기초 이론부터 실무 실습까지

머신 러닝 익히기

programmers

Part 05. SVM과 Decision Tree

정 정 민

programmers

Chapter 12. SVM

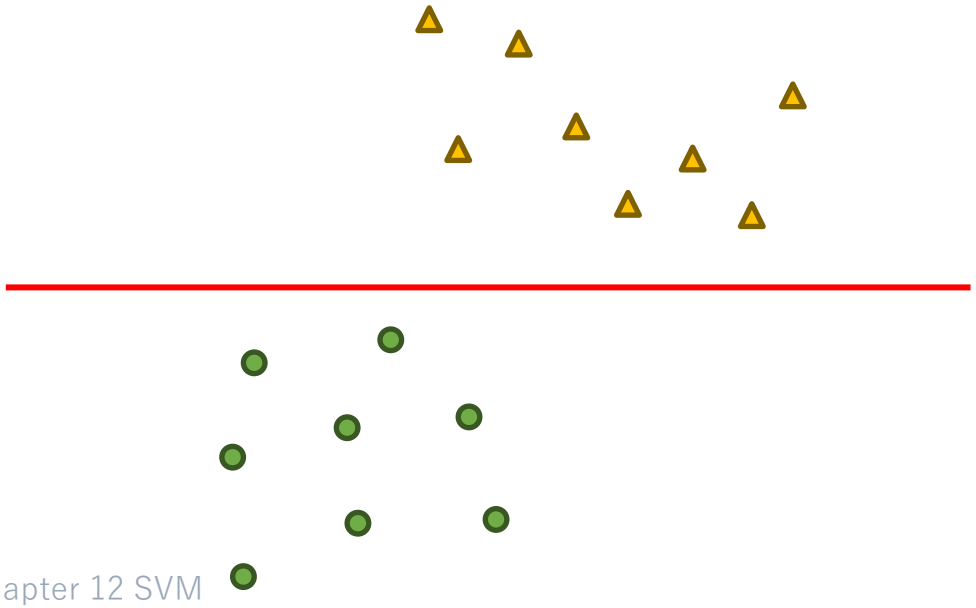
1. 선형 SVM
2. 비선형 SVM
3. SVR, Support Vector Regression

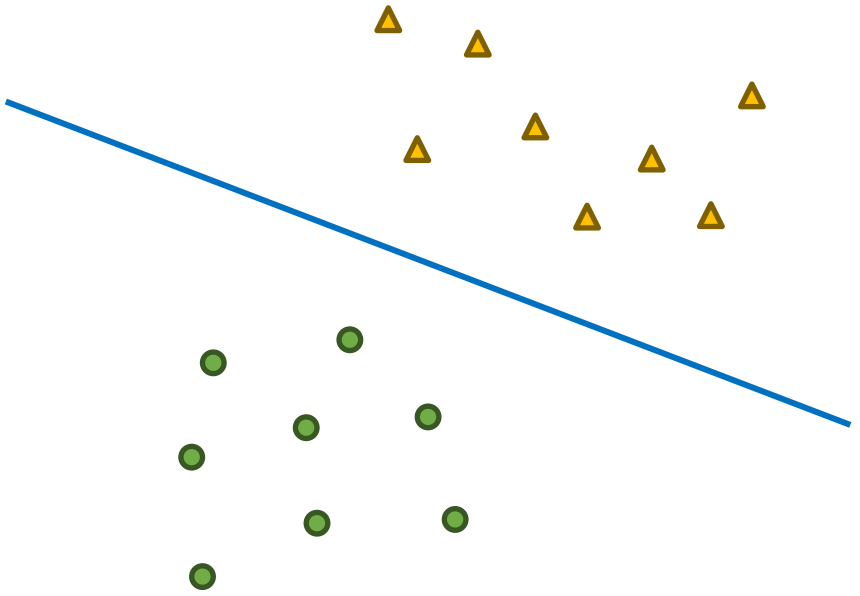
선형 SVM

강의 실습 코드 : 링크

삼각형과 원을 나누는 선!

- 아래 두 경우 중 원과 삼각형을 더 잘 나눈 경우는 어디일까요?
- 잘 나눈다는 정의 필요
  - 정확히 나누었는가? (빨강, 파랑)
  - 일반화가 잘 되었는가? (파랑)
- 왜 파란색이 더 일반화가 좋을까?



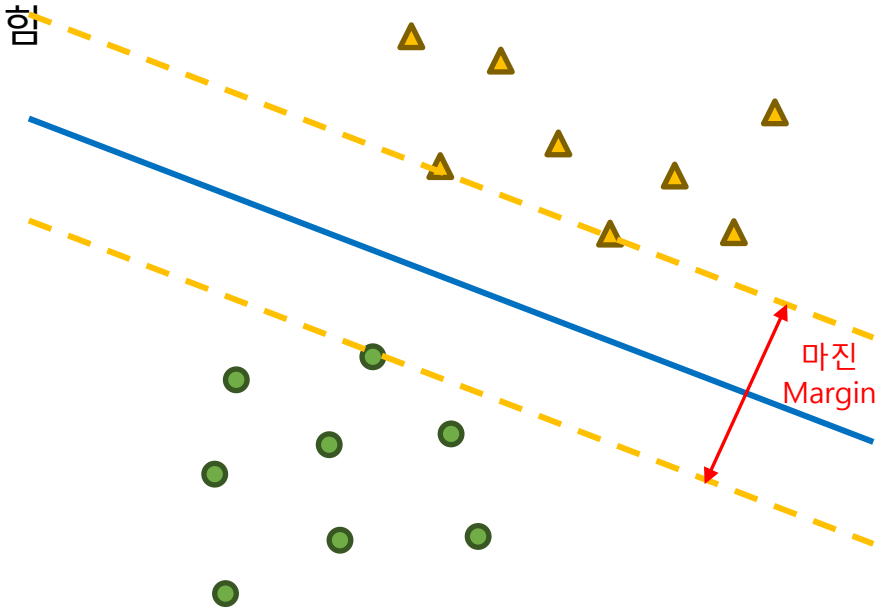


Margin 과 Support Vector

- 파란 분류 선의 경우
- 원과 삼각형을 잘 나누고 있을 뿐 아니라,
- 각 클래스의 데이터 샘플로부터 가장 멀리 위치해 있음
  - 그렇기 때문에 일반화 성능이 좋음

- 샘플로부터 분류 선까지의 거리를 마진(margin)이라고 함

- 이 샘플은 학습 데이터 중 일부에 해당하며
- 마진을 구성하는 이 데이터를 서포트 벡터(support vector)라고 함



선형 SVM의 목적

- SVM은 두 데이터를 나누는 직선(고차원의 경우 초평면) 을 찾고자 함

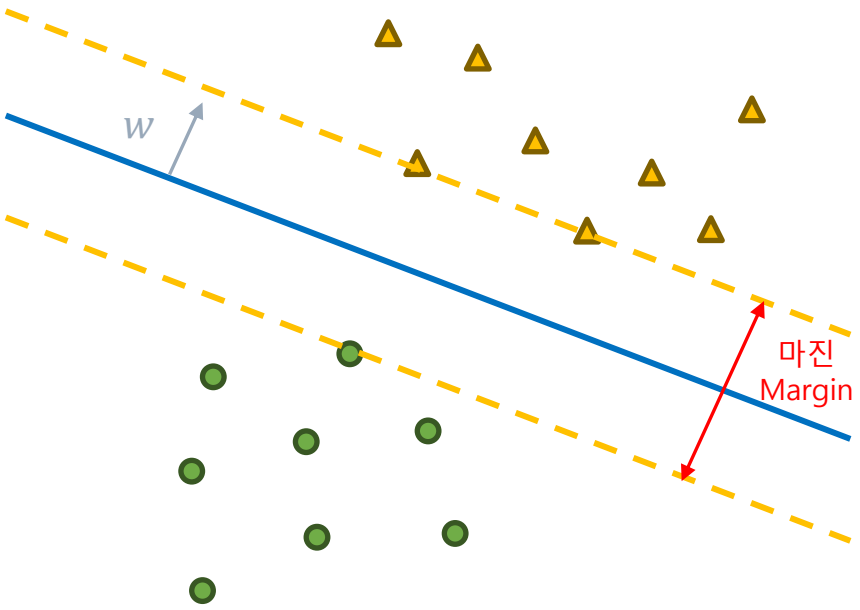
- 이때, 찾고자 하는 직선(혹은 초평면)과 평행한 두 개의 직선(초평면)을 만들 수 있고
- 이 두 직선 사이의 거리를 마진이라 함

- SVM의 경우 마진을 최대화 하는 최적 직선을 만드는 것이 목적
  - 이때의 직선을 '최대 마진 초평면'이라고 함

- 최대 마진 초평면의 선형 방정식 수식은 아래와 같음

$$w^T x + b = 0$$

- w와 b는 학습을 통해 찾아야 하는 값으로
  - w는 최대 마진 초평면의 법선 벡터
  - 그리고 b는 편향값에 해당



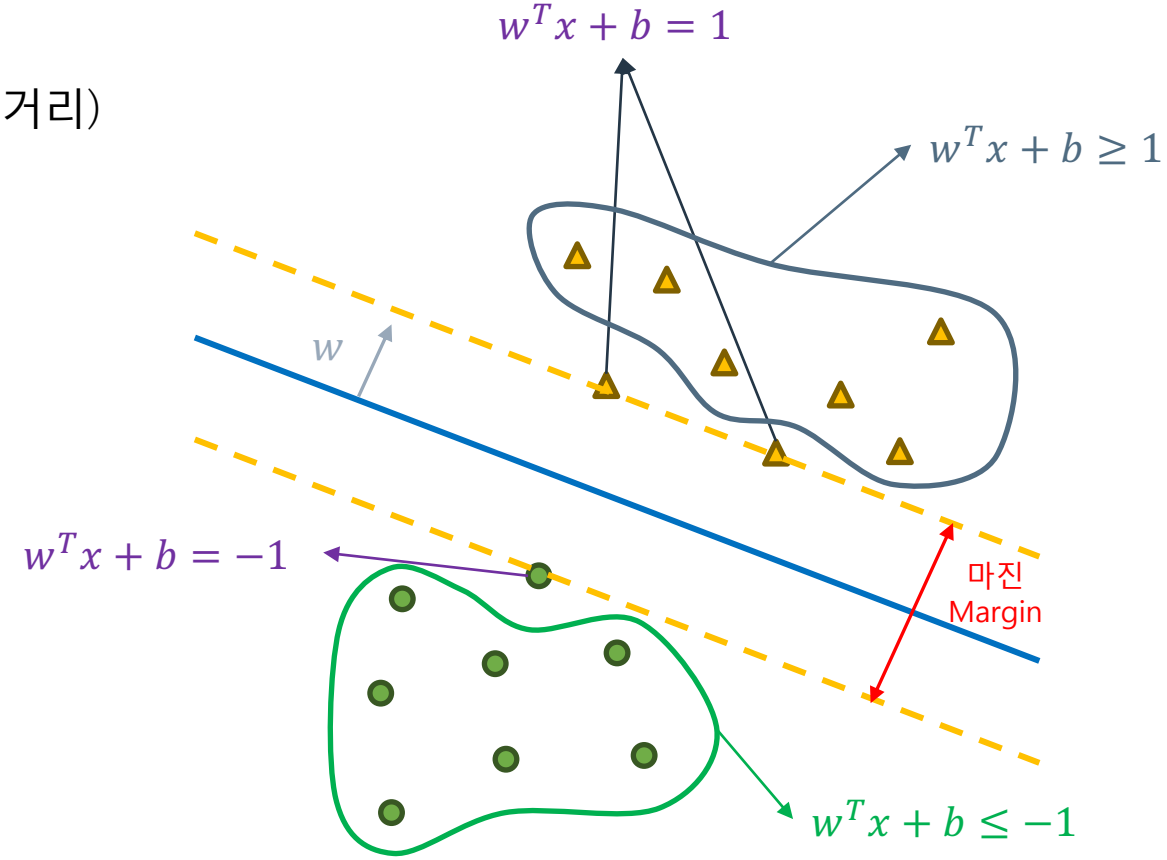
최적화 문제 1

- 평행한 두 직선위에 서포트 벡터(support vector)가 존재
- 서포트 벡터에 대해 $|w^T x + b| = 1$를 만족

- $w^T x + b \geq 1$인 영역에서는 $y = 1$을, $w^T x + b \leq -1$인 영역에서는 $y = -1$을 만족
- 이를 한번에 쓰면 $y_i(w^T x_i + b) \geq 1$

- 이때, 마진은 아래와 같이 표현 가능 (점과 직선 사이의 거리)

$$Margin = \frac{2}{\|w\|}$$



최적화 문제 2

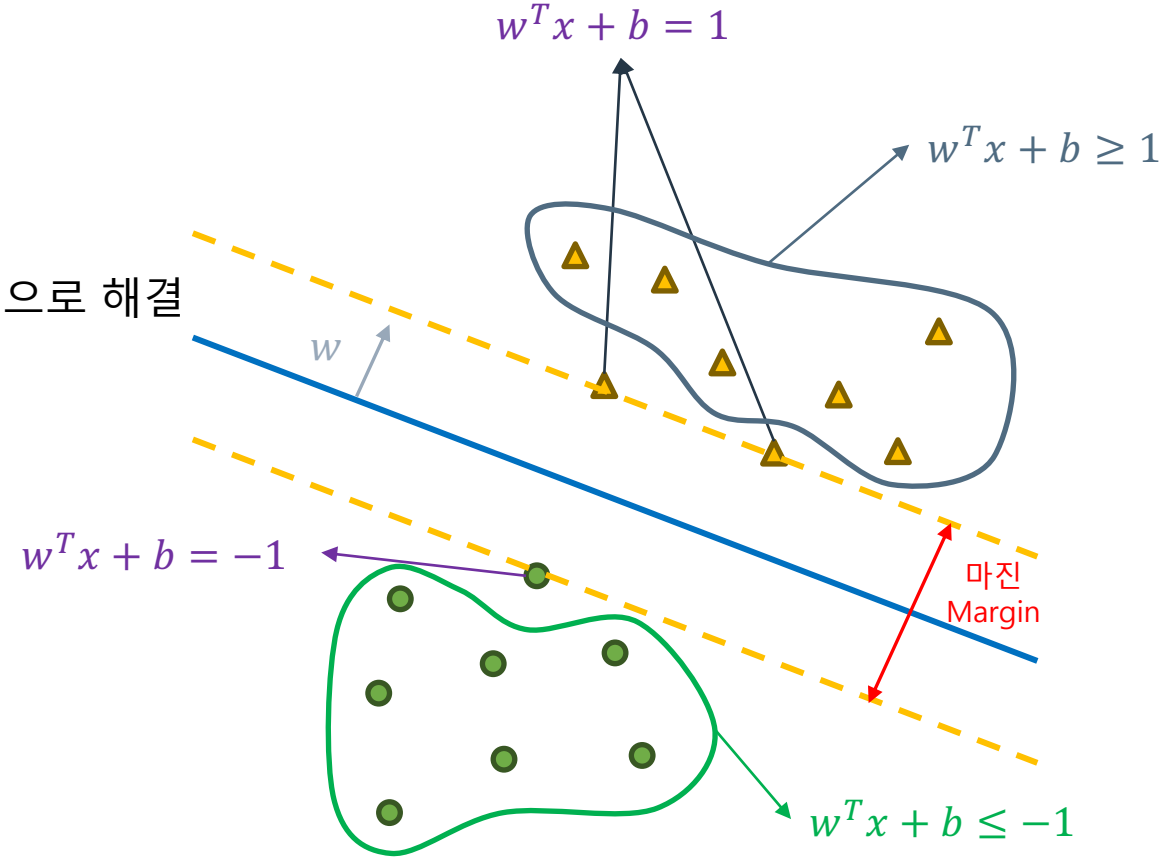
- 최적화 목표 : Margin을 키우면서 & 모든 데이터를 알맞게 분류

$$\max_{w,b} \frac{2}{\|w\|}$$
$$subject\ to\ y_i(w^T x_i + b) \geq 1\ for\ 1 \leq i \leq n$$

- 이를 계산이 용이하며 더욱 간단히 쓰면

$$\min_{w} \frac{1}{2}\|w\|^2$$

- 이를 풀기 위해서는 라그랑주 승수법 및 쌍대 문제 방식으로 해결
  - 이 과정은 수업의 범위를 넘어섬
  - 링크의 348 페이지 Appendix E를 통해 확인 가능

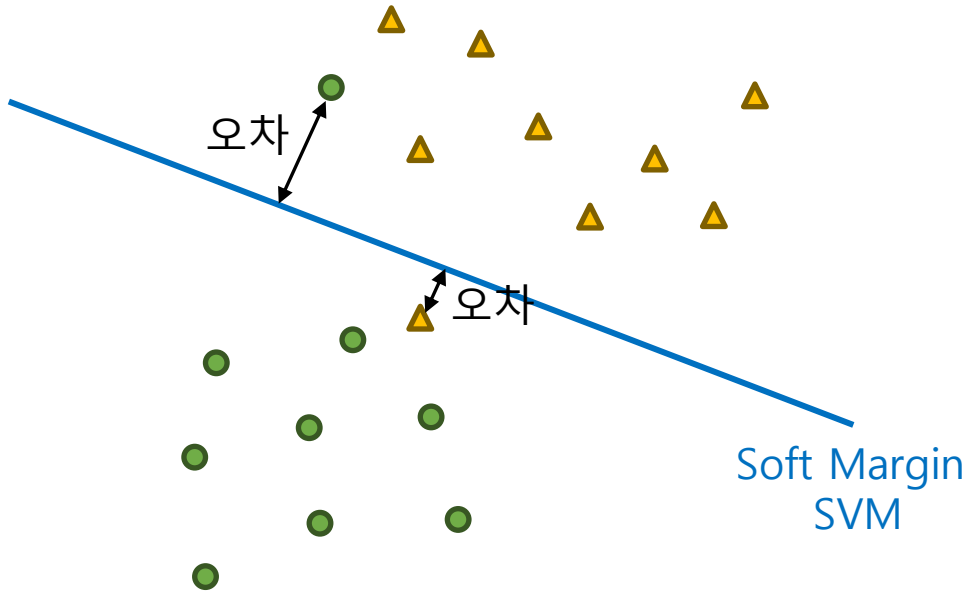


하드 마진 SVM과 소프트 마진 SVM

- 앞선 예시 데이터는 모든 데이터가 이쁘게 나뉘어 있음
- 따라서 어떠한 오분류도 허용하지 않고 완벽한 선형 모델로 분리가 가능
- 이를 하드 마진 SVM 이라고 함

- 하지만, 일반적으로는 어느 정도 데이터가 섞인 경우가 흔함
- 따라서 완벽한 선형 분리가 불가능한 경우가 ↑

- 어느 정도의 오분류를 허용하면서 오차 발생에 따른 패널티를 비용 함수에 부과
- 이를 통해 일반화 성능을 올릴 수 있음
- 이를 소프트 마진 SVM 이라고 함



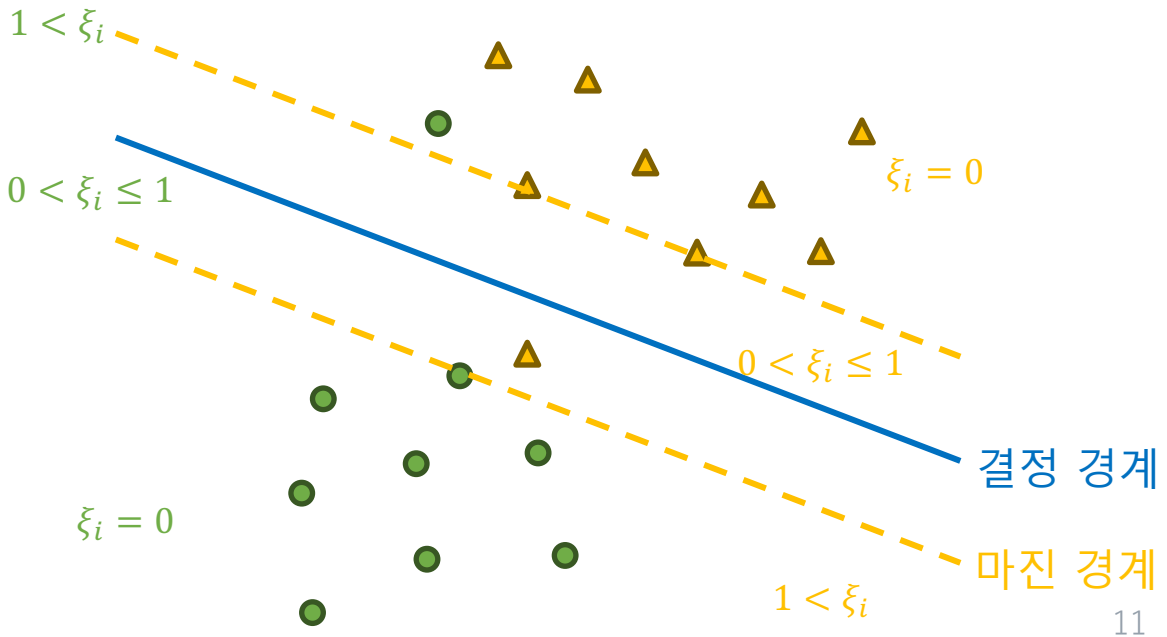
슬랙 변수 (slack variable)

- 소프트 마진 SVM에서 사용하는 개념으로
- 완벽하게 선형 분리되지 않는 데이터에 대해 SVM을 적용할 수 있도록 함

- 각 데이터 포인트(i)당 하나의 슬랙 변수 ξ_i가 할당되며
- 이 변수는 해당 데이터 포인트가 마진을 얼마나 위반하는지를 수치적으로 나타냄

- 마진을 위반하지 않은 데이터 : $\xi_i = 0$
  - 서포트 벡터와
  - 서포트 벡터보다 멀리 있는 데이터

- 마진을 위반한 데이터
  - 마진 경계 ~ 결정 경계 : $0 < \xi_i \leq 1$
  - 결정 경계 이후 : $1 < \xi_i$
    - 이 경우는 올바르지 못한 클래스로 분류 됨



소프트 마진 SVM의 최적화 함수

- 소프트 마진 SVM은 하드 마진 SVM 최적화 과정에 규제 페널티(ξ_i)를 도입해 일반화한 최적화 식을 사용

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$subject\ to\ y_i(w^T x_i + b) \geq 1 - \xi_i\ and\ \xi_i \geq 0\ for\ 1 \leq i \leq n$$

- 목적
  - 마진의 크기를 최대화
  - 마진 위반을 최소화

- C : 일반화를 위한 하이퍼파라메터로 마진 크기와 규제 사이의 중요도 변수
- $1 - \xi_i$: 규제가 적용될 데이터 포인트에 대해, 결정 경계에서 ξ_i 의 거리 만큼 벗어날 수 있음을 허용하는 과정
- $\xi_i \geq 0$: ξ_i이 음수를 갖을 수 없음을 조건으로 제시

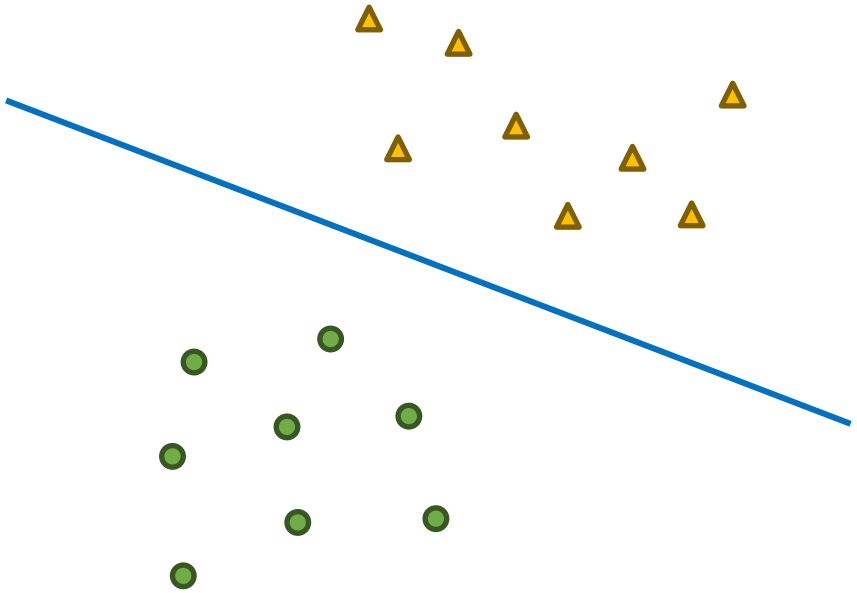
비선형 SVM

강의 실습 코드 : 링크

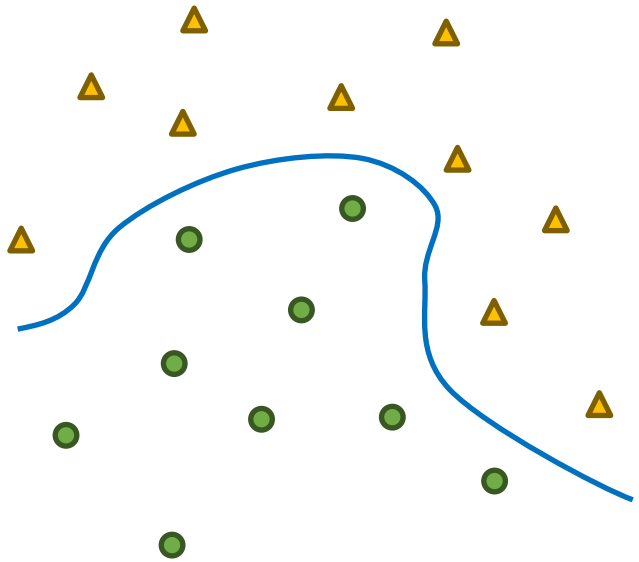
선형 SVM 모델의 한계

- 데이터의 복잡성으로 인해 선형 결정 경계로 데이터를 분류할 수 없는 경우가 있음
- 특히 데이터가 휘어진 형태로 분포한다면 선형 SVM 으로는 분류할 수 없음

- 비선형 SVM은 이러한 데이터에 대해 효과적으로 작동



선형으로 분리될 수 있는 데이터 분포



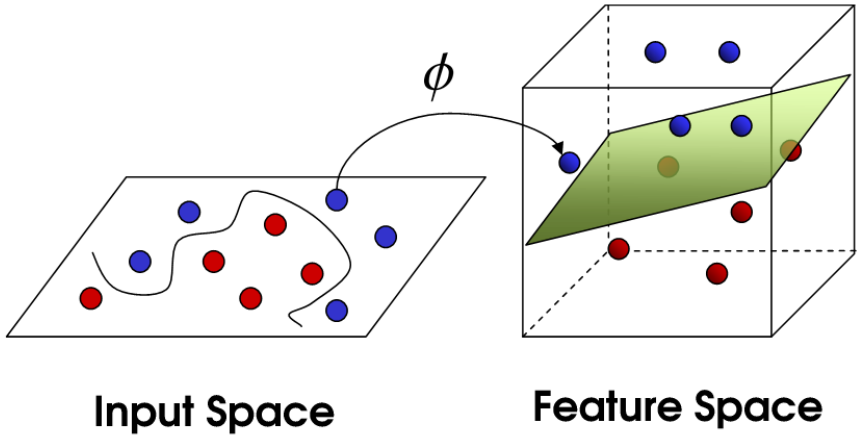
고차원에서의 데이터

- 선형적으로 분리할 수 없는 데이터를 고차원으로 변형하면
- 고차원의 초평면으로 분리할 수 있는 형태로 변환이 가능

- 왜 그런가요??
  - 차원이 증가하면 데이터 포인트 간의 상대 거리는 증가
  - 각 차원에서 데이터끼리 차지하는 공간이 확장
  - 그러면서 비슷한 특성을 공유하는 데이터들은 특정한 축 혹은 방향으로 군집될 가능성이 ↑

- 따라서, 고차원의 데이터일수록 선형으로 분류할 수 있는 가능성이 높음

- 저차원의 데이터를 고차원의 데이터로 옮기는 과정 함수를 정의할 수 있고
- 이른 Mapping function(ϕ) 이라고 부름



https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f

고차원 데이터가 갖는 문제

- 고차원의 데이터는 선형 분류할 수 있는 가능성이 있지만
- 계산량이 늘어난다는 단점이 있음

- 확장한 고차원이 원본 데이터의 차원보다 훨씬 크다면
- 모델의 복잡도가 늘어나고 효율성이 떨어짐

- 딜레마에 봉착!
  - 차원을 높여 선형 분류를 하고 싶은데
  - 계산량과 복잡성이 덩달아 늘어남

- 이를 해결하기 위해 커널 트릭이 제시됨
  - 높은 차원의 장점을 취하면서도
  - 계산의 복잡성이 증가되지 않는 기법

커널 트릭 (Kernel Trick)

- 데이터를 선형 분류하기 위해서는 데이터 포인트 사이의 내적 계산이 수행되어야 함
  - 내적은 두 임의의 vector 사이의 유사도를 측정 & 선형 경계를 생성하데 사용됨
  - 고차원에서 선형 분류를 해야하는 비선형 SVM 에서도 이 과정이 필요함
- 즉, 고차원에서 정상적으로 계산을 한다면 아래의 과정이 필요

$$\phi(x)^T\phi(x')$$

- 하지만 이는 계산량이 매우 많이 소모됨 (ϕ 의 결과로 고차원 vector이므로)

- 이때, 고차원의 내적 연산의 결과와 똑같은 결과를 보여주는 저차원 vector 끼리의 연산 함수가 있다면?

$$K(x, x') = \phi(x)^T\phi(x')$$

- 그렇다면 고차원으로 데이터를 변형하지 않고도,
  저차원의 데이터 만으로도 고차원 데이터를 활용한 내적 연산의 효과를 누릴 수 있음
  - 그래서 이름에 trick이라는 말이 들어갔네요!

다양한 커널의 종류

- 고차원의 내적 연산과 똑같은 결과를 보여주는 함수를 커널(kernel)이라고 하며
- 일반적으로 비선형 SVM에서 많이 사용하는 커널 함수는 아래와 같음

- **다항 커널 (Polynomial Kernel)**
  - 다양한 차수 설정으로 여러 식을 근사할 수 있지만 과적합의 위험이 있음

$$K(x, x') = (\gamma x_1^T x_2 + r)^d$$

- **RBF 커널 (Radial Basis Function Kernel) 혹은 가우시안 커널 (Gaussian Kernel)**
  - 다양한 데이터에 적용하면서도 유연성이 높아 범용성이 높음

$$K(x, x') = e^{-\gamma \|x_1 - x_2\|^2}$$

- **시그모이드 커널 (Sigmoid Kernel)**
  - 이진 분류에 최적화되며 RBF 커널에 비해 성능이 떨어짐

$$K(x, x') = \tanh(\gamma x_1^T x_2 + r)$$

비선형 SVM의 최적화 문제

- 소프트 마진 SVM과 비슷한 구조를 갖고 있음
- 다만 원래 차원(저차원)이 아니라 고차원에서의 데이터 분류가 가능하도록 하는 조건이 들어감

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$subject\ to\ y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i\ and\ \xi_i \geq 0\ for\ 1 \leq i \leq n$$

- 목적
  - 마진의 크기를 최대화
  - 마진 위반을 최소화

- $y_i(w^T\phi(x_i) + b)$: 고차원 데이터에서 선형 분류가 가능함을 표시

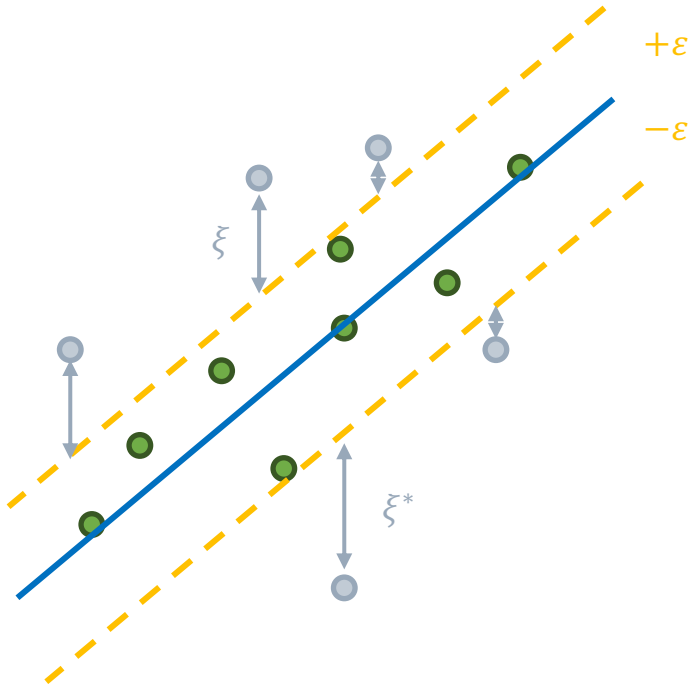
SVR, Support Vector Regression

강의 실습 코드 : 링크

회귀 문제에 적용되는 SVR

- 회귀 문제로 확장한 SVM 방법을 SVR (Support Vector Regression)이라고 함

- 주어진 데이터에서 가능한 많은 데이터 포인트를 포함하는 마진 구역을 설정
  - 이 마진 구역은 사용자가 선언한 허용 오차(ε) 내부의 구역
- 그 마진 구역 안에서 회귀선(혹은 초평면)을 찾는 것을 목표로 함



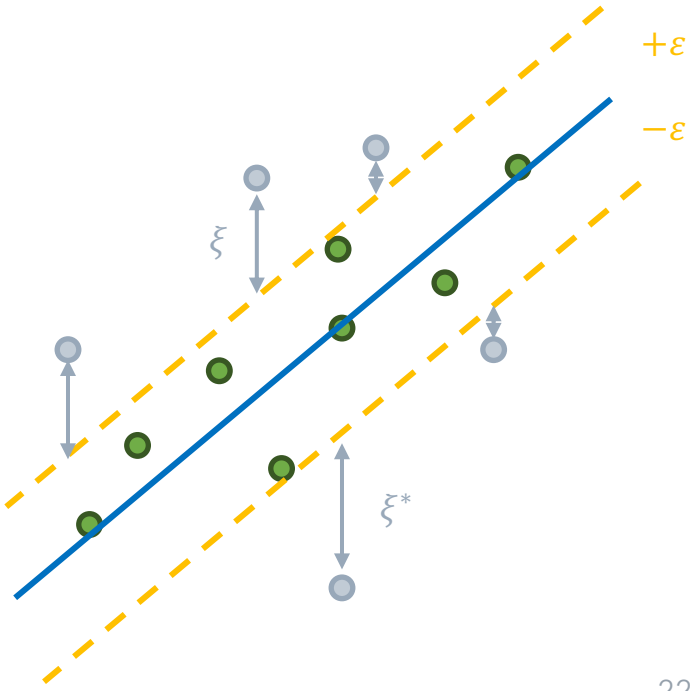
SVR의 최적화 문제

- 커널 함수를 적용한 SVR의 최적화 함수는 아래와 같음

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

$$subject\ to,$$
$$y_i - (w^T\phi(x_i) + b) \le \varepsilon + \xi_i\ and$$
$$(w^T\phi(x_i) + b) - y_i \le \varepsilon + \xi_i^*\ and$$
$$\xi_i, \xi_i^* \ge 0\ for\ 1 \le i \le n$$

- 정답과 예측값 사이의 차이가 ε 이라는 변수 안에서 허용될 수 있음을 내포
- 양쪽 방향의 슬랙 변수 (ξ_i, ξ_i^*)를 도입해 많은 데이터 포인트를 포괄하도록 강제



E.O.D