# COMP4388: **MACHINE LEARNING**

Support Vector Machines

Dr. Radi Jarrar
Department of Computer Science
Birzeit University

جامعة بيرزيت
BIRZEIT UNIVERSITY

## SVM

- SVM is a strong discriminative classifier that was originally used for binary classification by Vapnik 1998
- Currently one of the most common ML algorithms that is used to solve multiclass classification problems
- A support vector machine classifier determines which class label is associated with a given N-dimensional feature vector in the feature space

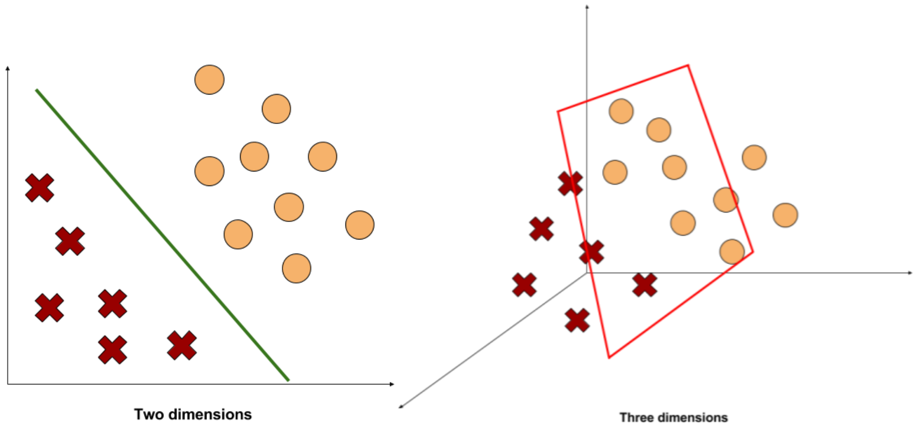
## SVM (2)

- SVM can be seen as a surface that defines a boundary between data points plotted in a multidimensional space
- SVM aims to create a **linear** flat surface that splits data points belonging to different classes
- This flat surface is called a **hyperplane**
- **The hyperplane separates the data into homogeneous groups on each side**

## SVM (3)

- The main advantage of SVM is that it can model highly complex relationships
- It is used in many fields such as image classification, hand-writing recognition, text categorisation, bioinformatics, and many other fields

## SVM (4)

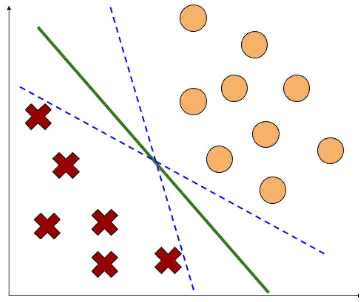


## SVM (5)

- In the simplest case, SVM can solve binary classification problems of classes that can be separated linearly (as in the previous figure)
- It can be also extended to scenarios in which the data points are not linearly separable
- The task of SVM is to find the line that separates the two classes

## SVM (6)

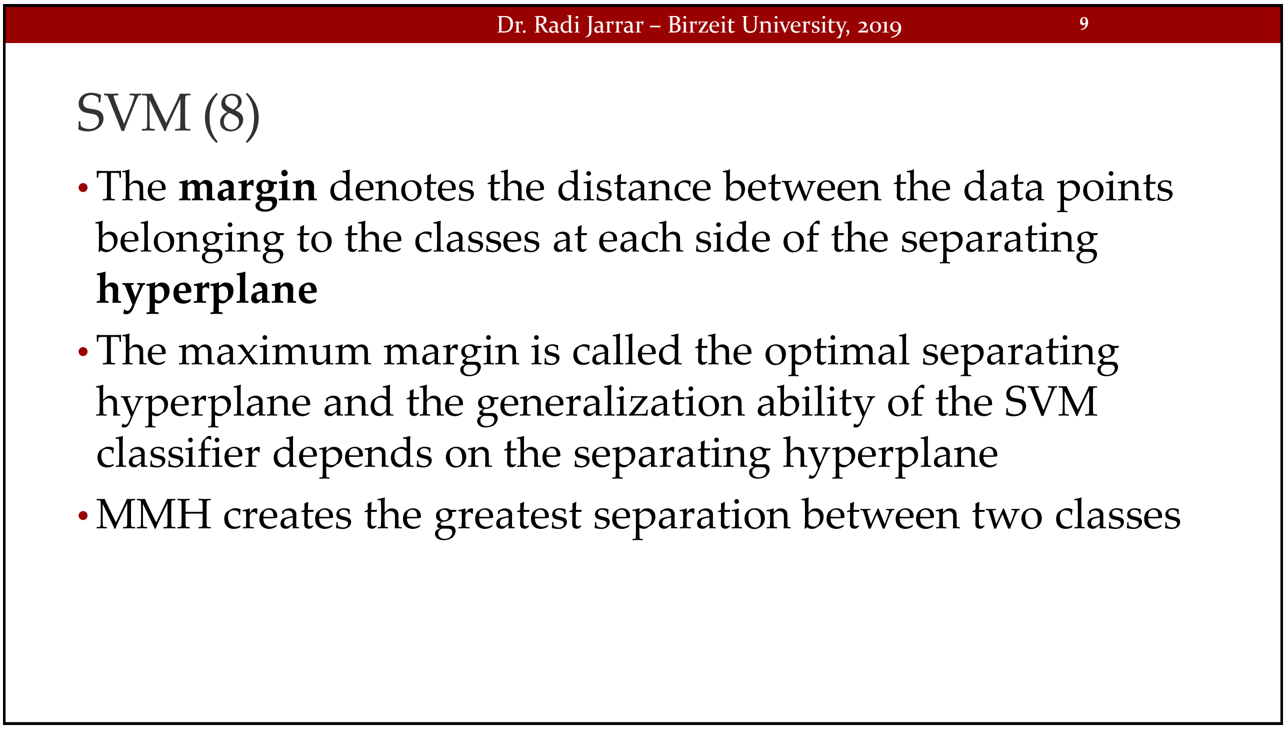
- Simply said, there are many options for the line



- The task of SVM is to identify which is the best line to separate the classes and maximises the margin between data points that fall on both sides of the line

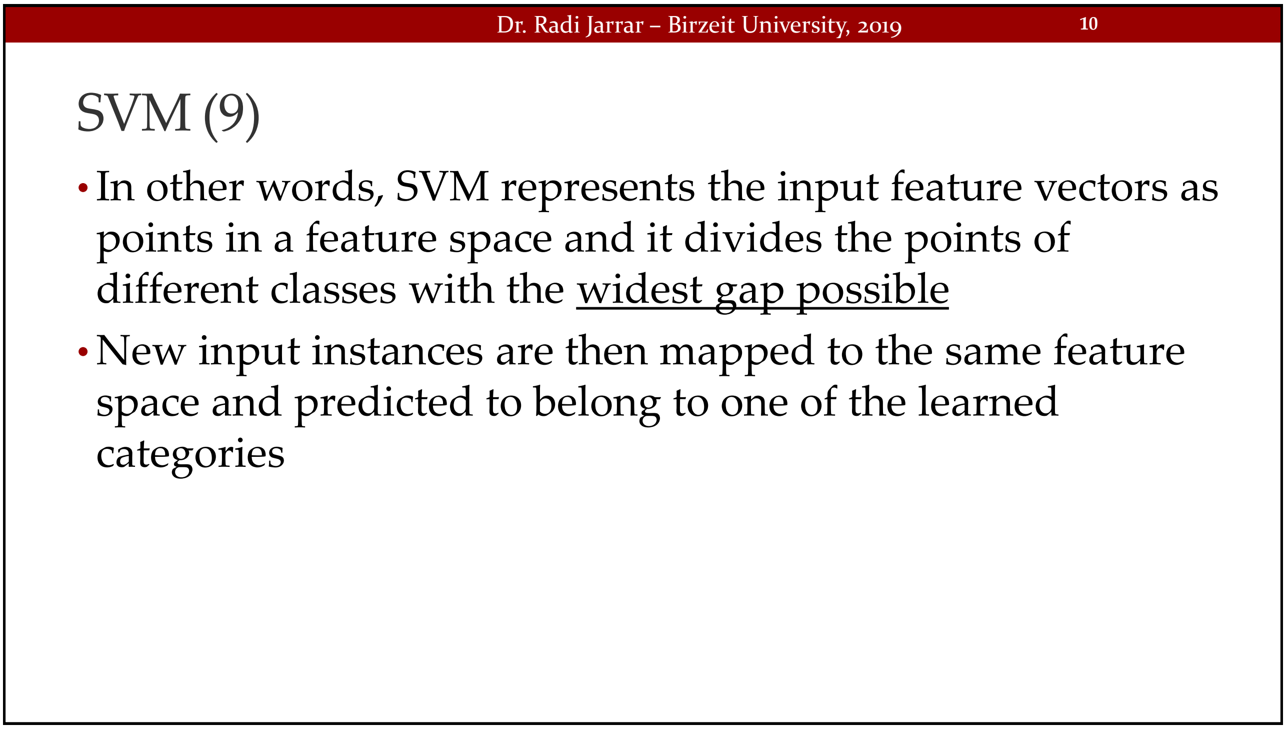
## SVM (7)

- Given a set of training data $D_{train} = \{(f_1, y_1), (f_2, y_2), \ldots, (f_i, y_i)\}$ where $f_i$ in $R^n$ is an n-dimensional input feature vector and $y_i$ in {-1, 1} is the associated class label with each input feature vector
- The support vector machine classifier constructs a separation hyperplane as a separation surface that maximizes the margin between the positive and negative examples



## SVM (8)

- The **margin** denotes the distance between the data points belonging to the classes at each side of the separating **hyperplane**
- The maximum margin is called the optimal separating hyperplane and the generalization ability of the SVM classifier depends on the separating hyperplane
- MMH creates the greatest separation between two classes



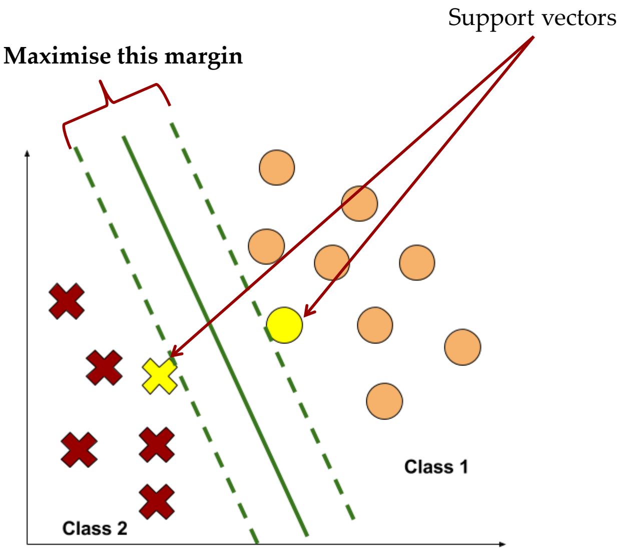
## SVM (9)

- In other words, SVM represents the input feature vectors as points in a feature space and it divides the points of different classes with the widest gap possible
- New input instances are then mapped to the same feature space and predicted to belong to one of the learned categories

# SVM (10)

- Support vectors are the data points that are the closest to the maximum margin
- Each class must have at least one support vector
- The support vectors alone can define the maximum margin hyperplane which is an important feature for SVM. How?

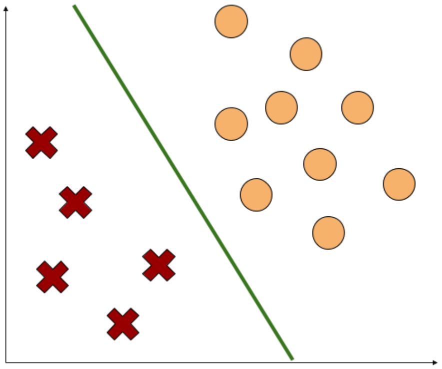
# SVM (11)



## Maximum Margin Hyperplane

- The maximum margin can be found through the convex hull concept
- The maximum margin is as far away possible from the outer boundaries from the two groups of data points (convex hull)
- The maximum margin heperplane is the perpendicular bisector of the shortest line between the two convex hulls
- This is achieved using quadratic programming (optimisation)

## Maximum Margin Hyperplane (2)



Two dimensions

## Maximum Margin Hyperplane (3)

- Another method involves searching through the space to find all possible hyperplanes in order to find a set of two parallel planes that divide the points into homogeneous groups and they are themselves as far apart as possible

## Maximum Margin Hyperplane (4)

- In an n-dimensional feature space, the following equation applies

$$\vec{w} \, . \, \vec{x} + b = 0$$

- where $\vec{w}$ stands for a vector of *n* weights and is the *bias* parameter
- This, in particular, is very similar to the equation to specify a line in 2D space:

$$y = (mx + b)$$

## Maximum Margin Hyperplane (5)

- That formula is used to find a set of weights that specify hyperplanes as follows

$$\vec{w}.\vec{x} - b \geq +1$$

$$\vec{w}.\vec{x} - b \leq -1$$

- The hyperplanes are specified such that all data points of one class fall above the first hyperplane and all the data points of the second class fall under the second hyperplane (possible as the data is linearly separable)

## Maximum Margin Hyperplane (6)

- The distance between two planes is defined as

$$\frac{2}{\left\|\vec{w}\right\|}$$

- where $\left\|\vec{w}\right\|$ is the Euclidean Norm (the distance from the origin to vector w)
- Minimising $\left\|\vec{w}\right\|$ results in maximising the distance between two planes

# Maximum Margin Hyperplane (7)

- This can then be expressed as an optimisation problem as

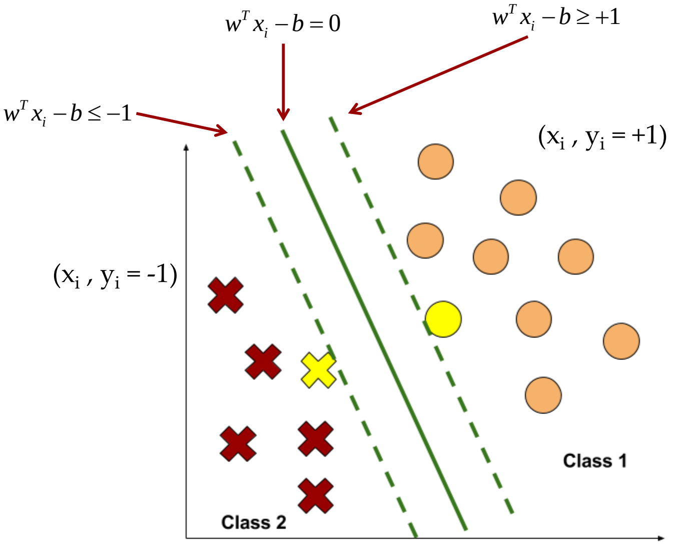
$$\min \frac{1}{2}\|\vec{w}\|^2 \quad subject\ to\ y_i(\vec{w}\ .\ \vec{x}_i - b) \geq 1, \forall\, \vec{x}_i$$

- which can be written as

$$\min \frac{1}{2} w^T x \quad subject\ to\ y_i(w^T x_i - b) \geq 1, \forall\, \vec{x}_i$$

where $(w^T x_i) - b > 0$ if $y_i=1$
$(w^T x_i) - b < 0$ if $y_i=-1$

# Maximum Margin Hyperplane (8)



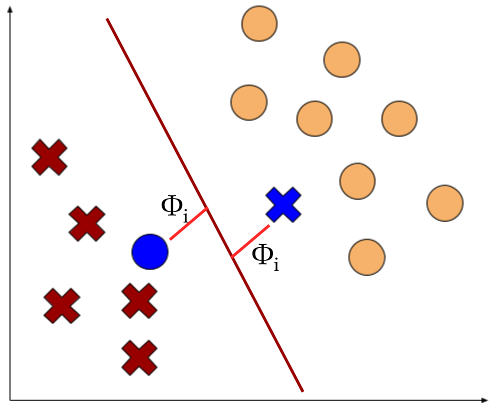
## Maximum Margin Hyperplane (9)

- For a non-linear classification problems, the support vector machine projects the original training data into a higher dimensional feature space using kernel functions to find the maximum-margin hyperplane
- Projection kernels are used and support vector machines are then based on the resolution of the following optimization problem

## Maximum Margin Hyperplane (10)

- A ***slack*** variable is added in the case of non-linearly separable data
- This results in creating a soft-margin that allows some data points to fall on the incorrect side of the margin

# Maximum Margin Hyperplane (11)



# Maximum Margin Hyperplane (12)

- A cost value is added to all data points that violate the constraints
- The algorithm is then trying to minimise the total cost as follows

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\varphi_i \mathit{subject\ to}\ y_i(w^T x_i - b) \geq 1-\varphi_i, \forall \vec{x}_i, \varphi_i \geq 0$$

# The Kernel Trick

- Another method used in SVM to solve non-linearity between classes in SVM is by using the ***kernel trick***
- This would map a non-linear relationship into a linear one

# The Kernel Trick (2)

- Non-linear kernels add more dimensions to the data in an attempt to create separation between data points
- Indeed, the kernel trick involves adding new features that express some mathematical relationship between measured characteristics
- The kernel function has this general form:

$$K(x_i, x_j) = y(x_i)^T . y(x_j)$$

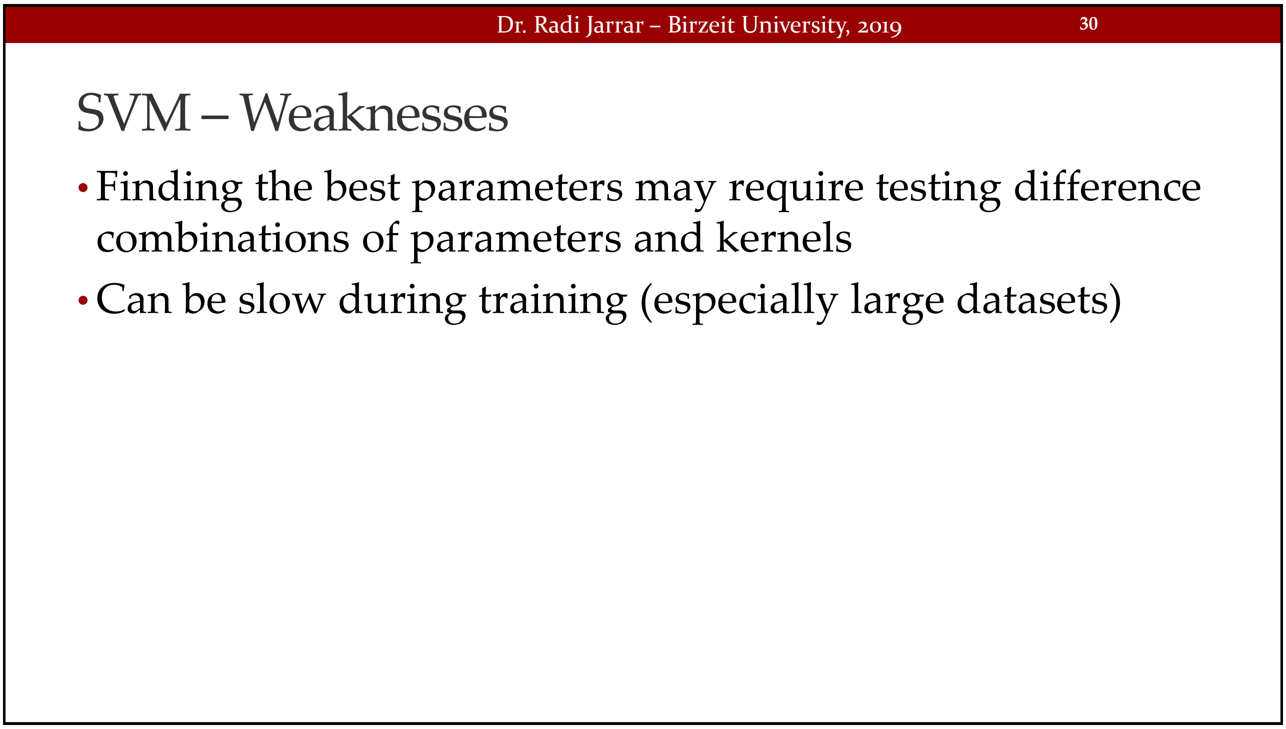
- where $y(x_i)$ is the mapping function mapping data into another space

# The Kernel Trick (3)

- There is a number of kernel function:
  - Linear function: $K(x_i, x_j) = x_i^T x_j$
  - Polynomial function: $K(x_i, x_j) = (x_i^T x_j + 1)^d$
    - a polynomial function of degree d
  - Sigmoid function: $K(x_i, x_j) = \tanh(k x_i^T x_j - d)$
    - where kappa and delta are kernel parameters
  - RBF:
    - Similar to RBF neural network $K(x_i, x_j) = exp\left(\frac{-\left\|x_i - x_j\right\|^2}{2\sigma^2}\right)$

# SVM – Strengths

- Can be used for classification and regression problems
- Not overly influenced by noise and outliers
- Highly accurate in means of performace
- Maybe easier than ANN



## SVM – Weaknesses

- Finding the best parameters may require testing difference combinations of parameters and kernels
- Can be slow during training (especially large datasets)