データマイニング

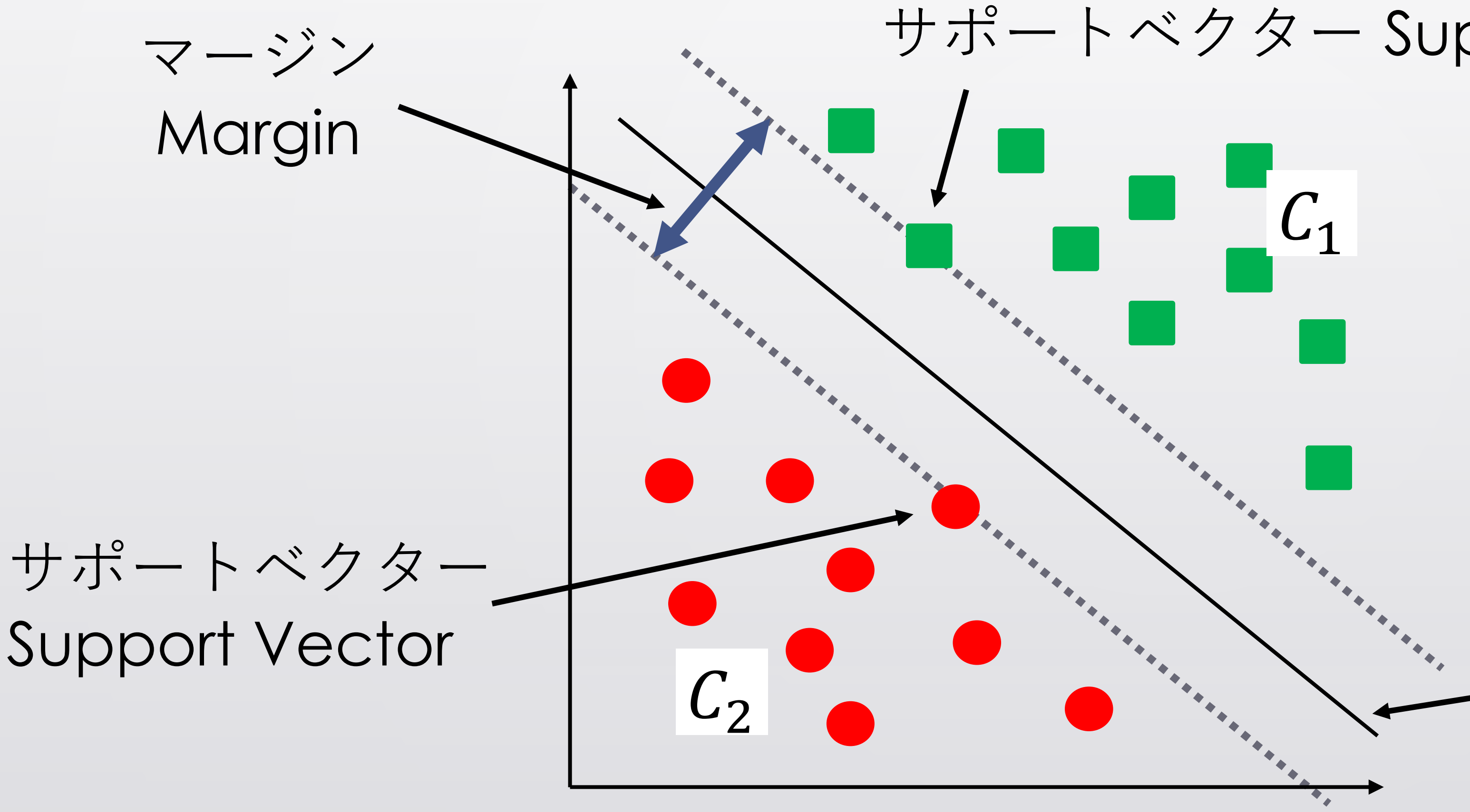
Data Mining

9: 分類④ Classification

土居 裕和 Hirokazu Doi

長岡技術科学大学 Nagaoka University of Technology

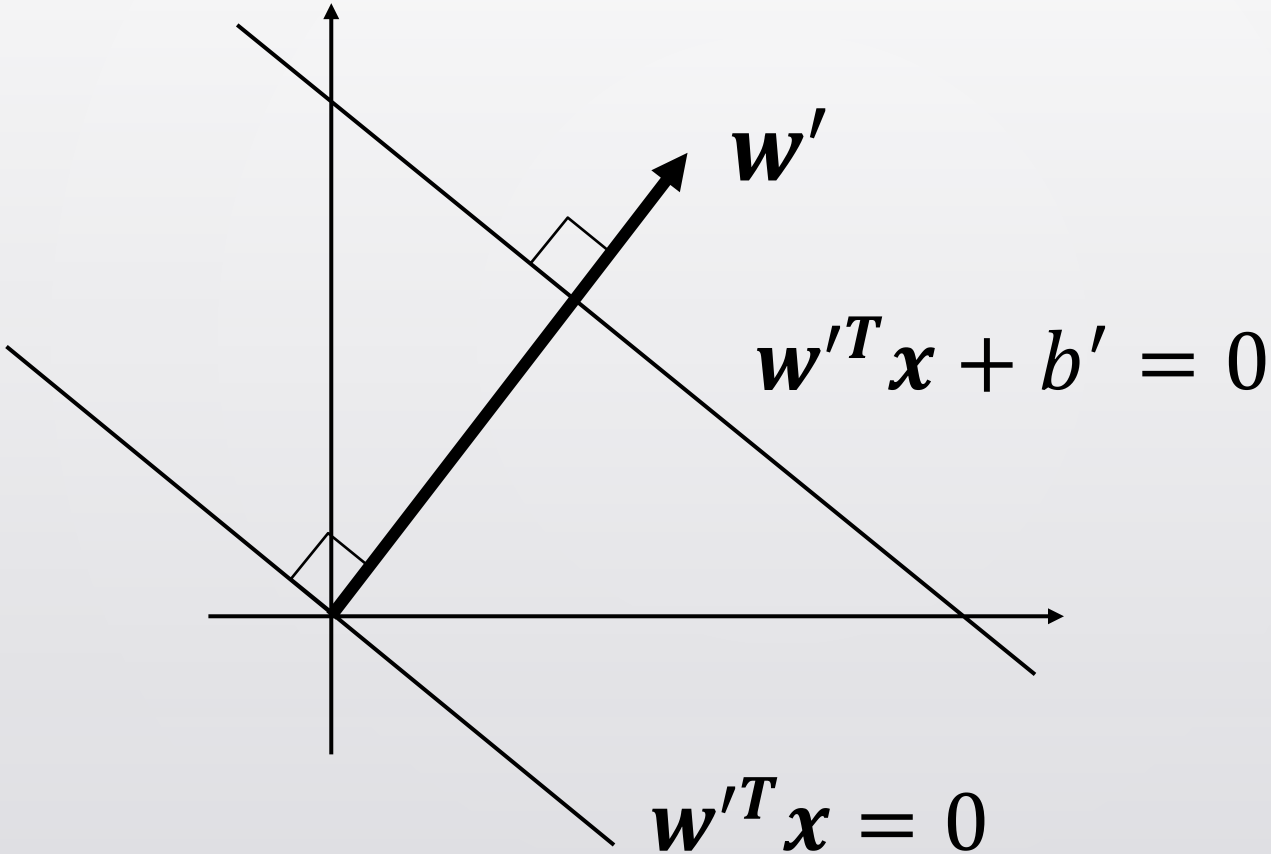
サポートベクターマシン Support Vector Machine



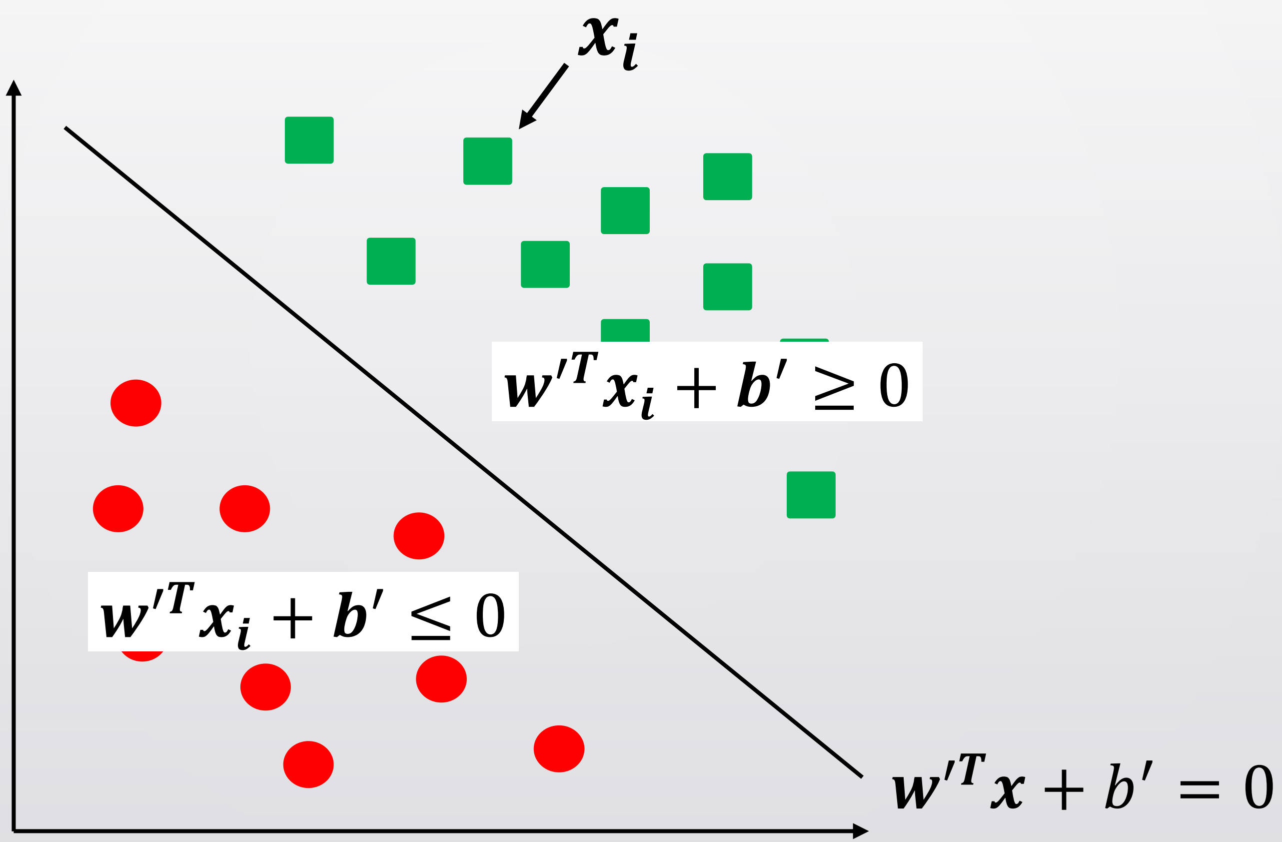
マージンを最大化するように決定境界を決定する

Draw decision boundary so that margin is maximized

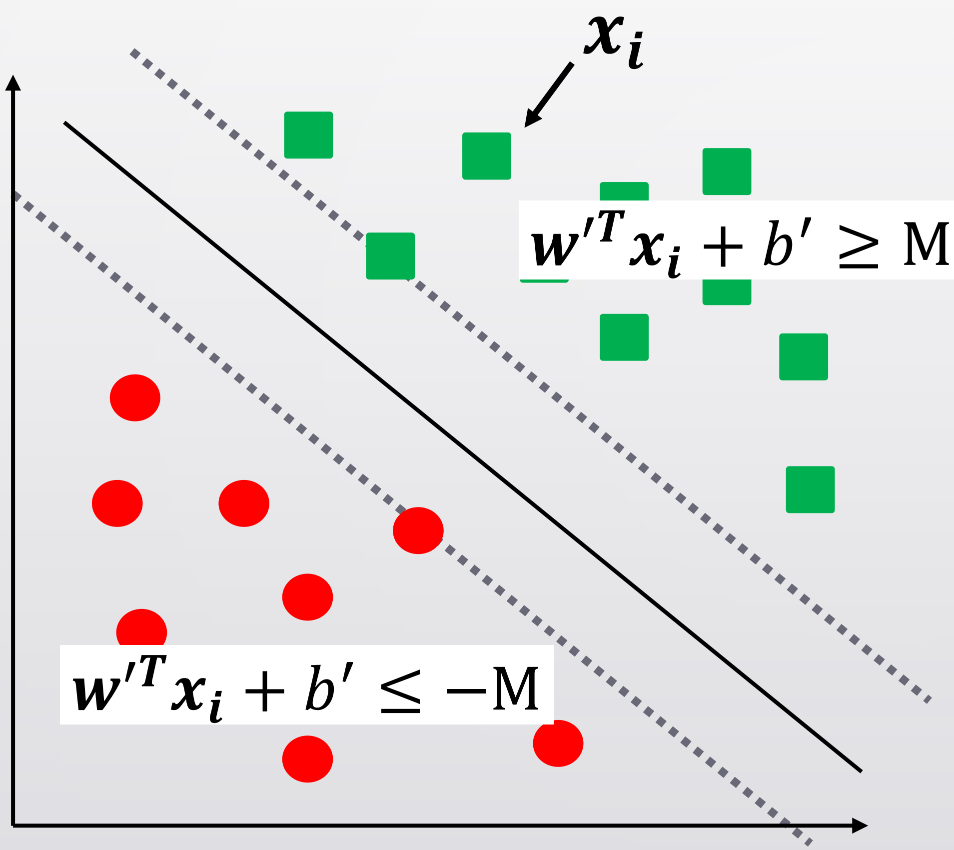
平面の方程式 Equation of a plane



サポートベクターマシン Support Vector Machine



サポートベクターマシン Support Vector Machine



$x_i \in C_1$の場合　$\boldsymbol{w'^T x_i} + b' \geq \mathrm{M}$

In case of $x_i \in C_1$

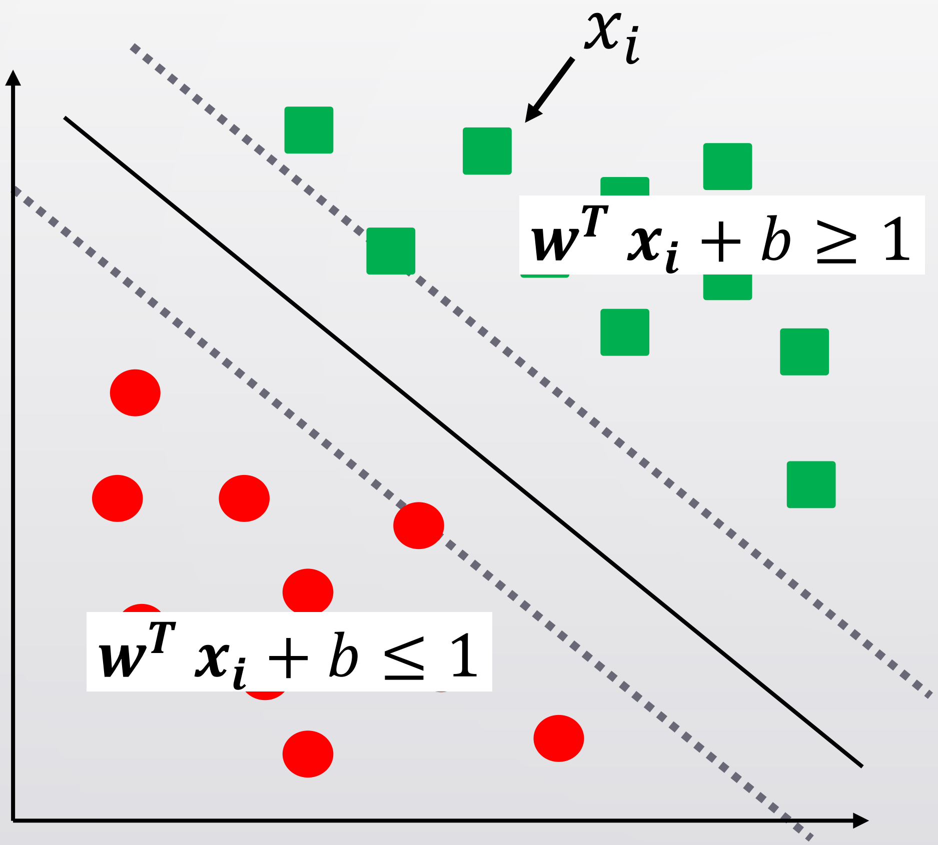
$x_i \in C_2$の場合　$\boldsymbol{w'^T x_i} + b' \leq -\mathrm{M}$

In case of $x_i \in C_2$

すべてのx_iに対して For all x_i

$$\left|\boldsymbol{w'^T x_i} + b'\right| \geq M$$

サポートベクターマシン Support Vector Machine



すべてのx_iに対して As for all x_i

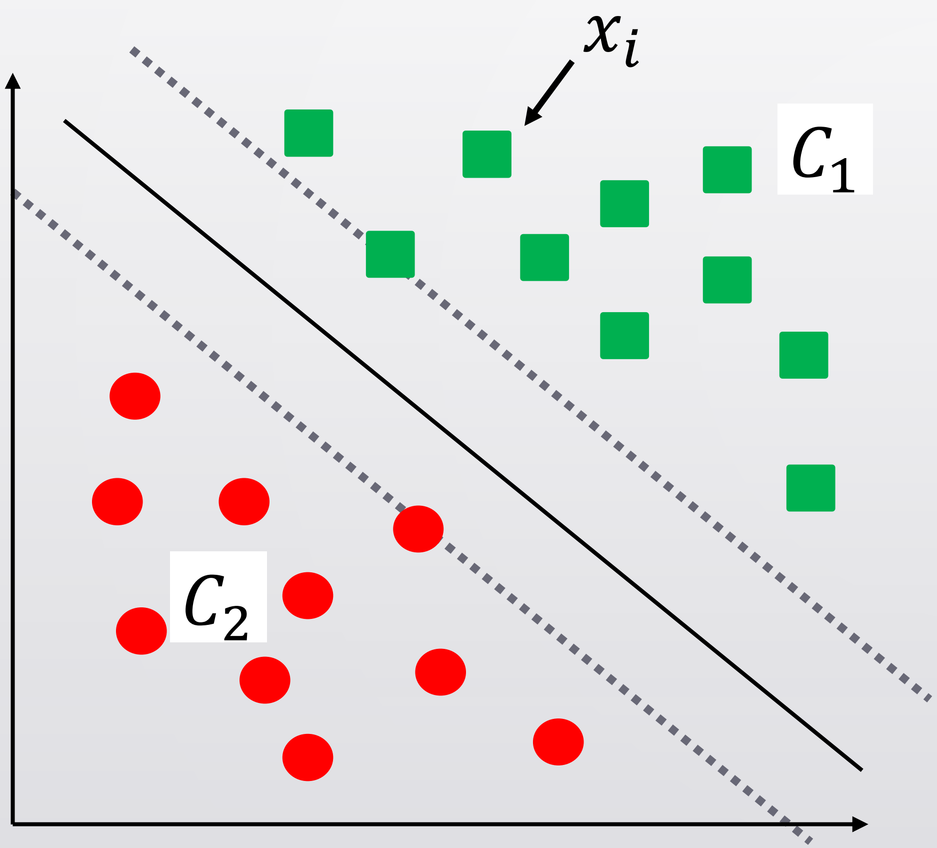
$$\left|\boldsymbol{w}'^T \boldsymbol{x}_i + b'\right| \geq M$$

$$\Downarrow$$

$$\left|\boldsymbol{w}^T \boldsymbol{x}_i + \boldsymbol{b}\right| \geq 1$$

$$\boldsymbol{w} = \frac{1}{M}\boldsymbol{w}' \qquad b = \frac{1}{M}b'$$

サポートベクターマシン Support Vector Machine



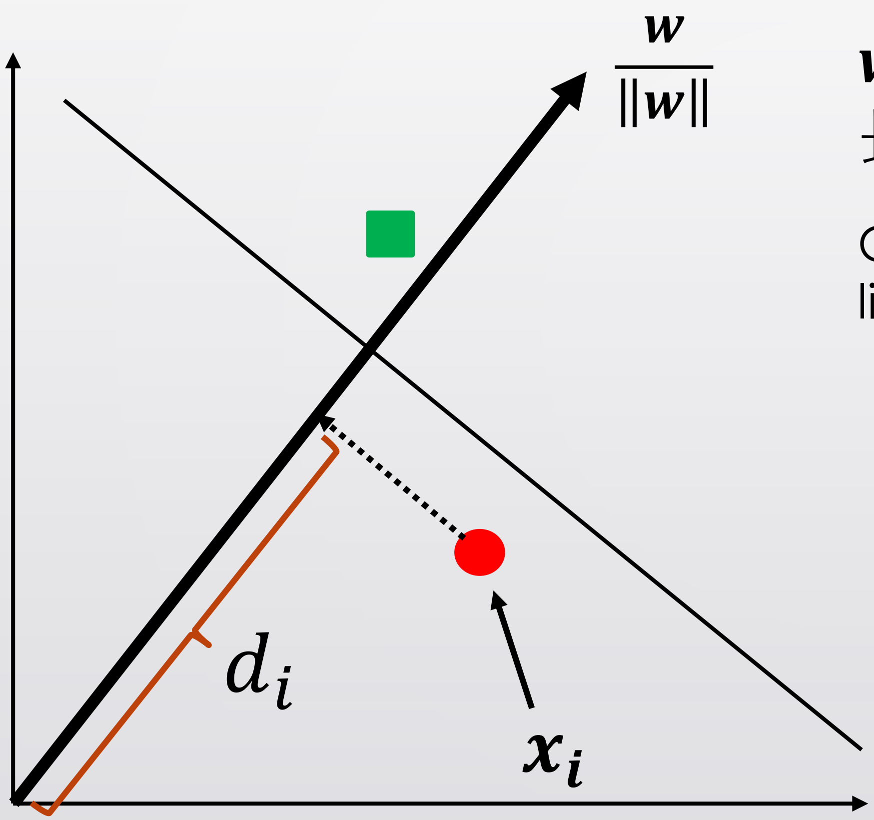
$x_i \in C_1$の場合 $\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b \geq 1$

In case of $x_i \in C_1$

$x_i \in C_2$の場合 $\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b \leq -1$

In case of $x_i \in C_2$

サポートベクターマシン Support Vector Machine

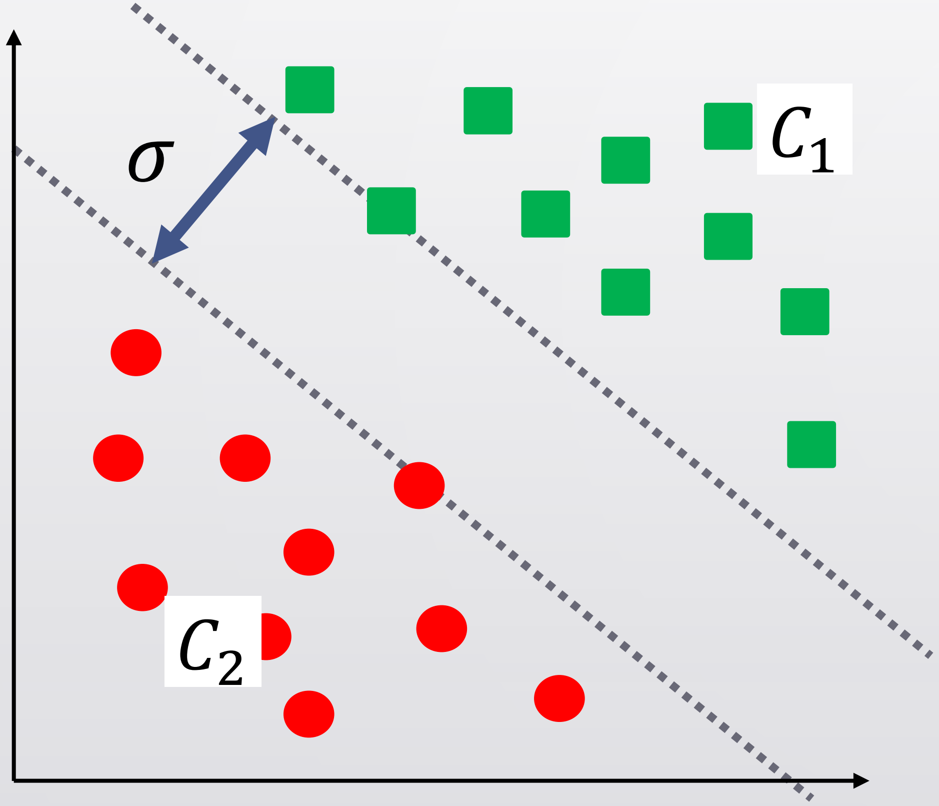


$\boldsymbol{w}$と平行な向きに対する$\boldsymbol{x}_i$の射影の長さd_iを計算する

Calculate the length d_i of projection of $\boldsymbol{x}_i$ onto the line parallel to $\boldsymbol{w}$

$$d_i = \frac{\boldsymbol{w}^T \boldsymbol{x}_i}{\|\boldsymbol{w}\|}$$

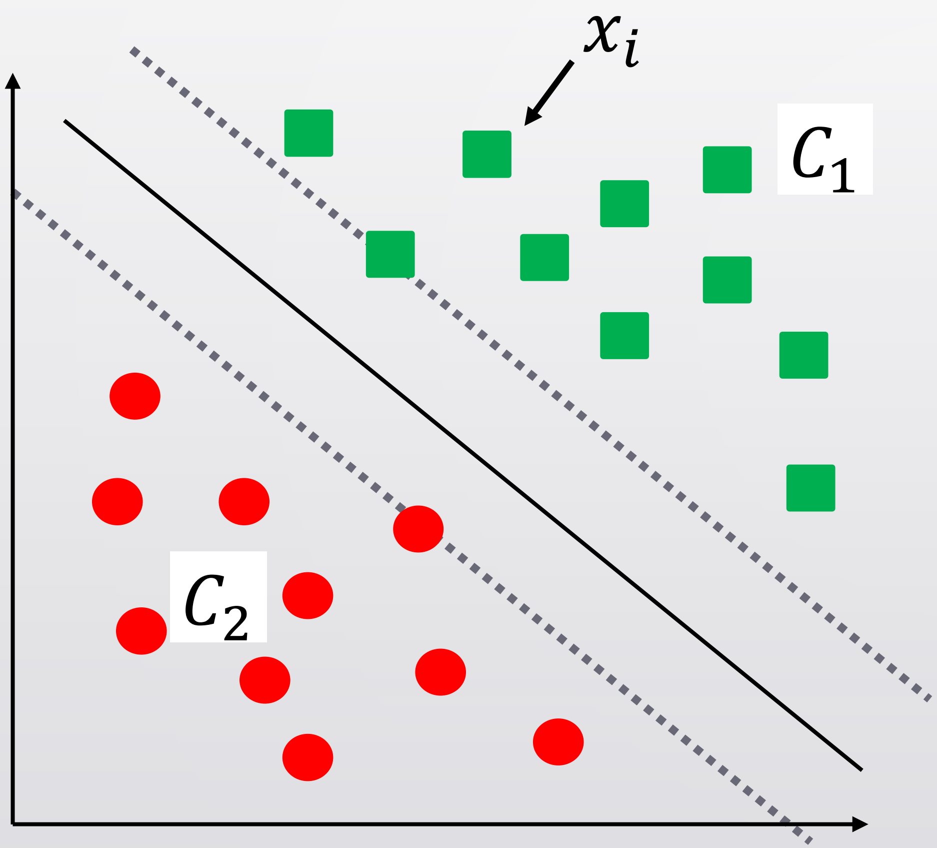
サポートベクターマシン Support Vector Machine



$$\sigma = min_{x_{i \in C_1}} d_i - max_{x_{i \in C_2}} d_i$$

$$= min_{x_{i \in C_1}} \frac{\boldsymbol{w}^T \boldsymbol{x}_i}{\|\boldsymbol{w}\|} - max_{x_{i \in C_2}} \frac{\boldsymbol{w}^T \boldsymbol{x}_i}{\|\boldsymbol{w}\|}$$

サポートベクターマシン Support Vector Machine



$x_i \in C_1$の場合 $\boldsymbol{w}^T\boldsymbol{x}_i + b \geq 1$

In case of $x_i \in C_1$

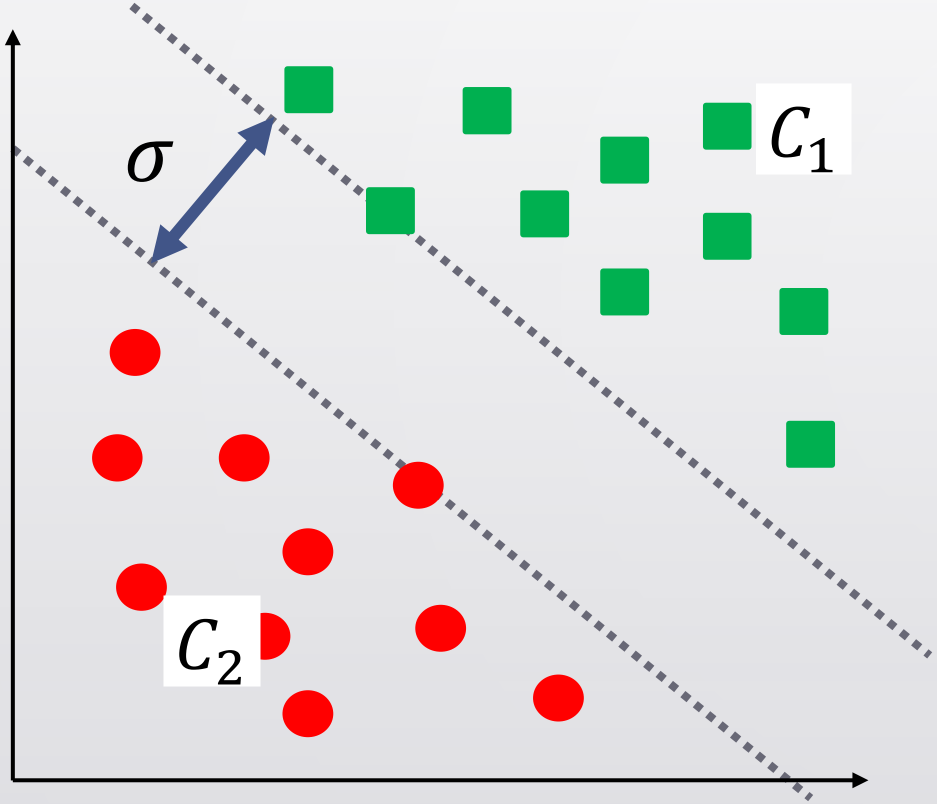
なので Then $min_{x_{i \in C_1}} \frac{\boldsymbol{w}^T\boldsymbol{x}_i}{\|\boldsymbol{w}\|} = \frac{1-b}{\|\boldsymbol{w}\|}$

$x_i \in C_2$の場合 $\boldsymbol{w}^T\boldsymbol{x}_i + b \leq -1$

In case of $x_i \in C_2$

なので Then $max_{x_{i \in C_2}} \frac{\boldsymbol{w}^T\boldsymbol{x}_i}{\|\boldsymbol{w}\|} = \frac{-1-b}{\|\boldsymbol{w}\|}$

サポートベクターマシン Support Vector Machine

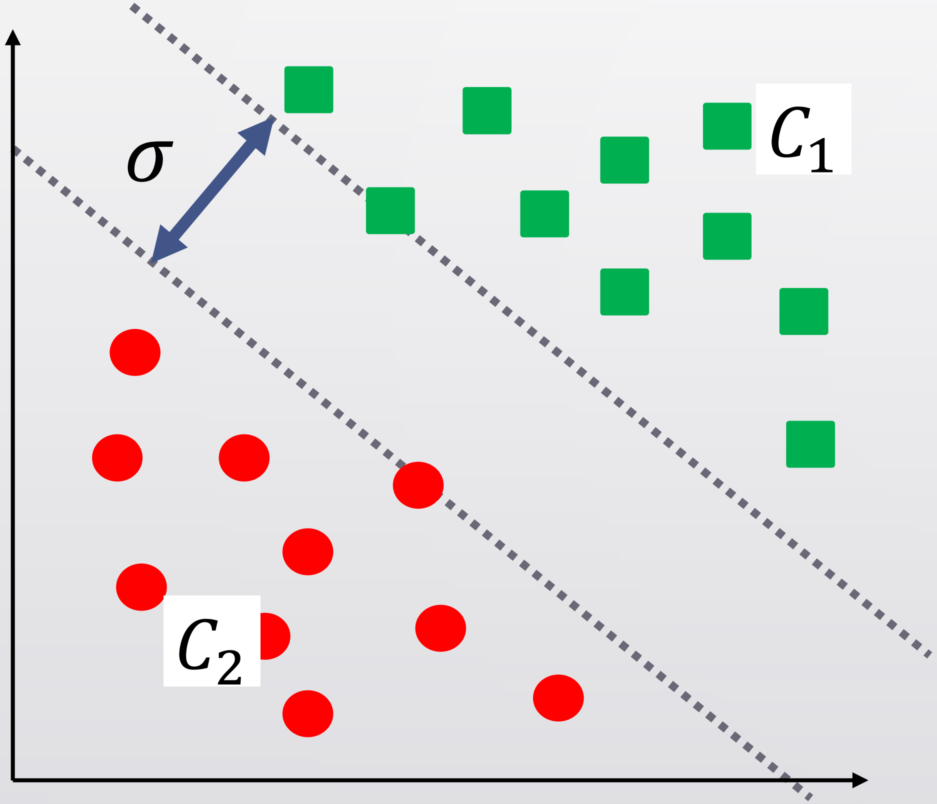


$$\sigma = min_{x_{i \in C_1}} \frac{\boldsymbol{w}^T \boldsymbol{x_i}}{\|\boldsymbol{w}\|} - max_{x_{i \in C_2}} \frac{\boldsymbol{w}^T \boldsymbol{x_i}}{\|\boldsymbol{w}\|}$$

$$= \frac{1 - b}{\|\boldsymbol{w}\|} - \frac{-1 - b}{\|\boldsymbol{w}\|}$$

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

サポートベクターマシン Support Vector Machine



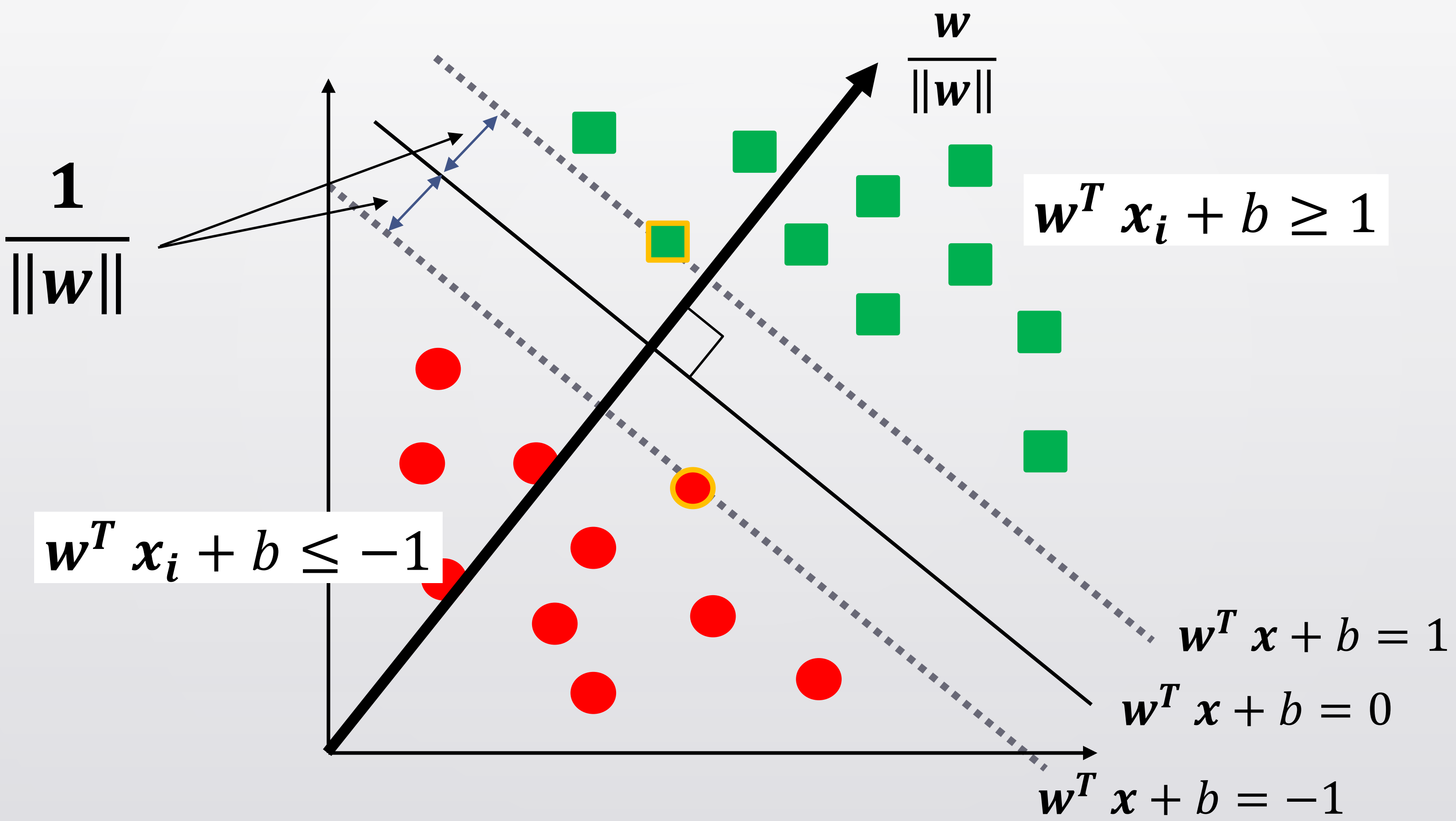
$\left|\boldsymbol{w}^T\boldsymbol{x}_i + \boldsymbol{b}\right| \geq 1$という制約の下で

Under the constraint that $\left|\boldsymbol{w}^T\boldsymbol{x}_i + \boldsymbol{b}\right| \geq 1$

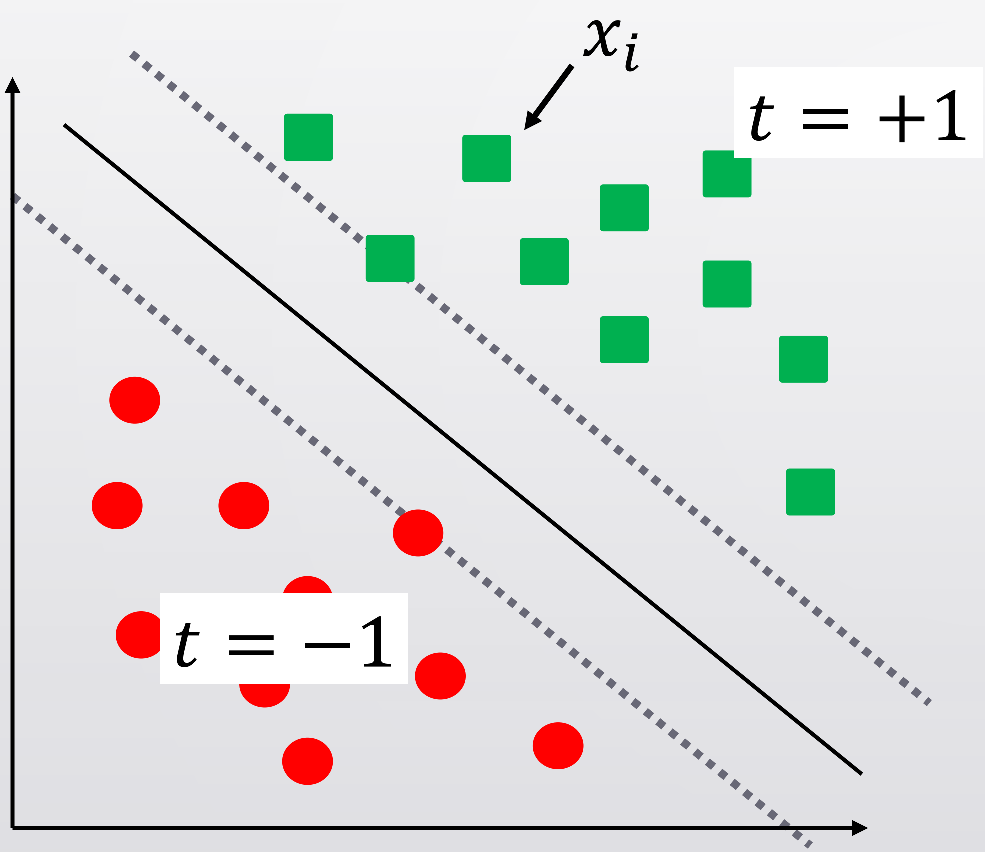
$\sigma = \frac{2}{\|\boldsymbol{w}\|}$ を最大化する Maximize $\sigma = \frac{2}{\|\boldsymbol{w}\|}$

$\|\boldsymbol{w}\|^2 = \boldsymbol{w}^T\boldsymbol{w}$を最小化する Minimize$\|\boldsymbol{w}\|^2 = \boldsymbol{w}^T\boldsymbol{w}$

サポートベクターマシン Support Vector Machine



サポートベクターマシン Support Vector Machine

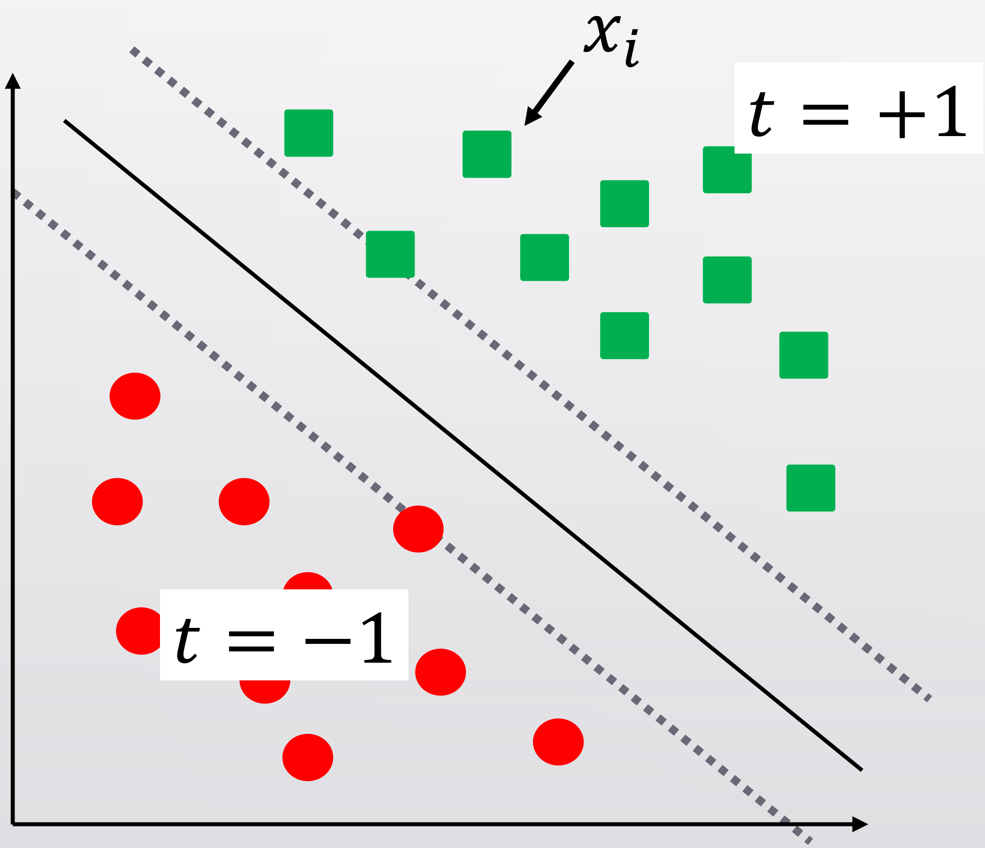


t_i: $\boldsymbol{x_i}$のクラスを表す変数

A variable representing the class of $\boldsymbol{x_i}$

$t_i = \{-1, +1\}$

サポートベクターマシン Support Vector Machine



$x_i \in C_1$の場合 $\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b \geq 1$

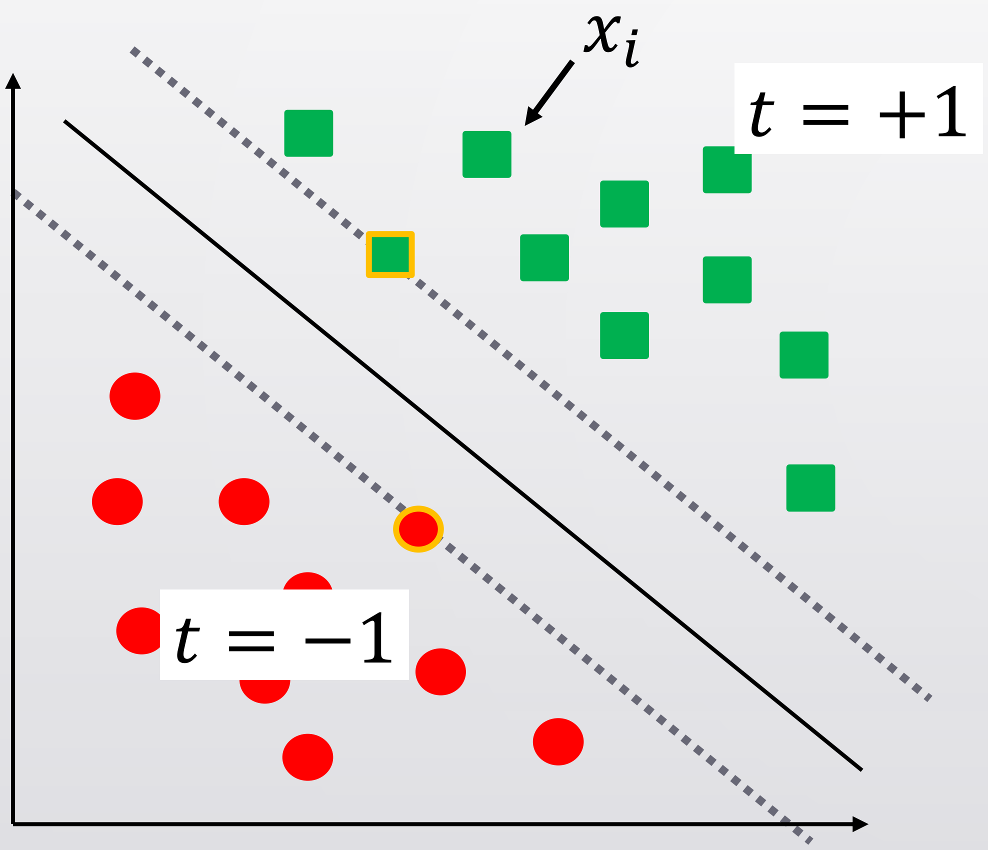
In case of $x_i \in C_1$

$x_i \in C_2$の場合 $\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b \leq -1$

In case of $x_i \in C_2$

$$t_i(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b) \geq 1$$

ハードマージンSVM Hard Margin SVM



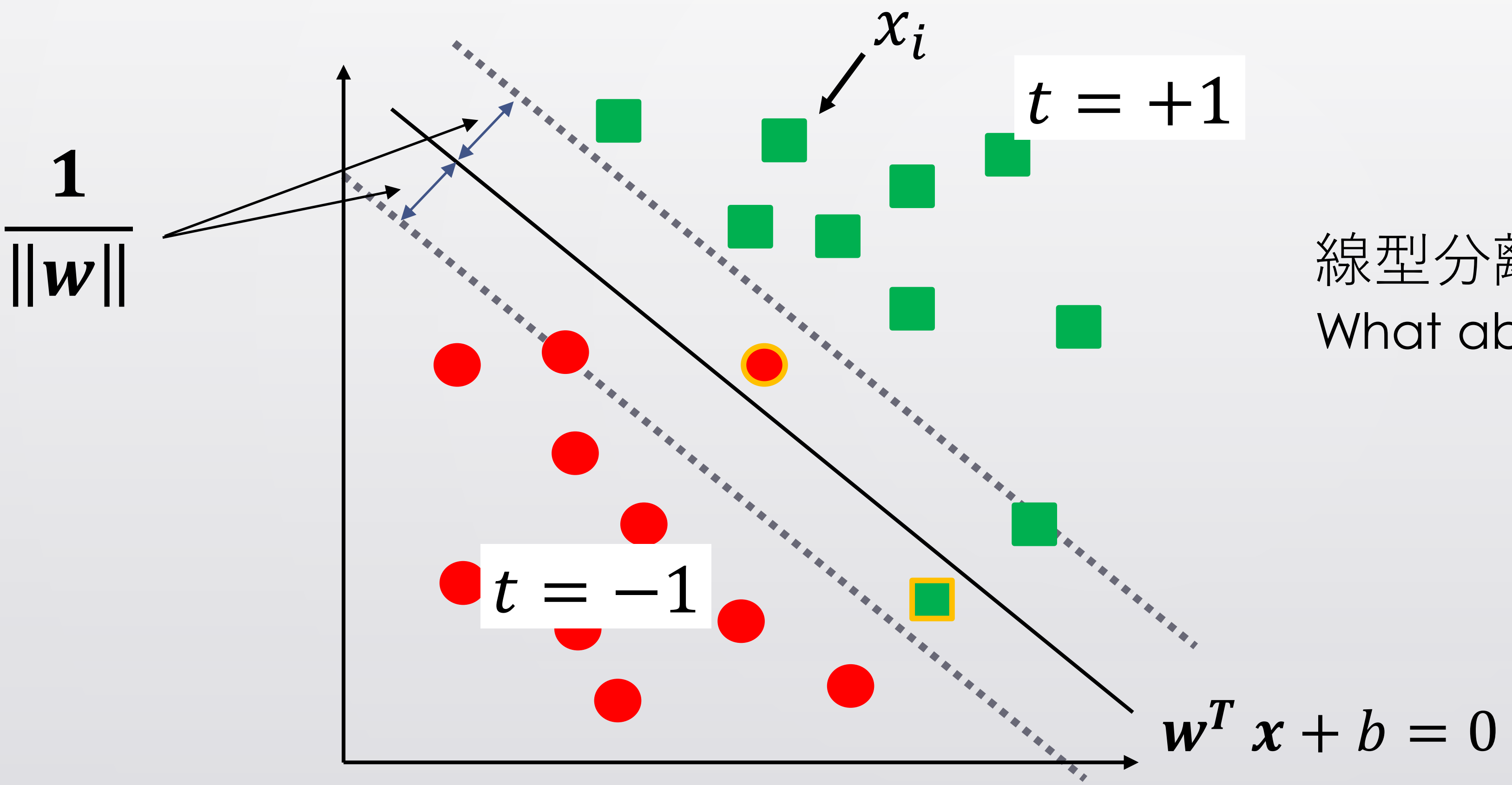
ハードマージンSVMは線型分離可能な問題に適用

Hard margin SVM is applicable to linearly separable data

$$\min_{\boldsymbol{w},\, b} \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$subject\ to \quad t_i\left(\boldsymbol{w}^T\boldsymbol{x_i} + b\right) \geq 1$$

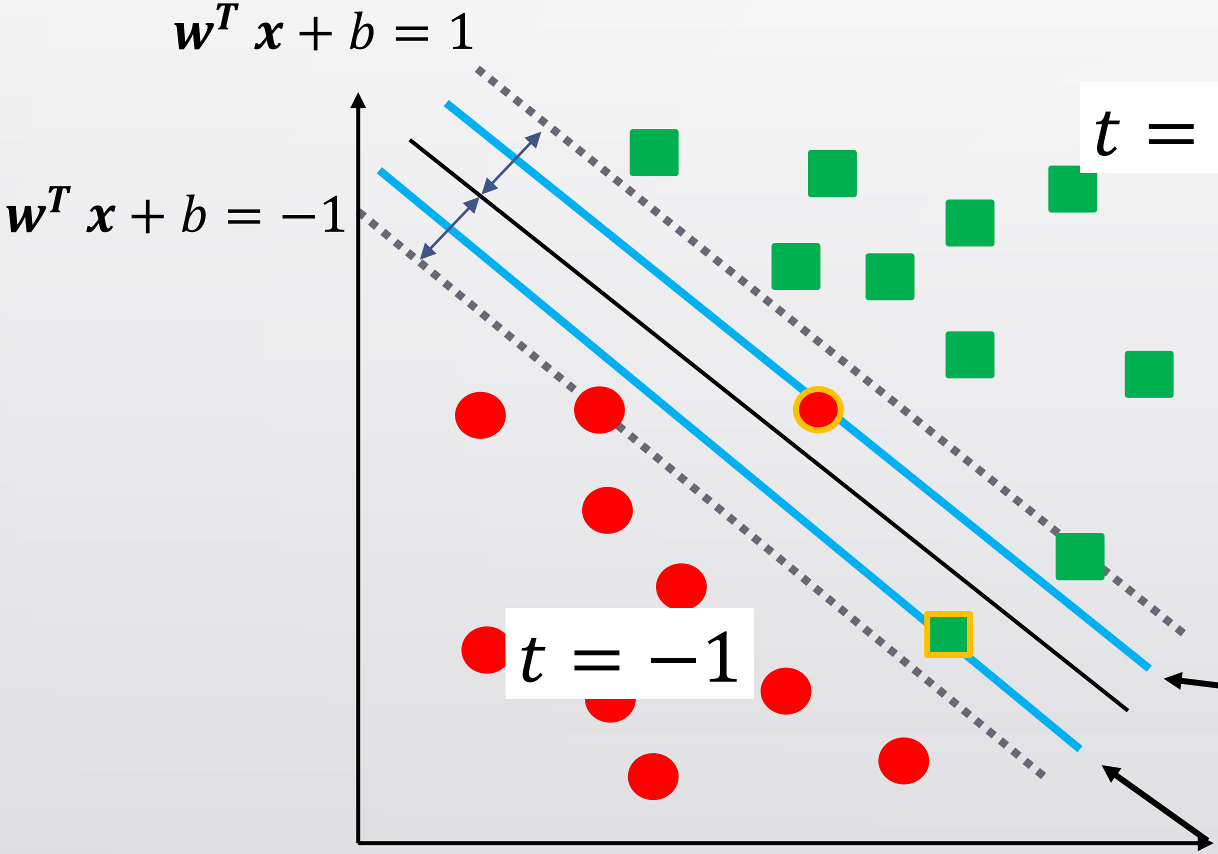
ソフトマージンSVM Soft Margin SVM



線型分離不可能な場合は？

What about linearly inseparable case?

ソフトマージンSVM Soft Margin SVM



データ$\boldsymbol{x_i}$がマージンの内側にξ_iだけはみだすことを許容する

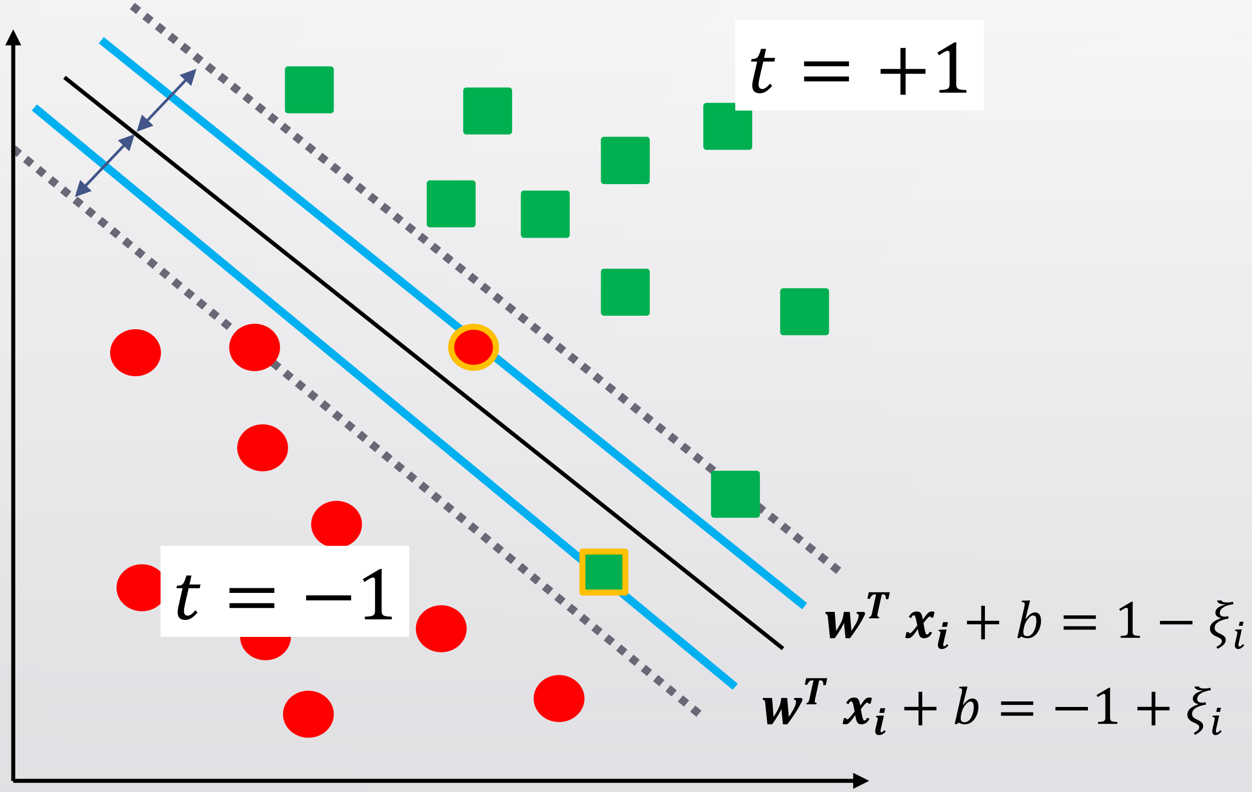
Data $\boldsymbol{x_i}$ is permitted to enter inside the margin by ξ_i

ξ_i: スラック変数 Slack variable

$$\boldsymbol{w}^T\boldsymbol{x_i}+b=1-\xi_i$$

$$\boldsymbol{w}^T\boldsymbol{x_i}+b=-1+\xi_i$$

サポートベクターマシン Support Vector Machine



$x_i \in C_1$の場合 $\boldsymbol{w}^T\boldsymbol{x_i} + b \geq 1 - \xi_i$

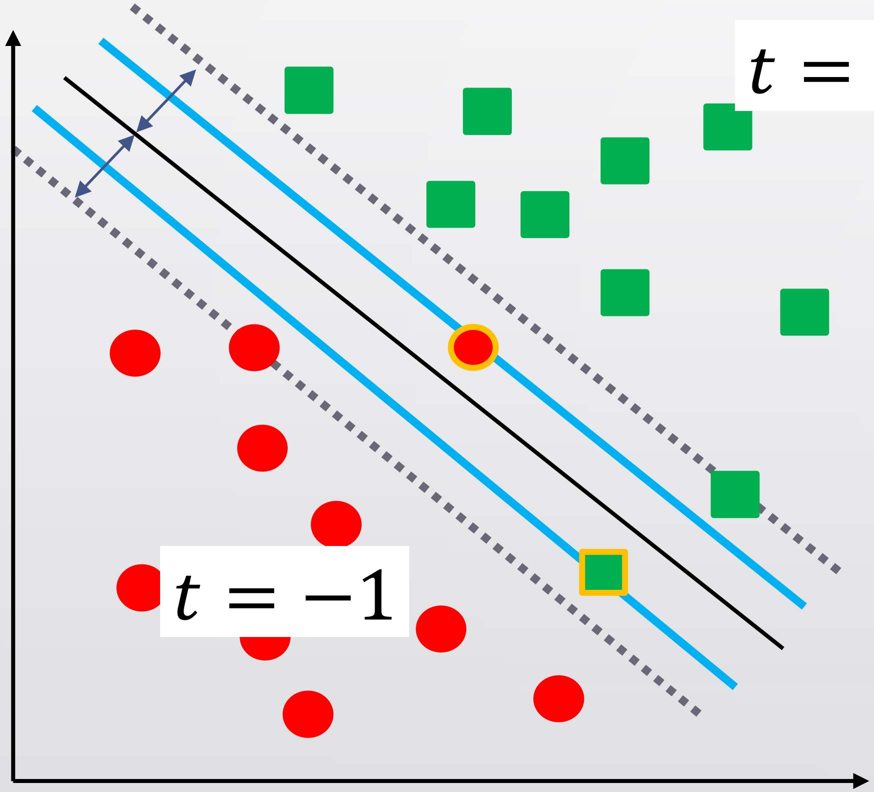
In case of $x_i \in C_1$

$x_i \in C_2$の場合 $\boldsymbol{w}^T\boldsymbol{x_i} + b \leq -1 + \xi_i$

In case of $x_i \in C_2$

$$t_i\left(\boldsymbol{w}^T\boldsymbol{x_i} + b\right) \geq 1 - \xi_i$$

ソフトSVM Soft Margin SVM



ソフトマージンSVMは線型分離不可能な問題に適用

Soft margin SVM is applicable to linearly inseparable data

$$\min_{\boldsymbol{w},\, b} \frac{1}{2} \|\boldsymbol{w}\|^2$$

$$\text{subject to} \quad t_i\left(\boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{x}_{\boldsymbol{i}} + b\right) \geq 1 - \xi_i$$

ハードマージンSVM Hard Margin SVM

<主問題> Primal Problem

$$\boxed{min_{\boldsymbol{w},\, b}\ \frac{1}{2}\|\boldsymbol{w}\|^2 \quad subject\ to \quad t_i\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b\right) \geq 1}$$

$$L(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{n} \alpha_i \left(1 - t_i\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b\right)\right)$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \quad \frac{\partial L}{\partial b} = 0 \quad \frac{\partial L}{\partial \boldsymbol{\alpha}} = 0$$

KKT条件を考慮

Take KKT condition into account

ハードマージンSVM Hard Margin SVM

<双対問題> Dual Problem

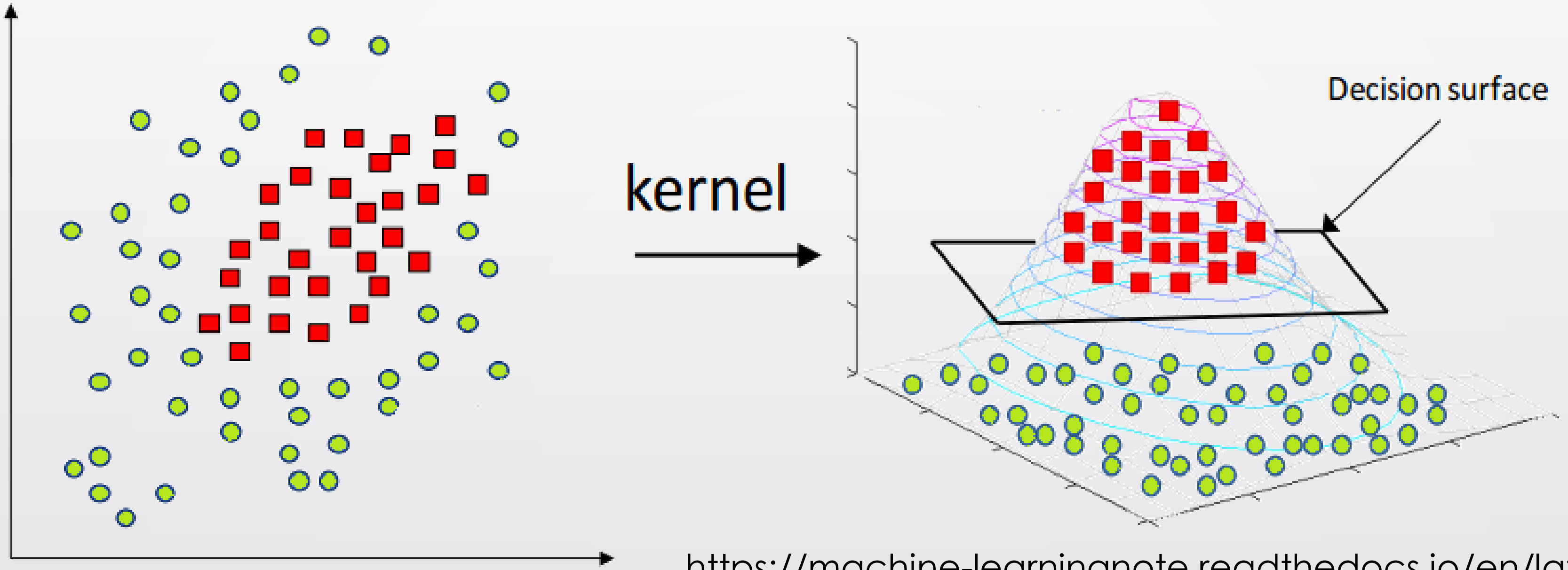
$$max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j t_i t_j \boldsymbol{x_i^T}\boldsymbol{x_j} + \sum_{i=1}^{n}\alpha_i$$

$$subject\ to\ \sum_{i=1}^{n}\alpha_i t_i = 0,\ \alpha_i \geq 0$$

αを求めると、それを用いて$\boldsymbol{w}$と$\boldsymbol{b}$の最適解が分かる

Optimal values of $\boldsymbol{w}$ and b are obtained based on $\boldsymbol{\alpha}$ meeting the constraints above

カーネル法 Kernel Methods



https://machine-learningnote.readthedocs.io/en/latest/algorithm/svm.html

データを高次元空間に写像することで、線型分離不可能な問題を線型分離可能にする

Transforming linearly inseparable problem to linearly separable one by mapping data to higher-dimensional space

基底関数 Basis Function

基底関数φはm次元データ$\boldsymbol{x_i}$を$p(p > m)$次元データ$\boldsymbol{\varphi}(\boldsymbol{x_i})$に変換する

Basis function φ transforms m-dimensional data $\boldsymbol{x_i}$ to p-dimensional data $\varphi(\boldsymbol{x_i})$

$$\boldsymbol{x_i} = (x_{i1}, x_{i2}, \cdots, x_{im})$$

$$\boldsymbol{\varphi}(\boldsymbol{x_i}) = \left(\varphi_1(\boldsymbol{x_i}), \varphi_2(\boldsymbol{x_i}), \cdots, \varphi_p(\boldsymbol{x_i})\right)$$

例 Example

$$\boldsymbol{x_i} = (x_{i1}, x_{i2})$$

$$\boldsymbol{\varphi}(\boldsymbol{x_i}) = \left(x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}, \sqrt{2c}x_{i1}, \sqrt{2c}x_{i2}, c\right)$$

基底関数 Basis Function

<双対問題>

$$max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j t_i t_j \boldsymbol{\varphi}(\boldsymbol{x_i})^{\boldsymbol{T}}\boldsymbol{\varphi}(\boldsymbol{x_j}) + \sum_{i=1}^{n}\alpha_i$$

$$subject\ to\ \sum_{i=1}^{n}\alpha_i t_i = 0,\ \alpha_i \geq 0$$

αを求めると、それを用いて$\boldsymbol{w}$と$\boldsymbol{b}$の最適解が分かる

Optimal values of $\boldsymbol{w}$ and b are obtained based on $\boldsymbol{\alpha}$ meeting the constraints above

カーネル関数 Kernel Function

$$max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j t_i t_j \boldsymbol{\varphi}(\boldsymbol{x_i})^{\boldsymbol{T}}\boldsymbol{\varphi}(\boldsymbol{x_j}) + \sum_{i=1}^{n}\alpha_i$$

$\boldsymbol{\varphi}(\boldsymbol{x_i})^{\boldsymbol{T}}\boldsymbol{\varphi}(\boldsymbol{x_j})$の計算はコストが大きい

A lot of resource is required to compute $\boldsymbol{\varphi}(\boldsymbol{x_i})^{\boldsymbol{T}}\boldsymbol{\varphi}(\boldsymbol{x_j})$

$$\underline{\boldsymbol{k}(\boldsymbol{x_i}, \boldsymbol{x_j})} = \boldsymbol{\varphi}(\boldsymbol{x_i})^{\boldsymbol{T}}\boldsymbol{\varphi}(\boldsymbol{x_j})$$

カーネル関数 Kernel Function

カーネル関数は, 基底関数$\boldsymbol{\varphi}$による写像の内積と一致する

Kernel function equals to the dot product of mappings by basis function $\boldsymbol{\varphi}$

多項式カーネル Polynomial Kernel

$$k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \gamma\left(x_i^T x_j + c\right)^d$$

<u>例 Example</u>

$$\boldsymbol{x_i} = (x_{i1}, x_{i2}) \quad \gamma = 1, d = 2$$

$$\boldsymbol{\varphi}(\boldsymbol{x_i}) = \left(x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}, \sqrt{2c}x_{i1}, \sqrt{2c}x_{i2}, c\right)$$

$$k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \left(\boldsymbol{x_i^T} \boldsymbol{x_j} + c\right)^2$$

$$= \left(x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}, \sqrt{2c}x_{i1}, \sqrt{2c}x_{i2}, c\right)^T \left(x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}, \sqrt{2c}x_{i1}, \sqrt{2c}x_{i2}, c\right)$$

$$= \boldsymbol{\varphi}(\boldsymbol{x_i})^{\boldsymbol{T}} \boldsymbol{\varphi}(\boldsymbol{x_j})$$

カーネルトリック Kernel Trick

基底関数をカーネル関数に置き換えることで、少ない計算コストで双対問題を解ける

Dual problem can be solved with reduced computational cost by replacing basis function with kernel function

$$max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j t_i t_j \boldsymbol{\varphi}(\boldsymbol{x_i})^{\boldsymbol{T}}\boldsymbol{\varphi}(\boldsymbol{x_j}) + \sum_{i=1}^{n}\alpha_i$$

$$= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j t_i t_j k(\boldsymbol{x_i}, \boldsymbol{x_j}) + \sum_{i=1}^{n}\alpha_i \quad = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j t_i t_j (\boldsymbol{x_i^T}\boldsymbol{x_j} + c)^2 + \sum_{i=1}^{n}\alpha_i$$

その他のカーネル Other Kernels

- linear $k(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \cdot \mathbf{x}_2$
- polynomial $k(\mathbf{x}_1, \mathbf{x}_2) = (\gamma\, \mathbf{x}_1 \cdot \mathbf{x}_2 + c)^d$
- Gaussian or radial basis $k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\gamma \left\|\mathbf{x}_1 - \mathbf{x}_2\right\|^2\right)$
- sigmoid $k(\mathbf{x}_1, \mathbf{x}_2) = \tanh(\gamma\, \mathbf{x}_1 \cdot \mathbf{x}_2 + c)$

https://www.analyticsvidhya.com/blog/2021/07/svm-support-vector-machine-algorithm/