

シミュレーション工学

モンテカルロ法(3) 発展的なアルゴリズム

慶應義塾大学大学院理工学研究科基礎理工学専攻物理情報専修

渡辺宙志

はじめに

モンテカルロ法とは

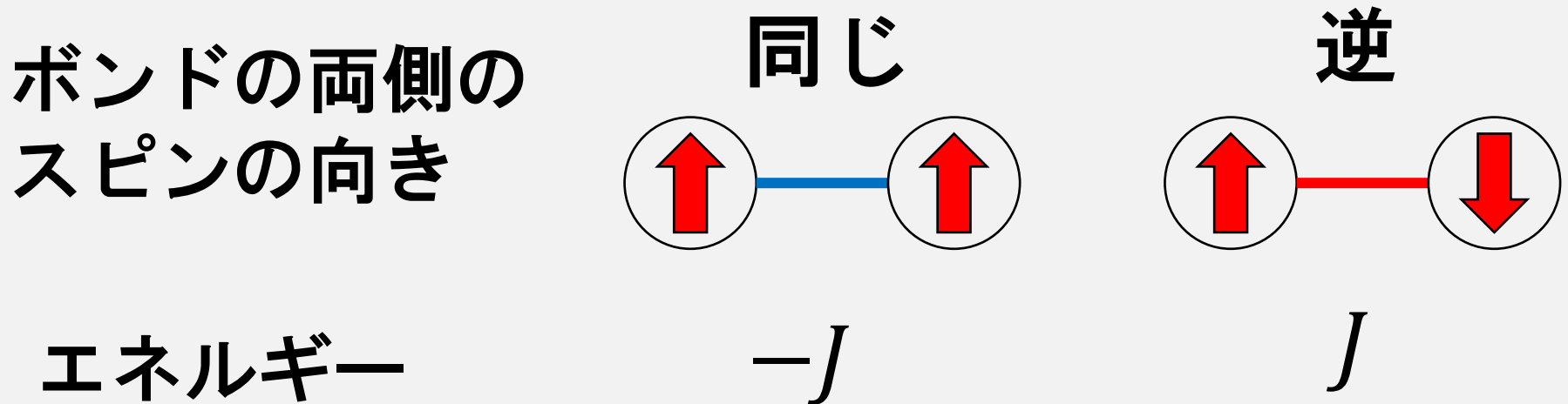
- 数値計算では、何かの和や積分の推定値を計算することが多い
- 和の形を変形することで、異なるアルゴリズムが生まれる

本講義の目的

- モンテカルロサンプリングと分配関数の関係を知る
- アルゴリズムを工夫することで、効率的なサンプリングができることを知る

イジング模型

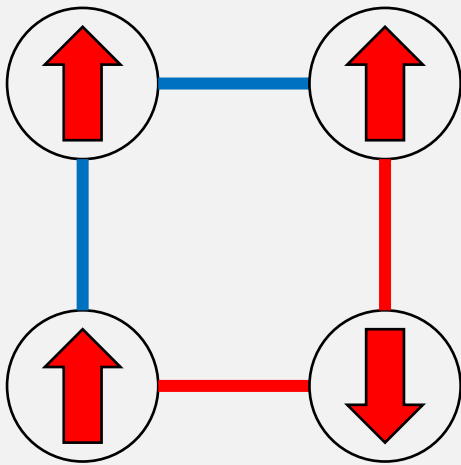
- 格子の各点に**スピン**(小さな磁石)がある
- スピンは「上」と「下」の状態がある
- 隣り合うスピンをつなぐ線を**ボンド**と呼ぶ



$J > 0$ ならスピンは揃いたがる(強磁性的)
 $J < 0$ ならスピンは逆向きを好む(反強磁性)

イジング模型

格子の上の全てのボンドについてエネルギーの和を取り、この系の全エネルギーとする



$$E = -2J + 2J = 0$$

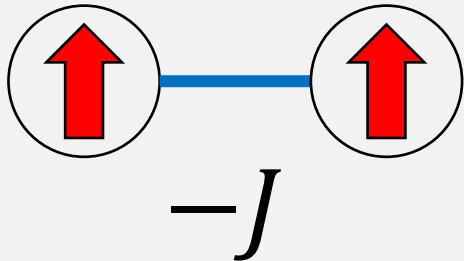
このような模型をイジング模型(Ising Model)と呼び、
磁性体の簡単なモデルになっている
以下、強磁性 ($J > 0$) の場合を考える

イジング模型

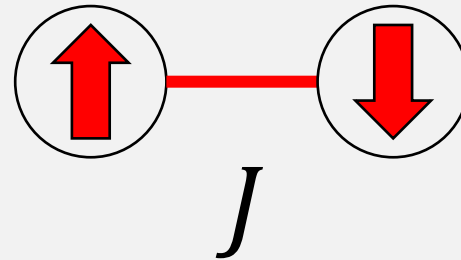
i 番目のスピンの状態を σ_i とする

σ_i の値は1(上向き)か-1(下向き)のいずれか

$$\sigma_i = 1 \quad \sigma_j = 1$$



$$\sigma_i = 1 \quad \sigma_j = -1$$



両方まとめて $-J\sigma_i\sigma_j$ と書ける

イジング模型

全系のエネルギーは以下のように書ける

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j$$

系の全てのボンドについて和をとるという意味

全系のエネルギーを与える量をハミルトニアンとよぶ

ボルツマン重み

系の状態に通し番号をつけ、 i 番目の状態のエネルギーを E_i とする



この状態の出現確率がボルツマン重みに比例する

$$\exp(-\beta E_i)$$

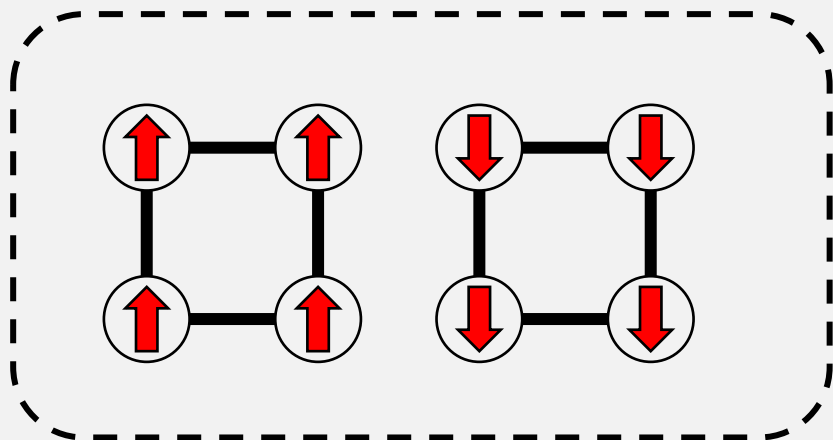
ボルツマン定数 k_B

逆温度 $\beta = 1/k_B T$

この時、エネルギーの温度依存性を知りたい

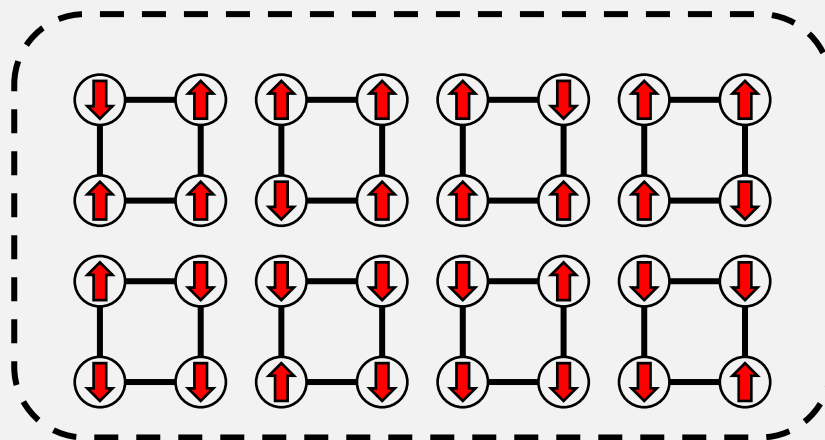
エネルギーの振る舞い

低温



スピンのそろった状態
出現確率は**高い**が、総数が**少ない**
→エネルギー重視
→低温で支配的

高温



スピンのバラバラの状態
出現確率は**低い**が、総数が**多い**
→エントロピー重視
→高温で支配的

エネルギーの振る舞い

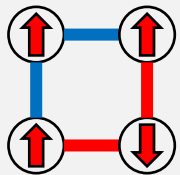
エネルギーの期待値の温度依存性

$$U(\beta) = Z^{-1} \sum_i E_i \exp(-\beta E_i) \quad Z = \sum_i \exp(-\beta E_i)$$

エネルギー E をとる状態の数を $g(E)$ とすると

$$U(\beta) = Z^{-1} \int E g(E) \exp(-\beta E) dE \quad Z = \int g(E) \exp(-\beta E) dE$$

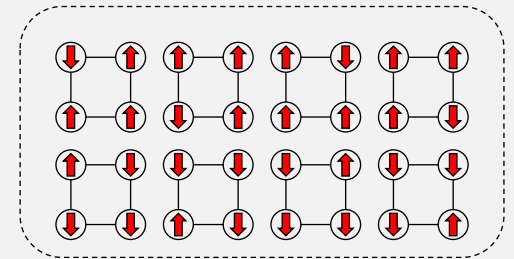
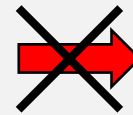
状態からエネルギー
の計算は簡単



$$E = 0$$

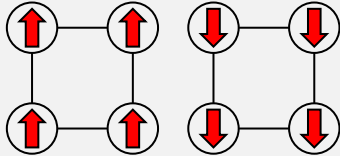
あるエネルギーをとる状態の
数の計算は大変

$$E = 0$$



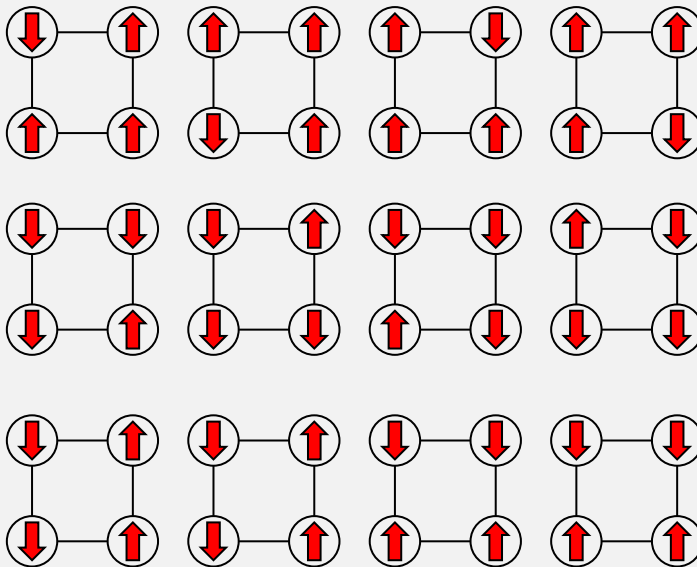
N=4の場合の状態数

$$E = -4J$$



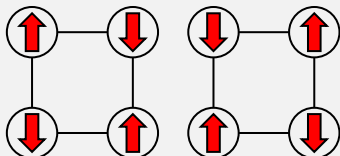
$$g(-4J) = 2$$

$$E = 0$$



$$g(0) = 12$$

$$E = 4J$$



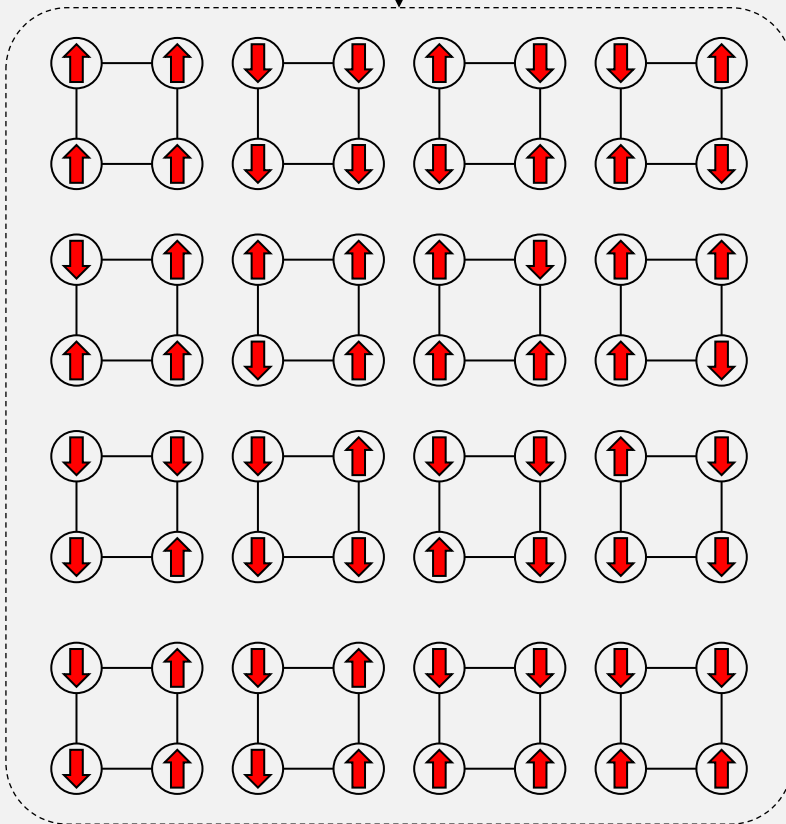
$$g(4J) = 2$$

これを一般のNで求めるのは極めて大変 ➡ モンテカルロサンプリング

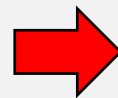
単純サンプリング

1. 全ての可能な状態から無作為に一つ選ぶ
2. その状態のエネルギー E_i と重み w_i を計算する
3. 1-2を繰り返し、重み付きで平均をとる

この中から無作為に一つ選ぶ



スピンの数が N 個なら、状態は 2^N 個
ほとんどの状態は重みが小さいので
サンプリング効率が悪い



マルコフ連鎖モンテカルロ法

マルコフ連鎖モンテカルロ法

サンプリング候補の決め方

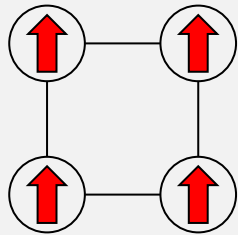
現在の状態から遷移可能な状態を限定し、その中から一つ無作為に選ぶ

イジング模型への適用

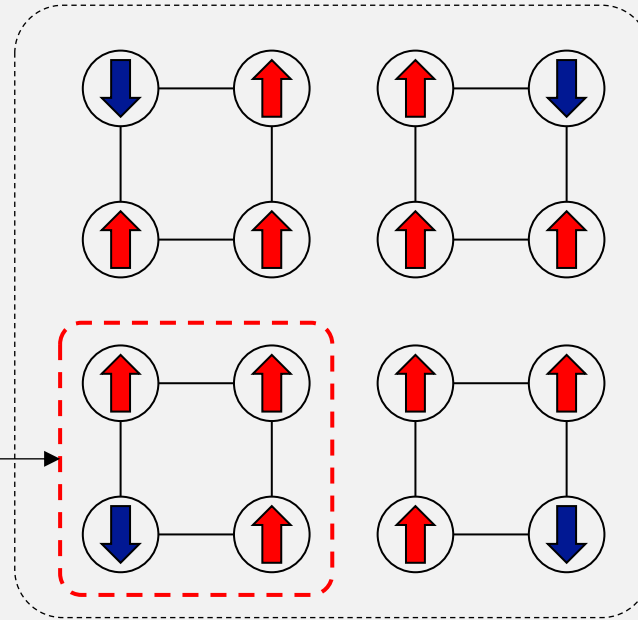
1. 現在の状態を A とし、スピンを一つ無作為に選ぶ
2. 選んだスピンを反転させた状態を提案状態 B とする
3. それぞれのエネルギーから遷移確率 $P(A \rightarrow B)$ を計算し、遷移させるか決める
4. 遷移しなかった場合は状態はそのまま、遷移した場合は提案状態を現在の状態にして1.へ

マルコフ連鎖モンテカルロ法

現在の状態



遷移可能な状態



この中から無作為に一つ選ぶ

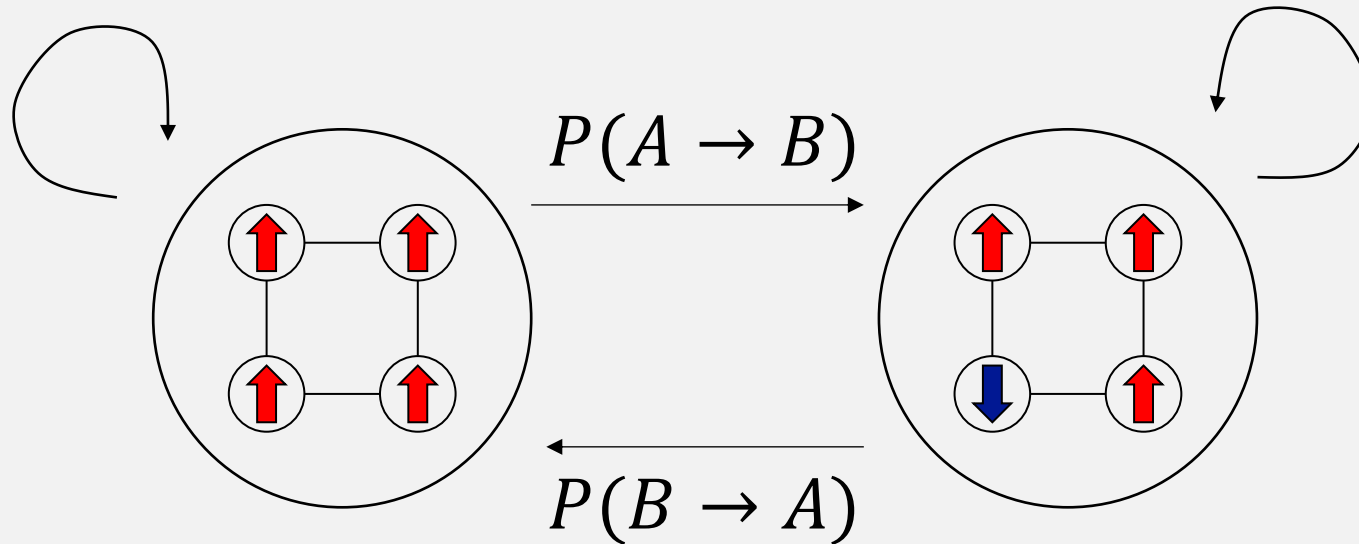
スピンの数が N 個なら総状態数は 2^N 、遷移可能な状態は N

➡ 遷移可能な状態が限定された

マルコフ連鎖モンテカルロ法

$$P(A \rightarrow A)$$

$$P(B \rightarrow B)$$



現在の状態A

現在の状態B

$$E_A = -4J$$

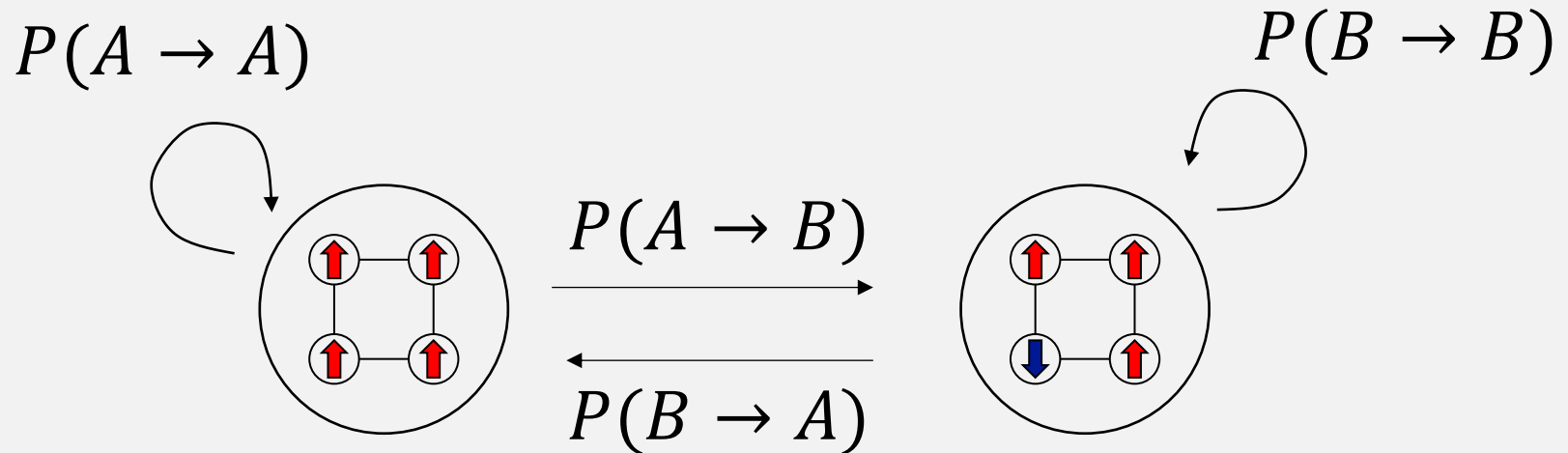
$$E_B = 0$$

$$w_A = \exp(-\beta E_A)$$

$$w_B = \exp(-\beta E_B)$$

$$E_A < E_B \text{ なので } w_A > w_B$$

マルコフ連鎖モンテカルロ法



確率の保存条件

$$P(A \rightarrow A) + P(A \rightarrow B) = 1$$

$$P(B \rightarrow B) + P(B \rightarrow A) = 1$$

詳細釣り合い条件

$$\frac{P(A \rightarrow B)}{P(B \rightarrow A)} = \frac{w_B}{w_A}$$

マルコフ連鎖モンテカルロ法

メトロポリス法(Metropolis method)

重みが大きくなる場合は必ず遷移、そうでない場合は確率的に遷移させる

$$P(B \rightarrow A) = 1$$

$$P(A \rightarrow B) = \frac{w_B}{w_A} = \frac{\exp(-\beta E_B)}{\exp(-\beta E_A)} = \exp(-\beta \Delta E)$$

$$\Delta E = E_B - E_A > 0 \text{なので、} \exp(-\beta \Delta E) < 1$$

提案状態のエネルギーが、現在の状態より低ければ必ず遷移
高ければ確率 $p = \exp(-\beta \Delta E) < 1$ で遷移

マルコフ連鎖モンテカルロ法

イジング模型にマルコフ連鎖モンテカルロ法を適用した場合のアルゴリズム(メトロポリス法を採用した場合)

1. スピンを一つ無作為に選ぶ
2. 選んだスピンを反転させた状態を提案状態とする
3. 現在の状態とのエネルギー差 ΔE を計算する
4. エネルギーが下がる場合($\Delta E < 0$)なら必ず遷移。そうでなければ確率 $p = \exp(-\beta\Delta E)$ で遷移
5. 1-4を繰り返す

一度に一つのスピンだけひっくり返すので
Single-Spin-Flip Algorithmと呼ぶ

ここまでのまとめ(復習)

状態に重みがある場合の期待値を厳密に計算することは難しい
その理由は同じ重みを持つ状態の数がわからないから
→サンプリングによる期待値の推定

単純サンプリング

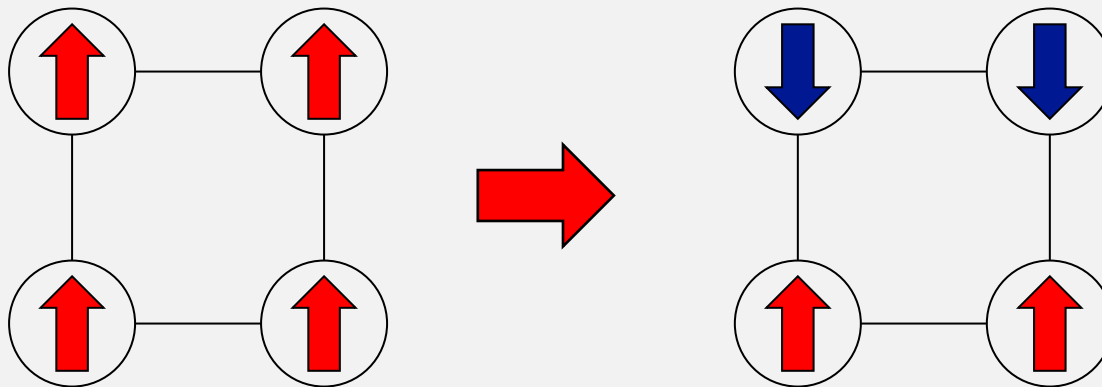
全ての状態から無作為に状態をサンプリングする
重みの小さい状態ばかりサンプリングするため非効率

マルコフ連鎖モンテカルロ法

現在の状態をもとに、サンプリングする状態を限定する
現在の状態と提案状態の重みを比較し、適切な遷移確率を
作ることによって重みに比例したサンプリングができる
(**重み付きサンプリング**)→効率的

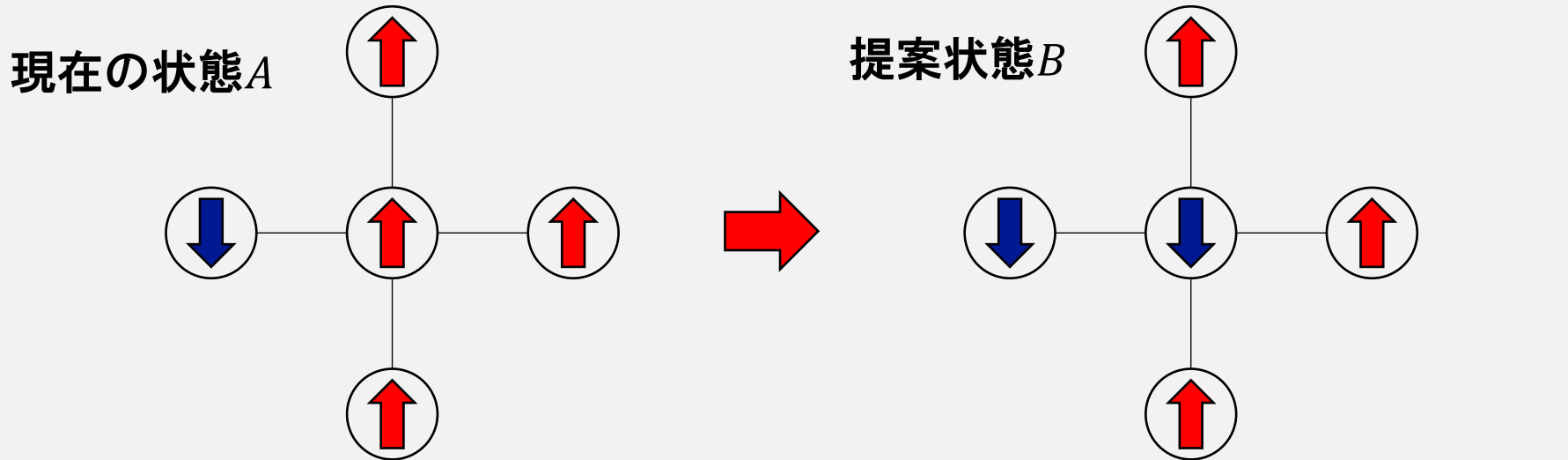
サンプリングの工夫

一度に一つしかスピンをひっくり返せないのは非効率的？



複数のスピンを同時にひっくり返せたら計算が加速するのでは？

スピンを一つだけひっくり返す場合



ひっくり返る

同じ向きのスピンの数 $n_+ = 3$
逆向きのスピンの数 $n_- = 1$

同じ向きのスピンの数 $n_+ = 1$
逆向きのスピンの数 $n_- = 3$

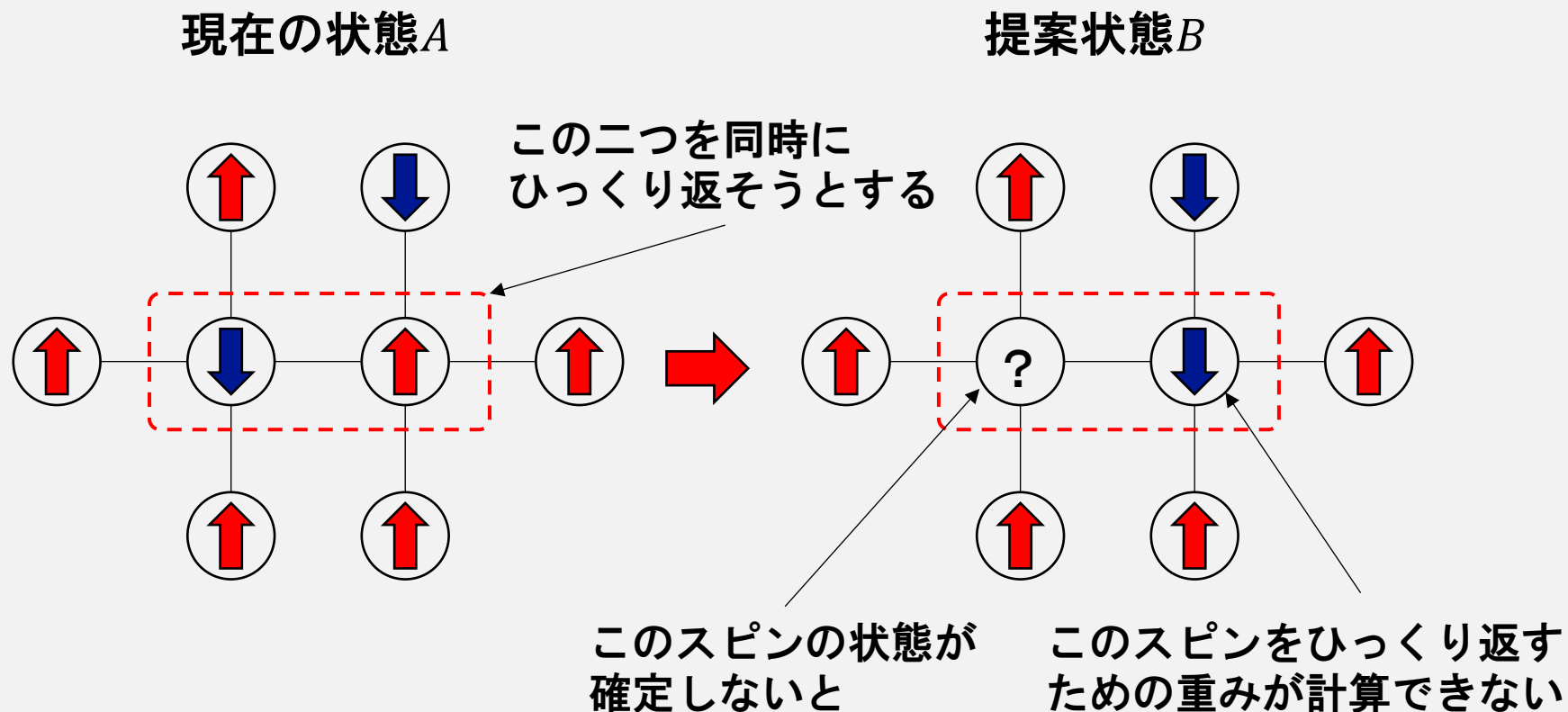
$$E_A = (n_- - n_+)J$$

$$E_B = (n_+ - n_-)J$$

$$\Delta E = E_B - E_A = 2(n_+ - n_-)J$$

エネルギー差が同じ向きのスピンの数を数えるだけで計算できる

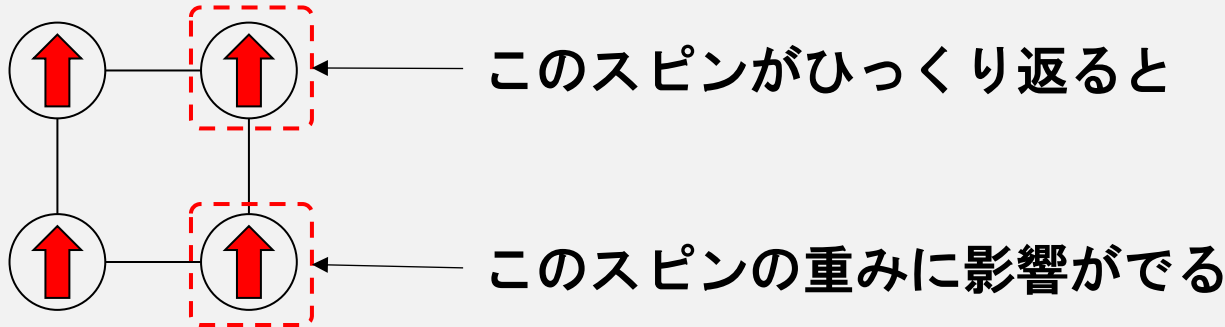
複数のスピンをひっくり返す場合



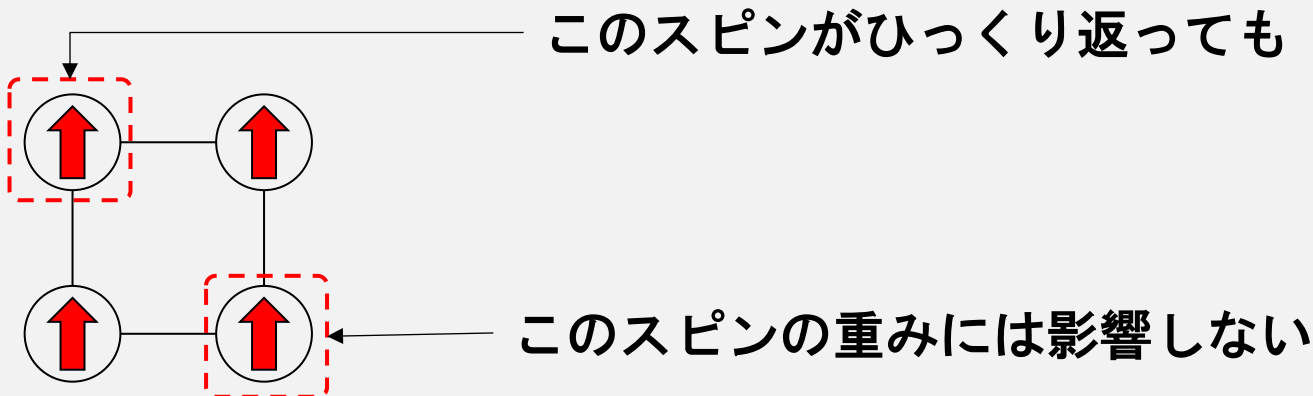
あるスピンの状態が確定していないと、隣のスピンをひっくり返す試行ができない

独立なスピン

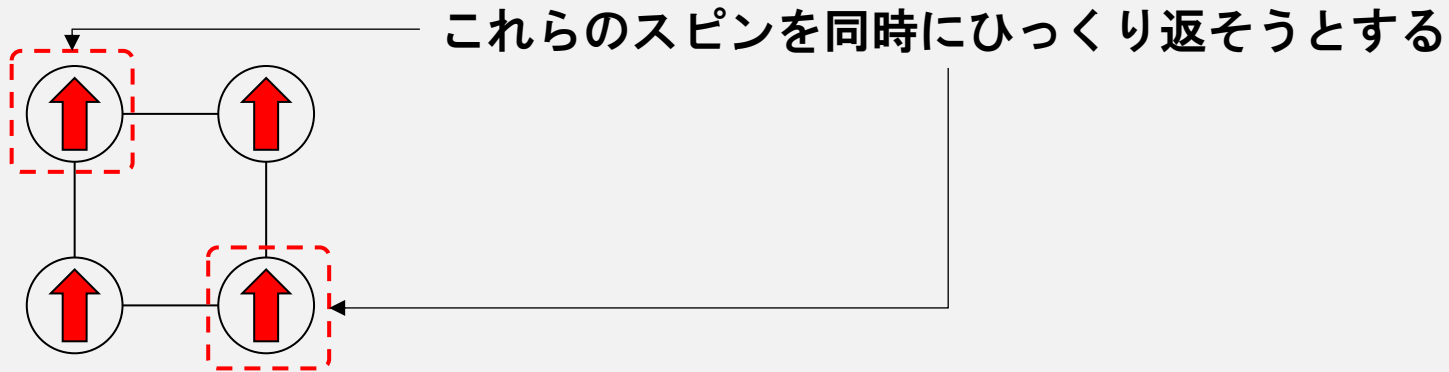
あるスピンのひっくり返ると、ボンドで繋がったスピンの重みが変わる



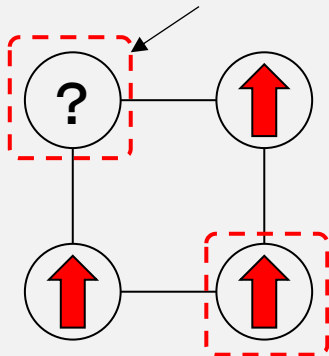
ボンドで接続されていないスピンは影響を受けない



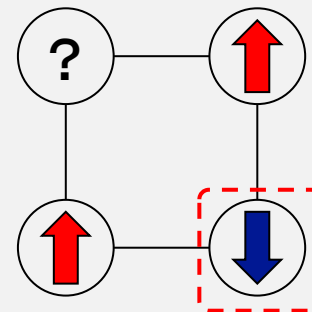
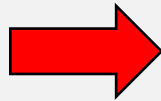
独立なスピン



このスピンをひっくり返す
試行の結果はまだ知らない

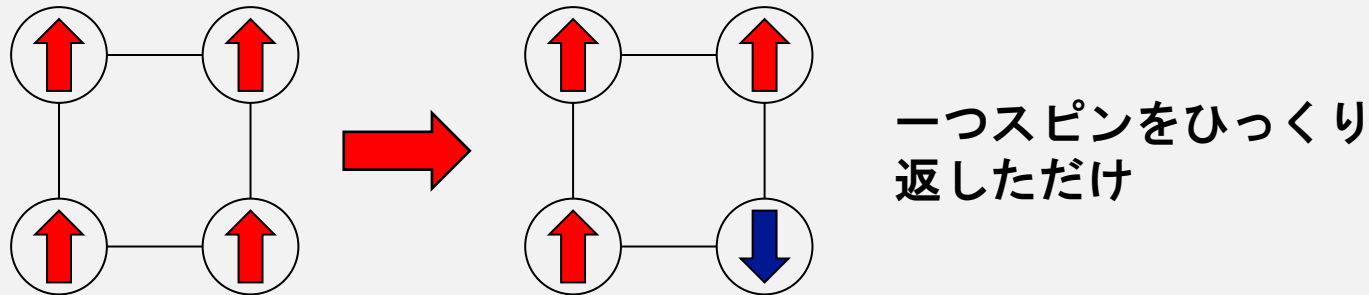


わからないまま確率を計算して
ひっくり返してしまう

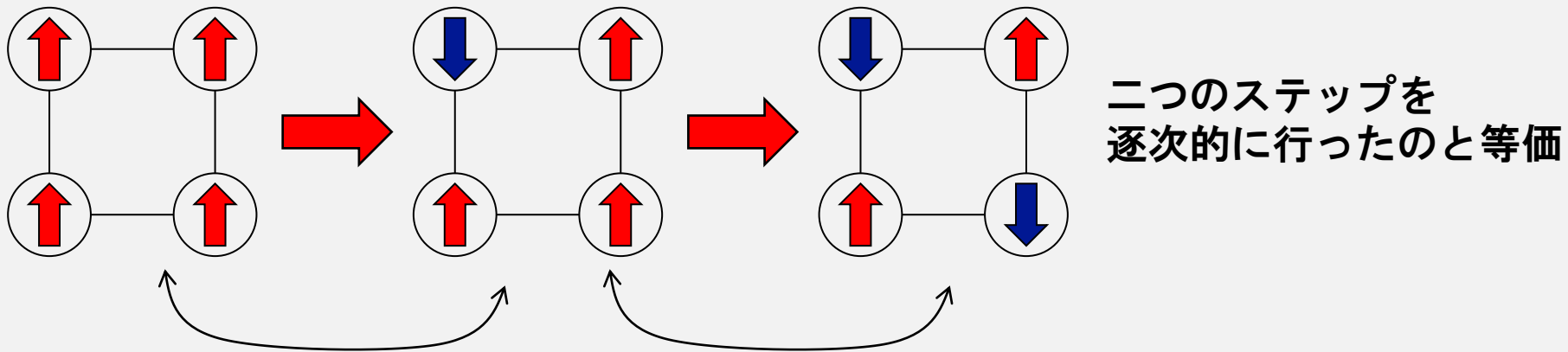


独立なスピン

もしスピンのひっくり返っていなければ？

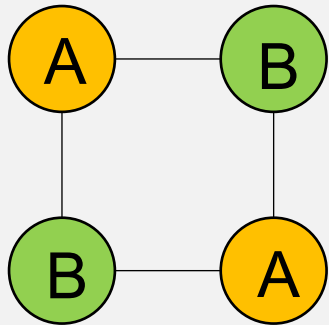


もしスピンのひっくり返っていたら？

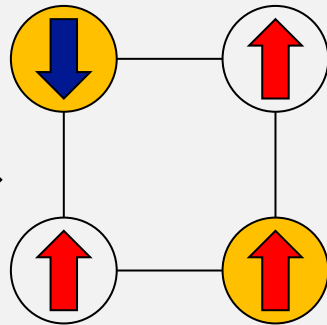
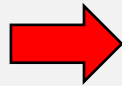
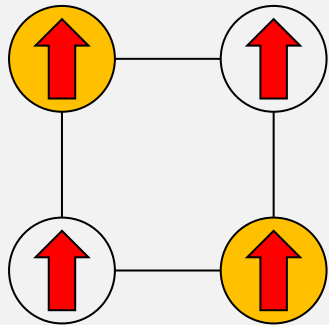


これらの状態間で詳細釣り合い条件が成り立っているので
正しいサンプリングができている

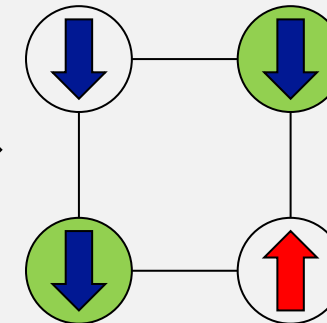
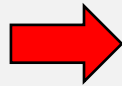
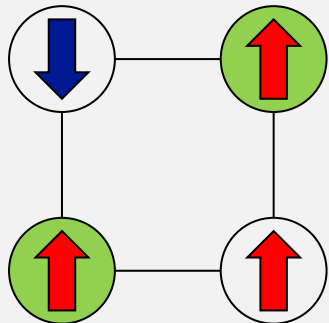
独立なスピン



ボンドで繋がっていない二つのグループに分け、
Aサイト、Bサイトと名前をつける



Bサイトのスピンを固定し、
Aサイトのスピンを全て独立
にひっくり返す試行を行う



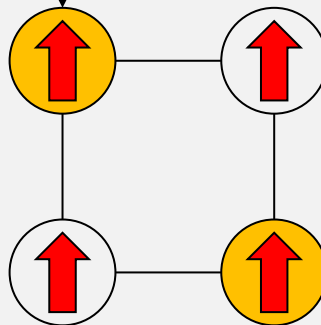
Aサイトのスピンを固定し、
Bサイトのスピンを全て独立
にひっくり返す試行を行う

独立なスピン

Q. 何がうれしいの? → A. 並列化できる

スレッド1

`flip(i)`



スレッド2

`flip(j)`

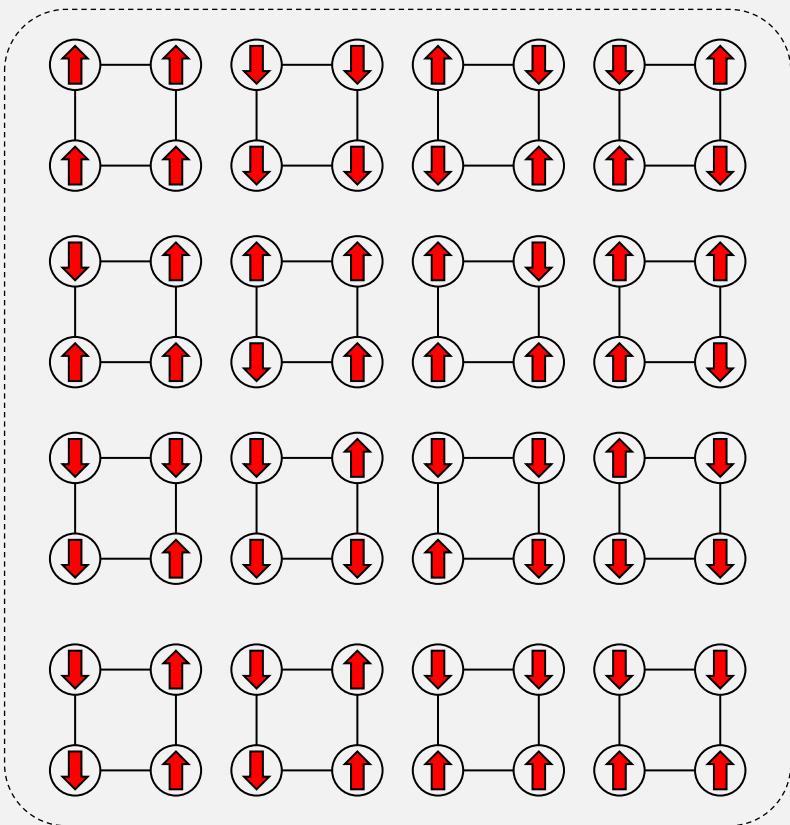
```
def flip(i, spin, beta):  
    dE = calc_energy_diff(i)  
    if check_flip(dE):  
        spin[i] = -spin[i]
```

他のスレッドの結果を待たずに自分の担当のスピンをひっくり返してよい

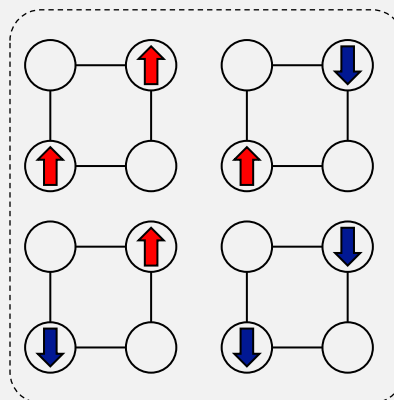
分配関数による表現

16個の状態を

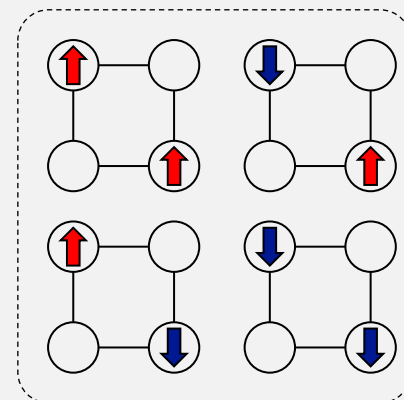
4x4個の状態に分離した



=



x



分配関数による表現

$$Z = \sum_{\{\sigma\}} \exp(-\beta H) = \sum_{\{\sigma_A\}} \sum_{\{\sigma_B\}} \exp(-\beta H)$$

全てのスピンの状態についての和
 2^N 通り

A、Bサイトそれぞれのスピンの状態についての和
 $2^{N/2}$ 通り

ハミルトニアンを代入する

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j$$

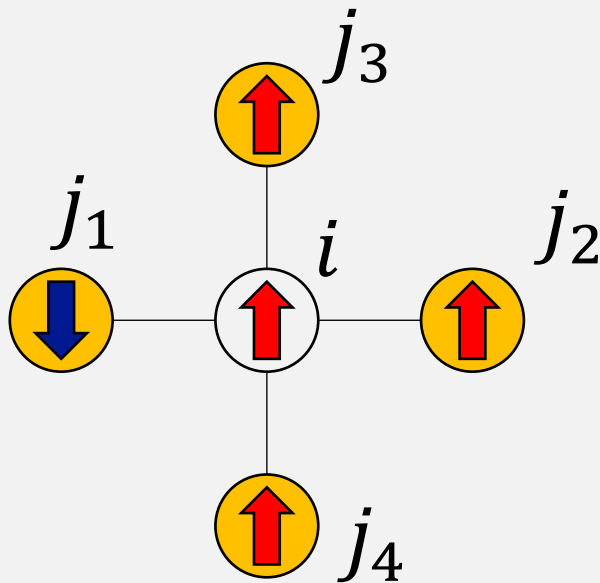

$$Z = \sum_{\{\sigma_A\}} \sum_{\{\sigma_B\}} \exp \left(\sum_{\langle i,j \rangle} \sigma_i \sigma_j \right)$$

分配関数による表現

$$Z = \sum_{\{\sigma_A\}} \sum_{\{\sigma_B\}} \exp \left(\sum_{\langle i,j \rangle} \sigma_i \sigma_j \right)$$

外側を固定して

内側の和についてのみサンプリングする



$$\sigma_i \sigma_{j_1} + \sigma_i \sigma_{j_2} + \sigma_i \sigma_{j_3} + \sigma_i \sigma_{j_4} = h_i \sigma_i$$

外側のスピンの固定されているので、
それをまとめてしまう

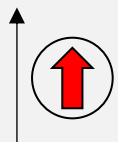
$$h_i \equiv \sigma_{j_1} + \sigma_{j_2} + \sigma_{j_3} + \sigma_{j_4}$$

分配関数による表現

$$Z = \sum_{\{\sigma_A\}} \sum_{\{\sigma_B\}} \exp \left(\sum_{\langle i,j \rangle} \sigma_i \sigma_j \right) = \sum_{\{\sigma_A\}} \sum_{\{\sigma_B\}} \exp \left(\sum_i h_i \sigma_i \right)$$

$$Z = \sum_{\{\sigma\}} \exp(-\beta H) = \sum_{\{\sigma_A\}} \sum_{\{\sigma_B\}} \exp(-\beta H) \quad \text{だったから}$$

$$H = -J \sum_i h_i \sigma_i \quad \text{というハミルトニアンを計算している気持ちになった}$$

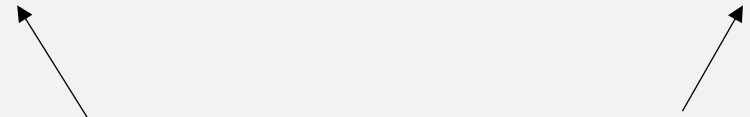


ローカルな磁場下にある独立なスピンの集まり

→ 相互作用が消えてしまった

→ 相互作用がないので好き勝手できる(例えば並列化)

分配関数による表現

$$Z = \sum_{\{\sigma_A\}} \sum_{\{\sigma_B\}} \exp \left(\sum_{\langle i,j \rangle} \sigma_i \sigma_j \right) = \sum_{\{\sigma_B\}} \sum_{\{\sigma_A\}} \exp \left(\sum_{\langle i,j \rangle} \sigma_i \sigma_j \right)$$


和の順序を入れ替えば、Aサイトについて同じことができる

アルゴリズム

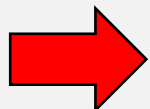
1. Aサイトのスピンを固定し、Bサイトのスピンのみ更新する
2. Bサイトのスピンを固定し、Aサイトのスピンのみ更新する
3. 以下繰り返し

計算の手間は変わらないが、更新しようとするスピンのみ更新が全て独立であるために並列化が可能になる

サンプリングの工夫のまとめ

- 分配関数の和の取り方を工夫すると、もとの系とは異なる形のハミルトニアンが現れる→異なる物理系をシミュレーションしているかのように見える
- 多重和の外側を固定し、内側だけサンプリングすると、スピンの独立になるために計算が楽になることがある

もっともうまく分配関数を変形すれば、より効率的なサンプリングが可能になるのでは？

 クラスターアルゴリズム


クラスターアルゴリズム

イジング模型のハミルトニアン

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j$$

その分配関数

$$Z = \sum_{\{\sigma\}} \exp(-\beta H)$$

$$Z = \sum_{\{\sigma\}} \exp \left(\sum_{\langle i,j \rangle} K \sigma_i \sigma_j \right)$$


ハミルトニアンを代入した
 $K \equiv \beta J$

$$= \sum_{\{\sigma\}} \prod_{\langle i,j \rangle} \exp(K \sigma_i \sigma_j)$$

指数関数の肩の和を外に出して
積に書き換えた

これを書き直す

クラスターアルゴリズム

$$\exp(K\sigma_i\sigma_j) = \begin{cases} e^K & \sigma_i = \sigma_j \\ e^{-K} & \sigma_i \neq \sigma_j \end{cases} \quad \text{であるから}$$

$$\begin{aligned} \exp(K\sigma_i\sigma_j) &= e^{-K} + \delta_{\sigma_i,\sigma_j}(e^K - e^{-K}) \\ &= e^K \left[e^{-2K} + \delta_{\sigma_i,\sigma_j} \underline{(1 - e^{-2K})} \right] \\ &\quad \quad \quad \equiv p \\ &= e^K \left[(1 - p) + p\delta_{\sigma_i,\sigma_j} \right] \end{aligned}$$

ここまではただの等式変形

クラスターアルゴリズム

以下のデルタ関数を使った恒等式を使って書き直す

$$x + y = \sum_{b=0}^1 x\delta_{b,0} + y\delta_{b,1}$$

$$e^K \left[(1 - p) + p\delta_{\sigma_i, \sigma_j} \right]$$

$$= \sum_{b_{ij}} e^K \left[(1 - p)\delta_{b_{ij},0} + p\delta_{\sigma_i, \sigma_j}\delta_{b_{ij},1} \right]$$

新たに導入された b_{ij} はスピン i とスピン j を結ぶボンド上で定義されており、**ボンド自由度**と呼ぶ

ボンド変数

スピン変数

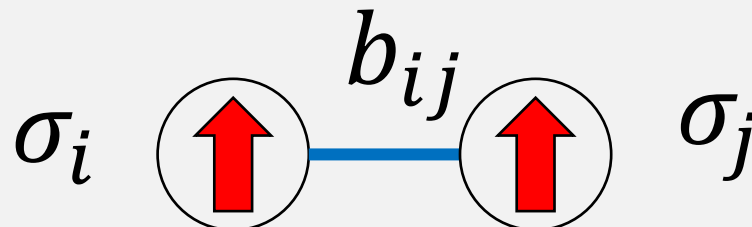
σ_i はサイト i 上で定義される
値として 1 か -1 をとる

1 の時に上向き、-1 の時に下向きと定義する

ボンド変数

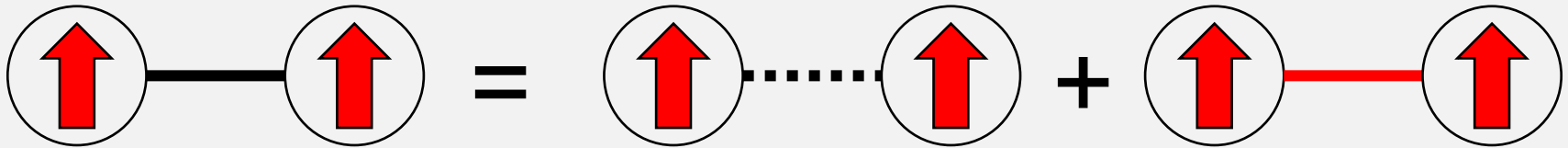
b_{ij} はスピン i と j をつなぐボンド上で定義される
値として 1 か 0 をとる

1 の時に active、0 の時に inactive であると定義する



クラスターアルゴリズム

$$\exp(K\sigma_i\sigma_j) = \sum_{b_{ij}} e^K \left[\underbrace{(1-p)\delta_{b_{ij},0}}_{\text{inactive}} + \underbrace{p\delta_{\sigma_i,\sigma_j}\delta_{b_{ij},1}}_{\text{active}} \right]$$



新たにボンド自由度を導入し、スピンの相互作用の重みをactiveなボンドとinactiveなボンドの和に分けた

クラスターアルゴリズム

以上の式変形をまとめると

$$Z = \sum_{\{\sigma\}} \exp(-\beta H)$$
$$= \sum_{\{\sigma_i\}} \sum_{\{b_{ij}\}} \prod_{\langle i,j \rangle} e^K \left[(1-p) \delta_{b_{ij},0} + p \delta_{\sigma_i, \sigma_j} \delta_{b_{ij},1} \right]$$

隣り合うスピンに関する積

ボンドに関する和

スピンに関する和

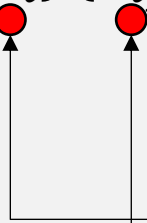
等価変形なのにボンドに関する自由度が増えた

ボンド変数

- 新たにボンド自由度が生まれ $b_{ij} = 0$ か $b_{ij} = 1$ の値をとる
- 分配関数が多重和で書かれているので、スピンに関するサンプリングとボンドに関するサンプリングを交互にとれる
- ボンド b_{ij} が active ($b_{ij} = 1$) であるか、inactive であるかの確率は、スピン状態に依存する重みで決まる

$$Z = \sum_{\{\sigma_i\}} \sum_{\{b_{ij}\}} \prod_{\langle i,j \rangle} \underline{e^K \left[(1-p)\delta_{b_{ij},0} + p\delta_{\sigma_i,\sigma_j}\delta_{b_{ij},1} \right]}$$

$\equiv S_{ij}$



スピン状態を固定した状態で

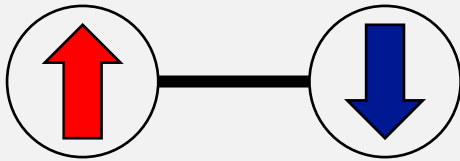
ボンド状態のサンプリングを考える

ボンド変数

スピン変数を固定

$$S_{ij} = e^K \left[(1 - p) \delta_{b_{ij},0} + p \delta_{\sigma_i, \sigma_j} \delta_{b_{ij},1} \right]$$

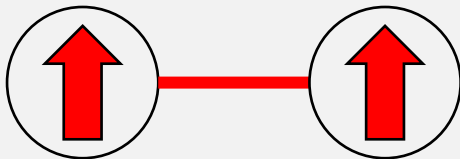
両側のスピンの向きが逆向き ($\sigma_i \neq \sigma_j$) の時



$$S_{ij} = e^K \left[(1 - p) \delta_{b_{ij},0} \right]$$

$b_{ij} = 0$ の場合しか重みを持たない
→ 必ずinactive

両側のスピンの向きが同じ向き ($\sigma_i = \sigma_j$) の時



$$S_{ij} = e^K \left[(1 - p) \delta_{b_{ij},0} + p \delta_{b_{ij},1} \right]$$

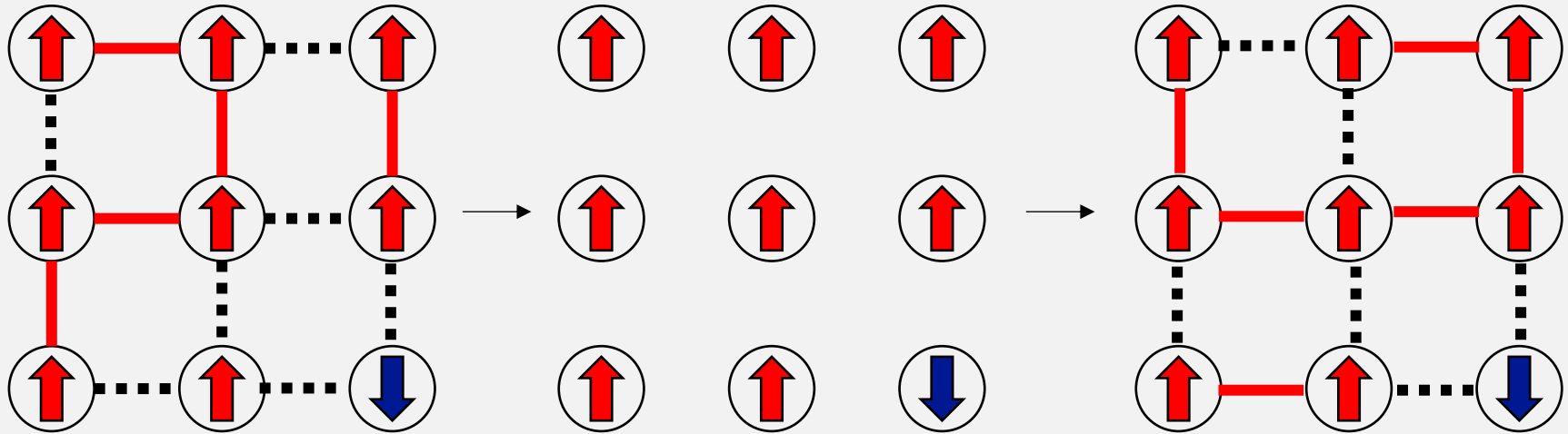
確率 $1 - p$ でinactive、確率 p でactive

ボンド変数

現在の状態

ボンド状態を忘れる

新しいボンド状態を与える



隣り合うスピンの同じ向きの場合

- 確率 p で active
- 確率 $1 - p$ で inactive


隣り合うスピンの逆向きの場合

- 必ず inactive

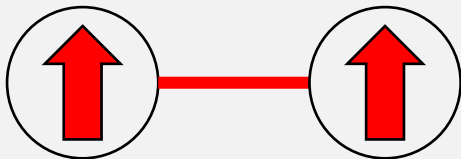
$$p \equiv 1 - e^{-2K}$$

スピン変数

ボンド変数を固定

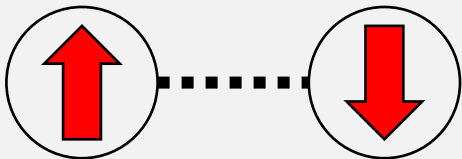

$$S_{ij} = e^K \left[(1 - p) \delta_{b_{ij}, 0} + p \delta_{\sigma_i, \sigma_j} \delta_{b_{ij}, 1} \right]$$

ボンドがactive($b_{ij} = 1$)の時



$$S_{ij} = e^K p \delta_{\sigma_i, \sigma_j}$$

ボンドがinactive($b_{ij} = 0$)の時

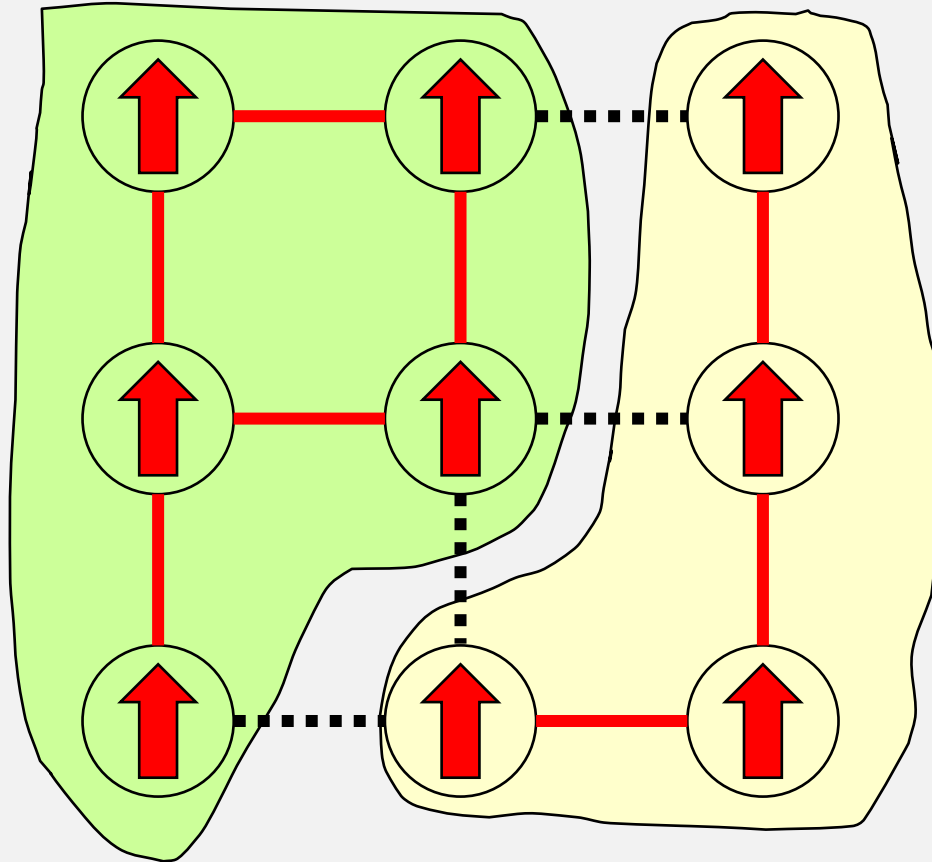


$$\underline{S_{ij} = e^K (1 - p)}$$

スピンの相互作用が含まれない
→ スピンの相互作用が切れている

クラスター

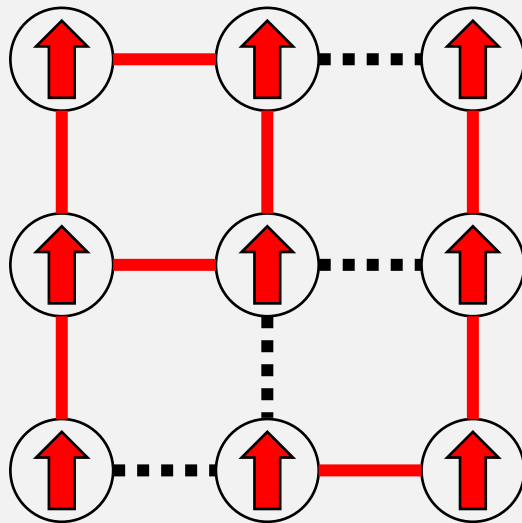
activeなボンドでつながっているスピンをクラスターと呼ぶ



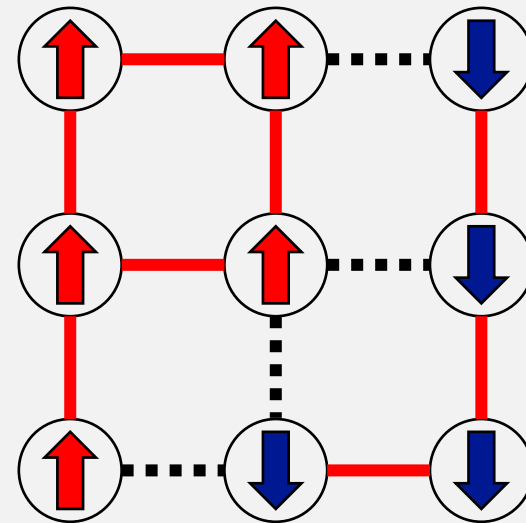
クラスター

inactiveなボンドでつながるスピンは相互作用を持たない
→ activeなボンドでつながるスピンを一斉にひっくり返しても
状態の重みが変わらない

状態A

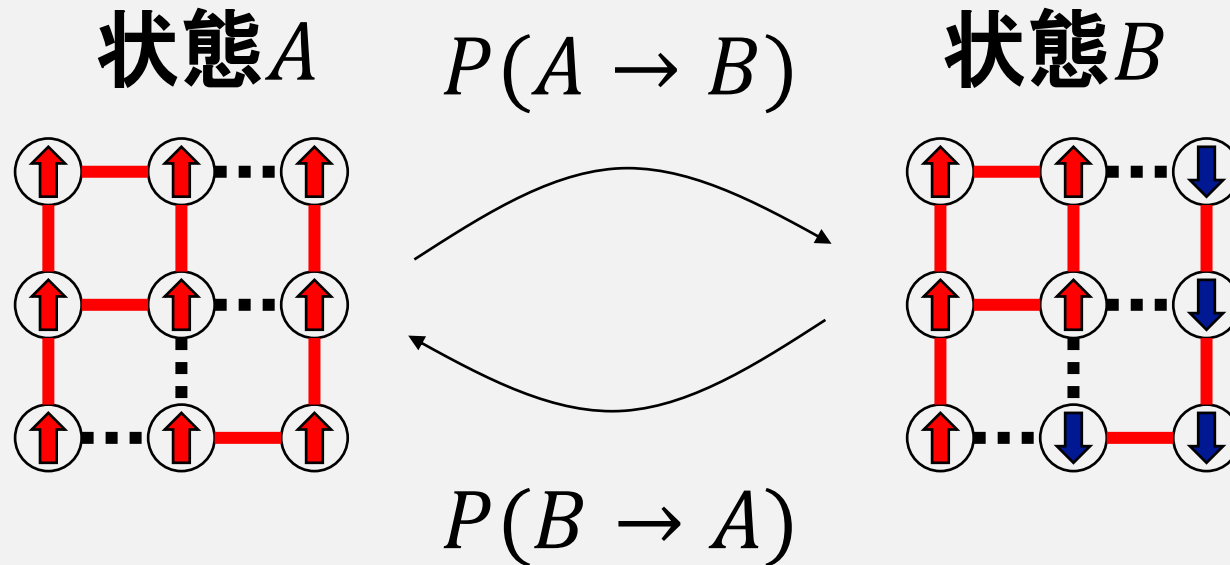


状態B



$$W_A = W_B$$

クラスターアップデート



熱浴法を採用すると

$$P(A \rightarrow B) = \frac{w_B}{w_A + w_B} = \frac{1}{2}$$

$$P(B \rightarrow A) = \frac{w_A}{w_A + w_B} = \frac{1}{2}$$

クラスターアップデート

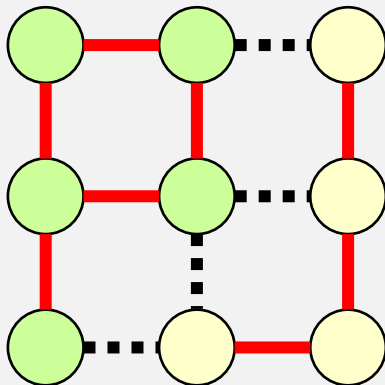
$$Z = \sum_{\{b_{ij}\}} \sum_{\{\sigma_i\}} \prod_{\langle i,j \rangle} e^K \left[(1-p) \delta_{b_{ij},0} + p \delta_{\sigma_i, \sigma_j} \delta_{b_{ij},1} \right]$$

$\{b_{ij}\}$

$\{\sigma_i\}$

ボンド状態を固定した状態で
スピン状態のサンプリングを考える

activeなボンドでつながるスピンを一斉にひっくり返しても重みが等しい
→ 同じクラスターに所属するスピンを全てランダムにひっくり返してよい



クラスターごとに、元のスピン配置を忘れて
ランダムにスピンを割り当ててよい

※同じクラスターに属すスピンは必ず同じ向き

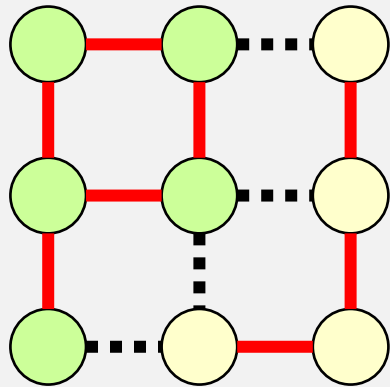
クラスターアルゴリズム

1. スピンを固定してボンド状態を更新する
 - 隣り合うスピンの向きが同じ向きなら確率 p でactive
 - 逆向きならinactive
2. ボンド状態を固定してスピン状態を更新する
 1. activeなボンドでつながったスピンをクラスターに分ける
 2. クラスターごとに新しいスピン状態を与える
3. 1-2のステップを繰り返す

以上の方法をSwendsen-Wangのアルゴリズムと呼ぶ

クラスターアルゴリズム

$$Z = \sum_{\{b_{ij}\}} \sum_{\{\sigma_i\}} \prod_{\langle i,j \rangle} e^K \left[(1-p) \delta_{b_{ij},0} + p \delta_{\sigma_i, \sigma_j} \delta_{b_{ij},1} \right]$$



外側にボンド状態に関する和がある
ボンド状態は、たとえば左図のよう
なグラフを表している
→あらゆるグラフに関する和

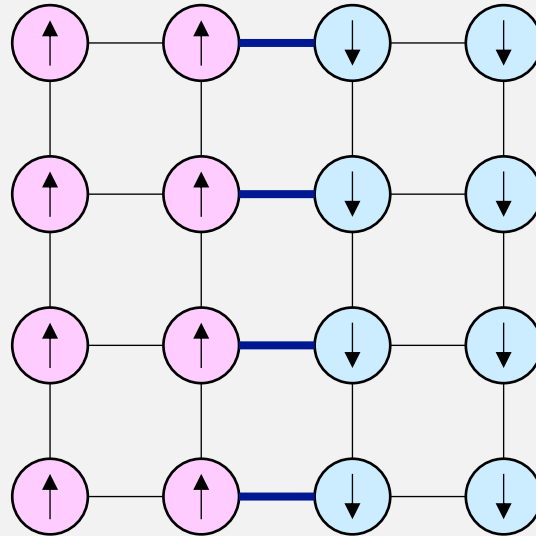
- このグラフ表現をFortuin-Kasteleyn表現と呼ぶ
- グラフ表現を利用してスピンを更新するアルゴリズムをクラスターアルゴリズムと呼ぶ
- Swendsen-Wangはクラスターアルゴリズムの一種

クラスターアルゴリズム

クラスターアルゴリズムは非常に早い

しかし、早い理由はスピンをまとめて更新するからではない

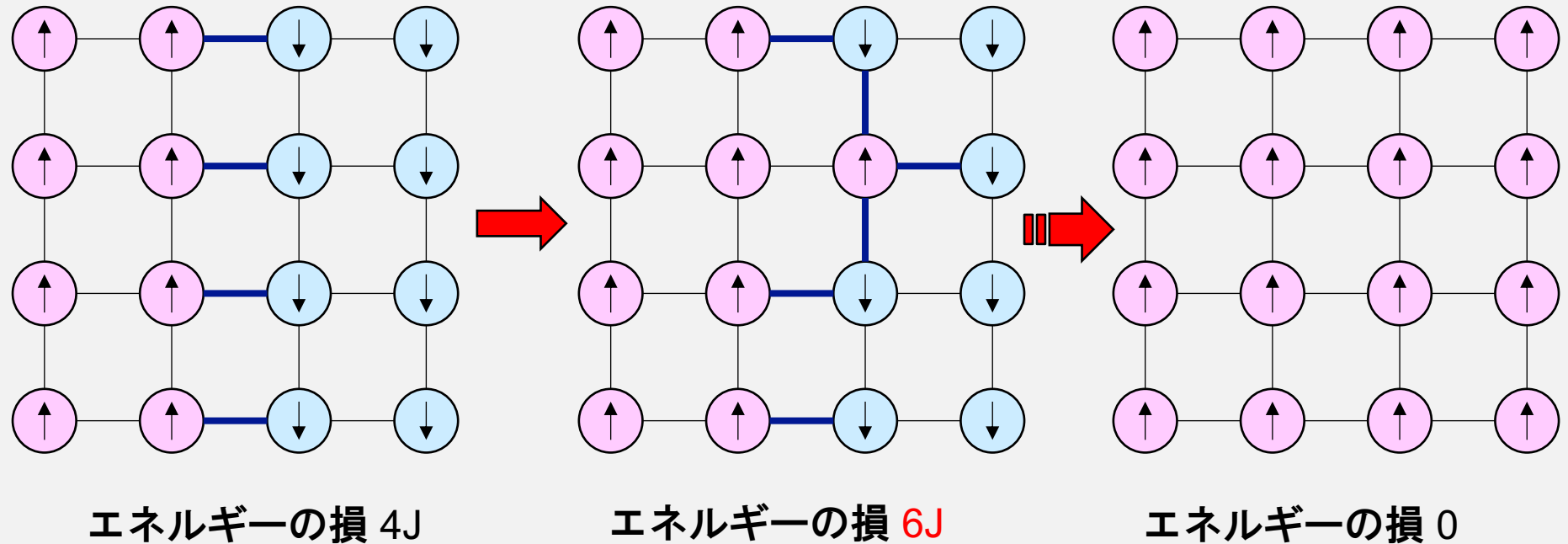
低温である時、以下のようなドメインウォールができたとする



低温だから全てのスピンのそろった状態になりたい

クラスターアルゴリズム

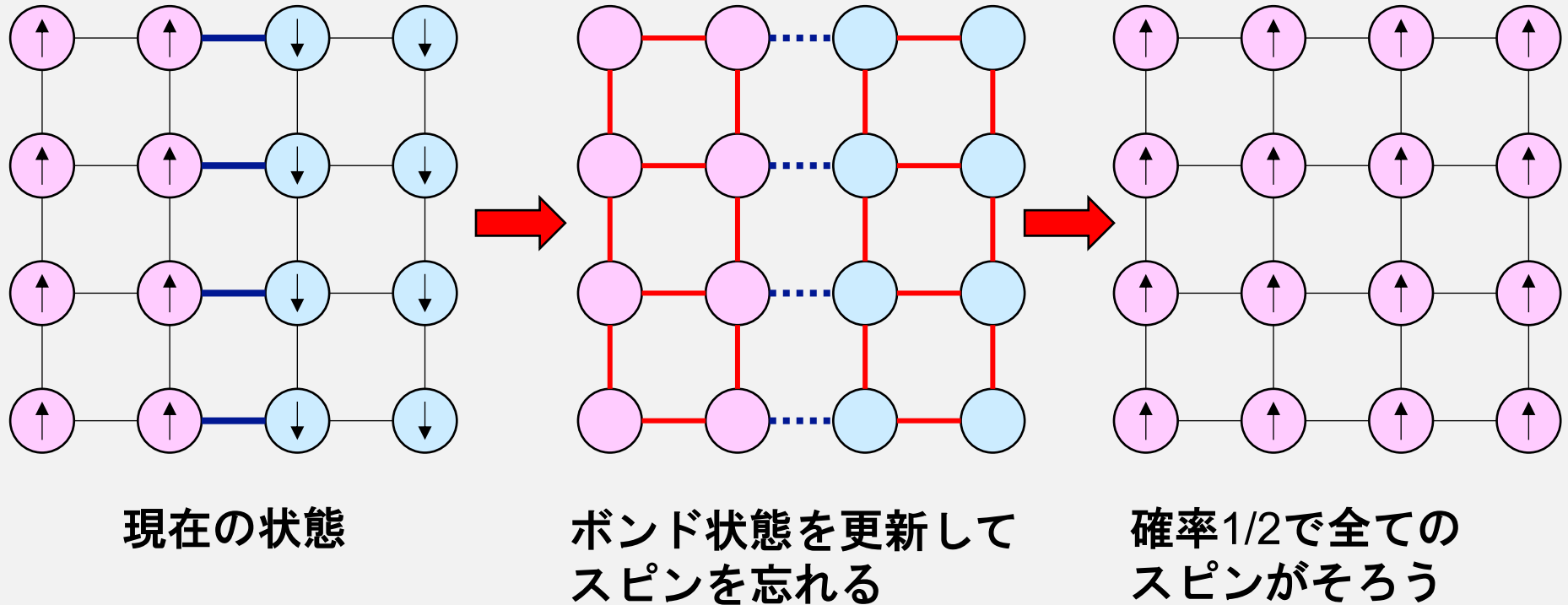
Single-Spin-Flipでドメインウォール解消を試みる



途中で必ずエネルギーが高い状態を経由する必要がある
低温であるほど、高エネルギー状態は出現しづらい
→極めて緩和が遅くなる

クラスターアルゴリズム

Swendsen-Wangの場合



クラスターアルゴリズムでは、ドメインウォールを非常に早く解消できる→緩和が本質的に早い

クラスターアルゴリズムのまとめ

- 分配関数を式変形し、新たにボンドの自由度を追加するとグラフの和の形に書ける(Fortuin-Kasteleyn表現)
- グラフ表現を利用したサンプリングアルゴリズムをクラスターアルゴリズムと呼ぶ
- Inactiveなボンドでつながったスピンは重みがスピン状態に依存しない→実質的に相互作用が切れている
- スピンをお互いに相互作用しないクラスターに分けることができる
- クラスターはそれぞれ独立にスピン状態を与えられる
- ボンド変数とスピンを交互に更新することで、効率的なサンプリングができる
- Swendseng-Wangはクラスターアルゴリズムの一種で、他にもWolffのアルゴリズムなどがある

ここまでのまとめ

- 状態 i の出現確率が w_i に比例する時、 w_i を状態 i の重みと呼ぶ
- 重みの総和 $Z = \sum_i w_i$ を分配関数と呼ぶ
- マルコフ連鎖モンテカルロ法とは分配関数をサンプリングにより推定する手法
- 分配関数を変形すると、異なる物理系のように見える
- 同じ分配関数であるにも関わらず、変形をすることで並列化できたり緩和を加速できたりする

Fortuin-Kasteleyn表現は1972年には知られていた。
しかしSwendsen-Wangアルゴリズムの提案は1987年
まだ見ぬ新たなブレイクスルーがあるかも・・・？

Estimator

確率変数 \hat{X} の母集団のパラメータを知りたい

\hat{X} の取りうる値 Y_k とその実現確率 p_k が既知なら
母集団の期待値の厳密な表記は

$$E[\hat{X}] = \sum_k Y_k p_k$$

一般にはこの和を厳密に計算できない
→サンプリングで推定

Estimator

確率変数 \hat{X} の母集団のパラメータを知りたい

確率変数 \hat{X} を N 回観測して標本 $\{X_i\}$ を得た
ここから \hat{X} の母集団のパラメータを推定したい

母集団の期待値

期待値のestimator

$$E[\hat{X}] \sim \bar{X} = \frac{1}{N} \sum_i X_i$$

知りたい量を標本で表す関数をestimatorと呼ぶ

Estimator

N が大きい時、中心極限定理により、 \bar{X} の分布はガウス分布に近づく

\bar{X} の分散

\hat{X} の分散(母分散)

$$V[\bar{X}] = \frac{1}{N} V[\hat{X}] = \frac{1}{N(N-1)} \sum_i (X_i - \bar{X})^2$$

• \bar{X} の分散のestimator

これが統計誤差だから、なるべく小さくしたい

Improved Estimator

全体では厳密な和を取れないが、状態がいくつかの**グループ**に分かれ、その中の**部分和**は厳密に取れるとする


$$E[\hat{X}] = \sum_k Y_k p_k = \sum_g \sum_{k \in g} Y_k p_k = \sum_g \bar{Y}_g p_g$$

ここは計算できないが

ここは厳密に計算できる

すると、対応するestimatorの分散が小さくなる
これを**improved estimator**と呼ぶ

Improved Estimator

$$E[\hat{X}] = \sum_k Y_k p_k = \sum_g \sum_{k \in g} Y_k p_k = \sum_g \bar{Y}_g p_g$$


確率 p_g で状態 g となる確率変数 \hat{X}_I を考えると
もとの確率変数 \hat{X} よりも分散が小さくなる

$$V[\hat{X}] > V[\hat{X}_I]$$

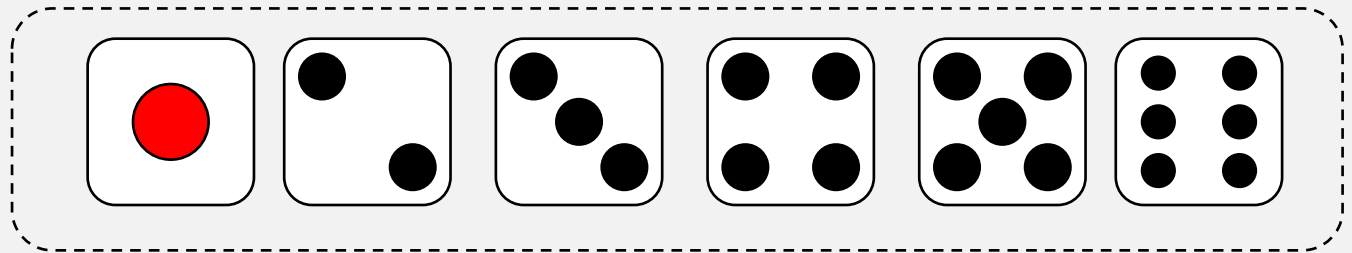
対応するestimatorの分散も小さくなる
これをimproved estimatorと呼ぶ

Improved Estimatorの例



サイコロの場合

母集団



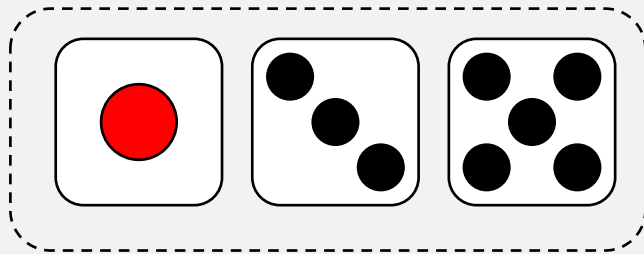
期待値 $E[\hat{X}] = \frac{7}{2}$

分散 $V[\hat{X}] = \frac{35}{12}$

Improved Estimatorの例

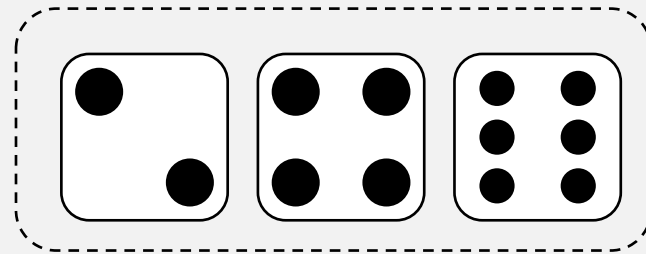
偶数と奇数でグループ分けする

奇数グループ



平均3

偶数グループ



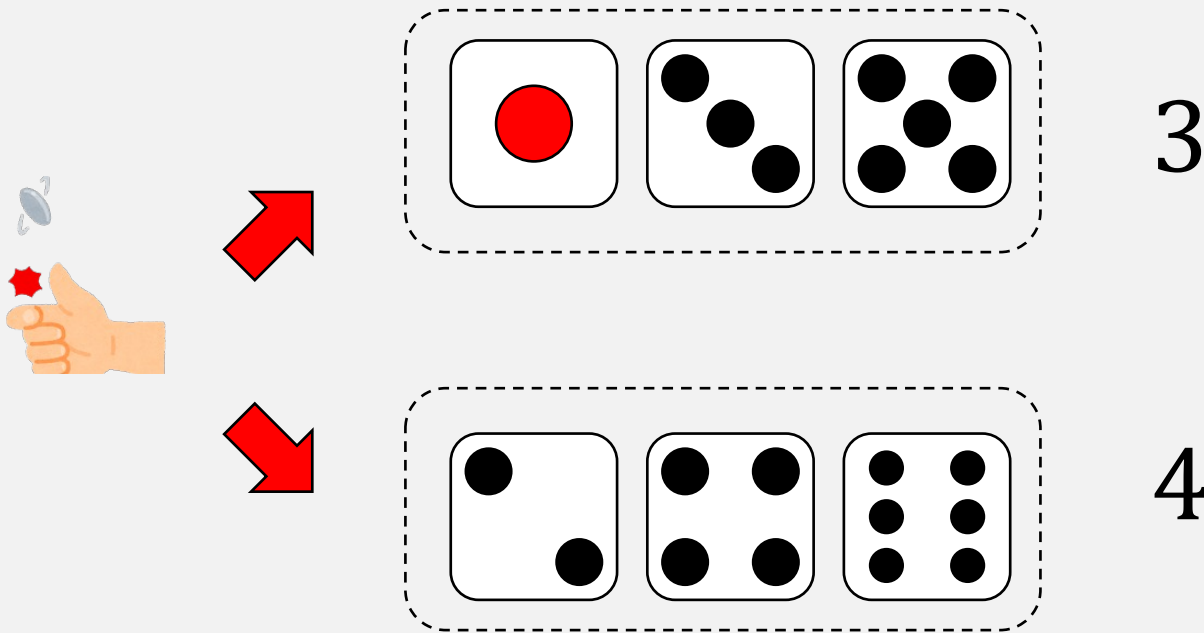
平均4

全体の和は取れないが、グループ内では期待値が
厳密に求まったとする

6つの事象があるサイコロ→2つの事象があるコイン

Improved Estimatorの例

コインを弾いて、表ならサイコロの奇数、裏なら偶数が出たことにする



確率変数 \hat{X}_I は、確率 $1/2$ で3、確率 $1/2$ で4

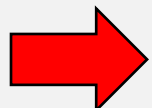
Improved Estimatorの例

確率変数 \hat{X}_I は、確率1/2で3、確率1/2で4

期待値 $E[\hat{X}_I] = \frac{3}{2} + \frac{4}{2} = \frac{7}{2}$

分散 $V[\hat{X}_I] = \frac{1}{4} \times \frac{1}{2} + \frac{1}{4} \times \frac{1}{2} = \frac{1}{4}$

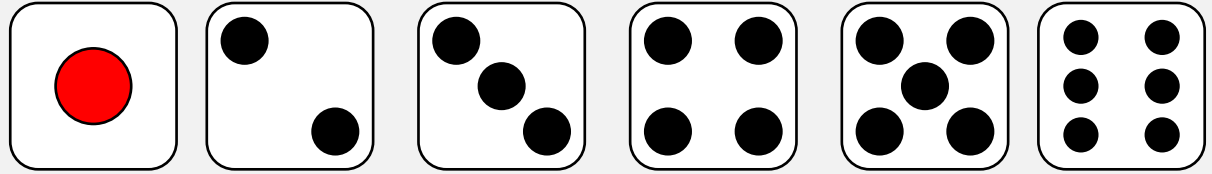
$$V[\hat{X}_I] = \frac{1}{4} < V[\hat{X}] = \frac{35}{12}$$

 分散が小さくなった

Improved Estimatorの例

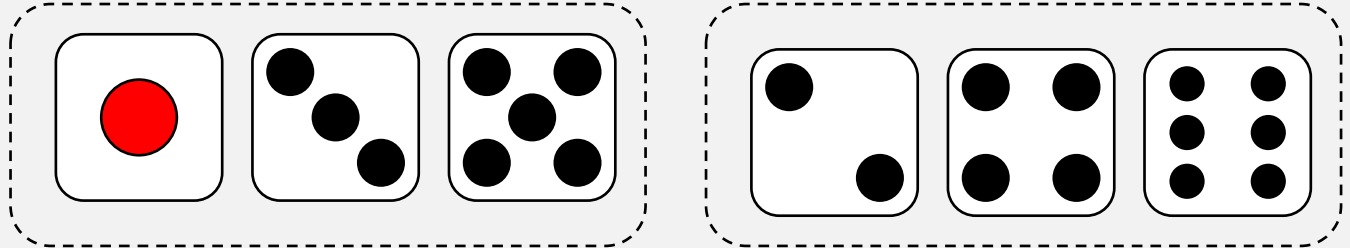
もとの確率変数

$$\{\hat{X}\} =$$



グループ分けして部分和をとった確率変数

$$\{\hat{X}_I\} =$$



部分和により「グループ内の期待値が十分に求められるだけのサンプル数」を得している

→実質的にサンプル数が増えている→分散が減る

Improved Estimatorの例

- 全体では厳密な和を取れないが
- 状態がいくつかのグループに分かれ
- その中の部分和は厳密に取れるなら
- グループに関するサンプリングにより
- エラーバーが小さくなる

そんな都合の良い話があるか？

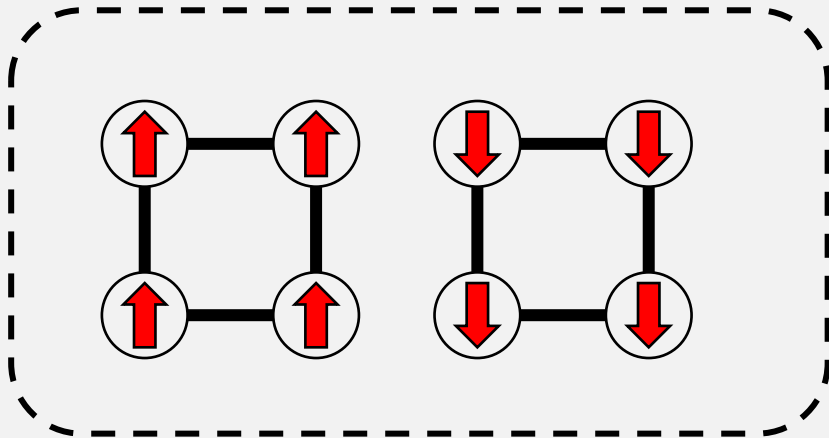
→ グラフ表現による Improved Estimator

イジング模型

自発磁化 $m = \frac{1}{N} \sum_i \sigma_i$

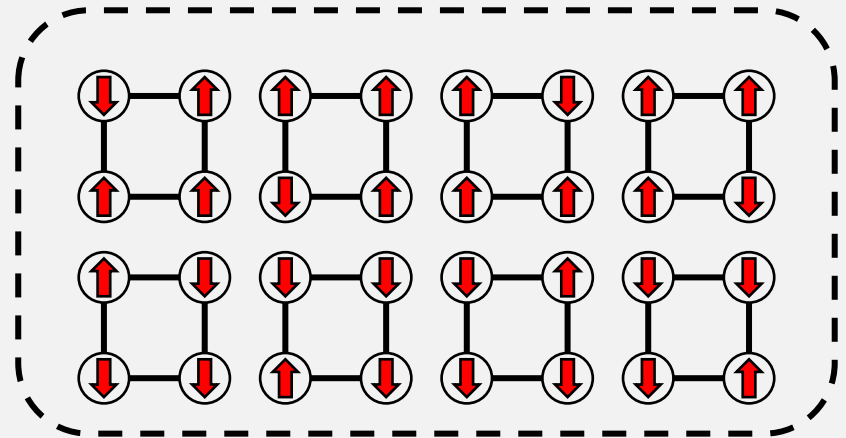
N はスピンの数

低温



$$m^2 > 0$$

高温



$$m^2 \sim 0$$

自発磁化の二乗 m^2 はこの系の秩序変数となる

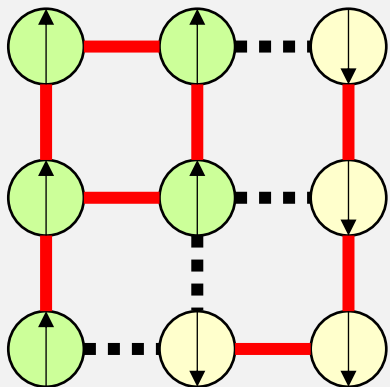
イジング模型の例

自発磁化の期待値の表式

$$\langle m^2 \rangle = Z^{-1} \sum_{\{\sigma_i\}} \left(\frac{1}{N} \sum_i \sigma_i \right)^2 w(\{\sigma_i\})$$

このグラフ表現を考える

イジング模型の例



クラスターを「グループ」だと思おう
クラスターに通し番号 k をつける
クラスター内のスピンは全て同じ向き
→ $\tilde{\sigma}_k$ とする
クラスター内のスピン数を n_k とする

$$m = \frac{1}{N} \sum_i \sigma_i = \frac{1}{N} \sum_k n_k \tilde{\sigma}_k$$



スピン個別の和を

クラスターごとの和にまとめる

イジング模型の例

$$m = \frac{1}{N} \sum_i \sigma_i = \frac{1}{N} \sum_k n_k \tilde{\sigma}_k \quad \text{を} \langle m^2 \rangle \text{の式へ代入する}$$

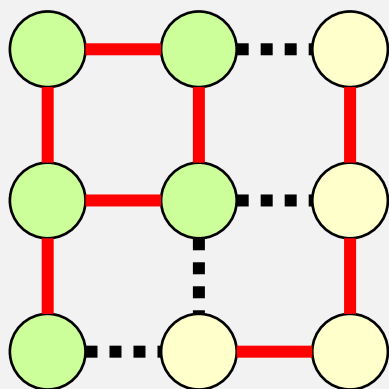
$$\langle m^2 \rangle = \frac{Z^{-1}}{N^2} \sum_{\{g\}} w(g) \sum_{\{\tilde{\sigma}_k\}} \left(\sum_k n_k \tilde{\sigma}_k \right)^2$$

この部分和が厳密に計算できる

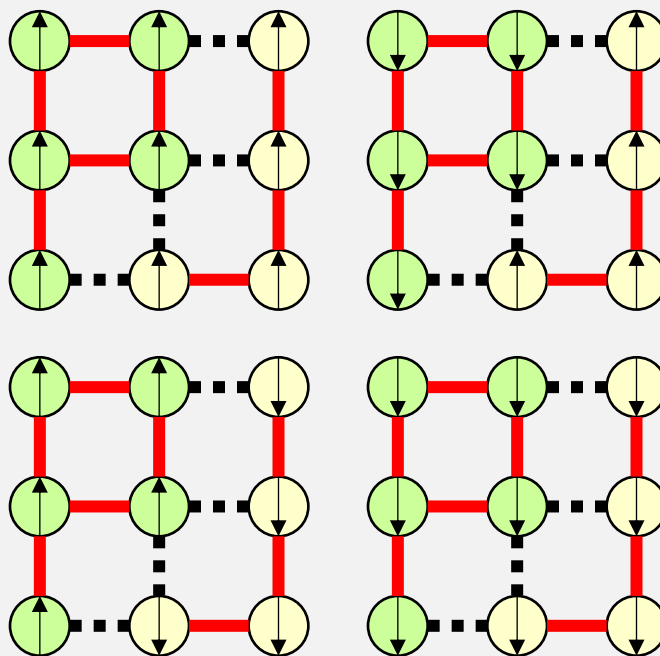
イジング模型の例

$$\langle m^2 \rangle = \frac{Z^{-1}}{N^2} \sum_{\{g\}} w(g) \sum_{\{\tilde{\sigma}_k\}} \left(\sum_k n_k \tilde{\sigma}_k \right)^2$$

ここでボンド状況を固定



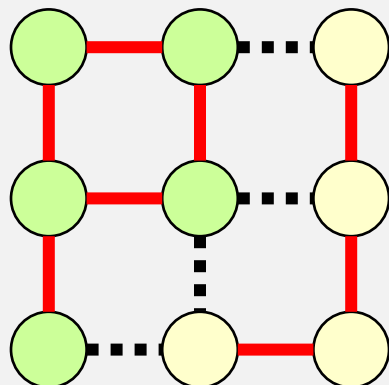
各クラスターのスピン状態は
全て同じ重み



イジング模型の例

クラスター1

$$n_1 = 5$$



クラスター2

$$n_2 = 4$$

$$\tilde{\sigma}_k = \pm 1 \text{ だから } \tilde{\sigma}_k^2 = 1$$

$$\sum_{\{\tilde{\sigma}_k\}} \left(\sum_k n_k \tilde{\sigma}_k \right)^2 = \sum_{\{\tilde{\sigma}_k\}} (n_1^2 \tilde{\sigma}_1^2 + n_1 n_2 \tilde{\sigma}_1 \tilde{\sigma}_2 + n_2^2 \tilde{\sigma}_2^2)$$

$$= \sum_k n_k^2$$

$$\tilde{\sigma}_k = \pm 1 \text{ だから } 0$$

グラフを固定した場合のスピン状態の部分和が厳密に計算できた

イジング模型の例

グラフ表現した物理量

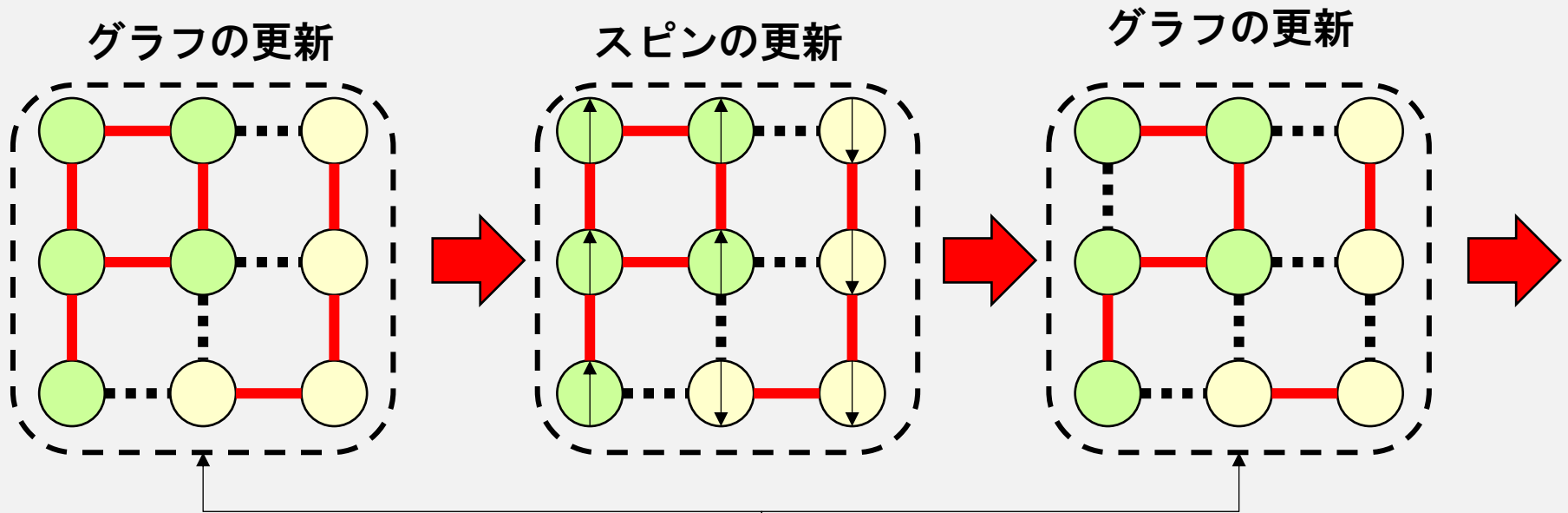
$$\langle m^2 \rangle = \frac{Z^{-1}}{N^2} \sum_{\{g\}} w(g) \sum_{\{\tilde{\sigma}_k\}} \left(\sum_k n_k \tilde{\sigma}_k \right)^2$$

部分和をとった物理量

$$\langle m^2 \rangle = \frac{Z^{-1}}{N^2} \sum_{\{g\}} w(g) \sum_k n_k^2$$

系をグラフ表現した時、クラスターサイズの二乗の平均が自発磁化の二乗になる

アルゴリズム



これらのグラフは重み $w(g)$ で出現する

グラフ更新のタイミングで以下の量を計算すると、これが improved estimator になっている

$$m^2 = \frac{1}{N^2} \sum_k n_k^2$$

数値計算例

スピン系で秩序変数としてよく用いられるBinder比を計算する

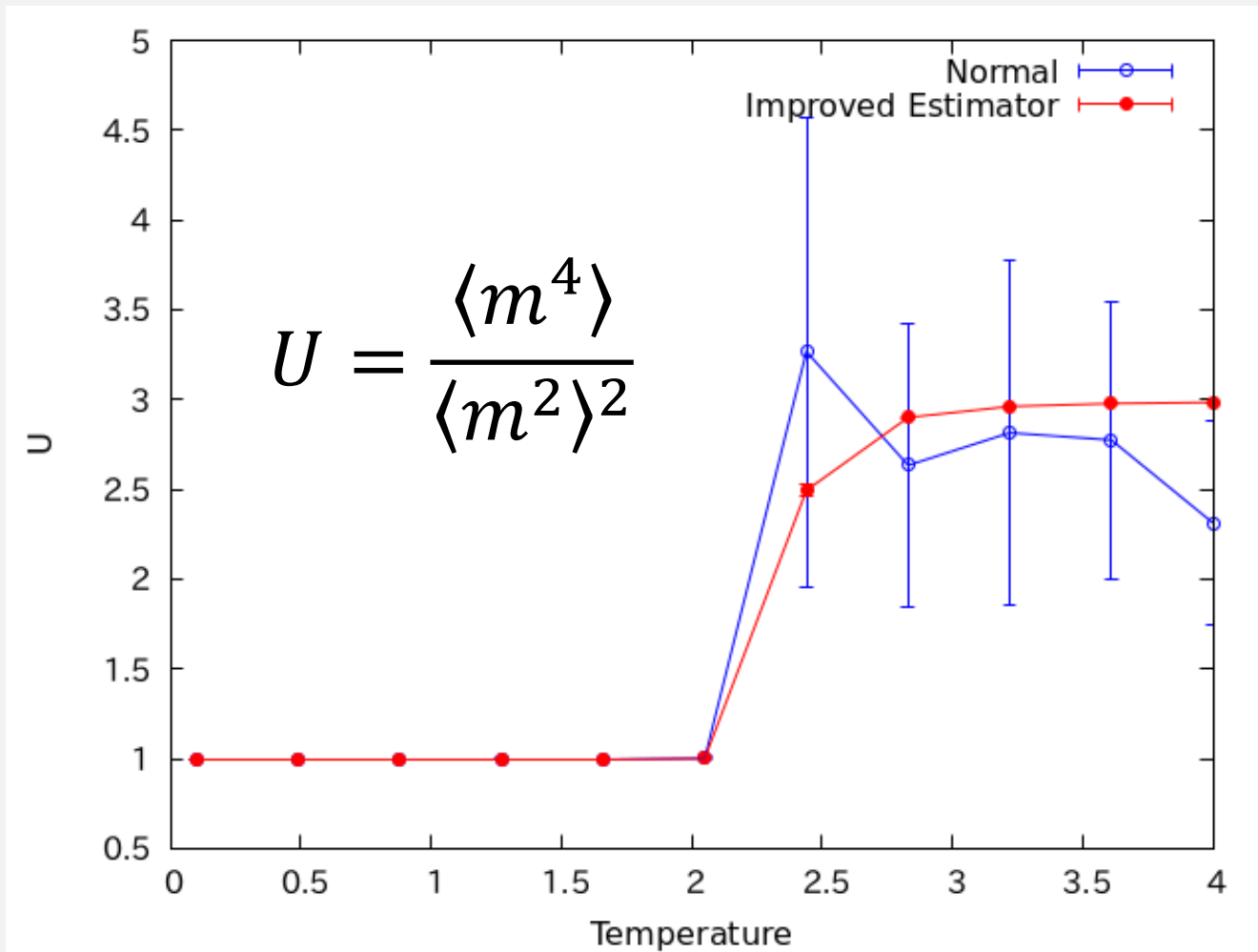
$$U = \frac{\langle m^4 \rangle}{\langle m^2 \rangle^2} \quad \text{高温で1、低温で3となる}$$

それぞれのモーメントのimproved estimator

$$m^2 = \frac{1}{N^2} \sum_k n_k^2$$

$$m^4 = \frac{1}{N^4} \left(\sum_k n_k^2 + 6 \sum_{k < k'} n_k^2 n_{k'}^2 \right)$$

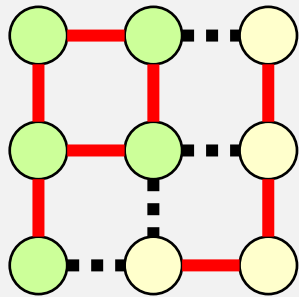
数値計算例



低温ではどちらも高精度だが、高温で通常のestimatorの精度が悪くなる

なぜ精度が改善するか？

グラフ状態
クラスター数 N_c

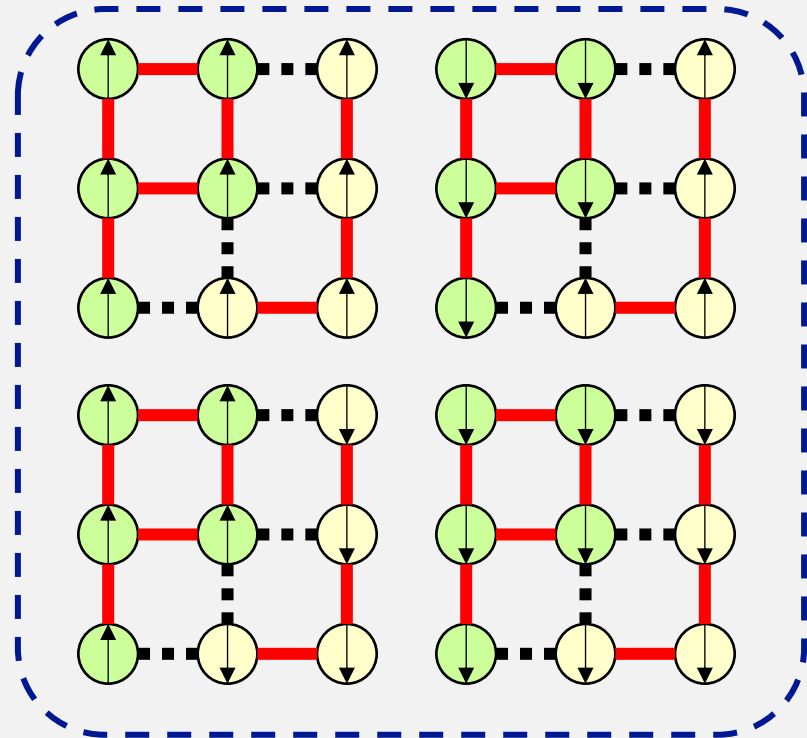


ボンドがつながる確率

$$p = 1 - e^{-2K}$$

高温($K \rightarrow 0$)で $p \rightarrow 0$

対応するスピンの状態数 $= 2^{N_c}$



高温ではほとんどのボンドが切れている

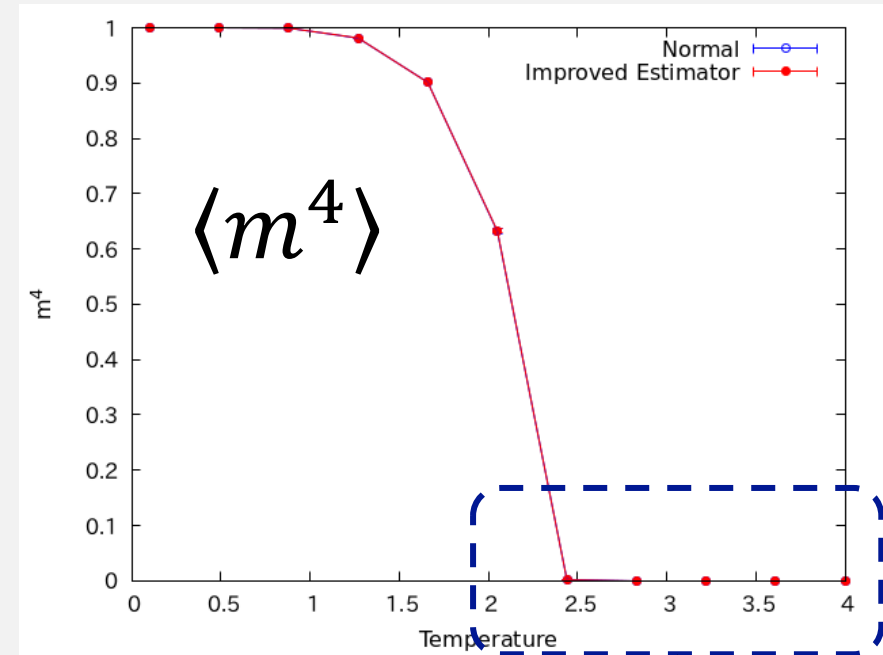
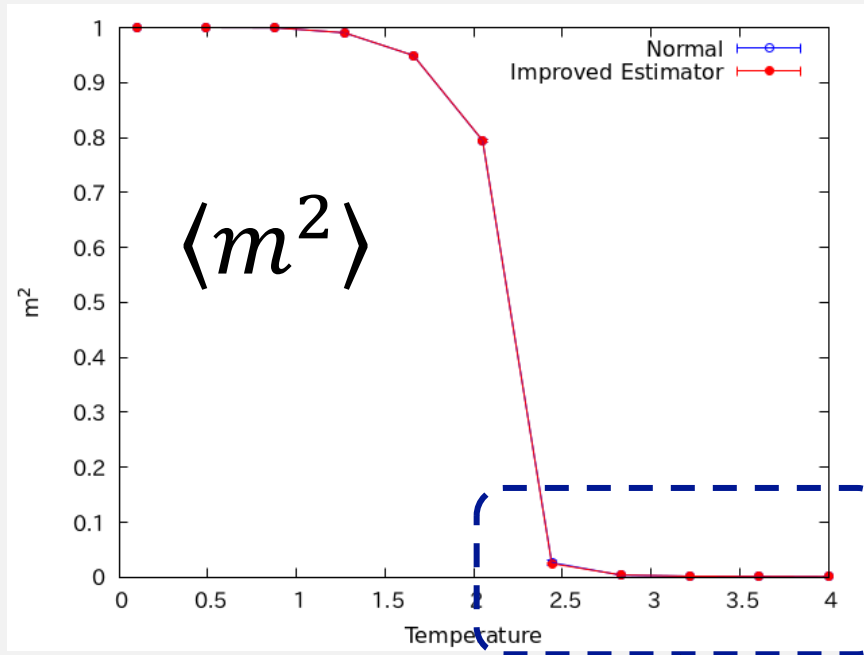
→ 多数の小さいクラスターができる

→ 部分和が厳密に取れるスピンの状態数が増える

→ 精度がよくなる

なぜ精度が改善するか？

2次と4次のモーメントそれぞれの推定の精度はどちらも問題ない



$$U = \frac{\langle m^4 \rangle}{\langle m^2 \rangle^2}$$

しかし、高温で微小な量の比を計算する必要があり、精度が極端に落ちる

Improved Estimatorのまとめ

- 全体の和は計算できないが、状態をグループにわけた時に、グループ内の和は厳密に取れる場合がある
- 部分和を厳密に取った場合に対応するestimatorを **improved estimator** と呼ぶ
- Improved estimatorの分散は必ず小さくなる

本日のまとめ

- 状態の重みの総和を分配関数とよぶ
- 分配関数を書き直す＝見方を変える
- 見方を変えると、効率がよくなる場合がある
- 同じものを異なる数式で表現すると、異なる世界が見えてくる
- 何十年も前の発見を活用してブレイクスルーが生まれることがある

※ Fortuin-Kasteleyn表現が1972年、Swendsen-Wangが1987年