STAT 243 Final Project: Adaptive Rejection Sampling Package Development

Mengfei Jiang (SID: 26941889), Ji Wang, Zhuangdi Li, Alexander Fredh-Ojala

December 16, 2015

Contents

1 Description

The main function *ars* takes 4 inputs, namely (1) target log-concave density function, (2) sample size, (3) lower bound and (4) upper bound, and returns a sample by Adaptive Rejection Sampling. The lower bound and the upper bound define the support for the target density, and it is assumed that the user input the **analytically correct bounds** when defining the density. $-/+\infty$ are default values respectively. The function will then draw samples from the target density after validity checks.

The pakcage is located in Mengfei Jiang's Github repository: `https://github.com/MengfeiJiang` named as **stat243-project**.

2 Approach

The general approach is described by the below pseudo-code:

Let A be a set of points $x_i, h(x_i), h'(x_i)$. The tangents at these points define the upper hull h_u, the chords define the lower hull h_l.
Repeat
Sample x from $s(x)$, the normalized exponential of the upper hull
Sample u from a $Unif(0,1)$
    if $u < exp(h_l(x) - h_u(x))$ then accept x (squeezing)
    else perform the rejection test
        evaluate function value h at x
        if $u < exp(exp(h(x) - h_u(x)))$ *then* accept x
        else reject x
        endif
        update upper and lower hull by adding $x, h(x), h'(x)$ to A
    endif
until the user required number of points are sampled

Descriptions of each function are detailed as follows.

2.1 Input Validity Check

Check support boundaries Three checks are performed: lower bound is smaller than upper bound, density is neither $-/+\infty$ or smaller than 0 at bounds, and density is not 0 everywhere within the bounds.

Check density convergence The function stops when the integral of the unnormalized density diverges within the bounds.

Check log-concavity of the density Log-concavity check is performed locally at the neighbourhood of each of the abscissae, and we require $h_u(x_i)$ be greather than or equal to $h_l(x_i)$. The function stops whenever $h_l(x_i)$ exceeds $h_u(x_i)$. Hence, log-concavity check is performed as sampling proceeds.

2.2 Distribution Shape Determination

Mode finding Mode of the target density is to be used to determine the density shape and as one of the initialization points, and hence, R's *optim* function is called to find such mode.

Shape determination We categorize the shape of a distributi on into 4 categories, (1) uniform distribution with constant density within bounds, (2) monotonely decreasing density, (3) monotonely increasing density, and (4) density mode occurring within bounds. R's built-in *runif* function is called to generate samples in the first shape category, while the following steps of ARS are performed to generate samples for other categories.

2.3 Boundary Shrink

To avoid numerical issues, valid boundaries are shrinked to such an extent that the target density is not numerically zero at and within bounds if it is in the first place.

2.4 Initialization

Initial abscissae are the lower bound, the upper bound, the mode, one point to the left of the mode and another point to the right of the mode. h value, i.e. the $log(density)$ value, and h' value is calculated for each of the initial point.

2.5 Upper hull and lower hull

Vectorized auxiliary functions u and l are such that take a vector of x's and output the corresponding values at the upper hull and the lower hull. Each time $h(x)$ is evaluated, the two functions are also updated. Log-concavity check is performed for all abscissae every time the upper and lower hulls are updated.

2.6 Intersections of upper hulls

Intersections of the upper hulls are determined by the following formula:

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

, for $i = 1, 2, \ldots, n$, and $z_0 =$ lower bound.
z's are updated each time the upper hull and the lower hull are updated.

2.7 Sampling

CDF CDF up to each intersection point is calculated as follows:

$$cdf(z_0) = 0$$
$$cdf(z_i) = \frac{1}{constant} \sum_{j=0}^{i} \frac{1}{h'(x_{j+1})} exp(h_u(z_{j+1})) - exp(h_u(z_j))$$

, where

$$constant = \sum_{j=0}^{n-1} \frac{1}{h'(x_{j+1})} exp(h_u(z_{j+1})) - exp(h_u(z_j))$$

Inverse-CDF To sample from the customized distribution, we need to inverse the cdf. First, draw a vector of $Unif(0, 1)$ random variables $\boldsymbol{u}$. Then, for each u_j, we find the largest z_i such that cdf(z_i) is smaller than u_j. Then, the x_j sampled from u_j is generated as such to form candidate sample points $\boldsymbol{x}$:

$$x_j = z_i + \frac{1}{h'(x_{i+1})} log\Big[1 + \frac{h'(x_{i+1}) * constant * (u_j - cdf(z_i))}{exp(h_u(z_i))}\Big]$$

, for $i = 0, 1, 2, \ldots, n-1$.

Squeeze test and rejection test A vector of $Unif(0,1)$ random variables $\boldsymbol{w}$ is generated to perform squeeze test. All the candidate points up to the first rejected point are accepted and stored in the final result to be returned. Then, rejection test is performed on the rejected point, which will be added to the final result if accepted. The upper hulls and lower hulls are then updated, and log-concavity check is performed again.
If the number of accepted points in the result is smaller than the sample size, go back to the sampling step and sample until enough.

3 Testing

Only main function test is detailed in the report. Unit tests are performed while the package is being developed. The main function is tested on both log-concave and non-log-concave distributions.

3.1 Log-concave Distributions

3.1.1 Normal

3.1.2 Truncated Normal

3.1.3 Uniform

3.1.4 Exponential

3.1.5 Gamma

3.1.6 Beta

3.1.7 Logistic Distribution

3.1.8 Extreme Value Distribution

3.1.9 Laplace

3.1.10 Chi-squared with dof >= 2

3.1.11 Weibull Distribution

3.2 Non Log-concave Distributions

3.2.1 Chi-squared with dof = 1

3.2.2 Student's t Distribution

3.2.3 Cauchy distribution

3.2.4 Pareto Distribution

3.2.5 F-Distribution

4 Logistics

4.1 Code

4.2 Testing

4.3 Report