сообщения
объединенного
института
ядерных
исследований
дубна

E11 - 12817

E.Běták

PROGRAM FOR PROBABILITIES CALCULATION WITHIN THE GENERALIZED URN MODEL

1979

E11 - 12817

E.Běták

PROGRAM
FOR PROBABILITIES CALCULATION
WITHIN THE GENERALIZED URN MODEL

Бетак Э. **E11 - 12817**

Программа для вычисления вероятностей в обобщенной урновой схеме

Написана программа на языке ФОРТРАН-4 для вычисления громоздких формул обобщенной урновой схемы. С ее помощью получаются соответствующие вероятности для заданных параметров модели. Практическое применение программы ограничено требованием машинного времени, которое при определенных значениях параметров сильно возрастает. Программа написана для расчетов с максимумом 10 цветами и 30 выборами из урнового "ящика". Приведен иллюстрационный пример расчета с 2 цветами и 10 выборами.

Работа выполнена в Лаборатории теоретической физики ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1979

Běták E. **E11 - 12817**

Program for Probabilities Calculation Within the Generalized Urn Model

A FORTRAN-4 program is written for evaluation of tedious formulae of the generalized urn model. It yields the corresponding probabilities for given values of the model parameters. The practical use of the program is limited by the required machine time, which grows seriously for some sets of parameters. The program is written for calculations with maximally 10 colours and 30 draws from the urn. A sample calculation with 2 colours and 10 draws is presented for an illustration.

The investigation has been performed at the Laboratory of Theoretical Physics, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1979

1. INTRODUCTION

Thirty years ago, Friedman [1] developed the so-called urn model, which is capable to describe a wide group of probability distributions with possible physical interpretations. Some commonly used distributions (like binomal, Poisson's and Polya's ones) can be obtained as special cases within such a model (see ref. [2] for detailes). Friedman's model has been recently generalized and discussed by Blažek [2]. The urn model can be presented as follows: Let us consider a "box", filled with a set of objects (like nucleons, quarks, etc.), commonly referred to as balls throughout this paper. These balls are not identical, but they differ by some property, say colour. There are r colours, and a_j balls ($(j = 1,2,\ldots,r)$) of each kind closed in the box. A ball of the f-th colour is drawn at random and put back. Then $\{c_{fj}\}_{j=1,\ldots,r}$ new balls are added into the box, all the balls are mixed together and a next draw may occur. We are interested in the probability $P_K^N[\{a\},\{c\},f,r]$ that the ball of the f-th colour is drawn just K times in total of N draws $(0 \leq K \leq N,\ N \geq 1)$. This probability is [2]

$$P_K^N[\{a\},\{c\},f,r] =$$

$$= \frac{(c_{ff})^K}{(\Sigma' c_{fi})^N} \; \frac{\Gamma\left(\frac{\Sigma' a_i}{\Sigma' c_{fi}}\right)}{\Gamma\left(\frac{a_f}{c_{ff}}\right)} \times$$

$$\times \sum_{\substack{j_1=1\\(j_1\neq f)}}^{r} \sum_{\substack{j_2=1\\(j_2\neq f)}}^{r} \cdots \sum_{\substack{j_{N-K}=1\\(j_{N-K}\neq f)}}^{r} \left\{ \frac{\Gamma\left(\dfrac{\alpha_f + c_{j_1,f} + \dots + c_{j_{N-K},f}}{c_{ff}} + K\right)}{\Gamma\left(\dfrac{\Sigma'(\alpha_i + c_{j_1,i} + \dots + c_{j_{N-K},i})}{\Sigma' c_{fi}} + K\right)} \times \right.$$

$$\times \prod_{\sigma=1}^{N-K} \sum_{\tau_\sigma=(\tau_\sigma)_{min}}^{(\tau_\sigma)_{max}} \left[\frac{\Gamma\left(\dfrac{\alpha_f + c_{j_1,f} + \dots + c_{j_{\sigma-1},f}}{c_{ff}} + \tau_\sigma\right)}{\Gamma\left(\dfrac{\alpha_f + c_{j_1,f} + \dots + c_{j_\sigma,f}}{c_{ff}} + \tau_\sigma\right)} \times \right. \tag{1a}$$

$$\times\ c_{f,j_\sigma} \frac{\Gamma\left(\dfrac{\alpha_{j_\sigma} + c_{j_1,j_\sigma} + \dots + c_{j_{\sigma-1},j_\sigma}}{c_{f,j_\sigma}} + \tau_\sigma + 1\right)}{\Gamma\left(\dfrac{\alpha_{j_\sigma} + c_{j_1,j_\sigma} + \dots + c_{j_{\sigma-1},j_\sigma}}{c_{f,j_\sigma}} + \tau_\sigma\right)} \times$$

$$\left. \left. \times \frac{\Gamma\left(\dfrac{\Sigma'(\alpha_i + c_{j_1,i} + \dots + c_{j_\sigma,i})}{\Sigma' c_{fi}} + \tau_\sigma\right)}{\Gamma\left(\dfrac{\Sigma'(\alpha_i + c_{j_1,i} + \dots + c_{j_{\sigma-1},i})}{\Sigma' c_{fi}} + \tau_\sigma + 1\right)} \right] \right\}$$

for $K \neq N$,

and

$$P_N^N[\{\alpha\}, \{c\}, f, r] =$$

$$= \frac{(c_{ff})^N}{(\Sigma' c_{fi})^N} \frac{\Gamma\left(\dfrac{\Sigma'\alpha_i}{\Sigma' c_{fi}}\right)}{\Gamma\left(\dfrac{\alpha_f}{c_{ff}}\right)} \frac{\Gamma\left(\dfrac{\alpha_f}{c_{ff}} + N\right)}{\Gamma\left(\dfrac{\Sigma'\alpha_i}{\Sigma' c_{fi}} + N\right)} \qquad \text{for } K = N, \tag{1b}$$

where the set of integers r_σ is given by

$$r_0 = 0$$

$$(r_\sigma)_{min} = r_{\sigma-1}; \; (r_\sigma)_{max} = K \qquad (2)$$

for $\sigma = 1, 2, \ldots, N - K$, and the symbol Σ' means summation over i from 1 to $\mathfrak{r}$.

For calculations of expressions (1) a computer program has been written. This paper gives the program listing and its description.

2. BASIC CHARACTERISTICS OF THE PROGRAM

The program is written in FORTRAN-4 [3] and runs successfully at SIEMENS 4004 at Bratislava and CDC 6500 at Dubna. Here the CDC version is presented. Calculations at computers with lower precision (32 bit machines, like EC, IBM and SIEMENS) require the use of double precision. Also at 48 bit machines the double precision is suitable. The program is very modest in memory; but the time requirements can grow tremendously for more colours and/or large N. The program is limited to $\mathfrak{r} \le 10$, $N \le 30$; but these limits can easily be changed by simple adjusting the dimensions of the arrays. The sample run for two distributions with $\mathfrak{r} = 2$, $N = 10$ (K running from 0 to 10), and the Γ function values produced by the subroutine GAMMA from ref. [4], needs 4 seconds compilation and 106 seconds execution time at CDC 6500 (the degree of optimalization OPT =1).

3. NOTATION, INPUT AND OUTPUT

Notation throughout the program basically follows that of formulae (1) and (2). All the important variables are placed into the (unlabelled) COMMON block. Most of the notations are self-explanatory; the remaining ones are:

IJ — Array of the subscripts $\{j_i\}$ in the multiple summation of (1a);

IT — Array $\{r_\sigma\}$;

IF,IR — Denoted as f and $\mathfrak{r}$ within the formulae;

IND — Control variable. See the description of QQQ and XQW below.

QQQ,XQW Limits in overflow control for the Γ function. If the argument of the Γ function is larger than QQQ or closer to negative integer than XQW, the calculation is interrupted and the control variable IND is set equal to 1 and the corresponding value of P_K^N is not produced. The values of XQW and QQQ are preset in the program listing to 0.000001 and 100.0, respectively.

J1, J2, J3 Some values of subscripts which are to be transferred from one subroutine to another.

Input is formed of four cards for one parameter set:

Card No.1	IR, IF	Dimension r and the selected colour f for the calculation. IR.LE.0 causes the stop of the program.
Card No.2	N,IKK	N is the number of draws, and the calculation is done for all values of K from IKK(1) to IKK(3) with the step IKK(2).
Card No.3	AL	Initial numbers of balls.
Card No.4	C	Array of c-coefficients.

Any desired number of card sets can be combined within the single program run.

Output starts by stating which values XQW and QQQ are used through the calculation. Then the input variables are printed in two groups. The first one repeats the values from the input cards Nos. 1,3 and 4 (in the same order); the second one contains the values of N and K. To each combination of parameters the resulting value of P_K^N is printed just after the corresponding K value. For $K = N$, also the sum $\sum_K P_K^N$ is given.

4. PROGRAM UNITS

The program is splitted into a number of subroutines. The main program reads in the input data, calls the subroutine GAMA2 and finally prints the results.

Subroutine GAMA2 evaluates the first line of the r.h.s. of (1). The rest of calculation is done by subroutine GENER.

Subroutine GENER prepares the multiple summation in the second line of the r.h.s. of (1). Then it gives the control to subroutine MASTER.

Subroutine MASTER evaluates the factor on the second line of the r.h.s. of eq. (1).

The tandem of subroutines GENSUM and SUMM calculates the rest of the r.h.s. expression in eq. (1).

Two little subroutines ORDER and ORDER1 are self-explanatory from their listings.

The last subroutine presented in the listing is GAMAC. It calculates the (complex) value GF of the logarithm of the Γ function of the parameter X. This is preferred to the values of Γ function, as the logarithms can (sometimes) overcome the limited range of numbers within the computers, especially for IBM- like machines. The values are supplied here by the function GAMMA from ref. [4], where the reader is referred to.

The program listing is given in the appendix.

5. TEST RUN INPUT AND OUTPUT

As all the input values are repeated as a part of the output, only a copy of the latter is represented.

The author is indebted to Dr. M.Blažek for pointing out this theme.

```
PROGRAM DISTRI      73/73    OPT=1                                    FTN 4.6+460

          PROGRAM DISTRI          (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
          COMMON AL(10),C(10,10),IJ(30),IT(31),
         1       IF,IND,IR,J1,J2,J3,K,N,NK,P,P1,QQQ,SUM,XQW
          DIMENSION IKK(3)
          XQW=0.000001
          QQQ=100.
          WRITE(6,300) XQW,QQQ
        2 READ(5,500) IR,IF
          IF (IR.LE.0) GOTO 1000
          READ(5,500) N,IKK
          READ(5,520) AL
          READ(5,520) ((C(I,J),J=1,IR),I=1,IR)
          WRITE(6,100) IR,IF,(AL(I),I=1,IR)
          WRITE(6,110) ((C(I,J),J=1,IR),I=1,IR)
          IKL=IKK(1)+1
          IKS=IKK(2)
          IKH=IKK(3)+1
          SUP=0.
          DO 1 IK=IKL,IKH,IKS
          K=IK-1
          NK=N-K
          IF (K.GT.N) GOTO 1
          CALL GAMA2
          IF (IND.EQ.0) SUP=SUP+P
          IF (IND.EQ.0) WRITE(6,120) N,K,P
          IF (IND.NE.0) WRITE(6,130) N,K
          IF (K.EQ.N) WRITE(6,140) SUP
        1 CONTINUE
          GOTO 2
     1000 WRITE(6,600)
          STOP
      100 FORMAT(1X,2I4,10F8.2)
      110 FORMAT(5X,15F8.2)
      120 FORMAT(83X,2I4,G17.8)
      130 FORMAT(83X,2I4,17H  GAMMA TOO LARGE)
      140 FORMAT(115X,F11.8//)
      300 FORMAT(1H1,1X,2G12.5//)
      500 FORMAT(16I5)
      520 FORMAT(16F5.1)
      600 FORMAT(22H CALCULATION COMPLETED)
          END
```

```
      SUBROUTINE GAMA 2
      COMPLEX CGR, CG2, GF
      COMMON AL(10),C(10,10),IJ(30),IT(31),
     1       IF,IND,IR,J1,J2,J3,K,N,NK,P,P1,QQQ,SUM,XQW
      IND = 0
      X = AL(IF)/C(IF,IF)
      CALL GAMAC (X,CGR)
      S1 = 0.
      S2 = 0.
      DO 20 I=1,IR
      S1 = S1 + AL(I)
   20 S2 = S2 + C(IF,I)
      X = S1/S2
      CALL GAMAC(X,CG2)
      S = C(IF,IF)**K*REAL(CEXP(CG2-CGR))/S2**N
      IF (NK.EQ.0) GOTO 1100
      S1 = 0.
      CALL GENER
 1000 P = S*P1
      RETURN
 1100 X = AL(IF)/C(IF,IF) + N
      CALL GAMAC(X,CGR)
      S1=0.
      S2=0.
      DO 1200 I=1,IR
      S1=S1+AL(I)
 1200 S2=S2+C(IF,I)
      X=S1/S2+N
      CALL GAMAC(X,CG2)
      P1=REAL(CEXP(CGR-CG2))
      GOTO 1000
      END
```



```
      SUBROUTINE GENER
      COMMON AL(10),C(10,10),IJ(30),IT(31),
     1       IF,IND,IR,J1,J2,J3,K,N,NK,P,P1,QQQ,SUM,XQW
      N1=NK
      J3=1
      CALL ORDER1
  122 IF (IJ(N1).EQ.IR) GOTO 140
      IJ(N1)=IJ(N1)+1
      CALL MASTER
      GOTO 122
  140 N1=N1-1
      IF (N1.EQ.0) GOTO 199
      IF (IJ(N1).EQ.IR) GOTO 140
      IJ(N1)=IJ(N1)+1
      J3=N1+1
      CALL ORDER1
      N1=NK
      GOTO 122
  199 RETURN
      END
```



```
      SUBROUTINE MASTER
      COMPLEX CGR,CG2
      COMMON AL(10),C(10,10),IJ(30),IT(31),
     1       IF,IND,IR,J1,J2,J3,K,N,NK,P,P1,QQQ,SUM,XQW
      DO 1 I=1,NK
      IF (IJ(I).EQ.IF) GOTO 1000
    1 CONTINUE
      X=AL(IF)
      DO 40 IJI=1,NK
      INDJ=IJ(IJI)
      X=X+C(INDJ,IF)
   40 CONTINUE
      X=X/C(IF,IF)+K
      CALL GAMAC(X,CGR)
      ARG=0.
      DO 50 IJI=1,IR
      ARG=ARG+AL(IJI)
      DO 60 IJII=1,NK
      INDJ=IJ(IJII)
   60 ARG=ARG+C(INDJ,IJI)
      X=0.
      DO 80 IJI=1,IR
   80 X=X+C(IF,IJI)
      X=ARG/X+K
      CALL GAMAC(X,CG2)
      IF (IND.EQ.1) GOTO 1000
      PG=REAL(CEXP(CGR-CG2))
      SUM=0.
      CALL GENSUM
      P1=P1+PG*SUM
 1000 RETURN
      END
```

SUBROUTINE GENSUM 73/73 OPT=1 FTN 4.6+460

```
      SUBROUTINE GENSUM
      COMMON AL(10),C(10,10),IJ(30),IT(31),
     1       IF,IND,IR,J1,J2,J3,K,N,NK,P,P1,QQQ,SUM,XQW
      K1=K+1
      IT(1)=1
      N1=NK+1
      IF (N1.LE.1) GOTO 200
      J1=2
      J2 =N1
      CALL ORDER
      CALL SUMM
   22 IF (IT(N1).EQ.K1) GOTO 40
      IT(N1)=IT(N1)+1
      CALL SUMM
      GOTO 22
   40 N1=N1-1
      IF (N1.EQ.1) GOTO 99
      IF (IT(N1).EQ.K1) GOTO 40
      IT(N1)=IT(N1)+1
      J1=N1+1
      J2=NK+1
      CALL ORDER
      CALL SUMM
      N1=NK+1
      GOTO 22
   99 RETURN
  200 CALL SUMM
      GOTO 99
      END
```

SUBROUTINE SUMM 73/73 OPT=1 FTN 4.6+460

```
      SUBROUTINE SUMM
      COMPLEX G1,G2,G5,G6
      COMMON AL(10),C(10,10),IJ(30),IT(31),
     1       IF,IND,IR,J1,J2,J3,K,N,NK,P,P1,QQQ,SUM,XQW
      P=1.
      DO 80 IS=1,NK
      ITS=IT(IS+1)-1
      ARG=AL(IF)
      IS1=IS-1
      IF (IS1.EQ.0) GOTO 2
      DO 1 IJI=1,IS1
      INDJ=IJ(IJI)
    1 ARG=ARG+C(INDJ,IF)
    2 X=ARG/C(IF,IF)+ITS
      CALL GAMAC(X,G1)
      IJS=IJ(IS)
      X=(ARG+C(IJS,IF))/C(IF,IF)+ITS
      CALL GAMAC(X,G2)
      ARG=AL(IJS)
      IF(IS1.EQ.0) GOTO 4
      DO 3 IJI=1,IS1
      INDJ=IJ(IJI)
    3 ARG=ARG+C(INDJ,IJS)
    4 XXXX=ARG+ITS*C(IF,IJS)
      ARG=0.
      ARGC=0.
      DO 6 IJI=1,IR
      ARG=ARG+AL(IJI)
      ARGC=ARGC+C(IF,IJI)
      IF (IS1.EQ.0) GOTO 6
      DO 5 IJII=1,IS1
      INDJ=IJ(IJII)
    5 ARG=ARG+C(INDJ,IJI)
    6 CONTINUE
      ARG3=ARG
      DO 7 IJI=1,IR
    7 ARG3=ARG3+C(IJS,IJI)
      X=ARG3/ARGC+ITS
      CALL GAMAC(X,G5)
      X=ARG/ARGC+ITS+1.
      CALL GAMAC(X,G6)
      TERM = REAL(CEXP(G1+G5-G2-G6))*XXXX
   80 P=P*TERM
      SUM=SUM+P
 1000 RETURN
      END
```

SUBROUTINE ORDER 73/73 OPT=1 FTN 4.6+460

```
      SUBROUTINE ORDER
      COMMON AL(10),C(10,10),IJ(30),IT(31),
     1       IF,IND,IR,J1,J2,J3,K,N,NK,P,P1,QQQ,SUM,XQW
      DO 1 I=J1,J2
    1 IT(I)=IT(I-1)
      RETURN
      END
```

```
SUBROUTINE ORDER1       73/73    OPT=1                        FTN 4.6+452

      SUBROUTINE ORDER 1
      COMMON AL(10),C(10,10),IJ(30),IT(31),
     1       IF,IND,IR,J1,J2,J3,K,N,NK,P,P1,QQQ,SUM,XQW
      DO 1 I=J3,NK
    1 IJ(I)=1
      CALL MASTER
      RETURN
      END
```

```
SUBROUTINE GAMAC        73/73    OPT=1                        FTN 4.6+452

      SUBROUTINE GAMAC(X,GF)
      COMPLEX GF
      COMMON AL(10),C(10,10),IJ(30),IT(31),
     1       IF,IND,IR,J1,J2,J3,K,N,NK,P,P1,QQQ,SUM,XQW
      Y=0.
      IF((X.LE.XQW.AND.ABS(FLOAT(INT(X))- X).LE.XQW).OR.X.GE.QQQ)GOTO999
      CALL GAMMA(2,X,Y,GR,GI)
      GF=CMPLX(GR,GI)
 1000 RETURN
  999 IND=1
      GOTO 1000
      END
```

```
 .1000E-05   100.00

 2    1     1.00        20.00
      10.00       2.00        2.00     10.0

                                        10    0     .27261716
                                        10    1     .26787098
                                        10    2     .19173664
                                        10    3     .12096195
                                        10    4     .70975769E-01
                                        10    5     .39336798E-01
                                        10    6     .20572981E-01
                                        10    7     .99115778E-02
                                        10    8     .42492426E-02
                                        10    9     .14804368E-02
                                        10   10     .32646736E-03
                                                                      1.00000000

 2    1     1.00        20.00
      10.00       3.00        3.00     10.0

                                        10    0     .81933546
                                        10    1     .74485041E-01
                                        10    2     .36870096E-01
                                        10    3     .22941393E-01
                                        10    4     .15557132E-01
                                        10    5     .10934441E-01
                                        10    6     .77452293E-02
                                        10    7     .53935313E-02
                                        10    8     .35940630E-02
                                        10    9     .21504378E-02
                                        10   10     .98117920E-03
                                                                      1.00000000

CALCULATION COMPLETED
```

REFERENCES

1. Friedman B. Comm. Pure Appl.Math.,1949, 2, p.59.
2. Blažek M. Acta Phys.Slov., 1979, 29, p.3.
3. Comm.Assoc.Comp.Mach.,1964,7,p.590.
4. Bardin C., et al. Comp.Phys.Comm.,1972,3,p.73.

Received by Publishing Department
on September 29 1979.