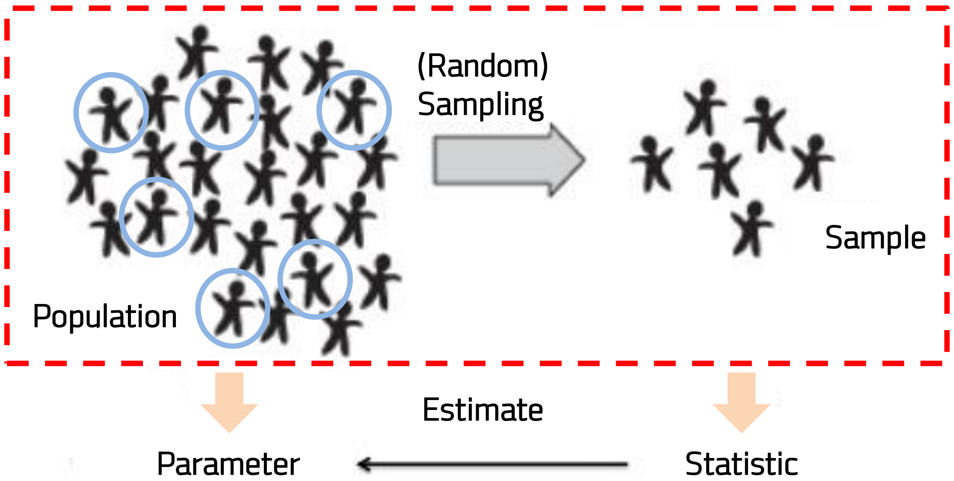
Sampling

INDEX

Simple sampling

❖ Sampling(표본추출)

- 모집단 전부를 대상으로 조사를 시행할 수 없을 때,
  모집단의 일부를 객관적으로 추출하는 방법.



모집단의 평균(μ)
모집단의 표준편차(σ)

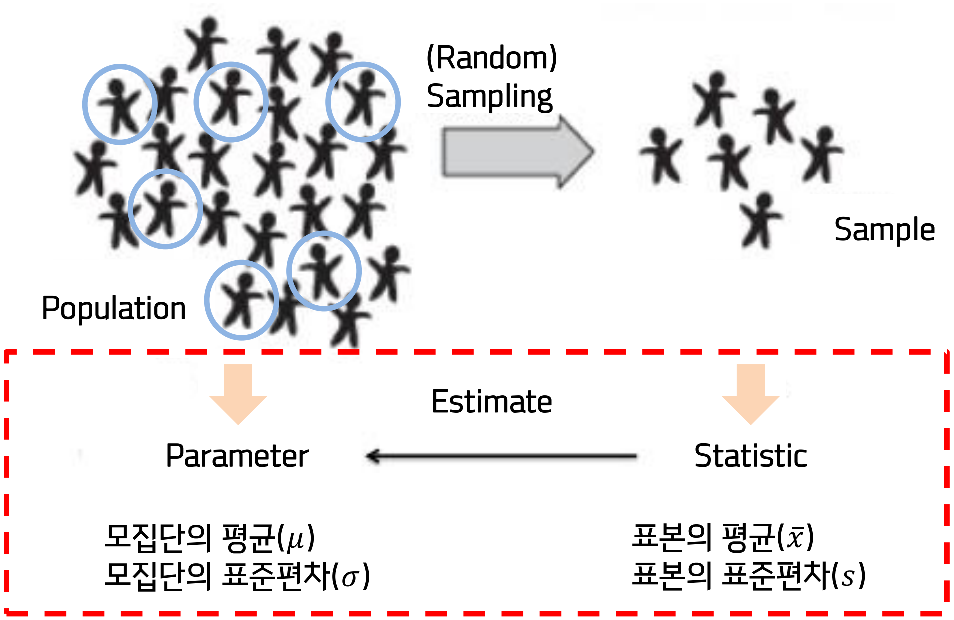
표본의 평균($\bar{x}$)
표본의 표준편차(s)

Simple sampling

❖ Sampling의 목적

- 표본에서 얻은 추정치를 이용해 모수를 알고자 함.

  예) 모집단의 평균, 모집단의 표준편차, ...

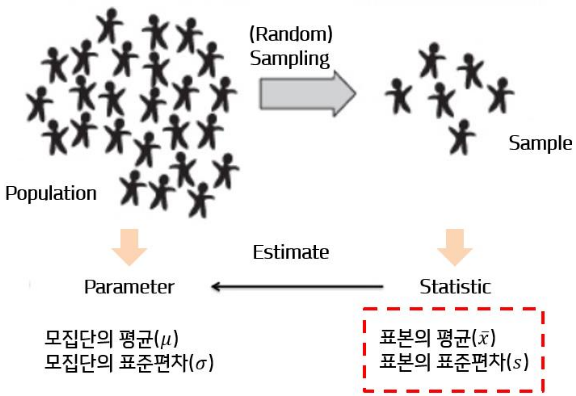


Simple sampling

01
02
03
04

❖ Sample의 추정치를 구하는 방법

- Maximum Likelihood Estimate(MLE)
  - 주어진 정보에 대해 우도를 최대화 하는 추정치를 구함



  - 우도(Likelihood function)

    주어진 데이터 : $x_1, x_2, \ldots, x_n$

    구하고자 하는 모수 : p, $0 \leq p \leq 1$

    x_i : binary(동전의 앞뒤), p : 성공(1)이 나올 확률

    $$L(p) = L(p; x_1, x_2, \ldots, x_n) = \mathrm{f}(\mathrm{x}_1; \mathrm{p}) * \mathrm{f}(\mathrm{x}_2; \mathrm{p}) * \ \ldots\ * \mathrm{f}(\mathrm{x}_n, \mathrm{p})$$
    $$= p^{\sum x_i}(1-p)^{n-\sum x_i}$$

    $L(p)$를 최대화하는 p값을 구한다. 보통 미분을 취해 얻을 수 있다.

  - 예) 동전던지기의 결과 1, 1, 1, 0, 1이 나왔다고 하자. 이때 1이 나올 확률은?

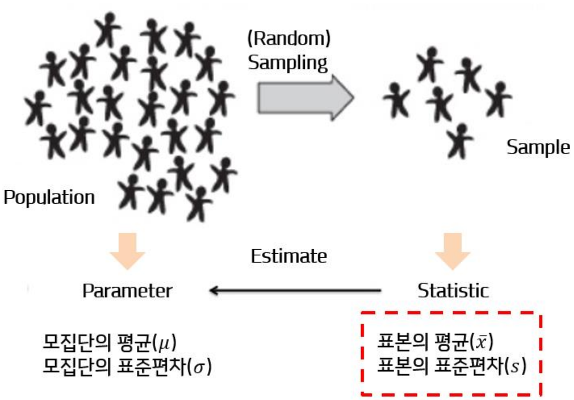
    $$\ln(L(p)) = 4 * \ln(p) + (5-4) * \ln(1-p)$$

    $$\frac{d \ln(L(p))}{dp} = 4 * \frac{1}{p} - 1 * \frac{1}{1-p} = 0 \quad \text{->} \quad \hat{p} = 0.8$$

Simple sampling

01
02
03
04

❖ Sample의 추정치를 구하는 방법



- Least Squares Estimate(LSE)
  - 적합한 모형의 예측값과 실제값과의 **오차제곱합을 최소**로하는 추정량을 구함
  - 오차제곱합(Sum of Squred Error)_SSE

    $SSE = \sum(y - \hat{y})^2$

    오차의 제곱합을 최소로하는 모형의 parameter를 구한다. 미분으로 구할 수 있다.

  - 예) 회귀분석에서 β를 구하는 대표적인 방법론

    $\hat{y} = \beta_0 + \beta_1 x,\ S(\beta_0, \beta_1) = \sum(y - \beta_0 - \beta_1 x)^2$ 이다.

    S는 오차제곱합.

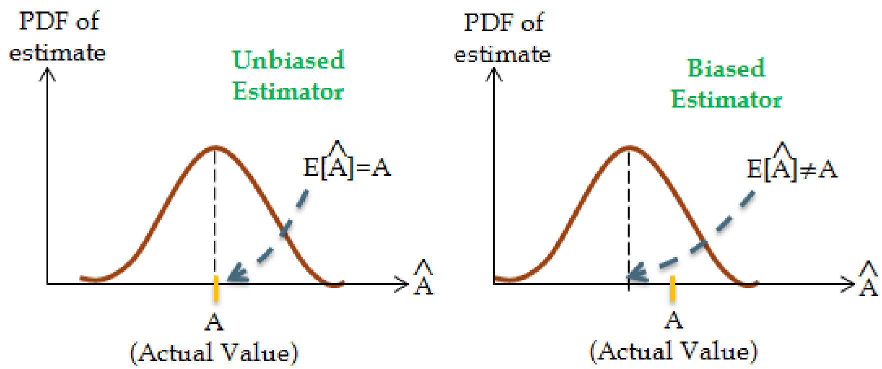
    이때 S를 β_0, β_1 에 대해 각각 미분하면 S를 최소화 하는 β_0, β_1를 구할 수 있다.

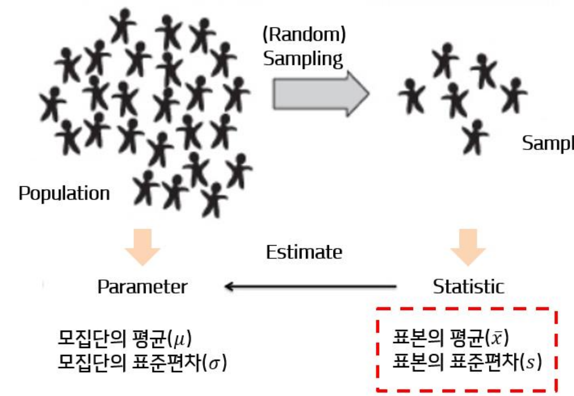
    $\beta = (X^T X)^{-1} X^T Y$

❖ Sample의 추정치의 성능

- Unbiased estimator(불편추정량)

$$\underline{\boldsymbol{Bias = E(\hat{\theta}) - \theta}}$$





- 최소분산 불편추정량

Rao-Blackwell 정리

$X_1, X_2, \dots, X_n$을 $f(x;\theta)$로부터의 확률표본이라 하자. $\boldsymbol{\hat{\theta} = t}(X_1, X_2, \dots, X_n)$이 θ에 대한 불편추정량이고 $Y = u(X_1, X_2, \dots, X_n)$가 θ에 대한 충분통계량일 때

$$\phi(Y) = E\left(\hat{\theta} \middle| Y\right)$$

는 θ에 대해 불편추정량이고, 모든 $\theta \in \Theta$에 대해

$$Var_\theta\left(\phi(Y)\right) \le Var_\theta\left(\hat{\theta}\right)$$

이다.

Simple sampling

❖ Sampling 방법

- 난수 추출
  - U(0, 1)로 부터 0부터 1사이의 실수 값 하나를 추출하고 이를 주어진 사건에 mapping해서 표본을 추출하는 방법

예) 당번 몰아주기

당번 이름	철수	영희	민수
난수 범위	(0.000, 0.333)	(0.333, 0.666)	(0.666, 1.000)

Simple sampling

01

02

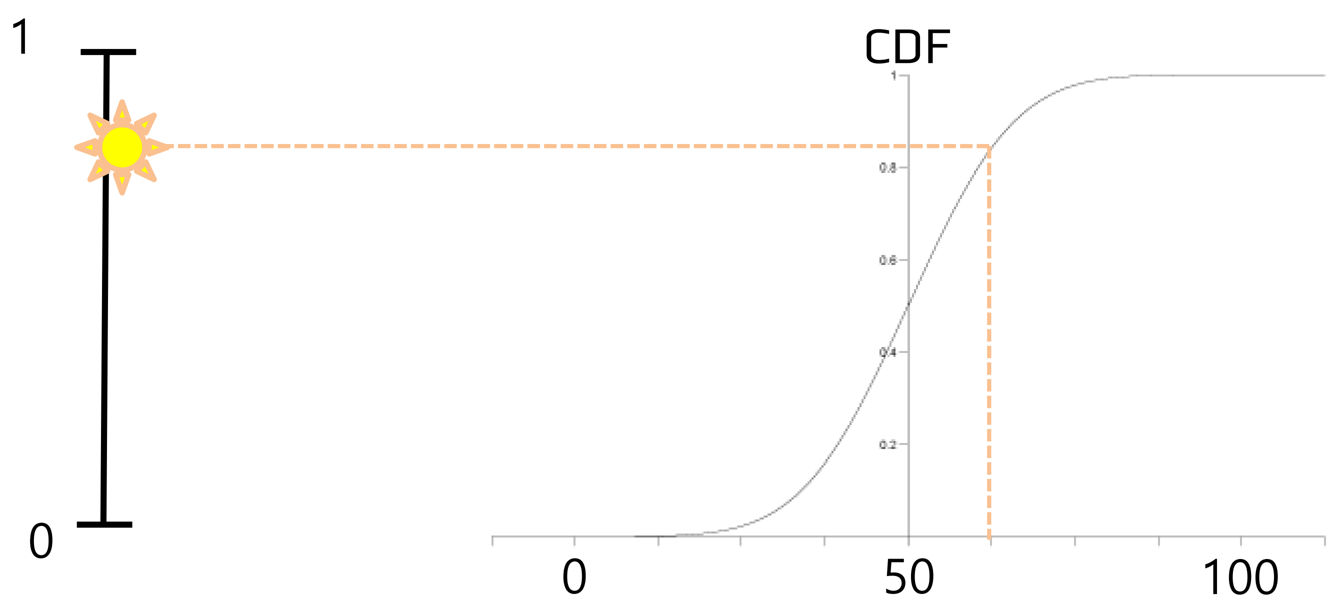
03

04

❖ Sampling 방법

- 특정 분포를 따르는 표본 추출(Inverse CDF)
  - U(0, 1)로 부터 0부터 1사이의 실수 값 하나를 추출하고 이를 주어진 분포에 mapping해서 표본을 추출하는 방법
  - If $U \sim Unif(0,1)$, then $X = F^{-1}(U)$ is a simulation from $f(x)$ [단, $F(x) = f(X \leq x)$, F는 f의 누적 확률 분포]

예) '신'이 이번에 만든 인간에게 어떤 능력을 $N(50, \sigma)$에 기반해서 주는 경우

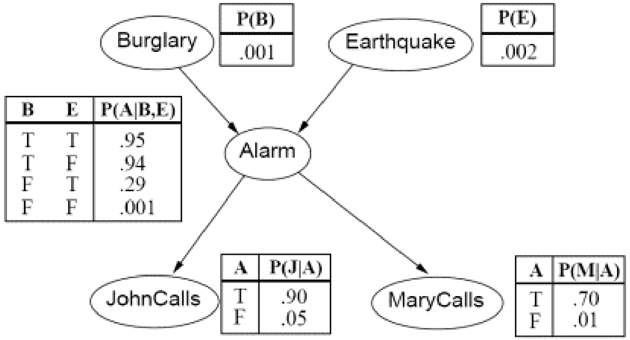


MCMC

❖ Markov Chain Monte Carlo

- Markov Chain + Monte Carlo (MCMC)
  - 목표 확률분포(Target Probability Distribution)로부터 샘플을 얻는 방법

- 앞의 simple sampling과 차이
  - 종속적인 고차원의 변수



- 난수 추출, inverse CDF 등을 사용하기 어렵다

❖ Markov Chain

- 정의
  - 마르코프 연쇄의 구성에 기반한 확률 분포로부터 원하는 분포의 정적(stationary) 분포를 갖는 표본을 추출하는 알고리즘의 한 부류
  - 많은 단계(step)를 거친 이후(정적 분포에 수렴)의 상태에서의 표본추출은 목표로 하는 분포로부터 추출된 표본처럼 사용
- Markov assumption
  - 확률변수(random variable)가 어떤 상태(state)에 도달할 확률이 오직 바로 이전 시점의 상태(state)에만 달려있다.
  - $P(X_k|X_{k-1}, X_{k-2}, \dots, X_2, X_1, X_0) = P(X_k|X_{k-1})$
- Mixing
  - 정적 분포에 수렴
  - 초기값에 의존하지 않음
  - sampling하기 좋은 상태에 도달했다는 의미
  - 얼마나 빨리 수렴하는지가 관건. 수렴 안하는 경우도 존재

01
02
03
04

❖ Markov Chain

- 예제
  - A사(오렌지 주스)와 A'사(오렌지 주스를 다루는 A가 아닌 회사)가 있다.
  - A사의 시장 점유율 : 20%, A'사의 시장 점유율 : 80%
    초기 상태 $S_0 = (0.2 \quad 0.8)$
  - A사가 광고를 진행해서 시장 점유율을 높이고자 함(1주마다 적용)
    전이 확률 $P = \begin{array}{c c} & \begin{array}{cc} A & A' \end{array} \\ \begin{array}{c} A \\ A' \end{array} & \begin{pmatrix} 0.9 & 0.1 \\ 0.7 & 0.3 \end{pmatrix} \end{array}$
  - $S_1 = S_0 * P = (0.2 \quad 0.8) * \begin{pmatrix} 0.9 & 0.1 \\ 0.7 & 0.3 \end{pmatrix} = (0.74 \quad 0.26)$
  - $S_2 = S_1 * P = (0.74 \quad 0.26) * \begin{pmatrix} 0.9 & 0.1 \\ 0.7 & 0.3 \end{pmatrix} = (0.848 \quad 0.152)$
  - $S_3 = S_2 * P = (0.8696 \quad 0.1304)$
  - $S_4 = S_3 * P = (0.87392 \quad 0.12608)$
  - $S_5 = S_4 * P = (0.874784 \quad 0.125216)$
  - $S_{10} = S_9 * P = (0.8749999309 \quad 0.12500000691)$
  - $S_k = S_{k-1} * P$
  - $S = S * P$ => 수렴상태에 도달(Mixing)

★정칙 행렬★
정사각행렬 A에 대해 A^n의 모든 성분이 양수, 역행렬을 갖는다.

❖ Markov Chain

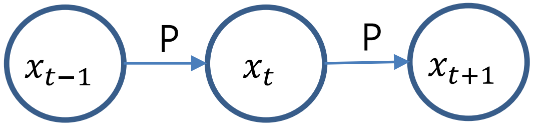
- 정적 분포(stationary distribution)에 수렴하는 조건
  - 전이행렬이 정칙(regular)인 경우 정칙 마르코프 연쇄(regular Markov chains)라고 부르며 이때 정적 분포에 도달할 수 있다.
- 정칙 마르코프 연쇄의 성질
  - $S = S * P$로 정적 분포를 구할 수 있다.
  - 초기행렬 S_0에 어떤 값에 상관없이 S_k는 정적 분포 S에 수렴한다.
  - 전이행렬의 거듭제곱 P^k는 하나의 행렬 $\bar{P}$에 수렴한다.
- 정적 분포를 행렬연산으로 구하기
  - $S = [S_1 \ \ S_2] = [S_1 \ S_2]\begin{bmatrix} 0.9 & 0.1 \\ 0.7 & 0.3 \end{bmatrix} = S * P$
  - $0.9 * S_1 + 0.7 \ * S_2 = S_1, \ 0.1 \ * S_1 + 0.3 * S_2 = S_2, \ (S_1 + S_2 = 1)$
  - 연립방정식을 풀면 $S_1 = \frac{7}{8} = 0.875, S_2 = \frac{1}{8} = 0.125$

$$S_{10} = (0.8749999309 \quad 0.12500000691)$$

But, 전이가 되어, mixing되어도 근접한 표본들은 correlated.

Not i.i.d.

❖ Markov Chain



- Markov chain sample은 i.i.d.가 아님
  - 하나의 Chain이 Mixing된 후
    t시점에서의 표본과 t-1, t+1등 가까운 시점에서의 표본은 correlated
  - 따라서 Mixing된 후의 표본들을 모아보면 i.i.d.라고 할 수 없다
    (엄밀하게는 independent가 아니기 때문일 것)
  - i.i.d.표본을 사용하는 모형에 사용할 수 없게 됨
  - 물론, Mixing 전의 표본을 사용하는 것 보다는 훨씬 낫다

- 해결 방법
  - 빠른 iteration(time)내로 Mixing을 하면 충분히 많은 횟수 뒤에서는
    correlate를 무시할 정도로 independent일 것이라 가정
  - 여러 초기값을 이용(Monte Carlo 방법 이용)해서
    여러 chain의 정적분포에서 표본추출

01
02
03
04

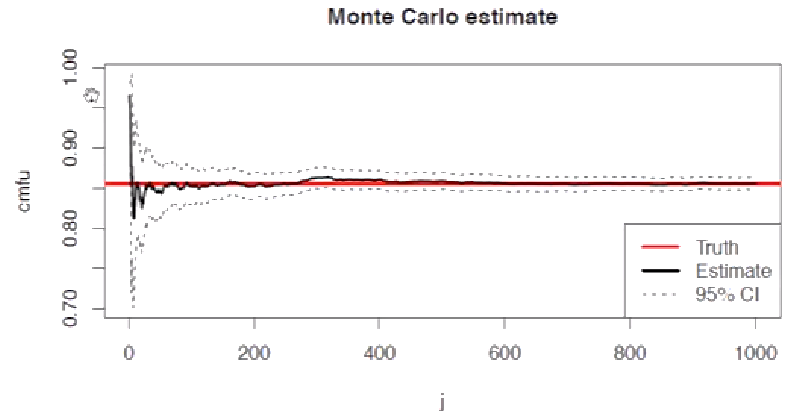
❖ Monte Carol

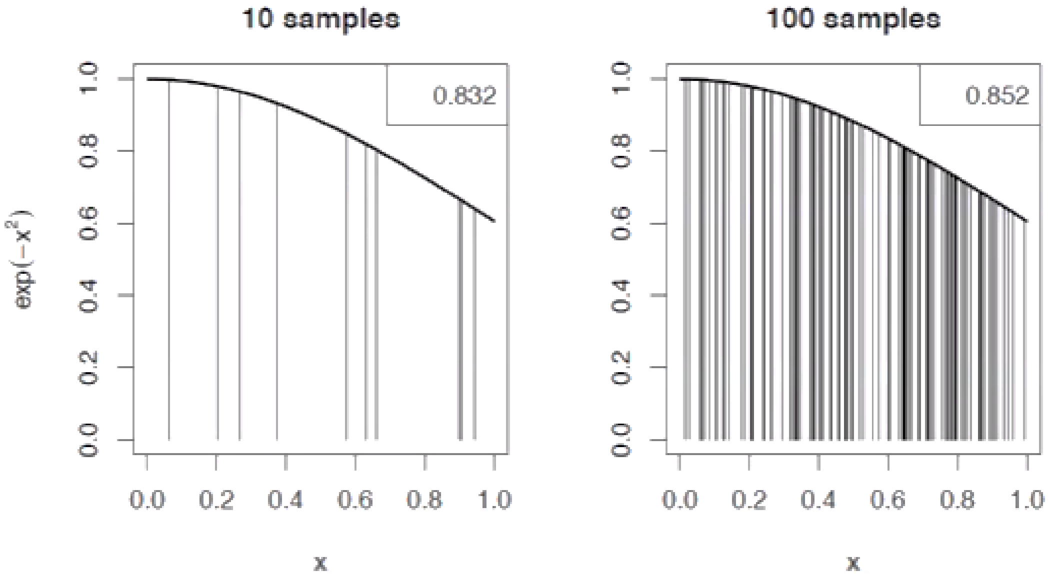
- 정의
  - 계산하려는 값이 복잡한 경우에 난수를 이용하여 함수의 값을 확률적으로 (근사적으로) 계산하는 알고리즘
- 식
  - $\boldsymbol{E_f[h(x)]} = \int_{\boldsymbol{x}} \boldsymbol{h(x)f(x)dx} \ (= \boldsymbol{h_J})$ (계산하고 싶은 것)
  - 함수 f의 조건에서 h(x)라는 값의 모든 x에 대한 기대 값
  - 통계적 시각으로 보면 h(x)는 확률변수, f는 확률 분포라고 생각
  - $\boldsymbol{\widehat{h_J}} = \frac{1}{J}\sum_{j=1}^{J} \boldsymbol{h(x_j)}$ (몬테카를로 방법을 이용한 추정치, 근사치)
  - $x_j \sim f(x), i.i.d.$ (independent and identically distributed)
  - 대수의법칙(SLLN), 중심 극한 정리(CLT)에 의해 $\boldsymbol{\widehat{h_J}} \rightarrow \boldsymbol{N}\left(E_f[h(x)], v_J\right)$ $(v_J = \frac{1}{J}V_f\widehat{[h(x)]} = \frac{1}{J^2}\sum_{j=1}^{J}[h(x_j) - \boldsymbol{\widehat{h_J}}]^2)$

01
02
03
04

❖ Monte Carol

- 예제
  - $\int_0^1 e^{-\frac{x^2}{2}}dx$ 를 구하고자 한다
  - $h(x) = e^{-\frac{x^2}{2}}$
  - $f(x) = 1,\ X \sim Unif(0,1)$
  - $\widehat{\boldsymbol{h}_J} = \frac{\mathbf{1}}{J}\sum_{j=1}^{J} \boldsymbol{h}(\boldsymbol{x}_j)$

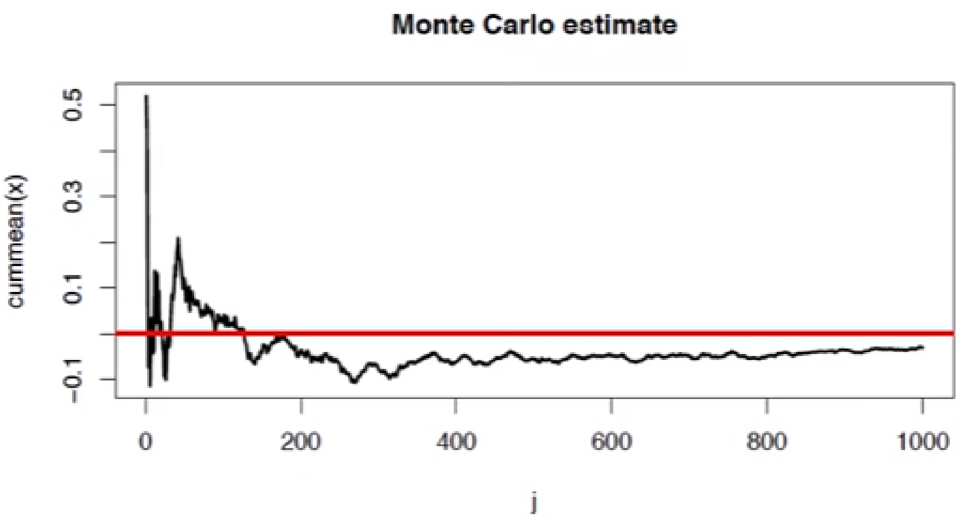




❖ Monte Carol

- 예제

  - $\int_{-\infty}^{\infty} x \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$ 를 구하고자 한다
  - $h(x) = x$
  - $f(x) = \phi(x),\ x \sim N(0,1)$
  - x가 정규분포의 평균 근처에서 많이 sampling되며 그때의 h(x)를 평균 내어 근사치를 구한다는 개념
  - $\widehat{\boldsymbol{h}_J} = \frac{1}{J}\sum_{j=1}^{J} \boldsymbol{h}(\boldsymbol{x}_j)$



01
02
03
04

❖ Monte Carol

- Bound
  - Hoeffding Bound

$$P_D\left(\widehat{h_J} \notin \left[h_J - \epsilon, h_J + \epsilon\right]\right) \leq 2e^{-2M\epsilon^2}$$

D : sample

$\widehat{h_J}$: estimator

h_J : true

M : number of sample

ϵ : positive real number

$2e^{-2M\epsilon^2} < \delta$ 라는 δ(허용 오차 가능 정도)를 정의하면

$\mathrm{M} \geq \frac{\ln\left(\frac{2}{\delta}\right)}{2\epsilon^2}$으로 표본의 개수를 구할 수 있다.

  - Chernoff Bound

$$P_D\left(\widehat{h_J} \notin \left[h_J(1-\epsilon), h_J(1+\epsilon)\right]\right) \leq 2e^{-\frac{Mh_J\epsilon^2}{3}}$$

$$\mathrm{M} \geq 3\frac{\ln\left(\frac{2}{\delta}\right)}{h_J\epsilon^2}$$

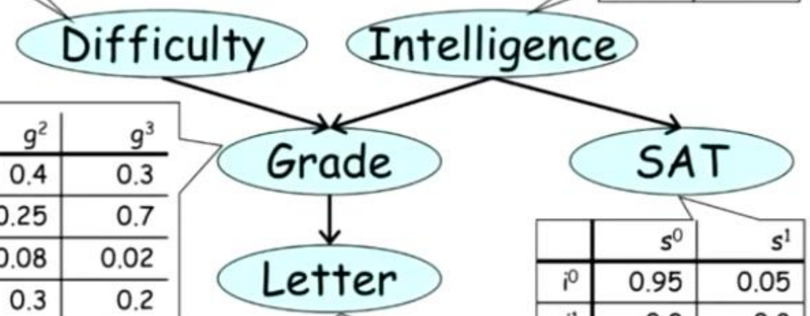
MCMC

❖ Forward sampling from a BN

- 주어진 Network에서 sampling을 하고 싶음

d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3



	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

  - 주어진 분포에 근거
  - $(d_0, i_0, g_2, s_0, l_1)$
  - $(d_1, i_0, g_3, s_0, l_0)$
  - …

- Goal : estimate $P(Y = y \mid E = e)$

❖ Rejection sampling

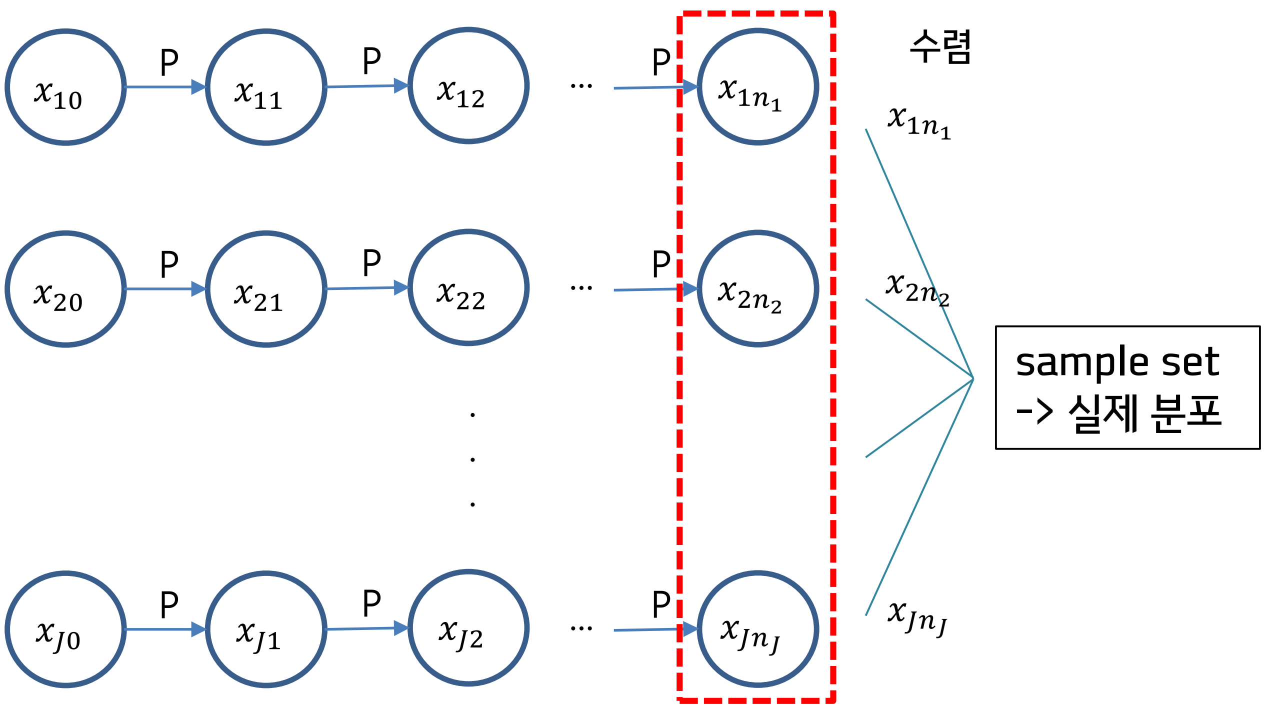
- Goal : estimate $P(Y = y \mid E = e)$
  - BN으로부터 표본을 생성
  - E=e의 조건이 맞지 않는 표본은 버림
  - $P(Y = y)$를 계산

- 단점
  - 표본의 결과가 너무 적음
    - Naïve Bayesian classifier와 같이 남성, 175cm, 73kg, 시력 1.1, 남동생 있음, 서울 거주, 고려대학교 재학, 25세, 옆구리에 점이 3개가 삼각형 모양으로 있음
    - 이런 조건을 만족하는 표본이 아니라면 사용 불가능
    - 표본이 많을 수 없음
  - 즉, 주어진 조건에 해당하는 변수의 개수가 많아짐에 따라 필요한 표본의 개수가 exponentially 증가

❖ MCMC

- Markov Chain + Monte Carlo(초기값 다양성 및 수학적 근거 확보)
  - 목표 확률분포(Target Probability Distribution)로부터 표본을 얻는 방법
  - Markov Chain의 전이확률 P를 MCMC 기법들을 통해 mixing되도록 설정, 여러 초기값 설정으로 i.i.d.를 확보한 뒤 각각의 chain에서 표본을 얻는다.

전이행렬(P)를 아는 경우



모르는 경우

(보통의 경우이다)

What?
P를 구한다.

How?
- Gibbs sampling
- Metropolis Hastings

❖ MCMC

- 장점
  - 사용하기 쉽다
  - 초기치에 상관없이 정적분포로 수렴하는 수학적 배경이 있다.

- 단점
  - 파라미터가 많이 필요하다.
    - 언제 Mixing 되었다고 할 수 있는가(stopping rule)
    - 몇 개의 iteration을 진행할 것인가
    - 몇 개의 표본을 추출할 것인가
  - 꽤 느리게 수렴할 수 있다.
  - Chain이 작동하는 것을 설명하기가 어렵다.

Gibbs sampling

01
02
03
04

❖ Gibbs sampling

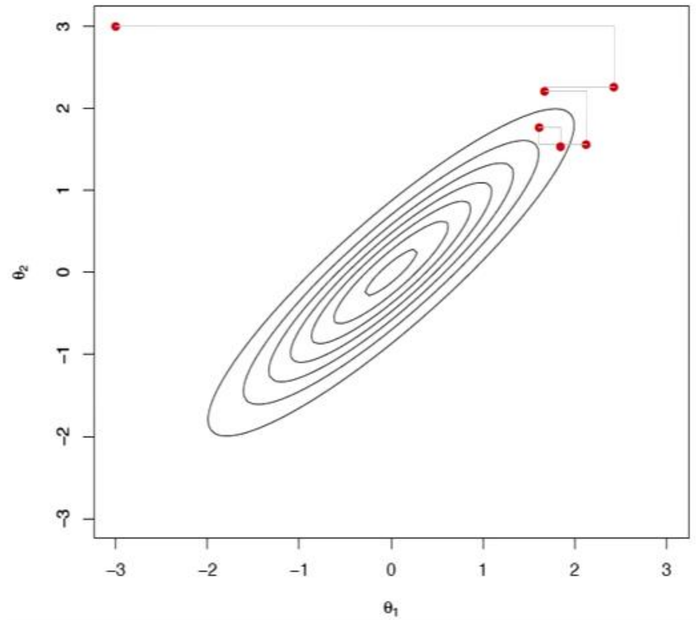
- 정의
  - 두개 이상의 확률 변수의 결합 확률 분포로부터 일련의 표본을 생성하는 확률적 알고리즘
  - Metropolis Hastings Algorithm의 특별한 예

- Target distribution
  - $P_\Phi(X_1, \dots, X_n)$

- Idea
  - $P_\Phi(X_1|X_{-1}) = P_\Phi(X_1|X_2, \dots, X_n)$
  - 초기값 x벡터 임의로 할당
  - for (i in 1:n){

    sample $x_i \sim P_\Phi(X_i|X_{-i})$

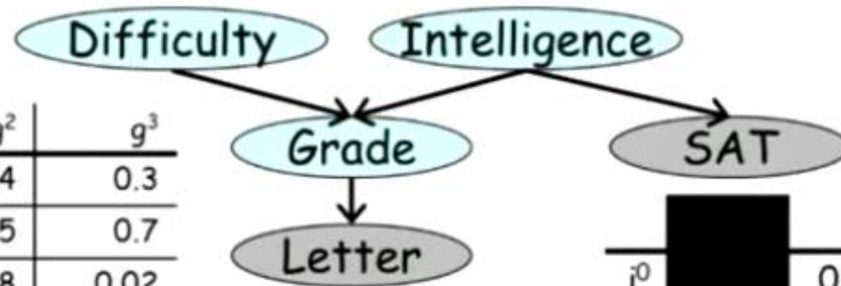
    }
  - Set $x_{new} \leftarrow x_{old}$



Gibbs sampling

❖ Gibbs sampling

- 예제



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	l^0	[redacted]
g^1	0.1	[redacted]
g^2	0.4	[redacted]
g^3	0.99	[redacted]

	[redacted]	s^1
i^0	[redacted]	0.05
i^1	[redacted]	0.8

- $\widetilde{P_\Phi}(D, I, G \mid l_0, s_1)$에서 표본 추출
- 초기값 임의로 d_0, i_0, g_1설정
- $P_\Phi(D \mid i_0, g_1, l_0, s_1)$에서 d_1추출
- $P_\Phi(I \mid d_1, g_1, l_0, s_1)$에서 i_1 추출
- $P_\Phi(G \mid d_1, i_1, l_0, s_1)$에서 g_3 추출
  - d_1, i_1, g_3, l_0, s_1이 하나의 표본이 됨
  - 이를 통해 다음 표본을 계속해서 뽑아 냄.(MCMC의 한 예로 볼 수 있다.)

Gibbs sampling

❖ Gibbs sampling

- 장점(계산상 이득)
  - for (i in 1:n){

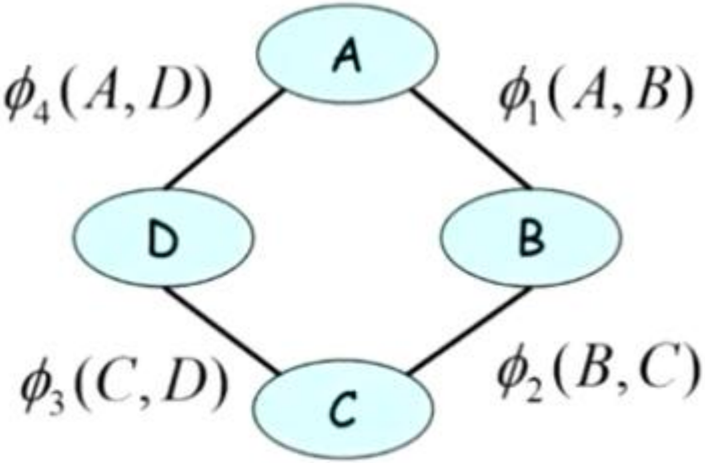
    sample $x_i \sim P_\Phi(X_i|X_{-i})$
  - $P_\Phi(X_i|X_{-i}) = \frac{P_\Phi(X_i,X_{-i})}{P_\Phi(X_{-i})} = \frac{\widetilde{P_\Phi}(X_i,X_{-i})}{\widetilde{P_\Phi}(X_{-i})}$ Network에서 Factor로 생각

    $$\propto \prod_{j:X_i \in Scope[C_j]} \phi_j(X_i, x_{j,-i})$$
  - $P_\Phi(A|b,c,d)$

$$= \frac{\widetilde{P_\Phi}(A,b,c,d)}{\sum_{A'} \widetilde{P_\Phi}(A',b,c,d)}$$

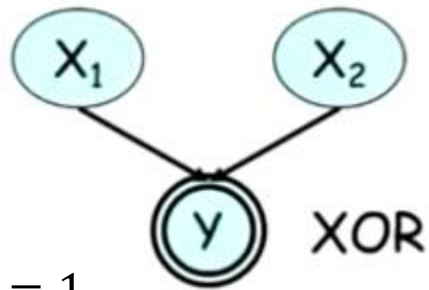
$$= \frac{\phi_1(A,b)\cancel{\phi_2(b,c)}\cancel{\phi_3(c,d)}\phi_4(A,d)}{\sum_{A'} \phi_1(A',b)\cancel{\phi_2(b,c)}\cancel{\phi_3(c,d)}\phi_4(A',d)}$$

$$\propto \phi_1(A,b)\phi_4(A,d)$$



Gibbs sampling

❖ XOR 문제



X_1	X_2	Y	Prob
0	0	0	0.25
0	1	1	0.25
1	0	1	0.25
1	1	0	0.25

- Y=1이 주어진 상황
  - $x_1 = 0, x_2 = 1$ 초기 할당
  - $P(X_1 = 0|x_2 = 1, y = 1) = 1$
    $P(X_1 = 1|x_2 = 1, y = 1) = 0$ => $x_1 = 0$ 할당
  - $P(X_2 = 0|x_1 = 0, y = 1) = 0$
    $P(X_2 = 1|x_1 = 0, y = 1) = 1$ => $x_2 = 1$ 할당
  - 앞으로 수차례 진행해도 이 결과가 바뀌지 않음 (0, 1, 1)
  - 초기할당이 다른 경우에는 다른 표본이 나오지만 그 또한
    다른 상황으로 변할 여지가 없음
  - 이는 Non-mixing chain이다. 다른 경우가 나올 확률이 0 (Non-regular)
  - 즉, 정적분포로 수렴하지 않으므로 표본추출이 용이하지 않다.
- Factor에 noise추가로 regular로 만들 수 있지만 좋은 방법은 아니다.
- 위 방식으로 regular를 만들어줘도, mixing까지 속도는 느리다.

Gibbs sampling

❖ Summary

- 특징
  - Joint distribution에서 어려운 sampling 문제를 보다 계산이 적고 쉬운 conditional distribution에서의 sampling 문제로 변환

- 장점
  - MCMC의 가장 쉬운 예
  - 계산의 이득을 많이 취함

- 단점
  - Mixing까지 오래가는 경우가 종종 있음
  - Factor의 곱으로부터 sampling 할 수 있는 경우에 한해서 적용 가능

Metropolis Hastings Algorithm

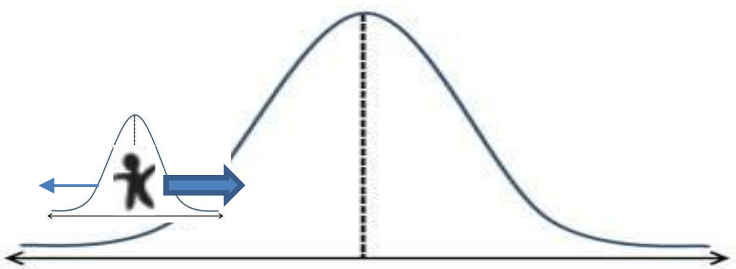
❖ Metropolis Hastings Algorithm

- 정의
  - 직접적으로 표본을 얻기 어려운 확률 분포로부터 표본의 수열을 생성하는데 사용하는 기각 표본 추출 알고리즘
  - 일반적인 적용에는 제약이 있지만 보통 더욱 빠르고 사용하기 쉽다.

- Idea
  - q를 이용해 샘플링을 진행하고, a로 방향성을 체크한 뒤 최종적으로는 f에 근사
    - $f(x)$: target density (정적 분포에 비례하게 설정. 굳이 정규화할 필요 없다)
    - x_j : current value
    - $q(x|x_j)$: proposal distribution(아무 분포나 가능, 보통 대칭형 사용)
  - Sample $x^* \sim q(x|x_j)$
  - Acceptance probability $a(x_j, x^*) = \min\left\{1, \frac{f(x^*)}{f(x_j)} \frac{q(x_j|x^*)}{q(x^*|x_j)}\right\}$

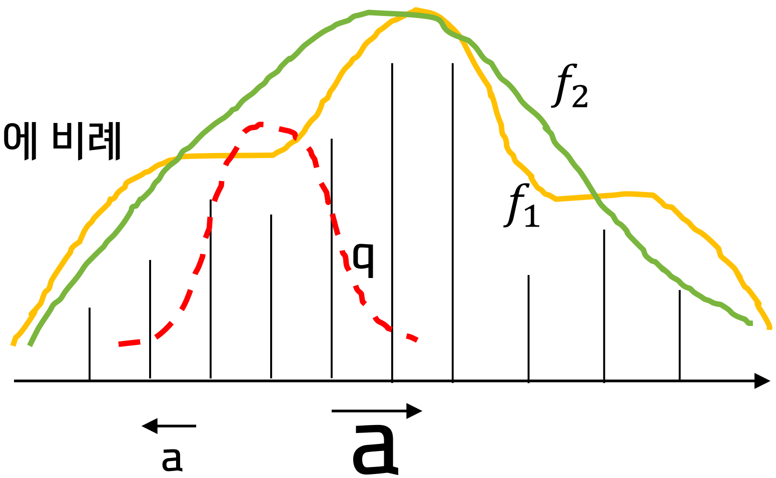
    x^*가 x_j보다 그럴 듯 한지 확인
  - Set $x_{j+1} = x^*$ with probability $a(x_j, x^*)$, o.w. $x_{j+1} = x_j$
  - $x_j \rightarrow X$ where $X \sim f(x)$

- x_j는 i.i.d.가 아니다.



❖ Metropolis Hastings Algorithm

- 도식화
  - f : target density
    - 주어진 Stationary distribution에 비례
      - » 정확한 분포를 알지 못함
    - 가정이 필요
      - » 최대한 형태를 보존
      - » 아는 분포로 근사



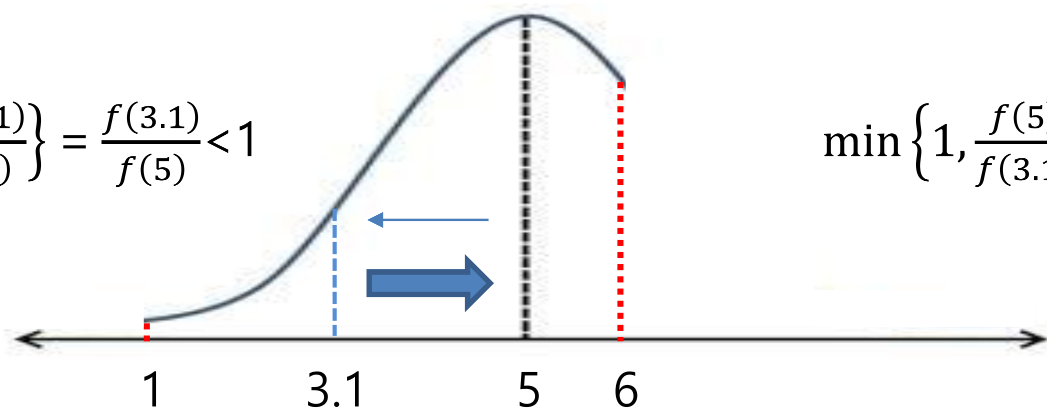
  - q : proposal distribution
    - f를 잘 찾아가는 것이 목표
    - 가정이 필요
      - » 보통 symmetric한 분포를 가정(정규분포 등)
    - 찾아가는 단계에서 표본이 나오게 되며, 이를 모아보면 f의 형태를 따라가게 됨
    - 표본은 q에서 파라미터를 update하는 역할
  - a : acceptance probability
    - q가 여기저기 움직이며 f를 찾아다닐 때, 정확히 움직이는 것인지 아닌지 판단
- q와 a의 조합(곱)이 우리가 찾고자 하는 전이행렬이다.

Metropolis Hastings Algorithm

❖ Metropolis Hastings Algorithm

- 예) 정규분포 찾아가기 (가정 : $q(x|x_j) = q(x_j|x)$, q : symmetric)
  - $a(x_j, x^*) = \min\left\{1, \frac{f(x^*)}{f(x_j)}\frac{q(x_j|x^*)}{q(x^*|x_j)}\right\} = \min\left\{1, \frac{f(x^*)}{f(x_j)}\right\}$
  - $f(x) = N(5,9)I(1 \le x \le 6), i.e.\ f(x) \propto \exp\left(-\frac{(x-5)^2}{18}\right)I(1 \le x \le 6)$
  - $q(x|x_j) = N(x_j, 1)$, $x_0 = 5$, q로부터 표본 3.1이라 하자.

$\min\left\{1, \frac{f(3.1)}{f(5)}\right\} = \frac{f(3.1)}{f(5)} < 1$

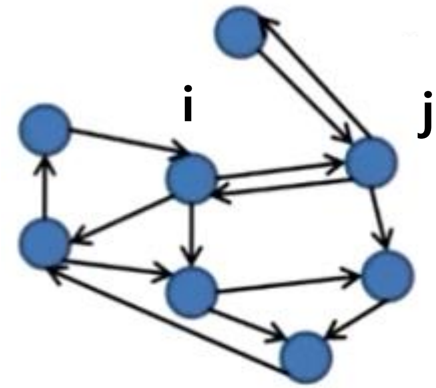


$\min\left\{1, \frac{f(5)}{f(3.1)}\right\} = 1$

Metropolis Hastings Algorithm

❖ Metropolis Hastings Algorithm

- Reversible chain
  - π : Stationary distribution, T : 전이행렬
  - $\pi(i)T(i \rightarrow j) = \pi(j)T(j \rightarrow i)$
  - 모든 i, j에 대해 성립할 경우, Detailed balance 라고 한다.



- Theorem
  - Chain이 detailed balance이고 T가 regular면 T는 유일한 정적분포 π를 갖는다. -> 즉, detailed balance인 T를 찾자.

$$\sum_i \pi(i)T(i \rightarrow j) = \sum_i \pi(j)T(j \rightarrow i) = \pi(j)\sum_i T(j \rightarrow i) = \pi(j)$$

[Stationary distribution의 정의, 정의를 다른 방식으로 보는 관점 제공]

예)

$\sum_i \pi(i)T(i \rightarrow 2)$

$= \pi(2)T(2 \rightarrow 1) + \pi(2)T(2 \rightarrow 2) + \pi(2)T(2 \rightarrow 3)$

$= \pi(2)[T(2 \rightarrow 1) + T(2 \rightarrow 2) + T(2 \rightarrow 3)]$

$= \pi(2)$

to

from

	1	2	3
1			
2		$T(i \rightarrow j)$	
3			

Metropolis Hastings Algorithm

❖ Metropolis Hastings Chain

- Proposal distribution(q)
  - $Q(x \rightarrow x')$
- Acceptance probability(a)
  - $a(x \rightarrow x')$
- 각 상태 x에서 x’로 가는 상황을 Q에 기반해서 결정
- a가 수락을 의미하는 경우, 다음단계의 상태를 x’로 결정해준다.

  그렇지 않다면 다음단계의 상태를 계속 x로 유지한다.
- 즉,

  $$T(x \rightarrow x') = Q(x \rightarrow x')a(x \rightarrow x') \;\; if \; x' \neq x$$

  $$T(x \rightarrow x) = Q(x \rightarrow x) + \sum_{x' \neq x} Q(x \rightarrow x')[1 - a(x \rightarrow x')]$$

Metropolis Hastings Algorithm

❖ Acceptance probability(a)

- 만족시킬 조건 : Detailed balance

$$\pi(x)T(x \to x') = \pi(x')T(x' \to x)$$

(a는 detailed balance를 유지하는 장치)

$$\pi(x)Q(x \to x')a(x \to x') = \pi(x')Q(x' \to x)a(x' \to x)$$

$$\frac{a(x \to x')}{a(x' \to x)} = \frac{\pi(x')Q(x' \to x)}{\pi(x)Q(x \to x')} = \rho < 1$$

$$\to a(x \to x') = \rho, a(x' \to x) = 1$$

$$a(x \to x') = \min\left[1, \frac{\pi(x')Q(x' \to x)}{\pi(x)Q(x \to x')}\right]$$

Metropolis Hastings Algorithm

❖ Proposal distribution(Q)

- Q는 a 안에서 비율의 형태로 존재한다.

  $$a(x \to x') = \min\left[1, \frac{\pi(x')Q(x' \to x)}{\pi(x)Q(x \to x')}\right]$$

  Q는 0이 되어서는 안된다. $Q(x \to x') > 0, Q(x' \to x) > 0$

- Q는 Mixing을 이루기 위해 전체를 훑어볼 수 있어야 한다.
- 하지만 너무 광범위하게 탐색하는 경우 a가 정말 낮은 값을 갖게 될 수 있다.

Metropolis Hastings Algorithm

❖ Gibbs sampling and Metropolis Hastings Chain

- 성능 비교
  - MH의 경우 Gibbs보다 더 잘 Mixing되는 것을 확인 가능
  - MH도 사용한 Q에 따라 성능 차이가 있음