AN OPTIMUM ALLOCATION ALGORITHM FOR MULTIVARIATE SURVEYS

James Bethel, U.S. Department of Agriculture

1. Introduction

The problem of optimal sample allocation for multipurpose surveys can be viewed more generally as a problem in convex programming and, as such, there are many ways to obtain a numerical solution. Huddleston, Claypool and Hocking (1970) have applied a nonlinear programming method devised by Hartley and Hocking (1963) to this problem, and Kokan (1963) has discussed some standard nonlinear programming techniques with respect to optimal allocation. While these and other, more general methods are available, most of them are difficult to program and computationally burdensome, and not all are guaranteed to converge. In this paper an algorithm is presented which is relatively simple to program and which converges quickly, even on small computers. The proof is beyond the scope of this paper, but it can be shown that the algorithm is guaranteed to converge. First, the allocation model and the algorithm will be described. Then, after a discussion of various issues related to the implementation of the algorithm, an example using data from an agricultural survey will be presented.

Consider the case of stratified random sampling with p variables of interest. Suppose it is required that the j-th variable, $1 \le j \le p$, satisfy

$$\mathrm{var}(\bar{y}_j) = \sum_{i=1}^{L} w_i^2 S_{ij}^2 \left(\frac{1}{n_i} - \frac{1}{N_i} \right) \le v_j .$$

Let

$$x_i = \begin{cases} \dfrac{1}{n_i} & \text{if } n_i \ge 1 \\ \infty & \text{otherwise.} \end{cases}$$

Assume the cost function

$$C = \sum_{i=1}^{L} c_i n_i = \sum_{i=1}^{L} \frac{c_i}{x_i} .$$

Now the problem reduces to minimizing

$$C = C(x) \tag{1}$$

subject to the constraints

$$\sum_{i=1}^{L} a_{ji} x_i \le 1 , \quad 1 \le j \le p \tag{2}$$

where

$$a_{ji} = \left[w_i^2 S_{ij}^2 \right] \Big/ \left[v_j + \sum_{i=1}^{L} \frac{w_i^2 S_{ij}^2}{N_i} \right] ,$$

The discussion will be limited to this allocation model, since Kokan (1963) shows how it can be adapted to cover virtually any sampling situation.

2. The algorithm

Consider this informal argument: For fixed values of k, the set

$$S_k = \{ x: \sum \frac{c_i}{x_i} = k \} \tag{3}$$

forms a convex hyperboloid, while the set

$$F = \{ x: a_j' x \le 1, \ 1 \le j \le p \} \tag{4}$$

forms a convex polygon below S_k. As k increases, S_k moves downward toward the upper boundary of the feasible region F and the point where these sets meet is the optimal solution to (1) and (2).

For any hyperplane

$$H = \{x: a'x = 1\} \tag{5}$$

Kokan and Khan (1967) show that H and S_k are tangent (for some suitable k) at the point t, where

$$t_i = \begin{cases} (c_i a_i)^{1/2} / \left(a_i \sum_{i=1}^{L} (c_i a_i)^{1/2} \right) & \text{if } a_i \ne 0 \\ \infty & \text{otherwise.} \end{cases} \tag{6}$$

Consider the $a_j = (a_{j1}, a_{j2}, \ldots, a_{jL})'$, as defined by (2). Let $H_j = \{x: a_j'x = 1\}$ and suppose that $t_j = (t_{j1}, t_{j2}, \ldots, t_{jL})'$ is the point where H_j and S_k are tangent. If $t_j \in F$, then, as Kokan and Khan (1967) show, t_j is the optimal solution to (1). Unfortunately, this is rarely the case.

Suppose $H = \{x: a'x = 1\}$ and $t = t(H)$ is the point where, for some suitable k, H is tangent to S_k. The cost C(t) can be written as a function of the coefficients a_i:

$$C(t) = \sum_{i=1}^{L} \frac{c_i}{t_i} \tag{7}$$

$$= \sum_{i=1}^{L} c_i / (c_i a_i)^{1/2} \Big/ \left(a_i \sum_{i=1}^{L} (c_i a_i)^{1/2} \right)$$

$$= \sum_{i=1}^{L} (c_i a_i)^{1/2} \sum_{i=1}^{L} (c_i a_i)^{1/2}$$

$$= \left(\sum_{i=1}^{L} (c_i a_i)^{1/2} \right)^2 .$$

For convenience, write

$$G(H) = C(t(H)) = C(t). \tag{8}$$

The algorithm begins by selecting one of the H_j as an initial value $H^{(1)} = \{x: a^{(1)\prime}x = 1\}$. For example, take

$$H^{(1)} = H_1.$$

Choose $H^{(2)}$ to satisfy

$$G(H^{(2)}) = \max_{0 \le \beta \le 1} \left[\sum_i (c_i (a_i^{(1)} + (1-\beta) a_{2i})^{1/2} \right] \tag{9}$$

In a sense, $H^{(2)}$ is the convex combination of $H^{(1)}$ and H_2 which maximizes G. Now find $H^{(3)}$ by repeating this process with H_3, replacing $a_i^{(1)}$ with $a_i^{(2)}$ and a_{2i} with a_{3i} in formula (9).

In general, take $H^{(n+1)}$ to satisfy

$$G(H^{(n+1)}) = \max_{0 \le \beta \le 1} \left[\sum (c_i (a_i^{(n)} + (1-\beta) a_{j_n i})^{1/2} \right] \tag{10}$$

where $j_n = n+1 \pmod p$. For the sake of discussion, one iteration will be considered to consist of calculating $H^{(1)}, H^{(2)}, \ldots, H^{(p)}$.

Denote the optimal solutions to (1) and (2) by x^*. At the n–th step, estimate x^* with $x^{(n)}$ where $x^{(n)}$ is the point of contact between $H^{(n)}$ and $S_{G(H^{(n)})}$. Clearly $G(H^{(n+1)}) \ge G(H^{(n)})$, so that $S_{G(H^{(n)})}$ moves downward towards F as n increases and, consequently, $x^{(n)}$ approaches x^*. While it is true that $x^{(n)}$ always violates some constraint, the violations, for large n, will be negligible.

3. Implementing the Algorithm

The algorithm converges for any starting value $H^{(1)}$, but, intuitively, it makes sense to maximize $G(H^{(j)})$, since the starting value $x^{(1)}$ should be as close as possible to F. Thus take

$$H^{(1)} = H^{(j_0)}$$

where

$$G(H^{(j_0)}) \ge G(H^{(j_b)})$$

for all j.

Since

$$g_j(\alpha) = \Sigma_i (c_i (\alpha a_i^{(n)} + (1-\alpha) a_{j i}))^{1/2}$$

is concave in α, a direct search can be carried out for the maximum on [0,1]. This was accomplished by using a small positive value β and finding

$$\max \{ g_j(0), g_j(\beta), g_j(2\beta), \ldots, g_j(k\beta), \ldots, g_j(1) \}. \tag{11}$$

This was done by starting with $g_j(0)$ and stopping when

$$g_j(k\beta) \ge g_j((k+1)\beta).$$

This is not as inefficient as it may appear. Even on problems with many constraints there are usually only two or three which determine the solution; that is, most of the constraints will not be searched since any value $k > 0$ results in lowering the cost. Also, the value of k which maximizes g_j is usually quite small, almost always less than .5. Thus this searching method wastes little time searching over constraints which are unnecessary and usually will not expend too many steps in finding the optimal k

In order for convergence to occur, β must be decremented. This was done after each iteration by replacing β with $\lambda\beta$, where $0 \le \lambda \le 1$. If β decreases too quickly, it will require many steps to obtain (11). If it decreases too slowly, many iterations will be necessary to obtain convergence. Initially taking $\beta = .05$ and setting $\lambda = .90$ seems to work well.

As noted above, $x^{(n)}$ always violates some constraint. The convergence criterion used was to require that the maximum relative constraint violation be no larger than ε. For example, if the variance requirement is $v_j = \gamma$, then setting a convergence criterion of ε would mean that $\text{var}(\bar{y}_j) \le \gamma(1+\varepsilon)$ must hold for each j.

4. Example

The example is drawn from an agricultural survey done by the United States Department of Agriculture (USDA). Population totals and standard deviations were estimated from previous data and are given in Table 1. Here the variance constraint is that all coefficients of variation must be less than or equal to .08, with a convergence criterion of .01, thus the effective requirement is that all CVs be no larger than .0808. The allocations are given in Table 2; each column corresponds to one iteration, as described in the previous section. Note that columns 3, 4, and 5 are the same, indicating that β was too large to refine the allocation. Table 3 gives the actual coefficients of variation resulting from the allocations in Table 2. Notice that the final CVs for all variables except variables 1 and 4 (cattle and dairy cattle) are smaller than .08. This indicates that these are the "binding" constraints and that the optimal solution lies on the intersection of the constraint hyperplanes associated with these two variables.

The program to implement this was written in PASCAL and run on a Zilog System 8000. It took 12 seconds of CPU time. Several problems of this size (including the one given in Huddleston, Claypool and Hocking, 1970) have been run with this program, all have taken less than 30 seconds of CPU time.

TABLE 1: SAMPLING INFORMATION FOR ILLINOIS AGRICULTURAL SURVEY (BY STRATUM).

Stratum Size	Stratum Cost	S_{i1}	S_{i2}	S_{i3}	S_{i4}	S_{i5}	S_{i6}	S_{i7}	S_{i8}	S_{i9}*
58112	6	78	1528	543	4	80	75	27	22	59
2390	6	51	3696	787	5	111	86	28	22	480
87	6	68	3057	665	23	58	17	12	32	556
2440	6	59	2381	1869	35	95	74	45	38	43
17833	6	73	5433	3462	6	242	195	72	33	88
2813	6	124	8600	1530	3	252	183	73	31	690
693	6	98	4051	2264	41	211	148	65	152	111
96	6	91	4603	527	28	256	113	78	58	804
29415	140	99	936	529	9	58	56	14	12	188
10031	140	21	789	367	2	46	62	24	13	158
9664	140	13	207	72	5	67	34	23	14	38
TOTAL	(000)	2058	105133	30427	245	11450	9354	1849	1152	6171

*The variables are, respectively, number of cattle, bushels of stored corn, bushels of stored soybeans, number of dairy cattle; acres of planted corn, soybeans, wheat, and hay; number of hogs.

TABLE 2: SAMPLE ALLOCATION.

Stratum	Interation 1	2	3	4	5	6
1	2453	2197	2212	2212	2212	2192
2	66	75	75	75	75	75
3	3	6	6	6	6	6
4	78	253	248	248	248	247
5	704	859	853	853	853	967
6	189	176	176	176	176	177
7	37	87	85	85	85	85
8	5	9	9	9	9	9
9	326	320	320	320	320	316
10	24	26	26	26	26	26
11	14	31	30	30	30	30

TABLE 3: COEFFICIENT OF VARIATION.

Variable*	Interation 1	2	3	4	5	6
1	.0800	.0800	.0800	.0800	.0800	.0802
2	.0483	.0457	.0458	.0458	.0458	.0445
3	.0893	.0818	.0820	.0820	.0820	.0785
4	.0955	.0795	.0799	.0799	.0799	.0800
5	.0257	.0222	.0223	.0223	.0223	.0219
6	.0529	.0240	.0241	.0241	.0241	.0237
7	.0545	.0471	.0473	.0473	.0473	.0467
8	.0548	.0471	.0473	.0473	.0473	.0471
9	.0818	.0796	.0796	.0796	.0796	.0797

*The variables are, respectively, number of cattle, bushels of stored corn, bushels of stored soybeans, number of dairy cattle; acres of planted corn, soybeans, wheat, and hay; number of hogs.

REFERENCES:

Kokan, A.R. (1963). Optimum allocation in multivariate surveys. J.R. Statist. Soc. A, 126, 557–565.

Kokan, A.R., and Khan, S. (1967). Optimum allocation in multivariate surveys: an analytical solution J.R. Statist. Soc. B, 29, 115–125.

Hartley, H.H., and Hocking, R.R. (1963). Convex programming by tangential approximation, Management Science, 9, 600–612.

Huddleston, H.F., Claypool, P.L., and Hocking, R.R. (1970). Optimum sample allocation to strata using convex programming. App. Stat. 19, 273–278.