Ch05_서포트 벡터 머신_0721

SVM

: 서포트 벡터 머신(SVM) 은 매우 강력한 선형이나 비선형 분류, 회귀, 이상치 탐색에 사용할 수 있는 다목적 머신러닝 모델이다.

▼ 5.1 선형 SVM 분류

SVM

- **SVM의 아이디어**

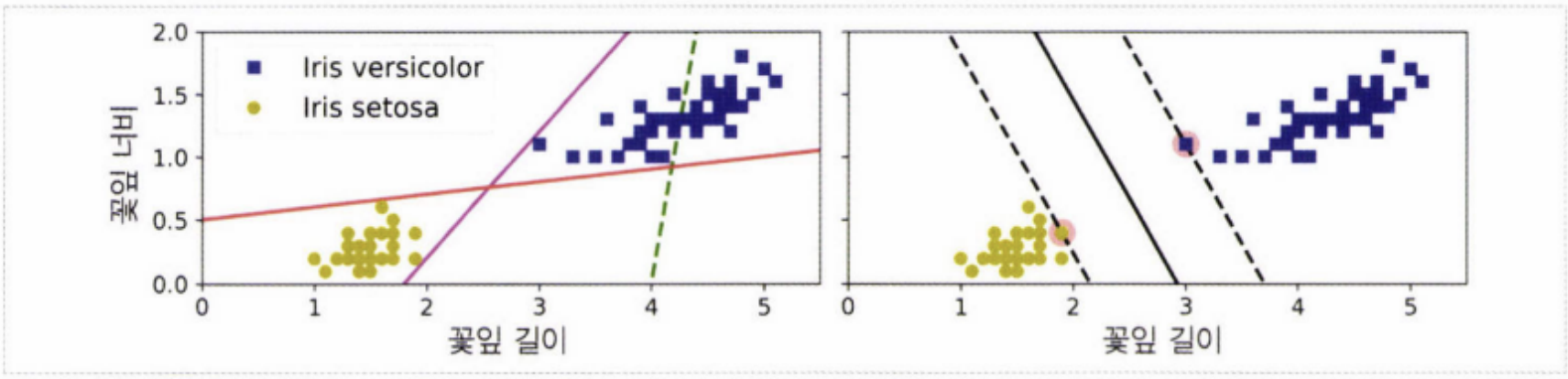


그림 5-1 라지 마진 분류

→ 붓꽃 데이터셋의 일부를 나타낸 그림이다.

→ 두 클래스가 직선으로 확실히 잘 나뉘어지며, 선형적으로 구분된다.

→ 왼쪽 그래프에서, 세 개의 선형 분류기로 결정 경계(점선)가 만들어진 것을 볼 수있다. 이 결정 경계를 만든 모델은 클래스를 적절하게 분류하지 못하고 있다고 할 수 있다.

다른 두 모델은 훈련 세트에 대해 완벽하게 동작한다. 하지만 결정 경계가 샘플에 너무 가까워 새로운 샘플에 대해서는 잘 작동하지 못한다.

→ 오른쪽 그래프는 SVM 분류기의 결정 경계를 나타낸다(실선). 이 직선은 두 개의 클래스를 나누고 있고, 제일 가까운 훈련 샘플로부터 가능한 한 멀리 떨어져 있다.

(1) SVM 분류기는 클래스 사이에 가장 폭이 넓은 도로를 찾는 것으로, 라진 마진 분류라고 한다.

(2) 도로 바깥쪽에 훈련 샘플을 더 추가해도 결정 경계에는 영향을 미치지 않는다. 도로 경계에 위치한 샘플에 의해 전적으로 결정된다. 이런 샘플을 서포트 벡터라고 한다.

▼ 5.1.1 소프트 마진 분류

하드 마진 분류

: 모든 샘플이 도로 바깥쪽에 게 분류되어 있는 경우이다.

- **하진 마드 분류의 문제점**

  : 데이터가 선형적으로 구분될 수 있어야 제대로 작동하며, 이상치에 민감하다.

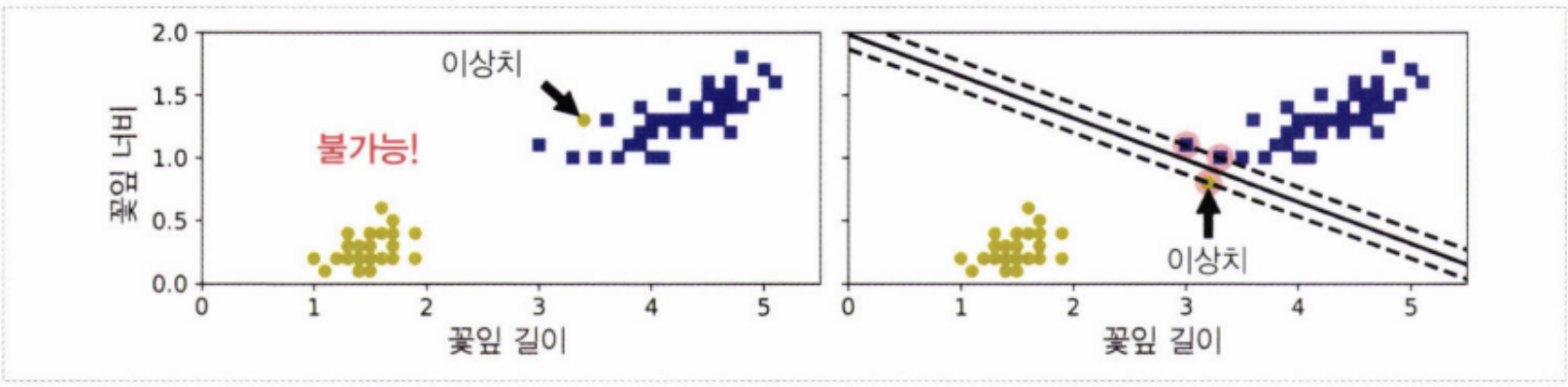


그림 5-3 이상치에 민감한 하드 마진

→ 붓꽃 데이터셋에 이상치가 하나 존재하는 경우의 그래프이다.

→ 왼쪽 그래프에서는 하드 마진을 찾을 수 없다.

→ 오른쪽 그래프의 결정 경계는 이상치가 없는 경우(그림 5-1)의 결정 경계와 매우 다르고, 일반화가 잘 되지 않는다.

소프트 마진 분류

: 하드 마진의 문제점을 해결하기 위해, 도로의 폭을 가능한 넓게 유지하는 것과 마진오류(샘플이 도로 중간이나, 반대쪽에 있는 경우) 사이에 적절한 균형을 잡는 것을 말한다.

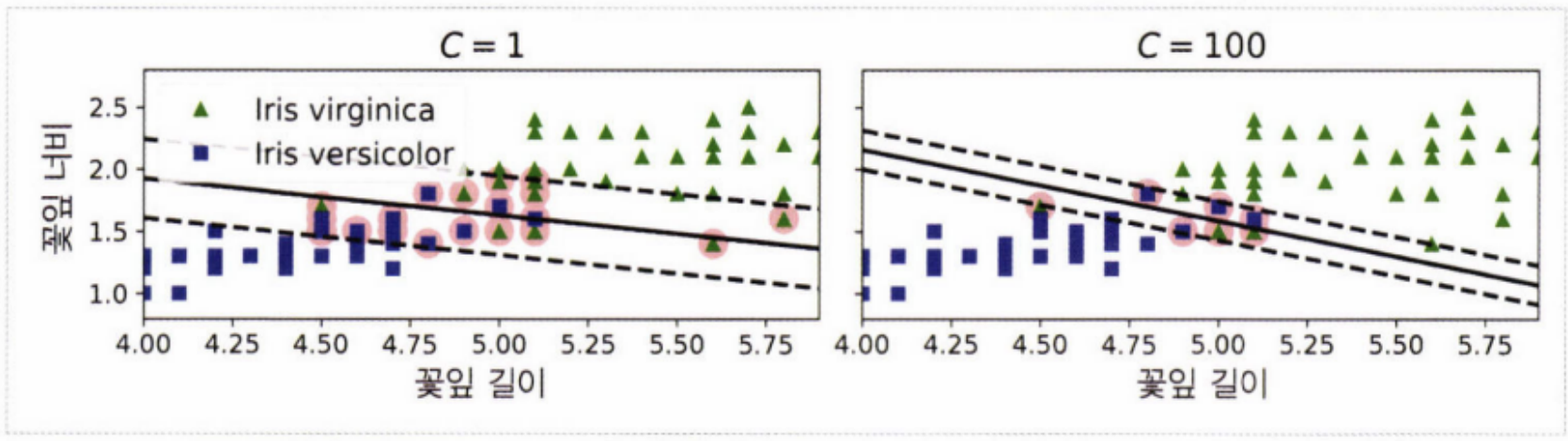


그림 5-4 넓은 마진(왼쪽) 대 적은 마진 오류(오른쪽)

→ 사이킷런의 SVM 모델을 만들 때, 여러 하이퍼파라미터 중 C를 이용하여 마진을 결정할 수 있다.

→ 마진 오류는 일반적으로 적은게 좋다.

사이킷런 코드

▼ 5.2 비선형 SVM 분류

비선형 SVM 분류

선형적으로 분류할 수 없는 데이터셋의 경우 비선형 SVM분류를 사용한다.

비선형 데이터셋을 다루는 방법은 다항 특성과 같은 특성을 더 추가하여, 선형적으로 구분되는 데이터셋이 만들어질 수 있도록 하는 것이다.

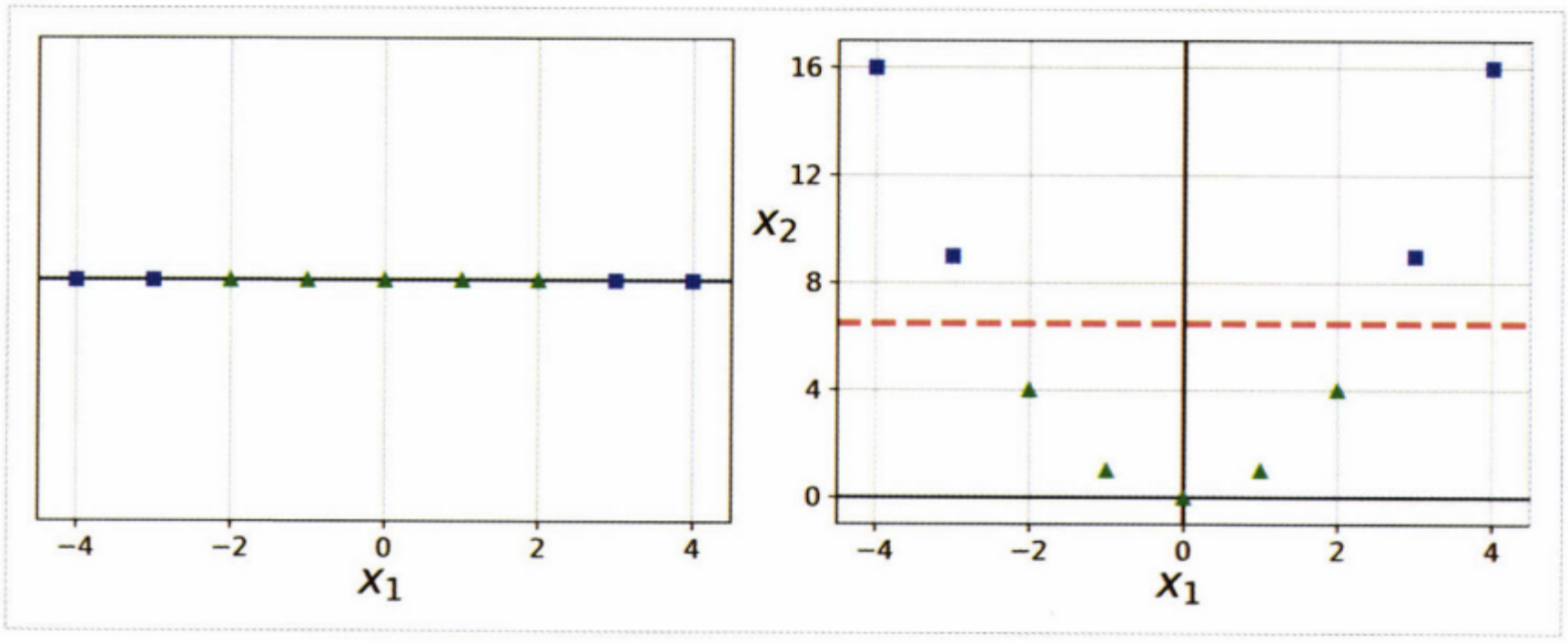


그림 5-5 특성을 추가하여 선형적으로 구분되는 데이터셋 만들기

→ 왼쪽 그래프

: 하나의 특성 만을 가진 간단한 데이터셋을 나타낸다. 이 데이터셋은 선형적으로 구분이 안 됩니다. 하지만 두 번째 특성 $X_2 = (X_1)^2$ 을 추가하여 만들어진 2차원 데이터셋은 완벽하게 선형적으로 구분할 수 있다.

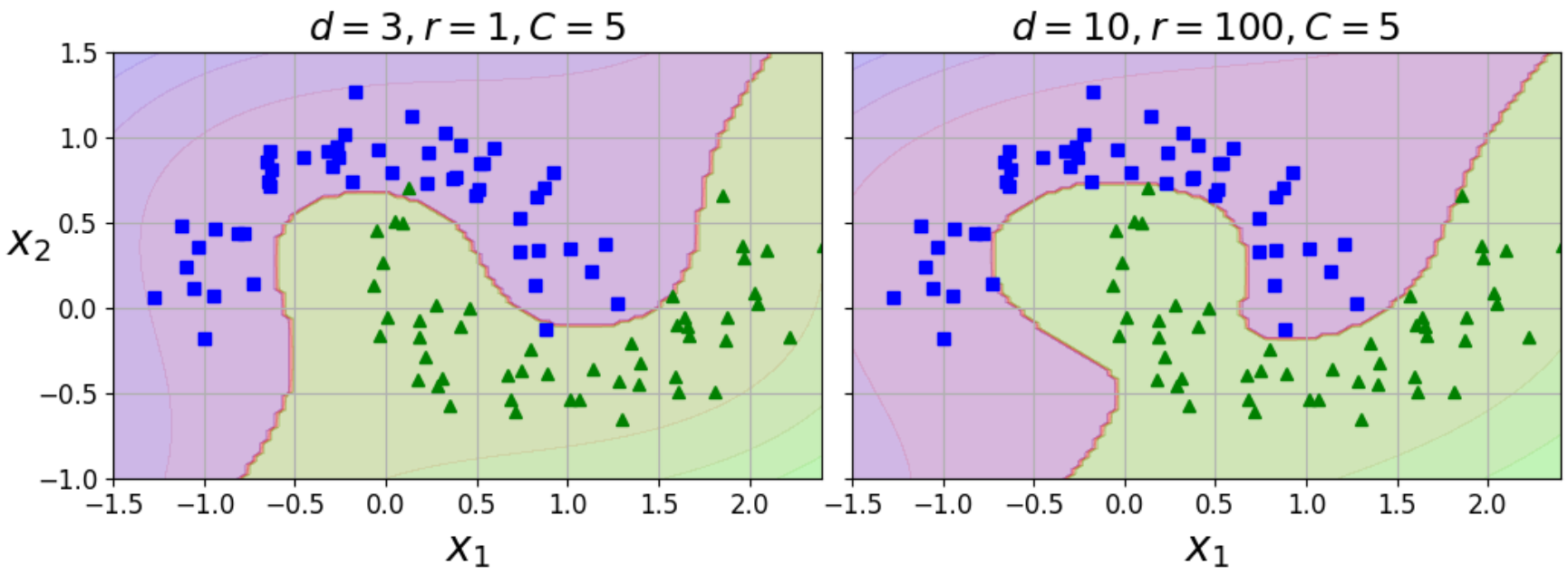
사이킷런 구현

▼ 5.2.1 다항식 커널

커널 트럭

: 커널 트릭은 실제로는 특성을 추가하지 않으면서 다항식 특성을 많이 추가한 것과 같은 결과를 얻을 수 있다. 실제로는 어떤 특성도 추가하지 않기 때문에 엄청난 수의 특성 조합이 생기지 않는다.

SVC 파이썬 클래스 구현



▼ **5.2.2 유사도 특성**

유사도 특성

: 비선형 특성을 다루는 기법 중 하나는, 각 샘플이 특정 랜드마크와 얼마나 닮았는지 측정하는 유사도 함수로 계산한 특성을 추가하는 것 이다.

ex)

- **가정**

- 앞의 데이터에 1차원 데이터셋에 두 개의 랜드마크 $x_1 = -2, x_1 = 1$을 추가한다. (그림 5-8의 왼쪽 그래프)

- $y = 0.3$인 가우시안 방사 기저함수(RBF)를 유사도 함수로 정의한다.

식 5-1 가우시안 RBF

$$\phi_\gamma(\mathbf{x}, \ell) = \exp\left(-\gamma \left\| \mathbf{x} - \ell \right\|^2\right)$$

→ 이 힘수의 값은 0(랜드마크에서 아주 멀리 떨어진 경우)부터 1(랜드마크와 같은 위치일 경우) 까지 변화하며 종 모양으로 나타난다.

- $x = -1$

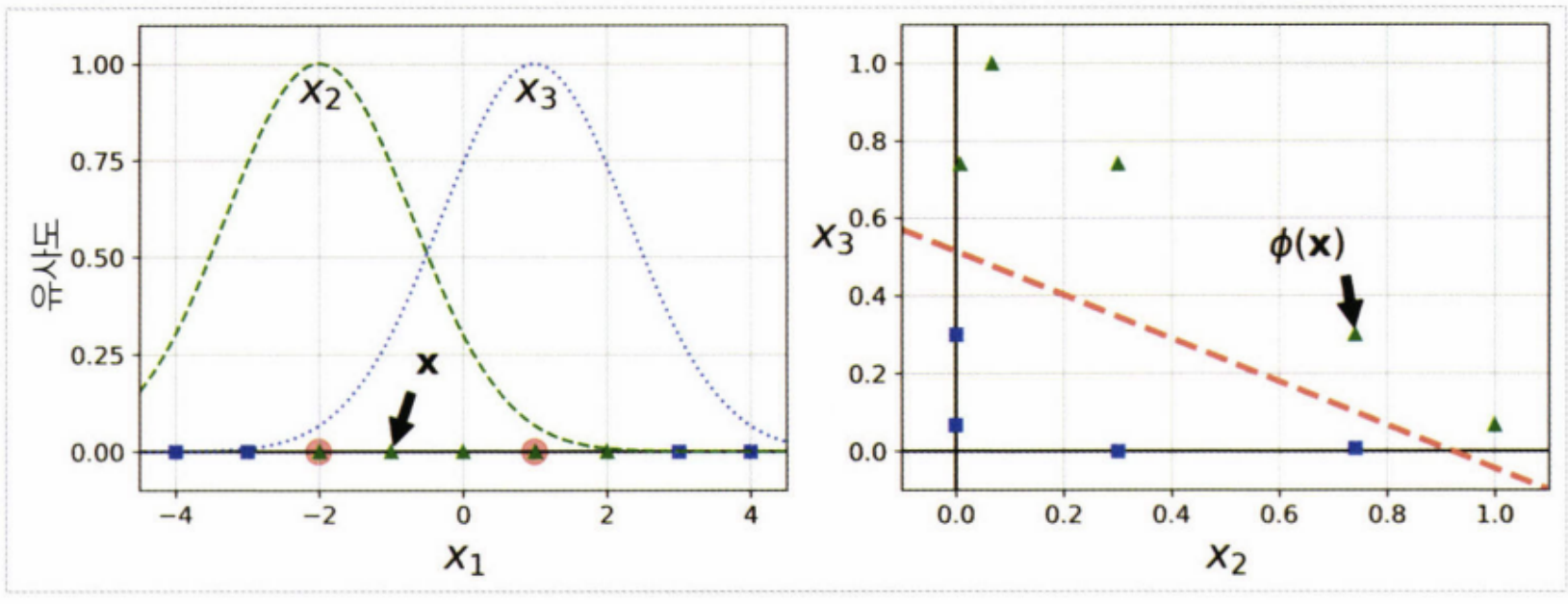


그림 5-8 가우시안 RBF를 사용한 유사도 특성

→ 이 샘플은 첫 번째 랜드마크에서 1만큼 떨어져 있고, 두 번째 랜드마크에서 2만큼 떨어져 있습니다. 따라서 새로 만든 특성은 $x_2 = \exp(-0.3X1^2) = 0.74$ 와 $x_3 = \exp(-0.3X2^2) = 0.30$ 이다.

→ 그래프 5-8의 오른쪽 그래프는 변환된 데이터 셋을 보여준다. 그림에서 볼 수 있듯이, 선형적으로 구분이 가능하다.

- **랜드마크 선택**

  : 간단한 방법은 데이터셋에 있는 모든 샘플 위치에 랜드마크를 설정하는 것이다. 이렇게 하면 차원이 매우 커지고, 변환된 훈련 세트가 선형적으로 구분될 가능성이 높다.

  단점은 훈련 세트에 있는 n개의 특성을 가진 m개의 샘플이 m개의 특성을 가진 m개의 샘플로 변환된다는 것이다(원본 특성은 제외한다고 가정) . 훈련 세트가 매우 클 경우 동일한 크기의 아주 많은 특성이 만들어진다.

▼ 5.2.3 가우시안 RBF 커널

: 다항 특성 방식과 마찬가지로, 유사도 특성 방식도 머신러닝 알고리즘에 유용하게 사용될 수 있다.

추가 특성을 모두 계산하려면 연산 비용이 많이 드는데 특히 훈련 세트가 클 수록 더 그렇다. 이때, 커널 트릭을 사용하면 유사도 특성을 많이 추가하는 것과 같은 비슷한 결과를 얻을 수 있다.

> **가우시안 RBF 커널을 시용한 SVC 모델**

▼ 5.2.4 계산 복잡도

> **계산 복잡도**

LinearSVC 파이썬 클래스는 선형 SVM을 위한 최적화된 알고리즘을 구현한 li linear 라이브러리를 기반으로 한다. 이 라이브러리는 커널 트력을 원하지 않지만 훈련 샘플과 특성 수에 거의 선형적으로 늘어난다. 이 알고리즘의 훈련 간 복잡도는 략 $O(m * n)$ 정도이다.

정밀도를 높이면 알고리즘의 수행 시간이 길어진다. 이는 허용오차 하이퍼파미터 ϵ 으로 조절한다(사 킷런에서는 매개변수 tol). 대부분의 분류 문제는 허용오차를 기본값으로 두면 잘 작동한다.

SVC는 커널 트릭 알고리즘을 구현한 libsvm 라이브러를 기반으로로 한다. 훈련의 시간 복잡도는 보통 $O(m^2 * n)$과 $O(m^3 * n)$ 사이다. 이것은 훈련 샘플 수가 커지면(예를 들면 수십만 개 샘플) 엄청 나게 느려진다는 것을 의미한다. 따라서 복잡하지만 작거나 중간 규모의 훈련세트에 이 알고리이 적합하다.

하지만 특성의 개수 , 특히 희소 특성(각 샘플에 0이 아닌 특성이 몇 개 없는 경우)인 경우에는 잘 확정된다. 이런 경우,알고리즘의 성능이 샘플이 가진 0이 아닌 특성의 평균 수에 거의 비례한다.

> **사이킷런 파이썬 클래스 비교**

표 5-1 SVM 분류를 위한 사이킷런 파이썬 클래스 비교

파이썬 클래스	시간 복잡도	외부 메모리 학습 지원	스케일 조정의 필요성	커널 트릭
LinearSVC	$O(m \times n)$	아니오	예	아니오
SGDClassifier	$O(m \times n)$	예	예	아니오
SVC	$O(m^2 \times n) \sim O(m^3 \times n)$	아니오	예	예

▼ **5.3 SVM 회귀**

SVM 회귀

SVM 알고리즘은 다목적으로 시용할 수 있다. 선형,비선형 분류뿐만 아니라 선형, 비선형 회귀에도 사용할 수 있다.

SVM을 분류가아니라 회귀에 적용하는 방법은 목표를 반대로 하는 것이다. 일정한 마진 오류 안에서 두 클래스의 도로 폭이 가능한 최대가 되도록 하는 대신, SVM 회귀는 제한된 마진 오류(즉, 도로 밖의 샘플) 안에서 도로 안에 가능한 많은 샘플이 들어가도록 학습한다. 도로의 폭은 하이퍼파라미터 ϵ으로 조절한다.

마진 안에서는 훈련샘플이 추가되어도 모델의 예측에는 영향이 없다. 그래서 이 모델을 “ϵ에 민감하지 않다” 고 한다.

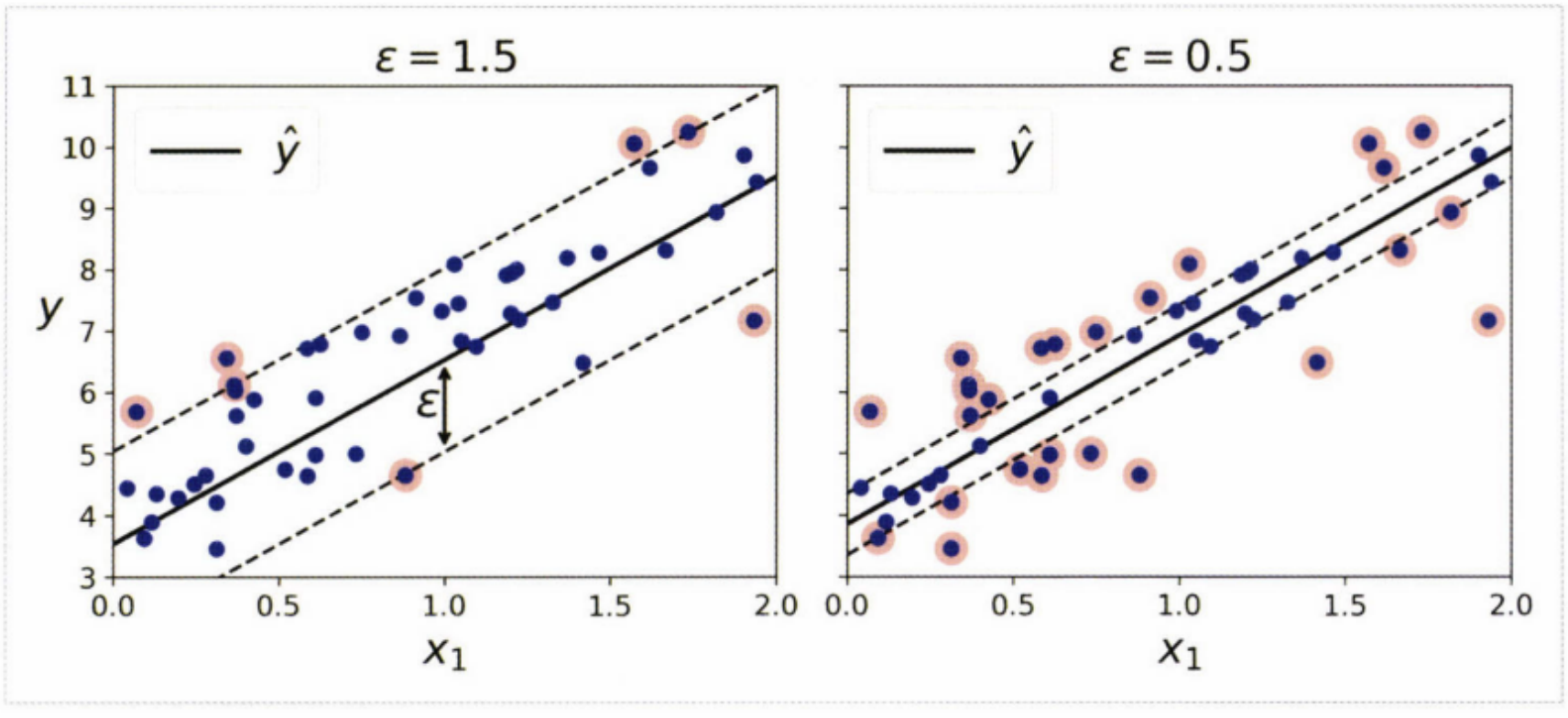


그림 5-10 SVM 회귀

→ 무작위로 생성한 선형 데이터셋에 훈련시킨 두 개의 선형 SVM 회귀 모델을 나타낸다. 하나는마진을 크게 ($\epsilon = 1.5$) 하고 다른 하나는 마진을 작게 (ε=0.5) 하여 나타낸 것이다.

사이킷런의 LinearSVR 시용 → 선형 SVM 회귀 적용

▼ **5.4 SVM 이론**

- **선형 SVM 분류기**

- **표기법**

: 펀향 θ_0과 입력 특성의 기중치 θ_1 에서 θ_n까지 전체 모델 파라미터를 하나의 벡터 θ에 넣는다.

: 모든 샘플에 편향에 해당히는 입력값 $x_0 = 1$을 추가한다.

→ 이 장에서는, 편향을 b, 가중치 벡터를 w라고 표기한다. 따라서, 입력 특성 벡터에 편향을 위한 특성이 추가되지 않는다.

▼ 5.4.1 결정 함수와 예측

선형 SYM 분류기 모델은 단순히 결정 함수 $w^T x + b = w_1 x_1 + \ldots + w_n x_x + b$를 계산해서 새로운 샘플 x의 클래스를 예측한다. 이때, 결과값이 0보다 크면 예측된 클래스 $\hat{y}$은 양성 클래스(1), 0보다 작으면 음성클래스(0)이 된다.

식 5–2 선형 SVM 분류기의 예측

$$\hat{y} = \begin{cases} 0 & \mathbf{w}^T \mathbf{x} + b < 0 \text{일 때} \\ 1 & \mathbf{w}^T \mathbf{x} + b \geq 0 \text{일 때} \end{cases}$$

- **ex. iris**

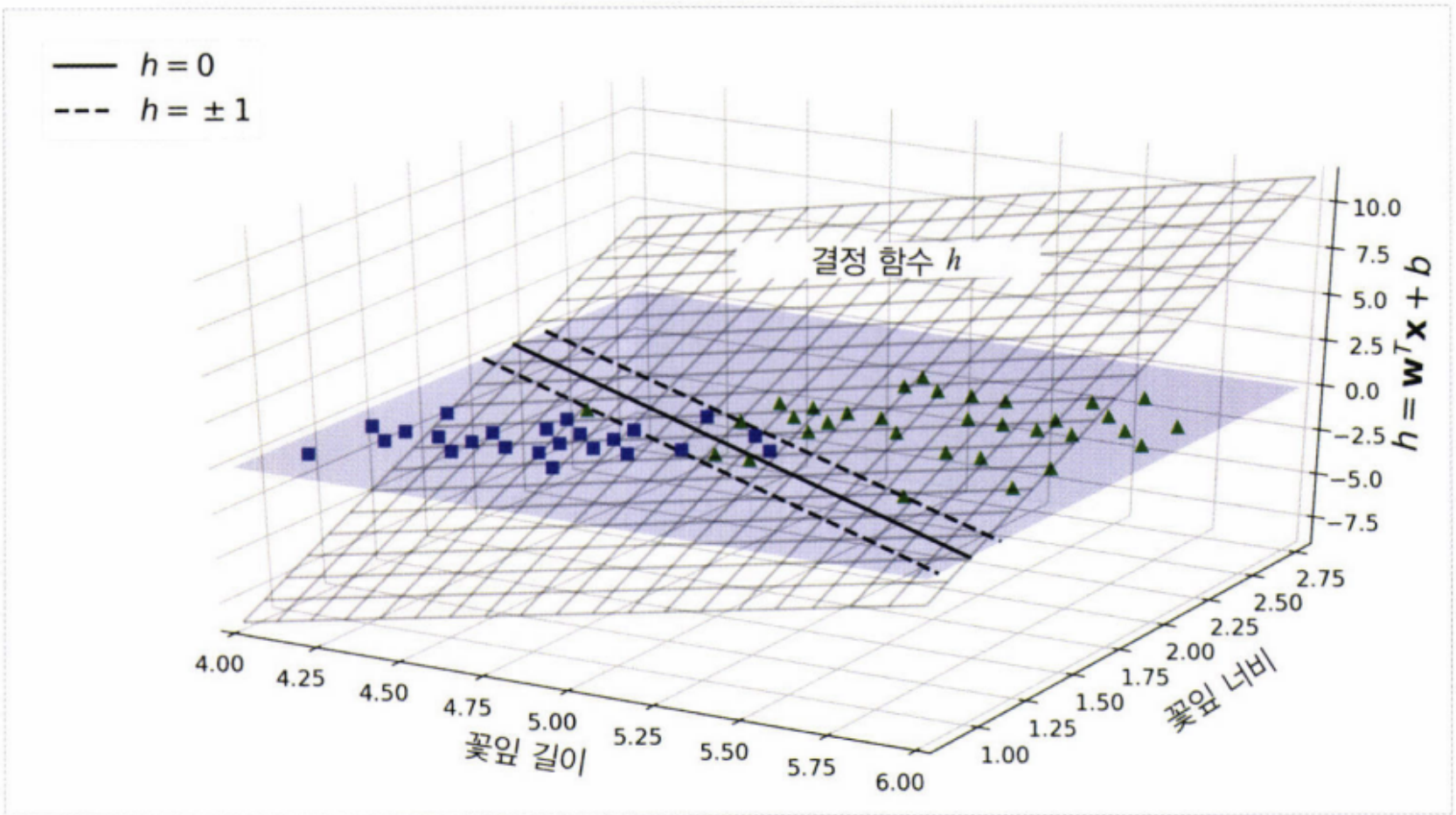


그림 5-12 iris 데이터셋의 결정 함수

→ 특성이 두개(꽃잎 너비와 길이)인 데이터셋이기 때문에 2차원 평면이다.

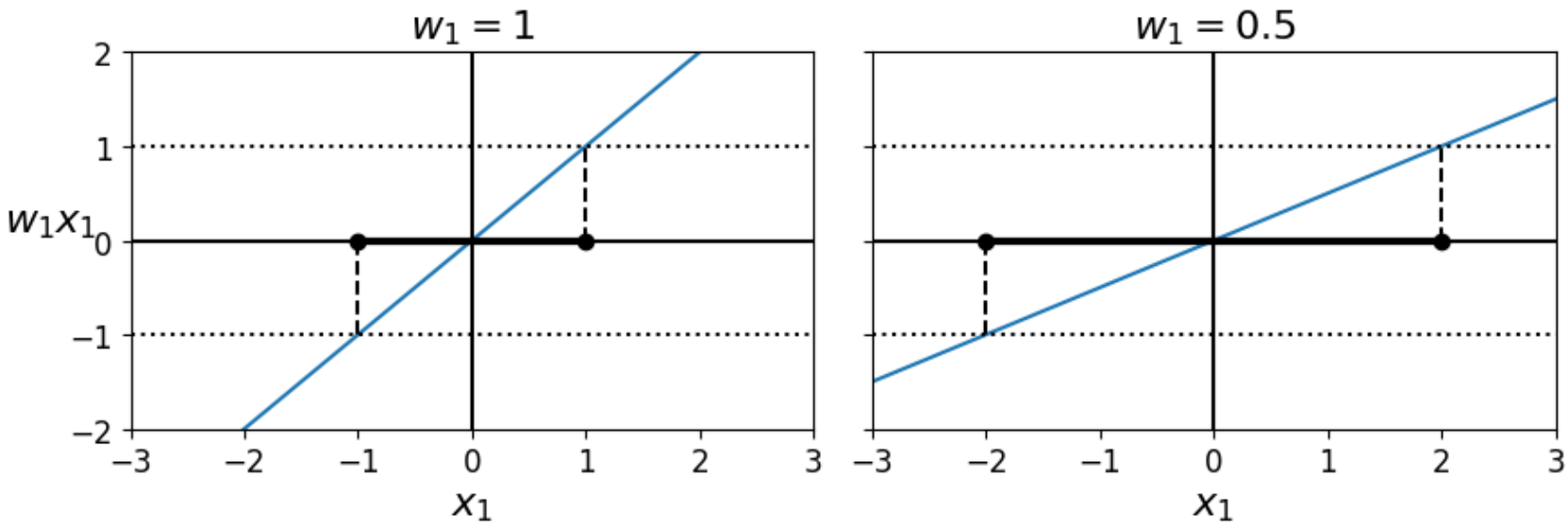
→ 결정 경계는 결정 함수의 값이 0인 점들로 이루어져 있다. 이는 두 평면의 교차점으로 직선이다(굵은 실선).

→ 점선은 결정 힘수의 값이 1 또는 -1 인 점들을 나타낸다. 이 선분은 결정 경계에 나란하고 일정한 거리만큼 떨어져서 마진을 형성하고 있다

선형 SVM 분류기를 훈련한다는 것은 마진 오하나도 발생하지 않거나 (하드 마진) 제한적인 마진 오류를 가지면서 (소프트 마진) 가능한 마진을 크게 하는 w를 찾는 것이다.

▼ 5.4.2 목적 함수

- 결정함수의 기율가는 가중치 벡터의 노름 $||w||$ 와 같다, 이 기울기를 2로 나누면 결정 함수의 값이 ± 1이 되는 점둘이 결정 경계로부터 2배만큼 더 멀어진다. 즉, 기울기를 2로 나누는 것은 마진에 2를 곱하는 것과 같다.



-

  → 가중치 벡터 w가 작을수록 마진은 커진다 .

- **하드 마진 선형 SVM 분류기의 목적 함수**

: 마진을 크게 하기 위해 IIwII 를 최소화 하고자 한다. 다항 마진 오류를 하나도 만들지 않으려면 (하드 마진) , 결정 힘수가 모든 양성 훈련 샘플에서는 1보다 커야 하고 음성 훈련 샘플에서는 - 1보다 작아야 한다.

음성 샘플 ($y^{(i)} = 0$일 때 $t^{(i)} = -1$로, 양성 샘플 ($y^{(i)} = 1$일 때 $t^{(i)} = 1$ 로 정의하면 앞서 말한 제약 조건을 모든샘플 에서 $t^{(i)}\left(w^T x^{(i)} + b\right) \geq 1$로 표현할 수 있다. 따라서 하드 마진 선형 SVM 분류기의 목적함수를 다음과 같이 제약이 있는 최적화 문제로 표현할 수 있다.

식 5-3 하드 마진 선형 SVM 분류기의 목적 함수

$$\underset{\mathbf{w}, b}{\text{minimize}} \ \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

[조건] $i = 1, 2, \cdots, m$일 때 $\quad t^{(i)}\left(\mathbf{w}^T\mathbf{x}^{(i)} + b\right) \geq 1$

- 소프트 마진 분류기의 목적 함수

식 5-4 소프트 마진 선형 SVM 분류기의 목적 함수[20]

$$\underset{w, b, \zeta}{\text{minimize}} \ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m}\zeta^{(i)}$$

[조건] $i = 1, 2, \cdots, m$일 때 $\quad t^{(i)}\left(\mathbf{w}^T\mathbf{x}^{(i)} + b\right) \geq 1 - \zeta^{(i)}$이고 $\quad \zeta^{(i)} \geq 0$

▼ 5.4.3 콰드라틱 프로그래밍

콰드라틱 프로그래밍(QP)

: 하드 마진과 소프트 마진 문제 처럼 선형적인 제약 조건이 있는 볼록 힘수의 이차 최적화 문제를 말한다. 문제의 공식은 다음과 같다.

식 5-5 QP 문제

$$\underset{\mathbf{p}}{\text{minimize}}\ \frac{1}{2}\mathbf{p}^T\mathbf{H}\mathbf{p}+\mathbf{f}^T\mathbf{p}$$

[조건] $\mathbf{A}\mathbf{p}\le\mathbf{b}$

여기서
- $\mathbf{p}$는 n_p차원의 벡터(n_p = 모델 파라미터 수)
- $\mathbf{H}$는 $n_p \times n_p$ 크기 행렬
- $\mathbf{f}$는 n_p차원의 벡터
- $\mathbf{A}$는 $n_c \times n_p$ 크기 행렬 (n_c = 제약 수)
- $\mathbf{b}$는 n_c차원의 벡터

이때, 다음과 같이 QP 파라미터를 지정하면 하드 마진을 갖는 선형 SVM 분류기의 목적 함수를 간단하게 검증할 수 있다.

- $n_p = n+1$, 여기서 n은 특성 수입니다(편향 때문에 +1이 추가되었습니다).
- $n_c = m$, 여기서 m은 훈련 샘플 수입니다.
- $\mathbf{H}$는 $n_p \times n_p$ 크기이고 왼쪽 맨 위의 원소가 0(편향을 제외하기 위해)인 것을 제외하고는 단위행렬입니다.
- $\mathbf{f} = \mathbf{0}$, 모두 0으로 채워진 n_p차원의 벡터입니다.
- $\mathbf{b} = \mathbf{1}$, 모두 1로 채워진 n_c차원의 벡터입니다.
- $a^{(i)} = -t^{(i)}\dot{\mathbf{x}}^{(i)}$, 여기서 $\dot{\mathbf{x}}^{(i)}$는 편향을 위해 특성 $\dot{\mathbf{x}}_0 = 1$을 추가한 $\mathbf{x}^{(i)}$와 같습니다.

하드 마진 선형 SVM 분류기 훈련

: 이미 준비되어 있는 QP 알고리즘에 관련 파라미터를 전달한다. 결과 벡터 p는 편향 $b = p_0$ 와 특성 가중치 $w_i = p_i$ ($i = 1, 2..,$)을 담고 있다.

- 소프트 마진 문제에서도 비슷하게 QP 알고리즘을 사용할 수 있다.
- 커널 트력을 사용하려면 제약이 있는 최적화 문제를 다른 형태로 바꿔야한다.

▼ 5.4.4 쌍대 문제

- 원문제는 제약이 있는 최적화 문제가 주어지면 쌍대 문제라고 하는 깊게 관련된 다른 문제로 표현할 수 있다. 일반적으로 쌍대 문제 해는 원 문제 해의 하한값이지만, 어떤 조건하에서는 원 문제와 똑같은 해를 제공한다. SVM는 이 조건을 만족한다. 따라서 원 문제 또는 쌍대 문제 중 하나를 선택하여 풀 수 있다. 둘 다 같은 해를 제공한다.

식 5-6 선형 SVM 목적 함수의 쌍대 형식

$$\underset{\alpha}{\text{minimize}} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha^{(i)}\alpha^{(j)}t^{(i)}t^{(j)}\mathbf{x}^{(i)^T}\mathbf{x}^{(j)} - \sum_{i=1}^{m}\alpha^{(i)}$$

[조건] $i = 1, 2, \cdots, m$일 때 $\alpha^{(i)} \geq 0$

이 식을 최소화하는 벡터 $\hat{\alpha}$ 을 찾았다면 [식 5-7J을 사용해 원 문제의 식을 최소화히는 $\hat{w}$과 $\hat{b}$을 계산할 있다.

식 5-7 쌍대 문제에서 구한 해로 원 문제의 해 계산하기

$$\hat{\mathbf{w}} = \sum_{i=1}^{m}\hat{\alpha}^{(i)}t^{(i)}\mathbf{x}^{(i)}$$

$$\hat{b} = \frac{1}{n_s}\sum_{\substack{i=1 \\ \hat{\alpha}^{(i)}>0}}^{m}\left(t^{(i)} - \hat{\mathbf{w}}^T\mathbf{x}^{(i)}\right)$$

훈련 샘플 수가 특성 개수보다 작을 때 원 문제보다 쌍대 문제를 푸는것이 더 빠르다. 중요한 것은 원 문제에서는 적용이 안 되는 널 트릭을 가능하게 한다.

▼ 5.4.5 커널 SVM

2치원 데이터셋에 차 다항식 변환을 적용하고, 선형 SVM 분류기를 변환된 이 훈련 세트에 적용한다고 하자. [식 5-8] 은 적용하고자 하는 2차 다항식 매핑함수 ϕ이다.

식 5-8 2차 다항식 매핑

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

변환된 벡터는 2차원이 아니고 3치원이 된다. 두 개의 2차원 벡터 a와 b에 2차 다항식 매핑을 적용한 디음 변환된 벡터로 점곱을 하면 다음과 같다.

식 5-9 2차 다항식 매핑을 위한 커널 트릭

$$\phi(\mathbf{a})^T \phi(\mathbf{b}) = \begin{pmatrix} a_1^2 \\ \sqrt{2}a_1a_2 \\ a_2^2 \end{pmatrix}^T \begin{pmatrix} b_1^2 \\ \sqrt{2}b_1b_2 \\ b_2^2 \end{pmatrix} = a_1^2b_1^2 + 2a_1b_1a_2b_2 + a_2^2b_2^2$$

$$= (a_1b_1 + a_2b_2)^2 = \left(\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}^T \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right)^2 = (\mathbf{a}^T \mathbf{b})^2$$

변환된 벡터의 점곱이 원래 벡터의 점곱의 제곱과 같다.

$$\phi(\mathbf{a})^T \phi(\mathbf{b}) = (\mathbf{a}^T \mathbf{b})^2$$

핵심은, 모든 훈련 샘플에 변환 ϕ를 적용하면 쌍대 문제(식 5-6) 에 점곱 $\phi(x_{(i)})^T \phi(x_{(j)})$ 가 포함될 것 이다 . 하지만 ϕ가 식[5-8]에 정의된 2차 다항식 변환이라면 변환된 벡터와 점곱을 간단하게 바꿀 수 있다. 그래서 실제로 훈련 샘플을 변환할 필요가 없다. 즉, 식 [5-6]에 있는 점곱을 제곱으로 바꾸면 된다. 결과값은 실제로 훈련 샘플을 어렵게 변환해서 선형 SVM 알고라즘을 적용하는 것과완전히 같다. 하지만 이 기법이 전체 과정에 필요한 계산량 측면에서 훨씬 효율적이다.

다항식 커널

: 함수 $K(a, b) = (a^T b)^2$ 을 2차 다항식 커널이라고 한다. 머신러닝에서 커널은 변환 ϕ 를 계산하지 않고 (또는 ϕ를 모르더라도) 원래 벡터 a와 $$b$에 기반하여 점곱 $\phi(a)^T \phi(b)$ 를 계산할 수는 함수다. 다음은 [식 10] 에 가장 널리 사용되는 커널의 일부이다.

식 5-10 일반적인 커널

선형: $K(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T \mathbf{b}$

다항식: $K(\mathbf{a}, \mathbf{b}) = (\gamma \mathbf{a}^T \mathbf{b} + r)^d$

가우시안 RBF: $K(\mathbf{a}, \mathbf{b}) = \exp(-\gamma \|\mathbf{a} - \mathbf{b}\|^2)$

시그모이드[26]: $K(\mathbf{a}, \mathbf{b}) = \tanh(\gamma \mathbf{a}^T \mathbf{b} + r)$

▼ 5.4.6 온라인 SVM