ベイズ統計学と MCMC
～メトロポリス・ヘイスティングス法の matlab による実現～

林　田　　　実

１　はじめに

筆者がベイズ統計学に最初に興味を持ったのは、1 次元 2 値回帰モデル問題をあざやかな手法で解いた例を紹介した、坂元(1985)にさかのぼる。そこでは、神無川の１年間の降雨の記録から極めて自然な、降雨確率を推計しており、ベイズ的手法の卓越性と、誤解を恐れずに言えば、一種の「おもしろみ」に非常に感銘を受けた。その後、ベイズ統計学の大家であるシカゴ大学の A. Zellner 教授の下での在外研究を経て、筆者の関心は「金融税制が金融市場に与える影響の実証分析」に移ったため、しばらく、ベイズ統計学そのものの研究からは遠ざかっていたが、北九州市立大学経済学会で晴山教授および山崎教授の退官記念号を出版する運びとなったこの機会を利用して、ベイズ統計学のホットな話題である MCMC（マルコフ連鎖モンテカルロ法）について書いてみることにした。MCMC は周知のようにコンピュータの強力な計算能力、シミュレーション能力を利用した推定方法である。しかしながら、昨今のコンピュータの能力をもってしても、プログラミングの善し悪しで、推計が成功したり、失敗したりすることがある。そのため、ネット上で公開されている MCMC のプログラムは、得てして、難解な場合が多い。そこで、本論では、MCMC の理論的な枠組みに忠実でありながら、同時に、プログラムとしての視認性の良さを同時に追求したプログラムを作成し、その実行結果についても紹介していきたいと思う。

２　ベイズ推定

ここで、ベイズ推定の概略を述べておきたい。そのためには、伝統的な非ベイズ統計の紹介から始めるのが順当であろう。例えば、表が出る確率θのコインを 1 回投げる試行を考える。さらに、コイン投げの結果、表が出た場合１を、裏が出た場合０をとる確率変数 X を考える。この時、X=1 となる確率はθ、X=0 となる確率は(1-θ)であるから、X が x(=0 or

1)となる確率はまとめて、次のように書くことができる。

$$\theta^x(1-\theta)^{1-x}$$

この式は、x を所与として考えれば、θの関数と見ることもできるので、これを尤度と言う。伝統的な統計学の主要な推定理論では、この尤度を最大にするようにθ(パラメータ)を決定する（最尤法)。この手続きを n 回のコイン投げで考えてみよう。n 回のコイン投げの結果、X 回表が出たとする。X が x(=0,1,・・・,n)という値をとる確率はよく知られているように二項分布に従い、

$$\binom{n}{x}\theta^x(1-\theta)^{n-x}$$

となる。この式は前述のように x を所与と考えれば(n は当然、所与)、θの関数であって、尤度である。最尤法の手続きによれば、この尤度を最大化すれば良いのであるから、式２をθについて最大化する。その結果、θ=x/n の時に尤度が最大となることが分かる(θについての点推定)。θ=x/n という推定結果は、結局、n 回のコイン投げで表が出た割合を示しているから、直感的にも納得のいく結論となっている。

次に、θの点推定値 x/n はどのような性質をもっているのであろうか。ここで、「性質」と言ったが、x/n は確定値なのだから、性質も何もあったものではないと思いがちである。しかし、それでは、話が終わってしまう。そこで、次のように考える。x はもともと確率変数 X の実現値であった。したがって、x/n ではなく、X/n を考えれば、その性質についていろいろ議論ができるわけである。周知のように X/n は独立同一分布から発生したデータの標本平均でもあるので、n がある程度大きければ(この場合 30 もあれば十分である)、中心極限定理が使えて、X/n は平均がθ、分散がθ(1-θ)/n の正規分布に従うと考えて良い。そうであれば、X/n をθの点推定量とした場合、その期待値はθであること（不偏性）が分かるし、有意性検定（例えば、θ=1/2 が正しいか否かの検定）も構築することができる。

以上のことを一般的に書いておこう。θが与えられたとき X=x となる確率を p(x|θ)と表記する。これは、x を所与と考えれば尤度でもある。尤度を最大にするθ=$\hat{\theta}$は x の関数であるから、$\hat{\theta}(x)$と書ける。さらに x がもともと確率変数であったことを想起すると$\hat{\theta}(X)$であるから、$\hat{\theta}(X)$も確率変数となり、種々の性質を持つ。この性質を用いて推定、検定を行うというわけである。

このように伝統的な統計学ではθは確定値であって、確率変数とは全く考えられていない。しかしながら、ベイズ統計学ではθが確率変数であることを議論の出発点にする。それ故、今日までベイズ統計学と伝統的統計学の間では哲学的な論争が絶え間なく行われているが本稿では、その点には深く立ち入らない。ともかく、θが確率変数であることを受け入れて先に進もう。θは確率変数であるから、データ X との同時分布が存在するはずである。そして、この同時分布は条件付き分布を用いて当然、次のように表記できる。

$$p(x,\theta)=p(x|\theta)p(\theta)=p(\theta|x)p(x)$$

この式から、ただちに次式を得る。

$$p(\theta|x) = p(x|\theta)p(\theta)/p(x) \propto p(x|\theta)p(\theta)$$

ここで、p(θ)を事前分布、p(θ | x)を事後分布、p(x | θ)を尤度、p(x)を周辺尤度と呼ぶ。ベイズ統計学では、θに関する事前分布（分析者が持つ事前の予想）を出発点にして、データを眺め、事前分布に修正を加え、それを事後分布という形で表現していることになる。ここで、x はもはや、確率変数として考える必要がないことに注意されたい（事後分布は x が与えられたときのθの条件付き分布である。）また、周辺尤度は確定値であるから、事後分布は事前分布×尤度に比例しており、後述する事後分布からのサンプリングでは周辺尤度は無視して良い。事後分布は通常の確率分布であるから、これを直接用いて、θに関する種々の推定を行う（例えば、事後分布のモードや平均でθの点推定値とする）。従って伝統的統計学の$\hat{\theta}$(X)の分布とベイズ統計学の事後分布が 2 つの体系において、相対的に同一な場所を占める概念であることが分かる。ちなみに、特定の事前分布を持ち込むことに懐疑的であれば、θに一様分布を仮定すれば良い[1]。その意味で、筆者は、ベイズ統計学は伝統的統計学を包含するものであると考えている。また、事前分布を想定することは、経済学などの社会科学ではむしろ無意識的に行われていると筆者には思われる。従って、ベイズ統計学と社会科学との相性は良いのではないかとも考えている[2]。

こうして、ベイズ統計学では、自然な事前分布に対して、事後分布をもとめさえすれば良いことが分かるが、実際には、この事後分布を解析的に求めることができないことが多い。そのような場合に適応する手法として編み出されたものが MCMC である。次節では、MCMC の中でも少し高度なメトロポリス・ヘイスティングス法（M-H 法）をとりあげる。

３　メトロポリス・ヘイスティングス法（M-H 法）によるベイズ推定

MCMC の内容に入る前に、ベイズ統計学、伝統的統計学に共通して利用できる乱数を使った推定法について述べておく。再び、表の出る確率がθであるようなコインを n 回投げてθに関する推定を行う場合を考える。前述のように、表の出る回数を X とすれば、θは$\hat{\theta}$=X/n で点推定できる。従って、$\hat{\theta}$(X)の確率分布がこの推定問題の核心である。2 節では$\hat{\theta}$の分布が、中心極限定理により、正規分布になることを指摘した。ところで、もし中心極限定理を知らない分析者がいたとすれば、彼には$\hat{\theta}$の分布を知る他の術はないのであろうか。コンピュータの出現以前は、その回答は否定的なものであったが、現在はそうではない。

---

[1] n 回のコイン投げの実験では、θに事前分布として一様分布を仮定すると、θはベータ分布に従うことが分かる。その平均は(x+1)/(n+2)で、標本が大きい場合には x/n に一致する。

[2] 例えば、最近、脚光を浴びている論文として、Smets F. and Wouters R.(2003)がある。これは、従来カリブレーションと言われていたパラメータに関する設定をベイズ統計学の枠組みに取り込んだものである。

統計学の素人であっても$\hat{\theta}(X)$の分布のおおよその形はすぐに求めることができる。そのやり方（シミュレーション）の原理はシンプルである。すなわち、真のパラメータθを適当な値に設定し、実際に n 回のコインを投げてみるのである。そして、その結果として出た表の回数 X の記録をとる。これで、1 個の X/n の値を得る。さらに、もう一度、n 回、コインを投げて、もう 1 個の X/n の値を得る。これを繰り返すと無数の X/n の実現値を得ることができる。この無数の X/n の実現値から得られるヒストグラムは、ほぼ X/n の確率分布を再現していると考えて良い。無論、実際にコインを投げることは必要ではなく、全てはコンピュータによって乱数を発生させて、実現することができる。このようなシミュレーションの本質は、$\hat{\theta}(X)$の確率分布を知りたいときに、$\hat{\theta}(X)$の分布からのサンプリング（ここでは、実際に n 回のコインを投げて、表の出る割合を計算できること）が行えれば、良いことがわかる。このサンプリングができれば、それからヒストグラムを描くことは容易である。

さて、伝統的統計学の$\hat{\theta}(X)$の確率分布に相当するベイズ統計学のそれは、事後分布であった。したがって、ベイズ統計学では、事後分布からのサンプリングが可能であれば、事後分布を用いた統計的推定は原理的に全て実行可能である。無論、事後分布の形が解析的にもとまるのであれば、あえてシミュレーションをする必要はない。これは、中心極限定理によって$\hat{\theta}(X)$の分布が分かっていればシミュレーションをする必要がないことと同値である。しかしながら、分析者の自由な発想に基づく事前分布を仮定すると事後分布はすぐに解析的に導出不能となってしまう。このような時に、事後分布からのサンプリングを行って、活躍する手法が MCMC ということになる。

まず、平均μ、分散 1 の正規分布から 10 個の独立なデータが得られており、その標本平均が 5.0981 であるとする[3]。課題はμについての推定である。まず、μに関する尤度は平均 5.0981、分散 1/10 の正規分布と一致することはすぐに分かる。次に、事前分布とし自由度 5 の t 分布を仮定する。従って、事後分布は正規分布と t 分布との積からなることが分かる。Matlab の関数を使うと以上のことは以下の 3 つの関数で簡潔に記述できる。

```
%----------------------------------------------------------------------------------------------------

% μの尤度関数
function y = likelihood( myu, data ) ;

    y = ( 1 / sqrt( 2*pi* ( 1/data(2))  ) ) * exp( -(1/2) * ( myu - data(1) ) ^ 2 / (1/data(2)) ) ;

end

%----------------------------------------------------------------------------------------------------

% μの事前分布～t(5)
function y = prior( x ) ;

    y = pdf( 't', x, 5 ) ;

end
```

[3] この例は伊庭他『計算統計Ⅱマルコフ連鎖モンテカルロ法とその周辺』岩波、177-179 頁、例 7 を用いた。

```
%-----------------------------------------------------------------------------------------------
% パラメータ x に関する事後分布
function y = posterior( x, data ) ;

    y = prior( x ) * likelihood( x, data ) ;

end
```

ここで、尤度（関数 likelihood）では引数が「myu」のパラメータと「data」なるデータとなっていることに注意されたい。なぜなら、尤度はパラメータとデータで完全に記述できるからである。これは、事後分布（関数 posterior）についても同様である。なお、プログラム中の data(1)は 5.0981 を、data(2)は 10 を意味する。

さて、このようにして求められた事後分布から直接サンプリングできるのであれば、問題は容易であるが、本問ではそれが困難である。そこで、メトロポリス・ヘイスティングス法(M-H 法)を用いる。M-H 法の概略はこうである。すなわち、事後分布にできるだけ近いサンプリング可能な分布（提案分布）からサンプリングを行い、別に定められた採択確率を用いて、このサンプリング結果を採択、あるいは棄却する。このようにして得られた一連のサンプル系列は、一定の期間を経ると（burn-in)、事後分布からのサンプリングとみなして良いことが知られている[4]。そこで、提案分布として、平均 5.0981、分散 1/10 の正規分布を利用してみよう。以下に、提案分布からのサンプリングプログラム（関数 proposal_ran）と提案分布の密度関数（関数 proposal_pdf）を与えるプログラムを記す。

```
% 提案分布。これは、事後分布がμの事前分布（自由度５のｔ分布）と N(μ, 1/n)との積になっていることから選択
% この提案分布は myu0 に依存しないので、独立連鎖の一例である。

function y = proposal_ran( myu0 ) ;

    global data ;

    y = data( 1 ) + sqrt( 1/data(2) ) * randn( 1 ) ;
end

%-----------------------------------------------------------------------------------------------
% 提案分布の密度関数。この提案分布は直前のμのサンプリング値 myu0 とは独立に定義されている（独立連鎖）。
function y = proposal_pdf( myu0, myu_prime ) ;

    global data ;

    y = 1 / sqrt( 2 * pi / data(2) ) * exp( -(1/2) * ( myu_prime - data(1) ) ^ 2 / ( 1/data(2) ) ) ;

end
```

提案分布からのサンプリングプログラムは引数 myu0 が与えられている。この myu0 は提案分布によって、サンプリングされた 1 期前の結果値を示している。このように、一般には提案分布からのサンプリングはその直前のサンプリング結果に依存するように設定されることが多い。しかしながら、プログラム中にこの myu0 が使われていないことからもわかるように、本問題の提案分布からのサンプリングは過去のサンプリング値に依存しない。これを独立連鎖という。次に、提案分布の密度関数を関数 proposal_pdf に示す。引数は、

[4] 例えば、Chib(2001)を見よ。

myu0 と my_prime である。myu0 についてはすでに述べた。myu_prime は提案分布からサンプリングされた結果値を示す。この両者を引数として、密度関数 y が定義されている。この提案分布は独立連鎖を用いているので、密度関数の定義に myu0 が出てきていないことに注意されたい。

最後に、提案分布からサンプリングされた結果をある一定の採択確率で取捨選択するプログラムを示す。

```
%-----------------------------------------------------------------------------------------
% MH アルゴリズムによるサンプリング
function y = selection( myu0, myu_prime, data ) ;
    u = rand( 1, 1 ) ; % (0,1)区間の一様乱数
    bunsi = posterior( myu_prime, data ) * proposal_pdf( myu_prime, myu0 ) ;
    bunbo = posterior( myu0    , data)  * proposal_pdf( myu0, myu_prime ) ;
    selection_p = min( 1, bunsi / bunbo ) ;
    if u < selection_p
        y = myu_prime ;
    else
        y = myu0     ;
    end ;
end
```

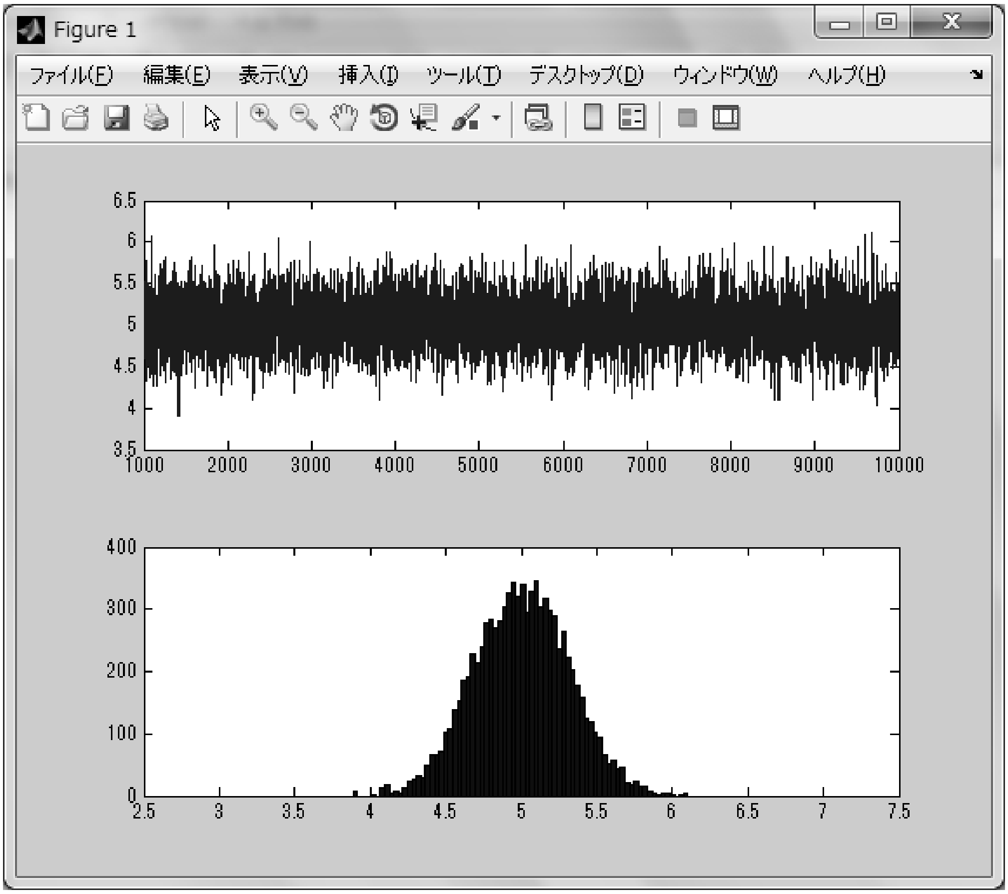
関数 selection は提案分布からの一期前のサンプリング結果 myu0 と今回のサンプリング結果 myu_prime およびデータを引数としている。データは採択確率で事後分布を使う関係から本質的である。そして、今回のサンプリング結果を採択確率で採択し出力する。もし、採択しない場合は、一期前のサンプリング結果が出力される。なお、採択確率は以下のように定義されている。ここではあえて、matlab によるコーディングで示す。最後の selection_p が採択確率である。

```
bunsi = posterior( myu_prime, data ) * proposal_pdf( myu_prime, myu0 ) ;

bunbo = posterior( myu0     , data ) * proposal_pdf( myu0, myu_prime ) ;
selection_p = min( 1, bunsi / bunbo )
```

このようにして、１万回のサンプリングを行い、burn-in として 1000 回のサンプリング結果を捨て去って得られたサンプリング系列から得られた、時系列のプロットとヒストグラムは以下のようになる。なお、全体を統括する関数プログラムは付録１に掲載している。

図１　独立連鎖による M-H アルゴリズムの結果（事前分布はｔ分布）



さて、先の例では提案分布に正規分布を設定しており、事後分布のサンプリング結果もそれを強く反映しているように思われるので、M-H 法と言っても、全く意外性はないように感じられる。そこで、提案分布にランダムウォークを使った例を考えてみる[5]。提案分布を変えるだけであるから、尤度、事前分布、事後分布を与える matlab 関数は全て同一である。提案分布は以下のように、ランダムウォークに従い、誤差項に平均０、分散 tau_square=0.5 を持つ正規分布を用いた（関数 proposal_ran_rw）。

```
%----------------------------------------------------------------------------------------------
% 提案分布。これは、新しいサンプルが酔歩過程から生じるとしている（酔歩連鎖）。
function y = proposal_ran_rw( myu0 ) ;
    global tau_square ;
     y = myu0 + sqrt( tau_square ) * randn( 1 ) ;
end
```

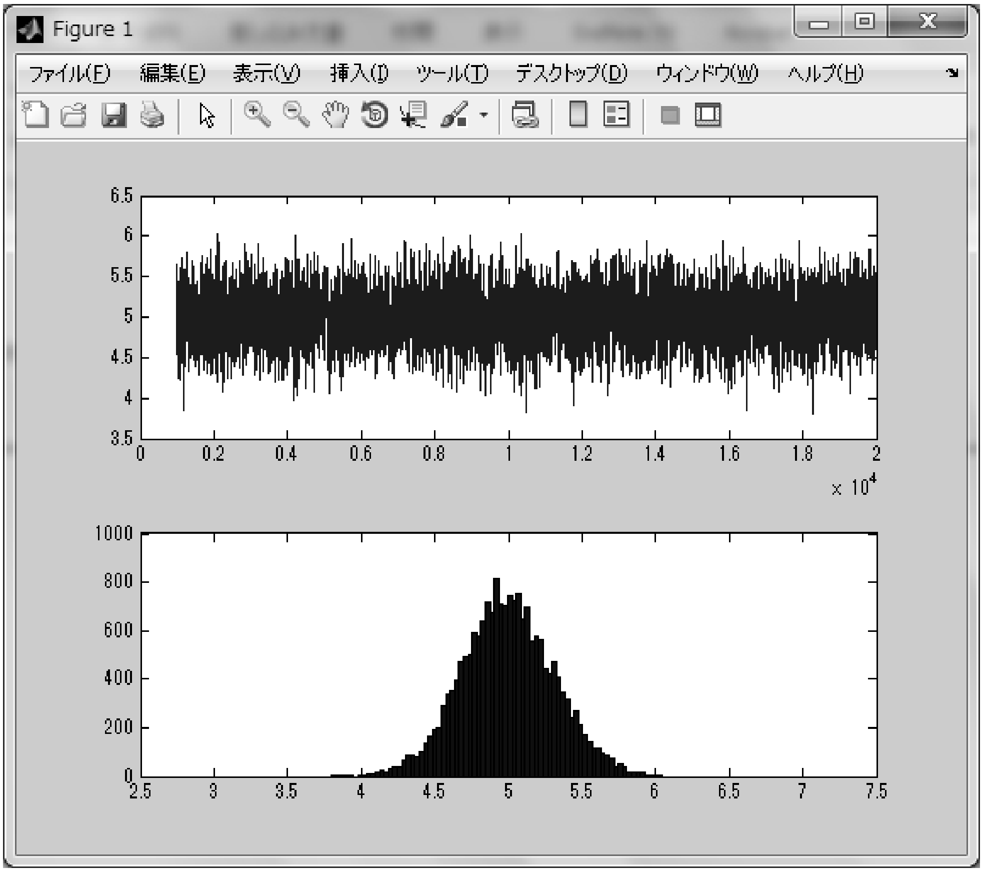
[5] 伊庭他(2005) 179-181 頁参照。

これに対応した提案分布の密度関数は以下のようになる（関数 proposal_pdf_rw）。

```
%-----------------------------------------------------------------------------------------------
% この提案分布は直前のμのサンプリング値 myu0 とは独立ではなく、ランダムウオークに従う（酔歩連鎖）。
function y = proposal_pdf_rw( myu0, myu_prime ) ;
    global tau_square ;
    y = 1 / sqrt( 2 * pi / tau_square ) * exp( -(1/2) * ( myu_prime - myu0 ) ^ 2 / tau_square ) ;
end
```

最後に、提案分布からサンプリングされた結果をある一定の採択確率で取捨選択するプログラムは、以前のプログラムと同じで良い。ただし、ランダムウォークの誤差項の分散 tau_square はグローバル変数として扱っているので、「global tau_square ;」を挿入しておくこと。サンプリング回数を 20000 回とし、burn-in を 1000 回とした結果を次の図に示す[6]。

図２　酔歩連鎖による M-H アルゴリズムの結果（事前分布はｔ分布）



6 先の例と同様、全体を統括する matlab 関数を付録２に掲げた。

図２は、提案分布として、図 1 の正規分布と全く違った、ランダムウォークを使っているにもかかわらず、ほとんぼ同じものとなっていることに注意されたい。

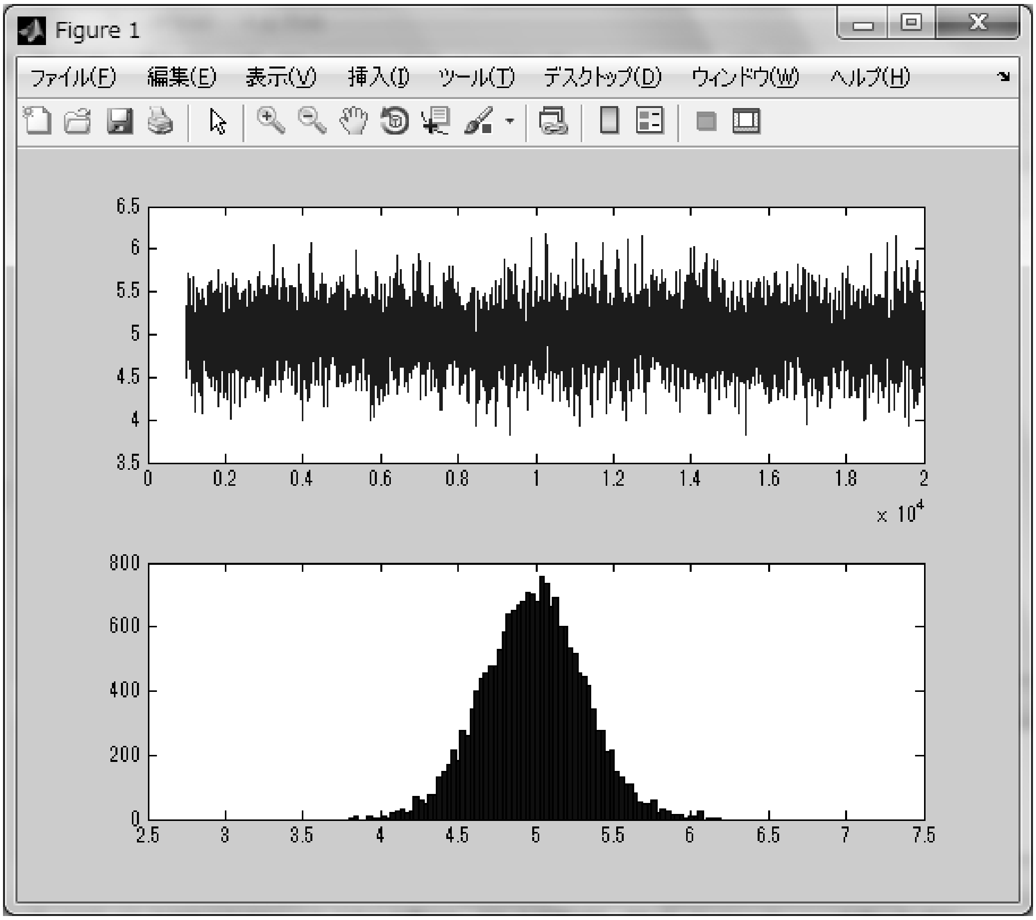
最後に、ランダムウォークと言っても、1 期前のサンプリング結果値を平均とし、分散 0.5 の正規分布に従う確率分布が提案分布となっているにすぎず、その意味で提案分布が正規分布の呪縛から逃れていないのではないかと危惧する方のために、ランダムウォークの誤差項が区間(-0.5, 0.5)の一様分布に従う例を紹介しよう。これまでの例と異なるのは提案分布からのサンプリングプログラムと提案分布の密度関数を与えるプログラムである。それらはそれぞれ、以下のようになる。

```
%----------------------------------------------------------------------------------------------------
% 誤差項が一様分布に従う酔歩連鎖の提案分布。これは、新しいサンプルが酔歩過程から生じるとしている（酔歩連鎖）。
function y = proposal_ran_rw_unif( myu0 ) ;
     y = myu0 + rand(1,1) - 0.5;
end

%----------------------------------------------------------------------------------------------------
% この提案分布の密度は一定値となる。
function y = proposal_pdf_rw_unif( myu0, myu_prime ) ;
     y = 1;
end
```

提案分布の密度関数を与える関数「proposal_pdf_rw_unif」は提案分布が一様乱数であることを反映して、一定値になっている。このプログラムを実行した結果を次に示す。

図３　酔歩連鎖（誤差項は一様分布）による M-H アルゴリズムの結果（事前分布は t 分布）



事後分布の形状が見事に再現されており、M-H 法の強力さを示す好例となっている。

４　おわりに

本論では、比較的難易度が高いと思われる M-H 法（メトロポリス・ヘイスティングス法）を、視認性の良い matlab プログラムで実現可能であることを示した。そこでは、引数に本質的なパラメータ、データを明示的に取り入れることによって、何を計算するのに、何が必須であるのかが明瞭になるようにコーディングを行った。また、M-H 法の強力さが分かる例として、一様分布を誤差項に持つランダムウォークを提案分布にする例も示した。伊庭他（2005）には、この他にも興味深い具体例があげられている。今後は、同書の具体例を参考にしながら、順次、matlab によるコーディング例を追加していきたいと思っている。

付録１

図１独立連鎖による M-H アルゴリズムの結果（事前分布はｔ分布）のためのプログラム

```
% 以下は、MH アルゴリズム（独立連鎖）によるサンプリング例
% 参考にしたのは以下の文献、伊庭他『計算統計Ⅱ』岩波、177-179 例 7

function mh ;

x = [ 5.293437097;
     6.786756911;
     6.698467713;
     5.216369926;
     4.279878011;
     3.550147211;
     4.901963292;
     4.477433903;
     4.014444256;
     5.7623089  ]

% x_mean が 5.0981 になってる。
x_mean = mean( x )

% サンプル数
n = size( x, 1 )

% 繰り返し回数
draw_no = 10000 ;

% 捨てるサンプル数
burn_in = 1000  ;

% 以下、MH アルゴリズムによるサンプリング

global data ;

data = [ 5.0981  10 ] ;


myu0 = 0 ;
for i = 1 : draw_no ;

    myu_prime = proposal_ran( [] ) ;

    y(i) = selection( myu0, myu_prime, data ) ;

        if y(i) == myu0

            hantei(i) = 0 ;

        else

            hantei(i) = 1 ;

        end

    myu0 = y(i) ;

    time( i ) = i ;

end ;

saitakuritu = sum( hantei ) / ( draw_no - burn_in )


% 以下、サンプリング結果の図示
subplot( 2,1,1 )

plot( time( burn_in : draw_no ), y( burn_in : draw_no ) );


subplot( 2,1,2 )

hist( y( burn_in : draw_no ), 3:0.03:7 );

end
```

付録２

図２酔歩連鎖による M-H アルゴリズムの結果（事前分布は t 分布）のためのプログラム

```
% 以下は、MH アルゴリズム（酔歩連鎖）によるサンプリング例
% 参考にしたのは以下の文献、伊庭他『計算統計Ⅱ』岩波、179-181、例７続き
function mh ;

x = [ 5.293437097;
     6.786756911;
     6.698467713;
     5.216369926;
     4.279878011;
     3.550147211;
     4.901963292;
     4.477433903;
     4.014444256;
     5.7623089  ]

% x_mean が 5.0981 になってる。
x_mean = mean( x )

% サンプル数
n = size( x, 1 )

% 繰り返し回数
draw_no = 20000 ;

% 捨てるサンプル数
burn_in = 1000  ;

% 以下、MH アルゴリズムによるサンプリング
global tau_square  data ;

data = [ 5.0981  10 ] ;


myu0      = 0   ;
tau_square = 0.5 ;

for i = 1 : draw_no ;

    myu_prime = proposal_ran_rw( myu0 ) ;

    y(i) = selection_rw( myu0, myu_prime, data ) ;

        if y(i) == myu0

            hantei(i) = 0 ;

        else

            hantei(i) = 1 ;

        end

    myu0 = y(i) ;

    time( i ) = i ;

end ;

saitakuritu = sum( hantei ) / ( draw_no - burn_in )


% 以下、サンプリング結果の図示
subplot( 2,1,1 )

plot( time( burn_in : draw_no ), y( burn_in : draw_no ) );

subplot( 2,1,2 )

hist( y( burn_in : draw_no ), 3:0.03:7 );

end
```

参考文献

伊庭幸人、種村正美、大森裕浩、和合肇、佐藤整尚、高橋明彦『計算統計Ⅱマルコフ連鎖モンテカルロ法とその周辺』岩波書店、2005

坂元慶行『カテゴリカルデータのモデル分析』共立出版株式会社、1985

Chib, S., "Markov Chain Monte Carlo Methods: Cmputation and Inference," in Heckman, J. and Leamer, E. (eds.), *Handbook of Econometrics*, 2001, vol.5, North-Holland, pp.3569-3649.

Ishiguro, M. and Sakamoto, Y., "A Bayesian Approach to Binary Response Curve Estimation," *Ann. Inst. Statist. Math.*, 1983, 35 Part B, pp.115-137.

Smets F. and Wouters R., "An Estimated Dynamic Stochastic General Equilibrium Model of the Euro Area," *Journal of the Europian Economic Association*, September 2003, 1(5), pp.1123-1175.