DSL Seminar: MCMC (4)

Kyung-han Kim

Data Science Lab

February, 2023

목차

- Metropolis-Hastings Algorithm 복습하기
- R로 Metropolis-Hastings algorithm 구현하기
  - 베이즈통계 2차 과제 Q1
  - 통계전산 7차 과제 Q2
- 과제 안내
  - 통계전산 7차 과제 Q1

Markov Chain Monte Carlo (MCMC)

- MCMC는 어떤 irreducible aperiodic Markov Chain의 stationary distribution π로부터 random sample을 만들어 내는 기법입니다.
- Markov chain에서 한 시점의 값은 직전 시점의 값에 영향을 받기 때문에 이러한 방식은 더이상 iid sampling이 아니고,
초기 sample들은 initial state의 영향을 강하게 받습니다.
- 그렇지만 t가 커지면서 Markov chain이 π에 수렴해 가면,
그때쯤부터는 MCMC를 통해 얻은 sample이 π에서 sampling된 것이라고 간주할 수 있습니다.
- 실제로 사용할 때는, 우리가 sample을 만들어 내기를 원하는 확률분포인 target distribution을 π라고 간주하고 MCMC를 사용합니다.

MCMC의 사용 목적

- 두 수업 모두에서 MCMC를 사용하는 기본 목적은 동일합니다.
- MCMC는 iid sampling이 불가능할 정도로 복잡한 확률분포에서 표본을 얻어내기 위해 사용하는 기법입니다.
- 통계전산에서 target distribution은 iid samling이 까다로운 모든 확률분포를 대상으로 합니다.
- 베이즈통계에서 target distribution은 주로 posterior distribution입니다.
  - (Posterior) $\propto$ (Prior) $\times$ (Likelihood)

M-H Algorithm: step-by-step (1)

1. Initialize $t = 0, X^{(0)} = x_0 \in \mathcal{X}$.
   - Markov chain을 결정짓기 위한 조건 중 하나가 Initial value $(X^{(0)})$ 입니다.
   - 시점을 0으로 초기화하고, 초깃값을 설정하는 단계입니다.
   - t를 계속 증가시킬 경우 초깃값은 최종 결과에 큰 영향을 미치지 않지만, 되도록 reasonable하게 정해주는 것이 좋습니다.

M-H Algorithm: step-by-step (2) (Continuous)

1. Initialize $t = 0, X^{(0)} = x_0 \in \mathcal{X}$.
2. Sample $X' \sim q(X'|X^{(t)})$.

- $q(X'|X^{(t)})$: 평균이 $X^{(t)}$인 proposal density q에서 X'를 생성한다.
- Proposal density는 기본적으로 임의로 설정할 수 있습니다.
  주로 정규분포를 많이 사용합니다.
  - 평균은 $X^{(t)}$라고 치자. 분산은? (acceptance rate와 관련 있음)
- Proposal density가 symmetric하면 계산이 용이합니다.
- 우리가 원하는 target distribution에서는 iid sample을 만들기 힘드니, 보다 쉽게 난수를 생성할 수 있는 다른 확률분포 (proposal) 에서 **후보군**을 만들어 낸 다음에, 그 중 적합한 것들만 골라 내서 사용하려는 목적입니다!
- Target distribution을 따르는 모수에 제약조건이 있다면 그 조건에 맞게 sampling을 해 줘야 합니다.
  - Ex] 무조건 양수여야 한다.

M-H Algorithm: step-by-step (3)

1. Initialize $t = 0, X^{(0)} = x_0 \in \mathcal{X}$.
2. Sample $X' \sim q(X'|X^{(t)})$.
3. Sample $U \sim U(0, 1)$
   - Metropolis-Hastings algorithm에는 다음 시점이 될 만한 값을 생성해 놓고, 그 값을 **사용할 것인지 여부를 결정**하는 단계가 있습니다.
   - 이 U는 해당 단계에서 사용하기 위해 미리 생성하는 값입니다.
   - 다음 단계에서 U와 threshold 값을 비교해서 무엇이 더 큰지에 따라 X'를 사용할지 말지 정합니다.

M-H Algorithm: step-by-step (4)

1. Initialize $t = 0, X^{(0)} = x_0 \in \mathcal{X}$.
2. Sample $X' \sim q(X'|X^{(t)})$.
3. Sample $U \sim U(0,1)$
4. If $U < \frac{\pi(X')q(X^{(t)}|X')}{\pi(X^{(t)})q(X'|X^{(t)})}$, then $X^{(t+1)} = X'$. else, $X^{(t+1)} = X^{(t)}$
   - $\alpha = \frac{\pi(X')q(X^{(t)}|X')}{\pi(X^{(t)})q(X'|X^{(t)})}$가 U와 비교되는 threshold 값입니다.
   - 만약 X'가 $X^{(t)}$보다 그럴싸하다면, $\pi(X') > \pi(X^{(t)})$,
     다시 말해, pmf 값이 $x = X'$일 때가 $x = X^{(t)}$일 때보다 클 것입니다.
     이 경우 $\alpha > 1$을 기대할 수 있습니다.
   - 반대의 경우 $\alpha < 1$일 가능성이 클 것입니다.
   - 이번 단계의 목적은 min$(1, \alpha)$의 확률로 X'를 채택하는 것입니다.
     - $\alpha > 1 \to$ min$(1, \alpha) = 1$: 무조건 사용
     - $\alpha < 1 \to$ min$(1, \alpha) = \alpha$: α의 확률로 사용 ($1 - \alpha$: 기존 값 유지)
   - $P(U < \alpha) =$ min$(1, \alpha)$이므로 U와 α를 비교해 위의 목적을 달성!
   - q는 두 가지 목적으로 α 계산식에 포함됩니다. (symmetric이면 약분)
     - 1) Detailed Balance Condition을 만족시키기 위해
     - 2) q가 asymmetric한 경우 그러한 제약사항을 보정해 주기 위해

M-H Algorithm: step-by-step (5)

1. Initialize $t = 0, X^{(0)} = x_0 \in \mathcal{X}$.
2. Sample $X' \sim q(X'|X^{(t)})$.
3. Sample $U \sim U(0,1)$
4. If $U < \frac{\pi(X')q(X^{(t)}|X')}{\pi(X^{(t)})q(X'|X^{(t)})}$, then $X^{(t+1)} = X'$. else, $X^{(t+1)} = X^{(t)}$
5. $t = t + 1$, Go to 2.
   - 다음 시점의 값이 결정되었으므로 시점 (t)을 1 늘려 주고 다시 위의 절차를 반복해 또다른 sample을 얻습니다.
   - 초깃값은 다시 지정해 줄 필요가 없으므로 2로 돌아갑니다.

M-H Algorithm: 단계별 의미

1. 초깃값을 설정한다.
2. 다음 sample이 될 후보 값을 뽑는다.
3. 다음 sample을 사용할 확률과 비교할 U 값을 뽑는다.
4. 후보 값을 사용할 확률 α와 U를 비교해 사용 여부를 정한다.
   사용하기로 나왔다면 다음 시점 값은 후보 값이 되고,
   아니라면 이전 시점 값을 그대로 다음 시점에도 사용한다.
5. 시점을 1 늘리고 다시 돌아가 절차를 반복한다.

MCMC 수렴 여부 확인하기

- 앞서 언급한 절차대로 MCMC를 잘 구성하더라도, 그 결과물로 얻게 되는 Markov chain 값 (=random sample from $\pi(\cdot)$) 들은 $\pi(\cdot)$에 수렴하지 않을 수도 있습니다.
- 그래서 MCMC로 sample을 만들어 낸 후에는,
  해당 sample이 내가 원하는 target distribution π에 잘 수렴했는지,
  수렴했다면 언제쯤부터 수렴하는지를 직접 확인하는 절차가 필요합니다.
- 과제에서는 주로 아래의 내용들을 확인하기를 요구합니다:
  - Trace plot (Time series plot)
  - Histogram
  - Auto-correlation plot
  - Acceptance probability
  - Effective Sample Size (ESS)
- 베이즈통계에서는 Sampling이 성공적이었다는 전제 하에,
  아래의 내용을 추가로 확인해 보기를 요구하기도 합니다.
  - Posterior mean
  - 95% HPD interval

(1) Trace plot (시계열그림)

- Trace plot을 통해 시간에 따른 sample 값의 변화를 확인합니다.
- MCMC가 잘 수렴했다면, 일정 시점 이후의 sample들은 특정 값 주위에서 큰 변동 없이 움직여야 합니다.
- ts.plot() 함수를 이용합니다.

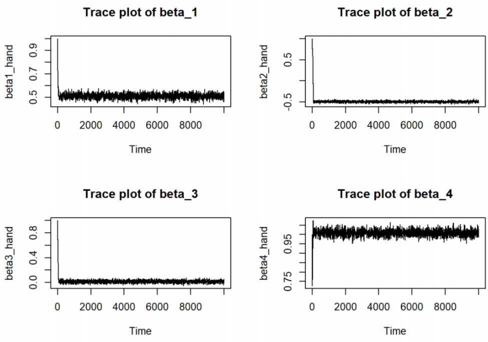


Figure 1: Example of well-converged trace plot

(1)-1 Burn-in period

- 앞선 trace plot에서 볼 수 있듯이, MCMC가 수렴하는 데까지는 일정 정도의 iteration (time)이 필요합니다.
  수렴하기 이전의 sample들은 우리의 target distribution π에서 나왔다고 보기 어렵습니다.
- 따라서 수렴 이전의 sample은 사용하는 것이 부적절하다고 판단, 만들어진 MCMC sample의 앞부분 일부를 제거하는 과정이 필요합니다.
- 이 과정을 **Burn-in**이라고 하며, MCMC에서 필수적인 요소입니다!
- 아래의 경우 감점 사유에 해당될 수 있습니다:
  - 통계전산 과제에서 burn-in을 사용하라고 했음에도 누락하는 경우
  - 베이즈통계 과제에서는 따로 언급이 없더라도 burn-in을 하는 것을 권장 (불시에 감점될 수 있음)
  - burn-in을 적용했더라도 burn-in period가 reasonable하지 않은 경우
  - Number of Iteration이 불충분한 경우 (권장: n=10,000)

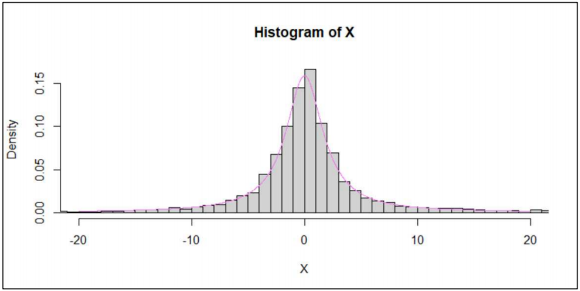
(1)-2 Thinning

- 의미: 전체 sample에서 일정 간격을 정해,
  해당 번째수에 해당하는 sample만 사용하는 것
- Ex] 총 5,000개의 sample에 thinning=5를 적용하면,
  $x_1, x_2, \cdots, x_{4999}, x_{5000}$ $(n = 5000)$을 모두 사용하는 것이 아니라,
  $x_5, x_{10}, x_{15}, \cdots, x_{4995}, x_{5000}$ $(n = 1000)$만 사용한다.
- Sample 값 간의 상관관계를 줄이기 위해 사용합니다.
- 통계전산과 베이즈통계 중에서는 베이즈통계에서 더 비중 있게 다룹니다.
  통계전산 과제에서는 다루지 않습니다.
- 주의] Acceptance rate는 반드시 Thinning **이전**에 계산해야 합니다!

(2) Histogram (히스토그램)

- 히스토그램을 통해 MCMC sample이 target distribution과 분포가 일치하는지 확인할 수 있습니다. (hist() 함수를 사용)
  - 단, target distribution을 직접 그릴 수 있는 경우에 한해 요구합니다.

다음은 히스토그램을 그려 보겠습니다. 문제에서 주어진 코시분포의 pdf까지 겹쳐서 그려보겠습니다.



보라색으로 그려진 곡선이 코시분포입니다.
히스토그램이 pdf와 상당히 유사한 형태를 보이며 sampling이 이루어졌다는 것을 볼 수 있습니다.

Figure 2: Histogram with target density function

(3) Auto-correlation plot (자기상관그림)

- MCMC는 더이상 iid sampling이 아니기 때문에, sample들 간에 correlation이 있습니다.
- 이 correlation의 정도를 시각화해 주는 것이 Auto-correlation plot입니다.
- Autocorrelation은 빠르게 0에 가깝게 감소하는 것이 바람직합니다.

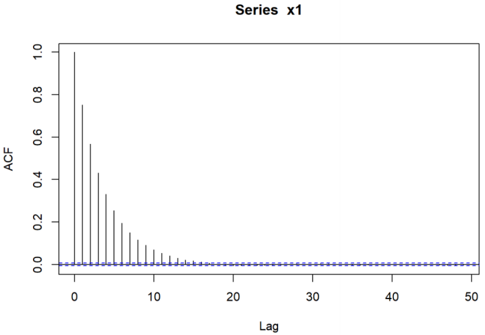


Figure 3: Example of Autocorrelation plot

(4) Acceptance Probability/Rate

- Metropolis-Hastings algorithm에서는 random sample이 만들어졌다고 해서 해당 sample을 무조건 사용하지는 않습니다.
- Threshold 값인 α를 π와 q로 계산해 새롭게 제안된 sample인 X'을 실제로 사용할 확률을 구하게 됩니다.
- 이 때 $X^{(t+1)} = X'$가 되면 이를 'accept'라고 합니다.
- Accpetance rate $= \frac{\text{Total number of accept}}{\text{Total number of Iteration}}$ 로 정의됩니다.
- R에서 acceptance rate를 구하는 방법에는 2가지가 있습니다:
  - Metropolis-Hastings algorithm 코드 내에서 accept 변수를 지정, X'가 accept될 때마다 accept $=$ accept $+$ 1.
    이 때 acceptance rate는 $\frac{\text{accept}}{\text{n_iter}}$가 된다.
  - length(unique(sample))/length(sample)로 구하기! (편해서 추천)
- 바람직한 acceptance rate는 20%에서 50% 사이입니다.
  Acceptance rate 100%는 결코 바람직하지 못합니다. 왜일까요?

(5) Effective Sample Size (ESS)

- MCMC를 이용한 표본 생성은 Dependent sampling입니다.
- Dependent sampling은 기본적으로 independent sampling보다 성능이 좋지 못합니다.
- 그렇다면 구체적으로 얼마나 성능이 떨어질까요?
- ESS는 현재 내가 가진 MCMC sample이, 길이가 어느 정도인 independent sample과 비슷한 효과를 내는지 알려 줍니다.
- Ex] MCMC로 sample을 10,000개 만들었는데 ESS=2,000 이다 = 내가 가진 MCMC sample은 iid sample 2,000개와 비슷하다.
- ACF 값이 클수록 ESS는 작아집니다.
- ESS < 100 인 경우 감점 사유입니다. (베이즈통계)

M-H Algorithm: R Implementation

- 지금부터 1시까지 R로 Metropolis-Hastings algorithm을 구현해 봅니다.
- 문제는 아래의 과제를 활용합니다:
  1. 베이즈통계 2차 과제 Q1
  2. 통계전산 7차 과제 Q2
- 질문이 있으시면 바로바로 해 주세요!
- 서로서로 자유롭게 대화하셔도 좋습니다!
- 두 문제를 모두 완료하셨을 경우 R파일을 올려 주세요!
- 오늘의 과제는 위의 두 문항과, 통계전산 7차 과제 Q1입니다.