Heisenberg model

The Heisenberg model can be seen as a generalization of the Ising model, where a more realistic model of classical magnetization is represented by the Hamiltonian

$$H = -J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j, \tag{1}$$

where $\vec{S}_i \in \mathbb{S}^2 \subset \mathbb{R}^3$ belongs the surface of a 3-sphere.

While for the XY model (spins in $\mathbb{S}^1$) it is advantageous to use polar coordinates, it is unclear and possibly system dependent if in the 3D spin case the added complexity and FLOP introduced by spherichal coordinates, justifies the memory savings.

Implement a MC simulation of the Heisemberg model with a Metropolis algorithm. Outside a few changes in the update probability computation and the representation of you cluster, you should not need to modify the code from previous exercises much. Remember to normalize your vectors if you use a cartesian representation.

Taks 1: Compute the critical temperature for $J = 1$. You can use either the binder cumulants, or the magnetic susceptibility. You should find $T_c \approx 1.443$.

Task 2: Compute the autocorrelation time, either for E or $|\vec{M}|$, at T_c and find the critical dynamical exponent given by the relation $\tau \propto L^{z_c}$.

The cluster algorithm can be extended to system with spins of arbitrary dimensions, with the trick of considering reflections around a random plane at each MC step. For each step select a random unit vector $\vec{r}$ which is chosen to be orthogonal to the reflection plane. Decide if you want to implement Swedsen-Wang or Wolff, grow a cluster with bond probability

$$p_{i,j} = 1 - \exp\Big[-2\beta(\vec{S}_i \cdot \vec{r})(\vec{S}_j \cdot \vec{r})\Big] \tag{2}$$

and flip the spins as

$$\vec{S}_i{}' = \vec{S}_i - 2\vec{r}\,(\vec{S}_i \cdot \vec{r}). \tag{3}$$

Task 3: Repeat the computation of T_c and z_c using the Swendsen-Wang or Wolff algorithm.

Unsupervised machine learning

We can use PCA and k-means to investigate condensed matter systems. One of the paradigmatic systems is the two-dimensional ferromagnetic Ising model. It has two phases separated by a well-known critical temperature at $T_c = 2.26$.

Task 1: Generate $10^3 - 10^4$ samples of L-by-L Ising systems using Metropolis or a cluster Monte Carlo method, with $L = 10$ for smaller systems, and $L = 32$ for larger systems at different temperatures.

Task 2: Perform PCA on these data points and find the 2 directions u_1 and u_2 with the largest variance. Make a scatter plot of the configurations in the u_1-u_2 plane. Describe and interpret your results. What is the difference between the small and the large system?

Hint: Use fit(PCA, X; maxoutdim = 2) of the package MultivariateStats.

We can use k-means to determine the boundary between the three (two) phases, so that when we are given a new data point, we can determine which phase it belongs to.

Task 3: Find the three stabilized centroids using k-means and divide the u_1-u_2 plane into three phase regions.

Hint: Use kmeans(X, 3) of the package Clustering.

A very interesting phenomena can be observed for the data point in u_1 and u_2 direction.

Task 4 a): Plot the mean value of the absolute value of the data points in the u_1 direction as a function of temperature and compare it with the mean magnetization of the corresponding spin configurations.

Task 4 b): Plot the mean value of the absolute value of the data points in the u_2 direction as a function of temperature and compare it with the susceptibility of the corresponding spin configurations.