

Differential unit weightings

Individual units, may have differential weights attached to them, e.g. as a result of varying sample selection probabilities from a survey. Thus in a 2 level model we may have differential weights attached to the level 2 units.

Weighting in MLn

In MLn a command **weight** was available. It is also present in *MLwiN* version 1.0x, but not documented. It applies a simple set of independent weights to each level 1 unit. This gives the same results for the standard regression weighting in single level models but is not really appropriate for multilevel models. In addition there is a bug associated with it which operates in the following way.

In MLn, whenever START or NEXT is typed with batch mode on, then a weight vector(w1) is formed as $w1 = w^{-0.5}$ where w is the column of user supplied weights, typically all less than unity. There is an incorrect line of program code which takes the square root of this newly formed vector. This line of code is executed at every iteration. The repeated taking of the square root moves the weights vector towards the unit vector and the results finally converge at or very close to the unweighted solution.

If batch is turned off and the model is estimated by repeated typing of the NEXT command then at every iteration weights of $(w^{-0.5})^{0.5}$ are used. This is because repeated typing of NEXT resets w1 at each iteration. Thus repeated typing of NEXT with batch off converges to a different solution than running with BATCH ON. By entering the squares of the required weights and using the next command the intended results will be obtained.

MLwiN drives the background MLn server by running an iteration at the time. Hence we will expect different answers with *MLwiN* run in default mode. The bug does unfortunately mean that MLN weighted analyses in the past were wrong. That is, people would have put in weights found they got very similar or identical answers to unweighted and will have concluded that weights did not make much difference.

A general weighting procedure

We recommend the following. In future releases this will be incorporated transparently within the software.

Two cases need to be distinguished. In the first the weights are independent of the random effects at the level. In this case we adopt the following procedure.

Consider the case of a 2 level model. Denote by w_j the weight attached to the j -th level 2 unit and by w_{ij} the weight attached to the i -th level 1 unit within the j -th level 2 unit such that

$$\sum_i w_{i|j} = n_j, \quad \sum_j w_j = J \quad (1)$$

where J is the total number of level 2 units and $N = \sum_j n_j$ the total number of level 1 units. That is, the lower level weights within each immediate higher level unit are scaled to have a mean of unity, and likewise for higher levels. For each level 1 unit we now form the final, or composite, weight

$$w_{ij} = N w_{i|j} w_j / \sum_{i,j} w_{i|j} w_j = N w_{i|j} w_j / \sum_j n_j w_j \quad (2)$$

Denote by Z_u, Z_e respectively the sets of explanatory variables defining the level 2 and level 1 random coefficients and form

$$\begin{aligned} Z_u^* &= W_j Z_u, \quad W_j = \text{diag}\{w_j^{-0.5}\} \\ Z_e^* &= W_{ij} Z_e, \quad W_{ij} = \text{diag}\{w_{ij}^{-0.5}\} \end{aligned} \quad (3)$$

We now carry out a standard estimation but using Z_u^*, Z_e^* as the random coefficient explanatory variables.

For a 3 level model, with an obvious extension to notation, we have the following

$$\begin{aligned} \sum_i w_{i|jk} &= n_{jk}, \quad \sum_j w_{j|k} = J_k, \quad \sum_k w_k = K, \quad N = \sum_{jk} n_{jk}, \quad J = \sum_k J_k \\ w_{ijk} &= N w_{i|jk} w_{j|k} w_k / \sum_{ijk} w_{i|jk} w_{j|k} w_k, \quad w_{jk} = J w_{j|k} w_k / \sum_{jk} w_{j|k} w_k \end{aligned}$$

Denote by V^* the weighting matrix in this analysis. The fixed part coefficient estimates and their covariance matrix are given by

$$\begin{aligned} \hat{\beta} &= (X^T V^{*-1} X)^{-1} X^T V^{*-1} Y, \\ \text{cov}(\hat{\beta}) &= (X^T V^{*-1} X)^{-1} X^T V^{*-1} V V^{*-1} X (X^T V^{*-1} X)^{-1} \end{aligned} \quad (4)$$

with an analogous result for the random parameter estimates. *MLwiN* does not allow the computation of the covariance matrix estimates directly, but robust or sandwich estimators (Goldstein, 1995) may be used.

We are grateful to Bob Johnson (National Opinion Research Centre, University of Chicago) for pointing out that in survey work analysts often have access only to the final level 1 weights w_{ij} . In this case, say for a 2-level model, we can obtain the w_j by computing $w'_j = W_j J / \sum_j W_j$, $W_j = (\sum_i w_{ij}) / n_j$. For a 3-level model the procedure is carried out for each level 3 unit and the resulting w'_{jk} are transformed analogously.

A number of features are worth noting.

First, for a single level model this procedure gives the usual weighted regression estimator. Secondly, suppose we set a particular level 1 weight to zero. This is not equivalent to removing that unit from the analysis in a 2 level model since the level 2 (weighted) contribution remains. Nevertheless, this weighting may be appropriate if we wish to remove the effect of the unit only at level 1, say if it were an extreme level 1 outlier. If, however, we set a level 2 weight to zero then this is equivalent to removing the complete level 2 unit. If we wished to obtain estimates equivalent to removing the level 1 unit we would need to set all the level 2 (random coefficient) explanatory variables *for that level 1 unit* to zero also. This is easily done by defining an indicator variable for the unit (or units) with a zero corresponding to the unit in question and multiplying all the random explanatory variables by it.

In calculating residuals we may also wish to use the weights. This leads to the following results (using the notation in Goldstein (1995) for the level 2 residuals

$$\begin{aligned}\hat{p}_2 &= \Omega_2 Z_u^{*T} V^{*-1} \tilde{Y}, \\ \text{cov}(\hat{p}_2) &= \Omega_2 Z_u^{*T} V^{*-1} (V) V^{*-1} Z_u^* \Omega_2 \\ V &= E(\tilde{Y} \tilde{Y}^T)\end{aligned}\tag{5}$$

This provides a consistent estimator of the covariance matrix. Alternatively, we may use a sandwich estimator for the covariance matrix. For many purposes an unweighted estimator for the residuals is adequate, in which case the usual formulae apply and the *MLwiN* residuals window can be used.

A similar procedure applies for multilevel generalised linear models. Here the weighted explanatory variables at levels 2 and higher are as above. For the quasiliikelihood estimators (PQL and MQL) at level 1 the vector Z_e is that which defines the binomial variation. Thus, for binomial data, at level 1 a method of incorporating the weight vector is to use Z_e but to work with $w_{ij} n_{ij}$ instead of n_{ij} as the denominator.

The second situation is where the weights are not independent of the random effects at a level. This leads to complications which are discussed by Pfefferman et al. (1997). These authors conclude that, in this situation, the above procedure produces acceptable results in many cases but can give biased results in some circumstances and should be used with caution.

References

Goldstein, H. (1995). *Multilevel Statistical Models*. London, Edward Arnold: New York, Wiley.

Pfeffermann, D., Skinner, C. J., Holmes, D., Goldstein, H., et al. (1997). Weighting for unequal selection probabilities in multilevel models. *Journal of the Royal Statistical Society, B*, 60, 23-40.