Markov Chain Monte Carlo Simulations and Their Statistical Analysis – An Overview

Bernd Berg

FSU, August 26, 2008

Content

Probability Distributions and Sampling

In N experiments we may find an event A to occur n times. The **frequency definition** of the **probability** of the event is

$$P(A) \;=\; \lim_{N\to\infty} \frac{n}{N}\ .$$

Let $P(a,b)$ be the probability that $x^r \in [a,b]$ where x^r is a **random variable** drawn in the interval $(-\infty,+\infty)$ with a **probability density** $f(x) > 0$. Then,

$$P(a,b) = \int_a^b dx\, f(x) \quad \text{and} \quad f(x) = \lim_{y\to x} \frac{P(y,x)}{x-y}\ .$$

The **(cumulative) distribution function** of the random variable x^r is defined as

$$F(x) = P(x^r \le x) = \int_{-\infty}^{x} f(x')\, dx'\ .$$

For **uniform probability distribution** between $[0,1)$,

$$u(x) = \begin{cases} 1 & \text{for} \quad 0 \leq x < 1; \\ 0 & \text{elsewhere.} \end{cases}$$

The corresponding distribution function is

$$U(x) = \int_{-\infty}^{x} u(x')\, dx' = \begin{cases} 0 & \text{for} \quad x < 0; \\ x & \text{for} \quad 0 \leq x \leq 1; \\ 1 & \text{for} \quad x > 1. \end{cases}$$

It allows for the construction of general probability distributions. Let

$$y = F(x) = \int_{-\infty}^{x} f(x')\, dx' \ .$$

For y^r being a uniformly distributed random variable in $[0,1)$

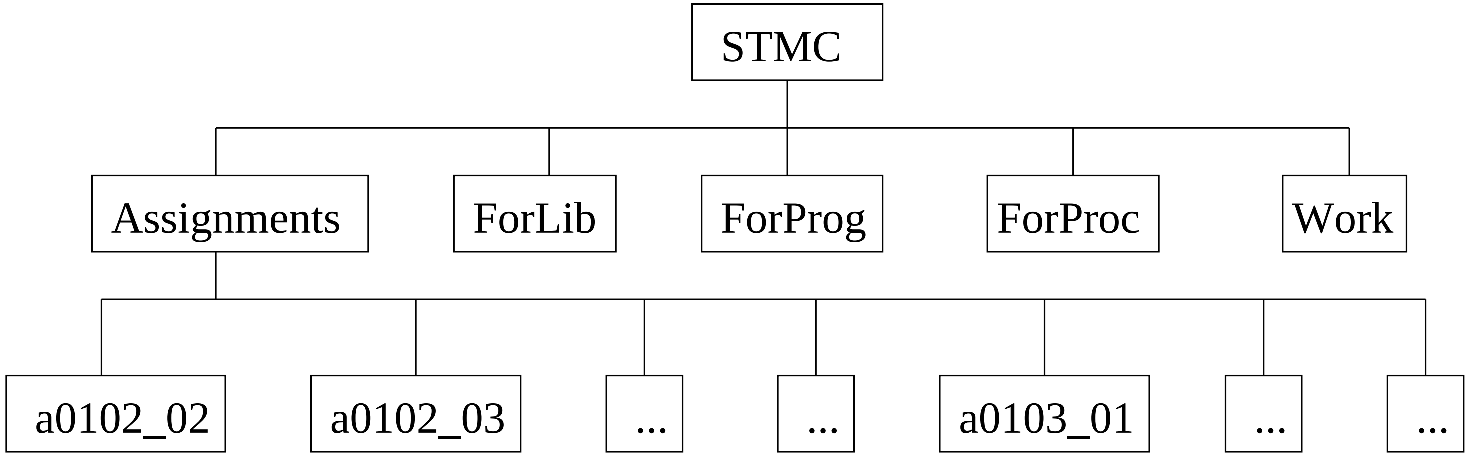
$x^r = F^{-1}(y^r)$ is then distributed according to the probability density $f(x)$.

Pseudo Random Numbers and Computer Code

It is sufficient to generate uniform (pseudo) random numbers. Control your random number generator! Therefore, a portable, well-tested generator should be choosen. My code supplies a generator by Marsaglia and collaborators whith an approximate period of 2^{110}. How to get it? Download `STMC.tgz` which unfolds under (Linux)

```
tar -zxvf STMC.tgz
```

into the directory structure shown below.



Topics to Follow:

Confidence Intervals and Sorting

Monte Carlo and Statistical Bootstrap

Parallel Computing with MPI

Central Limit Theorem: Convergence to Gaussian Sample Mean.

Binning (Blocking) and Jackknife Error Analysis

Various Difference Tests

Determination of Parameters (Fitting)

Statistical Physics and Markov Chain Monte Carlo Simulations

MC simulations of systems described by the Gibbs canonical ensemble aim at calculating estimators of physical observables at a temperature T. In the following we consider the calculation of the expectation value of an observable $\mathcal{O}$. All systems on a computer are discrete, because a finite word length has to be used. Hence,

$$\widehat{\mathcal{O}} = \widehat{\mathcal{O}}(\beta) = \langle \mathcal{O} \rangle \;=\; Z^{-1} \sum_{k=1}^{K} \mathcal{O}^{(k)} \, e^{-\beta \, E^{(k)}}$$

$$\text{where} \quad Z \;=\; Z(\beta) \;=\; \sum_{k=1}^{K} e^{-\beta \, E^{(k)}}$$

is the partition function. The index $k = 1, \ldots, K$ labels all configurations (or microstates) of the system, and $E^{(k)}$ is the (internal) energy of configuration k.

No direct way to generate the important configuration!

Markov Chain Monte Carlo

A Markov chain allows to generate configurations k with probability

$$P_B^{(k)} = c_B \, w_B^{(k)} = c_B \, e^{-\beta E^{(k)}} \,, \qquad c_B \;\; \text{constant} \,.$$

The state vector $(P_B^{(k)})$, for which the configurations are the vector indices, is called Boltzmann state. A Markov chain is a simple dynamic process, which generates configuration k_{n+1} stochastically from configuration k_n. Let the transition probability to create the configuration l in one step from k be given by $W^{(l)(k)} = W[k \to l]$. Then, the transition matrix

$$W = \left(W^{(l)(k)} \right)$$

defines the Markov process. Note, that this matrix is a very big and never stored in the computer. The matrix achieves our goal and generates configurations with the desired probabilities, when it satisfies certain properties.

Many Algorithms!

Metropolis, Heatbath, cluster, ...

Statistical Errors of Markov Chain MC Data

A typical MC simulation falls into two parts:

1. Equilibration without measurements.

2. Production with measurements.

Take Autocorrelations (error of error bars) into Account!

$\Rightarrow$ Comparison of Markov chain MC algorithms.

Generalized Ensembles

Replica exchange method (parallel computing, MPI), multicanonical, ...

Summary

- We intend to consider Statistics, MC and MCMC simulations, the Statistical Analysis of their data, parallel computing with MPI and, finally, advanced MCMC algorithms and simulations.
- Each method comes with its own Fortan code.

We will primarily train to get things up and running for interesting applications.