Sampling based inference

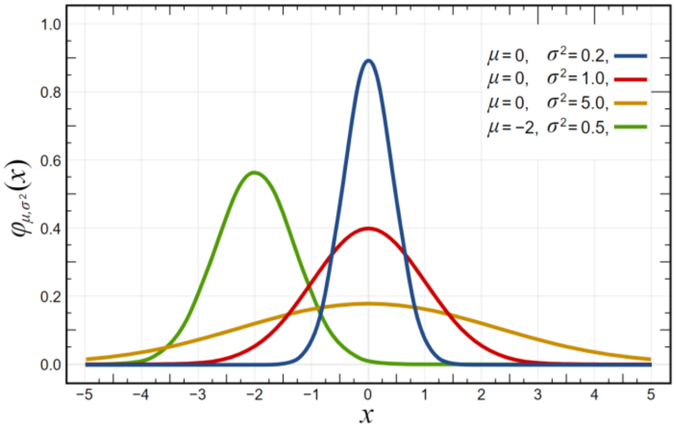
Overview

- Introduction to sampling
- Rejection sampling
- Importance sampling
- MCMC (Markov Chain Monte Carlo)
- M-H sampling (Gibbs sampling)

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

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?
  - Samples are extracted with the same probability
    - 무작위 추출 때, 모집단에서 각 개체가 추출될 확률은 모두 같다고 봄
- Is sampling difficult?
  - N개로 구간을 쪼갠 다음 각 구간의 중심값으로만 샘플링?
    →상태 개수가 N개로 고정되어 버림
    양 끝쪽의 샘플을 정확히 얻을 수 없음



Prof. Hyerim Bae (hrbae@pusan.ac.kr)

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

- When sampling is required

1. To solve an optimization problem. 높은 차원의 최적화 문제를 풀 때
2. To solve integral. 높은 차원의 적분을 할 때
3. To calculate a mean of distribution. 특정 확률밀도함수의 평균값 계산
4. To get a probability value at a point. 특정 확률값 계산

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

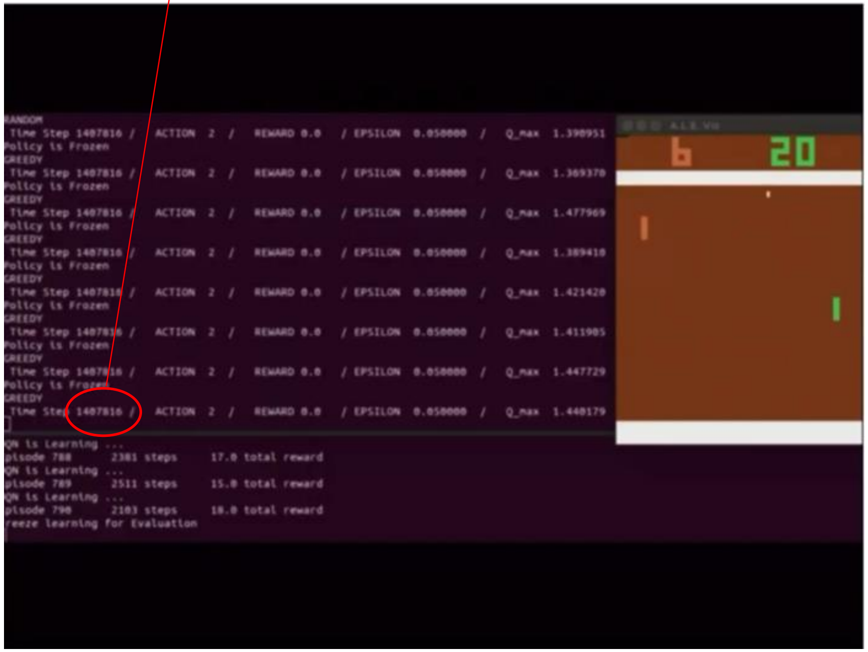
Application: RL Optimization using sampling

- Many trials to reach the optimum

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation 3
  end for
end for

Over 1.4 million trials



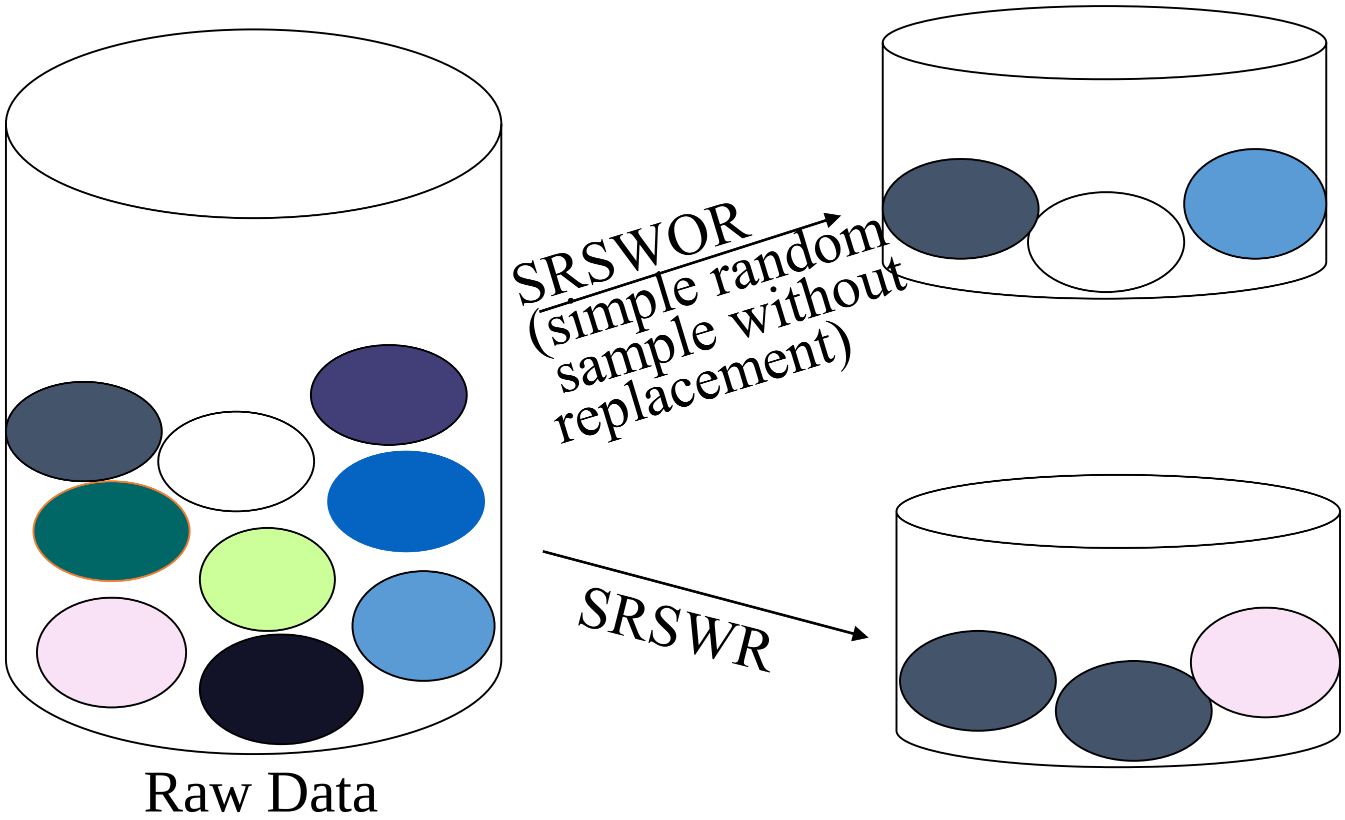
Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Random Sampling

- Key principle: Choose a representative subset of the data
- Note: Sampling may not reduce database I/Os (page at a time)
- **Simple random sampling**
  - There is an equal probability of selecting any particular item
- **Sampling without replacement**
  - Once an object is selected, it is removed from the population
- **Sampling with replacement**
  - A selected object is not removed from the population
- **Stratified sampling:**
  - Partition the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)
  - Used in conjunction with skewed data
- **Systematic sampling**
  - 모집단에 있는 관찰값들에게 번호를 임의로 배정한 후 추출 간격을 두어 추출
- **Cluster sampling**
  - 전체 자료를 집락으로 나누어, 몇 개의 샘플 집단을 랜덤 추출해 추출된 집락 내의 데이터를 전수 조사 하는 샘플링

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

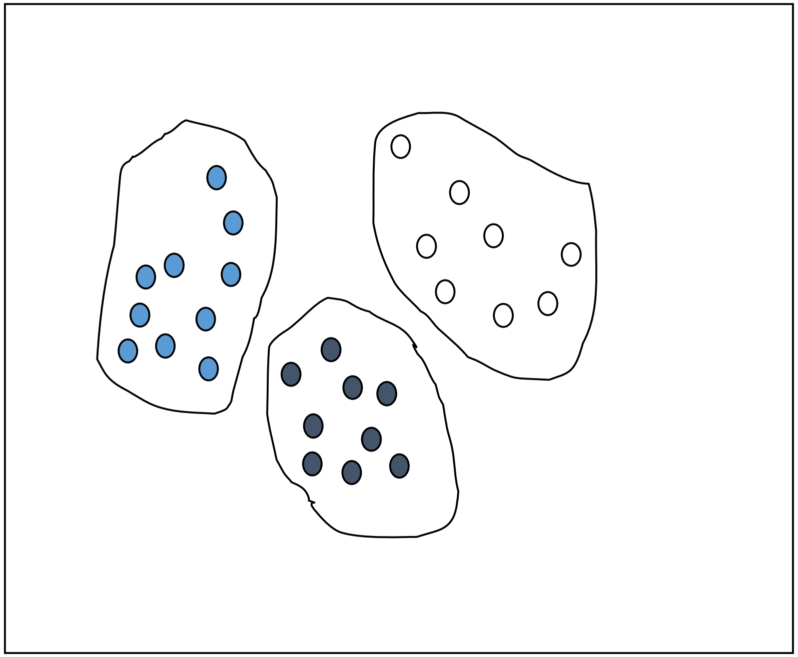
Sampling: With or without Replacement



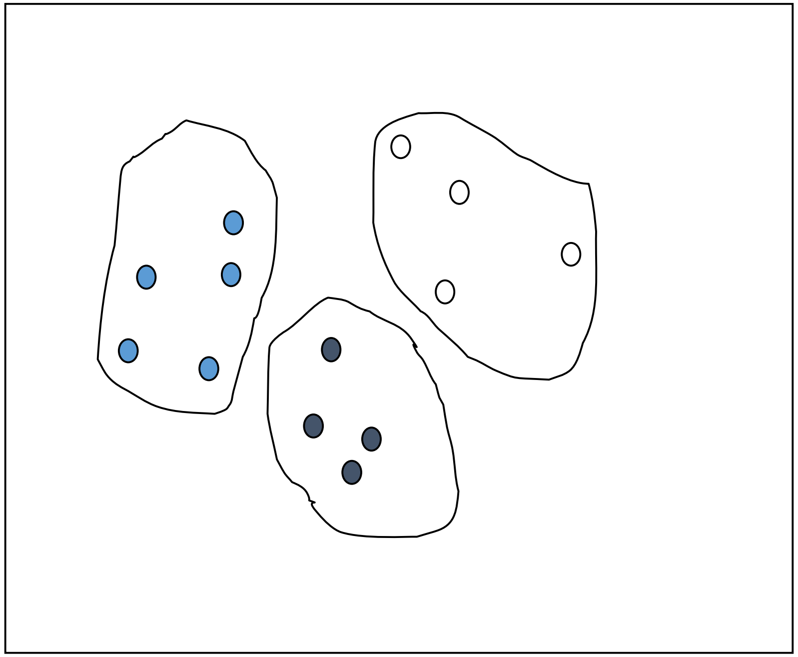
Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Sampling: Cluster or Stratified Sampling

Raw Data



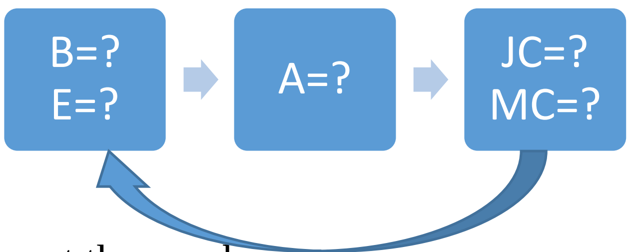
Cluster/Stratified Sample



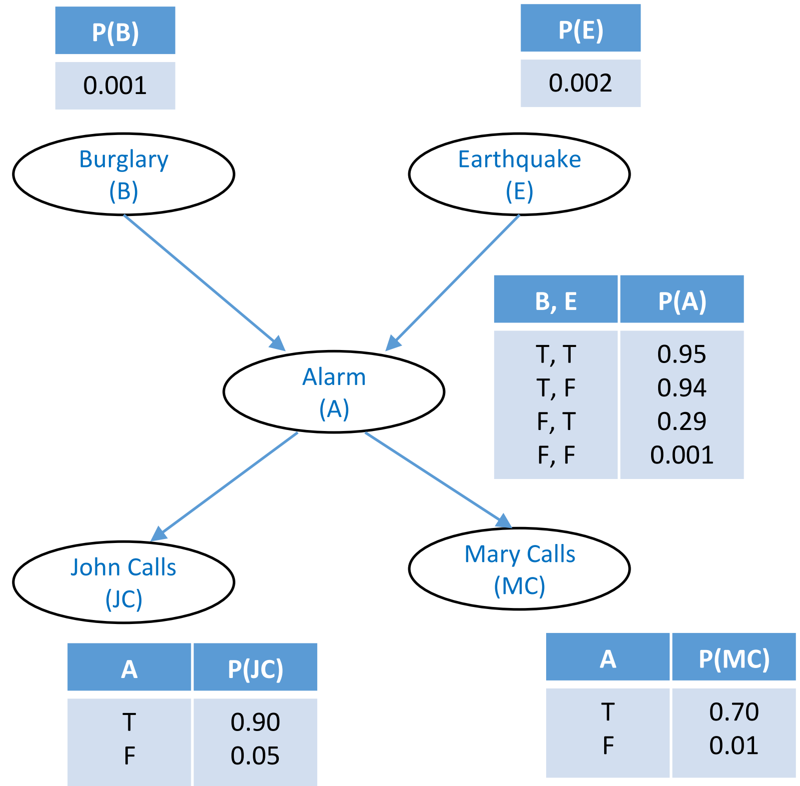
Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Forward sampling

- Generate samples based on the probability of the BN
  - Many samples

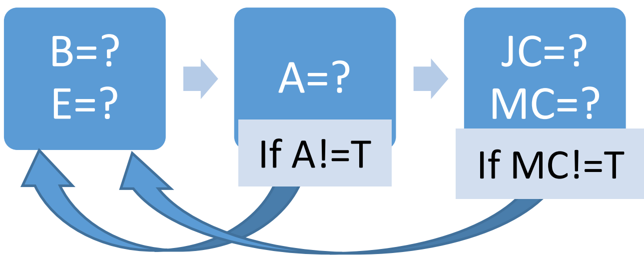


- Count the numbers
  - P(E=T|MC=T, A=T) ?
  - 계속 해본 뒤, 위의 경우를 만족하는 경우를 카운트
- What are problems of FS?
  - Error from random cases
    - Random case라 오차 발생 가능
  - It takes long time
    - 시행 횟수가 많을수록 시간이 많이 걸림



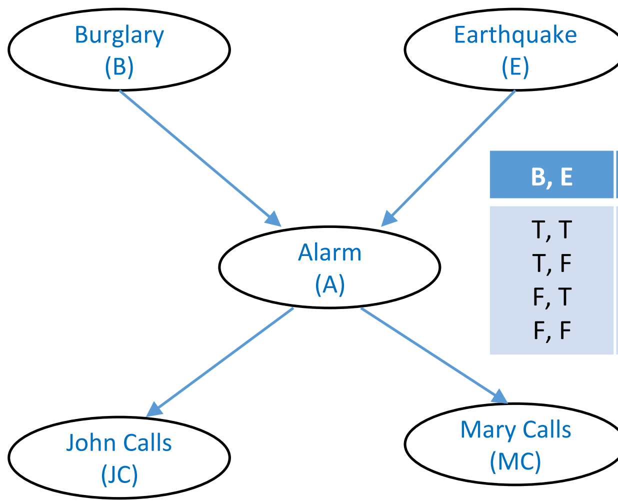
Rejection sampling

- If a sample doesn't satisfy the condition discard it.
  - If P(E|MC=T, A=T)
  - Count only the number of samples MC='T' and A='T'



  - Reject cases which do not satisfy the condition
    - 카운트하는 중간에 해당 조건을 만족못하면 카운트 하지 않음

- P(E|MC, A) =P(E, MC, A)/P(MC,A)
  - P(E, MC, A)/(# of accepted samples)



P(B)
0.001

P(E)
0.002

B, E	P(A)
T, T	0.95
T, F	0.94
F, T	0.29
F, F	0.001

A	P(JC)
T	0.90
F	0.05

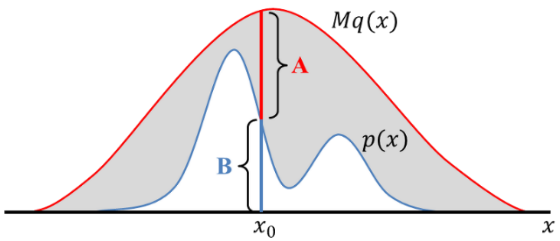
A	P(MC)
T	0.70
F	0.01

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

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 $p(x) \le Mq(x)$

```
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)
```

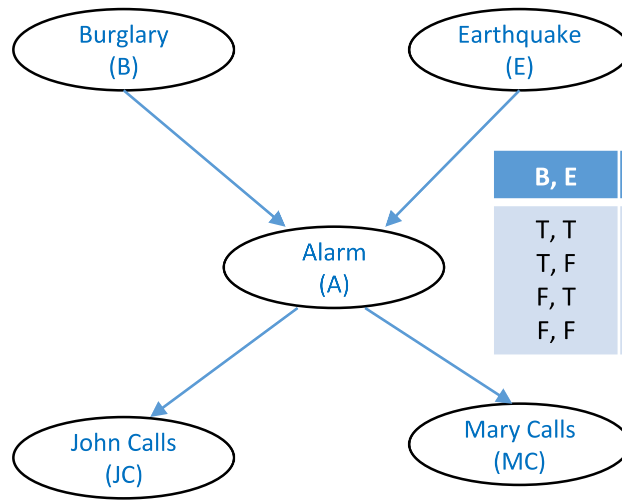
Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Importance sampling

- $P(E = T|MC = T, A = F) =?$
  - $SumSW = TotSW = 0$
  - Loop
    - sw(Sampling weight) = 1
      - ✓ B=F
      - ✓ E=F
      - ✓ A=F|B=F,E=F
        - sw=1*0.999
      - ✓ JC|A=T
      - ✓ MC=T|A=F
        - sw=sw*0.01
    - If E==T => SumSW=SumSW+sw
    - TotSW=TotSW+sw
  - Importance: SumSW/TotSW

P(B)
0.001

P(E)
0.002



B, E	P(A)
T, T	0.95
T, F	0.94
F, T	0.29
F, F	0.001

A	P(JC)
T	0.90
F	0.05

A	P(MC)
T	0.70
F	0.01

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

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

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

- 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

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

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 $X = \{x_1, x_2, ..., x_N\}$

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

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

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

Stanisław Marcin Ulam

- Markov Chain
  - Sate change
    - 시간에 따른 계의 상태 변화

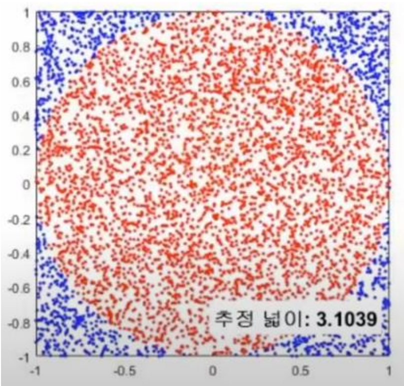
$$P(Z_i|Z_{i-1},\cdots,Z_1) = P(Z_i|Z_{i-1})$$

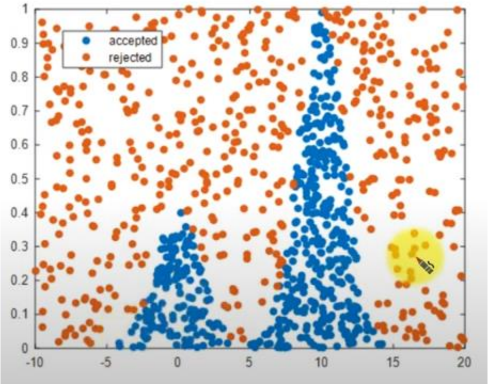
Andrei Andreyevich Markov

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

MCMC 원리

- MC





- MC
  - 가장 마지막에 뽑힌 샘플이 다음 번 샘플을 추천

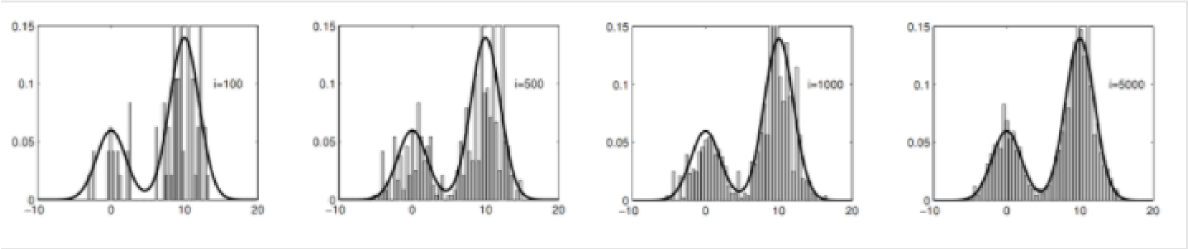
Prof. Hyerim Bae (hrbae@pusan.ac.kr)

MCMC

- Markov Chain
  - a stochastic model describing a sequence of possible events
    - 마코프 가정(Markov assumption)을 따르는 이산 시간 확률 과정
  - $P(Z_i|Z_{i-1},\cdots,Z_1) = P(Z_i|Z_{i-1})$
    - 특정 시점의 상태 확률은 단지 그 이전 상태에만 의존한다 =
  - Stationary Distribution
    - 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
  - to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables.
    - 랜덤 표본을 뽑아 함수의 값을 확률적으로 계산하는 알고리즘
- MCMC
  - 조건
    - Easy to generate samples: $P(Z_{n+1}|Z_n)$으로부터 표본을 생성하기 쉬워야 한다.
    - Target distribution is stationary distribution: 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

- Why do we use MCMC?
  1. Fast convergence: 빠른 수렴 속도
  2. 더 많은 확률분포 샘플링 가능
- Difference from the other sampling methods
  - Use historical records: 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

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

MCMC procedure

- Step1: randomly select the first sample
- Step2: recommend the next sample from it (use the proposed distribution)
- Step3: Accept or Reject
- Step4: repeat step 2 and 3

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Proposal Distribution

- If proposal distribution is asymmetric?
  - M-H (Metropolis Hastings)

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

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

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

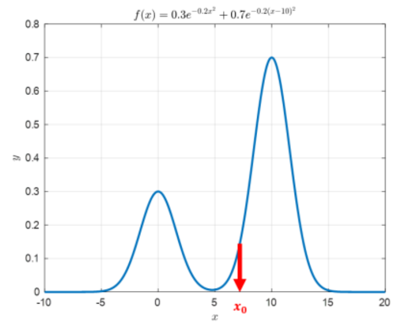
M-H algorithm (Metropolis-Hastings algorithm)

- $step1$
  - $p(z)$: $target\ density$
  - $q(z)$: $proposal\ density$
    - $z^{*} \sim q(z^{*}\ |z^{t})$: 새로뽑는 샘플이 이전 샘플에 종속
- $Step2$
  - $Calculate\ the\ ratio, r(z^{*}|z^{t}) = \frac{q\left(z^{t}|z^{*}\right)p(z^{*})}{q\left(z^{*}|z^{t}\right)p(z^{t})}$: 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

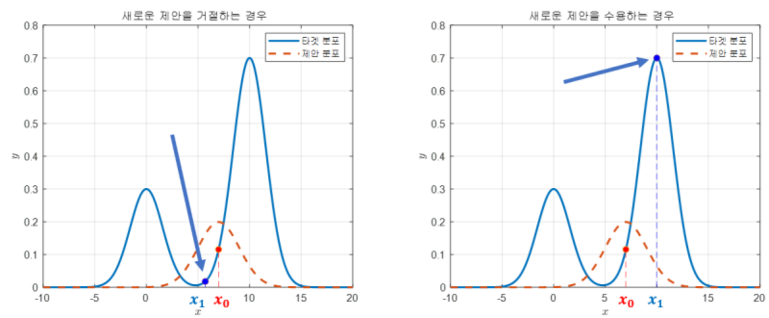
Metropolis 예시 $f(x) = 0.3e^{-0.2x^2} + 0.7e^{-0.2(x-10)^2}$

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

1. Random initialization



2. Next recommendation by proposed distribution



3. Accept or reject? ($r = \dfrac{p(x_1)}{p(x_0)} > 1$?)

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

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

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Exploration

- Accept 의 과정: Exploitation
  - 해당 봉우리만 계속 탐색
  - 제안 분포를 넓게 만들면?
- 패자부활전: Exploration
  - 다른 봉우리를 탐색할 수 있도록 유도
  - 제안 분포를 충분히 넓게 만들 수 없으므로

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Gibbs Sampling

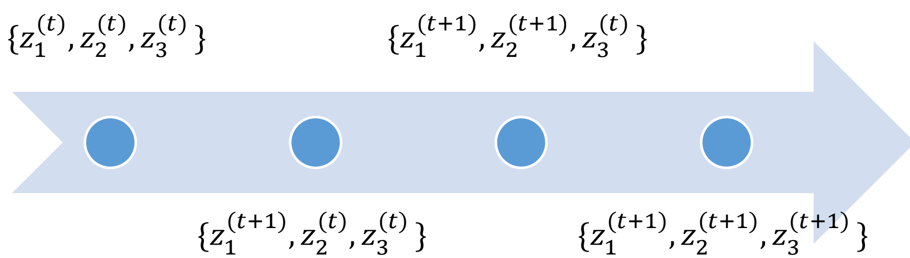
- $q(z^*|z^t) = P(Z_k^*, z_{-k}^*|z_{-k}^t) = P(Z_k^*|z_{-k}^t\ )$으로 놓으면
- $p(z^t)q(z^*|z^t) = \ q(z^t|z^*)p(z^*)$가 항상 성립
  → acceptance probability가 필요 없어짐$(\alpha(z^*|z^t) = 1)$
  → 항상 reversible
  → 항상 detailed balance equation이 자동으로 만족
- $p(z^t)q(z^*|z^t) = p(z_k^t, z_{-k}^t)p(z_k^*|z_{-k}^t)$

$$
\begin{aligned}
&= p(z_k^t, z_{-k}^t)p(z_{-k}^t)p(z_k^*|z_{-k}^t) \\
&= p\left(z_k^t|z_{-k}^t\right) p(z_k^*, z_{-k}^t) \\
&= \mathrm{q}\left(z^t\ |z^*\ \right)p(z^*\ )
\end{aligned}
$$

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Gibbs Sampling

- A special case of M-H algorithm
- *목표분포가 다변량분포일 때 주로 이용*
- *모든 단변량 조건부 밀도함수가 완전히 알려진 형태*
- *q라는 것을 새롭게 만들지 말고 기존에 P를 이용하는 것*
- *Latent variable* Z_k^*에 *대해서만 Update*
- Example
  - $p(z_1, z_2, z_3)$
  - $z_1 \sim p(z_1 | z_2^t, z_3^t) \rightarrow \mathrm{z}_1^{\mathrm{t+1}}$
  - $z_2 \sim p(z_2 | z_1^{t+1}, z_3^t) \rightarrow \mathrm{z}_2^{\mathrm{t+1}}$
  - $z_3 \sim p(z_3 | z_1^{t+1}, z_2^{t+1}) \rightarrow \mathrm{z}_3^{\mathrm{t+1}}$



Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Conclusion

- EM의 경우 saddle point/local optimum point가 존재하기 때문에 더 좋은 assignment가 있어도 stop될 수 있음
- But, Sampling Based Inference의 경우 local에서 탈출할 수 있어 시간이 많으면 MCMC가 더 좋은 성능을 보임
- 짧은 시간 안에 해야 한다면 MCMC…or VI(variational inference)사용