2nd problem set for Computer Simulations, 2024

Hand-in: see schedules for your group, linked in the TeachCenter.
Please upload/submit your program files in the TeachCenter *before* the conversation with the tutor.

Markov chain with Metropolis algorithm

In this simple example we will encounter the main features as well as the difficulties of Markov Chain Monte Carlo. Consider the normal distribution

$$\phi(x, x_0) = \frac{1}{\sqrt{2\pi}}\, e^{-\frac{(x-x_0)^2}{2}}$$

with variance $\sigma^2 = 1$ and average x_0. Generate a sequence of random numbers $x(t)$ which are distributed according to

$$\pi(x) = \frac{1}{2}\,[\,\phi(x, -\xi) \;+\; \phi(x, \xi)\,]$$

by employing the Metropolis–Hastings algorithm.

For the *proposal* probability $q_{\alpha\beta}$ of Metropolis-Hastings, use the following algorithm:
After step number t, the chain is at a value (a "state") $x_\alpha := x(t)$. Draw a proposal x_β from the neighborhood of x_α, e.g. from the distribution

$$x_\beta = x_\alpha + (r - 0.5)\Delta x,$$

where r is a uniform random number between 0 and 1, and Δx specifies the range in which x_β is proposed. If possible, adjust the parameter Δx such that *roughly* half of the proposed x_β are accepted.
(This is one of many reasonable choices for $q_{\alpha\beta}$. Note that ergodicity will be satisfied with this proposal probability: any real number x can be reached (if $\pi(x)$ permits) in a finite number of steps with finite probability.)

The *acceptance* probability for Metropolis-Hastings is specified in the lecture notes.

Analysis

Perform the steps a) to e) listed below for each of the cases (1) $\xi = 0$ (2) $\xi = 2$ and (3) $\xi = 6$.

a) Examine by way of a histogram whether the generated numbers $x(t)$ belong to the distribution $\pi(x)$. Calculate error bars for the histogram bars like in problem set 1 (i.e. disregarding autocorrelations), using the Bayesian equations. Specifically: do the heights of the two maxima agree within error bars ?

b) Calculate the average and the variance of a sample of N numbers $x(t)$. Are the results reasonable ? You can choose N yourself.

Check the time series for the presence of autocorrelations (steps c, d, and e):

c) By eye: plot the time series $x(t)$. (N sufficiently large). Can you recognize correlations? On which time scales ? You should look at the whole series and also zoom in!

d) Autocorrelation function:

   (i) You can use a library routine to calculate estimates for the autocorrelation function $\rho^E(t)$. (Note that most library routines (e.g. Matlab *xcov*, not the same as *xcorr* and *autocorrelation*) will use the average $\bar{x}$ like in the lecture notes in the equation *above* (1.38), instead of $\bar{x}_t$ and $\bar{y}_t$ in equation (1.38). For our simulations, and with large N, the differences will be small at the relevant small distances t and the library routines will therefore be sufficient.)

   (ii) *Plot* your estimate of the autocorrelation function. In case of a Markov chain, it usually (not always!) decays exponentially, often quickly. You should therefore use a semi-logarithmic plot. Verify that the function drops off exponentially and then decays into noise. You can easily recognize noise, because the autocorrelation function (for the algorithms considered by us) is monotonic and convex.

   Estimate (measure very roughly) the inverse slope of the autocorrelation coefficient, i.e. the time scale visible, and compare it with the time scales you have identified by eye in the time series.

   Note that it is *disadvantageous (!)* to plot the autocorrelation function far into the region of noise, since the relevant information at smaller t, before the noise, becomes more difficult or impossible to see. In our cases, sensible values for t_{max} can be quite small.

e) Perform a "binning analysis" (chapter 1.5.4 of the lecture notes) in the following way. For each value of $k = 1, 2, 4, 8, \ldots,$ up to about $N/2$:

   - block your N data points $x(t)$ into sequential blocks, leaving out any last unfilled block. There are then $N_{B,k} \approx N/k$ blocks.
   - calculate the averages $\bar{O}_{i,k}$ $(i = 1, \ldots, N_{B,k})$ of $x(t)$ *in each block*, and then the average $\bar{O}_{B,k}$ of the $\bar{O}_{i,k}$.
   - calculate the variance of these averages, $\sigma_k^2 = \frac{1}{N_{B,k}-1} \sum_{i=1}^{N_{B,k}} (\bar{O}_{i,k} - \bar{O}_{B,k})^2$

   After you have calculated σ_k^2 for all k, plot $\frac{1}{N_{B,k}} \sigma_k^2$ against $\log k$. The resulting curve should converge when k is much larger than the autocorrelation time, and the value plotted is then a good estimate of the variance of $\bar{O}_B$. At the largest k, the curve may fluctuate again. Why ? (N.B. When the autocorrelation times are small, the procedure may work better with a less drastic increase of k, e.g. values $k = 1, 2, 3, 4...$ or similar, and a plot versus k instead of $\log k$.)

   The ratio of the converged plotted value to the one at $k = 1$ should be similar to the time scales identified earlier (see eq. (1.71) in the lecture notes). Is it ? For each binning analysis of a Monte Carlo run, discuss whether you consider the curve to have converged. If it does not converge, what are the consequences for error estimates ??

f) At $\xi = 6$ there are autocorrelations on a large time scale. Discuss which influence Δx has on this time scale. What happens at small Δx, and at large Δx? Why are the autocorrelations large at $\xi = 6$?