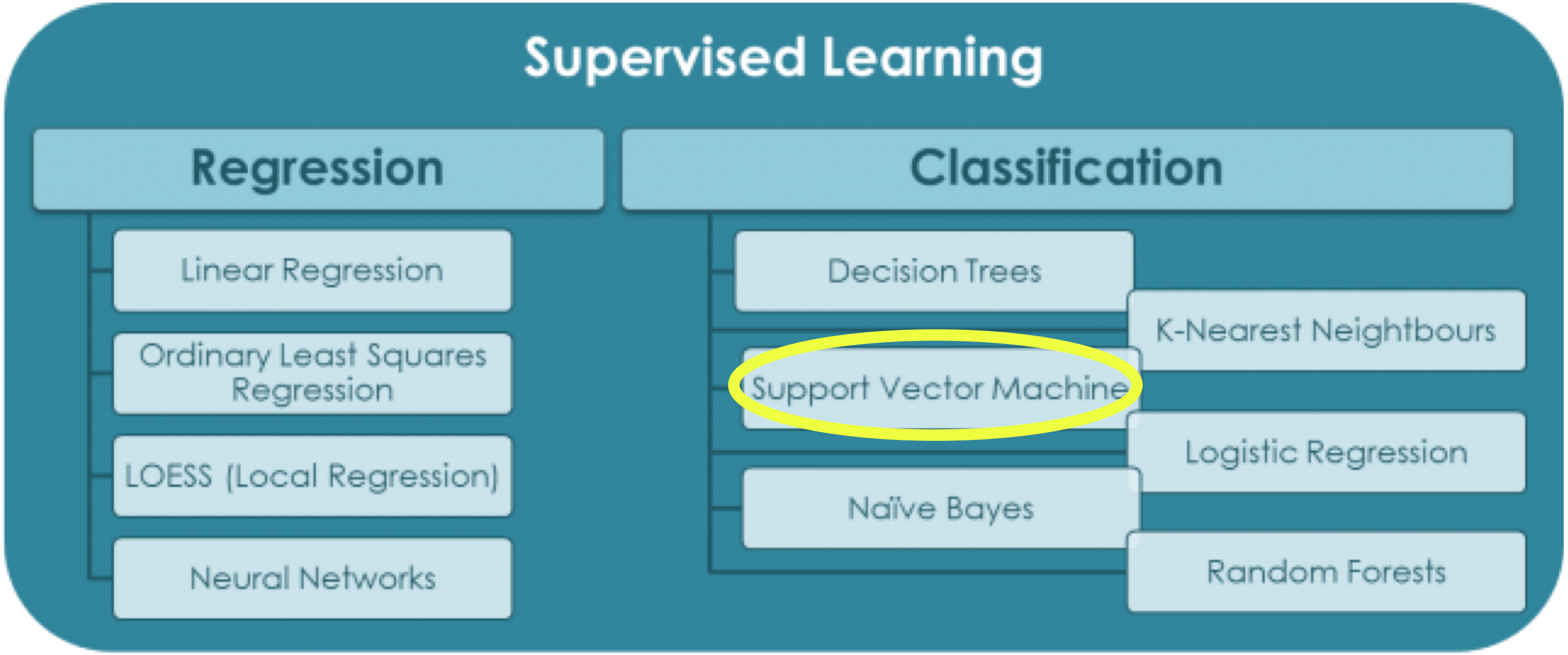
Ⅲ. 지도학습 (Supervised Learning)

- Classification

❷ SVM


Supervised Learning
Regression
Classification
Linear Regression
Ordinary Least Squares Regression
LOESS (Local Regression)
Neural Networks
Decision Trees
K-Nearest Neighbours
Support Vector Machine
Logistic Regression
Naïve Bayes
Random Forests

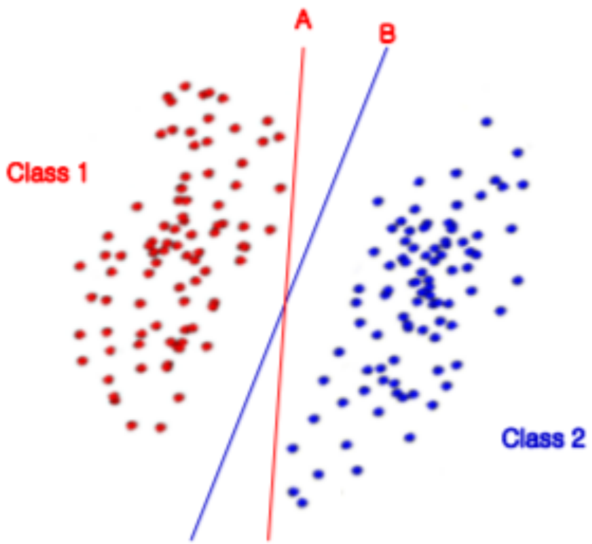
SVM 알고리즘의 기본 아이디어

- 분리 (Classification) 문제는 여러 클래스 간의 경계를 특정하는 문제로 볼 수 있음.
- 경계는 선형일 수도 있고 비선형일 수도 있음.
- 다차원 공간의 경우는 경계를 직관적으로 이해하기 어려움.

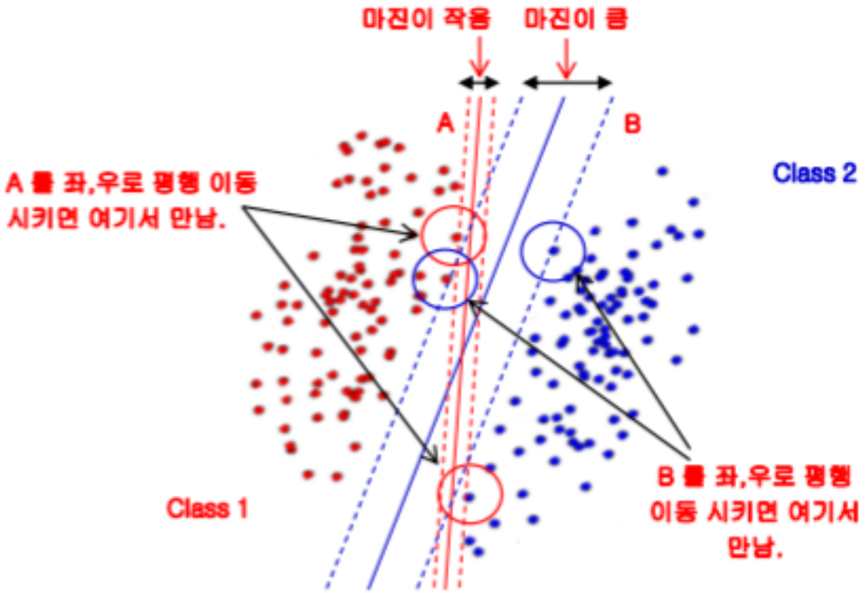
-> 수학적으로 판단함.

SVM 알고리즘의 기본 아이디어

- 왼쪽 그림에서 클래스 1 과 클래스 2를 구분하는 직선 2개 (A 와 B) 를 그려 보면, 두 직선은 훈련 데이터의 클래스를 완벽하게 구분하고 있음. 그러나 두 직선이 시험 데이터도 완벽하게 구분한다고 볼 수는 없음. -> 일반화 특성이 달라짐.
- 직관적으로 보면 직선 B 가 더 유용해 보임. 그 이유는 클래스 1쪽 과도 무리가 없어 보이고, 2쪽 과도 무리가 없어 보임.

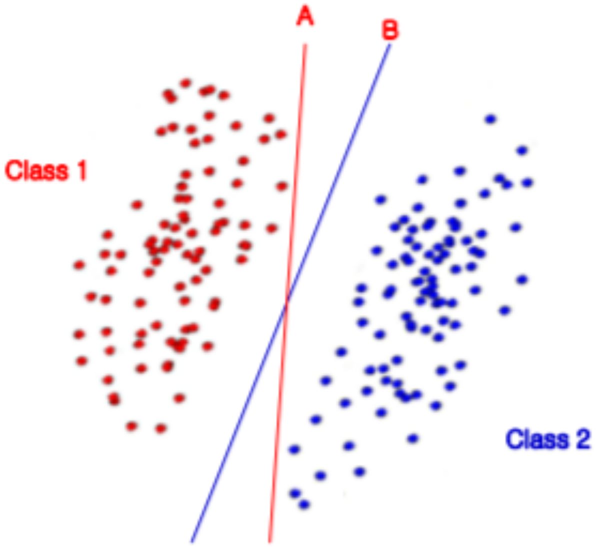


선형 분리

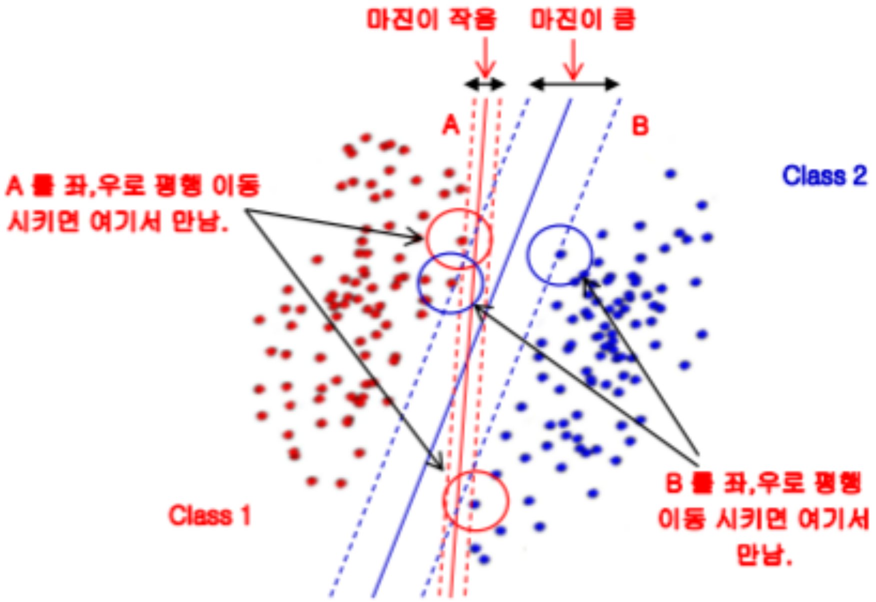


SVM 알고리즘의 기본 아이디어

- 두 직선을 오른쪽과 왼쪽으로 각 클래스의 점과 만날 때까지 각각 평행 이동 시켜 보면 B가 A보다 이동 폭이 큼.
- 이동 폭을 마진 (Margin) 이라 하고, **최대 마진을 갖는 경계를 찾는 것**이 **SVM 알고리즘의 기본 아이디어**임.
- 최대 마진 (초평면)으로 경계를 생성하면 과잉 적합 (Overfitting) 문제가 작고, 일반화 특성이 좋아짐.



선형 분리



SVM 알고리즘 이란?

- 2개의 범주를 분류하는 이진 분류기
- 다음 그림은 SVM의 개념을 설명하는 것이다. feature들은 그림과 같은 vector공간에 vector로 표시된다. 그림에서 보는 것처럼 하얀 색 vector들을 A그룹에 속하는 white point라고 하고, 그 반대로 검은색 vector들을 B그룹에 속하는 black point라고 하자.

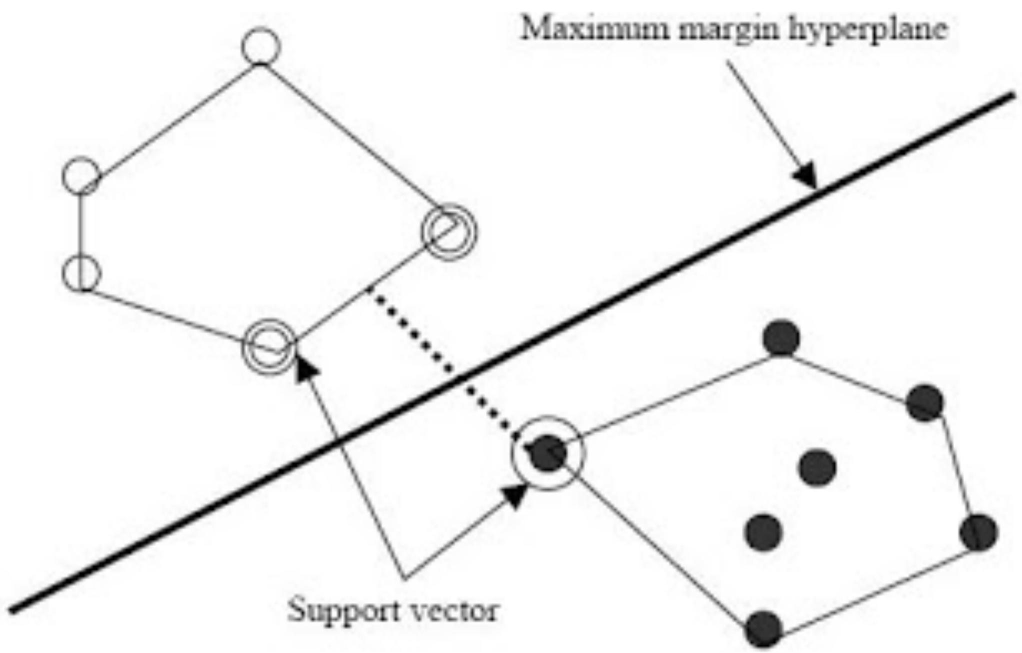


그림 4. SVM의 범주 분류

SVM 알고리즘 이란?

- 이러한 벡터 가운데 같은 범주를 기준으로 바깥으로 위치한 벡터들의 연결선으로 이루어진 닫혀진 다각형을 convex hull이라고 한다. convex hull안의 벡터들은 그룹을 분류하는 데 그다지 큰 영향을 미치지 않는다. 그룹을 분류하는데 가장 큰 영향을 미치는 것들은 바깥에 위치한 벡터들이다. 그룹을 분류하는 선, 면을 hyperplane이라고 한다.
- 그림에서 보는 것처럼 그룹을 나눌 수 있는 hyperplane은 무수히 많다.
  하지만, 직관적으로 그룹들의 convex hull에 속한 벡터들 중 가장 가까운 벡터와 수직거리로 가장 먼 거리를 가진 hyperplane이 2그룹을 효과적으로 분류할 것이다.
- 이러한 hyperplane을 maximum hyperplane이라고 부르고 이때 가장 가까운 벡터들을 support vector라고 한다.
  hyperplane이 재조정될 때는 support vector역시 재계산 되어야 한다. hyperplane은 선형 또는 비선형 모든 형태로 표현이 가능하며 일정 수식의 방정식으로 표현이 가능하기 때문에 간단한 수식으로 두 그룹을 분류할 수 있다.

SVM 알고리즘 이란?

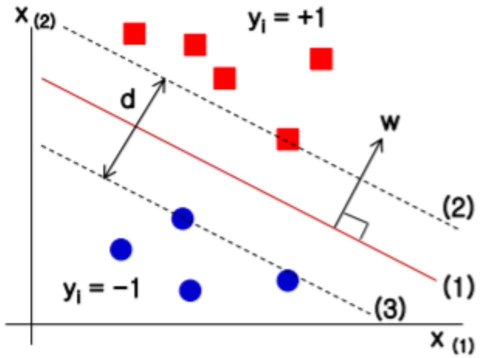
- 이제 해야 할 일은 이 두 그룹 간의 거리를 최대한으로 하여 categorization할 때 발생할 수 있는 오류를 최소화 해야 한다. 그룹 간 거리를 최대한으로 하기 다변수 함수의 최대, 최소 값 찾는 데 이용되는 라고랑지의 미정계수법을 사용한다.

SVM 알고리즘 이란?

- 그런데 SVM 역시 두 그룹간의 거리를 최대로 하는 가중치 값들(다변수)을 정하는 것이므로 다른 머신 러닝 방법과 유사한 측면이 있다. 그렇다면, 왜 Decision tree, Concept learning, 그리고 neural network같은 걸 쓰지 않고 SVM을 쓰는 걸까? 그것을 바로 target이 2그룹 중 하나로 분류되는 경우에 특화되어 있기 때문이다. 예를 들면, 2가지 그룹으로 분류하는 방법으로 Decision tree를 쓸 수도 있고, neural network를 쓸수도 있고 Concept learning을 할 수도 있다. 만약 training set이 선형적인 hyperplane으로 나눠질 수 있다면 모든 경우가 거의 비슷한 성능을 할 것이다. 하지만 비 선형적인 경우 neural network가 가장 좋은 성능을 내게 될 것이라고 하자. 하지만, 만약 이 비선형성이 보다 높은 차원에서 볼 때 선형성을 뒨다고 하면 차원을 확대해서 보다 더 빠르고 쉬운 선형적 머신 러닝 알고리즘을 사용할 수 있지 않을까? 우리는 두 개의 그룹으로 분류하기 때문에 차원의 수를 아주 많이 확대하지 않고도 training set을 선형적으로 바꿀 수 있을 것이라고 직관적으로 생각할 수 있다. 그런 의미에서 SVM이 효과적인 측면을 갖는다고 말할 수 있다.

SVM 알고리즘 예시 - 선형 분리가 가능한 경우

- 아래 그림에서 **평행한 두 직선의 거리 (d)를 최대화 하는 문제**임. -> w를 최소화하는 문제와 동일함.
- 제한 조건이 있는 상태에서 w를 최소화하기 위해 Lagrange 를 이용함. -> λ를 알 수 없으므로 Dual Lagrange 문제로 변환함
- Dual Lagrange 에서 λ 를 구한 후 (Quadratic Programming) 식 (8)로 w를 계산하고, 제한 조건에서 b 를 계산함.



$(1)\ w_1x_{(1)} + w_2x_{(2)} + b = 0 \rightarrow w\cdot x + b = 0$

$(2)\ w\cdot x + b = k \qquad (k > 0)$

$(3)\ w\cdot x + b = -k$

$$y = \begin{cases} +1 \leftarrow & w\cdot x + b > 0 \\ -1 \leftarrow & w\cdot x + b < 0 \end{cases}$$

$(4)\ d = \dfrac{2k}{\| w \|}$ ← d 를 최대화 = w 를 최소화

$(5)\ y_i(w\cdot x_i + b) \geq k \quad (i = 1, 2, \dots N)$ ← 제한 조건

$(6)\ \mathrm{argmin}\ \frac{1}{2}\| w \|^2, \quad constraint: y_i(w\cdot x_i + b) \geq k$

$$\boxed{(7)\ L_p = \frac{1}{2}\| w \|^2 - \sum_{i=1}^{N} \lambda_i \{ y_i(w\cdot x_i + b) - k \}}$$

Primal Lagrange (최소화)

$(8)\ \dfrac{\partial L_p}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{N} \lambda_i y_i x_i$

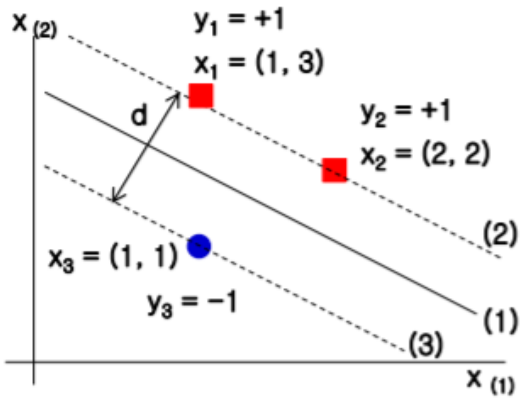
$(9)\ \dfrac{\partial L_p}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \lambda_i y_i = 0$

Dual Lagrange (최대화)

$$\boxed{(10)\ L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j x_i \cdot x_j}$$

SVM 알고리즘 예시 - 선형 분리가 가능한 경우

- 그림과 같이 3개의 훈련 데이터 (x1 , x2 , x3)와 클래스 데이터 (y1 , y2 , y3)가 있는 경우 Decision Boundary를 찾음.
- 훈련 데이터가 많은 경우 Quadratic Optimization 기법을 이용해야 하나, 직관적인 이해를 위해 훈련 데이터가 적은 경우로 단순화시켜 수작업으로 Decision Boundary를 찾음. (N = 3, i = {1, 2, 3}, j = {1, 2, 3})



$$(10)\ L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j x_i \cdot x_j$$

$$(11)\ L_D = \lambda_1 + \lambda_2 + \lambda_3 - \frac{1}{2} \times$$

Dual Lagrange가 최대가 되는 λ를 먼저 찾음.

$$(\lambda_1\lambda_1 y_1 y_1 x_1 x_1 + \lambda_1\lambda_2 y_1 y_2 x_1 x_2 + \lambda_1\lambda_3 y_1 y_3 x_1 x_3 + \lambda_2\lambda_1 y_2 y_1 x_2 x_1 + \lambda_2\lambda_2 y_2 y_2 x_2 x_2 + \lambda_2\lambda_3 y_2 y_3 x_2 x_3 + \lambda_3\lambda_1 y_3 y_1 x_3 x_1 + \lambda_3\lambda_2 y_3 y_2 x_3 x_2 + \lambda_3\lambda_3 y_3 y_3 x_3 x_3)$$

$$(12)\ L_D = \lambda_1 + \lambda_2 + \lambda_3 - \frac{1}{2}(10\lambda_1^2 + 8\lambda_1\lambda_2 - 4\lambda_1\lambda_3 + 8\lambda_2\lambda_1 + 8\lambda_2^2 - 4\lambda_2\lambda_3 - 4\lambda_3\lambda_1 - 4\lambda_3\lambda_2 + 2\lambda_3^2)$$

$$(13)\ L_D = \lambda_1 + \lambda_2 + \lambda_3 - \frac{1}{2}(10\lambda_1^2 + 16\lambda_1\lambda_2 - 8\lambda_1\lambda_3 + 8\lambda_2^2 - 8\lambda_2\lambda_3 + 2\lambda_3^2)$$

$$(14)\ \lambda_1 + \lambda_2 - \lambda_3 = 0 \ \longleftarrow\ (9)$$

$$(15)\ L_D = 2\lambda_1 + 2\lambda_2 - 2\lambda_1^2 - 2\lambda_1\lambda_2 - \lambda_2^2$$

$$(16)\ \frac{\partial L_D}{\partial \lambda_1} = 1 - 2\lambda_1 - \lambda_2 = 0$$

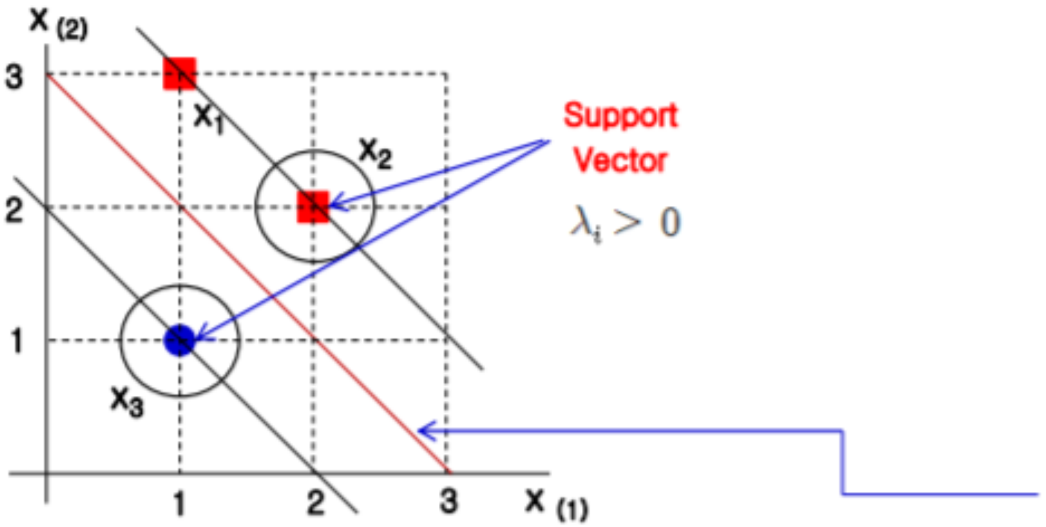
$$(17)\ \frac{\partial L_D}{\partial \lambda_2} = 1 - \lambda_1 - \lambda_2 = 0$$

$$(18)\ \lambda_1 = 0,\ \lambda_2 = 1,\ \lambda_3 = 1 \ \longleftarrow$$

이 값을 식 (8)에 대입하여 w 를 찾음.

SVM 알고리즘 예시 - 선형 분리가 가능한 경우

- 식 (23)이 두 클래스 (+1, -1)의 여러 구분선 중 마진이 최대인 직선의 방정식 (Decision Boundary) 임.
- Support Vector는 x2와, x3 임. -> λ > 0 인 곳이 Support Vector이고, 나머지 λ = 0 임.
- 아래 예시는 x2와 x3 만 있으면 Decision Boundary를 결정할 수 있다는 의미임.



$$(19)\ w = \lambda_1 y_1 x_1 + \lambda_2 y_2 x_2 + \lambda_3 y_3 x_3 \longleftarrow (8)$$

$$(20)\ w = (2,2) - (1,1) = (1,1)$$

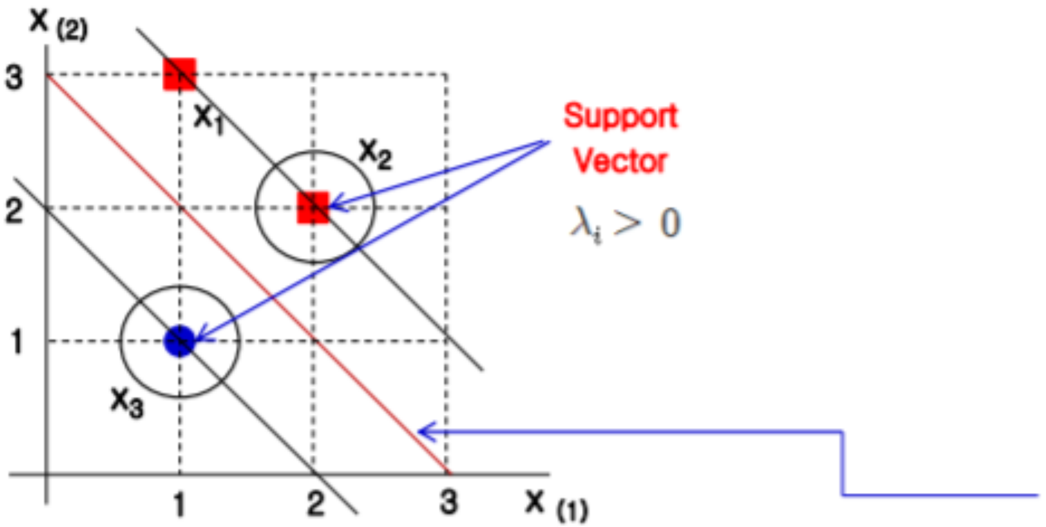
$$(21)\ x_2 : w \cdot x_2 + b > 0 \rightarrow 4 + b > 0$$
$$x_3 : w \cdot x_3 + b < 0 \rightarrow 2 + b < 0$$

$$(22)\ -4 < b < -2 \rightarrow b = -3$$

$$(23)\ x_{(1)} + x_{(2)} - 3 = 0 \longleftarrow \text{Decision Boundary}$$

SVM 알고리즘 예시 - 선형 분리가 가능한 경우

- 식 (22) 에서 b는 -4 ~ -2 사이에 있으며, 중간 지점인 -3 이 최대 마진을 갖는 지점임. (-4와 -2의 평균 지점)
- 인공 신경망 등의 분류 방식은 지정된 오류 이하로 학습되면 그 지점에서 Decision Boundary를 결정하는데, 이것은 여러 구분선 중 하나가 됨. -> 최적의 구분선이라 할 수는 없음.
- SVM 은 여러 구분선 중 마진이 최대가 되는 직선을 찾기 때문에 일반화 특성이 우수함.



$$(19)\ w = \lambda_1 y_1 x_1 + \lambda_2 y_2 x_2 + \lambda_3 y_3 x_3 \longleftarrow (8)$$

$$(20)\ w = (2,2) - (1,1) = (1,1)$$

$$(21)\ x_2 : w \cdot x_2 + b > 0 \rightarrow 4 + b > 0$$
$$x_3 : w \cdot x_3 + b < 0 \rightarrow 2 + b < 0$$

$$(22)\ -4 < b < -2 \rightarrow b = -3$$

$$(23)\ x_{(1)} + x_{(2)} - 3 = 0 \longleftarrow \text{Decision Boundary}$$

SVM 알고리즘 : 의사결정 경계면

- **일반적으로 SVM의 성능은 우수**한 것으로 알려져 있으나, **기술적 지표로 Feature 데이터를 구성한 경우에는 성능이 좋지 않음.**
- 그 이유는 분석에 사용된 데이터 세트가 분류에 적당한 데이터가 아니기 때문임. 더 나은 데이터 세트를 만들기 위해서는 **금융 이론**과 조화를 이루어야 함.