

In []:

04.03 스케일링

- OLS모형으로 회귀분석 실시 -> 조건수 큰 경우를 해결하기 위한 방안 1) 스케일링

1) 조건수

- 가장 큰 고유치와 가장 작은 고유치의 비율
- 회귀분석에선, 공분산행렬의 가장 큰 고유치와 가장 작은 고유치 비율
- 회귀분석에선, 조건수는 항상 양수 (공분산 행렬 = 대칭 + 양의정부호행렬. 양의정부호 = 고유값 모두 양수)

*공분산행렬은 대칭 + 양의정부호! (그래야 역행렬 가능)

- 조건수가 크면, 발생하는 오차에 대한 solution의 민감도가 커진다. (4page)

$$\begin{array}{c} Ax = b \quad <\text{조건수 작으면}> \\ \downarrow \\ \text{조금더오류} \quad \text{조금더오류} \\ \text{조건수 크면} \end{array} \iff \begin{array}{c} XW = Y \\ \downarrow \\ \text{조금더오류} \quad \text{조금더오류} \\ \text{조건수 크면} \end{array}$$

(4.5원, 4.5원은 때
W값의 변화↑↑
회귀방정식 (X 조건수↑하면)
문제!

- 조건수가 가장 작은 경우(단위행렬, 조건수 = 1), 가장 큰 경우(힐버트행렬, 조건수 = 15,000) (4,

In [3]:

```
b = np.ones(4)
x = sp.linalg.solve(A, b)
x
```

Out[3]:

```
array([1., 1., 1., 1.])
```

In [4]:

```
x_error = sp.linalg.solve(A + 0.0001 * np.eye(4), b)
x_error
```

Out[4]:

```
array([0.99990001, 0.99990001, 0.99990001, 0.99990001])
```

이렇게 연립방정식을 이루는 행렬의 조건수가 커지면 상수항 오차가 작은 경우에도 해의 오차가 커지게 된다. 오차가 없는 경우 해 x는 다음과 같다.

In [7]:

```
sp.linalg.solve(A, b)
```

Out[7]:

```
array([-4., 60., -180., 140.])
```

하지만 이 경우에는 계수행렬이나 상수벡터에 약간의 오차만 있어도 해가 전혀 다른 값을 가진다. 다음 코드는 계수행렬에 1/10000의 오차가 있을 때 해의 값이 전혀 달리되는 것을 보인다.

In [8]:

```
sp.linalg.solve(A + 0.0001 * np.eye(4), b)
```

Out[8]:

```
array([-0.58897672, 21.1225671, -85.75912499, 78.45650825])
```

- 결국, 회귀분석 시, 공분산행렬의 조건수가 크면 회귀분석을 사용한 예측값도 오차가 커진다.

조금의 오차(계수행렬A, 상수벡터에b)가 있어도

solution set(x)가 매우 크게 달라져 제대로된 회귀분석, 예측을 할 수 없다.

2) 회귀분석과 조건수

- 회귀분석에서 조건수가 커지는 경우 2가지 (6page)

1. 변수들의 단위 차이(m과 mm) -> 스케일이 크게 달라짐 -> 조건수 커짐 <=> 스케일링(평균, 표편일정하게)

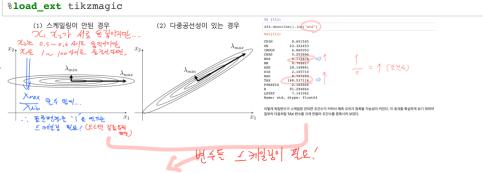
2. 다중공선성 -> 조건수 커짐 -> 변수 선택, PCA를 통한 차원축소(고유값 큰것들 위주로 남기기)

회귀분석에서 조건수가 커지는 경우는 크게 두 가지가 있다.

- 1번수들의 단위 차이로 인해 소수의 스케일이 크게 달라지는 경우. 이 경우에는 스케일링(scaling)으로 해결한다.
2. 다중공선성 즉, 상관관계가 큰 독립 변수들이 있는 경우. 이 경우에는 변수 선택이나 PCA를 사용한 차원 축소 등으로 해결한다.

이를 독립변수의 분포모양으로 설명하면 다음 그림과 같다.

In [9]:



[스케일링 전 미5결과]

```
%load_ext tikzmagic
%load_ext autoreload
%autoreload 2
feature_names = list(feature_names)
feature_names.remove('Intercept')
feature_names.append('Intercept')
feature_names = [f'feat{i}' for i in range(len(feature_names))]
feature_names[-1] = 'Intercept'
model1 = sm.OLS(y, sm.add_constant(X[feature_names]))
result1 = model1.fit()
print(result1.summary())
```

OLS Regression Results

	Intercept	X1	X2	X3	X4
Dep. Variable:	Intercept	0.741	0.734	0.734	0.734
Model:	OLS	R-squared:	0.742	0.742	0.742
Method:	Least Squares	F-statistic:	16128.34	16128.34	16128.34
Date:	Mon, 30 Oct 2019	P-value:	0.000	0.000	0.000
Time:	12:12:35	Log-Likelihood:	-16128.34	-16128.34	-16128.34
No. Observations:	598	AIC:	32256.68	32256.68	32256.68
Df Residuals:	592	BIC:	32259.68	32259.68	32259.68
Df Model:	6				
Covariance Type:	nonrobust				
	std err	t	P>t	t	P>t
Intercept	0.741	0.734	0.734	0.734	0.734
X1	0.742	0.742	0.742	0.742	0.742
X2	0.742	0.742	0.742	0.742	0.742
X3	0.742	0.742	0.742	0.742	0.742
X4	0.742	0.742	0.742	0.742	0.742

. 코딩 연습 노트북/0514_practice_math.ipynb?download=false

localh

[Warning] Standard errors assume that the covariance matrix of the errors is correctly specified.

[Warning] The covariance matrix of the errors appears to be very large. This might indicate that there are strong multicollinearity or other numerical problems.

In []:

04.04 범주형 독립변수

- OLS모형으로 회귀분석 실시 -> 조건수 큰 경우를 해결하기 위한 방안 1)스케일링 -> 2)범주형 독립변수 다루기

- 범주형 독립변수를 갖는 경우의 회귀모형 -> 더미변수화! (원핫인코딩)

ex) 혈액형 4개인 경우, $(1, 0)$ 이 아니라, $(1, 0, 0, 0)$, $(0, 1, 0, 0)$ 이렇게 가져야 함!
why? 분석 결과 해석을 하려면 특징 갯수만큼 원소를 갖는 카테고리 확률변수로 표시해줘야 함

- 더미변수화 : 풀랭크 방식(상수항 x), 축소랭크 방식(상수항 $0 = \text{기준값 존재}$)

1) 풀랭크 방식

- 기준 없이, 각 독립변수의 주체적인 변화를 분석하고 싶다면 풀랭크!
- 더미변수의 가중치는 각각 상수항이 됨 = y 절편 (1-3page)

풀랭크(full-rank) 방식에서는 더미변수의 값은 원핫인코딩(one-hot-encoding) 방식으로 지정한다. 즉 범주값이 2개인 경우에는

$$\begin{aligned} x_1 = A &\rightarrow d_{1A} = 1, d_{1B} = 0 \\ x_1 = B &\rightarrow d_{1A} = 0, d_{1B} = 1 \end{aligned}$$

이 된다. 이 값을 대입하면 더미변수의 가중치는 상수항이 된다.

$$\begin{aligned} x_1 = A &\rightarrow \hat{y} = w_{1A} + w_2x_2 + \dots + w_Dx_D \\ x_1 = B &\rightarrow \hat{y} = w_{1B} + w_2x_2 + \dots + w_Dx_D \end{aligned}$$

위 식은 $x_1 = A$ 인 데이터에 대해서는 $\hat{y} = w_{1A} + w_2x_2 + \dots + w_Dx_D$ 모형을 사용하고 $x_1 = B$ 인 데이터에 대해서는 $\hat{y} = w_{1B} + w_2x_2 + \dots + w_Dx_D$ 모형을 사용하게 된다는 뜻이다. 이렇게 하면 범주값이 달라졌을 때 상수항만 달라지고 다른 독립변수의 가중치(영향)은 같은 모양이 된다.

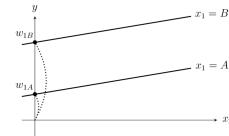


그림: 풀랭크 방식 더미변수 가중치의 의미

선행회귀모형에 범주형 독립변수가 있으면 더미변수의 가중치 이외에 별도의 상수항이 있으면 안된다. 만약 위의 모형에서 별도의 상수항 w_0 이 존재한다면 모형은 다음처럼 될 것이다.

$$\begin{aligned} x_1 = A &\rightarrow \hat{y} = (w_0 + w_{1A}) + w_2x_2 + \dots + w_Dx_D \Rightarrow w_0 \text{와 } w_{1A} \text{를 구분} \\ x_1 = B &\rightarrow \hat{y} = (w_0 + w_{1B}) + w_2x_2 + \dots + w_Dx_D \end{aligned}$$

이 경우에는 $w_0 + w_{1A}$ 나 $w_0 + w_{1B}$ 의 값은 구할 수 있어도 w_0 값과 w_{1A} 값을 분리할 수는 없다. 범주형 독립변수가 있으면 상수항은 포함시키지 않는다.

2) 풀랭크 방식 OLS 예시 (5, 7, 8, 9)

	time	value	month	범주형 독립변수
235	1939.583333	61.8	08	
236	1939.666667	58.2	09	
237	1939.750000	46.7	10	
238	1939.833333	46.6	11	
239	1939.916667	37.8	12	

더미변수의 값을 대입하면 다음과 같다.

$$\begin{aligned} x = 1 &\rightarrow d = (1, 0, 0, 0, \dots, 0) \rightarrow \hat{y} = w_1 \\ x = 2 &\rightarrow d = (0, 1, 0, 0, \dots, 0) \rightarrow \hat{y} = w_2 \\ x = 3 &\rightarrow d = (0, 0, 1, 0, \dots, 0) \rightarrow \hat{y} = w_3 \\ &\vdots \\ x = 12 &\rightarrow d = (0, 0, 0, 0, \dots, 1) \rightarrow \hat{y} = w_{12} \end{aligned}$$

월과 기준의 관계를 백스플롯으로 시각화하면 다음과 같다. 따라서 w_i 는 i 월의 기준의 표본평균값으로 계산된다.

In [6]:
model = sm.OLS.from_formula("value ~ C(month) + 0", df_nottem)
result = model.fit()
print(result.summary())

Month
장수 데이터로 표시되어 있기 때문에,
별도로 가중치를 모델에 일괄하여 주거나
여기서!

C(month)[01]	39.6950	0.518
40.715		
C(month)[02]	39.1900	0.518
40.210		
C(month)[03]	42.1950	0.518
43.215		
C(month)[04]	46.2900	0.518
47.310		

가장
장수
데이터로 표시되어 있기 때문에,
별도로 가중치를 모델에 일괄하여 주거나
여기서!

3) 축소랭크 방식

- 기준을 두고, 그 기준 대비 각 변수들의 변화를 분석하고 싶다면 축소랭크!

- 예) $H_0 : 1\text{월기온} = 2\text{월기온} \Leftrightarrow H_0 : w_2 = 0$ (기준 1월, 축소랭크 OLS)

- 더미변수의 가중치 = 기준값의 가중치 + 추가적으로 더해지는 가중치

축소랭크(reduced-rank) 방식에서는 특정한 하나의 범주값을 기준값(reference, baseline)으로 하고 기준값에 대응하는 더미변수의 가중치는 항상 1으로 놓는다. 다른 범주형 값을 가진다는 경우는 기준값에 추가적인 능성이 있는 것으로 기준이다.

예를 들어 다음 축소랭크 방식은 $x_1 = A$ 를 기준값으로 하는 경우이다.

$$x_1 = A \rightarrow d_{1A} = 1, d_{1B} = 0$$

$$x_1 = B \rightarrow d_{1A} = 0, d_{1B} = 1$$

반대로 $x_1 = B$ 를 기준값으로 하면 다음과 같아졌다.

$$x_1 = A \rightarrow d_{1A} = 1, d_{1B} = 0$$

$$x_1 = B \rightarrow d_{1A} = 0, d_{1B} = 1$$

이 값을 대입하면 기준값인 더미변수의 가중치는 상수항이 되고 나머지 더미변수의 가중치는 그 상수항에 추가적으로 더해지는 상수항이 된다. $x_1 = A$ 를 기준값으로 하는 경우와 같다.

$$x_1 = A \rightarrow \hat{y} = w_{1A} + w_2x_2 + \dots + w_Dx_D$$

$$x_1 = B \rightarrow \hat{y} = w_{1B} + w_2x_2 + \dots + w_Dx_D$$

그림: 축소랭크 방식 더미변수 가중치의 의미



4) 축소랭크 방식 OLS 예시 (10, 11, 12)

1월을 기준월로 하는 축소랭크 방식을 사용하면 더미변수는 다음과 같다.

$$x = 1 \rightarrow d = (1, 0, 0, 0, \dots, 0) \rightarrow \hat{y} = w_1$$

$$x = 2 \rightarrow d = (1, 1, 0, 0, \dots, 0) \rightarrow \hat{y} = w_1 + w_2 \quad \text{※ } w_2 = \text{기준값인 } 1\text{월기온}(w_1)$$

$$x = 3 \rightarrow d = (1, 0, 1, 0, \dots, 0) \rightarrow \hat{y} = w_1 + w_3 \quad \text{※ } w_3 = \text{기준값인 } 1\text{월기온}(w_1)$$

$$\vdots$$

$$x = 12 \rightarrow d = (0, 1, 0, 0, \dots, 1) \rightarrow \hat{y} = w_{12}$$

1월은 기준월이므로 더미변수는 다음과 같다.

$$x_1 = A \rightarrow d_{1A} = 1, d_{1B} = 0$$

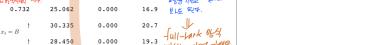
$$x_1 = B \rightarrow d_{1A} = 0, d_{1B} = 1$$

기준값 대비로 하는 더미변수는 다음과 같다.

$$x_1 = A \rightarrow \hat{y} = w_{1A} + w_2x_2 + \dots + w_Dx_D$$

$$x_1 = B \rightarrow \hat{y} = w_{1B} + w_2x_2 + \dots + w_Dx_D$$

그림: 축소랭크 방식 더미변수 가중치의 의미



기준을 찾는 축소랭크 방식

(기준을 찾았을 때 각 독립변수는 통제)

기준 X, 각 독립변수의 변화는 통제되는

= 통제되는

model = sm.OLS.from_formula("value ~ C(month)", df_nottem)

result = model.fit()

print(result.summary())

1월은 기준월이므로 더미변수는 다음과 같다.

$$x_1 = A \rightarrow d_{1A} = 1, d_{1B} = 0$$

$$x_1 = B \rightarrow d_{1A} = 0, d_{1B} = 1$$

기준값 대비로 하는 더미변수는 다음과 같다.

$$x_1 = A \rightarrow \hat{y} = w_{1A} + w_2x_2 + \dots + w_Dx_D$$

$$x_1 = B \rightarrow \hat{y} = w_{1B} + w_2x_2 + \dots + w_Dx_D$$

그림: 축소랭크 방식 더미변수 가중치의 의미

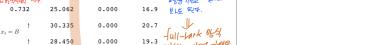


그림: 축소랭크 방식 더미변수 가중치의 의미

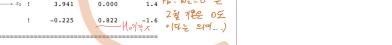


그림: 축소랭크 방식 더미변수 가중치의 의미

그림: 축소랭크 방식 더미변수

보스턴 집값 데이터는 CHAS라는 범주형 변수가 있고 이 변수는 0과 1 두 개의 값을 가진다.

만약 보스턴 집값 데이터에서 상수값 가중치를 가지는 모형을 만들면 축소 랭크 방식으로 더미변수 변환되어 있는 것과 같다. 즉 다음과 같은 두 개의 모형을 각각 회귀분석하는 경우라고 볼 수 있다.

- CHAS = 1 인 경우,

$$y = (w_0 + w_{\text{CHAS}}) + w_{\text{CRIM}} \text{CRIM} + w_{\text{ZN}} \text{ZN} + \dots$$

- CHAS = 0 인 경우,

$$y = w_0 + w_{\text{CRIM}} \text{CRIM} + w_{\text{ZN}} \text{ZN} + \dots$$

5) 보스턴 집값 데이터의 범주형 변수 (13page)

```
from sklearn.datasets import load_boston
boston = load_boston()
dfx = pd.DataFrame(boston.data, columns=boston.feature_names)
dfy = pd.DataFrame(boston.target, columns=['MEDV'])
df_boston = pd.concat([dfx, dfy], axis=1)

model1 = sm.OLS.from_formula("MEDV ~ " + "+ ".join(boston.feature_names), data=df)
model1 = model1.fit()
print(model1.summary())
 $\Rightarrow \text{CHAS 가 } \text{집값에 } \text{영향을 } \text{미친다!}$   

 $\Rightarrow 0 < 1$ 
```

	coef	std err	t	P> t	{ 0.025 }
Intercept	36.4595	5.103	7.144	0.000	26.432
CRIM	-0.1080	0.033	-3.287	0.001	-0.173
ZN	0.0464	0.014	3.382	0.001	0.019
INDUS	0.0206	0.061	0.334	0.738	-0.100
NOX	2.6867	0.862	3.118	0.002	0.994
RM	-17.7666	0.829	-21.651	0.000	-25.272
TX	-10.262	0.418	9.116	0.000	2.989
DIS	3.8099	0.418	9.116	0.000	2.989
AGE	4.631				

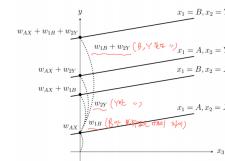
6) 두 개 이상의 범주형 변수가 있는 경우 (19page)

- 범주형 변수가 2개 이상이면 통합축소형 or 상호작용방식 (A, B / X, Y / 등..)

두 개 이상의 범주형 변수가 있는 경우에는 통합축소형 방식을 사용한다. 이 때 주의할 점은 모든 범주형 변수의 가중치는 기준값 상수항에 더해지는 상수항으로 처리된다. 예를 들어 x_1 은 A, B 의 두 가지 값을 가지고 x_2 는 X, Y 의 두 가지 값을 가지고 있을 때는 경우 상수항과 각 더미변수의 가중치의 합이 0이 되어야 한다.

$$\hat{y} = w_0 + w_{1A}x_1 + w_{1B}d_{1B} + w_{2X}x_2 + \dots + w_Dx_D$$

- w_{1A} : 기준값 $x_1 = A, x_2 = X$ 인 경우의 상수항
- w_{1B} : 기준값 $x_1 = B, x_2 = X$ 인 경우에 추가되는 상수항
- w_{2X} : 기준값 $x_1 = A, x_2 = Y$ 인 경우에 추가되는 상수항
- w_{2Y} : 기준값 $x_1 = B, x_2 = Y$ 인 경우에 추가되는 상수항



7) 범주형 독립변수와 실수 독립변수의 상호작용

- 실수독립변수에 영향을 주는 범주형 독립변수가 있다면, 상호작용항 생성해줘야함 (21page)

범주형 독립변수와 실수 독립변수의 상호작용

범주형 변수는 따라 다른 독립변수의 영향도(기여도)가 달라져서 상호작용을 한다.

만약 범주형 변수의 값이 달라질 때 상수항만 달라지는 것이 아니라 다른 독립변수들이 미치는 영향도 달라지는 모형을 원한다면 상호작용(interaction)을 쓰면 된다. 예를 들어 범주형 독립변수 x_1 과 실수 독립변수 x_2 를 가지는 회귀모형에서 연속값 독립변수 x_2 가 미치는 영향 즉 가중치가 범주형 독립변수 x_1 의 값에 따라 달라진다면 범주형 독립변수를 더미변수 d_1 으로 오클딩하고 연속값 독립변수 x_2 는 d_1 과의 상호작용 항 $d_1 \cdot x_2$ 를 추가하여 사용해야 한다.

이 모형은 다음과 같아진다.

$$\begin{aligned} \hat{y} &= w_0 + w_1x_1 \cdot w_2x_2 \\ &= w_0 + (w_{1A}d_A + w_{1B}d_B) \cdot (w_2x_2) \\ &= w_0 + w_{2A}d_Ax_2 + w_{2B}d_Bx_2 \end{aligned}$$

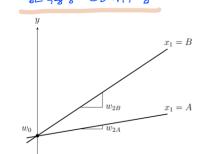
$$x_1 = A \text{ 일 때는 } d_A = 1, d_B = 0 \text{에서}$$

$$\hat{y} = w_0 + w_{2A}x_2$$

$x_1 = B \text{ 일 때는 } d_A = 0, d_B = 1 \text{에서}$

$$\hat{y} = w_0 + w_{2B}x_2$$

이므로 x_1 범주값에 따라 x_2 의 기울기가 달라지는 모형이 된다.



In []:

04.05 부분회귀

- Q : 기존의 모델에 새로운 독립변수를 추가하면, 가중치의 값이 달라질까? 그렇다.

con) 종속변수에 영향을 미치는 모든 독립변수를 회귀모형에 포함하지 않는 한 모형의 가중치는 항상 편향된 값이다.

con)

1. 처음에는, 독립변수를 최대한 많이 넣고 가중치 학습시키는 것이 좋음

(새로운 변수 추가 시, 가중치 값이 달라짐)

2. 순수한 상관관계를 확인하고 싶다면, 부분회귀로 각 변수의 순수한 스캐터플롯을 그려야 함

(그냥 스캐터플롯으로 다른 변수들의 영향이 복합되어있어 정확한 상관관계를 확인하기 어렵다.)

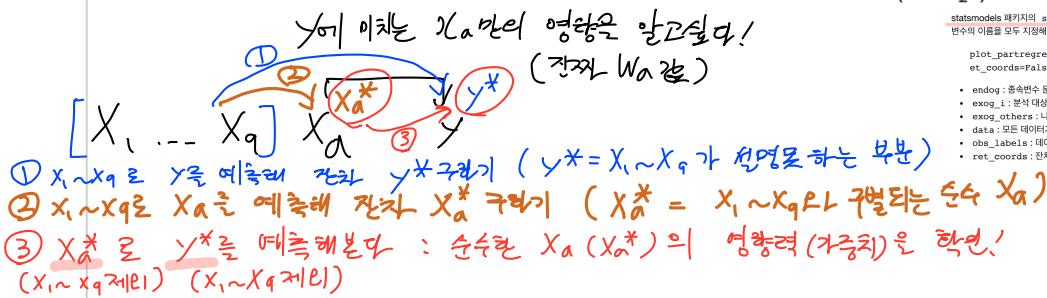
1) 새로운 독립변수가 추가된다면, 가중치는?

변한다. 안변하는 경우는 1) 새 변수와 종속변수가 독립일 때 or 2) 새 변수와 기존 변수들 간 독립일 때

증명해보기, 프리슈-워-로벨 정리

2) 부분회귀 플롯

- 각 변수들의 종속변수와의 순수한 상관관계를 볼 수 있음 (3-4p)



3) CCPR 플롯

- 부분회귀 플롯과 비슷한 역할. 하지만, 조금 정석적인 방법은 아님

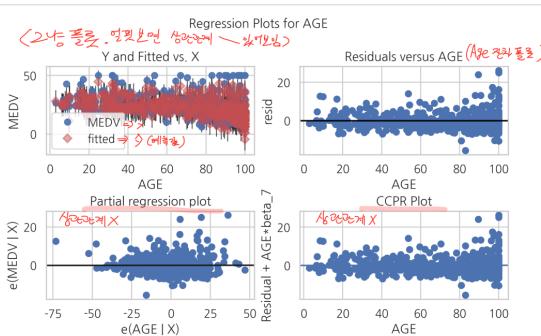
4) plot_regress_exog : 부분회귀 플롯, CCPR 플롯 함께 보여줌 (12page)

plot_regress_exog 명령은 부분회귀 플롯과 CCPR을 같이 보여준다.

```
plot_regress_exog(result, exog_idx)
• result: 회귀분석 결과 객체
• exog_idx: 분석 대상이 되는 독립변수 문자열
```

In [7]:

```
fig = sm.graphics.plot_regress_exog(result_boston, "AGE")
plt.tight_layout(pad=4, h_pad=0.5, w_pad=0.5)
plt.show()
```



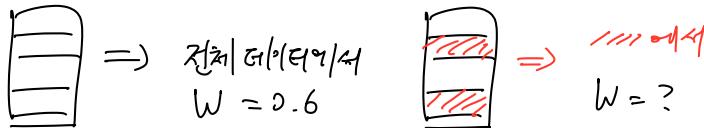
In []:

05.01 확률론적 선형 회귀모형

- 예측에는 오차가 있다. 그런데 그 오차가 얼마인지 모른다면? 아무것도 모르는 것과 같다.
- 오차가 어느정도 나는 건지 알아야 한다 => 확률론적 선형 회귀모형

1) 방법 1 : 부트스트래핑 (1page 상단)

- 표본 데이터가 달라질 때, 회귀분석의 결과는 어느정도 영향을 받는지를 알기 위한 방법
- 기존의 데이터를 re-sampling(재표본화) -> 회귀분석 재실시 (w주정) (기존의 N개 데이터에서 N개 데이터 선택하되, 중복 선택 가능하게 함)
- 이 방법은 연산소요시간이 너무 오래걸림 (1000번 리샘플링 - 회귀분석 반복!)



2) 방법 2 : 확률론적 선형회귀모형 (부트스트래핑 대체 방법)

- 사실 OLS 모델 자체는 확률과 상관 없는 모델 ==> 여기에 확률론을 접목시킨 것
- 데이터 = 확률변수에서 생성된 표본이라 가정
- 4가지 가정을 세팅한 것이 기본모형 => 확률론적 선형회귀모형
 - 1) 선형 정규분포 가정
 - 2) 외생성 가정 (exogeneity)
 - 3) 조건부 독립 가정
 - 4) 등분산성 가정

1) 선형 정규분포 가정 : 종속변수 y 는 정규분포를 따르고, 기대값(모수)은 독립변수 x 의 선형조합으로 결정됨 (5p)

\Leftrightarrow 잡음(y -기대값)은 0주변에서 분포한다.

\Leftrightarrow 잡음이 정규분포일 뿐이지, x 나 y 주변확률분포 자체가 정규분포일 필요 없음

선형 회귀분석의 기본 가정은 종속 변수 y 와 독립 변수 x 의 선형 조합으로 결정되는 기댓값과 고정된 분산 σ^2 을 가지는 가우

사인 점과 분포라는 것이다.

$y \sim N(\mu, \sigma^2)$

y 의 확률 밀도 함수는 다음과처럼 될 수 있다. 이 확률 밀도 평균 $\mu = E(y | w(x, \sigma^2))$,

$p(y | x, \theta) = p(y | w(x, \sigma^2))$

이 관계식을 잔음(stubrance)이라고 부르면 더 간단하게 표현할 수 있다.

$y = \mu + \epsilon$ ($\epsilon \sim N(0, \sigma^2)$)

$p(\epsilon | \theta) = N(0, \sigma^2)$

여기에서 주의할 점은

x, y 중 그 어느 것도 그 자체로 정규 분포일 필요는 없다.

는 것이다.

y 도 x 에 대해 조건부로 정규 분포를 이루는 것이지 y 자체가 무조건부로 정규분포는 아니다.

2) 외생성 가정 : 잡음의 기대값은 x 와 상관없이 0이다. (5p)

\Leftrightarrow 따라서, 잡음의 무조건부 기댓값 = 0, 잡음과 독립변수 x 는 독립 임을 증명할 수 있음

잡음 ϵ 의 기댓값은 독립 변수 x 의 크기에 상관없이 항상 0이라고 가정한다. 이를 외생성(Exogeneity) 가정이라고 한다.

$$E[\epsilon | x] = 0$$

3) 조건부 독립 가정 : i 번째 데이터에 대한 잡음, j 번째 데이터가 갖는 잡음은 서로 독립(공분산 = 0) (5p)

$\Leftrightarrow E[\text{잡음}_i * \text{잡음}_j] = 0$

\Leftrightarrow 잡음벡터의 공분산행렬 = 대각행렬(비대각성분 = 0) (왜냐면 $\text{cov}(\text{잡음}_i, \text{잡음}_j) = 0$)

i 번째 표본의 잡음 ϵ_i 와 j 번째 표본의 잡음 ϵ_j 의 공분산 값이 x 와 상관없이 항상 0이라고 가정한다.

$$\text{Cov}[\epsilon_i, \epsilon_j | x] = 0 \quad (i, j = 1, 2, \dots, N)$$

4) 등분산성 가정 : 데이터에 관계없이 잡음의 분산 값은 일정하다 (5p)

\Leftrightarrow 3) 조건과 함께, 결국 잡음벡터의 공분산 행렬은 항등행렬(대각성분 = 분산, 모두 같은 값)

i 번째 표본의 잡음 ϵ_i 와 j 번째 표본의 잡음 ϵ_j 의 분산 값이 표본과 상관없이 항상 같다고 가정한다.

3) 확률론적 선형회귀 모형 : 최대가능도 방법을 사용한 선형 회귀분석(OIS 방법처럼 찾을 수 없음)

- but, 결과는 OIS와 확률론적 모형(MLE)은 같다.

- 차이점 : MLE는 정답(w의 정답)이 존재한다는 가정 하, 정답과 가장 유사한 것을 찾아보자라는 접근(하지만 정답 근처의 다른 답이 나옴)
- OLS는 정답 없이 그냥 RSS를 최소화 하는 값 찾은 것

4) 잔차의 분포

- 확률론적 선형회귀모형 "잔차 = $e = y - \hat{w}^T x$ 도 정규분포따름"
- 확률론적 선형회귀모형에서는 잔차와 잡음이 다른 개념이다. (7page 상단) 잡음 ε (disturbance) 잔차 e (residual)

< 확률론적 선형회귀에서는 ε, e 다르다 >

$\tilde{W}^T x + \varepsilon \leftarrow$ 실제 실제!

1. $\tilde{W} \neq \hat{W}$ (정답은 있지만 예측값과 다르다)

2. $\varepsilon \neq e$ (잡음은 예측 기반)

$e = y - \hat{y}$
↳ 예측 기반
 $\varepsilon \Rightarrow \tilde{W}^T x + \varepsilon$
↳ 실제기반!

