Inference

Hyerim Bae
Department of Industrial Engineering, Pusan National University
hrbae@pusan.ac.kr

Contents

추론이란?

- 이미 알고 있는 지식(정보, 명제)를 이용하여 새로운 지식을 도출하는 과정

  - Deduction

    From general principles to specific instances

    $\forall x$ bird(x) $\rightarrow$ lays_egg(x)

    bird(Tweety)

    ---

    lays_egg(Tweety)

  - Induction
    - From specific instances to general principles / concepts / descriptions

    warning(MXMXP)

    warning(JXPMS)

    safe(JLMMX) → past exprience or data

    warning(LMXMJ)............................

    ---

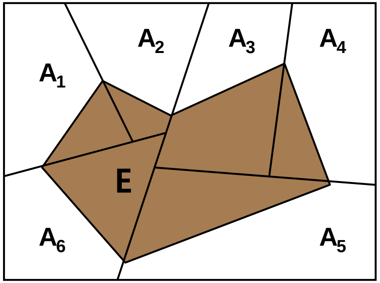
    warning(??)

    safe(??)

기계학습에서의 추론

- y=f(x)에서 f를 찾는 것
- f를 특징 짓는 파라미터를 알아내는 것

Bayes Rule



$$p(A|B) = \frac{p(A,B)}{p(B)} = \frac{p(B|A)p(A)}{p(B)}$$

$$p(A_i|E) = \frac{p(E|A_i)p(A_i)}{P(E)} = \frac{p(E|A_i)p(A_i)}{\sum_i p(E|A_i)p(A_i)}$$

- Based on the definition of conditional probability
  - $p(A_i|E)$ is posterior probability given evidence E
  - $p(A_i)$ is the prior probability
  - $P(E|A_i)$ is the likelihood of the evidence given A_i
  - p(E) is the preposterior probability of the evidence

Bayesian inference

- Let's see the rule again

Likelihood

Prior

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$$

Posterior

man or woman?

$$p(man|long\ hair)$$

Source: https://brunch.co.kr/@chris-song/59

Bayesian 추론

- “베이지안스럽게 추론하기”

Likelihood

Prior

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$$

Posterior

– $P(Model|Data) \cong P(Data|Model) \times P(Model)$

for i=1,2,… do
    estimate parameters from data
    recommend next input
    generate data from model and add
end for

bayes' theorem을 살펴보면

$posterior \propto likelihood \times prior$

이고

$P(Model \mid Data) \propto P(Data \mid Model) \times P(Model)$

이 된다.

즉, 현재까지 얻어진 모델 (prior)과 추가적인 실험 정보 (likelihood)를 통해
데이터가 주어졌을 때의 모델(Posterior)을 추정해나가는 방식이며 알고리즘은 다음과 같다.

(몇가지 초기 입력-결과값 데이터가 주어졌을 때)
for t = 1, 2, ... do
 1. 얻어진 데이터를 토대로 모델을 추정한다.
 2. 추정된 모델을 토대로 '모델 추정에 가장 유용할만한' 다음 입력값을 추천한다.
 3. 모델에 추천된 입력값을 넣어 결과값을 얻어내고, 이를 기존 데이터에 추가한다.
end for

Sampling을 통한 추론

- 우리나라 국민들의 평균 몸무게

$$\int f(x)p(x)dx$$

- 우리나라 국민들이 일 년동안 평균적으로 병원에 가는 횟수

$$\sum_i f(X_i)p(X_i)$$

- By sampling

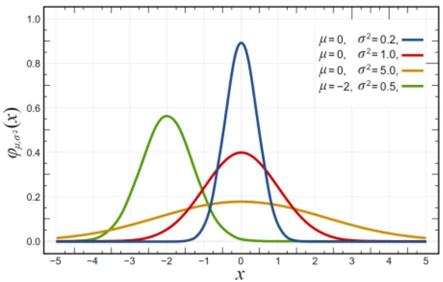
$$\sum_i^N \frac{f(x_i)}{N}$$

What is sampling

- Sampling: obtaining a small sample s to represent the whole data set N: 모집단과 유사한 일부 데이터를 추출
  - Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
  - Reduction of continuous signal to discrete signal
- Sampling in statistics
  - Extracting samples from a distribution $p(x)$

- Random sampling
  - Easy?
  - 무작위 추출 때, 모집단에서 각 개체가 추출될 확률은 모두 같다고 봄

- Is sampling easy?
  - N개로 구간을 쪼갠 다음 각 구간의 중심값으로만 샘플링?
    →상태 개수가 N개로 고정되어 버림
    양 끝쪽의 샘플을 정확히 얻을 수 없음



Why do we need sampling?

- Optimization problem solving
- Is it easy to know mean value?

$$\int f(x)p(x)dx \approx \sum_{i=1}^{N} \frac{f(X_i)}{N}$$

- Bayesian

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$$

Calculation is not easy

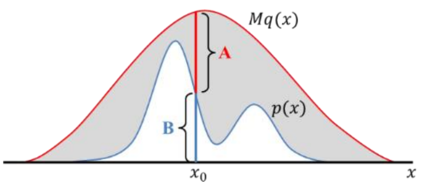
- 샘플링이 필요한 이유

1. 높은 차원의 최적화 문제를 풀 때
2. 높은 차원의 적분을 할 때
3. 특정 확률밀도함수의 평균값 계산
4. 특정 확률값 계산

Rejection sampling

- In the continuous variable
  - Use a known distribution q(x) (ex. *Normal*)
  - $u \sim U(0,1)$

$$if\ \frac{p(x)}{Mq(x)} < u, reject\ x$$



- How to find *M*?

$$p(accept) = \int \left\{ \frac{p(x)}{Mq(x)} \right\} q(x)dx = \frac{1}{M} \int p(x)\, dx$$

  - Acceptance rate is inversely proportional to M
  - Find a smallest M which satisfy $p(x) \leq Mq(x)$

```
def p(x):
    return st.norm.pdf(x, loc=30, scale=10) + st.norm.pdf(x, loc=80, scale=20)

def q(x):
    return st.norm.pdf(x, loc=50, scale=30)
```

```
[4] x = np.arange(-50, 151)
    k = max(p(x) / q(x))
```

```
def rejection_sampling(iter=1000):
    samples = []

    for i in range(iter):
        z = np.random.normal(50, 30)
        u=np.random.uniform(0,1)
        if u <= p(z)/(k*q(z)):
            samples.append(z)
    return np.array(samples)
```

Importance sampling

- The goal of sampling
  - Calculating the expectation of PDF
  - Calculating a certain probability

즉, 위 2가지만 잘 할 수 있으면 sampling을 여러 번 하지 않아도 됨(rejection sampling 보완)
→ **Importance sampling** 도입

$$E[f(z)] = \int_z f(z)p(z)dz = \int_z \frac{f(z)p(z)}{q(z)}q(z)dz \begin{cases} \cong \frac{1}{L}\sum_{l=1}^{L}\frac{p(z^l)}{q(z^l)}f(z^l) \\ E_{z\sim q}[\frac{p(z)}{q(z)}f(z)] \end{cases}$$

- z : 확률변수
- p : 확률변수 z의 확률분포 → 정확히 알 수 없다.
- f : 임의의 함수
- q : p에서 샘플 생성 어려울 때 → 비교적 샘플의 생성이 쉬운 q에서 샘플 생성
  → f 기댓값 계산

Code

```
[42] value_list = []
     for i in range(n):
         # sample from different distribution
         x_i = np.random.normal(mu_appro, sigma_appro)
         value = f_x(x_i)*(p_x.pdf(x_i) / q_x.pdf(x_i))

         value_list.append(value)
```

Algorithm 1: Sampling-importance-resampling

Input : the number of samples N,
nominal distribution p,
proposal distribution q

Output : samples $X = \{x_1, x_2, \ldots, x_N\}$

1. $X = \{\}$
2. $Z = \{\}$
3. **for** $n = 1 : N$ **do**
4. $\quad Z \leftarrow Z \cup \{z_n\} \quad z_n \sim q$
5. **end**
6. **for** $n = 1 : N$ **do**
7. $\quad w_n = \frac{p(z_n)/q(z_n)}{\sum_{i=1}^{N} p(z_i)/q(z_i)}$
8. **end**
9. generate N samples using w.

MCMC = MC (Monte Carlo) + MC (Markov Chain)

- Monte Carlo
  - 통계적 수치를 얻기 위한 Simulation
    - 무한히 많은 시도가 불가능하기 때문에
    - 유한한 시도만으로 추정

Stanisław Marcin Ulam

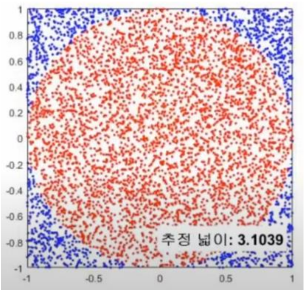
- Markov Chain
  - 시간에 따른 계의 상태 변화

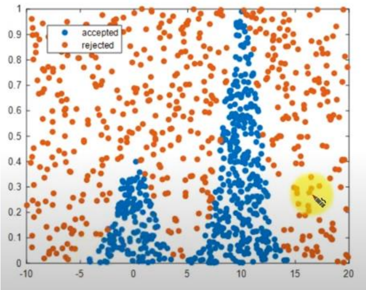
$$P(Z_i|Z_{i-1},\cdots,Z_1) = P(Z_i|Z_{i-1})$$

Andrei Andreyevich Markov

MCMC 원리

- MC





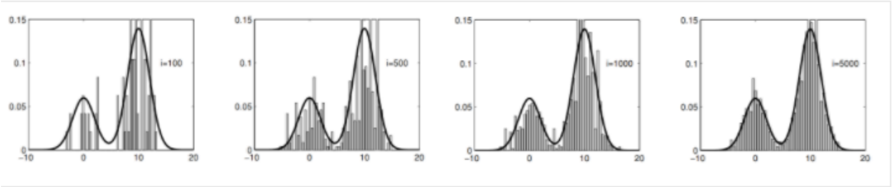
- MC
  - 가장 마지막에 뽑힌 샘플이 다음 번 샘플을 추천

MCMC

- Markov Chain
  - 마코프 가정(Markov assumption)을 따르는 이산 시간 확률 과정
  - 특정 시점의 상태 확률은 단지 그 이전 상태에만 의존한다 = $P(Z_i|Z_{i-1},\cdots,Z_1) = P(Z_i|Z_{i-1})$
  - 특정 조건을 만족한 상태에서 Markov Chain을 반복하다 보면 현재 상태의 확률이 직전 상태의 확률과 같아지게(수렴)된다. 평형 상태에 도달한 확률 분포를 정적 분포(Stationary Distribution)
    - 특정조건 : 기약, 에르고딕(양의 재귀적&비주기적)
- Monte Carlo
  - 랜덤 표본을 뽑아 함수의 값을 확률적으로 계산하는 알고리즘
- MCMC
  - 조건
    - $P(Z_{n+1}|Z_n)$으로부터 표본을 생성하기 쉬워야 한다.
    - 관심 있는 목표분포가 마르코프 연쇄의 정상분포가 되어야 한다.
  - Markov Chain에 기반한 확률 분포로 부터 표본을 추출하는 Monte Carlo방법
    - 초기값 Z_0를 지정한다
    - 마르코프 연쇄가 정상분포에 도달할 때까지 N개의 관측값 $\{Z_n, n = 1, \ldots, N\}$을 생성한다
    - 수렴진단 방법을 이용하여 알고리즘의 수렴 여부를 확인한다. 만약 수렴하지 않으면, 더 많은 관측값을 생성
    - 초기 B개의 관측값을 제거한 후, $\{Z_{B+1}, \ldots, Z_N\}$을 분석을 위한 표본으로 간주한다.
    - 생성한 표본을 이용하여, 요약통계량(평균, 중앙값, 표준편차, 분위수, 상관계수 등) 및 표본을 추정한다
  - Markov Chain Sample은 iid(Independent and Identically distributed)가 아님

MCMC

- MCMC를 쓰는 이유
  1. 빠른 수렴 속도
  2. 더 많은 확률분포 샘플링 가능
- 기존의 샘플링과 차이
  - 기존 샘플링의 문제점은 과거의 레코드는 사용하지 않음
  - Rejection sampling은 조건이 맞지 않으면 버림
  - Importance sampling은 개별 sample들이 다 독립이었음



MCMC가 특정 확률 분포를 샘플링 하는 과정

MCMC procedure

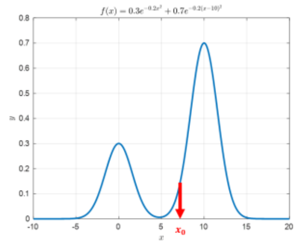
- Step1: 첫 샘플을 랜덤하게 선정
- Step2:해당 샘플에 의해서 다음번 샘플이 추천 (제안분포를 이용)
- Step3:특정 기준에 따라 추천된 샘플을 Accept or Reject
- Step4: repeat step 2 and 3

$$f(x) = 0.3e^{-0.2x^2} + 0.7e^{-0.2(x-10)^2}$$

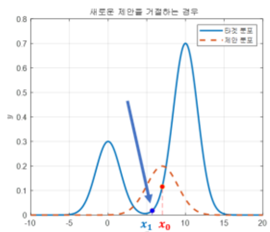
Metropolis 예시

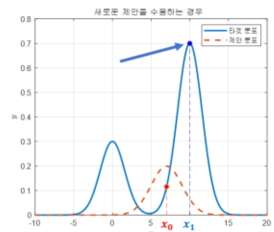
https://angeloyeo.github.io/2020/09/17/MCMC.html

1. Random initialization



2. Next recommendation by proposed distribution





3. Accept or reject? ($r = \frac{p(x_1)}{p(x_0)} >$

4. In the case r <1,
   - Give a chance to be sampled with the probability of r

$$\frac{p(x_1)}{p(x_0)} > \mathrm{u}\ ?$$

M-H algorithm (Metropolis-Hastings algorithm)

- MH(Metropolis-Hastings) is the basic algorithm of MCMC
- MH algorithm
  - MH란?
    - Target distribution인 $P(Z)$에 비례하는 임의의 함수 $f(Z)$를 계산할 수 있을 때 $P(Z)$를 MCMC샘플링 할 수 있는 알고리즘
    - 현재 z^t라고 assignment 되어있는 것을 다음 transient를 통해 어떠한 assignment로 바뀔지 candidate하는 것. 기존의 정보를 활용(z^t)에서 제안하는 것이 MCMC의 근간에서 따온 것
      → 이 것을 **Proposal distribution**이라 함
  - $f(z)$만으로는 $p(z)$ 샘플링 불가
    → $q(z^t|z)$라는 쉽게 샘플링이 가능한 조건부 확률 분포 proposal distribution이 필요
    이전 상태 z로 부터 다음 상태 z^t 예측
  - α(acceptance probability)에 따라 coin toss를 해서 accept가 일어나면 z^*(candidate)를 받아들이고
    reject되면 기존의 z^t를 사용

M-H algorithm (Metropolis-Hastings algorithm)

- $step1$
  - $p(z)$: $target\ density$
  - $q(z)$: $proposal\ density$
    - $z^{*} \sim q(z^{*}|z^{t})$: 새로뽑는 샘플이 이전 샘플에 종속
- $Step2$
  - $Calculate\ the\ ratio, r(z^{*}|z^{t}) = \dfrac{q\left(z^{t}\middle|z^{*}\right)p(z^{*})}{q\left(z^{*}\middle|z^{t}\right)p(z^{t})}$: q를 가우시안 분포로 사용
- $step3$
  - $u \sim U(0,1)$
  - $\alpha(z^{*}|z^{t}) = \min(1, r(z^{*}|z^{t}))$: 알파가 1보다 작거나 같은 값이 추출됨, 1이 뽑히면 r이 1보다 크므로 무조건 accept, 그렇지 않으면 패자 부활전
  - $\begin{cases} u < a(z^{*}|z^{t}) & z^{t+1} = z^{*} \quad (Accpet) \\ \quad else & z^{t+1} = z^{t} \quad (Reject) \end{cases}$
- M-H algorithm을 통해 Markov chain 생성
- Monte Carlo 방법 적용해서 샘플링