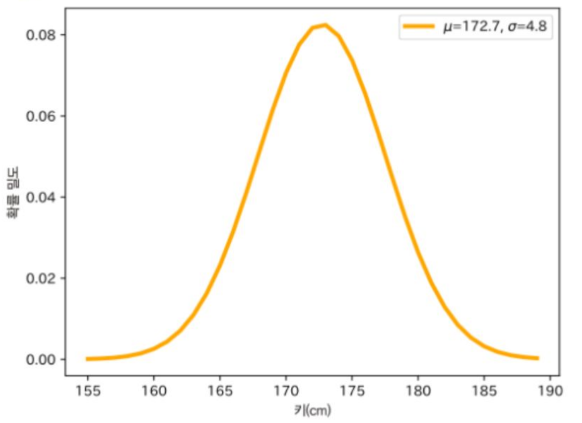
Chapter2. 최대 가능도 추정

2.1 생성 모델 개요

- 생성 모델의 목표
  - 특정 데이터 x의 확률 분포 p(x)를 모델링 → 그 집단에서 선택된 것 같은 유사 데이터를 새롭게 생성
    - 예. 그림2-1. 특정 데이터를 이용해 키 확률 분포를 정규 분포로 표현
    - **집단의 특성을 모델링하는 것이 목표**
→ 확률 변수의 매개변수 (예. 정규분포의 평균, 표준편차)를 추론

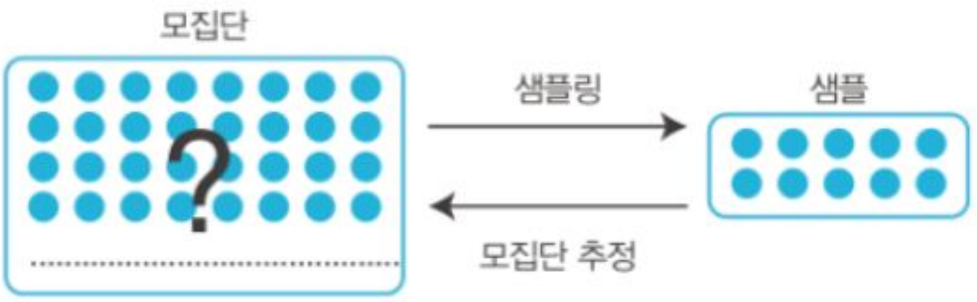
그림 2-1 정규 분포를 따르는 생성 모델 예



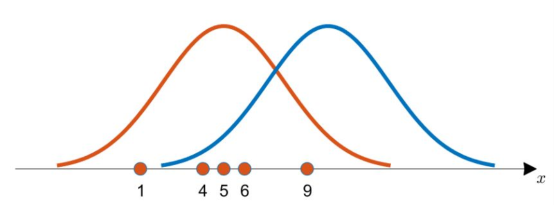
2.1 생성 모델 개요

- 모집단과 샘플
  - 일반적으로 모집단(대상의 전체 집합)의 규모는 방대함.
  - 통계학에서는 제한된 수의 **샘플**만으로 모집단의 특성을 추정

그림 2-2 모집단과 샘플의 관계

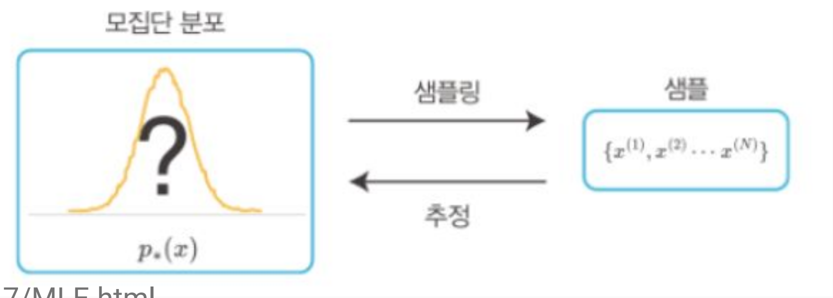


2.1 생성 모델 개요



- 모집단과 샘플
    - 생성 모델도 샘플을 이용하여 모집단을 추정
        - 모집단은 샘플을 뒷받침하는 확률 분포 → 모집단 분포
        - p*(x) 모집단을 현실적으로 알수 없으므로 모집단 분포를 추정
    - 모집단 분포 추정
        - 1. 모델링 : 모집단 분포를 '매개변수로 조정 가능한 확률 분포'로 비슷하게 표현한다고 **가정**
        - 2. 매개변수 추정 : 모델링 결과로 만들어진 확률 분포가 샘플 데이터에 부합하도록 추정 (여

그림 2-3 생성 모델에서 모집단 분포와 샘플의 관계



*https://angeloyeo.github.io/2020/07/17/MLE.html

2.1 생성 모델 개요

- (궁금점) 왜 정규분포로 모델링하지?
  - 중심극한정리
  - 계산의 용이성
  - 많은 종류의 데이터가 정규분포에 근사
    - 예. 자연현상 : 키, 몸무게, 혈압 등 / 제조공정 품질 : 부품치수, 중량 등 / 시험 점수 등
- 그럼 언제 다른 분포로 모델링하지?
  - 한쪽으로 치우친 데이터 분포가 명백하거나 정교한 분포 모델링이 필요한 경우
    - 예. 신뢰성 문제 등 (지수 분포, 포아송 분포 등)

2.2 실제 데이터로 생성 모델 구현

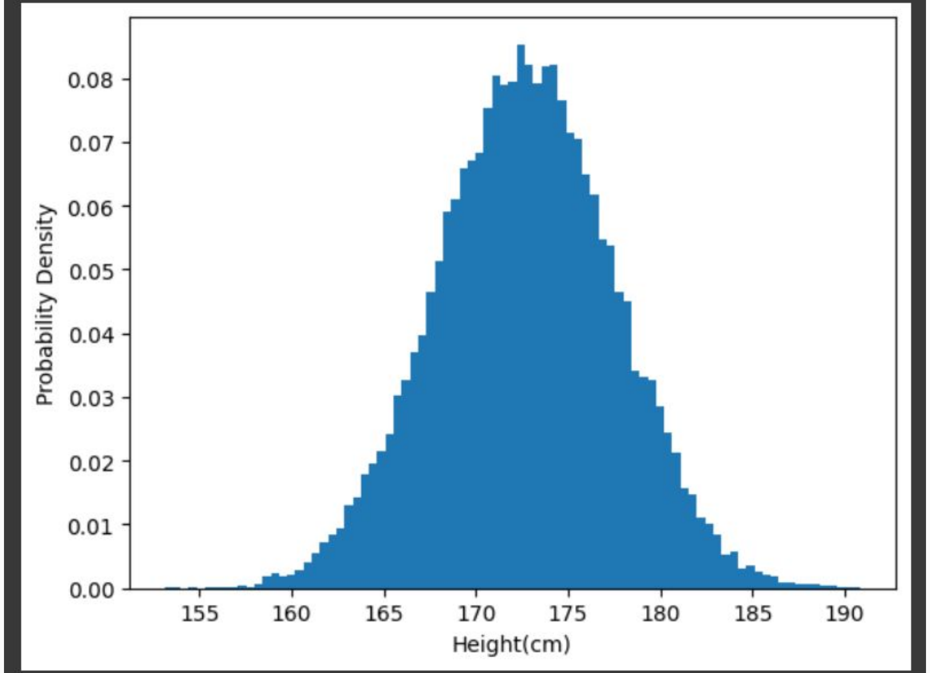
- 키 데이터셋 불러오기
    - 1993년 18세 홍콩인 키데이터(25,000개)
    - 정규 분포와 비슷함.

```python
import numpy as np
import matplotlib.pyplot as plt

path = 'height.txt'
xs = np.loadtxt(path)
print(xs.shape)

plt.hist(xs, bins='auto', density=True)
plt.xlabel('Height(cm)')
plt.ylabel('Probability Density')
plt.show()
```

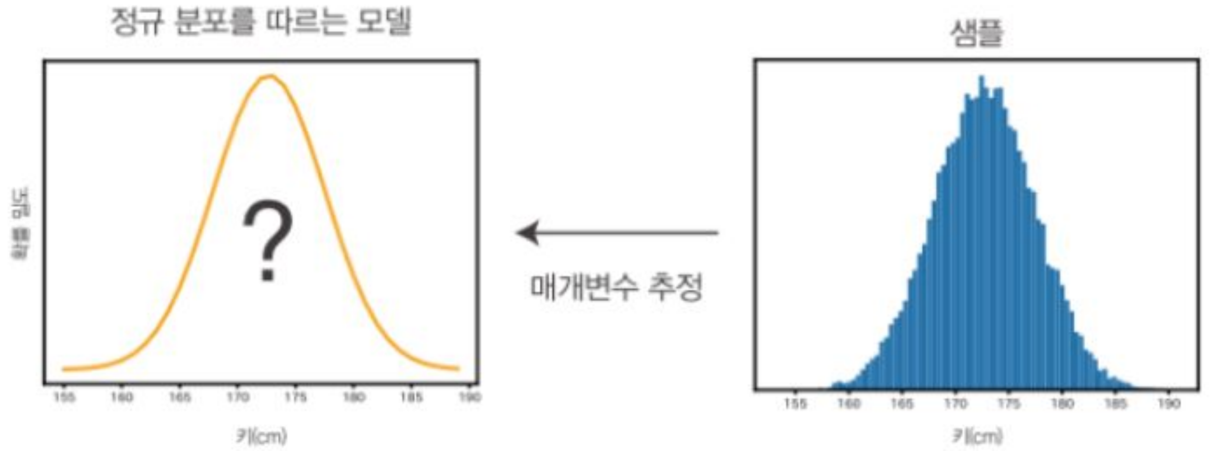
```
(25000,)
```



2.2 실제 데이터로 생성 모델 구현

- 정규 분포를 따르는 생성 모델
  - 모델링 : 키 데이터가 정규 분포라고 가정
  - 매개변수 추정 : 샘플을 기반으로 정규 분포의 매개변수를 추정

그림 2-5 정규 분포의 매개변수 추정



2.2 실제 데이터로 생성 모델 구현

- 정규 분포를 따르는 생성 모델
  - 25,000개 데이터의 평균, 표준편차로 정규 분포 그래프(노란색)를 그리면 데이터 분포(파란색)와 유사함 → 샘플로부터 정규 분포의 매개변수를 추정

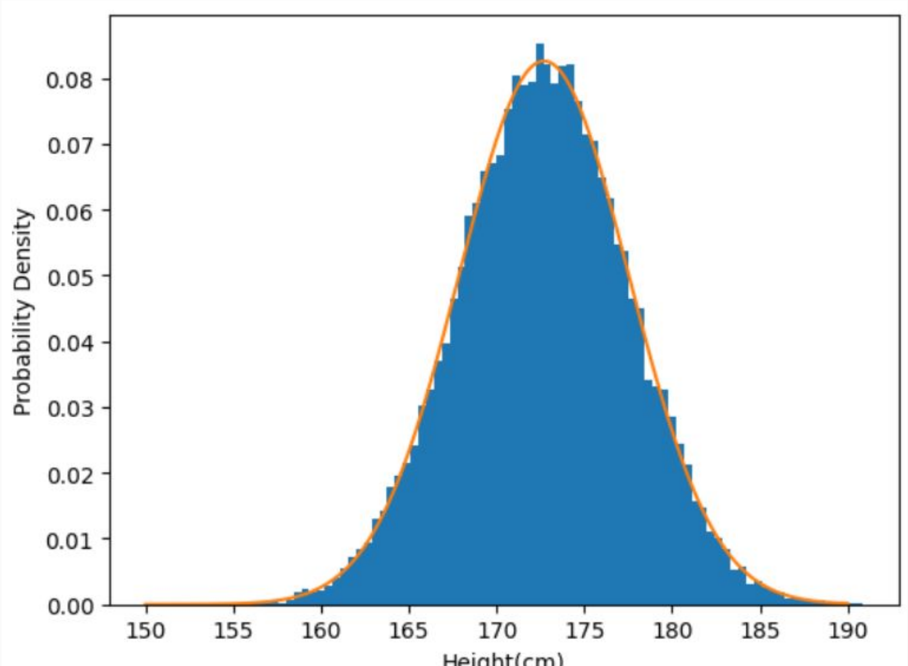
```python
import numpy as np
import matplotlib.pyplot as plt

path = 'height.txt'
xs = np.loadtxt(path)

mu = np.mean(xs)
sigma = np.std(xs)

# normal distribution
def normal(x, mu=0, sigma=1):
    y = 1 / (np.sqrt(2 * np.pi) * sigma) * np.exp(-(x - mu)**2 / (2 * sigma**2))
    return y
x = np.linspace(150, 190, 1000)
y = normal(x, mu, sigma)

# plot
plt.hist(xs, bins='auto', density=True)
plt.plot(x, y)
plt.xlabel('Height(cm)')
plt.ylabel('Probability Density')
plt.show()
```



2.2 실제 데이터로 생성 모델 구현

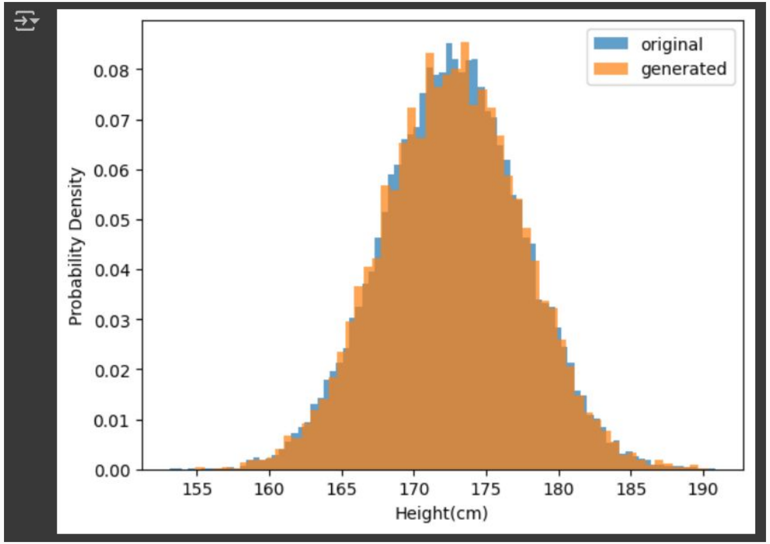
- (궁금점) 그럼 저 모델로 새로운 데이터를 만들수 있다는건가?
  - 원본 데이터로부터 평균, 표준편차를 추정하여 정규 분포 모델을 생성.
  - 추정된 정규 분포를 기반으로 새로운 데이터(samples)를 생성하여, 원본 데이터와 비교
    - (교재 2.4.1 절에 나오기는 함;)

```python
import os
import numpy as np


path = 'height.txt'
xs = np.loadtxt(path)
mu = np.mean(xs)
sigma = np.std(xs)

samples = np.random.normal(mu, sigma, 10000)

plt.hist(xs, bins='auto', density=True, alpha=0.7, label='original')
plt.hist(samples, bins='auto', density=True, alpha=0.7, label='generated')
plt.xlabel('Height(cm)')
plt.ylabel('Probability Density')
plt.legend()
plt.show()
```



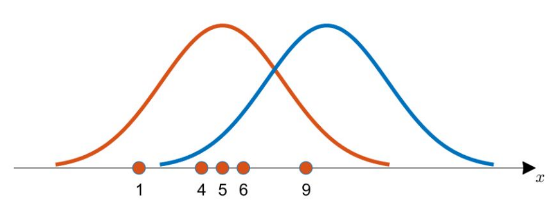
2.3 최대 가능도 추정 이론

- 가능도 최대화
  - N개의 관측 데이터를 얻었을 때, $\{x^{(1)}, x^{(2)} \dots x^{(N)}\}$ 다음 식에 따라 정규 분포의 매개변수를 추정
    - 아래 식은 추정된 값

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x^{(n)}$$

$$\hat{\sigma} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x^{(n)} - \hat{\mu})^2}$$

2.3 최대 가능도 추정 이론



- 가능도 최대화
  - 매개변수 θ 에 의해 모양이 결정되는 확률분포가 있다고 가정
  - 데이터 x를 얻을 수 있는 확률 밀도 $p(x;\theta)$.

아래 샘플을 가정. 각 데이터는 확률분포에 따라 독립적으로 생성된다고 가정.

$$\mathcal{D} = \{x^{(1)}, x^{(2)} \cdots x^{(N)}\}$$

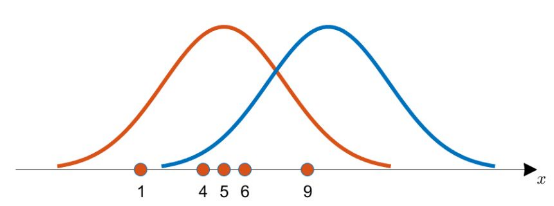
N개의 데이터를 얻을 수 있는 확률 밀도는 각 데이터의 확률 밀도의 곱 → 가능도

$$p(\mathcal{D};\theta) = p(x^{(1)};\theta)\ p(x^{(2)};\theta)\ \cdots\ p(x^{(N)};\theta) = \prod_{n=1}^{N} p(x^{(n)};\theta)$$

매개변수를 θ 로 설정했을 때 샘플D를 얻을 수 있는 확률 밀도

$$L(\theta) = p(\mathcal{D};\theta)$$

2.3 최대 가능도 추정 이론



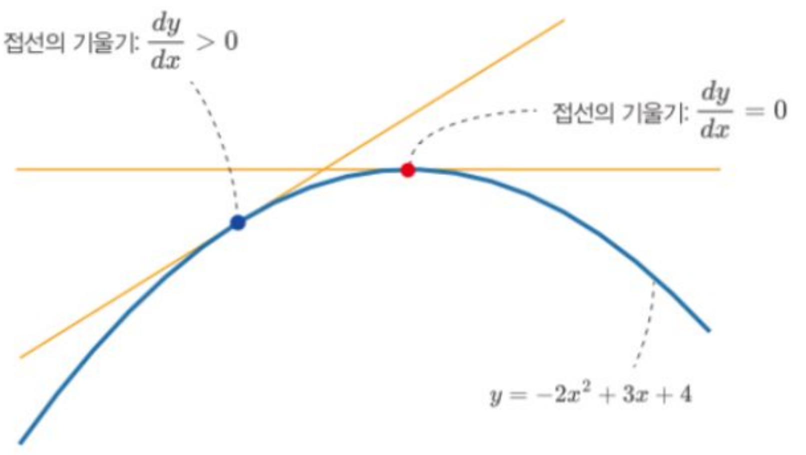
- 가능도 최대화
  - 최대 가능도 추정은 가능도 함수를 최대화하 θ는 를 찾는 기
  - 가능도를 최대화하는 매개변수 $\hat{\theta}$ 라고 하면, 이때 샘플이 관찰될 확률이 가장 높다는 뜻
  - 실제 가능도는 로그 가능도를 최대 $\log p(\mathcal{D}; \theta)$
    - 계산의 편리함 log ab = log a + log b

2.3 최대 가능도 추정 이론

- 미분을 사용하여 최댓값 찾기
  - 임의의 x에서 y가 변하는 비율
  - 접선의 기울기가 0일 때 y의 값이 최대값 (2차항 계수가 음수인 경우에만 해당)

  -> 로그 가능도의 매개변수에 대해 미분하고 0으로 설정해서 최대값을 찾자

그림 2-7 미분과 접선의 기울기



2.3 최대 가능도 추정 이론

- 정규 분포의 최대 가능도 추정

$$\mathcal{N}(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$p(\mathcal{D};\mu,\sigma) = \prod_{n=1}^{N}\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x^{(n)}-\mu)^2}{2\sigma^2}\right)$$

$$\begin{aligned}\log p(\mathcal{D};\mu,\sigma) &= \log\prod_{n=1}^{N}\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x^{(n)}-\mu)^2}{2\sigma^2}\right)\\ &= \log\prod_{n=1}^{N}\frac{1}{\sqrt{2\pi}\sigma} + \log\prod_{n=1}^{N}\exp\left(-\frac{(x^{(n)}-\mu)^2}{2\sigma^2}\right)\\ &= \log\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N + \sum_{n=1}^{N}\frac{-(x^{(n)}-\mu)^2}{2\sigma^2}\\ &= -\frac{N}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(x^{(n)}-\mu)^2\end{aligned}$$

$$\frac{\partial}{\partial\mu}(x-\mu)^2 = 2(x-\mu)\cdot\frac{\partial}{\partial\mu}(x-\mu)$$

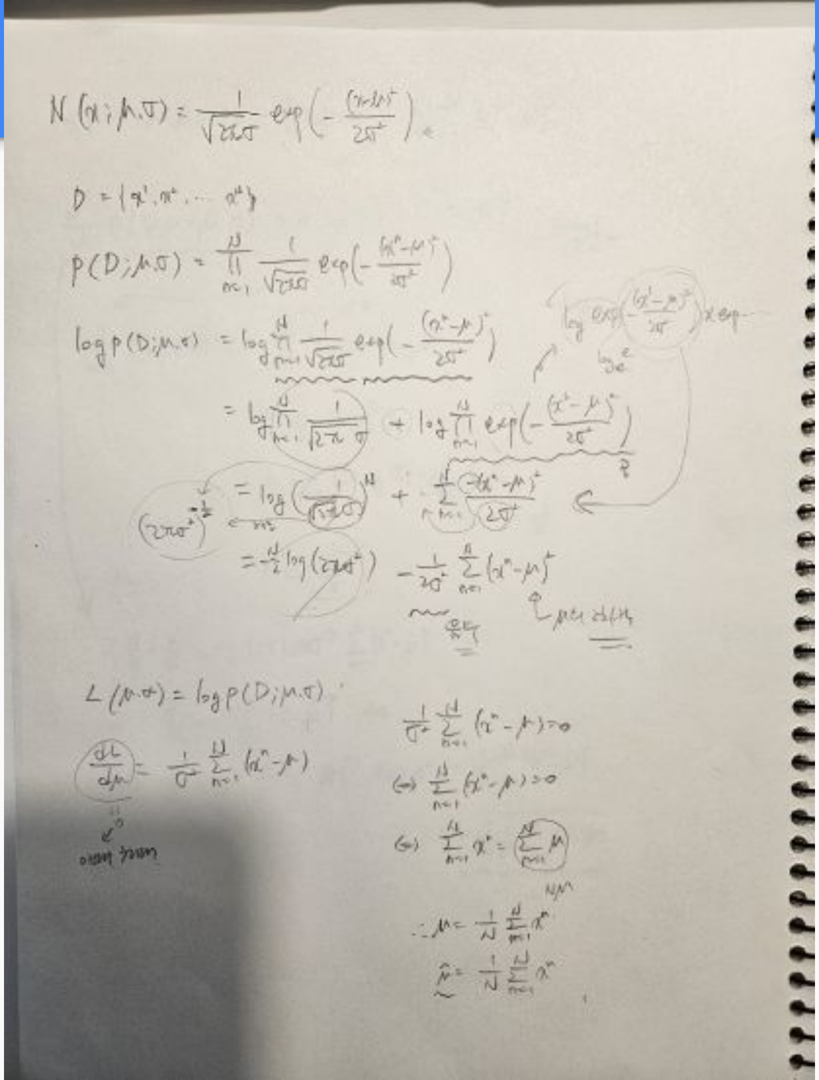
$$\frac{\partial L}{\partial\mu} = \frac{1}{\sigma^2}\sum_{n=1}^{N}(x^{(n)}-\mu)$$

$$\begin{aligned}&\frac{1}{\sigma^2}\sum_{n=1}^{N}(x^{(n)}-\mu) = 0\\ \Leftrightarrow\ &\sum_{n=1}^{N}(x^{(n)}-\mu) = 0\\ \Leftrightarrow\ &\sum_{n=1}^{N}\mu = \sum_{n=1}^{N}x^{(n)}\\ \Leftrightarrow\ &N\mu = \sum_{n=1}^{N}x^{(n)}\\ \therefore\ &\mu = \frac{1}{N}\sum_{n=1}^{N}x^{(n)}\end{aligned}$$

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N}x^{(n)}$$

2.3 최대 가능도 추정 이론

- 정규 분포의 최대 가능도 추정



2.3 최대 가능도 추정 이론

- 정규 분포의 최대 가능도 추정

$$\log p(\mathcal{D}; \mu = \hat{\mu}, \sigma) = -\frac{N}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(x^{(n)} - \hat{\mu})^2$$

$$\begin{aligned}
\frac{d}{d\sigma}(\log 2\pi\sigma^2) &= \frac{d}{d\sigma}(\log 2\pi + \log\sigma^2) && (\log ab = \log a + \log b)\\
&= \underbrace{\frac{d}{d\sigma}(\log 2\pi)}_{0} + \frac{d}{d\sigma}(\log\sigma^2) && (\text{상수의 미분은 } 0)\\
&= \frac{d}{d\sigma}(2\log\sigma) && (\log a^b = b\log a)\\
&= 2\frac{d}{d\sigma}\log\sigma && (\frac{d}{da}\log a = \frac{1}{a})\\
&= \frac{2}{\sigma}
\end{aligned}$$

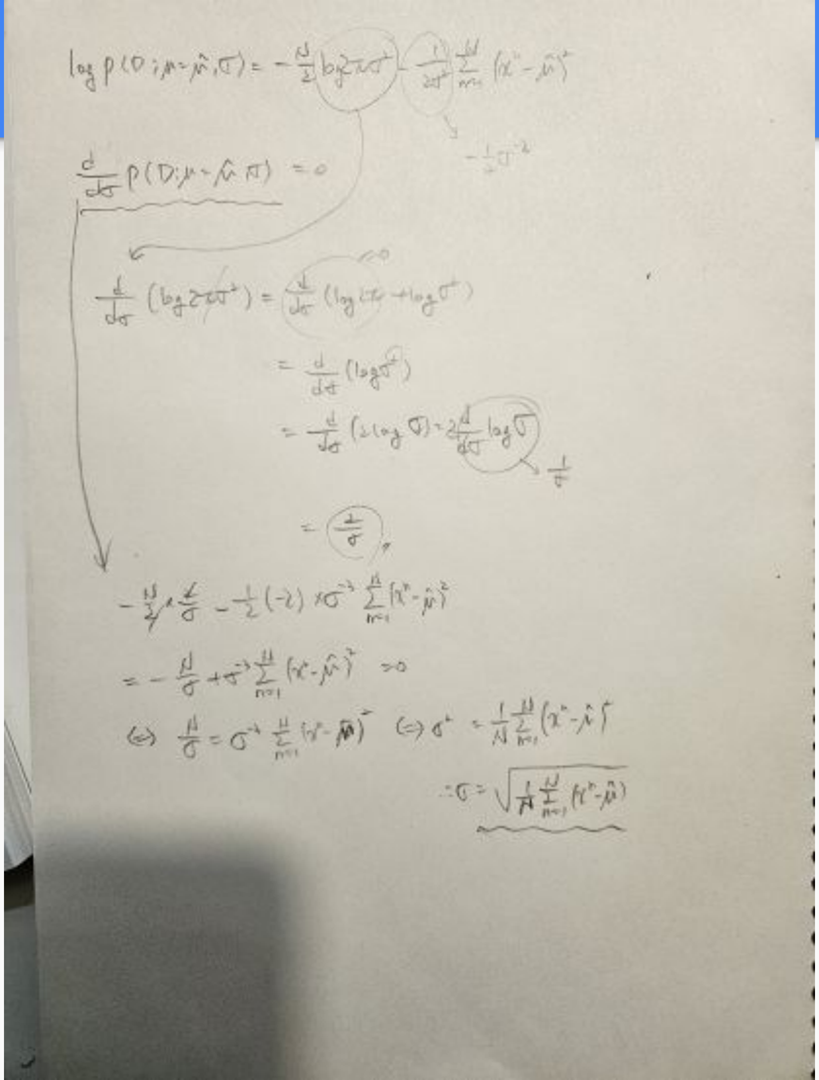
$$\begin{aligned}
\frac{\partial}{\partial\sigma}\log p(\mathcal{D}; \mu = \hat{\mu}, \sigma) &= -\frac{N}{2}\underbrace{\frac{\partial}{\partial\sigma}(\log 2\pi\sigma^2)}_{\frac{2}{\sigma}} - \frac{\partial}{\partial\sigma}\left(\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x^{(n)} - \hat{\mu})^2\right)\\
&= -\frac{N}{2}\frac{2}{\sigma} - \frac{1}{2}(-2)\sigma^{-3}\sum_{n=1}^{N}(x^{(n)} - \hat{\mu})^2\\
&= -\frac{N}{\sigma} + \sigma^{-3}\sum_{n=1}^{N}(x^{(n)} - \hat{\mu})^2
\end{aligned}$$

$$\begin{aligned}
& \frac{N}{\sigma} = \sigma^{-3}\sum_{n=1}^{N}(x^{(n)} - \hat{\mu})^2\\
\Leftrightarrow\quad & \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x^{(n)} - \hat{\mu})^2\\
\therefore\quad & \sigma = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x^{(n)} - \hat{\mu})^2}
\end{aligned}$$

$$\hat{\sigma} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x^{(n)} - \hat{\mu})^2}$$

2.3 최대 가능도 추정 이론

- 정규 분포의 최대 가능도 추정



2.4 생성 모델의 용도

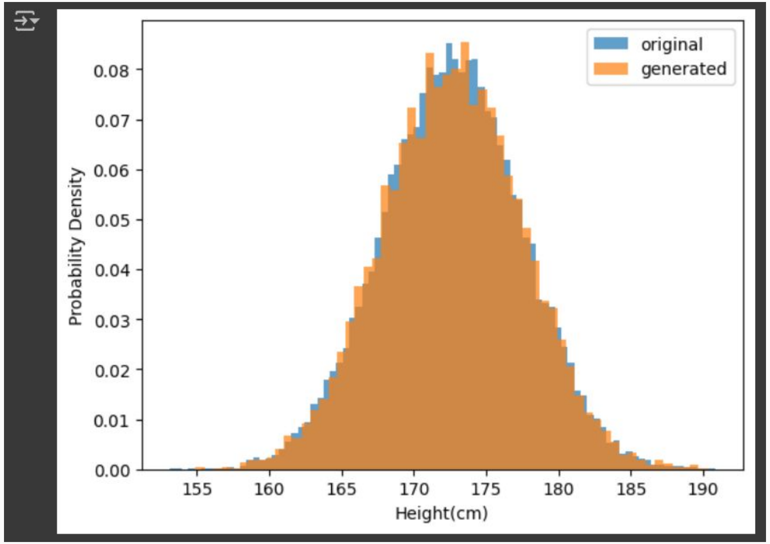
- 2.4.1. 새로운 데이터 생성
  - 매개변수 추정을 학습이라고도 함
  - 정규분포로부터 새로운 데이터를 생성할 수 있음
    - 생성 모델에서 1만개의 샘플을 추출 → 1만 개의 새로운 데이터를 생성 (노란색)
    - 기존 데이터와 분포가 유사 → 우리의 생성 모델이 특성을 잘 포착해 비슷한 데이터를

```python
import os
import numpy as np


path = 'height.txt'
xs = np.loadtxt(path)
mu = np.mean(xs)
sigma = np.std(xs)

samples = np.random.normal(mu, sigma, 10000)

plt.hist(xs, bins='auto', density=True, alpha=0.7, label='original')
plt.hist(samples, bins='auto', density=True, alpha=0.7, label='generated')
plt.xlabel('Height(cm)')
plt.ylabel('Probability Density')
plt.legend()
plt.show()
```



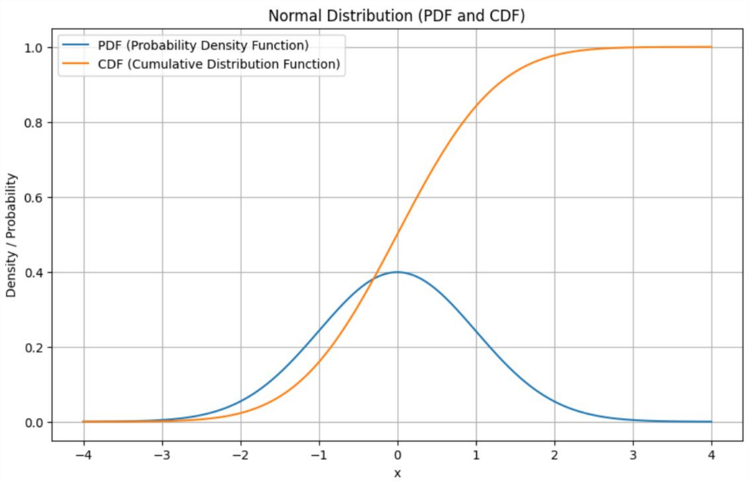
2.4 생성 모델의 용도

- 2.4.1. 새로운 데이터 생성
  - 더 복잡한 생성 모델이라면 다차원 데이터(이미지)를 새로 생성할 수 있음.
    - 생성 모델 만드는 과정은 동일
    - 모델링 + 매개변수 추정

그림 2-9 생성 모델로 생성한 얼굴 이미지(논문[3]에서 인용)

2.4 생성 모델의 용도

- 2.4.2 확률 계산
    - 확률 분포를 알면 어떤 값이 얼마나 발생하기 쉬운지(어려운지) 알수 있음
        - 확률 밀도 p(x)를 적분 (누적분포함수 이용)
        - 수식으로 풀수 없는 경우에는 몬테카를로 방법 (5.1.3절)
    -



2.4 생성 모델의 용도

- 2.4.2 확률 계산
  - 누적분포 함수를 이용
  - x < =1 구간의 면적.

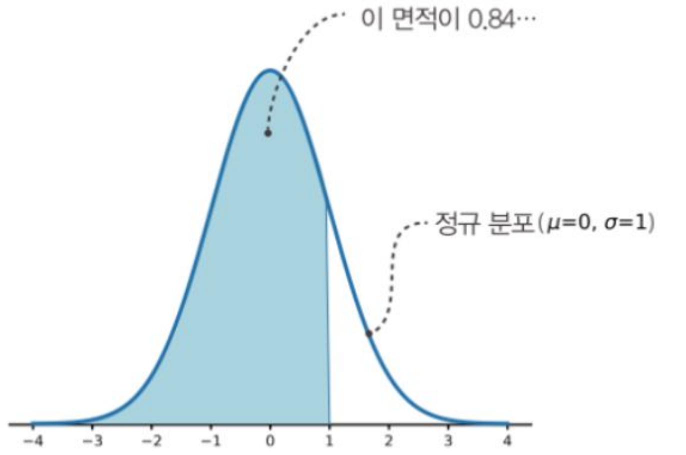
```python
from scipy.stats import norm

x = 1.0
p = norm.cdf(x, loc=0, scale=1) # loc 평균, scale 표준편차

print(p)
```

```
0.8413447460685429
```

그림 2-10 정규 분포(평균 0, 표준 편차 1)의 $x \leq 1$ 구간 면적



2.4 생성 모델의 용도

- 2.4.2 확률 계산
  - 키 데이터
    - 180 이상을 구할 때는 1-(180cm 이하 확률)

```python
import numpy as np
from scipy.stats import norm


path = 'height.txt'
xs = np.loadtxt(path)
mu = np.mean(xs)
sigma = np.std(xs)

p1 = norm.cdf(160, mu, sigma)
print('p(x <= 160):', p1)

p2 = norm.cdf(180, mu, sigma)
print('p(x > 180):', 1-p2)
```

```
p(x <= 160): 0.004271406830855
p(x > 180): 0.06541774339950823
```