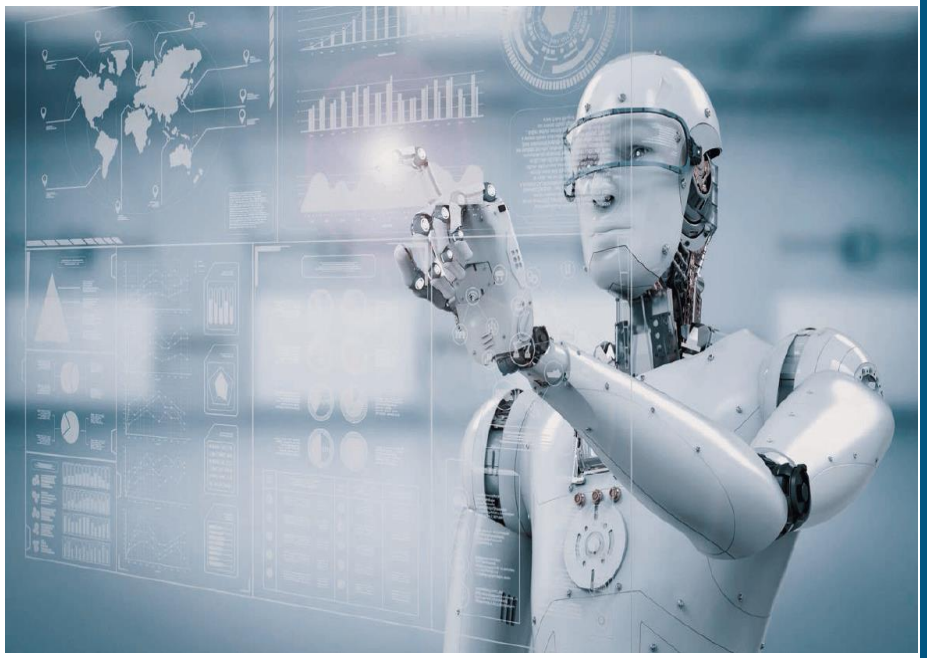


기계학습개론

- 회귀 알고리즘2 -

교수 이홍로

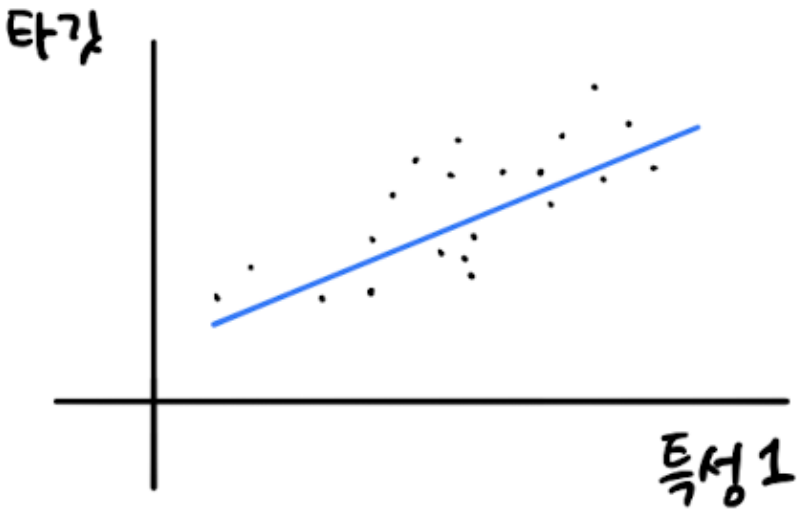
MP : 010-6611-3896
E-mail : hrlee@cnu.ac.kr
강의 홈페이지 : https://cyber.hanbat.ac.kr/

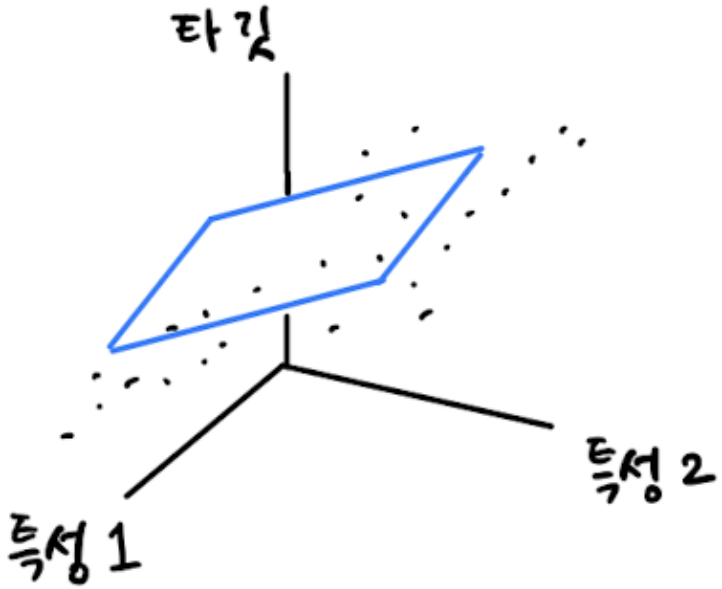
오늘의 강의 목표

- 다중 회귀 알고리즘
- 릿지 회귀 알고리즘
- 라쏘 회귀 알고리즘

다중 회귀 알고리즘

- **점수가 높은 모델을 생성하려면?**
  - 다양한 특성을 활용할수록 정확도는 높아진다.





다중 회귀 알고리즘

- 다양한 데이터 준비 (Width, Height, Length)

```
import pandas as pd
import numpy as np
url='https://drive.google.com/file/d/19RzKaMLzEJ7JsBY0ukaLMiay3K8xp-
4S/view?usp=sharing'
url='https://drive.google.com/uc?id=' + url.split('/')[-2]
df = pd.read_csv(url)

product_total = df.to_numpy()
product_weight = np.array(
    [5.9, 32.0, 40.0, 51.5, 70.0, 100.0, 78.0, 80.0, 85.0, 85.0,
     110.0, 115.0, 125.0, 130.0, 120.0, 120.0, 130.0, 135.0, 110.0,
     130.0, 150.0, 145.0, 150.0, 170.0, 225.0, 145.0, 188.0, 180.0,
     197.0, 218.0, 300.0, 260.0, 265.0, 250.0, 250.0, 300.0, 320.0,
     514.0, 556.0, 840.0, 685.0, 700.0, 700.0, 690.0, 900.0, 650.0,
     820.0, 850.0, 900.0, 1015.0, 820.0, 1100.0, 1000.0, 1100.0,
     1000.0, 1000.0]
     )
print(product_total)
```

다중 회귀 알고리즘

- 사이킷런을 이용하여 훈련 및 테스트 데이터 생성하기

```
from sklearn.model_selection import train_test_split


train_data, test_data, train_answer, test_answer = train_test_split(
product_total, product_weight, random_state=42)
```

다중 회귀 알고리즘

- 사이킷런 변환기를 적용하여 다중 특징 만들기

```
from sklearn.preprocessing import PolynomialFeatures


poly = PolynomialFeatures(include_bias=False)


poly.fit(train_data)


train_poly = poly.transform(train_data)
test_poly = poly.transform(test_data)


print(train_poly.shape, test_poly.shape)
print(poly.get_feature_names_out())
```

```
(42, 9) (14, 9) ['x0' 'x1' 'x2'
'x0^2' 'x0 x1' 'x0 x2' 'x1^2' 'x1
x2' 'x2^2']
```

다중 회귀 알고리즘

- 다중 회귀 알고리즘을 적용하여 학습시키기(모델생성 및 평가)

```
from sklearn.linear_model import LinearRegression


model = LinearRegression()
model.fit(train_poly, train_answer)


print('Training Set : ' + str(model.score(train_poly, train_answer)))
print('Test Set     : ' + str(model.score(test_poly, test_answer)))
```

```
Training Set : 0.9903183436982125
Test Set : 0.9714559911594111
```

다중 회귀 알고리즘

- 사이킷런 변환기를 적용하여 다중 특징 만들기(55개)

```
from sklearn.preprocessing import PolynomialFeatures

poly = PolynomialFeatures(degree=5, include_bias=False)

poly.fit(train_data)

train_poly = poly.transform(train_data)
test_poly = poly.transform(test_data)

print(train_poly.shape, test_poly.shape)
print(poly.get_feature_names_out())
```

특징의 수를 5제곱까지 생성

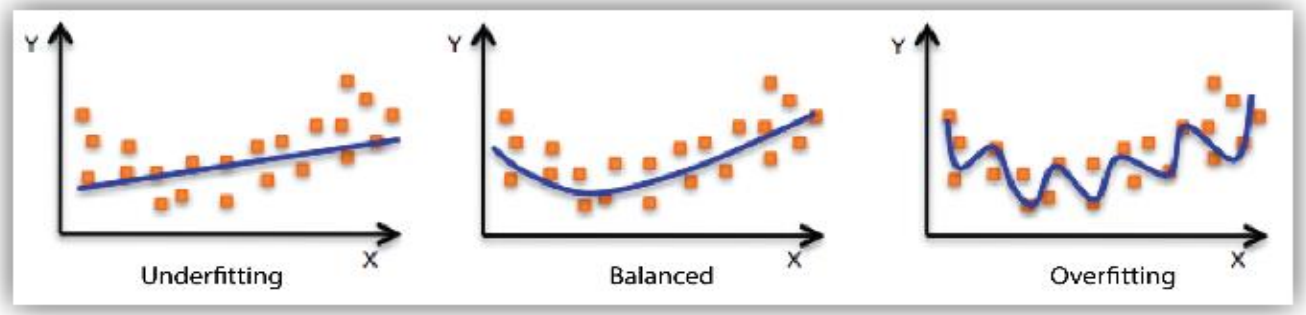
```
(42, 55) (14, 55) ['x0' 'x1' 'x2' 'x0^2' 'x0
x1' 'x0 x2' 'x1^2' 'x1 x2' 'x2^2' 'x0^3' 'x0^2
x1' 'x0^2 x2' 'x0 x1^2' 'x0 x1 x2' 'x0 x2^2'
'x1^3' 'x1^2 x2' 'x1 x2^2' 'x2^3' 'x0^4' 'x0^3
x1' 'x0^3 x2' 'x0^2 x1^2' 'x0^2 x1 x2' 'x0^2
x2^2' 'x0 x1^3' 'x0 x1^2 x2' 'x0 x1 x2^2' 'x0
x2^3' 'x1^4' 'x1^3 x2' 'x1^2 x2^2' 'x1 x2^3'
'x2^4' 'x0^5' 'x0^4 x1' 'x0^4 x2' 'x0^3 x1^2'
'x0^3 x1 x2' 'x0^3 x2^2' 'x0^2 x1^3' 'x0^2
x1^2 x2' 'x0^2 x1 x2^2' 'x0^2 x2^3' 'x0 x1^4'
'x0 x1^3 x2' 'x0 x1^2 x2^2' 'x0 x1 x2^3' 'x0
x2^4' 'x1^5' 'x1^4 x2' 'x1^3 x2^2' 'x1^2 x2^3'
'x1 x2^4' 'x2^5']
```

다중 회귀 알고리즘

- **Training Set와 Test Set의 모델 평가 다시 해보기**

```
model.fit(train_poly, train_answer)
print('Training Set : ' + str(model.score(train_poly, train_answer)))
print('Test Set     : ' + str(model.score(test_poly, test_answer)))
```

```
Training Set : 0.9999999999996433
Test Set : -144.40579436844948
```

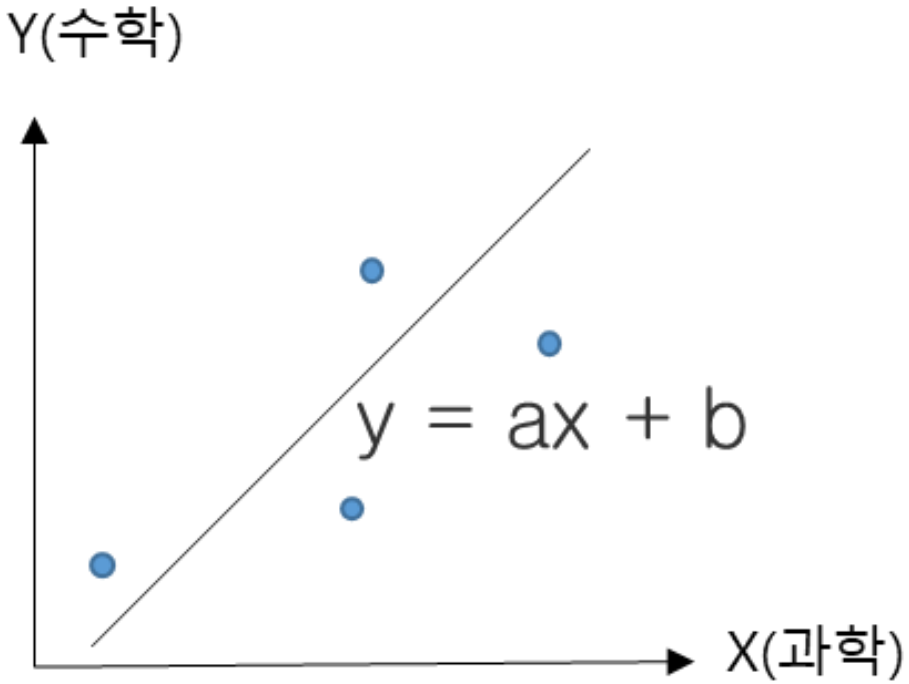


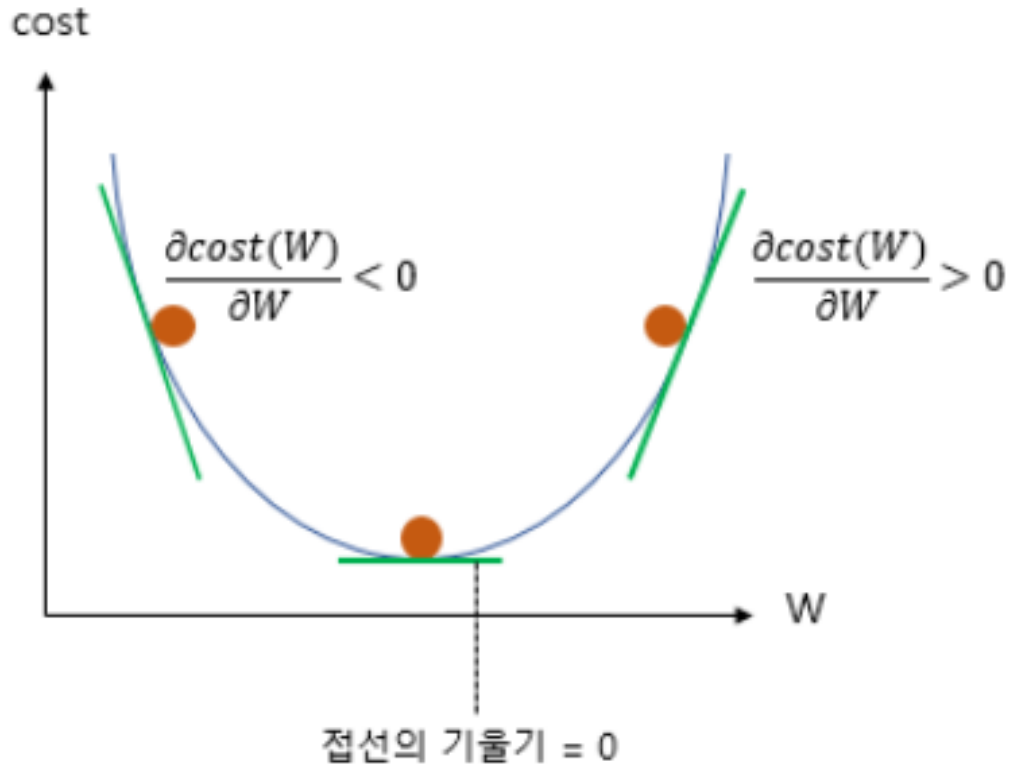
특징이 많아지면 강력한 모델이 생성되나, Training Set에만 최적화되어 Test Set 결과는 낮은 성능을 보인다.

규제 알고리즘(릿지, 라쏘)

- 선형 회귀 알고리즘의 경우

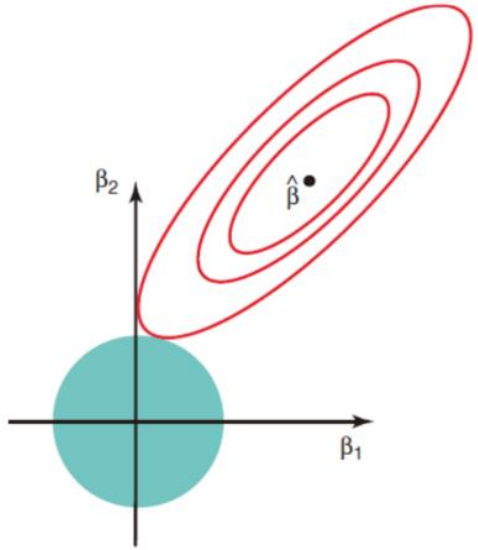
$$cost(W,b) = \frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)} - H(x^{(i)})\right]^2$$



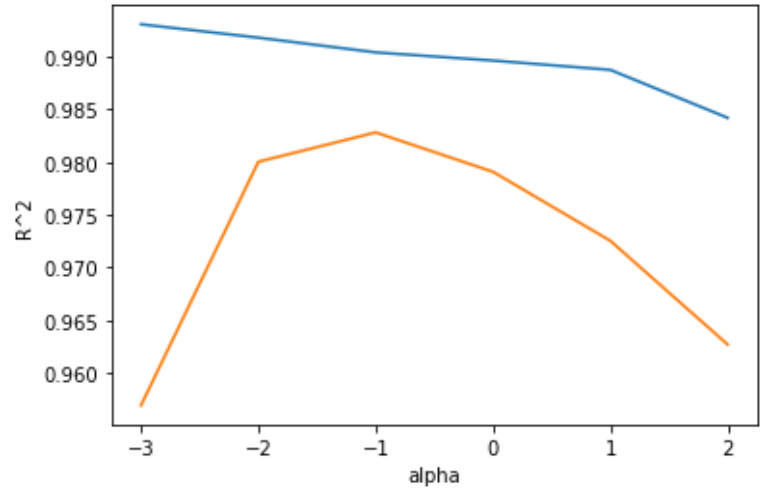


릿지 알고리즘

- **릿지 알고리즘의 경우**
  - 적절한 가중치와 편향을 찾기 위해 추가적인 제약조건을 주는 것
  - MSE가 최소가 되게 하는 가중치와 편향을 찾으면서 동시에, 가중치들의 합이 최소가 되게 한다는 것



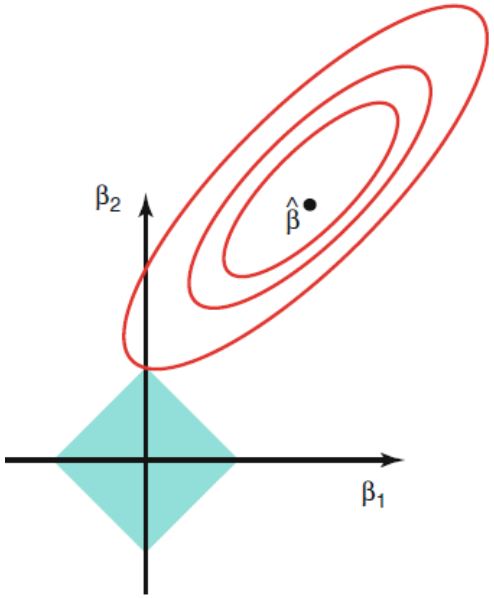
$$J(\boldsymbol{\theta}) = \mathrm{MSE}(\boldsymbol{\theta}) + \alpha\frac{1}{2}\sum_{i=1}^{n}\theta_i^2$$

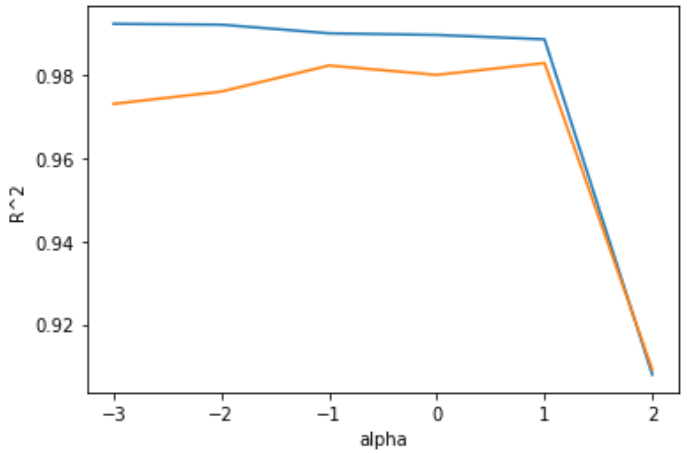


라쏘 알고리즘

- **라쏘 알고리즘**
  - 적절한 가중치와 편향을 찾기 위해 추가적인 제약조건을 주는 것
  - MSE가 최소가 되게 하는 가중치와 편향을 찾으면서 동시에, 가중치들의 절댓값의 합이 최소가 되게 한다는 것

$$J(\boldsymbol{\theta}) = \text{MSE}(\boldsymbol{\theta}) + \alpha \sum_{i=1}^{n} | \theta_i |$$





“어떤 일이 충분히 중요하다면
역경이 있더라도
그것을 반드시 해야만 한다.”

-엘론 머스크-