UiO : Matematisk institutt
Det matematisk-naturvitenskapelige fakultet

STK-4051/9051 Computational Statistics Spring 2021
Chapter 6 (and 5)

Instructor: Odd Kolbjørnsen, oddkol@math.uio.no

Bayesian approach

- Likelihood $f(\mathbf{y}|\theta)$
- Introduce a prior $p(\theta)$ describing knowledge about θ prior to data
- Bayes theorem:

$$f(\theta|\mathbf{y}) = \frac{f(\theta)f(\mathbf{y}|\theta)}{f(\mathbf{y})}$$

$$f(\mathbf{y}) = \int_\theta f(\theta)f(\mathbf{y}|\theta)d\theta$$

- Bayesian paradigm: All relevant information about θ is contained in the posterior distribution $p(\theta|\mathbf{y})$
  - $\hat{\theta}_{post} = E[\theta|\mathbf{y}] = \int_\theta \theta p(\theta|\mathbf{y})d\theta$
  - Credibility interval (one-dimensional): $\alpha = \Pr(a < \theta < b|\mathbf{y}) = \int_a^b \theta p(\theta|\mathbf{y})d\theta$
- Posterior: Updated knowledge based on both prior and data
- Numerical aspect: Bayesian approach change optimization to integration
- Many other integration problems both inside and outside statistics, will focus on

$$\mu = \int_\mathbf{x} h(\mathbf{x})f(\mathbf{x})d\mathbf{x}$$

- In many problems: $\mathbf{x}$ is high-dimensional

Image analysis – spatial structure

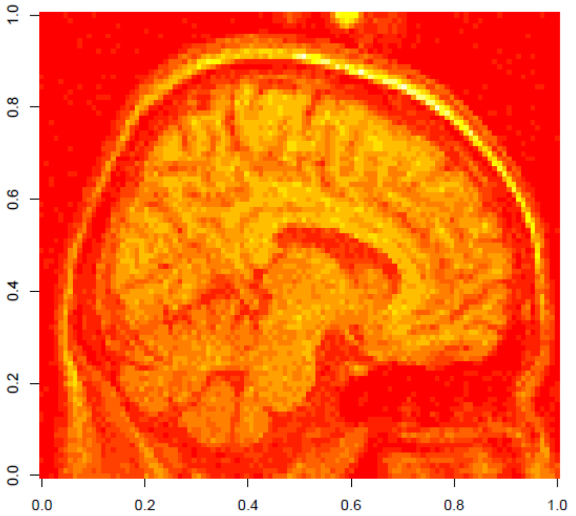
- Expect some smoothness in class-structure
- Markov Random field/Potts model:

$$\Pr(\mathbf{C}) = \Pr(C_{11}, \ldots, C_{n_1 n_2}) = \frac{1}{Z} e^{-\beta \sum_{||(i,j)-(i'j')||=1} I(C_{ij} \neq C_{i'j'})}$$

- Now interested in

$$\Pr(\mathbf{C}|\mathbf{y}) = \frac{\Pr(\mathbf{C}) \prod_{ij} f(y_{ij}|C_{ij})}{\sum_{\mathbf{C}'} \Pr(\mathbf{C}') \prod_{ij} f(y_{ij}|C'_{ij})}$$

- The sum in the denominator contains K^n terms,
  - $K =$ number of class
  - $n =$ number of pixels.
- Discrete type of "integration"

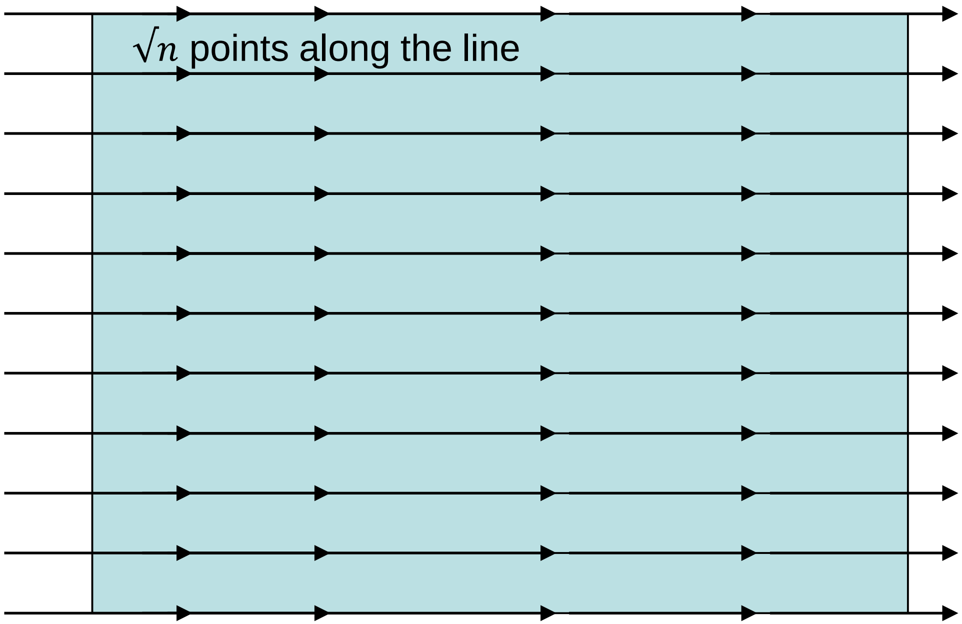


Chapter 5, 1D integration

- Newton-Cotes Quadrature
  - Riemann rule $$\left|\int_a^b f(x)\,dx - A_{\text{right}}\right| \le \frac{M_1(b-a)^2}{2n},$$
  - Trapezoidal rule $$\left|\int_a^b f(x)\,dx - A_{\text{trap}}\right| \le \frac{M_2(b-a)^3}{12n^2},$$
  - Simpsons rule $$\left|\int_a^b f(x)\,dx - A_{\text{S}}\right| \le \frac{M_4(b-a)^5}{180\,n^4}$$
  - Random Sampling $$\left|\int_a^b f(x)\,dx - A_{\text{r}}\right| \le \frac{M(b-a)}{\sqrt{n}}$$
  - **Efficient in 1D**
- Romberg integration (stable)
- Gauss quadrature (popular)
- Higher order approximations is often bounded by the maximum of the derivative of the corresponding order
- Software for exact integration
  - Mathematica & Maple

Integration in R^d (Fubini-style)

n =Number of «nodes»



$$I(S) = \iint_S f(x_1, x_2)\,dx_1 dx_2 = \int_c^d I(x_2)\,dx_2$$

$$I(x_2) = \int_a^b f(x_1, x_2)\,dx_1$$

$$\hat{I}(S) = \sum v_k \hat{I}(x_2^k) = \sum v_k \left(I(x_2^k) + E_1(x_2^k)\right)$$

$$= I(S) + E_2 + \sum v_k E_1(x_2^k)$$

The error is bounded by the integration error in each direction, but the number of nodes along each direction is $\sqrt{n}$

Curse of dimension for Quadrature formulas:

1D integral has convergence order:
$O(n^{-r})$
the Fubini integral in R^d has order:
$O(n^{-r/d})$

Monte Carlo method

- Aim (following notation from book):

$$\mu = E^{f(\mathbf{X})}[h(\mathbf{X})] = \begin{cases} \int_{\mathbf{x}} h(\mathbf{x})f(\mathbf{x}))d\mathbf{x} & \mathbf{x} \text{ continuous} \\ \sum_{\mathbf{x}} h(\mathbf{x})f(\mathbf{x}) & \mathbf{x} \text{ discrete} \end{cases}$$

- Monte Carlo:
  1. Simulate $\mathbf{X}_i \sim f(\mathbf{x}), i = 1, \ldots, n$
  2. Approximate μ by

$$\hat{\mu}_{MC} = \frac{1}{n}\sum_{i=1}^{n} h(\mathbf{x}_i)$$

- Properties:
  - Unbiased $E[\hat{\mu}_{MC}] = \mu$
  - If $X_1, \ldots, X_n$ are independent
    - Variance: $\text{var}[\hat{\mu}_{MC}] = \frac{1}{n}\text{var}[h(\mathbf{X})]$
    - Consistent: $\hat{\mu}_{MC} \to \mu$ as $n \to \infty$ if $\text{var}[h(\mathbf{X})] < \infty$
  - Estimate of variance:

$$\widehat{\text{var}}[\hat{\mu}_{MC}] = \frac{1}{n-1}\sum_{i=1}^{n}(h(\mathbf{x}_i) - \hat{\mu}_{MC})^2$$

The Fubini integral in R^d has order:

$$O\left(n^{-r/d}\right)$$

Monte Carlo method in R^d has order:

$$O\left(n^{-1/2}\right)$$

1) Independent of d
2) Does not depend on derivatives

- Main problem: How to simulate $\mathbf{X}_i \sim f(\cdot)$

Simulation techniques

- Exact methods
  - Inversion/transformation methods
  - Rejection sampling
- Approximate methods
  - Sampling importance resampling
  - Sequential Monte Carlo
  - Markov chain Monte Carlo (Chapter 7 and 8)
- Variance reduction methods
  - Importance sampling
  - Antithetic sampling
  - Control variates
  - Rao-blackwellization
  - Common random numbers

The inversion and the transformation methods

- Assume continuous distribution, density $f(x)$, CDF

$$F(x) = \int_{-\infty}^{x} f(u)du$$

- Assume $U \sim \text{Unif}[0, 1]$
- Define $X = F^{-1}(U)$:

$$\begin{aligned} \Pr(X \leq x) =& \Pr(F^{-1}(U) \leq x) \\ =& \Pr(U \leq F(x)) = F(x) \end{aligned}$$

  showing that $X \sim f(x)$!
- Assumes possible to generate U (good routines available)
- Assumes $F^{-1}(U)$ available
- Only applicable for univariate distributions
- Special case of transformation methods: $X = g(U)$
- Table 6.1: List of how to simulate most common distributions.

Table 6.1

TABLE 6.1 Some methods for generating a random variable X from familiar distributions.

Distribution	Method
Uniform	See [195, 227, 383, 538, 539, 557]. For $X \sim \text{Unif}(a, b)$; draw $U \sim \text{Unif}(0, 1)$; then let $X = a + (b - a)U$.
Normal(μ, σ^2) and Lognormal(μ, σ^2)	Draw $U_1, U_2 \sim$ i.i.d. Unif(0, 1); then $X_1 = \mu + \sigma\sqrt{-2\log U_1}\cos\{2\pi U_2\}$ and $X_2 = \mu + \sigma\sqrt{-2\log U_1}\sin\{2\pi U_2\}$ are independent $N(\mu, \sigma^2)$. If $X \sim N(\mu, \sigma^2)$ then $\exp\{X\} \sim \text{Lognormal}(\mu, \sigma^2)$.
Multivariate $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Generate standard multivariate normal vector, $\mathbf{Y}$, coordinatewise; then $\mathbf{X} = \boldsymbol{\Sigma}^{-1/2}\mathbf{Y} + \boldsymbol{\mu}$.
Cauchy(α, β)	Draw $U \sim \text{Unif}(0, 1)$; then $X = \alpha + \beta\tan\{\pi(U - \frac{1}{2})\}$.
Exponential(λ)	Draw $U \sim \text{Unif}(0, 1)$; then $X = -(\log U)/\lambda$.
Poisson(λ)	Draw $U_1, U_2, \ldots \sim$ i.i.d. Unif(0, 1); then $X = j - 1$, where j is the lowest index for which $\prod_{i=1}^{j} U_i < e^{-\lambda}$.
Gamma(r, λ)	See Example 6.1, references, or for integer r, $X = -(1/\lambda)\sum_{i=1}^{r}\log U_i$ for $U_1, \ldots, U_r \sim$ i.i.d. Unif(0, 1).
Chi-square (df $= k$)	Draw $Y_1, \ldots, Y_k \sim$ i.i.d. $N(0, 1)$, then $X = \sum_{i=1}^{k} Y_i^2$; or draw $X \sim \text{Gamma}(k/2, \frac{1}{2})$.
Student's t (df $= k$) and $F_{k,m}$ distribution	Draw $Y \sim N(0, 1)$, $Z \sim \chi^2_k$, $W \sim \chi^2_m$ independently, then $X = Y/\sqrt{Z/k}$ has the t distribution and $F = (Z/k)/(W/m)$ has the F distribution.
Beta(a, b)	Draw $Y \sim \text{Gamma}(a, 1)$ and $Z \sim \text{Gamma}(b, 1)$ independently; then $X = Y/(Y + Z)$.
Bernoulli(p) and Binomial(n, p)	Draw $U \sim \text{Unif}(0, 1)$; then $X = 1_{\{U<p\}}$ is Bernoulli(p). The sum of n independent Bernoulli(p) draws has a Binomial(n, p) distribution.
Negative Binomial(r, p)	Draw $U_1, \ldots, U_r \sim$ i.i.d. Unif(0, 1); then $X = \sum_{i=1}^{r}\lfloor(\log U_i)/\log\{1-p\}\rfloor$, and $\lfloor\cdot\rfloor$ means greatest integer.
Multinomial(1, ($p_1, \ldots, p_k$))	Partition [0, 1] into k segments so the ith segment has length p_i. Draw $U \sim \text{Unif}(0, 1)$; then let X equal the index of the segment into which U falls. Tally such draws for Multinomial(n, ($p_1, \ldots, p_k$)).
Dirichlet($\alpha_1, \ldots, \alpha_k$)	Draw independent $Y_i \sim \text{Gamma}(\alpha_i, 1)$ for $i = 1, \ldots, k$; then $\mathbf{X}^T = \left(Y_1/\sum_{i=1}^{k} Y_i, \ldots, Y_k/\sum_{i=1}^{k} Y_i\right)$.

Random number generator (RNG)

- Physical methods (HRNG/TRNG) (Hardware-/True-)
  - based on microscopic phenomena,
    e.g. thermal noise, photoelectric effect
  - Still need to correct for bias/sequence correlation
- Computational methods, (PRNG, Pseudo-)
  - linear congruential generator
    - $X_{n+1} = (aX_n + b) \bmod m$
  - Initialize
    - Set seed (reproducible randomness)
    - Using the computer's real time clock
  - Mersenne Twister (Mersenne prime $2^{19937}-1$)
    - Default many programs
    - Good enough for our use
  - Cryptographically secure approaches (CSPRNG)

Rejection sampling

- Difficult to simulate from $f(x)$ directly
- Easy to simulate from $g(x) \approx f(x)$.
- Assume $\exists \alpha \leq 1$ such that for all x: $f(x) \leq g(x)/\alpha \equiv e(x)$ (the envelope)
- Algorithm:
  1. Sample $Y \sim g(\cdot)$.
  2. Sample $U \sim \text{Unif}(0,1)$.
  3. If $U \leq f(Y)/e(Y)$, put $X = Y$, otherwise return to step 1
- Distribution of X:

$$\Pr(X \leq x) = \Pr(Y \leq x | U \leq \tfrac{f(Y)}{e(Y)}) = \frac{\Pr(Y \leq x, U \leq \frac{f(Y)}{e(Y)})}{\Pr(U \leq \frac{f(Y)}{e(Y)})}$$

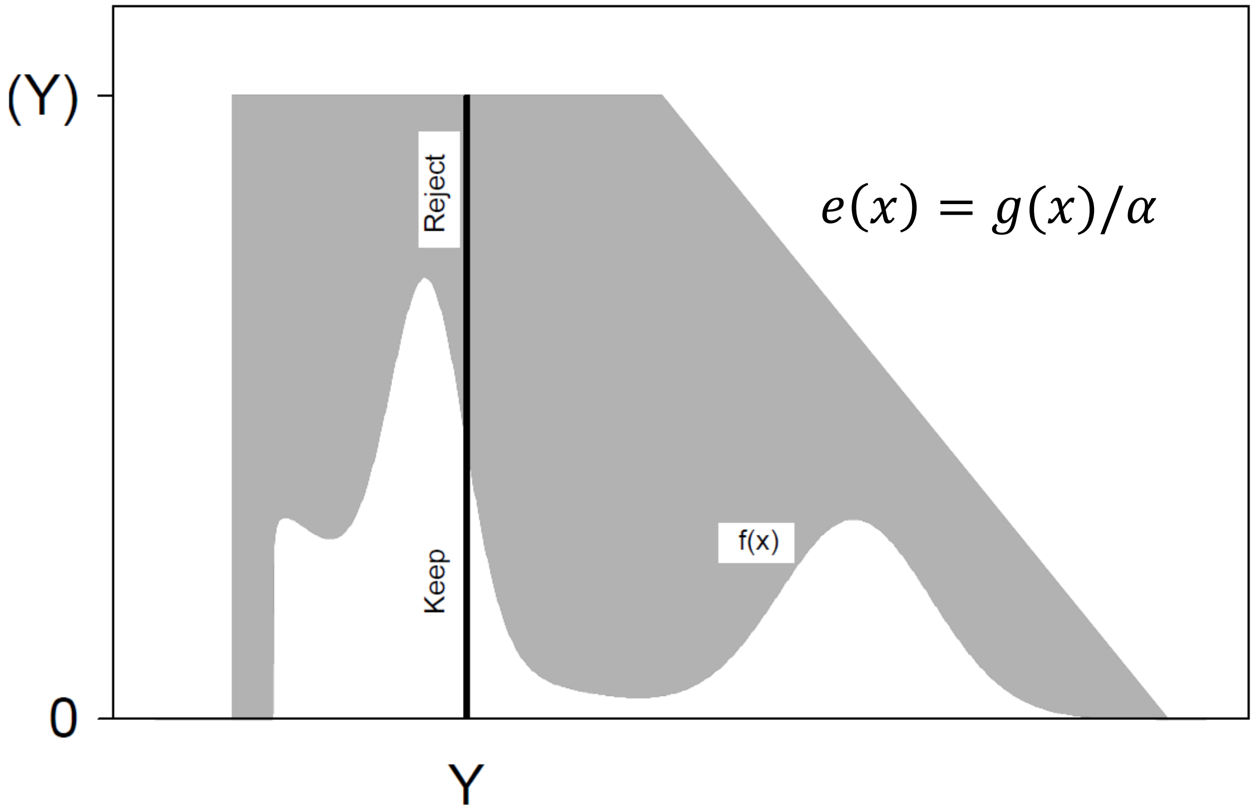
$$= \frac{\int_{-\infty}^{x} \int_{0}^{f(y)/e(y)} du g(y) dy}{\int_{-\infty}^{\infty} \int_{0}^{f(y)/e(y)} du g(y) dy} = \frac{\int_{-\infty}^{x} \frac{f(y)}{e(y)} g(y) dy}{\int_{-\infty}^{\infty} \frac{f(y)}{e(y)} g(y) dy}$$

$$= \int_{-\infty}^{x} f(y) dy = F(x)$$

- $\alpha = \Pr(U \leq \frac{f(Y)}{e(Y)})$ is the probability for acceptance
- α^{-1} is the expected number of iterations.

The envelope in rejection sampling

1. $U \sim \text{Unif}(0, 1)$ and accept if $U \le f(Y)/e(Y)$ is equivalent to
2. $U \sim \text{Unif}(0, e(Y))$ and accept if $U \le f(Y)$



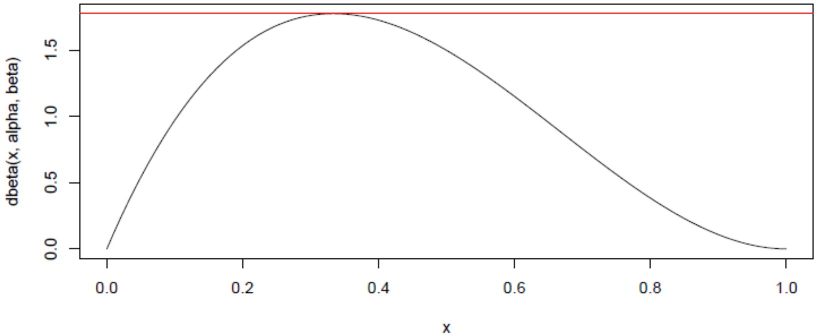
3. $U \sim Unif(0, 1)$ and accept if $U \le \alpha f(Y)/g(Y)$

Example rejection sampling

1. Aim: Simulate from Beta distribution:

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

2. $\arg\max_x f(x) = \frac{\alpha-1}{\alpha+\beta-2} = x^*$
3. Define $g(x) = 1; 0 < x < 1$. Then $g(x) \geq f(x)/f(x^*)$



4. Accept if $U \leq f(x)/f(x^*)$
5. `beta_rej.R`

What if the normalizing constant is unknown

- Assume $f(x) = c \cdot q(x),\ c -$ unknown
- If we can find, $\tilde{\alpha}$ such that:
  - $g(x) \geq \tilde{\alpha} q(x) \quad = \frac{\tilde{\alpha}}{c} f(x) = \alpha f(x)$

$$\boxed{\alpha = \frac{\tilde{\alpha}}{c}}$$

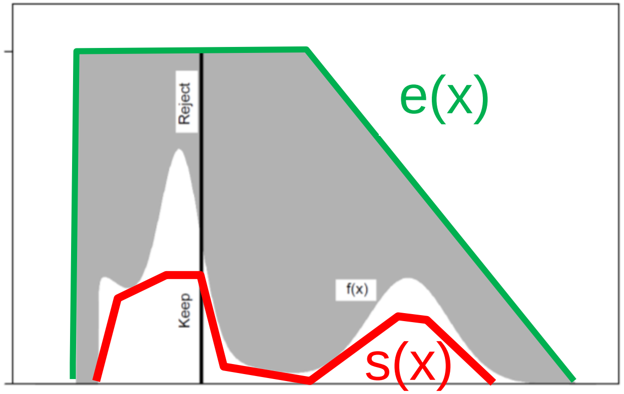
- Then:

$$U \leq \frac{\alpha f(Y)}{g(Y)} \Leftrightarrow U \leq \frac{\tilde{\alpha} q(Y)}{g(Y)}$$

- We need not know *c !*
- If we can estimate acceptance rate
- True acceptance rate is unknown: $\frac{\tilde{\alpha}}{c}$
- We can estimate c

Squeezed rejection sampling

- Assume
  - $\exists \alpha \leq 1$ and $g(\cdot)$ such that for all x: $f(x) \leq g(x)/\alpha \equiv e(x)$
  - $\exists s(x) \leq f(x)$ which is easy to evaluate
- Note: $U \leq s(Y)/e(Y)$ imply $U \leq f(Y)/e(Y)$

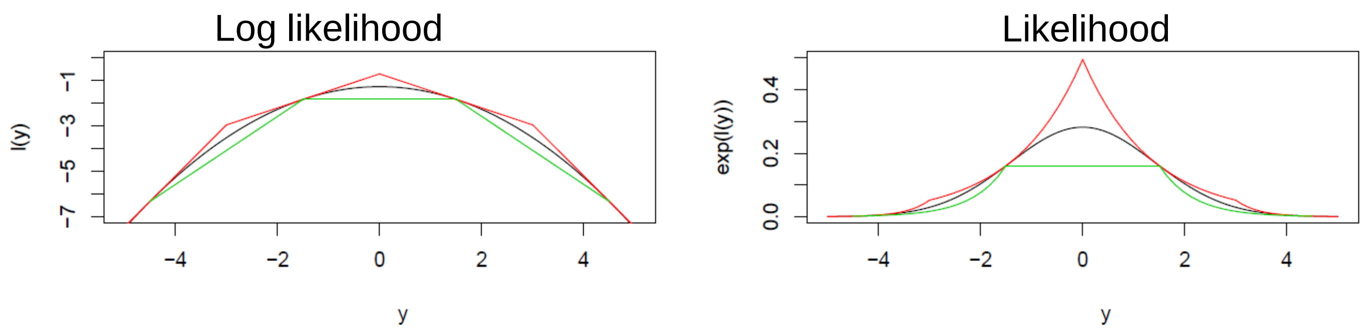


- Algorithm:
  1. Sample $Y \sim g(\cdot)$.
  2. Sample $U \sim \text{Unif}(0, 1)$.
  3. If $U \leq s(Y)/e(Y)$, accept $X = Y$
  4. If $U > s(Y)/e(Y)$, but $U \leq f(Y)/e(Y)$, accept $X = Y$
  5. If $U > f(Y)/e(Y)$, go to step 1

Adaptive rejection sampling

- Main challenge: Construct suitable envelope
- Assume now $l(x) = \log f(x)$ is concave and differentiable
- Choose initial points $x_1, x_2, ..., x_k$ such that $l'(x_1) > 0$, $l'(x_k) < 0$
- $e_k^*(x)$: Piecewise linear upper hull of $l(x)$

concave:
1) Never above the tangent in any point (super-gradient)
2) Never below the line connecting two points



- Proposal distribution:

$$g(x) = c\exp\{\ell(x_i) + \ell'(x_i)(x - x_i)\} \qquad \text{for } x \in [z_{i-1}, z_i]$$

$$z_i = \frac{\ell(x_{i+1}) - \ell(x_i) - x_{i+1}\ell'(x_{i+1}) + x_i\ell'(x_i)}{\ell'(x_i) - \ell'(x_{i+1})}$$

Possible to calculate c and also easily find $G(x)$ and $G^{-1}(x)$

- Also possible to define squeezing function

$$s_k^*(x) = \frac{(x_{i+1} - x)\ell(x_i) + (x - x_i)\ell(x_{i+1})}{x_{i+1} - x_i}$$

Adaptive rejection sampling

1. Start with $x_1, \ldots, x_k$ and calculate $e_k(x)$, $s_k(x)$, $g_k(x)$
2. Generate $x \sim g(x)$
3. If $U \leq s_k(Y)/e_k(Y)$, accept $X = Y$, goto step 6
4. If $U > s_k(Y)/e_k(Y)$, but $U \leq f(Y)/e_k(Y)$
    1. accept $X = Y$
    2. Add X to $\{x_1, \ldots, x_k\}$ and update to $e_{k+1}(x)$, $s_{k+1}(x)$, $g_{k+1}(x)$ and go to step 6
5. If $U > f(Y)/e(Y)$, reject and go to step 2
6. If not enough samples, return to step 2

Example: `ars.R`

```
library(ars)
```

Importance sampling

- Rewriting

$$\mu = \int h(\mathbf{x})f(\mathbf{x})d\mathbf{x} = \int \frac{h(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})}g(\mathbf{x})d\mathbf{x} = \frac{\int \frac{h(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})}g(\mathbf{x})d\mathbf{x}}{\int \frac{f(\mathbf{x})}{g(\mathbf{x})}g(\mathbf{x})d\mathbf{x}}$$

- Assume $X_1, \dots, X_n$ iid from $g(\mathbf{x})$. (We know how to sample from g(**x**))

- Two alternative estimates

$$\hat{\mu}^*_{IS} = \frac{1}{n}\sum_{i=1}^{n} h(\mathbf{X}_i)w^*(\mathbf{X}_i), \quad w^*(X_i) = \frac{f(\mathbf{X}_i)}{g(\mathbf{X}_i)}$$

$$\hat{\mu}_{IS} = \sum_{i=1}^{n} h(\mathbf{X}_i)w(\mathbf{X}_i), \quad w(\mathbf{X}_i) = \frac{w^*(\mathbf{X}_i)}{\sum_{j=1}^{n} w^*(\mathbf{X}_j)}$$

- $w^*(\mathbf{X}_i)$ called importance weights
- $w(\mathbf{X}_i)$ called the normalized importance weights

Importance sampling version 1

$$\hat{\mu}^*_{IS} = \frac{1}{n}\sum_{i=1}^{n} h(\mathbf{X}_i)w(\mathbf{X}_i), \quad w^*(\mathbf{X}_i) = \frac{f(\mathbf{X}_i)}{g(\mathbf{X}_i)}$$

$$E[w^*(\boldsymbol{X}_i)] = \int \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} g(\boldsymbol{x})d\boldsymbol{x} = \int f(\boldsymbol{x})d\boldsymbol{x} = 1$$

$$E[\hat{\mu}^*_{IS}] = \int h(\boldsymbol{x})\frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} g(\boldsymbol{x})d\boldsymbol{x} = \int h(\boldsymbol{x})f(\boldsymbol{x})d\boldsymbol{x} = \mu$$

$$\text{Var}[\hat{\mu}^*_{IS}] = \frac{1}{n}\text{Var}^g[h(\mathbf{X})w^*(\mathbf{X})]$$

- Can be unstable if $g(\mathbf{x})$ small when $f(\mathbf{x})$ large
- $g(\mathbf{x})$ should have heavier tails than $f(\mathbf{x})$.
- If only one $h(\mathbf{X})$ of interest, should choose

$$g(\mathbf{x}) \propto |h(\mathbf{x})|f(\mathbf{x})$$

- Often interested in many functions, focus on making variability of $w^*(\mathbf{X})$ small

Importance sampling version 2

$$\hat{\mu}_{IS} = \sum_{i=1}^{n} h(\mathbf{X}_i)w(\mathbf{X}_i), \quad w(\mathbf{X}_i) = \frac{w^*(\mathbf{X}_i)}{\sum_{j=1}^{n} w^*(\mathbf{X}_j)}$$

- Based on

$$\mu = \frac{\int \frac{h(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})} g(\mathbf{x})d\mathbf{x}}{\int \frac{f(\mathbf{x})}{g(\mathbf{x})} g(\mathbf{x})d\mathbf{x}} = \frac{\mu}{1} \approx \frac{\hat{\mu}^*_{IS}}{\hat{1}^*_{IS}} = \hat{\mu}_{IS}$$

$$\hat{\mu}^*_{IS} = \bar{t}, \quad t_i = t(\mathbf{X}_i) = h(\mathbf{X}_i)w^*(\mathbf{X}_i)$$

$$\hat{1}^*_{IS} = \bar{w}^*$$

- Why also estimate denominator?
  - What would be best if h(x) = c (constant)?
  - Correlations between nominator and denominator

Impact of normalization

$$\frac{\hat{\mu}_{IS}^*}{\hat{1}_{IS}^*} = \hat{\mu}_{IS} \qquad \hat{1}_{IS}^* = \bar{w}^*$$

- Taylor approximation of $1/\bar{w}^*$ around 1:

$$\frac{1}{\bar{w}^*} \approx 1 - (\bar{w}^* - 1) + (\bar{w}^* - 1)^2$$

giving

$$\hat{\mu}_{IS} \approx \bar{t}\,[1 - (\bar{w}^* - 1) + (\bar{w}^* - 1)^2]$$
$$= \bar{t} - (\bar{t} - \mu)(\bar{w}^* - 1) - \mu(\bar{w}^* - 1) + \bar{t}\,(\bar{w}^* - 1)^2$$

$$E[\hat{\mu}_{IS}] = E\{\bar{t} - (\bar{t} - \mu)(\bar{w}^* - 1) - \mu(\bar{w}^* - 1) + \bar{t}\,(\bar{w}^* - 1)^2\} + \mathcal{O}(n^{-2})$$
$$= \mu - \frac{1}{n}\mathrm{cov}[t(\boldsymbol{X}), w(\boldsymbol{X})] - 0 + \frac{\mu}{n}\mathrm{var}(w(\boldsymbol{X})) + \mathcal{O}(n^{-2})$$

$$\mathrm{var}[\hat{\mu}_{IS}] = E\left\{\left((\bar{t} - \mu) - \mu(\bar{w}^* - 1)\right)^2\right\} + \mathcal{O}(n^{-2})$$
$$= \frac{1}{n}\left[\mathrm{var}\big(t(\boldsymbol{X})\big) + \mu^2\mathrm{var}\big(w^*(\boldsymbol{X})\big) - 2\mu \cdot \mathrm{cov}[t(X), w^*(X)]\right] + \mathcal{O}(n^{-2})$$

$$\boxed{\mathrm{MSE}[\hat{\mu}_{IS}] - \mathrm{MSE}[\hat{\mu}_{IS}^*] = \frac{1}{n}\left(\mu^2\mathrm{var}[w^*(\mathbf{X})] - 2\mu\mathrm{cov}[t(\mathbf{X}), w^*(\mathbf{X})]\right) + \mathcal{O}(n^{-2})}$$

When is normalization better?

$$\text{MSE}[\hat{\mu}_{IS}] - \text{MSE}[\hat{\mu}^*_{IS}] = \frac{1}{n}\left(\mu^2 \text{var}[w^*(\mathbf{X})] - 2\mu \text{cov}[t(\mathbf{X}), w^*(\mathbf{X})]\right) + \mathcal{O}(n^{-2})$$

- Gain if

$$\text{cov}[t(\mathbf{X}), w^*(\mathbf{X})] > \frac{\mu \text{var}[w^*(\mathbf{X})]}{2}$$

$$\Updownarrow$$

$$\text{cor}[t(\mathbf{X}), w^*(\mathbf{X})] > \frac{\sqrt{\text{var}[w^*(\mathbf{X})]}}{2\sqrt{\text{var}[t(\mathbf{X})]}/\mu} = \frac{\text{cv}[w^*(\mathbf{X})]}{2\text{cv}[t(\mathbf{X})]}$$

- Example: `imp_samp_beta.R`

Coefficient of variation:
cv(X)=std(X)/E(X)

Effective sample size

- Assume $w_i = w(\mathbf{X}_i)$, $i = 1, \ldots, n$ are normalized weights
- Define effective sample size by

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{n} w_i^2}$$

Ex 1:	if $w_i = \frac{1}{n}$ for all i	$\widehat{N}_{eff} = n$
Ex 2:	if $w_i = 0, i \leq z, w_i = \frac{1}{n-z}, i > z$	$\widehat{N}_{eff} = n - z$
Ex 3:	if $w_i = 0, i \neq j, w_j = 1$	$\widehat{N}_{eff} = 1$