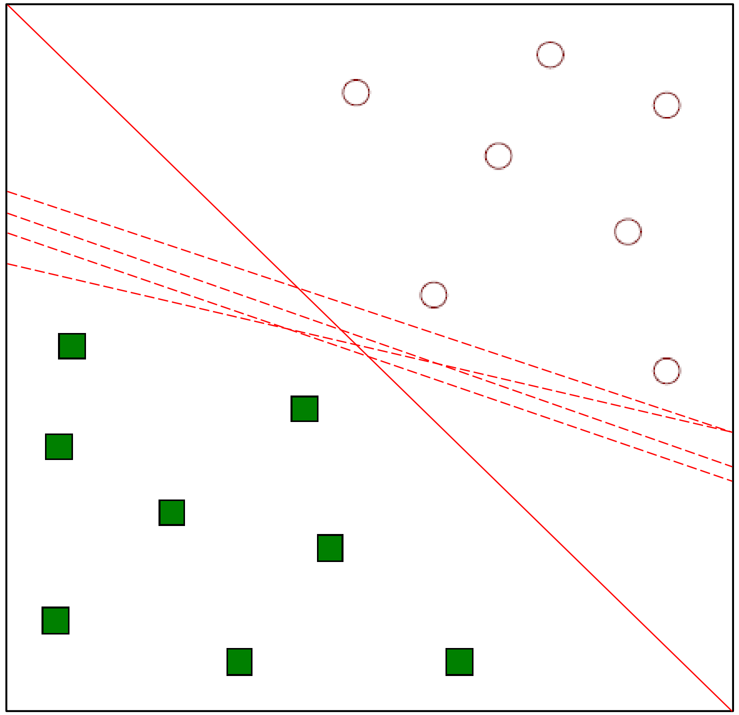
Data Mining

2017. 4. 20.

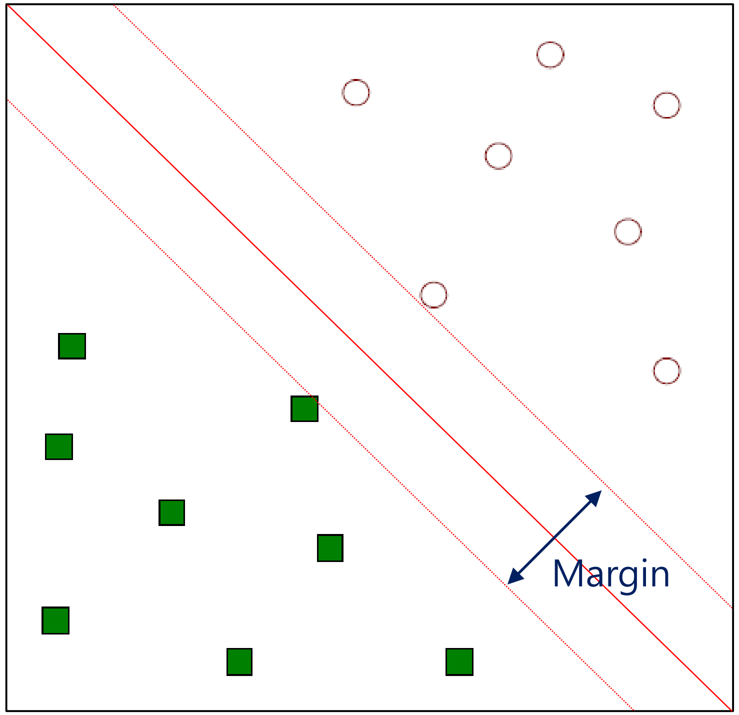
김영철

Support Vector Machines



두 클래스를 나누는 직선은 많다.
어떤 선을 가장 좋다고 할 것인가?

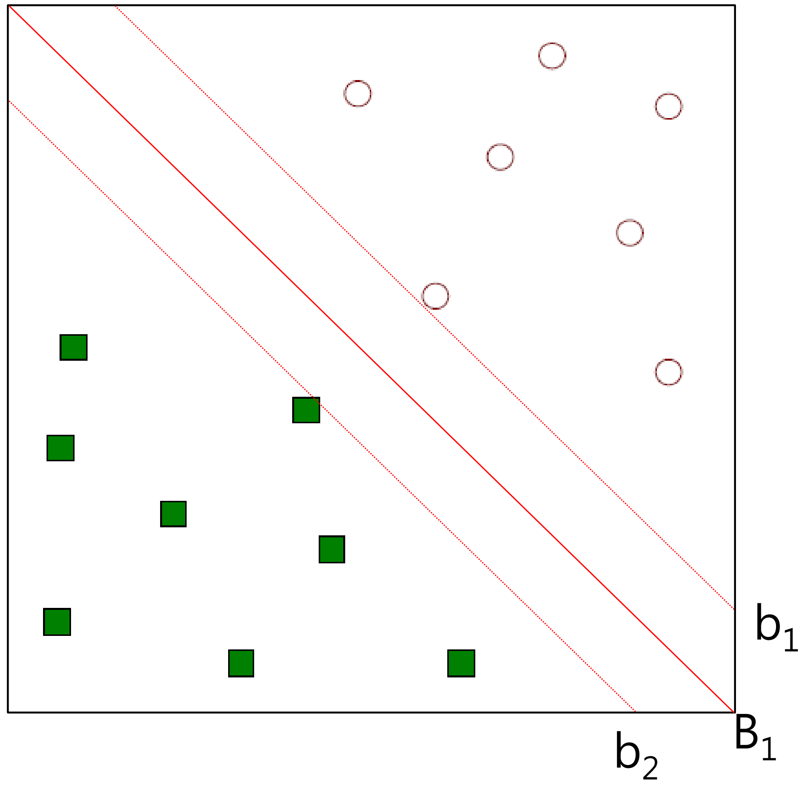
Support Vector Machines



두 클래스를 나누는 직선은 많다.
어떤 선을 가장 좋다고 할 것인가?

Margin이 최대가 되는 직선

Support Vector Machines



$$B_1 \Rightarrow w \bullet x + b = 0$$

$$b_1 \Rightarrow w \bullet x + b = 1$$

$$b_2 \Rightarrow w \bullet x + b = -1$$

w, x는 벡터

$$f(x) \Rightarrow 1 \quad if\ w \bullet x + b \geq 1$$
$$f(x) \Rightarrow -1 \quad if\ w \bullet x + b \leq -1$$

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

Support Vector Machines

$f(x) \Rightarrow 1 \quad if\ w\bullet x + b \geq 1$
$f(x) \Rightarrow -1 \quad if\ w\bullet x + b \leq -1$

$Margin = \frac{2}{|w|}$ ➔ 최대로 만드는 w 찾기

(1) $L(w) = \frac{|w|^2}{2}$ ➔ 최소로 만드는 w 찾기, 미분을 편하게 하기 위해 w를 제곱함

(2) $y_i(w\bullet x_i + b) \geq 1$

Support Vector Machines

$$L = f(\mathbf{x}) + \sum_{i=1}^{q} \lambda_i h_i(\mathbf{x})$$

(1) $L(w) = \frac{|w|^2}{2}$ ➔ f(x)

(2) $y_i(w \bullet x_i + b) \geq 1$ ➔ $1 - y_i(w \bullet xi + b) \leq 0$ ➔ $h_i(x)$

$$\frac{1}{2}\|w\|^2 - \sum_i \lambda_i (y_i (\vec{w} \bullet \vec{x}_i + b) - 1)$$

➔ $$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}$$

Support Vector Machines

$$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}$$

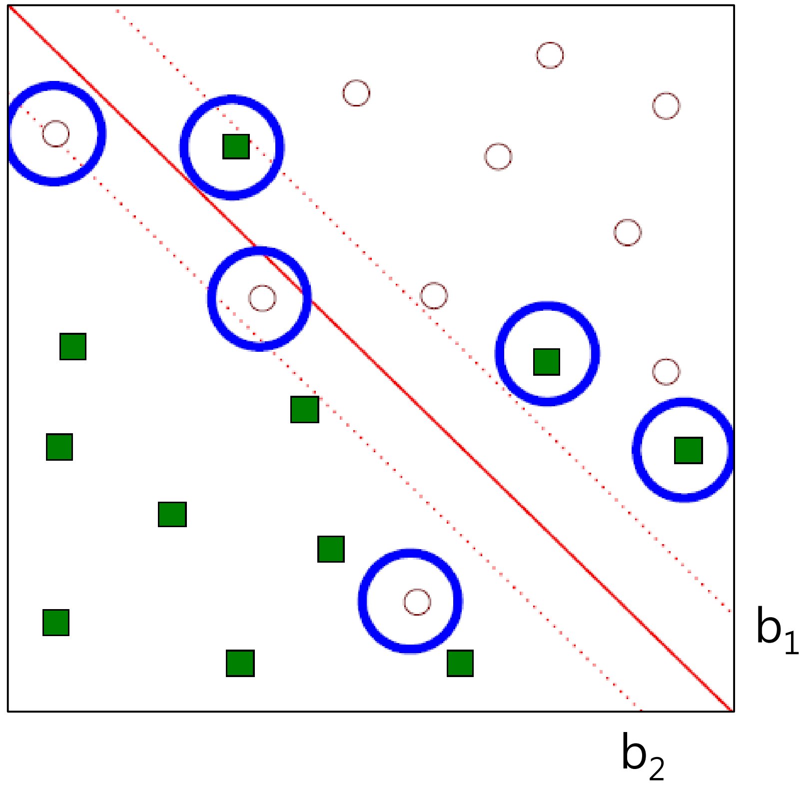
Support vectors

- x_i for nonzero λ_I *lies along the hyperplanes*

$$w = \sum_i \lambda_i y_i x_i \quad \textit{and find b using} \quad y_i(\vec{w} \bullet \vec{x}_i + b) = 1$$

if $\vec{w} \bullet \vec{x} + b > 0$ then assign to class 1

Support Vector Machines



선형으로 분리되지 않는 경우에는?
slack variable을 이용하여 에러를 허용

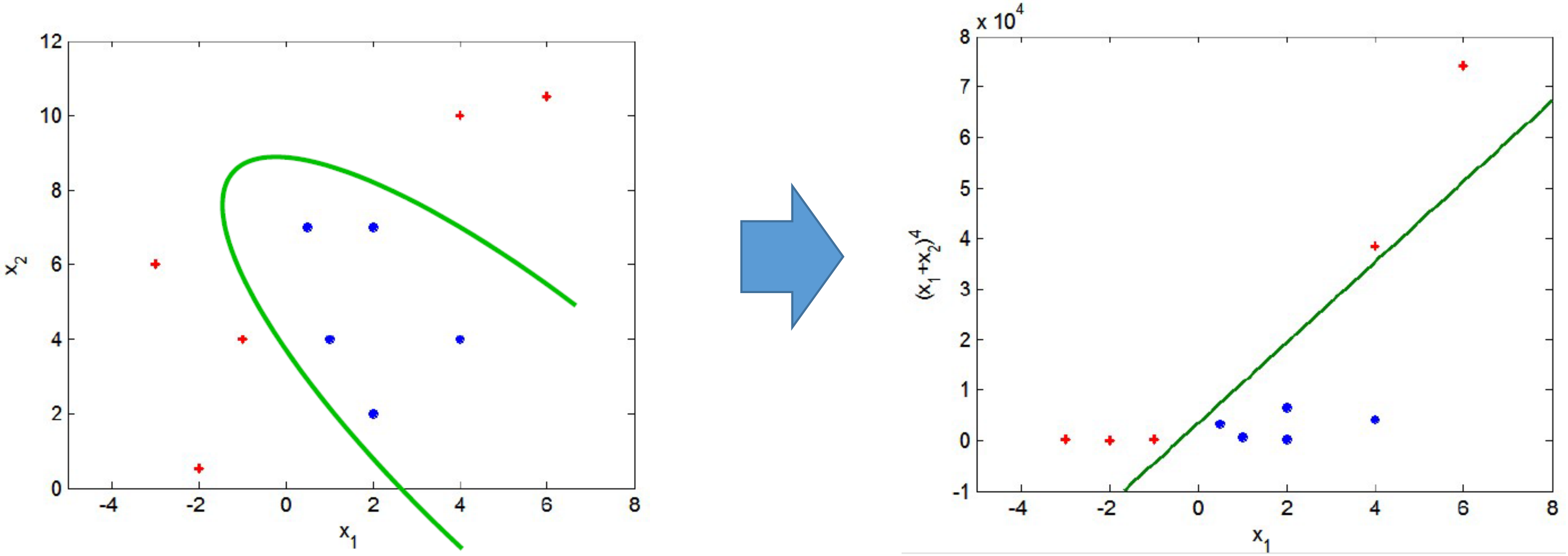
$b_1 \Rightarrow w \bullet x + b \geq 1 - \varepsilon_i$ (y_i=1)

$b_2 \Rightarrow w \bullet x + b \leq -1 + \varepsilon_i$ (y_i=-1)

Nonlinear Support Vector Machines

decision boundary가 그림과 같이 non linear하다면?

데이터를 더 높은 차원의 데이터로 변형



Kernel Method

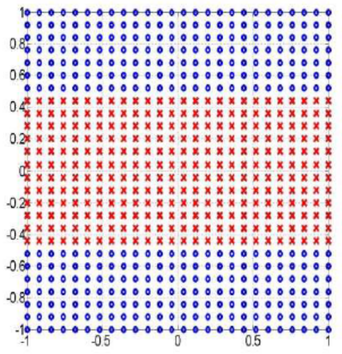
$$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}$$

$$\sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

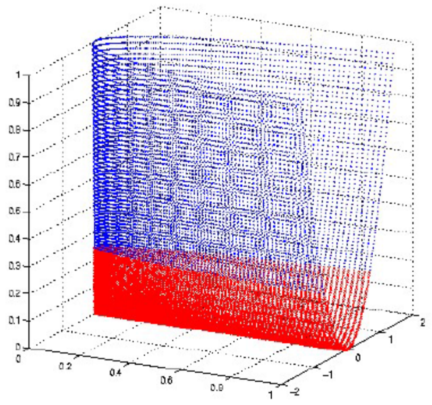
Kernel Method

kernel function k(x,y)가 Mercer's condition을 만족하면 <m(x), m(y)> = k(x,y)인 mapping m이 존재한다.

Kernel Method



$$\Phi(x_1,x_2)=(x_1^2,\sqrt{2}x_1x_2,x_2^2)$$



$$<\Phi(x),\Phi(y)> = k(x,y) = <x,y>^2$$

Gaussian kernel $k(x,y) = \exp(-\|x-y\|^2 / \sigma)$

Polynomial kernel $k(x,y) = (\alpha< x, y >+\beta)^d$

Hyperbolic tangent kernel $k(x,y) = \tanh(k< x, y >- \delta)$

http://blog.daum.net/_blog/BlogTypeView.do?blogid=0YW8F&articleno=136&categoryId=0&regdt=20160424213557