Exercises for Computational Physics (physics760)
WS 2020 / 21

Thomas Luu, Andreas Nogga, Marcus Petschlies, Andreas Wirzba

Exercise 4 (20 pts. total)

Homework (due Dec. 02 at 18:00)

Please note the due date of this homework. Submission of homework *requires* submitting your solutions (e.g. answers to questions, graphs, results in tabular form) in the form of a brief report (please no 100 page submissions!) **AND** a copy of your code that you used to do the simulations. Please address all questions and requirements written below in *italics* and red font in your homework report.

5 Error analysis of a Markov Chain

In this week's exercise we apply proper analysis techniques to estimate the statistical error of a quantity estimated from a Markov Chain (MC). We consider simulating the Long Distance Ising Model of last week's exercise.

To begin, first generate two MCs using the HMC algorithm you developed using the parameters

$$\beta J = .1; \quad \beta h = .5; \quad n = 5; \quad N_{md} = 100$$

for one MC and

$$\beta J = .1; \quad \beta h = .5; \quad n = 5; \quad N_{md} = 4$$

for the other MC (they only differ in N_{md}). Generate a long MC for each run (e.g. $N = 12800$ trajectories). Don't forget to disregard any *thermalization* trajectories if needed. In other words, you should have an ensemble,

$$\{\phi\} = \{\phi_1, \phi_2, \ldots, \phi_N\} ,$$

for each set of parameters. Now calculate the magnetization per configuration from these two MCs. You should now have two lists of calculated magnetizations,

$$\{m\} = \{m_1 = m(\phi_1), m_2 = m(\phi_2), \ldots, m_N = m(\phi_N)\} .$$

1: 2 pts. *Plot the first couple hundred trajectories of the MC history of $\{m\}$. How do the trajectories differ for the two MCs?*

Let's take a step back. Simulating this Markov Chain means sampling HS field configurations ϕ_1, $\phi_2, \ldots, \phi_N$ for some number N of trajectories with the HMC. Each ϕ_k is a sample for its underlying random variable Φ_k, k labelling the "*Monte-Carlo time*". Assuming the

Markov Chain has reached the equilibrium distribution (having left out sufficiently many Monte-Carlo time steps from the start), we use the usual *ensemble mean*,

$$\bar{m}_N = \frac{1}{N}\sum_{k=1}^{N} m(\phi_k)\,,$$

as the *Monte-Carlo estimator* of the magnetization $\langle m \rangle$.

For the ensemble mean, you should find roughly $\bar{m}_N \sim .49$. The goal now is to provide an error estimate for this quantity.

5.1 Autocorrelation

Correlations in data can dramatically affect estimated errors. In particular, if the ordered list of elements is highly correlated, then obviously the elements do not constitute a list of iids. To understand correlations, one analyzes the *autocorrelations* of the data.

A useful quantity in which to analyze the correlations is the autocorrelation function,

$$\Gamma^{(m)}(|k-l|) = \langle (m(\Phi_k) - \langle m \rangle)\,(m(\Phi_l) - \langle m \rangle)\rangle\ .$$

It should be clear from our previous discussions that due to the Markov Chain simulation the subsequent $m(\Phi_k)$ are no longer independent random variables. And thus correlations should be present in your data.

So to get an estimate $\Delta\bar{m}_N$ for the true error $\delta\bar{m}_N$ we need to estimate $\mathrm{Var}(M_N)$ and thus $\Gamma^{(m)}$.

2: 5 pts. *Implement the straightforward estimator* $C(\tau) = \bar{\Gamma}^{(m)}(\tau)/\bar{\Gamma}^{(m)}(0)$ *for the* normalized *autocorrelation function using*

$$\bar{\Gamma}^{(m)}(\tau) = \frac{1}{\#(k,l)} \sum_{(k,l)\,:\,\tau=|k-l|} (m_k - \bar{m}_N)\,(m_l - \bar{m}_N)$$

and plot the function of $C(\tau)$ *for your generated data sets.*

5.2 Blocking (also called "binning")

For this section use the data generated with $N_{md} = 100$. One can reduce the "negative" effects of autocorrelation by blocking the data. That is, take averages of successive blocks of b elements of the list (preserving the same order of the original list), which makes a new list that is N/b elements long, where N was the length of the original list. For example, assume you want to block your list $\{m_k\}$ by $b = 2$. You would construct this blocked list with elements $\{m_1^{(b)}, \ldots, m_{N/2}^{(b)}\}$ where $m_i^{(b)} = (m_{2i}^{(b)} + m_{2i+1}^{(b)})/2$. As discussed in the previous lecture, blocking reduces autocorrelation and makes the elements more like iid. This means that error estimates of blocked data are more reliable.

3: 5 pts. *Generate blocked data for b =2,4,8,16, 32, and 64, and calculate the autocorrelation for each blocked list. Does it behave the way you expect? With the blocked lists, estimate the* naive standard error *with* $\sigma/\sqrt{N/b}$ *and observe the behavior, where* σ *is the standard deviation of the blocked list.*

5.3 The bootstrap

The bootstrap procedure provides a means for giving an unbiased estimate of the error, as discussed in the previous lecture. The procedure is as follows:

- Given a list $\{m\}$ that is, say, 1000 elements long, make a new list ("bootstrap list") that is also 1000 elements long by randomly pulling elements from $\{m\}$ *with replacement* 1000 times. Label this new list $\{m\}_{bs}^{0}$.

- Repeat this procedure N_{bs} times, thereby making a list of *bootstrap lists*:

$$\{\{m\}\}_{bs} = \{\{m\}_{bs}^{0}, \{m\}_{bs}^{1}, \ldots, \{m\}_{bs}^{N_{bs}-1}\}\ .$$

- Now calculate the mean of each bootstrap list, which gives a list of numbers that is N_{bs} elements long,

$$\{\bar{m}\}_{bs} = \{\bar{m}_{bs}^{0}, \bar{m}_{bs}^{1}, \ldots, \bar{m}_{bs}^{N_{bs}-1}\}$$

- The *bootstrap* error δm of the quantity $\langle m\rangle$ is the standard deviation of $\{\bar{m}\}_{bs}$.

Increase N_{bs} until you find a stable value for δm. This is now your non-biased error estimate for $\bar{m}$.

4: 8 pts. *Code up the bootstrap procedure and calculate the bootstrap error for your blocked list of $\{m\}$. Investigate the stability of the error as a function of N_{bs}. How does the bootstrap error compare with your naive estimate from above?*