# Lecture 4

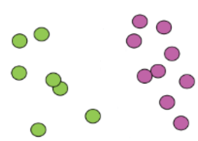
## Clustering and Classification

## Objectives

At the end of the session, you should be able to

1. discuss the principles of support vector machines (SVM); and
2. implement classification by SVM in Python.

## The Case of Binary Predictions



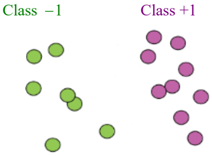
How do we separate the two classes?

Solution: Draw a line that separates the two data classes.

There are infinitely many such lines. How do you choose which one?

## Support Vector Machines (SVM)

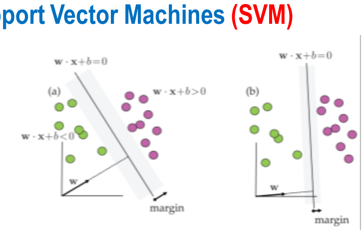
Class $-1$     Class $+1$



Construct a hyperplane $w \cdot x + b = 0$ where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$.

The optimization problem associated with SVM is to not only optimize a decision line which makes the fewest labeling errors for the data, but also optimizes the largest margin between the data.

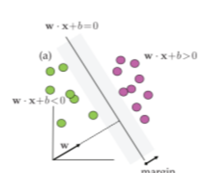
## Support Vector Machines (SVM)




The vectors (data points) closest to the margin are called support vectors.

## Support Vector Machines (SVM)

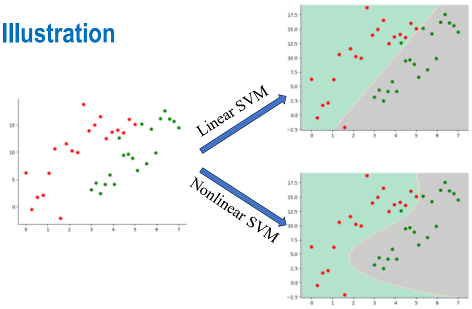
Consider data points $x_j \in \mathbb{R}^n$ with class $y_j \in \{\pm 1\}$.




Given the hyperplane $w \cdot x + b = 0$, the class of a new data point $x_j$ can be determined by whether $x_j$ is to the left or right of the hyperplane.

$$\mathbf{y}_j(\mathbf{w}\cdot\mathbf{x}_j + b) = \text{sign}(\mathbf{w}\cdot\mathbf{x}_j + b) = \begin{cases} +1 & \text{magenta ball} \\ -1 & \text{green ball.} \end{cases}$$

## Illustration




## Nonlinear SVM

Idea: Map the data into a nonlinear, higher-dimensional space.

$$x \to \Phi(x)$$

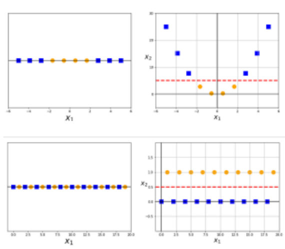
$\Phi(x)$ is now the new observables (features) of the data.

The optimization work now is learning the hyperplane

$$w \cdot \Phi(x) + b = 0$$

## Illustration

Image source: https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f

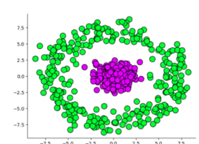


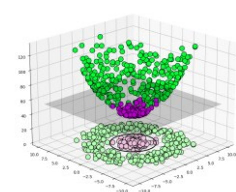
$$x \mapsto (x, x^2)$$

$$x \mapsto (x, x \bmod 2)$$

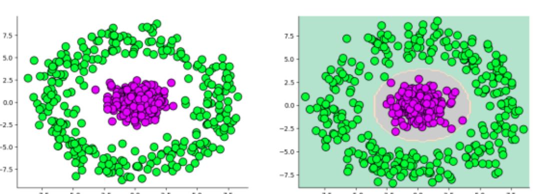
## Illustration

$$(x, y) \mapsto (x, y, x^2 + y^2)$$





## Illustration



## Nonlinear SVM

The optimization work now is learning the hyperplane

$$w \cdot \Phi(x) + b = 0$$

Problem: Nonlinearity induces potentially high computational cost.

Solution: Kernel "trick"

## Common Kernels

Radial Basis Function (RBF): $K\left(x_j, x\right) = \exp\left(-\gamma \left\| x_j - x \right\|^2\right)$

Polynomial kernel: $K\left(x_j, x\right) = \left(x_j \cdot x + 1\right)^N$

Linear Kernel

Sigmoid Kernel

Exponential Kernel

## END