of SVM

Case1: Linearly Separable



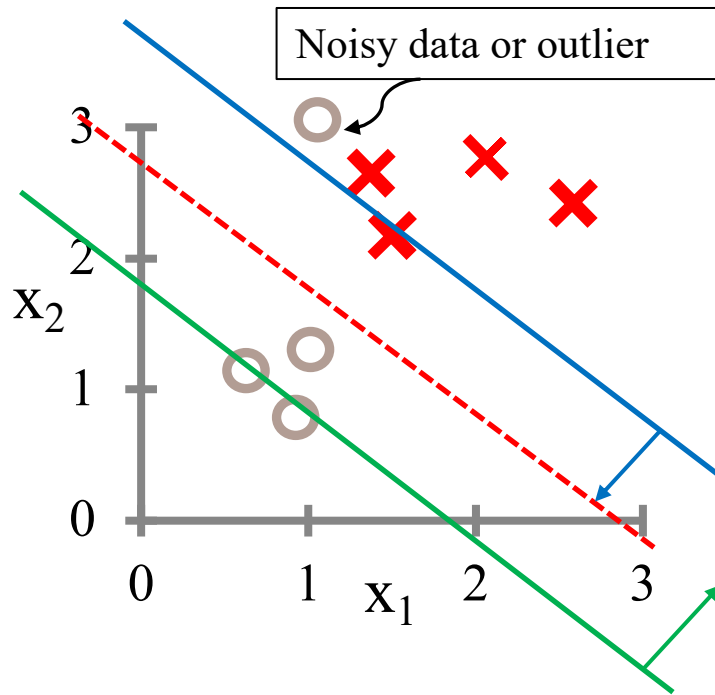
Hard margin SVM

Case2: None-Linearly Separable



Soft margin SVM

None-Linearly Separable



Linearly Separable:

Objective: $\text{Min } \frac{1}{2} \|\vec{w}\|^2$

Subject to: $y_i(\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0$

None-Linearly Separable:

Objective: $\text{Min } \frac{1}{2} \|\vec{w}\|^2 + c \sum \xi_i$

Subject to: $y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i$

where $\xi_i > 0$

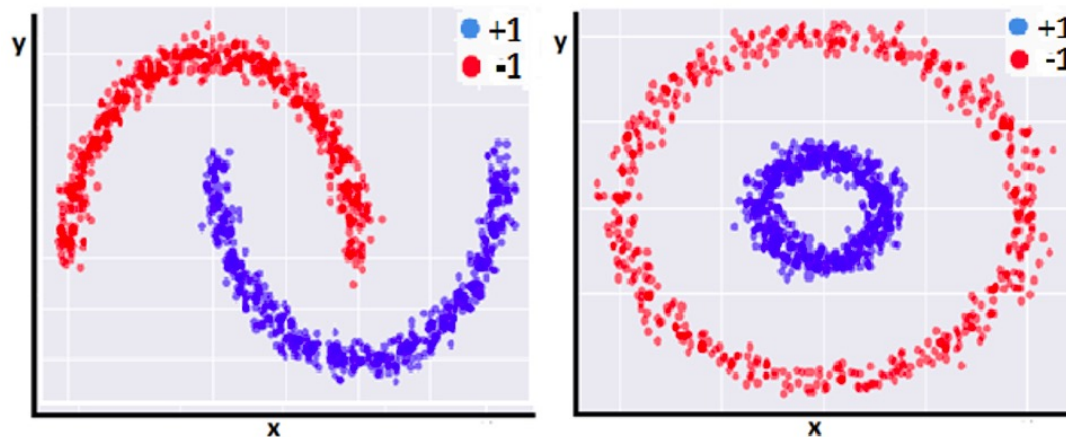
allow some missclassification.

e.g. $\xi_i = 3$, then the blue line moving down, and the green line moving up.

Larger c leads a narrower margin.

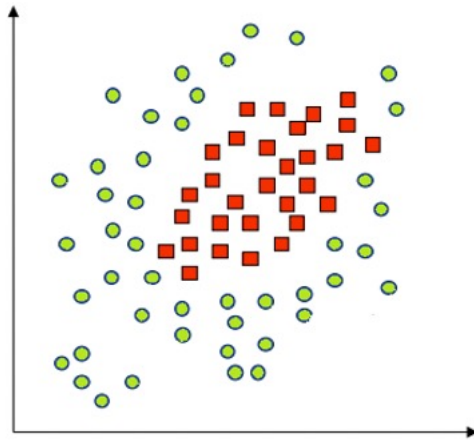
- Make sure there are as few misclassifications as possible.
- c value determine how important the ξ_i should be.
- Trade-off between maximizing the margin and allowing some misclassifications.

- What is the none-linearly separable data is not caused by the **noisy data and outliers**?
- What is the data are characteristically none-linearly seperable?

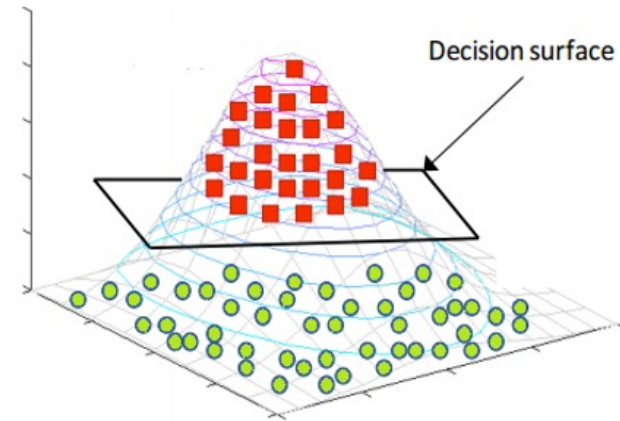


Kernel Function

Dimension Transformation



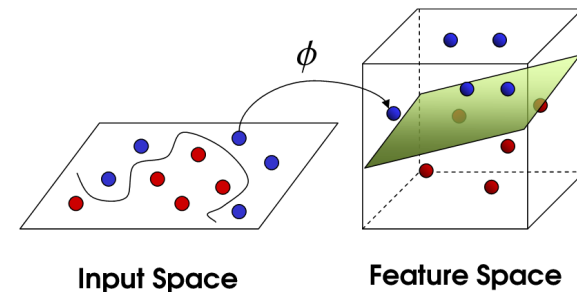
Kernel



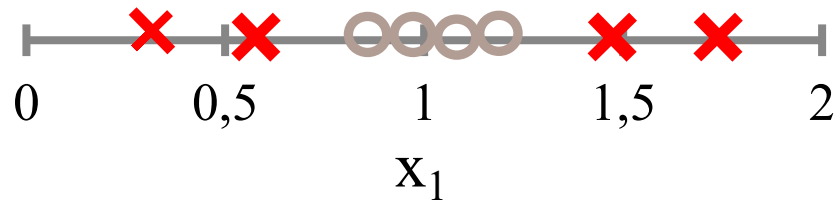
None-Linearly Separable

Linearly Separable

- A method of using a **linear classifier** to solve a **non-linear problem**.
- Kernel function is a mathematical function that is used to **transform input data** into a **higher-dimensional feature space**, where it may be more easily separated by a **hyperplane**.



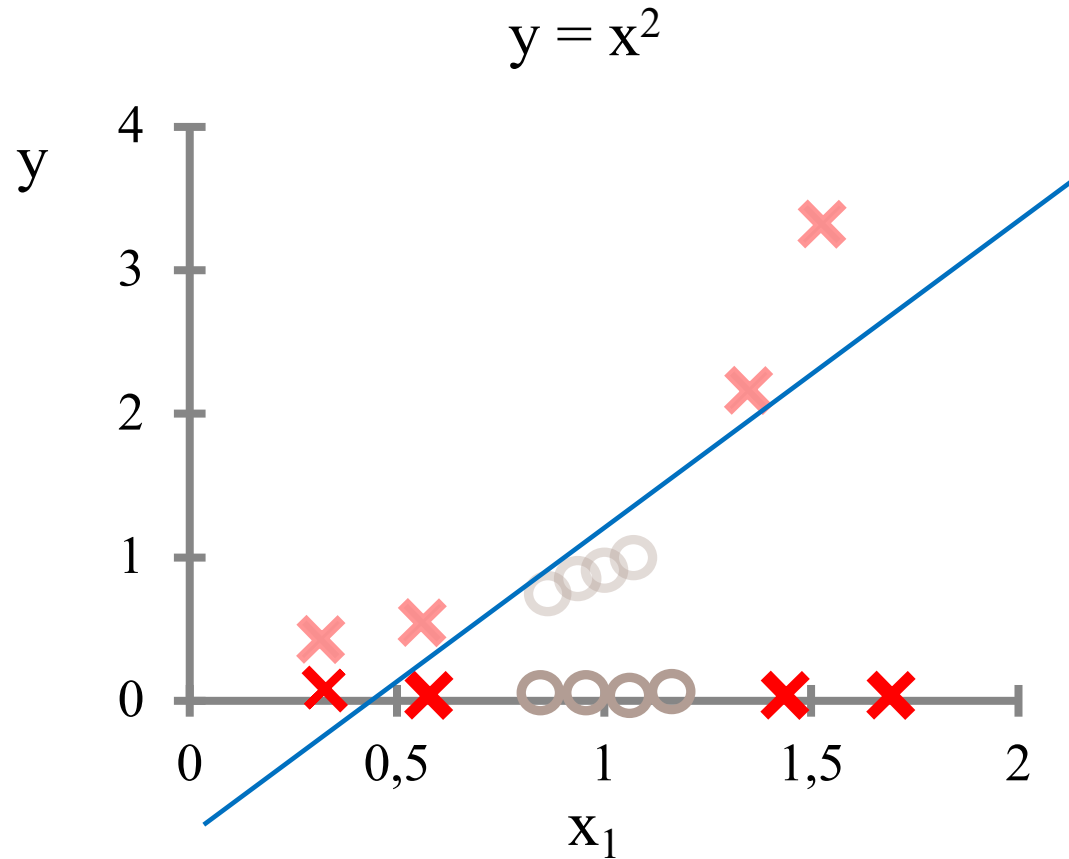
Dimension Transformation (cont.)



1-Dimensional data

None-Linearly Separable

Dimention Transformation (cont.)



Why we choose
square of x , not other
function?

In other words, how
do we decide how
to transform the
data?

SVM with Kernel Function

Let's say input x_i is transformed by a function $\phi(\vec{x}_i)$ into a higher dimensional space.

Hyperplane can be define: $\phi(\vec{x}_i) \cdot \vec{w} + b$

The original problem:

Objective: $\text{Min } \frac{1}{2} \|\vec{w}\|^2$

Subject to: $y_i(\vec{w} \cdot \phi(\vec{x}_i) + b) - 1 \geq 0$

Its dual problem:

$\max_{\lambda_i} L = \sum \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$ **Subject to:** $\sum \lambda_i y_i = 0$

It is the dot product of any two samples in higher dimensional space.
It is hard to calculate the relation in higher dimension.

Kernel Function (Trick)

Kernel function $K(\cdot, \cdot)$

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$

It calculates the **dot product of two sample's vectors in another space** without transforming all samples into that high dimension space.

Popular Kernel functions

Kernel	$K(x, x_j) =$	Kernel	$K(x, x_j) =$
Lineal	$x^\top x_j$	Powered	$-\ x - x_j\ ^\beta \quad 0 < \beta \leq 1$
Polynomial	$(a \times x^\top x_j + b)^d$	Log	$-\log(1 + \ x - x_j\ ^\beta) \quad 0 < \beta \leq$
RBF	$e^{(-\frac{\ x-x_j\ ^2}{\sigma^2})}$	Generalized Gaussian	$e^{-(x-x_j)^\top A(x-x_j)}$ where A is a symmetric PD matrix
Sigmoid	$\tanh(\sigma x^\top x_j + r)$	Hybrid	$e^{-\frac{\ x-x_j\ ^2}{\sigma^2}} \sim \dots$

Radial basis function kernel - RBF

Polynomial Kernel

How the **polynomial kernel** calculates high-dimensional relationships

$$(x_i x_j + b)^d$$

d – is the degree of the polynomial.
b – coefficient.

$$\begin{aligned} (x_i x_j + \frac{1}{2})^2 &= (x_i x_j + \frac{1}{2}) (x_i x_j + \frac{1}{2}) \\ &= (x_i x_j + x_i^2 x_j^2 + \frac{1}{4}) \\ &= \left(x_i, x_i^2, \frac{1}{2}\right) \cdot \left(x_j, x_j^2, \frac{1}{2}\right) \end{aligned}$$

Polynomial Kernel (cont.)

$$(x_i \times x_j + \frac{1}{2})^2 = (x_i x_j + \frac{1}{2}) (x_i x_j + \frac{1}{2})$$

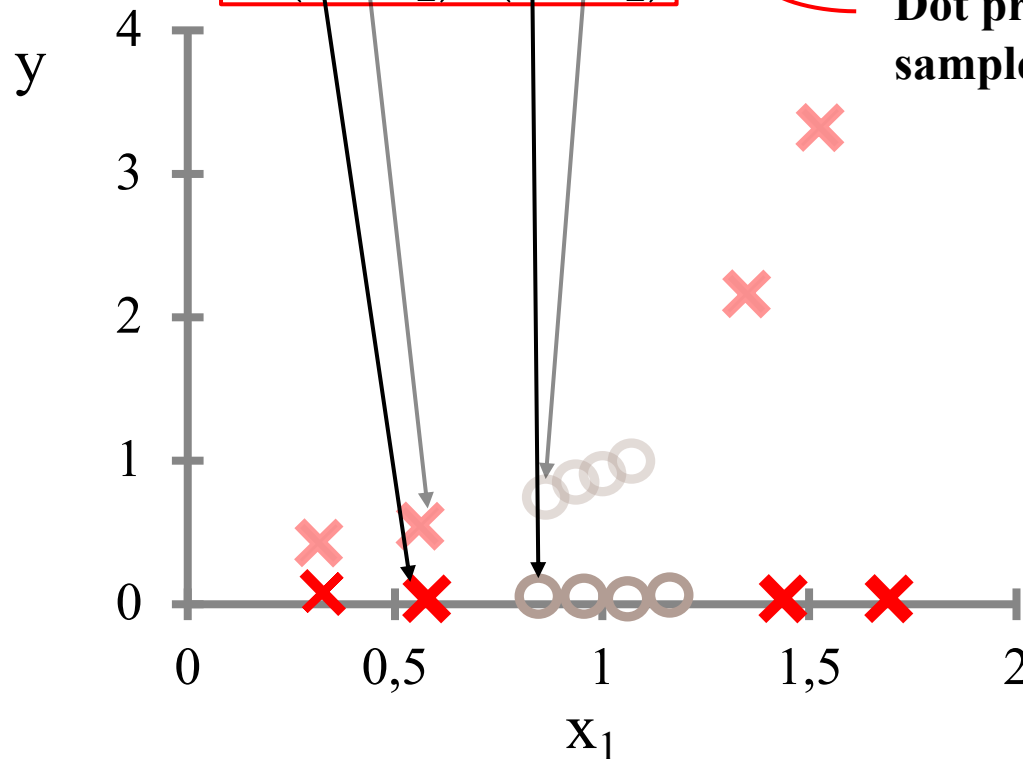
$$= (x_i x_j + x_i^2 x_j^2 + \frac{1}{4})$$

$$= \left(x_i, x_i^2, \frac{1}{2}\right) \cdot \left(x_j, x_j^2, \frac{1}{2}\right)$$

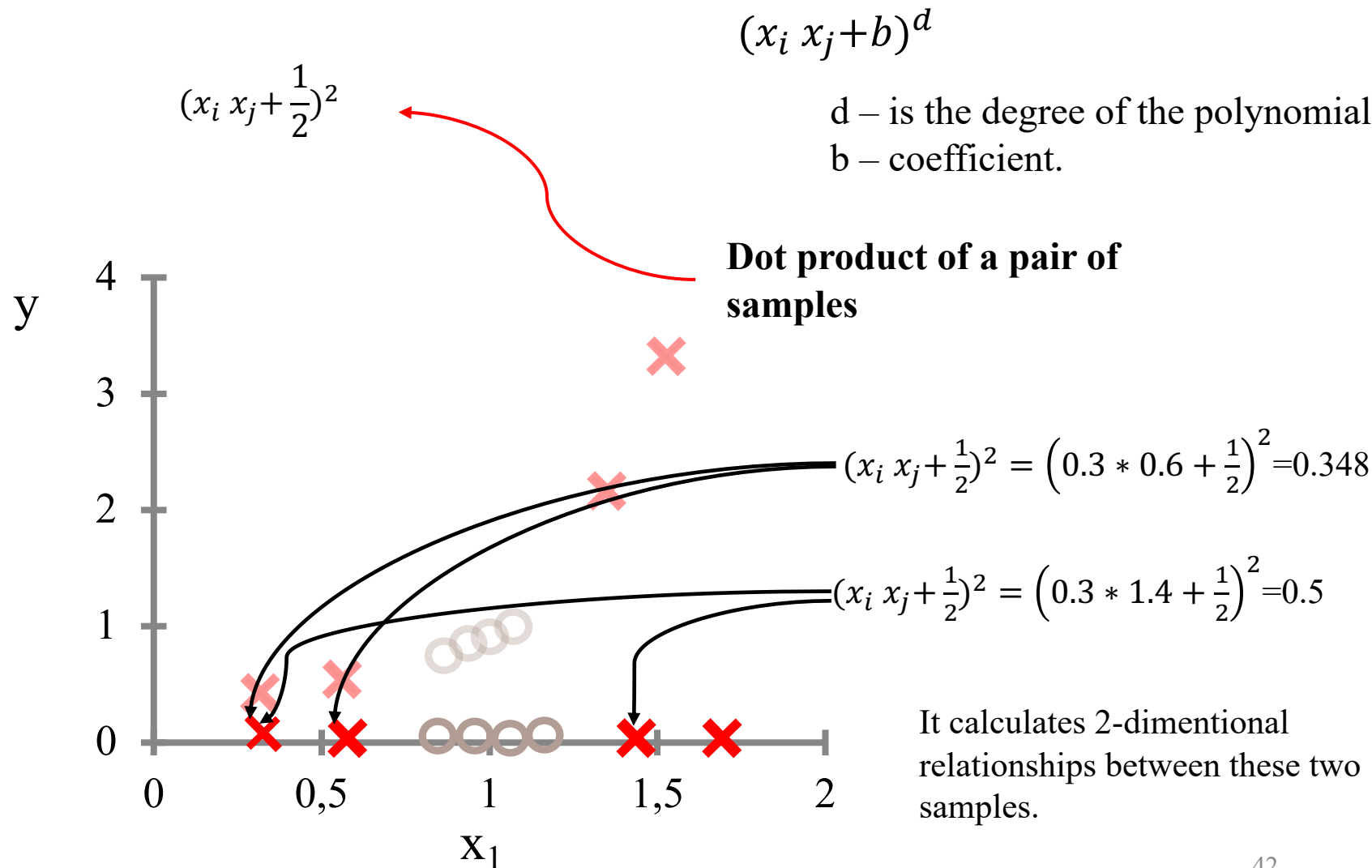
$$(x_i \times x_j + b)^d$$

d – is the degree of the polynomial
b – coefficient.

Dot product of a pair of samples




Polynomial Kernel (cont.)



Radial basis function kernel - RBF

RBF kernel: $e^{-\gamma(x_i - x_j)^2}$



γ (gamma), scaling the distance, or scaling the amount of influence two samples have on each other.

Like polynomial kernel, when you plug value to RBF function, it calculates **high dimensional relationship**.

$$e^{-\gamma(x_i - x_j)^2} = \text{high dimensional relationship}$$

Let's explain this with polynomial kernel first.

RBF(cont.)

$$(x_i x_j + b)^d$$

d – is the degree of the polynomial.

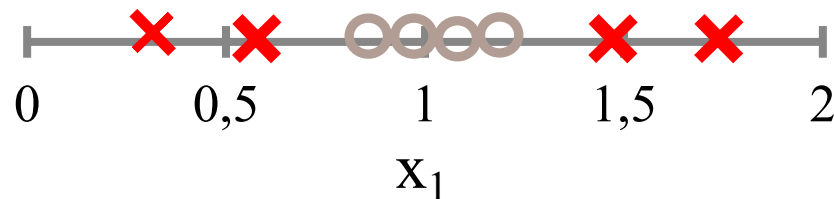
b – coefficient.

Dot product with a single
coordinates.

Let's set $b = 0$

$$(x_i x_j)^d = x_i^d x_j^d = (x_i^d) \cdot (x_j^d)$$

Original:



RBF(cont.)

$(x_i \ x_j + b)^d$ d – is the degree of the polynomial.
 b – coefficient.

Set $d = 1$:

$$(x_i \ x_j)^1 = x_i^1 \ x_j^1 = (x_i^1) \cdot (x_j^1)$$

Set $d = 2$:

$$(x_i \ x_j)^2 = x_i^2 \ x_j^2 = (x_i^2) \cdot (x_j^2)$$

New coordinates is just the square of the original value on the original axis.

.....

Set $d = \infty$:

$$(x_i \ x_j)^\infty = x_i^\infty \ x_j^\infty = (x_i^\infty) \cdot (x_j^\infty)$$


It gives us the dot product with coordinates for a infinite number of dimension.

$$(x_i \ x_j)^1 + (x_i \ x_j)^2 + \dots + (x_i \ x_j)^\infty = (x_i^1, x_i^2, \dots, x_i^\infty) \cdot (x_j^1, x_j^2, \dots, x_j^\infty)$$

RBF(cont.)

RBF kernel: $e^{-\gamma(x_i - x_j)^2} = e^{-\gamma(x_i^2 + x_j^2)} e^{\gamma 2x_i x_j}$

let's set $\gamma = \frac{1}{2}$

$$e^{-\frac{1}{2}(x_i - x_j)^2} = e^{-\frac{1}{2}(x_i^2 + x_j^2)} e^{x_i x_j}$$


Create the **Taylor Series Expansion** of this term

Taylor Series Expansion

In mathematics, the Taylor series or **Taylor expansion of a function** is *an infinite sum of terms* that are expressed in terms *of the function's derivatives* at a single point.

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \dots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \dots$$



$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

RBF(cont.)

RBF kernel: $e^{-\gamma(x_i - x_j)^2} = e^{-\gamma(x_i^2 + x_j^2)} e^{\gamma 2x_i x_j}$

$$e^{-\frac{1}{2}(x_i - x_j)^2} = \underbrace{e^{-\frac{1}{2}(x_i^2 + x_j^2)}}_s \boxed{e^{x_i x_j}}$$

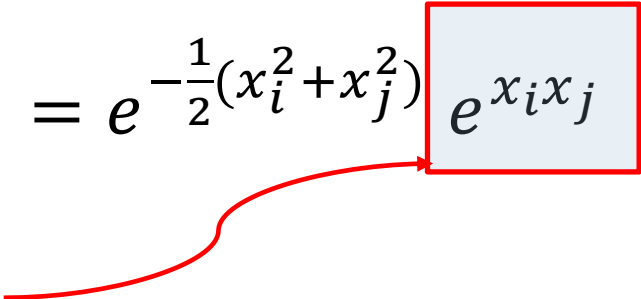
$$e^{x_i x_j} = (1, x_i, \sqrt{\frac{1}{2!}} x_i^2, \sqrt{\frac{1}{3!}} x_i^3, \dots, \sqrt{\frac{1}{\infty!}} x_i^\infty) \cdot (1, x_j, \sqrt{\frac{1}{2!}} x_j^2, \sqrt{\frac{1}{3!}} x_j^3, \dots, \sqrt{\frac{1}{\infty!}} x_j^\infty)$$

$$e^{-\frac{1}{2}(x_i^2 + x_j^2)} e^{x_i x_j} = s e^{x_i x_j}$$

$$= (s, s x_i, s \sqrt{\frac{1}{2!}} x_i^2, s \sqrt{\frac{1}{3!}} x_i^3, \dots, s \sqrt{\frac{1}{\infty!}} x_i^\infty) \cdot (s, s x_j, s \sqrt{\frac{1}{2!}} x_j^2, s \sqrt{\frac{1}{3!}} x_j^3, \dots, s \sqrt{\frac{1}{\infty!}} x_j^\infty)$$

RBF(cont.)

RBF kernel: $e^{-\gamma(x_i - x_j)^2} = e^{-\gamma(x_i^2 + x_j^2)} e^{\gamma 2x_i x_j}$

$$e^{-\frac{1}{2}(x_i - x_j)^2} = e^{-\frac{1}{2}(x_i^2 + x_j^2)} e^{x_i x_j}$$


Create the **Tayler Series Expansion** of this term $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$

$$e^{x_i x_j} = 1 + x_i x_j + \frac{(x_i x_j)^2}{2!} + \frac{(x_i x_j)^3}{3!} + \dots + \frac{(x_i x_j)^\infty}{\infty!}$$

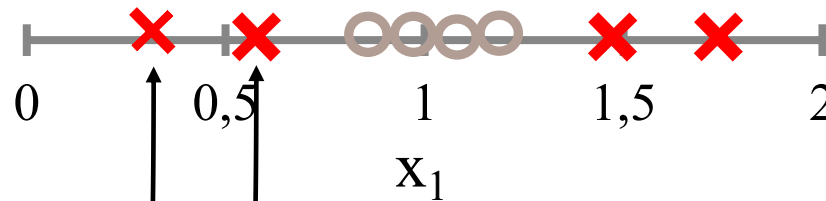


Turn it to dot product form

$$e^{x_i x_j} = (1, x_i, \sqrt{\frac{1}{2!}} x_i^2, \sqrt{\frac{1}{3!}} x_i^3, \dots, \sqrt{\frac{1}{\infty!}} x_i^\infty) \cdot (1, x_j, \sqrt{\frac{1}{2!}} x_j^2, \sqrt{\frac{1}{3!}} x_j^3, \dots, \sqrt{\frac{1}{\infty!}} x_j^\infty)$$

RBF(cont.)

Original:



RBF kernel ($\gamma=1$): $e^{-\gamma(x_i - x_j)^2} = e^{-1(0.3-0.6)^2} = 0.913$

The value we calculated at the end is the relationship between the two samples in **infinite-dimensions**.

About the author of SVM



Vladimir Naumovich Vapnik is a computer scientist, researcher, and academic. He is one of the main developers of the Vapnik–Chervonenkis theory of statistical learning and the co-inventor of the support-vector machine method and support-vector clustering algorithms. [Wikipedia](#)

Place of birth: [Soviet Union](#)

Awards: [Paris Kanellakis Award](#)

Education: [Russian Academy of Sciences](#), [National University of Uzbekistan](#)

h-index: 97

Notable students: [Bernhard Schölkopf](#), [Klaus-Robert Müller](#)

Academic advisor: [Alexander Lerner](#)

Support-Vector Networks

CORINNA CORTES
VLADIMIR VAPNIK
AT&T Bell Labs., Holmdel, NJ 07733, USA

corinna@neural.att.com
vlad@neural.att.com

Experiments with Digit Recognition

Classifier	Raw error, %
Human performance	2.5
Decision tree, CART	17
Decision tree, C4.5	16
Best 2 layer neural network	6.6
Special architecture 5 layer network	5.1

Degree of polynomial	Raw error, %	Support vectors	Dimensionality of feature space
1	12.0	200	256
2	4.7	127	~33000
3	4.4	148	$\sim 1 \times 10^6$
4	4.3	165	$\sim 1 \times 10^9$
5	4.3	175	$\sim 1 \times 10^{12}$
6	4.2	185	$\sim 1 \times 10^{14}$
7	4.3	190	$\sim 1 \times 10^{16}$

- Thank you!