STRATIFICATION OF A POPULATION : PROGRAMMING OF LAVALLEE AND HIDIROGLOU"S ALGORITHM

William Chen, Department of the Treasury
Washington D.C. 20220

Lavallee and Hidiroglov's(1988) developed an algorithm that minimizes the overall size of a random stratified sample by optimally choosing the boundary points of the strata. The boundary points are chosen for a given coefficient of variation for the estimator and a specific power allocation scheme. The current work presents a Fortran Program that perform the above optimal allocation. The computer algorithm is demonstrated with an application to real data.

1. INTRODUCTION

Lavallee and Hidiroglou(1988) developed an algorithm that minimizes the overall size of a random stratified sample by optimally choosing the boundary points of the strata. The boundary points are chosen for a given coefficient of variation for the estimator and a specific power allocation scheme. This allocation scheme enables estimation of the coefficients of variation among the strata that to be similar. A disadvantage of the Neyman allocation is that if we need to estimate each strata, the associated coefficients of variation may be quite different among the strata. However, an allocation that achieves equal coefficients of variation among the strata may require a much larger sample size. The approach developed by Lavallee and Hidiroglou offers a compromise between the Neymann allocation and the attainment of an equal coefficient of variation for each strata. It can be treated as a generalization of the Neymann allocation.

In section II we present the rule of optimum stratification developed by Lavallee and Hidiroglou(1988). In section III we discuss some possible computational problems that may arise when using this algorithm. In section IV we present the Fortran Program to perform the optimum stratification. In section V we give some simulated data examples as well as examples using real data to compare our results with the cumulative square root method.

II. THE ALGORITHM

Let us consider a finite ordered population of N units: $y_1, y_2, \ldots y_N$, with $y_i < y_{(i+1)}$ for $i=1,2,..(N-1)$. This population is to be stratified into L strata. The sampling scheme calls for n_h units to be drawn from each corresponding stratum of size N_h, $h=1,2,...L$, without replacement. Cochran(1977,p.91) defines the usual estimator of the population mean y_{st}. He gives the estimator of the population variance as:

$$\mathrm{Var}(\bar{y}_{st}) = \sum_{h=1}^{\ell} W_h^2 \sigma_h^2 (1 - f_h) / n_h$$

$$= \frac{1}{N^2} \sum_{h=1}^{\ell} \frac{N_h}{n_h} (N_h - n_h) \sigma_h^2 \qquad (2.1)$$

where

$$W_h = N_h / N , \quad f_h = n_h / N_h$$

If it is assumed that the desired level of precision for the estimated mean is specified by the coefficient of variation,c, and that the proportion of sampled units to be allocated to each of the L strata is a_h,(h=1,2,...L) where $\sum_{h=1}^{\ell} a_h = 1$. Then

$$\mathrm{Var}(\bar{y}_{st}) = c^2 \bar{y}^2 \quad \text{and} \quad n_h = n^* a_h \qquad (2.2)$$

where $\bar{y}$ is the population mean and n is the overall required sample size. If it is further assumed in the proportional allocation scheme where a_h is:

$$a_h = \frac{(W_h \mu_h)^p}{\sum_{h=1}^{\ell} (W_h \mu_h)^p} \qquad (2.3)$$

and substitute(2.3),(2.2) into (2.1) then simplified, this results in the overall sample size

$$n = \frac{N\left(\sum_1^{\ell} (W_h^2 \sigma_h^2)(W_h \mu_h)^{-p}\right)\left(\sum_1^{\ell} (W_h \mu_h)^p\right)}{Nc^2 \mu^2 + \sum_1^{\ell} W_h \sigma_h^2} \qquad (2.4)$$

where

$$W_h = \int f(y)dy, \quad \mu_h = \int y f(y)dy / W_h, \qquad (2.5)$$

$$\sigma_h^2 = \int y^2 f(y)dy / W_h - \mu_h^2 ,$$

h=1,2,...L.

To simplify expression, we may also let

$$F = Nc^2 \mu^2 + \sum_1^{\ell} W_h \sigma_h^2, \qquad A = \sum_1^{\ell} (W_h \mu_h)^p,$$

$$B = \sum_1^{\ell} (W_h \sigma_h)^2 (W_h \mu_h)^{-p} \qquad (2.6)$$

Also, let

$$K_h = Bp(W_h \mu_h)^{p-1} - Ap(W_h \sigma_h)^2 (W_h \mu_h)^{-p-1}$$

$$T_h = AW_h (W_h \mu_h)^{-p} \qquad (2.7)$$

Taking the partial derivative of n, in equation (2.4), with respect to $b_{(h)}$ and equating it to zero, Lavallee and Hidiroglou(1988) obtained the following quadratic form

$$(FT_h - FT_{(h+1)}) b_{(h)}^2 + (FK_{(h)} - 2\mu_h FT_{(h)} - FK_{(h+1)}$$
$$- 2\mu_{(h+1)} AB + 2\mu_h AB + 2\mu_{(h+1)} FT_{(h+1)}) b_{(h)} +$$
$$F(\sigma_h^2 + \mu_h^2) T_h - F(\sigma_{(h+1)}^2 + \mu_{(h+1)}^2) T_{(h+1)}$$
$$- AB(\mu_h^2 - \mu_{(h+1)}^2) = 0 \qquad (2.8)$$

Letting the coefficient of $b^2_{(h)}$ be labeled as α_h, the coefficient of $b_{(h)}$ as $\beta_{(h)}$, and the remaining terms as $\gamma_{(h)}$ then equation (2.8) can be written as

$$\alpha_h \, b^2_{(h)} + \beta_h \, b_{(h)} + \gamma_h = 0 \qquad (2.9)$$

Since the term α_h, β_h and γ_h are themselves functions of $b_{(1)}, b_{(2)}, \dots b_{(L-1)}$, Hence they developed the following iterative process:
Step 1: Start with arbitrary boundaries.
Step 2: Compute the sample proportion weight W'_h, the sample mean μ_h, and the sample variance estimate σ^2_h from equation (2.5) based on the boundaries defined in step 1.
Step 3: Replace the old set of boundaries by the new set of boundaries $b_{(1)}, b_{(2)}, \dots b_{(L-1)}$, these $b_{(i)}$ are roots of the quadratic equation, i.e.

$$b_{(h)} = \frac{-\beta_h + (\beta_h - 4 \alpha_h \gamma_h)^{1/2}}{2 \alpha_h} \qquad (2.10)$$

Step 4: Repeat step 2 and step 3 until two consecutive sets of boundaries are either identical or differ by a negligible amount.
In an actual computation the parameter defined in equation(2.5), can be replaced by sample estimates.

III. SOME DISCUSSIONS TO THE ALGORITHM

It should not be too surprising if the above suggested algorithm fails, as the algorithm has shown the following possible problems.
Problem 1: It is possible that the determinant given in equation(2.10) is negative. Usually this event happens when the given coefficient of variation is really small such as 0.01. If this is the case, the program given in the section iv may naturally terminate. This problem is avoided by letting the determinant equal zero. In this way we can have the rest information although the final calculated sample sizes may not reliable.
Problem 2: The new boundary points solved from equation (2.10) are not in strict increasing order. If this is the case, then some of the strata will have a zero count. This problem is avoided by using the natural condition of the root of equation. i.e. the roots $b_{(h)}$ must lie between $\mu_{(h)}$ and $\mu_{(h+1)}$.
The suggested algorithm can help us to find the optimum boundaries such that the required sample sizes will be less than cumulative square root method. (see examples in section v) Another draw back of the method is that it may get larger required sample if the last strata has only one unit or very few units. Then it may be a good idea to use Hidiroglou's(1986) suggestion to reserve the last strata as taken all stratum.

IV. THE PROGRAM

```
ccccccccccccccccccccccccccccccccccccccccccccccc
c     This Fortran Program can iterate the optimum
c     boundaries for the given data files. We only
c     assume that data file has sorted in increasing
c     order and number of strata not exceed ten.
c     usually, sorting the data file can be
c     accomplished by some utility program.
c     A. Input data includes:
c      1. C: coefficients of variation,
c      2. P: power of allocation scheme,
c      3. L: number of strata,
c      4. N: population sizes,
c     B. In each iteration, it will produce the
c        following output data information:
c      1. number of iteration,
c      2. the old bound of the strata,
c      3. the number of counts in each strata,
c      4. the weight in each strata,
c      5. the strata mean values,
c      6. the strata variances,
c      7. the cv values in each strata,
c      8. A,B,F values defined in equations (2.6)
c      9. K(h),T(h) values defined in equations
c         (2.7),
c     10. the coefficients of equation (2.9),
c     11. the created new bounds.
c     C. Possible error information includes:
c      1. the H strata has zero counts or use the
c         zero as denominator: it needs to
c         justify the bounds to some arbitrary
c         bounds.
c      2. negative values in the square roots:
c         it needs to relax the c value to some
c         larger values.
c     D. How to define the initial bound: use the
c        maximum data value subtract the minimum
c        data value divided by the number of
c        strata then evenly divided the range by
c        subinterval of equal length. If this
c        method fails it should modify the
c        initial bounds in some random fashion.
c     E. The stopping rule:
c        two consecutive sets of boundaries are
c        either identical or differ by a
c        negligible amount.
c
ccccccccccccccccccccccccccccccccccccccccccccccc
      implicit real*8(a h,o z)
      dimension bh(10),sum(10),sumsq(10),yhbar(10),
     +varyh(10),wh(10),rkh(10),th(10),alpha(10),
     +beta(10),gama(10),count(10),dbh(10),cv(10),a(10)
     +,y(2000)
      data c,p,sumtot,sumdsq,error/0.1,0.5,0.0,0.0,
     +0.001/
      data l,n,k,kk,num/3,440,1,10,1/
      data fn1,fn2,fn3/0.0,0.0,0.0/
      open(10,file='ud12:sample.dat',status='old')
1000  read(10,100,end=99)(y(i),i=k,kk)
      write(6,100)(y(i),i=k,kk)
      k=k+10
      kk=kk+10
      go to 1000
99    do 1020 i=1,n
      sumtot=sumtot+y(i)
1020  continue
      ybar=sumtot/float(n)
      write(6,120)sumtot,ybar
      do 1030 i=1,n
      sumdsq=sumdsq+(y(i)- ybar)**2
1030  continue
      sterro=sqrt(sumdsq/(n - 1))
      allcv=sterro/ybar
      write(6,130)sterro,allcv
      bh(1)=y(1)
```

```
      bh(l+1)=y(n)
      v=(y(n) -y(1))/float(l)
      begin=bh(1)
      do 1040 i=2,l
      bh(i)=begin+v
      begin=bh(i)
1040  continue
2000  write(6,140)num
      num=num+1
      do 1060 i=1,l
      count(i)=0.0
      sum(i)=0.0
      sumsq(i)=0.0
1060  continue
      do 1080 j=1,l
      do 1100 i=1,n
      if(y(i).ge.bh(j).and.y(i).le.bh(j+1))then
      count(j)=count(j)+1.0
      sum(j)=sum(j)+y(i)
      sumsq(j)=sumsq(j)+y(i)*y(i)
      end if
1100  continue
1080  continue
      write(6,160)(bh(i),i=1,l+1)
      write(6,180)(count(j),j=1,l)
      do 1120 i=1,l
      wh(i)=count(i)/n
      yhbar(i)=sum(i)/count(i)
      varyh(i)=sumsq(i)/count(i) - yhbar(i)**2
1120  continue
      do 1130 i=1,l
      cv(i)=sqrt(varyh(i))/yhbar(i)
1130  continue
      write(6,200)(wh(i),i=1,5)
      write(6,220)(yhbar(i),i=1,5)
      write(6,240)(varyh(i),i=1,5)
      write(6,250)(cv(i),i=1,5)
      suma=0.0
      sumb=0.0
      sumf=n*c*c*ybar*ybar
      do 1140 i=1,l- 1
      suma=suma+(wh(i)*yhbar(i))**p
      sumb=sumb+(wh(i)**2*varyh(i))*(wh(i)*
     +yhbar(i))**(- p)
      sumf=sumf+wh(i)*varyh(i)
1140  continue
      do 1160 i=1,l
      rkh(i)=sumb*p*(wh(i)*yhbar(i))**(p-1)-
     +suma*p*(wh(i)**2*varyh(i))*(wh(i)*yhbar
     +(i))**(-p - 1)
      th(i)=suma*wh(i)*(wh(i)*yhbar(i))**(-p)
1160  continue
      write(6,260)suma,sumb,sumf
      write(6,280)(rkh(i),i=1,5)
      write(6,300)(th(i),i=1,5)
      do 1180 i=1,l- 1
      alpha(i)=sumf*th(i)- sumf*th(i+1)
      beta(i)=sumf*rkh(i)- 2*yhbar(i)*sumf*th
     +(i) - sumf*rkh(i+1)+2*yhbar(i+1)*sumf*th
     +(i+1)+2*yhbar(i)*suma*sumb- 2*yhbar(i+1)
     +*suma*sumb
      gama(i)=sumf*th(i)*yhbar(i)**2+sumf*th(i)
     +*varyh(i) - sumf*th(i+1)*yhbar(i+1)**2-
     +sumf*th(i+1)*varyh(i+1)- suma*sumb*yhbar
     +(i)**2+suma*sumb*yhbar(i+1)**2
1180  continue
      do 1200 i=1,l- 1
      temp=beta(i)**2 - 4*alpha(i)*gama(i)
      if(temp.lt.0)temp=0.0
      dbh(i)=(- beta(i)+sqrt(temp))/(2*alpha(i))
      if(dbh(i).gt.yhbar(i+1))dbh(i)=yhbar(i+1)
      if(dbh(i).lt.yhbar(i))dbh(i)=yhbar(i)
      write(6,440)dbh(i)
1200  continue
      write(6,320)(alpha(i),i=1,5)
      write(6,340)(beta(i),i=1,5)
      write(6,360)(gama(i),i=1,5)
      iflag=0
      do 1220 i=2,l
      if(abs(bh(i)- dbh(i-1)).ge.error)then
      iflag=1
      end if
      bh(i)=dbh(i-1)
1220  continue
      write(6,380)(bh(i),i=1,l+1)
      if(iflag.eq.1)go to 2000
      do 1300 i=1,l
      fn1=fn1+(wh(i)**2*varyh(i))*(wh(i)*
     +yhbar(i))**(-p)
      fn2=fn2+(wh(i)*yhbar(i))**p
      fn3=fn3+wh(i)*varyh(i)
1300  continue
      sample=(n*fn1*fn2)/(n*c*c*ybar*ybar+fn3)
      do 1400 i=1,l
      a(i)=(wh(i)*yhbar(i))**p/fn2
1400  continue
      write(6,400)sample
      write(6,420)(a(i),i=1,5)
 100  format(2x,10(1x,f12.4))
 120  format(5x,'sumtot=',f20.6,3x,'ybar=',f20.6)
 130  format(5x,'the population standard error
     +is',f20.8,5x,'the overallcoefficient of
     +variation is',f20.8)
 140  format(5(/),5x,'the number of iteration is'
     +,25x,i6,/)
 160  format(5x,'the old bound of the strata is',
     +6(1x,f15.4))
 180  format(5x,'the count of the h strata is',
     +5(1x,f15.4))
 200  format(5x,'the weight of h strata is',
     +5(1x,f15.4))
 220  format(5x,'the mean value of h strata is',
     +5(1x,f15.4))
 240  format(5x,'the variance of h strata is',
     +5(1x,f15.4))
 250  format(5x,'the cv of h strata is',
     +5(1x,f15.4))
 260  format(5x,'a=',f15.4,3x,'b=',f15.4,3x,'f=',
     +f15.4)
 280  format(5x,'the rkh value of h strata is',
     +5(1x,f15.4))
 300  format(5x,'the th value of h strata is',
     +5(1x,f15.4))
 320  format(5x,'the alpha value of h strata is',
     +5(1x,f15.4))
 340  format(5x,'the beta value of h strata is',
     +5(1x,f15.4))
 360  format(5x,'the gama value of h strata is',
     +5(1x,f15.4))
 380  format(5x,'after replace with new bound',
     +6(1x,f15.4))
 400  format(5(/),5x,'the final total sample size
     + required',f10.2)
 420  format(5x,'the power allocation of a(i) is'
     +,5(1x,f10.4))
 440  format(20x,'the new bound is',f20.5)
      close(10)
      stop
      end
```

V. EXAMPLES AND APPLICATION

To illustrate results from previous section, we use data,computer generated,gross receipts of corporations,adjusted gross income of individuals in 1985. We compare the proposed method with the cumulative square root method in terms of the required sample sizes. The computed skewness for these populations is 1.464(for population 1), 11.563(for population 2),14.283(for population 3).

Example 1. Using a computer to generate 50 Chi-Square distribution with one degree of freedom. We want to cut into two strata or three strata.

use cum $f^{1/2}$ method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.25	1.0	1	39	2	1.14	0.001
						1.594
		2	11	5	0.34	5.204
		total		7		

use optimum method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.25	1.0	1	34	1	1.06	0.001
						0.987
		2	16	4	0.43	5.204
		total		5		

use cum $f^{1/2}$ method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.1	0.5	1	39	13*	1.14	0.001
						1.594
		2	11	11	0.34	5.204
		total		24		

use optimum method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.1	0.5	1	32	4	0.98	0.001
						0.989
		2	18	11	0.48	5.204
		total		15		

use cum $f^{1/2}$ method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.1	0.25	1	39	11*	1.14	0.001
						1.594
		2	11	11	0.34	5.204
		total		22		

use optimum method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.1	0.25	1	34	7	1.06	0.001
						1.158
		2	16	9	0.43	5.204
		total		16		

'*' means: allocation is required to satisfy coefficient of variation.

Example 2. Using 400 values of gross receipts of corporations in the United States. The population is divided into 3 or 4 strata. (Notice that the original data has been divided by billion.)

use cum $f^{1/2}$ method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.25	1.0	1	291	2	1.24	0.00
						2.37
		2	75	4	0.41	9.05
		3	34	9	1.24	197.60
		total		15		

use optimum method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.25	1.0	1	291	1	1.24	0.00
						2.45
		2	98	5	0.68	24.45
		3	11	4	0.90	197.60
		total		10		

use cum $f^{1/2}$ method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.25	0.5	1	291	3	1.24	0.00
						2.37
		2	75	5	0.41	9.05
		3	34	9	1.24	197.60
		total		17		

use optimum method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.25	0.5	1	302	2	1.27	0.00
						2.99
		2	87	4	0.63	25.74
		3	11	3	0.90	197.60
		total		9		

use cum $f^{1/2}$ method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.15	1.0	1	261	1	1.15	0.00
						1.44
		2	74	3	0.33	4.93
		3	43	5	0.30	13.18
		4	22	11	1.10	197.60
		total		20		

use optimum method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.15	1.0	1	260	1	1.15	0.00
						1.37
		2	102	3	0.51	8.36
		3	32	4	0.42	39.40
		4	6	4	0.77	197.60
		total		12		

use cum $f^{1/2}$ method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.05	1.0	1	261	2	1.15	0.00
						1.44
		2	74	6	0.33	4.93
		3	43	10	0.30	13.18
		4	22	22	1.10	197.60
		total		40		

use optimum method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.05	1.0	1	220	3	1.03	0.00
						0.77
		2	115	6	0.51	4.99
		3	54	12	0.47	23.48
		4	11	11	0.90	197.60
		total		32		

Example 3: The population size of 990 adjusted individual gross income for 1985 was used. We want to cut the population into three strata for different combination of c, p and l.

use cum $f^{1/2}$ method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.1	1.0	1	501	3	0.58	0.00
						0.83
		2	284	5	0.16	1.49
		3	205	7	0.41	9.05
		total		15		

use optimum method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.1	1.0	1	408	2	0.56	0.00
						0.64
		2	431	7	0.27	1.71
		3	151	5	0.41	9.05
		total		14		

use cum $f^{1/2}$ method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.05	1.0	1	501	12	0.58	0.00
						0.83
		2	284	19	0.16	1.49
		3	205	27	0.41	9.05
		total		58		

use optimum method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.05	1.0	1	364	5	0.55	0.00
						0.58
		2	446	24	0.27	1.61
		3	180	22	0.41	9.05
		total		51		

use cum $f^{1/2}$ method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.05	0.25	1	501	17	0.58	0.00
						0.83
		2	284	20	0.16	1.49
		3	205	21	0.41	9.05
		total		58		

use optimum method

0.05	0.25	1	460	14	0.57	0.00
						0.74
		2	396	18	0.25	1.82
		3	134	16	0.41	9.05
		total		48		

use cum $f^{1/2}$ method

C	P	Strata	N_h	n_h	C	$b_{(h)}$
0.01	1.0	1	501	127*	0.58	0.00
						0.83
		2	284	163	0.16	1.49
		3	205	205	0.41	9.05
		total		495		

0.01	1.0	1	345	42	0.55	0.00
						0.53
		2	442	194	0.27	1.49
		3	203	202	0.41	9.05
		total		438		

VI. CONCLUDING REMARKS

From examples 1 to 3 we can see that using optimum boundaries requires less sample than using the cumulative square root method. The discrepancy of the sample sizes will depend on the distribution of the data, given coefficient of variation, c, the power of allocation, p, and the number of strata l. For both methods, the total required sample sizes will heavily depend on the given c value while less on p value and l value. If the computed determinant is negative or final sample sizes larger than the population size N it means that the current suggested algorithm cannot stratify the given population into the specified precision c value. It is clear that the cumulative square root method can make up this difficulty since in the procedure of stratification it does not depend on the c-value.

ACKNOWLEDGEMENT

I am grateful for Pierre Lavallee and Michel A. Hidiroglou for their useful comments.

REFERENCES

Cochran,W.G.: Sampling Techniques,1977,(3rd. ed.) New York:John Wiley and Sons.

Hidiroglou,M.A.(1986): The construction of a self representing stratum of large units in survey design. The American Statistician, ,40,27-31.

Lavallee P. and Hidiroglou M.A.(1988): On the stratification of skewed populations. Survey Methodology, Vol. 14,No. 1,33- 43.