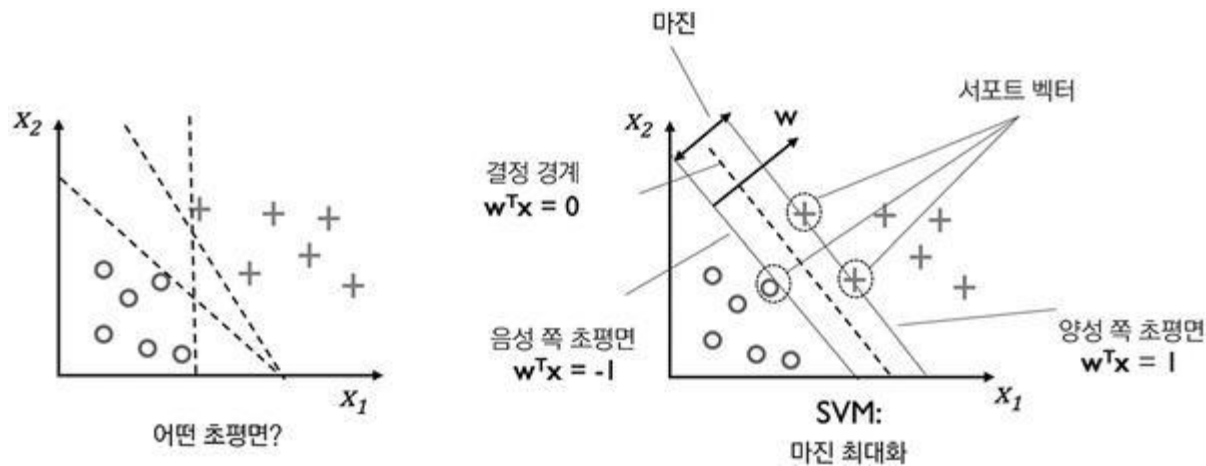


## chapter 5

### 서포트 벡터 머신

서포트 벡터 머신 support vector machine (SVM)은 매우 강력하고 선형이나 비선형 분류, 회귀, 이상치 탐색에도 사용할 수 있는 다목적 머신러닝 모델

→복잡한 분류 문제에 잘 들어맞으며 작거나 중간 크기의 데이터셋에 적합합니다.

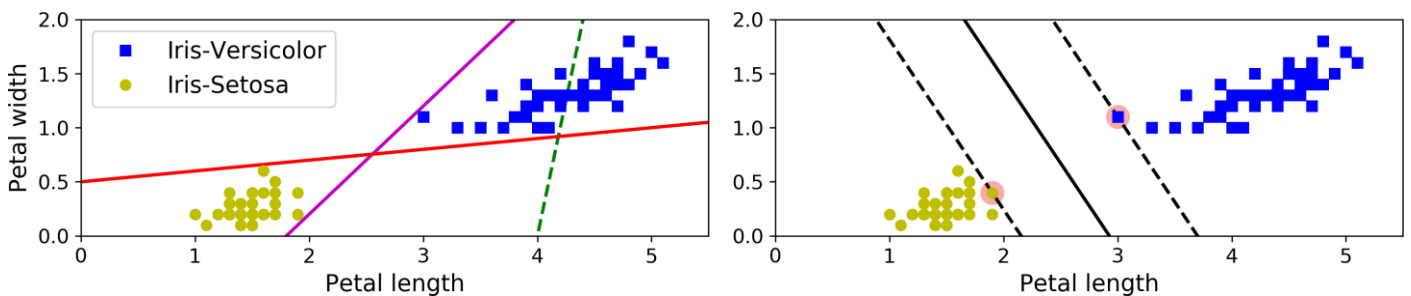


초평면 : 클래스를 구분하는 결정 경계 (n-1차원을 가짐)

마진 margin : 초평면에 가장 가까운 훈련 샘플 사이의 거리

서포트 벡터 support vector : 초평면에 가장 가까운 몇 개의 샘플 포인트

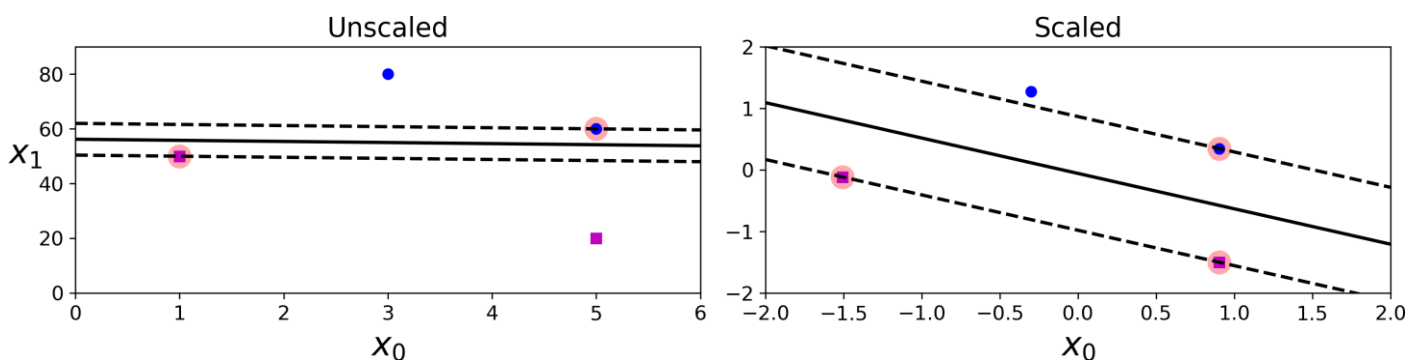
### 5.1 선형 SVM 분류



[그림 5-1]

오른쪽 그래프의 실선 : SVM 분류기의 결정경계, 두 개의 클래스를 나누고 있을 뿐만 아니라 제일 가까운 훈련 샘플로부터 가능한 한 멀리 떨어져 있다. → 클래스 사이에 가장 폭이 넓은 도로를 찾는 것

→ 라진 마진 분류 large margin classification 라고 한다.



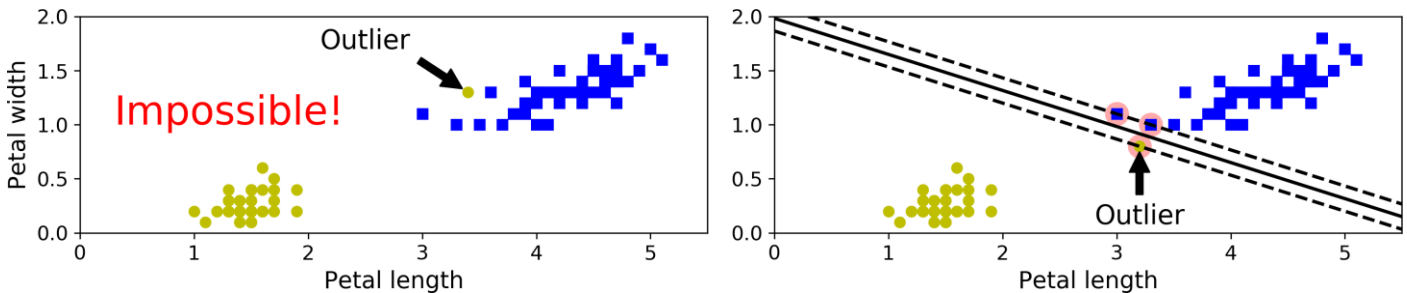
왼쪽 : 수직축의 스케일 > 수평축의 스케일 → 가장 넓은 도로가 거의 수평에 가까움 (도로의 간격을 수직으로 맞추려고 할 것이기 때문)

오른쪽 : 특성의 스케일을 조절하여 결정 경계가 훨씬 좋아진다.

### 5.1.1. 소프트 마진 분류

**하드 마진 분류** **hard margin classification** : 모든 샘플이 도로 바깥쪽에 올바르게 분류됨

문제점 : 데이터가 선형적으로 구분되어야 제대로 작동, 이상치에 민감



왼쪽 그래프 : 데이터셋에 이상치가 하나 있어 하드 마진을 찾을 수 없다.

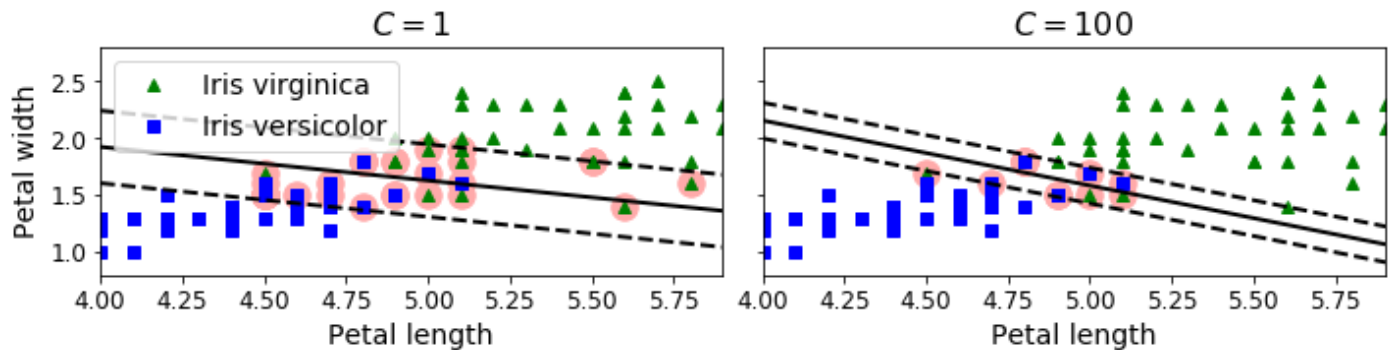
오른쪽 그래프 : [그림 5-1]의 결정 경계와 매우 다르고 일반화가 잘 될 것 같지 않음

**소프트 마진 분류** **soft margin classification**

도로의 폭을 가능한 한 넓게 유지하는 것과 **마진 오류** **margin violation** (샘플이 도로 중간이나 반대쪽에 있는 경우) 사이에 적절한 균형을 잡는다.

Scikit-Learn의 SVM 모델의 하이퍼 파라미터 C

C : Regularization parameter. The strength of the regularization is inversely proportional to C.



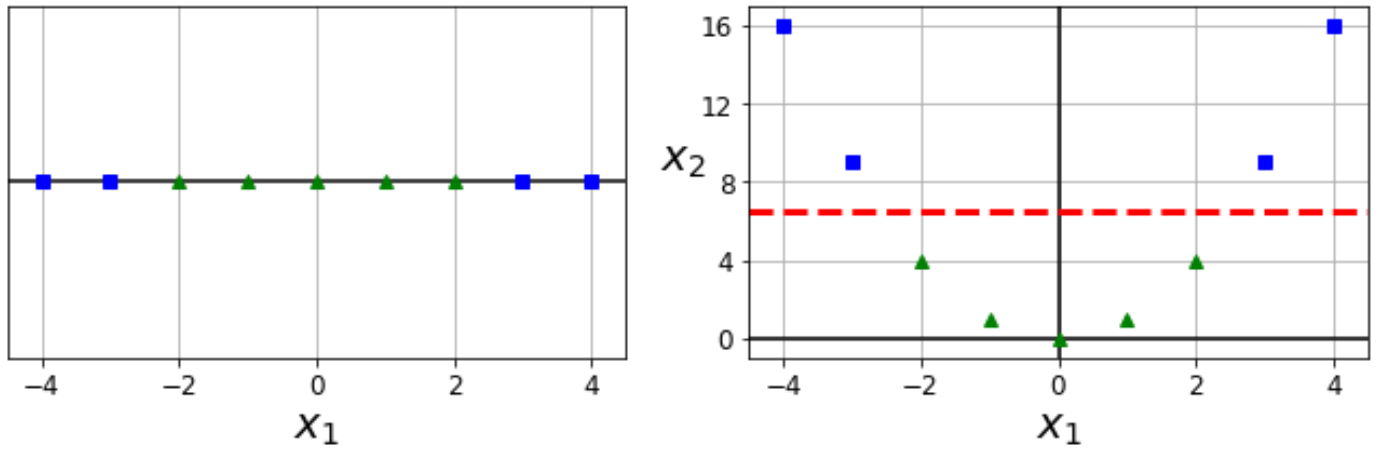
C를 낮게 설정 : 넓은 마진, 마진 오류가 많지만 일반화가 더 잘됨, 규제가 커짐

C를 높게 설정 : 좁은 마진, 적은 마진 오류, 규제가 낮아짐

→ SVM모델이 과대적합이라면 C를 감소시켜 모델을 규제할 수 있다.

## 5.2 비선형 SVM 분류

다항 특성과 같은 특성을 더 추가하여 비선형 데이터셋을 다룬다. → 선형적으로 구분되는 데이터셋이 만들어짐



[그림 5-5]

왼쪽 그래프 : 하나의 특성  $x_1$ 만을 가진 간단한 데이터셋 → 선형적으로 구분이 안 된다.

오른쪽 그래프 : 두 번째 특성  $x_2 = (x_1)^2$ 을 추가하여 만들어진 2차원 데이터셋 → 선형적으로 구분 가능

## 5.2.1 다항식 커널

`SVC(kernel='poly')`

다항식 특성을 추가하는 것은 간단하고 모든 머신러닝 알고리즘에서 잘 작동하지만

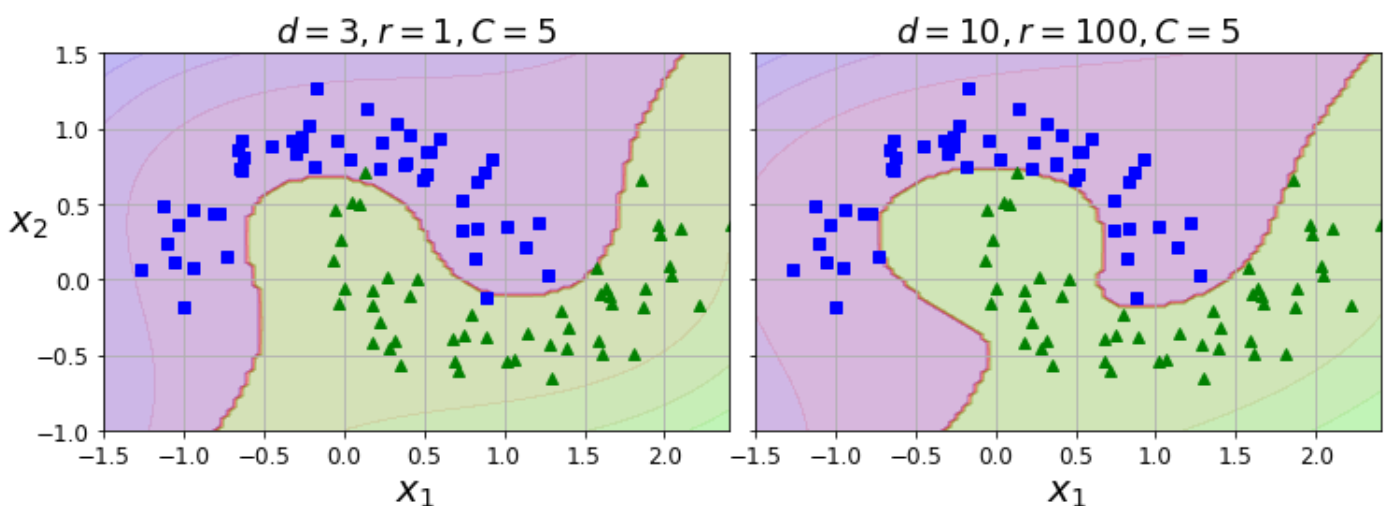
낮은 차수의 다항식 : 매우 복잡한 데이터셋을 잘 표현하지 못함

높은 차수의 다항식 : 굉장히 많은 특성을 추가 → 모델을 느리게만들

### 커널 트릭 kernel trick

실제로는 특성을 추가하지 않으면서 다항식 특성을 많이 추가한 것과 같은 결과를 얻을 수 있음

→ 사실 어떤 특성도 추가하지 않기 때문에 엄청난 수의 특성 조합이 생기지 않음



왼쪽 그래프 : 3차 다항식 커널을 사용한 SVM 분류기

오른쪽 그래프 : 10차 다항식 커널을 사용한 SVM 분류기

→ 모델이 과대적합 : 차수를 줄임 | 모델이 과소적합 : 차수를 늘려야 함

매개변수  $r(\text{coef0})$  : 모델이 높은 차수와 낮은 차수에 얼마나 영향을 받을지 조절한다.

다항식 커널

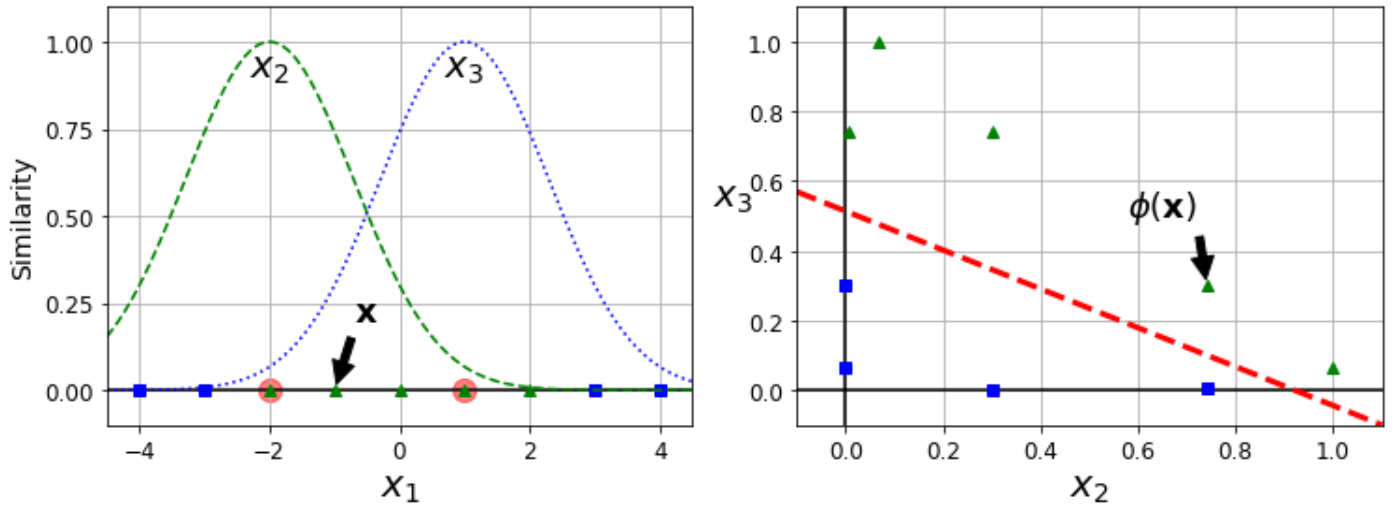
$$K(\mathbf{a}, \mathbf{b}) = (\gamma \mathbf{a}^T \mathbf{b} + r)^d$$

위 식을 보다시피 다항식 커널은 차수가 높아질수록 1보다 작은 값과 1보다 큰 값의 차이가 크게 벌어지므로  $r$ 을 적절한 값으로 지정하면 고차항의 영향을 줄일 수 있다.

### 5.2.2 유사도 특성

`SVC(kernel='rbf')`

각 샘플이 특정 **랜드마크 landmark**와 얼마나 닮았는지 측정하는 **유사도 함수 similarity function**로 계산한 특성을 추가하는 것.



[그림 5-8]

ex)

앞의 [그림 5-5]의 왼쪽 그래프를 나타내는 1차원 데이터셋에 두 개의 랜드마크  $x_1 = -2$ 와  $x_1 = 1$ 을 추가한다. 그리고  $\gamma = 0.3$ 인 가우시안 **방사 기저 함수 radial basis function (RBF)**를 유사도 함수로 정의한다.

가우시안 RBF

$$\phi_{\gamma}(x, l) = \exp(-\gamma \|x - l\|^2) = e^{(-\gamma \|x - l\|^2)}$$

$l$  : 랜드마크 지점,  $\gamma$  : 0보다 커야 하며 값이 작을수록 폭이 넓은 종 모양

→ 방사 기저 함수의 값 : 0(랜드마크에서 멀리 떨어짐)부터 1(랜드마크와 같은 위치)까지 변화하며 종 모양으로 나타남

$x_1 = 1$  샘플을 살펴 보았을 때 첫번째, 두번째 랜드마크에서 각각 1, 2만큼 떨어져 있으므로 새로 만든 특성은  $x_2 = \exp(-0.3 \times 1^2) \approx 0.74$ ,  $x_3 = \exp(-0.3 \times 2^2) \approx 0.30$ 이다. 오른쪽 그래프는 변환된 데이터셋을 보여준다

→ 이제 선형적으로 구분 가능, 멀리 떨어진 샘플들이 0에 가깝게 나타나게 된다.

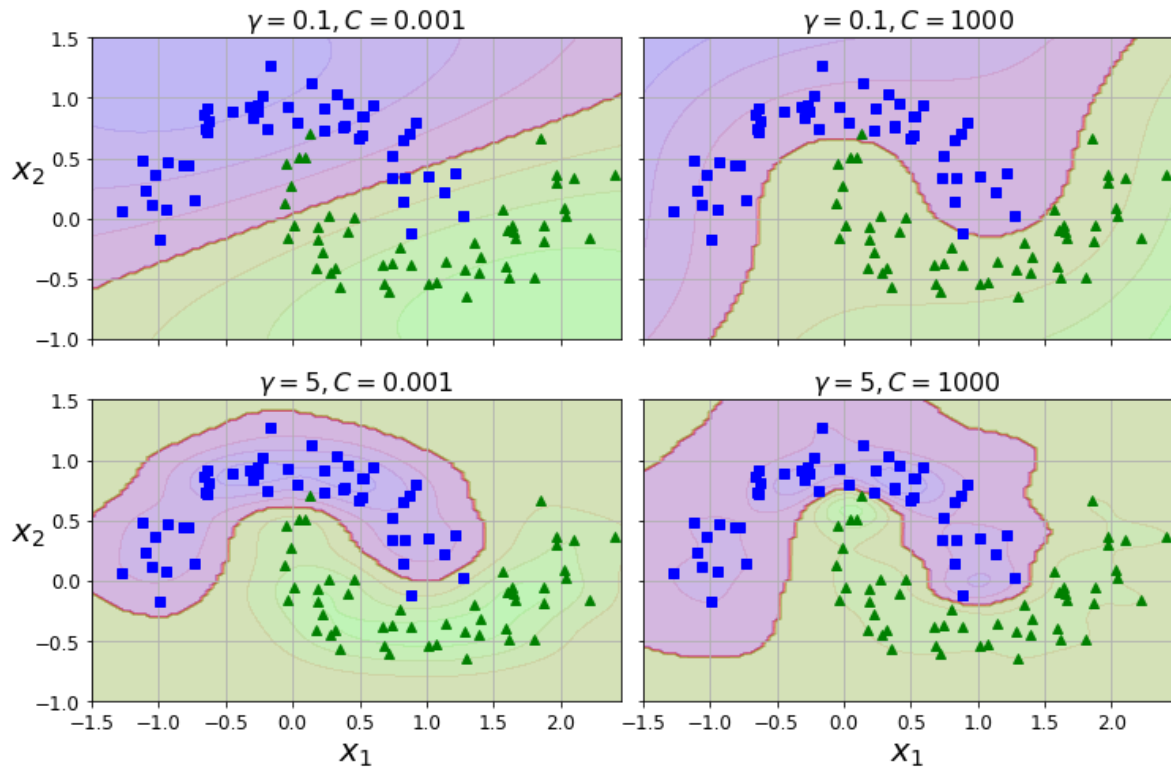
랜드마크 선택

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

단점 : 훈련 세트에 있는  $n$ 개의 특성을 가진  $m$ 개의 샘플이  $m$ 개의 특성을 가진  $m$ 개의 샘플로 변환된다는 것이다. (원본 특성은 제외한다고 가정) 훈련 세트가 매우 클 경우 동일한 크기의 아주 많은 특성이 만들어짐.

### 5.2.3 가우시안 RBF 커널

추가 특성을 모두 계산하려면 연산 비용이 많이 드는데, 이때 커널 트릭을 이용하여 유사도 특성을 많이 추가하는 것과 같은 비슷한 결과를 얻을 수 있다.



gamma( $\gamma$ ) 증가 → 종 모양 그래프가 좁아져서, 각 샘플의 영향 범위가 작아짐, 결정 경계가 조금 더 불규칙해지고 각 샘플을 따라 구불구불하게 휘어진다.

gamma( $\gamma$ ) 감소 → 넓은 종 모양 그래프를 만들며 샘플이 넓은 범위에 걸쳐 영향을 주므로 결정 경계가 더 부드러워짐

→하이퍼 파라미터  $\gamma$ 가 규제의 역할을 한다. (하이퍼 파라미터 C와 비슷하다)

모델이 과대적합 →  $\gamma$ 를 감소시켜야 한다.

모델이 과소적합 →  $\gamma$ 를 증가시켜야 한다.

어떤 커널을 사용할까

선형 커널을 먼저 시도한다. 특히 훈련 세트가 아주 크거나 특성 수가 많은 경우

훈련 세트가 너무 크지 않다면 가우시안 RBF 커널도 시도해보면 좋다. → 대부분의 경우 잘 들어맞음.

그 외 시간, 성능이 충분하다면 다른 커널을 좀 더 시도해볼 수 있다.

## 5.2.4 계산복잡도

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

정밀도를 높이면 알고리즘의 수행 시간이 길어진다. 이는 허용오차 하이퍼파라미터  $\epsilon$ 으로 조절한다.

LinearSVC : 선형 SVM을 위한 최적화 알고리즘을 구현한 liblinear 라이브러리를 기반으로 한다.

커널 트릭을 지원하지 않지만 훈련 샘플과 특성 수에 거의 선형적으로 늘어난다.

정밀도를 높이면 수행 시간이 길어짐. → 허용오차 하이퍼파라미터  $\epsilon$ 으로 조절 (매개변수 tol=0.0001)

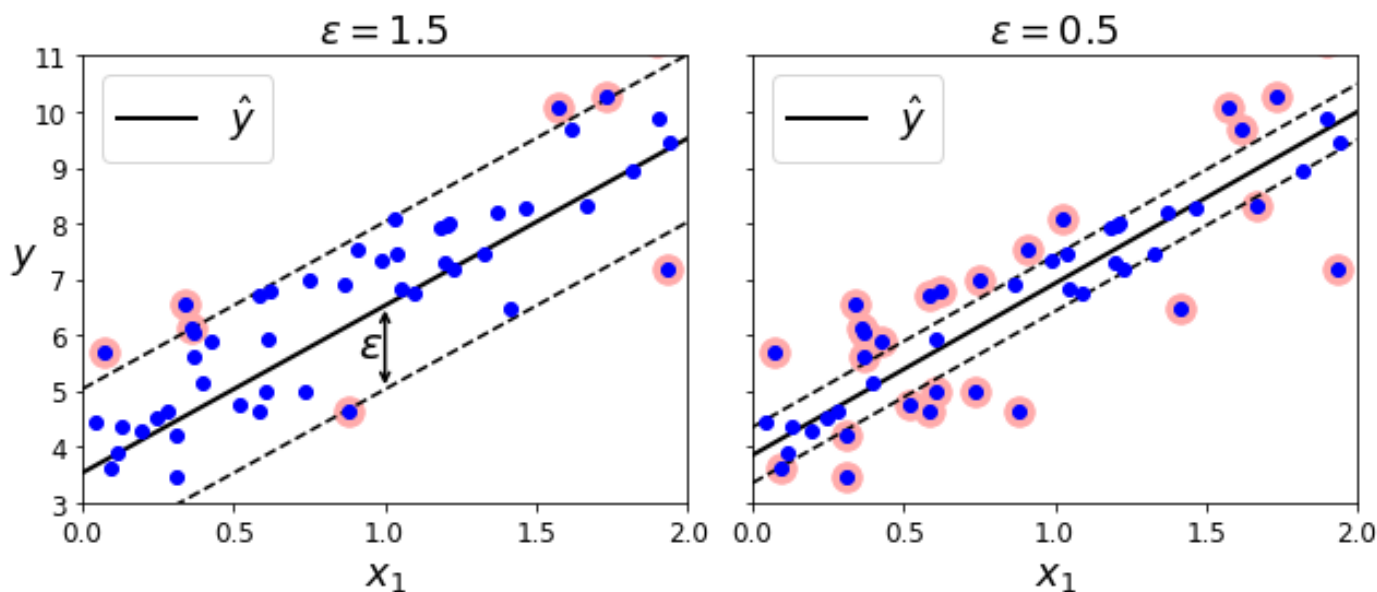
SVC: 커널 트릭 알고리즘을 구현한 libsvm 라이브러리를 기반으로 한다.

시간 복잡도가  $O(m^2 \times n) \sim O(m^3 \times n)$ 로 복잡하지만 작거나 중간 규모의 훈련 세트에 이 알고리즘이 잘 맞는다. 특성의 개수에는, 특히 **희소 특성 sparse features** (각 샘플에 0이 아닌 특성이 몇 개 없는 경우) 인 경우에는 잘 확장된다. → 알고리즘의 성능이 샘플이 가진 0이 아닌 특성의 평균 수에 거의 비례

### 5.3 SVM 회귀

일정한 마진 오류 안에서 두 클래스 간의 도로 폭이 가능한 한 최대가 되도록 하는 대신, SVM 회귀는 제한된 마진 오류(=도로 밖의 샘플) 안에서 도로 안에 가능한 한 많은 샘플이 들어가도록 학습한다.

도로의 폭은 하이퍼파라미터  $\varepsilon$ 으로 조절한다.( $\neq$ 허용오차의 하이퍼 파라미터  $\varepsilon$ )

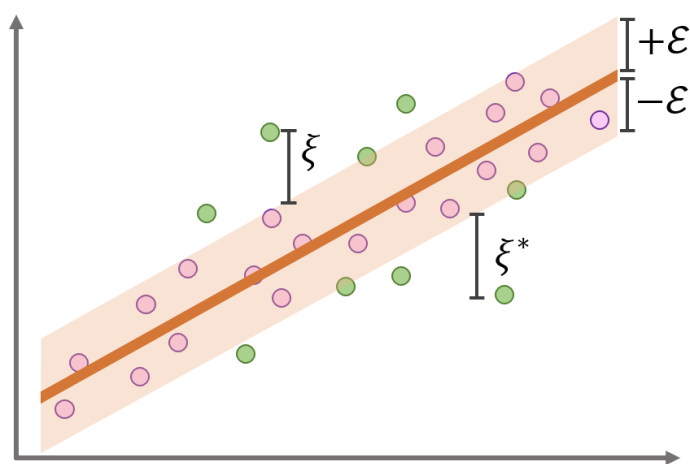


무작위로 생성한 선형 데이터셋에 훈련시킨 두 개의 선형 SVM 회귀 모델을 보여준다.

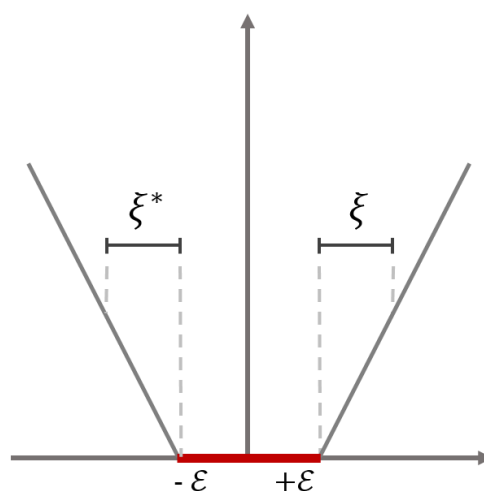
왼쪽 : 마진을 크게 ( $\varepsilon = 1.5$ )하여 만듦

오른쪽 : 마진을 작게 ( $\varepsilon = 0.5$ ) 하여 만듦

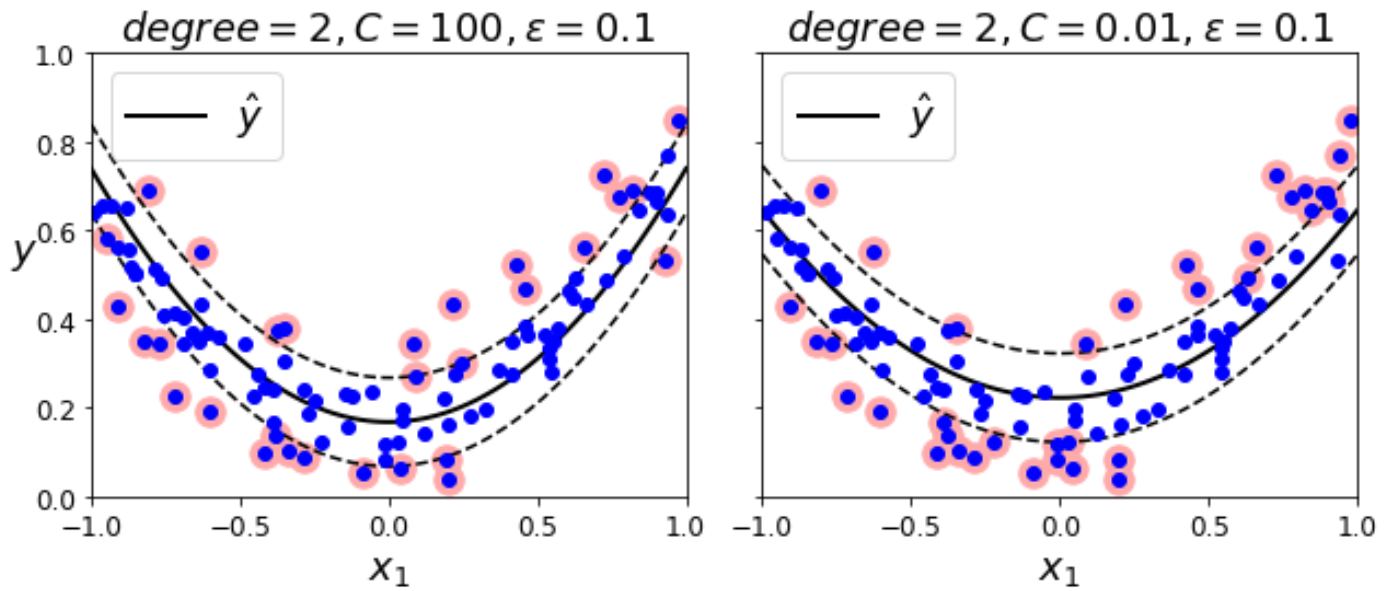
마진 안에서는 훈련 샘플이 추가되어도 모델의 예측에는 영향이 없다 →  $\varepsilon$ 에 민감하지 않다  $\varepsilon$ -insensitive



● : Support Vector



$\varepsilon$  - insensitive loss



[그림 5-11] : 임의의 2차방정식 형태의 훈련 세트에 2차 다항 커널을 사용한 SVM 회귀

**비선형 회귀 작업을 처리하려면 커널 SVM 모델을 사용한다.**

[그림 5-11]

왼쪽 그래프 : 규제가 거의 없다.(=아주 큰 C)

오른쪽 그래프 : 규제가 훨씬 많음(=작은 C)

LinearSVR : 필요한 시간이 훈련 세트의 크기에 비례해서 선형적으로 늘어남(=LinearSVC)

SVR : 훈련 세트가 커지면 훨씬 느려짐(=SVC)