A SCHEME FOR SAMPLING TIME ORDERED POPULATIONS

James H. Drew, GTE Laboratories Incorporated

1. INTRODUCTION

We describe one part of a two-stage scheme for sampling a time ordered population in which there are different levels of interest and background information for each of the stages.

Yearly telephone call records are accessible by means of daily computer tapes, each of which contains the records of all calls made on the network during that day. Those calls are generally ordered by the time of day at which a given call was initiated. For the following discussion, interest centers on the measurement of call intensity--the number of calls initiated during a unit of time--between pairs of nodes in the telephone network as a function of the time of day, say x. For a given node pair and a given day it was felt that the functional form of the call intensities within the hours of interest could be well approximated by a polynomial in x whose degree was known to lie between two specified integers k_1 and k_2.

Note that to estimate call intensities, it is necessary to select time intervals over which to record the number of calls initiated. The sampling task generated by these considerations is then to construct a series of times $x_1, x_2, \ldots, x_N$ and numbers of calls to be processed after those times, say $n_1, n_2, \ldots, n_N$ so that subsequent estimation of the call intensity functional form is sufficiently good in a sense to be made clear. If the i^{th} time interval is indexed by its start time x_i and its relative sample size w_i, then the problem is to choose a good set

$$\xi = \{(x_1,w,_1), (x_2,w_2), \ldots, (x_N,w_N)\}.$$

The problem thus becomes one of optimal experimental design, where the design is given by ξ. The classical literature of Kiefer and Wolfowitz[1] and Federov[2] is then useful in our situation. Furthermore, Cook and Nachtsheim[3] consider a related problem, and both their paper and the present one benefit from the work of Tsay.[4]

The second phase of the sampling scheme is the selection of days of the year for which to select daily tapes. Much less background inforformation is available here. No family of functional forms for daily call records was postulated, nor was there any special interest in the hypothesizing of such forms. One could, however, identify several factors which were believed to play some role in determining daily effects. It seems natural then to use this information to stratify the population of days and to use the inferential techniques of classical sampling theory as given in, say, Cochran.[5] That is, although we postulate a superpopulation model for the call records within a day, the days themselves constitute a fixed population which we interpret within the context of finite population theory.

However, in this paper we concentrate on the first phase of sampling--within a day.

2. WITHIN DAY DESIGN CONSIDERATIONS

2.1 Classical Optimal Design

Suppose the call intensity at time x is y(x) and the model

$$y(x_i) = \underset{\sim}{X}_{ik}\underset{\sim}{\beta}_k + \varepsilon_i$$

is appropriate, where

$$\underset{\sim}{X}_{ik} = (1, x_i, \ldots, x_i^k),$$
$$\underset{\sim}{\beta}_k = (\beta_0, \beta_1, \ldots, \beta_k)',$$

and ε_i is a random variable with mean 0 and variance σ^2. For the design

$$\xi = \{(x_1,w_1), \ldots, (x_N,w_N)\}$$

where

$x_i = i^{th}$ starting time for observing calls,

n_i = number of calls observed at i^{th} time,

$w_i = n_i/n.$, and $n = \sum_{i=1}^{N} n_i$,

the covariance matrix of the ordinary least squares estimator of $\underset{\sim}{\beta}_k$ is given by:

$$\underset{\sim}{V}\,\sigma^2$$

where

$$V^{-1} = (v_{rs}), \text{ and}$$

$$V_{rs} = \sum_{i=1}^{N} w_i x_i^{(r+s)}, \quad \begin{matrix} r = 0, 1, \ldots, k \\ s = 0, 1, \ldots, k. \end{matrix}$$

Let

$$d_k(x,\xi) = (1\ x\ x^2\ \ldots\ x^k)\ \underset{\sim}{V}(1\ x\ x^2\ \ldots\ x^k)',$$

and consider $\max_x d_k(x,\xi)$. A design ξ is called called G-optimal if it minimizes this latter quantity. It is a well known result of Kiefer and Wolfowitz[1] that if ξ is G-optimal, then $\max_x d_k(x,\xi) = k + 1$. We will use this fact to assess the nearness of our designs to the optimal.

To construct nearly optimal designs, we use the following basic algorithm, which is in the spirit of those algorithms given in Tsay[4] and Cook and Nachtsheim.[3]

(1) Begin with a design at prespecified points $x_1, \ldots, x_N$, and equal weights (= 1/N). In practice these points will be the closest spaced feasible sampling points. The weights can be represented by the 1×N vector (1 1 ... 1 1) which is just a scalar multiple of the actual vector of weights.

(2) Find $x^* \varepsilon \{x_1, x_2, \ldots, x_N\} = S$ such that $x^* = \max_S d_k(x,\varepsilon)$.

(3) Add one to the appropriate component of the vector (1 1 ... 1 1) and normalize so the new components sum to one.

(4) Continue in this way until $\max_x(x,\xi)$ becomes sufficiently close to k + 1.

Note that if we were using the exact round-off scheme of Federov,[2] the work of Tsay[4] implies convergence of this sequence of designs to the optimum.

Serious complaint might be lodged against the fixedness of the design points, particularly if the points were so sparse as to obstruct convergence to the optimum. Designs with an excessive number of initial points need not concern us, since in the limit it will happen that superfluous points are assigned zero weight. Experiments we have done indicate that if the number of equally spaced design points is somewhat more than twice the highest degree of the polynomials considered, then the algorithm converges to within a few tenths of a percent of the optimum. The algorithm has converged in all of our experiments.

2.2 Robustness

In our problem we were unwilling to specify the degree of the polynomial in the model for call intensities. Instead, k was specified to lie between k_1 and k_2. It was necessary, then to search for a design ξ which would be nearly optimal for this range of polynomial models.

First, it is necessary to rescale our measure of design goodness so different polynomial degrees can be compared. Since the maximum variance of a predicted value x from a k^{th} degree polynomial is k + 1, the values of $d_k(x,\xi)$ below have been transformed by substracting k + 1, and then dividing by k + 1. The transformed values are then the percentages (divided by 100) the realized $d_k(x,\xi)$ achieve in excess of the optimum.

The procedure we have adopted is easily described by its algorithm. Recall in the classical algorithm, one searches for x* such that $\max_x d_k(x,\xi) = d_k(x^*,\xi)$. Now, for $S = \{k_1, k_1 + 1, \ldots, k_2\}$ we search for x* minimizing

$$\max_S \max_x d_k(x,\xi)$$

Thus we generate a matrix of variances of predicted values (scaled), each row corresponding to the variances generated from a particular polynomial degree under consideration. The next design point puts its weight where the predictor variance is highest among all rows and columns of the matrix.

It is of interest to know what $d_k(x,\xi)$ values will be produced by the converged algorithm, or whether the algorithm converges at all. It is easily shown that if the algorithm converges, then the realized $(d_k x,\xi)$ values for the selected set of k's will all be equal. They will not all be zero for the simple reason that zero corresponds to the optimal design, and not all polynomials have the same optimal design. It is heartening to note that in our experiments, a few results of which are reproduced below, we do achieve near equality of the relevant $d_k(x,\xi)$ values.

Figure 1 shows the values of $\max_x d_k(x,\xi)$ for $k_1 < k < k_2$ associated with several designs generated by the scheme just described. These designs are supported by the x_i values 0.5, 1.0, 1.5, ..., 9.5, 10.0. The values of $\max_x d_k(x,\xi)$ have been rescaled by subtracting k + 1, and then dividing by (k + 1)/100. The displayed values therefore represent the percentage by which the $\max_x d_k(x,\xi)$ value exceeds its theoretical minimum. This quantity is labeled MV in Figure 1.

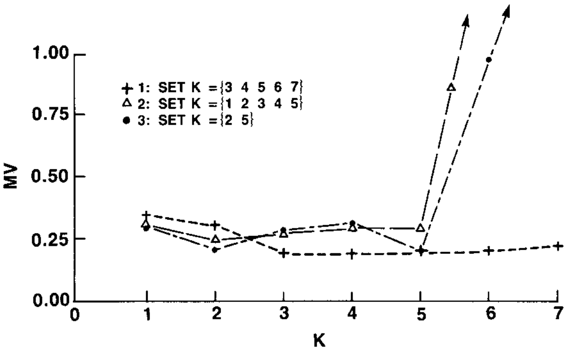

Figure 1. MV for Three Designs

We note in passing that the scaling of $d_k(x,\xi)$ can be modified to deal with certain other design specifications. For example, one might be less interested in the function for some values of x than for others, and this interest might be translated into upper bounds B_{Ix} on the predictor variance for certain intervals of x. It would then be reasonable to apply the algorithm with $d_k(x,\xi)$ replaced by

$$[d_k(x.\xi) - B_{Ix}]/B_{Ix}$$

In interpreting these graphs, one may ask how far above the optimal value of k + 1 $\max_x d_k(x,\xi)$ may be and still be reasonably good. Some heuristic insight is gained by noting that the equal spacing, equal weight design for N points with k = 1 is associated with a value of 4 (twice the theoretical minimum) when N becomes large. Since this design has some intuitive appeal, a value of MV, the scaled version of $\max_x d_k(x,\xi)$ even as high as 1.00 may not be excessive.

Given the genesis of the designs ξ_v, it is not surprising that each design performs well for the k values for which it was generated. In addition we note that, in general, designs generated for high degree polynomials tend to perform better for a wide range of k values, presumably because they put weight on many different x-values. However, it is an artifact of fitting polynomial models that a good design must put relatively heavy weight near the end points of the interval supporting the x-values. Unlike certain special trigonometric models that afford optimal designs which are equally spaced and weighted, near optimal and robust designs for polynomial models do not have equal weightings for equal spacings.

These points are illustrated by a plot of the design weights against the associated design points, shown for the design in Figure 2.

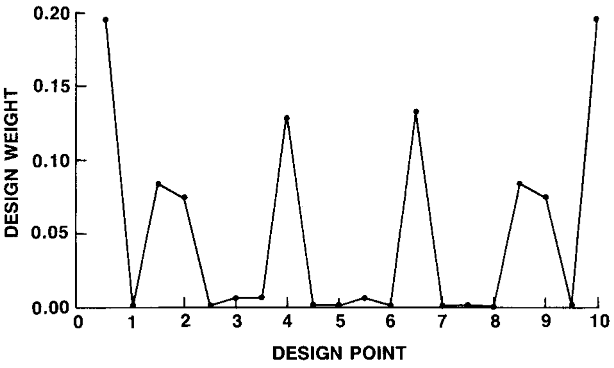

Figure 2. Design for k = 3 4 5

The algorithm can be modified to incorporate initial beliefs about the possible values k may actually be. Operationally, given relative weights $\{p_{ki}\}$ for the polynomial degrees $\{k_i\}$ we consider each of the degrees $k_1, \ldots, k_2$ selecting polynomials of degree k_i with probability p_i to use in the algorithm described above. The selection process is repeated for a large number of iterations of the algorithm. This selection process ensures that degrees appear with the required probabilities, and the same degree will appear at most once in any one iteration. Each set of polynomials is of a stochastic size, however. Some experiments suggest that the generation of many polynomial degrees for use in a single iteration produces convergence slightly faster than the use of a small number of pőlynomial degrees per iteration.

2.3 Constraints on the Design

Each time interval over which call records are processed contains calls between various node pairs in the network. It is the functional form of the call intensities for the calls between specified node pairs which are of interest. Hence each set of node pairs identifies a study domain of the population. In order to insure the precise measurement of call intensities for each of the node pairs, we would like to ensure with a specified probability 1 - α that at least c_{ij} call records for the j^{th} node pair are observed in the i^{th} time interval. It seems reasonable to suppose that $\{n_{ij}\}$, the number of records observed from the j^{th} node pair in the i^{th} time interval (i.e., that one beginning at time x_i), is distributed as a multinomial with parameters C_i and p_{ij}, where C_i is the total number of call records observed in the i^{th} time interval and p_{ij} is the probability that an observation in the i^{th} time interval is from the j^{th} node pair. We then require that C_i be chosen so that

$$P(n_{ij} > c_{ij} \,|C_i,p_{ij}| > 1 - \alpha_i.$$

The algorithm given in the previous section lends itself to the imposition of constraints on the values $\{n_i\}$. Let the constrained values of $\{n_i\}$ be called $\{C_i\}$ and consider the algorithm described above. Replace step (1), in which an arbitrary design was chosen to begin the design generation, by the vector $(C_1\ C_2, \ldots, C_N)$, scaled so the sum of the components is unity. Then the subsequent steps of the algorithm effectively add observations to those x-values for which $d(x,\xi)$ is large. As before, Tsay's work[4] shows that this procedure leads to optimal designs for specified k.

Using this procedure, we create a sequence of designs which span the gap between those which merely satisfy the constraints on the number of calls per time interval, and those which are arbitrarily close to the optimum. In that regard, note that for a given design with weight w_i and constraint C_i at time x_i, the constraints can be satisfied by choosing the total sample size $n_{\bullet}$ so that

$$n_{\bullet} = \max_i(C_i/w_i).$$

The sampler can then choose among designs which have a specified closeness to the optimal and which also satisfy the sample size constraints.

The speed at which designs with various constraints approach the optimal is illustrated in the following figure, in which the scaled values of $\max_x d_k(x,\xi)$ are given for the above algorithm with the stated constraint and the given polynomial degree.

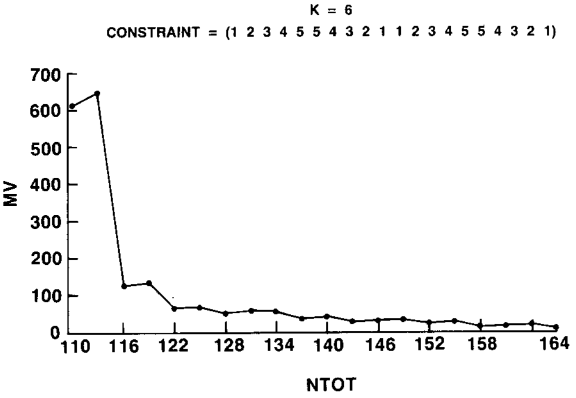

Figure 3. Constrained Design Toward Optimum

In these figures, NTOT is $n_{\bullet}$, computed as given above. For a given constraint, it is hardly surprising that a reasonable value of MV (say 20) is reached fairly quickly when the constrained weights are not too different from the optimal weights. It also appears that these graphs are about the same whatever the presumed degree of the polynomial model.

2.4 Equal Spacing and Weighting

It is of interest to note that designs with equal spacing and equal weights are not very good, in the sense that they have large $\max_x d_k(x,\xi)$ values for polynomial models. In fact, in the spacing of the designs considered above, with $x_i = i/2$, we note that with equal weights, namely $w_i = 0.05$ for $N = 20$, we calculate

$$\max_x d_1(x,\xi) = 3.71$$

$$\max_x d_2(x,\xi) = 7.41$$

$$\max_x d_3(x,\xi) = 11.25\,,$$

$$\max_x d_4(x,\xi) = 14.52$$

$$\max_x d_5(x,\xi) = 16.94$$

$$\max_x d_6(x,\xi) = 18.74$$

These are, of course, well above the theoretical minima for $\max_x d_k(x,\xi)$ and suggest that when polynomials appear to be reasonable models for the variable of interest, equal spacing and equal weighting designs are not as attractive as intuition suggests.

REFERENCES

1. Kiefer, J. and Wolfowitz, J. (1960), "The Equivalence of Two Extreme Problems," Canadian Journal of Mathematics 12, 363-366.

2. Federov, V.V. (1972), Theory of Optimal Experiments, New York: Academic Press.

3. Cook, R.D., and Nachtscheim, C.J. (1982), "Model Robust, Linear-Optimal Designs," Technometrics 24, 1, 49-54.

4. Tsay, J. (1976), "On the Sequential Construction of D-Optimal Designs," Journal of the American Statistical Association 71, 671-674.

5. Cochran, W.G. (1970), Sampling Techniques, 3rd Edition, New York: John Wiley and Sons.