Title
Terraces: An XLISP-STAT Program for Multilevel Modeling with Diagnostics

Permalink
https://escholarship.org/uc/item/8f9585tk

Author
James Hilden-Minton

Publication Date
2011-10-25
References


7 Final Remarks

TERRACE is a work in progress still very much at an embryonic stage of development. It is intended as an experimental platform for researching diagnostic and computational approaches to multi-level modeling, and no claims of accuracy are made. I welcome comments, questions, suggestions, and reports of any bugs.

In the near future, I plan to add methods for calculating Fisher information and plotting diagnostics. My long range plans include: supplementing and speeding-up the EM algorithm (c.f., Meng and Rubin, 1991, Jamshidian and Jennrich, 1990), implementing a Fisher scoring or Newton-Raphson algorithm, adding new diagnostics including ones derived from Cook’s idea of local influence (c.f., Beckman, Nachtsheim, and Cook, 1987), and adding jackknife and cross-validation methods.
Figures 2, 3, and 4 all display results which come from the MLR estimates. Corresponding results from MLF estimates are nearly identical up to a change in scale. Hence, they add nothing to the qualitative discussion. This is, however, not the case when we consider leverage or influence of a case on the estimate of some $u_j$. Figures 5 and 6 show the generalized leverage for each case on the uniqueness of Filter 1 based on MLR and MLF estimation, respectively.

Since the design is balanced, leverage on the fixed coefficients (not shown) is flat—as one would expect. The balanced design also explains why observations from the same filter have the same leverage on a given filter. The pattern in Figure 6 is intuitive. The first six observations come from Filter 1, and have the greatest leverage. The next twelve observations (7–18) are on the two other filters which come from the same manufacturer as the first, and they have less leverage than the first six observations. Finally, the last 18 observations exert no leverage on the first filter. Note that influence is a product of leverage and square of residual. So we have that observations which come from a given unit will have greater potential influence on the uniqueness of units which are more similar, which is agreeable from the empirical Bayes point of view.

On the other hand, when we consider influence and leverage (Fig. 5) using MLR estimation, we have that it is likely that an observation from one unit could have greater influence or leverage on the uniqueness of another unit. This situation seems rather counterintuitive to me, and I don’t have any good way to explain it. At present, I can’t rule out a coding mistake, but as terrace-proto and terrace-f-proto use the same code for case-deletion methods this seems an unlikely problem. Unfortunately, these diagnostics are original, and no other programs are known to contain them; otherwise, I validate this phenomenon. Nevertheless, if this difference between full and restricted ML estimation is real, it deserves some serious consideration, especially when one intends to use empirical Bayes estimates. (More research on this later.)

unit. More emphatically, if it is the unit itself which is an outlier, what business do we have selecting from it observations which we believe are “reasonable”. On the other hand, when we delete points we can greatly diminish our estimates of covariance components. Thus, if the estimation of variance components is our primary concern, we must be wary of downward biasing them by deletion of influential points. In this situation, I would recommend setting $A_j = 0$, before deleting observations.
Figure 5: Leverage on the uniqueness of Filter 1 via MLR estimation.

Figure 6: Leverage on the uniqueness of Filter 1 via MLF estimation.
Figure 4: $\hat{A}_2 - \hat{A}_{2(i)}$ for Aerosol data.

Figure 2 presents Cook’s distance for the fixed parameters $C(\gamma, i)$ vs. observation index. Lines connecting points indicate group memberships, identifying observations on the same filter. Observations 25 and 26 stand out as influential observations, which previous analysts have noted. But filter 5 (obs. 25–30) looks suspicious as a whole. The Studentized residuals plotted in Figure 3 add to this suspicion. In this plot, we see that five of the most extreme observations are on filter 5. Furthermore, there is an uncanny pattern in the residuals of the first four filters. In each group of six observations the first three are with aerosol 1 and the last three are with aerosol 2. If the model fitted well, we would not see such a pattern. Filter 5 strongly reverses the pattern. This suggests that filter 5 is influential on the estimates of $A_j$.

The plot of $\hat{A}_2 - \hat{A}_{2(i)}$ in Figure 4 shows that if any of observations 25, 26, 28, 29, or 30 are deleted then the resulting estimate of $A_2$ would greater. In fact, Christensen, Pearson, and Johnson (1992) delete 25 and 26, and the resulting estimate of $A_2$ quite near 0, but I recommend other remedies.\footnote{A more radical move would be to remove all of Filter 5 from the model. One could reason that it is not coincidental that some of the most influential observations are in the same unit, but that it is the unit itself which is atypical; therefore, delete the entire}
Figure 2: Cook’s distance for fixed parameters of Aerosol data.

Figure 3: Studentized residuals from Aerosol data.
:h-cdbb Returns the case-deletion building block $h_i$.
:k-cdbb Returns $k_{ij}$.

6 Example: Aerosol Data

Beckman, Nahtaehm, and Cook (1987) present analyze via Cook’s approach of local influence a subset of data originating from Kerschner, Ettinger, De-Field, and Beckman (1984). Subsequently, Christensen, Pearson, and John-
son (1992) diagnose the same data set via case-deletion. The data comes
from an experiment measuring the variability of aerosols penetrating high-
efficiency (HEPA) filters. The design is balanced with three replicates at
each point. There are three filters from each of two manufacturers, and each
filter in turn is tested with two kinds of aerosols. Thus, we have a total 36 ob-
servations, six observations nested within each of six filters. An appropriate
model is

$$y_{ijkl} = \nu_{kl} + A_j + \epsilon_{ijkl},$$

$$\nu_{kl} = \mu + M_l + F_{kl},$$

where $i = 1, 2, 3$, $A_j$ aerosols $j = 1, 2$, manufacturer $M_l$, $l = 1, 2$, and filters
$F_{kl}$, $k = 1, 2, 3$, together with $\sum A_j = \sum M_l = 0$, $\epsilon_{ijkl} \overset{iid}{\sim} N(0, \sigma^2)$, and
$F_{kl} \overset{iid}{\sim} N(0, \tau)$. This model is identical to the model that Beckman, et al, 
(1987) and Christensen, et al, (1992) consider, apparent in the combined equation

$$y_{ijkl} = \mu + A_j + M_l + F_{kl} + \epsilon_{ijkl}.$$

TERRACE produces MLF (MLR, respectively) estimates $\hat{\mu} = 0.9924$,
$A_2 = -0.1969$, $M_2 = 0.5970$, $\hat{\sigma}^2 = 0.6328(0.6547)$, and $\hat{\tau} = 0.1364(0.2532)$,
which agree with the estimates that the previous analysts have produced.\footnote{Actually, Christensen, et al, provide MLR estimates for $\hat{\mu}$, $A_2$, and $M_2$ which are in error, which can be easily verified by computing directly the BLUES given the given estimates $\hat{\sigma}^2 = 0.6546$ and $\hat{\tau} = 0.2537$.}

Also, both groups of analysts identify two influential points, which shall be
obvious in the ensuing graphical analysis. Our analysis shall go one step fur-
ther, by noting the nested structure of observations, to suggest that perhaps an entire unit (filter) be deleted from the model.
Johnson (1992) propose $C(\gamma, i)$ and $\hat{h}_i$, while Hilden-Minton (1993) discusses these and proposes extensions including that $r_i = d_i \sqrt{s_i - \hat{h}_i}$ which can be seen as a Studentized residual.

5.2 Methods in TERRACE

At present, TERRACE has five slots dedicated to case-deletion analysis. They are $\textbf{gamma-cdd}$ and $\textbf{u-cdd}$, which are to house case-deleted differences $\hat{\gamma} - \hat{\gamma}_{(i)}$ and $\hat{u}_j - \hat{u}_{j(i)}$ and slots $\textbf{d-cdab}$, $\textbf{s-cdab}$ and $\textbf{h-cdab}$ which house the case-deletion building blocks $d_i$, $s_i$ and $h_i$. The method $\textbf{update-cdab}$ fills the five slots using current estimates. Once $\textbf{update-cdab}$ is called a case deletion analysis may commence. If one re-estimates the parameters or somehow alters the model, one will need to update the case-deletion building blocks again.

After calling $\textbf{update-cdab}$, the user may call any of the following methods:

$\textbf{deleted-gamma}$ Returns one-step approximations of $\hat{\gamma}_{(i)}$.

$\textbf{deleted-u-list}$ Returns one-step approximations of $\hat{u}_{(i)}$.

$\textbf{cook-gamma}$ Returns lists of $C(\gamma, i)$ grouped by units.

$\textbf{cook-u-list}$ Returns a list of lists. Each sublist contains $C(u_j, i)$ for $j = 0, 1, \ldots, J - 1$.

$\textbf{leverage-gamma}$ Returns generalized leverage $\hat{h}_i$.

$\textbf{leverage-u-list}$ Returns generalized leverage $\hat{k}_{ij}$.

$\textbf{stud-resid}$ Returns Studentized residual $r_i$.

$\textbf{gamma-cdd}$ Returns case-deleted difference $\hat{\gamma} - \hat{\gamma}_{(i)}$ as rows of matrices.

$\textbf{u-cdd}$ Returns case-deleted difference $\hat{u} - \hat{u}_{(i)}$ as rows of a matrix.

$\textbf{d-cdab}$ Returns the case-deletion building block $d_i$.

$\textbf{s-cdab}$ Returns the case-deletion building block $s_i$. 

21
these ideas are experimental and as such their usefulness and reliability are largely unknown.

5.1 Theory

To begin, we want one-step approximation of the case-deleted estimators of $\gamma$ and $u_j$. This is simply done using:

\[ \hat{\gamma} - \hat{\gamma}_C = d_i V_{11} \hat{a}_{1i}, \]
\[ \hat{u}_j - \hat{u}_j_C = d_i \tau \hat{a}_{2ij}, \]

where (letting $j_i$ denote the unit of which the $i$th case is a member)

\[ \hat{a}_{1i} = \frac{(a_{1i} - S_{12i} C^{-1} a_{2i})/(1 - d_{21i} C^{-1} a_{2i})}{1/(1 - d_{21i} C^{-1} a_{2i})}, \]
\[ \hat{y}_i = \frac{(y_i - S_{y2i} C^{-1} a_{2i})/(1 - d_{21i} C^{-1} a_{2i})}{1/(1 - d_{21i} C^{-1} a_{2i})}, \]
\[ \hat{a}_{2i} = \frac{(a_{2i} - S_{22i} C^{-1} a_{2i})/(1 - d_{21i} C^{-1} a_{2i})}{1/(1 - d_{21i} C^{-1} a_{2i})}, \]
\[ s_i = \frac{1}{1 - d_{21i} C^{-1} a_{2i}}, \]
\[ h_i = d_i V_{11} a_{1i}, \]

with

\[ d_i = \frac{\hat{y}_i - \hat{a}_{1i} \hat{\gamma}}{s_i - h_i} \]

and

\[ \hat{a}_{2ij} = \left\{ \begin{array}{ll} \hat{a}_{2i} - S_{21j} V_{11} \hat{a}_{1i} & \text{if } j = j_i; \\ -S_{21j} V_{11} \hat{a}_{1i} & \text{if } j \neq j_i. \end{array} \right. \]

Using the above formulas, one may go on to define measures of influence for $\gamma$ and $u_j$ like that of Cook's distance to be

\[ C(\gamma, i) = (\hat{\gamma} - \hat{\gamma}_C) V_{11}^{-1} (\hat{\gamma} - \hat{\gamma}_C) = d_i^2 h_i \]

and

\[ C(u_j, i) = (\hat{u}_j - \hat{u}_j_C) V_{22}^{-1} (\hat{u}_j - \hat{u}_j_C) = d_i^2 k_{ij}. \]

(Let $k_{ij}$ be defined implicitly in the above equation.)

Also a notion of generalized leverage may be expressed by $\hat{h}_i = h_i / s_i$ for $\gamma$ and $\hat{k}_{ij} = k_{ij} / (s_i - h_i + k_{ij})$ for the $i$th case on $u_j$. Christensen, Pearson, and
initial estimates. One feasible way to begin—the way \texttt{terrace- proto} by
default begins—is to set $\sigma^2 = 0$ and $\tau^{-1} = 0$ and make one initial M step
to compute estimates for the other parameters. At this point all vital slots
will have feasible content, and the object may be allowed to iterate. If one
desires to start from an arbitrary point, one may use accessor methods to fill
the slots appropriately, but this can be difficult.

The method \texttt{m-step} updates \texttt{c-inv-list}, \texttt{v11}, \texttt{v22-list}, \texttt{gamma}, and
\texttt{u-list}. The method \texttt{em-step} gives us one complete iteration beginning
with \texttt{update-tau}, \texttt{update-sigma}, \texttt{m-step}, and ends with \texttt{update-llik-
history}.

The method \texttt{llik} computes the restricted log likelihood function

$$
\log[f(Y | \sigma^2, \tau)] = \frac{1}{2} \left\{ -(N - J) \log(\sigma^2) - J \log|\tau| + \log|V_{11}| \right.
\left. + \sum \log |C_j^{-1} - \sum Y_j e_j| \right\}.
$$

\texttt{update-llik-history} calls this message and includes its value to the list
in the slot \texttt{llik-hist}.

Finally, the method \texttt{ml} maximizes likelihood by calling \texttt{em-step} re-
peatedly until the relative change in log-likelihood is less than the key-word
argument precision, which is 0.00001 by default. If one wishes to use some
other convergence criteria, one may over-ride \texttt{ml} or use \texttt{em-step} directly.

4.4 Other methods

Other methods are described in sections 3.2 and 5.2.

5 Case-deletion diagnostics

My principal motivation to write TERRACES was to create an environment
were I could develop some experimental diagnostics for multi-level models.
In “Mixed model diagnostics via case-deletion” I develop several one-step
estimators of various parameters in the mixed model and propose several
diagnostic measures. In this section, I shall briefly describe these formula
and how they are implemented in TERRACE. It should be emphasized that

19
and stores them in the their designated slots.

Now, our current estimate of $\gamma$ is updated by :update-gamma which computes

$$\hat{\gamma} = V_{11}[\sum(S_{1yj} - S_{12j}C_{\hat{x}}^{-1}S_{2yj})].$$

Note that if you simply want the current estimate of $\gamma$ it is best to use :gamma which does not alter the contents of the slot.

Finally, :update-u-list computes and stores

$$\hat{u}_j = C_j^{-1}(S_{2yj} - S_{21j}\hat{\gamma}).$$

The method :m-step calls these methods in succession.

### 4.3.2 E step

In the E step, we update estimates of $\tau$ and $\sigma^2$. The method :update-tau computes and stores

$$\hat{\tau} = J^{-1}(\sum \hat{u}_j\hat{u}_j' + \sigma^2 \sum V_{22j}).$$

This formula uses the old estimate of $\sigma^2$. If one wanted to impose some structure on $\tau$, one may over-ride :update-tau with an appropriate method.

Next, :update-sigma computes and stores

$$\hat{\sigma}^2 = N^{-1}\{\sum \hat{e}_j\hat{e}_j + \sigma^2[F - \text{tr}(V_{11}\sum S_{12j}C_j^{-1}S_{21j}) + \text{tr}(\sum V_{22j}S_{22j})]\},$$

where

$$\hat{e}_j = Y_j - A_{1j}\hat{\gamma} - A_{2j}\hat{u}_j,$$

which are computed but not stored by :resid-list.

### 4.3.3 Iteration methods

The above formulas for updating key statistics assume that the model already has estimates for certain parameters. This means that it is necessary to have

---

2Technically, this is part of the true EM step, not just the estimation part. This is just convenient labeling.
4.2.3 :init-sufficient-stats

Once :remodel secures our level-two modeling choices, it calls :init-sufficient-stats. First, :init-sufficient-stats computes the design matrices $A_{1j}$ and $A_{2j}$ as specified in equation (5). These are stored in slots a1-list and a2-list. Next the following “sufficient” statistics are computed and stored in the appropriate slots:

$$
\begin{align*}
S_{11j} &= A'_{1j} A_{1j} \\
S_{12j} &= A'_{1j} A_{2j} \\
S_{22j} &= A'_{2j} A_{2j} \\
S_{1bj} &= A'_{1j} y_j \\
S_{2bj} &= A'_{2j} y_j
\end{align*}
$$

These statistics will be shown to be useful in the next section.

4.3 Estimation methods

The methods of this section follow the implementation of the EM algorithm as outlined in the technical appendix of Bryk & Raudenbush (1992). Much of the notation is the same.

4.3.1 M step

The maximization step amounts to computing the following using the current estimates for $\sigma^2$ and $\tau$.

$$
C_j = S_{22j} + \sigma^2 \tau^{-1},
$$

the inverse of which is computed and stored in the slot c-inv-list by the method :update-c-inv-list.

Next, the methods :update-v1 and :update-v22-list compute

$$
V_{11} = [\sum (S_{11j} - S_{12j} C_j^{-1} S_{21j})]^{-1}
$$

and

$$
V_{22j} = C_j^{-1} + C_j^{-1} S_{22j} V_{11} S_{12j} C_j^{-1},
$$
4.2 Initialization methods

Now that we have the slot structure, we need methods to fill these slots with necessary information. In this section I discuss the three main initialization methods: :isnew, :remodel, and :init-sufficient-stats.

4.2.1 :isnew

This method takes the parameters: y-list x-list z &key var-labels. These parameters must be in the form that their slots expect. See section 4.1. The method simply passes these values on to the appropriate slots, which indirectly causes the slots p-reg and n-list to be correctly filled.

The method :isnew is usually only called indirectly by the method :new which is sent to a prototype to create an object after its pattern. For instance, one may form a model by

(send terrace-prototype :new y-list x-list z :var-labels labels)

The method :new passes whatever parameters you give it onto :isnew, thus we use :isnew’s parameter list. The difficulty with using :new to construct a new TERRACE model is that one must carefully code the parameters. terrace-mlr-model does this initial coding for us.

4.2.2 :remodel

Thus far, the information that we have given our TERRACE object is independent of second level modeling choices. These choices are specified by z-ind and rand-ind. :remodel takes these arguments to their respective slot and calls the methods :init-sufficient-stats, :m-step, :ml, and :display, which are all dependent on modeling choices.

When :remodel is used to alter an analysis, it will empty certain slots as necessary.
z-ind A list of indices, where each index is a list of integers indicating which columns of \( z \) to regress a \( \beta_p \) on. There must be \( P \) such indices.

rand-ind An index, a list of integers, indicating which \( \beta_p \) to treat as random. Note that \( p = 0, 1, \ldots, P - 1 \).

p-reg P, number of level-one regressors.

n-list List containing \( n_j \) for \( j = 0, 1, \ldots, J - 1 \), where the \( j \)th unit contains \( n_j \) cases.

gamma A list representing the current estimate of \( \gamma \).

u-list A list of lists representing current estimates of each \( u_j \).

sigma Current estimate of \( \sigma^2 \).

tau Matrix representing current estimate of \( \tau \).

llik-history List containing history of the log-likelihood function in reverse-chronological order.

a1-list, a2-list Lists of submatrices of \( A_1 \) and \( A_2 \).

s11, s12, s22, s1y, s2y Lists of sufficient statistics used in computation.

c-inv-list, v11, v22-list Contain helpful subcomputations.

gamma-cdd List of matrices containing case-deleted differences in estimates of \( \gamma \).

u-cdd Matrix containing case-deleted differences in estimates of \( u \).

d-cddb, h-cddb, s-cddb Case-deletion building blocks.

Each slot has its own accessor method, e.g., \texttt{:var-labels, :y-list}. Many slots are not of direct interest to the user, but they play some internal role. Nevertheless, they all may be read by their accessor methods. Some accessor methods take optional values to replace old contents. But one must be careful when replacing information in slots. Accessor methods have few guards against nonsense. Some sensitive slots may not be written over, and it may be dangerous to the integrity of the model to over- ride this limitation.
To save effort and memory, it is best to alter existing TERRACE models rather than creating new one. So far, however, there is no method for altering the level-one modeling assumptions. In this case one must construct new objects.

4 The Object Structure of TERRACE

TERRACES is a package of XLISP-STAT objects. Presently, TERRACES has two prototypes: `terrace-proto`, which is constructed to carry out restricted maximum likelihood estimation via the EM algorithm, and `terraceef-proto`, which inherits from `terrace-proto` with a few modifications so that full information maximum likelihood estimation via the EM algorithm is obtained. Other prototypes which extend the model or apply a different computational approach are feasible descendants of `terrace-proto`. In this section I will focus on how `terrace-proto` works.

Objects in XLISP-STAT have slots and methods. Slots are memory locations in an object where information is stored, while methods specify how the object uses information. An object is in a sense the union of its form and function. First, the slots will be described. Then the host of methods will be introduced.

4.1 Slots

Here is the basic anatomy of TERRACE. The main prototype `terrace-proto` has the following slots:

`var-labels` List of labels for level-one regressors.

`y-list` A list of lists representing the response variable $Y_j$ for each of the $J$ units.

`x-list` A list of matrices representing the level-one background variables $X_j$ for each of the $J$ units. Each $X_j$ must have $P$ columns.

`z` Matrix of level-two background variables as columns. Must have $J$ rows, one for each unit.
randomly. Instead of constructing a new model, we may alter the old one by 
(send rat :remodel '((0 1) (1)) '(0)). This produces the following 
results.

Maximizing likelihood...

Iteration 1: -144.1985
Iteration 2: -143.2555
Iteration 3: -143.0424
Iteration 4: -142.9597
Iteration 5: -142.9223
Iteration 6: -142.9036
Iteration 7: -142.8936
Iteration 8: -142.8879
Iteration 9: -142.8846
Iteration 10: -142.8826
Final Iteration 11: -142.8814

TERRACE: Full Information Maximum Likelihood

<table>
<thead>
<tr>
<th>Parameter Estimates</th>
<th>(S.E.)</th>
<th>T:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept-0</td>
<td>18.8737</td>
<td>(13.6824)</td>
</tr>
<tr>
<td>Intercept-1</td>
<td>0.2151</td>
<td>(0.0849)</td>
</tr>
<tr>
<td>Week-1</td>
<td>0.1650</td>
<td>(0.0061)</td>
</tr>
</tbody>
</table>

Sigma^2: 97.9748

Tau (covariance)

Intercept 18.3594

Tau (correlation)

Intercept 1.0000
NIL
Iteration 22: -142.2263
Iteration 23: -142.2247
Iteration 24: -142.2233
Final Iteration 25: -142.2219

**TERRACE: Full Information Maximum Likelihood**

<table>
<thead>
<tr>
<th>Parameter Estimates</th>
<th>(S.E.)</th>
<th>T:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept-0</td>
<td>12.9382</td>
<td>(22.5847)</td>
</tr>
<tr>
<td>Intercept-1</td>
<td>0.2514</td>
<td>(0.1387)</td>
</tr>
<tr>
<td>Week-0</td>
<td>2.9677</td>
<td>(9.2202)</td>
</tr>
<tr>
<td>Week-1</td>
<td>0.1469</td>
<td>(0.0566)</td>
</tr>
</tbody>
</table>

Sigma^2: 88.9808

**Tau (covariance)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>58.5042</th>
<th>-13.4626</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week</td>
<td>-13.4626</td>
<td>3.7463</td>
</tr>
</tbody>
</table>

**Tau (correlation)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Correlation</th>
<th>T:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>1.0000</td>
<td>-0.9094</td>
</tr>
<tr>
<td>Week</td>
<td>-0.9094</td>
<td>1.0000</td>
</tr>
<tr>
<td>NIL</td>
<td>NIL</td>
<td>NIL</td>
</tr>
</tbody>
</table>

> rat

#<Object: 1527360, prototype = TERRACE-F-PROTO>

Foreshadowing our discussion in the next section, note that **rat** is reparented so that it now inherits from **terrace-f-proto** which is a prototype that carries out MLF estimation. The analogous method `change-to-mlr` will reparent a TERRACE model to inherit from **terrace-proto** which performs mlr estimation.

In **rat** the parameter Week-0, a.k.a. \( \gamma_{10} \), is a likely candidate removing from the model. Also we may want to make Week, a.k.a. \( \beta_{1j} \), vary non-
:ml Accepts key word argument :precision. Iterates EM algorithm until relative change in log-likelihood becomes less than precision.

3.3 Altering a TERRACE model

In our ongoing example, TERRACE has estimated the parameters of rat by restricted maximum likelihood. Suppose we are also interested in the mlf estimates. To compute them, we may construct an entirely new object using terrace-mlf-model or we may simply alter our existing object using the message :change-to-mlf.

> (send rat :change-to-mlf)
; loading "terrace-f.lsp"
Maximizing likelihood...

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-142.3632</td>
</tr>
<tr>
<td>2</td>
<td>-142.3303</td>
</tr>
<tr>
<td>3</td>
<td>-142.3124</td>
</tr>
<tr>
<td>4</td>
<td>-142.2998</td>
</tr>
<tr>
<td>5</td>
<td>-142.2898</td>
</tr>
<tr>
<td>6</td>
<td>-142.2816</td>
</tr>
<tr>
<td>7</td>
<td>-142.2746</td>
</tr>
<tr>
<td>8</td>
<td>-142.2685</td>
</tr>
<tr>
<td>9</td>
<td>-142.2631</td>
</tr>
<tr>
<td>10</td>
<td>-142.2584</td>
</tr>
<tr>
<td>11</td>
<td>-142.2541</td>
</tr>
<tr>
<td>12</td>
<td>-142.2503</td>
</tr>
<tr>
<td>13</td>
<td>-142.2468</td>
</tr>
<tr>
<td>14</td>
<td>-142.2437</td>
</tr>
<tr>
<td>15</td>
<td>-142.2408</td>
</tr>
<tr>
<td>16</td>
<td>-142.2382</td>
</tr>
<tr>
<td>17</td>
<td>-142.2358</td>
</tr>
<tr>
<td>18</td>
<td>-142.2336</td>
</tr>
<tr>
<td>19</td>
<td>-142.2316</td>
</tr>
<tr>
<td>20</td>
<td>-142.2297</td>
</tr>
<tr>
<td>21</td>
<td>-142.2279</td>
</tr>
</tbody>
</table>
:dispersion-11 Computes estimate of Cov(\(\hat{\gamma}\)).

:u-list Returns list of current estimates of \(u_j\).

:dispersion-22-list Computes list of estimates of Cov(\(\hat{u}_j\)).

:beta-list Computes list of empirical Bayes estimates of \(\beta_j\).

:beta-cov-list Computes list of estimates of the dispersion matrices for the empirical Bayes estimates of \(\beta_j\).

:y-hat-list Computes list of predicted \(Y_j\).

:resid-list Computes list of residuals \(Y_j - A_{1j}\hat{\gamma} - A_{2j}\hat{u}_j\).

:x-list Computes list of residuals \(Y_j - A_{1j}\hat{\gamma}\).

:sigma Returns current estimate of \(\sigma^2\).

:tau Returns current estimate of \(\tau\) as a covariance matrix.

:tau-corr Returns current estimate of \(\tau\) as a correlation matrix.

:p-reg Returns the number of level-one regressors.

:m-list Returns list of number of cases per unit.

:n-cases Returns total number of cases.

:j-units Returns number of units.

:f-list Returns list of number of fixed effects per level-one regressor.

:f-effects Returns total number of fixed effects.

:r-effects Returns number of random effects per unit.

:llik Computes current log-likelihood.

:llik-history Returns list of log-likelihood values in reverse chronological order. Accepts key-word argument :clear; with t the history is erased.

:em-step Computes one iteration of EM algorithm.
Figure 1: Plot of residual fit on rat.

Some methods require or optionally accept additional arguments, but most methods in TERRACE do not. This section closes with a list of the most useful methods for interacting with a TERRACE model, except case-deletion methods which will be introduced in section 5.2. The methods introduced here do not take arguments except where indicated, and details of these and other methods are discussed in the following sections.

:display Presents essential statistics in a readable format.

:remodel (Args: z-ind rand-ind). Reconstructs model according to z-ind and rand-ind, maximizes likelihood, and displays results.

:change-to-mlf Changes object to carry out full information maximum likelihood estimation, maximizes likelihood, and displays results.

:change-to-mlr Changes object to carry out restricted maximum likelihood estimation, maximizes likelihood, and displays results.

:gamma Returns current estimate of $\gamma$.

:gamma-stderr Computes standard errors for $\hat{\gamma}$. 
Note that this model required 21 iterations to “converge”.\(^1\) This rate of convergence is slow, albeit linear with about 10 percent loss of error with each iteration. Looking at the T-statistics on the estimates of the \(\gamma\)s, we see further evidence that this model may be over-determined, that we may want to set \(\gamma_{10}\) to 0. The estimate of \(\tau\) is given in two forms. The last line is essentially the the address of the object. This is what \texttt{rat} labels.

### 3.2 Interacting with a TERRACE model

In \texttt{XLISP-STAT}, one interacts with an object by sending it messages. The object then will interpret the message, selecting the appropriate method and returning the result of that method. For instance, suppose we want the residuals from \texttt{rat}. This is obtained by sending the message \texttt{:resid-list} which returns list of lists for each unit, in this case, for each rat. Thus,

\[
\begin{align*}
 &> \text{(send rat :resid-list)} \\
 &\quad \left(\begin{array}{cccccccc}
 -5.73852 & 0.855727 & (-4.40952 & 8.33141 & 1.07234 & -10.1867 \\
 -4.44581 & (-2.3497 & -14.0611 & 10.2274 & 7.516 & -0.195437 \\
 -7.7814 & -0.224566 & -7.66774 & (6.55588 & 6.92854 & -2.69879 \\
 1.67388 & 9.04655 & & & & \\
\end{array}\right)
\end{align*}
\]

Statements sending messages to an object may be part of more complex statements. For example suppose we want to plot residuals on rats. This is accomplished in the following:

\[
> \text{(plot-points (repeat (iseq 1 10) (send rat :n-list))} \\
\quad \text{(combine (send rat :resid-list)))}
\]

\#<Object: 1092520, prototype = SCATTERPLOT-PROTO>

The result is the plot in figure 1.

\(^1\)In TERRACE “convergence” is assumed when the relative change in log-likelihood is less than 0.00001.
Iteration 7: -146.3142  
Iteration 8: -146.2999  
Iteration 9: -146.2888  
Iteration 10: -146.2801  
Iteration 11: -146.2732  
Iteration 12: -146.2676  
Iteration 13: -146.2629  
Iteration 14: -146.2591  
Iteration 15: -146.2558  
Iteration 16: -146.2531  
Iteration 17: -146.2508  
Iteration 18: -146.2488  
Iteration 19: -146.2470  
Iteration 20: -146.2455  
Final Iteration 21: -146.2442

TERRACE: Restricted Maximum Likelihood

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimates</th>
<th>(S.E.)</th>
<th>T:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept-0</td>
<td>12.9382</td>
<td>( 36.4615)</td>
<td>0.3548</td>
</tr>
<tr>
<td>Intercept-1</td>
<td>0.2514</td>
<td>( 0.2239)</td>
<td>1.1227</td>
</tr>
<tr>
<td>Week-0</td>
<td>2.9677</td>
<td>( 12.1199)</td>
<td>0.2449</td>
</tr>
<tr>
<td>Week-1</td>
<td>0.1469</td>
<td>( 0.0744)</td>
<td>1.9734</td>
</tr>
</tbody>
</table>

Sigma^2: 90.4519

Tau (covariance)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimates</th>
<th>T:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>84.8807</td>
<td>-20.4320</td>
</tr>
<tr>
<td>Week</td>
<td>-20.4320</td>
<td>6.3300</td>
</tr>
</tbody>
</table>

Tau (correlation)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimates</th>
<th>T:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>1.0000</td>
<td>-0.8815</td>
</tr>
<tr>
<td>Week</td>
<td>-0.8815</td>
<td>1.0000</td>
</tr>
</tbody>
</table>
2. Full random effects.

\[ \beta_{0j} = \gamma_0 + \gamma_{01} z_j + u_{0j}, \]
\[ \beta_{1j} = \gamma_{10} + \gamma_{11} z_j + u_{1j}, \]

which is obtained with

(terrace-mlr-model x y rat z '((0 1) (0 1)) '(0 1)
 :var-labels '("Intercept" "Week").

And

3. Random intercept, common fixed slope.

\[ \beta_{0j} = \gamma_0 + \gamma_{01} z_j + u_{0j}, \]
\[ \beta_{1j} = \gamma_{10}, \]

which is obtained with

(terrace-mlr-model x y rat z '((0 1) (0)) '(0)
 :var-labels '("Intercept" "Week").

Let's load "terrace.lsp" and give the full model to the XLISP-STAT interpreter.

> (load "terrace")
; loading "terrace.lsp"
T
> (setf rat (terrace-mlr-model x y rat z '((0 1) (0 1))
 '((0 1) :var-labels '("Intercept" "Week")))

Maximizing likelihood...

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-148.6767</td>
</tr>
<tr>
<td>2</td>
<td>-146.6921</td>
</tr>
<tr>
<td>3</td>
<td>-146.4714</td>
</tr>
<tr>
<td>4</td>
<td>-146.4006</td>
</tr>
<tr>
<td>5</td>
<td>-146.3605</td>
</tr>
<tr>
<td>6</td>
<td>-146.3335</td>
</tr>
</tbody>
</table>
Before formulating a model, let’s first do a little data handling.

> (def rat-data (read-data-columns "rat.data" 7))
RAT-DATA
> (def y (combine (mapcar #'rest
 (select rat-data (iseq 1 5)))))
Y
> (def rat (repeat (iseq 10) 5))
RAT
> (def intercept (repeat 1 50))
INTERCEPT
> (def week (repeat (iseq 5) (repeat 10 5)))
WEEK
> (def x (list intercept week))
X
> (def z (rest (select rat-data 6)))
Z

Now we model the growth of the \(j\)th rat by

\[ y_j = \beta_{0j} + \beta_{1j} \text{week}_j + \epsilon_j. \]

Next we want to model how the growth curves of the individual rats are related. Here are some possible models:

1. Simple random effects,

\[ \beta_{0j} = \gamma_0 + u_{0j}, \]

with

\[ \beta_{1j} = \gamma_1 + u_{0j}. \]

This model may be obtained with

\[
\text{(terrace-mlr-model x y rat nil '((0) (0)) '(0 1)) :var-labels '("Intercept" "Week")).}
\]
(terrace-mlr-model x y group z ind-z rand-ind
  :var-labels labels).

Here x is a list of lists representing the level-one background variables, y is a list representing the outcome variable, while group is a list of group indicators for each of the N cases. These indicators should be integers from 1 to J. The lengths of y, group, and the sublists of x must all be equal. Next, z is a list of lists representing the background variables on the J units. The sublists of z must be of length J. To specify which columns of Z regress which β_p as in equation (2), we encode ind-z so that the pth element of ind-z (p = 0, 1, ..., P - 1), is a list of indicators of the columns of Z to be used. Here 0 indicates the unit (or uncrossed) carrier and the sublists of z are indicated with 1, 2, 3, ... Next, rand-ind is a list of indicators which β_p to treat as random. Finally, as an optional key-word parameter, the user may pass a list of strings in labels to act as the labels for the level-one regressors, otherwise TERRACE constructs default labels.

Given the above parameters, terrace-mlr-model constructs the model as an object inherited from terrace,proto, maximizes the restricted likelihood function, displays estimates of basic parameters and relevant statistics, and returns the object itself. The object is returned so that the user may label it using def or setf. TERRACE also includes the function terrace-mlf-model which will perform full information maximum likelihood estimation.

Another method for constructing a model in TERRACE will be presented in section 4.2.

For an example, we construct a model for the Rat Data from the HLM manual by Bryk, Raudenbush, Seltzer & Congdon (1989). These data consist of five weekly weight measurements taken on five rats with mother’s weight as a background variable. Issued with TERRACE, the file “rat.data” contains the following:

rat t1 t2 t3 t4 t5 z
1 61 72 118 130 176 170
2 65 85 129 148 174 194
3 57 68 130 143 201 187
4 46 74 116 124 157 156
5 47 85 103 117 148 155
6 43 58 109 133 152 150
As stated the above model treats each parameter $\beta_{jp}$ in (1) as randomly varying in (3) with variance $\tau_{pp}$, but sometimes we do not want to treat $\beta_{jp}$ as random. In this case, we want to impose $\tau_{pp} = 0$, whence the $p$th row and column of $\tau$ will be zero. In practice, we want to delete rows and columns of zeros from $\tau$. This is accomplished if we treat $u_{jp} = 0$ for all $j = 1, 2, \ldots, J$. Thus, modify (4) to be

$$Y_j = A_{1j} \gamma + A_{2j} u_j + r_j,$$

where $A_{1j} = X_j W_j$ and $A_{2j}$ a submatrix of $X_j$ containing the columns for which $\tau_{pp}$ are not constrained to be zero.

Given the data $X_j$, $Y_j$ for each of the $J$ units, data/design matrices $Z_p$ for each of the $P$ level-one regressors, and a subset of the $P$ regressors to be treated as having random effects, TERRACE can compute estimates of the parameters of the model (5), including $\sigma^2$ and $\hat{\tau}$ (with rows and columns of zeros deleted). After that is accomplished, TERRACE gives the user many options. The next section explains how to construct a model and estimate its parameters in TERRACE.

3 Getting Started with TERRACE

It is assumed that the user has a working knowledge of XLISP-STAT. The first chapter of Tiemey (1989) should suffice.

3.1 Constructing a model in TERRACE

To perform regression analysis in XLISP-STAT, one begins by creating a model using (regression-model x y), where x is a list of predictor variables, i.e., x is a list of lists, and y is a list representing the response variable. The result is an object inherited from, or patterned after, regression-proto. Usually one labels this object using (setf model (regression-model x y)) so that later one may send messages to the object such as (send model :residuals) which returns the list of residuals for the model model.

In a similar manner with TERRACE, we may construct our hierarchical model using
2 Description of a multi-level model

Suppose we have $N$ subjects naturally grouped into $J$ units, where there are $n_j$ subjects in the $j$th unit and $\sum_{j=1}^{J} n_j = N$. Further suppose that for the $J$ units we model

$$Y_j = X_j\beta_j + r_j,$$  \hspace{1cm} (1)

where each $X_j$ has dimensions $n_j \times P$, and

$$r_j \sim N(0, \sigma^2 I_{n_j}).$$

At the next level, we want to model each $\beta_{jp}$ ($j = 1, 2, \ldots, J$, and $p = 1, 2, \ldots, P$) with

$$\beta_{jp} = z_{jp}\gamma_p + u_{jp},$$  \hspace{1cm} (2)

where $z_{jp}$ is the $j$th row of an $(J \times f_p)$-matrix $Z_p$ and $u_{jp} \sim N(0, \tau_{pp})$. But the $u_{jp}$ are not independent; to get at their covariance structure, we “stack” the equations in (2) to obtain

$$\beta_j = W_j\gamma + u_j,$$  \hspace{1cm} (3)

where

$$u_j = (u_{j1}, u_{j2}, \ldots, u_{jP})',$$

$$\gamma = (\gamma_1', \gamma_2', \ldots, \gamma_P'),$$

and

$$W_j = \begin{pmatrix}
z_{j1} & 0 & \cdots & 0 \\
0 & z_{j2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & z_{jP}
\end{pmatrix}.$$

Note that $W_j$ is a $(P \times F)$-matrix, where $F = \sum_{p=1}^{P} f_p$. Now our second level model may be completely specified by (3) when we impose that each $u_{jp} \sim N(0, \tau)$. Also we impose that $\text{Cov}(u_j, r_{j'}) = 0$ for any $j$ and $j'$. Combining (1) and (3), one obtains

$$Y_j = X_jW_j\gamma + X_ju_j + r_j,$$  \hspace{1cm} (4)
TERRACES:
an XLISP-STAT package for multi-level modeling
with diagnostics

James Hilden-Minton

December 14, 1993

Work in Progress

1 Introduction

TERRACE is an XLISP-STAT package of objects with which one may interactively construct, estimate, and diagnose multi-level models. At present, the package accepts bi-level, or once clustered, data and estimates the parameters of the implied hierarchical linear model using restricted or full maximum likelihood estimation. The user can then interact with the model to extract estimates of the parameters, calculate diagnostic statistics, make graphs, and to perform whatever analyses the user may envision. The program (and notation) follows the implementation of the EM algorithm as outlined in the technical appendix of Bryk & Raudenbush (1992). The case-deletion diagnostics are developed in Hilden-Minton(1993) for a general mixed model and are here adapted to TERRACE.

The purpose of this paper is to show how to use TERRACE, explain briefly how it works, and to demonstrate its use. The basic bi-level model will be introduced.