Robust Model-based Sampling Designs.

Alan Welsh

Mathematical Sciences Institute
Australian National University

Joint work with Doug Wiens
**The Problem**

Consider a finite population of $N$ units in which we have

- an **auxiliary variable** $x$ with population values $x_1, \ldots, x_N$ which are known for all $N$ population units

- a **survey variable** $Y$ with population values $Y_1, \ldots, Y_N$ which are observed (under **complete response**) for a sample $s$ of $n \leq N$ population units

The **estimation/prediction problem** is to use $Y_i, i \in s$ and $x_i, i = 1, \ldots, N$ to estimate or predict the unknown **finite population total** $T = \sum_{i=1}^{N} Y_i$.

The **design problem** is to specify a rule using $x_i, i = 1, \ldots, N$ to **select the sample** of $n$ units so that the estimator or predictor has optimal properties.
**The Sugar Farm Population**

Consists of $N = 338$ sugar cane farms in Queensland (Chambers and Dunstan, 1986). The auxiliary variable $x$ is the area on each farm assigned to growing sugar cane. For calculation, we often use $x / \text{max}(x)$. The survey variable $Y$ is the gross value of cane and the goal is to predict the total gross value of cane from a sample of size $n = 30$. 
**The Ratio Model**

There is known to be an approximately linear relationship between $Y$ and $x$ which we may be willing to describe by the ratio model

$$Y_i = x_i \theta + \sigma_\varepsilon x_i^{1/2} \varepsilon_i, \quad i = 1, \ldots, N,$$

where $\theta$ and $\sigma_\varepsilon > 0$ are scalar, unknown parameters and \{\varepsilon_1, \ldots, \varepsilon_N\} are independent and identically distributed random variables with mean zero and variance one.

Under the ratio model, if sample selection is conditionally independent of the values of $Y$ given the values of $x$ and we have complete response in the sample, the model-based minimum mean squared error predictor of $T$ is the ratio estimator

$$\hat{T} = N \bar{x} \bar{Y}_s / \bar{x}_s,$$

where $\bar{x}$ is the population mean of $x$ and $\bar{x}_s$ and $\bar{Y}_s$ are the sample means of $x$ and $Y$ respectively.
The ratio estimator has prediction mean squared error

\[(1 - n/N)(N^2/n)(\bar{x}_r \bar{x}/\bar{x}_s)\sigma^2_\varepsilon,\]

where \(\bar{x}_r\) is the mean of the nonsample \(x\).

The optimal design is the purposive design which selects the \(n\) units with the largest \(x_i\).
But ...

The optimal design is **optimal when the ratio model holds**.

It is **not necessarily optimal if the ratio model does not hold** (e.g. if the mean does not go through the origin).

The optimal design **does not allow us to check** if the intercept equals zero or the form of the variance function \( g_0(x) = x^{1/2} \). (This can be difficult to check even when the data are collected over a wider range of \( x_i \) values.) The assumption of independence is probably reasonable here.
So ...

We could use **probability sampling** to select the sample.

Can we gain efficiency without going as far as the optimal design?

One approach is to assume that the ratio model is a good approximation (i.e. a **working model**) without necessarily being exactly correct, incorporate uncertainties about the working model and then construct **robust sampling designs** which give **good results** both for the working model and for models which are close to the working model.
**Key steps**

- Define a neighbourhood $\mathcal{M}$ of models $m$ of the working model
- Specify a criterion $\mathcal{L}_N(s,m)$ for choosing a sample $s$ for fitting a model $m$
- Choose $s$ to **minimise** $\mathcal{R}_N(s) = \max_{m \in \mathcal{M}} \mathcal{L}_N(s,m)$

It is important to be able to **compute** $\mathcal{R}_N$ **quickly**. We **maximise** $\mathcal{L}_N(s,m)$ **analytically**, work out how to compute $\mathcal{R}_N$ **quickly** and then **minimise** $\mathcal{R}_N$ using a genetic algorithm.
**MODEL NEIGHBOURHOOD**

The first two moments of $\gamma^{-1}(Y) = (\gamma^{-1}(Y_1), \ldots, \gamma^{-1}(Y_N))^T$ are

$$Z_N \theta + f_N \quad \text{and} \quad \sigma^2_\varepsilon G_N + H_N.$$  

Here

(i) $Z_N$ is a known $N \times p$ matrix with rows $z^T(x_i)$; $\theta$ is an unknown $p$-dimensional parameter and $\sigma_\varepsilon$ is an unknown non-negative scale parameter.

(ii) $f_N$ represents departures from linearity,

$G_N = \text{diag}\{g(x_1), \ldots, g(x_N)\}$ represents departures from

$G_{0,N} = \text{diag}\{g_0(x_1), \ldots, g_0(x_N)\}$

and $H_N$ represents departures from independence.

Subscripts $n$ and $N-n$ denote the submatrices or subvectors corresponding to the sample and nonsample rows or elements, respectively.
We order matrices by positive semidefiniteness and, for finite, positive constants $\tau^2_f, \tau^2_g, \tau^2_h$, we impose the bounds

\[ \mathcal{F} = \{ f : R^q \rightarrow R : Z_N^T f_N = 0 \text{ and } f_N^T f_N \leq \tau^2_f \}, \]
\[ \mathcal{G} = \{ g : R^q \rightarrow R^+ : 0 \leq G_N G^{-1}_{0,N} \leq (1 + \tau^2_g) I_N \}, \]
\[ \mathcal{H} = \{ H : H \text{ positive semi-definite and } ||H|| \leq \tau^2_h / N \}, \]

where $||\cdot||$ is any induced matrix norm such as the maximum absolute row (column) sum or the spectral radius.

Without (something like) these bounds the departures from the working model swamp that of the fitted component asymptotically.

$Z_N^T f_N = 0$ is an identifiability condition.
A Key Result

Suppose that $\mathcal{L}(H)$ is a function of positive semi-definite matrices $H$ which is monotonic with respect to the ordering by positive semi-definiteness, in that

$$H_1 \geq H_2 \Rightarrow \mathcal{L}(H_1) \geq \mathcal{L}(H_2).$$

For any induced matrix norm $\| \cdot \|$, define a class of positive semi-definite matrices

$$\mathcal{H} = \{ H \mid H \text{ positive semi-definite and } \| H \| \leq \tau^2 \}.$$

Then $\mathcal{L}(H)$ is maximized over $\mathcal{H}$ by $\tau^2 I_N$.

A statistical interpretation is that the least favourable situation for modelling dependence is actually the distribution under independence.
THE FORM OF $\mathcal{R}_N$

For $r$ and $s$ functions of the bounds $\tau_f$, $\tau_g$ and $\tau_h$,

$$\mathcal{R}_N = r \cdot \text{ch}_{\max}(M) + s \cdot \text{trace}(G_{0,N} M) + (1 - r - s) \cdot \text{trace}(M),$$

where $M = V^T U V$. Here $U$ and $V$ are the $N \times N$ matrices

$$U = N^{-1} \begin{pmatrix} n^{-1} \text{trace}(G_{0,N-n}) I_n & -n^{-1} 1_n 1_{N-n}^T G_{0,N-n}^{1/2} \\ -n^{-1} G_{0,N-n}^{1/2} 1_{N-n} 1_n^T & I_{N-n} \end{pmatrix},$$

$$V = \begin{pmatrix} (I_n - P_n) G_{0,n}^{-1/2} & 0_n 0_{N-n}^T \\ -Z_{N-n} (Z_n^T G_{0,n}^{-1} Z_n)^{-1} Z_n^T G_{0,n}^{-1} & I_{N-n} \end{pmatrix},$$

where the $n \times n$ matrix

$$P_n = G_{0,n}^{-1/2} Z_n (Z_n^T G_{0,n}^{-1} Z_n)^{-1} Z_n^T G_{0,n}^{-1/2}.$$
Working model: slope and intercept, $g_0(x) = x$
Loss: $r = s = 0.49$.
Best Random: 20,000 designs of size $n = 30$ generated by simple random sampling without replacement. (The process took 59 seconds on MATLAB.)
Optimal robust design.
## Component losses; random and optimal designs

<table>
<thead>
<tr>
<th>Source</th>
<th>Loss</th>
<th>Design</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Best</td>
<td>Random</td>
<td>Optimal</td>
</tr>
<tr>
<td>$f$</td>
<td>$r \cdot ch_{\text{max}}(M)$</td>
<td>0.025</td>
<td>0.025</td>
<td></td>
</tr>
<tr>
<td>$g$</td>
<td>$s \cdot tr(G_{0,N}M)$</td>
<td>0.188</td>
<td>0.177</td>
<td></td>
</tr>
<tr>
<td>$h$</td>
<td>$(1 - r - s) \cdot tr(M)$</td>
<td>0.037</td>
<td>0.033</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>$\mathcal{R}_N$</td>
<td>0.250</td>
<td>0.235</td>
<td></td>
</tr>
</tbody>
</table>
**SIMULATION**

For each design, we generated $y$-values from a model in the neighbourhood of the working model, calculated the population total $T$, fitted the working model to the selected sample data, calculated the predictor $\hat{T}$, and then averaged $(\hat{T} - T)^2$ over simulations to estimate the mean squared prediction error.

Each working and population model used $\gamma^{-1}(y) = \log(y)$ or $y^{1/2}$, but not necessarily the same for both models. We matched the intercept (‘int’) or no intercept (‘no-int’) working model with the actual model, because misspecification in the mean is allowed for through $f_N$. We used $g_0(x) = x$ in all cases.

Robust designs were specified by values of $r$ and $s$ or by choosing $r$ and $s$ so that the three terms in the expression for $\mathcal{R}_N$ contribute approximately a proportion $\beta_1$, $\beta_2$ and $\beta_3$ of $\mathcal{R}_N$. Thus $r$ and $s$ are approximately $r = \beta_1 \mathcal{R}_N / \text{ch}_{\text{max}}(M)$, $s = \beta_2 \mathcal{R}_N / \text{tr}(G_{0,N}M)$. 
The robust designs we considered were:

- **d1)** ‘Robust $[r, s] = [1/3, 1/3]$’: equal relative importance to misspecification of the mean, variance function and independence.

- **d2)** ‘Robust $[r, s] = [0, 1]$’: low concern for possible misspecification of the mean and independence but high concern for the variance.

- **d3)** ‘Robust $\beta = [1/3, 1/3]$’: equal concern for possible misspecification of the mean, variance and independence.

- **d4)** ‘Robust $\beta = [0.48, 0.48]$’: low concern for possible misspecification of independence but equal, high concern for the mean and variance.

These designs were each constructed for both an intercept model and a no-intercept model and are shown in Figure 3 (a) - (h).
Other designs included for comparison:

d5) ‘SRS’: Simple random sampling without replacement.

d6) ‘Largest $n$’: the largest $n = 30$ values of $x$.

d7) ‘Largest and smallest $n/2$’: the largest and smallest $n/2 = 15$ values of $x$. See Figure 3 (i).

d8) ‘D-optimal’ (homoscedastic): the largest 12 and smallest 18. See Figure 3 (j). (With weights inversely proportional to $g_0(x)$ the D-optimal design chooses the largest 14 and smallest 16 values, so is almost identical to d7).

d9) ‘Balanced’: A balanced design generated by the `sampling` library on R, available from the Comprehensive R Archive Network (CRAN). See Figure 3 (k).

d10) ‘Systematic’: A systematic design, again generated by the CRAN `sampling` library on R. See Figure 3 (l).
All 10 designs were applied with data simulated with and without an intercept, and for three combinations of transformations, making 60 sets of simulations. We generated 10,000 populations for each of the 60 cases.

<table>
<thead>
<tr>
<th>Population/sample transformations</th>
<th>log/log</th>
<th>sqrt/log</th>
<th>sqrt/sqrt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>int</td>
<td>no-int</td>
<td>int</td>
</tr>
<tr>
<td>d1) [1/3, 1/3]</td>
<td>295 (31)</td>
<td>45 (1)</td>
<td>153 (2)</td>
</tr>
<tr>
<td>d2) [0, 1]</td>
<td>184 (5)</td>
<td>45 (1)</td>
<td>182 (38)</td>
</tr>
<tr>
<td>d3) $\beta = [1/3, 1/3]$</td>
<td>242 (16)</td>
<td>45 (1)</td>
<td>105 (2)</td>
</tr>
<tr>
<td>d4) $\beta = [0.48, 0.48]$</td>
<td>201 (9)</td>
<td>44 (1)</td>
<td>93 (4)</td>
</tr>
<tr>
<td>d5) SRS</td>
<td>609 (84)</td>
<td>227 (45)</td>
<td>3.6m (1.8m)</td>
</tr>
<tr>
<td>d6) largest $n$</td>
<td>496 (11)</td>
<td>45 (1)</td>
<td>282 (9)</td>
</tr>
<tr>
<td>d7) $n/2$ &amp; $n/2$</td>
<td>190 (12)</td>
<td>70 (5)</td>
<td>168 (22)</td>
</tr>
<tr>
<td>d8) D-optimal</td>
<td>198 (8)</td>
<td>73 (3)</td>
<td>205 (26)</td>
</tr>
<tr>
<td>d9) Balanced</td>
<td>389 (59)</td>
<td>206 (68)</td>
<td>2088 (401)</td>
</tr>
<tr>
<td>d10) Systematic</td>
<td>575 (102)</td>
<td>188 (39)</td>
<td>85K (31K)</td>
</tr>
</tbody>
</table>
With **no intercept in the model** (even columns), the **optimal robust** designs and the ‘largest $n$’ design performed similarly – and well – even though they are quite different.

With **no intercept and the wrong transformation used in the working model** (column 4), SRS, Balanced and Systematic performed well; otherwise, they performed very poorly.

With an **intercept** in the model (odd columns), the **optimal robust $\beta$** designs, particularly $\beta = [0.48, 0.48]$, performed very well; the ‘largest and smallest $n/2$’ design also did reasonably well. As we should expect, the ‘largest $n$’ design performed poorly in these cases. Overall, the **D-optimal** design did not fare well.
CONCLUSION

The performance of all the robust designs, and in particular of the $\beta$ designs, was quite stable across the various situations. Qualitatively, this seems to derive from their greater spread of points throughout the design space. The results suggest that there are substantial gains to be made from using optimal robust designs in place of more naive designs tailored to unrealistically stringent conditions.