

部分的完全調査モンテカルロ法の推定精度の検証と 閾値の自動設定

川村 秀憲[†] 小野寺 将輝^{††} 大内 東[†]

ある事象が複数の独立な確率事象の組合せとして表現されており、それぞれの事象に評価値が割り当てられているときに評価値の期待値を推定する問題において、加重サンプリングモンテカルロ法と同じサンプル数でより推定精度を向上させることができる手法に、部分的完全調査モンテカルロ法がある。この手法は生起確率の閾値に基づいて対象事象を分割し、完全調査と加重サンプリングモンテカルロ法を複合的に実行することで推定を行うが、対象事象がどのような特徴を持っているときに精度を向上させることができるのか、また閾値はどのように設定したらよいのかについては明らかにされていない。本論文では、部分的完全調査モンテカルロ法の推定精度の向上が可能である条件について検討を行い、予備実験を行うことなく推定手順の実行中に閾値を自動調整する方法についての提案を行う。また、計算機実験により計算時間からの検証も含めて提案手法の有効性を確認する。

Accuracy of Partially Exhaustive Investigation Monte Carlo Method and Automatic Adjustment of Threshold Parameter

HIDENORI KAWAMURA,[†] MASAKI ONODERA^{††} and AZUMA OHUCHI[†]

We focus on a method to calculate the estimated value for a probabilistic model in which each event is defined as a combination of several other independent events. The partially exhaustive investigation Monte Carlo method has been proposed by the authors for such model, and it can show higher performance than simple weighted sampling Monte Carlo method on the condition that the used sample number for estimating is limited. This method divides the event into two parts by a threshold parameter; one is composed by high-probability events, and another is by remaining events. To make this method for practical use, these are unclear appropriate threshold setting and a condition that this method works effectively. In this paper, we investigate the condition that this method works effectively, and propose the partially exhaustive investigation Monte Carlo method with automatic threshold parameter setting.

1. はじめに

1953年にメトロポリスらによって提案されたモンテカルロ法¹⁾(以下、MC法)は、物理学や統計力学など様々な分野の問題へ応用されてきた²⁾⁻⁴⁾。MC法による典型的な計算の例としては、連続系においてある関数とその確率分布が与えられた場合の期待値の推定などがあげられる。また、最適化問題において不確実な要素を含んだ解の評価へMC法による期待値の推定値を用いる方法が提案されている⁵⁾。

一般に、MC法の推定精度は使用するサンプルの数

が多ければ多いほど向上する。具体的には、サンプル数 N に対し、誤差は $O(N^{-\frac{1}{2}})$ となる⁶⁾。十分な精度を確保したければ多数のサンプルが必要であるが、推定を何度も行う必要がある場合などには、計算コストの制約から十分な数のサンプリングを行うことが困難な場合がある。

一方、推定対象が複数の独立な確率事象の組合せで定義され、かつそれぞれの事象での確率分布に偏りが大きい場合、確率の大きい事象から順に確率の積を計算して列挙することによって、全事象を調査しなくても生起確率の高い結果を列挙可能な場合がある。我々はこの特徴を利用し、生起確率の高い事象に関しては完全調査を行い、それ以外の事象に対しては加重サンプリングに基づくモンテカルロ法を行うことで期待値の推定を行う部分的完全調査MC法(Partially Exhaustive Investigation MC法、以下PEXI-MC法)

[†] 北海道大学大学院情報科学研究科
Graduate School of Information Science and Technology, Hokkaido University

^{††} NTT ドコモ北海道
NTT DoCoMo Hokkaido

を提案した⁷⁾。

PEXI-MC 法では、期待値を推定する確率事象に対して閾値を設定し、それぞれ生起確率が閾値以上と閾値未満の事象に分割する。そして、閾値以上のものに関しては完全調査を行うことで期待値に寄与する値の真値を計算し、閾値未満のものに関しては加重サンプリングを行うことでその寄与値を求める。その後、それぞれの値を合算することで期待値の推定を行う。論文 7) では、閾値を適切に設定することで同じサンプル数でも加重サンプリング MC 法と比較して推定精度を向上させることができる場合があることを確かめ、具体的な応用例として複数種オークションからなる取引に参加するエージェントの意思決定設計でその有効性を確認した。

しかしながら、応用性を考えると上記の提案では不十分な点があった。まず、実際の応用ではサンプリング回数を固定するのが現実的であり、その際に PEXI-MC 法では使用できるサンプル数がサンプリング回数より減少する可能性があるが、それを考慮したうえでの精度に関する検討を行っていなかった。また閾値の設定に関して、推定を行う事象の確率分布に関する予備実験を行うことができればよいが、実際の応用では推定ごとに分布が変わってしまう場合が多く、そのたびに予備実験を行っては計算コストの増加が精度向上のメリットに釣り合わない。したがって、推定を行いながら適切な閾値を自動で調整する方法が望まれる。

本論文では、複数の独立な確率事象の組合せで推定対象が定義される場合において、サンプリング回数を固定した状況で確率事象がどのような条件を満たすときに PEXI-MC 法が有効なのかについて検証を行い、その知見を応用してサンプリングの過程で動的に閾値の調整が行われる閾値自動設定 PEXI-MC 法の提案を行う。この方法は、完全調査を行う部分から計算できる数値を利用して閾値を自動調整して適切な大きさまで動的に完全調査の対象を広げていくので、対象問題の性質をあまり考慮することなく PEXI-MC 法を適用することが可能となる。実装の観点からみると、単純な加重サンプリング MC 法に比べて 1 サンプルを抽出する際の計算コストは増大するが、閾値自動調整 PEXI-MC 法はサンプル数を節約したうえで精度向上が望めるので、評価値の計算コストによってはトータルの計算コストを減少させることができる場合があることを示す。

以降、2 章では対象とする問題と PEXI-MC 法の概要について、続く 3 章、4 章では PEXI-MC 法の推定精度に関する考察と検証実験の結果について述べる。

5 章では閾値自動設定 PEXI-MC 法の提案とその有効性の検証を行い、6 章、7 章では PEXI-MC 法に関する議論と本論文のまとめを述べる。

2. 部分的完全調査モンテカルロ法

2.1 対象問題

ここで取り扱う対象問題について述べる。まず、期待値の推定の対象となる事象は複数の独立な確率事象の組合せとして定義され、かつ各事象の生起確率は既知であるとする。また、それぞれの組合せについて評価値が割り当てられているものとし、問題の目的は評価値の期待値を精度良く推定することとする。

具体的に、生起する可能性のある事象 x_i の集合を X 、 x_i を定義する複数の事象 y_{jk} の集合を Y_j と定義する。

$$X = \{x_i = (y_1, \dots, y_N) \mid y_1 \in Y_1, \dots, y_N \in Y_N\} \quad (1)$$

$$Y_j = \{y_{j1}, \dots, y_{jN_j}\}, \quad j = 1, 2, \dots, N \quad (2)$$

x_i の生起確率を $p(x_i)$ 、 y_{jk} の生起確率を $p(y_{jk})$ とすると、 $p(x_i)$ は $p(y_{jk})$ の積として下記のように定義される。

$$p(x_i) = p(y_1) \times p(y_2) \times \dots \times p(y_N) \quad (3)$$

ただし、

$$\sum_{x \in X} p(x) = \sum_{y \in Y_j} p(y) = 1 \quad (4)$$

である。 x_i の評価値を $e(x_i)$ とすると、期待値の真値 E は以下のように定義できる。

$$E = \sum_{x \in X} p(x) \cdot e(x) \quad (5)$$

この期待値をできるだけ少ない計算コストで精度良く推定することが目的である。

2.2 部分的完全調査モンテカルロ法

次に PEXI-MC 法の概要を述べる⁷⁾。まず事前に確率事象 X に対して完全調査を行う事象と加重サンプリング MC 法を行う事象に分割するための生起確率の閾値 θ ($0 \leq \theta \leq 1$) を設定する。次に X の各事象について式 (3) に従って生起確率を計算できるが、確率の積の計算過程で閾値を下回った時点で計算を打ち切ることによって効率良く閾値以上の生起確率を持つ事象を列挙し、完全調査を行う。

ここで、完全調査のために列挙された事象集合を X_1 とし、 X から X_1 を除いた事象集合を X_2 とする。また、推定を行う際のトータルのサンプリング回数を M 、 X_1 、 X_2 からサンプルを抽出する際のサンプリング回数をそれぞれ m_1 、 m_2 とするとき、 $M = m_1 + m_2$ であり、 m_1 は X_1 の要素数と一致する。

まず, X_1 については完全調査を行うので X_1 からの推定値への寄与値 D_1 を以下のように計算する.

$$D_1 = \sum_{x \in X_1} p(x) \cdot e(x) \quad (6)$$

次に X_2 からの寄与値 D_2 を加重サンプリング MC 法で求める. X_2 に含まれる要素 $x \in X_2$ の生起確率を $p_2(x)$ とすると,

$$p_2(x) = p(x) \Big/ \sum_{x' \in X_2} p(x') \quad (7)$$

となるので, $p_2(x)$ に従って加重サンプリングを行うわけであるが, 計算コストを考えると $p_2(x)$ をすべて求めてから X_2 に含まれるサンプルを発生させることは容易ではない.

したがって, X 全体から $p(x)$ に従って加重サンプリングを行い, その中から X_1 に含まれるものを除外して X_2 に含まれるサンプルを準備するのが現実的である. このとき, サンプルが m_2 個そろそろまでサンプリングを続ける方法と, サンプリング回数を m_2 回に固定して得られるサンプルを用いる方法が考えられるが, 実用を考えた場合前者はあまり現実的な方法とはいえないので, ここでは後者の方法を採用する.

その際, 破棄される確率を考慮し, 実際に利用できる X_2 のサンプル数は理論的に $m'_2 = m_2 \cdot \sum_{x \in X_2} p(x) \leq m_2$ と見積もることができる. つまり, 完全調査も含めて X より M 回のサンプリングを行うが, 実際には $M' = m_1 + m'_2 \leq M$ 個のサンプルで推定を行うことになる.

このようにして X_2 から得られた m'_2 個のサンプルを $\xi_1, \xi_2, \dots, \xi_{m'_2}$ とすると, X_2 からの寄与値 D_2 は,

$$\begin{aligned} D_2 &= \sum_{x \in X_2} p(x) \cdot \frac{1}{m'_2} \sum_{i=1}^{m'_2} e(\xi_i) \\ &= \frac{1}{m_2} \sum_{i=1}^{m'_2} e(\xi_i) \end{aligned} \quad (8)$$

と計算できる.

X_1 と X_2 から計算できる寄与値 D_1, D_2 より, PEXI-MC 法による期待値の推定値 E_{PM} は次のように求めることができる.

$$E_{PM} = D_1 + D_2 \quad (9)$$

3. 推定精度に関する考察

ここでは加重サンプリング MC 法と比較して PEXI-MC 法の推定精度について議論する. MC 法の推定精度を評価する指標として推定結果の分散を用

いることができる⁶⁾. つまり, 分散が小さいほど推定精度が高いと見なすことができる.

まず, X より単純に M 個のサンプルを加重サンプリングした場合のサンプルを $\xi_1^*, \xi_2^*, \dots, \xi_M^*$ とする. このとき得られる加重サンプリング MC 法による期待値の推定値を E_M , 分散を $\text{var } E_M$ とすると, それぞれ下記のように見積もることができる.

$$E_M = \frac{1}{M} \cdot \sum_{i=1}^M e(\xi_i) \quad (10)$$

$$\text{var } E_M = \frac{1}{M} \text{var } e(\xi) \quad (11)$$

また, PEXI-MC 法によって得られる推定値 E_{PM} の分散 $\text{var } E_{PM}$ は, 式 (6), (8), (9) より

$$\begin{aligned} \text{var } E_{PM} &= \text{var}(D_1) + \text{var}(D_2) \\ &= \left(\frac{1}{m_2}\right)^2 m'_2 \cdot \text{var } e(\xi) \\ &= \frac{1 - \sum_{x \in X_1} p(x)}{M - m_1} \text{var } e(\xi) \end{aligned} \quad (12)$$

と見積もることができる. これは, 実際に推定に使用するサンプル数が $M' \leq M$ であることを考慮した分散である.

ここで, サンプリング回数が M として固定されている条件を考慮し, 式 (11) と式 (12) を比較する. 使用サンプル数の減少を考慮しても PEXI-MC 法による推定の分散が小さくなる場合は, 式 (11) と式 (12) より以下の条件を満たしていることになる.

$$\frac{M \cdot \left(1 - \sum_{x \in X_1} p(x)\right)}{M - m_1} \cdot \frac{\text{var } e(\xi)}{\text{var } e(\xi^*)} \leq 1 \quad (13)$$

ここで, $p(x)$ と $e(x)$ が無相関であり, $\text{var } e(\xi) \approx \text{var } e(\xi^*)$ が成り立つと仮定すると,

$$R = \frac{M}{M - m_1} \cdot \left(1 - \sum_{x \in X_1} p(x)\right) \leq 1 \quad (14)$$

を満たしている場合には, サンプル数の減少があったとしても PEXI-MC 法による推定精度の向上が望め, なおかつ R の値が小さければ小さいほど推定精度が良いと考えることができる.

ここで重要なのは, R が X_1 の完全調査による結果だけを用いて計算することが可能であるという点である. すなわち, X 全体を調べることなく, PEXI-MC 法で必然的に行われる X_1 の列挙結果から計算できる数値のみを用いて X 全体に対する PEXI-MC 法の推定精度が評価できることになる.

以上をまとめると, $p(x)$ と $e(x)$ が無相関である場合においては, X_1 の列挙の過程で R を観察すること

により閾値を自動調整し、加重サンプリング MC 法よりも良い精度で推定を行える可能性があるというのが本論文の核となるアイデアである。

4. 推定精度に関する検証実験

4.1 設定

ここでは考察の正当性を確認するためにいくつかの実験を行う。対象として、各目の生起確率が異なるさいころを複数同時に振り、出た目の組合せによって評価値が決まる状況において、評価値の期待値の推定を行う問題を取り上げる。

具体的に、生起確率が既知である 1~10 の目を持った 6 個のサイコロを同時に振るとすると、以下のようモデル化できる。

$$X = \{x_i = (y_1, \dots, y_6) \mid y_1 \in Y_1, \dots, y_6 \in Y_6\} \quad (15)$$

$$Y_j = \{y_{j,1}, \dots, y_{j,10}\}, j = 1, 2, \dots, 6 \quad (16)$$

すべての $p(y_{jk})$ は既知とし、 $p(x_i)$ はこれらの積として計算可能である。ただし、簡単化のために、

$$p(y_{1,k}) = \dots = p(y_{6,k}), k = 1, 2, \dots, 10 \quad (17)$$

とする。 $e(x_i)$ は、 $p(x_i)$ とは無相関に割り当てられた $N(100.0, 1.0)$ の正規乱数とする。また、比較的小さいサンプリング回数で期待値を推定する状況を想定し、サンプリング回数 M を目の組合せ総数の 0.1% である $M = 1000$ と固定して実験を行う。それぞれの実験結果は 1,000 試行の平均である。

4.2 実験 1

まず、PEXI-MC 法の基本的な特性を調べるために、図 1 に示すような生起確率の分散が異なる目の組合せを持った 3 種類の $p(y_{jk})$ の設定に対して適用を試み、あらかじめ求めておいた期待値の真値との誤差率を調べる。それぞれの設定は生起確率をランダムに割り当てたものから分散が異なるものを選択した。ここでは、 m_1 と誤差率の関係を明らかにするために閾値による列挙は行わず、あらかじめ $p(x_i)$ に従って降順にソートした x_i のデータベースから上位 m_1 個を X_1 とし、 X_2 は m_2 回の加重サンプリングを行って X_1 に含まれるものを破棄することで推定を行う。実験においては、 m_1 を M の 0% から 90% まで 10% 刻みで変化させ、完全調査とそれ以外のサンプルの配分の違いによる精度の差を比較する。なお、 $m_1 = 0$ のときの結果は加重サンプリング MC 法を用いた場合の結果と同等である。

図 2 は、各設定における期待値推定の誤差率を縦軸、 m_1 を横軸にとった実験結果である。加重サンプリング MC 法と比較して、 $p(x_i)$ の分散が小さい設定 1 に

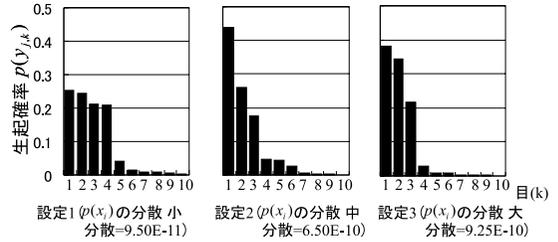


図 1 3 種類の $p(y_{jk})$ の設定
Fig. 1 Settings of $p(y_{jk})$ in the three cases.

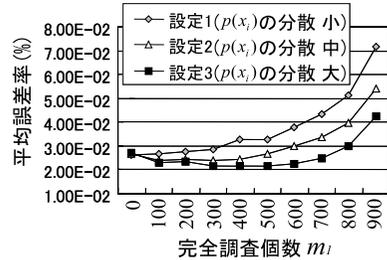


図 2 完全調査個数と平均誤差率の関係
Fig. 2 Average error rate depending on the sample number of exhaustive investigation.

おいては m_1 の値によらず PEXI-MC 法の誤差率が大きくなっている。一方で、設定 2 および設定 3 のような $p(x_i)$ の分散が大きい場合においては、 m_1 を適切に選ぶことで加重サンプリング MC 法よりも誤差率を小さくすることができる場合があることが確認できる。これらの実験結果から、 $p(x_i)$ の分散が大きい場合には m_1 を適切に設定することができれば推定精度を向上させることができる可能性があることが分かる。

4.3 実験 2

より広範に $p(x_i)$ の分散と推定精度の関係を調べるため、 $p(x_i)$ の分散が異なるようランダムに生成した $p(y_{jk})$ の設定を 100 種類用意し、実験 1 と同手順の PEXI-MC 法と加重サンプリング MC 法を比較する。ただし、ここでは代表例として $m_1 = 100$ の場合を示す。

図 3、図 4 は、設定ごとの $p(x_i)$ の分散を横軸にとり、縦軸を誤差率および推定値の分散として結果をプロットしたグラフである。各プロットはそれぞれの設定での 1,000 試行の平均に基づいており、また図中の線はそれぞれの値の 2 次の線形近似曲線を示している。結果より、 $p(x_i)$ の分散が大きい設定において加重サンプリング MC 法よりも PEXI-MC 法の推定精度が向上しており、また $p(x_i)$ の分散が大きいほど、精度向上の度合いが大きいことが確認できる。これらの結果から、 $p(x_i)$ の分散が大きいほど PEXI-MC 法

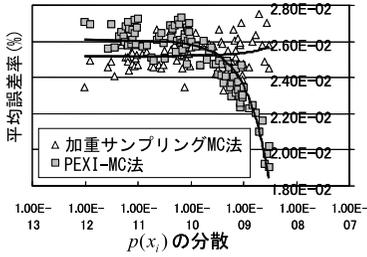


図 3 平均誤差率と $p(x_i)$ の分散の関係

Fig. 3 Average error rate depending on the variance of $p(x_i)$.

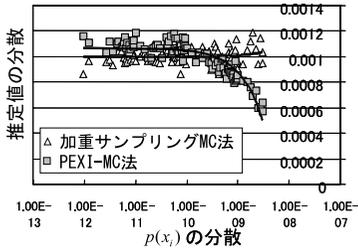


図 4 推定値の分散と $p(x_i)$ の分散の関係

Fig. 4 Variance of the estimate value depending on the variance of $p(x_i)$.

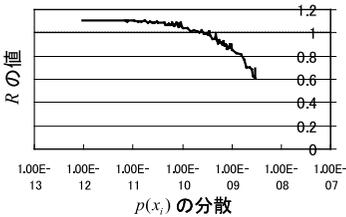


図 5 R の平均と $p(x_i)$ の分散の関係

Fig. 5 Average value of R depending on the variance of $p(x_i)$.

の有効性は向上すると考えられる。

図 5 は、各結果における R の値を縦軸にプロットしたグラフである。実験の設定上、 $e(x_i)$ は $p(x_i)$ と無相関に割り当てられているので、この値が 1 を下回るときに加重サンプリング MC 法と比較して PEXI-MC 法の推定精度の向上が期待できる。図 3、図 4 の結果と見比べると、加重サンプリング MC 法よりも PEXI-MC 法の推定精度が向上している点と R が 1 を下回る点がほぼ一致していることが確認できる。

4.4 実験 3

実験 1, 2 では m_1 を固定して実験を行ったが、実際の応用では閾値 θ によって X_1 が決定されるために m_1 を直接指定することはできない。そこで今回は θ の設定の影響を調べるために、実験 1 で用いた 3 種類の設定に対して異なる θ で PEXI-MC 法を適用

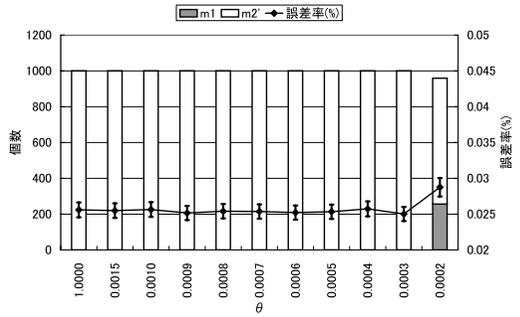


図 6 設定 1 ($p(x_i)$ の分散小) におけるサンプル数と閾値の関係
Fig. 6 Sample numbers depending on the value of θ in the first setting.

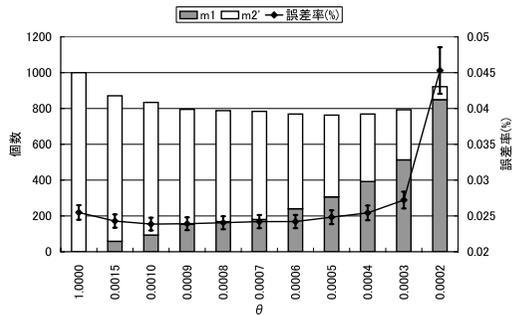


図 7 設定 2 ($p(x_i)$ の分散中) におけるサンプル数と閾値の関係
Fig. 7 Sample numbers depending on the value of θ in the second setting.

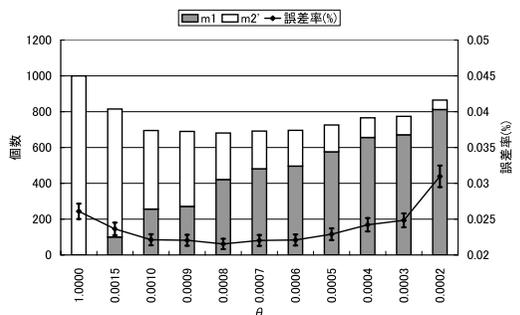


図 8 設定 3 ($p(x_i)$ の分散大) におけるサンプル数と閾値の関係
Fig. 8 Sample numbers depending on the value of θ in the third setting.

した際の推定精度、 m_1 、 m_2 について検証を行った。図 6、図 7、図 8 はそれぞれの設定において、横軸に θ 、縦軸に m_1 、 m_2 の値、および誤差率をとって結果をグラフ化したものである。グラフにおける縦棒は各誤差率の分散を表している。 $\theta = 1.0$ のときの結果は加重サンプリング MC 法を用いた場合と同等である。なお、ここではより正確に傾向を判断するために 10,000 試行の結果の平均を示している。

図 6 の設定 1 に関しては、 $p(x_i)$ の分散が小さく、ほとんどの θ に関してそれより大きい $p(x_i)$ を持つ

事象 x_i が存在しない．したがって，完全調査も行われないために誤差率もほぼ変化していない．唯一 $\theta = 0.0002$ と設定した場合にのみ m_1 が約 250 程度を示しており，完全調査と加重サンプリングが両方行われているが，誤差率は大きくなってしまっている．これは， $\theta = 0.002$ の設定によって列挙された X_1 の要素数の割に X_1 が占める生起確率が大きくならなかつたためであると考えられる．

一方，図 7，8 の設定 2 および設定 3 では， θ が小さくなるにつれて m_1 が多くなり，逆に m'_2 は少なくなっている．実際に使用するサンプルの合計数 M' は θ の減少にともなって下に凸な形状を示している．誤差率に関しては， θ の設定によって $m_1 = 0$ の結果より良くなっていることが確認できる．これは， θ の設定次第で m_1 の値の割に X_1 の占める生起確率を大きくとれるため，トータルで使用するサンプル数が少なくなっても R を 1 以下に抑えられているからと考えられる．

これらの実験結果から， θ を適切に設定することができれば $R < 1$ となるような適度な数の完全調査が行われ，提案手法を用いて誤差率を改善することが可能であると考えられる．さらに，閾値を過小な値に設定しなければ，生起確率の偏りが小さい場合には完全調査が行われないため，最低でも加重サンプリング MC 法と同等の精度を得ることが可能だと考えられる．

したがって，閾値の設定は PEXI-MC 法の有効性を決める重要な要因となる．

5. 閾値自動設定 PEXI-MC 法

5.1 閾値の自動設定

前述の考察より，式 (14) の R の値を 1 未満のできるだけ小さな値をとるように設定できれば推定誤差を小さくすることができる． R の値は完全調査を行う X_1 の列挙の過程で得られる情報のみで計算でき，また実験 2 より R は列挙する際の閾値の減少にともなって下に凸になる傾向があることが分かった．これらの特徴を考慮すると， R が最小を示すあたりまで逐次的に閾値を下げながら列挙を行えば，予備実験なしに閾値を自動設定することができると期待できる．

以上の点をふまえ，ここでは以下の手順のような閾値自動設定 PEXI-MC 法を提案する．

- (1) $X'_1 = \{\phi\}$ ， $R' = 1$ とする．また，以下の式に従って $p(x_i)$ の最大値を θ の初期値に設定する．

$$\theta = \prod_{j=1}^N \max_k^{N_j} y_{jk} \quad (18)$$

表 1 最終的な閾値と平均誤差率

Table 1 Finally selected threshold and average error rate.

	設定 1	設定 2	設定 3
閾値	1.0	0.000953	0.000786
誤差率 (%)	0.0250	0.0230	0.0216

表 2 サンプル数の平均値

Table 2 Average values of samples.

	M	M'	m_1	m'_2
設定 1	1,000	978	0 (22)	978
設定 2	1,000	783.769	153 (168)	630.769
設定 3	1,000	654.616	421 (481)	233.616

- (2) $p(x_i)$ が θ 以上の事象 x_i を列挙し，その集合を X_1 とする．さらに， X_1 に基づいて R を計算する．
- (3) $m_1 < M$ ， $R < 1.0$ ， $R < R'$ の諸条件を満たす場合， $X'_1 \leftarrow X_1$ ， $R' \leftarrow R$ ， $\theta \leftarrow \alpha \cdot \theta$ として閾値を下げ，再び (2) の手順に戻る．なお，ここで α ($0 < \alpha < 1$) は閾値の冷却率を表す固定パラメータである．
- (4) X'_1 を用いて PEXI-MC 法を実行する．

これらの手順を実装する場合には，閾値が更新されるたびに X_1 を初めから列挙し直すのでは列挙の効率が悪くなってしまうので， θ の更新に従って先ほどまでの続きから列挙を再開する工夫が必要である．これらを効率良く行えるよう作成した閾値自動設定 PEXI-MC 法の C 言語プログラムを付録に添付する．

5.2 実験 4

まず，閾値自動設定 PEXI-MC 法の有効性を確認するため，実験 2 と同様の問題に対して 1,000 試行の実験を行い，誤差率を比較する．実験を行うにあたって， $M = 1000$ ，冷却率は $\alpha = 0.9$ とした．

表 1 は，各設定において手順が停止した際の最終的な θ の値とそのときの誤差率を示している．設定 1 については閾値が 1.0 と設定され，加重サンプリング MC 法と同等の結果になっている．図 6，図 7，図 8 の結果とあわせるとほぼ適切な閾値を設定できているのが確認できる．特に設定 2，3 では，ほぼ誤差率が最小となる閾値が選ばれているのが確認できる．

表 2 は実際に推定に使用されたサンプル数である m_1 と m'_2 の平均値を示している． m_1 について，括弧内の数値は閾値自動設定の際に列挙した個数であり，推定に使用されたサンプル数は括弧外に示している．閾値設定は適切な閾値よりも 1 ステップ先まで進んだ後に終了するので，列挙を行ったが推定には使用しなかったサンプルが生じるために差がつくことがある．結果より，実際に使用したサンプル数 $M' = m_1 + m'_2$

は、サンプリング回数 M よりも減少していることが分かる。

5.3 実験 5

実験 4 では精度の面から提案法の有効性を確認した。しかし、提案手法は完全列挙の計算コストを考えると加重サンプリング MC 法と同じサンプリング回数を実行する場合には実行時間が多くかかってしまう可能性がある。もし同じサンプリング回数で閾値自動調整 PEXI-MC 法の計算時間が多くかかってしまうならば、それだけ加重サンプリング MC 法のサンプリング回数を多くすることができるので、提案手法が有効であるとはいえなくなってしまふ。一方、実験 4 の結果より推定に使用するサンプル数はサンプリング回数よりも減少する傾向があるので、各サンプルに対する評価時間がゼロではない場合、計算時間の面で提案手法が有効な場合もあると考えられる。有効な例としては、たとえばゲーム木の探索において、各状態の静的評価関数の計算時間がサンプリングにかかる時間よりも大きくなってしまふような場合があげられる。

ここでは、実験 4 の設定 ($M = 1000$, 冷却率 $\alpha = 0.9$) をそのまま用い、計算時間の観点から提案手法と加重サンプリング MC 法の比較を行う。まず評価値の計算時間が 0 ms となるように実装し、それ以外にかかる計算時間を実測する。その後、実験とは別にサンプル 1 つあたりに必要な評価時間を想定し、サンプル個数との積をとって 1,000 試行の平均計測時間と合計して全体の計算時間とする。なお、実験は Windows XP, Pentium4 2.26 GHz, 256 MB RAM のマシンで行った。

図 9, 図 10, 図 11 は、それぞれの設定において計測した実測時間に表 2 で示されたサンプル個数に基づいた評価時間を加えたものである。なお、サンプル 1 つあたりの評価時間は $0 \mu\text{s} \sim 100 \mu\text{s}$ の範囲で設定し、サンプル個数との積をとることで評価時間としている。

設定 1 に関しては、想定する評価時間にかかわらず提案手法の計算時間が加重サンプリング MC 法の計算時間を上回っている。これは、閾値を自動設定する手間をかけたにもかかわらず完全調査が行われなかったため、ほぼ通常の加重サンプリング MC 法が実行され、サンプルの評価回数の減少もあまり見込めずに提案手法の手間がそのまま余分な計算になっているためである。したがって、設定 1 のような状況では提案手法によって精度が向上しないだけでなく、計算時間も多くかかってしまう可能性がある。

一方、設定 2 および設定 3 に関しては、評価時間が

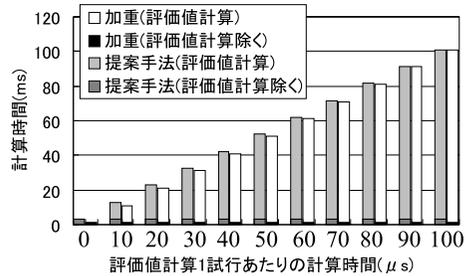


図 9 期待値推定 1 試行の計算時間 (設定 1)

Fig. 9 Calculation time for estimating the expected value in the first setting.

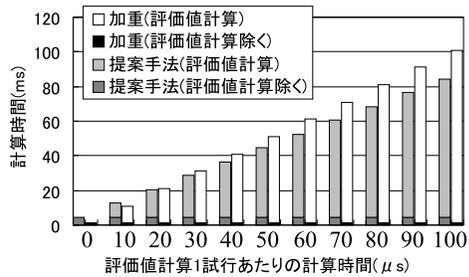


図 10 期待値推定 1 試行の計算時間 (設定 2)

Fig. 10 Calculation time for estimating the expected value in the second setting.

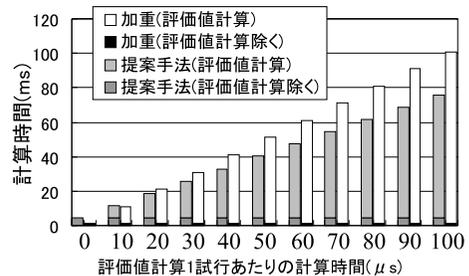


図 11 期待値推定 1 試行の計算時間 (設定 3)

Fig. 11 Calculation time for estimating the expected value in the third setting.

大きくなっていくに従って加重サンプリング MC 法よりも提案手法の計算時間が少なくなっていることが確認できる。これは、実際に推定に使用するサンプル数が少なくなっている分だけ評価時間の合計が少なくてすむからである。したがって、完全列挙と閾値自動調整の手順を含めた計算時間に対してサンプル 1 つあたりの評価時間が大きいような問題においては、提案手法によってトータルな計算時間が節約でき、かつ推定精度の向上を望むことができるといえる。

6. 議 論

本論文で取り扱った PEXI-MC 法は単純に発生させた乱数によってサンプルを抽出して推定するのでは

なく、推定対象となる事象の性質を積極的に利用することで精度の向上を図っている。他の MC 法に関する古典的な工夫として、推定対象から既知である主部を差し引いてサンプリングを行う主部分離法や、対象を複数の小区間に分類し、それぞれの区間で個別に MC 法を適用する層別サンプリングなど、推定対象に関する特徴や知識を積極的に利用することによって推定精度の向上を実現できることが知られている⁶⁾。

より PEXI-MC 法と類似した手法として、n-Queen 問題など探索木で表現される探索空間の中から解の総数を見積もるといった問題において、完全探索とモンテカルロ法を組み合わせた複合 MC 法という手法が寺田によって提案されている⁸⁾。複合 MC 法では、根からランダムに木を降下しながら探索する過程において、探索点がある一定のカットオフレベル以上に達した場合にはそれ以降の枝を完全調査することで解の総数の推定精度の向上を試みている。この手法は、木のバランスが悪いほど効果が大きくなると考えられ、寺田は計算機実験によって有効性を確認している。

木のバランスの悪さと生起確率の偏り、カットオフレベルと列挙の閾値の対比から、複合 MC 法と PEXI-MC 法は基本的アイデアが類似した手法であるといえる。異なる点としては、複合 MC 法は木をたどることで解の有無を探索するバイナリモデルを対象としているが、PEXI-MC 法は確率的事象における評価値の期待値推定という実数値を扱うモデルを対象としていること、PEXI-MC 法は推定精度が向上する条件などを理論的に説明してその知識を積極的に利用しているが、複合 MC 法はどのような場合に有効性が期待できるのか明らかになっていないといったことがあげられる。対象問題と手順を工夫することで PEXI-MC 法を複合 MC 法と同様の問題に適用可能である可能性があるため、今後検討する必要がある。

また、これらの MC 法の工夫と比較して、今回提案した PEXI-MC 法は完全調査部分の生起確率の和と事象の数という関係から閾値と精度の関係を考察し、閾値の自動調整を可能としているという特徴がある。閾値自動調整の方法は他の MC 法へ応用可能である可能性があり、今後詳細に検討を進めていく必要がある。

7. おわりに

本論文では、部分的完全調査モンテカルロ法の推定精度の向上が可能である条件について検討を行い、予備実験を行うことなく推定手順の実行中に閾値を自動調整する方法についての提案を行った。また、計算機実験により計算時間からの検証も含めて提案手法の有

効性を確認した。今回の提案は複数の独立な確率事象の組合せで表される問題を取り上げたが、原理的にはある事象を取り上げた際にその生起確率が分かれば本手法は適用可能である。今後はより手法を一般化し、応用可能性を広げていく必要がある。

参考文献

- 1) Metropolis, N., et al.: Equation of State Calculations by Fast Computing Machines, *The Journal of Chemical Physics*, Vol.21, No.6, pp.1087-1092 (1953).
- 2) 福島孝治: モンテカルロ法の前線—サイコロ振って積分する方法, 科研費特定領域研究「確率的情報処理への統計学的アプローチ」主催, 日本神経回路学会, 電子情報通信学会基礎・境界ソサイエティ共催, 若手研究者・学生向けに最新技術をわかりやすく紹介する講演会「確率的アルゴリズムによる情報処理」発表資料 (2003).
- 3) 宮武 修, 中山 隆: モンテカルロ法, 日刊工業新聞社 (1970).
- 4) Doucet, A., de Freitas, N. and Gordon, N.: *Sequential Monte Carlo Methods in Practice*, Springer-Verlag (2001).
- 5) 喜多 一, 佐野泰仁: GA による不確実な適度関数の最適化, 人工知能学会誌, Vol.18, No.5, pp.510-516 (2003).
- 6) 津田孝夫: モンテカルロ法とシミュレーション, 培風館 (1977).
- 7) 川村秀憲, 小野寺将輝, 大内 東: 部分的完全調査モンテカルロ法の提案と TAC エージェントの意思決定への応用, 情報処理学会論文誌, Vol.36, No.12, pp.317-328 (2005).
- 8) 寺田 実: 複合モンテカルロ法による探索木の解総数の見積り, 情報処理学会論文誌, Vol.38, No.8, pp.1492-1497 (1997).

付 録

A.1 閾値自動調整 PEXI-MC 法の C 言語ソー

スコード

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define MAX_SAMPLE 10000
#define MAX_RESUME 10000
#define MAX_EVENT 20
#define MAX_STATE 20
#define alpha 0.9

struct Model{
    int num_event, num_state[MAX_EVENT];
    double prob[MAX_EVENT][MAX_STATE];
};

struct Event{
    int state[MAX_EVENT+1];
```

```

double prob;
};

static struct X {
int n, max_smpls;
double sum_prob, sum_vals, theta;
} X1;

static struct Resume {
struct Event event[MAX_RESUME];
int rsmd_pnt[MAX_RESUME], n, crt;
} R;

void enum_X1_2(struct Model M,
struct Event E, int pos){
int i,j;
double prob;
if(pos < M.num_event){
prob = E.prob;
for(j = E.state[pos];
j < M.num_state[pos]; j++){
E.state[pos] = j;
E.prob = prob * M.prob[pos][j];
if(E.prob >= X1.theta){
E.state[pos + 1] = 0;
enum_X1_2(M, E, pos + 1);
}
else{
for(i=0; i < pos + 1; i++)
R.event[R.n].state[i] = E.state[i];
R.rsmd_pnt[R.n] = pos;
R.event[R.n++].prob = prob;
break;
}
}
}
else{
X1.sum_prob += E.prob;
X1.sum_vals += value(m, E) * E.prob;
X1.n++;
}
}

void enum_X1(struct Model M){
int i, n = R.n;
for(i = R.crt; i < n; i++)
enum_X1_2(M, R.event[i],
R.rsmd_pnt[i]);
R.crt = n;
}

void reset_X1(){
X1.n = R.crt = 0;
X1.sum_prob = X1.sum_vals = 0.0;
X1.theta = R.event[0].prob = 1.0;
R.event[0].state[0] = 0;
R.rsmd_pnt[0] = 0;
R.n = 1;
}

double calc_X1(struct Model M,
int max_smpls){
double cond1, cond2 = 1.0;
double pre_sum_prob = 0.0;
double pre_sum_val = 0.0;
double pre_theta = 1.0;
int i, pre_n;
reset_X1();
X1.max_smpls = max_smpls;
for(i=0; i<M.num_event; i++){
X1.theta *= M.prob[i][0];
while(1){
enum_X1(M);
cond1 = (double)max_smpls
/ (double)(max_smpls - X1.n)
* (1.0 - X1.sum_prob);
if(cond1 < cond2){
pre_sum_prob = X1.sum_prob;
pre_sum_val = X1.sum_vals;
pre_theta = X1.theta;
pre_n = X1.n;
cond2 = cond1;
}
else if(cond1 > cond2){
X1.sum_prob = pre_sum_prob;
X1.sum_vals = pre_sum_val;
X1.theta = pre_theta;
X1.n = pre_n;
break;
}
}
X1.theta *= alpha;
}
return(X1.n == 0 ? 0.0 :
X1.sum_vals / X1.sum_prob );
}

double calc_X2(struct Model M,
int smple_num){
int i,j,k,n = 0;
struct Event event;
double rnd, prob;
double sum_vals = 0.0, sum_prob = 0.0;
for(i = 0; i<smple_num-X1.n; i++){
prob = 1.0;
for(j = 0; j<M.num_event; j++){
rnd = (double)rand()
/ (double)RAND_MAX;
k = 0;
while (rnd > 0.0)
rnd -= M.prob[j][k++];
event.state[j] = k-1;
prob *= M.prob[j][k - 1];
}
if(prob < X1.theta){
sum_vals += value(M, event);
sum_prob += prob;
n++;
}
}
return( sum_vals / (double)n );
}

double peximc(
struct Model M, int smple_num){
return( calc_X1(M, smple_num)
* X1.sum_prob + (1.0 - X1.sum_prob)
* calc_X2(M, smple_num) );
}

```

A.2 ソースコードの説明

ソースコードにおけるマクロ定義部の変数の意味は下記のとおりである。実行する際は対象問題の規模と計算機のメモリ量にあわせて調整する必要がある。

MAX_SAMPLE サンプルの最大数
MAX_RESUME 列挙の最大再開点数
MAX_EVENT 最大事象数
MAX_STATE 各事象の最大要素数
alpha パラメータ α

Model 型は対象問題, X 型は確率事象, Event 型は個別の事象を表す. 推定を行う際にはまず, Event 型を引数としてその事象に対応する評価値を返す double 型の value 関数を準備する. ここで, Event 型の state[0, 1, ..., j, ...] は評価される事象 $y_1, y_2, \dots, y_{j+1}, \dots \in Y_j$ を表す. 次に, Model 型の変数を 1 つ準備し, num_event に N , num_state[j] に N_j , prob[i][j] に $p(y_{ij})$ をそれぞれ代入する. これらの準備ができれば, Model 型の変数とサンプリング回数を引数として peximc 関数を呼び出すことで推定結果が double 型で返ってくる.

(平成 17 年 3 月 24 日受付)

(平成 18 年 4 月 4 日採録)



川村 秀憲 (正会員)

1973 年 5 月 3 日生. 1996 年北海道大学工学部情報工学科卒業, 2000 年同大学大学院工学研究科博士後期課程期間短縮修了. 同年同大学院工学研究科助手. 2004 年同大学院情報科学研究科助手, 現在に至る. 飛行船ロボット, マルチエージェントシステム, 複雑系工学, 観光情報学等の研究に従事. 博士(工学). 人工知能学会, 日本オペレーションズ・リサーチ学会, 観光情報学会各会員.



小野寺将輝

1981 年 1 月 2 日生. 2003 年北海道大学工学部情報工学科卒業, 同年同大学大学院工学研究科修士課程システム情報工学専攻入学. 2005 年 NTT ドコモ北海道入社. 現在に

至る.



大内 東 (正会員)

1945 年 8 月 19 日生. 1968 年北海道大学工学部応用物理学科卒業, 1974 年同大学大学院工学研究科博士後期課程修了. 同年同大学工学部助手, 助教授, 1989 年同大学大学院工学研究科教授. 2004 年同大学院情報科学研究科教授, 現在に至る. 飛行船ロボット, DNA コンピューティング, マルチエージェントシステム, 医療システム等の研究に従事. 工学博士. 日本オペレーションズ・リサーチ学会, 医療情報学会, 観光情報学会各会員.