

焼きなまし法と増加再標本焼きなまし法の統一視点

今村安伸^{†1} 樋口直哉^{†2} 篠原武^{†2} 平田耕一^{†2} 久保山哲二^{†3}

概要: 増加再標本焼きなまし法 (Annealing by Increasing Resampling, AIR) は、少しずつ再標本サイズを増やしながらか山登りを行う確率的な最適化法である。本稿では、従来の焼きなまし法 (SA) と AIR の統一的な視点を導入する。この視点では、SA と AIR の両方を確率的な変動を伴う目的関数の確率的な山登りと見ることができる。生じる確率の変動は、SA では logit, AIR では probit である。Logit と probit は近似の関係にあるため、AIR は SA の近似とみなすことができる。次元縮小射影のピボット探索や焼きなましによるクラスタリングなどの最適化問題に対する実験結果もまた、AIR が SA の近似であることを支持している。さらに、目的関数が多数の標本を必要とする場合、AIR は結果の品質を犠牲にすることなく、SA よりもはるかに高速に動作することも確認できる。

キーワード: 焼きなまし法, 増加再標本焼きなまし法, 最適化, logit, probit, メタヒューリスティクス

Annealing by Increasing Resampling in the Unified View of Simulated Annealing

YASUNOBU IMAMURA^{†1} NAOYA HIGUCHI^{†2} TAKESHI SHINOHARA^{†2}
KOUICHI HIRATA^{†2} TETSUJI KUBOYAMA^{†3}

Abstract: Annealing by Increasing Resampling (AIR) is a stochastic hill-climbing optimization by resampling with increasing size for evaluating an objective function. In this paper, we introduce a unified view of the conventional Simulated Annealing (SA) and AIR. In this view, we generalize both SA and AIR to a stochastic hill-climbing for objective functions with stochastic fluctuations, i.e., logit and probit, respectively. Since the logit function is approximated by the probit function, we show that AIR is regarded as an approximation of SA. The experimental results on sparse pivot selection and annealing-based clustering also support that AIR is an approximation of SA. Moreover, we show that when an objective function requires a large number of samples, AIR is much faster than SA without sacrificing the quality of the results.

Keywords: simulated annealing, annealing by increasing resampling, optimization, logit, probit, meta-heuristics

1. はじめに

類似検索は、高次元データからの情報検索のための重要な有名タスクの一つである。この一つの解決法として、Simple-Map[14]や Sketch[7]の様な次元縮小射影を用いて高次元データを低次元へ射影することで、効率的な索引付け及び高速な検索を実現することが可能となる。低次元への射影は対象データに合った少ない軸の選択[4]を意味し、このとき、組合せ最適化問題が発生する。

焼きなまし法[10] (Simulated Annealing, SA) は、与えられた目的関数の大域最適値を探索するための確率的な手法である。焼きなまし法は温度のパラメータを用いる。最初に、焼きなまし法は高温で始まる。このとき、探索空間内のとても広い範囲を探索対象としている。つづいて、温度をゆっくりと下げることによって、探索範囲を徐々に狭めていく。最終的には、局所探索と同様にふるまう。

増加再標本焼きなまし法 (Annealing by Increasing Resampling, AIR) では、評価に標本集合を用いる最適化問題で、目的関数は個々の標本に対する評価値を平均などで総合するものに適用可能である。探索の開始時点では、小さな再標本を用いて評価を行い、探索が進むにつれて評価に用いる再標本を徐々に大きくして行く。

小さな再標本による評価は、標本全体を用いるものに比べると大きな誤差を発生する確率が高く、探索の初期段階では SA と同様にランダムな遷移を行う。大きな再標本での評価は誤差が小さくなるので、最終段階では、AIR は SA と同様に局所探索を行う。つまり、AIR は SA のような挙動を実現している。また、AIR は、目的関数に用いる集合サイズが大きな場合に特に顕著な速度面の優位性を有する。

AIR が SA の効率的な実現法となっていることについては、実験的には、すでに報告済みである[9]。本論文では、SA と AIR は、いずれも確率的揺らぎをもった目的関数を用いた山登りであるという統一的な視点でとらえることができることを理論的に示す。ベーカー関数[3]を用いる SA の評価の揺らぎは logit で説明でき、AIR のそれは probit で説明できる。Logit は probit で近似できるので、AIR は SA の近似として説明できる。

^{†1} システム工房コロン
System Studio Colun

^{†2} 九州工業大学
Kyushu Institute of Technology

^{†3} 学習院大学
Gakushuin University

また、実験的には、SA は大域的最適解を求めるためには高温時に多くの計算が必要であることが知られている。これに対し、AIR では高温時には再標本サイズを小さくするため、結果の品質を損なうことなく、効率よく大域的最適解を求めることができる。このように評価用の標本数が大きいときにはAIRがSAより速度的に非常に有利であることがわかる。

2. SA と AIR の統一視点

ここで、焼きなまし法 SA[10]と増加再標本焼きなまし法 AIR[9]に共通の視点を与えておく。用いる記号や表記法を表1に示す。ここでは、目的(エネルギー)関数 $E: U \times S \rightarrow \mathbb{R}$ の最小化問題を扱うことにする。ただし、 U は解空間、すなわち可能な解すべての集合、 S は標本データ集合である。最適化の目標は、 E を大域的に最小化する x^* 、つまり、すべての $x \in U$ に対して、 $E(x^*, S) \leq E(x, S)$ となる x^* を求めることである。また、評価に S 全体を用いる場合は、略記法として $E(x)$ を用いる、つまり、 $E(x) = E(x, S)$ である。

| 記法 | 意味 |
|-----------------------|---|
| $t \in \mathbb{N}$ | 時刻 (0, 1, 2, ...) |
| $T(t) \geq 0$ | 時刻 t における温度 (単調減少) |
| S | 目的関数 E の評価用のデータ集合 |
| $s(t) \in \mathbb{N}$ | 時刻 t における再標本サイズ ($\leq S $) |
| U | 解空間 |
| $x, x' \in U$ | 解空間の要素 |
| $N(x) \subseteq U$ | $x \in U$ の近傍 |
| $E(x, S')$ | $x \in U$ のデータ集合 $S' \subseteq S$ に対する評価値 |
| ω | 一様乱数 ($0 \leq \omega < 1$) |
| $P(\Delta E, T(t))$ | 評価値差 ΔE 温度 $T(t)$ における受理確率 |

表1 記号と記法

2.1 焼きなまし法

SA において、つぎの状態が悪くなっても受け入れることを許す確率を受理確率[2] (*acceptance probability*) あるいは受理基準[13] (*acceptance criterion*) という。アルゴリズム1にSAの概要を示す。2状態 x と x' の標本 S に対する評価値の差を ΔE とする。

$$\Delta E = E(x', S) - E(x, S)$$

SA では、2種類の受理確率がよく用いられる。一つは、メトロポリス (Metropolis[12]) 関数 P_M で、最初に SA[10] で用いられたもので、標準的に用いられている。

$$P_M(\Delta E, T) = \min\{1, \exp(-\Delta E/T)\}$$

もう一つは、ベーカー (Baker[3]) 関数 P_B (または、heat bath 関数[2]) であり、ボルツマンマシン[1]の文脈で導入されたヘイスティングス (Heistings[8]) 関数の特別な場合である。

procedure SA

```

x ← 任意の初期状態;
for t = 1 to ∞ do
    x' ← N(x) から選んだ任意の状態;
    ΔE ← E(x') - E(x);
    ω ← rand(0, 1);
    if ω ≤ P(ΔE, T(t)) then x ← x';
    
```

アルゴリズム 1 焼きなまし法 (SA)

procedure AIR

```

x ← 任意の初期状態;
for t = 1 to ∞ do
    x' ← N(x) から選んだ任意の状態;
    S' ← Sからランダムに選んだ標本,
    ただし, |S'| = s(t) とする;
    if E(x', S') - E(x, S') ≤ 0 then x ← x';
    
```

アルゴリズム 2 増加再標本焼きなまし法 (AIR)

$$P_B(\Delta E, T) = \frac{1}{1 + \exp(\Delta E/T)}$$

さて、状態 x に続いて $x' \in N(x)$ が選ばれる条件を考えよう。メトロポリス関数に対しては、 $\omega \leq \exp(-\Delta E/T)$ であるので、つぎの条件が得られる。

$$\Delta E + T \cdot \log(\omega) \leq 0$$

ベーカー関数に対しては、

$$\exp(\Delta E/T) \leq \frac{1 - \omega}{\omega}$$

となるので、つぎの条件が得られる。

$$\Delta E + T \cdot \text{logit}(\omega) \leq 0 \tag{1}$$

ここで、 logit はつぎで定義されるものである。

$$\text{logit}(\omega) = -\log\left(\frac{1 - \omega}{\omega}\right)$$

最小化においては、 ΔE が0以下であれば、より良い状態 x' に遷移する。受理条件 (1) の左辺は、温度Tに比例する外乱 (確率的揺らぎ) を伴った ΔE と見ることができる。

2.2 増加再標本焼きなまし法

AIR においては、目的関数は、データ集合 S から選んだ標本集合 S' を用いるものとし、個々の評価値の平均を最小化する問題を扱う。 S' のサイズが小さいほど評価値は大きく揺らぐという性質をもつ。AIRはこのことを利用した探索手法である。アルゴリズム2にAIRの概要を示す。

$N = |S|$ とし、2状態 x と x' の標本 S' に対する評価値の差 $E(x', S') - E(x, S')$ は標準偏差 σ の正規分布に従うと仮定する。目的関数は、独立な標本集合に対する評価値の平均で求められるので、この仮定は中心極限定理から合

理的である。

そうすると、標本数 $n = |S'|$ の標本集合 S' に対する x と x' 評価値の差は、標準誤差を持った正規分布に従う。言い換えると、 $E(x', S') - E(x, S')$ の値は、 ΔE に標準偏差 $\frac{\sigma}{\sqrt{n}} \cdot \sqrt{\frac{N-n}{N-1}}$ の揺らぎを持ったものである。よって、 ω を 0 から 1 の範囲の一様乱数とすると、AIR における受理条件は、つぎで与えられる。

$$\Delta E + \frac{\sigma}{\sqrt{n}} \cdot \sqrt{\frac{N-n}{N-1}} \cdot \text{probit}(\omega) \leq 0 \quad (2)$$

ここで、 probit は標準正規分布の累積確率密度関数の逆関数である。 ω を 0 から 1 の範囲の一様乱数とすると、 $\text{probit}(\omega)$ は正規分布に従うことに注意されたい。

AIR では再標本によって標本 S の部分標本 S' を選択している。次の反復時には S' とは独立に部分標本を選ぶ必要がある。よく似た方法として、部分標本 S' を保持して、差分のみの少数の標本を S' 付加して用いることが考えられる。もとの S' に対する評価のための計算の途中結果を再利用できるので、評価を高速化するメリットがある。しかし、この方法は AIR では用いてはならない。各時点での次状態の選択に関する確率的試行は独立になるようにしなければならない。このように本来、AIR では部分標本を試行毎に独立に選ぶ必要があるが、処理の効率化のために、現在の部分標本を再利用してもよいが、時々には独立な選択により部分標本を取り換えなくてはならない。

2.3 焼きなまし法の一般化

次状態候補の受理条件は、ベーカー関数に基づいた SA では式 (1) であり、AIR では式 (2) である。

$$\Delta E + T \cdot \text{logit}(\omega) \leq 0 \quad (1)$$

$$\Delta E + \frac{\sigma}{\sqrt{n}} \cdot \sqrt{\frac{N-n}{N-1}} \cdot \text{probit}(\omega) \leq 0 \quad (2)$$

どちらも、新の評価値の差 ΔE と確率的揺らぎをもった項との和である。それらの揺らぎは、それぞれ、 logit と probit で与えられる。正規分布はロジスティック分布で近似でき

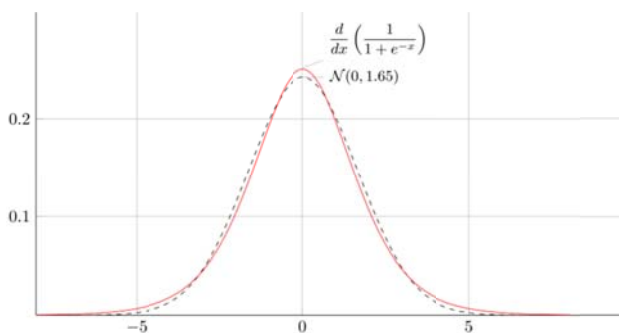


図 1 logit と probit

procedure Unified_Annealing

```

x ← 任意の初期状態;
for t = 1 to ∞ do
    x' ← N(x) から選んだ任意の状態;
    ω ← rand(0, 1);
    if E(x') - E(x) + α(t) · Φ-1(ω) ≤ 0 then
        x ← x';
    
```

アルゴリズム 3 SA と AIR の共通視点

ることが知られている。つまり、 $\sigma_0 = 1.65$ のとき、図 1 に示すように

$$\text{logit}(\omega) \approx \sigma_0 \cdot \text{probit}(\omega)$$

である[5]。

したがって、SA と AIR の次状態候補の受理条件を一般化して、

$$\Delta E + \alpha(t) \cdot \Phi^{-1}(\omega) \leq 0 \quad (3)$$

とすることができる。ここで、 α は時刻 t に関して単調減少であり (AIR では、再標本サイズ n は、関数 s で与えられることに注意)、 Φ^{-1} は、確率分布の累積密度関数の逆関数である。アルゴリズム 3 は、SA と AIR に対する統一的視点を与える一般化アルゴリズムである。

2.4 焼きなましスケジュール

温度を下げていくスケジュールは、SA にとって、効率と精度の両面から重要である。ここで、SA でよく用いられている指数冷却スケジュールに対応する AIR の再標本サイズのスケジュールの関係について述べておこう。温度スケジュールは、つぎで与えられるとする。

$$T = T_0 \cdot T_r \quad (0 < T_r < 1)$$

ここで、 T および T_0 、 T_r は、それぞれ、現在の温度、初期温度、温度比である。 T_r は、時刻に対して単調増加であることに注意されたい。 N および n_0 、 n を、それぞれ、最大標本数、初期標本数、現在の標本数とする。また、 σ_0 をおおよそ 1.65 とし、 σ を個々の標本の評価値の標準偏差とする。

次状態候補の受理条件は、ベーカー関数に基づいた SA では式 (1) であり、AIR では式 (2) である。どちらも、新の評価値の差 ΔE と確率的揺らぎをもった項との和である。それらの揺らぎが等しくなるようにするには、

$$T \cdot \sigma_0 = \frac{\sigma}{\sqrt{n}} \cdot \sqrt{\frac{N-n}{N-1}}$$

とすればよいことがわかる。これより、

$$n = \frac{N}{(N-1)T_0^2 T_r^2 \frac{\sigma_0^2}{\sigma^2} + 1}$$

さらに、 $T = T_0$ のとき、 $T_r = 1$ 、 $n = n_0$ であるので、

$$T_0^2 \frac{\sigma_0^2}{\sigma^2} = \frac{N - n_0}{(N - 1)n_0}$$

よって、

$$s(t) = n = \frac{N}{\frac{N - n_0}{n_0} T_r^2 + 1} \quad (4)$$

時刻 t における再標本サイズ n は、 t の関数によって与えられる。ここで、 $s(0) = n_0$ であることに注意しよう。式 (4) によって、SA における $T = T_0 \cdot T_r$ の冷却スケジュールと AIR における再標本サイズのスケジュールの橋渡しができる。SA と AIR で同じ温度比 T_r を用いるようにすれば、公平な比較実験を行うことができる。

3. 実験結果

3.1 MCMC における logit と probit

AIR がどの程度、焼きなまし法の近似になっているのかについて、焼きなまし法の基盤となっている MCMC (メトロポリス・ヘイスティングス法) を用いた分布の推定実験を行った。離散的に求めた実際の分布との相関係数を精度とし、その補確率を誤差とした。

対象関数としては、二つの正規分布を混合したつぎの 1 次元関数を用いた。

$$y = \frac{0.3e^{-(x-1)^2} + 0.7e^{-(x+2)^2}}{\sqrt{\pi}}$$

受理条件としては、以下の 6 種類を比較した。

i. メトロポリス法:

$$\omega \leq \min\{1, \exp(-\Delta E/T)\}$$

ii. ベーカー (ヘイスティングス) 法:

$$\omega \leq \frac{1}{1 + \exp(\Delta E/T)}$$

iii. $\Phi^{-1}(\omega) = \log(\omega)$:

$$\Delta E + T \cdot \log(\omega) \leq 0$$

iv. $\Phi^{-1}(\omega) = \text{logit}(\omega)$:

$$\Delta E + T \cdot \text{logit}(\omega) \leq 0$$

v. $\Phi^{-1}(\omega) = 1.60 \cdot \text{probit}(\omega)$:

$$\Delta E + T \cdot 1.60 \cdot \text{probit}(\omega) \leq 0$$

vi. $\Phi^{-1}(\omega) = 1.65 \cdot \text{probit}(\omega)$:

$$\Delta E + T \cdot 1.65 \cdot \text{probit}(\omega) \leq 0$$

状態遷移回数を 10^5 から 10^8 まで変化させて、MCMC による誤差比較を行った結果を図 2 に示す。メトロポリス法 (i) と \log (iii) およびベーカー法 (ii) と logit (iv) については、第 2 章で示したように数学的に等しく、ここで見られる様な差異は有意な差ではない。遷移回数 10^5 や 10^6 での実験では手法による誤差にはほとんど差が見られなかったが、 10^7 や 10^8 での実験では有意

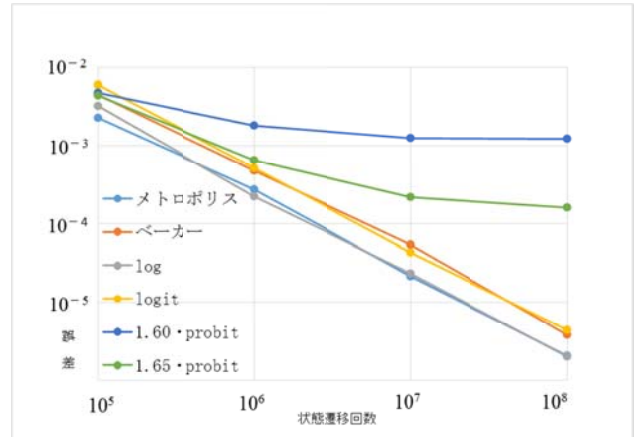


図 2 MCMC における誤差比較

な差が現れている。このことから、おおよそ遷移回数が 10^7 を超える付近から、 logit と probit との差異が現れるのではないかとと思われる。

この実験は、メトロポリス・ヘイスティングス法の誤差に関する実験であるが、焼きなまし法の代替として AIR を用いる場合には温度 T の存在が σ を吸収してくれるため、常に最適な σ_0 として機能する。また、焼きなまし法では温度 T を変化させるために、本実験で確認された 10^7 回という上限回数はおそらく、温度帯ごとの上限回数となるのではないかと考える。このため、焼きなまし法においては、より多い遷移回数にも耐えるものと考えられる。

3.2 次元縮小射影のピボット選択

本実験では、動画のコマ送り画像から抽出した約 700 万件 64 次元の特徴データを用いた。特徴間の類似度 (相違度) は、 L_1 距離を用いた。次元縮小射影としては、Simple-Map [14] を用いた。Simple-Map の射影は、空間内の点をピボットとし、ピボットとデータ間の距離を射影像として用いる。Simple-Map は、三角不等式を満たす任意の距離空間に適用できるという特徴を持っている。ピボットを n 個用いることで、 n 次元の L_∞ 距離空間に射影できる。2 点間の距離は射影後に縮むことはあっても伸びることはない。空間検索においては、ある程度低次元に射影して、できるだけ距離の縮みが小さくなるようにすることが重要である。本実験では、目的関数として距離保存率 (射影前の距離の総和に対する射影後の距離の総和) を用いて、その最大化を行った。ピボット探索のための評価用には、 $N=5,000$ ペアのデータ点をテストケースごとに無作為に抽出して用いた。

実験環境としては、2.53GHz Intel Core i5、主記憶 8GB の mac OS X 環境上にて測定を行った。ここでは、SA と AIR とが近似になっていることを示すために、遷移回数を揃えた結果として報告する。遷移回数を 11×10^3 回、 40×10^3 回、 400×10^3 回でそれぞれそろえて行った。実験結果を表 2 に示す。

| 手法 | 遷移回数 ($\times 10^3$) | 処理時間 (秒) | 距離保存率 | |
|-----|---------------------------|-------------|--------|--------|
| | | | 平均 | 標準偏差 |
| SA | 11 | 149.7 | 57.06% | 0.2763 |
| | 40 | 511.1 | 57.36% | 0.2379 |
| | 400 | 4864 | 57.52% | 0.1485 |
| AIR | 11 | 28.80 | 57.10% | 0.2260 |
| | 40 | 70.49 | 57.35% | 0.1333 |
| | 400 | 592.5 | 57.57% | 0.1547 |

表 2 ピボット探索における SA と AIR の比較

SA においても AIR においても、遷移回数を増やすことで同じくらいのペースでスコア平均の上昇する様子が観測できた。処理速度については、SA より AIR の方が 5 倍から 8 倍程度高速になっており、その差は、遷移回数が増えるほど大きくなっていることが確認できる。

3.3 焼きなましに基づくクラスタリング

本実験では、SA を用いたクラスタリング手法に着目する。最小化する典型的な目的関数は、各点と最も近いクラスタ中心との二乗誤差の和 (SSE) である。2013 年に Merendino と Celebi は、ガウス遷移を用いた中心摂動に基づく SA クラスタリングアルゴリズム SAGM を提案した[11]。SAGM は MMC と SMC の二つの冷却スケジュールを採用している。彼らは UCI 機械学習リポジトリ[6]の 10 個のデータセットを使用した実験を通して、SAGM(SMC)が他の SA アルゴリズムよりも大幅に速く収束することを報告した。表 3 に実験で使用したデータセットの諸元を示した。ここで、 N はデータ数、 d は属性数、 k はクラス数である。

SAGM(MMC)と SAGM(SMC)を C++で実装し、それらの AIR 版についても互換スケジュールにて、それぞれ AIR(MMC)と AIR(SMC)として実装した。実験環境は、Intel(R) Core(TM) i7-7820X CPU @ 3.60GHz、主記憶 64GB の Windows10 環境上にて Windows Subsystem for Linux を用いて Ubuntu を動かす、その上での測定を行った。初期ピボット(クラスタ中心)はサンプルから無作為に選んだものをテストケースとして、それぞれ 100 テストケースで実験を行った。解の品質は SSE によって評価され、値が小さいほど結果がよいことを示している。表 4 で示す SSE 平均の比較より、SA と AIR の間には品質面で大きな差がないことが確認できる。また、表 5 で示す処理時間平均では、9 番のデータセットを除くすべてのデータセットで AIR は SA より高速に動作している。9 番のデータセットではデータサイズが最も小さい($N=178$)ため、SA がわずかに速い。実行時間に対するデータサイズ N の影響を観察するため、図 4 に MMC および SMC におけるデータサイズに伴う SA と AIR の処理時間比を示している。図から分かるように、データサイズが大きくなるほど AIR の方がより速くなり、再標本による高速化の効果が現れていることが分かる。

| ID | データセット名 | N | d | k |
|----|---------------------|-------|-----|-----|
| 1 | Ecoli | 336 | 7 | 8 |
| 2 | Glass | 214 | 9 | 6 |
| 3 | Ionosphere | 351 | 34 | 2 |
| 4 | Iris Bezdek | 150 | 4 | 3 |
| 5 | Landsat | 6435 | 36 | 6 |
| 6 | Letter Recognition | 20000 | 16 | 26 |
| 7 | Image Segmentation | 2310 | 19 | 7 |
| 8 | Vehicle Silhouettes | 846 | 18 | 4 |
| 9 | Wine Quality | 178 | 13 | 7 |
| 10 | Yeast | 1484 | 8 | 10 |

表 3 データセット一覧

| ID | MMC | | SMC | |
|----|--------------|--------------|--------------|--------------|
| | SAGM | AIR | SAGM | AIR |
| 1 | 17.55 | 17.53 | 17.60 | 17.56 |
| 2 | 18.91 | 19.05 | 18.98 | 19.08 |
| 3 | 630.9 | 638.8 | 630.9 | 646.8 |
| 4 | 6.988 | 6.986 | 6.988 | 6.991 |
| 5 | 1742 | 1742 | 1742 | 1742 |
| 6 | 2732 | 2720 | 2738 | 2722 |
| 7 | 411.9 | 395.2 | 413.1 | 396.2 |
| 8 | 225.7 | 224.6 | 225.8 | 224.6 |
| 9 | 37.83 | 37.81 | 37.85 | 37.82 |
| 10 | 58.90 | 59.08 | 58.90 | 59.04 |

表 4 解の品質 (SSE 平均)

| ID | MMC | | SMC | |
|----|-------|-------|-------|-------|
| | SAGM | AIR | SAGM | AIR |
| 1 | 1.727 | 1.376 | 0.282 | 0.182 |
| 2 | 0.719 | 0.629 | 0.125 | 0.091 |
| 3 | 1.549 | 0.857 | 0.281 | 0.095 |
| 4 | 0.139 | 0.110 | 0.022 | 0.015 |
| 5 | 17.40 | 1.051 | 3.216 | 0.125 |
| 6 | 167.4 | 13.72 | 28.82 | 1.803 |
| 7 | 3.285 | 0.606 | 0.523 | 0.056 |
| 8 | 1.298 | 0.367 | 0.219 | 0.035 |
| 9 | 0.786 | 0.814 | 0.141 | 0.149 |
| 10 | 2.854 | 0.971 | 0.497 | 0.088 |

表 5 処理時間平均(sec)

4. 結論

再標本評価による山登り法が焼きなまし法の近似となることを、logitとprobitの近似関係から理論的に示した。また、これについては、MCMC における誤差比較実験を行い、実際に同傾向の結果が得られることについても示した。

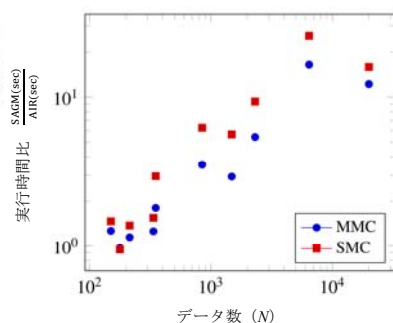


図 3 データ数と SA と AIR の実行時間比

温度 T と関連の深い再標本サイズ n は、総標本数 N に近付くまでは N の影響をほとんど受けないため、全てのデータセットにおいて同程度の n を用いることが可能になる。このため、処理時間比較の実験において N の大きなデータセットでは特に顕著な結果を得ることができた。

今回のクラスタリング問題の実験では比較対象の SA-Java があまりに重かったために、最大の N は 2000 であり、より大きなデータ集合での実験を行えなかったが、ビッグデータが流行る昨今、 N が大きくなればなるほどに有利になる AIR は、大きなアドバンテージを有すると思う。これは、従来の SA では適用が難しかった分野への応用の可能性を示すものである。

謝辞 SAGMとそのためのスケジュールSMCとMMCをソースコードとして提供してくださったM. Emre Celebi教授に感謝します。また、第1著者は、いくつかのプログラミングコンテストに謝意を表したいと考えています。本論文の最も重要な動機は、これらのコンテストの経験に基づいている。プログラミングコンテストにおけるハイパーパラメータのチューニングについて、AIRを用いてチューニングを行っている。あまり時間をかけずに大量のハイパーパラメータを適用してもほどよくチューニングすることができる。

参考文献

- [1] Aarts, E. and Korst, J. (1989). Simulated annealing and Boltzmann machines: A stochastic approach to combinatorial optimization and neural computing, Wiley.
- [2] Anily, S. and Federgruen, A. (1987). Simulated annealing methods with general acceptance probabilities, *App. Prob.*, 24, pp. 657–667.
- [3] Barker, A. A. (1965). Monte Carlo calculations of the radial distribution functions for a proton-electron plasma, *Aust. J. Phys.*, 18, pp. 119–133.
- [4] Bustos, B., Navarro, G., and Chávez, E. (2001). Pivot selection techniques for proximity searching in metric spaces, In *Proc. Computer Science Society, SCCC'01, XXI International Conference of the Chilean, IEEE*, pp. 33–40.
- [5] Demidenko, E. (2013). *Mixed Models: Theory and Applications with R*, Wiley, 2nd edition.
- [6] Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences, <http://archive.ics.uci.edu/ml>.
- [7] Dong, W., Charikar, M., and Li, K. (2008). Asymmetric distance estimation with sketches for similarity search in high-dimensional spaces, In *Proc. 31st ACM SIGIR*, pp. 123–130.
- [8] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, 57, pp. 97–109.
- [9] Imamura, Y., Higuchi, N., Kuboyama, T., Hirata, K., and Shinohara, T. (2017). Pivot selection for dimension reduction using annealing by increasing resampling, In *Proc. Learn. Wissen Daten Analysen, (LWDA'17)*, pp. 15–24.
- [10] Kirkpatrick, S. and Gelatt Jr., C. D. (1983). Optimization by simulated annealing. *Science*, 220, pp. 671–680.
- [11] Merendino, S. and Celebi, M. E. (2013). A simulated annealing clustering algorithm based on center perturbation using Gaussian mutation, In *Proc. FLAIRS Conference*, pp. 456–461.
- [12] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., and Teller, A. H. (1953). Equation of state calculations by fast computing machines, *J. Chem. Phys.*, 21, pp. 1086–1092.
- [13] Schuur, P. C. (1997). Classification of acceptance criteria for the simulated annealing algorithm, *Math. Oper. Res.*, 22, pp. 266–275.
- [14] Shinohara, T. and Ishizaka, H. (2002). On dimension reduction mappings for approximate retrieval of multidimensional data, In Arikawa, S. and Shinohara, A., editors, *Progress in Discovery Science*, volume 2281 of LNCS, pp. 224–231, Springer.