Sample

Convenience : 가용한 표본 표집, 편향이 일어나기 쉬움

Quota : 표본의 성질 특정해서 표집

Probability Sample

SRS : 불가능한 경우가 없이 모든 확률이 동등한 표집

- 가능한 경우의 수 : ${}_NC_n = \dfrac{N!}{n!(N-n)!}$

- 특정한 원소가 포함되었을 확률 : $\dfrac{{}_{N-1}C_{n-1}}{{}_NC_n}$

Cluster : 군집 중 하나 또는 여럿을 표집

Stratified : 군집 내에서 SRS 표집

Bias

Selection : 특정 집단을 편애, 배제하여 생김

Response : 응답이 정직하지 않아서 생김

Non-response : 응답하지 않아서 생김

Experiment

RCT : 실험집단, 통제집단을 랜덤하게 나누어 처치

Observational : 필요한 집단 모집, 관찰

A/B Testing : 두 표본이 같은 모집단에 속한지 확인하는 방법

Pandas

Domain : 데이터의 타입

`df[0:1]` : 행 추출

`df[[True, False]]` : 특정 행 추출

시리즈에 대한 논리 연산은 부울 시리즈를 반환, 논리 연산자도 가용

- `df["col"]=="spam" ¦ df["col"]=="egg“`

`s.isin(["spam", "egg"])` : 리스트 안의 원소 중 하나인지의 부울 시리즈 반환

`df.query("col=='spam' or col=='egg'")` : 문자열 쿼리를 해석

`df.loc[[0, 1, 2, 3], ["col0", "col1", "col2"]]` : 데이터프레임의 해당 행과 열 반환

`df.loc[0:3, "col0":"col2"]` : Inclusive Slicing으로 접근 가능

`df.loc[0, "colname"]` : 열과 행 레이블로도 접근 가능

`df.loc[[True, False], [True, True]]` : 부울 이터러블도 가용

`df.loc[[0,1,2,3,4]]` : 인덱스와 완전히 일치한다면 인덱스가 해당 순서가 되도록 정렬한 데이터프레임 반환

`df.sample(10)` : 숫자만큼의 데이터를 무작위로 표집

`df.size` : 데이터프레임의 관측치 수

`df.shape` : 데이터프레임의 행, 열 개수 튜플

`df.index` : 데이터프레임의 인덱스 이터러블

`df.columns` : 데이터프레임의 열 레이블 이터러블

`s.map(lambda x: -x)` : 시리즈에 함수 적용해서 시리즈로 반환

`df.drop("label", axis=0)` : 축이 0이면 행 삭제, 1이면 열 삭제

`group.filter(lambda x: True)` : 그룹에 대해 함수 실행 후 참이면 새 데이터프레임에 넣고 거짓이면 넣지 않은 후 반환

basic python	re	pandas
	re.findall	df.str.findall
str.replace	re.sub	df.str.replace
str.split	re.split	df.str.split
’ab’ in str	re.search	df.str.contain
len(str)		df.str.len
str[1:4]		df.str[1:4]

EDA 고려사항

1. Structure : Table, Matrix, ... / csv, tsv, json, xml, ...
2. Granularity : 각 레코드가 나타내는 대상
3. Temporality : 데이터가 모아진 시간, 시간과 위치의 관계, null time value
4. Faithfulness : 아웃라이어, null value

- null value를 드랍하면 편향 가능성, Mean, Hot Deck Imputation

변수형

- Quantitative : Continuous, Discrete
- Qualitative : Ordinal, Nominal

Distribution

모든 변수의 합은 1(100%), 모든 대상은 하나의 항목에만 속함

Mode(최빈값) : Unimodal, Bimodal, Multimodal

Skewed Right : 오른쪽 꼬리, 평균 > 중위수

Skewed Left : 왼쪽 꼬리, 평균 < 중위수

사분위수

First(Lower, Q1) : 하위 25% = 상위 75% = 25 퍼센타일

Third(Upper, Q3) : 하위 75% = 상위 25% = 75 퍼센타일

IQR = Q3 - Q1

Overplotting

- 값이 겹쳐서 구별 불가능한 경우
- Jitter로 구별 가능

히스토그램

- 분포를 나타내는 플롯, 전체 면적이 약 100%
- $Width = 2\dfrac{IQR(x)}{\sqrt[3]{n}}$

상자 플롯

- 상자의 선은 각 사분위수를 나타냄
- Whisker : 각 사분위수에서 1.5IQR만큼 벗어난 값
- 수염 바깥쪽 값은 아웃라이어가 됨

시각화 원칙

1. Scale : 축이 여럿 있으면 혼동의 가능성이 있음
2. Conditioning : 특정한 사항을 강조하기 위해 플롯을 선택

- Juxtaposition : 나란히 배열
- Superposition : 겹치게 배열

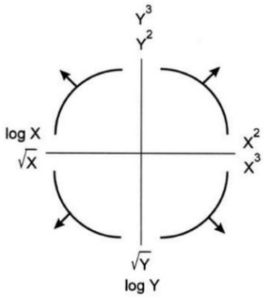
3. Perception : 델타 값이 일정하거나 연속적인 컬러맵을 사용

- Sequential Colormap : 낮은 값에서 높은 값으로의 진행
- Diverging Colormap : 0을 중심으로 낮고 높은 값이 위치

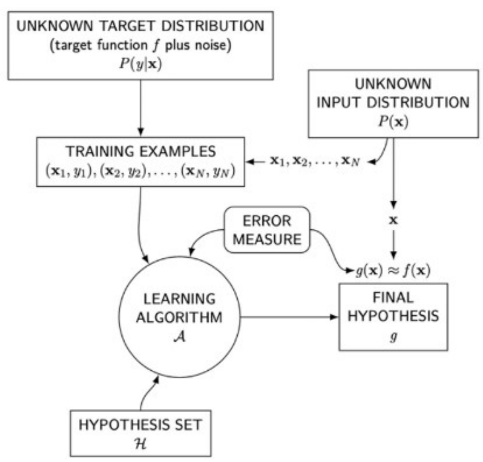
4. Marking : 기준선의 고정, 즉 Jiggling 회피
5. Context : 플롯을 설명하는 제목, 범례, 캡션 등
6. Smoothing

- KDE : 가우시안 커널에서 α는 낮을수록 좁은 커널이, 높을수록 넓은 커널이 만들어짐

7. Transformation



모델링 용어



Prediction : 모델을 사용해 출력을 결정(예측)
Estimation : 모델의 파라미터를 결정하기 위해 데이터를 사용

y : 실제 관측치

$\hat{y}$: 예측한 관측치

θ : 모델의 파라미터

$\hat{\theta}$: 최적의 파라미터

Target Distribution $f(x)$: 실제 분포 함수

Final Hypothesis $g(x)$: 가설함수

Training Examples : 학습을 위한 레이블된 데이터

Error Measure : 손실함수

Hypothesis Set H : 가능한 모든 가설의 집합

Learning Algorithm A : 가설함수를 결정하는 알고리즘

Input Space X : $\mathbb{R}^d$인 d차원 공간

Input Vector x : $x \in X$인 입력 벡터(독립변수, 공변량, 피쳐)

Output Space Y : 1이나 -1을 값으로 갖는 공간

Output y : 종속변수 값, $y_n = f(x_n) + \epsilon$

Constant Model

$\hat{y} = \theta$

Loss Function

Average Loss(Empirical Risk, Objective Function)

- $R(\theta) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y_i})$

MAE

L1 Loss : $|y - \hat{y}|$

$$\frac{d}{d\theta}|y_i - \theta| = \begin{cases} -1 & \theta < y_i \\ 1 & \theta > y_i \end{cases}$$

$$\frac{d}{d\theta}R(\theta) = \frac{1}{n}\left(\sum(-1) + \sum 1\right)$$

따라서 MAE의 $\hat{\theta}$는 중위수

부드럽지 않고 값을 최소화하기 어려움, 아웃라이어에 강함

$x > \theta$에서 n이 곱해지면 데이터의 수를 $n:1$로 나누는 분위가 $\hat{\theta}$가 됨 (ex: $n = 4$면 80 Percentile)

MSE

L2 Loss : $(y - \hat{y})^2$

$$\frac{d}{d\theta}R(\theta) = \frac{-2}{n}\sum_{i=1}^{n}(y_i - \theta)$$

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

따라서 MSE의 $\hat{\theta}$는 평균

부드럽고 값을 계산하기 쉬움, 아웃라이어에 약함

Huber Loss

일정 구간에는 MSE, 다른 구간에는 MAE 사용

$\hat{\theta}$는 중위수와 평균 사이의 값

퍼셉트론

$$h(x) = sign\left(\sum_{i=0}^{d} w_i x_i\right) = sign(w^\top x)$$

이때, $w_0 \equiv -threshold \equiv b$이고, $x_0 \equiv 1$(정확히는 초기값이 1)

PLA는 데이터가 선형적으로 나누어질 수 있다는 가정에서 출발

$h(x)$가 잘못 분류한 데이터에 대해 $w \leftarrow w + y_n x_n$(모든 데이터에 대해 옳게 될 때까지 반복)

선형회귀

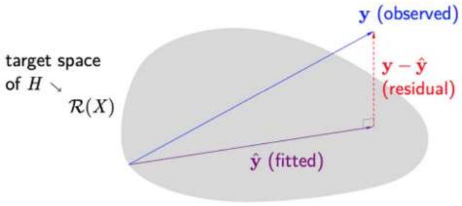
$$h(x) = \sum_{i=0}^{n}(w_i x_i) = w^\top x$$

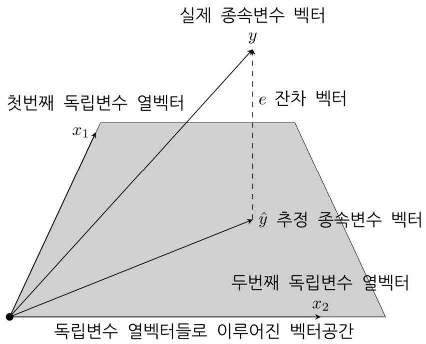
OLS

$$E_{\mathrm{in}}(\mathrm{h}) = \frac{1}{\mathrm{N}}\sum_{\mathrm{n}=1}^{\mathrm{N}}(\mathrm{h}(\mathrm{x_n}) - \mathrm{y_n})^2$$

Convex Function인 E_{in}의 최솟값은 $\nabla E_{\mathrm{in}} = 0$이 되는 값으로, 이때의 $w = (X^\top X)^{-1}X^\top y$(단, $X^\top X$가 역행렬을 지닐 때, 즉 N이 $d+1$보다 훨씬 클 때)

Hat Matrix





$\hat{y} = Hy$로 표현됨

$H = X(X^\top X)^{-1}X^\top$

$H = H^2 = H^\top$ 이고, $HX = X$

$M = I - H$역시 Hat Matrix이고, $HM = MX = 0$임

H의 영역은 x값의 Column Space의 영역이 되고, 다른 말로 $\hat{y}$는 y의 $R(X)$에 대한 정사영이 된다