Kobe University Repository : Kernel

PDF issue: 2026-11-07

期待値の評価方法に関する考察 : 重点的サンプリング,棄却法,Metropolis-Hastingsアルゴリズムの比較

谷崎, 久志

(Citation)
国民経済雑誌,178(4):73-85

(Issue Date)
1998-10

(Resource Type)
departmental bulletin paper

(Version)
Version of Record

(JaLCDOI)
https://doi.org/10.24546/00209247

(URL)
https://hdl.handle.net/20.500.14094/00209247

期待値の評価方法に関する考察：重点的サンプリング，棄却法，Metropolis-Hastingsアルゴリズムの比較

谷　崎　久　志

1　はじめに

計量経済学，統計学において，最も基本的な主題の一つは期待値を推定することである。通常，確率変数には連続型を仮定するのが慣例であり，したがって，一部の特殊な場合を除いては，期待値に含まれる積分を明示的に評価するということが難しい場合が多い。このような積分が明示的に評価できない場合に，いかに期待値を求めるかという問題はかなり厄介な問題である。従来では，数値積分による積分値の近似が主流であった。しかし，台形公式，シンプソン公式，ガウス型積分公式等（例えば，Press, Teukolsky, Vetterling and Flannery (1992) を参照せよ）の数値積分法は，コンピュータの進歩が著しい今日でさえ次元が大きくなると計算量が膨大になり，実行不可能になることがよく知られた事実である。[1] よって，計算量を増大させないような様々な期待値の評価方法がこれまで考案されてきた。本稿では，重点的サンプリング (Geweke (1988, 1989a, 1989b), Koop (1994), Shao (1989)), 棄却法 (Boswell, Gore, Patil and Tallie (1993), Chib and Greenberg (1995), Knuth (1981), O' Hagen (1994)), Metropolis-Hastingsアルゴリズム[2] (Chib and Greenberg (1995, 1996), Geweke (1996, 1997), Tierney (1994)) による期待値の評価方法を紹介し，その精度を考察する。

重点的サンプリングとは，モンテカルロ積分という方法を用いて，積分値を乱数によって近似する方法である。それに対して，棄却法とMetropolis-Hastingsアルゴリズムとは，任意の密度関数から乱数を生成する方法であり，期待値は単に乱数の算術平均によって与えられる。言い換えると，棄却法とMetropolis-Hastingsアルゴリズムとは，積分計算をせずに期待値を求める方法である。Metropolis-Hastingsアルゴリズムは繰り返し計算によって，収束先が目的とする密度関数からの乱数となっているというところに特徴がある (Markov Chain Monte Carlo と呼ばれる)。棄却法とは，一つの独立な乱数を生成する方法であり，Independence Monte Carlo と呼ばれ，Markov Chain Monte Carlo とは区別される。

本稿では，この3つの方法で期待値を計算し，シミュレーションによって，精度を比較す

る。予想される結論は，棄却法が最も精度面からは優れているが，計算時間がかかるという欠点を持つということである。

2　期待値の評価方法

確率変数 x の分布関数を $f(x)$ とする。$g(\cdot)$ をある既知の関数としたとき，$\mathrm{E}(g(x))$ を評価することを考える。すなわち，$g(x)$ の期待値は

$$\mathrm{E}(g(x))=\int g(x)f(x)\mathrm{d}x \tag{1}$$

として与えられる。

$g(\cdot)$ や $f(\cdot)$ の関数形に依存するが，一般的には，多くの場合で，(1) 式を明示的に評価することは不可能である。そのような場合には，モンテカルロ法によって，(1) 式を近似的に評価する方法が採用される。(1) 式を近似的に評価するするためのモンテカルロ法には，いくつかの種類があるが，重点的サンプリング，棄却法，Metropolis-Hastings アルゴリズムが有名な方法である。

2.1　重点的サンプリング (Importance Sampling)

重点的サンプリングを行うためには，分析者によって適切に特定化されなければならない重点的密度関数 (Importance Density) と呼ばれる密度関数を導入する必要がある。重点的密度関数を $f_*(x)$ として定義する。

重点的サンプリングによる期待値の評価方法によると，重点的密度関数 $f_*(x)$ から N 個の乱数 $x_1, x_2, \cdots, x_N$ を生成し，その乱数によって期待値が近似的に評価されるのである。したがって，重点的密度関数 $f_*(x)$ は簡単に乱数の生成出来る密度関数 (例えば，一様分布，正規分布等) が選ばれなければならない。

期待値の近似は次のように行われる。

$$\begin{aligned}\mathrm{E}(g(x))&=\int g(x)f(x)\mathrm{d}x\\&=\int g(x)\frac{f(x)}{f_*(x)}f_*(x)\mathrm{d}x\\&=\int g(x)\omega(x)f_*(x)\mathrm{d}x\\&\approx\frac{1}{N}\sum_{i=1}^{N}g(x_i)\omega(x_i)\\&\equiv\overline{g(x)}\end{aligned}$$

2 行目では，重点的密度関数 $f_*(x)$ を用いて，積分が変形される。3 行目では，ウエイト関数 (Weight Function) が用いられ，その定義は，$\omega(x)\equiv f(x)/f_*(x)$ である。4 行目

では，重点的密度関数 $f_*(x)$ から生成された N 個の乱数 $x_1, x_2, \cdots, x_N$ を用いて，積分値が近似的に評価される。この重点的サンプリングに基づいた積分方法をモンテカルロ積分と呼ぶ。5行目では，近似的に評価された期待値（または，積分値）を $\overline{g(x)}$ として定義される。

推定値 $\overline{g(x)}$ の漸近的な性質は，中心極限定理により，

$$\frac{\overline{g(x)}-\mathrm{E}(g(x))}{\hat{\sigma}/\sqrt{N}} \rightarrow N(0,1)$$

となることが証明される（例えば，Geweke (1988, 1989a, 1989b) を参照せよ）。ただし，$\hat{\sigma}^2 \equiv (1/N)\sum_{i=1}^{N}(g(x_i)\omega(x_i)-\overline{g(x)})^2$ とする。

2.2 棄却法 (Rejection Sampling)

重点的サンプリングでは，期待値における積分値をモンテカルロ積分によって近似するという方法を用いた。しかし，本節と次節では，$f(x)$ から直接乱数を生成し，期待値を評価するという方法を採用する。$x_1, x_2, \cdots, x_N$ を $f(x)$ からの N 個の乱数とした場合，$E(g(x))$ は単に以下のように求められる。

$$\mathrm{E}(g(x))=\int g(x)f(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} g(x_i) \equiv \overline{g(x)}$$

このように，本節と次節で紹介する方法によると，生成したい密度関数 $f(x)$ からの乱数を生成することが必要となる。

$f_*(x)$ をサンプリング密度関数 (Sampling Density) として，$\omega(x)$ を以下のように定義する。

$$\omega(x)=\frac{f(x)}{\gamma f_*(x)}$$

これは受容確率 (Acceptance Probability) と呼ばれる。棄却法を実行するためには，以下の仮定を必要とする。

$$\gamma \equiv \sup_x \frac{f(x)}{f_*(x)} < \infty$$

棄却法は以下に示される(i)−(iv)の方法によって行われる。

(i) $f_*(\cdot)$ から乱数（z とする）を生成する。

(ii) $\omega(z)$ を求め，区間 $[0,1]$ の一様乱数 u を生成する。

(a) $u \leq \omega(z)$ のとき，z を $f(\cdot)$ からの乱数とする。すなわち，$x_i = z$ とする。ただし，x_i は第 i 番目の $f(x)$ からの乱数とする。

(b) $u > \omega(z)$ のとき，(i)に戻る。

(iii) そして，(ii)(a)が実現するまで，(i)と(ii)(b)が繰り返される。

(iv) $i=1,2,\cdots,N$について，繰り返し，N個の$f(x)$からの乱数$x_1, x_2, \cdots, x_N$を得る。

棄却法の妥当性については，例えば，Boswell, Gore, Patil and Tallie (1993)，Chib and Greenberg (1995)，Knuth (1981)，O' Hagen (1994) 等を参照せよ。

サンプリング密度関数$f_*(x)$について考えるために，例として，$f(x)$と$f_*(x)$が共に$N(\mu, \sigma^2)$と$N(\mu_*, \sigma_*^2)$の正規分布に従う場合をとりあげる。このとき，条件$\gamma<\infty$が満たされるためには，$\sigma_*^2>\sigma^2$となることが簡単に証明される[3]。これは$f_*(x)$は$f(x)$より分散の大きい分布でなければならないということを意味する。

条件$\gamma<\infty$が満たされている場合には，棄却法を用いることによって，任意の密度関数から乱数を生成することが可能となる。しかし，棄却法は次のような問題点を持つ。

(1) γを計算する必要があるが，γは必ずしも存在しない場合がある。この場合には，棄却法を適用することは出来なくなる。

(2) 受容確率$\omega(\cdot)$がゼロに近い場合（すなわち，γが無限大の場合）は，(i)と(ii)(b)が永遠に繰り返されるため，そのような場合には計算時間が非常にかかる。

棄却法による乱数$x_1, x_2, \cdots, x_N$は，互いに独立で，密度関数$f(x)$からの乱数となる。したがって，$E(g(x))$の推定値$g(x)$は，中心極限定理によって，

$$\frac{\overline{g(x)}-E(g(x))}{\hat{\sigma}/\sqrt{N}} \to N(0,1)$$

という結果が得られる。ただし，$\hat{\sigma}^2=(1/N)\sum_{i=1}^{N}(g(x_i)-\overline{g(x)})^2$とする。

棄却法については，例えば，Boswell, Gore, Patil and Taillie (1993)，O' Hagan (1994) and Geweke (1996) を参照せよ。

2.3 Metropolis-Hastings アルゴリズム

Tierney (1994)，Chib and Greenberg (1995, 1996)，Geweke (1996) 等は Metropolis-Hastings アルゴリズムについて研究した。Metropolis-Hastings アルゴリズムというのも，棄却法と同じく，任意の密度関数から乱数を生成する方法である。

$f(x)$からxの乱数を生成する場合，Metropolis-Hastings アルゴリズムによると，棄却法と同じく，別の密度関数（すなわち，サンプリング密度関数）を導入する必要がある。その密度関数は，通常，$f_*(x|z)$として置かれる。同様に，$f_*(x|z)$はサンプリング密度関数と呼ばれる。サンプリング密度関数$f_*(z|x)$は次の2つの条件を満たさなければならない。

(1) サンプリング密度関数$f_*(x|z)$については，素早く，かつ，簡単に乱数が生成出来るような密度関数が選ばれなければならない。

(2) 乱数を生成したい密度関数 $f(x)$ とサンプリング密度関数 $f_*(x \mid z)$ とは，x の分布する範囲が同じようなものである必要がある。

条件(1)は，棄却法においても必要な条件である。しかし，棄却法が実行可能であるためには，サンプリング密度関数が目的とする密度関数よりも幅広く分布する必要があり，上の条件(2)とは異なったものであるということに注意せよ。

Metropolis-Hastings アルゴリズムを実行するためには，まず，受容確率 $\omega(x,z)$ を次のように定義する。

$$\omega(x,z)=\begin{cases}\min\left(\dfrac{f(z)f_*(x\mid z)}{f(x)f_*(z\mid x)},\ 1\right), & \text{if } f(x)f_*(z\mid x)>0,\\ 1, & \text{otherwise.}\end{cases}$$

受容確率 $\omega(x,z)$ を用いて，Metropolis-Hastings アルゴリズムは以下のように行われる。

(i) x の初期値を x_0 をする。

(ii) x_{i-1} を与えたもとで，$f_*(\cdot \mid x_{i-1})$ からの乱数（z とする）を生成する。また，区間 $[0,1]$ の一様乱数（u とする）を生成する。

(iii) 受容確率 $\omega(x_{i-1},z)$ を計算し，次の(a)か(b)のいずれかを選ぶ。

(a) $u \le \omega(x_{i-1},z)$ のとき，$x_i=z$ とする。

(b) $u>\omega(x_{i-1},z)$ のとき，$x_i=x_{i-1}$ とする。

(iv) $i=1,2,\cdots,N$ について，(ii)と(iii)を繰り返す。

(v) 十分に大きな N について，x_N は $f(x)$ からの乱数とみなされる。

Metropolis-Hastings アルゴリズムの基本的な結果は，$N\longrightarrow\infty$ のとき，

$$\frac{1}{N}\sum_{i=1}^{N}g(x_i) \rightarrow \int g(x)f(x)\,dx \equiv E(g(x)), \qquad (2)$$

となることが知られている。ただし，$g(\cdot)$ はある関数とする。N個の乱数のうち最初の10－20%の乱数は，通常，期待値の評価のためには使われない。

サンプリング密度関数 $f_*(z \mid x)$ の選択が，Metropolis-Hastings アルゴリズムにおいては，最大の問題点である。サンプリング密度関数 $f_*(z \mid x)$ の分散は大きすぎたり，小さすぎたりするのは乱数の生成には効率的ではない（例えば，Chib and Greenberg (1995) を参照せよ）。すなわち，サンプリング密度関数 $f_*(z \mid x)$ は，乱数を生成したい密度関数 $f(x)$ と同じくらいの範囲に分布する必要がある（Chib and Greenberg (1996) を参照せよ）。

また，$f_*(z \mid x)$ の関数形について，$f_*(z \mid x)=f_*(z-x)$ のとき，ランダム・ウォーク・チェイン（Random Walk Chain）と呼ばれ，$f_*(z \mid x)=f_*(z)$ のとき，インディペンデンス・チェイン（Independence Chain）と呼ばれる。

Metropolis-Hastings アルゴリズムにおいては，$f(x)$ 必ずしも密度関数である必要はな

い。受容確率 $\omega(x,z)$ の計算方法を見ると，$f(x)$ は密度関数の核（kernel）であっても何ら問題は生じないからである。

Metropolis-Hastings アルゴリズムにおいては，$x_1, x_2, \cdots, x_N$ は厳密な意味での乱数ではないことに注意せよ。$x_1, x_2, \cdots, x_N$ は相互に相関，しかも，正の相関を持つからである。$x_1, x_2, \cdots, x_N$ の系列がエルゴード性という性質を持っていれば[4]，Metropolis-Hastings アルゴリズムによって得られた $x_1, x_2, \cdots, x_N$ は（2）という性質を持つことが知られている。

3　モンテカルロ実験

本節では，前節で紹介した重点的サンプリング，棄却法，Metropolis-Hastings アルゴリズムの精度比較をモンテカルロ実験によって行う。目的とする密度関数 $f(x)$ を標準正規分布，サンプリング密度関数 $f_*(x)$ を平均 μ_*，分散 σ_*^2 の正規分布の場合を考え，サンプリング密度関数におけるパラメータの値は $\mu_*=0.0,1.0,2.0,3.0$，$\sigma_*=0.5,1.0,1.5,2.0,3.0,4.0$ とする[5]。また，本節では，$g(x)=x$ のみを取り上げる。したがって，$\overline{g(x)}=\bar{x}$ ということになる。$\bar{x}^{(i)}$ は，第 i 回目のシミュレーションにおける $E(x)$ の推定値を表すものとする。本節におけるモンテカルロ実験の目的は，μ_*，σ_*^2 の値によって，$\bar{x}$ のバイアス（BIAS）や平均二乗誤差の平方根（RMSE）がいかに影響を受けるかを考察する。$\bar{x}$ のバイアス（BIAS）や平均二乗誤差の平方根（RMSE）を以下のように定義する。

$$\mathrm{BIAS}=\frac{1}{G}\sum_{i=1}^{N}\bar{x}^{(i)},\quad \mathrm{RMSE}=\frac{1}{G}\sum_{i=1}^{N}(\bar{x}^{(i)})^2$$

ただし，$G=1000$ とする。RMSE は，N が大きくなるにつれて，ゼロに近づくため，$\sqrt{N}\times$RMSE によって，比較を行う。BIAS はゼロに近いもの，$\sqrt{N}\times$RMSE は 1 に近いものが望ましい値である。

表 1－3 のＩＳ，ＲＳ，MHはそれぞれ，重点的サンプリング，棄却法，Metropolis-Hastings アルゴリズムによる結果を意味する。N は生成した乱数の数を表し，Metropolis-Hastings アルゴリズム（MH）の場合は N の20%の乱数を余分に生成し，後半の N 個の乱数で分析を行った（すなわち，$1.2N$ 個の乱数を生成し，後半の N 個の乱数を使って，BIAS，RMSE を求めた。ただし，表 3 の計算時間は $1.2N$ 個の乱数の生成による計算時間である）。また，表 1－3 のＲＳの項目の“－”は，棄却法が適用できない場合（すなわち，γ が存在しない場合）を表す。表 3 では，ＣＰＵが DX2 66Mz の DOS/V パソコンと Watcom FORTRAN 77 Version 10.6（Sybase，Inc.）の FORTRAN コンパイラを用いて，$G=1000$ 回の計算時間（単位は秒）を表す。

表から得られる結論は以下の通りである。

表 1 は BIAS を表し，ゼロに近いものが望ましい。$\mu_*=0.0$ のときは，σ_* の値にかかわ

表1：BIAS による比較

μ_* \ σ_*		N	0.5	1.0	1.5	2.0	3.0	4.0
0.0	IS	50	.012	.007	.004	.002	−.001	−.004
		100	−.024	.003	.003	.002	.000	−.001
		200	−.018	.001	−.001	−.001	−.001	−.001
		500	−.005	.001	.000	.001	.001	.001
		1000	−.009	.001	.000	.001	.001	.001
	RS	50	−	−	.007	.004	.000	−.002
		100	−	−	.000	.000	.002	−.001
		200	−	−	−.001	.001	.002	−.001
		500	−	−	.000	−.001	−.001	−.002
		1000	−	−	−.001	−.001	−.001	.000
	MH	50	.010	.011	.014	.014	.009	.010
		100	.006	.006	.007	.007	.005	.005
		200	.000	.000	−.001	.000	.000	−.001
		500	−.001	−.001	−.002	−.001	−.001	−.002
		1000	−.001	−.001	−.002	−.002	−.001	−.001
1.0	IS	50	.214	.007	.007	.007	.010	.010
		100	.061	−.003	.003	.004	.005	.005
		200	.062	−.002	.002	.002	.004	.006
		500	.136	.002	.001	.002	.003	.004
		1000	.104	.002	.001	.001	.001	.002
	RS	50	−	−	.003	.002	.005	−.001
		100	−	−	−.002	−.003	.002	.002
		200	−	−	−.002	−.002	.001	−.001
		500	−	−	−.001	−.001	−.001	−.001
		1000	−	−	−.001	.001	−.001	−.001
	MH	50	.147	.382	.369	.230	.112	.066
		100	.141	.374	.361	.225	.104	.060
		200	.133	.368	.354	.216	.095	.053
		500	.132	.367	.354	.217	.096	.054
		1000	.132	.366	.354	.217	.095	.053
2.0	IS	50	.398	.021	.007	.004	.004	.013
		100	.316	−.022	−.002	.001	.002	.006
		200	.327	−.026	.001	.002	−.001	.003
		500	.363	.003	.002	.000	.001	.002
		1000	.331	.003	.002	.000	.000	.000
	RS	50	−	−	−.008	−.003	−.003	.005
		100	−	−	−.005	−.005	−.002	.001
		200	−	−	−.004	−.005	−.004	−.003
		500	−	−	−.002	−.002	−.001	−.001
		1000	−	−	−.001	−.002	.000	−.001
	MH	50	.051	.047	.225	.278	.178	.116
		100	.045	.042	.221	.274	.174	.110
		200	.039	.036	.215	.268	.167	.103
		500	.038	.035	.216	.266	.166	.101
		1000	.037	.035	.215	.266	.166	.101
3.0	IS	50	.269	.184	−.006	.008	−.001	−.004
		100	.419	−.046	−.013	−.002	.001	−.002
		200	.420	−.064	−.004	.001	.001	−.005
		500	.345	.056	.003	.001	−.001	−.001
		1000	.375	.021	.005	.000	.001	.000
	RS	50	−	−	−.009	.000	−.005	−.006
		100	−	−	.000	.000	−.005	−.004
		200	−	−	.000	−.004	−.001	.000
		500	−	−	.001	−.001	.002	.001
		1000	−	−	.000	.000	.000	.000
	MH	50	.014	.012	.057	.185	.205	.148
		100	.009	.006	.052	.181	.201	.144
		200	.003	.000	.047	.178	.195	.137
		500	.002	−.001	.045	.178	.194	.137
		1000	.001	−.001	.045	.177	.194	.137

表2：$\sqrt{N}\times$RMSE による比較

μ_0 \ σ_0		N	0.5	1.0	1.5	2.0	3.0	4.0
0.0	IS	50	5.316	1.012	0.886	0.942	1.109	1.244
		100	14.541	0.995	0.869	0.941	1.133	1.290
		200	14.100	1.004	0.901	0.967	1.105	1.247
		500	10.757	0.697	0.870	0.945	1.100	1.238
		1000	16.821	0.975	0.862	0.929	1.105	1.256
	RS	50	–	–	0.992	1.003	1.022	1.003
		100	–	–	0.993	1.007	1.007	1.001
		200	–	–	1.011	1.045	1.014	1.013
		500	–	–	0.991	0.993	1.028	1.005
		1000	–	–	1.002	0.980	0.990	1.001
	MH	50	0.913	0.979	1.232	1.212	1.154	1.117
		100	0.945	1.010	1.270	1.232	1.178	1.155
		200	0.949	1.022	1.284	1.261	1.184	1.145
		500	0.924	0.983	1.221	1.193	1.147	1.115
		1000	0.914	0.980	1.225	1.210	1.164	1.110
1.0	IS	50	7.600	2.285	1.094	0.996	1.066	1.218
		100	46.036	2.400	1.066	1.018	1.138	1.263
		200	39.541	2.379	1.055	1.020	1.160	1.282
		500	27.042	2.314	1.046	1.011	1.152	1.287
		1000	48.105	2.392	1.085	1.020	1.123	1.279
	RS	50	–	–	1.026	1.009	1.023	1.008
		100	–	–	1.014	0.994	1.014	0.996
		200	–	–	1.021	1.004	1.003	0.992
		500	–	–	1.003	0.999	1.001	0.992
		1000	–	–	1.044	1.007	1.010	0.992
	MH	50	1.111	1.089	1.022	1.106	1.102	1.090
		100	1.155	1.148	1.068	1.145	1.133	1.111
		200	1.154	1.165	1.073	1.151	1.173	1.145
		500	1.108	1.125	1.036	1.110	1.133	1.086
		1000	1.108	1.116	1.034	1.107	1.104	1.073
2.0	IS	50	1.904	8.105	1.914	1.324	1.219	1.251
		100	25.759	15.371	1.957	1.321	1.237	1.348
		200	20.530	15.071	1.917	1.311	1.221	1.329
		500	13.396	12.050	1.889	1.320	1.215	1.325
		1000	34.026	13.714	1.968	1.332	1.213	1.289
	RS	50	–	–	1.000	1.056	1.015	1.013
		100	–	–	0.992	1.000	1.002	1.005
		200	–	–	0.988	1.009	1.008	1.013
		500	–	–	1.006	0.990	1.026	1.027
		1000	–	–	0.986	0.997	0.988	1.003
	MH	50	1.059	1.016	0.886	0.920	1.050	1.065
		100	1.092	1.035	0.920	0.975	1.090	1.090
		200	1.112	1.051	0.918	0.963	1.112	1.122
		500	1.070	1.025	0.924	0.947	1.050	1.068
		1000	1.066	1.018	0.913	0.932	1.047	1.061
3.0	IS	50	7.587	14.461	4.828	1.974	1.447	1.432
		100	44.925	71.006	4.739	2.045	1.419	1.411
		200	38.752	63.265	4.690	2.064	1.461	1.411
		500	26.570	44.291	4.714	1.988	1.492	1.414
		1000	45.829	66.466	4.894	2.076	1.444	1.417
	RS	50	–	–	1.033	0.994	0.995	1.018
		100	–	–	1.052	0.988	0.968	0.987
		200	–	–	0.997	1.013	0.982	1.006
		500	–	–	1.001	1.001	0.960	0.994
		1000	–	–	0.968	0.030	0.978	1.012
	MH	50	0.991	0.978	0.956	0.878	0.957	1.025
		100	1.021	1.009	0.964	0.889	0.987	1.066
		200	1.032	1.021	0.982	0.915	0.979	1.065
		500	0.996	0.985	0.948	0.891	0.979	1.029
		1000	0.991	0.980	0.943	0.876	0.965	1.000

表3：計算時間による比較（単位：秒）

μ_0 \ σ_0		N	0.5	1.0	1.5	2.0	3.0	4.0
0.0	IS	50	2.91	2.69	2.86	2.86	2.91	2.91
		100	5.76	5.38	5.71	5.71	5.76	5.77
		200	11.53	10.77	11.42	11.48	11.54	11.53
		500	28.78	26.85	28.51	28.61	28.67	28.84
		1000	57.40	53.83	56.91	57.23	57.45	57.62
	RS	50	−	−	5.22	6.98	10.39	13.95
		100	−	−	10.49	14.00	20.88	27.90
		200	−	−	20.98	27.96	41.69	55.75
		500	−	−	52.46	69.92	104.19	139.58
		1000	−	−	104.80	139.57	208.39	278.04
	MH	50	5.17	4.94	5.17	5.16	5.11	5.16
		100	10.27	9.83	10.21	10.27	10.27	10.32
		200	20.88	19.72	20.43	20.54	20.54	20.54
		500	51.24	49.32	51.13	51.30	51.30	51.41
		1000	102.54	98.54	102.22	102.55	102.60	102.77
1.0	IS	50	2.85	2.86	2.86	2.86	2.86	2.86
		100	5.76	5.71	5.72	5.71	5.71	5.76
		200	11.53	11.48	11.42	11.42	11.47	11.54
		500	28.84	28.62	28.56	28.61	28.61	28.73
		1000	57.67	57.35	56.96	57.18	57.28	57.45
	RS	50	−	−	7.80	8.24	11.10	14.34
		100	−	−	15.55	16.48	22.19	28.73
		200	−	−	31.09	32.90	44.32	57.34
		500	−	−	77.78	82.12	110.51	143.30
		1000	−	−	155.44	164.22	221.14	286.82
	MH	50	5.05	5.17	5.10	5.05	5.11	5.16
		100	10.22	10.27	10.17	10.16	10.22	10.22
		200	20.33	20.54	20.27	20.27	20.38	20.49
		500	50.86	51.30	50.64	50.64	50.91	51.08
		1000	101.72	102.60	101.28	101.39	101.83	102.21
2.0	IS	50	2.92	2.91	2.86	2.86	2.86	2.85
		100	5.83	5.71	5.77	5.72	5.77	5.77
		200	11.59	11.54	11.48	11.54	11.48	11.54
		500	28.89	28.83	28.67	28.72	28.73	28.78
		1000	57.78	57.62	57.40	57.39	57.40	57.51
	RS	50	−	−	25.65	13.51	13.34	15.87
		100	−	−	51.36	27.02	26.80	31.91
		200	−	−	102.66	53.99	53..50	63.49
		500	−	−	256.50	134.90	133.30	158.35
		1000	−	−	513.51	269.96	266.72	316.65
	MH	50	5.16	5.10	5.11	5.11	5.05	5.11
		100	10.22	10.27	10.27	10.22	10.21	10.22
		200	20.49	20.54	20.44	20.38	20.32	20.44
		500	51.30	51.36	51.14	50.86	50.75	50.97
		1000	102.54	102.71	102.61	101.72	101.56	102.05
3.0	IS	50	2.91	2.86	2.91	2.92	2.86	2.86
		100	5.77	5.83	5.76	5.76	5.71	5.76
		200	11.58	11.53	11.53	11.53	11.48	11.48
		500	28.89	28.89	28.83	28.78	28.73	28.78
		1000	57.89	57.78	57.61	57.62	57.45	57.51
	RS	50	−	−	188.45	30.98	18.13	18.73
		100	−	−	378.45	62.12	36.47	37.46
		200	−	−	755.88	123.91	72.89	74.70
		500	−	−	1892.79	309.78	182.41	187.19
		1000	−	−	3786.53	619.29	364.33	373.82
	MH	50	5.16	5.16	5.11	5.11	5.11	5.11
		100	10.27	10.27	10.21	10.27	10.16	10.21
		200	20.54	20.55	20.49	20.44	20.32	20.44
		500	51.30	51.35	51.24	51.19	50.81	50.98
		1000	102.54	102.71	102.44	102.28	101.56	101.94

らず，ＩＳ，ＲＳ，MHのいずれもBIASはゼロに近く良好な結果となった。しかし，ＩＳについては，μ_*が大きくなっても，σ_*を大きくすればBIASは減少するということが読み取れる。ＲＳはμ_*，σ_*の値にかかわらず，推定値のバイアスは存在しないと結論づけられる。MHについては，Nを増やせばBIASは小さくなっていくが，$N=500$と$N=1000$とではそれほどの差はなく，真の値への収束は非常に遅いと言える。しかも，μ_*とσ_*の値によっては，BIASがかなり大きい場合もある（例えば，$\mu_*=1.0$かつ$\sigma_*=0.5$, 1.0, 1.5, 2.0の場合，$\mu_*=2.0$かつ$\sigma_*=1.5$, 2.0, 3.0の場合，$\mu_*=3.0$かつ$\sigma_*=2.0$, 3.0, 4.0の場合）。

表2は$\sqrt{N}\times$RMSEを表し，1に近いものが望ましい。どの場合についても，ＲＳが最も1に近く，期待値の推定には最も適切な手法であると言える。ＩＳについては，σ_*が小さい（すなわち，$\sigma_*=0.5,1.0$）ときは極端に$\sqrt{N}\times$RMSEが大きく，乱数の数Nを増やしても，この値は減少しない。さらにＩＳについては，1に最も近づく場合は，μ_*とσ_*の値に依存しているため，μ_*とσ_*の選択に一般的な法則性を見つけることが難しい。MHについては，それほど極端に1から離れるものはなく良好な結果と言えるが，ＲＳと比較するとかなり劣る。

表3の計算時間については，ＩＳが最も計算時間がかからない方法である。ＩＳ，MHはNに比例するが，μ_*やσ_*には依存しない（すなわち，サンプリング密度関数には計算時間は依存しない）。しかし，ＲＳの場合，$\mu_*=0.0,1.0$で$f(x)$の期待値に近いときは，σ_*が大きくなれば計算時間が増大するが，μ_*が$f(x)$の期待値から離れるにつれて，σ_*が大きくなると計算時間が短くなる。このように，$f_*(x)$が$f(x)$から離れるにつれて，計算時間は増大するという結果が得られる。よって，ＲＳについては，$f(x)$の期待値が未知である以上，$f_*(x)$の分散σ_*^2を大きくとっておく方が望ましいと言える。

4　おわりに

本稿では，一般的な連続型密度関数のもとで，モンテカルロ法を用いて3種類の期待値の評価方法を紹介し，その精度の比較を行った。重点的サンプリングとは，モンテカルロ積分によって一般的には評価不可能な積分値を求めるというものである。それに対して，棄却法やMetropolis-Hastingsアルゴリズムは共に任意の密度関数から直接乱数を生成し，乱数の単純平均によって期待値をもとめようとするものである。両者の違いは，前者がIndependence Monte Carloに対し，後者はMarkov Chain Monte Carloという点で異なる。換言すると，前者は任意の密度関数から互いに独立な乱数を生成する方法であるが，後者は収束先のみが正確な意味での乱数ということになる。よって，期待値を推定する際に，同じ精度の推定値を得るためには，Metropolis-Hastingsアルゴリズムは棄却法よりも多くの乱数を

必要とする。またさらに，棄却法はサンプリング密度関数に依存しないが[6]，Metropolis-Hastings アルゴリズムはサンプリング密度関数の選択に大きく影響される[7]。

モンテカルロ実験（表１－３）で得られた結論は，以下のとおりである。

(1) ＩＳ，ＭＨでは，サンプリング密度関数 $f_*(x)$ の選択の仕方によっては，推定値にバイアスが生じる場合がある。ＲＳについては，サンプリング密度関数に依存せずに，どの場合でもバイアスのない推定値を得ることが出来る。

(2) RMSE 基準では，μ_*，σ_* の値にかかわらず，ＲＳが最も良い推定値を与える。ＭＨについては，それほど極端に１から離れるものはなく良い結果と言えるが，ＲＳと比較するとかなり劣る。

(3) 最も計算時間がかからない方法はＩＳである。ＩＳ，ＭＨは N に比例するが，μ_* や σ_* に計算時間は依存しない。しかし，ＲＳの場合は，$f_*(x)$ が $f(x)$ から離れるにつれて，計算時間は増大する。

以上の結果から，ＲＳが最も良い推定値を与えるが，実行不可能な場合（すなわち，γ が存在しない場合）もあるということを心に留めておくべきである。よって，任意の場合に適用可能な方法としては，BIAS の大きさと RMSE の大きさから判断して，危険がより少ないという意味でＩＳよりむしろＭＨが好まれるであろう。しかし，ＭＨにしても，サンプリング密度関数の選択は重要な問題であるということを，最後に付け加えておく。

注

1　単純に，台形公式による数値積分を例にとると，k 次元で，積分区間の分割が N 個の場合，計算時間は N^k に比例する。よって，k が大きくなると，極端に計算量は増大する。本稿では，重点的サンプリング，棄却法，Metropolis-Hastings アルゴリズム等のような計算時間がそれほど次元 k に依存しないような方法を取り上げ比較する。

2　Metropolis-Hastings アルゴリズムの特殊ケースとして，Gibbs サンプリングという方法もある。Gibbs サンプリングについては，Arnold (1993)，Geman and Geman (1984)，Geweke (1996, 1997) を参照せよ。

3　$f(x)$ と $f_*(x)$ を $N(\mu, \sigma^2)$ と $N(\mu_*, \sigma^2_*)$ として，γ を求める。γ は以下のように変形される。

$$\begin{aligned}\gamma &\equiv \sup_x \frac{f(x)}{f_*(x)} \\ &\equiv \sup_x \frac{\sigma_*}{\sigma}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2+\frac{1}{2\sigma_*^2}(x-\mu_*)^2\right) \\ &= \sup_x \frac{\sigma_*}{\sigma}\exp\left(-\frac{1}{2}\frac{\sigma_*^2-\sigma^2}{\sigma^2\sigma_*^2}\left(x-\frac{\mu\sigma_*^2-\mu_*\sigma^2}{\sigma_*^2-\sigma^2}\right)^2-\frac{(\mu-\mu_*)^2}{\sigma_*^2-\sigma^2}\right)\end{aligned}$$

γ が存在するためには，x の範囲は $-\infty<x<\infty$ なので，$\sigma^2_*-\sigma^2>0$ である必要がある。このとき，

$$\gamma=\frac{\sigma_*}{\sigma}\exp\left(-\frac{(\mu-\mu_*)^2}{\sigma_*^2-\sigma^2}\right)$$

となる。

4 Metropolis-Hastingsアルゴリズムの(i)−(v)によると，x_iとx_{i+m}との相関は，mが大きくなるにつれて，ゼロになることは容易に理解できることであろう。

5 重点的サンプリング，棄却法，Metropolis-Hastingsアルゴリズムの3つとも，同一のサンプリング密度関数を採用する。特に，Metropolis-Hastingsアルゴリズムでは，$f_*(x\mid z)=f_*(x)$となり，Independence Chainと呼ばれるサンプリング密度関数となっている。

6 しかし，サンプリング密度関数の選択によっては，γが存在しない場合もあるので，この意味においては，棄却法はサンプリング密度関数に依存すると言える。γが存在しさえすれば，棄却法による乱数は正確な意味での乱数となるので，サンプリング密度関数には依存しないことになる。

7 サンプリング密度関数の選択によって，収束のスピードが大きく異なる。例えば，十分大きいNについて，x_NがMetropolis-Hastingsアルゴリズムによって生成したい密度関数$f(x)$からの乱数になっているとする。このとき，x_{N+1}もまたMetropolis-Hastingsアルゴリズムを実行することによって$f(x)$からの乱数になっている。しかし，x_Nはx_{N+1}と相関を持つので，Metropolis-Hastingsアルゴリズムによる乱数は互いに独立ではない。この点が棄却法とは異なる。

参 考 文 献

Arnold, S.F., 1993, "Gibbs Sampling," in *Handbook of Statistics*, Vol.9, edited by Rao, C.R., pp.599−625, North-Holland.

Boswell, M.T., Gore, S.D., Patil, G.P. and Taillie, C., 1993, "The Art of Computer Generation of Random Variables," in *Handbook of Statistics*, Vol.9,edited by Rao, C.R., pp.661−721, North-Holland.

Chib, S. and Greenberg, E., 1995, "Understanding the Metropolis-Hastings Algorithm," *The American Statistician*, Vol.49, No.4, pp.327−335.

Chib, S. and Greenberg, E., 1996, "Markov Chain Monte Carlo Simulation Methods in Econometrics," *Econometric Theory*, Vol.12, No.4, pp.409−431.

Geman, S. and Geman, D., 1984, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. Pami-6, No.6, pp.721−741.

Geweke, J., 1988, "Antithentic Acceleration of Monte Carlo Integration in Bayesian Inference," *Journal of Econometrics*, Vol.38, pp.73−90.

Geweke, J., 1989a, "Modeling with Normal Polynomial Expansions," in *Economic Complexity: Chaos,Sunspots,Bubbles and Nonlinearity*, edited by Barnett, W.A., Geweke, J. and Shell, K., Cambridge University Press.

Geweke, J., 1989b, "Bayesian Inference in Econometric Models Using Monte Carlo Integration," *Econometrica*, Vol.57, pp.1317−1339.

Geweke, J., 1996, "Monte Carlo Simulation and Numerical Integration," in *Handbook of Computational Economics*, Vol.1, edited by Amman, H.M., Kendrick, D.A. and Rust, J., pp.731－800,North-Holland.

Geweke, J., 1997, "Posterior Simulators in Econometrics," in *Advances in Economics and Econometrics: Theory and Applications*, Vol.3, edited by Kreps, D. and Wallis, K.F., pp.128－165,Cambridge University Press.

Knuth, D.E., 1981, *The Art of Computer Programming*, Vol.2 (Seminumerical Algorithms, 2nd ed.), Addison-Wesley.

Koop, G., 1994, "Recent Progress in Applied Bayesian Econometrics," *Journal of Economic* Surveys, Vol.8, No.1, pp.1－34.

O'Hagan, A., 1994, *Kendall's Advanced Theory of Statistics*, Vol.2B (Bayesian Inference), Edward Arnold.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P., 1992, *Numerical Recipes in FORTRAN* (*Second Edition*), Cambridge University Press.

Shao, J., 1989, "Monte Carlo Approximations in Bayesian Decision Theory," *Journal of the American Statistical Association*, Vol.84, No.407, pp.727－732.

Tierney, L., 1994, "Markov Chains for Exploring Posterior Distributions," *The Annals of Statistics*, Vol.22, No.4, pp.1701－1762.