2019-11-06

computational Statistics

HW#6

192STG11 우나영

computational Statistics

HW#6

1. Problem

적분 근사 방법은 크게 Numerical Integration 과 Monte Carlo(MC) Simulation 방법 두 가지가 있다. Numerical Integration 은 함수의 전 구간을 n 개의 grid 로 나눈 후 각각의 구간을 polynomial 함수로 근사하는 방법이다. p 차원 함수의 경우 Numerical Integration 에 필요한 grid 개수가 n^p개로 차원이 커짐에 따라 grid 의 개수가 기하급수적으로 증가한다. 따라서 Numerical Integration 은 차원이 높아짐에 따라 계산량이 많아져 오차가 커지기 때문에 수렴하지 않는다. 반면에 MC 는 함수의 likelihood 를 이용해 grid 를 잡기 때문에 모든 구간을 systematic 하게 grid 로 나누는 Numerical Integration 과 비교해 훨씬 효율적이다. 따라서 고차원에서는 Numerical Integration 대신 MC 를 사용할 수밖에 없다.

MC 는 target 함수에서 샘플을 무작위로 뽑아 통계량을 계산하는 기법이다. 예를 들면 $X_1, \dots, X_n$이 iid 로 f 라는 분포를 따를 때 MC 를 이용해 continuous 함수 h(x)의 expected value 를 구해보자.

$$X_1, \dots, X_n \sim f\ , iid\ , \quad \mu = E[h(x)] = \int h(x)f(x)dx$$

$$\widehat{\mu_{MC}} = \frac{1}{n}\sum h(x_i) \quad where\ n\ is\ the\ numer\ of\ sample\ , x_i\ is\ sample\ from\ dist^n\ f$$

$$then\ \widehat{\mu_{MC}} \overset{n\to\infty}{\longrightarrow} \mu\ (by\ SLLN, Strong\ Law\ of\ Large\ Number)$$

f 에서 랜덤하게 iid 로 sample size n 개를 뽑은 후 이를 h(x)에 대입한 값의 합을 n 으로 나눠 $\widehat{\mu_{MC}}$을 구한다. $\widehat{\mu_{MC}}$는 sample size n 이 커짐에 따라 SLLN 에 의해 참값으로 수렴한다. 하지만 이렇게 간편한 MC simulation 을 수행하기 위해서는 target 함수에서 샘플을 iid 로 뽑을 수 있어야 한다는 점이 전제되어야 한다. 이론적으로 iid 샘플을 생성하는 방법은 분포함수(CDF)의 역함수를 이용하는 것이다.

< inverse CDF thm >

$$For\ any\ continuous\ dist^n\ function\ F$$

$$\text{if u} \sim \text{unif(0,1)}\ \ then\ X = F^{-1}(u)\ has\ the\ dist^n\ function\ F$$

$$\text{if } F^{-1}\ has\ closed\ form\ and\ u_1, \dots, u_n \sim u(0,1)\ iid$$

$$\text{then } X_i = F^{-1}(u_i) \text{ which means } X_i \sim F \text{ iid}$$

하지만 대부분의 함수에서 F^{-1}가 closed form 으로 존재하지 않는다. 따라서 inverse CDF thm 에 따라 샘플링을 할 수 있는 분포는 거의 없다. 본 과제에서는 임의의 분포에서 iid 샘플을 뽑는 방법으로 Rejection Sampling 과 SIR(Sampling Importance Resampling)기법의 이론에 대한 소개하고 예제를 통해 이들을 구현해 보도록 하겠다.

1. Rejection Sampling

만약 target 함수를 f 라 할 때 f(x)를 정확하게 혹은 적어도 unknown 상수에 비례하는 정도까지 계산할 수 있다면 Rejection Sampling 을 이용해 정확한(exact) target 함수로부터 iid 샘플링 할 수 있다. Rejection Sampling 은 f 로부터 직접 샘플을 뽑는 대신 샘플을 뽑기 쉬운 분포 g 로부터 후보 샘플을 뽑은 후 $e(x) = \frac{g(x)}{\alpha} \geq f(x)\ for\ all\ x\ for\ f(x) > 0, given\ constant\ \alpha \leq 1$ 의 성질을 갖는 envelope 함수를 이용하여 f 분포를 따르지 않는 샘플을 reject 하는 방식으로 f 의 iid 샘플을 뽑는다. Rejection Sampling 의 과정은 아래와 같다.

1. Sample Y ~ g
2. Sample U ~ Unif(0,1)
3. Reject Y if U>f(Y)/e(Y). 이때 Y 는 target 함수 f 의 sample 이 될 수 없다. 따라서 다시 1 번으로 돌아간다.
4. Otherwise, keep the value of Y. Set X=Y. 그리고 X 는 target 함수 f 의 sample 이 된다. 필요한 만큼의 sample size 가 생길 때까지 1 번으로 돌아가 위와 같은 과정을 반복 수행한다.

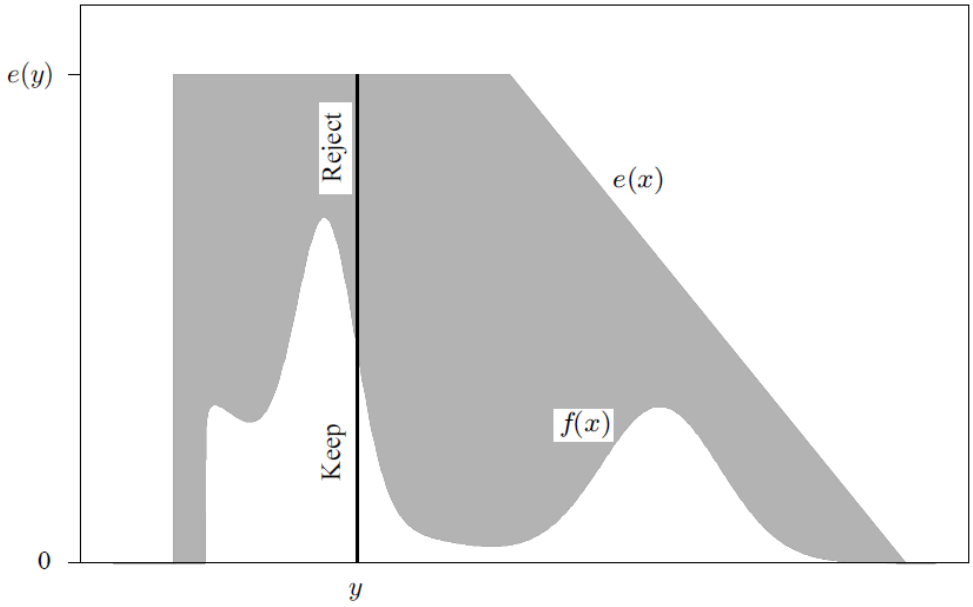


FIGURE 6.1 Illustration of rejection sampling for a target distribution f using a rejection sampling envelope e.

그림에서 알 수 있듯이 envelope 함수 e 는 모든 범위에서 f 보다 크거나 같은 함수로 f 를 감싸고 있기 때문에 봉투(envelope)함수라고 부른다. Envelope 함수와 target 함수의 높이 차이가 작을수록 accept 확률이 높아질 것이다. Envelope 이 $e(x) = \frac{g(x)}{\alpha} \geq f(x)$ 이기 때문에 α가 작을수록 envelope 이 target 함수보다 높아져 accept 확률이 낮아진다. 따라서 α 값은 expected proportion of acceptance 로 값이 클수록 accept 될 확률이 높아져 샘플이 빠르게 생성될 수 있다. Rejection Sampling 을 통해 구한 샘플이 실제 f 분포의 샘플인지 확인하기 위해 아래와 같은 증명을 할 수 있다.

$$\begin{aligned} \mathrm{P}(\mathrm{X} \leq \mathrm{y}) &= \mathrm{P}\left[\mathrm{Y} \leq \mathrm{y} | \mathrm{U} \leq \frac{f(y)}{e(y)}\right] \\ &= \mathrm{P}\left[\mathrm{Y} \leq \mathrm{y} \text{ and } \mathrm{U} \leq \frac{f(y)}{e(y)}\right] \Big/ \mathrm{P}\left[\mathrm{U} \leq \frac{f(y)}{e(y)}\right] \\ &= \int_{-\infty}^{y} \int_{0}^{f(z)/e(z)} du g(z) dz \Big/ \int_{-\infty}^{\infty} \int_{0}^{f(z)/e(z)} du g(z) dz \\ &= \int_{-\infty}^{y} f(z) dz \end{aligned}$$

앞서 언급했듯이 target 함수를 정확히 모르더라도 unknown 상수에 비례한다는 것까지 계산이 가능하다면 Rejection Sampling 을 이용할 수 있다. 특히 베이지안에서는 unnormalized posterior 의 샘플을 생성할 때 유용하다. Posterior 는 likelihood 와 prior 에 비례한다는 사실은 알려져 있지만 normalizing constant 를 계산의 어려움으로 인해 정확한 posterior 는 구하기 어렵다. 그러나 Rejection Sampling 을 이용한다면 정확한 posterior 를 구하지 않더라도 정확한 posterior 로부터 iid 샘플을 생성할 수 있다.

2. SIR(Sampling Importance Resampling)

iid 샘플을 뽑는 두 번째 방법은 SIR(Sampling Importance Resampling)이다. SIR 의 알고리즘은 아래와 같다. 참고로 f 는 target density, g 는 envelope density 그리고 $w^* = f(Y_i)/g(Y_i)$으로 unstandardized weight 이고 $w = \frac{f(Y_i)/g(Y_i)}{\Sigma f(Y_i)/g(Y_i)}$은 standardized weight 이다.

1. Sample $Y_1, \ldots Y_m \sim g \; iid \; where \; g \; is \; envelope$
2. Calculate standardized weight $\mathrm{w}(Y_1), \ldots, \mathrm{w}(Y_m)$
3. Resample $X_1, \ldots, X_n \; from \; Y_1, \ldots, Y_m$ with replacement and probability $\mathrm{w}(Y_1), \ldots, \mathrm{w}(Y_m)$

Then X_i from SIR $\xrightarrow{m \to \infty}$ follows $target \; function \; f$

SIR 의 기본 아이디어는 envelope 함수 g 로부터 생성한 m 개의 iid 샘플의 weight f/g 를 계산한다. 각각의 원소들이 뽑힐 확률이 Standardized weight 인 m 개의 iid 샘플에서 다시 n 개의 샘플을

복원 추출(with replacement)한다. Weight f/g 가 높을수록 f 하고 가까운 값들이기 때문에 resampling 될 확률이 높아진다. SIR 을 통해 생성된 X_i, i = 1, … , n은 m 이 커짐에 따라 X_i들의 분포가 target density f 로 근사(approximate)한다. SIR 과 관련된 이슈는 크게 두가지로 첫 번째 m 과 n 의 크기와 두번째 envelope 함수인 g 이다. 생성된 n 개의 샘플의 분포가 target density f 로 근사하기 위해 n 과 m 의 크기가 중요하다. 하지만 n/m 의 크기는 g 함수에 dependent 하기 때문에 더 중요한 것은 envelope 함수인 g 이다. g 가 f 와 비슷하다면 n/m 이 작지 않아도 되지만 다르다면 n/m 이 매우 작아야 한다. 그렇다면 g 는 어떤 함수여야 할지 알아보자. g 함수는 두 가지 조건을 따라야한다. 첫 번째는 support(g)≥ support(f)로 g 가 f 를 포함하고 있어야한다. 두 번째는 g 가 f 보다 분포의 꼬리가 더 두꺼워(heavier)야 한다. 즉 weight 인 f/g 가 너무 빨리 커져서는 안된다. 다시 말하면 분포의 꼬리에서 f 가 g 보다 density 가 클 경우 weight 인 f/g 가 기하급수적으로 커지기 때문에 resampling 된 샘플들이 f 의 꼬리부분에서 많이 나타나게 될 것이다.

SIR 는 샘플의 분포가 m 이 커짐에 따라 target density 에 approximation 인 반면 Rejection Sampling 는 샘플의 분포가 exact target density 이다. 그러나 SIR 는 n 개의 resampling 샘플을 뽑기 위해 사전에 정해진 m 개를 sampling 하면 되지만 Rejection Sampling 은 envelope 에 따라서 acceptance rate 가 달라지기 때문에 n 개의 샘플을 얻기 위해 필요한 후보 샘플의 사이즈가 random 이다. 아래는 SIR 과 Rejection Sampling 의 차이점을 표로 정리한 것이다.

	Rejection Sampling	SIR
sample size	dependent on acceptance rate n × acceptance rate where n is candidate sample size	n , predefined
approximation	exactly target function	approximate if m goes to ∞

2. Result

1.1 Rejection Sampling : example 6.1 Gamma(r, 1) Sampling

Y 가 f(y)에서 생성되었을 때 X=t(Y)는 Gamma(r,1) , r≥ 1분포를 갖는다고 한다.

$$f(y) = \frac{t(y)^{r-1}t'(y)exp\{-t(y)\}}{\Gamma(r)}$$

$$where\ t(y) = a(1+by)^3\ for -\frac{1}{b} < y < \infty, a = r - \frac{1}{3}, b = \frac{1}{\sqrt{9a}}$$

위의 문제를 Rejection Sampling 을 이용하여 Gamma(r, 1) where r≥ 1의 샘플을 뽑기 위해 아래와 같은 set up 이 선행된다.

$$Draw\ Z \sim N(0,1)\ which\ is\ easy\ to\ sample\ pdf\ and\ U \sim U(0,1)$$

$$target\ function\ to\ generate\ is\ proportional\ to\ q(y) = exp\{alog\left\{\frac{t(y)}{a}\right\} - t(y) + a\}$$

$$envelope\ function\ \mathrm{e(y)} = \exp\{\frac{-y^2}{2}\}$$

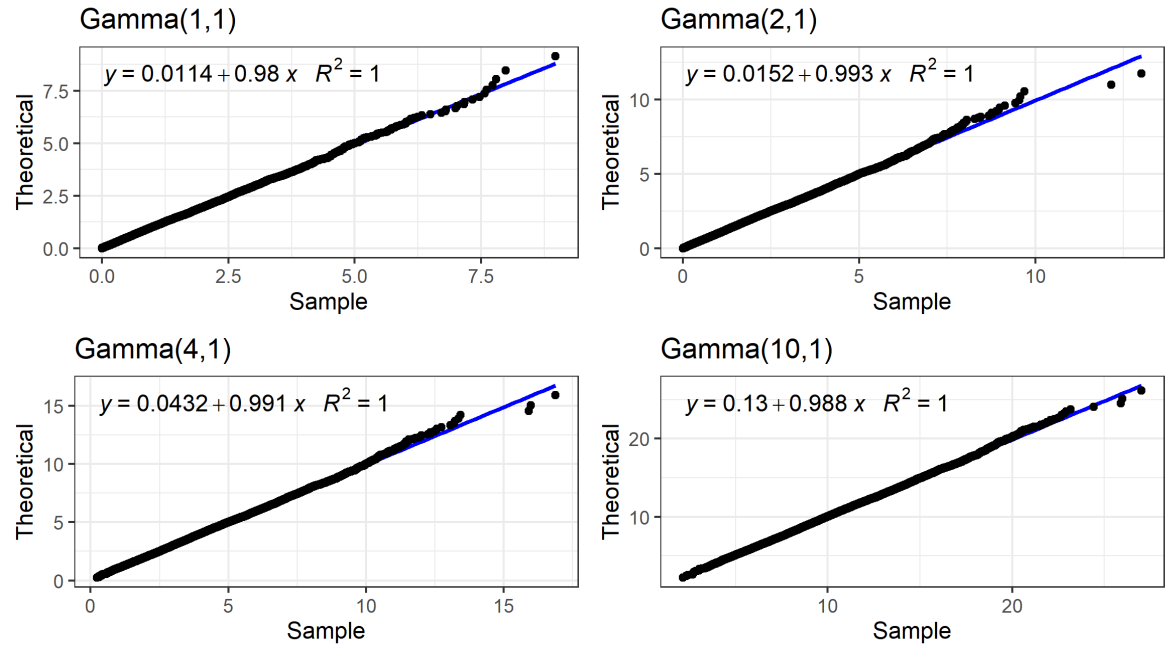
$$Accept\ Z\ and\ set\ X = t(Z)$$

$$if\ t(Z) > 0\ and\ U \leq \frac{q(z)}{e(z)} = \exp\left\{\frac{z^2}{2} + alog\left\{\frac{t(z)}{a}\right\} - t(z) + a\right\}$$

$$Then\ X\ is\ sample\ from\ Gamma(r,1)\ where\ \mathrm{r} \geq 1$$

아래의 결과는 N(0,1)로부터 후보 샘플을 10,000 개를 뽑은 후 Rejection Sampling 을 이용하여 생성한 샘플이 target density 분포를 갖는지 확인하기 위해 아래와 같은 QQ plot 을 그렸다. 또한 전체 후보 샘플 중 accept 비율을 확인하기 위해 acceptance rate 를 계산했다.

ex. 6.1 Gamma Deviates



r	1	2	4	10
Acceptance rate	95.82%	98.15%	99.25%	99.75%

QQ plot 이 거의 직선을 보여주고 R^2값이 1 이라는 사실을 통해 Rejection Sampling 으로 생성된 sample 의 quantile 과 실제 target 함수의 quantile 이 거의 유사하다는 것을 확인할 수 있다. 이는 sample 이 정확히(exactly) target 함수에서 생성된 것과 다름 없음을 보여준다. Acceptance rate 가 가장

높을 때는 r 이 10 일때로 99.75%가 accept 되었고 가장 낮을 때는 r 이 1 일때로 95.82%이다. 가장 acceptance rate 가 낮을 때는 5%미만으로 reject 가 발생한 것으로 보아 target 함수에 적합한 envelope 함수로 sampling 했다는 것을 알 수 있다.

1.2 Rejection Sampling : example 6.2 Bayesian Posterior Sampling

$$observed : 8, 3, 4, 3, 1, 7, 2, 6, 2, 7$$

$$likelihood\ L(\lambda|x) \sim Pois(\lambda)\ \ prior\ log(\lambda) \sim N(log(4), 0.5^2) = f(\lambda)$$

$$\text{Target function(Unnormalized Posterior) } q(\lambda|x) = \text{f}(\lambda)\text{L}(\lambda|\text{x})$$

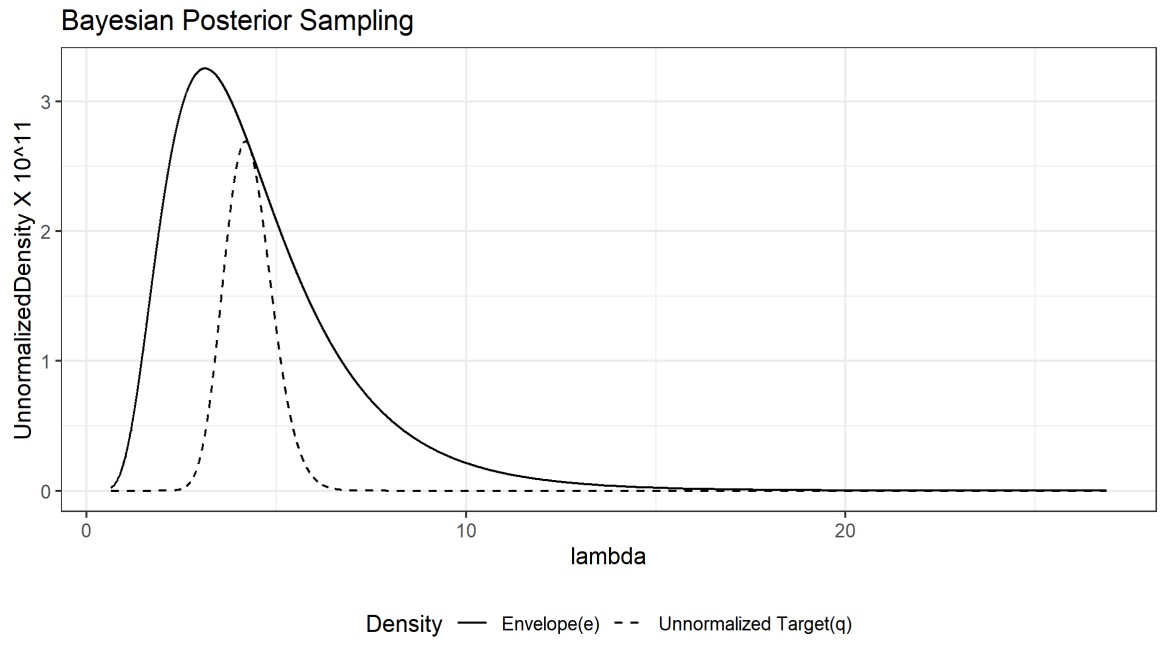
$$\leq \text{f}(\lambda)\text{L}(\bar{x}|x)=e(\lambda|\text{x})$$

$$Draw\ \lambda_i\ from\ prior\ lognormal(log(4), 0.5^2)\ and\ U_i \sim U(0,1)$$

$$\text{Accept } \lambda_i\ if\ U_i < \frac{f(\lambda_i)L(\lambda_i|x)}{f(\lambda_i)L(\bar{x}|x)}\ o.w.Reject$$

아래는 prior 로부터 후보 샘플을 10,000 개 뽑은 후 target 함수인 unnormalized posterior 함수와 envelope 함수를 비교한 그래프, acceptance rate 그리고 Posterior Moment 를 구해본 결과이다.

ex. 6.2 Bayesian Posterior Sampling



acceptance rate	$E(q(x))$	$Var(q(x))$	$E(q(x)^2)$	$E(q(x)^3)$
28.99%	4.29	0.3966	18.76	83.90

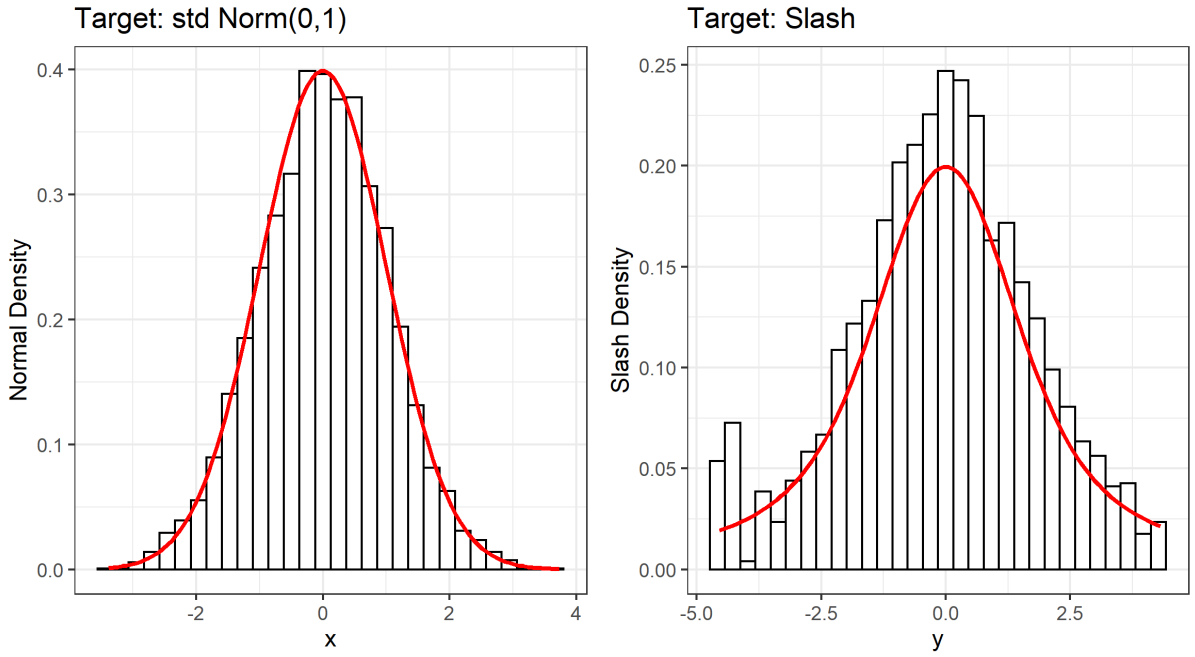
예상했던 대로 target 함수인 unnormalized function 은 Envelope 함수보다 작거나 같다. Acceptance rate 는 28.99%로 envelope 함수가 target 함수의 샘플을 뽑는데 효과적이지 않다는 것을 알 수 있다.

2.1 SIR : example 6.3 Slash and Standard Normal

$$Y = \frac{X}{U}\ follows\ slash\ density\ f(y)\ if\ X \sim N(0,1)\ and\ U \sim Unif(0,1)$$

$$f(y) = \begin{cases} \dfrac{1-\exp(-\frac{y^2}{2})}{y^2\sqrt{2\pi}}, & y \neq 0 \\ \dfrac{1}{2\sqrt{2\pi}}, & y = 0 \end{cases}$$

slash density has lower height but heavier tail than standard normal density



좌측 그래프는 envelope 함수인 slash 분포에서 population 을 m(=100,000)개 뽑아 target 함수인 standard normal 을 따르는 n(=10,000)개의 샘플을 resampling 했다. 우측은 좌측과 반대로 envelope 을 standard normal 로 target 을 slash 로 정하였다. 그 결과 좌측은 n 개의 샘플의 히스토그램이 target density 를 따르지만 우측의 경우 target density 에 벗어나 있다. 이러한 현상이 발생한 이유는 envelope 함수 때문이다. 좌측의 경우 envelope 이 slash 이고 target 이 standard normal 으로 envelope 이 target 보다 분포의 꼬리가 두껍기 때문에 weight=f/g where f is target and g is envelope 가 왜곡되지 않았다. 하지만 우측의 경우 envelope 이 standard normal 이고 envelope 이 slash 로 envelope 이

target 보다 분포의 꼬리가 두껍지 않아 weight 가 꼬리 쪽에서 급격히 커졌다. 따라서 왜곡된 weight 로 resampling 하였기 때문에 최종 샘플이 target density 를 따르지 않는 결과가 발생했다. 이를 통해 envelope 함수는 target 함수보다 분포의 꼬리가 두터워야 함을 알 수 있다. 이러한 사실은 SIR 뿐만 아니라 모든 sampling 기법에 적용된다.

2.2 SIR : example 6.2 Bayesian Posterior Sampling

1.2 에서 Rejection Sampling 을 통해 베이지안 사후 분포의 샘플을 생성했다면 이번에는 SIR 을 이용해 사후 분포의 샘플을 생성하고 두 샘플을 비교해 볼 것이다.

$$Envelope\ function\ prior\ f(\lambda) \sim lognormal(\log(4), 0.5^2)$$

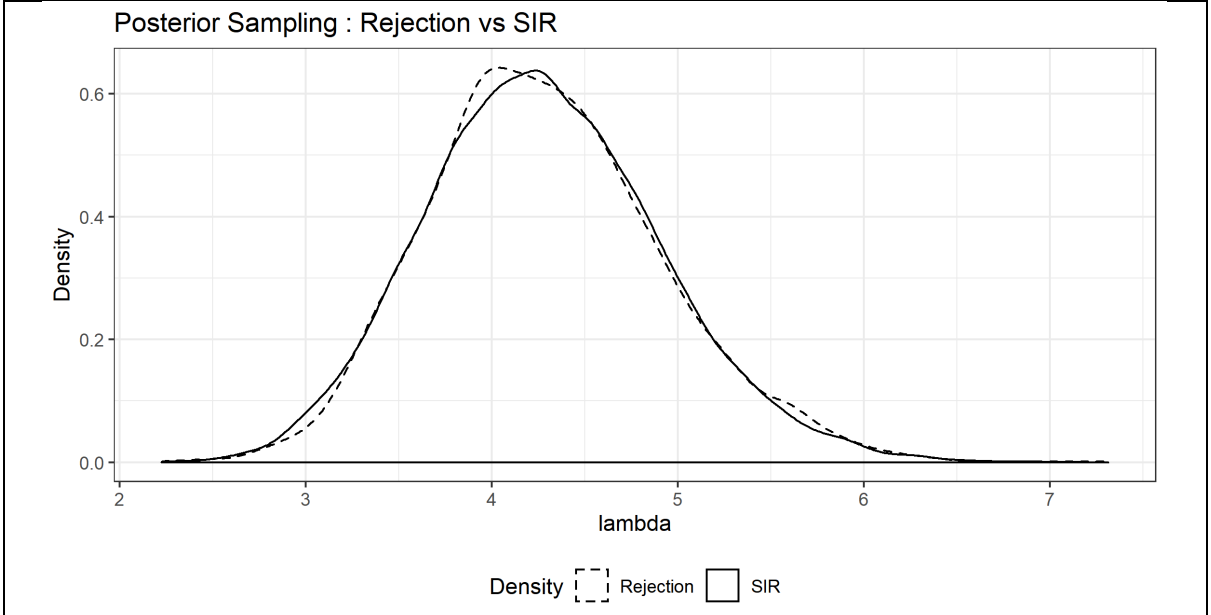
$$Likelihood\ L(\lambda|x) \sim pois(\lambda)$$

$$Target\ density\ Unnormalized\ Posterior\ f(\lambda|x) = f(\lambda) \times L(\lambda|x)$$

$$Weight\ should\ be\ L(\lambda|x) = \frac{f(\lambda|x)}{f(\lambda)}$$

envelope 함수인 prior 로부터 m(=100,000)개의 sample 을 생성한 후 Likelihood 로 weight 를 계산하고 n(=10,000)개를 resampling 한다. 아래는 Rejection Sampling 과 SIR 을 통해 구해진 sample 의 density 그래프와 summary 이다. 아래의 결과를 통해 Rejection Sampling 과 SIR 의 샘플이 유사함을 알 수 있다.

ex. 6.2 Bayesian Posterior Sampling (Rejection Sampling vs SIR)



	Min	Q1	Q2	Q3	Max
Rejection	2.31	3.86	4.25	4.68	7.31
SIR	2.22	3.84	4.25	4.69	6.89

3. Discussion

이번 과제를 통해 고차원의 함수의 적분을 근사하는 MC 를 하기 위해 필요 조건인 iid 샘플 생성 방법을 배웠다. MC 는 target function 으로부터 iid 샘플을 생성해 통계량을 구하는 시뮬레이션 기법이다. MC 이론이 성립하기 위한 필수 조건이 바로 iid 샘플이다. 이론적으로 분포의 iid 샘플은 inverse CDF 를 통해 구할 수 있지만 실제로 대부분의 분포함수들이 inverse CDF 의 closed form 이 정의되지 않기 때문에 iid 샘플을 생성할 수 없다. 따라서 본 과제에서는 Rejection sampling 과 SIR 을 통해 iid 샘플 생성하는 방법을 구현했다. Rejection Sampling 과 SIR 모두 target function f 로부터 직접 샘플링 하지 않고 f 보다 크거나 같은 envelope 함수로부터 후보 샘플을 생성한 후 각 방법의 rule 에 따라 f 분포를 따르는 샘플을 선별한다. 두 방법 모두 envelope 함수에 따라 샘플의 quality 가 결정된다. Envelope 함수는 target function f 와 유사하고 f 를 적당히 포함하는 동시에 f 보다 분포의 꼬리가 두꺼운(heavier)것이 효율적이다. Result 2.1 의 결과를 통해 envelope 이 target function f 보다 분포의 꼬리가 두껍지 않을 때 왜곡된 샘플을 얻을 수 있다는 결론을 얻을 수 있었다. Result 2.2 의 결과를 통해 Rejection Sampling 과 SIR 두 방법을 이용한 베이지안 사후 분포의 sample 이 유사함을 알 수 있었다. 따라서 주어진 문제의 특성에 따라 두 가지 샘플링 방법 중 하나를 택해 iid 샘플을 생성하면 될 것이다. 이때 유의할 점은 Rejection Sampling 은 exact target function 으로부터의 샘플 생성인 반면 SIR 은 샘플의 분포가 approximation to target function 이라는 것이다. 또한 Rejection Sampling 은 sampling 하기 전에 acceptance rate 을 알 수 없기 때문에 후보 샘플의 사이즈와 최종 샘플의 사이즈가 random 인반면 SIR 은 정해진 m 개의 후보 샘플 사이즈를 뽑아 n 개의 최종 샘플을 resampling 할 수 있다.

4. Appendix

```
rm(list=ls())
setwd('C:/Users/dnskd/Desktop/19-2/계특/과제/hw6')
# HW1.1 generate Gamma(r, 1) using Rejection Sampling

rjt <- function(r, n){

  set.seed(1)
  a <- function(r) r - 1/3; b <- function(r) 1/sqrt(9 * a(r))
  z <- rnorm(n, 0, 1)
  z <- z[which(z > -1/b(r))]
  u <- runif(length(z), 0, 1)
  t <- function(r, z) a(r)*(1+b(r)*z)^3

  tz <- t(r, z)
  smp <- tz[which(u <= exp(z^2/2 + a(r)*log(tz/a(r)) - tz + a(r)))]
  rate <- length(smp)/length(z)

  return(list(smp=smp, rate=rate, size=length(z)))
}

# accpeted rate
```

```
rjt(1, 10^4)$rate  #0.9582577
rjt(2, 10^4)$rate  #0.9815
rjt(4, 10^4)$rate  #0.9925
rjt(10, 10^4)$rate #0.9975

# QQplot
library(tidyverse)
library(ggpmisc)
library(gridExtra)
smp1 <- rjt(1, 10^4)$smp
smp2 <- rjt(2, 10^4)$smp
smp4 <- rjt(4, 10^4)$smp
smp10 <- rjt(10, 10^4)$smp

qftn <- function(vt, r){
  df <- vt %>% data.frame
  colnames(df) <- "sample"
  df$rank <- rank(df$sample)
  df$pr <- df$rank/(nrow(df)+1)
  df$qgam <- qgamma(df$pr, shape=r)
  my.formula <- y ~ x
  ggplot(data=df, aes(x = sample, y = qgam)) +
    geom_smooth(method = "lm", se=FALSE, color="blue", formula = my.formula) +
    stat_poly_eq(formula = my.formula,
                 aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),
                 parse = TRUE) +geom_point()+labs(title=paste0("Gamma(",r,",1)"),x="Sample",
y="Theoretical")+theme_minimal()+theme_bw()
}
p1 <- qftn(smp1, 1)
p2 <- qftn(smp2, 2)
p3 <- qftn(smp4, 4)
p4 <- qftn(smp10, 10)
g <- arrangeGrob(p1,p2,p3,p4, ncol=2)
ggsave('qqplot.png', g)

# HW1.2 generate Bayesian Posterior using Rejection Sampling

n <- 10^4
x <- c(8, 3, 4, 3, 1, 7, 2, 6, 2, 7)

set.seed(1)
lam <- rlnorm(n, log(4), 0.5)
u <- runif(n, 0, 1)

prior <- c(); likeli <- c()
for( i in 1:length(lam)){
  prior[i] <- dlnorm(lam[i], log(4), 0.5)
  likeli[i] <- prod(dpois(x, lam[i]))
}
smp <- lam[which(u < likeli/prod(dpois(x, mean(x))))]

# plot
q <- prior*likeli; e <- prior*prod(dpois(x, mean(x)))
df <- data.frame(lam=lam, q=q, qq=q*10^11, e=e, ee=e*10^11)
df %>% ggplot()+geom_line(aes(x=lam, y=qq, linetype="Unnormalized
Target(q)"))+geom_line(aes(x=lam, y=ee, linetype="Envelope(e)"))+labs(x="lambda",
y="UnnormalizedDensity X 10^11", title="Bayesian Posterior Sampling")+
  theme_minimal()+theme_bw()+scale_linetype_manual("Density",values=c("Unnormalized
Target(q)"= 2,"Envelope(e)"=1))+theme(legend.position = "bottom")
ggsave('hw12.png')
```

```
# acceptance rate
length(smp)/length(lam)

# posterior moment
mean(smp)             # 1st moment
var(smp)              # variance
var(smp)+mean(smp)^2 # 2nd moment
mean(smp^3)           # 3rd moment


# HW 2.1 slash example using SIR

slash <- function(z){
  ifelse(z!=0, (1-exp(-z^2/2))/z^2*sqrt(2*pi), 1/(2*sqrt(2*pi)))
}

# case 1) envelope(g) : slash / target(f) : std normal

## step 1. sample m from envelope(slash)
set.seed(1)
m <- 10^5
x <- rnorm(m, 0, 1)
u <- runif(m, 0, 1)
y <- x/u # slash sample

## step 2. calculate weight
w <- dnorm(y, 0, 1)/slash(y)
stdw <- w/sum(w)

## step 3. resample n from m with replacement and weight
n <- 5000
sirsmp <- sample(y, n, replace = T, prob=stdw)

# case 2) envelope(g) : std normal / target(f) : slash

## step 1. sample m from envelope(std normal)
set.seed(1)
y <- rnorm(m, 0, 1)

## step 2. calculate weight
w <- slash(y)/dnorm(y, 0, 1)
stdw <- w/sum(w)

## step 3.resample n from m with replacement and weight
sirsmp2 <- sample(y, n, replace=T, prob=stdw)


## plot to compare
if(!require(extraDistr)) install.packages('extraDistr'); library(extraDistr)
smpdf1 <- sirsmp %>% data.frame()
colnames(smpdf1) <- "x"
smpdf2 <- sirsmp2 %>% data.frame()
colnames(smpdf2) <- "y"

p1 <- ggplot(smpdf1, aes(x=x))+geom_histogram(aes(y=..density..),bins=30,color="black",
fill="white")+
  stat_function(fun=dnorm,col='red',size=1)+labs(x="x",y="Normal Density",title="Target: std
Norm(0,1)")+theme_minimal()+theme_bw()
p2 <- ggplot(smpdf2, aes(x=y))+geom_histogram(aes(y=..density..),bins=30,color="black",
fill="white")+
```

```
  stat_function(fun=dslash,col='red',size=1) + labs(x="y", y="Slash Density",title="Target:
Slash")+theme_bw()+theme_minimal()
g <- arrangeGrob(p1, p2, ncol=2)
ggsave('sir.png',g)

# HW 2.2 Bayesian Posterior Sampling using SIR

## step 1. sample m from envelope(prior)
m <- 10^5
n <- 10^4
x <- c(8, 3, 4, 3, 1, 7, 2, 6, 2, 7)
y <- rlnorm(m, log(4), 0.5)

## step 2. calculate weight(likelihood)
w <- c()
for(i in 1:length(y)) w[i] <- prod(dpois(x,y[i]))

## step 3. resample n from m with replacement and weight
sirb <- sample(y, n, replace = T, prob=w)


## compare Rejection and SIR
# 1. density graph
rejdf <- smp %>% data.frame; colnames(rejdf) <- "smp"
sirdf <- sirb %>% data.frame; colnames(sirdf) <- "smp"
ggplot(data=rejdf,
aes(x=smp))+geom_density(aes(linetype='Rejection'))+geom_density(data=sirdf,aes(linetype='SIR'
))+theme_minimal()+theme_bw()+
  labs(x="lambda",y="Density",title="Posterior Sampling : Rejection vs
SIR")+theme(legend.position = "bottom")+
  scale_linetype_manual("Density",values = c('Rejection' = 2, 'SIR' = 1))
ggsave('rejvssir.png')

# 2. summary
summary(smp)
summary(sirb)
```