Week 03

Support Vector Machine

Prof. Ph.D. Woo Youn Kim

Chemistry, KAIST

Contents

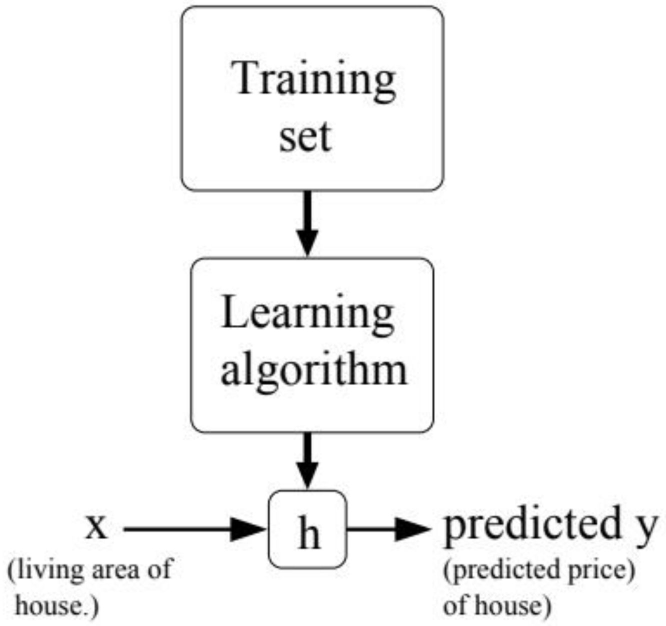
- Review on linear regression/classification
- Nonlinear problem
- Constrained minimization
- Support vector machine
- Support vector regression

Andrew Ng's lecture note
https://wikidocs.net/5719

Principle of machine learning

Under the previous probabilistic assumptions on the data, least-squares regression corresponds to finding the maximum likelihood estimate (MLE) of θ.

This is thus one set of assumptions under which least-squares regression can be justified as a very natural method that's just doing MLE.



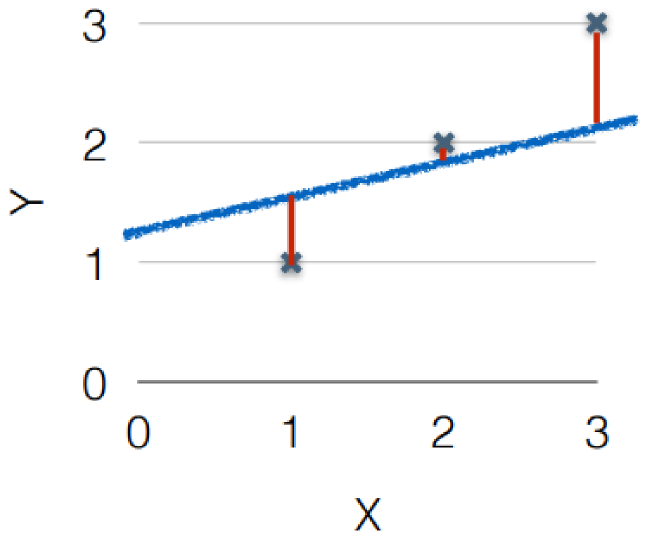
Linear regression review

Least-mean square (LMS) cost function

$$\frac{(H(x^{(1)}) - y^{(1)})^2 + (H(x^{(2)}) - y^{(2)})^2 + (H(x^{(3)}) - y^{(3)})^2}{3}$$

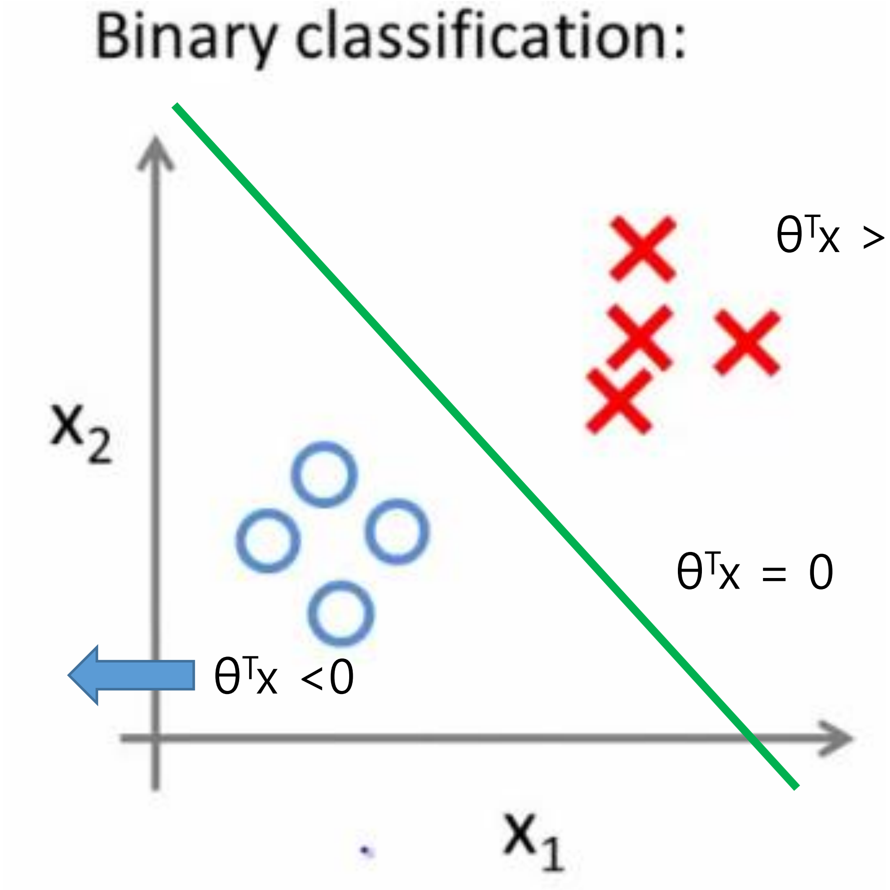
$$cost = \frac{1}{m} \sum_{i=1}^{m} (H(x^{(i)}) - y^{(i)})^2$$

$$H(x) = Wx + b$$



KAIST

Linear classification review



$\theta^T x > 0$ ➡ $y = h(\theta^T x) = 1$

$y = h(\theta^T x) = 1$ ⬅ $\theta^T x < 0$

For 2D, the dimension of θ will be two.

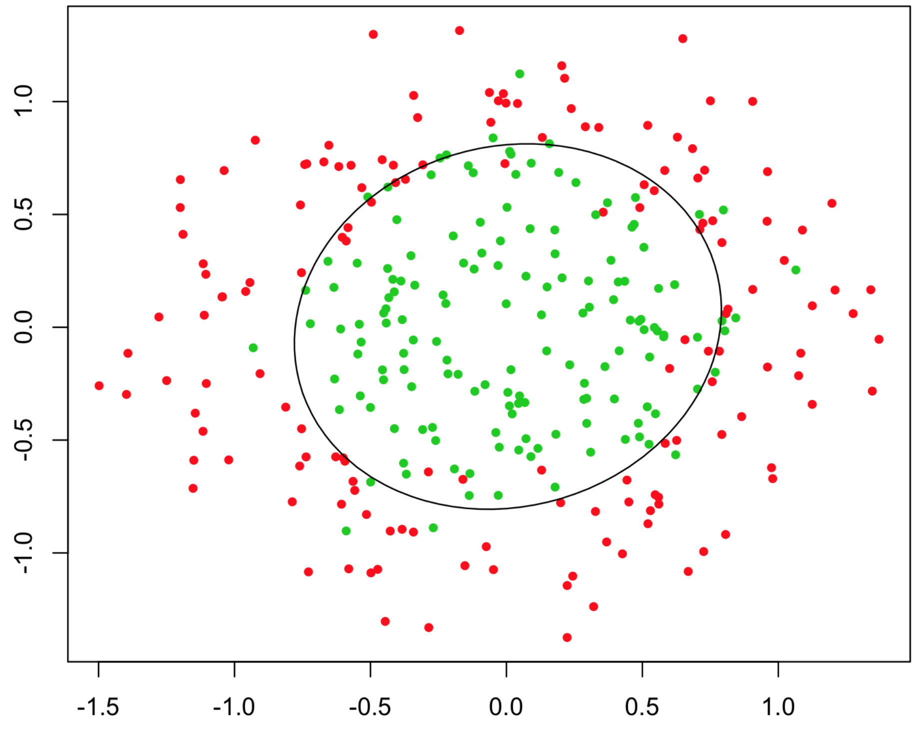
Thus, the **decision boundary** will be a linear line.

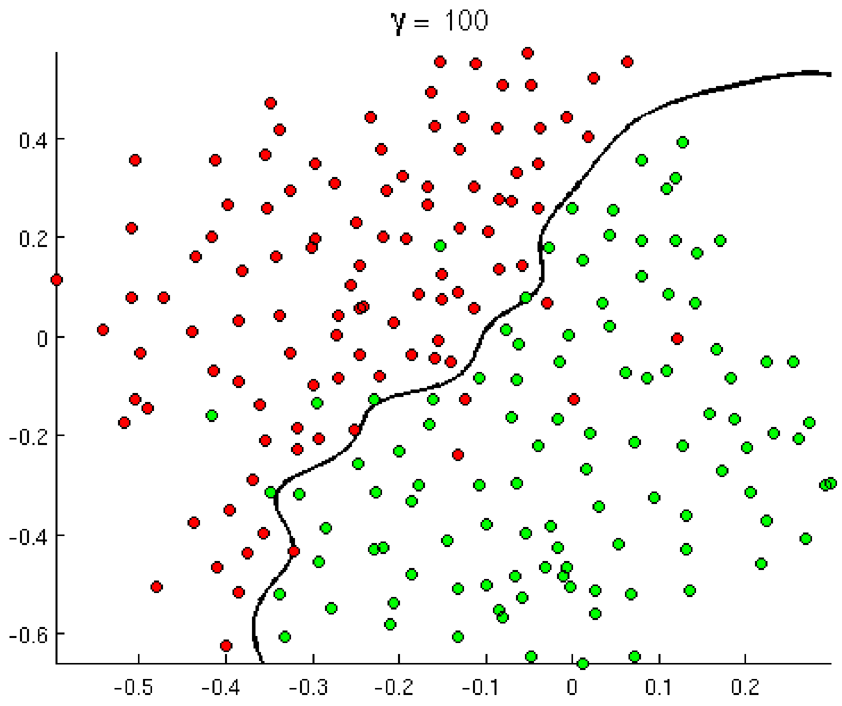
$$Cost = -\sum_{i=1}^{m}\left[y^{(i)} \ln h\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \ln\left(1 - h\left(x^{(i)}\right)\right)\right]$$

Cross entropy or softmax

Nonlinear problem

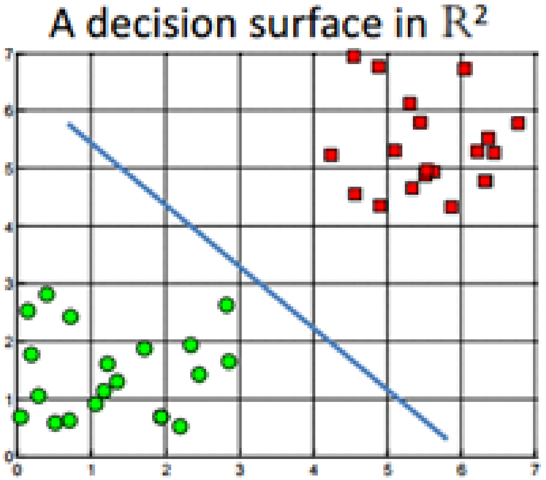
Non-linear decision boundary

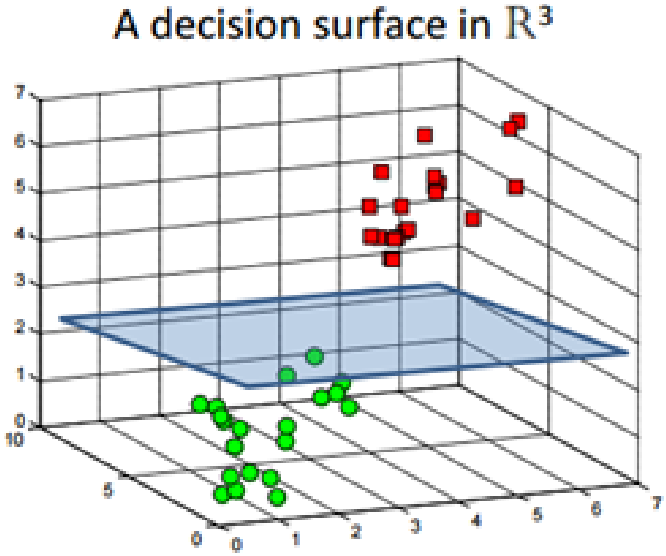




➔ non-linear method

n-dimensional space



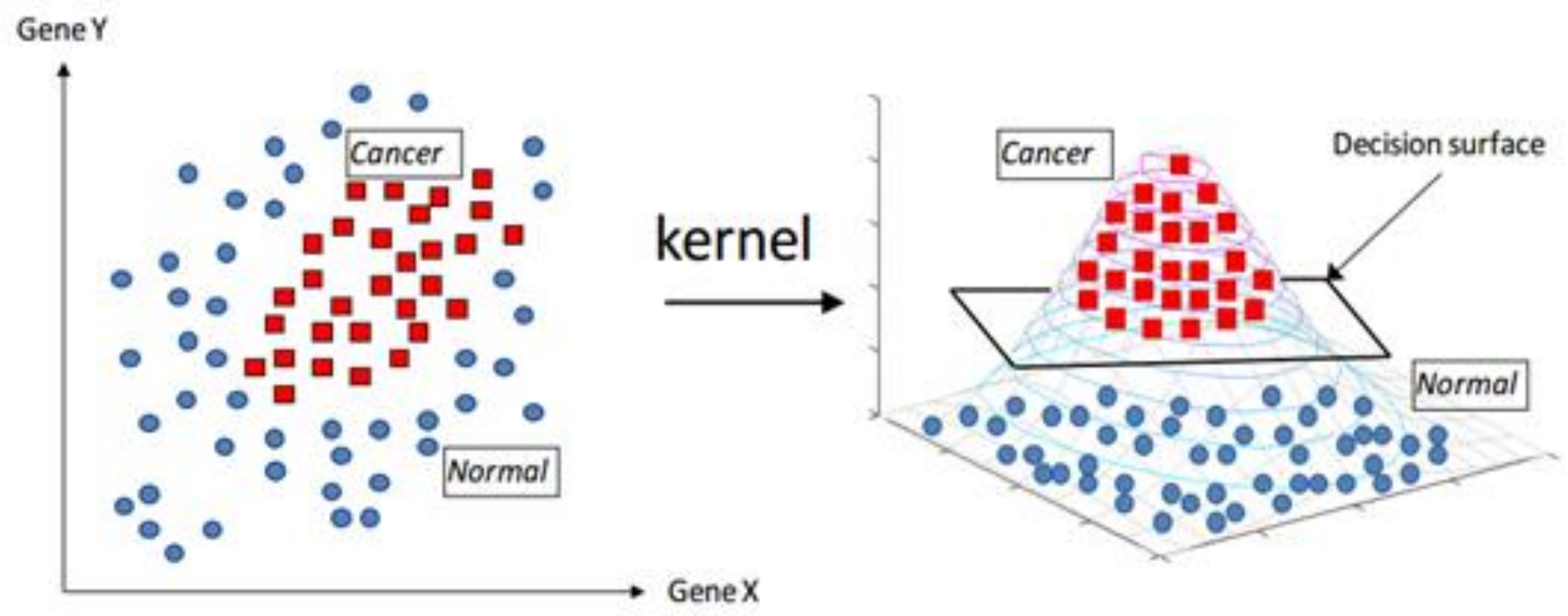


Data mapping: 2D ➔ n-dimensional space

Decision boundary: line ➔ hyperplane in (n-1)D subspace

Much easier to draw the decision boundary in a high-dimensional space

n-dimensional space



Data mapping: 2D ➔ n-dimensional space

Decision boundary: line ➔ hyperplane in (n-1)D subspace

Much easier to draw the decision boundary in a high-dimensional space

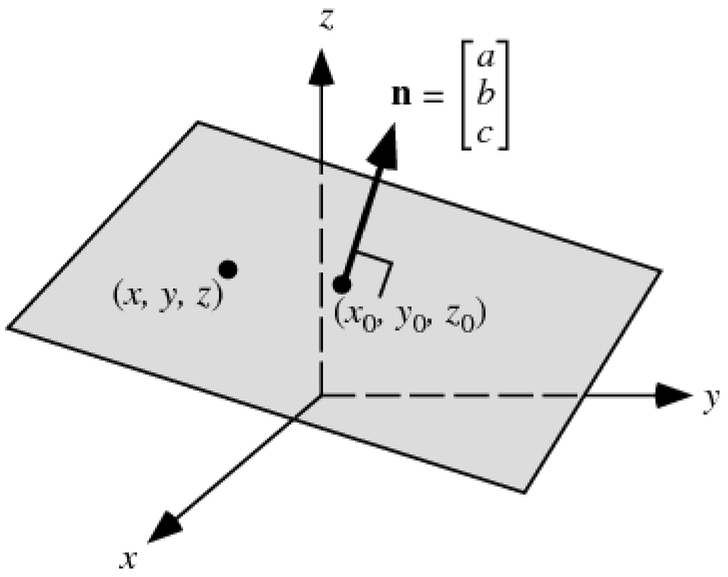
Decision boundary

Normal vector of a decision surface in 3D space:

$$\vec{v} = (a, b, c)$$

The equation of the decision surface with the normal vector above and a certain distance from the origin

$$ax + by + cz + d = 0$$



How about in a nD space?

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = c$$

http://mathworld.wolfram.com/Plane.html

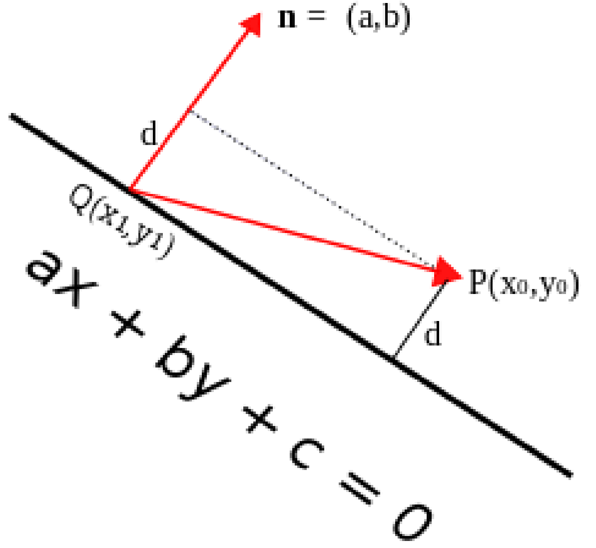
Distance between a plane and a point

- 2D line: distance between the line and a point

$$\text{D(line, P)} = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}$$

$$\text{D(line, origin)} = \frac{|c|}{\sqrt{a^2+b^2}} \qquad c = -\sqrt{a^2 + b^2} \times D(\text{line}, \text{origin})$$

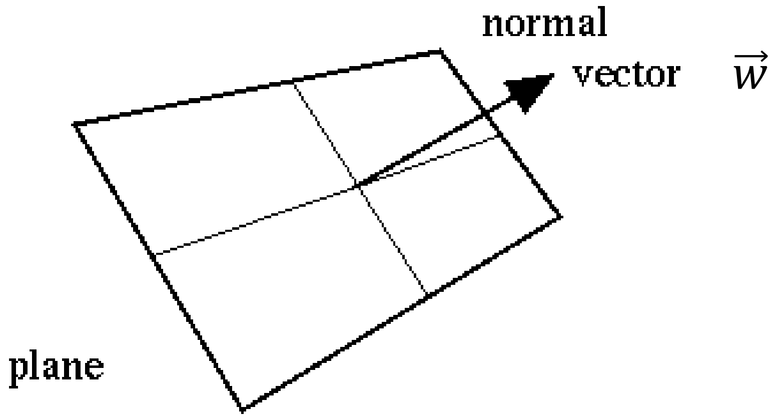
origin = (x_0=0, y_0=0)



- nD hyperplane: distance between the plane and a point

$$\text{D(plane, } \vec{x}) = \frac{|\vec{w} \cdot \vec{x} + b|}{\|w\|}$$

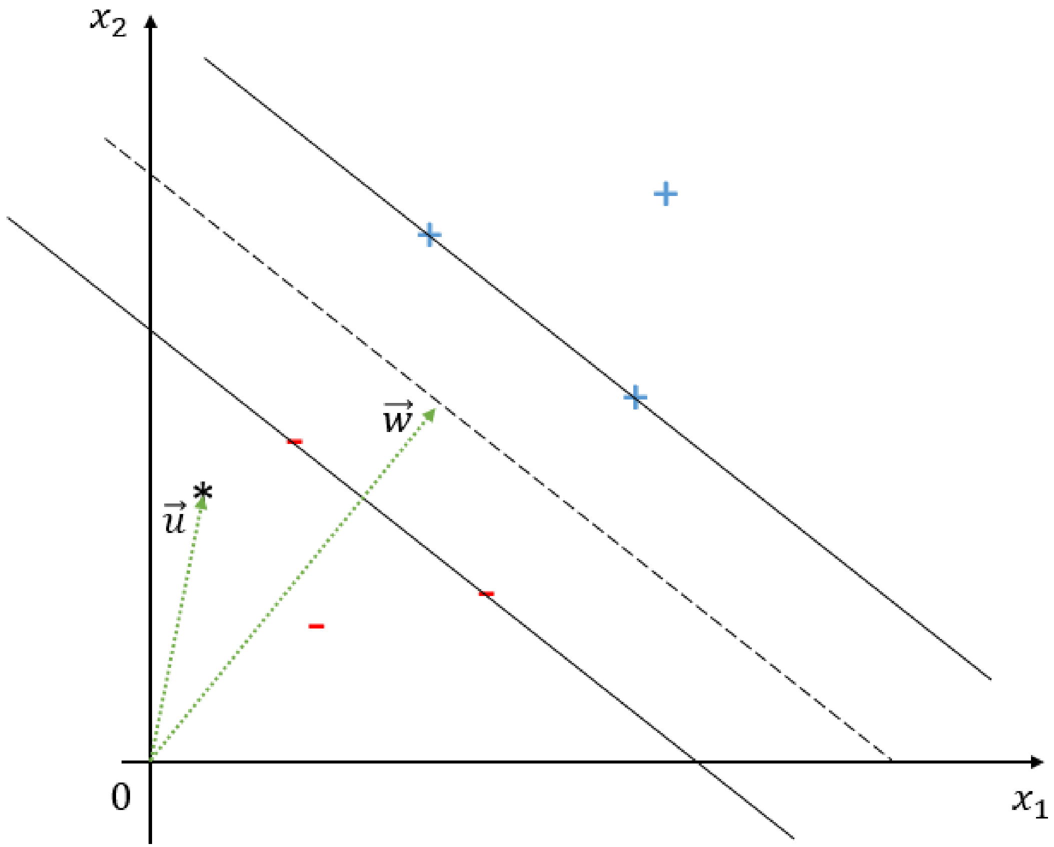
$$\text{D(plane, origin)} = \frac{|b|}{\|w\|} \qquad b = -\|w\| \times D(\text{line}, \text{origin})$$



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

KAIST

Decision rule



Decision boundary (the dashed line)

a hyperplane with a normal vector ***w***.

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

Decision rule:

for a new input ***u***, determine whether it belongs to + or -.

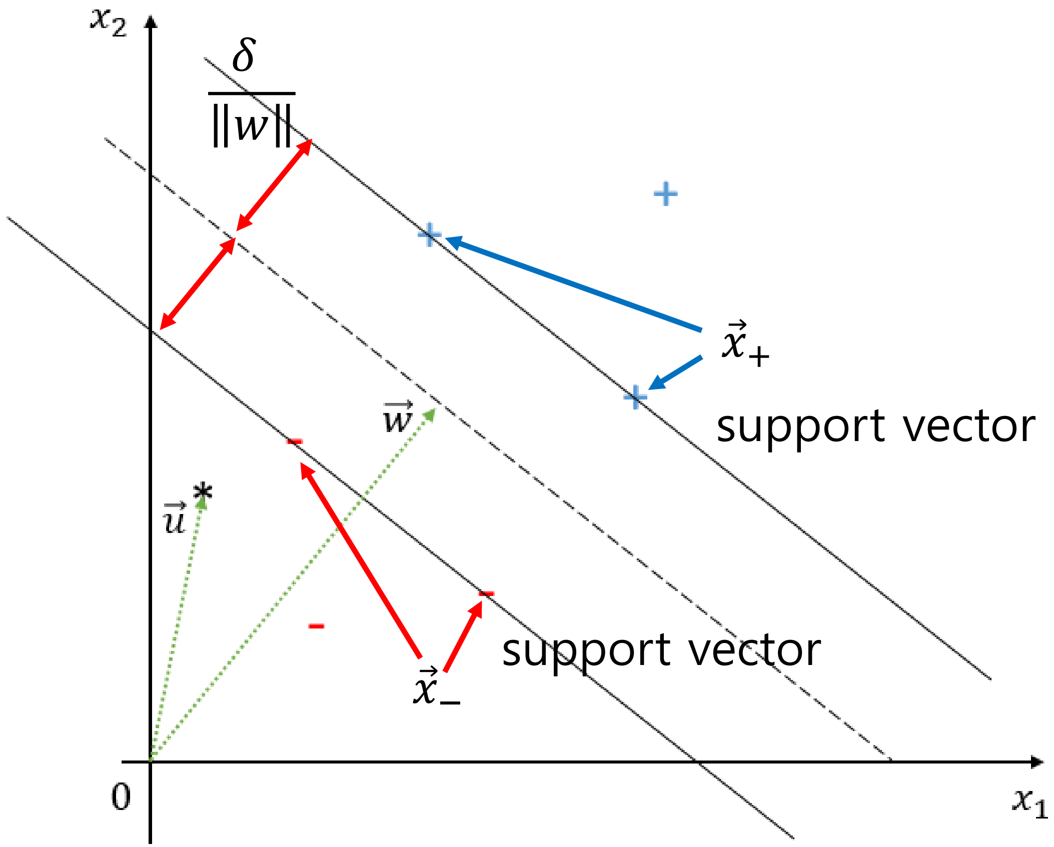
if the u is below the decision boundary, - otherwise +

$$\text{if } \vec{w} \cdot \vec{u} + b > 0, +$$

$$\text{if } \vec{w} \cdot \vec{u} + b < 0, -$$

where w and b should be determined.

Support vector



An optimal decision boundary given by $\vec{w} \cdot \vec{x} + b$=0 may be in an equidistance from the two solid lines in the figure.

$$\vec{w} \cdot \vec{x}_+ + b \geq \delta$$

$$\vec{w} \cdot \vec{x}_- + b \leq -\delta$$

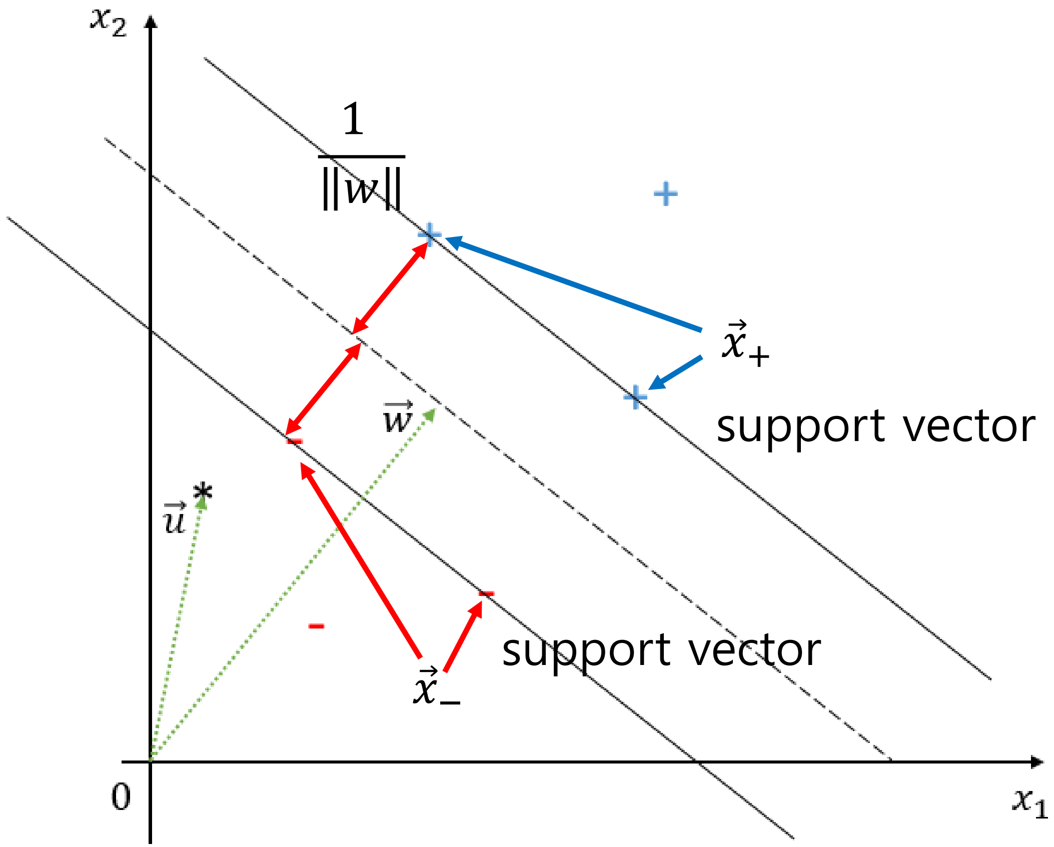
δ: geometric margin

w and b can be scaled by any constant.

Normalization of the geometric margin to 1

$$\vec{w} \cdot \vec{x}_+ + b \geq 1$$
$$\vec{w} \cdot \vec{x}_- + b \leq -1$$

Support vector



$$\vec{w} \cdot \vec{x}_+ + b \geq 1$$
$$\vec{w} \cdot \vec{x}_- + b \leq -1$$

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$$

where $y_i = \begin{cases} +1 \text{ for } \vec{x}_+ \\ -1 \text{ for } \vec{x}_- \end{cases}$

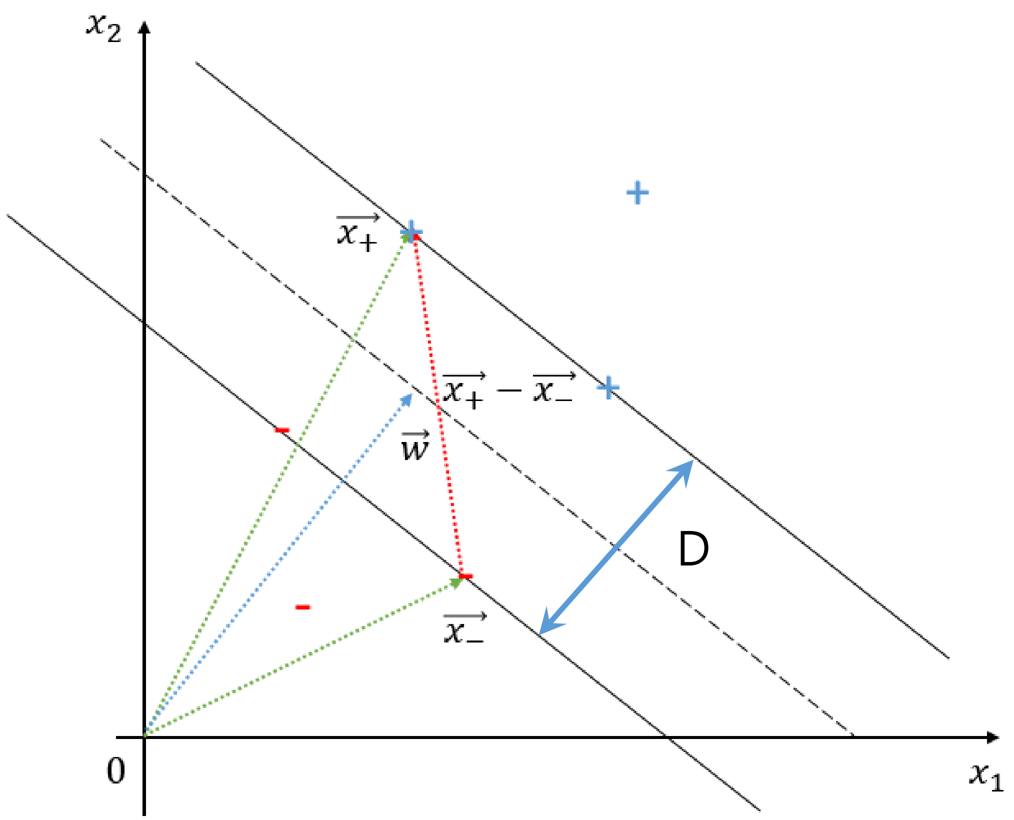
Thus, the support vectors satisfy

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0$$

Determining $\boldsymbol{w}$ and b satisfying the above equation ➔ decision boundary

KAIST

Optimal margin classifier



To find out an optimal decision boundary, we aim to find $\boldsymbol{w}$ and b to maximize D (geometric margin).

For the support vectors

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0$$

Distance between two solid lines (surfaces)

$$D = \frac{\vec{w}}{\|\vec{w}\|} \cdot (\vec{x}_+ - \vec{x}_-)$$

for $\vec{x}_+$: $\vec{w} \cdot \vec{x}_+ + b = 1 \rightarrow \vec{w} \cdot \vec{x}_+ = 1 - b$

for $\vec{x}_-$: $\vec{w} \cdot \vec{x}_- + b = -1 \rightarrow \vec{w} \cdot \vec{x}_+ = -1 - b$

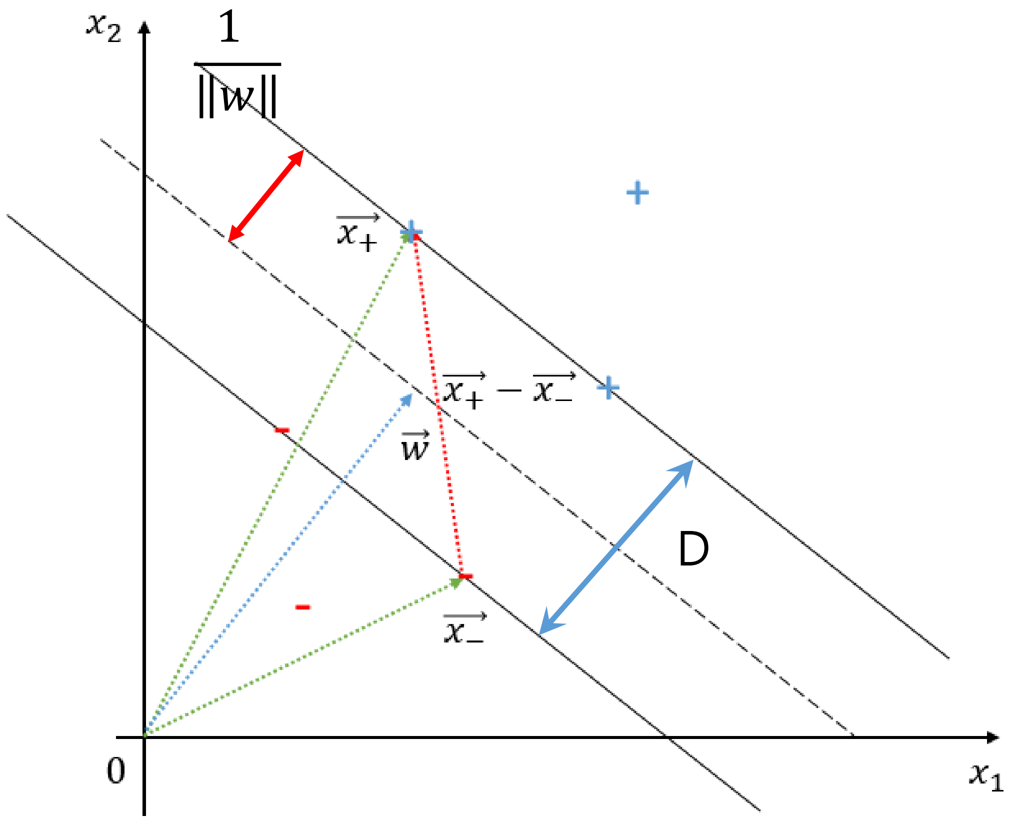
Thus, $D = \frac{1}{\|\vec{w}\|}(\vec{w} \cdot \vec{x}_+ - \vec{w} \cdot \vec{x}_-)$

$$= \frac{1}{\|\vec{w}\|}(1 - b + 1 + b) = \frac{2}{\|\vec{w}\|}$$

$$\max \frac{1}{\|\vec{w}\|} = \min \|\vec{w}\| \rightarrow \min \frac{1}{2}\|\vec{w}\|^2$$

KAIST

Optimal margin classifier



$$D = \frac{2}{\|\vec{\boldsymbol{w}}\|}$$

We aim to minimize the following for accurate classification.

$$\min \frac{1}{2} \|\vec{w}\|^2$$

with the following constraint

$$y_{\boldsymbol{i}}(\vec{w} \cdot \vec{x}_{\boldsymbol{i}} + b) - 1 = 0$$

Constraint minimization!

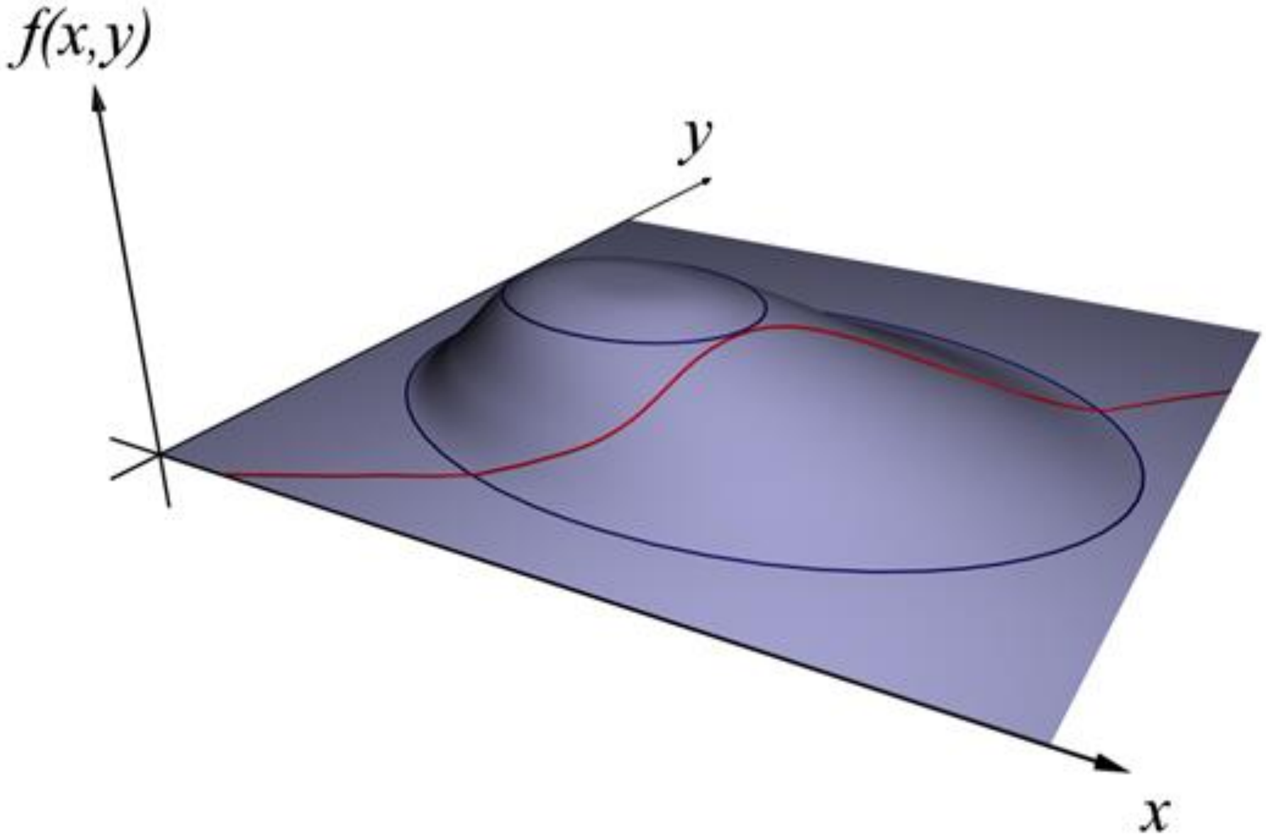
$$\min \frac{1}{2} \|\vec{w}\|^2 \quad \text{s.t.} \quad y_{\boldsymbol{i}}(\vec{w} \cdot \vec{x}_{\boldsymbol{i}} + b) - 1 = 0$$

KAIST

Constrained minimization

Constrained minimization

- Lagrange undetermined multiplier



Minimization without constraint

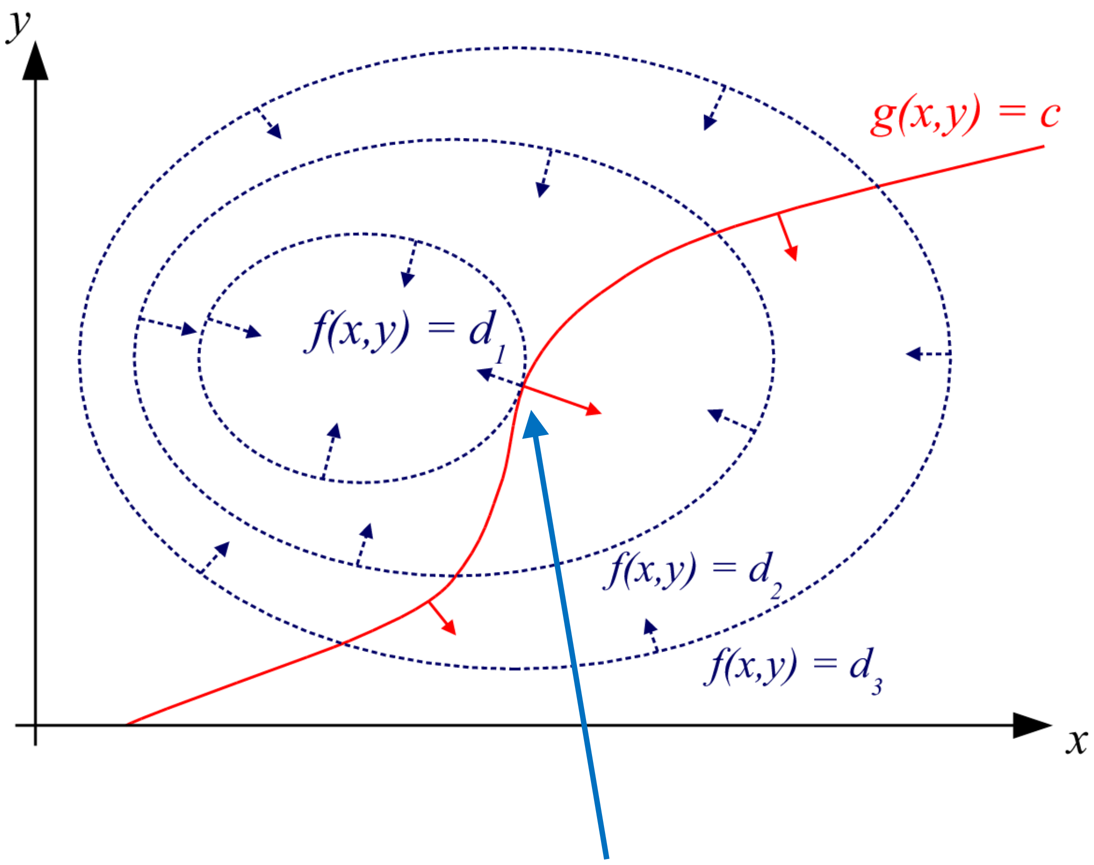
$$\frac{\partial f(x,y)}{\partial x}+\frac{\partial f(x,y)}{\partial y}=0$$

Minimization with constraint

$f(x, y)$=d s.t. $g(x, y)$=c

Constrained minimization

- Lagrange undetermined multiplier



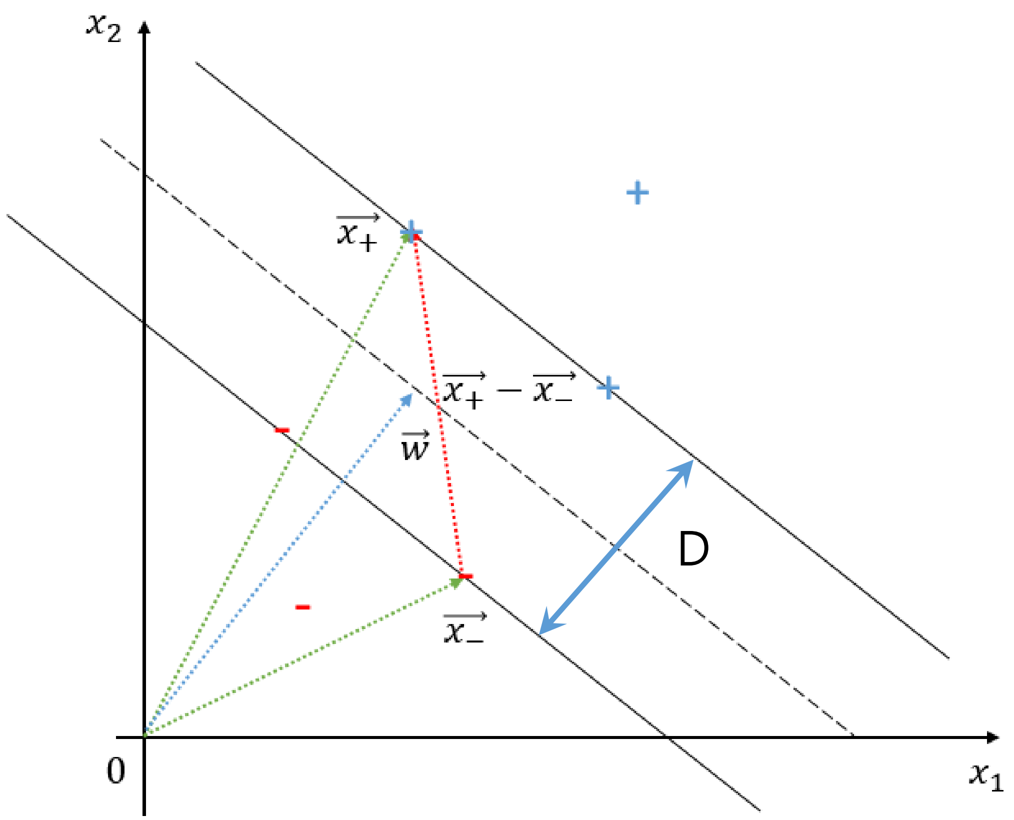
Minimization with constraint

$$L(x, y, \alpha) = f(x, y) - \alpha[g(x, y) - c]$$

$$\frac{\partial L(x,y,\alpha)}{\partial x} + \frac{\partial L(x,y,\alpha)}{\partial y} = 0$$

f(x,y) becomes minimum (or maximum) under the constraint g(x,y)=c

Constrained minimization



$$\min \frac{1}{2}\|\vec{w}\|^2 \quad \text{s.t.} \quad y_{\boldsymbol{i}}(\vec{w}\cdot\vec{x}_{\boldsymbol{i}}+b)-1=0$$

➔ constraint minimization

$$L=\frac{1}{2}\|\vec{w}\|^2-\sum_{\boldsymbol{i}}\alpha_i\left[y_{\boldsymbol{i}}(\vec{w}\cdot\vec{x}_{\boldsymbol{i}}+b)-1\right]$$

$$\frac{\partial L}{\partial \vec{w}}=\vec{w}-\sum_{\boldsymbol{i}}\alpha_i\, y_{\boldsymbol{i}}\vec{x}_{\boldsymbol{i}}=0$$

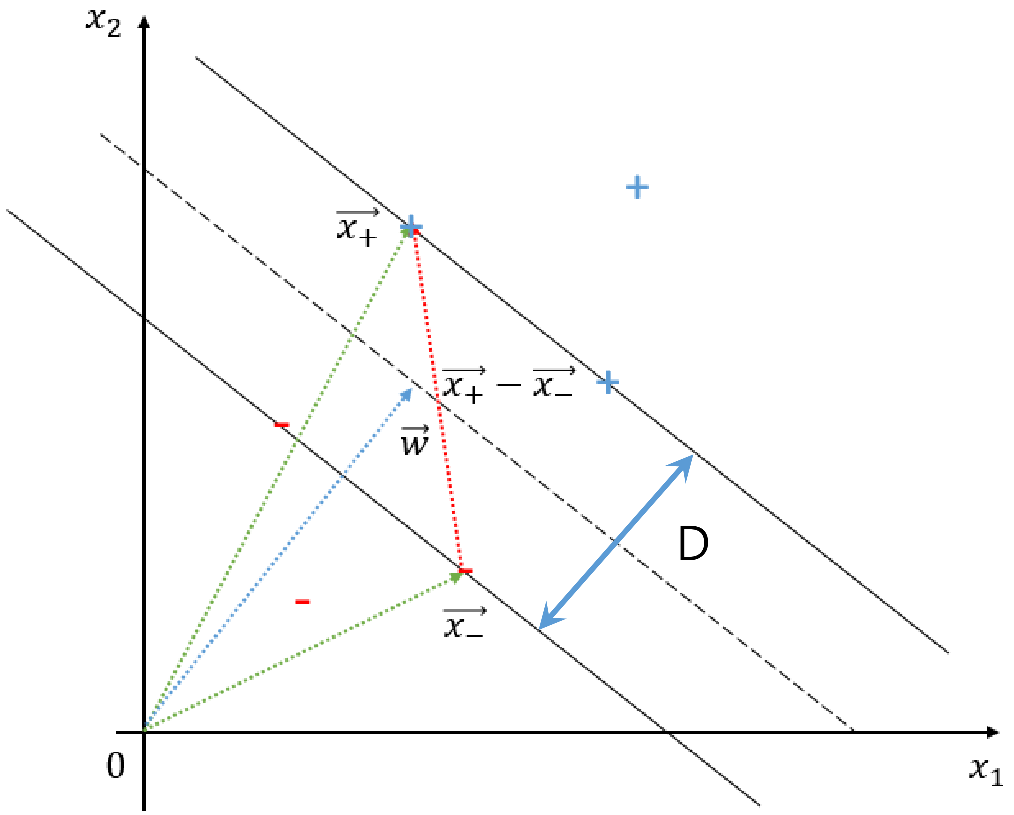
At the minimum

$$\vec{w}=\sum_{\boldsymbol{i}}\alpha_i\, y_{\boldsymbol{i}}\vec{x}_{\boldsymbol{i}} \qquad \text{eq.1}$$

linear combination of data points $\{\mathbf{x}_i\}$

KAIST

Constrained minimization



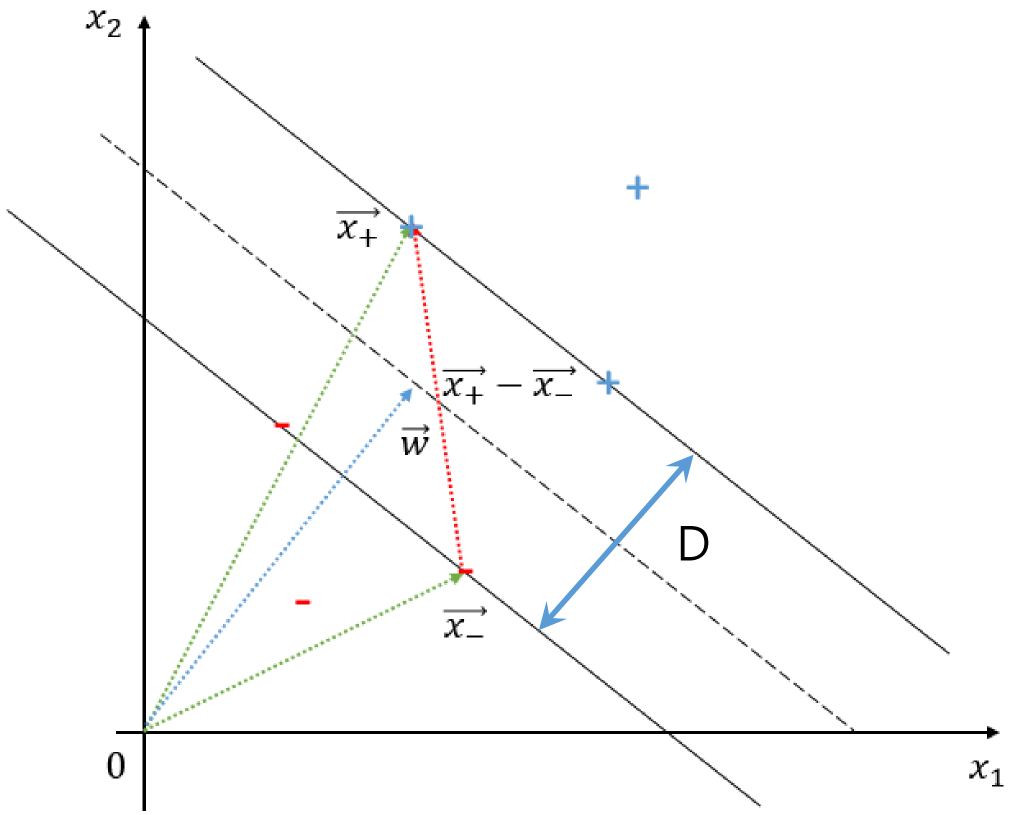
$$\min \frac{1}{2}\|\vec{w}\|^2 \quad \text{s.t.} \quad y_i(\vec{w}\cdot\vec{x}_i + b) - 1 = 0$$

At the minimum

$$\vec{w} = \sum_i \alpha_i y_i \vec{x}_i \qquad \text{eq.1}$$

$$f(x) = \sum_i \alpha_i y_i \vec{x}_i \cdot \vec{x} + b \begin{cases} = 1 & \text{for } + \quad \text{eq.2} \\ = -1 & \text{for } - \quad \text{eq.3} \end{cases}$$

Constrained minimization



$$\min \frac{1}{2}\|\vec{w}\|^2 \quad \text{s.t.} \quad y_{\boldsymbol{i}}(\vec{w} \cdot \vec{x}_{\boldsymbol{i}} + b) - 1 = 0$$

➔ constrained minimization

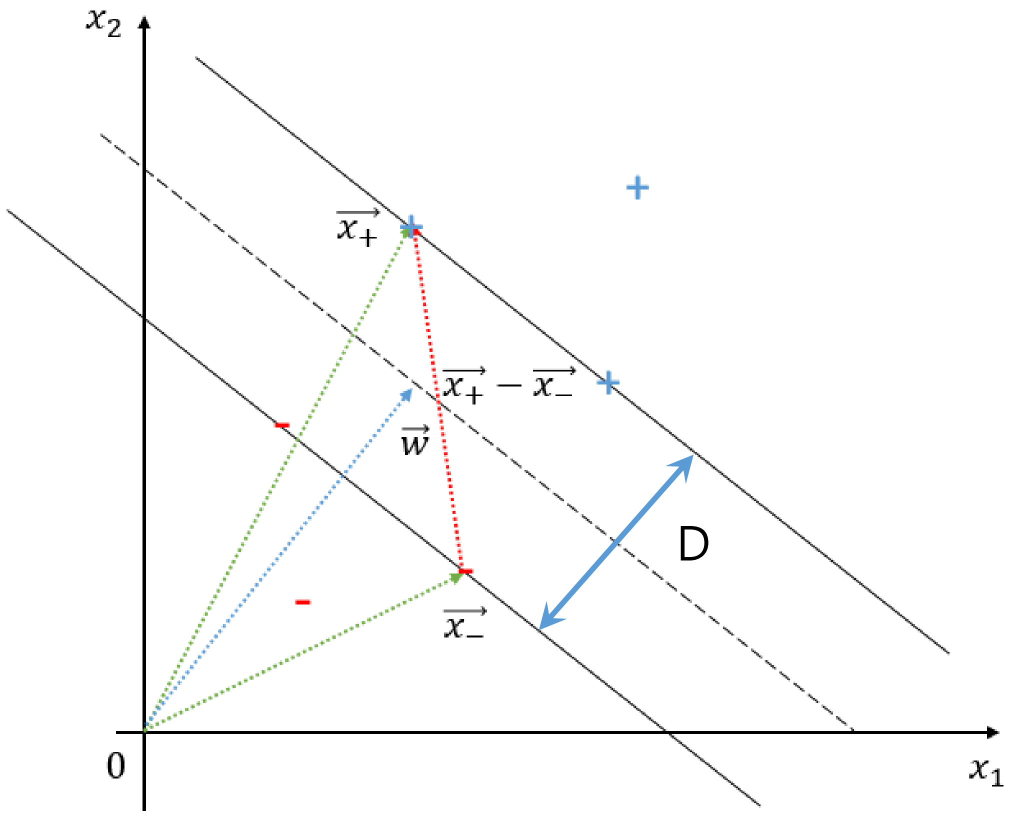
$$L = \frac{1}{2}\|\vec{w}\|^2 - \sum_{\boldsymbol{i}} \alpha_i \left[y_{\boldsymbol{i}}(\vec{w} \cdot \vec{x}_{\boldsymbol{i}} + b) - 1\right]$$

$$\frac{\partial L}{\partial b} = -\sum_{\boldsymbol{i}} \alpha_i y_{\boldsymbol{i}} = 0$$

At the minimum

$$\sum_{\boldsymbol{i}} \alpha_i y_{\boldsymbol{i}} = 0 \qquad \text{eq.4}$$

Constrained minimization



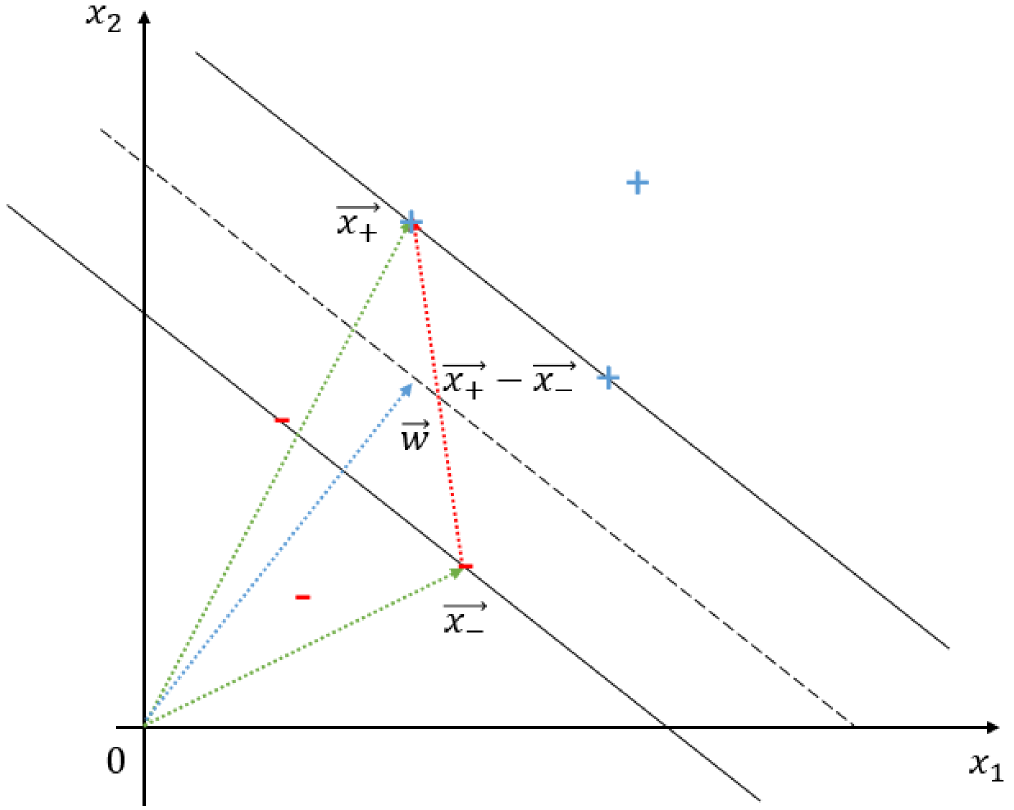
At the minimum

$$f(x) = \sum_{\boldsymbol{i}} \alpha_i\, y_{\boldsymbol{i}} \vec{x}_{\boldsymbol{i}} \cdot \vec{x} + b \begin{cases} = 1 & \text{for} + \\ = -1 & \text{for} - \end{cases} \quad \begin{matrix} \text{eq.2} \\ \text{eq.3} \end{matrix}$$

$$\sum_{\boldsymbol{i}} \alpha_i\, y_{\boldsymbol{i}} = 0 \qquad \text{eq.4}$$

eq.2, 3, 4 ➔ α_i

Constrained minimization



Using eq.1 and 2

$$\vec{w} = \sum_{i} \alpha_i y_i \vec{x}_i \qquad \sum_{i} \alpha_i y_i = 0$$

$$L = \frac{1}{2}\|\vec{w}\|^2 - \sum_{i} \alpha_i \left[y_i(\vec{w}\cdot\vec{x}_i + b) - 1\right]$$

Then,

$$L = \sum_{i} \alpha_i - \frac{1}{2}\sum_{i}\sum_{j} \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j)$$

Support vector machine

Support vector machine (SVM)

Decision boundary (the dashed line)

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

Decision rule:

If $f(x)=\vec{w}\cdot\vec{x}+b>0, +$

If $f(x)=\vec{w}\cdot\vec{x}+b<0, -$

where w and b should be determined.

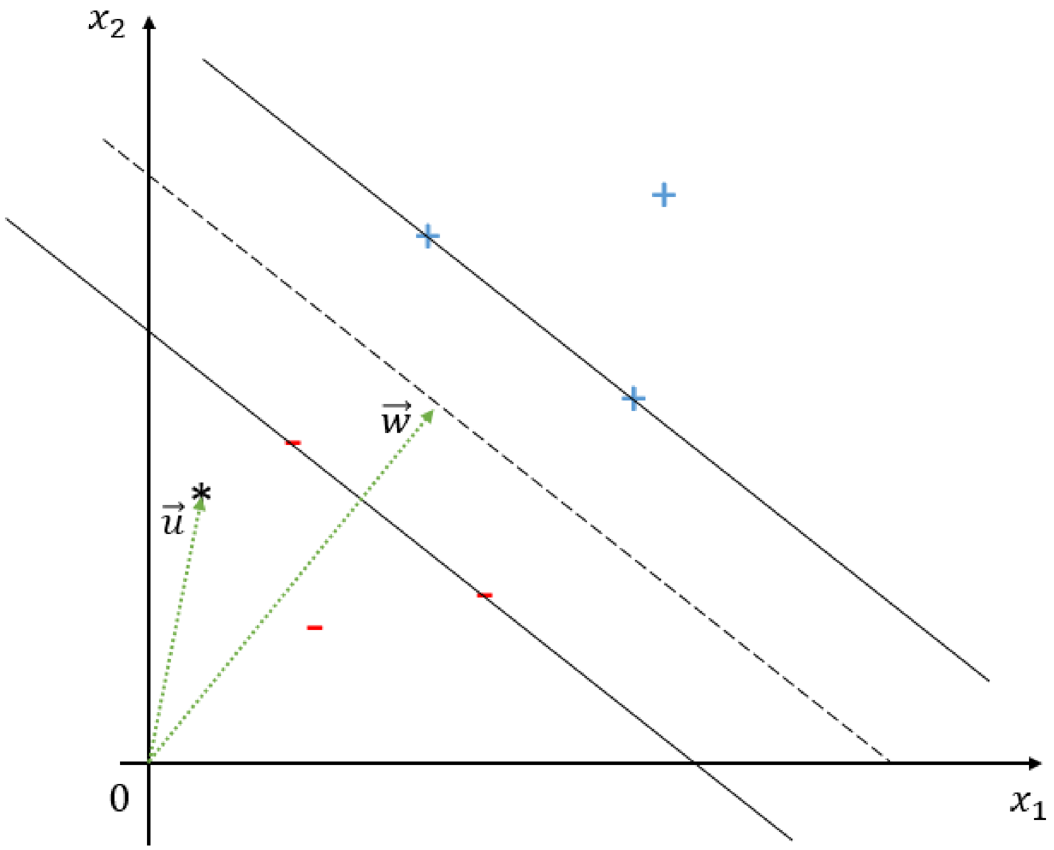
At the minimum of ***L***, $\sum_{\boldsymbol{i}}\alpha_i y_{\boldsymbol{i}}=0$ eq.4

Thus, for a new input x,

$$f(x)=\vec{w}\cdot\vec{x}+b$$

$$=(\sum_i \alpha_i y_i \vec{x}_i)\cdot\vec{x}+b$$

$$=\sum_i \alpha_i y_i (\vec{x}_i\cdot\vec{x})+b$$



Support vector machine (SVM)

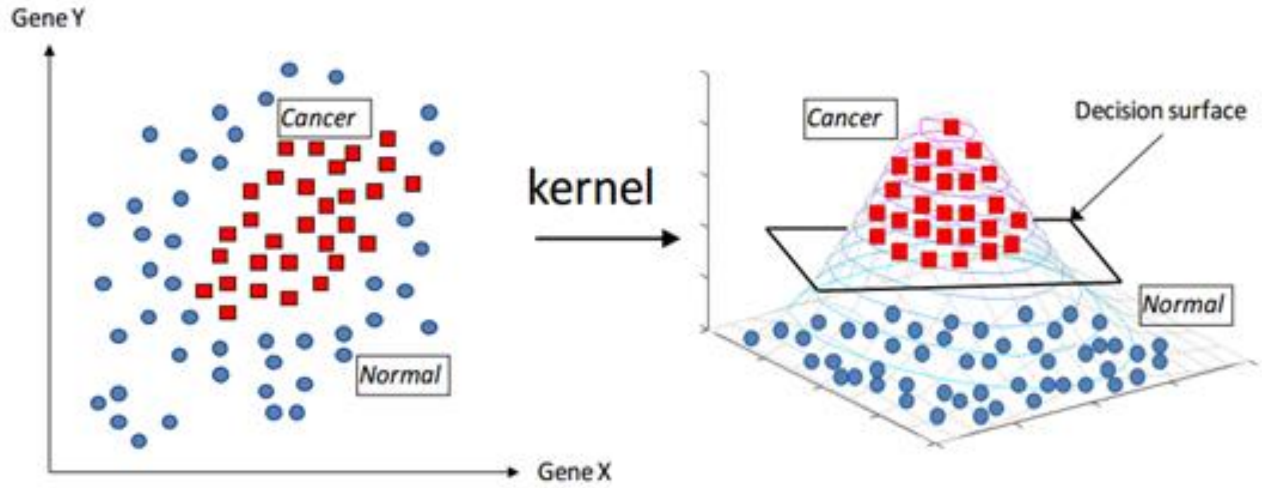
$$L = \sum_{i} \alpha_i - \frac{1}{2}\sum_{i}\sum_{j} \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j)$$

One can obtain $\{\alpha_i\}$ to minimize L and then determine ***w*** and *b*.

In addition, one can further minimize it by adjusting the inner product $\vec{x}_i \cdot \vec{x}_j$

1. transformation to a high dimensional space: $\vec{x}_i \rightarrow \phi(\vec{x}_i)$ $\phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$ Then, $\vec{x}_i \cdot \vec{x}_j \rightarrow \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$

➔feature mapping



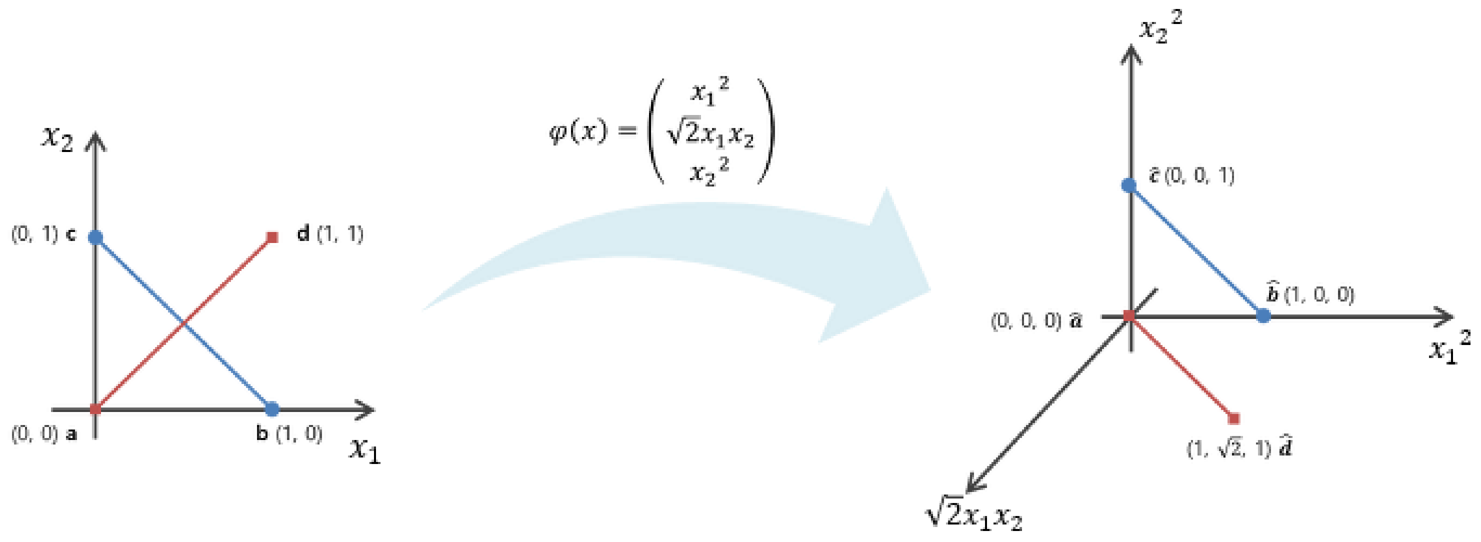
Support vector machine (SVM)

$\phi(\vec{x}_i)$ itself may be very expensive to calculate if it is an extremely high dimensional vector.
Instead, one can directly evaluate the inner product without explicitly finding $\phi(\vec{x}_i)$.

2. kernel: $\vec{x}_i \cdot \vec{x}_j \rightarrow \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) \rightarrow k(\vec{x}_i, \vec{x}_j)$

Homogeneous polynomial	$k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} \cdot \mathbf{x_j})^d$
Inhomogeneous polynomial	$k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} \cdot \mathbf{x_j} + 1)^d$
Gaussian kernel	$k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma \lVert \mathbf{x_i} - \mathbf{x_j} \rVert^2)$ for $\gamma > 0$.
Hyperbolic tangent	$k(\mathbf{x_i}, \mathbf{x_j}) = \tanh(\kappa \mathbf{x_i} \cdot \mathbf{x_j} + c)$, for some (not every) $\kappa > 0$ and $c < 0$

Example 1: homogeneous polynomial



$\mathbf{a} = (0,0)^{\mathrm{T}},\ t_{\mathbf{a}} = 1$

$\mathbf{b} = (1,0)^{\mathrm{T}},\ t_{\mathbf{b}} = -1$

$\mathbf{c} = (0,1)^{\mathrm{T}},\ t_{\mathbf{c}} = -1$

$\mathbf{d} = (1,1)^{\mathrm{T}},\ t_{\mathbf{d}} = 1$

$$\Phi_1(\mathbf{x}) = \begin{pmatrix} x_1{}^2 \\ \sqrt{2}x_1x_2 \\ x_2{}^2 \end{pmatrix}$$

$\mathbf{a} = (0,0)^{\mathrm{T}} \rightarrow \hat{\mathbf{a}} = (0,0,0)^{\mathrm{T}}$

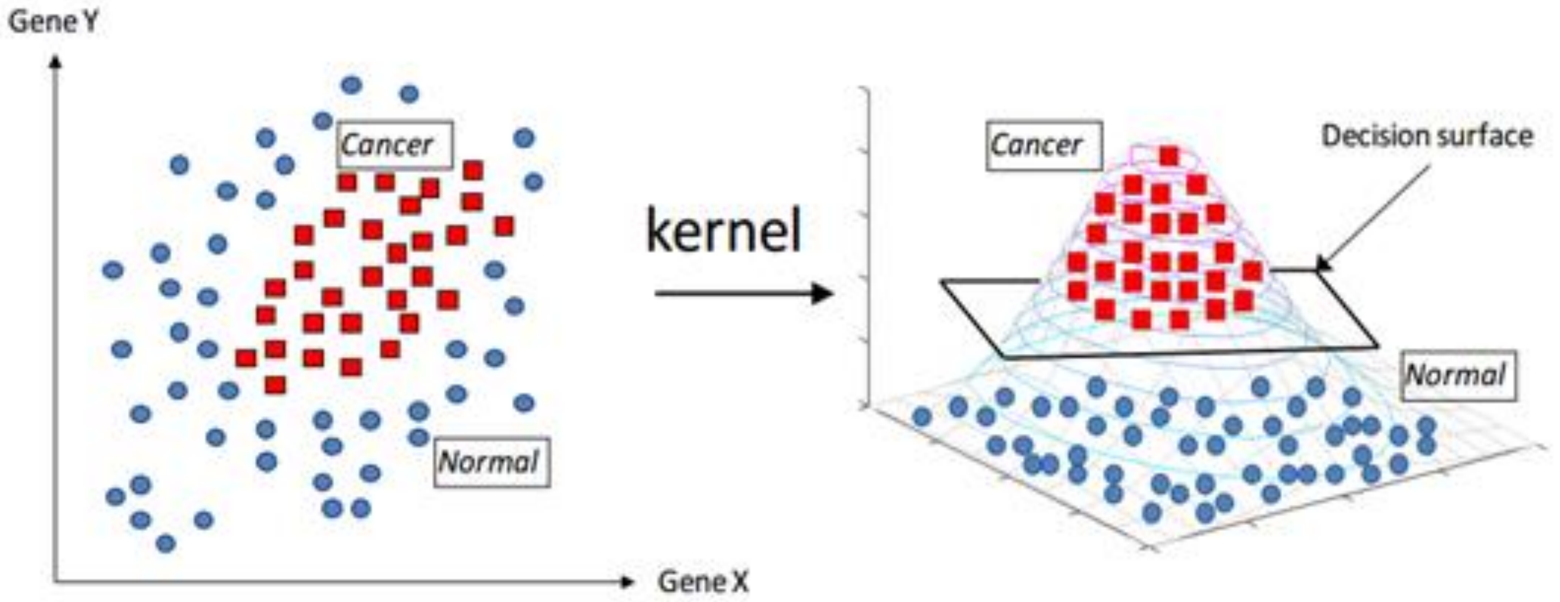
$\mathbf{b} = (1,0)^{\mathrm{T}} \rightarrow \hat{\mathbf{b}} = (1,0,0)^{\mathrm{T}}$

$\mathbf{c} = (0,1)^{\mathrm{T}} \rightarrow \hat{\mathbf{c}} = (0,0,1)^{\mathrm{T}}$

$\mathbf{d} = (1,1)^{\mathrm{T}} \rightarrow \hat{\mathbf{d}} = (1,\sqrt{2},1)^{\mathrm{T}}$

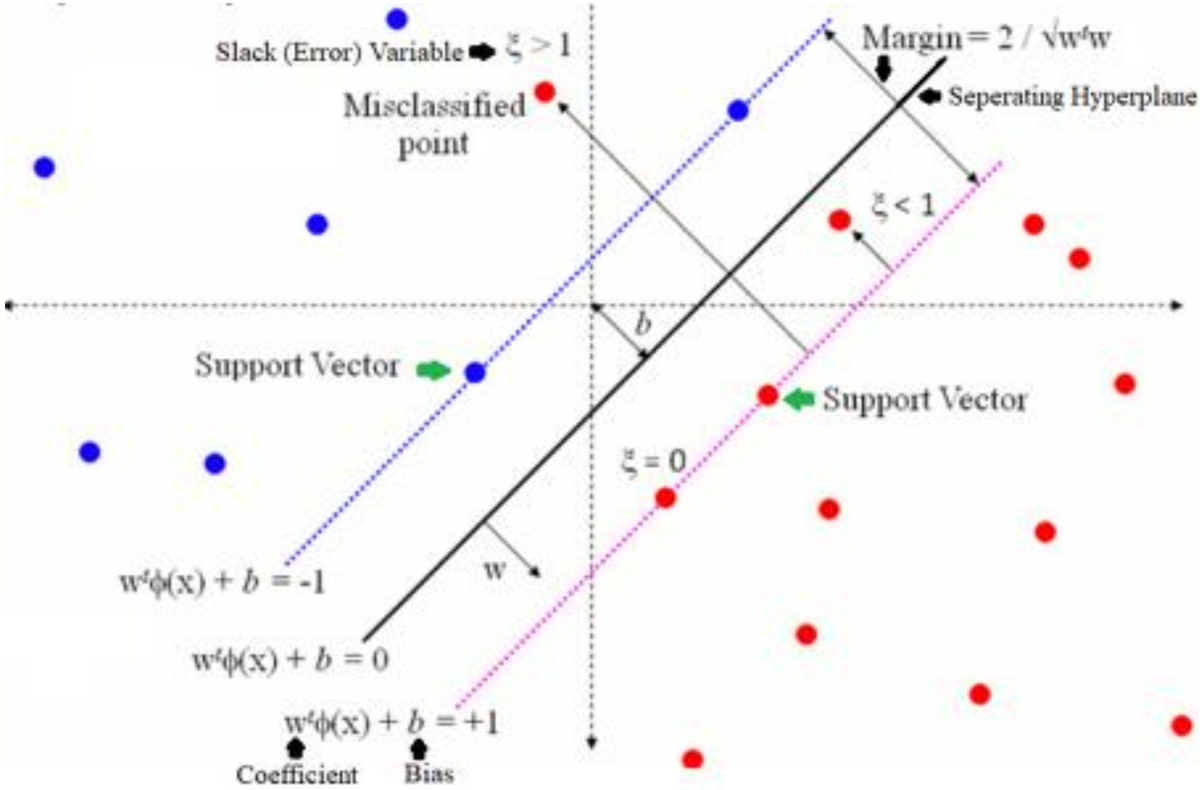
Example 2: Gaussian kernel

$$f(\mathbf{x}) = \sum_{i}^{N} \alpha_i y_i \exp\left(-||\mathbf{x} - \mathbf{x}_i||^2 / 2\sigma^2\right) + b$$



Regularization

$$\min \frac{1}{2} \|\vec{w}\|^2 \quad \text{s.t.} \quad y_i(\vec{w} \cdot \vec{x}_i + b) = 1$$

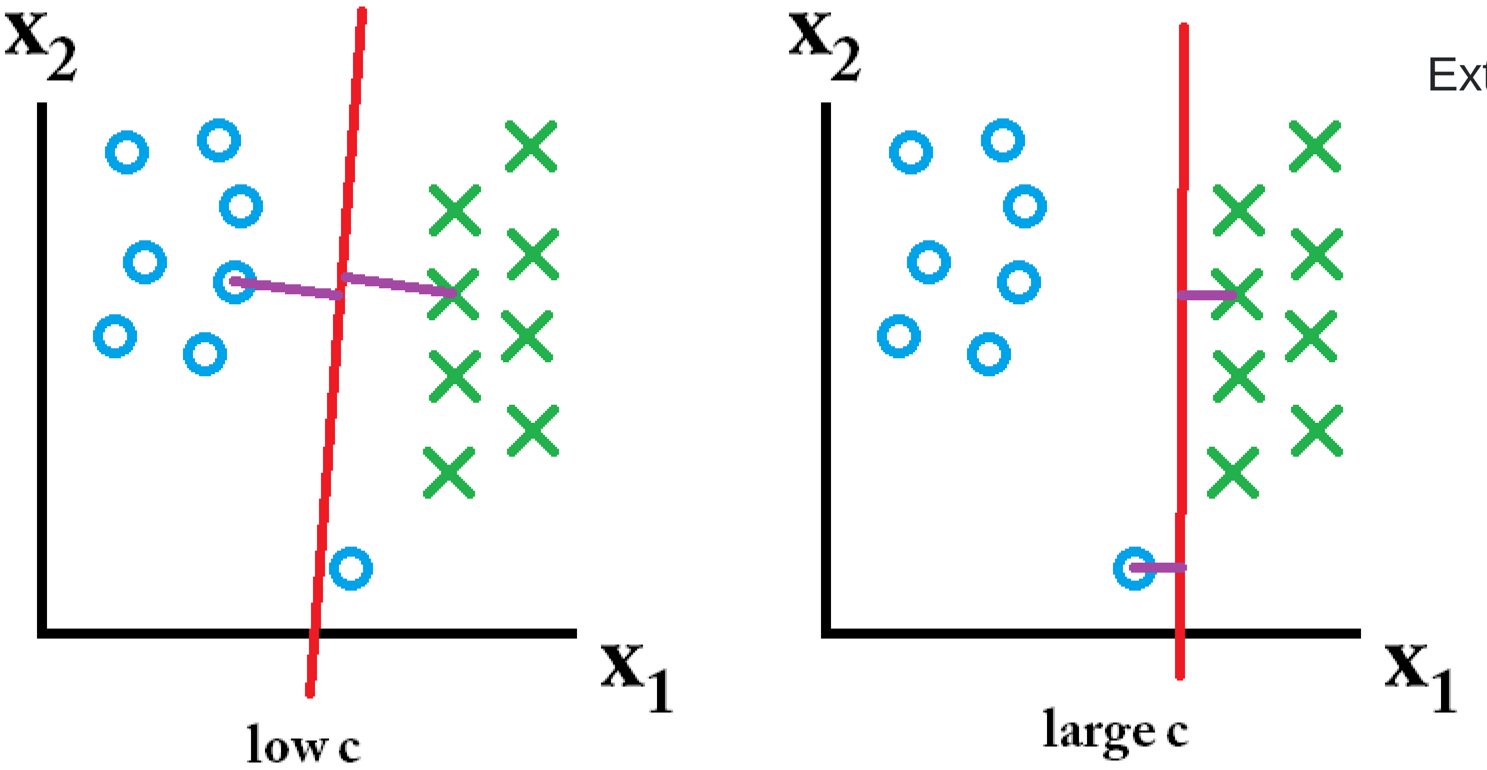


$$\min \frac{1}{2} \|\vec{w}\|^2 + C \sum_i \zeta_i \quad \text{s.t.} \quad y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0$$

to avoid overfitting

Regularization

The control parameter C



Extent of avoiding misclassifying data points

(i) small C

large ζ_i ➔ large margin with more misclassifying

(ii) large C

small ζ_i ➔ small margin with less misclassifying

$$\min \frac{1}{2}\|\vec{w}\|^2 + C\sum_i \zeta_i \qquad \text{s.t.} \quad y_i(\vec{w}\cdot\vec{x}_i + b) \geq 1 - \zeta_i, \qquad \zeta_i \geq 0$$

KAIST

Support vector regression

Kernel ridge regression (KRR)

SVM for nonlinear classification

$$\min \frac{1}{\mathbf{2}} \|\vec{\boldsymbol{w}}\|^2 \quad \text{s.t.} \quad y_{\boldsymbol{i}}(\vec{w} \cdot \vec{x}_{\boldsymbol{i}} + b) = 1$$

$$f(\boldsymbol{x}) = \vec{w} \cdot \vec{x} + b = \sum_i \alpha_i y_i (\vec{x}_i \cdot \vec{x}) + b = \sum_i \alpha_i y_i k(\vec{x}_i, \vec{x}) + b \begin{cases} > 0 & \text{for } + \\ < 0 & \text{for } - \end{cases}$$

Kernel ridge regression (KRR) combines a ridge regression (linear least squares) with the **kernel trick**

➔ nonlinear regression

$$\min \frac{1}{\mathbf{2}} \|\vec{\boldsymbol{w}}\|^2 \quad \text{s.t.} \quad (\vec{w} \cdot \vec{x}_{\boldsymbol{i}} + b) - y_{\boldsymbol{i}} = 0$$

Prediction for a given x

$$f(x) = \vec{w} \cdot \vec{x} + b = \sum_i \alpha_i (\vec{x}_i \cdot \vec{x}) + b = \sum_i \alpha_i k(\vec{x}_i, \vec{x}) + b$$

Support vector regression (SVR)

Kernel ridge regression (KRR)

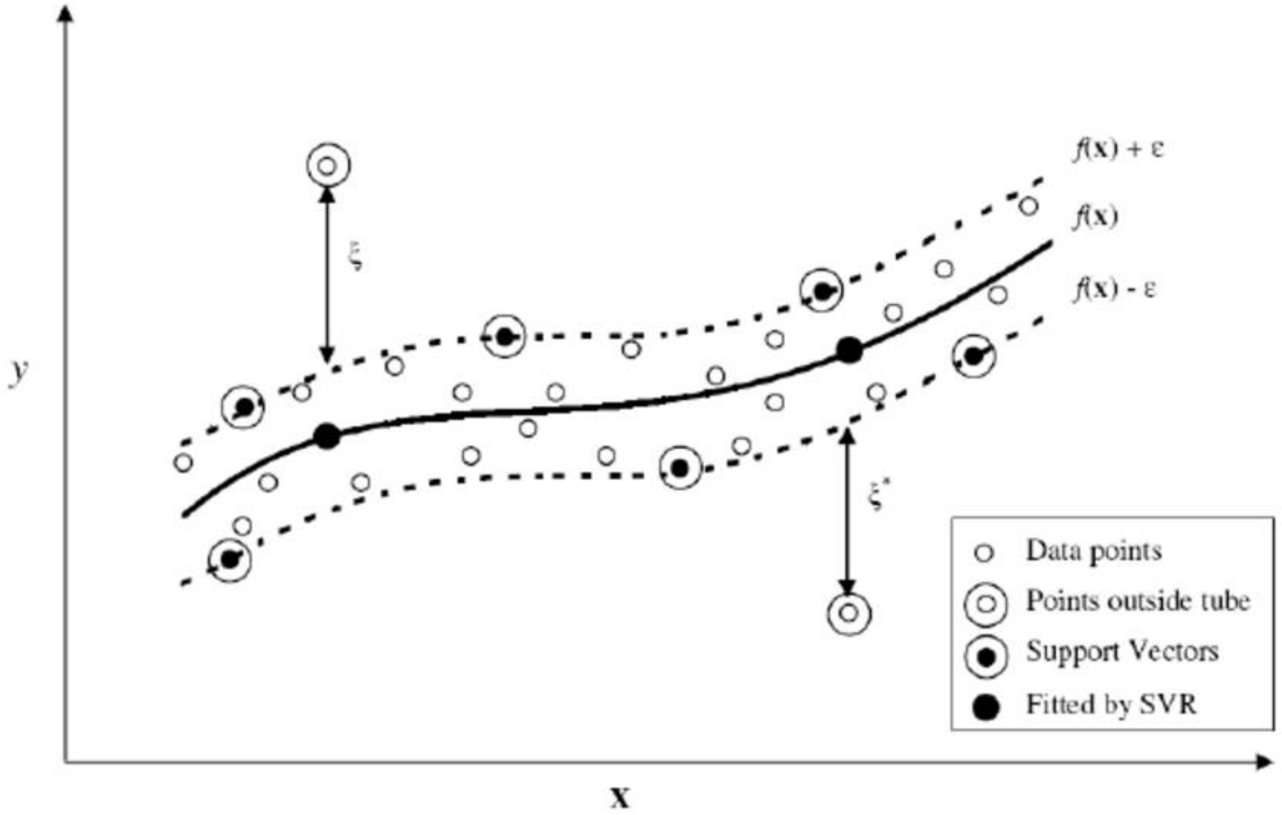
$$\min \frac{1}{2}\|\vec{\boldsymbol{w}}\|^2 \quad \text{s.t.} \quad (\vec{w} \cdot \vec{x}_{\boldsymbol{i}} + b) - y_{\boldsymbol{i}} = 0$$

$$f(x) = \sum_i \alpha_i k(\vec{x}_i, \vec{x}) + b$$

Support vector regression (SVR)

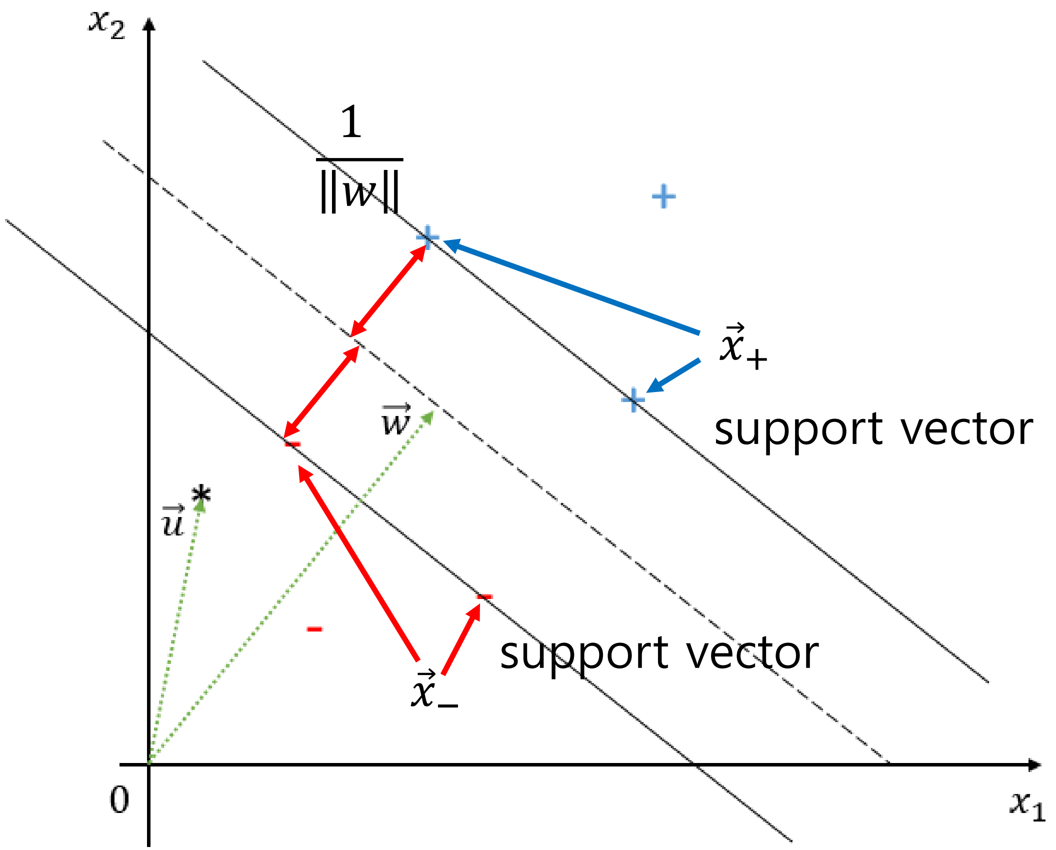
$$\min \frac{1}{2}\|\vec{\boldsymbol{w}}\|^2 \quad \text{s.t.} \; -\varepsilon < (\vec{w} \cdot \vec{x}_{\boldsymbol{i}} + b) - y_{\boldsymbol{i}} < \varepsilon$$

$$f(x) = \sum_i \alpha_i k(\vec{x}_i, \vec{x}) + b$$



KAIST

Summary



$$\min \frac{1}{2}\|\vec{\boldsymbol{w}}\|^2 \quad \text{s.t.} \quad y_{\boldsymbol{i}}(\vec{\boldsymbol{w}} \cdot \vec{x}_{\boldsymbol{i}} + b) - 1 = 0$$

to avoid overfitting

$$\min \frac{1}{2}\|\vec{\boldsymbol{w}}\|^2 + C\sum_i \zeta_i$$

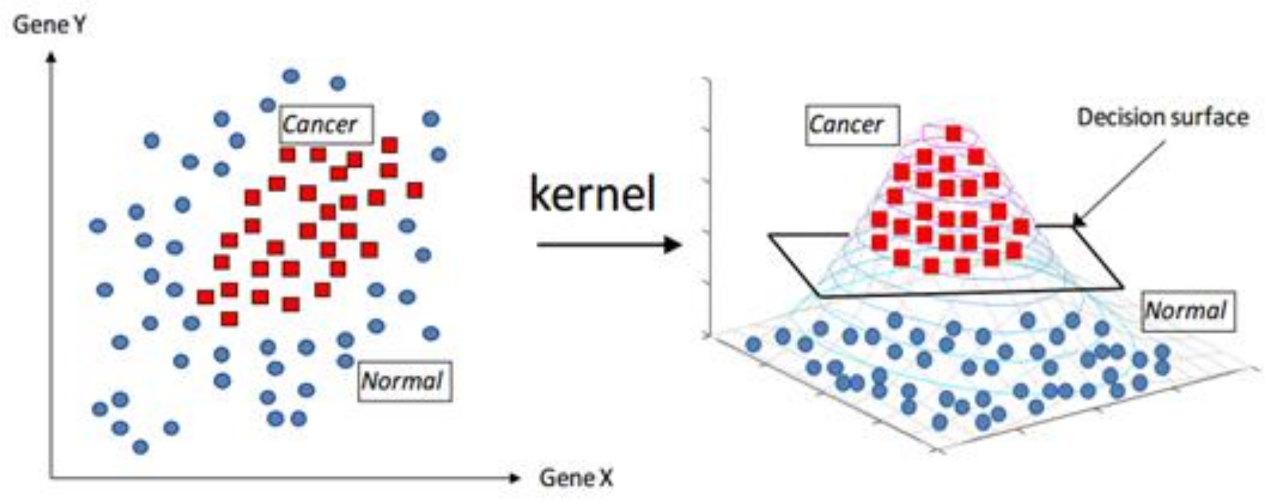
$$\text{s.t.} \quad y_{\boldsymbol{i}}(\vec{\boldsymbol{w}} \cdot \vec{x}_{\boldsymbol{i}} + b) \geq 1 - \zeta_i, \qquad \zeta_i \geq 0$$

Summary

1. transformation to a high dimensional space: $\vec{x}_i \rightarrow \phi(\vec{x}_i)$

$\rightarrow$ feature mapping

2. kernel: $\vec{x}_i \cdot \vec{x}_j \rightarrow \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) \rightarrow k(\vec{x}_i, \vec{x}_j)$



Best "off-the-shelf" classification methods

KAIST

New terms

Decision boundary
Support vector
Geometric margin
Functional margin
Optimal margin classifier
Constraint minimization
Support vector machine
Feature mapping
Kernel trick
Regularization
Kernel ridge regression
Support vector regression