5. *Reweighting*

The configurations generated in a Monte Carlo simulation contain a huge amount of information, from which we usually distill a couple of numbers.

It would be a shame to waste all that information. ***Reweighting*** is a method which allows us to "expand" the results from the original simulation, performed at coupling β_0, say, to *any* other β sufficiently close to the simulation point without performing any additional simulations.

For concreteness, let us consider here a Monte Carlo simulation of 2D Ising model at the critical coupling $\beta_c = \frac{1}{2}\log(1+\sqrt{2}) \approx 0.44$. MC simulation gives us a series of configurations $\Phi_1, \Phi_2 \ldots \Phi_N$, and measurements of some observable $O_i = O(\Phi_i)$.
The standard estimate of the expectation value:

$$\langle O \rangle_{\beta_c} \equiv \frac{1}{Z} \sum_{\{s\}} O(s)\, e^{-\beta_c E(s)} \approx \frac{1}{N} \sum_i O_i$$

Now, the basic principle of reweighting is very simple: consider measurements O_i, E_i from the same configuration Φ_i. Now

$$\frac{1}{N}\sum_i O_i e^{-(\beta-\beta_c)E_i} = \frac{\sum_{\{s\}}\left[Oe^{-(\beta-\beta_c)E}\right]e^{-\beta_c E}}{Z_{\beta_c}} = \frac{Z_\beta}{Z_{\beta_c}}\langle O\rangle_\beta$$

$$\frac{1}{N}\sum_i e^{-(\beta-\beta_c)E_i} = \frac{\sum_{\{s\}}\left[e^{-(\beta-\beta_c)E}\right]e^{-\beta_c E}}{Z_{\beta_c}} = \frac{Z_\beta}{Z_{\beta_c}}$$

$$\longmapsto \qquad \boxed{\langle O\rangle_\beta = \frac{\sum_i O_i\, e^{-(\beta-\beta_c)E_i}}{\sum_i e^{-(\beta-\beta_c)E_i}} = \frac{\left\langle O\, e^{-(\beta-\beta_c)E}\right\rangle_{\beta_c}}{\left\langle e^{-(\beta-\beta_c)E}\right\rangle_{\beta_c}}}$$

Note: $\sum_i$ goes over measurements, E_i, O_i must be measured from the same configuration.

In practice: during a MC run, write down all measurements E_i and various O_i in a file. After the run, use eq. $\square$ to calculate $\langle O\rangle_\beta$.

Alternative view: reweighting using histograms

Using histograms (probability distributions) for reweighting is the "original" reweighting method:

- During a MC run (at β_c), measure histogram $h_{\beta_c}(E)$.
- We know that $h_\beta(E) \propto n(E)\exp(-\beta E)$, where $n(E)$ is the number of states at energy E.
- Thus, we can reweight h: $h_\beta(E) \propto h_{\beta_c}(E)\exp[-(\beta-\beta_c)E]$.
- Finally, if $O(E)$ is a function of energy, we can calculate

$$\langle O\rangle_\beta = \frac{\sum_E O(E)h_\beta(E)}{\sum_E h_\beta(E)} = \frac{\sum_E O(E)h_{\beta_c}(E)\exp[-(\beta-\beta_c)E]}{\sum_E h_{\beta_c}(E)\exp[-(\beta-\beta_c)E]}$$

- **Main weakness:** this works only if $O[\{s\}] = O[E(\{s\})]$ is a function of energy, whereas $\square$ works for arbitrary observable.

Note: histogram method can be obtained directly from $\Box$ using $O = \delta_{E,E'}$.)

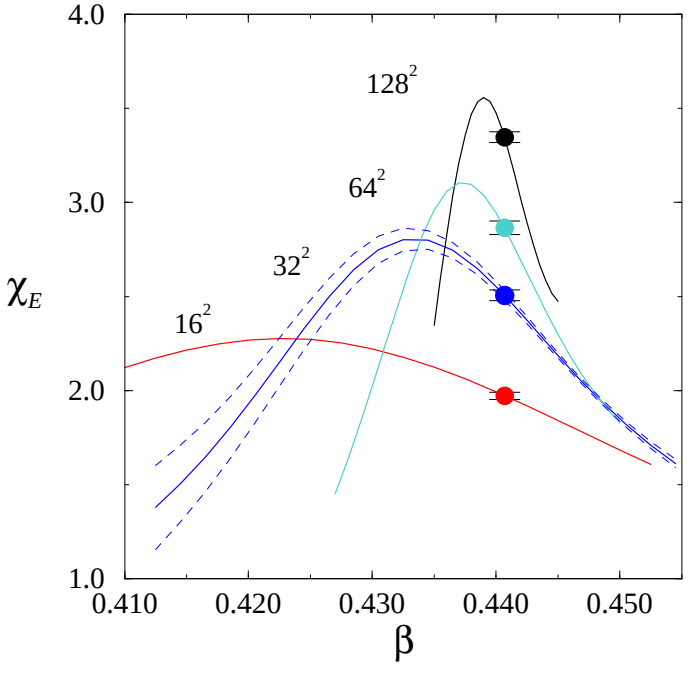
History:

- First proposed in '59 [Salzburg et al, J.Chem.Phys 30 (1959) 60]
- First used by McDonald and Singer 1967, no success (reweighting range too small?)
- Shown to be very effective by **Ferrenberg and Swendsen** [PRL 61 (1988) 2635]; now the whole thing goes under the name F-S reweighting.
- Multihistogram method: F+S [PRL 63 (1989) 1195; Computers in Physics, Sep/Oct 1988]

Example: 2d Ising, $V = 16^2 \ldots 128^2$. Energy susceptibility (Heat capacity)

$$\chi_E = -\frac{1}{V}\frac{\partial \langle E \rangle}{\partial \beta} = \frac{1}{V}\langle (E - \langle E \rangle)^2 \rangle$$

Susceptibility diverges with a critical exponent $\chi_E \sim L^{\alpha/\nu}$ when $L \to \infty$ and we are at $\beta = \beta_c$:



Points: simulated values, curves: reweighted data. Dashed lines show the error band for 32^2 (for clarity, not shown for others).

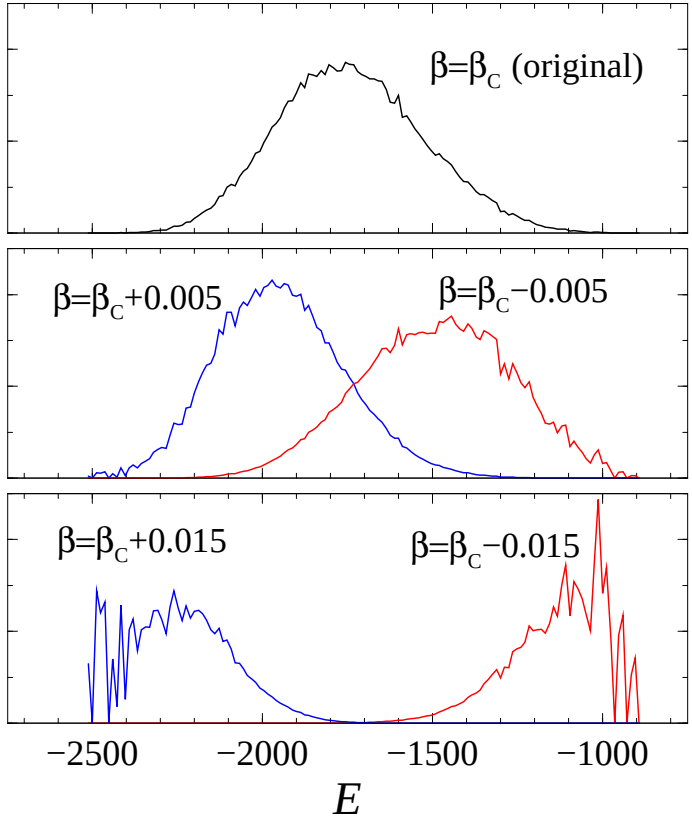
Histograms:

Energy histograms can be obtained through

$$h_\beta(E) = \frac{\sum \delta(E, E_i) e^{-\delta\beta E_i}}{\sum e^{-\delta\beta E_i}},$$

Histograms at 64^2 (probability distributions of E). Original simulation at $\beta = \beta_c \approx 0.44$.

The histograms become "exponentially" worse when the reweighting distance increases.



5.1. Error analysis in reweighting

- Errors analyzed in previous plot using jackknife error analysis Use of ***jackknife*** or ***bootstrap*** methods strongly recommended! (Return to that soon)
- Errors increase when reweighting distance increases. Must not do reweighting too far from original simulation point! (how far?)
- Note: using normal error analysis in upstairs and downstairs of

$$\langle O \rangle_\beta = \frac{\langle O\, e^{-(\beta-\beta_c)E} \rangle_{\beta_c}}{\langle e^{-(\beta-\beta_c)E} \rangle_{\beta_c}}$$

  is usually not reliable: the exponential factors make the 'observable' inside $\langle \cdot \rangle$ very skewed. Furthermore, this *overestimates* errors, since the deviations up and down are correlated!

- Calculating the errors using ***jackknife*** (for details of jackknife, see sect. 6.)

  1. Divide the data (N measurements) into M blocks, length $m = N/M \gg \tau$ (this to make individual blocks statistically independent).
  2. For each m, *delete* block m from the full data, and calculate

     $$\langle O \rangle_\beta^m = \frac{\sum_i O_i \, e^{-(\beta-\beta_c)E_i}}{\sum_i e^{-(\beta-\beta_c)E_i}}$$

     where the sums go over the $N-m$ measurements which do not belong to block m.
  3. Calculate the error through

     $$\delta\langle O \rangle_\beta = \sqrt{\frac{M-1}{M} \sum_m (\langle O \rangle_\beta^m - \langle O \rangle_\beta)^2}$$

     where $\langle O \rangle_\beta$ is either the full dataset reweighted value, or, usually the average of $\langle O \rangle_\beta^m$. The difference is very small, but the average of $\langle O \rangle_\beta^m$ is used for *bias correction*.

5.2. How far one can reweight?

When we reweight, the expectation value of the histogram shifts sideways. The *bulk* and *expectation value* of the shifted histogram must be well within the original histogram! I.e. if we perform the simulation at β_0, we should only go to β where

$$|\langle E\rangle_\beta - \langle E\rangle_{\beta_0}| = |\delta\langle E\rangle| = \lesssim \langle (E - \langle E\rangle_{\beta_0})^2\rangle^{1/2}$$

This actually should be valid for *any* observable, not only E!
Valid reweighting range $\lesssim$ 'fluctuation' range:

- Simulation at non-critical point: $\Delta\beta \propto 1/\sqrt{V}$.
- At a critical point (as in our example): $\Delta\beta \propto 1/L^x$, where x is some critical exponent. In our case above, $x = (d + \alpha/\nu)/2$ (I think), where $d = 2$ is the dimensionality.

PRAGMATIC VIEW: check that the 'mass' of the reweighted histogram does not shift too far away into tails of the original histogram, where there is insufficient amount of data! Naturally, more statistics $\rightarrow$ slightly larger

range.

Nevertheless, scaling laws are bad:

- Reweighting range $\propto 1/\sqrt{V}$.
- If we insist on increasing range $\delta\beta \to n\delta\beta$, and if we assume that $\delta\langle E\rangle_{n\delta\beta} = n\delta\langle E\rangle_{\delta\beta}$, then we have to increase statistics by a factor $\sim \exp[n^2 - 1]$ to achieve comparable accuracy (tail of a Gaussian)!

  The reweighting range cannot practically be increased by statistics. Much better to perform new simulations with different β.

Reweighting with respect to ***external magnetic field*** h**:** Ising model

$$Z_{\beta,h} = \sum_s \exp[-\beta E + hM] \quad \textbf{where} \quad M = \sum_i s_i$$

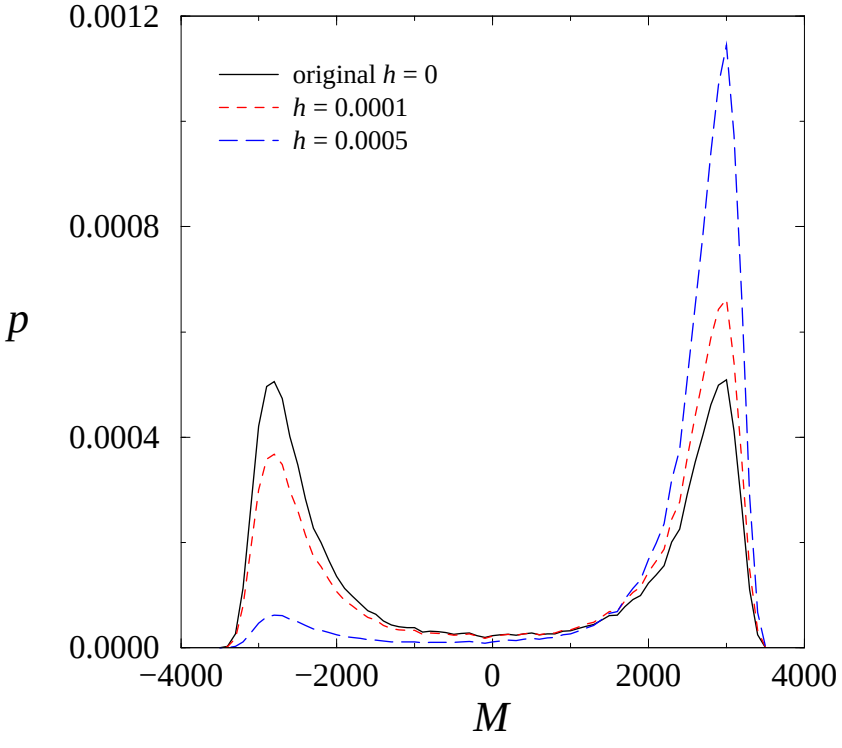
If the original simulation was performed with h_s, reweighting in h is completely analogous to eq. □:

$$\langle O \rangle_h = \frac{\sum_i O_i \, e^{(h-h_s)M_i}}{\sum_i e^{(h-h_s)M_i}}$$

Thus, even if originally $h_s = 0$ (no field in the original simulation), we can reweight to finite external field.

Reweighting can be done simultaneously in β and h. (How? And why the original histogram reweighting is not practical in this case?)

Magnetization histogram $p(M)$, size 64^2, in simulation $h = 0$, and $\beta = \beta_c$



5.3. Reweighting with respect to arbitrary parameters

Reweighting wrt. arbitrary parameters g^a, which couple to action/Hamiltonian $S(\vec{g};\{\phi\})$. The partition function is

$$Z_{\vec{g}} = \int [d\phi] \exp[-S(\vec{g};\{\phi\})]$$

Simulation is performed with $g^a = g_0^a$, and we measure $S_i(\vec{g}_0) = S(\vec{g}_0;\{\phi\}_i)$ and $S_i(\vec{g}) = S(\vec{g};\{\phi\}_i)$ (note that configs are generated with $S(\vec{g}_0;\{\phi\})$). Reweighting observable O:

$$\langle O \rangle_{\vec{g}} = \frac{\sum_i O_i \, \exp[-(S_i(\vec{g}) - S_i(\vec{g}_0))]}{\sum_i \exp[-(S_i(\vec{g}) - S_i(\vec{g}_0))]}$$

Usually the action factorizes to form $S = \sum_a g^a S^a$. In this case it is sufficient to measure the "pieces" of action S^a, and we can reweight wrt. any component g^a.

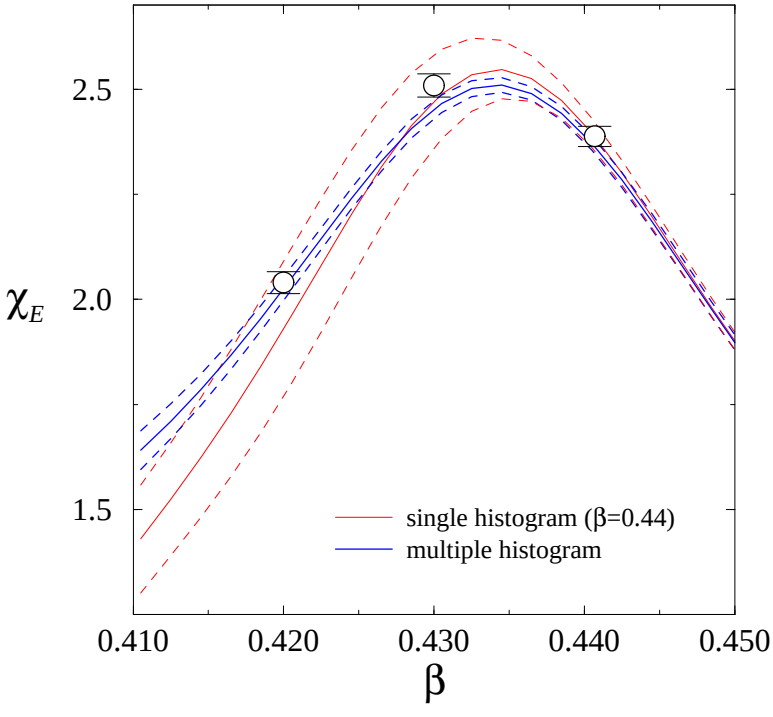
$g^1 = -\beta$, $S^1 = E$ recovers standard Ising reweighting.

Finding Lee-Yang zeroes of the partition function: reweight to **complex** β, and find those where $Z_\beta = 0$.

5.1. Multiple histogram reweighting

Multiple histogram method joins together several sets of data, run at different β-values, in an optimized way.

Example:
Susceptibility χ_E, volume 32^2. The data from 3 runs is joined together with multiple histogram reweighting (blue curve). Errors are much smaller than with single histogram reweighting.



Powerful method – should be part of standard toolbox for everybody who can join points from different simulations. However, somewhat cumbersome to use.

- Perform R Monte Carlo simulations, at couplings β_i, with length N_i.
- Measure the energy distributions $p_i(E) = H_i(E)/N_i$, and the autocorrelation time τ_i.
- True distribution is given by

$$p_i(E) = n(E)\, e^{-\beta_i E + f_i},$$

where $n(E)$ is the density of states (does not depend on β), and f_i is (dimensionless) free energy: $f_i = -\log Z_{\beta_i}$.

$\mapsto$ Each of the simulations gives us an estimate of $n(E) = p_i(E) e^{\beta_i E - f_i}$ (we don't know f_i, so that the normalization is unknown). Since the

MC runs were performed at different β_i, each of the run yields a reliable estimate of $n(E)$ only at limited range of E.

- **Optimization:** we obtain an improved estimate for $n(E)$ by combining all runs together:

$$n(E) = \sum_{i=1}^{R} r_i(E) p_i(E) e^{\beta_i E - f_i}, \quad \text{where} \quad \sum_{i=1}^{R} r_i(E) = 1 \text{ for all } E.$$

Note that the relative weights $r_i(E)$ are independent at each E: $\rightarrow$ optimization.

- $r_i(E)$ are determined by minimizing the (error)^2 in $n(E)$ (now follows somewhat technical derivation):
  - What is the uncertainty in histogram values? Assuming that $H_i(E)$ is Poisson distributed around the 'true' value $\bar{H}_i(E)$, we obtain

$$\delta^2 H_i(E) = g_i \bar{H}_i(E) = g_i N_i n(E) e^{-\beta_i E - f}.$$

Here $g_i = 1 + 2\tau_i$ takes into account the autocorrelations in run i.

– Thus, (error)^2 in $n(E)$

$$\delta^2 n(E) = \sum_i r_i^2(E) \frac{\delta^2 H_i(E)}{N_i^2} e^{2(\beta_i E - f_i)} = \sum_i r_i^2(E) \frac{g_i n(E)}{N_i} e^{\beta_i E - f_i}$$

– Minimize $\delta^2 n(E)$ wrt. $r_i(E)$ with condition $C \equiv \sum_i r_i(E) = 1$. Use Lagrange multipliers (try it):

$$\frac{\partial}{\partial r_i(E)} [\delta^2 n(E) + \lambda C] = 0 \quad \mapsto \quad r_i(E) = \frac{N_i g_i^{-1} e^{-\beta_i E + f_i}}{\sum_{j=1}^{R} N_j g_j^{-1} e^{-\beta_j E + f_j}}$$

- Thus, the optimized expression for $n(E)$ is

$$n(E) = \frac{\Sigma_{i=1}^{R} g_i^{-1} H_i(E)}{\Sigma_{j=1}^{R} N_j g_j^{-1} e^{-\beta_j E + f_j}}$$

- The coefficients f_i are then determined by solving

$$e^{-f_i} = \sum_E n(E) e^{-\beta_i E}$$

To solve this equation use some iterative method (Newton-Raphson). f_i's are determined up to an additional constant.

- Observable expectation values:

$$\langle O \rangle_\beta = \frac{\Sigma_E O(E) n(E) e^{-\beta E}}{\Sigma_E n(E) e^{-\beta E}}$$

As with the single histogram method, this can be formulated without resorting to $H_i(E)$. Let E_i^a be the energy measurement number a ($a = 1 \ldots N_i$) from run number i. The expression for the free energy

f_β becomes

$$e^{-f_\beta} = \sum_{i=1}^{R} \sum_{a=1}^{N_i} \frac{g_i^{-1} e^{-\beta E_i^a}}{\sum_{j=1}^{R} N_j g_j^{-1} e^{-\beta_j E_i^a + f_j}} \,.$$

f_i's are then solved from $e^{-f_i} = e^{-f_{\beta_i}}$, in analogy to the second eq. in the box above. The expectation value of O at reweighted β:

$$\langle O \rangle_\beta = \sum_{i=1}^{R} \sum_{a=1}^{N_i} \frac{O_i^a g_i^{-1} e^{-\beta E_i^a - f_\beta}}{\sum_{j=1}^{R} N_j g_j^{-1} e^{-\beta_j E_i^a + f_j}}$$

6. Error analysis: jackknife & bootstrap

As discussed before, it is no problem to calculate the expectation values and *statistical error estimates* of "normal" observables from Monte Carlo. However, often we have to calculate functions which depend (possibly non-linearly) on the *expectation values* of some quantities.

As a common example, we may have 2 observables, a and b, and we want the *correlation coefficient*

$$\rho = \frac{\sum_i (a_i - \langle a \rangle)(b_i - \langle b \rangle)}{\sqrt{\sum_i (a_i - \langle a \rangle)^2 \sum_j (b_j - \langle b \rangle)^2}} = \frac{\langle (a - \langle a \rangle)(b - \langle b \rangle) \rangle}{\sqrt{\langle (a - \langle a \rangle)^2 \rangle \langle (b - \langle b \rangle)^2 \rangle}}$$

where, as usual, $\langle a \rangle = 1/N \sum_i a_i$.

Now, if we write the measurements a_i, b_i in a file, it is of course no problem to calculate ρ. However, *what is the error of our result for ρ?* We cannot construct a sigle-configuration quantity ρ_i, which we could plug in the autocorrelation analysis.

One methdod to do the analysis would be to divide the measurements

$(a, b)_i$ in M blocks (bins), with block length $\gg \tau$, the autocorrelation time. Then we can calculate ρ_m for each block, and use the naive error formula for the error. However, this is not optimal: the value of ρ_m may vary a lot from block to block.

Jackknife and bootstrap methods are nowadays standard ways to calculate the error in this case.

– R.G. Miller, *the jackknife – a review*, Biometrika 61 (1974) pg. 1–17.

– B. Efron, *Computers and the theory of statistics: thinking the unthinkable*, SIAM Review, vol 21, No. 4 460 (1979)

– B. Efron, *The Jackknife, the Bootstrap and Other Resampling Plans*, Society for Industrial and Applied Mathematics (1982)

6.1. Jackknife

Jackknife method is a systematic way of obtaining the "standard deviation" error of a set of stochastic measurements:

1. Calculate average ρ (or some function f) from the full dataset
2. Divide data $(a, b)_i$ into M blocks, with block length $\gg \tau$. This is done in order to get rid of autocorrelations; if there are no correlations, block length can be 1.
3. For each $m = 1 \ldots M$, take away block m and calculate the average $\rho_{(m)}$ using the data from all other blocks.
4. Estimate the error of ρ by calculating the deviation of $\rho_{(m)}$'s from ρ:

$$\delta\rho = \sqrt{\frac{M-1}{M} \sum_{m=1}^{M} (\rho_{(m)} - \rho)^2}$$

The factor $(M-1)/M$ is there to give the correct result if we look at the errors of simple observables. For example, assume that we have a random observable a_i. Dividing these into M blocks, we can calculate an average from each block as a_m. Now we obtain the jackknife-blocked average as $a_{(m)} = 1/(M-1)\sum_{m' \neq m} a_{m'}$. Thus,

$$\sqrt{\frac{M-1}{M}\sum_m (a_{(m)} - \langle a \rangle)^2} = \sqrt{\frac{\sum_m (a_m - \langle a \rangle)^2}{(M-1)M}}$$

i.e. we obtain the standard (blocked) error estimate.

Why does jackknife work? $\rho_{(m)}$ contains almost the full set of data, thus, they are quite close to the final (right) value. Indeed, *each jackknifed block* $\sim$ *a new MC average of length* $N-m$ (However, these are naturally not independent!). This is why jackknife and bootstrap are often called ***resampling*** methods: they construct pseudo-independent 'new' simulation results.

Note: parametrically,

$$\rho_{(m)} - \rho \sim \delta\rho/\sqrt{M} \ ,$$

i.e. the distribution of the results of these new "simulations" is *narrower* than $\delta\rho$, the error. Thus, we cannot really consider jackknife sets to be "new" simultations (these would have distribution with width $\delta\rho$).

We want an estimate of $f(\langle a \rangle)$, where a is some quantity which we measure from simulation. The jackknife error estimate is naturally

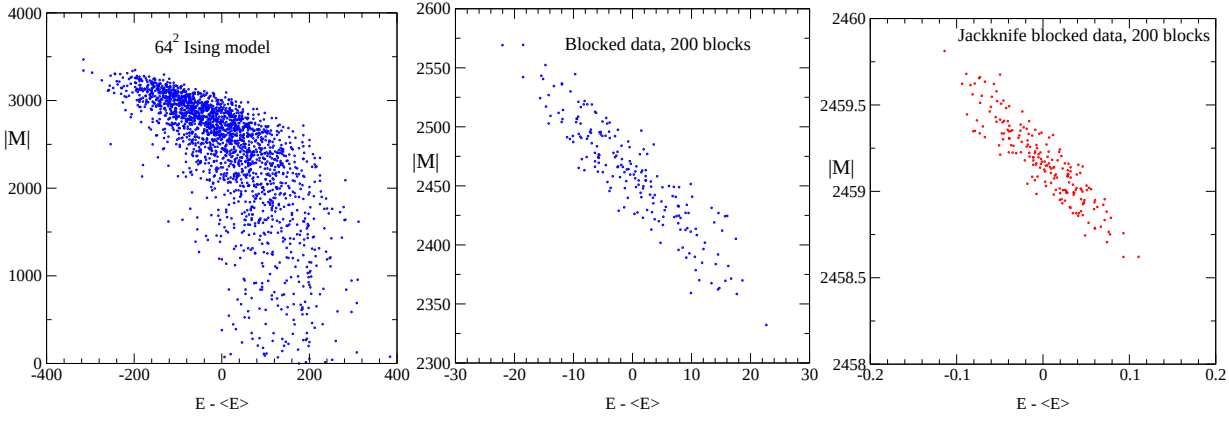
$$\delta f = \sqrt{\frac{M-1}{M} \sum_m (f_{(m)} - \langle f \rangle)^2} \approx |f'(\langle a \rangle)| \delta a$$

where $f_{(m)} = f(a_{(m)})$. This is typically a good approximation, because $f_{(m)}$ is close to $\langle f \rangle$. (Here $\langle f \rangle = \sum_m f_{(m)}/M$ *usually*; it could also be $\langle f \rangle = f(\langle a \rangle)$).

Jackknife blocked data makes it easy to 'chain' the blocks through consecutive functions: for example, if we want $g(f(\langle a \rangle))$, where a is some statistical measurable, we can form $a_{(m)}$, average of the observable a

over jackknifed block m. Now we can calculate $f_{(m)} = f(a_{(m)})$ and $g_{(m)} = g(f_{(m)})$, from which the error is

$$\delta g(f(\langle a \rangle)) = \sqrt{\frac{M-1}{M} \sum_m (g_{(m)} - \langle g \rangle)^2}$$



Above: 64^2 Ising at β_c;
a) sample of $(E - \langle E \rangle, |M|)$-data,
b) block averages with 200 blocks of length 2000,
c) jackknife block averages with same blocks.

Each jackknife block is very close to the final value, and the distribution of the jackknifed data is an approximation of the error ellipse (or domain, in general) of the $(\langle E \rangle, \langle |M| \rangle)$-data.

Feeding these blocks through some (non-linear) function, the distribution of the results gives us directly the error of the function value.

Let us calculate the correlation coefficient of $a = (E - \langle E \rangle)$ and $b = (|M| - \langle |M| \rangle)$. Doing this with 200 jackknife blocks as above, we have to calculate $\langle a \rangle$, $\langle b \rangle$ and $\langle ab \rangle$ for each of the 200 jackknife block, each containing $199 \times 2000 = 398000$ points. Lot of operations!

In this case, I obtain $\rho = -0.7139(15)$.

One could also (incorrectly) assume that each of the $\langle \cdot \rangle$'s in ρ are statistically independent. Then I would obtain, using standard independent error propagation,[1] $\rho = -0.7139(67)$. The error is 4 times too large!

---

[1] Independent errors: $\delta^2 f(a,b) = [\partial f/\partial a]^2 \delta^2 a + [\partial f/\partial b]^2 \delta^2 b$

6.2. Bootstrap

Bootstrap method is closely related to jackknife, but it mimics the resampling more closely. It works as follows:

1. Divide data $(a, b)_i$ into M blocks, with block length $\gg \tau$ (independent blocks).
2. From the set of the M blocks, pick randomly M blocks, not trying to avoid double sampling. Thus, some blocks may not get selected at all, some once, some twice etc.
3. Calculate the quantity of interest over the selected data – for example, the correlation coefficient ρ^*.
4. Repeat steps 2 and 3 a large number of times, say N_B times, each time using an independent set of random numbers to generate the bootstrap sample. The new correlation coefficients are $\rho^*_1, \rho^*_2, \ldots, \rho^*_{N_B}$.

5. Find values a and b so that these bracket the central 68% of the ρ^* values:

$$\frac{\#\{\rho_i^* < a\}}{N_B} = 0.16 \qquad \frac{\#\{\rho_i^* > b\}}{N_B} = 0.16$$

In effect, by generating a large number (N_B) of bootstrap samples one is generating the distribution function of the final result. The values a and b define the "1-σ" cumulants of this distribution.

6. The bootstrap estimate of the standard deviation can be now given as

$$\delta\rho = \frac{b-a}{2}\,.$$

Or, more accurately, we can give asymmetric errors

$$\rho = \rho_0{}^{+(b-\rho_0)}_{-(\rho_0-a)}$$

where $\rho_0 = \langle\rho\rangle$.

7. The points 5-6 above can be usually substituted with the bootstrap estimate of the standard deviation:

$$\delta\rho = \sqrt{\frac{1}{N_B - 1}\sum_i (\rho_i^* - \langle\rho^*\rangle)^2}$$

where $\langle\rho^*\rangle = \sum_i \rho_i^*/N_B$.

In jackknife/bootstrap literature there is no initial blocking – successive datapoints are considered to be statistically independent (no autocorrelations)! However, this is a minor modification: by blocking initially we obtain statistically independent measurements.

What is a sufficient number of blocks? One should have at least several tens of samples, preferably hundreds. N_B in the bootstrap can go to $\sim$ 1000.

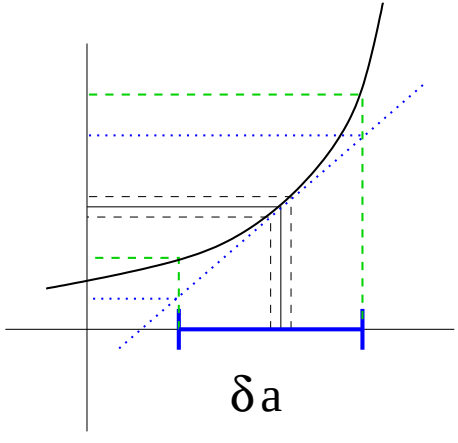
Note: let us have measurements a_i, with expectation value $\langle a \rangle$ and "error" δa. Let us block this into M "normal" blocks a_m, jackknife block averages $a_{(m)}$ and N_B bootstrap averages a_i^*. Now

1. $a_m - \langle a \rangle \sim \delta a \sqrt{M}$
2. $a_{(m)} - \langle a \rangle \sim \delta a / \sqrt{M}$
3. $a_i^* - \langle a \rangle \sim \delta a$

Thus, jackknife "resamples" the distribution on a very narrow range, whereas bootstrap gives the "right" range.

For example, we may have a function $f(\langle a \rangle)$ which happens to be significantly non-linear in range δa.

For jackknife, $\delta f \sim |f'(\langle a \rangle)| \delta a$.



Example: reweighting

$$\langle O \rangle_\beta = \frac{\sum_i O_i \, e^{-(\beta-\beta_c)E_i}}{\sum_i e^{-(\beta-\beta_c)E_i}}$$

If the number of bootstrap samples is very large, it can become expensive to evaluate the sums over the measurements (for each bootstrap block, we have a sum over the full number N measurements!). This can be optimized by precalculating sums over the original blocks.

Bootstrap and jackknife are also used in *fits* to the data.

6.3. Bias estimation

- For concreteness, let us again have a function $f(\langle a \rangle)$. The true value $f(\langle a \rangle)$ is obtained when we calculate the average of a using infinitely many samples a_i.
- Let us now denote the average over N samples $a_1 \ldots a_N$ as $\langle a \rangle_N$, from which we obtain an estimate $f(\langle a \rangle_N)$ Now, if we take the *average over many samples of size* N, we can calculate the expectation value $\langle f(\langle a \rangle_N) \rangle$.
- ***Bias*** is now defined as

  $$\text{Bias} = \langle f(\langle a \rangle_N) \rangle - f(\langle a \rangle)$$

  i.e. the deviation of the expectation value using N measurements from the one using ∞ measurements.
- Note that the bias is a well-defined quantity, the expectation values do not have “errors” (like a single $f(\langle a \rangle_N)$ has).

- Many quantities do not have a bias – for example, simple observables like magnetization M. However, a function of $\langle M \rangle$ like $\langle M \rangle^2$ has a bias!
- The original purpose for the jackknife method was bias reduction (Quenouille, J.Roy.Statist.Soc.Ser.B, 1949).

- Calculate jackknife block averages $a_{(n)}$ from data a_i, and let $f_{(n)} = f(a_{(n)})$ (let me assume here that the original data a_i are already "blocked", if needed).
- Let $\langle f_{(\cdot)} \rangle = \dfrac{1}{N} \sum_n f_{(n)}$.
- Now the Queinouille estimate for bias is

$$\text{Bias} = (N-1)(\langle f_{(\cdot)} \rangle - f(\langle a \rangle)).$$

- This leads to the ***bias-corrected jackknifed estimate*** of f:

$$\tilde{f}(\langle a \rangle) = f(\langle a \rangle) - \text{Bias} = N f(\langle a \rangle) - (N-1)\langle f_{(\cdot)} \rangle$$

- The usual rationale for bias correction goes as follows: if we assume that

$$\langle f(\langle a \rangle_N) \rangle = f(\langle a \rangle) + \frac{c_1}{N} + \frac{c_2}{N^2} + \dots$$

  Now, because each jackknife blocked set contains only $N - 1$ measurements,

$$\langle f_{(\cdot)} \rangle = \langle f(\langle a \rangle_{N-1}) \rangle.$$

  Now we can eliminate the "error" term $\propto 1/N$, giving us the bias correction result.

- While in principle the bias correction can be performed, in practice it is often of very limited use. Usually the bias correction is completely overwhelmed by the statistical error of the sample.