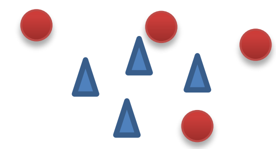
Classification

- The question is
  - Put them in the right bucket
- Supervised learning
  - The data has a true class. The categories are clearly labeled.
  - Accuracy is the guiding consideration

Clustering



- The question is
  - Try separate them in two groups
- Unsupervised learning
  - The data doesn’t have a true class. The categories are not clearly labeled.
  - Groups are shaped by similarity
  - There may be multiple ways to cluster them
  - Either by color, or by shape

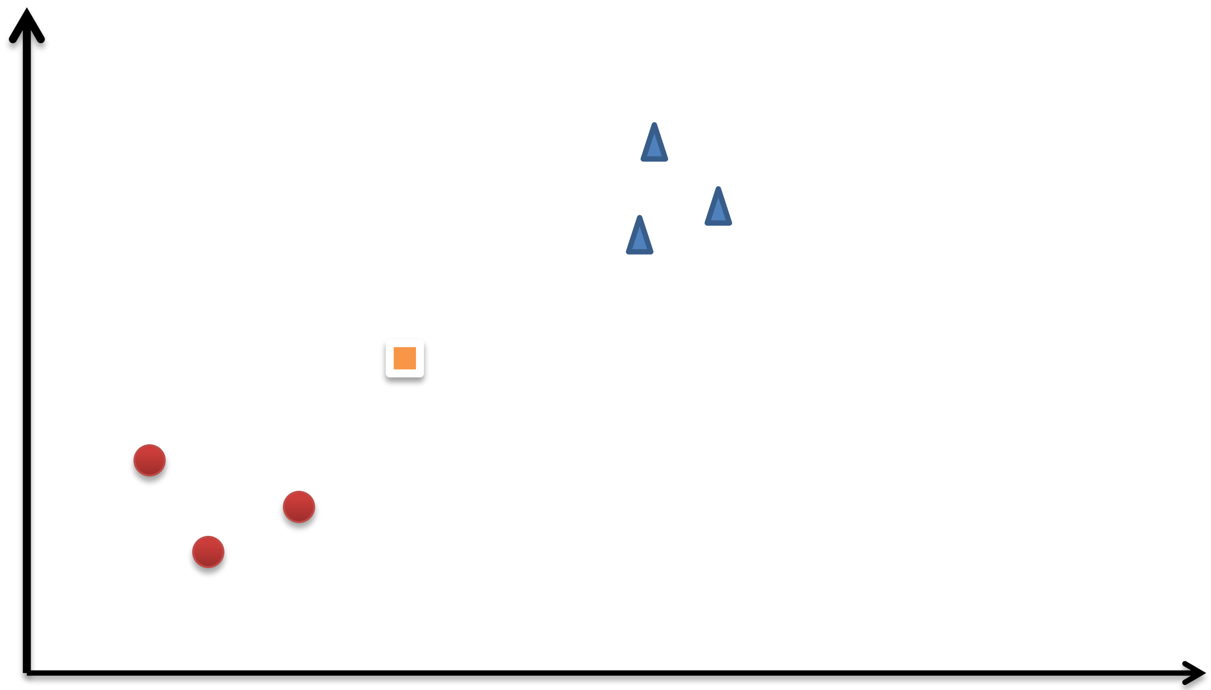
Classification Application Examples

- Spam filters
  - Whether the email is legitimate email
  - Whether it should be delivered to you

- Fraud detection
  - When a payment required is generated from a credit card
  - Credit card company calculated a probability that this is a legitimate transaction

- Psychological diagnosis
  - Classify people into a particular diagnostic category based on a range of symptoms

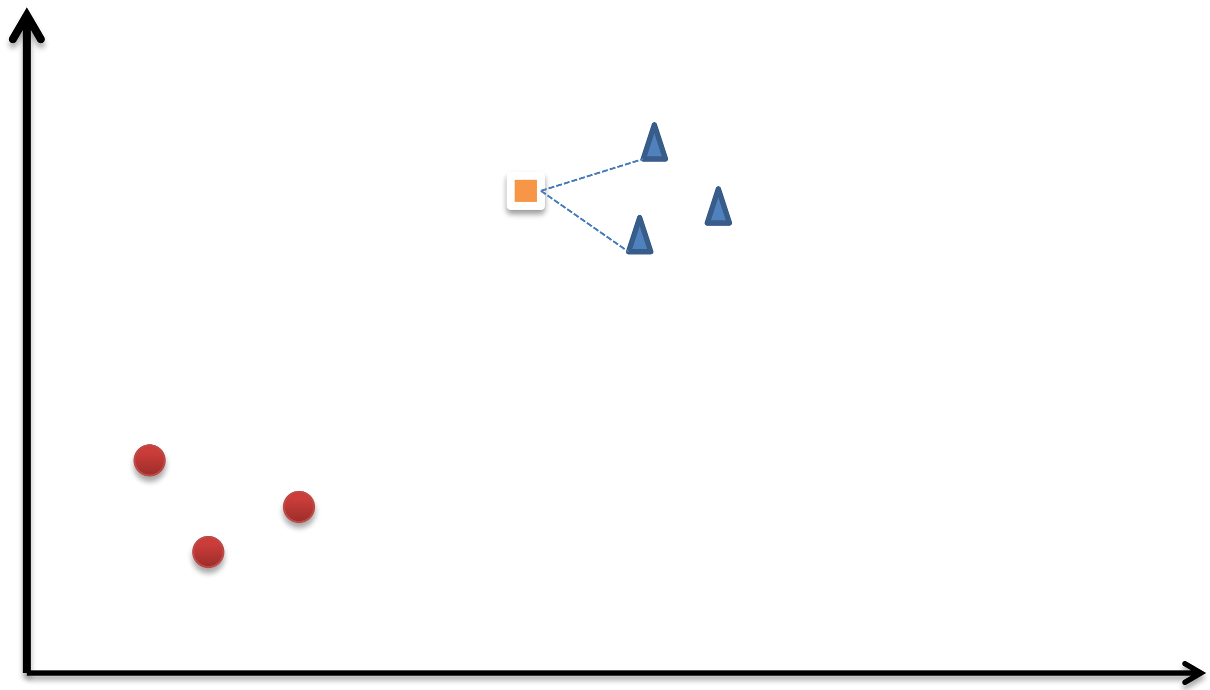
K-NN (K nearest neighbors)



There are two known groups: the red dots, and the blue dots

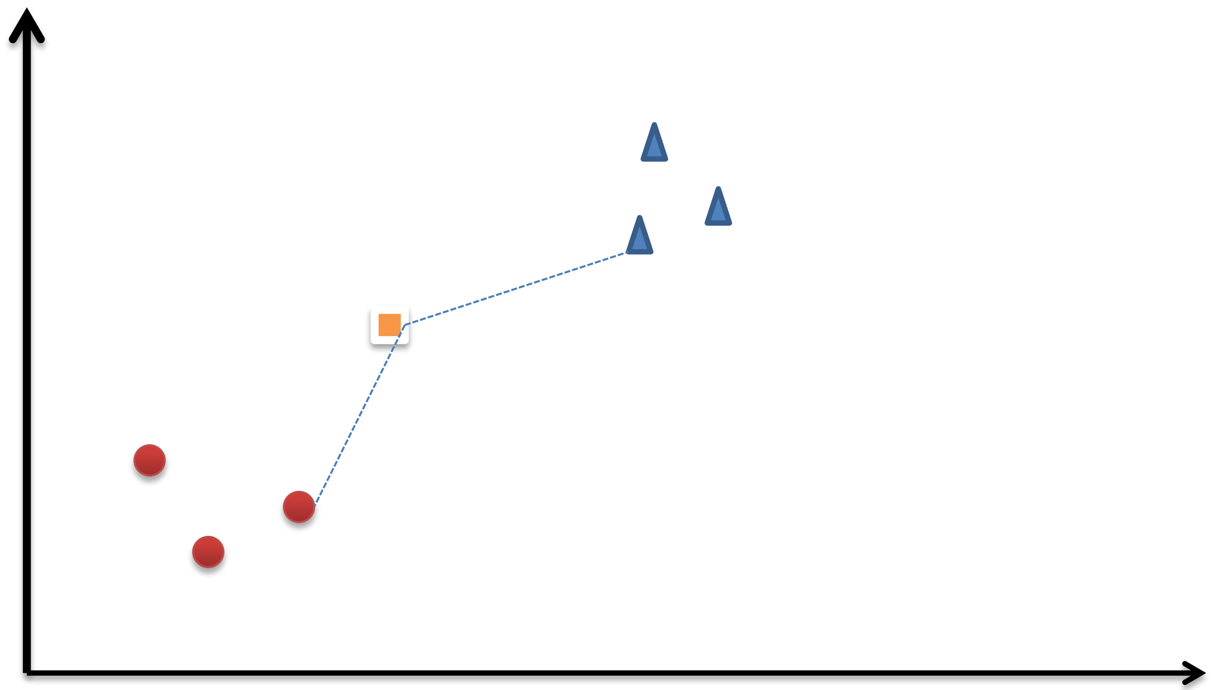
Our task is to determine, which group should the new case (square) belong to.

K-NN (K nearest neighbors)



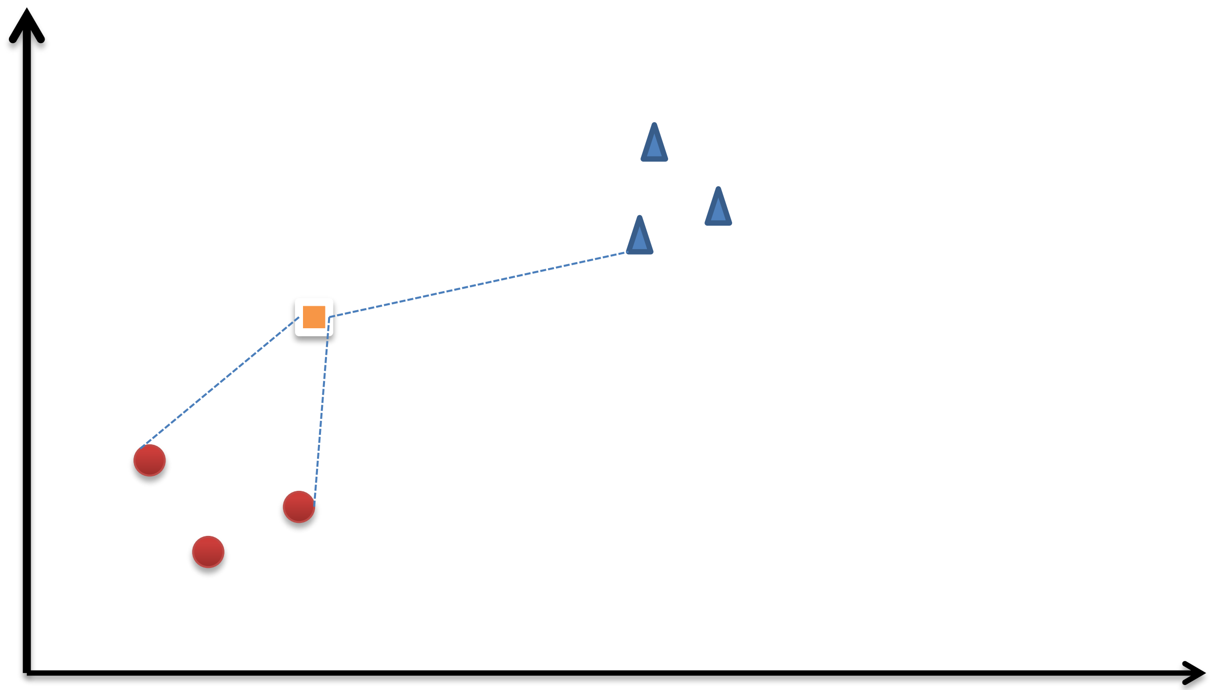
When we are using 2 nearest neighbor method.
We first find out which two dots are the closest.
Then we know that the new case is the blue group.

K-NN (K nearest neighbors)



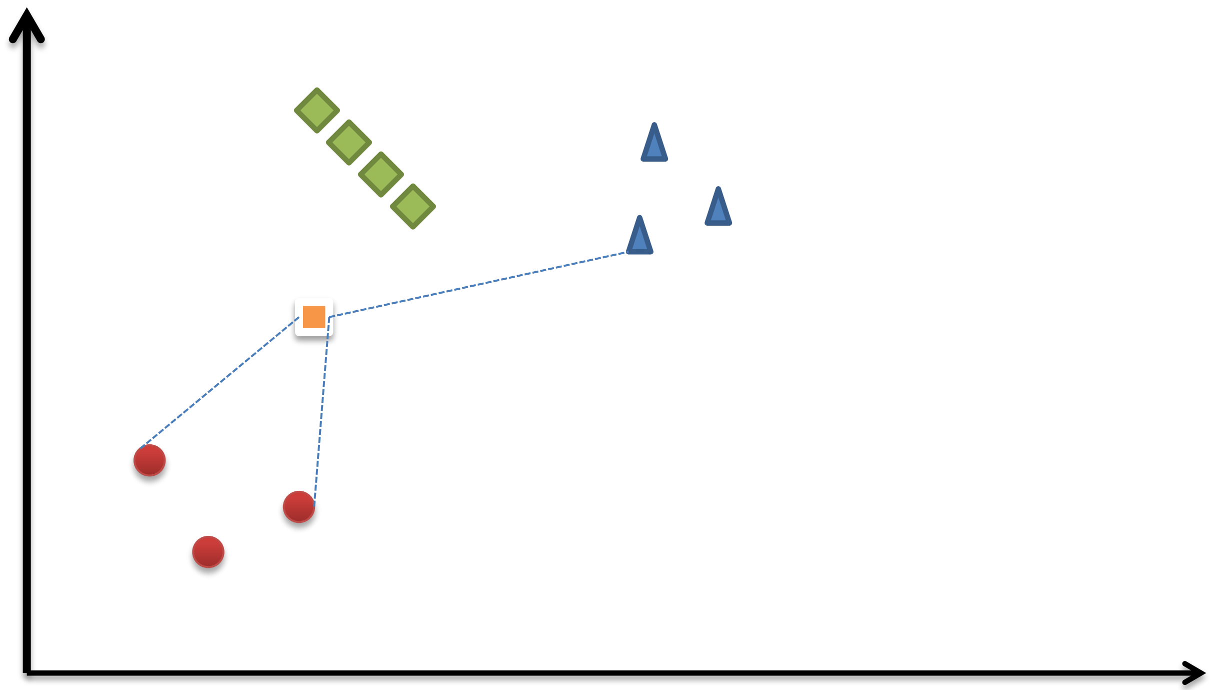
But what if the new case is here?
Then the vote is 1:1
So, we should use K=3 at least.

K-NN (K nearest neighbors)



For example, the 3 nearest neighbor is shown in the picture.
Then the new case is in the red
The accuracy is 66% (2:1)

K-NN (K nearest neighbors)



What if there are 3 groups?
We should at least use K = 5

K-NN Example

Product Name	Stability	Strength	Customer Rating / Class
Type-1	7	7	Bad
Type-2	7	4	Bad
Type-3	3	4	Good
Type-4	1	4	Good

A new product: stability = 3, strength = 7, Class = ?

K-NN Example
Calculate the similarity using Euclidean Distance

$$d(p,q) = d(q,p) = \sqrt{(q1-p1)^2 + (q2-p2)^2 + \cdots + (qn-pn)^2}$$

Product Name	Stability	Strength	Class	Distance to the new product (3,7)
Type-1	7	7	Bad	Sqrt((7-3)*(7-3)+(7-7)*(7-7))=4
Type-2	7	4	Bad	5
Type-3	3	4	Good	3
Type-4	1	4	Good	3.6

K-NN Example
Rank the distance

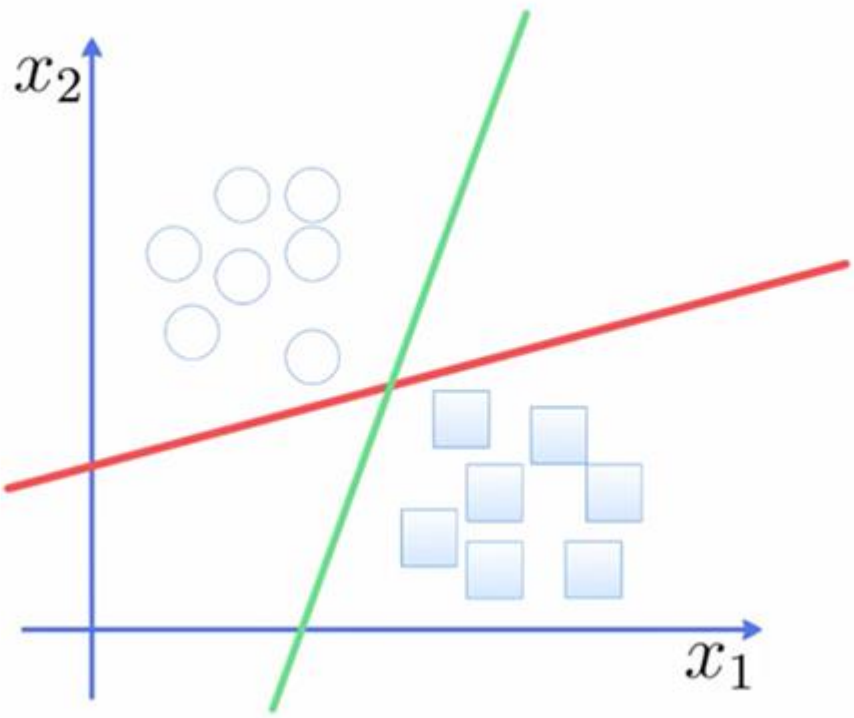
$$d(p,q) = d(q,p) = \sqrt{(q1-p1)^2 + (q2-p2)^2 + \cdots + (qn-pn)^2}$$

Product Name	Stability	Strength	Class	Distance	Rank
Type-1	7	7	Bad	4	3
Type-2	7	4	Bad	5	4
Type-3	3	4	Good	3	1
Type-4	1	4	Good	3.6	2

When using K=1,2,3, classify the new product as Good
Usually when there are two groups, we use K = 3 at least.
Next, let's see another example of KNN in Excel to classify the follower species.

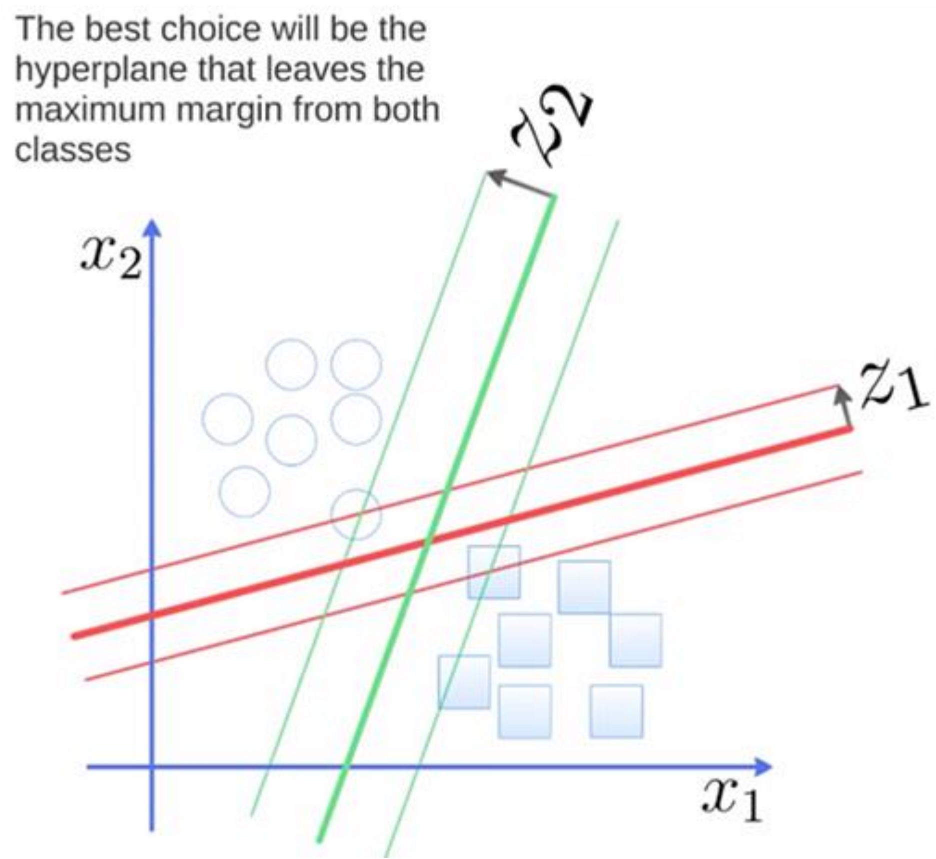
Support Vector Machine

- We have two groups
  - The square group and the circle group
  - X1 and X2 are called features,
  - Which one is the better separating hyperplane?



Support Vector Machine

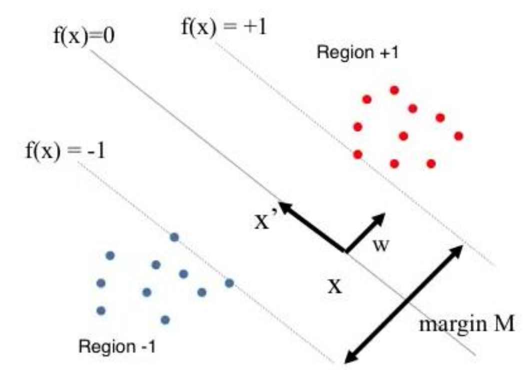
- Steps to locate the Best separating hyperplane
  - Step1: calculate the distance between a line to the closest points in each group
  - Step2: find the hyperplane that maximize the margin

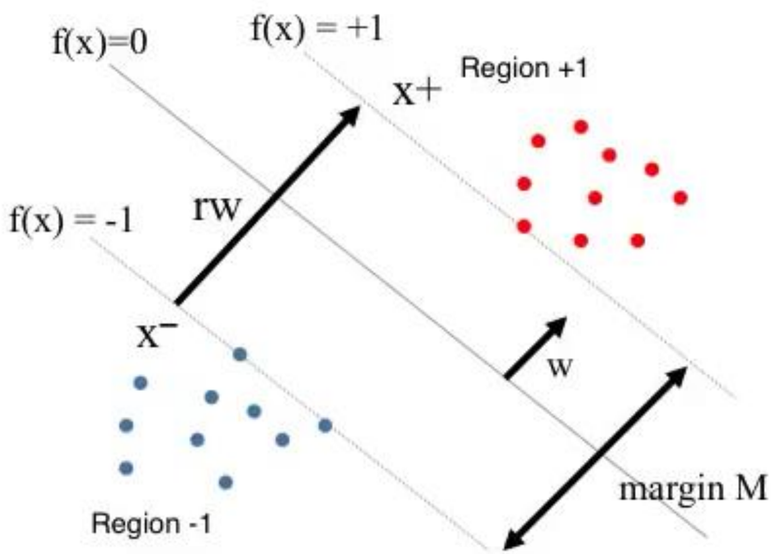


Support Vector Machine – a little algebra

- The separating hyperplane is defined by $f(x) = \vec{w} \cdot x + b = 0$
- The $\vec{w}$ vector must be perpendicular to the hyperplane
- $f(x) = +1$ and $f(x) = -1$ are parallel to $f(x) = 0$
- Pick any X_+ on $f(x) = +1$ and the corresponding X_- on $f(x) = -1$, we have
- $\vec{w} \cdot (X_+) + b = +1$, and $\vec{w} \cdot (X_-) + b = -1$
- $X_+ = X_- + r\vec{w}$, where r is a scaling factor
- Remember, we are using an existing dataset to train our classifier, and then use the classifier to classify new observations, or test it on a test dataset.

Suppose x_1 and x_2 are on the hyperplane, so we have
$\vec{w} \cdot x_1 + b = 0$
$\vec{w} \cdot x_2 + b = 0$
So we have
$\vec{w} \cdot (x_1 - x_2) = 0$
$\vec{w}$ is orthogonal to $(x_1 - x_2)$





To calculate the margin:

$\vec{w} \cdot (X_+) + b = +1 \Rightarrow$
$\vec{w} \cdot (X_- + r\vec{w}) + b = +1 \Rightarrow$
$r\|\vec{w}\|^2 + \vec{w} \cdot X_- + b = +1 \Rightarrow$
$r\|\vec{w}\|^2 = 2$
$$r = \frac{2}{\|\vec{w}\|^2}$$

Support Vector Machine
– a little algebra

- To find the best separating hyperplane, do a optimization like below

$$w^* = argmax \frac{2}{\|\vec{w}\|^2}$$

s.t. all training data points are one the correct side of the margin

- Or

$$w^* = argmin_w \sum_j {w_j}^2$$

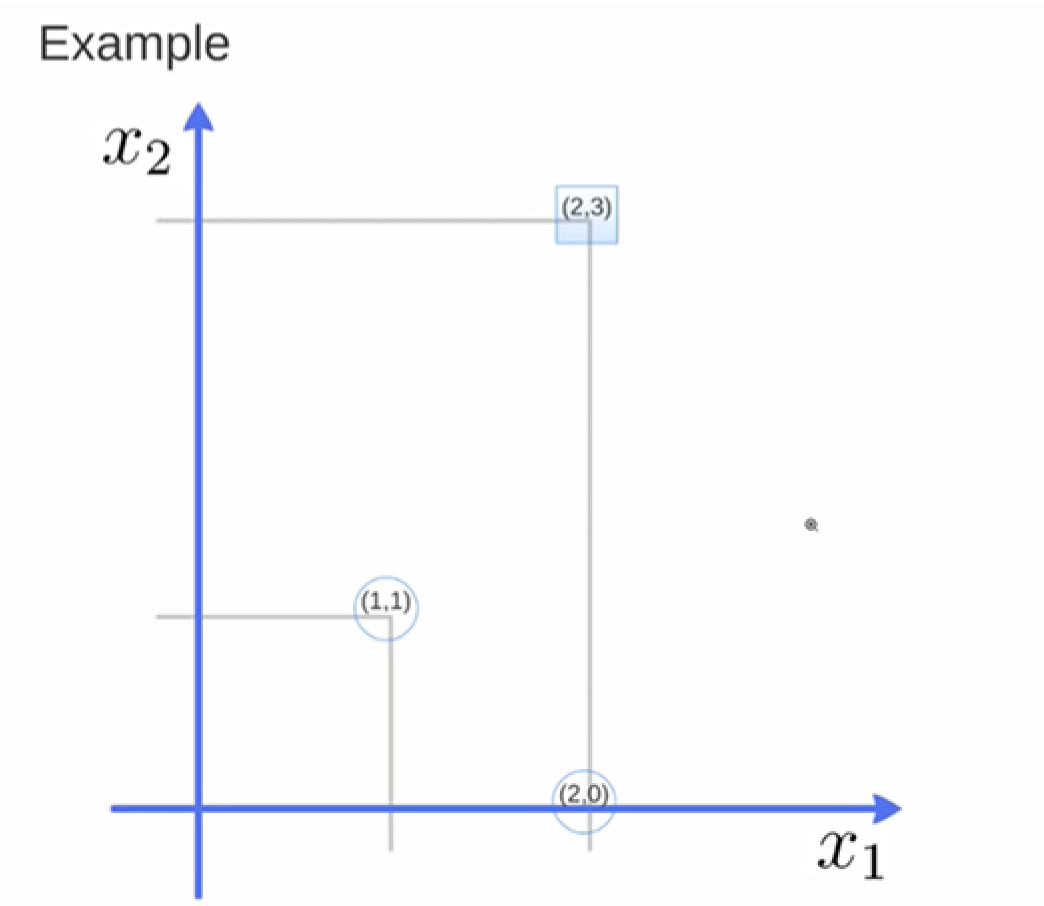
s.t. $y^{(i)}\left(\vec{w} \cdot x^{(i)} + b\right) \geq +1$

The true class

The calculated class

- As to how to solve, let’s leave that to the computer

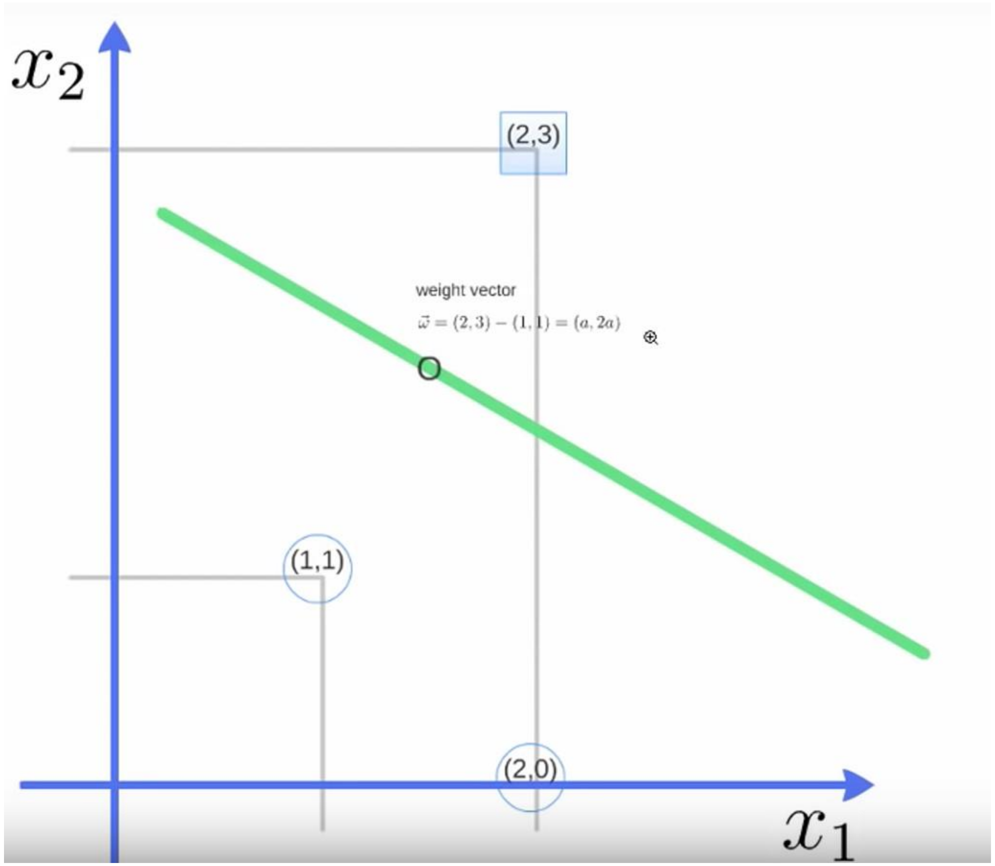
Support Vector Machine – an easy example



The training dataset has three data points and two groups

The square group
The circle group

Support Vector Machine – an easy example



After solving the super complicated optimization problem

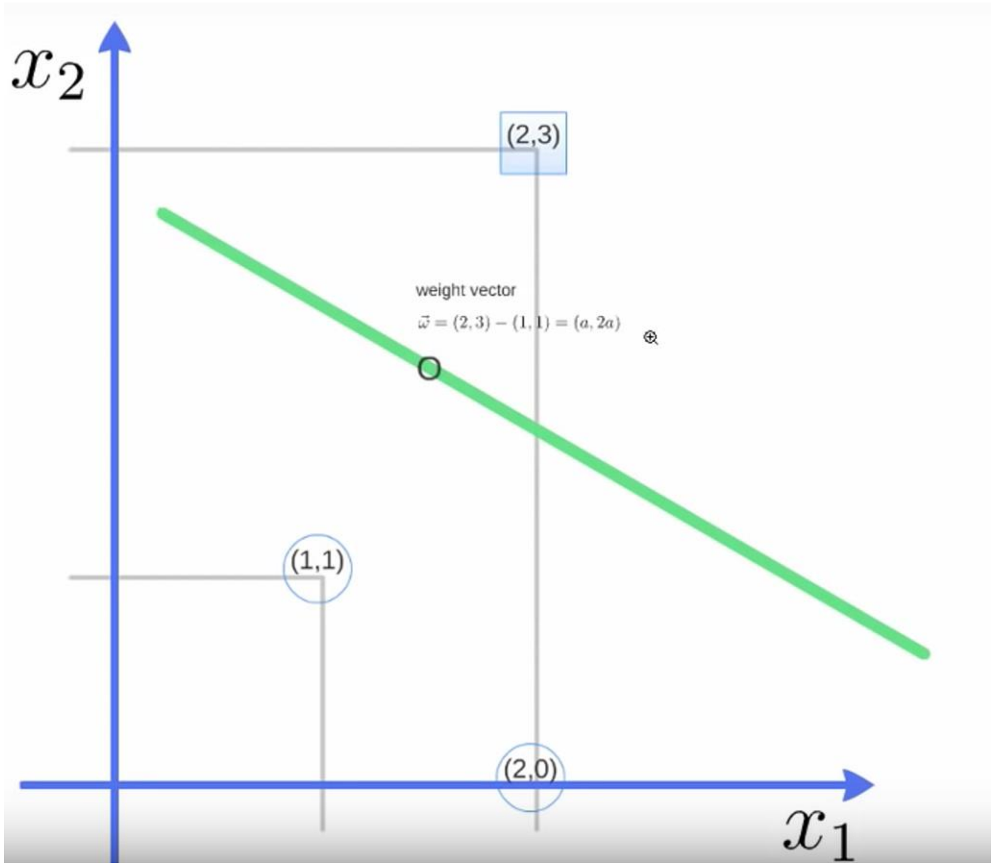
$$w^* = argmin_w \sum_j {w_j}^2$$

s.t. $$y^{(i)}(\vec{w} \cdot x^{(i)} + b) \geq +1$$

with our eyeballs, we know that the best hyperplane is the green line.

Perpendicular to the connection line between (2,3) and (1,1)

Support Vector Machine – an easy example



The weight vector must take the form of

$$a(2-1, 3-1) = (a, 2a)$$

Solve the functions:

$$\begin{cases} a + 2a + b = -1 \\ 2a + 6a + b = 1 \end{cases}$$

It doesn't matter if we use 1 or 2.
The solution is: $a = \frac{2}{5}, b = -\frac{11}{5}$

$$\vec{w} = \left(\frac{2}{5}, \frac{4}{5}\right)$$

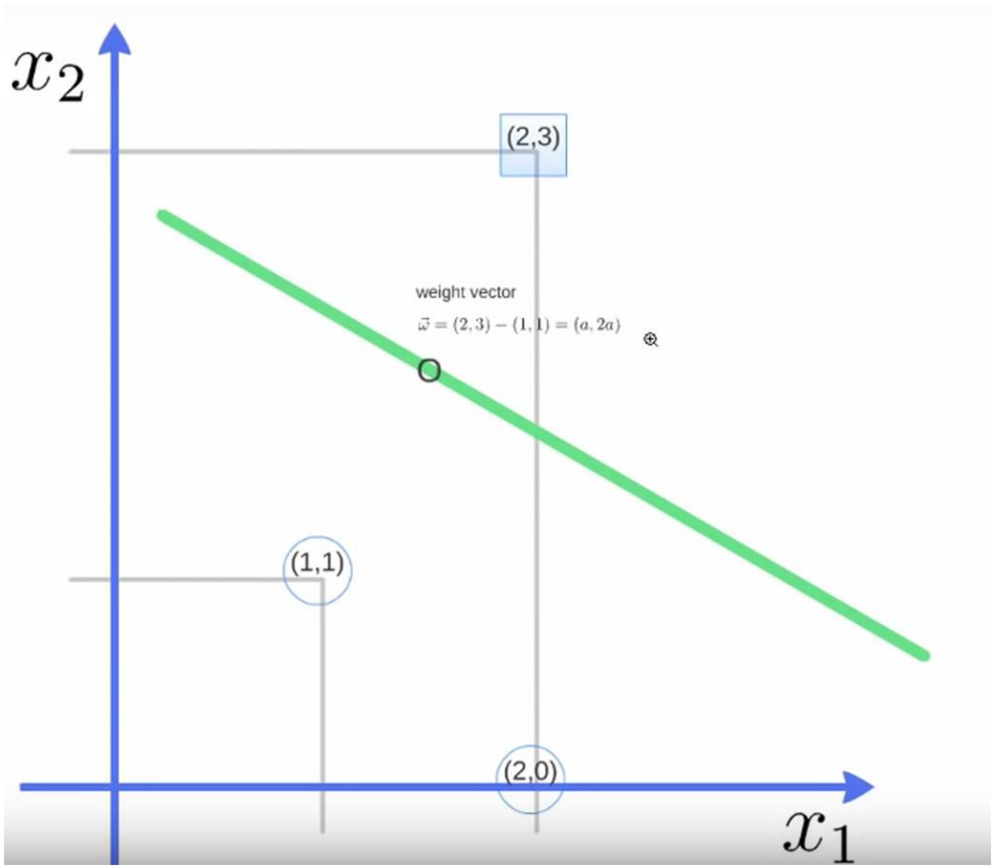
The classifier is

$$f(x) = \frac{2}{5}x_1 + \frac{4}{5}x_2 - \frac{11}{5}$$

or

$$f(x) = 2x_1 + 4x_2 - 11$$

Support Vector Machine – an easy example



Next we use the classifier

$$f(x) = 2x_1 + 4x_2 - 11$$

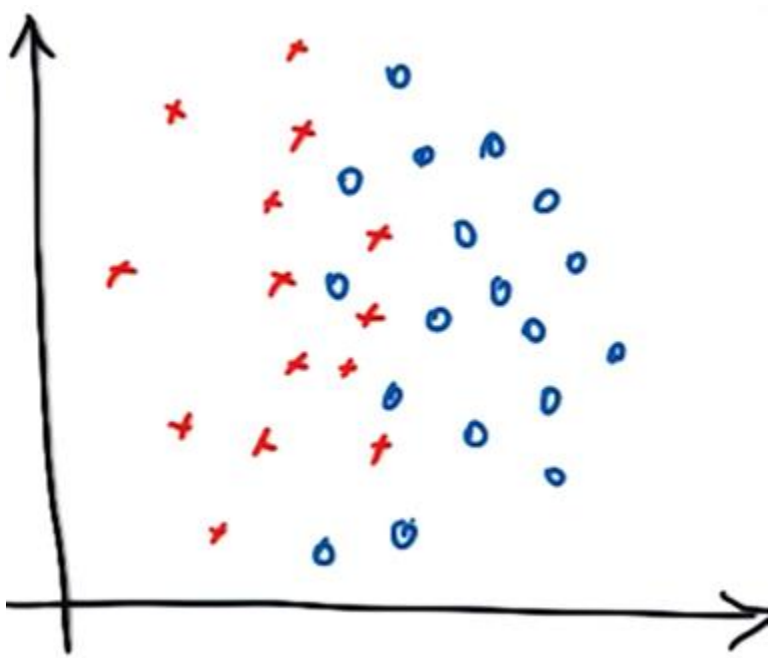
to classify new observations

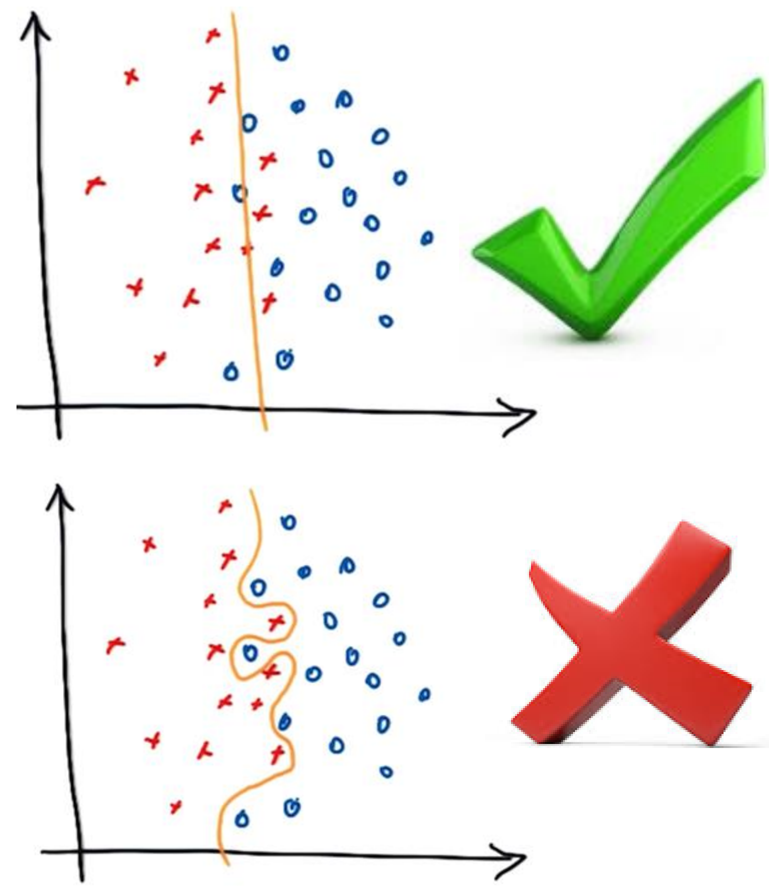
For example: $z(1,3)$ should be classified as a square since $f(z) > 0$

How about $y(1,2)$?

Support Vector Machine – the reality is not easy

- What if the two groups are not linearly separable?
  - There is no straight line you can draw to separate red and blue perfectly
  - Solution 1: still draw a straight line
  - Solution 2: draw a wield curved line





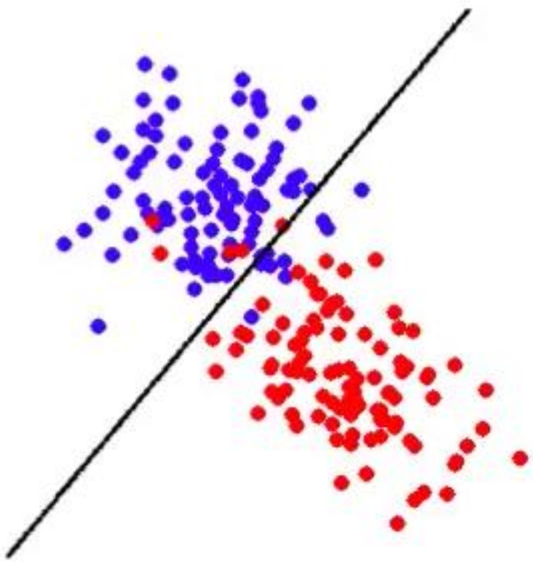
Support Vector Machine – the reality is not easy

- Now what we want?
  - A large margin
  - Few misclassified points
  - We allow some of the points to violate the margin constraints
  - But assign a cost for that

The cost or penalty

$$w^* = argmin_w \sum_j {w_j}^2 + R \sum_i \epsilon^{(i)}$$

s.t. $$y^{(i)}(\vec{w} \cdot x^{(i)} + b) \geq +1 - \epsilon^{(i)}$$

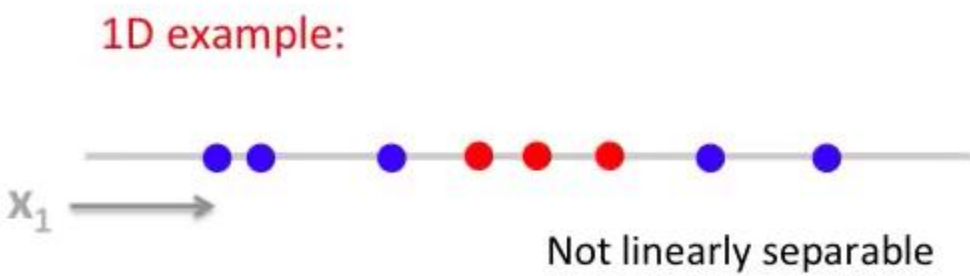


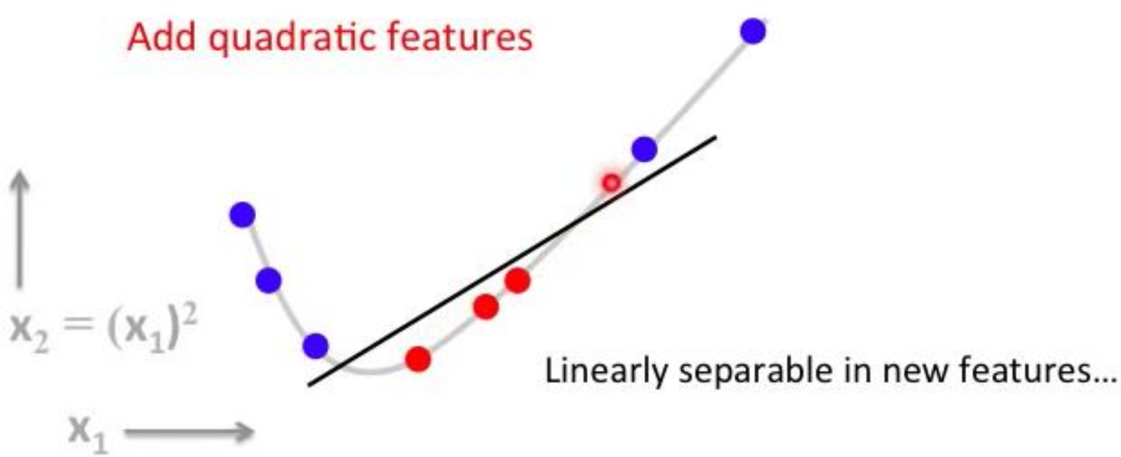
If R is large, we try to make sure every point is on the right side of the margin.

If R is small, we tolerate the violations, but try to maximize the margin

Another Solution – Kernel Trick

- Not linearly separable?
- Let’s add another feature.





We generate a secondary feature from the existing feature

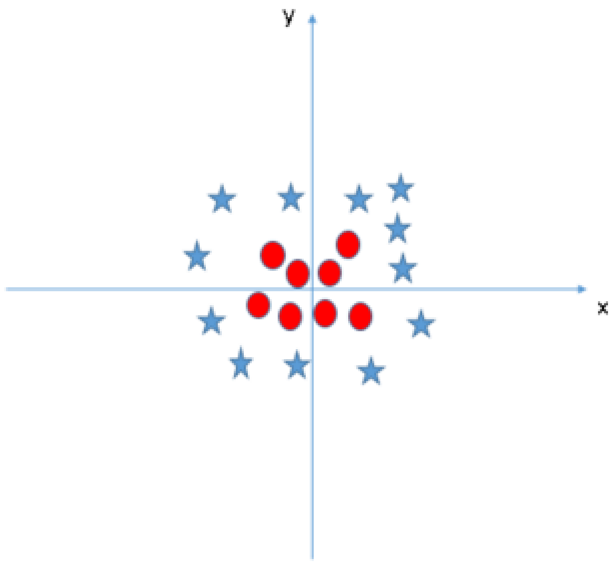
$x_2 = {x_1}^2$ is the Kernel function

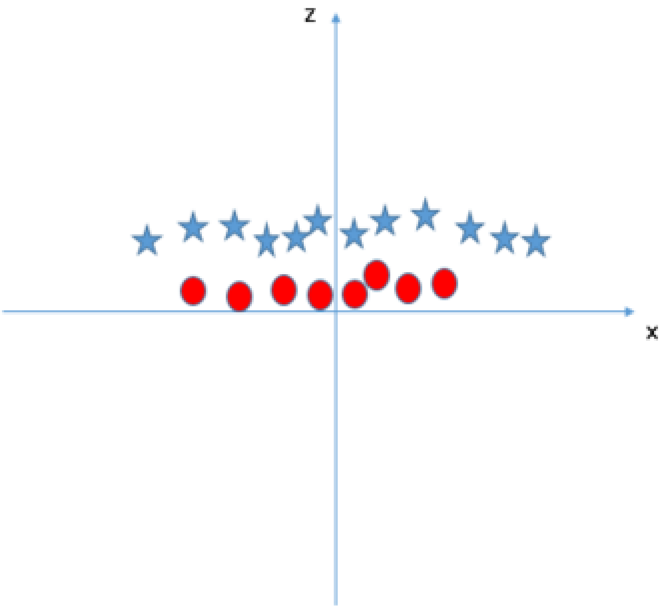
Another Solution – Kernel Trick

We generate a new feature from the existing feature

$$z = x^2 + y^2$$

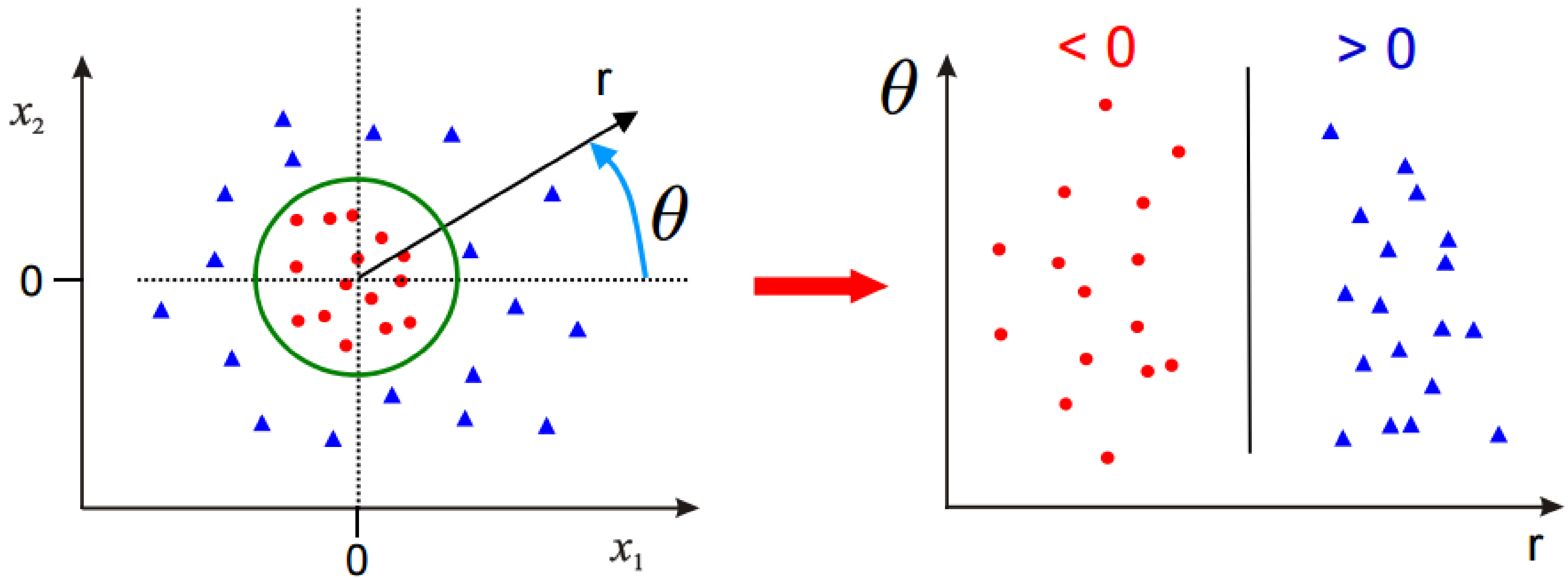
- Not linearly separable?
- Let's add another feature.





Another Solution – Kernel Trick

- Not linearly separable?
- Let's try polar coordinates.



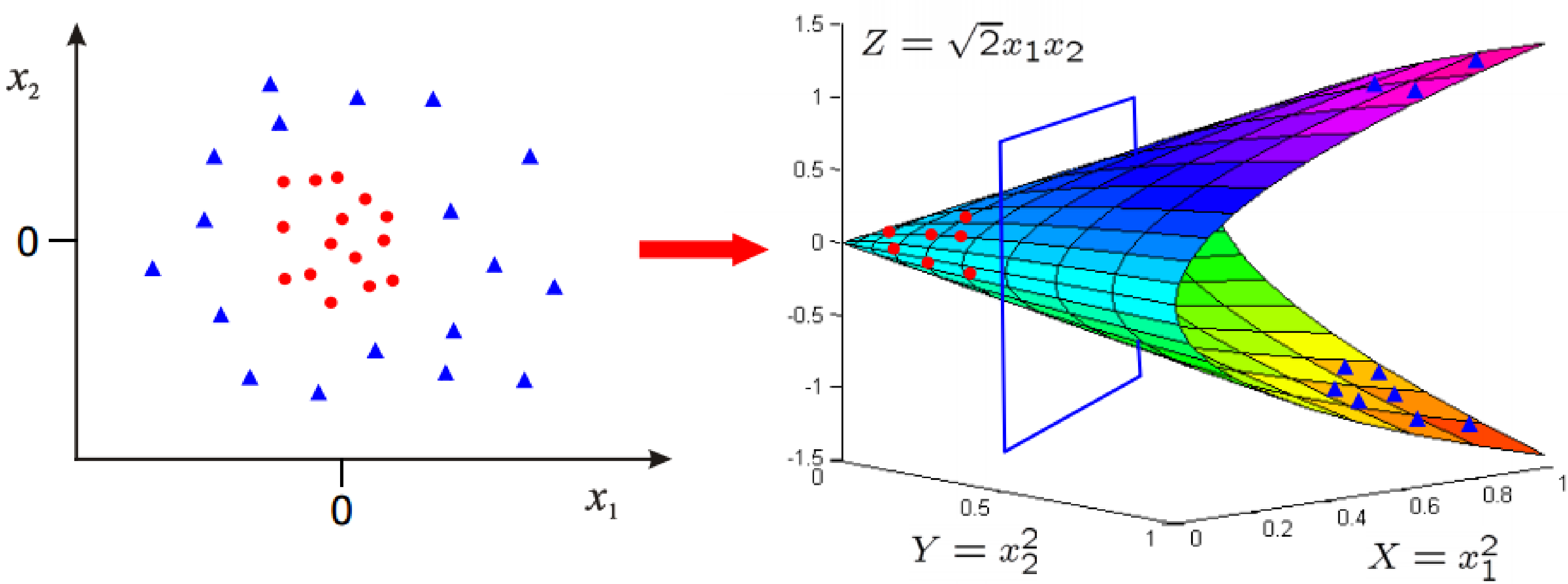
- Linear separable in the polar coordinates. The feature map is:

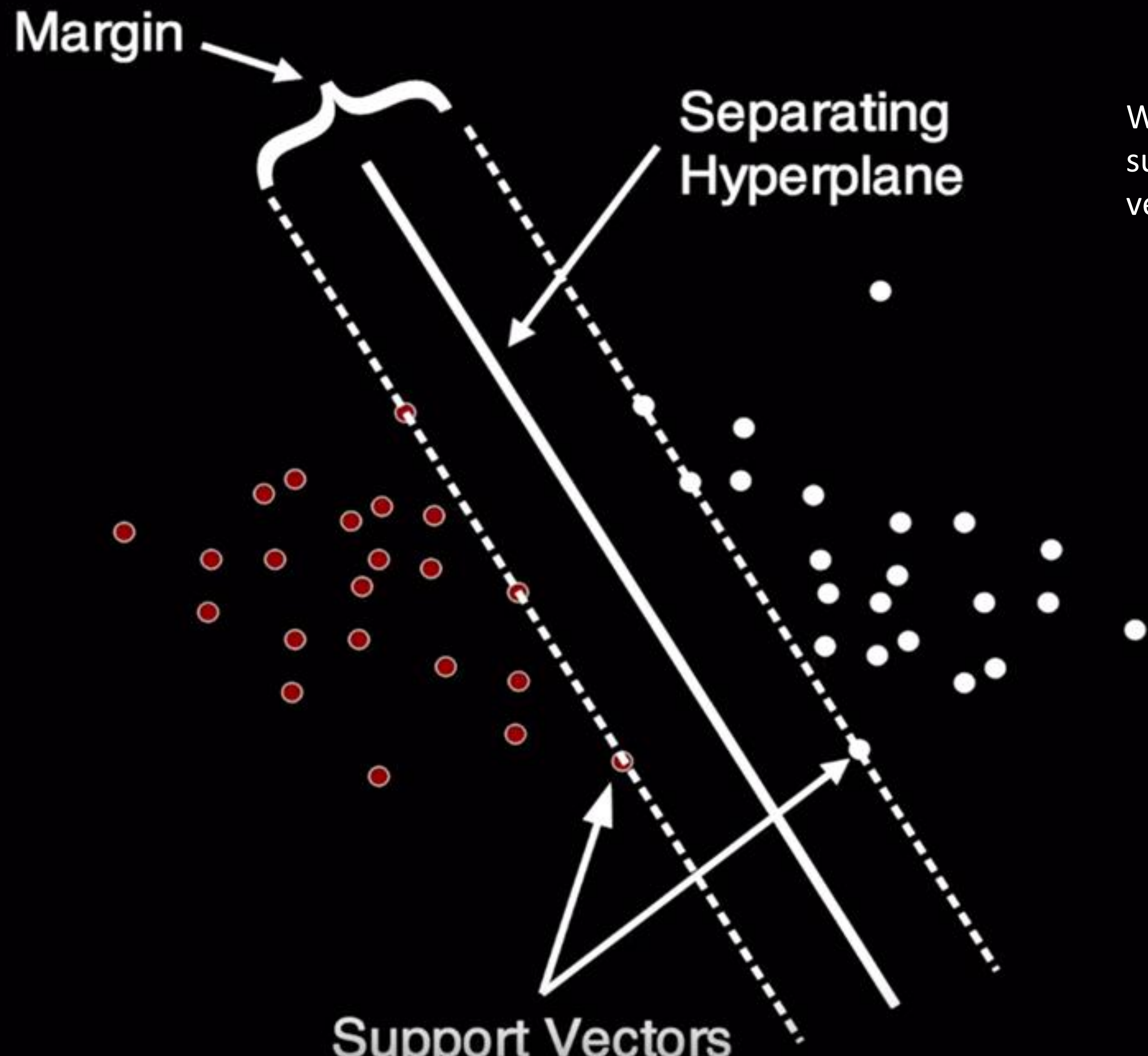
$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \to \begin{pmatrix} r \\ \theta \end{pmatrix} \qquad \mathbb{R}^2 \to \mathbb{R}^2$$

Another Solution – Kernel Trick

- Not linearly separable?
- Let's map the feature into higher dimension.
- So the problem can still be solved by a linear classifier
- $\emptyset(x)$ is a feature map
- Simply map x to $\emptyset(x)$ where data is separable

$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \qquad \mathbb{R}^2 \rightarrow \mathbb{R}^3$$





What are support vectors???