Diego ZAPPA

CALIBRATION CONFIDENCE REGIONS USING EMPIRICAL LIKELIHOOD AND THE VBA-EXCEL TOOL
Calibration confidence regions using Empirical Likelihood and the VBA-Excel tool

Diego Zappa
Istituto di Statistica
Università Cattolica del Sacro Cuore - Milano
e-mail: diego.zappa@unicatt.it

Keywords: Multivariate Calibration, Empirical Likelihood.

1. Introduction

Statistical calibration deals with the inference on the unknown values of explanatory variables given a vector of response variables. It results particularly useful when it is much simpler (or much more economically convenient) to measure (or to fix) the responses of an experiment and then to find the levels of the explanatory variables that may have produced them. Examples may be found in chemistry, biometrics, where the interest is on the causes of the experiments, or in engineering where targets must be suitably chosen to let the process work.

In Section 2 we briefly describe the multivariate calibration problem, stressing some of the difficulties in the construction of confidence regions in a parametric context, and some recent non parametric proposals. In Section 3, using a semiparametric approach, a new methodology for calibration is proposed, based on empirical likelihood. Then in Section 4 an application of the proposal to a semiconductor process is presented. In the Appendix it is described the Excel spreadsheet and the related VBA program to make effective the proposal.
2. Multivariate Calibration: object, problems, recent proposals

A brief introduction to multivariate calibration is deemed necessary for the remainder of the paper. Many references can be found e.g. in Brown (1993), Naes et alii (2002) and Zappa, Salini (2005).

According to most of the literature on this topic, we consider two steps.

1) The calibration step. An experiment is run. \( n \) observations on \( q \) response variables \( Y = \{ Y_1, Y_2, ..., Y_q \} \) and \( p \) explanatory variables \( X = \{ X_1, X_2, ..., X_p \} \) are collected in the matrices \( Y_1 \) and \( X \). A vector \( g(\cdot) (1 \times q) \) of (not necessarily invertible) transfer functions is used to link the two sets of variables. \( g(\cdot) \) is usually supposed to be composed of linear models. Let \( E_1 \) be a \((1 \times q)\) vector of random variables (r.v.s). This vector will represent measurement errors, and we will suppose, for the sake of simplicity, that it is additive with respect to \( g(\cdot) \). Then the calibration model is defined by:

\[
Y_1 = g(\Theta, X) + E_1
\]

where \( \Theta \) is a matrix of unknowns parameters.

2) The prediction step. Analogously to the previous step, suppose that a \( Y_2 \) \((m \times q)\) set of response data is available, where \( m \) represents the number of replications of the experiment in unknown homogeneous conditions. Then the prediction model is

\[
Y_2 = \hat{g}(\hat{\Theta}, 1 \times \xi) + E_2
\]

where \( \hat{\Theta} \) has been estimated at step 1, \( E_2 \) is a \((m \times q)\) matrix of measurement errors and we are interested in the unknown levels \( \xi \) of the \( X \) variables. Independently from the properties of \( g(\cdot) \), some problems may occur when \( q > p \). Using maximum likelihood based on multinormal distributional assumptions, it may be shown that the confidence region for \( \xi \) depends on a quantity usually called \( R \), which is a measure of the unreliability of the \( Y_2 \) sample set to calibrate \( \xi \). Among others, the main problem is that the confidence region has an anomalous behaviour with respect to \( R \): it increases as \( R \) decreases and decreases as \( R \) increases. When \( q < p \) the problem may have no solution (see Brown 1993 for details), mainly because the space spanned by \( \xi \) is greater than the one spanned by \( Y_1 \). Thus no uniqueness in the solution arises. Different additional problems, mostly related to the maximum likelihood procedure, are due to the nature of \( X \): if it is supposed to be fixed we will talk of controlled calibration; if it is random, calibration will be said to be uncontrolled.

Zappa and Salini (2005) present a semiparametric solution to some of the above problems using a data depth procedure (see e.g. Liu and Singh, 1993). Their proposal requires no distributional assumptions, nor the choice of a linear transfer function, as well as the application of no multivariate technique in order to reduce the complexity of the problem. Besides, a non-empty, non-infinity confidence region is found for \( q \geq p \). This proposal has two major properties: 1) all the information included in the set of variables is used, and 2) a preliminary (non-parametric) test is run to verify if the
hypothesis of linear relationship between the set of variables is true. The counterpart of this approach (like most of the statistics that use a depth function) is the relevant computational effort needed. At present no sufficiently powerful (fast and reliable) software has been prepared, and most of the available algorithms (as the one specifically implemented by the authors) have been programmed for research reasons. Additionally, it may be used only for uncontrolled calibration experiments and it does not apply when the dataset is small.

All these considerations will support the following new proposal.

3. Nonparametric calibration regions based on Empirical likelihood

Many are the references on empirical likelihood (EL). The most relevant is certainly Owen (2001) where historical background, main results, applications in very different fields and computational details may be found.

Briefly on EL: let \( F_z \) be the distribution function of the continuous r.v. \( Z \), and \( Z_n \) be an iid sample from \( Z \). Supposing that \( w_i \) is the weight that \( F_z \) places on observation \( z_i \), define the EL for the whole sample as \( \prod w_i \); the maximum is achieved for \( w_i = 1/n \forall i \) and then the empirical likelihood ratio will be \( R(F_z) = \prod n w_i \). Starting from these preliminaries, we may define the profile likelihood ratio of a vector of parameters \( \theta \) and its corresponding EL confidence region. Supposing that \( T(F) = \sum w_i h(Z_n) \) is an estimator of \( \theta \), where \( h(\cdot) \) is a vector of functions that may itself be dependent on \( F_z \), then, on the domain of \( \theta \), the empirical profile likelihood will be defined as

\[
R(\theta) = \max(\prod n w_i \mid T(F_z) = \theta, w_i \geq 0, \sum w_i = 1)
\]  

(3)

and the corresponding EL confidence region will be

\[
\{ \theta \mid R(\theta) \geq r_0 \} = \{ T(F_z) \mid \prod n w_i \geq r_0, w_i \geq 0, \sum w_i = 1 \}
\]  

(4)

where \( r_0 \) must be chosen such that the coverage probability equals a chosen \((1-\alpha)\) probability level. Theorem 3.6 in Owen (2001) shows that

\[
-2 \log R(\xi) \overset{d}{\to} \chi^2_p \quad \text{as} \quad n \to \infty
\]  

(5)

(where \( \overset{d}{\to} \) means convergence in distribution) which is a result analogous to Wilk’s theorem. Then \( r_0 \) will be, at least asymptotically, the \((1-\alpha)\) percentile of \( \chi^2_p \). More generally, the constraints in (3) may be substituted by what Owen calls “estimating equations”, that is a vector of functions, \( m(\cdot) \), such that \( \sum \sum w_i m(z_i, \theta) = 0 \).

To solve (3) some routines are actually freely available in \( R^\circ \), \( Gauss^\circ \) and very probably in other program language. One of the contribution of this work is the implementation of a routine freely available from the Author to solve (3) in VBA-Excel\(^\circ\). An advantage of this software is the possibility of being used by most researchers. Details of the algorithm and of the implementation of the dual problem of (3) are reported in the Appendix.
The application of EL to calibration problems may present some difficulties and at least two problems.

The difficulties concern the implementation of the constraints. In the prediction step, stacking equations (1) and (2) and assuming to keep the estimate of $\Theta$ fixed, the unknown $\xi$ must be found subject to the constraints needed to find the solution to the unknowns in (1). As it is customary, supposing that a standard least squares is implemented in (1) and supposing that $g(\cdot)$ is at least twice differentiable, we should include in (3) the following set of $(q \cdot p \times q)$ constraints (these are the estimating equations) corresponding to the matrix of gradients equated to zero:

$$
\begin{bmatrix}
\frac{\partial g(\Theta)}{\partial \Theta} \left[ 1 \cdot \xi \right] X \end{bmatrix}_{(m+n) \times q} \circ W = 0
$$

where the first argument is a $(q \cdot p) \times (m+n)$ matrix, $I$ is a $(m \times 1)$ vector of ones and $\circ$ is the Hadamard product. More generally if equations (1) and (2) were a set of generalized linear models with random structure belonging to the exponential family, then with some little additional complications we should have constraints like (6) (for the application of EL to GLM see Kolaczyk, 1994). Note also that the matrix $W$ may be thought of either $q$ different columns $[w_1, w_2, \ldots, w_q]$ of vectors of weights or $W = w_1 \cdot I^T$ where $I$ is $(q \times 1)$. In the former case we assume that the $q$ models in (1) are to be estimated independently, that is $q$ different experiments may be run. Then the maximum in (3) becomes $\prod_{i=1}^{n+m} \prod_{j=1}^{(n+m)w_j}$. In the latter case, as it is much more frequent, we assume that only one experiment is run and then the $q$ models are jointly examined and then estimated. Now, supposing that (1) and (2) are models linear in the parameters, with obvious notation we may write

$$
Y_1 = \alpha + XB + E_1
$$

$$
Y_2 = 1^\wedge \alpha + 1 \cdot \xi^\wedge B + E_2
$$

The constraints (6) become

$$
\begin{bmatrix}
1 \\
1 \\
(1 \times 1) X
\end{bmatrix}_{(n+m) \times q} \circ W = 0
$$

where $\otimes$ is the Kronecker product. If $q=1$ and $W = w_1 \cdot I^T$, (9) corresponds to:

$$
\begin{align}
\sum_{j=1}^{m} e_g w_g + \sum_{i=1}^{n} e_i w_i &= 0 \\
\sum_{j=1}^{m} \xi_j e_g w_g + \sum_{i=1}^{n} x_i e_i w_i &= 0 \\
\sum_{j=1}^{m} \xi_j p e_g w_g + \sum_{i=1}^{n} x_p e_i w_i &= 0
\end{align}
$$

where

$$
e_g = (y_2 - \hat{\alpha} - \sum_{s=1}^{p} \hat{\beta}_s \xi_s)
$$

$$
e_i = (y_1 - \hat{\alpha} - \sum_{s=1}^{p} \hat{\beta}_s x_i)
$$
and $y_1, y_2$ are, respectively, the data coming from the calibration and the prediction step, and $w_{\xi_j}$ is the weight of $e_{\xi_j}$. The dual problem concerning the log of (3) with the constraints (10) is shown in the Appendix. Additional constraints, e.g. on the deviance of the residuals, may be added (see §3.2 for some details). Without loss of generality, in the rest of this paragraph we will make use only of the constraints in (10).

The first problem is that, given $y_2$, not unique solutions to (8) may exist: all the $\xi$s that belong to the calibration model and that satisfy the constraints in (10) will have $R(\xi)=1$. The following remark gives a solution to this problem.

**Remark:** Using (9) in (3), the calibration region will be closed at least in probability.

Suppose for simplicity and without loss of generality that $q=m=1$. Suppose that the variables have been centred (this is often the case especially in controlled calibration): then $\hat{\alpha}=0$. Let $d=||\xi-X||_2$ be the distance of $\xi$ from the observed set $X$. As $X$ is fixed, it is sufficient that, e.g., the $j$-th element of $\xi$ diverges, so that $d\to\infty$. It follows that $\lim_{||\xi||_2\to\infty} \xi_j e_{\xi_j}=\infty$: that is, if the whole vector $\xi$ does not simultaneously change as $\xi_j$ increases, the contribution to the estimating equations in (10) goes to infinity. In order to fulfil the constraints to zero in (10), the weight $w_{\xi\to0}$ and then $R(\xi)\to0$. But even for small fixed $e_{\xi_j}$, that is even if we choose the whole vector $\xi$ systematically close to the model, the previous limit will be $\infty$ and then $R(\xi)\to0$. These considerations support the conjecture that $\xi$s too far from $X$ are not good calibrating solutions and this will be exploited in the following. Then, as $d\to\infty$ we may have only $R(\xi)\geq0$ or $R(\xi)=1$ respectively, if $\xi$ does or does not belong to the model (that is for a subset of measure zero). In the latter situation the contribution to the estimating equations in (10) is zero, as it is shown in the next equation

$$\lim_{||\xi||_2\to\infty} \xi_j (y_{21} - \hat{\beta}^T \xi) = \lim_{||\xi||_2\to\infty} \left[-\hat{\beta}_j/(y_{21} - \hat{\beta}^T \xi)^2 \right]^{-1} = 0 \quad (11)$$

and then $w_i=1/(n+1)$ $\forall i$. Obviously for small or moderate $d$ the contribution to the equations in (10) is at least moderate and the weights will not be necessarily all zeros. Thus the problem of the closure of the confidence region, at least in probability, for any combination of $q\leq p$ is solved. Then, once $R(\xi)$ has been computed, we may exploit (5) to operatively find the confidence region.

The second problem is that, if e.g. $p>q$, then the estimates satisfying $R(\xi)=1$ may be infinite, and the choice of a unique calibration point appears indeterminate. We propose the introduction of a penalty function that will downweight those $\xi$ which will increase the overall entropy to the joint sets $(X,Y_i)$ collected in the calibration step. Consider that this step is usually run in a very accurate and precise conditions. Then it seems reasonable that any solution “far” from the set $(X,Y_i)$ may result not fully accurate. This proposal introduces a measure coherent with the previous considerations based on $d$ and similar to the inconsistency measure in the classical approach: the greater the entropy the less is the prediction ability of the $Y_2$ set with respect to $Y_1$. 


Let \((X_\xi Y) = \begin{bmatrix} 1 & \xi & Y_2 \\ X & Y_1 \end{bmatrix}\). To measure entropy, we have chosen the global variability of the second order

\[
\Delta_2(X_\xi Y) = \left( \sum_{i,j}^{n+m} [(x_\xi y)_i - (x_\xi y)_j]^2 w_i w_j \right)^{0.5}
\]

(12)

where \((x_\xi y)_{i,j}\), for \(i,j=1,...,n+m\), is the \(i,j\)th row of the matrix \((X_\xi Y)\). \(\Delta_2\) does not need the estimate of any additional parameter and it is based on Euclidean distances. Note that the distance between the \(i\)th and the \(j\)th row is weighted by the estimated \(w\). Then it balances the increase in \(d\) and the relative importance of the new \(\xi\) with respect to the set \(X\). Moreover, a property of (12) is that, for any \(\xi\) such that \(w_\xi = 0\), \(\Delta_2(X_\xi Y) = \text{const}\), that is any solution far from the calibration set is to be considered indifferent for \(\Delta_2\). This proposal may be applied also to the Maximum likelihood (ML) approach, but in this case no weights may be used in (12).

3 An example: the Critical Dimension process (CD process)

3.1 Introduction

In microelectronics\(^1\), calibration is one of the key methodology to guarantee the highest precision and, where possible, a reduction of costs. Many are the steps requested to produce what is elementarily called a chipset. Among these very many steps the so called Critical Dimension (CD) process is considered (as its name suggests) very critical. This is also due to the progressive shrinking of CD. In very few words, CD concerns the control of a lithography process whose aim is to "print" on a wafer surface the map of geometric structures and to control that the distance or the shape of structures, such as channels, boxes, holes etc., has a particular profile. See Fig. 1 for an example of profile on a chipset.

![Fig. 1: An example of profile](image)

\(^1\) I wish to thank Gabriella Garatti of STMicroelectronics in Agrate for having proposed me the CD problem, provided the dataset and helped me in giving a simple technical explanation of the problem.
To control this process key factors are: image contrast, image focus and exposure dose. Visual test based methods have played significant roles in both production and development environments. For example, the use of checking completely developed and cleared photoresist patterns from a dose focus matrix is very common in semiconductor industry. While visual tests are easy to implement, they are not easy to automate. Scanning electron microscopes (SEM) can measure patterned features. However they are very expensive, and can be either time-consuming or destructive, and thus not suitable for run-to-run monitoring. Moreover, electrical measurements can provide information on final CD linewidths, but cannot provide reliable resist profile information. Besides, the airborne base contamination of chemically amplified photoresist is a yield-limiting factor in deep UV lithography and will remain so as device features continue to shrink.

Because of all the above reasons, the availability of a reliable calibration algorithm is needed in order to find the best treatment for pre-chosen CDs.

The real data in the very small dataset reported in Tab. 1 have been masked because of industrial privacy. We can say only that the original data have been translated and rotated. The variables are:

- \( X_1 \) is the Dose variable (3 levels)
- \( X_2 \) is the Image Focus (3 levels)
- \( X_3 \) is the covariate Airborne base contamination
- \( Y \) is the response, CD.

At present no optimal experimental procedure has been implemented. For the sake of simplification, and according to the standard literature for this process, a linear combination of the variables has been preferred. The estimate of the parameters in model (7) applied to data in Tab. 1, gives

\[
\hat{Y} = -1.117 -0.337 \cdot X_1 -0.346 \cdot X_2 + 1.757 \cdot X_3
\]

(13)

with residual variance \( \hat{\sigma}^2 = 0.1315 \). The \( x_3 \) unknown level (Fixed) will be fixed equal to 0 and we will choose for \( Y \) the Target = 0. This means that the expected best prediction model is
\[ 0 = -1.117 -0.337 \cdot X_1 -0.346 \cdot X_2 \]  

(14)

3.2 Application of the constraints in (9)

The output of the VBA program applied to the above dataset is reported in Fig. 2 and Fig. 3. For this calibration problem the parameters, \( n,p,q,m \) are 9,2,1,1 respectively plus one covariate. Note that \( n \) is very small and that \( p>q \), that is the two problems described in §3 are present.

Looking at the profile EL in Fig. 2, it turns out, as expected, a locus of maximum corresponding to the locus of the estimated model plus an unexpected local suboptimum in the neighbour of (0,0), where some relevance should be attributed to a region a bit far from the calibration model. Such a region is absolutely absent in the ML profile. From the corresponding contour plot some empirical evidence appears on the asymptotic closure of the confidence region applying (5), while in no way this will happen using the ML approach. Observing Fig. 3, the \( \Delta_2 \) surface in EL is evidently different from the one in ML where no weights are attributed to the data. Observing the contour plot of \( R(\xi)/\Delta_2 \) it can be found that the optimum calibration point is (-1.58,-1.63) for EL and (-1.51,-1.52) using ML.

<table>
<thead>
<tr>
<th>( R(\xi) )</th>
<th>EL</th>
<th>ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contour plot of ( R(\xi) )</td>
<td>![Contour plot of R(\xi) for EL]</td>
<td>![Contour plot of R(\xi) for ML]</td>
</tr>
</tbody>
</table>

**Fig. 2:** \( R(\xi) \): Profile EL and ML and their contour plot
In order to give some insights to the region around the origin, we have made use of simulation to study the behaviour of EL around (0,0). We have simulated 100 times the experiment generating the value of target form a $N(\hat{\alpha}, \hat{\sigma}^2)$. The $R(\xi)$ and $R(\xi)/\Delta_2$ surfaces which result as the average of the 100 EL profile surfaces are almost the same. This is the reason for which we report in Fig. 4 only $R(\xi)$ (and at the right the estimated standard deviation of the surface). The maximum of the averages is around the coordinate (0,0.6) with $R(\xi)=0.86$ and the 3rd quartile equals the high value of 0.951. A significant reduction of variance is evident around (0,0.6) which may be considered a very good candidate to be the best calibration value for the average value.
of $-1.117$ even if the point calibrated value using (14) returns the slightly bias of $-0.346 \cdot 0.6 = -0.208$.

The bias is almost expected, because the constraints in (9) refer only to the estimated coefficients of the model in (7). This choice is mainly due to the sake of simplicity but the addition of new constraints is relatively simple as it will be described later. For example using only (9) the residual variance will not be forced to be equal to the one estimated in the calibration step and then a sort of incoherence with the application of the prediction step may arise (we refer to the search of calibration points or region in (8) subject to any or most of the parameters estimated in (7): e.g. the residual variance computed using (8) must equal the one estimated in (7)). In fact without specific constraints on the residual variance, Fig. 5a and Fig. 5b report the plot of the weighted and unweighted residual variance respectively, computed on the same domain of the EL in Fig. 3. In both figures we have subtracted 1.1315 that is the estimated residual variance of model (7). It turns out that the weighted variance differs from 1.1315 not more than 0.125 that is about 11%. As this percentage is not small, then we have added to (10) also the constraint

$$
(e_{\xi}^2 w_{\xi} + \sum_{i=1}^{n} e_{i}^2 w_{i}) - 1.1315 = 0
$$

Fig. 6 reports the resulting surface and the contour plot of $R(\xi)/\Delta^2$. It appears clear that this surface is very similar to the one in Fig. 3 but the optimum calibration point is $\{-1.48, -1.51\}$. Using the same simulation experiment described above, The maximum of the averages is around the coordinate (-0.4,0.6) with $R(\xi)=0.81$ and with the bias of -0.073 which is less than the one found in the previous experiment.

In Fig. 7 we report the surface of the difference between the two approaches (the former minus the latter) which is positive almost everywhere: the introduction of (15) has almost obviously reduced the EL. Note the strange behaviour of the surface. A justification may be found in Fig. 8, which is the scatter plot of the variable in Fig. 5(a) and Fig. 7. The greater is the difference between the estimated variance and the expected one, the greater is the difference in EL, that is the less will be the estimated weight for that peculiar solution.

![Fig. 4](image-url)  
\(a\) Average and \(b\) standard deviation of $R(\xi)$ for 100 replications of the experiment
Fig. 5: (a) weighted residual variance (b) unweighted residual variance

Fig. 6: (a) EL surface and (b) contour plot of $R(\xi)/\Delta_2$ using (10) and (15)

Fig. 7: Difference of the EL surface in Fig. 3 minus the EL surface in Fig. 6.

Fig. 8: Scatter plot of the variable in Fig. 5(a) (x axis) and Fig. 7 (y axis)
4 Conclusions

Empirical likelihood seems to be very promising for a variety of problems. A relevant property is the possibility to exploit some standard results of the parametric likelihood theory in a non parametric context. Certainly the main limit to a wide application of EL is the availability of a computational tool. To solve this limit we hope the freeware spreadsheet and the connected VBA program we have implemented will be a useful tool in making EL wider known, especially, as it has been in our experience, in those communities where small dataset, very high experimental costs and the difficulty to test parametric assumptions are very frequent in the daily labour. At present it has been specifically prepared for the calibration problem but it can be easily adjustable to many others contexts.

5 APPENDIX: EL computational tool in Excel

5.1 The dual problem of (3)

Computationally, the problem of finding the maximum of (3) with the constraints (10) can be more efficiently solved by implementing a dual problem. (10) corresponds to

\[
\begin{bmatrix}
e_{\xi_1} & \ldots & e_{\xi_m} & e_1 & \ldots & e_n \\
\xi_1 e_{\xi_1} & \ldots & \xi_1 e_{\xi_m} & x_{11} e_1 & \ldots & x_{1n} e_n \\
\vdots & \ldots & \vdots & \vdots & \ldots & \vdots \\
\xi_p e_{\xi_1} & \ldots & \xi_p e_{\xi_m} & x_{p1} e_1 & \ldots & x_{pn} e_n
\end{bmatrix}
\begin{bmatrix}
w_{\xi_1} \\
\vdots \\
w_{\xi_m} \\
w_1 \\
\vdots \\
w_n
\end{bmatrix} = 0
\]

that will be written as \(X_{e_{\xi}}^T \cdot w_{\xi} = 0\). Let \(A_{nxm}^* = \{a_{ij}^{-1}\}\) for any cells \(i,j\). Then, using the generic hints of Owen (2001) pag.60, the peculiar solution for our calibration problem is

\[
\frac{1}{n+m} x_{z_{\lambda}}^* \quad \text{where} \quad x_{z_{\lambda}} = 1 + X_{e_{\xi}} \lambda
\]

(16)

and \(\lambda\) (a \(p+1\) vector) is a Lagrange multiplier that satisfies the \(p+1\) equations

\[
\frac{1}{n+m} [(X_{e_{\xi}} \odot (x_{z_{\lambda}}^* \mathbf{1})^{(n+m)+1}) \mathbf{1}] = 0
\]

The duality results by substituting (16) in (the log of) (3)

\[
\log \mathcal{R}(\lambda) = - [\log (x_{z_{\lambda}}^{*})]^T \mathbf{1}
\]
which must be minimized over $\lambda$ subject to the constraints that the argument of log is positive and that $\Sigma w_i = 1$.

5.2 The Excel spreadsheet needed for the EL calibration macro (dual problem)

Before giving some insights to the VBA macro used for the computations of the example described in §3, some details must be given on the spreadsheet, where data and some formulas have to be stored. We illustrate how (16) has been computed. The implementation of the problem (3) subject to (10) is reported at §5.4. The choice of giving much more emphasis to the dual problem is due to a greater numerical stability ($q$ are the unknown parameters) then the direct problem ($m+n$ are the unknown parameters). Fig. 9 represents a generic spreadsheet where in the range A2:D11 the data needed by the EL problem are stored (the notation reflects the one used throughout the paper) and where bold reference to a cell (e.g. A14) means that a Formulas, reported below, must be stored.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X1</td>
<td>X2</td>
<td>X3</td>
<td>Y</td>
<td>weight</td>
<td>w_scaled</td>
</tr>
<tr>
<td>2</td>
<td>$\xi_1$</td>
<td>$\xi_2$</td>
<td>x3_fixed</td>
<td>Y2_fixed</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>X11</td>
<td>x21</td>
<td>x31</td>
<td>y11</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>X12</td>
<td>X22</td>
<td>x32</td>
<td>y12</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>X13</td>
<td>X23</td>
<td>x33</td>
<td>y13</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>6</td>
<td>X14</td>
<td>X24</td>
<td>x34</td>
<td>y14</td>
<td>E6</td>
<td>F6</td>
</tr>
<tr>
<td>7</td>
<td>λ_0</td>
<td>l_b0</td>
<td>1_b0</td>
<td>1</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>8</td>
<td>λ_1</td>
<td>l_b1</td>
<td>1_b1</td>
<td>1</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>9</td>
<td>λ_2</td>
<td>l_b2</td>
<td>1_b2</td>
<td>1</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>10</td>
<td>λ_3</td>
<td>l_b3</td>
<td>1_b3</td>
<td>1</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>11</td>
<td>obiett b0</td>
<td>B22</td>
<td>Sum of obiett for numerical control</td>
<td>If numerical problems exist</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Constraints</td>
<td>B17</td>
<td>EL (R(xsi))</td>
<td>E17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>obiett b0</td>
<td>B23</td>
<td>C23</td>
<td>D23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>obiett b1</td>
<td>B24</td>
<td>D24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>obiett b2</td>
<td>B25</td>
<td>D25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 9 Excel spreadsheet for problem in Tab. 1
Variables used in Formulas
RES_MQ := (Y-b0-b1*X1-b2*X2-b3*X3)
res_mq_i := ith element of the vector RES_MQ

Formulas
E6: \(=1/10^2(1+b0*\text{res}_mq_4+b1*X14*\text{res}_mq_4+b2*X24*\text{res}_mq_4+b3*X34*\text{res}_mq_4)^{-1}\)
F6: \(+\text{weight} / \text{SOMMA(weight)}\)
G6: \(=\text{if}(w_{scaled} \leq 0; 1; 0)\)
A14: \(=\text{ASS(obiett b0+obiett b1 + obiett b2 + obiett b3)}\)
B17: \(=\text{SOMMA(vinc w pos)}\)
E17: \(=\text{PRODOTTO(n*(w_{scaled}))} \quad [n \text{ in our example is 10]}\)
B22: \(=\text{SOMMA}(\text{RES}_MQ)/(1+b0*(\text{RES}_MQ)+b1*X1*(\text{RES}_MQ)+b2*X2*(\text{RES}_MQ)+b3*X3*(\text{RES}_MQ)))/10\)
B23: \(=\text{SOMMA}(X1*(\text{RES}_MQ)/(1+b0*(\text{RES}_MQ)+b1*X1*(\text{RES}_MQ)+b2*X2*(\text{RES}_MQ)+b3*X3*(\text{RES}_MQ)))/10\)
B24: \(=\text{SOMMA}(X2*(\text{RES}_MQ)/(1+b0*(\text{RES}_MQ)+b1*X1*(\text{RES}_MQ)+b2*X2*(\text{RES}_MQ)+b3*X3*(\text{RES}_MQ)))/10\)
B25: \(=\text{SOMMA}(X3*(\text{RES}_MQ)/(1+b0*(\text{RES}_MQ)+b1*X1*(\text{RES}_MQ)+b2*X2*(\text{RES}_MQ)+b3*X3*(\text{RES}_MQ)))/10\)
C23: \(=\text{SOMMA}(\text{ASS(B22:B25)})\)
D23: \(=\text{SOMMA}(\text{ASS(B22)})\)
D24: \(=\text{SOMMA}(\text{ASS(B22:B23)})\)
D25: \(=\text{SOMMA}(\text{ASS(B22:B24)})\)

5.3 The VBA macro

`Matrix where results will be stored`
Dim MATRICE(10000, 6)
`the width of the EL domain: in the example it is symmetric and centred in (0,0)`
ampiez = width
`the grid of the EL surface`
precisione = numb_steps
stepxsi1 = 2*width / numb_steps
stepxsi2 = 2*width / numb_steps
incrxsi2 = 0
Do While incrxi2 <= 2*width
  Range("b2").Select
  ActiveCell.FormulaR1C1 = -width + incrxi2
  Incrxi1 = 0
  Do While incrxi1 <= 2*width
    Range("a2").Select
    ActiveCell.FormulaR1C1 = -width + incrxi1
    Range(Cells(18, 2), Cells(21, 2)) = "0"
  End Do
  Call SolverReset
  Call SolverOK(SetCell:=Range("a14"), MaxMinVal:=3, valueof:=0,
    ByChange:=Range("b18:b21"))
  Call SolverAdd(Range("b17"), 2, ",=0")
  Call SolverOptions(Derivatives:=2, precision:=0.000000001, Iterations:=1000)
Call SolverSolve(UserFinish:=True)

contr = Range("c23").Value
' If Solver does not find an appropriate solution in 1 step only try a sequential conditional procedure in 4 steps
If contr > 0.01 Then

'Loop 4 times the previous Call using in SolverOk
'1-step: SetCell:=Range("d23")
'2-step: SetCell:=Range("d24")
'3-step: SetCell:=Range("d25")
'4-step: SetCell:=Range("a14")

End If
contr = Range("c23").Value
If contr > 0.01 Then
' Pause the program or print somewhere a warning
End If

'Save in Matrice cells: A2, B2, F2, E17, C23
incrxs1 = incrxs1 + stepxsi1
Loop
Incrxs2 = incrxs2 + stepxsi2
Loop
'Save MATRICE somewhere and draw the EL surface

Note that the Call Solver corresponds to a repeated use of the next Solver (Italian version) interface
5.4 The Excel spreadsheet needed for the EL calibration macro *(direct problem)*

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X1</td>
<td>X2</td>
<td>X3</td>
<td>Y</td>
<td>weight</td>
<td>vinc w pos</td>
</tr>
<tr>
<td>2</td>
<td>ξ1</td>
<td>ξ2</td>
<td>x3_fixed</td>
<td>Y2_fixed</td>
<td>w1</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>X11</td>
<td>x21</td>
<td>x31</td>
<td>y11</td>
<td>z\</td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>X12</td>
<td>x22</td>
<td>x32</td>
<td>y12</td>
<td>w3</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>X13</td>
<td>x23</td>
<td>x33</td>
<td>y13</td>
<td>w4</td>
<td>...</td>
</tr>
<tr>
<td>6</td>
<td>X14</td>
<td>x24</td>
<td>x34</td>
<td>y14</td>
<td>w5</td>
<td>F6</td>
</tr>
<tr>
<td>7</td>
<td>X15</td>
<td>X25</td>
<td>x35</td>
<td>y15</td>
<td>w6</td>
<td>...</td>
</tr>
<tr>
<td>8</td>
<td>X16</td>
<td>X26</td>
<td>x36</td>
<td>Y16</td>
<td>w7</td>
<td>...</td>
</tr>
<tr>
<td>9</td>
<td>X17</td>
<td>X27</td>
<td>x37</td>
<td>Y17</td>
<td>w8</td>
<td>...</td>
</tr>
<tr>
<td>10</td>
<td>X18</td>
<td>X28</td>
<td>x38</td>
<td>Y18</td>
<td>w9</td>
<td>...</td>
</tr>
<tr>
<td>11</td>
<td>X19</td>
<td>X29</td>
<td>x39</td>
<td>Y19</td>
<td>w10</td>
<td>...</td>
</tr>
</tbody>
</table>

**Comment:** CELL A14 is the objective of Solver (EL)

**Constraints**

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>Positive?</td>
<td>B17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Sum to 1?</td>
<td>B18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>obiett b0</td>
<td>B19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>obiett b1</td>
<td>B20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>obiett b2</td>
<td>B21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>obiett b3</td>
<td>B22</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Variable used in Formulas**

RES_MQ := (Y-b0-b1*X1-b2*X2-b3*X3)

**Formulas**

\begin{align*}
F6: &= +SE(weight<=0;1;0) \\
A14: &= +PRODOTTO(n*(weight)) \quad [n \text{ in our example is } 10] \\
B17: &= +SOMMA(SE(weight<=0;1;0)) \\
B18: &= +SOMMA(weight) \\
B19: &= +ASS(SOMMA(weight*(RES_MQ))) \\
B20: &= +ASS(SOMMA(weight*X1(RES_MQ))) \\
B21: &= +ASS(SOMMA(weight*X2(RES_MQ))) \\
B22: &= +ASS(SOMMA(weight*X3(RES_MQ)))
\end{align*}
Then Solver must be programmed as reported in the next picture

![Solver Interface](image)

**References**


