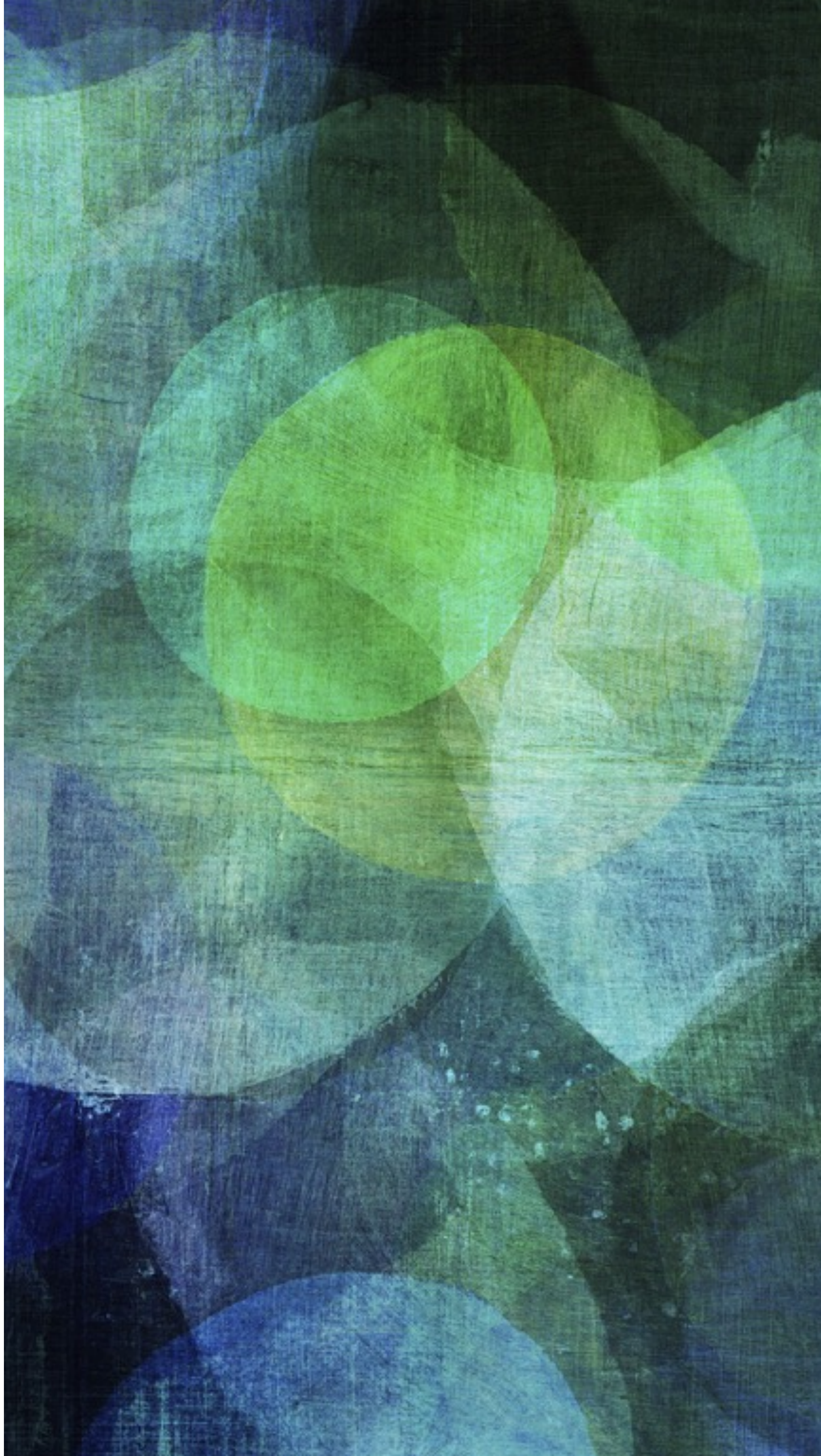
SUPPORT VECTOR MACHINE



SUPPORT VECTOR MACHINE

SVM is difficult to understand. Most people use it as a black box tool. But it is useful to have an idea about how the algorithm works.

To understand SVM you need to understand the following:

- Ideas of support vector and kernel
- Large margin classification (building off of our understanding of logistic regression).

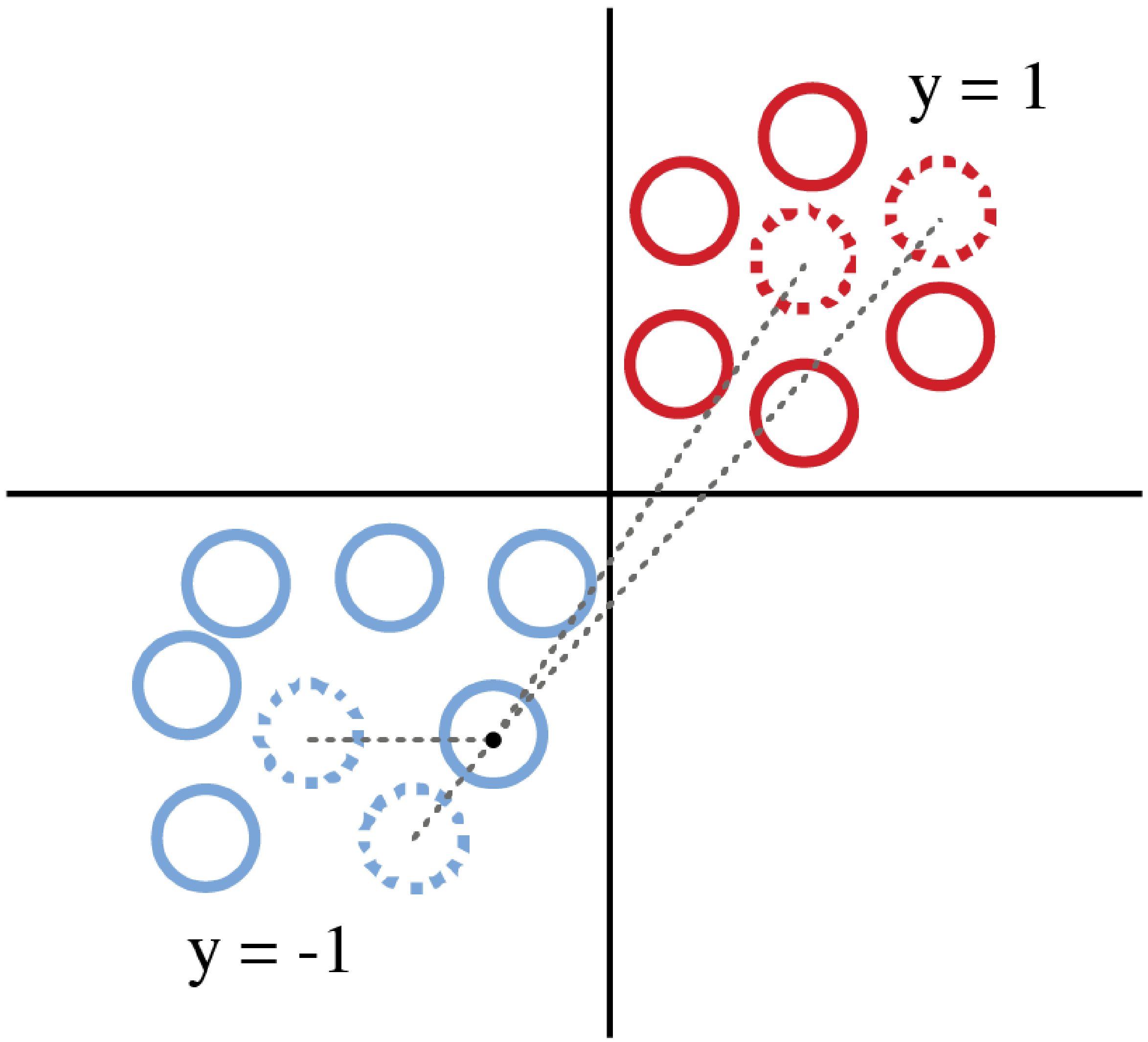
Ideas of Support Vectors and Kernels

Support Vectors: a subset of the training examples $\mathbf{x}$

Kernel: a similarity function $K(x^i, x^j)$ that measures how much sample x^i is similar to sample x_j

SVM cares about the similarity between a given sample to the support vectors.

Support Vector and Kernels



Support vectors are often a subset of the training set.

Kernel defines the similarity between a given sample to all support vectors.

What does kernel look like?

Linear kernel: this is just the dot product between the two vectors.

$$K(x', x) = x' \cdot x$$

Nonlinear kernel:

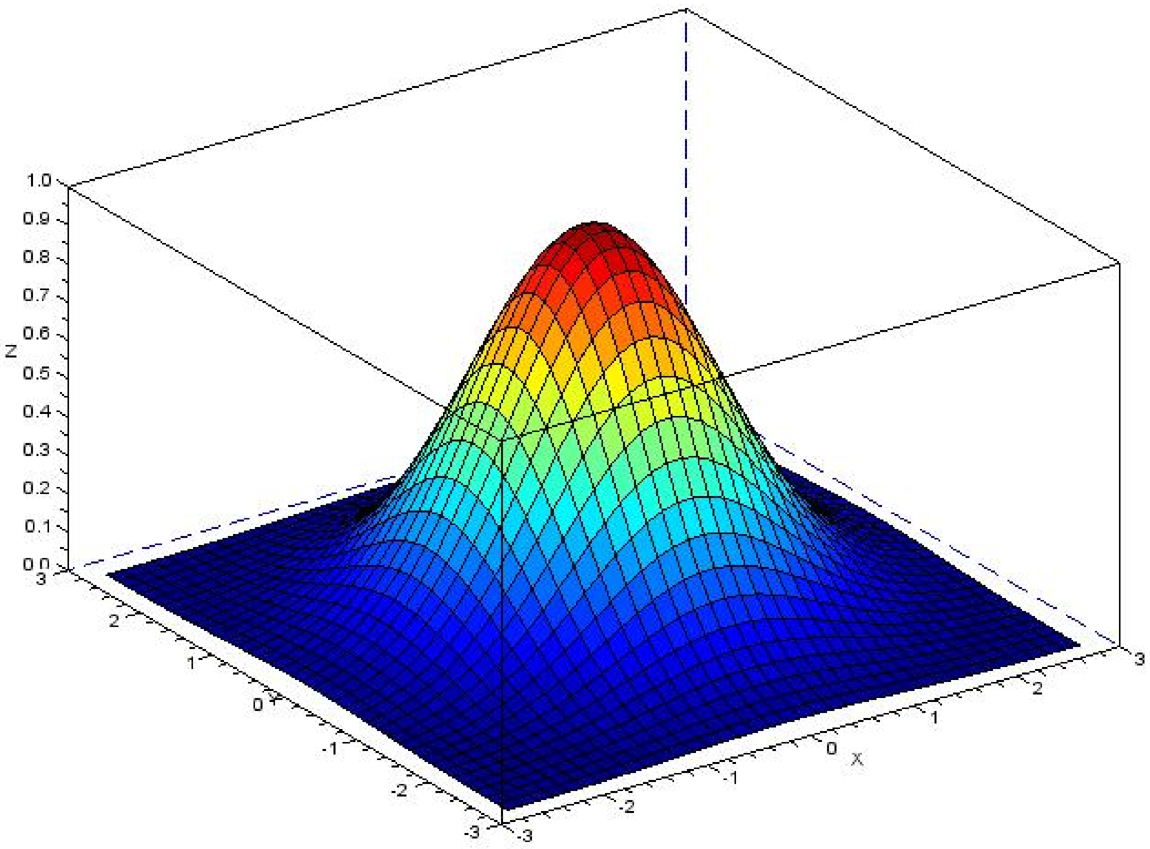
$$K(x', x) = \phi(x') \cdot \phi(x)$$

The common choices of K are:

Polynomial: $K(x', x) = (x' \cdot x)^d$

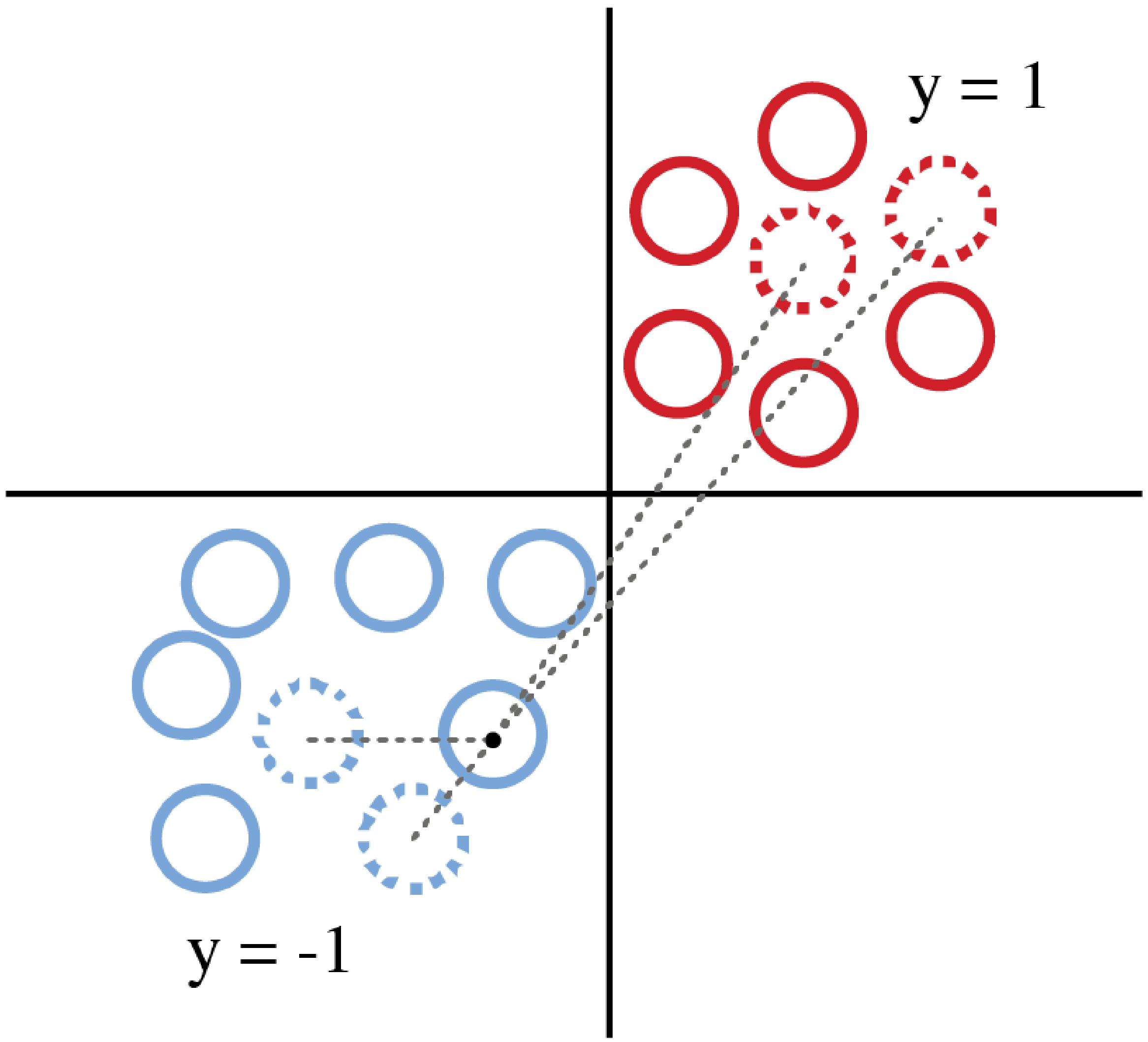
Gaussian: $K(x', x) = exp(-\frac{||x - x'||}{2\sigma})$

Understanding Kernels a Bit More



This is a gaussian kernel. If x^i and x^j are close to each other, $K(x^i, x^j)$ will be high.

SVM Also Create a New Set of Nonlinear Features



$$f_1^i = K(x^1, x^i)$$

$$f_2^i = K(x^2, x^i)$$

...

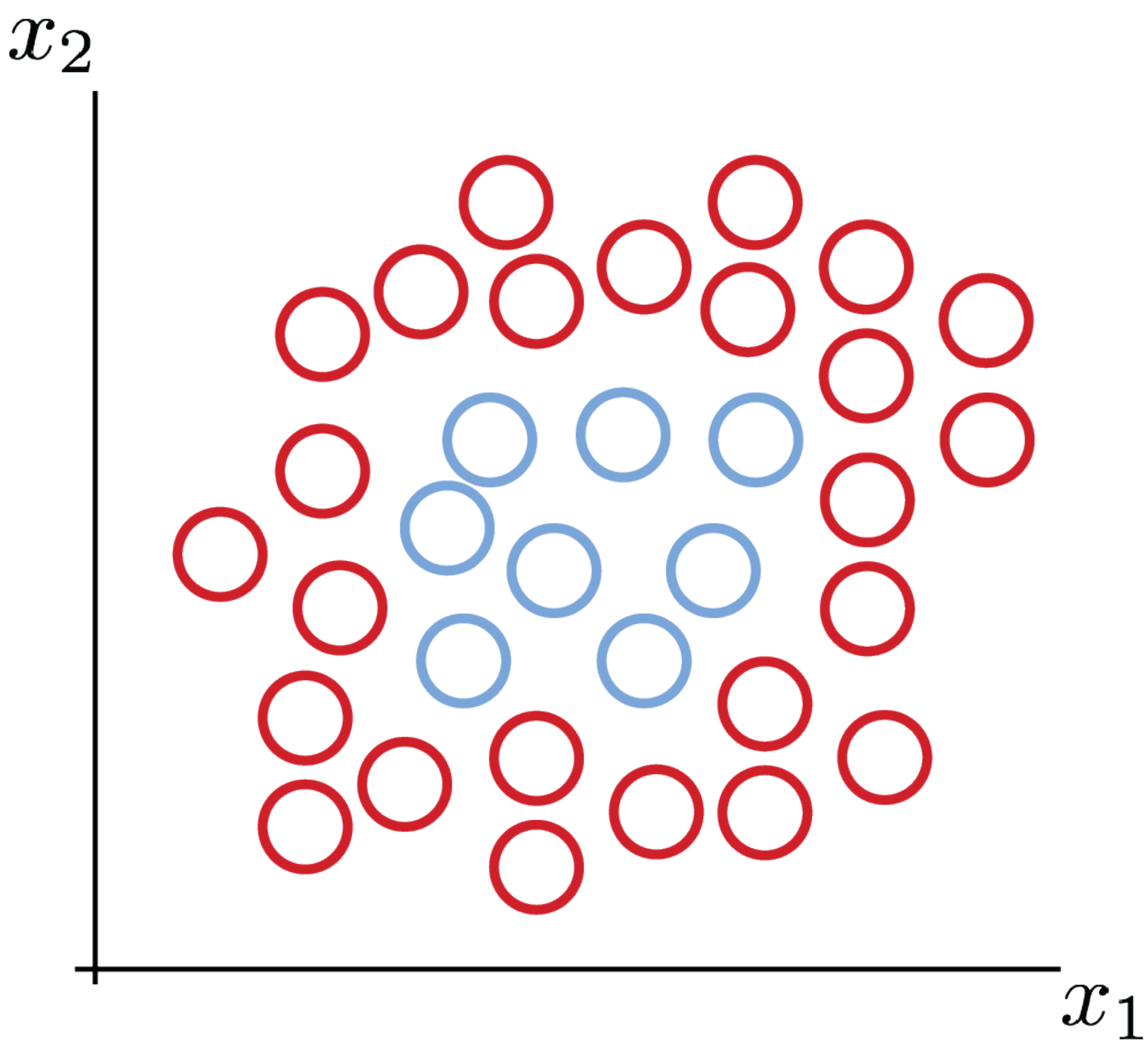
Features of a given sample are defined by the similarity between the support vectors and that sample.

Nonlinear Kernels Create Nonlinear Features

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$\begin{aligned}
(u \cdot v)^2 &= (u_1 v_1 + u_2 v_2)^2 \\
&= u_1^2 v_1^2 + 2 u_1 v_1 u_2 v_2 + u_2^2 v_2^2 \\
&= (u_1^2, u_2^2, \sqrt{2} u_1 u_2) \cdot (v_1^2, v_2^2, \sqrt{2} v_1 v_2) \\
&= \phi(u) \cdot \phi(v)
\end{aligned}$$

Nonlinear features allow nonlinear classification.



$$h(x) = \theta_0 + \theta_1 x_1^2 + \theta_2 x_2^2 + \ldots$$
$$\theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 \ldots$$

$$f_1 = x_1^2,\ f_2 = x_2^2$$
$$f_3 = x_1 x_2,\ f_4 = x_1^2 x_2,\ \ldots$$

$$h(x) = g(\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \ldots)$$

SUPPORT VECTOR MACHINE

SVM is difficult to understand. Most people use it as a black box tool. But it is useful to have an idea about how the algorithm works.

To understand SVM you need to understand the following:

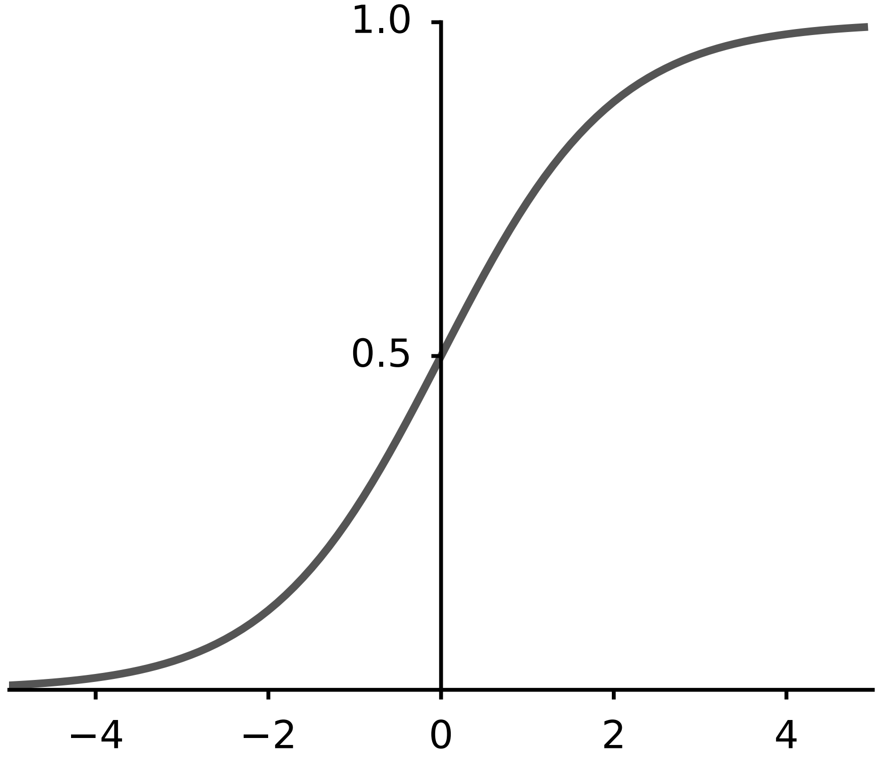
- Ideas of support vector and kernel
- **Large margin classification**

Large Margin Classification

Let's start with a logistic regression idea. Remember in logistic regression we had:

$$h(x) = g(z)$$

$$h(x^i) = \frac{1}{1+exp(-\theta^\intercal x^i)}$$



if $y = 1$, $h(x) \approx 1$, $z >> 0$
if $y = 0$, $h(x) \approx 0$, $z << 0$

$$z = \theta^\intercal x$$

SVM is Large Margin Classification

SVM approaches the classification problem differently from logistic regression in many ways.

First, class labels in SVM, $y \in \{-1, 1\}$ by convention (not $\{0, 1\}$).

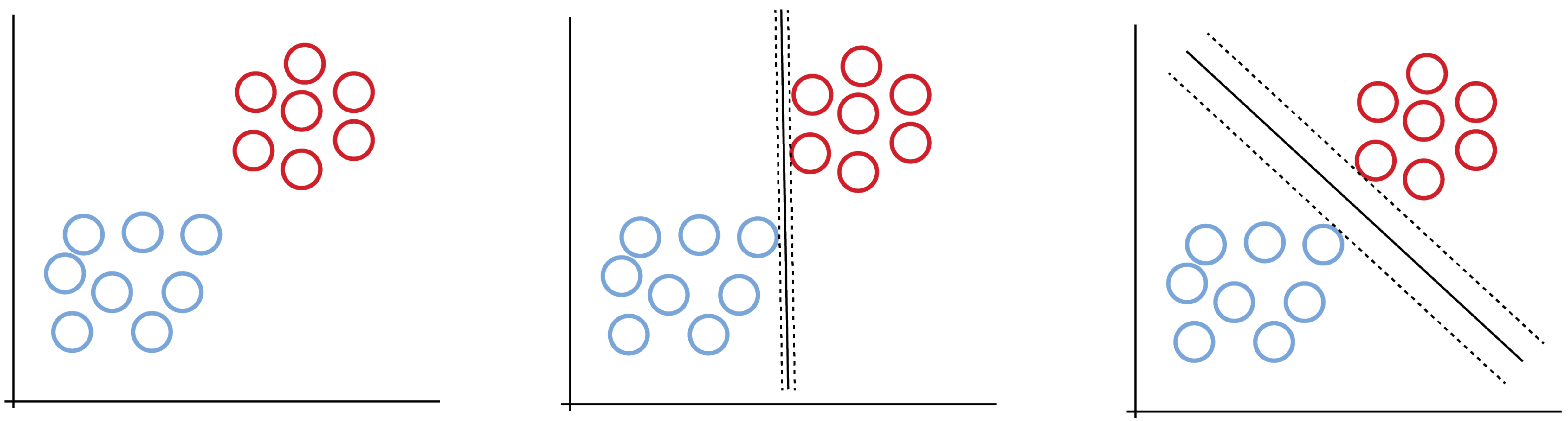
Second, SVM pays attention to margin:

$$\text{if } y = 1,\ h(x) = 1,\ z = \theta^\intercal x >= 1 \text{ (not just } z > 0)$$
$$\text{if } y = -1,\ h(x) = -1,\ z = \theta^\intercal x <= -1 \text{ (not just } z < 0)$$

In fact, it make this margin condition a requirement or a constraint of the algorithm.

Intuition for Large Margin Classification

SVM selects the decision boundary that maximizes the margin.
Here's how it works in linearly separable case.



SVM is Large Margin Classification

The optimization objective of SVM:

$min \frac{1}{2}||\theta||^2$

subject to:

$z = \theta^{\intercal} x >= 1$, if $y = 1$

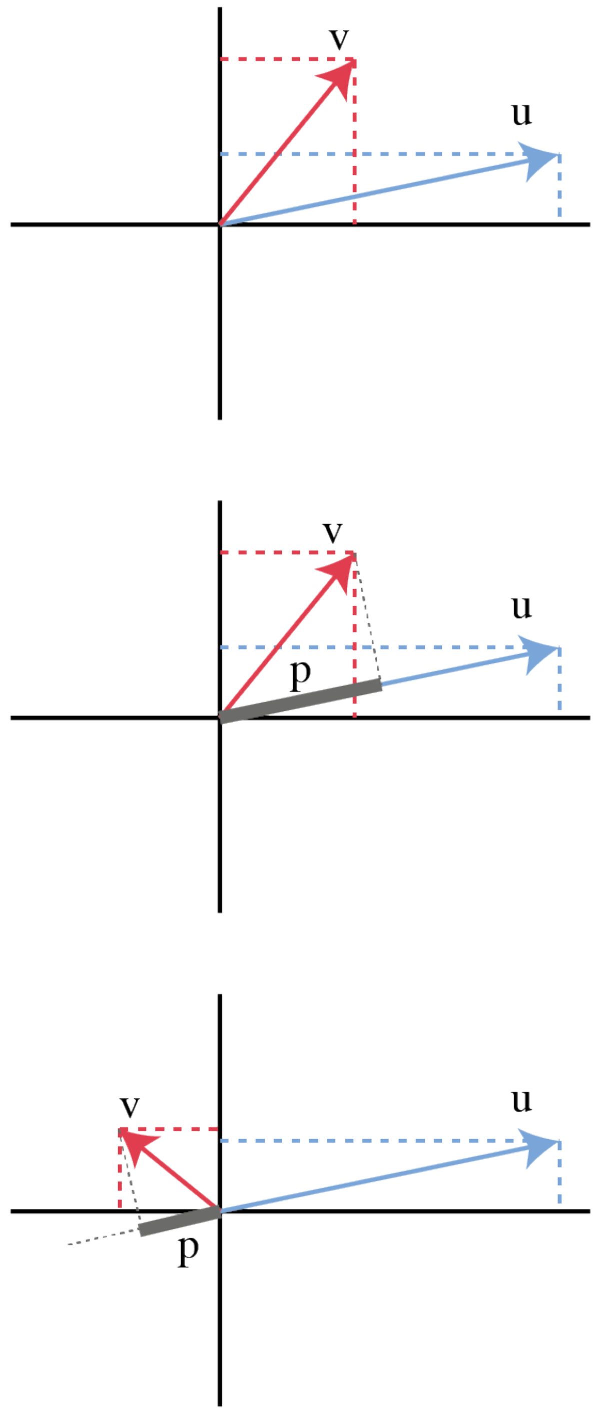
$z = \theta^{\intercal} x <= -1$, if $y = -1$

How does optimizing the objective lead to selecting the right decision boundary?

To understand this you need to know two facts:

1. Facts about vector inner products
2. The fact that θ vector is perpendicular to decision boundary

Vector Inner Product



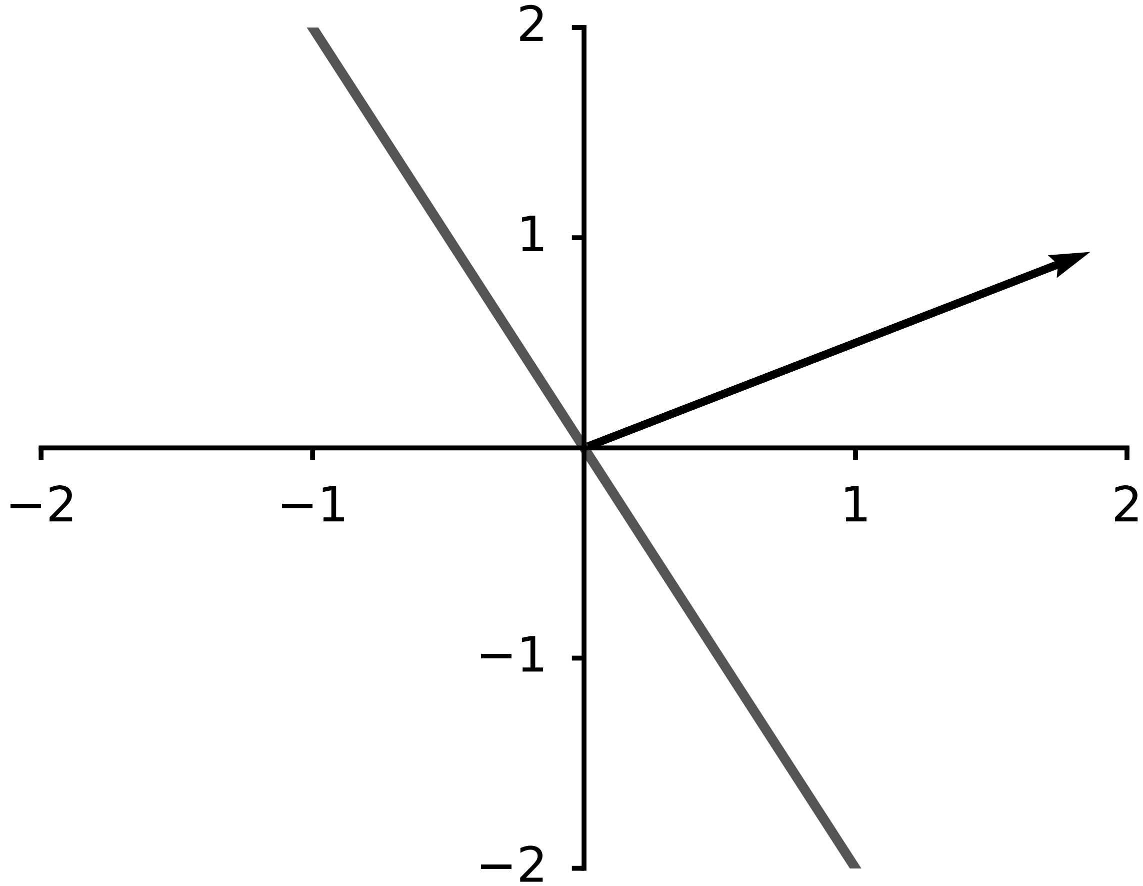
$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$u^\intercal v = u_1 v_1 + u_2 v_2$$

$$u^\intercal v = p \cdot ||u||$$

Note that if the angle between u and v are more than 90° then p will be negative.

The θ Vector is Perpendicular to Decision Boundary



$$h(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$\theta_0 = 0$, $\theta_1 = 2$, $\theta_2 = 1$

decision boundary:
$2x_1 + x_2 = 0$

theta vector: $[2, 1]$

Selecting the Right Decision Boundary

$min \frac{1}{2}||\theta||^2$

subject to:

$\theta^\intercal x >= 1$, if $y = 1$

$\theta^\intercal x <= -1$, if $y = -1$

Swap the constraint using the inner product fact:

$\theta^\intercal x = p \cdot ||\theta|| >= 1$, if $y = 1$

$\theta^\intercal x = p \cdot ||\theta|| <= -1$, if $y = -1$

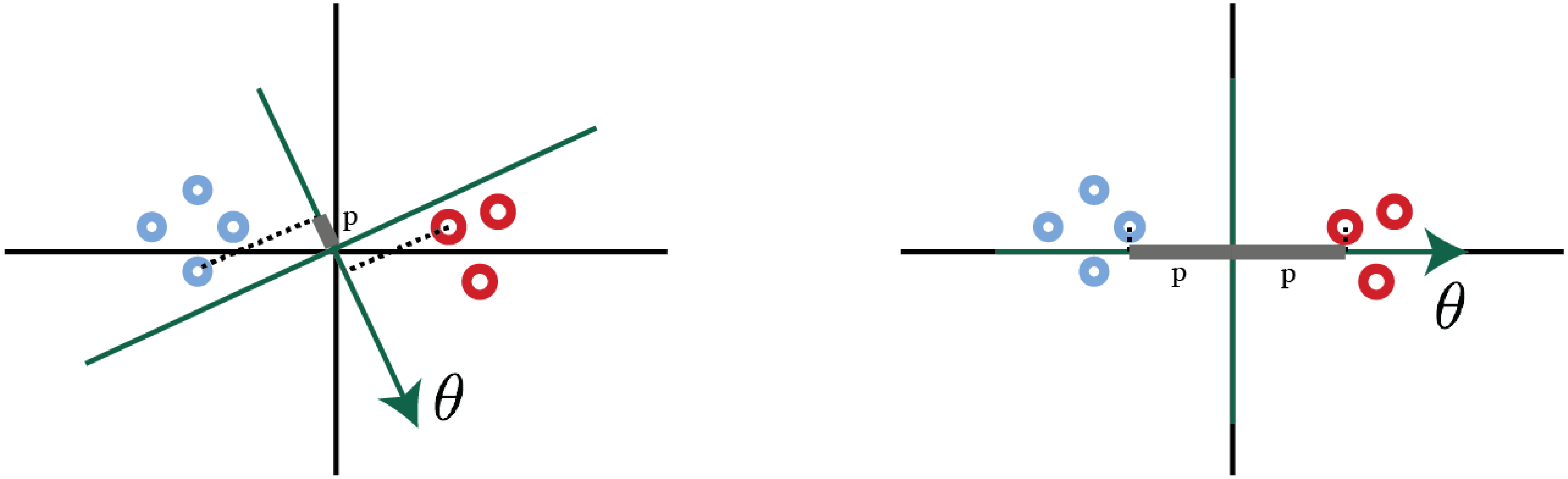
Selecting the Right Decision Boundary

$$min \frac{1}{2} ||\theta||^2$$

subject to:

$p \cdot ||\theta|| >= 1$, if $y = 1$

$p \cdot ||\theta|| <= -1$, if $y = -1$



SUPPORT VECTOR MACHINE SUMMARY

- SVM use support vector and kernel to project data points into (non)linear dimensional space and form a new set of features.
- Choice of nonlinearity is determined by choice of kernels.
- Then SVM uses optimization to classify data points by maximum margin principles, yielding the most effective decision boundary.