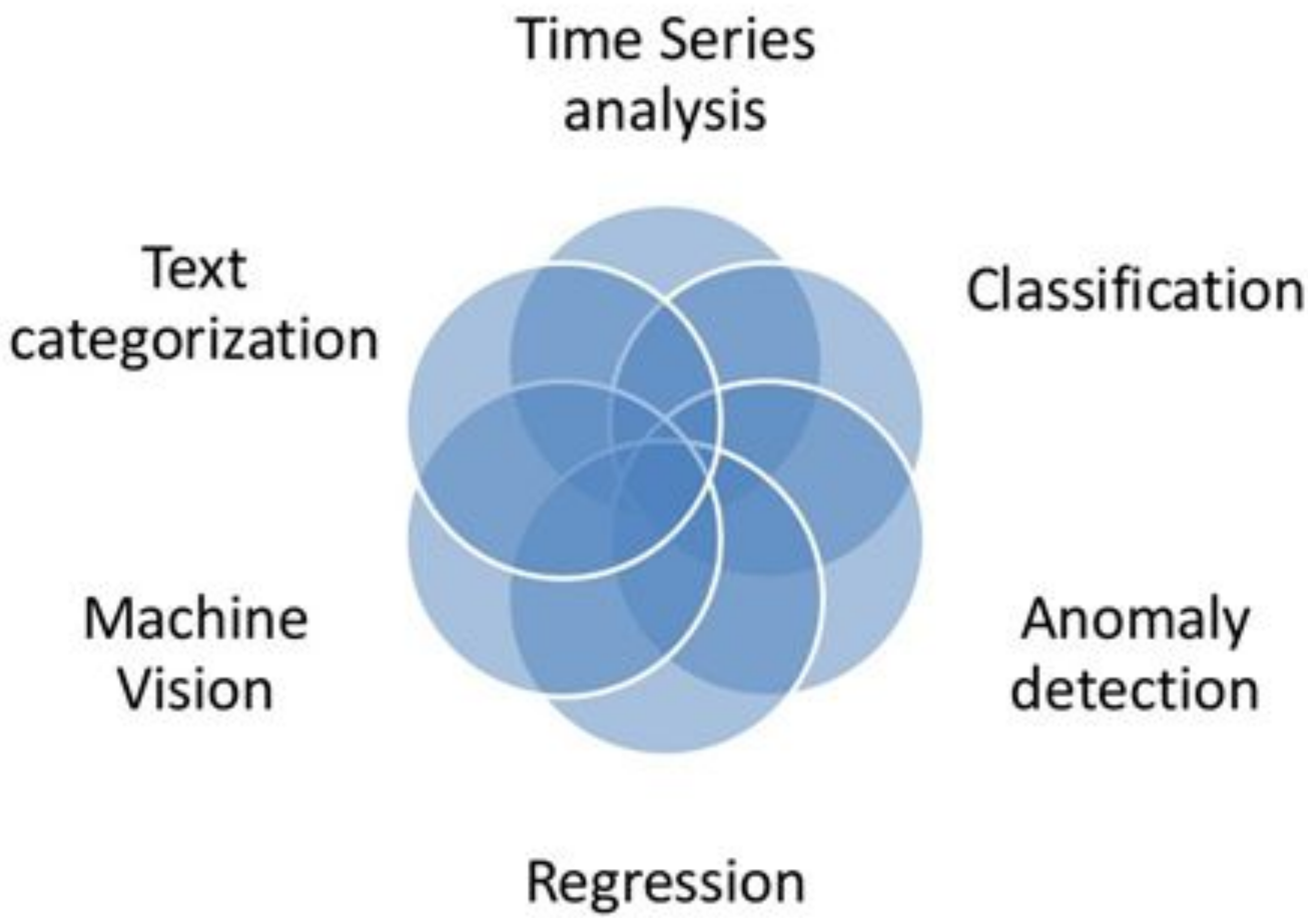
SVM

SUPPORT VECTOR MACHINE

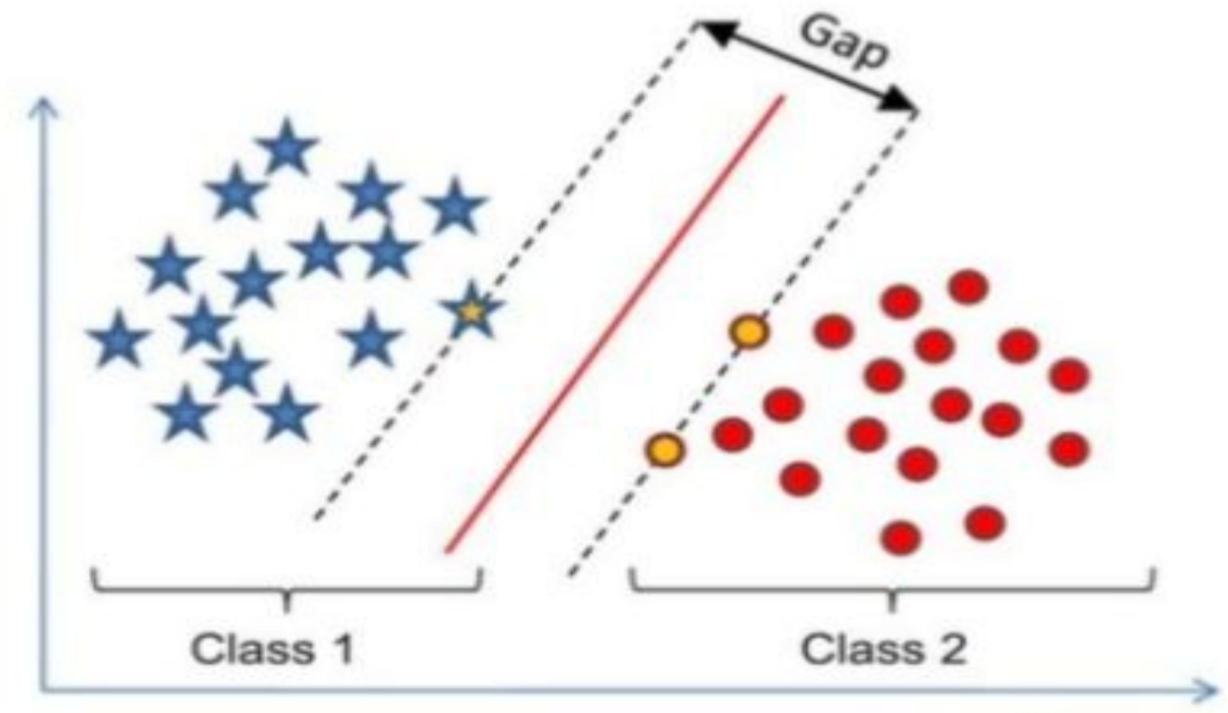
SVM: Support Vector Machine

- In Machine Learning, Support Vector Machines are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis.
- Properties of SVM :
  - Duality
  - Sparseness
  - Kernels
  - Margin
  - Convexity

SVM Applications



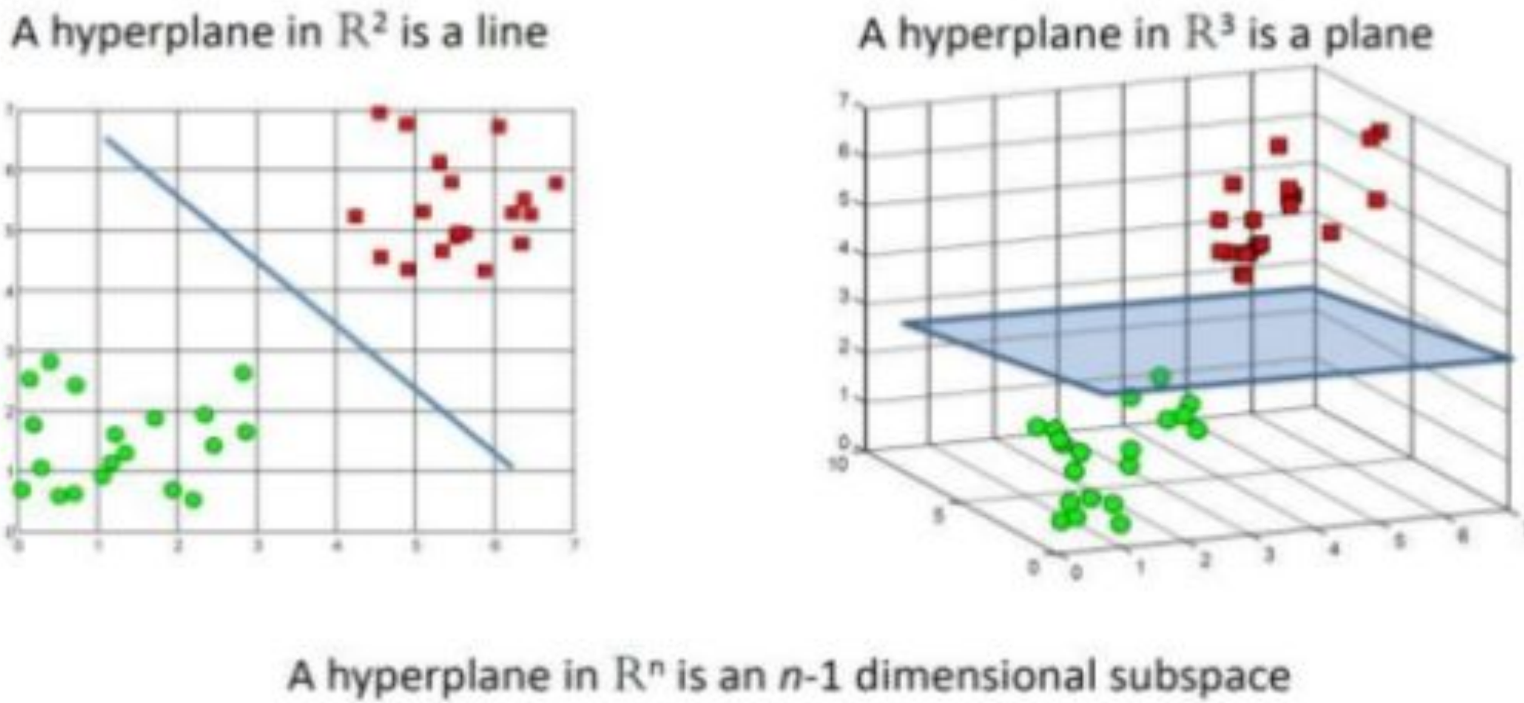
Basic Concept of SVM



Find a linear decision surface ("hyperplane") that can separate classes and has the largest distance (i.e., largest "gap" or "margin") between border-line patients (i.e., "support vectors")

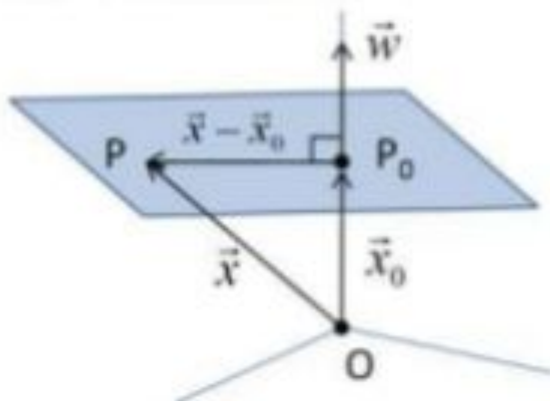
Hyperplane as a Decision Boundary

- A **hyperplane** is a linear decision surface that splits the space into two parts;
- A hyperplane is a binary classifier.



Equation of a Hyperplane

Consider the case of $\mathbb{R}^3$:



An equation of a hyperplane is defined by a point (P_0) and a perpendicular vector to the plane ($\vec{w}$) at that point.

Define vectors: $\vec{x}_0 = \overrightarrow{OP_0}$ and $\vec{x} = \overrightarrow{OP}$, where *P* is an arbitrary point on a hyperplane.

A condition for *P* to be on the plane is that the vector $\vec{x}-\vec{x}_0$ is perpendicular to $\vec{w}$:

$\vec{w}\cdot(\vec{x}-\vec{x}_0)=0$ or

$\vec{w}\cdot\vec{x}-\vec{w}\cdot\vec{x}_0=0$ define $b=-\vec{w}\cdot\vec{x}_0$

$$\boxed{\vec{w}\cdot\vec{x}+b=0}$$

The above equations also hold for $\mathbb{R}^n$ when n>3.

Understanding the basics

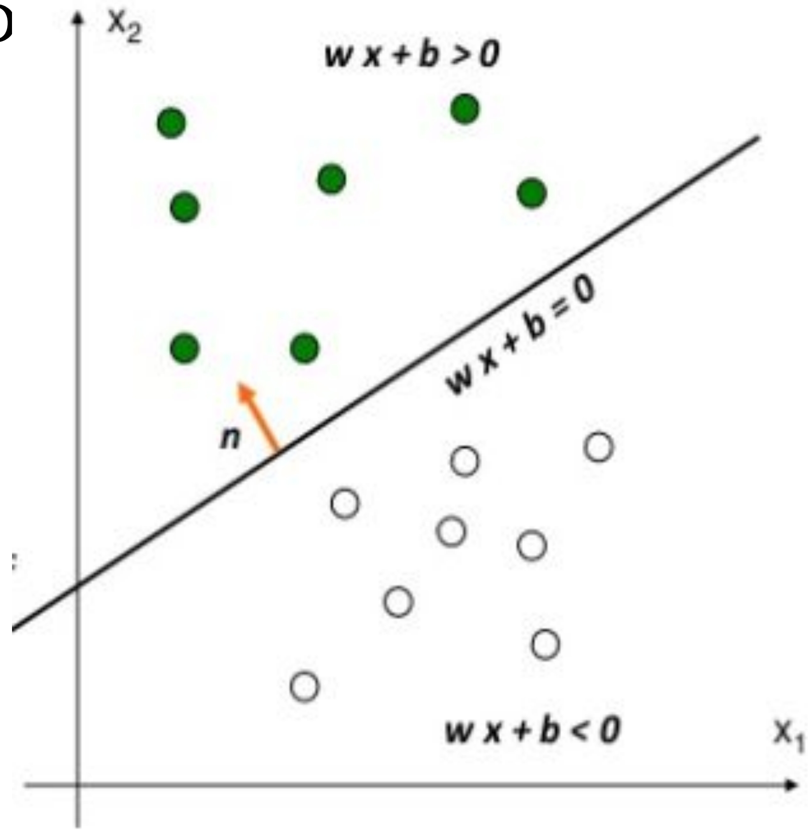
- g(x) is a linear functio

$$g(x) = W X + b$$

- A hyperplane in the feature space.

- (Unit-length) normal of the hyperplane:

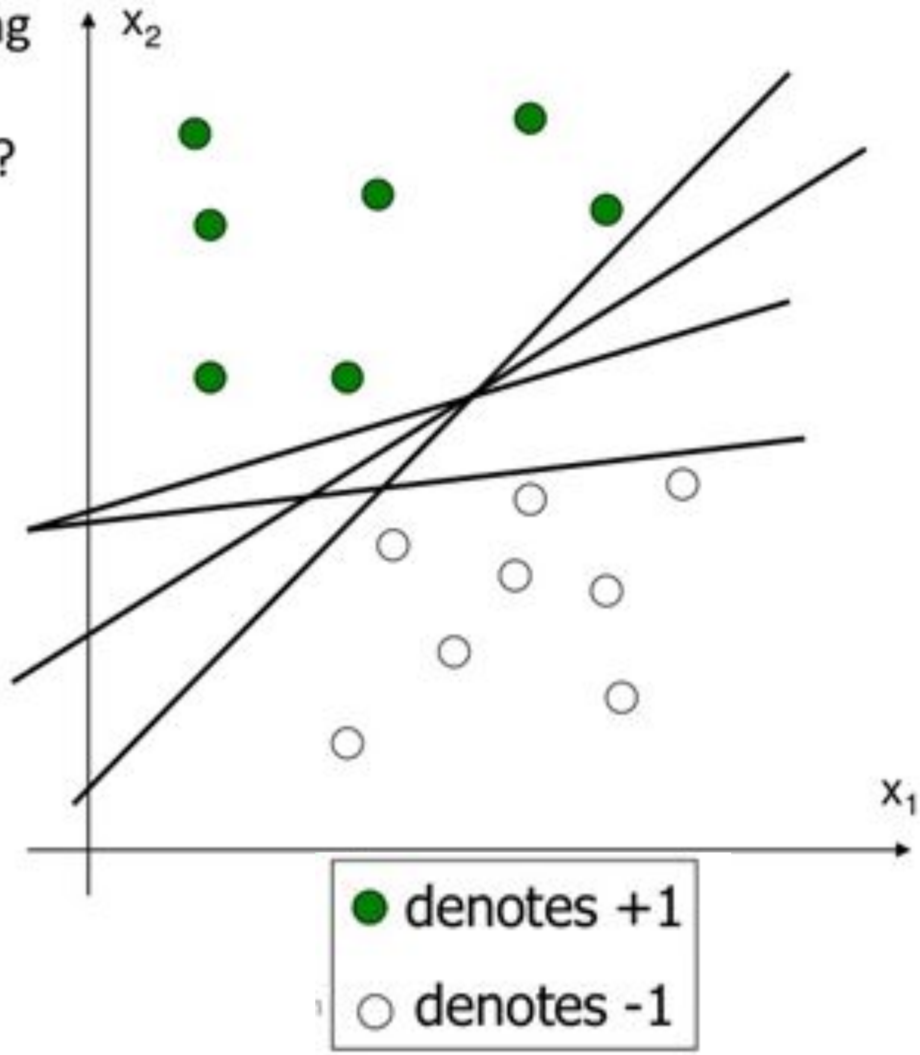
$$\mathbf{n} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$$



Understanding the basics

How to classify these points using a linear discriminant function in order to minimize the error rate?

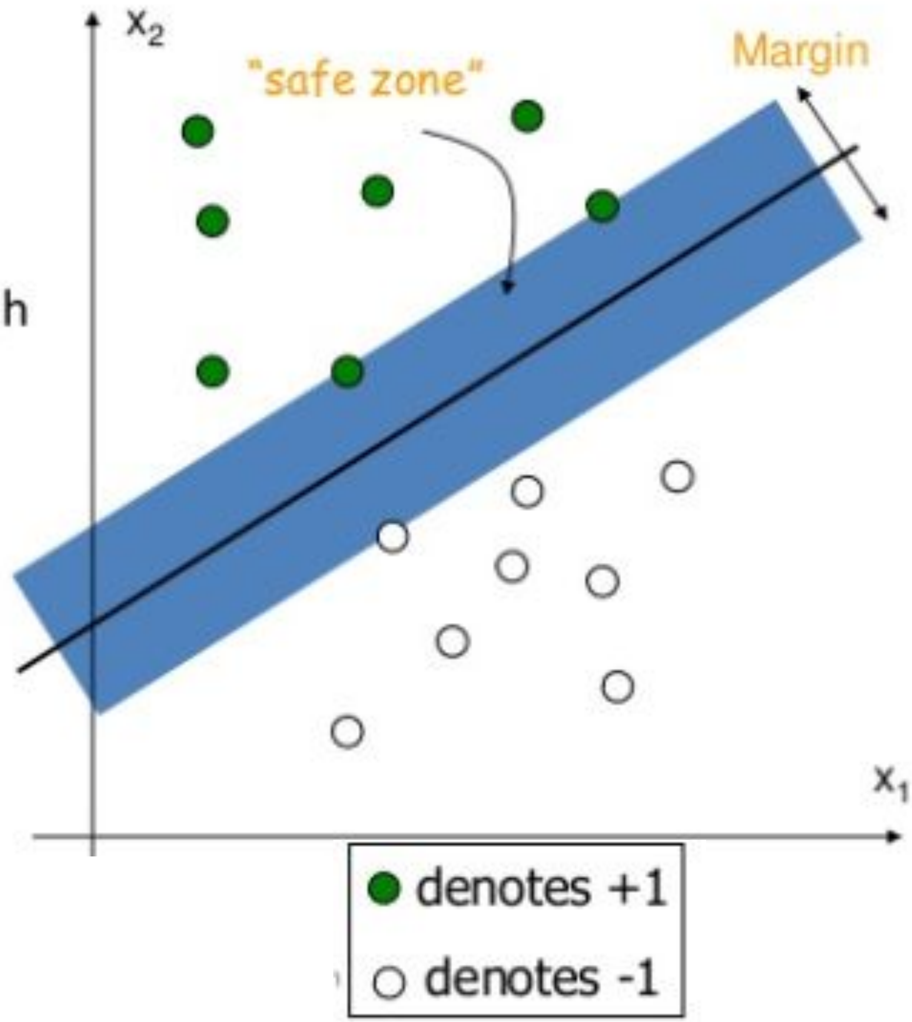
- Infinite number of answers!
- Which one is the best?



Understanding the basics

- The linear discriminant function (classifier) with the maximum margin is the best
- Margin is defined as the width that the boundary could be increased by before hitting a data point
- Why it is the best?

  Robust to outliners and thus strong generalization ability



Understanding the basics

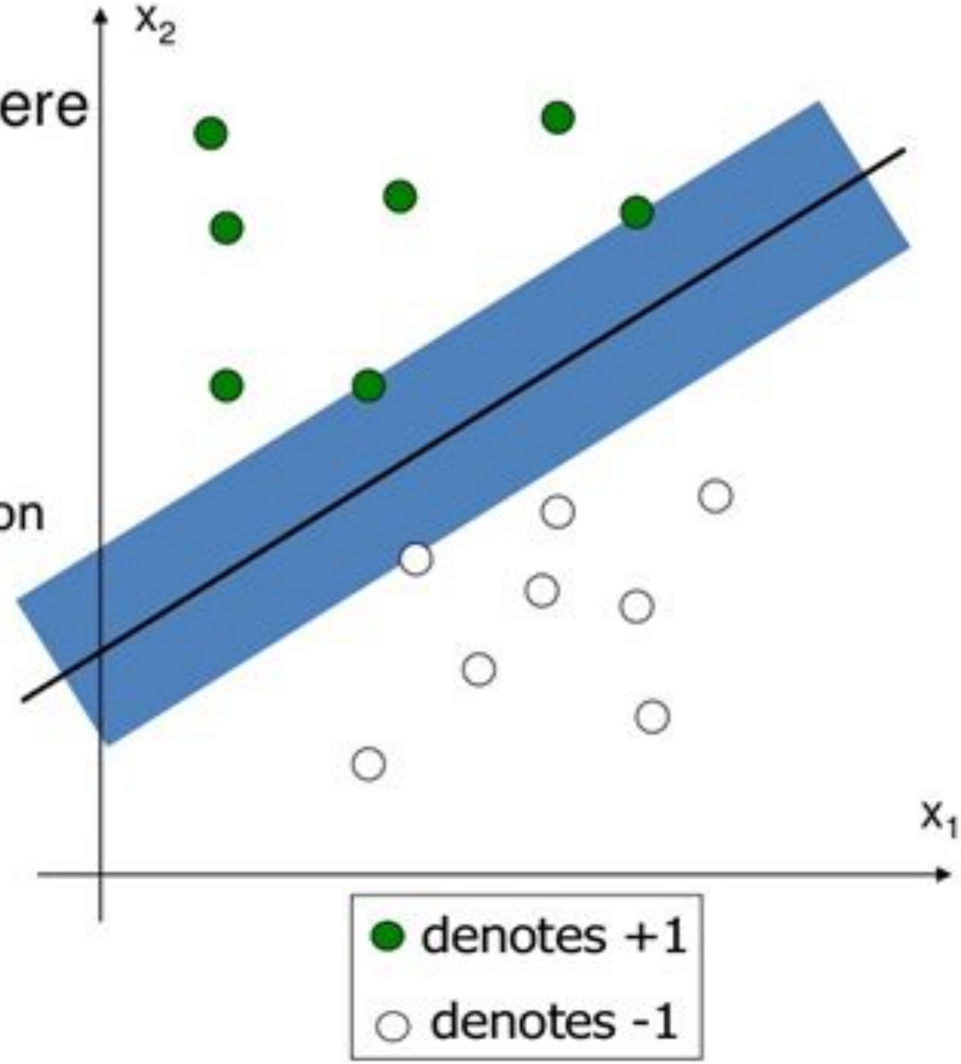
- Given a set of data points:

$\{(\mathbf{x}_i, y_i)\},\ i = 1, 2, \cdots, n,$ where

$$\boldsymbol{For\ y_i = +1, W\,X_i + b > 0}$$
$$\boldsymbol{For\ y_i = -1, W\,X_i + b < 0}$$

- With a scale transformation on both *w* and *b*, the above is equivalent to

$$\boldsymbol{For\ y_i = +1, W\,X_i + b > +1}$$
$$\boldsymbol{For\ y_i = -1, W\,X_i + b < -1}$$



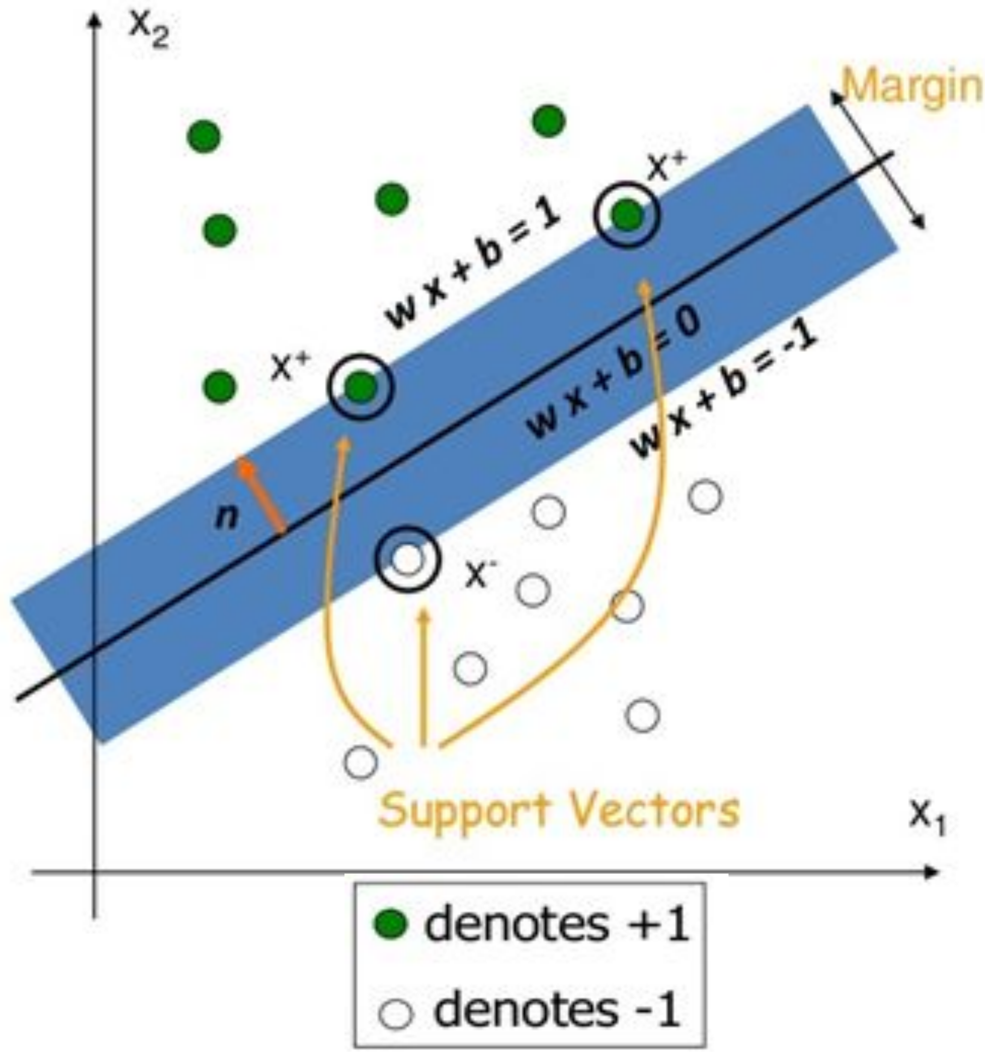
Understanding the basics

- We know that

$$W\,X^{+} + b = +1$$
$$W\,X^{-} + b = -1$$

- The margin width is:

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



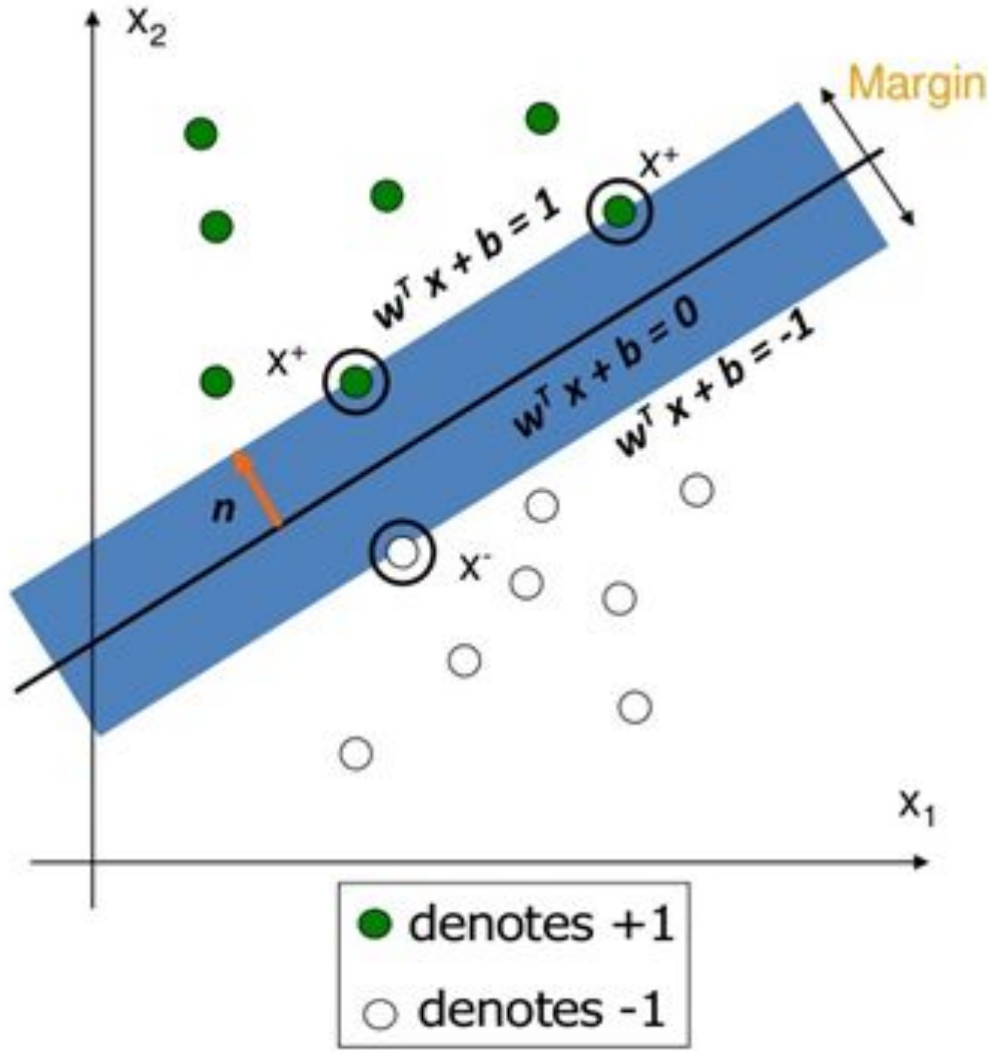
Understanding the basics

- Formulation:

$$\text{maximize} \quad \frac{2}{\|\mathbf{w}\|}$$

such that

$$For\ y_i = +1, W\,X_i + b > +1$$
$$For\ y_i = -1, W\,X_i + b < -1$$



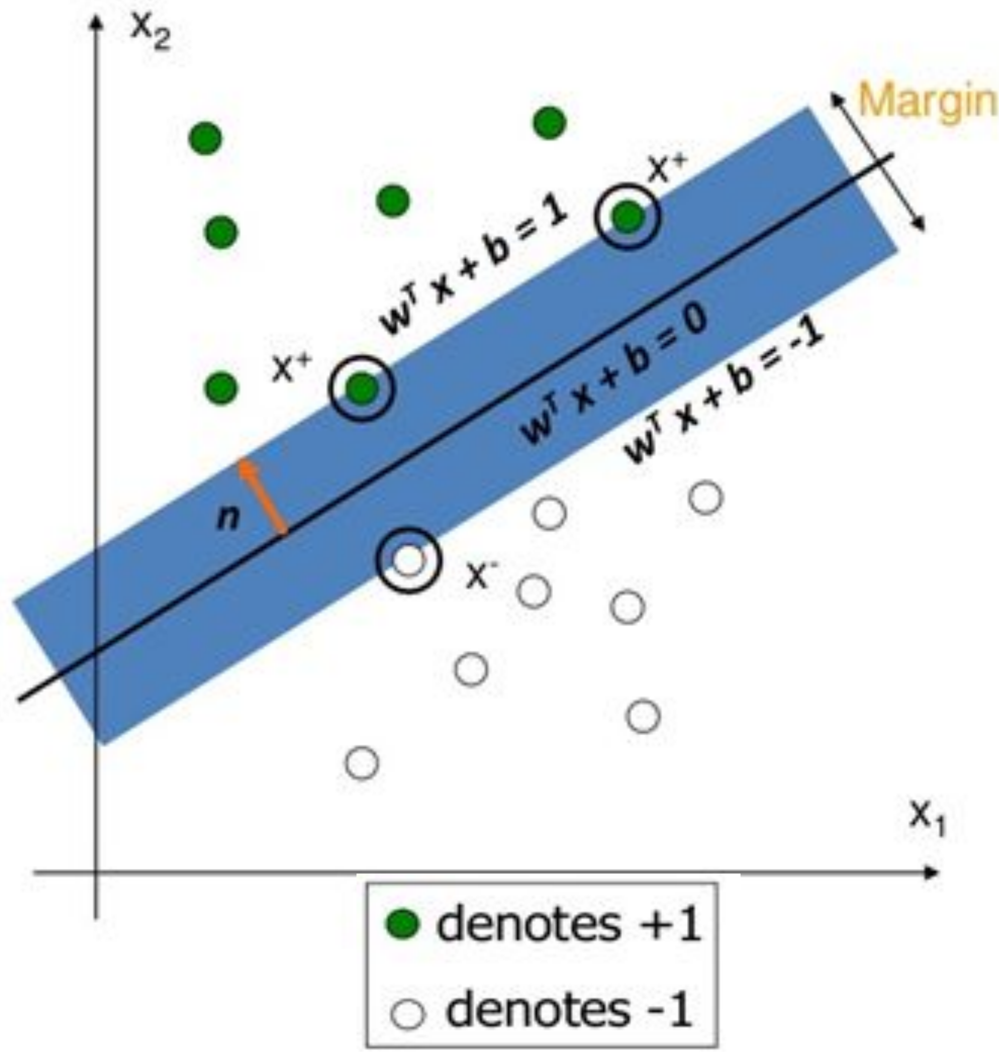
Understanding the basics

- Formulation:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

such that

$$\textbf{\textit{For } } \boldsymbol{y_i = +1, W\,X_i + b > +1}$$
$$\textbf{\textit{For } } \boldsymbol{y_i = -1, W\,X_i + b < -1}$$



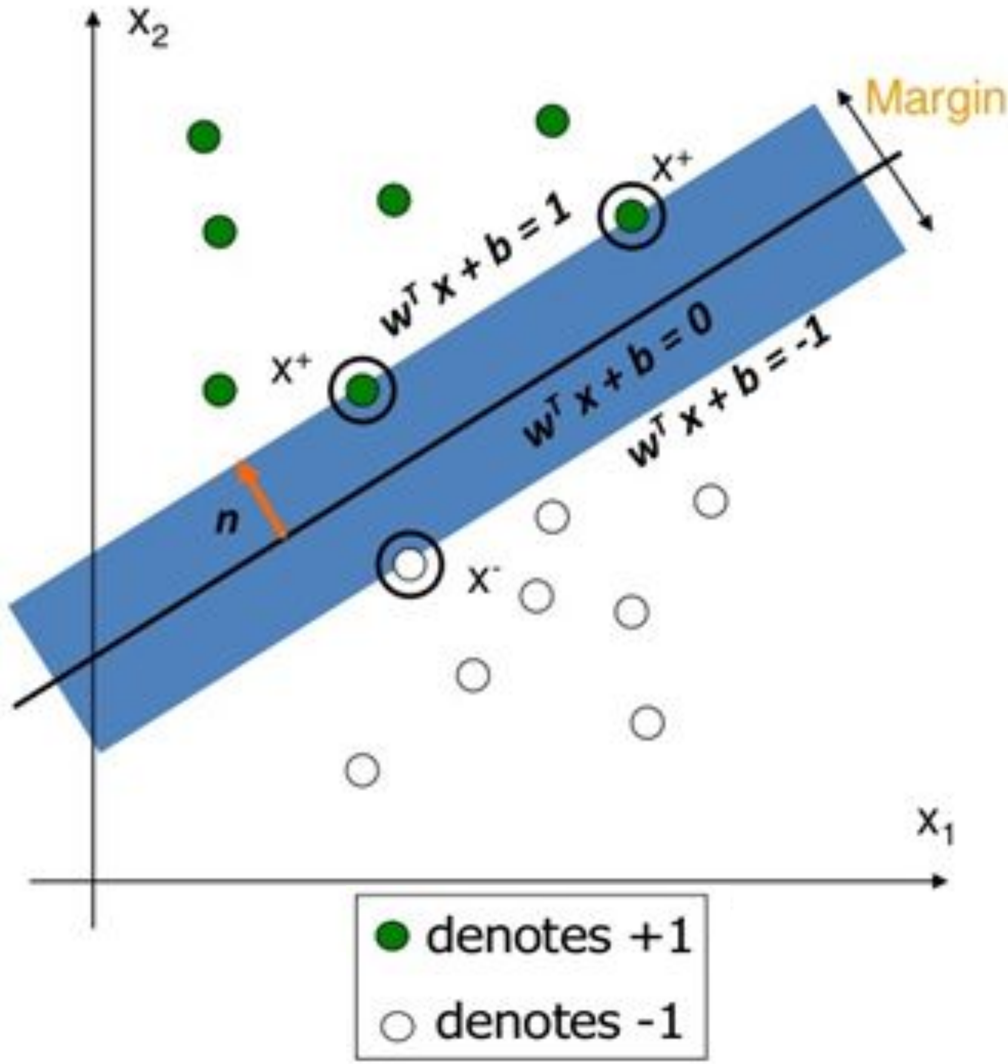
Understanding the basics

- Formulation:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

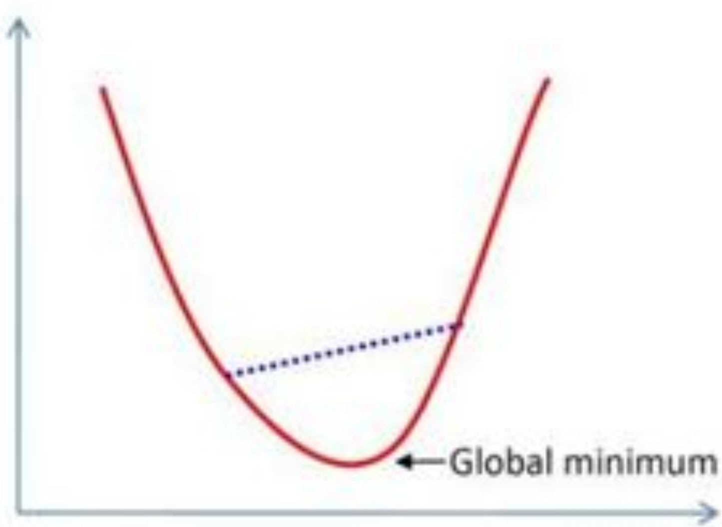
such that

$$\mathbf{y_i(W\,X + b) \geq 1}$$

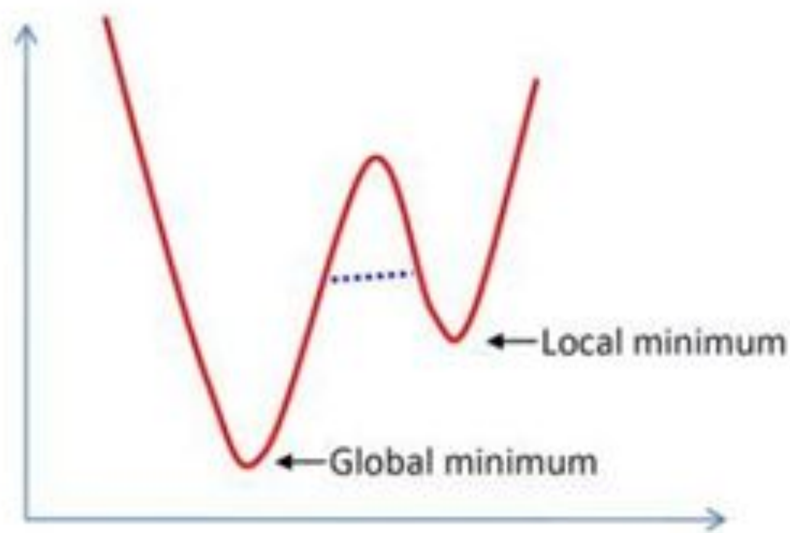


Basics of optimization: Convex functions

- A function is called ***convex*** if the function lies below the straight line segment connecting two points, for any two points in the interval.
- Property: Any local minimum is a global minimum!



Convex function



Non-convex function

Basics of optimization: Quadratic Programming

- Quadratic programming (QP) is a special optimization problem: the function to optimize ("*objective*") is quadratic, subject to linear *constraints*.
- Convex QP problems have convex objective functions.
- These problems can be solved easily and efficiently by greedy algorithms (because every local minimum is a global minimum).

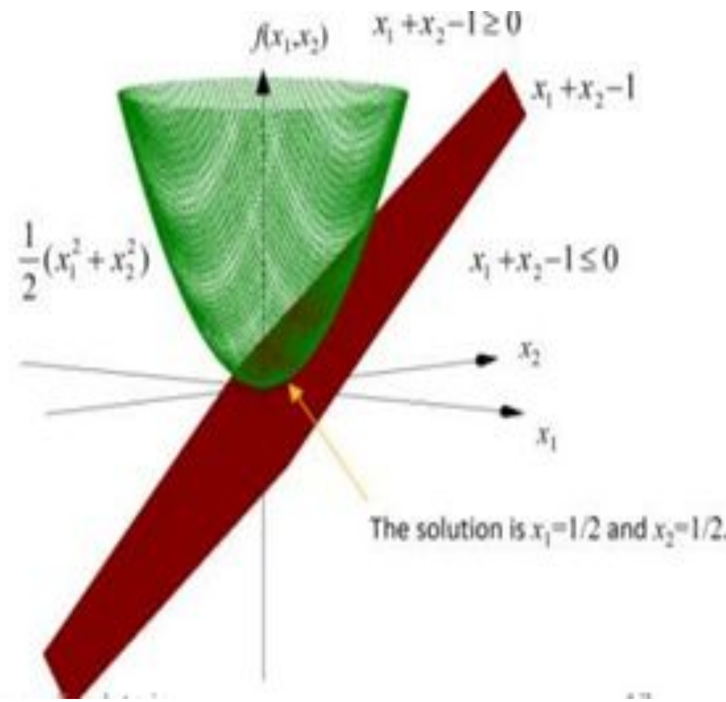
Consider $\vec{x} = (x_1, x_2)$

Minimize $\underbrace{\frac{1}{2}\|\vec{x}\|_2^2}_{\text{quadratic objective}}$ subject to $\underbrace{x_1 + x_2 - 1 \geq 0}_{\text{linear constraints}}$

This is QP problem, and it is a convex QP as we will see

We can rewrite it as:

Minimize $\underbrace{\frac{1}{2}(x_1^2 + x_2^2)}_{\text{quadratic objective}}$ subject to $\underbrace{x_1 + x_2 - 1 \geq 0}_{\text{linear constraints}}$



SVM optimization problem: Primal formulation

Minimize $\boxed{\frac{1}{2}\sum_{i=1}^{n} w_i^2}$ subject to $\boxed{y_i(\vec{w}\cdot\vec{x}_i + b) - 1 \geq 0}$ for $i = 1,\ldots,N$

Objective function — Constraints

- This is called ***"primal formulation of linear SVMs"***
- It is a convex quadratic programming (QP) optimization problem with *n* variables (w_i, i= 1,…,n), where n is the number of features in the dataset.

SVM optimization problem: Dual formulation

- The previous problem can be recast in the so-called "*dual form*" giving rise to "***dual formulation of linear SVMs***".

Minimize $\underbrace{\boxed{\frac{1}{2}\sum_{i=1}^{n} w_i^2}}_{\text{Objective function}}$ subject to $\underbrace{\boxed{y_i(\vec{w}\cdot\vec{x}_i + b) - 1 \geq 0}}_{\text{Constraints}}$ for $i = 1,\ldots,N$

- Apply the method of Lagrange multipliers.

Define Lagrangian $\Lambda_P(\vec{w}, b, \vec{\alpha}) = \frac{1}{2}\sum_{i=1}^{n} w_i^2 - \sum_{i=1}^{N} \alpha_i\left(y_i(\vec{w}\cdot\vec{x}_i + b) - 1\right)$

a vector with n elements

a vector with N elements

- We need to minimize this Lagrangian with respect to and simultaneously require that the derivative with respect to vanishes , all subject to the constraints that $\alpha_i > 0$

SVM optimization problem: Dual formulation

If we set the derivatives with respect to $\vec{w}, b$ to 0, we obtain:

$$\frac{\partial \Lambda_P(\vec{w}, b, \vec{\alpha})}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\frac{\partial \Lambda_P(\vec{w}, b, \vec{\alpha})}{\partial \vec{w}} = 0 \Rightarrow \vec{w} = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i$$

We substitute the above into the equation for $\Lambda_P(\vec{w}, b, \vec{\alpha})$ and obtain "*dual formulation of linear SVMs*":

$$\Lambda_D(\vec{\alpha}) = \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j$$

We seek to maximize the above Lagrangian with respect to $\vec{\alpha}$, subject to the constraints that $\alpha_i \geq 0$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$.

It is also a convex quadratic programming problem but with *N* variables (α_i, $i = 1,\ldots,N$), where *N* is the number of samples.

Maximize $\underbrace{\boxed{\sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j}}_{\text{Objective function}}$ subject to $\underbrace{\boxed{\alpha_i \geq 0 \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0}}_{\text{Constraints}}$.

Then the w-vector is defined in terms of α_i: $\vec{w} = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i$

And the solution becomes: $f(\vec{x}) = sign(\sum_{i=1}^{N} \alpha_i y_i \vec{x}_i \cdot \vec{x} + b)$

SVM optimization problem: Benefits of Using Dual formulation

1) No need to access original data, need to access only dot products.

Objective function: $\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \boxed{\vec{x}_i \cdot \vec{x}_j}$

Solution: $f(\vec{x}) = sign(\sum_{i=1}^{N} \alpha_i y_i \boxed{\vec{x}_i \cdot \vec{x}} + b)$

2) Number of free parameters is bounded by the number of support vectors and not by the number of variables (beneficial for high-dimensional problems).

Thank You