ガンマ乱数生成のアルゴリズムについて

谷崎久志（神戸大学大学院経済学研究科）

1990 年代になってから，Gibbs Sampler や Metropolis-Hastings Algorithm などの Markov Chain Monte Carlo (MCMC) という乱数生成の手法がベイズ推定に応用されるようになりました．応用の手順としては，まず，パラメータに事前分布を仮定し，モデルから得られる尤度関数から，事後分布を求めます．そして次に，事後分布から直接乱数を生成して，その乱数を用いてパラメータに関する推論を行うというものです．MCMC の利用に伴い，非常に複雑なモデルでも実証分析が簡単に行えるようになり，応用範囲が爆発的に拡大しました．

私自身もこの手法をよく使います（と言っても，私の場合は似非(えせ)ベイジアンと呼ばれる部類の者です）．ベイズ推定を行うと，多くの場合，ガンマ乱数が必要になってきます．単純な線形回帰モデルを例にすると，誤差項に正規分布を仮定すれば，分散の事後分布は逆ガンマ分布になります（これは分散の事前分布の仮定にも依存します）．ガンマ乱数の逆数が逆ガンマ乱数なので，このためにガンマ乱数生成のアルゴリズムが必要になります．ベイズ推定以外の他の例としても，ガンマ乱数が生成できれば，自由度が正の整数以外（すなわち，正の実数）の t 分布やカイ 2 乗分布の乱数生成も行うことができるようになります．

ご存知の通り，ガンマ分布とは密度関数が $f(x)=x^{\alpha-1}\exp(-x)/\Gamma(\alpha)$ となる分布です(ただし，$x>0$，$\alpha>0$)．αはシェイプ・パラメータ（Shape Parameter）と呼ばれるパラメータです．過去の研究から，いくつか代表的なガンマ乱数生成のアルゴリズムを調べていたところ，α が 1 より小さいか大きいかによって，アルゴリズムが異なるということが分かりました．その理由としては，αの値によって $f(x)$ の関数形が変化するからです．すなわち，α が 1 以上の場合は $f(x)$ は一つの山を持ち最大値が存在しますが，α が 1 より小さい場合は x が 0 に近づくにつれて $f(x)$ は無限大になります．伏見正則著『乱数』(1989，東京大学出版会）には，両方のケースについて簡単なアルゴリズムが紹介されています．$0<\alpha<1$ の場合は Best（1983，*Computing*, Vol.30，pp.185-188)，$\alpha\geqq1$ の場合は Cheng（1977，*Applied Statistics*，Vol.26，pp.71-75）がよいとされています（ただし，後者は簡単ではあるが最速なアルゴリズムではありません)．その他にも，$\alpha\geqq1$ の場合に関して，高速な正規乱数が生成できれば，Minh（1988，*ACM Transactions on Mathematical Software*，Vol.14，pp.261-266）の方法も有用です．

αの値によってアルゴリズムを使い分けるのは煩わしいので，すべての $\alpha>0$ について同じアルゴリズム，しかも，簡単なアルゴリズム，さらに，出来れば高速なアルゴリズムでガンマ乱数が生成出来ないものかを考えてみることにしました．その結果，これに関する論文は H. Tanizaki (2008)，“A Simple Gamma Random Number Generator for Arbitrary Shape Parameters,” *Economics Bulletin*, Vol.3，No.7，pp.1-10 に掲載されました．論文は http://economicsbulletin.vanderbilt.edu/2008/volume3/EB-07C10012A.pdf からダウンロード出来るようになっています．そこで考案されたアルゴリズムを以下に紹介したいと思います．

[アルゴリズム]

（ⅰ）αを与えたもとで，n，b_1，b_2，c_1，c_2 を次のようにセットする．$0.0<\alpha\leqq0.4$ のとき $n=1/\alpha$，$0.4<\alpha\leqq4$ のとき $n=1/\alpha+(\alpha-0.4)/(3.6\alpha)$，$4<\alpha$ のとき $n=1/\sqrt{\alpha}$ とおく．$b_1=\alpha-1/n$，$b_2=\alpha+1/n$ とする．また，$0.0<\alpha\leqq0.4$ のとき $c_1=0$，$0.4<\alpha$ のとき $c_1=b_1(\log(b_1)-1)/2$ とおく．さらに，$c_2=b_2(\log(b_2)-1)/2$ とする．

（ⅱ）区間 (0,1) で，2 つの独立な一様乱数 v_1，v_2 を生成する．そして，$w_1=c_1+\log(v_1)$，$w_2=c_2+\log(v_2)$，$y=n(b_1w_2-b_2w_1)$ を計算する．

（ⅲ）$y<0$ のとき（ⅱ）に戻る．

（ⅳ）$x=n(w_2-w_1)$ とおく．もし $\log(y)\leqq x$ ならば $\exp(x)$ をパラメータαのガンマ乱数として採用し，もし $\log(y)>x$ ならば（ⅱ）に戻る．

アルゴリズムをごく簡単に解説すると，X をガンマ分布に従う確率変数とするとき，$X=Y^n$ として Y の密度関数を求めると，$g(y)=ny^{n\alpha-1}\exp(-y^n)/\Gamma(\alpha)$ となります．そうすると，$\alpha\geqq 1/n$ について，$y\geqq 0$ の範囲で $g(y)$ は最大値を持つことになり，$f(x)$ の $\alpha\geqq 1$ のケースの手法をそのまま適用することが出来ます．上記のアルゴリズムでは $g(y)$ に Ratio of Uniforms という方法が採用されています．さらに，n を α の関数，すなわち，$n=n(\alpha)$ と考え，しかも，$\alpha\to 0$ のとき $n\to\infty$（または，$\alpha\to\infty$ のとき $n\to 0$）となるような何らかの減少関数とすれば，$\alpha>0$ の範囲で Y の乱数を生成することが出来るようになります．乱数生成が出来るだけ高速になるように選ばれたものが上記のアルゴリズムのステップ（ｉ）での α と n との関係（α の値によって，n と α との関係が異なる）ということになります．紙面の都合上，詳細は論文をご覧ください．

アルゴリズムの特徴は，(1) α の値によって乱数生成アルゴリズムを変更する必要はない，(2) 乱数生成のスピードは，既存のアルゴリズムに比べて，それほど遅くはない（決して高速とは言えないが，使い物になる程度のスピードはある），ということです．実は，もっとレベルの高い他の専門誌にも投稿しましたが，査読者の意見は極端に真っ二つに分かれて最終的には残念ながら棄却ということになってしまいました．反対意見としては，「複雑なアルゴリズムであろうと，一旦，誰かがプログラムを書いてしまえば，それを利用すれば済むことで，簡単なアルゴリズムを作る必要性はなく，最速なアルゴリズムの方がいいに決まっている」ということでした．これはこれで一理あるのですが，例えば，NAG数値計算ライブラリ（http://www.nag-j.co.jp/nag_lib.htm）やIMSL数値計算ライブラリ（http://www.vnij.com/products/imsl/index.html）等を使う場合は金銭的に費用がかかるということや引数等の使い方を調べるのが面倒なときがあったりします．また，一般的に高速なアルゴリズムはステップ数（すなわち，プログラム行数）が長いという傾向があるようで，他人の論文を見て自分でプログラムを書く場合は，特に，エラーの対処に時間がかかるときがあります（ときには，何回見直してもどこが間違えたのか分からないときもあります）．ここで紹介したアルゴリズムは，ステップ数も短く，既存の統計パッケージや統計ライブラリをそのまま利用するのではなく，1 から自分でプログラムを組むという方には有用かと思います．

というわけで，皆さん，一度，上記のガンマ乱数の生成アルゴリズムをお試しください．少なくとも授業の教材にはなると私は思っているのですが，いかがでしょうか．