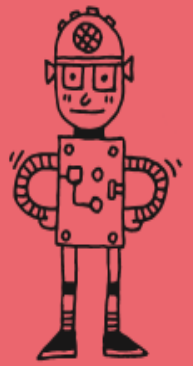
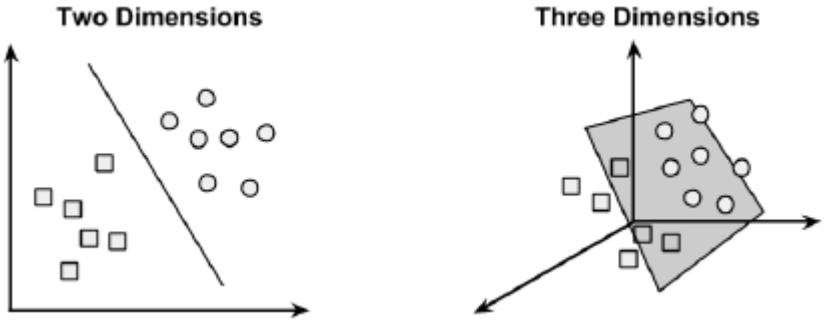
SVM

데이터사이언스 학과
김 승 환
swkim4610@inha.ac.kr

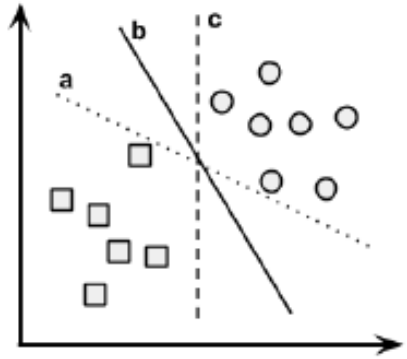


1 SVM(Support Vector Machine)

Classification with hyperplanes

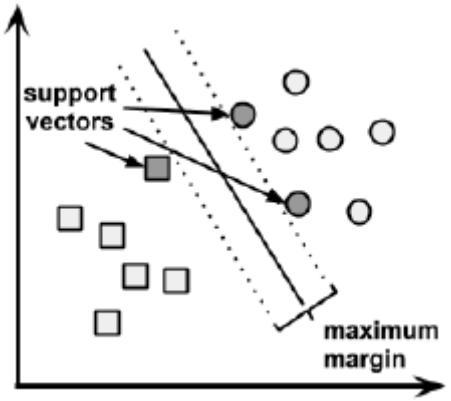


The task of the SVM algorithm is to identify a line that separates the two classes. As shown in the following figure, there is more than one choice of dividing line between the groups of circles and squares. Three such possibilities are labeled a, b, and c. How does the algorithm choose?



1 SVM(Support Vector Machine)

MMH(Maximum Margin Hyperplane)



The support vectors (indicated by arrows in the figure that follows) are the points from each class that are the closest to the MMH; each class must have at least one support vector, but it is possible to have more than one.

결국, Support Vector를 찾고 이를 이용하여 MMH를 구하는 것이다.
문제는 선형분리가 쉽지 않다는 것이다. 이 때, 데이터 차원을 고차원으로 변환하여 분리가 잘되도록 하는 것이 핵심이다.

2 MMH

$$(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i), \ldots, (x_N, y_N) \qquad y = \begin{cases} +1 \\ -1 \end{cases}$$

y = 1, -1을 구분하는 직선 방정식을 $f(x) = w^T x - w_0$ 라고 하자. 우리의 목적은 w를 구하는 것이다. 판별 함수에서 y = +1 이 되는 x 값을 x_+ 라고 하고 y = -1이 되는 x 값은 x_- 라고 하면 아래의 식을 만족한다.

$$f(x_+) = w^T x_+ - w_0 > 0$$

$$f(x_-) = w^T x_- - w_0 < 0$$

x_+ 중에 판별함수 값을 가장 작게 하는 x_+를 x^+ 라고 하고, x_- 중에 판별함수 값을 가장 크게 하는 x_-를 x^- 라고 하자. x^+, x^- 는 각 클래스에 속한 데이터 중에 가장 경계선이 있는 최전방 데이터이고 이 데이터를 서포트 벡터라고 한다.

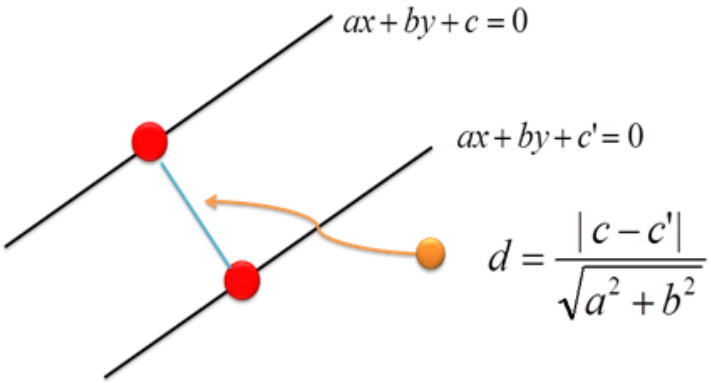
2 MMH

서포트 벡터도 아래 식을 만족한다. 부등호로 문제를 풀기 어려우니 좌에서 우로 변형해 풀어보자.

$$f(x^+) = w^T x^+ - w_0 > 0 \qquad f(x^+) = w^T x^+ - w_0 = +1$$

$$f(x^-) = w^T x^- - w_0 < 0 \qquad f(x^-) = w^T x^- - w_0 = -1$$

판별 경계선 $f(x) = w^T x - w_0$ 와 x^+, x^- 사이 거리는 절편만 다른 두 직선의 점의 거리로 아래와 같다.

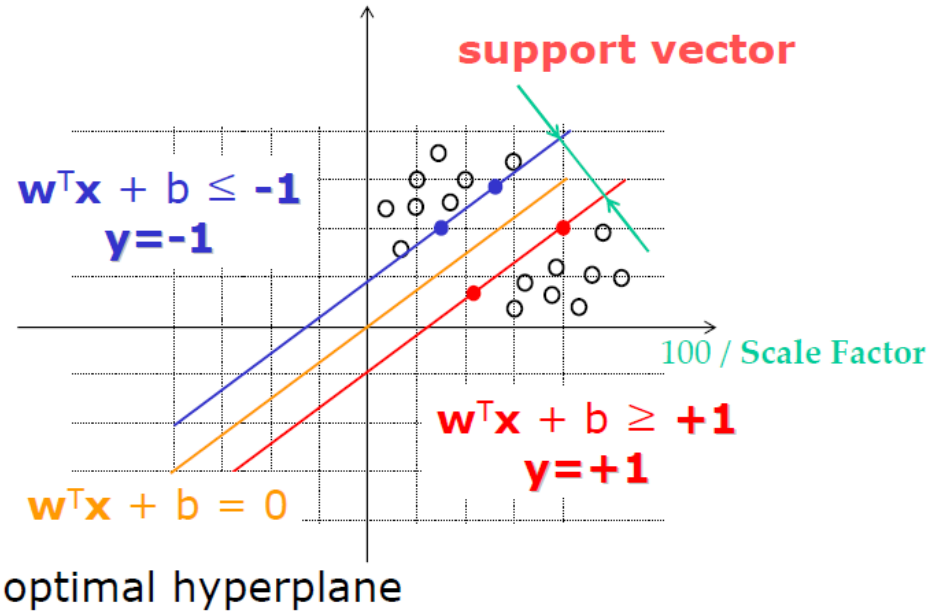


$$\frac{w^T x^+ - w_0}{\|w\|} = \frac{1}{\|w\|}$$

$$-\frac{w^T x^- - w_0}{\|w\|} = \frac{1}{\|w\|}$$

2 MMH

MMH는 아래와 같이 구할 수 있다.



두 평면간의 거리는 $\frac{2}{||w||}$ 가 되고 이 거리를 최대화하는 w를 구하는 것이다.

$\frac{2}{||w||}$를 최대화 하는 것과 ||w|| 를 최소화하는 것이 같으므로 손실함수는 $L = \frac{1}{2}||w||^2 = \frac{1}{2}w^Tw$ 이 된다.

2 MMH

또한, 모든 데이터에 대해 아래 조건을 만족해야 한다.

$y_i = 1$ 일 때, $f(x_i) \geq 1$ 이여야 하고, $y_i = -1$ 일 때, $f(x_i) \leq -1$ 이여야 한다.

$$y_i \cdot f(x_i) = y_i \cdot (w^T x_i - w_o) \geq 1 \quad (i = 1, \ldots, N) \quad \Rightarrow \quad y_i \cdot (w^T x_i - w_o) - 1 \geq 0 \quad (i = 1, \ldots, N)$$

그러므로, 최종적으로 구한 손실함수는 아래와 같다.

$$L = \frac{1}{2} w^T w - \sum_{i=1}^{N} a_i \{y_i \cdot (w^T x_i - w_o) - 1\}$$

손실함수를 최소화하는 w, w_0, a_i 를 구하면 된다.

$$\frac{\partial L}{\partial w} = 0, \frac{\partial L}{\partial w_0} = 0 \qquad w = \sum_{i=1}^{N} a_i y_i x_i \qquad 0 = \sum_{i=1}^{N} a_i y_i \qquad a_i \geq 0 \quad (i = 1, \ldots, N) \qquad a_i = 0 \text{ if } x_i \notin \{x^+, x^-\}$$

a_i 는 서포트벡터만 계산에 기여하므로 서포트 벡터가 아닌 값은 모두 0 이다.

$$\Rightarrow \quad L = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y_i y_j x_i^T x_j \quad \Leftarrow \quad a_i \text{ 만 구하면 되는 문제로 바뀌었다.}$$

2 MMH

또한, 모든 데이터에 대해 아래 조건을 만족해야 한다.

$y_i = 1$ 일 때, $f(x_i) \geq 1$ 이여야 하고, $y_i = -1$ 일 때, $f(x_i) \leq -1$ 이여야 한다.

$$y_i \cdot f(x_i) = y_i \cdot (w^T x_i - w_o) \geq 1 \quad (i = 1, \ldots, N) \quad \Rightarrow \quad y_i \cdot (w^T x_i - w_o) - 1 \geq 0 \quad (i = 1, \ldots, N)$$

그러므로, 최종적으로 구한 손실함수는 아래와 같다.

$$L = \frac{1}{2} w^T w - \sum_{i=1}^{N} a_i \{y_i \cdot (w^T x_i - w_o) - 1\}$$

손실함수를 최소화하는 w, w_0, a_i 를 구하면 된다.

$$\frac{\partial L}{\partial w} = 0, \frac{\partial L}{\partial w_0} = 0 \qquad w = \sum_{i=1}^{N} a_i y_i x_i \qquad 0 = \sum_{i=1}^{N} a_i y_i \qquad a_i \geq 0 \quad (i = 1, \ldots, N) \qquad a_i = 0 \text{ if } x_i \notin \{x^+, x^-\}$$

a_i 는 서포트벡터만 계산에 기여하므로 서포트 벡터가 아닌 값은 모두 0 이다.

$$\Rightarrow \quad L = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y_i y_j x_i^T x_j \quad \Leftarrow$$

a_i 만 구하면 되는 문제로 바뀌었다.

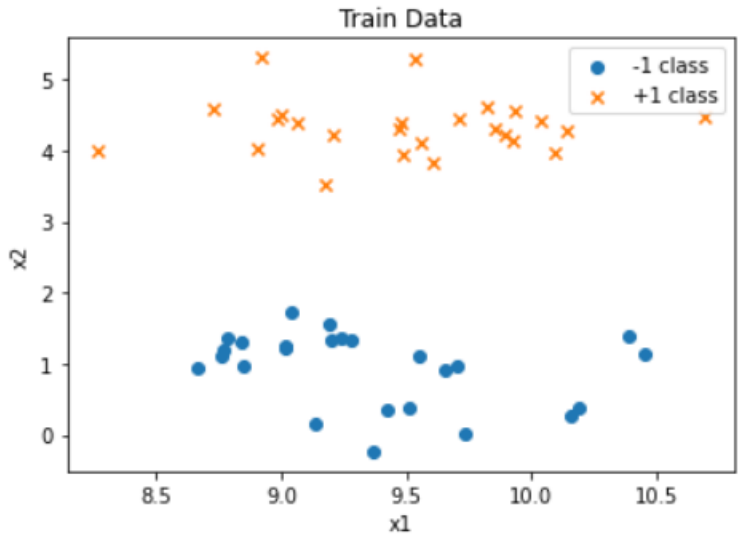
3 Linear SVM 실습

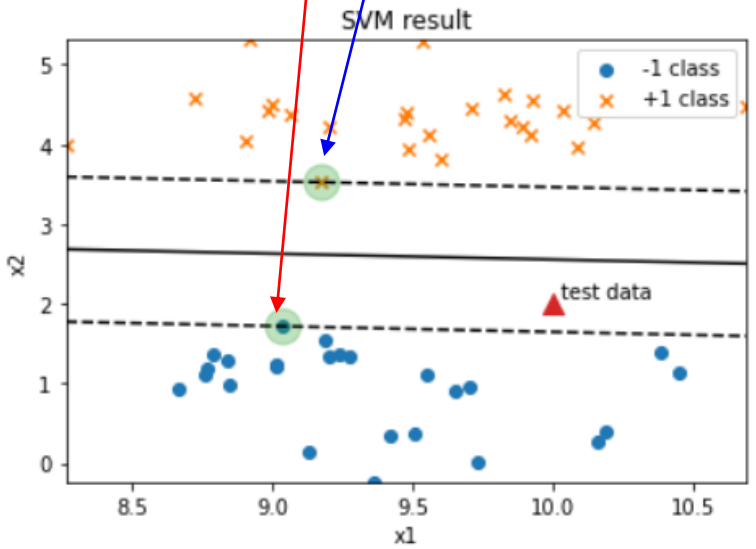
$\frac{2}{||\mathbf{w}||}$ 를 최대화 하는 것은 $L = \frac{1}{2}||w||^2 = \frac{1}{2}w^T w$ 을 최소화 하는 것과 같다.

MMH는 아래와 같이 구할 수 있다.

```
1 model.support_vectors_   # 서포트 벡터 값
```

```
array([[9.03715314, 1.71813465],
       [9.17124955, 3.52485535]])
```

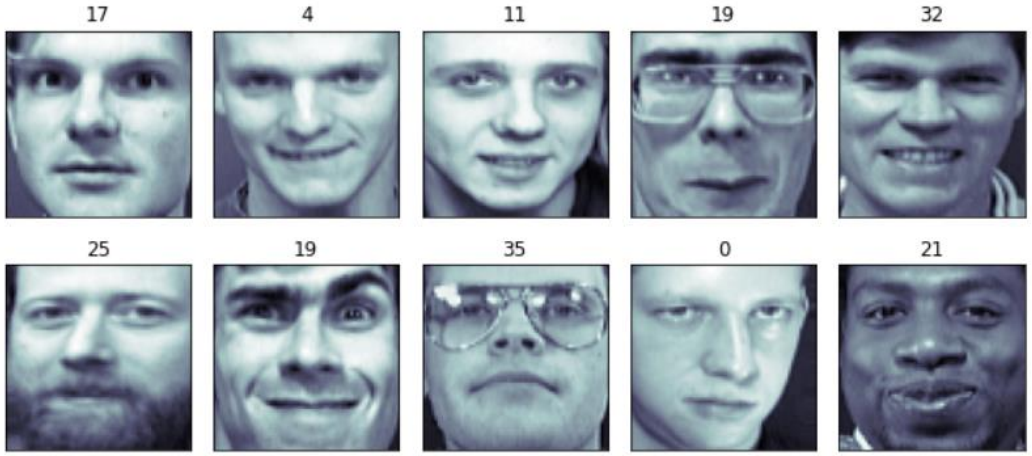




SVM_basic.ipynb

3 Linear SVM 실습

올리베티 얼굴이미지를 Linear SVM모형으로 인식한 결과, 93%의 정확도를 보임

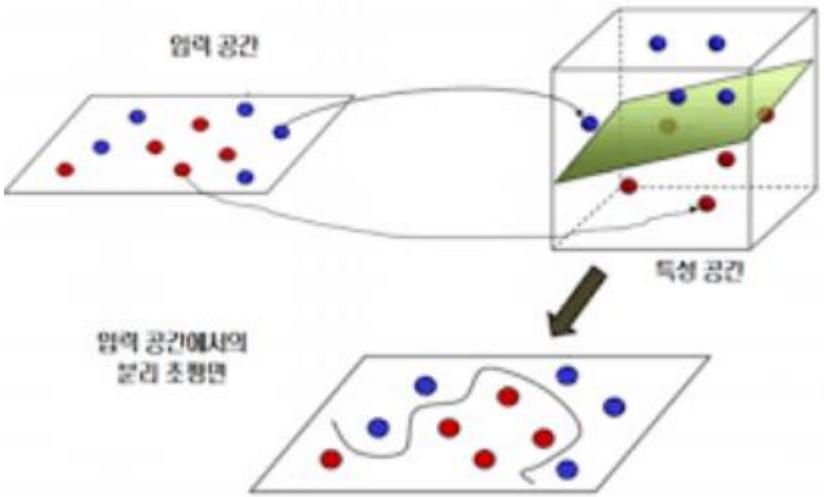


	precision	recall	f1-score	support
0	0.86	1.00	0.92	6
1	1.00	1.00	1.00	5
2	1.00	1.00	1.00	4
3	0.50	1.00	0.67	2
4	1.00	0.50	0.67	2
5	1.00	1.00	1.00	5
6	0.83	0.83	0.83	6
7	1.00	0.67	0.80	3
8	0.67	1.00	0.80	2
9	1.00	1.00	1.00	3
10	1.00	1.00	1.00	6

4 Nonlinear Mapping

비선형 매핑은 아래 그림으로 이해할 수 있다.

분리가 어려운 입력공간을 분리가 가능한 특성공간으로 확장하는 함수를 구하는 문제다.



4 Nonlinear Mapping

XOR 문제를 생각해 보자. 선형 식으로 분리가 안된다.

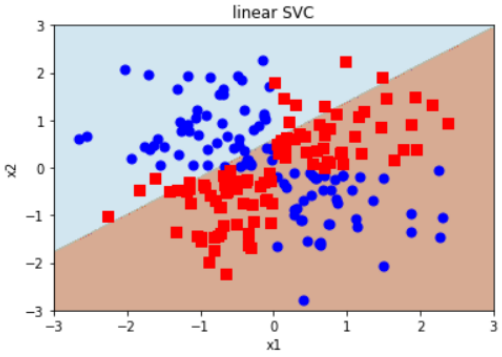
이를 우측과 같이 비선형 매핑을 하면 선형 MMH를 구할 수 있다. 딥러닝의 Hidden Layer와 같은 개념이다.

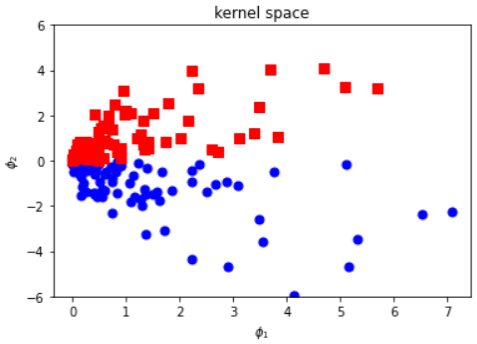
딥러닝에서 히든 노드(hidden Node) 수를 증가하는 것은 매핑의 차원을 증가시키는 것과 같은 개념이다.

여기서, $\Phi(\vec{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^T.$ 를 Kernel Trick 함수라고 부른다.

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

x_1^2	$\sqrt{2}x_1x_2$	x_2^2	y
0	0	0	0
0	0	1	1
1	0	0	1
1	$\sqrt{2}$	1	0





3 Nonlinear Mapping

대표적으로 사용되는 커널 함수는 아래와 같다.

어떤 함수가 최적인지 모르고 함수에 따른 파라미터도 모르기 때문에 시행착오법을 통해 결정한다.

초평면의 최대 마진을 구하기 위해서 훈련 데이터 개수의 제곱에 해당하는 계산량 필요하기 때문에 빅데이터에 대해 계산할 경우, 계산 속도 이슈가 존재한다.

`kernel = "linear"`: 선형 SVM. $k(x_1, x_2) = x_1^T x_2$

`kernel = "poly"`: 다항 커널. $k(x_1, x_2) = (\gamma(x_1^T x_2) + \theta)^d$

- `gamma`: γ
- `coef0`: θ
- `degree`: d

`kernel = "rbf"` 또는 `kernel = None`: RBF 커널. $k(x_1, x_2) = \exp\left(-\gamma||x_1 - x_2||^2\right)$

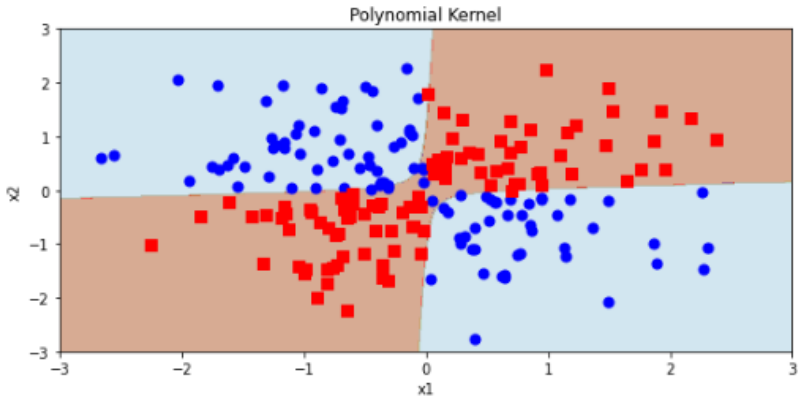
- `gamma`: γ

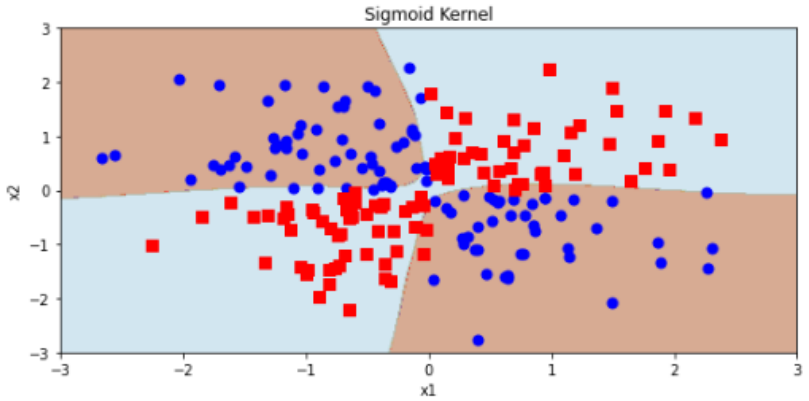
`kernel = "sigmoid"`: 시그모이드 커널. $k(x_1, x_2) = \tanh(\gamma(x_1^T x_2) + \theta)$

- `gamma`: γ
- `coef0`: θ

3 Nonlinear Mapping

아래는 Polynomial Kernel과 Sigmoid Kernel을 적용한 결과다.





감사합니다