統計数理研究所彙報
第29巻 第1号 (1981)

分布の逆関数の数値計算に関する一注意

東京工業大学 佐藤定夫

(1981年5月 受付)

まえがき

確率分布の数値計算において，その逆関数，いわゆる％点を求めることが，しばしば要請される．一般に逆関数の計算には，Newton 法，線型内挿法，補間法などが良く知られているが [1], [2], 実際の関数ではこれらの方法が使えないことがある．たとえば，Newton 法の場合，確率分布の導関数，すなわちその密度関数を求めなければならないが，しばしば，それはもとの分布関数を求めるより困難である．

筆者が，最近計算した例をあげると $F(z)$ は $z \geqq 0$ における分布関数で

$$F(z) = (1-\rho)^{\lambda} \sum_{m=0}^{\infty} \frac{\rho^m \Gamma(\lambda+m)}{m!\,\Gamma(\lambda)} \{\varphi(\lambda+m,\ z)\}^2$$

ここで
$$\varphi(\mu,\ z) = \frac{1}{\Gamma(\mu)} \int_0^{\infty} dt\, t^{\mu-1} e^{-t} (1-e^{-z/t})$$
$$0 \leqq \rho < 1,\quad 0 < \lambda$$

で与えられた．右辺に現われている $\varphi(\mu,\ z)$ も一つの分布関数であるが，これは $z=0$ の近くで性質が悪く，効率良く計算するのは困難である．特に $G(z) = \varphi(1,\ z)$ を考えると

$$G(z) = 1 - \int_0^{\infty} dt\, e^{-t-z/t}$$

$$G'(z) = \int_0^{\infty} dt \frac{1}{t} e^{-t-z/t}$$

となり，$G'(z)$ の方が $G(z)$ よりさらに求めにくい．$G(z)$ は独立な指数分布の積の分布であり，この種の例は分布関数の数値計算では特別なことではない．

このような場合，何らかの関数近似を用いて，逆補間を行なう必要があるが，筆者は後述のような有理関数近似による方法を用いることにより，多くの場合逆関数が効率良く計算できることを発見し，実行している．この有理関数近似の方法は逆関数を求める方法としてはまだ知られていないようなので，ここに記しておく．

1. 有理関数近似による逆補間のアルゴリズム

1次の有理関数

$$y = (Ax+B)/(Cx+1)$$

による近似を考える．近似点は3個必要であり，それを $x_i\,(i=1,\ 2,\ 3)$ とする．もし $f^{-1}(p)$ を求めるのであれば，$y_i = f(x_i) - p$ とおき関数の対応を逆に考えて，方程式

$$x_i=(Ay_i+B)/(Cy_i+1)$$

を解けば，求める $f^{-1}(p)$ の近似値は B である．少し計算をすると，次のアルゴリズムが得られる．

(1) 近似点 x_i を決めて $y_i \leftarrow f(x_i)-p$ とする．$j=1$.

(2) $x_0 \leftarrow x_1-y_1(y_3-y_2)/D$

ここで $D=y_3(y_1-y_2)/(x_1-x_2)-y_2(y_1-y_3)/(x_1-x_3)$.

(3) $y_0 \leftarrow f(x_0)-p$ とする．$y_0 \approx 0$ なら終り．

(4) $(x_j,\ y_j)$ を $(x_0,\ y_0)$ に変更する．

$j=1+\mathrm{mod}\ (j,\ 3)$ にして (2) にもどる．

実際には1の手順はあらかじめ適当なサブルーチンで求めておいた関数値を用いる．(4) の手順は cyclic に行なうのが最も良く，近似点を選択しない方が良い．これは他の補間法の場合も同様なので注意しておく．実際のプログラム例は最後に付されている．

2. 収束の速さについて

近似点が，真値に近く関数 $f(x)$ がなめらかな場合について調べておく．Δx_i を x_i と真値との差の絶対値とするとき，有理関数近似による近似値の誤差の評価は次式で与えられる．x^* を真値として

$$\Delta x_0 \sim L\,\Delta x_1\,\Delta x_2\,\Delta x_3$$

$$\text{ここで}\quad L=-\frac{1}{6}\{f\,;x\}_{x=x^*},\quad \{f;x\}=\frac{f'''}{f'}-\frac{3}{2}\left(\frac{f''}{f'}\right)^2.$$

$\{f;\ x\}$ は Schwartz の導関数である．証明は後に付す．

なお，Newton 法，線型内挿法の場合は

$$\Delta x_0 \sim K(\Delta x_1)^2 \qquad \text{(Newton 法)}$$

$$\Delta x_0 \sim K\Delta x_1\,\Delta x_2 \qquad \text{(線型内挿法)}$$

$$K=\frac{1}{2}\frac{f''}{f'}\Big|_{x=x^*}$$

となる．これらのことから収束のオーダー(*)は Newton 法は2次，線型内挿法は約1.62次，有理関数近似の場合約1.84次である．

Newton 法の step 数を基準にしたとき，以上のことから他の方法の必要 step 数は，下表のようになる．

(理論上の step 数の比較)

Newton 法	1	2	3	4	5	6	7	8	9	10
線型内挿法	1	3	4	6	7	9	10	12	13	15
有理関数	1	2	3	4	6	7	8	9	10	11

(注意) 初期値はすでに与えられたものとし，その誤差はすべて同程度とする．

(*) 近似値の列を z_n，真値を α とする．

$$\lim_{n\to\infty}\ (z_{n+1}-\alpha)/(z_n-\alpha)^k=c\neq 0$$

ならば収束のオーダーは k であるという．

step 数は新しい近似値を求める回数であり，Newton 法では $f(x)$ と $f'(x)$ の2個の関数値を必要とし他の方法では $f(x)$ だけを必要とする．このことからも前述したような，$f'(x)$ の計算が困難な関数に対しては Newton 法は線型内挿法に対してさえもすでに有利とは言いがたいことがわかる．

3. 具体例での比較

(1) $\lambda=4$ の Γ 分布

$$F(x)=\frac{1}{6}\int_0^x e^{-t}t^3\,dt=1-\frac{1}{6}e^{-x}(x^3+3x^2+6x+6)$$

の逆関数を計算してみる．$F(x)$ は倍精度計算で $x>0.1$ のときは上式で求め，$x\leqq 0.1$ のときは巾級数により計算した．Newton 法の場合は

$$F'(x)=\frac{1}{6}e^{-x}x^3$$

も計算する．

初期値は 0，1，2，4，8 とし，収束の判定は

$$|F(x)-p|<10^{-5}p\,(1-p)$$

で行なった．

P (%)	0.1	1	5	50	95	99	99.9	99.99
$F^{-1}(P)$	0.43	0.82	1.37	3.67	7.75	10.05	13.06	15.91
Newton 法	6	3	4	3	3	5	7	9
線型内挿法	27	7	6	4	5	8	12	15
有理関数	8	5	5	5	3	5	8	7

(2) $F(x)=1-e^{-x-\sqrt{x}},\ x\geqq 0$

0，0.5，1，2，4 を初期値とし収束の判定は (1) と同じとする．近似値が $x\leqq 0$ となる場合があるので，$x\leqq 0$ に対して

$$F(x)=x,\quad F'(x)=0.9$$

として計算を続行させた．

P (%)	0.1	1	5	50	95	99	99.9	99.99
$F^{-1}(F)$	1.0E-6	9.9E-5	2.4E-3	0.22	1.69	2.90	4.73	6.63
Newton 法	(**)	(**)	(**)	4	4	5	4	7
線型内挿法	31(*)	19(*)	12(*)	7	6	9	7	11
有理関数	18(*)	11(*)	9(*)	3	3	4	5	7

(*) $x<0$ が現われた　(**) 振動して収束しない

4. エラーの発生に関する注意

一般的に，近似値が関数の定義域を飛び出すエラーを防ぐ事は困難である．前節の例2でわかるように，あらかじめ分布関数の定義域を拡張して計算を続行させるのが一方法である．これは Newton 法の場合は有効ではないが，線型内挿法や有理関数近似の場合は前述のような単純な修正でも計算可能である．しかし，一般的に $x=0$ の近傍での逆関数の計算は困難であるから，初期値は $x=0$ の近くで細かく取った方が良い．後に付してあるサブルーチンでは初期値として $2^n\times10^{-6}(n=1,\cdots,30)$ を取っているが，この程度でほとんど不都合は生じない．定義域が実数全体にわたる場合は，かえって初期値を一般的に設定するのは困難であるが，二分法と有理関数近似を組合せる事によってかなりの場合解決できる．

なお，2次以上の多項式による逆補間については論じなかったが，これは実際上実用にならないからである．これは高次多項式が単調でないために，分布関数のような単調関数を近似するには適さず，定義域からの飛び出し，近似値の振動による発散等のエラーが発生しやすいからである．

5. あとがき

この方法は，筆者が統計数理研究所で計算の手伝いをしているときに見つけたものである．その計算は無線通信のフェージング対策としてダイバーシチ切替受信をしたときの受信電力の分布の計算で，分布関数は種類も多く，かなり複雑であるので，個々の分布関数の性質を詳しく検討してから逆関数を計算するわけにゆかず，すでに述べたように密度関数の方が求め難い方が普通であった．さらに関数値の計算に要する時間も無視できず，効率が良く，しかも汎用性のある方法はないかと思い見つけたものである．

プログラム例.

```
      PROGRAM EXAMPL
      IMPLICIT REAL*8(A-H, O-Z)
      COMMON/INV/VL, ER, IS
      DIMENSION VQ(30)
      EXTERNAL BUNPU
      READ (5, 1) ND
      READ (5, 3) (VQ(I), I=1, ND)
      A=FINVS (BUNPU)
      WRITE (6,200)
      DO 10 I=1, ND
      P=VQ (I)
      PE=P*100.
      A=FINV(P)
      WRITE(6,300) PE, A, ER, IS
   10 CONTINUE
    1 FORMAT (6I5)
    3 FORMAT (D20.6)
  200 FORMAT (1H , 8X, 'PROB (%)', 12X, 'X', 15X, 'ERROR', 3X, 'STEP'/)
  300 FORMAT (1H , F20. 6, F20. 3, D10. 2, I5)
      STOP
      END

      REAL FUNCTION BUNPU*8(Z)
      IMPLICIT REAL*8(A-H, O-Z)
      BUNPU=Z*Z/(1D+0+Z*(1D+0+Z))
      RETURN
      END
```

```
      REAL FUNCTION FINVS*8(F)
      IMPLICIT REAL*8(A-H, O-Z)
      COMMON/INV/VL, ER, IS
      COMMON/DH/FA(3), X(3)
      EXTERNAL F
      DATA E/1.0D-5/
      G (Q, R, S)=(Q-R)*S/A-(Q-S)*R/B
      CALL HIT(F)
      RETURN
      ENTRY FINV(P)
      EP=E*P*(1.0D+0-P)+1D-15
      IS=0
      CALL HIV(P)
   30 FINVS=X(1)
      A=FINVS-X(2)
      B=FINVS-X(3)
      IF (A*B. EQ. 0.D+0) RETURN
      D=G(FA(1), FA(2), FA(3))
      IF (D. EQ. 0.D+0) RETURN
      FINVS=FINVS+FA(1)*(FA(2)-FA(3))/D
      VL=F(FINVS)
      IS=IS+1
      ER=VL-P
      IF (DABS(ER). LT. EP) RETURN
      L=1+MOD(IS, 3)
      FA(L)=ER
      X(L)=FINVS
      IF (IS. LT. 16) GO TO 30
      RETURN
      END

      SUBROUTINE HIT(F)
      IMPLICIT REAL*8(A-H, O-Z)
      DIMENSION XH(50), FH(50)
      COMMON/DH/FA(3), X(3)
      DATA XH(1), FH(1)/2*0.D+0/
      A=1D-6
      AB=2.0
      N=30
      N1=29
      DO 10 I=2, N
      XH (I)=A
      FH (I)=F(A)
   10 A=A*AB
      RETURN
      ENTRY HIV(P)
      DO 20 I=2, N1
      K=I
      IF( P. LT. FH(I)) GO TO 30
   20 CONTINUE
   30 K=K-2
      DO 40 L=1, 3
      J=K+L
      X(L)=XH(J)
   40 FA(L)=FH(J)-P
      RETURN
      END
```

付記 誤差の評価の導出

本文中にあげた x_0 の式は

$$x_0=\begin{vmatrix} x_1 & y_1 & x_1y_1 \\ x_2 & y_2 & x_2y_2 \\ x_3 & y_3 & x_3y_3 \end{vmatrix} \Bigg/ \begin{vmatrix} 1 & y_1 & x_1y_1 \\ 1 & y_2 & x_2y_2 \\ 1 & y_3 & x_3y_3 \end{vmatrix}$$

と書けることに注意する．平行移動により $f(0)=0$ とし，真値は $x=0$ として良い．このとき $y_i=f(x_i)$ である．$f'(0)>0$ と仮定し，$f(x)$ のテイラー展開

$$f(x)=f'(0)x+\frac{1}{2}f''(0)x^2+\frac{1}{6}f'''(0)x^3+o(x^3)$$

を考えて，上式に代入する．

まず，分子の行列式は

$$f'(0)\begin{vmatrix} x_1 & x_1 & x_1y_1 \\ x_2 & x_2 & x_2y_2 \\ x_3 & x_3 & x_3y_3 \end{vmatrix}+\frac{1}{2}f''(0)\begin{vmatrix} x_1 & x_1^2 & x_1y_1 \\ x_2 & x_2^2 & x_2y_2 \\ x_3 & x_3^2 & x_3y_3 \end{vmatrix}$$
$$+\frac{1}{6}f'''(0)\begin{vmatrix} x_1 & x_1^3 & x_1y_1 \\ x_2 & x_2^3 & x_2y_2 \\ x_3 & x_3^3 & x_3y_3 \end{vmatrix}+o(x^6)$$

(注 以下 $o(x^n)$ は $x_i(i=1, 2, 3)$ の n 個の積に対する small order を表わす.)

この第1項は0であり，さらに y_i にテイラー展開を代入して

$$分子=\frac{1}{4}f''(0)^2x_1x_2x_3\begin{vmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{vmatrix}$$
$$+\frac{1}{6}f'''(0)f'(0)\,x_1x_2x_3\begin{vmatrix} 1 & x_1^2 & x_1 \\ 1 & x_2^2 & x_2 \\ 1 & x_3^2 & x_3 \end{vmatrix}+o(x^6)$$

そして，分母は

$$f'(0)^2\begin{vmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{vmatrix}+o(x^3)$$

ゆえに

$$x_0=-\frac{1}{6}\{f;x\}_{x=0}\,x_1x_2x_3+o(x^3)$$

以上

参 考 文 献

[1] Blum, E.K. (1972). *Numerical Analysis and Computation: Theory and Practice*, Addison-Wesley Pnbl.

[2] Berezin, I.S. and Zhidkov, N.P. (1965). *Computing Methods.*, Vol. 1, 2, Pergamon press.

A Remark on the Numerical Calculations of the Inverse of Statistical Distribution Functions

by Sadao Sato
(Tokyo Institute of Technology)

The use of rational function is proved to be often advantageous for calculation of the inverse of the statistical distribution functions, especially when the calculation of the density is difficult and the Newton method is inapplicable. In this paper the algorithm of the calculation is presented, as well as a discussion on the velocity of convergence. The number of necessary steps are compared between the Newton method, linear interpolation method and our method of rational functions for two concrete examples. Some remarks are also stated about the various causes of calculation errors.