古典 $O(3)$ 模型の効率的な Monte Carlo シミュレーション

tarotene

2019 年 1 月 25 日

グラフ $G=(V,E)$ 上の連続スピン系でスピン次元が 3 のものを考える．例えば Heisenberg 模型はこれに合致する．この時，各頂点 $v\in V$ 上のスピン変数を $\boldsymbol{\sigma}_v$ と書くと，全系のスピン配位の情報は

$$\boldsymbol{\sigma}_v=\begin{pmatrix}\cos\phi\sin\theta\\ \sin\phi\sin\theta\\ \cos\theta\end{pmatrix} \tag{1}$$

なる極座標 (ϕ,θ)（$0\leq\phi<2\pi, 0\leq\theta<\pi$）で管理できる．これを離散化し，

$$\phi=\frac{2\pi}{P_\phi}i_\phi \tag{2}$$

$$\theta=\frac{\pi}{P_\theta}i_\theta \tag{3}$$

なる整数変数 (i_ϕ,i_θ)（$0\leq i_\phi<P_\phi, 0\leq i_\theta<P_\theta$）で持っておくことにする．但し，$P_\phi,P_\theta$ は**メッシュパラメータ**であり，地球儀をそれぞれ緯度・経度方向に何等分に区切るかということを表す．

Monte Carlo シミュレーションではよく，**シングルフリップ**という方法で系の状態を更新する．シングルフリップでは，ランダムに頂点 $r_v\in V$ を選び，その頂点上のスピン $\boldsymbol{\sigma}_{r_v}$ をある確率 p でランダムな方向 $\boldsymbol{\sigma}'_{r_v}$（$\neq\boldsymbol{\sigma}_{r_v}$）に振り向けるということを繰り返し行い，確率 p を制御することで目当ての定常分布を達成できるようにする．定常分布として温度 T のカノニカル分布を取れば，温度 T の平衡状態が再現する．ここで，乱数が必要となる箇所が 2 箇所ある．それは，頂点 r_v の選択と振り向ける方向 $\boldsymbol{\sigma}'_{r_v}$ の選択である．前者は一様分布の普通の乱数を離散化すれば良くて，後者はちょっと工夫が必要となるが，振り向ける方向が球面上で一様分布すれば良さそうである．

まず，通常の直交座標 (x,y,z) で見た時の球面上の一様分布乱数は，今の場合は 3 次元の Gauss 乱数を取り，2-ノルムを規格化することで得られる[*1]．これを式 (1) の規則に従って極座標 (ϕ,θ) に変換するには，

$$\phi=\begin{cases}\arctan(y/x)+\pi & \text{for } 0\leq x\\ \arctan(y/x)+2\pi & \text{for } x<0, 0\leq y\\ \arctan(y/x) & \text{else}\end{cases} \tag{4}$$

$$\theta=\arccos z \tag{5}$$

を計算する必要がある．すなわち，単純に逆三角関数を取るのみではダメである．しかし，多くのプログラミング言語では ϕ の計算を組み込み関数によって正しく行える．例えば，C++ の標準ライブラリ `cmath` では，

[*1] 多くの場合，分散共分散行列をパラメータとする疑似乱数生成器が必要となる．分散共分散行列は単位行列に取れば良い．また，乱数が ${}^{\mathrm{t}}(0,0,0)$ の時だけ除外するように注意すれば良い．

`std::atan2(y, x)` というメンバ関数が

$$\phi = \begin{cases} \arctan(y/x) & \text{for } 0 \leq x \\ \arctan(y/x) + \pi & \text{for } x < 0, 0 \leq y \\ \arctan(y/x) - \pi & \text{else} \end{cases} \tag{6}$$

を与えるし，Java ではクラス `Math` に `public static double atan2(y, x)` というメソッドが存在し同じ結果を与える．多くの場合 `atan2` という名前の関数ではこのように $-\pi \leq \phi < \pi$ が値域になるので，結果に π を足せば (1) の規則に持っていける．すなわち，直交座標 (x, y, z) で見た一様分布乱数 $\boldsymbol{r}$ を入力とし，整数変数 (i_ϕ, i_θ) を出力とする関数を作ることが出来る[*2]．

相互作用が 2 体の場合，Hamiltonian の一般形は f 及び g を任意の関数として

$$H(\{\boldsymbol{\sigma}_i\}) := -\sum_{\langle i,j\rangle \in E} f(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) - \sum_{i \in V} g(\boldsymbol{\sigma}_i) \tag{7}$$

である．例えば，普通の Heisenberg 模型ならば

$$f(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) = J_{i,j} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j \tag{8}$$

$$g(\boldsymbol{\sigma}_i) = \boldsymbol{h}_i \cdot \boldsymbol{\sigma}_i \tag{9}$$

と取れば良い．系の状態は，各々のスピン $\boldsymbol{\sigma}_i$ が属する 2 次元球面 $\left(S^2\right)_i$ の直積空間の元 $\mu \in \prod_{i \in V} \left(S^2\right)_i$ として表現される．

シングルフリップによって系の状態を μ から ν へ更新することを考える．この場合，状態 μ と ν はあるサイト i のスピン $\boldsymbol{\sigma}_i$ を除いて等しいので，この時のエネルギー変化 $\Delta E_{\nu,\mu} := H(\nu) - H(\mu)$ は

$$\Delta E_{\nu,\mu} = -{\sum_j}' \left\{ f(\boldsymbol{\sigma}'_i, \boldsymbol{\sigma}_j) - f(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) \right\} - \left\{ g(\boldsymbol{\sigma}'_i) - g(\boldsymbol{\sigma}_i) \right\} \tag{10}$$

で与えられる．但し，$\boldsymbol{\sigma}'_i$ は状態 ν におけるサイト i のスピンであり，和 ${\sum_j}'$ は $\langle i, j\rangle \in E$ であるような j について実行される．一般に，定常分布として温度 $T = 1/(k_{\mathrm{B}}\beta)$ のカノニカル分布を達成するには，例えばフリップ確率を $p = \min\left\{1, \mathrm{e}^{-\beta \Delta E_{\nu,\mu}}\right\} =: p^{(\mathrm{M})}_{\nu,\mu}(\beta)$ に取るのが良い．$p^{(\mathrm{M})}_{\nu,\mu}(\beta)$ は Metropolis 確率と呼ばれる．Metropolis 確率の指数関数部分の計算は，式 (10) を用いると

$$\mathrm{e}^{-\beta \Delta E_{\nu,\mu}} = \exp\left[\beta \left\{ g(\boldsymbol{\sigma}'_i) - g(\boldsymbol{\sigma}_i) \right\}\right] \times {\prod_j}' \exp\left[\beta \left\{ f(\boldsymbol{\sigma}'_i, \boldsymbol{\sigma}_j) - f(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) \right\}\right] \tag{11}$$

より，事前に

$$p^{(1)}_{\nu,\mu}(\beta) := \exp\left[\beta \left\{ g(\boldsymbol{\sigma}'_i) - g(\boldsymbol{\sigma}_i) \right\}\right] \tag{12}$$

$$p^{(2)}_{\nu,\mu}(\beta) := \exp\left[\beta \left\{ f(\boldsymbol{\sigma}'_i, \boldsymbol{\sigma}_j) - f(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_j) \right\}\right] \tag{13}$$

[*2] 規格化ができないケース $\boldsymbol{r} = {}^{\mathrm{t}}(0, 0, 0)$ が無きにしもあらずなので，1 つずつ（ないしはある程度のまとまった個数ずつ）の乱数を MKL 等で引いて，規格化できるものはまとめて処理し，$\left\{ \left(i^{(j)}_\phi, i^{(j)}_\theta \right) \right\}$ という集合が j について連番になるようにすると良い．その時，規格化できないものがあれば残りの連中で詰める必要がある．よって，乱数の多次元配列を確保するのにはノードベースの連結リストが適していることが分かる．あるいは，ノードベースのデータ構造を利用する場合は，規格化出来るものだけ `push_back(●)` して，要素数がある数を超えたら順次 `pop_front()` によって利用するというスタイルでも良い．

を計算して配列に格納しておけば，時間計算量を削減することが出来る[*3]．具体的には，1 回の $\mathrm{e}^{-\beta\Delta E_{\nu,\mu}}$ の計算にグラフの平均次数 k_{ave} 程度の乗算が必要となる[*4]．

一方，空間計算量については，$p^{(1/2)}_{\nu,\mu}(\beta)$ がそれを全て決める．1 体スピンの寄与 $p^{(1)}_{\nu,\mu}(\beta)$ については，始状態 μ におけるサイト i のスピン $\boldsymbol{\sigma}_i$ と終状態 ν におけるそれ $\boldsymbol{\sigma}'_i$ が確定すれば求まるので，スピン変数 $\boldsymbol{\sigma}_i$ の状態数を P_{tot} とすれば $P_{\mathrm{tot}}(P_{\mathrm{tot}}-1)$ 要素が必要である．また，同様に 2 体スピンの寄与 $p^{(2)}_{\nu,\mu}(\beta)$ については，相互作用する相手 $\boldsymbol{\sigma}_j$ の状態も自由度として加味する必要があるため，$P^2_{\mathrm{tot}}(P_{\mathrm{tot}}-1)$ 要素が必要である．合計で

$$P_{\mathrm{tot}}(P_{\mathrm{tot}}-1)+P^2_{\mathrm{tot}}(P_{\mathrm{tot}}-1)=P_{\mathrm{tot}}(P^2_{\mathrm{tot}}-1) \tag{14}$$

だけの要素が必要となる．

今考えている模型では各スピンの状態数は $P_{\mathrm{tot}}=P_\phi\times P_\theta$ なので，例えば $P_\phi=24, P_\theta=12$ で 8byte の浮動小数点数を用いる場合は

$$P_{\mathrm{tot}}(P^2_{\mathrm{tot}}-1)=8\times 288\times(288^2-1)\sim 10\times 300^3=270\times 10^6 \tag{15}$$

より，270MB から 300MB は見込む必要がある．しかし，これは現代的なハードウェアの構成からすると現実的なメモリ使用量であり，手元の PC でも十分に実現可能である．

$P_\phi=1, P_\theta=2$ と取れば，よく知られた Ising 模型が得られる．Ising 模型でよく行われるのは，d 次元の立法格子で自分自身と全ての配位 $i_z=1,2,\ldots,2d$ についての合計 $(2d+1)$ 個のスピンを入力とする $\mathrm{e}^{-\beta\Delta E_{\nu,\mu}}$ の計算である．Ising 模型ではスピンの情報が 2 値で表現され，振り向ける方向も逆向き（`NOT`）の一通りであり，また 2 値であることを利用したマルチスピンコーディングなども可能なので，極めて効率の良い情報圧縮ができてしまうのである．しかし，これをナイーブに拡張して球面上の連続スピン系（の離散化）に持っていこうとすると，$\mathrm{e}^{-\beta\Delta E_{\nu,\mu}}$ を保存するための配列の要素数が $\mathcal{O}(P^{k+2}_{\mathrm{tot}})=\mathcal{O}(P^{2(d+1)}_{\mathrm{tot}})$ となることが確かめられる．これは，グラフ G の形状を自由に変えたりした時の次数 k の変化に対して明らかにロバストではない．この意味で，本稿で扱うアルゴリズムとデータ構造はこのようなグラフ構造に対して（次数が頂点によって変わる場合などのややこしいコーナーケースの処理を除けば）ロバストである．

以上のアルゴリズムを実際に運用するにあたっては，整数変数のペア $(i_\phi,i_\theta)_i$ とスピンそのものの向き $\boldsymbol{\sigma}_i={}^{\mathrm{t}}(\sigma_{i,x},\sigma_{i,y},\sigma_{i,z})$ を相互変換するルーチンが必要となる．前者から後者へのルーチンを `decodeSpin()`，その逆を `encodeSpin()` と書くと，まず最初に $\mathrm{e}^{-\beta\Delta E_{\nu,\mu}}$ の配列を用意するのに `decodeSpin()` を呼び出し，計算は直交座標乱数を極座標乱数に落とす際に `encodeSpin()` が使われる以外は $\mathrm{e}^{-\beta\Delta E_{\nu,\mu}}$ の配列を介して (i_ϕ,i_θ) だけで行われ，一定時間おきにスナップショットを撮るのには `decodeSpin()` が用いられると言った具合に，使う場面は多い[*5]．

[*3] 必要に応じてその都度指数関数を計算する戦略だと，in-place になる代わりに凡そ 10 倍の速度低下が見込まれる．勿論，これは現時点でのハードウェア性能とアルゴリズムを前提とした話であり，将来的に大きく改善される可能性はある．しかし，シンプルな加算や乗算を前提とした計算量解析が出来るためには，内部的に演算回数が不定のルーチンを基本単位とするアルゴリズムは好ましくないという気持ちがある．指数関数の計算ルーチンは，gcc や icc などの比較的よく利用されるコンパイラでは $\exp(a)=\exp(\text{切りの良い } a)\exp(a \text{ の残り})$ と分解し，$\exp(\text{切りの良い } a)$ を配列，$\exp(a \text{ の残り})$ を任意次数の Taylor 展開で求めるようである [1]．この時，配列のサイズと Taylor 展開の次数がそれぞれ空間計算量と時間計算量に対応するが，これらを変えると加算や乗算の回数が大きく変化し，全体のアルゴリズムに影響を与える．よって，ベストプラクティスとしてはまず指数関数演算の回数を最小化しておき，どうしても必要な場合には内部的に調節し直すのが良いと筆者は考えている．一方，引いた一様乱数 p_r の対数 $\log p_r$（$p_r>0$）をエネルギー変化 $\Delta E_{\nu,\mu}$ と比較するという戦略を取れば，また話が異なってくる．この場合は，負のオーバーフロー値 $d_{\min}$ に対応する引数 $p_r=\exp[-|d_{\min}|]$ より小さな値を弾く必要がある．いずれの場合も，値そのものの精度を著しく欠いてはいけないという制約がある．

[*4] 次数が頂点毎に異なる場合でも，シングルフリップの場合は頂点のランダムな選択がこれを保証する．

[*5] 組込の逆三角関数の呼び出しがあるので，先に述べた指数関数の計算量云々の話は，本当はこれとの差し引きで語られなければい

References

[1] 高速な倍精度指数関数 *exp* の実装. URL: https://www.slideshare.net/herumi/exp-9499790 (visited on 01/25/2019).

けない．しかし，逆三角関数の計算精度はメッシュパラメータ $\left(P_\phi, P_\theta\right)$ に応じたまで落とすことが出来るので，その考慮も必要である（どうせ `INT()` で打ち切るのだから，その打ち切り精度以上の計算はサボれるという意味）．