

5장 서포트 벡터 머신 (1부)

주요 내용

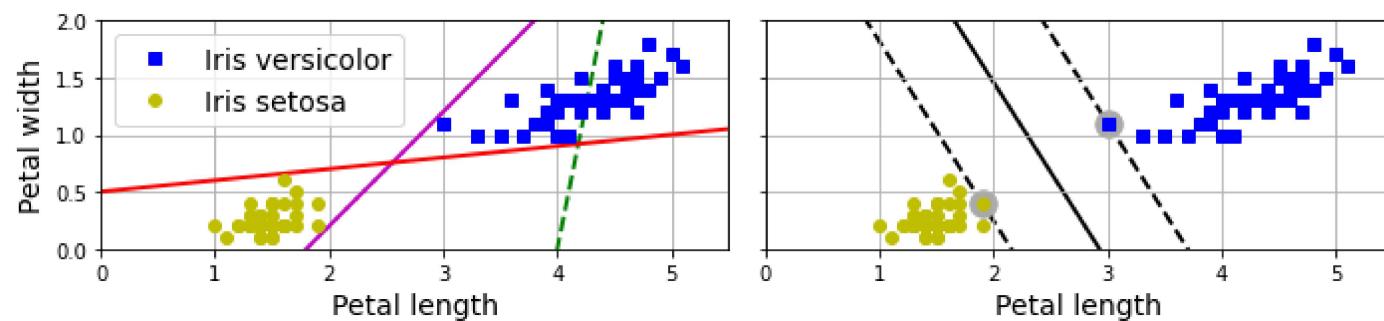
- 선형 SVM 분류
- 비선형 SVM 분류
- SVM 회귀
- SVM 이론 (2부)

5.1 선형 SVM 분류

선형 SVM 아이디어

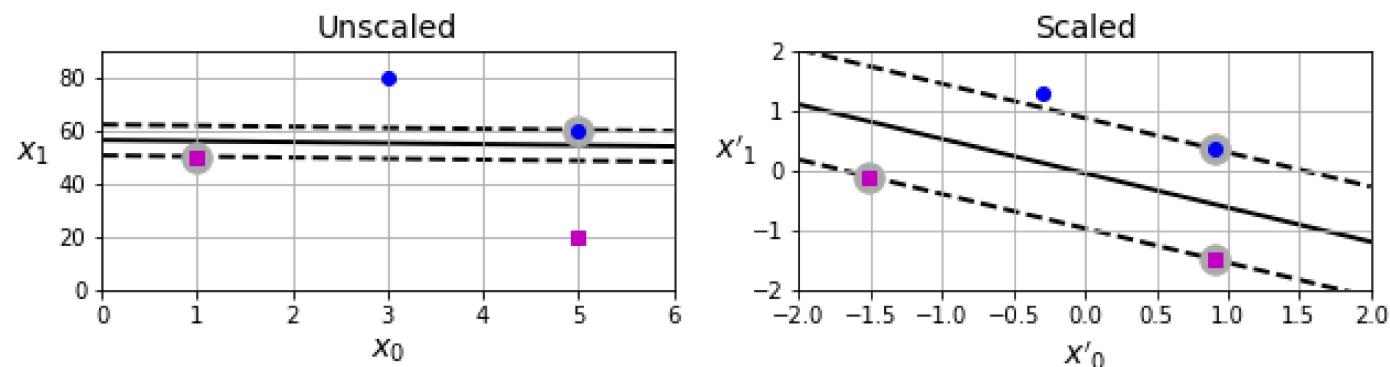
- 마진 margin: 클래스를 구분하는 도로의 폭
- 큰 마진 분류 large margin classification>: 마진을 최대로 하는 분류

	원편 그래프	오른편 그래프
분류기:	선형 분류	라지 마진 분류
실선:	결정 경계	결정 경계
일반화:	일반화 어려움	일반화 쉬움



서포트 벡터

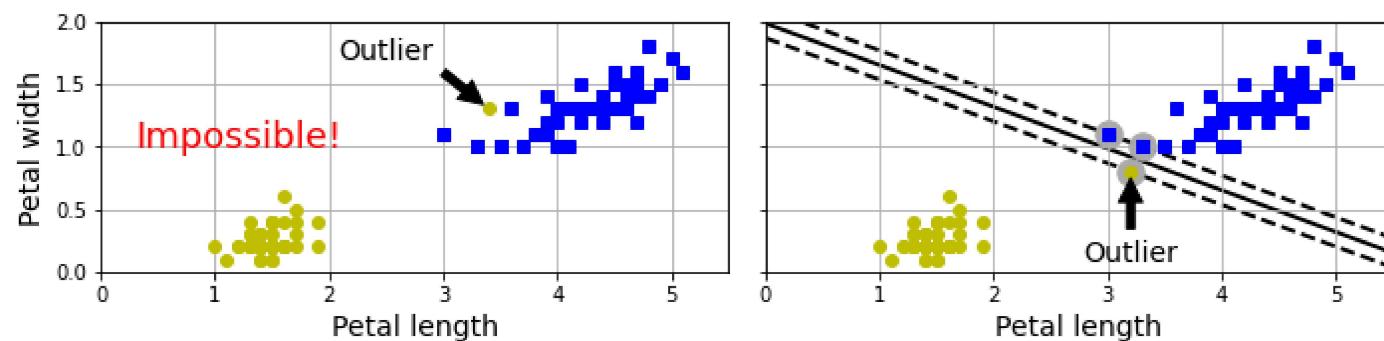
- 도로의 양쪽 경계에 위치하는 샘플 (아래 그림에서 동그라미 표시됨)
- 서포트 벡터 사이의 간격, 즉 마진이 최대가 되도록 학습
- 특성 스케일을 조정하면 결정경계가 훨씬 좋아짐.



하드 마진 분류

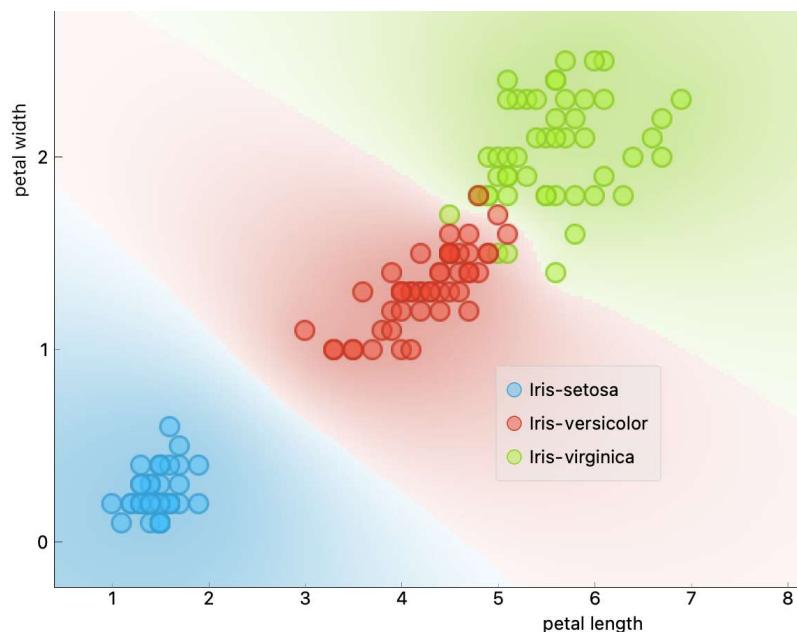
- 모든 훈련 샘플이 도로 바깥쪽에 올바르게 분류되도록 하는 마진 분류
- 훈련 세트가 선형적으로 구분되는 경우에만 가능
- 이상치에 민감함

	왼편 그래프	오른편 그래프
이상치:	타 클래스에 섞임	타 클래스에 매우 가까움
하드 마진 분류:	불가능	가능하지만 일반화 어려움



소프트 마진 분류

- 마진 오류를 어느 정도 허용하면서 도로의 폭을 최대로 넓게 유지하는 마진 분류
- **마진 오류** margin violation: 결정 경계 도로 위에 또는 결정 경계를 넘어 해당 클래스 반대 편에 위치하는 샘플. 소프트 마진 분류의 **서포트 벡터** 구성.
- 꽃잎 길이와 너비 기준의 버지니카와 버시컬러 품종 분류: 소프트 마진 분류만 가능



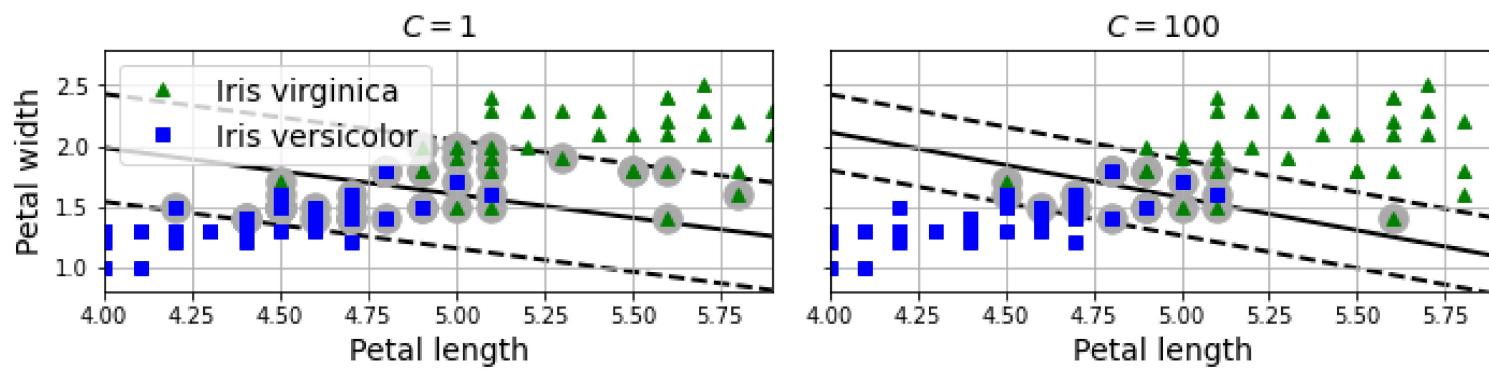
예제: 버지니까 품종 여부 판단

- 사이킷런의 선형 SVM 분류기 `LinearSVC` 활용
 - 데이터셋 표준화 스케일링이 중요해서 기본적으로 함께 사용.

```
svm_clf = make_pipeline(  
    StandardScaler(),  
    LinearSVC(C=1, random_state=42))
```

- 규제 하이퍼파라미터 `C`
 - 작을 수록 마진 오류를 강함
 - 클 수록 적은 규제: 모델의 자유도 증가. 마진(결정 경계 도로의 폭)이 작아져서 과대적합 가능성을 키움.
 - `C=float("inf")`로 지정하면 하드 마진 분류 모델이 됨.

	원편 그래프	오른편 그래프
C	작게	크게
마진	크게	작게
분류	덜 정교하게	보다 정교하게



선형 SVM 분류 지원 모델

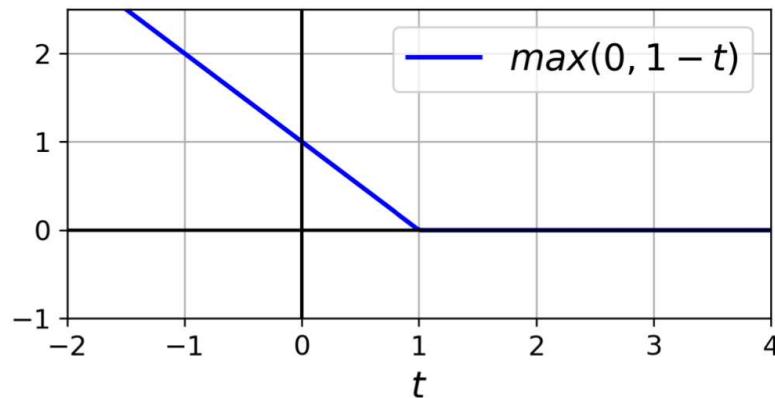
- 선형 분류는 LinearSVC 모델이 제일 빠르지만 'SVC + 선형 커널' 조합도 가능.

```
SVC(kernel="linear", C=1)
```

- SGDClassifier + hinge 손실함수 + 규제: 규제는 데이터셋 크기(m)에 반비례.

```
SGDClassifier(loss="hinge", alpha=1/(m*C))
```

- hinge 손실 함수: 어긋난 예측 정도에 비례하여 손실값이 선형적으로 커짐.



5.2 비선형 SVM 분류

방식 1: 특성 추가 + 선형 SVC

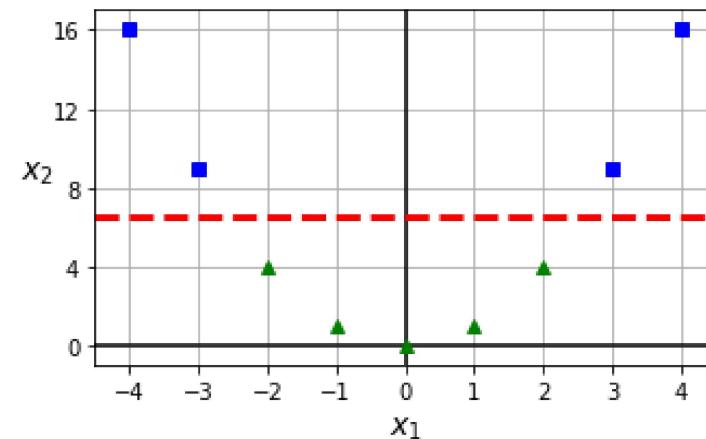
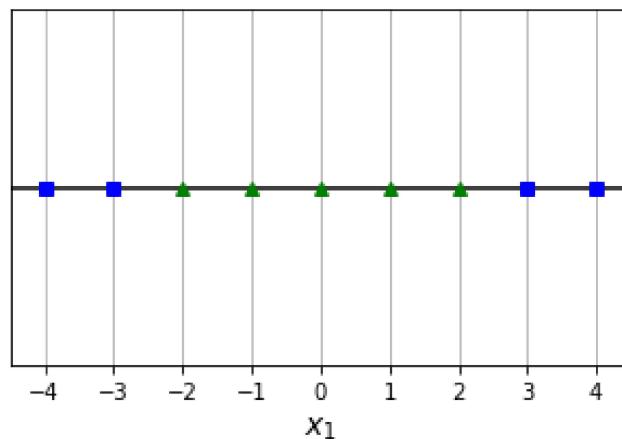
- 다항 특성 활용: 다항 특성을 추가한 후 선형 SVC 적용
- 유사도 특성 활용: 유사도 특성을 추가한 후 선형 SVC 적용

방식 2: SVC + 커널 트릭

- 커널 트릭: 새로운 특성을 실제로 추가하지 않으면서 동일한 결과를 유도하는 방식
- 다항 커널
- 가우시안 RBF(방사 기저 함수) 커널

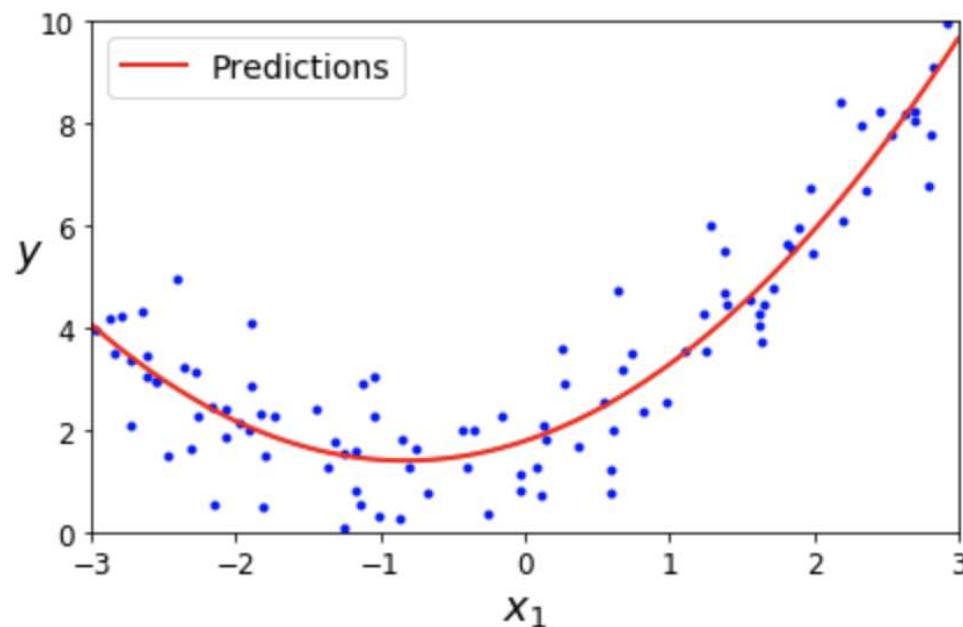
다항 특성 추가 + 선형 SVC

- 예제 1: 특성 x_1 하나만 갖는 모델에 새로운 특성 x_1^2 을 추가한 후 선형 SVM 분류 적용

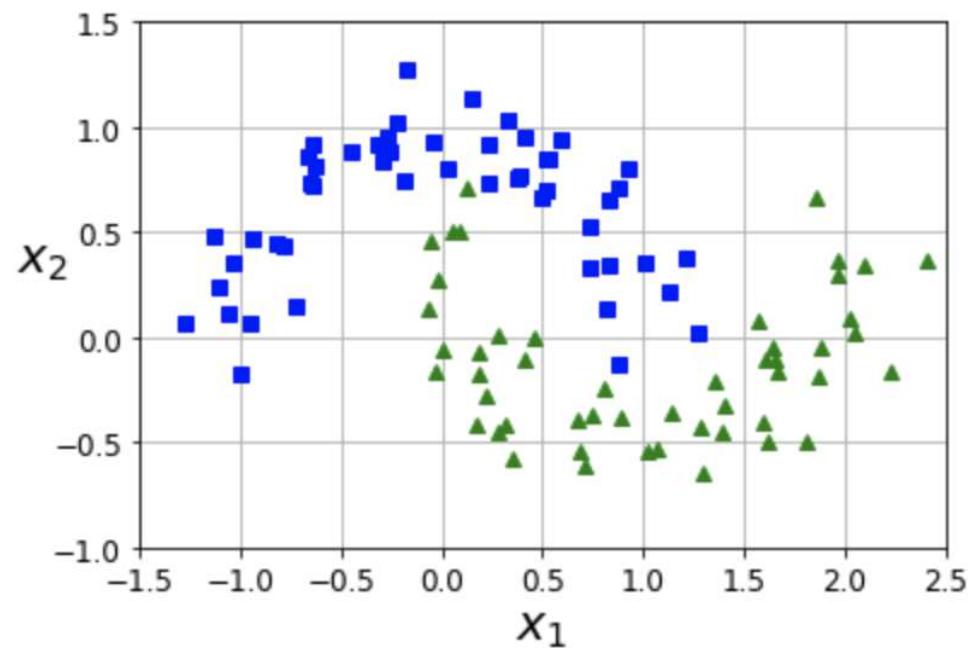


- 다항 특성 + 선형 회귀(4장) 방식과 유사한 기법

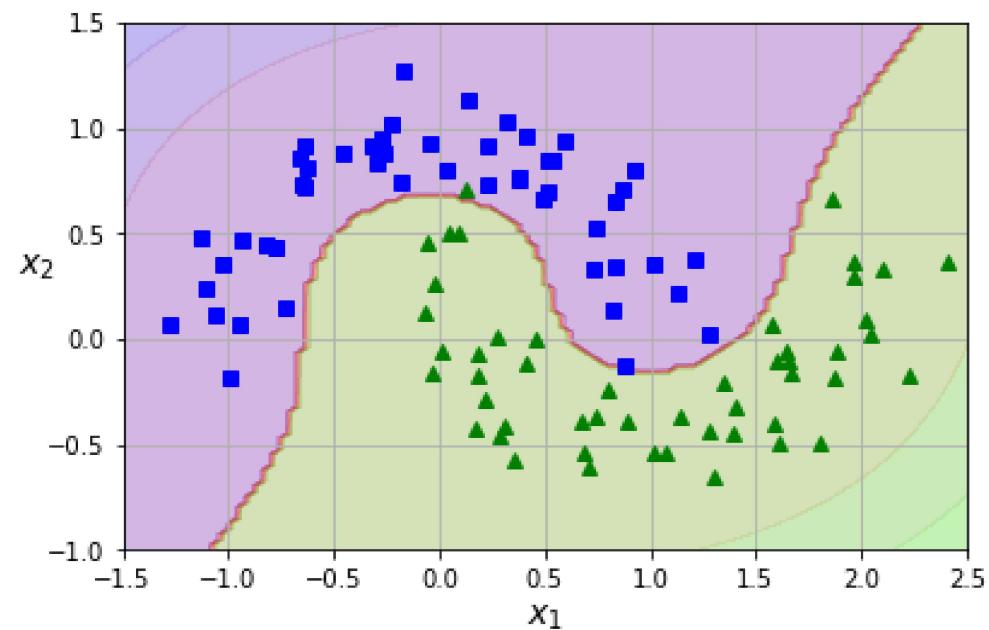
$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2$$



- 예제 2: moons 데이터셋. 마주보는 두 개의 반원 모양으로 두 개의 클래스로 구분되는 데이터



```
# 3차 항까지 추가
polynomial_svm_clf = make_pipeline(
    .... PolynomialFeatures(degree=3),
    .... StandardScaler(),
    .... LinearSVC(C=10, max_iter=10_000, random_state=42))
```



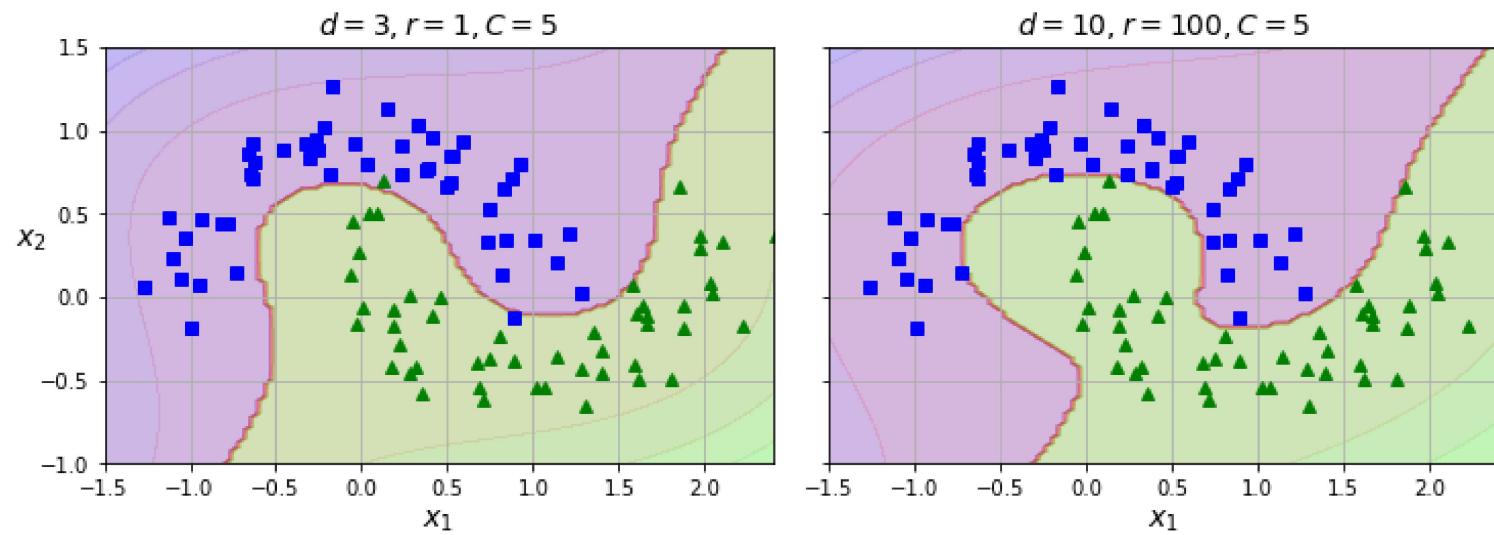
SVC + 다항 커널

- SVM 모델을 훈련시킬 때 다항 특성을 실제로는 추가하지 않으면서 수학적으로 추가한 효과 활용
- 예제: moons 데이터셋

```
poly_kernel_svm_clf = make_pipeline(  
    StandardScaler(),  
    SVC(kernel="poly", degree=3, coef0=1, C=5))
```

- `coef0=1`: 고차항의 중요도 지정. 아래 이미지에서 r_0 이 가리킴.

	원편 그래프	오른편 그래프
degree	3차 다항 커널	10차 다항 커널
coef0(r)	높은 차수 강조 조금	높은 차수 강조 많이



유사도 특성

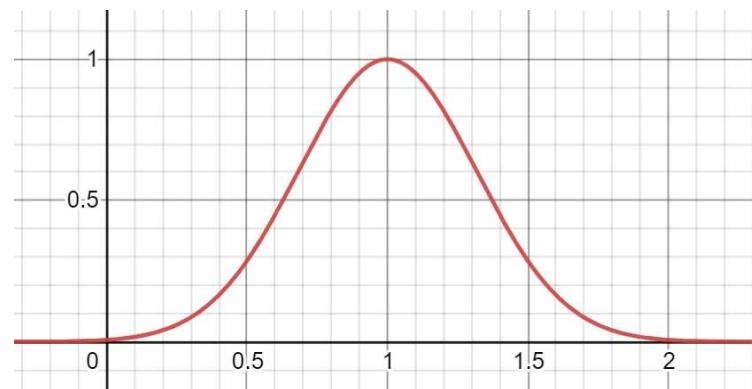
- 유사도 함수: 랜드마크 landmark 샘플과 각 샘플 사이의 유사도 similarity 측정
- 가우시안 방사 기저 함수(RBF, radial basis function)

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

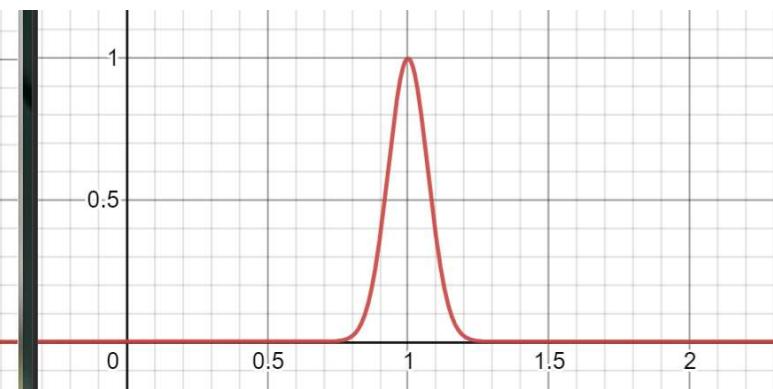
- ℓ : 랜드마크
- γ : 랜드마크에서 멀어질 수록 0에 수렴하는 속도를 조절함
- γ 값이 클수록 가까운 샘플 선호, 즉 샘플들 사이의 영향을 보다 적게 고려하여 모델의 자유도를 높이게 되어 과대적합 위험 커짐.

- 예제

$$\exp(-5 \|\mathbf{x} - 1\|^2)$$

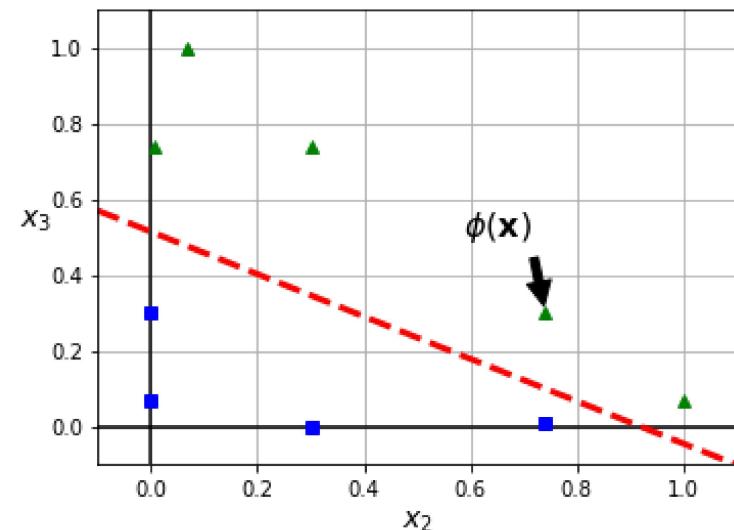
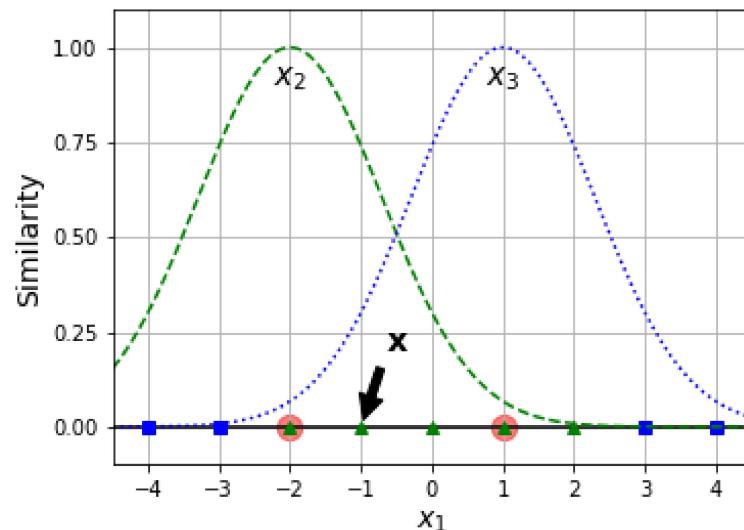


$$\exp(-100 \|\mathbf{x} - 1\|^2)$$



유사도 특성 추가 + 선형 SVC

- 예제
 - 랜드마크: -2와 1
 - x_2 와 x_3 : 각각 -2와 1에 대한 가우시안 RBF 함수로 계산한 유사도 특성
 - 화살표가 가리키는 점: $\mathbf{x} = -1$



랜드마크 지정

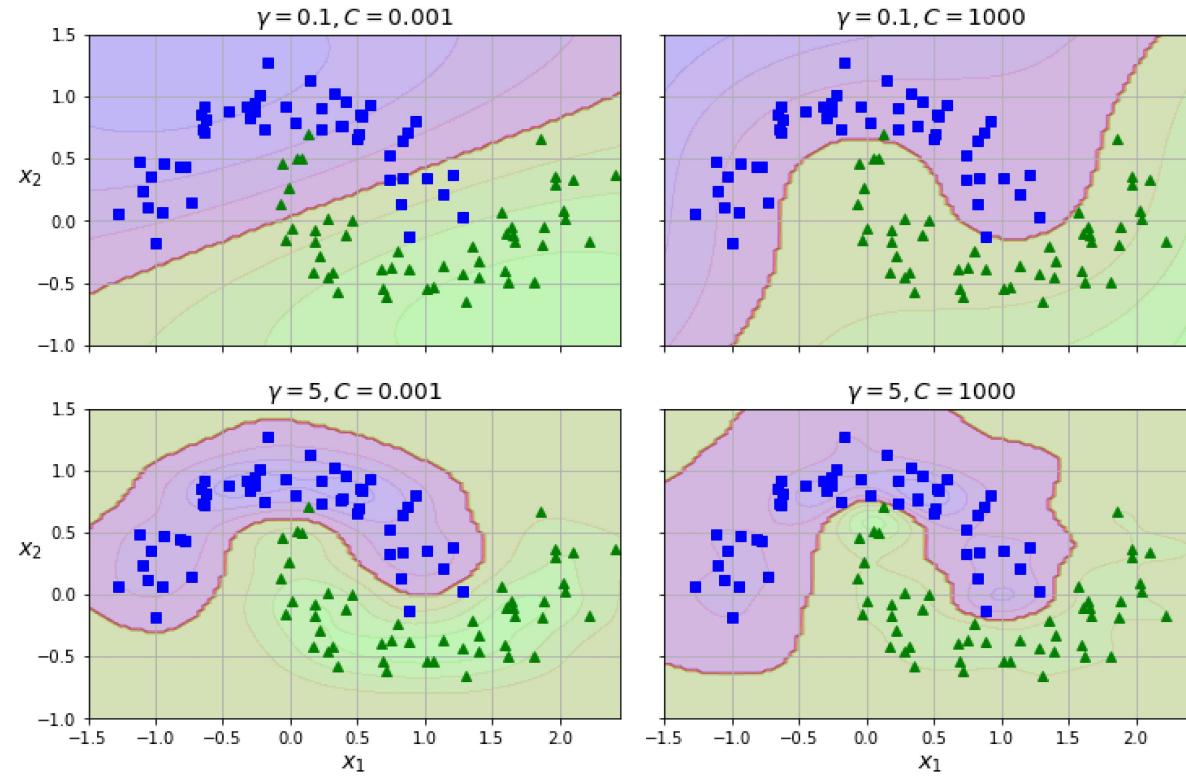
- 어떤 샘플을 랜드마크로 지정하면 좋은지 모름
- 따라서 모든 샘플을 랜드마크로 지정
- 장점: 차원이 커지면서 선형적으로 구분될 가능성이 높아짐.
- 단점: 훈련 세트가 매우 클 경우 동일한 크기의 아주 많은 특성이 생성됨.

SVC + 가우시안 RBF 커널

- 유사도 특성을 실제로는 추가 하지 않으면서 수학적으로 추가한 효과 활용

```
rbf_kernel_svm_clf = make_pipeline(  
    StandardScaler(),  
    SVC(kernel="rbf", gamma=5, C=0.001))
```

- 예제: moons 데이터셋



SVM 분류 모델 계산 복잡도

클래스	시간 복잡도(m 샘플 수, n 특성 수)	외부 메모리 학습	스케일 조정	커널	다중 클래스 분류
LinearSVC	$O(m \times n)$	미지원	필요	미지원	OvR
SVC	$O(m^2 \times n) \sim O(m^3 \times n)$	미지원	필요	지원	OvR
SGDClassifier	$O(m \times n)$	지원	필요	미지원	OvR

5.3 SVM 회귀

SVM 분류 vs. SVM 회귀

- SVM 분류
 - 목표: 마진 오류 발생 정도를 조절(C 이용)하면서 결정 경계 도로의 폭을 최대한 넓게 하기
 - 마진 오류: 도로 위에 위치한 샘플
- SVM 회귀
 - 목표: 마진 오류 발생 정도를 조절(C 이용)하면서 지정된 폭의 도로 안에 가능한 많은 샘플 포함하기
 - 마진 오류: 도로 밖에 위치한 샘플
 - 도로의 폭: epsilon 하이퍼파라미터로 지정

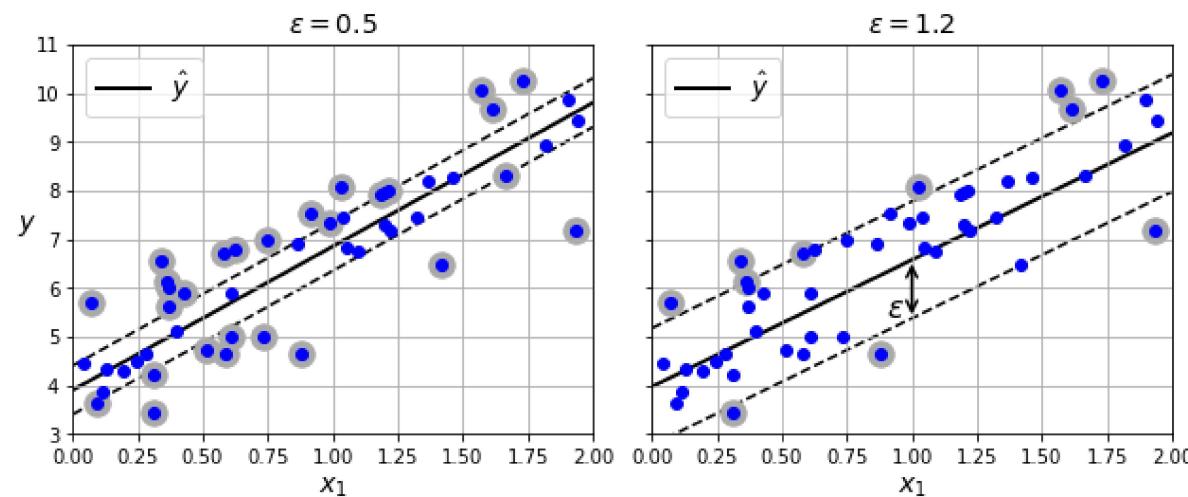
선형 SVM 회귀

- 예제: LinearSVR 활용. `epsilon` 은 도로의 폭 결정

```
svm_reg = make_pipeline(  
    StandardScaler(),  
    LinearSVR(epsilon=0.5, random_state=42))
```

epsilon 하이퍼파라미터

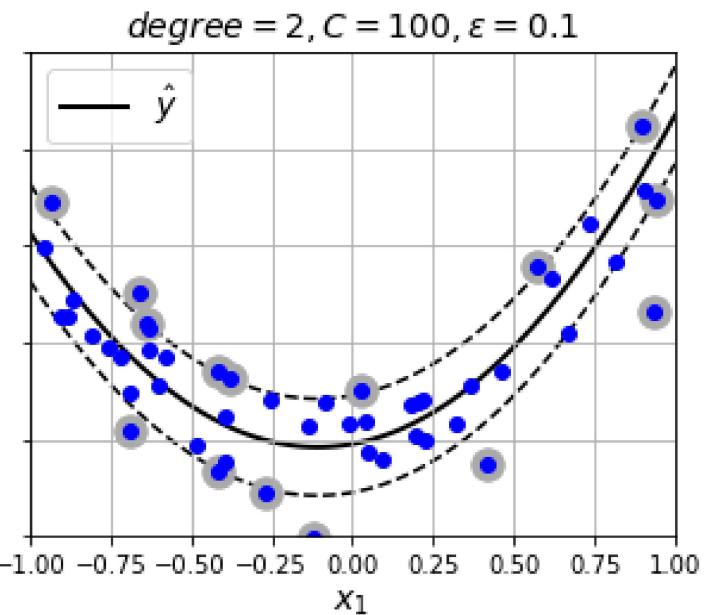
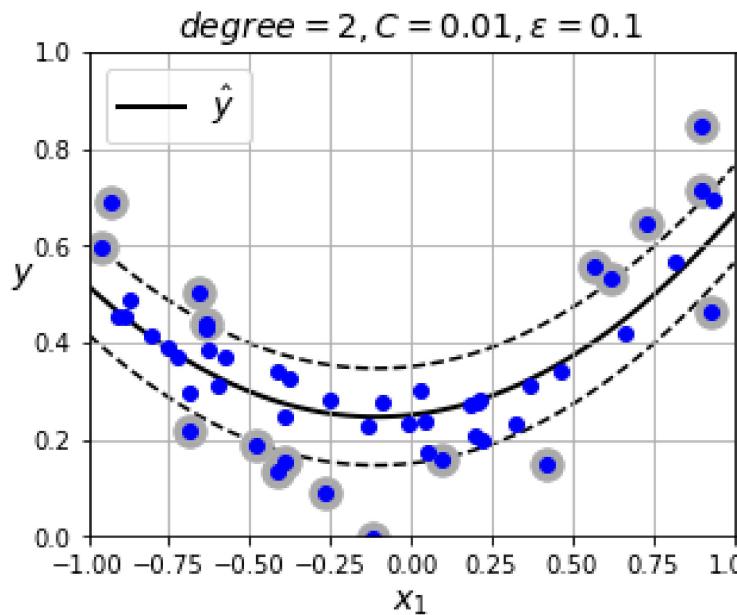
- 마진 안쪽, 즉 도로 위에 포함되는 샘플이 많아져도 예측에 영향 주지 않음. 이유는 마진 오류가 변하지 않기 때문. 즉 `epsilon` 만큼의 오차는 무시됨.
- `epsilon` 이 작을 수록 도로폭이 좁아지기에 보다 많은 샘플을 마진 안쪽으로 포함시키기 위해 도로의 굴곡이 심해짐. 따라서 `epsilon` 이 클 수록 규제가 약해지는 효과 발생.



비선형 SVM 회귀

- SVC와 동일한 커널 트릭을 활용하여 비선형 회귀 모델 구현
- 예제: SVR + 다항 커널

```
# SVR + 다항 커널
svm_poly_reg2 = make_pipeline(
    StandardScaler(),
    SVR(kernel="poly", degree=2, C=100))
```



원편 그래프($C=0.01$)	오른편 그래프($C=100$)
규제 보다 강함 (낮은 자유도)	규제 보다 약함 (높은 자유도)
샘플에 덜 민감	샘플에 더 민감
마진 오류 보다 많이	마진 오류 보다 적게

SVM 회귀 모델 계산 복잡도

- LinearSVR
 - LinearSVC와 유사
 - 시간 복잡도가 훈련 세트의 크기에 비례해서 선형적으로 증가
- SVR
 - SVC와 유사
 - 훈련 세트가 커지면 매우 느려짐