ST3456 Modern statistical methods II

Lab 7 - 20/11/18

Bernardo Nipoti

Lab 7: Metropolis-Hastings for continuous distributions

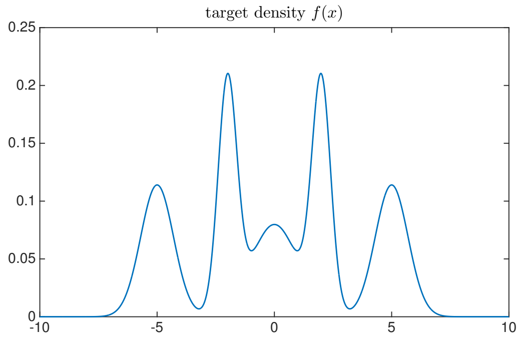
Download the files `ST3456_lab7.R` and `ST3456_lab7_solutions.R` from the course webpage

`http://www.bernardonipoti.com/teaching/st3456_1819`

We want to use the Metropolis-Hastings algorithm to simulate from the target distribution f, displayed below and defined as

$$f(x) = \frac{1}{5} f_N(x; -5, 0.7) + \frac{1}{5} f_N(x; -2, 0.4) + \frac{1}{5} f_N(x; 0, 1) + \frac{1}{5} f_N(x; 2, 0.4) + \frac{1}{5} f_N(x; 5, 0.7),$$

where $f_N(x; \mu, \sigma)$ denotes the density of a normal random variable with mean μ and standard deviation σ.



The function `target()` allows f to be evaluated.

Part 1. Independent Metropolis-Hastings

1a. Write a function `iMH()` to implement the independent M-H algorithm with normal proposal $g(x)$ with mean 0 and variance S^2. Recall that the probability of accepting a generated value is given by

$$\alpha(x, y) = \min\left\{\frac{f(y)g(x)}{f(x)g(y)}, 1\right\}$$

Specifically `iMH()` should have, as input, N (number of iterations), x_0 (initial state) and S (st. dev. of the proposal), and return the generated sample and the acceptance rate.

1b. Set $N = 10\,000$, $x_0 = 0$ and $S = 3$. Generate a sample by using `iMH()` and use it to estimate mean (=0) variance (=12.06) of f and to build a histogram approximating the true distribution f.

1c. Study the effect of parameter S on the histogram you obtain (and related estimates for mean and variance). Specifically, consider $S = 1$ (too small?) and $S = 20$ (too large?).

Part 2. Random walk Metropolis-Hastings

2a. Write a function `rwMH()` to implement a random walk M-H algorithm with normal distribution for the jumps of the random walk. That is, given that the chain is at state x, a new value y is obtained by simulating $Y = x + \varepsilon$, where $\varepsilon \sim N(0, S^2)$. Recall that the probability of accepting a generated value is given by

$$\alpha(x, y) = \min\left\{\frac{f(y)}{f(x)}, 1\right\}$$

Specifically `rwMH()` should have, as input, N (number of iterations), x_0 (initial state) and S (st. dev. of ε), and return the generated sample and the acceptance rate.

2b. Set $N = 10\,000$, $x_0 = 0$ and $S = 3$. Generate a sample by using `rwMH()` and use it to estimate mean (=0) and variance (=12.06) of the true distribution and to build a histogram approximating the true distribution f.

2c. Visualise the traceplot of generated values to qualitatively assess the mixing of the chain.

2d. Repeat parts 2b and 2c with $S = 0.1$ (too small?).

2e. Tuning of the parameter S. Study how the acceptance rate changes with S and identify a value S^* which gives an acceptance rate approximately equal to 50%.

Part 3. Comparison via simulation study

3a. Compare the performance of independent M-H (with $S = 3$) and randow walk M-H (with $S = S^*$, as found in part 2e). For each generated sample, the comparison is made by estimating the variance (=12.06) of f. Repeat 20 times and, for each method, compute the mean absolute deviation, given by

$$\text{m.a.d.}(\hat{V}) = \frac{1}{20}\sum_{i=1}^{20} |\hat{V}_i - 12.06|,$$

where $\hat{V}_i$ is the estimate obtained at the ith replicate. You should see that the two algorithms have pretty similar performances.