

Linear Regression & SVM

23.01.31 / 8기 백민준

CONTENTS

01. Introduction

- 지도학습
- 회귀와 분류

02. Linear Regression

- Linear Regression
- 회귀계수 추정방법
- 회귀계수의 의미
- 변수 수 증가에 따른 문제

03. Regularization

- Ridge
- Lasso

04. Logistic Regression

- Logistic Regression
- Binary Cross Entropy

05. SVM

- SVM
- SVM의 수리적 모델링
- 비선형 SVM
- 커널 함수

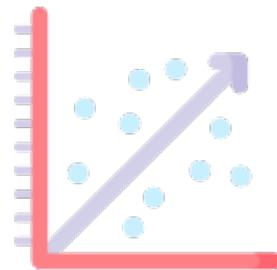
06. SUMMARY

- Summary
- Reference

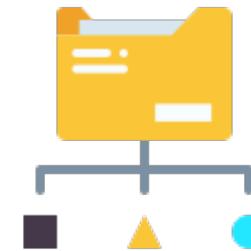
지도 학습(Supervised Learning)

주어진 입력과 출력 데이터가 있고, 출력 데이터로 모델을 학습한 후,
입력으로부터 출력을 예측하고자 할 때 사용되는 머신러닝 방법

→ 레이블(Y, 정답)이 존재하는 학습 방법, 모델에게 정답을 가르쳐주는 학습 방법



회귀(Regression)



분류(Classification)

회귀(Regression)

변수들 간의 함수적 관계를 탐색

연속적 수치, 연속형 자료를 예측



Ex) 몸무게로 키 예측

분류(Classification)

이미 정해진 몇 개의 class label 중 하나를 예측
(사전에 어떤 카테고리로 나누어질지 결정되어 있음)

이산적 수치, 범주형 자료를 예측

이진(Binary) 분류와 다중(Multiclass) 분류



Ex) 이메일의 스팸 여부

지도학습(Supervised Learning)



데이터를 받으면, 우리가 해야 하는 task가 무엇인지를 명확히 해야 한다.
이후에 어떤 방법론을 사용할 것인지 결정한다.

02. Linear Regression

회귀분석 (Regression)

회귀분석 (Regression) : 변수들 사이의 함수적 관계를 탐색하는 과정

독립 / 설명변수 (X) : 다른 변수에 영향을 주는 변수

종속 / 반응변수 (Y) : 다른 변수로부터 영향을 받는 변수

Y: Supervised & 연속형 자료 → Regression

선형회귀분석 (Linear Regression) : 독립변수와 종속변수 간 선형 상관관계를 모델링하는 기법

→ 데이터들에 가장 가까운 선형식($\hat{y} = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$)을 찾는다

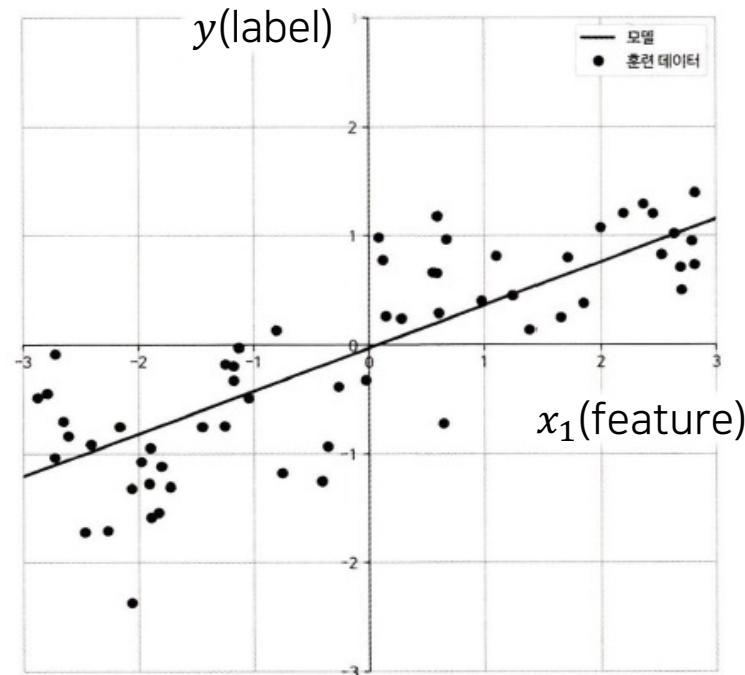
(**선형회귀분석 ⊂ 회귀분석**)

단순선형회귀분석 (Simple Linear Regression) : 독립변수 1개로 종속변수를 예측 (선형 관계)

다중선형회귀분석 (Multiple Linear Regression) : 독립변수 k개로 종속변수를 예측 (선형 관계)

02. Linear Regression

단순선형회귀(Simple Linear Regression)



단순선형회귀(Simple Linear Regression) : 독립변수 1개, 선형관계

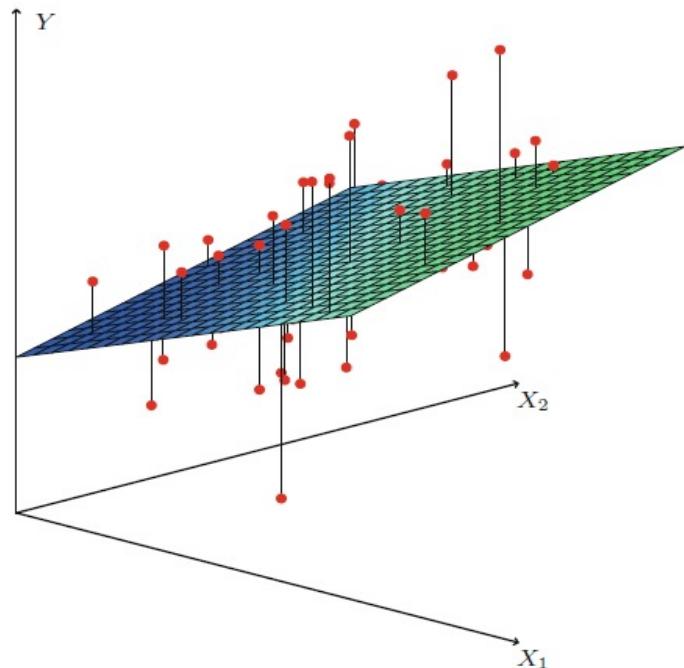
$$\hat{y} = \beta_0 + \beta_1 x_1$$

입력데이터가 1개인 경우,
 y 절편이 β_0 , 기울기가 β_1 인 1차 함수로
출력 데이터를 표현할 수 있다.

회귀계수(**학습 파라미터**): β_0, β_1

02. Linear Regression

다중선형회귀(Multiple Linear Regression)



다중선형회귀(Multiple Linear Regression) : 독립변수 k개, 선형관계

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

독립변수가 여러 개인 경우,
위와 같이 절편과 기울기로 이루어진 초평면(hyperplane)의 식을
데이터들을 설명하는 모델로 설정할 수 있다.

회귀계수(**학습 파라미터**): $\beta_0, \beta_1, \beta_2, \dots, \beta_k$

Ex) 특성이 2개인 경우, 회귀모델은
3차원 공간에 있는 데이터를 설명하는 평면이 된다.

회귀계수 β 를 추정하는 방법

1. 최소자승법 (Ordinary Least Squares Method, OLS)

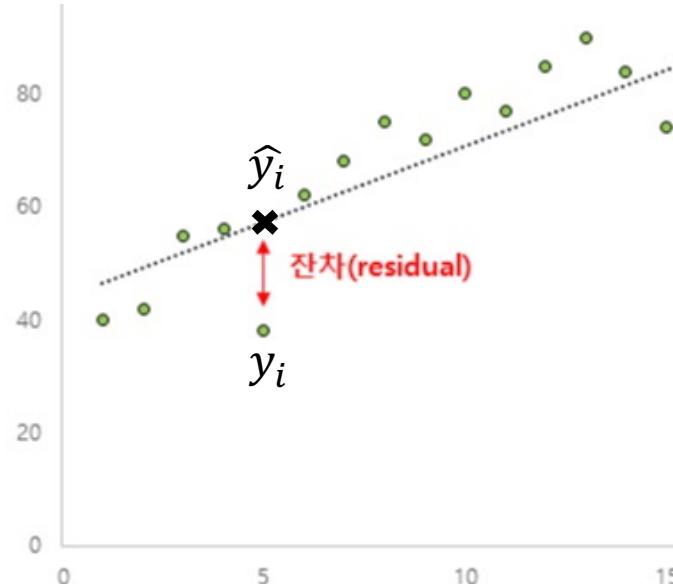
- (1) 정규방정식을 이용
- (2) 특잇값 분해로 유사역행렬을 구하여 정규방정식을 변형
- (3) 경사 하강법 (Gradient Descent, GD)

2. 최대우도법 (Maximum Likelihood Estimation, MLE)

02. Linear Regression

최소자승법 (OLS)

최소자승법(OLS): 잔차제곱합(SSE)을 최소화하는 회귀계수를 추정하는 방법
→ 데이터와 회귀식 간 차이를 가장 작게 만드는 회귀계수를 찾는다



잔차(residual): 실제 출력변수와 예측한 출력변수의 차이

$$e_i = y_i - \hat{y}_i$$

잔차제곱합(Sum of Squared Errors, SSE):

$$SSE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

평균제곱오차(Mean of Squared Errors, MSE):

$$MSE = \frac{1}{n - k - 1} \sum_{i=1}^n e_i^2 = \frac{1}{n - k - 1} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

02. Linear Regression

최소자승법 (OLS): (1) 정규방정식 이용

$$\min_{\beta} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$



$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (\text{정규방정식})$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{k1} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{kn} \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}$$

증명) $SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \cdots - \hat{\beta}_k x_{ik})^2 = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$

공식 이용) $\sum_{i=1}^n a_i = a_1 + a_2 + \cdots + a_n = [a_1 \dots a_n] \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}}_{\alpha^T \alpha} = \alpha^T \alpha \quad (\text{where } \alpha = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix})$

$$= (\mathbf{y}^T - (\mathbf{X}\beta)^T)(\mathbf{y} - \mathbf{X}\beta) = \mathbf{y}^T \mathbf{y} \boxed{- \mathbf{y}^T (\mathbf{X}\beta)} - \boxed{(\mathbf{X}\beta)^T \mathbf{y}} + \boxed{(\mathbf{X}\beta)^T (\mathbf{X}\beta)}$$

↳ $-2(\mathbf{X}\beta)^T \mathbf{y}$ 가 된다 ↳ $\beta^T \mathbf{X}^T \mathbf{X}\beta$

$\mathbf{y}^T (\mathbf{X}\beta) = (\mathbf{X}\beta)^T \mathbf{y}$ 이기 때문

$$= \mathbf{y}^T \mathbf{y} - 2\beta^T \mathbf{X}^T \mathbf{y} + \beta^T \mathbf{X}^T \mathbf{X}\beta$$

$$\frac{\partial}{\partial \beta} (SSE) = -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X}\beta = 0 \rightarrow (\mathbf{X}^T \mathbf{X})\beta = \mathbf{X}^T \mathbf{y}$$

if $(\mathbf{X}^T \mathbf{X})$ is invertible, $\beta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$



02. Linear Regression

최소자승법 (OLS): (2) 특잇값 분해로 유사역행렬을 구하여 정규방정식을 변형

$\hat{\beta} = (X^T X)^{-1} X^T y$ (정규방정식)의 단점:

1. 모델이 복잡해질 수록 연산 시간이 크게 증가
2. $(X^T X)$ 의 역행렬을 구할 수 없는 때 사용할 수 없다.



특잇값 분해를 이용해서 유사역행렬을 구하고,
이를 이용해 기존의 정규방정식을 변형해보자!

$$X^+ = V \Sigma^+ U^T \quad (\text{유사역행렬})$$

$$\hat{\beta} = (X^T X)^{-1} X^T y \rightarrow \hat{\beta} = X^+ y$$

역행렬과 달리, 유사역행렬은 항상 구할 수 있다.

정규방정식을 이용한 방법보다 계산 복잡도가 낮다

scikit-learn 패키지에서 사용된다

참고) 유사역행렬을 구하는 방법:

$$X = U \Sigma V^T \quad (\text{특잇값 분해 공식})$$

유사역행렬은 $X^+ = V \Sigma^+ U^T$ 로 계산된다.

Σ^+ 를 계산하기 위해 알고리즘이 먼저 Σ 를 구한다(특잇값 분해 이용).

그 다음 어떤 낮은 임곗값보다 작은 모든 수를 0으로 바꾼다.

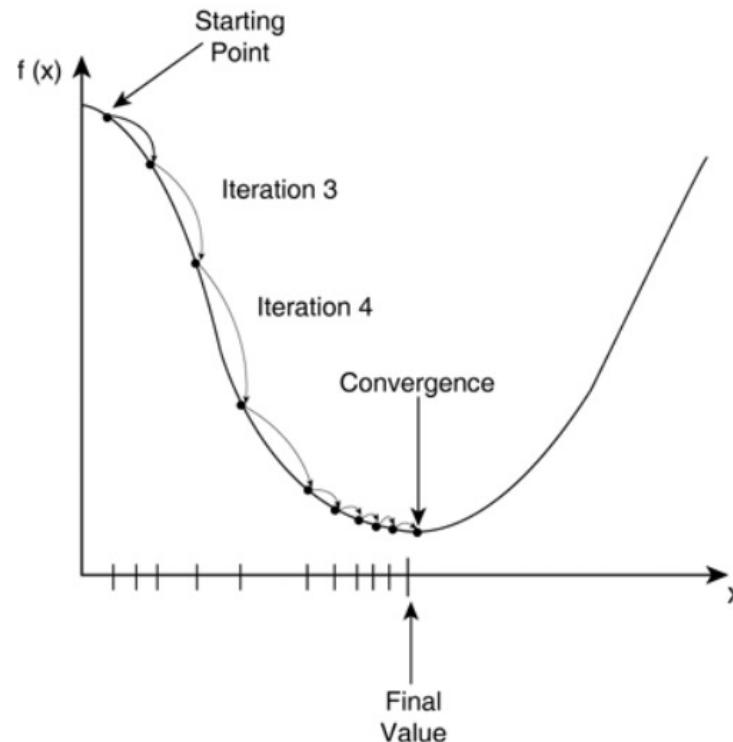
그 다음 0이 아닌 모든 값을 역수로 치환한다.

마지막으로 만들어진 행렬을 전치한다.

(핸즈온 머신러닝, 163p)

02. Linear Regression

최소자승법 (OLS): (3) 경사하강법



손실함수를 최소화시키기 위해 매개 변수를 반복적으로 조정하는 과정

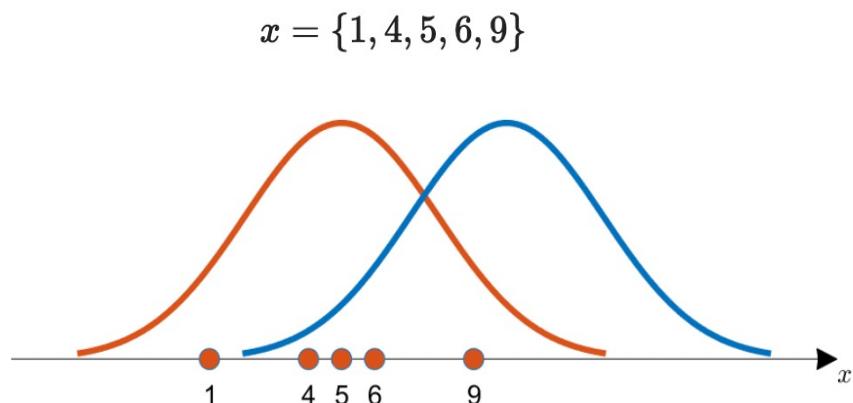
손실함수로 MSE 값을 사용한다.

→ 머신러닝에서의 선형회귀 목적함수가 딥러닝에서의 회귀 손실함수와 같다.
딥러닝에서 다루는 경사하강법을 머신러닝에서도 적용할 수 있다.

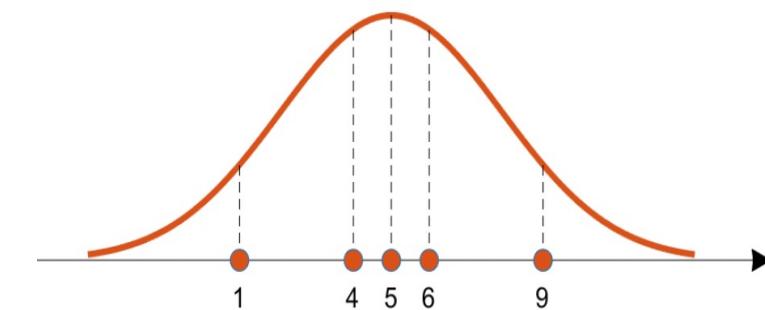
반복적인 학습을 통해 모델의 최적 파라미터를 찾는 것이 목표

02. Linear Regression

최대우도법 (Maximum Likelihood Estimation, MLE)



데이터 x 는 주황색 곡선과 파란색 곡선 중 어떤 곡선으로부터 추출되었을 확률이 더 높을까?



우도(Likelihood): 데이터가 어떤 분포로부터 만들어졌을 가능성

수치적으로 이 가능도를 계산하기 위해서는 각 데이터 샘플에서 후보 분포에 대한 높이를 계산해서 다 곱한다.

$$P(x|\theta) = \prod_{k=1}^n P(x_k|\theta)$$

02. Linear Regression

최대우도법 (Maximum Likelihood Estimation, MLE)

우도함수(Likelihood function)

$$P(x|\theta) = \prod_{k=1}^n P(x_k|\theta)$$

로그 우도함수 (Log-likelihood function)

$$L(\theta|x) = \log P(x|\theta) = \sum_{i=1}^n \log P(x_i|\theta)$$

위 식을 최대화시키는 θ 값을 찾는다:

$$\frac{\partial}{\partial \theta} L(\theta|x) = \frac{\partial}{\partial \theta} \log P(x|\theta) = \sum_{i=1}^n \frac{\partial}{\partial \theta} \log P(x_i|\theta) = 0$$

최대우도법(MLE): **관측치가 주어진 상황에서
우도(likelihood)를 최대화시키는 분포의 모수를 찾는 방법**

회귀계수 β 를 추정하는 방법 (정리)

1. 최소자승법 (Ordinary Least Squares Method, OLS)

(1) 정규방정식을 이용: $\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

(2) 특잇값 분해로 유사역행렬을 구하여 정규방정식을 변형: $\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \rightarrow \hat{\beta} = \mathbf{X}^+ \mathbf{y}$

(3) 경사 하강법 (Gradient Descent, GD):

손실함수를 최소화시키기 위해 매개 변수를 반복적으로 조정하는 과정

2. 최대우도법 (Maximum Likelihood Estimation, MLE):

관측치가 주어진 상황에서 우도(likelihood)를 최대화시키는 분포의 모수를 찾는 방법

02. Linear Regression



회귀계수(기울기)의 의미

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

회귀계수 $\beta_1, \beta_2 \dots \beta_k$ 는 (β_0 제외) X가 한 단위 증가했을 때 Y의 평균적인 변화량을 의미한다.

(이때 다른 입력변수는 고정되어 있다고 가정)

따라서 회귀계수는 변수의 중요도로 간주할 수 있고, 회귀식을 통해 주요 인자를 판단할 수 있다.

Ex) 민준의 몸무게를 예측하고자 한다.

X_1 : 한 달 동안 야식을 먹은 횟수 (단위: 회)

X_2 : 한 달 동안 운동한 시간 (단위: hour)

Y: 그 달의 민준의 몸무게 (단위: kg)

$$\rightarrow Y = 1.5X_1 - 3X_2 + 66$$

해석) 민준이가 야식을 먹는 횟수가 1회 더 증가할 때마다 민준의 몸무게는 평균적으로 1.5kg씩 증가한다.

민준이가 운동을 하는 시간이 1시간 더 증가할 때마다 민준의 몸무게는 평균적으로 3kg씩 감소한다.

운동을 한 시간 하는 것이 야식을 한 번 안 먹는 것 보다 그 효과가 평균적으로 2배 더 크다.

02. Linear Regression

스케일링의 중요성

회귀분석은 스케일링에 매우 민감한 분석방법이다.

입력변수들의 범위 / 분포가 크게 다른 경우 스케일링을 하는 것이 바람직하다.

Ex) 지역별 소비자 물가지수를 예측하고자 한다.

Y: 지역별 소비자 물가지수

X_1 : 해당 지역의 30층 이상 빌딩의 평균 가격 (단위: 원)

X_2 : 해당 지역의 김밥 한 줄 평균 가격 (단위: 원)

→ 스케일링 미적용시 회귀계수 해석이 의미가 없을 수도 있다.

1. Minmax scaling은 동일한 변화율에 따른 차이를 비교하고자 할 때 용이

Ex) 빌딩과 김밥의 가격이 동일하게 10% 증가했을 때 Y 변화량의 차이를 비교하고자 할 때

!!

2. Standard scaling은 변수들이 분포에서 상대적으로 얼마나 변화했는지 알고자 할 때 용이

Ex) 빌딩과 김밥의 가격 분포에서, Z-score가 각각 1만큼 증가했을 때 Y의 변화량을 알고자 할 때

다중선형회귀: 변수 수의 증가에 따른 문제

다중선형회귀는 출력변수와 여러 입력 변수들 간의 복잡한 관계를 모델링하기 때문에 단순선형회귀보다 높은 예측력을 보이는 모델을 만들 수 있다.

그러나 변수가 너무 많아지면 **과대적합** 또는 **다중공선성** 문제를 야기할 수 있다.

과대적합은 출력변수와 상관성이 없는 입력변수 때문에 선형회귀 식이 복잡해지고 이로 인해 학습 데이터에 과대적합되어 테스트 데이터에 대한 예측력이 감소하는 것을 의미한다.

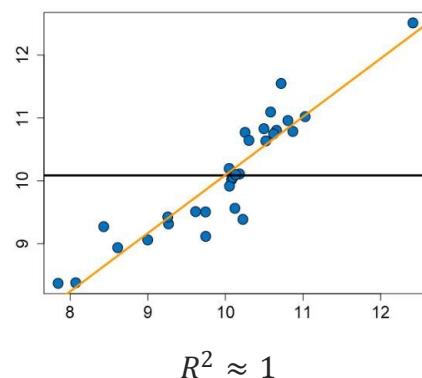
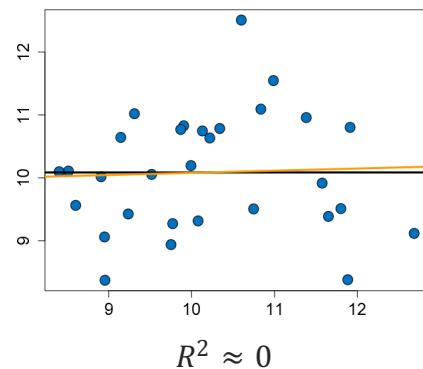
다중공선성은 입력변수들의 상관성이 높은 것을 의미하며, 학습 데이터에 따라 추정되는 회귀계수의 변동성이 심해진다는 문제가 발생한다.

02. Linear Regression

선형회귀 평가지표: 결정계수 R^2

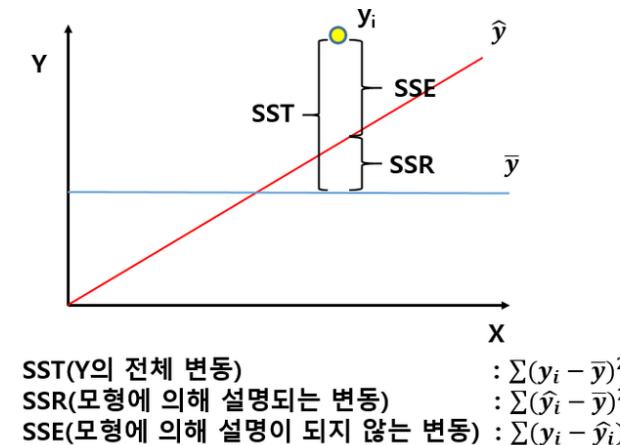
회귀 모델에서 독립변수가 종속변수를 얼마나 설명해주는지를 가리키는 지표로, 0 ~ 1 값을 가진다.

R^2 은 선형회귀 예측값 \hat{Y} 가 \bar{Y} 대비 얼마나 실제값 Y 를 잘 설명하는지를 의미함. R^2 이 1에 가까울 수록 선형회귀 모형의 설명력이 높다는 것을 뜻함 (0에 가까우면 평균 \bar{Y} 로 예측한 성능과 같다)



$$R^2 = \frac{SSR}{SST}$$

설명되는 변동 / 전체 변동
-> 그러니까 클수록 좋음!
설명되는 부분이 많은거니까



02. Linear Regression

과소적합(Underfitting) vs 과대적합(Overfitting)

과소적합(Underfitting): 모델이 너무 단순하거나 잘 학습이 되지 않아 데이터에 대해 설명을 잘 못하는 경우

과대적합(Overfitting): 모델이 훈련 데이터에만 과도하게 적합되어 일반성이 떨어지는 경우

→ 잘 일반화되어 **강건(Robust)하게** 작동하는 모델이 좋은 모델



02. Linear Regression

과대적합(Overfitting) 판단지표: R^2_{adj} 와 R^2_{pred}

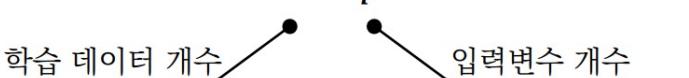
변수 수가 증가하면 자연스럽게 R^2 이 증가한다.

 변수 수가 증가할수록 Penalty 를 줌

다중선형회귀분석에서는 R^2 보다는 R^2_{adj} 와 R^2_{pred} 지표가 모형의 성능을 더욱 정확하게 평가한다.

Adjusted R^2 는 모형에 사용된 입력 변수 수만큼 Penalty를 주는 지표로써 식은 다음과 같다

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1} = 1 - \frac{SSE(n - 1)}{SST(n - p - 1)}$$



Predicted R^2 는 학습 데이터로 만든 회귀 모형을 검증 데이터 $\{(x_i, y_i), i = 1 \dots k\}$ 로 예측 성능을 평가한 지표로 다음 식과 같다

$$R^2_{pred} = 1 - \frac{\sum_{i=1}^k (y_i - \hat{y}_i)^2}{\sum_{i=1}^k (y_i - \bar{y})^2}$$

→ R^2 에 비해 R^2_{adj} 와 R^2_{pred} 가 현저히 낮다면 과대적합 존재

다중공선성(Multicollinearity)

다중공선성은 입력변수 간 상관계수가 높은 것을 의미함

Ex) X_1, X_2, X_3 로 Y 를 예측하는데, $X_3 = 2*X_1$ 인 경우

입력변수들의 상관성이 높으면 학습 데이터에 따라 추정되는 회귀계수의 변동성이 심함 (회귀계수의 불안정성 문제 발생).

따라서 회귀계수는 더 이상 출력변수에 대한 상대적인 설명력으로 해석하기 어려움

또한 회귀계수 β_i 는 X_i 를 제외한 다른 입력변수는 고정되어 있고, X_i 만 한 단위 증가했을 때 출력변수의 변화량을 의미하였는데, 상관관계가 존재하면 다른 입력변수가 고정된다는 가정이 맞지 않아 해석이 유효하지 않음

다중공선성(Multicollinearity) 판단 지표: 상관계수, VIF

입력변수 간 **상관계수 행렬을** 구하여 다중공선성을 파악할 수 있다.

가장 많이 사용하는 판단 지표는 **VIF(Variation Index Factor)**이다. VIF는 다중공선성을 가지는 변수를 발견하기 위해, k 번째 변수에 대해 다음과 같은 모델을 수립하고 모델의 설명력 R^2 으로 VIF_k 를 계산한다:

$$X_k = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{k-1} X_{k-1} + \beta_{k+1} X_{k+1} + \cdots + \beta_p X_p$$

$$VIF_k = \frac{1}{1 - R_k^2}$$

일반적으로 VIF가 10 이상인 경우 입력변수가 다중공선성 문제를 일으킨다고 한다

규제, 정규화 (Regularization)

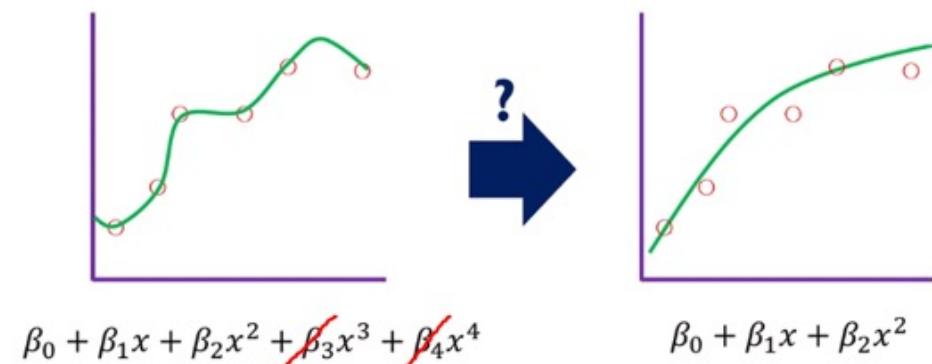
일반적으로 과대적합과 다중공선성은 불필요한 또는 중복되는 입력 변수가 있을 때 발생
→ 불필요한 입력변수들을 없애보자!

규제, 정규화 (Regularization) :

회귀계수에 관한 규제항을 추가함으로써 영향력이 없는 입력변수의 효과를 제거하는 기법

회귀계수를 축소하는 규제항 $f(\hat{\beta})$ 을 추가:

$$\min_{\beta} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + f(\hat{\beta})$$



규제가 있는 선형회귀 (Regularized Linear Regression): Ridge, Lasso

!!

Ridge 회귀에서는 **회귀계수의 제곱합을** $f(\hat{\beta})$ 에 대입 : L2 규제

$$\text{Minimize} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \hat{\beta}_j^2$$

Lasso 회귀에서는 **회귀계수의 절대값 합**을 $f(\hat{\beta})$ 에 대입 : L1 규제

$$\text{Minimize} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p |\hat{\beta}_j|$$

λ 는 하이퍼 파라미터로 크면 클수록 보다 많은 회귀계수를 0으로 (또는 0으로 가깝게) 만듦

03. Regularization

Ridge & Lasso의 최적화 표현

Ridge와 Lasso는 다음과 같은 최적화 방식으로도 표현 가능하다 (상단: Ridge, 하단: Lasso)

$$\text{Minimize} \left\{ \sum_{i=1}^n \left(y_i - \hat{\beta}_0 - \sum_{j=1}^p \hat{\beta}_j x_{ij} \right)^2 \right\} \text{Subject to } \sum_{j=1}^p \hat{\beta}_j^2 \leq s$$

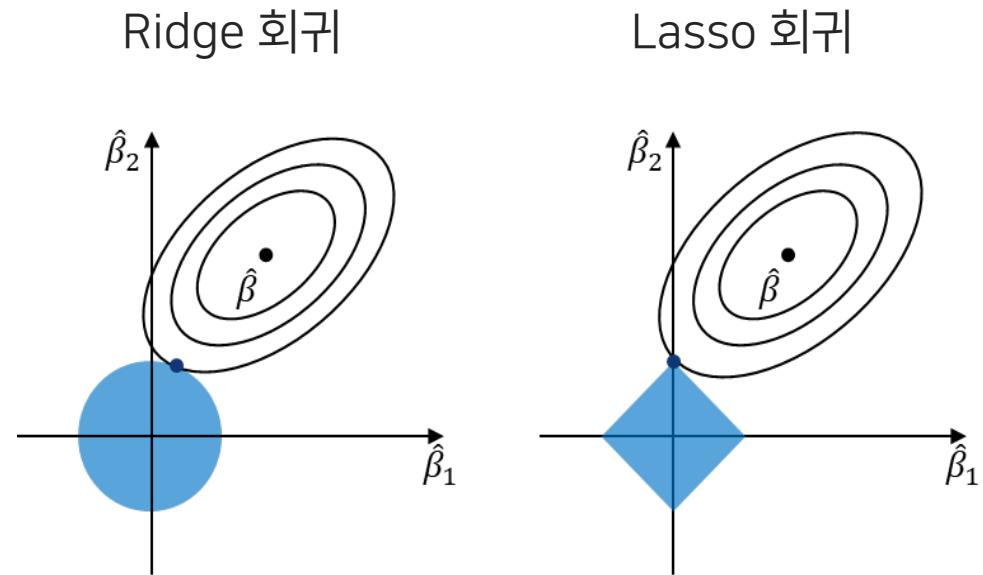
$$\text{Minimize} \left\{ \sum_{i=1}^n \left(y_i - \hat{\beta}_0 - \sum_{j=1}^p \hat{\beta}_j x_{ij} \right)^2 \right\} \text{Subject to } \sum_{j=1}^p |\hat{\beta}_j| \leq s$$

Ex) Ridge 회귀: X1, X2로 Y를 예측, $s = 10$
 $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

→ SSE를 최소화하는 방향으로 하되,
 β_1 과 β_2 의 제곱합이 10보다 작게!
 $\beta_1 = 3, \beta_2 = -4$ (X)
 $\beta_1 = 1, \beta_2 = 2$ (O)

03. Regularization

Ridge & Lasso



하늘색 영역: 회귀계수가 가질 수 있는 영역
검은색 타원: SSE가 같은 지점을 연결한 그림
(가운데로 갈수록 오차가 작아짐)
Ridge와 Lasso 모두 SSE를 희생하여 계수를 축소하는 방법
Lasso의 경우 회귀계수가 0이 될 수 있지만, Ridge는 불가능

Ridge는 회귀계수를 0에 가까운 수로 축소하는 반면, Lasso는 회귀계수를 완전히 0으로 축소함

데이터에 따라 성능이 좋은 알고리즘은 다르다. 입력변수들이 전반적으로 비슷한 수준으로 출력변수에 영향을 미치는 경우에는 Ridge가, 출력변수에 미치는 입력변수의 영향력 편차가 큰 경우에는 Lasso가 좋을 가능성이 큼



Logistic Regression

샘플이 특정 클래스에 속할 확률을 추정하고, 추정 확률이
50%가 넘으면 모델은 그 샘플이 해당 클래스에 속한다고 예측하고,
넘지 않으면 클래스에 속하지 않는다고 예측하는 이진 분류 모델

회귀 분석을 이용하여 분류 task를 해결하는 모델

로지스틱 회귀의 핵심은 선형회귀분석을 이용하여 '클래스 1에 속할 확률'을 예측하는 것

($\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$ 로 확률 p를 예측해보자!)



확률 예측: 범위가 안 맞는다?!

로지스틱 회귀의 핵심은 선형회귀분석을 이용하여 '클래스 1에 속할 확률'을 예측하는 것

그런데 다음과 같이 '='로 연결해서 예측을 할 경우, 범위가 어긋나는 문제 발생:

$$\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k = p$$

우변 p 는 확률의 기본 공리에 따라 $[0,1]$ 사이 값만 갖는 반면,

좌변 $\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$ 는 $(-\infty, +\infty)$ 의 범위를 갖는다.

04. Logistic Regression



p를 바로 예측할 수 없다면 로짓을 먼저 예측한다!

p 를 예측하는 대신 $\ln \frac{p}{1-p}$ 을 예측한다면?

$$\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k = \ln \frac{p}{1-p}$$

→ 양변 모두 $(-\infty, +\infty)$ 로 범위가 맞춰진다!

$\ln \frac{p}{1-p}$ 를 **로짓**이라고 한다.

즉, 회귀분석을 통해 로짓 $\ln \frac{p}{1-p}$ 을 예측하고, 여기서 식을 잘 정리해 p 를 알아내자!

$$\ln \frac{p}{1-p} = \boldsymbol{\beta}^T \mathbf{x} \quad \longrightarrow \quad \frac{p}{1-p} = e^{\boldsymbol{\beta}^T \mathbf{x}} \quad \longrightarrow \quad p = \frac{1}{1 + e^{-(\boldsymbol{\beta}^T \mathbf{x})}}$$

04. Logistic Regression

Logistic Regression

$$\ln \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$



$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}}$$

1. 입력변수들로 선형회귀분석을 하여
로짓을 추정한다.

(이때, $\hat{\beta}_i$ 는 X_i 가 한 단위 증가했을 때 로짓의 증가량을 의미)

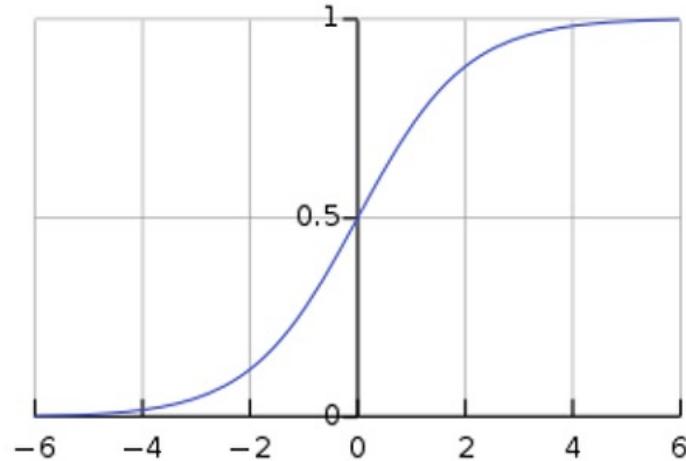
2. 식을 잘 정리해서 '입력이 1로 분류될 확률' p 를 구한다



3. p값이 0.5보다 크면 클래스 1로 분류,
p값이 0.5보다 작으면 클래스 0으로 분류

04. Logistic Regression

Sigmoid function



$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}.$$

정의역: 실수 전체
치역: (0, 1)

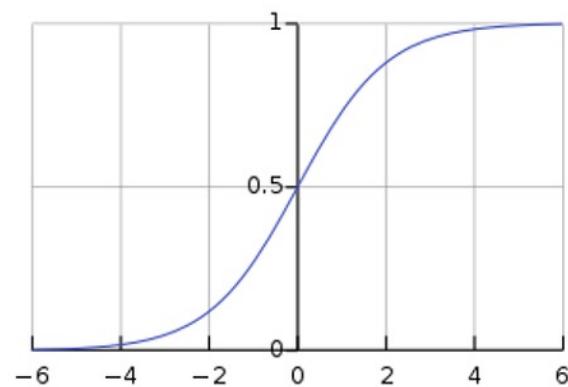
어떤 값을 넣든 **0과 1사이 값**으로 반환해준다!

Logistic Regression

결국 로지스틱 회귀는 주어진 입력변수들의 선형결합을 sigmoid함수에 넣고, 그 결과값을 확률값으로 이해하는 알고리즘으로 이해할 수 있다.

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k)}}$$

위 결과값이 0.5보다 크다면 1로, 0.5보다 작다면 0으로 분류한다.



$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}.$$

04. Logistic Regression

Logistic Regression의 손실함수

Logistic Regression은 OLS(최소자승법)이 아닌 MLE(최대우도법)을 통해 회귀계수 $\beta_0, \beta_1, \beta_2, \dots, \beta_k$ 를 추정한다.

(이 때 우도함수를 그대로 사용하지 않고 변형된 공식을 사용한다.)

$$\text{Max Likelihood} = \prod_{i:y_i=1} p(\mathbf{x}_i) \prod_{i:y_i=0} (1 - p(\mathbf{x}_i))$$



$p(\mathbf{x}_i)$: 클래스 1에 속할 확률
 $1 - p(\mathbf{x}_i)$: 클래스 0에 속할 확률

$\prod_{i:y_i=1} p(\mathbf{x}_i)$:
 실제 클래스가 1인 데이터를 1로 예측할 확률

$\prod_{i:y_i=1} (1 - p(\mathbf{x}_i))$:
 실제 클래스가 0인 데이터를 0으로 예측할 확률

$$\text{Min Loss} = -\sum_i^n y_i \ln\{p(\mathbf{x}_i)\} + (1 - y_i) \ln\{1 - p(\mathbf{x}_i)\}$$

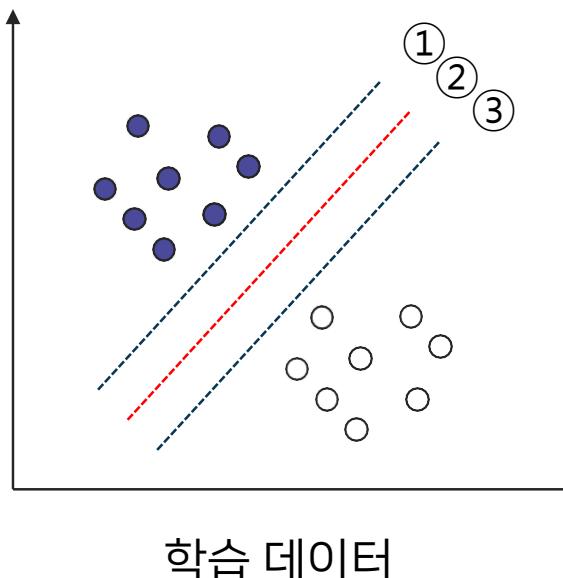
y_i 가 1인 경우 $-\ln\{p(\mathbf{x}_i)\}$ 가 손실함수값이 되고,
 y_i 가 0인 경우 $-\ln\{1 - p(\mathbf{x}_i)\}$ 가 손실함수값이 된다.

이 손실함수를 **이진 크로스 엔트로피(Binary Cross Entropy)**라고 부른다.

SVM이란?

SVM: 클래스(출력변수)가 다른 데이터를 명확하게 구분할 수 있는 **초평면(hyperplane)**을 구하는 알고리즘으로, 초평면 해 공간에서 마진을 최대화하는 하나의 초평면을 탐색한다

아래 그림에서 각 분류기는 모두 학습 데이터를 잘 분류하여 오류가 0이다. 그럼 모두 똑같이 좋은 분류기일까?

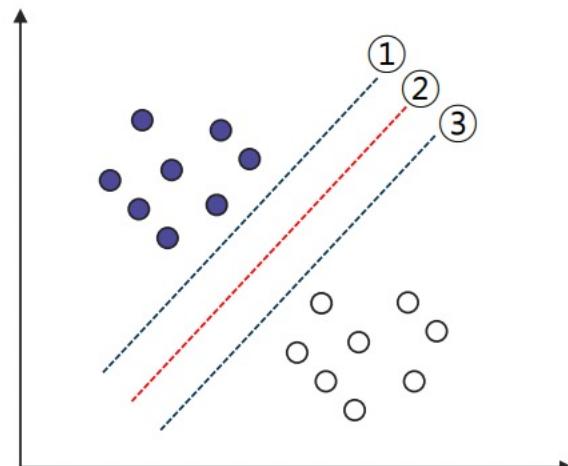


SVM

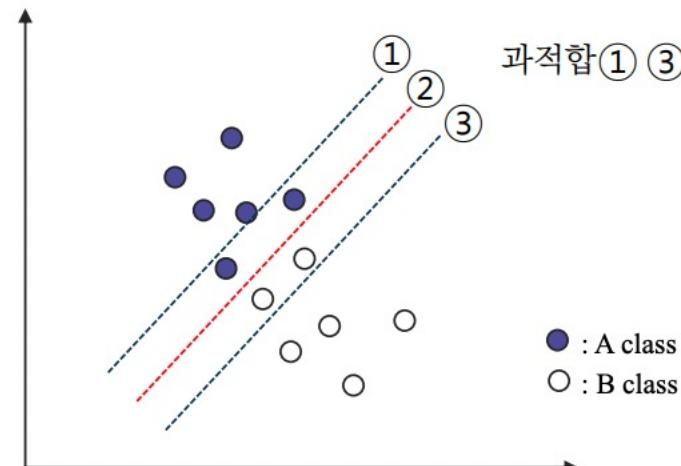
아래 그림에서 각 분류기는 모두 학습 데이터를 잘 분류하여 오류가 0이다. 그럼 모두 똑같이 좋은 분류기일까?

→ No! 각 클래스로부터 모두 멀리 떨어진 2번 분류기가 제일 좋은 분류기이다.

1번과 3번 분류기는 과적합 위험이 존재 (테스트 데이터를 제대로 분류하지 못함)



학습데이터



테스트 데이터

분류기와 각 클래스 데이터 간의 마진을 크게 하여 **일반화 능력을 최대화**하는 것이 SVM의 핵심 아이디어

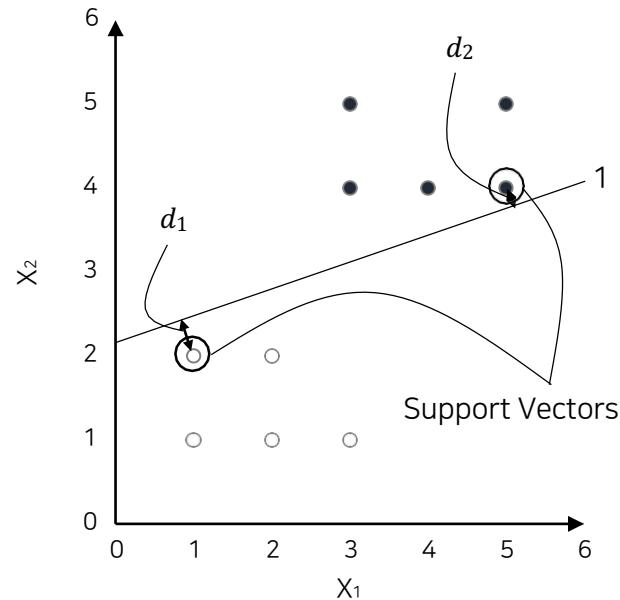
05. SVM

SVM

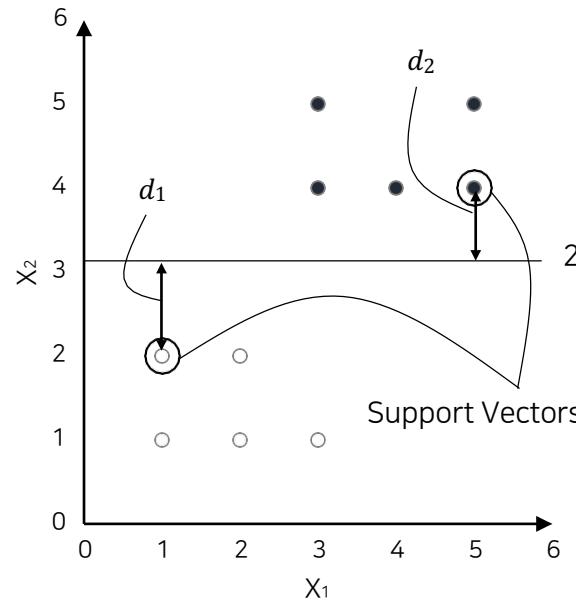
SVM은 Support Vector Machine의 약자로 마진을 최대로 하는 초평면을 구하는 알고리즘

Support Vector: 두 그룹 각각에서 초평면과 가장 가까운 데이터

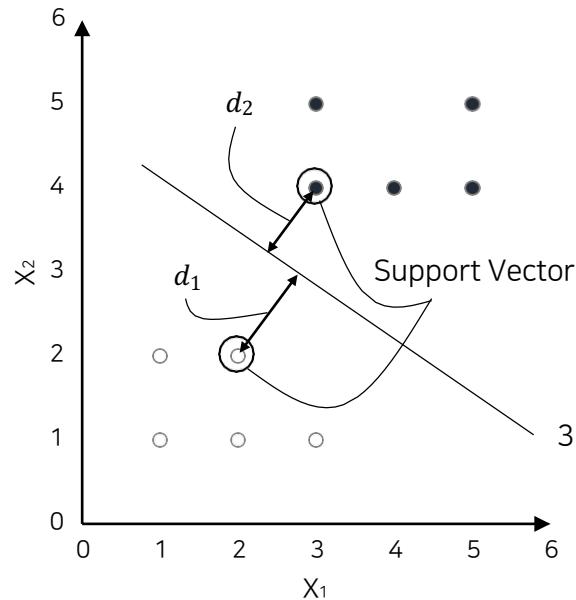
Margin: 두 개의 Support Vector 각각과 초평면과의 최소 거리 합



$$\text{Margin} = d_1 + d_2 = 0.31$$

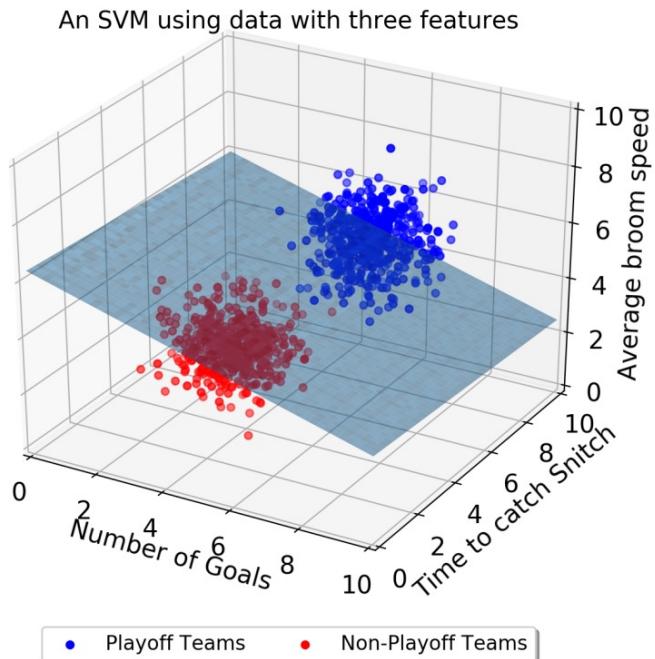


$$\text{Margin} = d_1 + d_2 = 1$$



$$\text{Margin} = d_1 + d_2 = 1.06$$

SVM 참고사항



SVM은 기본적으로 이진 분류를 위한 알고리즘
→ 다중 분류로도 확장이 가능

참고) SVM 초평면으로 분류하는 알고리즘이다.
무조건 직선으로 분류해야 한다는 고정관념은 버릴 것!



SVM의 수리적 모델링

초평면:

$$f(\mathbf{X}) = w_0 + w_1X_1 + w_2X_2 + \dots + w_pX_p$$

(p차원에서 형성되는 초평면)

목적 함수:

Max. M(margin)

제약조건:

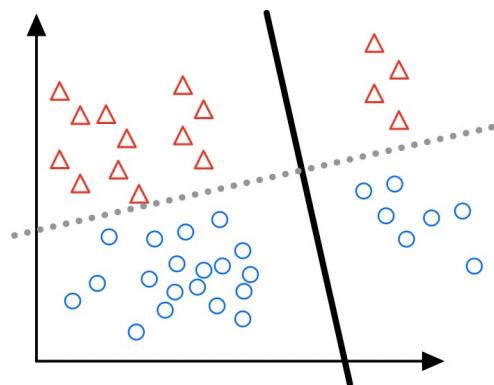
- 1) 대부분의 데이터는 초평면과의 최소거리가 Support Vector와의 초평면의 최소거리보다 커야 한다.
- 2) 일부 데이터에 대해서만 초평면과의 최소거리가 Support Vector와 초평면의 최소거리보다 작아지는 것을 허용한다.

SVM의 목적

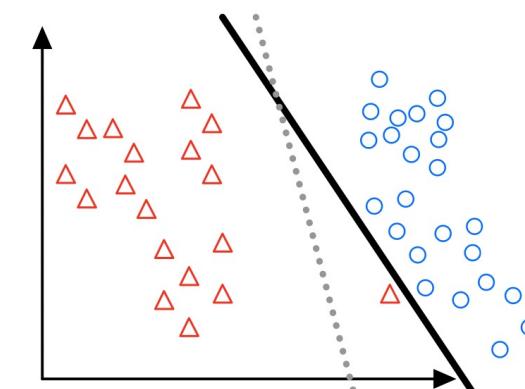
SVM의 목적:

- 1) 마진의 최대화 : 일반화 성능 향상
- 2) 오분류의 최소화 : 정확도 향상

그러나 문제는, 위 두 목표가 서로 상충관계

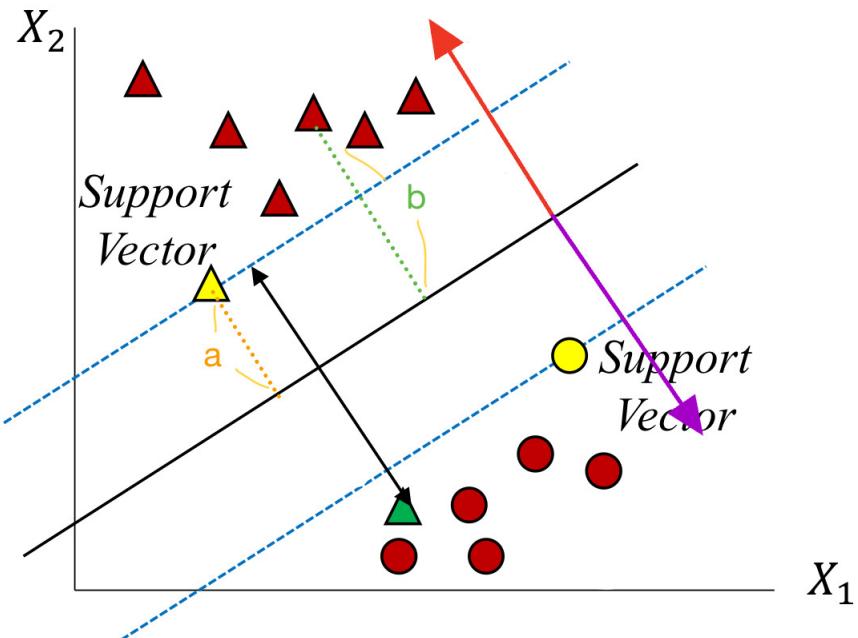


마진의 최대화



오분류의 최소화

SVM의 수리적 모델링: 제약조건



- 1) 대부분의 데이터는 초평면과의 최소거리가 Support Vector와 초평면 간 최소거리보다 커야 한다 ("가급적 오분류가 없었으면 좋겠어")
Ex) 세모는 세모 서포트 벡터보다 위로 더 멀리 있어야 바람직하다
 - 2) 일부 데이터에 대해서만 초평면과의 최소거리가 Support Vector와 초평면의 최소거리보다 작아지는 것을 허용한다. ("마진을 더 크게 해서 일반화 성능을 크게 높일 수 있다면 어느 정도는 오분류가 나도 괜찮아")
- 1, 2번 문장은 모두 방향을 고려하는 개념

SVM의 수리적 모델링

초평면:

$$f(\mathbf{X}) = w_0 + w_1X_1 + w_2X_2 + \dots + w_pX_p$$

(p차원에서 형성되는 초평면)

목적 함수:

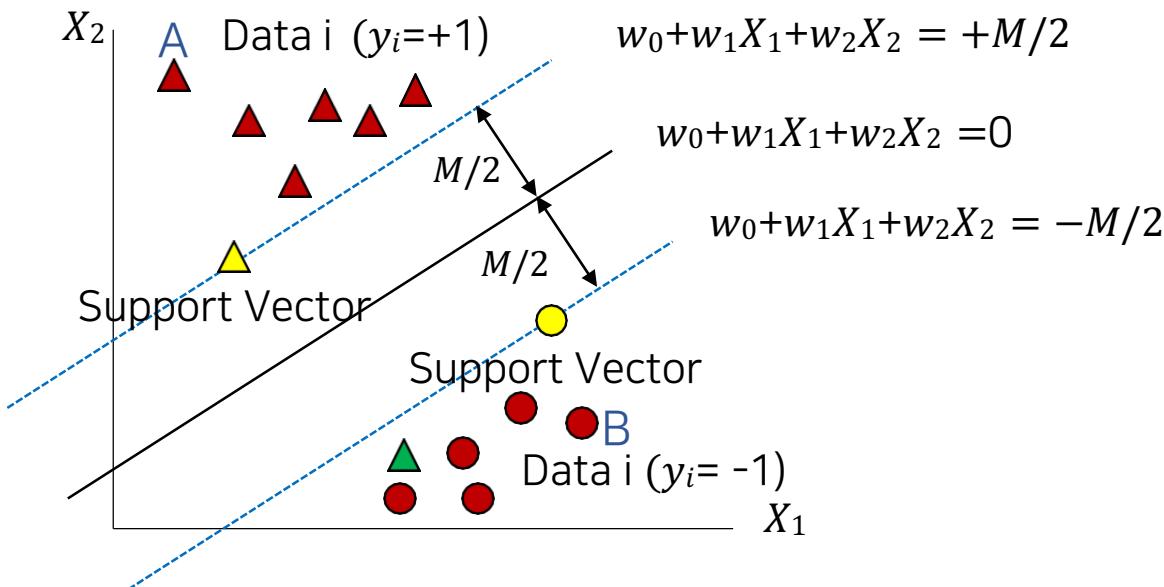
Max. M(margin) 마진의 최대화

제약조건:

- 1) 대부분의 데이터는 초평면과의 최소거리가 Support Vector와의 초평면의 최소거리보다 커야 한다.
- 2) 일부 데이터에 대해서만 초평면과의 최소거리가 Support Vector와 초평면의 최소거리보다 작아지는 것을 허용한다.

이제 수식으로 표현해보자!

SVM의 수리적 모델링: 제약요인

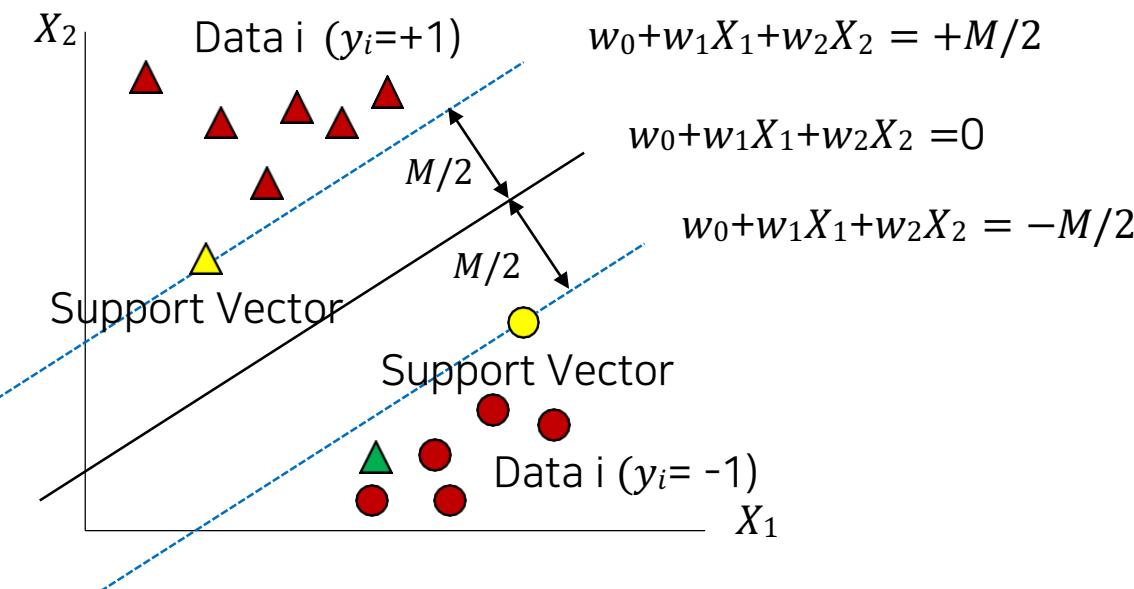


우리가 찾아야 하는 직선을 $w_0 + w_1X_1 + w_2X_2 = 0$ 으로 두자
(w_0, w_1, w_2 , 을 추정해야 한다)

$w_0 + w_1X_1 + w_2X_2 = 0$ 에 평행하고,
각 클래스의 서포트 벡터를 지나는 직선의 방정식은 다음과 같다:
(1) $w_0 + w_1X_1 + w_2X_2 = +M/2$
(2) $w_0 + w_1X_1 + w_2X_2 = -M/2$

직선 (1)보다 위에 있는 점들은 $w_0 + w_1X_1 + w_2X_2 \geq +M/2$
직선 (2)보다 아래에 있는 점들은 $w_0 + w_1X_1 + w_2X_2 \leq -M/2$

SVM의 수리적 모델링: 제약요인



1) 대부분의 데이터는 초평면과의 최소거리가 Support Vector와 초평면 간 최소거리보다 커야 한다
(가급적 오분류를 최소화하자)

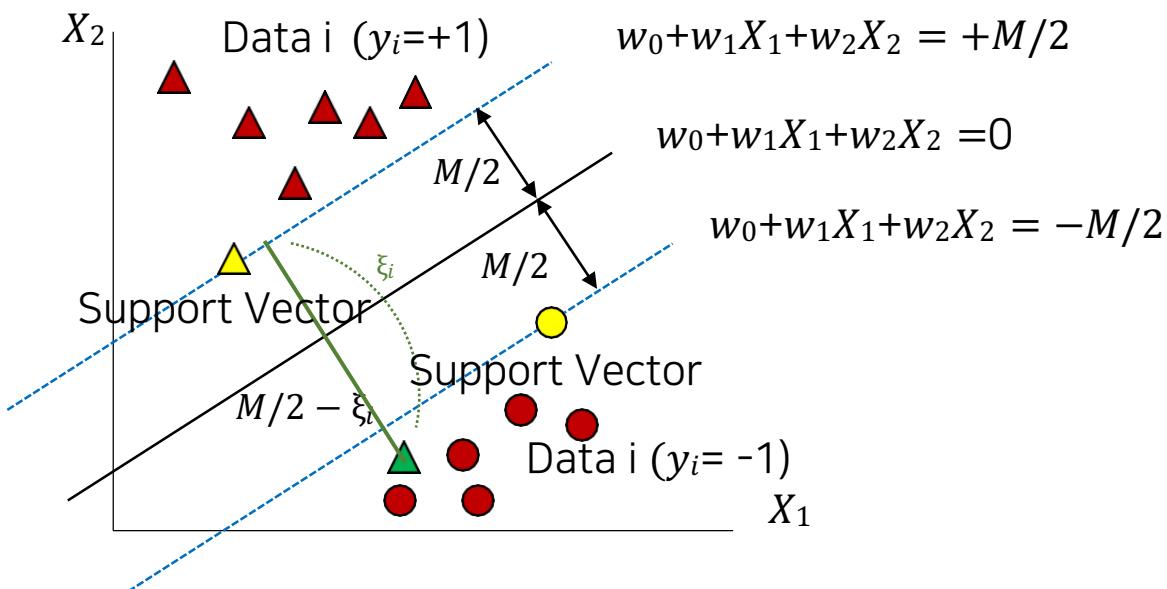
- ▲ For Data i ($y_i=+1$), $w_0 + w_1X_{i1} + w_2X_{i2} \geq +M/2$
 - For Data i ($y_i=-1$), $w_0 + w_1X_{i1} + w_2X_{i2} \leq -M/2$
- ($i = 1, \dots, n$)

두 식을 한 번에 표현하면 다음과 같다:

$$y_i(w_0 + w_1X_{i1} + w_2X_{i2}) \geq M/2$$

$$y_i(\mathbf{W} \cdot \mathbf{X}_i + w_0) \geq M/2$$

SVM의 수리적 모델링: 제약요인



2) 일부 데이터에 대해서만 초평면과의 최소거리가 Support Vector와 초평면의 최소거리보다 작아지는 것을 허용한다.
(일반화 성능이 크게 좋아진다면 어느 정도 오분류는 괜찮아)

▲ For Data i ,

$$y_i(\mathbf{W} \cdot \mathbf{X}_i + w_0) \geq M/2 - \xi_i \quad (i = 1, \dots, n)$$

ξ_i (크사이):

1) 자신이 원래 속해 있어야 하는 클래스의 서포트 벡터를 지나면서 분류 초평면과 평행한 초평면과 오분류를 일으키는 데이터까지의 거리

→ 슬랙 변수(Slack Variable):
오분류되는 데이터를 허용하면서 모델을 더욱 강건하게 만드는 요소

$$\xi_i \geq 0$$

05. SVM

SVM의 수리적 모델링: 제약요인 !!

$$\left. \begin{array}{l} \textcircled{1} \quad \textcolor{red}{\bullet} \quad y_i(\mathbf{W} \cdot \mathbf{X}_i + w_0) \geq M/2 \\ \textcircled{2} \quad \textcolor{green}{\triangle} \quad y_i(\mathbf{W} \cdot \mathbf{X}_i + w_0) \geq M/2 - \xi_i \end{array} \right\}$$

양변을 $M/2$ 로 나누고,
회귀계수들 & 크사이를 재정의

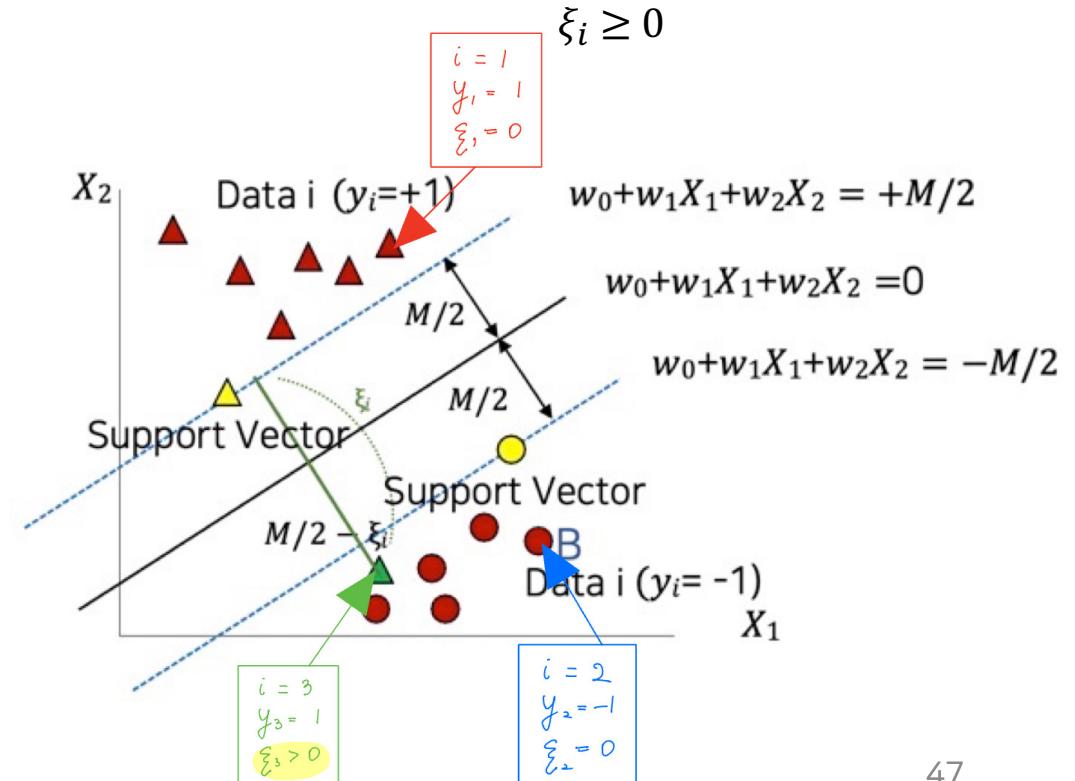
$$y_i(\mathbf{W} \cdot \mathbf{X}_i + w_0) \geq M/2 - \xi_i \quad \xi_i \geq 0$$

$$y_i[(\mathbf{W} \cdot \mathbf{X}_i) + w_0] \geq 1 - \xi_i \quad (i = 1, \dots, n)$$

ξ_i (크사이):

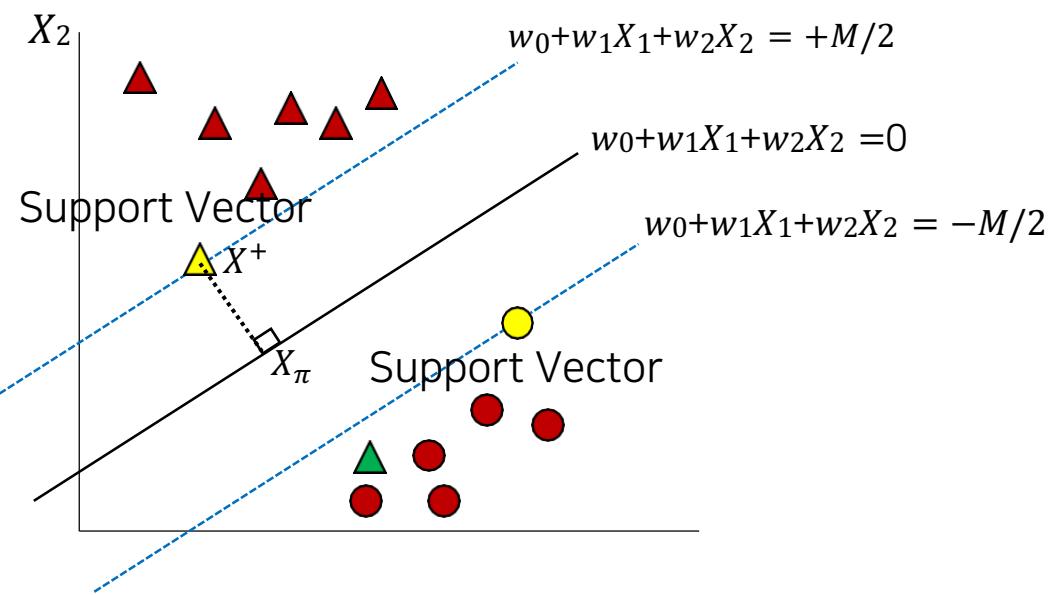
1) 오분류된 데이터라면,
자신이 원래 속해 있어야 하는 클래스의 서포트 벡터를 지나면서
분류 초평면과 평행한 초평면과 오분류를 일으키는 데이터까지의 거리

2) 잘 분류된 데이터라면,
 $\xi_i = 0$



05. SVM

SVM의 수리적 모델링: 목적함수



서포트 벡터(세모)를 X^+ 라고 하고,
 X^+ 에서 분류 초평면에 내린 수선의 발을 X_π 라고 하자.

$$\begin{aligned}
 & w_0 + w_1X_1 + w_2X_2 = \frac{M}{2} \\
 & \frac{2w_0}{M} + \frac{2w_1}{M}X_1 + \frac{2w_2}{M}X_2 = 1 \\
 & w_0 + w_1X_1 + w_2X_2 = 1 \\
 & w_0 + \mathbf{W} \cdot \mathbf{X} = 1
 \end{aligned}$$

양변을 $\frac{M}{2}$ 로 나눈다
계수들을 재정의
내적으로 표현

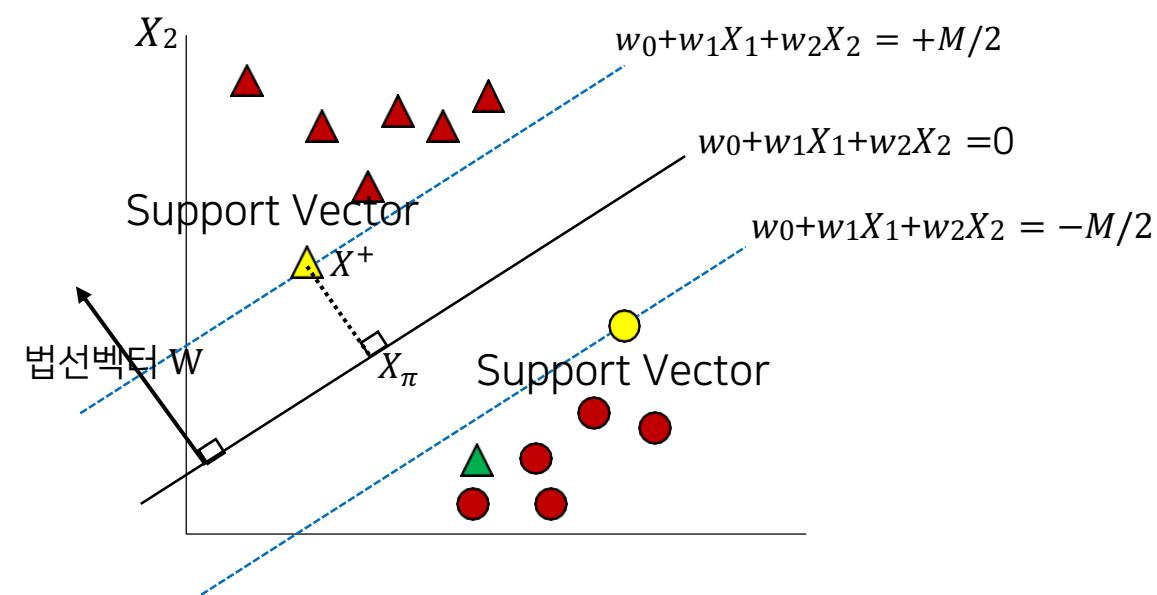
같은 방식으로 $w_0 + w_1X_1 + w_2X_2 = 0$ 는 $w_0 + \mathbf{W} \cdot \mathbf{X} = 0$ 이,
 $w_0 + w_1X_1 + w_2X_2 = -\frac{M}{2}$ 는 $w_0 + \mathbf{W} \cdot \mathbf{X} = -1$ 표현 가능

$$\begin{cases}
 w_0 + \mathbf{W} \cdot \mathbf{X} = 1 & X^+ \text{는 이 위의 점} \\
 w_0 + \mathbf{W} \cdot \mathbf{X} = 0 & X_\pi \text{는 이 위의 점} \\
 w_0 + \mathbf{W} \cdot \mathbf{X} = -1 &
 \end{cases}
 \rightarrow
 \begin{cases}
 w_0 + \mathbf{W} \cdot \mathbf{X}^+ = 1 \\
 w_0 + \mathbf{W} \cdot \mathbf{X}_\pi = 0 \\
 \mathbf{W}(\mathbf{X}^+ - \mathbf{X}_\pi) = 1
 \end{cases}$$

vector

05. SVM

SVM의 수리적 모델링: 목적함수



서포트 벡터(세모)를 X^+ 라고 하고,
 X^+ 에서 분류 초평면에 내린 수선의 발을 X_π 라고 하자.

직선 $w_0 + w_1X_1 + w_2X_2 = 0$ 의 법선 벡터는 $\|w\| = \sqrt{w_1^2 + w_2^2}$ 이다.
 $\|w\|$ 와 $(X^+ - X_\pi)$ 는 평행한 벡터 (사이각 0°)

$$\|w\| (X^+ - X_\pi) = 1$$

$$\|w\| \cdot \|X^+ - X_\pi\| \cdot \cos 0^\circ = 1$$

두 벡터의 내적 공식 이용
 $a \cdot b = \|a\| \cdot \|b\| \cdot \cos \theta$
 θ는 a와 b의 사이각

$$\|X^+ - X_\pi\| = \frac{M}{\|w\|}$$

이건 마진의 $\frac{1}{2}$, 우리가 구하고 싶은 건 마진 문제

$$\therefore M = \frac{2}{\|w\|}$$

SVM의 수리적 모델링: 마진의 최대화 & 하이퍼 파라미터 C

$$\text{Max } M = \text{Max} \frac{2}{\|W\|} = \text{Min} \frac{\|W\|}{2} = \text{Min} \frac{\|W\|^2}{2}$$

$$\text{목적함수 } \text{Min} \frac{\|W\|^2}{2} + C \sum_i \xi_i$$

$C \sum_i \xi_i$: 오분류되는 데이터를 허용하는 정도

C: 하이퍼 파라미터

C값이 작을 수록 데이터를 보다 유연하게 분류한다.
(오분류에 관대하다)

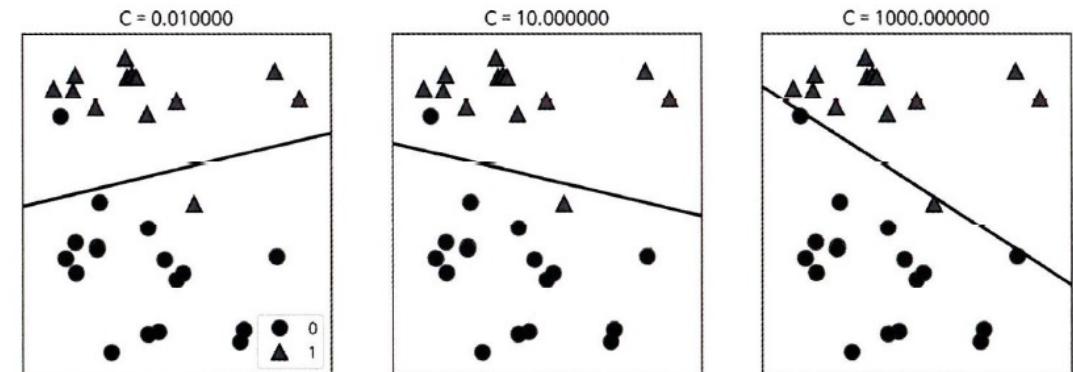
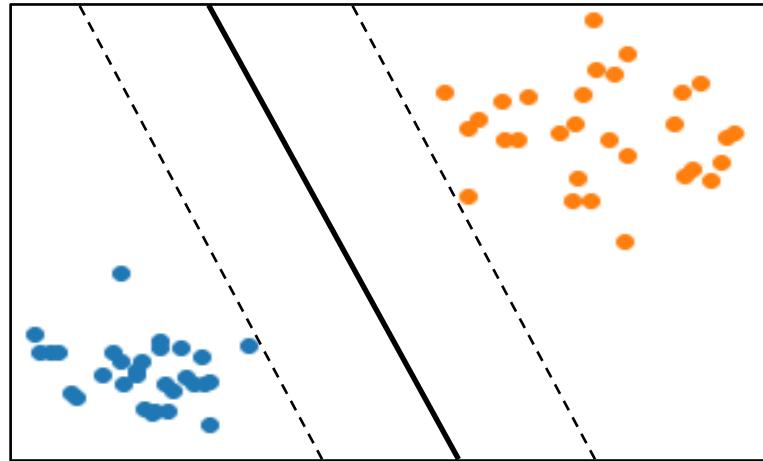


그림 2-16 forge 데이터셋에 각기 다른 C 값으로 만든 선형 SVM 모델의 결정 경계

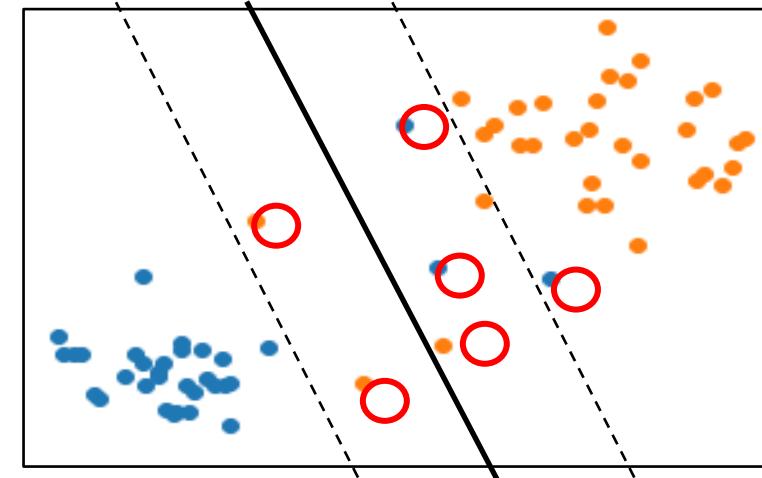
하드 마진 분류 vs 소프트 마진 분류

하드 마진 분류



모든 샘플을 올바르게 분류해야 합니다.
데이터가 선형적으로 구분되어 있어야 하며,
이상치에 민감합니다.
 C 값을 높게 설정한 경우입니다.

소프트 마진 분류



샘플을 조금 더 유연하게 분류합니다.
마진의 크기와 마진 오류 사이에서
적절한 균형을 잡습니다.
 C 값을 낮게 설정한 경우입니다.

SVM의 수리적 모델링: 결론

초평면:

$$f(\mathbf{X}) = w_0 + w_1X_1 + w_2X_2 + \dots + w_pX_p$$

(p차원에서 형성되는 초평면)

따라서 SVM을 학습한다는 것은 최적화 문제

(Quadratic Programming Problem)를 푸는 것이며,

최적화 문제를 풀면 다음과 같은 해를 얻는다:

목적 함수:



$$\text{Min} \frac{\|\mathbf{W}\|^2}{2} + C \sum_i \xi_i$$

(C는 하이퍼 파라미터, C값이 작을 수록 데이터를 보다 유연하게 분류)

$$\mathbf{W} = \sum_i \alpha_i y_i \mathbf{X}_i$$

최적해(Optimal Solution): \mathbf{W}

쌍대해(Dual Solution): α_i

제약조건:

$$y_i[(\mathbf{W} \cdot \mathbf{X}_i) + w_0] \geq 1 - \xi_i \quad (i = 1, \dots, n)$$

$$\xi_i \geq 0$$

05. SVM

최적화 문제의 해를 구하는 과정

Optimization Problem (C-SVM)

✓ Objective function

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i$$

✓ Constraints

$$s.t. y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i$$

Solution

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$



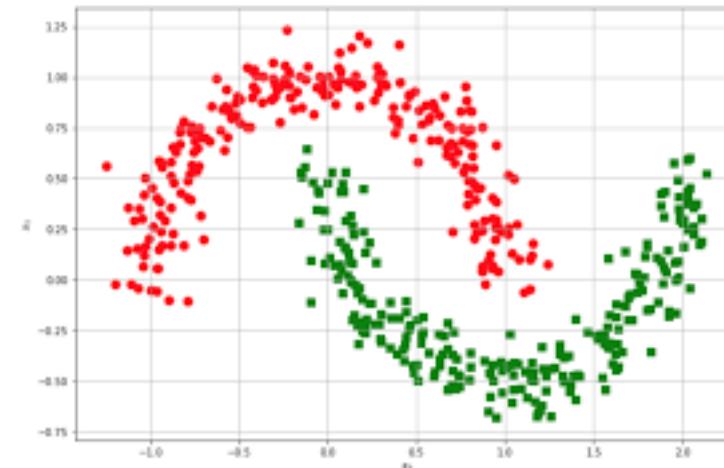
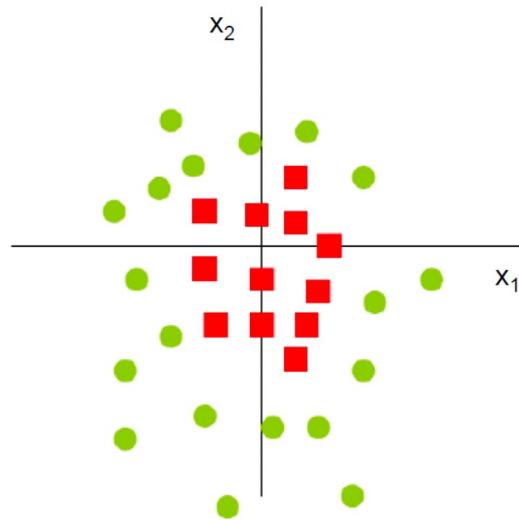
라그랑주승수법(Lagrange multiplier method)

$$\begin{aligned} \min L_P(\mathbf{w}, b, \alpha_i) &= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^N \mu_i \xi_i \\ s.t. \quad \alpha_i &\geq 0 \end{aligned}$$



$$\begin{aligned} \max L_D(\alpha_i) &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ s.t. \quad \sum_{i=1}^N \alpha_i y_i &= 0 \quad and \quad 0 \leq \alpha_i \leq C \end{aligned}$$

비선형 SVM



Linearly **inseparable** data

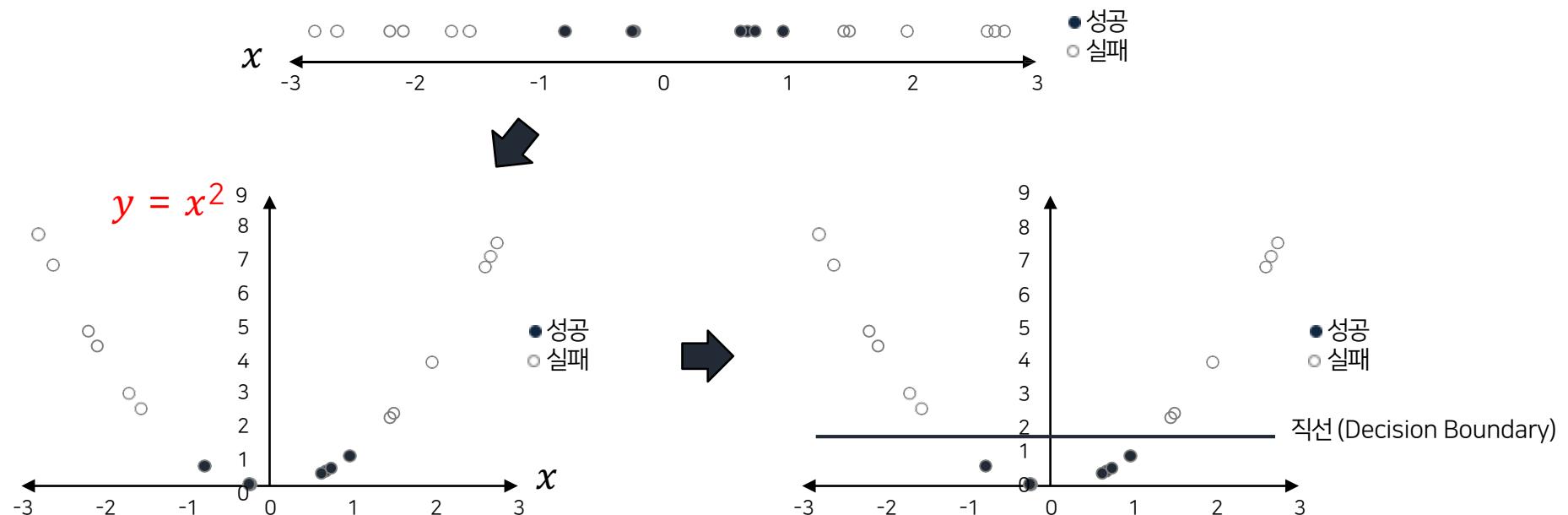
위의 2차원 데이터를 SVM으로 잘 분류할 수 있으신가요?

05. SVM

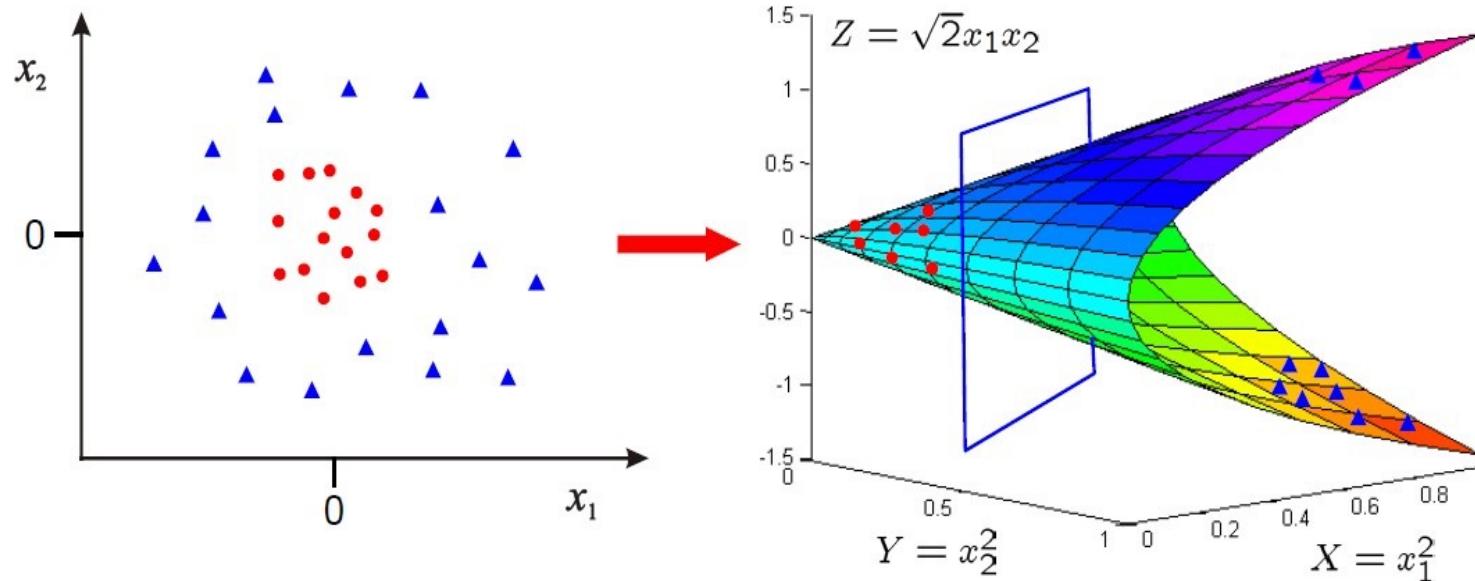
비선형 SVM

비선형 SVM은 **커널 함수(kernel Function)**를 사용하여 마진을 최대로 하는 초평면을 구하는 알고리즘

선형 SVM으로 분류하지 못하는 경우에 사용



비선형 SVM

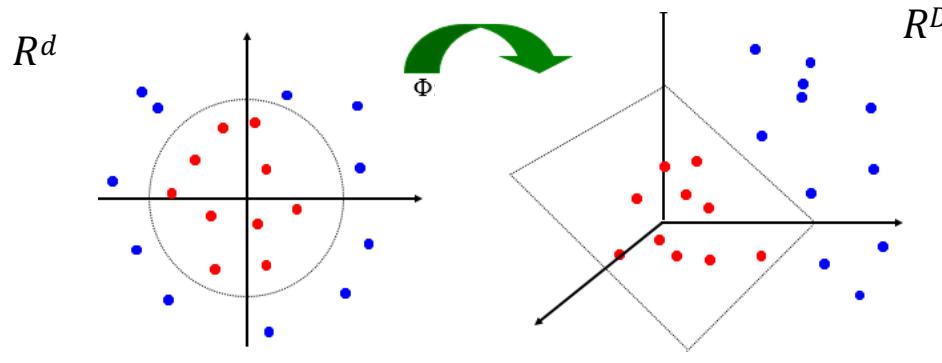


$$\phi : \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \rightarrow \begin{pmatrix} X_1^2 \\ X_2^2 \\ \sqrt{2}X_1X_2 \end{pmatrix} \quad R^2 \rightarrow R^3$$

Feature Map (Function)

비선형 SVM의 수리적 모델링

비선형 SVM은 Feature Map ϕ 를 이용해서 SVM 학습 문제를 재정의한다.



$$\phi: X \rightarrow \phi(X) \quad R^d \rightarrow R^D$$

따라서 R^D 차원에서 선형 SVM의 최적화 문제를 풀면 된다.

R^D 차원에서의 초평면

$$f(X) = \mathbf{W} \cdot \phi(X) + w_0$$



$$\text{Min} \quad \frac{\|\mathbf{W}\|^2}{2} + C \sum_i \xi_i$$

$$\text{s.t. } y_i[(\mathbf{W} \cdot \phi(X_i)) + w_0] \geq 1 - \xi_i \quad i = 1, \dots, n$$

$$\xi_i \geq 0$$

05. SVM

비선형 SVM의 수리적 모델링

Original dimension: R^d

$$f(\mathbf{X}) = \mathbf{W} \cdot \mathbf{X} + w_0$$



Hyperplane의 수리적 표현

High dimension: R^D

$$f(\mathbf{X}) = \mathbf{W} \cdot \phi(\mathbf{X}) + w_0$$



$$\text{Min} \quad \frac{\|\mathbf{W}\|^2}{2} + C \sum_i \xi_i$$

$$\text{s.t. } y_i[(\mathbf{W} \cdot \mathbf{X}_i) + w_0] \geq 1 - \xi_i \quad i = 1, \dots, n$$

$$\xi_i \geq 0$$

Hyperplane을 찾기 위한 최적화식

$$\text{Min} \quad \frac{\|\mathbf{W}\|^2}{2} + C \sum_i \xi_i$$

$$\text{s.t. } y_i[(\mathbf{W} \cdot \phi(\mathbf{X}_i)) + w_0] \geq 1 - \xi_i \quad i = 1, \dots, n$$

$$\xi_i \geq 0$$



커널 함수(Kernel Function)

그러나 $D \gg d$ 인 경우 변수가 너무 많아져서 학습 파라미터인 \mathbf{W} 벡터를 구하기 힘들어지는 현상이 발생함
이 문제를 해결하기 위해서 쌍대해를 이용하는 방법을 소개한다

Original dimension: R^d

$$f(\mathbf{X}) = \mathbf{W} \cdot \mathbf{X} + w_0$$

두 방식은 같은 예측 결과를 낸다



$$\mathbf{W} = \sum_i \alpha_i y_i \mathbf{X}_i$$

$$f(\mathbf{X}) = \sum_i \alpha_i y_i \underline{\mathbf{X}_i^T \mathbf{X}} + w_0$$

데이터의 내적

이제 R^D 차원 문제에서 SVM 문제를 쌍대해를 이용하여 표현하면 다음과 같다.

High dimension: R^D

$$f(\mathbf{X}) = \mathbf{W} \cdot \phi(\mathbf{X}) + w_0$$

두 방식은 같은 예측 결과를 낸다



$$f(\mathbf{X}) = \sum_i \alpha_i y_i \underline{\phi(\mathbf{X}_i)^T \phi(\mathbf{X})} + w_0$$

Feature Map의 내적

이 식에서 $k(\mathbf{X}_i, \mathbf{X}_j) = \phi(\mathbf{X}_i)^T \phi(\mathbf{X}_j)$ 로 정의하고 k 를 **커널 함수(Kernel Function)**이라고 부른다.

커널 함수(Kernel Function)

$$f(\mathbf{X}) = \sum_i \alpha_i y_i \phi(\mathbf{X}_i)^T \phi(\mathbf{X}) + w_0$$

$k(\mathbf{X}_i, \mathbf{X}_j) = \phi(\mathbf{X}_i)^T \phi(\mathbf{X}_j)$ 로 정의하고 k 를 **커널 함수(Kernel Function)**이라고 부른다.

커널 트릭(Kernel Trick): 데이터를 직접 변환해서 내적을 하는 것이 아니라, 변환된 곳에서의 내적을 바로 나타내는 트릭 (ϕ 를 모르더라도 혹은 $\phi(\mathbf{X}_i)$ 를 계산하지 않더라도 내적값인 $\phi(\mathbf{X}_i)^T \phi(\mathbf{X}_j)$ 만 알면 초평면을 구할 수 있다!)

기존의 방법: 데이터 → 변환 → 내적
커널 트릭: 데이터 → skip → 내적

결국 커널 함수는 변환을 안 거치고 바로 내적 결과를 끌어올 수 있게 하는 함수



비선형 SVM의 수리적 모델링

비선형 SVM에서 사용하는 대표적인 커널 함수는 아래와 같이 4 종류가 있다

선형 함수(Linear)

$$k(\mathbf{X}_i, \mathbf{X}_j) = \mathbf{X}_i^T \mathbf{X}_j$$

다항 함수(Polynomial)

$$k(\mathbf{X}_i, \mathbf{X}_j) = (1 + \mathbf{X}_i^T \mathbf{X}_j)^d$$

방사형 함수(Radial Basis)

$$k(\mathbf{X}_i, \mathbf{X}_j) = \exp(-\gamma (\mathbf{X}_i - \mathbf{X}_j)^2)$$

시그모이드 함수(Sigmoid)

$$k(\mathbf{X}_i, \mathbf{X}_j) = \tanh(\kappa \mathbf{X}_i^T \mathbf{X}_j + c), \kappa > 0 \text{ and } c < 0$$

차원 증가

- 
1. 비선형성이 증가하여 복잡한 분류 문제를 해결
 2. 그러나 과대적합 될 위험도 증가



커널 트릭 (Kernel Trick) 예제

커널 함수 k 는 Feature Map $\phi(\mathbf{X}_i)$ 을 내적한 함수로, 커널 함수의 예제를 아래와 같이 살펴보자.
2차원 데이터 $\mathbf{X} = [X_1, X_2]$ 가 주어졌을 때 커널 함수 $k(X_1, X_2) = (1 + \mathbf{X}_i^T \mathbf{X}_j)^2$ 라고 하자.

$$\begin{aligned} k(\mathbf{X}_i, \mathbf{X}_j) &= (1 + \mathbf{X}_i^T \mathbf{X}_j)^2 = 1 + X_{i1}^2 X_{j1}^2 + 2X_{i1}X_{j1}X_{i2}X_{j2} + X_{i2}^2 X_{j2}^2 + 2X_{i1}X_{j1} + 2X_{i2}X_{j2} \\ &= [1 \ X_{i1}^2 \ \sqrt{2}X_{i1}X_{i2} \ X_{i2}^2 \ \sqrt{2}X_{i1} \ \sqrt{2}X_{i2}]^T [1 \ X_{j1}^2 \ \sqrt{2}X_{j1}X_{j2} \ X_{j2}^2 \ \sqrt{2}X_{j1} \ \sqrt{2}X_{j2}] \\ &= \phi(\mathbf{X}_i)^T \phi(\mathbf{X}_j) \text{ 여기서 } \phi(\mathbf{X}) = [1 \ X_1^2 \ \sqrt{2}X_1X_2 \ X_2^2 \ \sqrt{2}X_1 \ \sqrt{2}X_2] \end{aligned}$$



Multiclass SVM

Multiclass SVM은 2개 이상의 클래스를 분류할 수 있는 알고리즘이다.

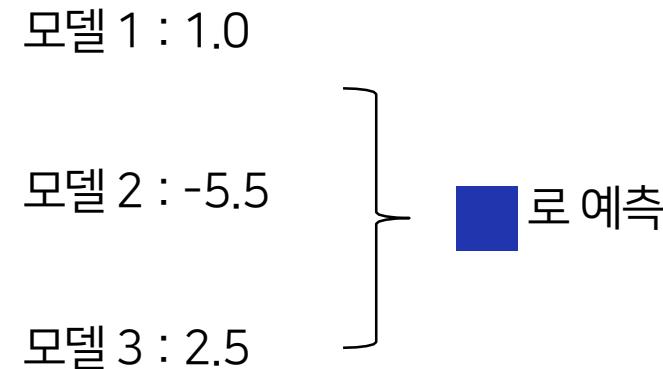
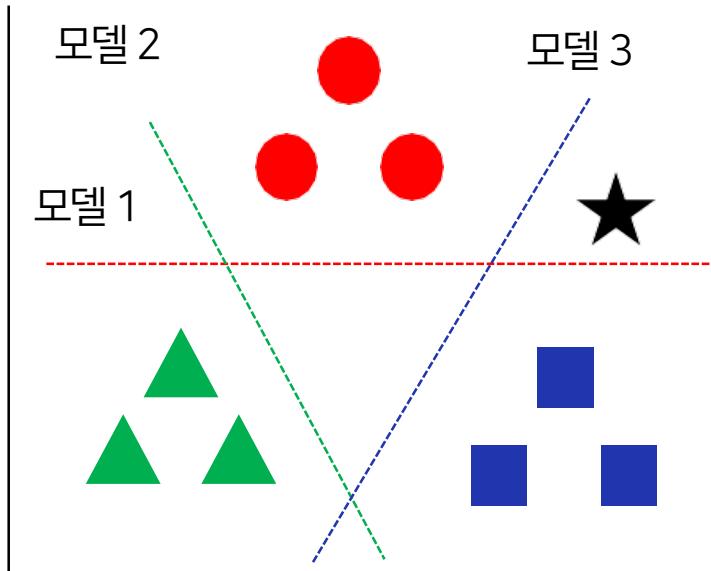
Multiclass SVM을 수행하는 방법은 일반적으로 2가지로 나뉜다:

- 1) One-Versus-Rest 방법: 클래스 별로 해당 클래스와 나머지 클래스를 구별하는 모델을 만듦.
이를 통해 새로운 데이터가 들어왔을 때 각 모델 중 가장 높은 출력값을 가진 클래스로 분류하는 방법

- 2) One-Versus-One 방법: m 개의 클래스 중 2개의 클래스를 선택하여 초평면을 만드는데, 이를 모든 조합에 대해 만든다(총 mC_2 개의 초평면이 생성됨). 새로운 데이터가 들어오면 각 모델마다 클래스를 예측해서 가장 많이 예측된 클래스로 분류하는 투표시스템이다.

One-Versus-Rest

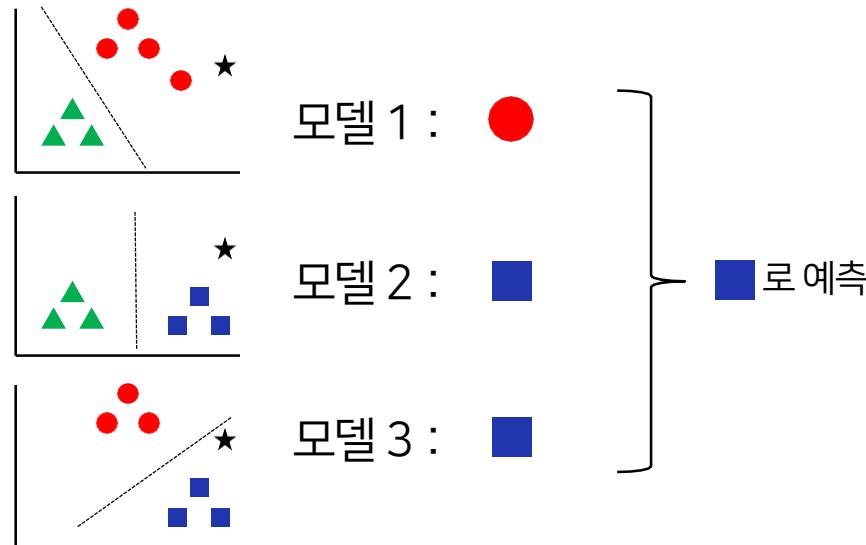
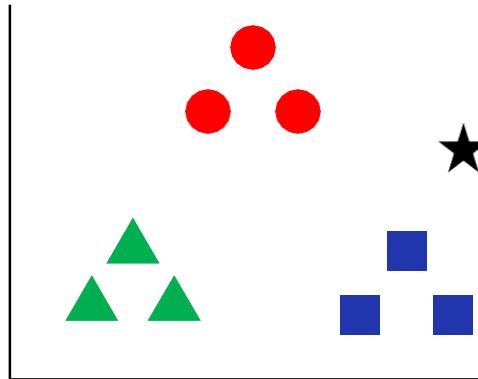
테스트 데이터: ★



클래스 별로 해당 클래스와 나머지 클래스를 구별하는 모델을 만들고,
이를 통해 새로운 데이터가 들어왔을 때 각 모델 중 가장 높은 출력값을 가진 클래스로 분류하는 방법

One-Versus-One

테스트 데이터: ★



m 개의 클래스 중 2개를 선택하여 초평면을 만드는데, 이를 모든 조합에 대해 만든다(총 mC_2 개의 초평면이 생성된다).

새로운 데이터가 들어오면 각 모델마다 클래스를 예측해서 가장 많이 예측된 클래스로 분류하는 투표시스템이다.

선형회귀분석 (Linear Regression)

선형회귀분석 (Linear Regression) : 독립변수와 종속변수 간 선형 상관관계를 모델링하는 기법

단순선형회귀분석 (Simple Linear Regression) : $\hat{y} = \beta_0 + \beta_1 x_1$ (독립변수 1개, 선형관계)

다중선형회귀분석 (Multiple Linear Regression) : $\hat{y} = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$ (독립변수 k개, 선형 관계)

회귀계수(학습 파라미터): $\beta_0, \beta_1, \beta_2, \dots, \beta_k$

회귀계수 $\beta_1, \beta_2, \dots, \beta_k$ 는 (β_0 제외) X가 한 단위 증가했을 때 Y의 평균적인 변화량을 의미한다.

회귀계수 추정방법: (1) 최소자승법 (OLS) : 정규방정식 이용, SVD, 경사하강법

(2) 최대우도법 (MLE)

06. Summary

Ridge, Lasso

다중선형회귀에서 변수의 수가 너무 많아지면 과대적합 또는 다중공선성 문제가 발생할 수 있다.

- 과대적합 판단지표: R_{adj}^2 와 R_{pred}^2
- 다중공선성 판단지표: 상관계수 (행렬), VIF

규제, 정규화 (Regularization) : 회귀계수에 관한 규제항을 추가함으로써 영향력이 없는 입력변수의 효과를 제거

Ridge 회귀에서는 회귀계수의 제곱합을 $f(\hat{\beta})$ 에 대입 (L2 규제): Minimize $\sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \hat{\beta}_j^2$

Lasso 회귀에서는 회귀계수의 절대값 합을 $f(\hat{\beta})$ 에 대입 (L1 규제): Minimize $\sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p |\hat{\beta}_j|$

Ridge는 회귀계수를 0에 가까운 수로 축소하는 반면, Lasso는 회귀계수를 완전히 0으로 축소함

06. Summary

Logistic Regression

샘플이 특정 클래스에 속할 확률을 추정하고, 추정 확률이 50%가 넘으면 모델은 그 샘플이 해당 클래스에 속한다고 예측하고, 넘지 않으면 클래스에 속하지 않는다고 예측하는 이진 분류 모델

회귀 분석을 이용하여 분류 task를 해결하는 모델

로지스틱 회귀의 핵심은 선형회귀분석을 이용하여 p (클래스 1에 속할 확률)을 예측하는 것

$$\ln \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

입력변수들로 선형회귀분석을 하여
로짓을 추정한다.



$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}}$$

식을 잘 정리해서
'입력이 1로 분류될 확률' p 를 구한다



p 값이 0.5보다 크면 클래스 1로 분류,
 p 값이 0.5보다 작으면 클래스 0으로 분류

Logistic Regression의 손실함수(Binary Cross Entropy):

$$\text{Min Loss} = - \sum_i^n y_i \ln\{p(x_i)\} + (1 - y_i) \ln\{1 - p(x_i)\}$$

06. Summary

SVM

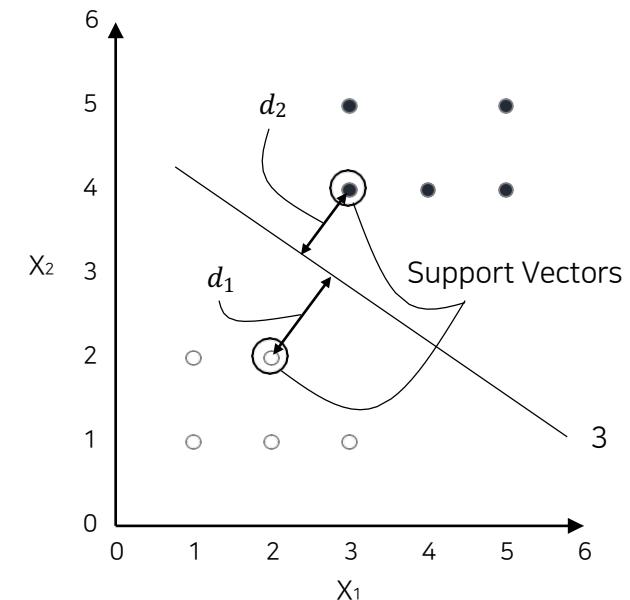
SVM: 클래스(출력변수)가 다른 데이터를 명확하게 구분할 수 있는 초평면(hyperplane)을 구하는 알고리즘으로, 초평면 해 공간에서 마진을 최대화하는 하나의 초평면을 탐색한다.

Support Vector: 두 그룹 각각에서 초평면과 가장 가까운 데이터

Margin: 두 개의 Support Vector 각각과 초평면과의 최소 거리 합

SVM은 기본적으로 이진 분류를 위한 알고리즘 → 다중 분류로도 확장이 가능

- 1) One-Versus-Rest
- 2) One-Versus-One



$$\text{Margin} = d_1 + d_2 = 1.06$$

06. Summary

SVM의 수리적 모델링

초평면: $f(\mathbf{X}) = w_0 + w_1X_1 + w_2X_2 + \dots + w_pX_p$

목적 함수: $\text{Min} \frac{\|\mathbf{w}\|^2}{2} + C \sum_i \xi_i$

제약조건: $y_i[(\mathbf{W} \cdot \mathbf{X}_i) + w_0] \geq 1 - \xi_i \quad (i = 1, \dots, n)$

$$\xi_i \geq 0$$



$$\mathbf{W} = \sum_i \alpha_i y_i \mathbf{X}_i$$

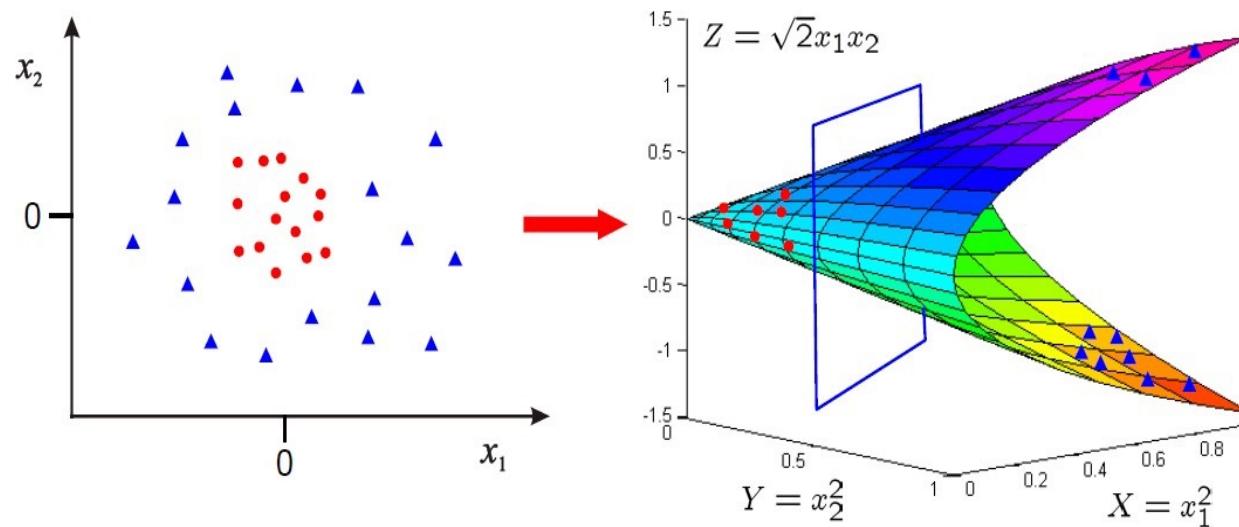
최적해(Optimal Solution): \mathbf{W}

쌍대해(Dual Solution): α_i

비선형 SVM

비선형 SVM은 커널 함수(Kernel Function)를 사용하여 마진을 최대로 하는 초평면을 구하는 알고리즘

선형 SVM으로 분류하지 못하는 경우에 사용



출처 & 참고문헌

- 김창옥 교수님 <머신 러닝과 산업 응용> 강의안
- 6기 박준우님 <Supervised Learning> 세션 자료
- 7기 김예진님 <Supervised Learning> 세션 자료
- 공돌이의 수학정리노트 Blog
- <핸즈온 머신러닝> 2판

DATA SCIENCE LAB

발표자 백민준 010-3029-6815
E-mail: mjoon0309@gmail.com