Accept/Reject Sampling Methods & Stratified Sampling

Prof. Nicholas Zabaras
University of Notre Dame
Notre Dame, IN, USA

Email: nzabaras@gmail.com
URL: https:/www.zabaras.com/

September 27, 2017

Contents

- Monte Carlo integration – a review, Sampling from a discrete distribution, Reverse sampling for continuous distributions
- Transformation Methods, Box-Muller Algorithm, sample from the bivariate normal distribution, sampling from the multivariate Gaussian, simulation my composition
- Accept-Reject Methods, Log-concave densities
- Monahan's Accept/Reject Method
- Conditional MC, Stratified Sampling, Systematic Sampling

Following closely:

- C.P. Roberts and G. Casella, *Monte Carlo Statistical Methods*, Chapter 2 (google books, slides, video)
- J S Liu, Monte Carlo Strategies in Scientific Computing, Chapter 2
- A. Doucet, Statistical Computing and Monte Carlo Methods (2007)

The Problem of Interest

- $\boldsymbol{x} \equiv$ vector of *k*-random variables.
- $\pi(\boldsymbol{x}) \equiv$ distribution function. In the context of Bayesian analysis, this will be the posterior distribution.
- Goal is to evaluate $\mathbb{E}_{\pi}\{f(\boldsymbol{x})\}$

$$\mathbb{E}_{\pi}\{f(\boldsymbol{x})\} \equiv \frac{\int f(\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x}}{\int \pi(\boldsymbol{x})d\boldsymbol{x}}$$

- $\int \pi(\boldsymbol{x})d\boldsymbol{x}$ is the normalizing constant.
- In most algorithms to be discussed in this and the following lectures, we do not need to know this normalization constant.

Monte Carlo Integration: Review

- Monte Carlo integration evaluates $\mathbb{E}_\pi\{f(\mathbf{x})\}$ by drawing samples $\{\boldsymbol{x}_t; t = 1,\dots N\}$ from $\pi(\boldsymbol{x})$ and then approximating

$$\mathbb{E}_\pi\left(f(\mathbf{x})\right) \equiv \frac{1}{N}\sum_{t=1}^{N} f(\mathbf{x}_t), \mathbf{x}_t \; i.i.d.$$

- That is, the population mean of $f(X)$ is estimated by a sample mean.
- When $\boldsymbol{x}_t$ are i.i.d., LLN ensures that the approximation can be accurate as desired by increasing N.
- N is the number of samples we use to approximate $\mathbb{E}_\pi\left(f(\mathbf{x})\right)$
- As we have seen, the convergence rate is $\frac{1}{\sqrt{N}}$.

Sampling From an Arbitrary Distribution

- Let us assume that we need to estimate

$$\mathbb{E}_{\pi}\left(f(\mathbf{x})\right) = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \mathbf{x} \in \mathcal{X}$$

- The basic steps in MC are the following:
  - Generate i.i.d samples

$$\mathbf{x}_i \sim \pi(\mathbf{x})$$

But how do we sample from an arbitrary distribution?

  - Evaluate

$$\hat{f} = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i)$$

- We have shown that this estimator is unbiased:

$$\mathbb{E}(\hat{f}) = \mathbb{E}_{\pi}(f(\mathbf{x}))$$

- The variance of the estimator is:

$$Var(\hat{f}) = \frac{Var(f(\mathbf{x}))}{N}$$

Sampling From an Arbitrary Distribution

- Consider an arbitrary probability density $\pi(\mathbf{x})$

- Monte Carlo approximation is given by

$$\hat{\pi}_N(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X^{(i)}}(\mathbf{x}) \text{ , } where\ X^{(i)} \overset{i.i.d.}{\sim} \pi$$

- For any function $f : \mathcal{X} \to \mathbb{R}$

$$\mathbb{E}_{\hat{\pi}_N}(f) = \frac{1}{N}\sum_{i=1}^{N} f(X^{(i)}) \cong \mathbb{E}_{\pi}(f)$$

But how do we sample from an arbitrary distribution?

or more precisely:

$$\mathbb{E}_X\left[\mathbb{E}_{\hat{\pi}_N}(f)\right] = \mathbb{E}_{\pi}(f)\ and\ Var_X\left(\mathbb{E}_{\hat{\pi}_N}(f)\right) = \frac{Var_{\pi}\ (f)}{N}$$

Sampling Uniform Random Variables

- Most sampling algorithms rely on generating uniform random variables in [0, 1].
- We only have algorithms for generating pseudo-random numbers which look like they are i.i.d. $\mathcal{U}$ [0, 1].
- There are several standard uniform random number generators available.

Sampling From a Discrete Distribution

- Consider $\mathcal{X} = \{1, 2, 3\}$ and

$$\pi(X = 1) = 1/6,\ \pi(X = 2) = 2/6,\ \pi(X = 3) = 1/2$$

- Define the cdf of X for $x \in [0, 3]$ as

$$F_X(x) = \sum_{i=1}^{3} \pi(X = i)\mathbb{I}(i \leq x)$$

and its inverse for $u \in [0, 1]$

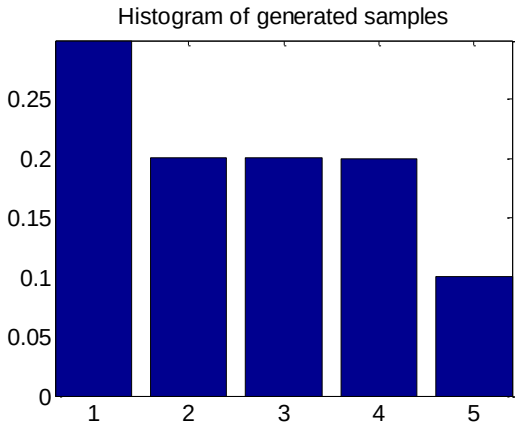
$$F_X^{-1}(u) = \inf\{x \in \mathcal{X}; F_X(x) \geq u\}$$

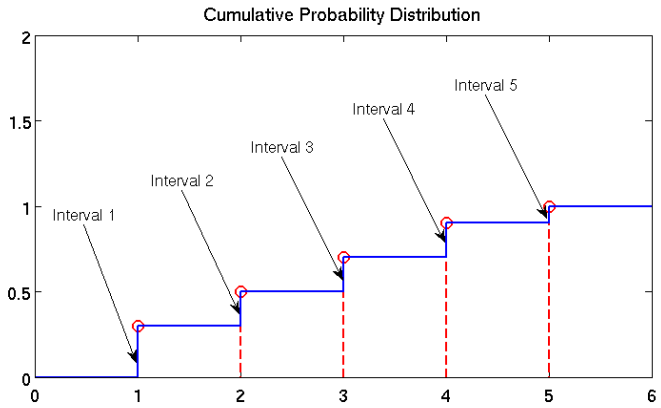
Sampling from a Discrete Distribution

- Consider the distribution of a discrete random variable
- To sample from this discrete distribution, sample $u \sim \mathcal{U}[0,1]$
- Find $X = F_X^{-1}(u)$
- The probability of u falling in the vertical interval i is equal to the probability $\pi(X = i)$.
- We will show next:

$$\Pr[x = k] = \Pr[P_{k-1} < u \le P_k],$$

$$where\ P_k = \sum_{j=1}^{k} p_j$$





MatLab Implementation

Sampling from a Discrete Distribution

- Consider a discrete random variable X taking values $\{1,2,3,\ldots,\infty\}$ with probability $\Pr[X=k]=p_k, \text{ with } \sum_{i=1}^{\infty} p_i = 1$
- We want by sampling $u \sim \mathcal{U}[0,1]$ to generate samples $x \sim p(X)$
- Let $P_k \in [0,1], s.t.\ P_k = \Pr[X \le k] = \sum_{j=1}^{k} p_j$
- Draw $u \sim \mathcal{U}[0,1]$.
- Then set $x=k$, where $k = \min\{k \in \{1,2,3,\ldots,+\infty\} \text{ such that } u \le P_k\}$

$$Note: \Pr[x=k] = \Pr[P_{k-1} < u \le P_k] = \int_{P_{k-1}}^{P_k} 1\,du$$

$$= P_k - P_{k-1} = \sum_{j=1}^{k} p_j - \sum_{j=1}^{k-1} p_j = p_k, \text{ i.e. the desired distribution}$$

Sampling from a Continuous Distribution

- Assume the distribution has a density, then the CDF takes the form:

$$Note: F_X(x) = Pr(X \le x) = \int_{-\infty}^{+\infty} \pi(u)\mathbb{I}(u \le x)du = \int_{-\infty}^{x} \pi(u)du$$

- Algorithm: $u \sim \mathcal{U}[0,1]$ and then set $X = F_X^{-1}(u)$. Do we have $X \sim \pi$?

$$Proof: Pr(X \le x) = Pr\left(F_X^{-1}(u) \le x\right) = \Pr\left(u \le F_X(x)\right) \text{ since } F_X \text{ is non-decreasing}$$

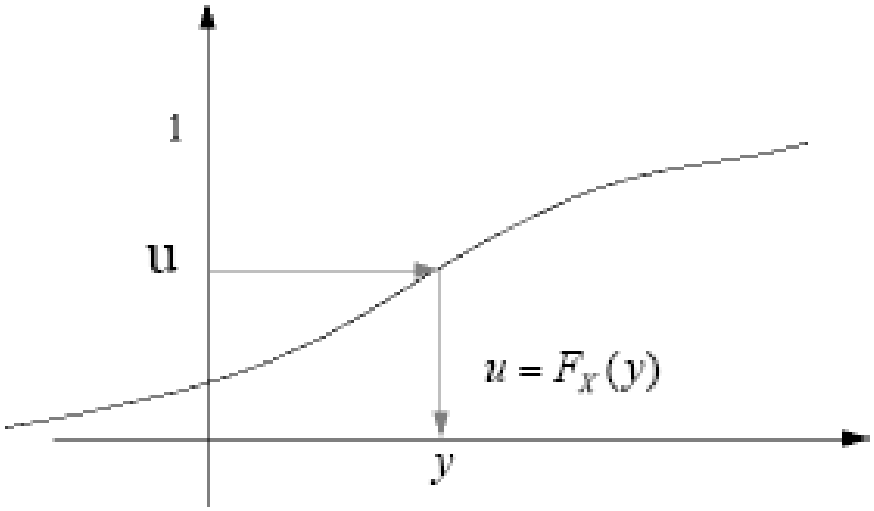
$$= \int_0^1 \mathbb{I}(u \le F_X(x))du \text{ since } u \sim \mathcal{U}[0,1]$$

$$= F_X(x) \text{ which is precisely the CDF of } \pi$$

Reverse Sampling for Continuous Distributions

- Simplest method
- Used frequently for non-uniform random number generation
- Sample a random number u from $\mathcal{U}(0,1)$
- Set x=y, where u=F_X(y) (i.e. $x = F_X^{-1}(u)$)
- Simple, but you need the functional form of F.

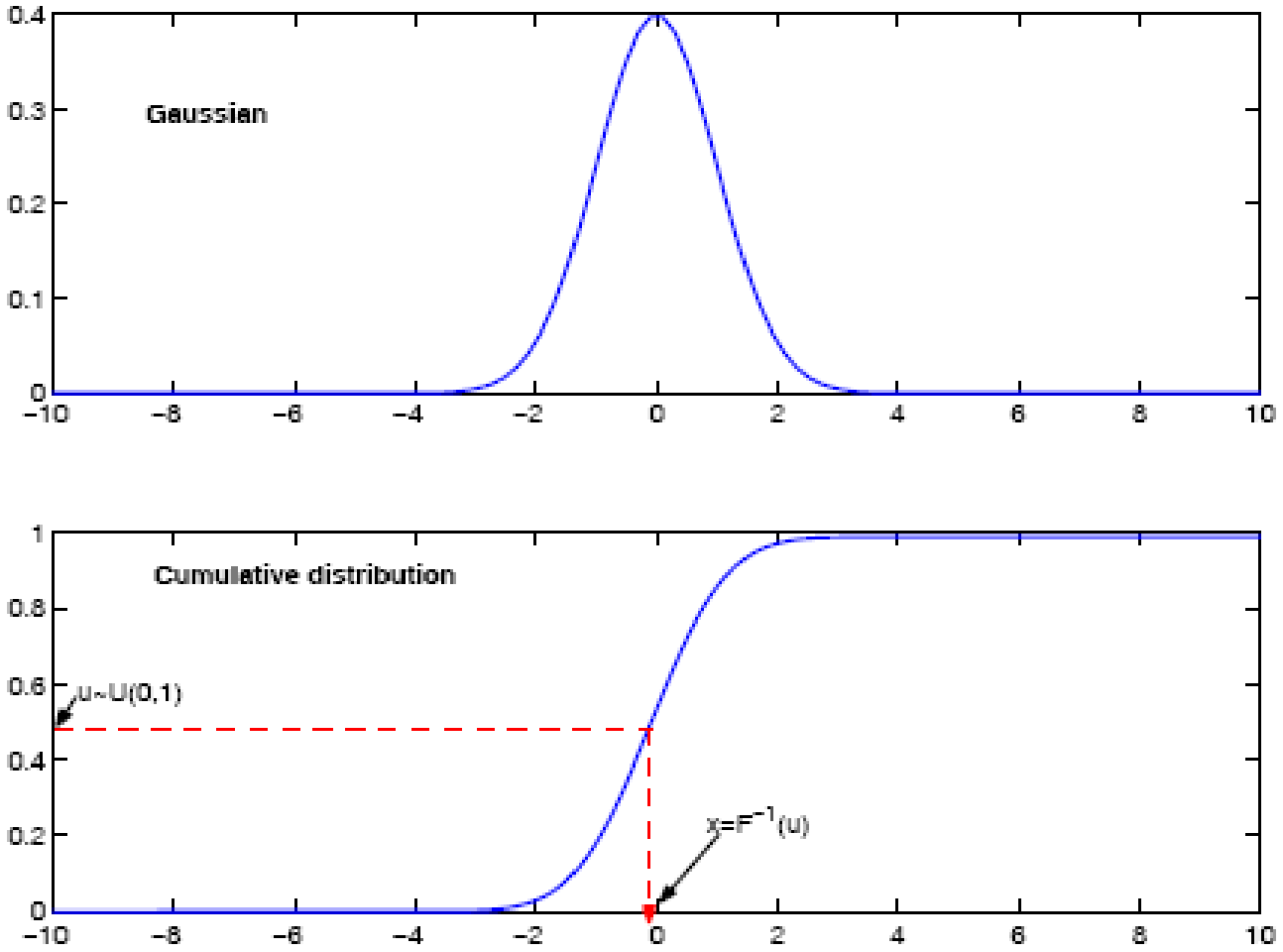
$$Indeed: \Pr[x \le y] = \Pr[u \le F_X(y)]$$
$$= \int_0^{F_X(y)} 1 du = F_X(y)$$



Sampling using the inverse of the CDF

Sampling from a Continuous Distribution

- The distribution and the CDF of a normal distribution



Inverse Method: Exponential Distribution

- Consider the exponential distribution with parameter λ:

$$\pi(x) = \lambda e^{-\lambda x} \mathbb{I}_{[0,+\infty)}$$

- The CDF of X is:

$$F_X(x) = \int_{-\infty}^{x} \pi(z)dz = \begin{cases} 0 \; if \;\; x \le 0 \\ 1 - e^{-\lambda x} \; if \;\; x > 0 \end{cases}$$

- Thus the inverse CDF is:

$$1 - e^{-\lambda x} = u \Leftrightarrow x = -\frac{1}{\lambda}\log(1-u) = F_X^{-1}(u)$$

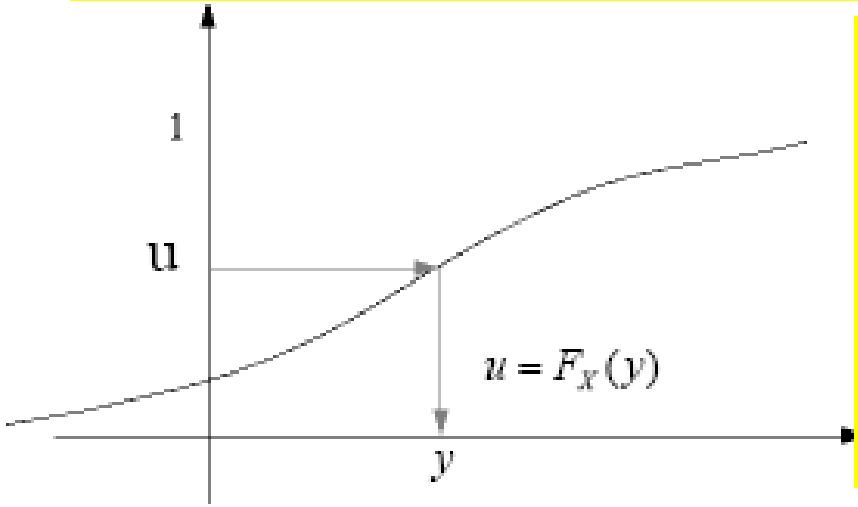
- Inverse method: $u \sim \mathcal{U}[0,1]$, then $X = -\frac{1}{\lambda}\log(1-U) \sim \pi \equiv \mathcal{E}xp(\lambda)$

- Similarly if $u \sim \mathcal{U}[0,1]$, then $X = -\frac{1}{\lambda}\log U \sim \mathcal{E}xp(\lambda)$. Indeed:

$$\Pr(X \le x) = \Pr(-\log U \le \lambda x) = \Pr\left(U \ge e^{-\lambda x}\right) = 1 - e^{-\lambda x} \Rightarrow X \sim \mathcal{E}xp(\lambda)$$

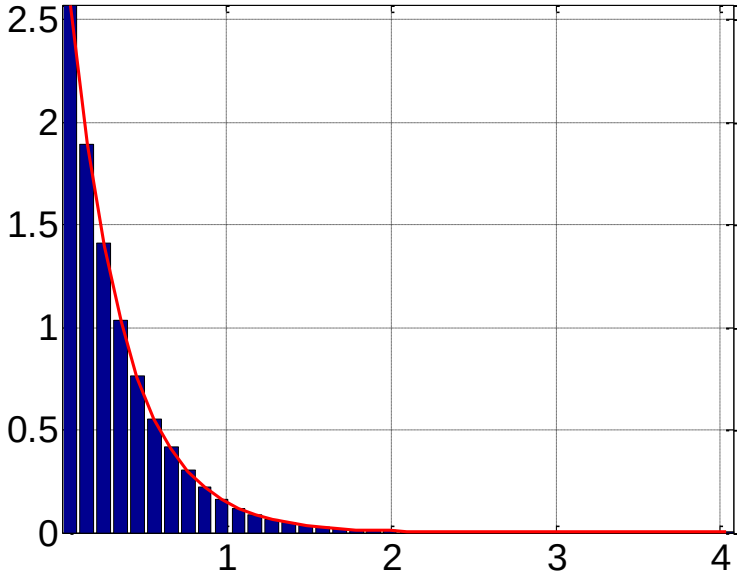
Inverse Method: Exponential Distribution

Exponential(λ): $\pi(x)= \lambda e^{-\lambda x}$, $F_X(y) = 1 - e^{-\lambda y}$, $x= -\ln(1-u)/\lambda$



As *F(y)* is area under *π(y)*, *$y = F^{-1}(u)$* prescribes that

- **Choose *u=(0,1]*, then find value y that has that *fraction u of area to the left of y, or u=F(y)***
- **Return that value of *x=y*.**



MatLab Implementation

Inverse Method: Example

- Consider N i.i.d. random variables $X_i \sim f_X$ (with CDF F_X). We are interested to sample from the following distribution:

$$Z = \max\left(X_1, ..., X_N \right)$$

- You can use an inverse approach as follows:

$$F_Z(z) = \Pr\left(X_1 \le z, ..., X_N \le z \right)$$

$$= \prod_{i=1}^{N} \Pr\left(X_i \le z \right) = \left[F_X(z) \right]^N$$

- Thus for any $U \sim \mathcal{U}[0,1]$, we can sample $Z \sim f_Z$ as follows:

$$Z = F_Z^{-1}(U) = F_X^{-1}(U^{1/N})$$

Inverse Method: Limitations

- Practical and simple method for univariate distributions.
- Limited to cases where the inverse cdf has an analytical form that can be tabulated.
- Its practical use is very limited.

Transformation Methods

- We have seen an example of the transformation method:

$$If\ u \sim \mathcal{U}[0,1] \Rightarrow X = -\frac{1}{\lambda}\log(1-U) \sim \mathcal{E}xp(\lambda)$$

$$If\ u \sim \mathcal{U}[0,1] \Rightarrow X = -\frac{1}{\lambda}\log U \sim \mathcal{E}xp(\lambda)$$

- We use the fact that π is related to other transformations easier to sample from.
- These methods are specific to some distributions, e.g.

$$\text{If } X_i \sim \mathcal{E}xp(1)\ then: Y = 2\sum_{i=1}^{\nu} X_i \sim \chi^2_{2\nu}, \quad Y = \beta\sum_{i=1}^{\alpha} X_i \sim \mathcal{G}amma(\alpha,\beta),$$

$$Y = \frac{\sum_{i=1}^{\alpha} X_i}{\sum_{i=1}^{\alpha+\beta} X_i} \sim \mathcal{B}e(\alpha,\beta)$$

Transformation Methods

- Starting with samples from the uniform distribution, these transformations are very simple to apply:

$$Y = -2\sum_{i=1}^{\nu} \log\left(U_i\right) \sim \chi^2_{2\nu},$$

$$Y = -\beta\sum_{i=1}^{\alpha} \log\left(U_i\right) \sim \mathcal{Gamma}(\alpha, \beta)$$

$$Y = \frac{\sum_{i=1}^{\alpha} \log\left(U_i\right)}{\sum_{i=1}^{\alpha+\beta} \log\left(U_i\right)} \sim \mathcal{Be}(\alpha, \beta)$$

- With this approach, we cannot generate Gamma random variables with a non-integer shape parameter α.

- Cannot generate χ^2_1 which will give us a $\mathcal{N}(0,1)$ variable.

Box Muller Algorithm to Sample Gaussians

- Consider (*x*,*y*) coordinates $X_1 \sim \mathcal{N}(0,1)$ *and* $X_2 \sim \mathcal{N}(0,1)$, then the polar coordinates (R,θ) of this point are independent and distributed according to

$$R^2 = X_1^2 + X_2^2 \sim \mathcal{Exp}\left(\frac{1}{2}\right)$$
$$\theta \sim \mathcal{U}[0, 2\pi]$$

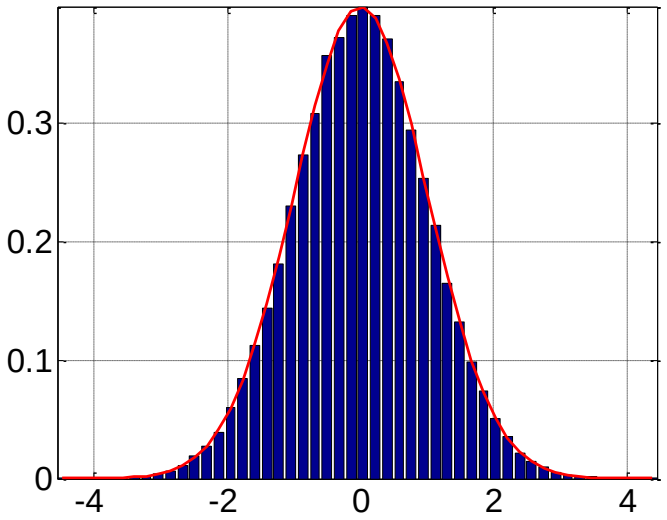
$If\ X_1 \sim \mathcal{N}(0,1),\ \ldots, X_n \sim (0,1),$ then $X_1^2 + \ldots + X_n^2 \sim x_n^2$. For n=2, $x_{n=2}^2 \sim \mathcal{E}xp\left(\frac{1}{2}\right)$.

- It is simple to simulate $R = \sqrt{-2\log(U_1)}$ *and* $\theta \sim 2\pi U_2$, where

$U_1 \sim \mathcal{U}[0,1]$ *and* $U_2 \sim \mathcal{U}[0,1]$. *Then*

$$X_1 = R\cos\theta = \sqrt{-2\log(U_1)}\cos(2\pi U_2)$$
$$X_2 = R\sin\theta = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

- By construction X_1 and X_2 are two independent $\mathcal{N}(0,1)$ random variables. A plot of one of them is shown here.



MatLab Implementation

Box Muller Algorithm to Sample Gaussians

- Let us see a simpler approach to sample from a spherical Gaussian. We sample z_1, z_2 uniformly from a unit circle: $p(z_1, z_2) = \frac{1}{\pi}\mathbb{I}(z\ inside\ unit\ circle)$. This can be done by sampling uniformly on the cube $[-1\ 1]^2$ and disregarding the samples outside the circle.

- Consider polar coordinates (r, θ). You can show that $\left|\frac{\partial(z_1,z_2)}{\partial(r,\theta)}\right| = \left|\frac{\partial(rcos\theta, rsin\theta)}{\partial(r,\theta)}\right| = r$.

- Define $y_1 = (-2lnr^2)^{1/2}cos\theta, y_2 = (-2lnr^2)^{1/2}sin\theta$. With simple algebra you can show that $\left|\frac{\partial(y_1,y_2)}{\partial(r,\theta)}\right| = -\frac{2}{r}$.
- The distribution of y_1, y_2 can now be computed as:

$$p(y_1, y_2) = p(z_1, z_2)\left|\frac{\partial(z_1, z_2)}{\partial(r, \theta)}\right|\left|\frac{\partial(y_1, y_2)}{\partial(r, \theta)}\right|^{-1} = \frac{1}{\pi}r\frac{r}{2} = \frac{r^2}{2\pi} = \frac{1}{2\pi}exp\left(-\frac{1}{2}(y_1^2 + y_2^2)\right)$$

- Thus we obtain the desired result:

$$p(y_1, y_2) = \frac{1}{\sqrt{2\pi}}exp\left(-\frac{1}{2}y_1^2\right)\frac{1}{\sqrt{2\pi}}exp\left(-\frac{1}{2}y_2^2\right)$$

Sampling from the Bivariate Normal Density

To sample from the bivariate density,

1) Generate two, uncorrelated, standard normal variates, z_1 *and* z_2.

$$z \sim \mathcal{N}\left(\mu = 0, \sigma^2 = 1\right)$$

2) Compute the correlated X_1 and X_2

$$X_1 = \mu_1 + \sigma_1 z_1$$
$$X_2 = \mu_2 + \sigma_2 \left[z_1 \rho + z_2 \sqrt{1 - \rho^2} \right]$$

3) X_1 *and* X_2 will have means μ_1 and μ_2, standard deviations σ_1 and σ_2, and correlation ρ.

Sampling from the Bivariate Normal Density

Example 1. data are sampled from the standard bivariate normal distribution

Example 2. data are sampled from the bivariate normal distribution with

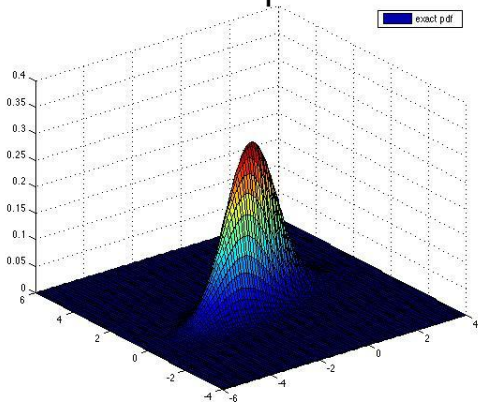
Mean vector $\{\mu_1, \mu_2\} = \{-1, 1\}$

Covariance matrix $\Sigma = \begin{bmatrix} 1.44 & 0.3 \\ 0.3 & 0.25 \end{bmatrix}$

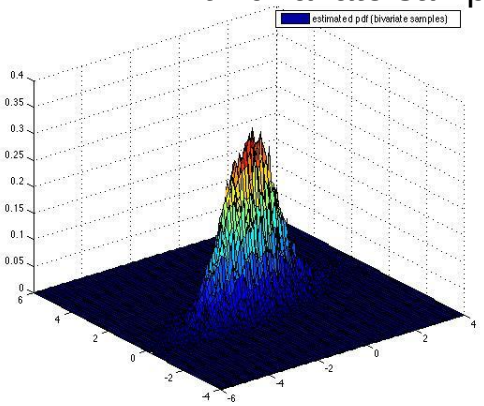
Sampling from the Bivariate Normal Density

Validation of the algorithm

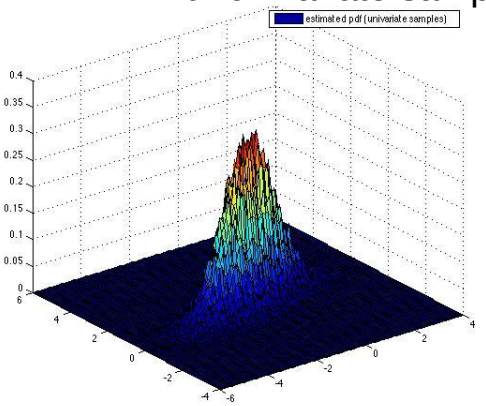
Exact pdf



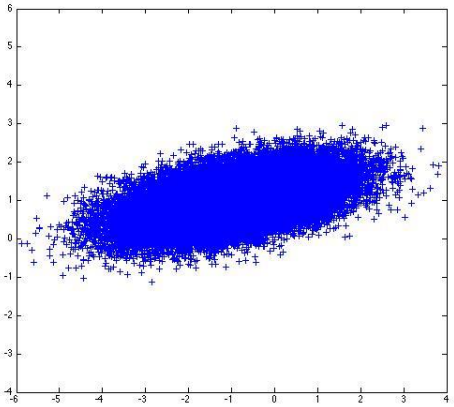
PDF with bivariate samples

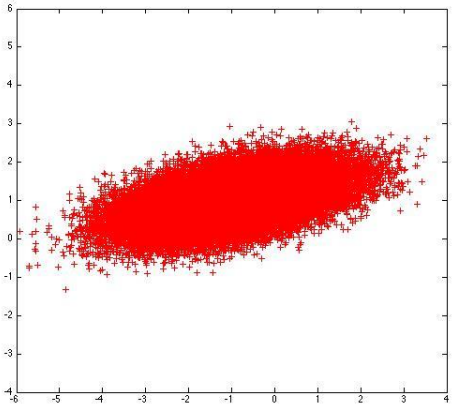


PDF with univariate samples



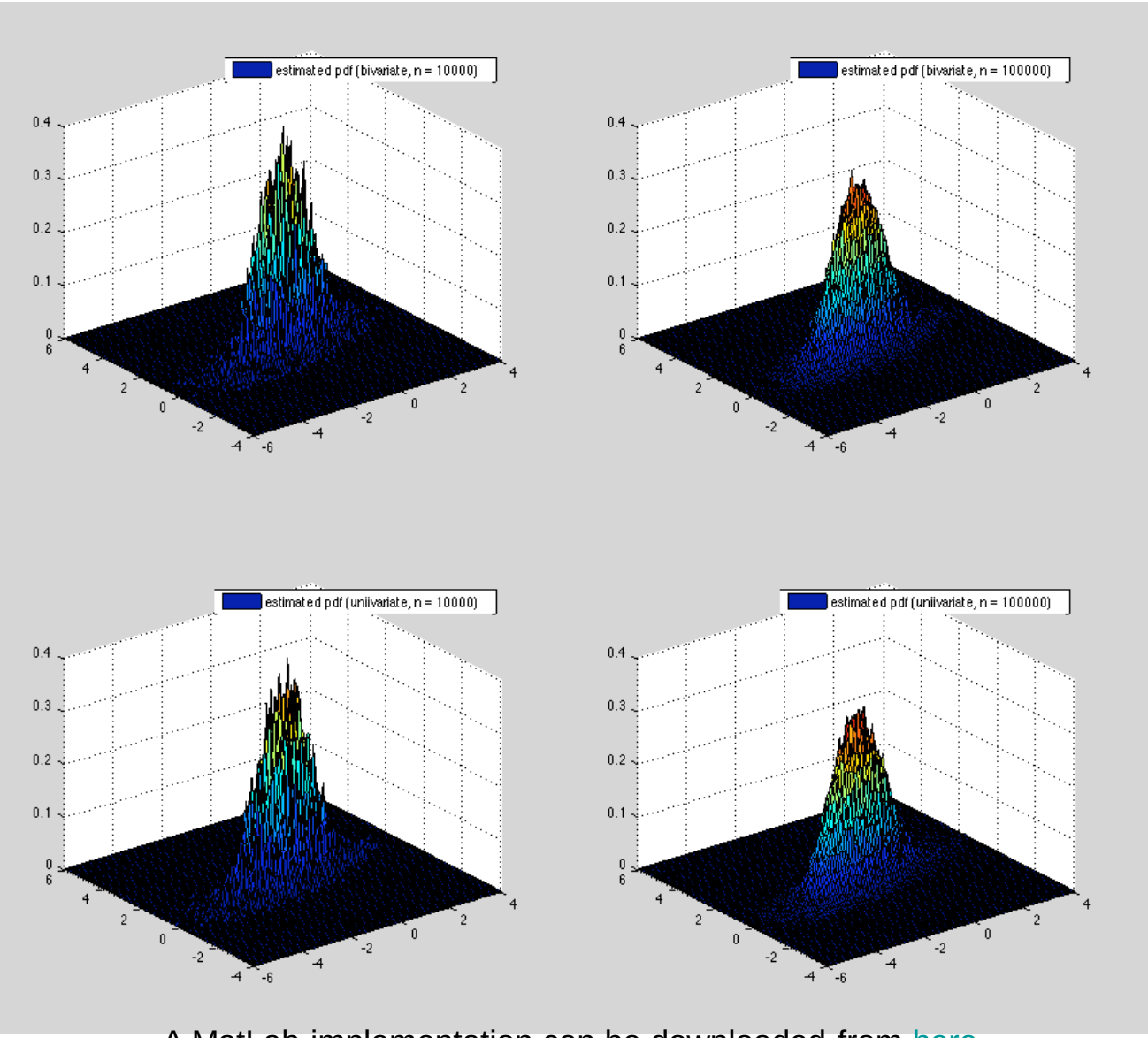
The pdf generated from samples compared with the true standard bivariate Gaussian pdf





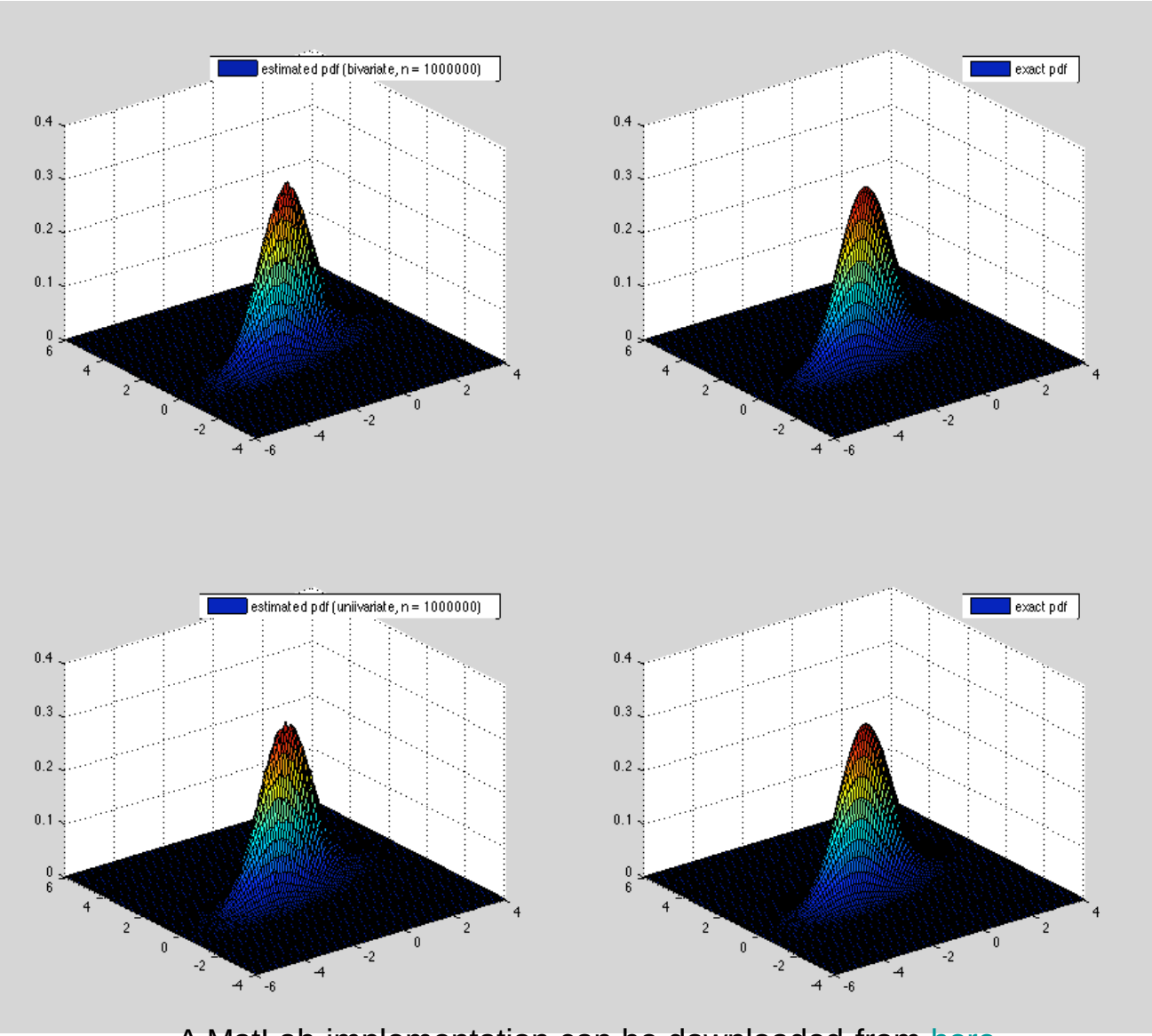
A MatLab implementation can be downloaded from here.

Sampling from the Bivariate Normal Density



A MatLab implementation can be downloaded from here.

Sampling from the Bivariate Normal Density



A MatLab implementation can be downloaded from here.

Sampling from the Multivariate Gaussian

- In principle, the same approach can work:
  - Sample $\boldsymbol{u} \sim \mathcal{U}[0,1]^d$
  - Set $\boldsymbol{x} = F_X^{-1}(\boldsymbol{u})$
- How about for the multivariate Gaussian: $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
  - Draw $z_i \sim \mathcal{N}(0,1)$ i.i.d.
  - Introduce the Cholesky decomposition of $\boldsymbol{\Sigma}$: $\boldsymbol{\Sigma} = \boldsymbol{S}\boldsymbol{S}^T$
  - Set

$$\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{S}\boldsymbol{z}$$

$$Indeed: \mathbb{E}\left[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T \right] = \mathbb{E}\left[\boldsymbol{S}\boldsymbol{z}\boldsymbol{z}^T \boldsymbol{S}^T \right] =$$

$$= \boldsymbol{S}\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^T \right] \boldsymbol{S}^T = \boldsymbol{S}\boldsymbol{S}^T = \boldsymbol{\Sigma}$$

Simulation by Composition

- Assume we have:

$$\pi(x) = \int \bar{\pi}(x, y)dy$$

where it is easy to sample from $\bar{\pi}(x, y)$ but difficult to sample directly from $\pi(x)$.

- In this case, it is sufficient to sample

$$(X, Y) \sim \bar{\pi} \Rightarrow X \sim \pi$$

- One samples from $\bar{\pi}(x, y) = \bar{\pi}(y)\bar{\pi}(x \mid y)$ as follows:

$$Y \sim \bar{\pi}, then\ X \mid Y \sim \bar{\pi}(. \mid Y)$$

Sampling from a Mixture of Distributions

- Assume we want to sample from:

$$\pi(x) = \sum_{i=1}^{p} \pi_i \times \pi_i(x),\ \pi_i > 0,\ \sum_{i=1}^{p} \pi_i = 1,\ \ \pi_i(x) \geq 0,\ \int \pi_i(x)dx = 1.$$

- We can introduce $Y \in \{1, 2, ..., p\}$ and

$$\bar{\pi}(x, y) = \pi_y \times \pi_y(x) \Rightarrow \begin{cases} \int \bar{\pi}(x, y)dy = \pi(x) \\ \int \bar{\pi}(x, y)dx = \bar{\pi}(y) = \pi_y \end{cases}$$

- To sample from $\pi(x)$, then sample $Y \sim \bar{\pi}$ (discrete distribution such that $\Pr(Y = k) = \pi_k$) then sample

$$X \mid Y \sim \bar{\pi}(. \mid Y) = \pi_Y$$

Sampling from a Scale Mixture of Gaussians

- Consider a scale mixture of Gaussians, i.e.

$$\pi(x) = \int \mathcal{N}(x; 0, 1/y)\bar{\pi}(y)dy$$

- By using different mixing distributions $\bar{\pi}(y)$, we obtain distributions $\pi(x)$ which are $\mathcal{T}$-Student's, α-stable, Laplace, logistic, etc.

- If for example, $Y \sim \chi^2_\nu$ *and* $X|Y \sim \mathcal{N}(0, \nu/y)$, then X is marginally distributed according to a t-student with ν degrees of freedom.

- Condition upon Y, X is Gaussian. This structure is used to develop efficient MCMC algorithms.

Andrews, DF and Mallows, CL, "Scale Mixtures of Normal Distributions", JRSS B, Vol. 36, No. 1. (1974), pp. 99-102. Get it from JSTOR

Rejection Sampling

- More generally, we would like to sample from $\pi(x)$ defined on $\mathcal{X}$ only known up to a proportionality constant, $\pi \propto \pi^*$

- The method relies on samples generated from a *proposal distribution q(x) on X. q might also be known up to a normalizing constant,* $q \propto q^*$

- We need q(x) to dominate p(x), i.e.

$$M = \sup_{x \in X} \frac{\pi^*(x)}{q^*(x)} < +\infty$$

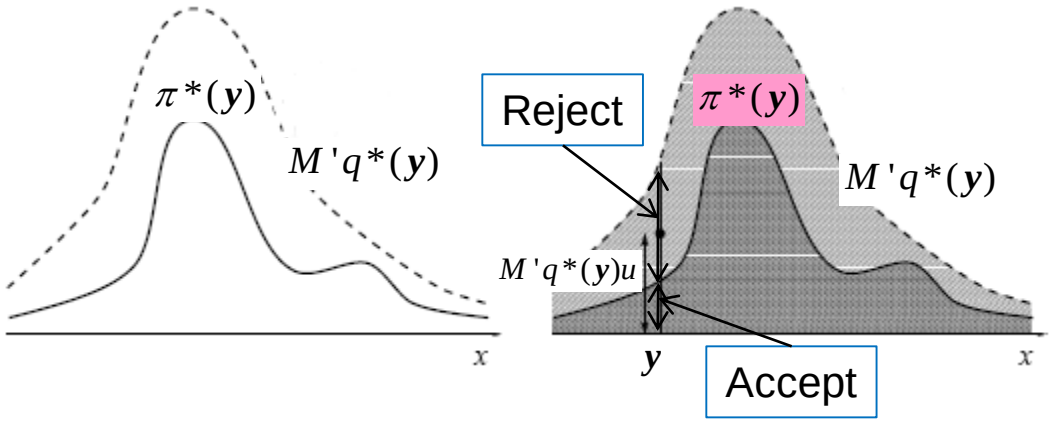
- This implies that $\pi^*(x) > 0 \Rightarrow q^*(x) > 0$ but also that the tails of $q^*(x)$ must be thicker than the tails of $\pi^*(x)$.

Accept/Reject Algorithm

- More generally, we would like to sample from $\pi(x)$, but it's easier to sample from a *proposal distribution q(x)*
- q(**x**) satisfies $\pi^*(\mathbf{x}) < M' q^*(\mathbf{x})$ for all **x** for some $M' \geq M$
- Procedure:
  - Sample y from q(y) and u from $\mathcal{U}[0,1]$
  - Accept (set x=y) with probability $\pi^*(y) / M' q^*(y)$ (i.e. if $u \leq \pi^*(y) / M' q^*(y)$)
  - Reject otherwise and repeat.
- The accepted $x^{(i)}$ are sampled from $\pi(x)$!
- If M' is too large, we will rarely accept samples.

B. D. Flury, Rejection Sampling made easy, SIAM review, 1990
J. Halton, Reject the rejection technique, J. Scientific Computing, 1992

Rejection Sampling



- Set *i=1*
- Repeat until *i=N*
  - Sample $y \sim q(y)$ *and* $u \sim \mathcal{U}_{(0,1)}$
  - If $u < \frac{\pi^*(y)}{M'q^*(y)}$ then accept (set $x^{(i)}=y$) and increment the counter i
    - ✓ Otherwise, reject

The distribution $\pi(y)$ needs to be known only up to a normalizing constant:

$$\pi(y) = \frac{\pi^*(y)}{Z}, \; Z = \int \pi^*(y)dy$$

Rejection Sampling

- Step 1:Sample $y \sim q^*(y)$ *and* $u \sim \mathcal{U}_{(0,1)}$
- Step 2: Set $x=y$ if $u < \frac{\pi^*(y)}{M' q^*(y)}$. Otherwise return to step 1.

$$Define: J = \begin{cases} 1 & \text{if the proposed sample } \boldsymbol{y} \text{ is accepted} \\ 0 & \text{otherwise} \end{cases}$$

$$Indeed: \Pr[J=1] = \int \Pr[J=1 \mid y] q(y) dy = \frac{1}{\int q^*(y)dy} \int \frac{\pi^*(y)}{M' q^*(y)} q^*(y)dy = \frac{Z}{M' \int q^*(y)dy} \Rightarrow$$

Distribution of accepted $\boldsymbol{x}$:

$$p(y = x \mid J = 1) = \frac{p(y = x \text{ and } J = 1)}{p(J = 1)} = \frac{p(J = 1 \mid y = x)\, p(y = x)}{p(J = 1)} =$$

$$= \underbrace{\frac{\pi^*(x)}{M' q^*(x)}}_{\Pr[J=1 \mid \boldsymbol{y}=\boldsymbol{x}]} \underbrace{q(x)}_{p(\boldsymbol{y}=\boldsymbol{x})} \underbrace{\frac{M' \int q^*(y)dy}{Z}}_{\Pr[J=1]} = \frac{\pi^*(x)}{Z} = \pi(x)$$

$$Note: From \Pr[Y \text{ accepted}] = \frac{\int \pi^*(y)dy}{M' \int q^*(y)dy} \Rightarrow large\ M' \ leads\ to\ very\ low\ acceptance$$

Rejection Sampling: Proof

- We can alternatively show that $\Pr\left(Y \le x \,|\, Y \text{ accepted}\right) = \Pr\left(X \le x\right) \forall x \in \mathcal{X}$
- Indeed, it is clear that:

$$\Pr(Y \le x \text{ and } Y \text{ accepted}) = \int_0^1 \int_{-\infty}^{x} \mathbb{I}\left(u \le \frac{\pi^*(y)}{M' q^*(y)}\right) q(y) \times 1 \, dy \, du = \int_{-\infty}^{x} \frac{\pi^*(y)}{M' q^*(y)} q(y) dy = \frac{\int_{-\infty}^{x} \pi^*(y) dy}{M' \int_{\mathcal{X}} q^*(y) dy}$$

- Also

$$\Pr[Y \text{ accepted}] = \int \Pr[\text{accepted} \,|\, y] q(y) dy = \frac{1}{\int q^*(y) dy} \int \frac{\pi^*(y)}{M' q^*(y)} q^*(y) dy = \frac{\int \pi^*(y) dy}{M' \int q^*(y) dy}$$

- Finally:

$$\Pr(Y \le x \,|\, Y \text{ accepted}) = \frac{\Pr(Y \le x \text{ and } Y \text{ accepted})}{\Pr(Y \text{ accepted})} = \frac{\frac{\int_{-\infty}^{x} \pi^*(y) dy}{M' \int_{\mathcal{X}} q^*(y) dy}}{\frac{\int_{\mathcal{X}} \pi^*(y) dy}{M' \int_{\mathcal{X}} q^*(y) dy}} = \int_{-\infty}^{x} \pi(y) dy$$

Accept/Reject Efficiency

- The acceptance probability *Pr(Y accepted)* is a measure of efficiency.
- The expected number of trials before accepting a candidate is

$$\frac{1}{\Pr(Y \text{ is accepted})}$$

- The number of trials before success is thus an unbiased estimate of *1/Pr(Y is accepted).*
- This is important as we will see later when discussing the Metropolis Hastings algorithm.

Accept/Reject Efficiency

- The number of trials before a candidate sample is accepted is a geometric distribution.

- Indeed note that:

$$\Pr\left(k^{th} \ \ proposal\ is\ accepted\right) = (1-\gamma)^{k-1}\gamma$$

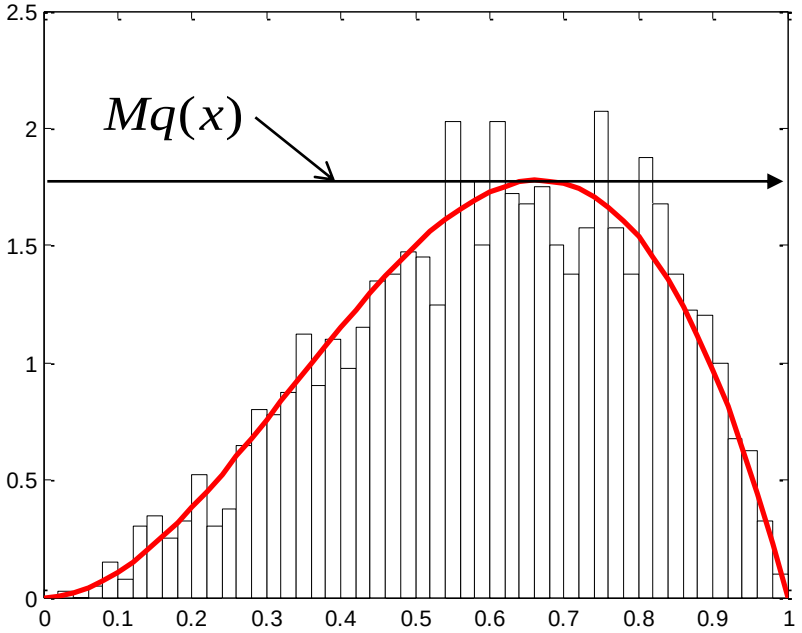
$$\gamma = \Pr(Y\ is\ accepted) = \frac{\int \pi^*(y)dy}{M'\int q^*(y)dy}$$

- The mean of the geometric distribution is $1/\gamma$, thus the number of trials before success is thus an unbiased estimate of *1/Pr(Y is accepted).*

- This is important as we will see later when discussing the Metropolis Hastings algorithm.

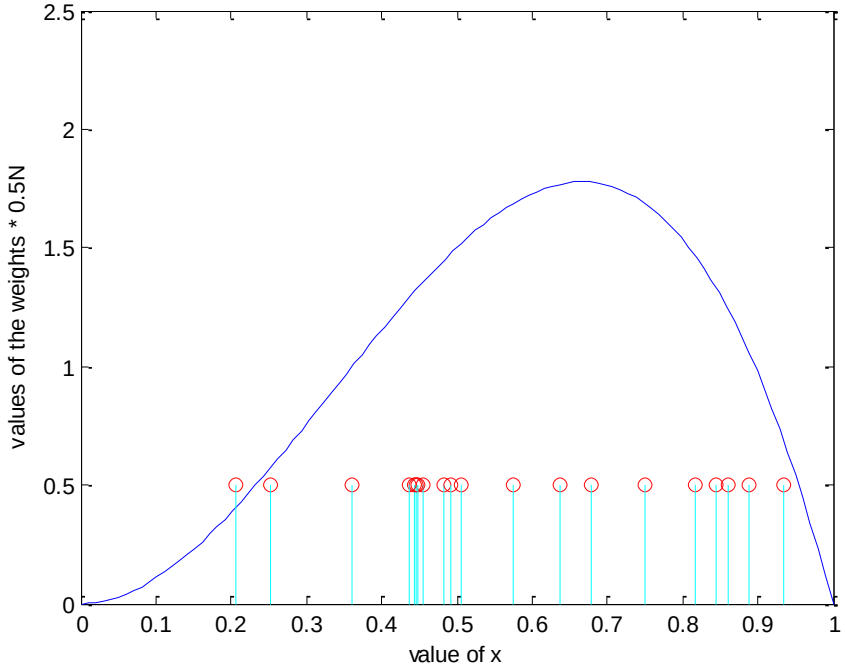
Rejection Sampling: Beta Distribution

- Consider as a target PDF $\pi(x)= \pi^*(x)= \mathcal{Beta}(3,2)$
- We take as proposal $q=q^* \sim \mathcal{U}[0,1]$ and $M'=M=48/27$



$$For\ \mathcal{Be}(x;\alpha,\beta),\ q \propto \mathcal{U}[0,1]$$

$$find: M = \sup_{x\in[0,1]} \frac{x^{\alpha-1}(1-x)^{\beta-1}}{1}$$



- For a C++ implementation of this example see here.
- Two MatLab implementations are also available.

Example: Accept-Reject Sampling

- In this example, the target distribution & the proposal distribution are

$$\pi^*(x) = \exp\left(-x^2/2\right)\left\{\sin^2(6x) + 3\cos^2(x)\sin^2(4x) + 1\right\}$$

$$q^*(x) = \exp\left(-x^2/2\right)$$

- The proposal distribution is a standard normal.
- The maximum of each of the trigonometric functions inside the target is *1*
  - Hence a reasonable upper bound for the ratio of the target to the proposal is *5*

$$\sup\left(\frac{\pi^*(x)}{q^*(x)}\right) \le 5$$

The adjoining figure illustrates the sampling procedure. $M' = 5$

1. Plot the target distribution and 5 times the proposal distribution to show the manner in which the proposal envelopes the target
2. Generate a sample x from the normal distribution
3. Evaluated acceptance probability as

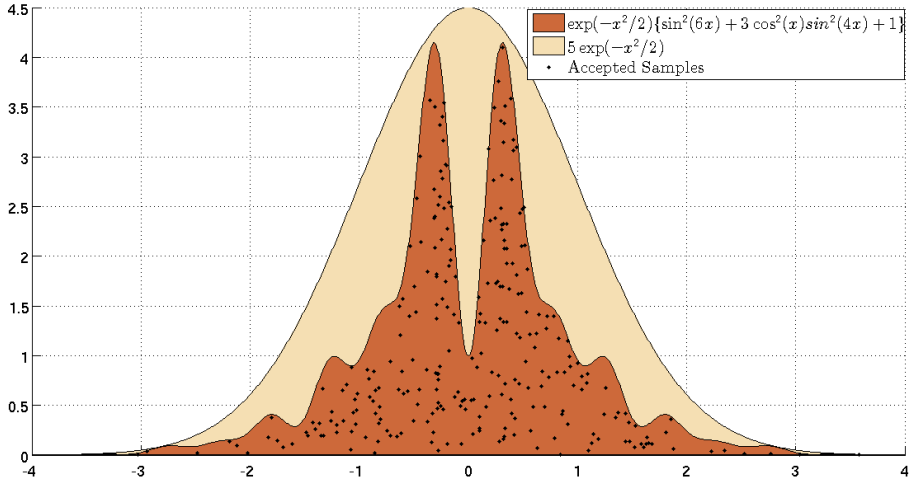
$$\frac{\pi^*(x)}{5q^*(x)}$$

4. Generate a uniform random number U. If it is less than the acceptance probability we accept the sample.
5. Plot the point as

$$(x, 5Uq^*(x))$$

This picture resulted from 500 samples drawn from the proposal distribution

Link to MatLab implementation



Example: Accept-Reject Sampling

- The target distribution and the proposal distribution are given as

$$\pi^*(x) = \exp\left(-x^2/2\right)\left\{\sin^2(6x) + 3\cos^2(x)\sin^2(4x) + 1\right\}$$

$$q^*(x) = \exp\left(-x^2/2\right)$$

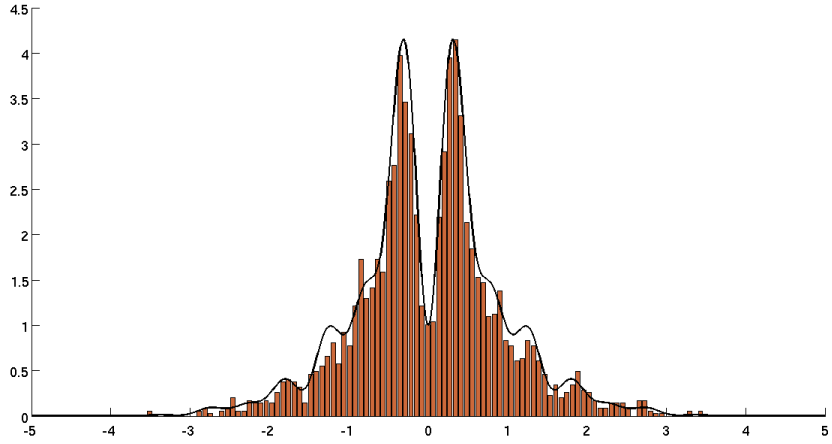
- The proposal distribution is a standard normal.
- The maximum of each of the trigonometric functions inside the target is *1*
  - Hence a reasonable upper bound for the ratio of the target to the proposal is *5*

$$\sup\left(\frac{\pi^*(x)}{q^*(x)}\right) \leq 5$$

$$M' = 5$$

Link to MatLab implementation

The adjoining figure shows the histogram plot obtained from 5000 samples drawn from the proposal distribution

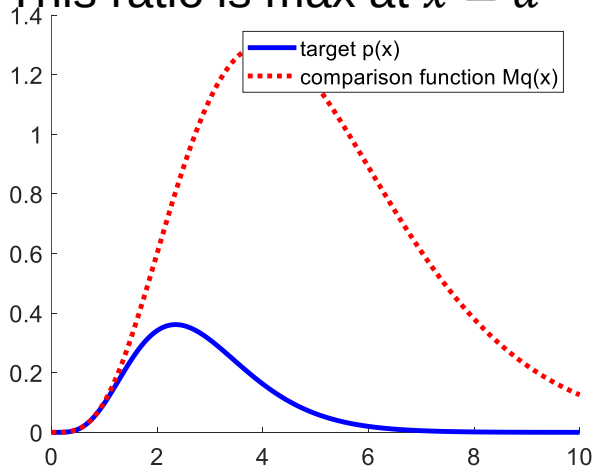


Sampling from the Gamma

- Consider sampling from the Gamma distribution $Ga(x|a,\lambda) = \frac{\lambda^a}{\Gamma(a)} x^{a-1} e^{-\lambda x}$
- One can show that if $X_i \sim \mathcal{E}xp(\lambda)(iid)$ and $Y = X_1 + \cdots + X_k$, then $Y \sim Ga(x|k,\lambda)$.
- This transformation cannot be used for non-integer a. In this case we can sample from $Ga(x|a,\lambda)$ using accept/reject with a proposal $\mathrm{q}(\mathrm{x}) = Ga(x|k = \lfloor a \rfloor, \lambda - 1)$. With this choice

$$\frac{p(x)}{q(x)} = \frac{x^{a-1}\lambda^a e^{-\lambda x}}{x^{k-1}(\lambda-1)^k e^{-(\lambda-1)x}} \frac{\Gamma(k)}{\Gamma(a)} = \frac{\Gamma(\lfloor a \rfloor)\lambda^a}{\Gamma(a)(\lambda-1)^k} x^{a-k} e^{-x}$$

- This ratio is max at $x = a - \lfloor a \rfloor$. Thus $M = \frac{Ga(a-k|a,\lambda)}{Ga(a-k|k,\lambda-1)}$.



Link to MatLab implementation
From PMTK3

Acceptance-Rejection for the Gamma

- We revisit sampling from $Ga(x|a,\lambda=1) \equiv f(x) = \frac{1}{\Gamma(a)} x^{a-1} e^{-x}, a > 1$. A suitable proposal is the Cauchy $h(x|b,c) = \frac{\sqrt{c}/\pi}{1+c(x-b)^2}$. The CDF is given as $H(x) = \frac{1}{2} + \frac{1}{\pi} arctan\left(\sqrt{c}(x-b)\right)$. The inverse is $x = \frac{1}{\sqrt{c}} tan\left(\pi(H(x) - 0.5)\right) + b$.
- Since we have an analytical expression of the CDF of h(x) we can sample easily from it. To use it as a proposal distribution, we generalize to be sure that is nowhere less than the Gamma.
- You can show $(a > 1, x \geq 0)$ the following inequality holds:

$$\mathrm{f}(x) = \frac{1}{\Gamma(a)} x^{a-1} e^{-x} \leq \frac{1}{\Gamma(a)} \frac{e^{-(a-1)}(a-1)^{a-1}}{1 + \frac{\left(x-(a-1)\right)^2}{2a-1}} \equiv g(x)$$

$$= \frac{1}{\Gamma(a)} e^{-(a-1)}(a-1)^{a-1} \pi \sqrt{2a-1}\, h\left(x|a-1, c = \frac{1}{2a-1}\right)$$

- Thus we have shown that:

$$f(x) \leq Kh\left(x|b = a-1, c = \tfrac{1}{2a-1}\right), \text{ where } K = \frac{1}{\Gamma(a)} e^{-(a-1)}(a-1)^{a-1}\pi\sqrt{2a-1}.$$

- U. Dieter & J. Ahrens, Acceptance Rejection Techniques for Sampling from the Beta and Gamma Distributions, 1974

Adaptive Rejection for the Gamma

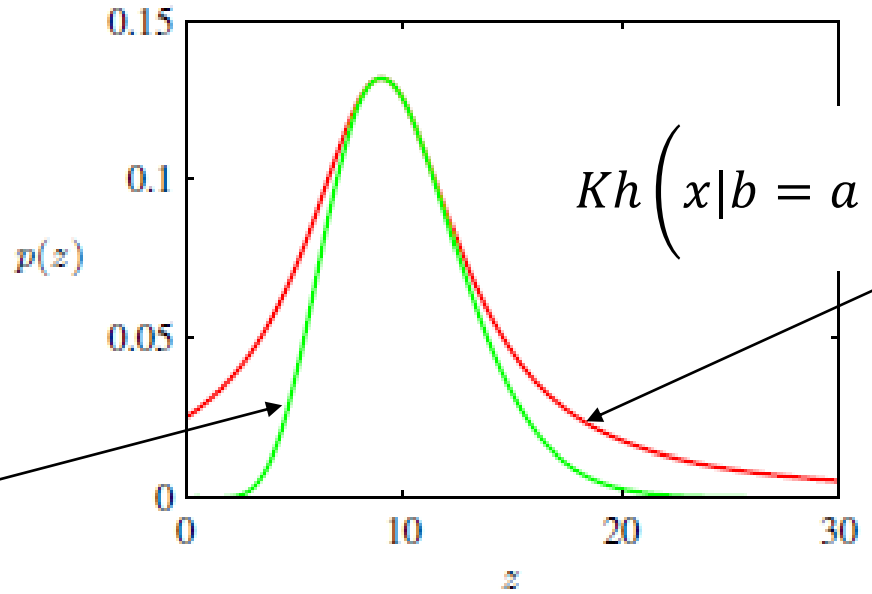
The overall accept-reject algorithm now takes the form:

1. Set $b \leftarrow a-1, A \leftarrow a+b$, and s $\leftarrow \sqrt{A}$. // $b = a-1, A = 2a-1, s = \sqrt{2a-1}$.
2. Generate $u \sim \mathcal{U}_{(0,1)}$. Set $t \leftarrow s\ tan\big(\pi(u-0.5)\big)$ and $x \leftarrow b+t$. // x =sample from the Cauchy.
3. If $x < 0$ go to 2.
4. Generate u'. If $u' > exp\left(b\ ln\frac{x}{b} - t + ln\left(1+\frac{t^2}{A}\right)\right)$ go to Step 2. Otherwise deliver x.

- U. Dieter & J. Ahrens, Acceptance Rejection Techniques for Sampling from the Beta and Gamma Distributions, 1974

Adaptive Rejection for the Gamma

- We sample from $Ga(x|a,\lambda) = \frac{1}{\Gamma(a)} x^{a-1}\lambda^a e^{-\lambda x}, a > 1$,with a proposal distribution the Cauchy $h(x) = \frac{\sqrt{c}/\pi}{1+c(x-b)^2}$, b $= a-1, c = \frac{1}{2a-1}$. With this choice the proposal distribution (times a constant K) is nowhere less than the Gamma.



$$Kh\left(x|b=a-1, c=\frac{1}{2a-1}\right)$$

$$Ga(x|a,\lambda=1) = \frac{1}{\Gamma(a)} x^{a-1}e^{-x}$$

- U. Dieter & J. Ahrens, Acceptance Rejection Techniques for Sampling from the Beta and Gamma Distributions, 1974

Alternative Rejection Sampling (RS) Algorithm

- In the standard accept/reject algorithm, the candidate is sampled before u. This is not necessary.
- (Beskos et al., 2005): Let $(Y_n, I_n)_{n\geq 1}$ be a sequence of i.i.d. random variables in $\mathcal{X}\times\{0,1\}$ such that $Y_1 \sim q$

$$\Pr(I_1 = 1 \mid Y_1 = y) = \frac{\pi^*(y)}{Cq^*(y)} \; \forall y \in \mathcal{X}$$

Define $\tau = \min\{i \geq 1 : I_i = 1\}, \text{then } Y_\tau \sim \pi$

- This scheme does not assume any order for the simulation of Y and I and, besides the conditional property given in the proposition, does not restrict the construction of I.
- This result is useful if we can construct conditions for the acceptance or rejection of the current proposed element Y from minimal information about it.

A. Beskos and G. Roberts, The Annals of Applied Probability, Vol 15(4) (2005) pp. 2422–2444.

Accept/Reject as a Sampling Through Composition

- One can re-write:

$$\pi(x) = \sum_{i=1}^{\infty} p_i \pi(x),\ p_i = p(1-p)^{i-1}\ and$$

$$p = \Pr\left(U \leq \frac{\pi^*(X)}{Mq^*(X)} \right)$$

- Instead of simulating from the geometric distribution $\mathcal{Geo}$(p) directly which is impossible, one simulates an element which admits this probability distribution (see Peterson and Kronmal, 1982)

Arthur V. Peterson, Jr. and Richard A. Kronmal, On Mixture Methods for the Computer Generation of Random Variables, The American Statistician, Vol. 36, No. 3, Part 1 (Aug., 1982), pp. 184-191

Accept/Reject Example

- The target π is given by $\pi(x) \propto \pi^*(x) = e^{-\frac{x^2}{2}} m(x),\ m(x) \le M\ \forall x \in \mathcal{X}$

- If we use $q(x) = q^*(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\ (normalized)$

  then we have: $\frac{\pi^*(x)}{q^*(x)} \le M' = \sqrt{2\pi} M\ and\ \Pr(Y\ accepted) = \frac{\int_{\mathcal{X}} \pi^*(y)dy}{M'}$

- If instead we use $q^*(x) = e^{-\frac{x^2}{2}}$, then we have:

$$\frac{\pi^*(x)}{q^*(x)} \le M\ and\ \Pr(Y\ accepted) = \frac{\int_{\mathcal{X}} \pi^*(y)dy}{M\int_{\mathcal{X}} q^*(y)dy} = \frac{\int_{\mathcal{X}} \pi^*(y)dy}{M\sqrt{2\pi}} = \frac{\int_{\mathcal{X}} \pi^*(y)dy}{M'}$$

- Once again we see that we do not need the normalizing constant of q*.

Accept/Reject – Bayesian Estimation Example

- Consider a Bayesian model: prior $\pi(x)$ and likelihood $f(x|\theta)$

- The posterior distribution is given by:

$$\pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\int f(x|\theta)\pi(\theta)d\theta} \propto \pi^*(\theta|x), \text{ where } \pi^*(\theta|x) = f(x|\theta)\pi(\theta)$$

- We can use the prior distribution as a candidate distribution $q(\theta) = q^*(\theta) = \pi(\theta)$ as long as

$$\sup_{\theta\in\Theta} \frac{\pi^*(\theta|x)}{q^*(\theta)} = \sup_{\theta\in\Theta} f(x|\theta) \le M, \text{ i.e. } M = f(x|\theta^{MLE})$$

- The likelihood is often bounded and can use the rejection procedure. Samples from $\pi(x)$ are accepted with probability

$$\frac{\int_x \pi^*(x)dx}{M\int_x q^*(x)dx} = \frac{\int_\Theta \pi(\theta)f(x|\theta)d\theta}{M\int_\Theta \pi(\theta)d\theta} = \frac{\int_\Theta \pi(\theta)f(x|\theta)d\theta}{M}$$

Rejection Sampling in High-Dimensions

- Consider the following target distribution

$$\pi(\boldsymbol{x}) = \pi^*(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) = \frac{1}{(2\pi)^{d/2}} e^{-\frac{\sum_{i=1}^{d} x_i^2}{2}}$$

- We take the following reasonably good proposal distribution ($\sigma>1$)

$$q^*(\boldsymbol{x}) = \mathcal{N}(0, \sigma^2 \boldsymbol{I}) = \frac{1}{(2\pi\sigma^2)^{d/2}} e^{-\frac{\sum_{i=1}^{d} x_i^2}{2\sigma^2}}$$

- Note that: $\frac{\pi^*(\boldsymbol{x})}{q^*(\boldsymbol{x})} = \sigma^d e^{-\frac{1}{2}\sum_{i=1}^{d} x_i^2 (1-\frac{1}{\sigma^2})} \leq \sigma^d \equiv M$

$$Pro\left[Proposal\ Accepted\right] = \frac{Z}{M\int q^*(\boldsymbol{y})d\boldsymbol{y}} = \frac{1}{\sigma^d} \to 0 \quad when\ d \to \infty$$

The efficiency of RS decreases exponentially with dimensionality

Sampling From an Arbitrary PDF

- Inverting the CDF:
  - Not practical in high dimensions
  - Often the CDF is not known (e.g. when the density is known up to a normalizing factor)

- Rejection sampling:
  - It only requires π or q to be known up to a normalizing constant
  - We need to have a proposal density $q^*(\mathbf{x})$ from which we can draw easily samples. This is not feasible in high dimensions.
  - We need to find a bounding constant

$$M \geq \frac{\pi^*(\mathbf{x})}{q^*(\mathbf{x})}, \forall \mathbf{x}$$

  - How do you construct q*(x) automatically?

Envelope Rejection Method

- Assume we have: $q_L^*(x) \le \pi^*(x) \le M' q^*(x)$

- We can modify the accept/reject algorithm as follows:

  I. Sample Y~q and u~$\mathcal{U}$(0,1).

  II. If $u \le \frac{q_L^*(Y)}{M' q^*(Y)}$, then return Y;

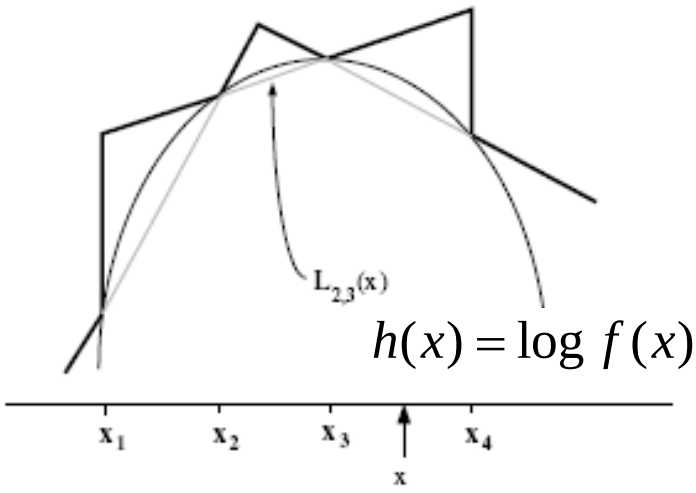
  III. Otherwise, accept *Y* if $u \le \frac{\pi^*(Y)}{M' q^*(Y)}$, otherwise return to step I.

Log-Concave Densities

- Consider the class of univariate log-concave densities, i.e. we have:

$$\frac{\partial^2 \log \pi(x)}{\partial x^2} < 0, \pi(x) = f(x) \Big/ \int_{\mathcal{X}} f(x)dx$$

- The idea is to construct automatically a piecewise linear upper and lower bounds for the target PDF.



$\{x_i\}_{i=0}^{n+1}$ in the support of $\pi(x)$

1. *Define the line* $L_{i,i+1}$ *through* $(x_i, h(x_i))$ *and* $(x_{i+1}, h(x_{i+1}))$ *as shown in the Fig.*

2. *Define on* $[x_0, x_{n+1}]$

$$\bar{h}_n(x) = \min\left(L_{i-1,i}(x), L_{i+1,i+2}(x)\right), \underline{h}_n(x) = L_{i,i+1}(x)$$

On $[x_0, x_{n+1}]^c$: $\underline{h}_n(x) = -\infty$ *and*

$\bar{h}_n(x) = \min\left(L_{0,1}(x), L_{n,n+1}(x)\right)$ *such that*

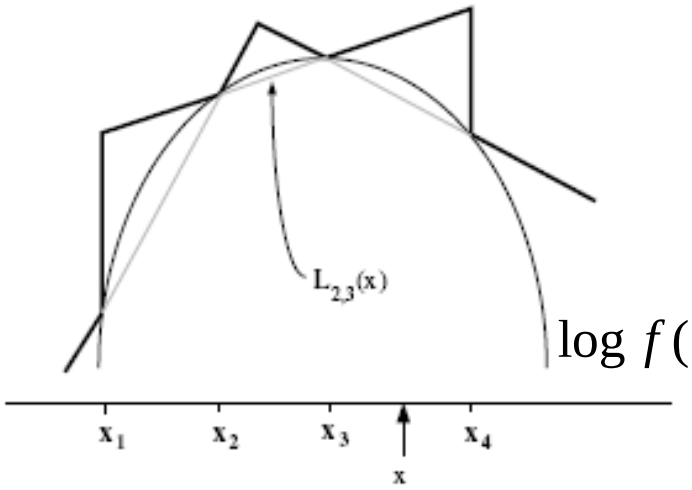
$$\underline{f}_n(x) = \exp \underline{h}_n(x) \le \pi(x) \le \exp \bar{h}_n(x) = \bar{f}_n(x) = \bar{w}_n g_n(x)$$

Log-Concave Densities

- Consider the class of univariate log-concave densities, i.e. we have:

$$\frac{\partial^2 \log \pi(x)}{\partial x^2} < 0, \pi(x) = f(x) \Big/ \int_x f(x)dx$$

- The idea is to construct automatically a piecewise linear upper and lower bounds for the target PDF.



$$\underline{h}_n(x) \le h(x) \equiv \log \pi(x) \le \overline{h}_n(x) \Rightarrow$$

$$\underline{f}_n(x) = e^{\underline{h}_n(x)} \le f(x) \le e^{\overline{h}_n(x)} \le \overline{f}_n(x) \Rightarrow$$

$$\underline{f}_n(x) \le f(x) \le \overline{f}_n(x) = \overline{w}_n g_n(x),$$

$\overline{w}_n$ is the normalized constant of $\overline{f}_n(x)$ and $g_n(x)$ is a density easy to sample from.

Adaptive Rejection Sampling

- Let S_n be a set of points x_i, $i=0,1,\ldots,n+1$. Initialize $n=0$ and S_0.
- At iteration n≥1
  - Generate Y~$g_n(x)$, u~$\mathcal{U}[0,1]$
  - If $U \leq \frac{f_n(Y)}{w_n f_n(Y)}$, then return Y;

    Otherwise, update $S_{n+1} = S_n \cup \{Y\}$

Gilks, WR, Wild, P, "Adaptive rejection sampling for Gibbs sampling" Applied Statistics, Vol. 41 (1992), pp. 337-348. Get it from JSTOR

Adaptive Rejection Sampling: Example

- Consider *n* data (x_i, Y_i) of integer value data Y_i where

$$Y_i \mid x_i \sim \textit{Poisson}(\exp\{a + bx_i\}) = \frac{e^{(a+bx_i)y_i} e^{-e^{(a+bx_i)}}}{y_i!}$$

- Consider as the prior the following:

$$\pi(a,b) = \mathcal{N}(a;0,\sigma^2)\mathcal{N}(b;0,\tau^2)$$

- We have for the posterior:

$$\pi(a,b \mid x_{1:n}, y_{1:n}) \propto \exp\left\{a\sum y_i + b\sum y_i x_i - e^a \sum e^{x_i b}\right\} e^{-\frac{a^2}{2\sigma^2}} e^{-\frac{b^2}{2\tau^2}} \Rightarrow$$

$$\textit{Full Conditional}: \log \pi(a \mid x_{1:n}, y_{1:n}, b) = a\sum y_i - e^a \sum e^{x_i b} - \frac{a^2}{2\sigma^2} + \textit{non } a\textit{-dependent terms} \Rightarrow$$

$$\frac{\partial^2 \log \pi(a \mid x_{1:n}, y_{1:n}, b)}{\partial a^2} = -e^a \sum e^{x_i b} - \sigma^{-2} < 0$$

- Thus $\pi(a \mid x_{1:n}, y_{1:n}, b)$ and similarly $\pi(b \mid x_{1:n}, y_{1:n}, b)$ are log-concave, and adaptive rejection sampling can be applied.

Monahan's Accept/Reject Method

- Assume that we want to sample from a CDF of the form:

$$F(x) = P(X \le x) = \frac{H\left(-G(x)\right)}{H(-1)}$$

- Here *G(x)* is a given CDF and

$$H(x) = \sum_{n=1}^{\infty} a_n x^n,$$

$$with: \; 1 = a_1 \ge a_2 \ge \ldots \ge 0$$

- We want to achieve this by taking samples only from *G* and $\mathcal{U}[0,1]$.

J. F. Monahan, Extension of von Newmann's method for generating random variables, Mathematics of Computation, 33(147) (1979) 1065-1069.

Monahan's Accept/Reject Method

- For example, let

$$F(x)=1-\cos\frac{\pi x}{2}=\frac{\left(-x^2\right)+\frac{\pi^2}{48}\left(-x^2\right)^2+\ldots+\frac{\pi^{2i-2}}{2^{2i-3}(2i)!}\left(-x^2\right)^i+\ldots}{(-1)+\frac{\pi^2}{48}(-1)^2+\ldots+\frac{\pi^{2i-2}}{2^{2i-3}(2i)!}(-1)^i+\ldots}$$

- To derive this note that:

$$\cos x=\sum_{i\geq 0}(-1)^i\frac{x^{2i}}{(2i)!}\Rightarrow 1-\cos\frac{\pi x}{2}=1-\sum_{i\geq 0}(-1)^i\frac{\left(\frac{\pi x}{2}\right)^{2i}}{(2i)!}=-\sum_{i\geq 1}\frac{\pi^{2i}}{2^{2i}(2i)!}\left(-x^2\right)^i\Rightarrow$$

$$1-\cos\frac{\pi x}{2}=\frac{-\sum_{i\geq 1}\frac{\pi^{2i-2}\left(-x^2\right)^i}{2^{2i-3}(2i)!}}{\left(\frac{\pi^2}{8}\right)^{-1}}=\frac{\sum_{i\geq 1}\frac{\pi^{2i-2}}{2^{2i-3}(2i)!}\left(-x^2\right)^i}{\sum_{i\geq 1}\frac{\pi^{2i-2}}{2^{2i-3}(2i)!}(-1)^i}\ (\textit{use } x=1 \textit{ for the denominator approximation})$$

$$\textit{Thus}: G(x)=x^2,\ H(x)=x+\frac{\pi^2}{48}x^2+\ldots+\frac{\pi^{2i-2}}{2^{2i-3}(2i)!}x^i+\ldots$$

Monahan's Accept/Reject Method

$$F(x) = P(X \le x) = \frac{H\left(-G(x)\right)}{H(-1)}, G \text{ is a CDF}, H(x) = \sum_{n=1}^{\infty} a_n x^n, s.t. 1 = a_1 \ge a_2 \ge \ldots \ge 0$$

- The sampling algorithm is as follows:

- *Repeat*
  - *Generate* $X \sim G$ *and set* $K \leftarrow 1$
  - *Repeat*
    - *Generate* $U \sim G$ *and* $V \sim \mathcal{U}[0,1]$
    - *If* $U \le X$ *and* $V \le \frac{a_{K+1}}{a_K}$ *then* $K \leftarrow K+1$, *otherwise stop.*

Until K *odd, return* X

Monahan's Accept/Reject Method

- We define the event

$$A_n : X = \max\left(X, U_1, U_2, ..., U_n\right) \text{ and } Z_1 = Z_2 = \cdots = Z_n = 1$$

- The U_i's are the random variables generated in the inner loop of the algorithm and the Z_i's are Bernoulli random variables equal to consecutive values

$$\mathbb{I}_{V \le \frac{a_{K+1}}{a_K}}$$

- We can show:

$$P\left(X \le x, A_n\right) = a_n G(x)^n,$$

$$P\left(X \le x, A_n, A_{n+1}^c\right) = P\left(X \le x, A_n\right) - P\left(X \le x, A_n, A_{n+1}\right)$$

$$= a_n G(x)^n - a_{n+1} G(x)^{n+1}$$

- Indeed notice that:

$$P\left(X \le x, A_n\right) = \left(1 \times G(x)\right) \times \left(\frac{a_2}{a_1} \times G(x)\right) \times ... \times \left(\frac{a_n}{a_{n-1}} \times G(x)\right) = a_n G(x)^n$$

Monahan's Accept/Reject Method

- The probability that X is accepted is:

$$P(K\ odd) = \sum_{n=1}^{\infty} a_n (-1)^{n+1} = -H(-1)$$

- This can be shown simply using $P\left(X \le \infty, A_n, A_{n+1}^c \right) = a_n - a_{n+1}$

$$P(K\ odd) = \left(a_1 - a_2\right) + \left(a_3 - a_4\right) + \ldots = \sum_{n=1}^{\infty} a_n (-1)^{n+1} = -H(-1)$$

- The returned X has then a distribution:

$$F(x) = P(X \le x \mid X\ returned) = \frac{P(X \le x, X\ returned)}{P(X\ returned)}$$

Monahan's Accept/Reject Method

- The returned X has then a distribution:

$$P(X \le x \mid X \text{ returned}) = \frac{P(X \le x, X \text{ returned})}{P(X \text{ returned})} = \frac{\sum_{n=1,3,5,..} P(X \le x, A_n, A_{n+1}^c)}{P(X \text{ returned})}$$

$$= \frac{\left(a_1 G(x)^1 - a_2 G(x)^2\right) + \left(a_3 G(x)^3 - a_4 G(x)^4\right) + \ldots}{-H(-1)} = \frac{\sum_{n=1}^{\infty} a_n G(x)^n (-1)^{n+1}}{-H(-1)} = \frac{-H\left(-G(x)\right)}{-H(-1)}$$

$$F(x) = P(X \le x \mid X \text{ returned}) = \frac{H\left(-G(x)\right)}{H(-1)}$$

Conditional Monte Carlo

- Let $\ell = \mathbb{E}\left[H(\boldsymbol{X})\right] = \int H(\boldsymbol{x})\, p(\boldsymbol{x}) d\boldsymbol{x}$ be some expected performance measure of a computer simulation model, where ***X*** is the input random variable (vector) with a pdf *p(**x**)* and *H(**X**)* is the sample performance measure (output random variable).

- Suppose that there is a random variable (or vector), ***Y*** ~ *g(**y**), such that the conditional expectation* $\mathbb{E}$ [H(**X**) I **Y** = **y**] *can be computed* analytically.

- Since $\ell = \mathbb{E}\left[H(\boldsymbol{X})\right] = \mathbb{E}\left[\mathbb{E}\left[H(\boldsymbol{X}) \mid \boldsymbol{Y}\right]\right]$, it follows that $\mathbb{E}$ **[H(X) I Y]** ***is an unbiased estimator of ℓ.*** *Furthermore, it is readily seen*

$$Var\left[\mathbb{E}\left(H(\boldsymbol{X}) \mid Y\right)\right] \le Var\left[H(\boldsymbol{X})\right]$$

- **Thus using the random variable $\mathbb{E}$ [H(X) I Y] *instead of H(X), leads to variance reduction.***

- The equation above is derived from

$$Var[U] = \mathbb{E}\left[Var(U \mid V)\right] + Var\left[\mathbb{E}(U \mid V)\right]$$

  for any pair of random variables *(U, V).*

- The conditional Monte Carlo idea is referred to as ***Rao-Blackwellization*****.**

Conditional Monte Carlo

- Step 1: Generate a sample $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_N$ from $g(\boldsymbol{y})$.
- Step 2: Calculate $\mathbb{E}\left[H(\boldsymbol{X}) \mid Y_k\right], k = 1, 2, \ldots, N$ analytically
- Step 3: Estimate $\ell = \mathbb{E}\left[H(\boldsymbol{X})\right]$ by

$$\hat{\ell}_c = \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}[H(X) | Y_k]$$

The Algorithm requires that a random variable $\boldsymbol{Y}$ be found, such that $\mathbb{E}$ [H(**X**) |**Y** = **y**] is known analytically for all **y.** Moreover, for the Algorithm to be of practical use, the following conditions must be met:

(a) $\boldsymbol{Y}$ should be easy to generate.

(b) $\mathbb{E}$[H(**X**)|**Y** =**y**] should be readily computable for all values ***y.***

(c) $\mathbb{E}$[Var(H(**X**)I**Y)**] should be large relative to Var($\mathbb{E}$[H(**X**)I **Y**]).

Conditional Monte Carlo: Example – Random Sums

- Consider the estimation of

$$\ell = \Pr\left[S_R \le x\right] = \mathbb{E}\left[\mathbb{I}_{S_R \le x}\right], \ \ where: S_R = \sum_{i=1}^{R} X_i$$

- *R* is a random variable with a given distribution and the $\{X_i\}$ are i.i.d. with $X_i \sim F$ and independent of *R*. Let F^r be the cdf of the random variable S_r, for fixed $R = r$.

- Noting that

$$F^r(x) = \Pr\left[\sum_{i=1}^{r} X_i \le x\right] = F\left[x - \sum_{i=2}^{r} X_i\right]$$

- We obtain

$$\ell = \mathbb{E}\left[\mathbb{E}\left[\mathbb{I}_{S_R \le x} \mid \sum_{i=2}^{R} X_i\right]\right] = \mathbb{E}\left[F\left(x - \sum_{i=2}^{R} X_i\right)\right]$$

- As an estimator of ℓ based on conditioning, we can take

$$\hat{\ell}_c = \frac{1}{N}\sum_{k=1}^{N} F\left(x - \sum_{i=2}^{R_k} X_{ki}\right)$$

Stratified Sampling

- We wish to estimate

$$\ell = \mathbb{E}\left[H(\boldsymbol{X})\right] = \int H(\boldsymbol{x})\, p(\boldsymbol{x}) d\boldsymbol{x}$$

- Assume that there exists a random variable Y taking values in $\{1, \ldots, m\}$, say, with known probabilities $\{p_i, i = 1, \ldots, m\}$, and we assume that it is easy to sample from the conditional distribution of $\boldsymbol{X}$ given Y.

- The events $\{Y = i\}$, $i = 1, \ldots, m$ form disjoint subregions, or *strata, of the sample space Ω, hence the name stratification.*

- *Using the* conditioning formula, we can write

$$\ell = \mathbb{E}\Big[\mathbb{E}\left[H(\boldsymbol{X} \mid Y)\right]\Big] = \sum_{i=1}^{m} p_i \mathbb{E}\left[H(\boldsymbol{X}) \mid Y = i\right]$$

R. Y. Rubinstein, D. P. Kroese, Simulation and the Monte Carlo Method, 2007

Stratified Sampling Estimator

- This representation suggests that we can estimate ℓ *via the following stratified sampling estimator:*

$$\hat{\ell}^S = \sum_{i=1}^{m} p_i \frac{1}{N_i} \sum_{j=1}^{N_i} H(\boldsymbol{X}_{ij})$$

where X_{ij} is the j-th observation from the conditional distribution of $\boldsymbol{X}$ given $Y = i$. Here N_i *is the sample size assigned to the i-th stratum.*

- *The variance of the stratified sampling* estimator is given by

$$Var\left[\hat{\ell}^S\right] = \sum_{i=1}^{m} \frac{p_i^2}{N_i} Var[H(X)|Y=i] = \sum_{i=1}^{m} \frac{p_i^2 \sigma_i^2}{N_i}$$

where $\sigma_i^2 = Var\left[H(\boldsymbol{X}) \mid Y = i\right]$

- How the strata should be chosen depends very much on the problem at hand. However, for a given particular choice of the strata, the sample sizes $\{N_i\}$ *can be obtained in an* optimal manner.

Stratified Sampling

- Assuming that a maximum number of N samples can be collected, that is, $\sum_{i=1}^{m} N_i = N$, the optimal value of N_i is given by

$$N_i^* = N \frac{p_i \sigma_i}{\sum_{j=1}^{m} p_j \sigma_j}$$

which gives a minimal variance of

$$Var\left[\hat{\ell}^{*s}\right] = \frac{1}{N}\left(\sum_{i=1}^{m} p_i \sigma_i\right)^2$$

- This theorem asserts that the minimal variance of $\hat{\ell}^s$ is attained for sample sizes N_i that are proportional to $p_i\sigma_i$. A difficulty is that although the probabilities p_i are assumed to be known, the standard deviations $\{\sigma_i\}$ are usually unknown.

- In practice, one would estimate the $\{\sigma_i\}$ from "pilot" runs and then proceed to estimate the optimal sample sizes, N_i^*, from the equation above.

- A simple stratification procedure, which can achieve variance reduction without requiring prior knowledge of σ_i^2 and $H(\boldsymbol{X})$, is presented next.

Stratified Sampling

- Let the sample sizes N_i be proportional to p_i, that is, $N_i = p_i N$, $i = 1, \ldots m$. Then

$$Var\left[\hat{\ell}^s\right] \leq Var\left[\hat{\ell}\right]$$

- Substituting $N_i = p_i N$ in $Var\left[\hat{\ell}^s\right] = \sum_{i=1}^{m} \frac{p_i^2 \sigma_i^2}{N_i}$ yields $Var\left[\hat{\ell}^s\right] = \frac{1}{N} \sum_{i=1}^{m} p_i \ \sigma_i^2$

- The result now follows from

$$NVar\left[\hat{\ell}\right] = Var[H(X)] \geq \mathbb{E}[Var(H(X)|Y)] = \sum_{i=1}^{m} p_i \ \sigma_i^2 = NVar\left[\hat{\ell}^s\right]$$

where we used

$$Var\left[H(\boldsymbol{X})\right] = \mathbb{E}\left[Var\left(H(\boldsymbol{X})|Y\right)\right] + Var\left[\mathbb{E}\left(H(\boldsymbol{X})|Y\right)\right] \geq \mathbb{E}\left[Var\left(H(\boldsymbol{X})|Y\right)\right]$$

and

$$\sigma_i^2 = Var\left[H(\boldsymbol{X})|Y = i\right]$$

Systematic Sampling Method

- The proposition in the slide before states that the estimator $\hat{\ell}^S$ *is more accurate than the conditional MC (CMC) estimator* $\hat{\ell}$.

- It effects stratification by favoring those events {Y = i} *whose probabilities* p_i *are largest.* Intuitively, this cannot, in general, be an optimal assignment, since information on σ_i^2 *and H(**X**) is ignored.*

- In the special case of equal weights (p_i =1/m and N_i =N/m), *the estimator*

$$\hat{\ell}^S = \sum_{i=1}^{m} p_i \frac{1}{N_i} \sum_{j=1}^{N_i} H(\boldsymbol{X}_{ij})$$

reduces to $\hat{\ell}^S = \frac{1}{N} \sum_{i=1}^{m} \sum_{j=1}^{N/m} H(\boldsymbol{X}_{ij})$

and the method is known as the *systematic sampling method*.

Stratified Sampling

- The stratification process is more obvious when partition of the stochastic space is possible:

$$\ell = \mathbb{E}\left[H(\boldsymbol{X})\right] = \int_D H(\boldsymbol{x})\, p(\boldsymbol{x}) d\boldsymbol{x} = \sum_{m=1}^{M} Z_m \int_{D_m} H(\boldsymbol{x})\, p_m(\boldsymbol{x}) d\boldsymbol{x},$$

$$where: p_m(\boldsymbol{x}) = \mathbb{I}_{D_m}(\boldsymbol{x}) \frac{p(\boldsymbol{x})}{Z_m} \, (conditional\ PDF),$$

$$Z_m = \Pr\left[\boldsymbol{x} \in D_m\right] = \int_{D_m} p(\boldsymbol{x}) d\boldsymbol{x},\ D = \bigcup_{m=1}^{M} D_m,\ D_m\ are\ disjoint$$

- The stratified sampling algorithm can be easily implemented assuming that the conditional PDFs and normalization factors Z_m are known.

Stratified Sampling: Algorithm

- Step 1: For *m=1,…,M*, draw N_m samples $\left\{ \boldsymbol{x}_i^{(m)} \right\}_{i=1}^{N_m}$ from the conditional PDF p_m.

- Evaluate the estimator of $\int_{D_m} H(\boldsymbol{x}) p_m(\boldsymbol{x}) d\boldsymbol{x}$ at domain m:

$$\hat{\ell}_m = \frac{1}{N_m} \sum_{i=1}^{N_m} H(\boldsymbol{x}_i^{(m)})$$

- Then the overall estimator of

$$\ell = \mathbb{E}\left[H(\boldsymbol{X}) \right] = \sum_{m=1}^{M} Z_m \int_{D_m} H(\boldsymbol{x}) p_m(\boldsymbol{x}) d\boldsymbol{x}$$

is as follows:

$$\hat{\ell} = \sum_{m=1}^{M} Z_m \hat{\ell}_m = \sum_{m=1}^{M} Z_m \frac{1}{N_m} \sum_{i=1}^{N_m} H(\boldsymbol{x}_i^{(m)})$$

Stratified Sampling: Algorithm

- The variance of the stratified sampling estimator is:

$$Var\left[\hat{\ell}\right] = \sum_{m=1}^{M} Z_m^2 Var\left[\hat{\ell}_m\right] = \sum_{m=1}^{M} Z_m^2 \frac{1}{N_m} Var_{D_m}\left[H(\boldsymbol{x})\right]$$

- Note that for the choices:

$$Z_m = 1/M,\ N_m = N/M:$$

$$Var\left[\hat{\ell}\right] = \frac{1}{MN} \sum_{m=1}^{M} Var_{D_m}\left[H(\boldsymbol{x})\right]$$

- If we select D_m such that $Var_{D_m}\left[H(\boldsymbol{x})\right]$ is small on average, i.e.

$$\frac{1}{M} \sum_{m=1}^{M} Var_{D_m}\left[H(\boldsymbol{x})\right] < Var\left[H(\boldsymbol{x})\right]$$

(e.g. $H(\boldsymbol{x})$ is relatively homogeneous within D_m), then we do achieve variance reduction.

Stratified Sampling: Example

- Consider the following case:

$$p(x) = 1,\ x \in [0,1]$$

$$H(x) = \begin{cases} 1/k, 0 \le x < 1/2 \\ k, 1/2 \le x \le 1 \end{cases}$$

- It can easily be shown that:

$$Var_{[0,1]}\left[H(x)\right] = \frac{(k^2-1)^2}{4k^2} \to \infty \ as \ k \to \infty.$$

- However, the variance of the stratified sampling estimator is zero since:

$$Var_{[0,1/2]}\left[H(x)\right] = Var_{[1/2,1]}\left[H(x)\right] = 0.$$

Stratified Sampling: Example

- Often $H(\boldsymbol{x})$ is not known explicitly.
- Then, the only way to select the partition domains D_m is by drawing samples of $\boldsymbol{x}$ and evaluating $H(\boldsymbol{x})$.
- In that respect, the applicability of the stratified sampling method is limited.