RANDOM COEFFICIENT MODELS FOR TREND, SEASONALITY, LOCATION, ...

JAN DE LEEUW

ABSTRACT. Meet the abstract. This is the abstract.

1. INTRODUCTION

1.1. **Data.** Consider a univariate time series of, say, ozone level measurements, observed hourly over a number of years at one or more observation stations. Suppose n_T is the total numbers of observations, some observations may be missing.

2. MODEL

We suppose that ozone level is determined, at least partially, by hour-of-the-day, day-of-the-week, month-of-the-year, by year, and by location. In order to model this dependency we use a (linear) random coefficient model [1].

We create K indicators (dummies) G_k, coding hour, day, month, year, and location. Each G_k has n_T rows and m_k columns. An indicator for hours has 24 columns, the one for days 7 columns, the one for months 12 columns, and so on. Define corresponding vectors $\underline{\beta}_k$ of random regression coefficients, and assume

$$\underline{y} = \eta + \sum_{k=1}^{K} G_k \underline{\beta}_k + \underline{\varepsilon}. \tag{1}$$

The disturbances $\underline{\varepsilon}$ are assumed to have mean zero, dispersion matrix $\sigma^2 I$, and to be uncorrelated with the random regression coefficients. The vector η captures the fixed effects, and it can incorporate another linear or non-linear regression model.

Date: November 7, 2008 — 13h 52min — Typeset in TIMES ROMAN.

[1] We follow the Dutch Convention of underlining random variables introduced by Van Dantzig, discussed by Hemelrijk [1966], and used systematically in De Leeuw and Meijer [2008a].

2.1. **Matrix Expression.** To derive convenient matrix expressions we define the $n_T \times \sum_{k=1}^K m_k$ matrix

$$G = \begin{bmatrix} G_1 & \mid & G_2 & \mid & \cdots & \mid & G_K \end{bmatrix},$$

and the $\sum_{k=1}^K m_k$-element vector

$$\underline{\beta} = \begin{bmatrix} \underline{\beta}_1 \\ \underline{\beta}_2 \\ \vdots \\ \underline{\beta}_K \end{bmatrix}.$$

This allows us to write

$$\mathbf{E}(y) = \eta + Gb, \tag{2a}$$

where $b = \mathbf{E}(\underline{\beta})$. If we assume, in addition, that the vectors of random regression coeficients $\underline{\beta}_k$ are not correlated with each other, then

$$\mathbf{V}(\underline{y}) = \sum_{k=1}^K G_k V_k G_k' + \sigma^2 I,$$

where $V_k = \mathbf{V}(\underline{\beta}_k)$. This can also be written as

$$\mathbf{V}(\underline{y}) = GVG' + \sigma^2 I,$$

where

$$V = \bigoplus_{k=1}^K V_k.$$

2.2. **Regression Parametrization.** The regression coefficients b show the expected (fixed) effects of hour, day, month, year, and so on. These effects can be plotted in separate time-plots, or in a joint plot of Gb against time. It could be that we want to make sure these time plots are smooth, and consequently we may want to impose smoothness constraints on the regression coefficients.

Suppose Z_k is an $m_k \times p_k$ matrix, for example of polynomials or of sines/cosines. Now require

$$\underline{\beta}_k = Z_k \gamma_k + \underline{\delta}_k,$$

and we have a total of $\sum_{k=1}^K p_k$ regression parameters. The unrestricted model has all Z_k equal to the identity. Note that $\mathbf{V}(\underline{\delta}_k) = \mathbf{V}(\underline{\beta}_k) = V_k$.

For a matrix form we define

$$Z = \bigoplus_{k=1}^{K} Z_k.$$

$$\gamma = \begin{bmatrix}\gamma_1\\ \gamma_2\\ \vdots\\ \gamma_K\end{bmatrix}, \quad\text{and}\quad \underline{\delta} = \begin{bmatrix}\underline{\delta}_1\\ \underline{\delta}_2\\ \vdots\\ \underline{\delta}_K\end{bmatrix}.$$

Then

$$\mathbf{E}(y) = \eta + GZ\gamma, \tag{3a}$$

while we still have

$$\mathbf{V}(\underline{y}) = GVG' + \sigma^2 I. \tag{3b}$$

3. Estimation

3.1. **Initial Estimates of the Regression Coefficients.** Ordinary (unweighted) least squares estimates of γ can be computed by minimizing

$$\sigma_{OS}(\gamma) = (y - \eta - GZ\gamma)'(y - \eta - GZ\gamma) \tag{4a}$$

We can always write y in the form $y = \eta + Gb_\eta + e$, where b_η is any least squares estimate, i.e. any b_η such that $G'(y - \eta - Gb_\eta) = G'e = 0$. This means that

$$\sigma_{OS}(\gamma) = (b_\eta - Z\gamma)'G'G(b_\eta - Z\gamma) + e'e, \tag{4b}$$

and we can compute the estimate of γ, for given η, in two steps: first compute any least squares estimates b_η and then minimize (4b). This gives

$$\hat{\gamma}_{OS} = (Z'G'GZ)^{-1}Z'G'Gb_\eta.$$

The matrix $G'G$ is the *Burt matrix* of the design, i.e. the supermatrix with as elements the cross tables of the K factors. For balanced designs, for example with complete observations on 24 hours in a day and 7 days in a week, this implies various simplications. Because Gb_η is the same for all least squares estimates, it does not matter which one we use. The least squares estimate always exists and is an unbiased estimate of γ.

Note that if η contains unknowns, we may have to use *alternating least squares*. We iteratively alternate minimization of (4a) over η for given γ and minimization of (4b) over γ for given η.

3.2. **Loss Function.** The loss function we minimize to get final parameter estimates in the current version of the program is, except for some irrelevant constants, the negative multinormal log-likelihood (FIML). This is

$$\mathscr{L}(\gamma,W)=\log\mathbf{det}(W)+(y-\eta-GZ\gamma)'W^{-1}(y-\eta-GZ\gamma), \tag{5a}$$

where

$$W=GVG'+\sigma^2 I. \tag{5b}$$

This can be rewritten in the computationally more efficient form [De Leeuw and Meijer, 2008b, p. 21]

$$\mathscr{L}_j=(n_j-p)(\log\sigma^2-\frac{s_j^2}{\sigma^2})+\log\mathbf{det}(W_j)+(b_j-Z_j\gamma)'W_j^{-1}(b_j-Z_j\gamma), \tag{6a}$$

where

$$W_j=\Omega+\sigma^2(X_j'X_j)^{-1}, \tag{6b}$$

and where

$$b_j=(X_j'X_j)^{-1}X_j'y_j, \tag{6c}$$

$$s_j^2=\frac{1}{n_j-p}(y_j-X_jb_j)'(y_j-X_jb_j), \tag{6d}$$

are the ordinary least squares regression coefficients and mean squared error. Of course the FIML loss over all groups is simply

$$\mathscr{L}_F(\gamma,\sigma^2,\Omega)=\sum_{j=1}^m\mathscr{L}_j(\gamma,\sigma^2,\Omega).$$

Alternatively, we can use the restricted or residual (REML) loss function. This is defined as

$$\mathscr{L}_R(\sigma^2,\Omega)=\min_\gamma\mathscr{L}_F(\gamma,\sigma^2,\Omega)+\log\mathbf{det}(\sum_{j=1}^m Z_j'W_j^{-1}Z_j). \tag{7}$$

3.3. **Reparametrization.** If Ω is positive semidefinite and $\sigma^2>0$ then it follows that V_j is non-singular. But W_j can still be singular if $X_j'X_j$ is singular, and the null-space of matrices Ω and $X_j'X_j$ have a non-trivial intersection.

It is not uncommon that the matrices X_j are singular, because there are too few observations in the corresponding group. In the case the formulas Subsection 3.2 do not apply any more, and we must to be more careful. Note that (6d) also does not apply when $p=n_j$, no matter if X_j is singular or non-singular.

We avoid the problem with singularity by reparametrizing the problem using the QR-decomposition $X_j = Q_j R_j$, where Q_j is $n_j \times r_j$ and satisfies $Q_j' Q_j = I$, while R_j is $r_j \times p$, and upper triangular. Here $r_j = \mathbf{rank}(X_j) \leq \min(n_j, p)$. Then

$$\underline{y}_j = \tilde{X}_j \underline{\tilde{b}}_j + \underline{\varepsilon}_j, \tag{8a}$$

$$\underline{\tilde{b}}_j = \tilde{Z}_j \gamma + \underline{\tilde{\delta}}_j, \tag{8b}$$

with $\tilde{X}_j = Q_j$, $\underline{\tilde{b}}_j = R_j \underline{b}_j$, $\tilde{Z}_j = R_j Z_j$, and $\underline{\tilde{\delta}_j} = R_j \underline{\delta}_j$. Thus

$$\mathbf{E}\left(\begin{bmatrix}\underline{\varepsilon}_j \\ \underline{\tilde{\delta}}_j\end{bmatrix}\begin{bmatrix}\underline{\varepsilon}_j' & \underline{\tilde{\delta}}_j'\end{bmatrix}\right) = \begin{bmatrix}\sigma^2 I & 0 \\ 0 & \tilde{\Omega}_j\end{bmatrix}. \tag{8c}$$

with

$$\tilde{\Omega}_j = R_j \Omega R_j' = \sum_{g=1}^{G} \xi_g R_j C_g R_j' = \sum_{g=1}^{G} \xi_g \tilde{C}_{jg}.$$

We can now apply our likelihood functions (and our algorithm) to this new reparametrized system (8). Note that the parameters are exactly the same as in the old parametrization, but the new first-level regressors are orthonormal, and there is a different Ω_j for each j.

3.4. **Initial Estimate of the Variance Components.** Define $\underline{\hat{b}}_j = \tilde{X}_j' \underline{y}_j$ and

$$\underline{H}_j = (\underline{\hat{b}}_j - \tilde{Z}_j \gamma)(\underline{\hat{b}}_j - \tilde{Z}_j \gamma)'$$

Then $b_j - Z_j \gamma = \underline{\tilde{\delta}}_j + \tilde{X}_j' \underline{\varepsilon}_j$ and

$$\mathbf{E}(\underline{H}_j) = \sum_{g=1}^{G} \xi_g \tilde{C}_{jg} + \sigma^2 I.$$

Of course γ is unknown, but we have the least squares estimate $\hat{\gamma}$, which is consistent under quite general conditions. This suggest to minimize the linear least squares loss function

$$\sigma(\xi, \sigma^2) = \sum_{j=1}^{m} \mathbf{tr}\ (H_j - \sum_{g=1}^{G} \xi_g \tilde{C}_{jg} - \sigma^2 I)^2,$$

where

$$H_j = (b_j - \tilde{Z}_j \hat{\gamma})(b_j - \tilde{Z}_j \hat{\gamma})', \tag{9}$$

and $b_j = \tilde{X}_j' y_j$.

3.5. **Algorithm.** Our algorithm uses the *method of scoring* to minimize the FIML or REML loss function. More precisely, we compute FIML estimates by minimizing $\mathscr{L}_F(\gamma,\sigma^2,\xi)$ and we compute REML estimates by minimizing $\mathscr{L}_R(\gamma,\sigma^2,\xi) = \mathscr{L}_{(}\gamma,\sigma^2,\xi) + \log\mathbf{det}(\sum_{j=1}^m \tilde{Z}_j'\tilde{W}_j^{-1}\tilde{Z}_j)$.

Suppose θ is the current parameter vector with the $m+G+1$ elements (γ,σ^2,ξ), $g(\theta)$ is the value of the partials, and $H(\theta)$ is the expected value of the matrix of second derivatives. The update formula for scoring is

$$\theta^{(k+1)} = \theta^{(k)} - H(\theta^{(k)})^{-1}g(\theta^{(k)})$$

Formulas for the necessary first derivatives and for the expected values of the second derivatives are in De Leeuw and Meijer [2008b, p. 33-39]. We adapt them here to our context and simplify them slightly.

Matters can be simplified considerably, by observing that the expected values of the mixed second partials of γ and the variance components σ^2 and ξ are zero. Thus

$$\gamma^{(k+1)} = \gamma^{(k)} - \left[\left.\frac{\partial^2\mathscr{L}}{\partial\gamma\partial\gamma}\right|_{\gamma=\gamma^{(k)}}\right]^{-1}\left.\frac{\partial\mathscr{L}}{\partial\gamma}\right|_{\gamma=\gamma^{(k)}}.$$

Since

$$\left.\frac{\partial\mathscr{L}}{\partial\gamma}\right|_{\gamma=\gamma^{(k)}} = -2\sum_{j=1}^m \tilde{Z}_j'\tilde{W}_j^{-1}(\tilde{b}_j - \tilde{Z}_j\gamma^{(k)}),$$

$$\left.\frac{\partial^2\mathscr{L}}{\partial\gamma\partial\gamma}\right|_{\gamma=\gamma^{(k)}} = 2\sum_{j=1}^m \tilde{Z}_j'\tilde{W}_j^{-1}\tilde{Z}_j,$$

we see that, both for FIML and REML,

$$\gamma^{(k+1)} = \left[\sum_{j=1}^m \tilde{Z}_j'\tilde{W}_j^{-1}\tilde{Z}_j\right]^{-1}\sum_{j=1}^m \tilde{Z}_j'\tilde{W}_j^{-1}\tilde{b}_j. \tag{10}$$

Thus in each scoring iteration the update of $\gamma^{(k)}$ is the weighted least squares estimate of γ using the current variance components as weights. It does not depend on the current value of γ.

To update the variance components for FIML we have, using H_j from Equation (9),

$$\frac{\partial\mathscr{L}_F}{\partial\sigma^2} = -\sum_{j=1}^m\left\{(n_j - r_j)\left(\log\sigma^2 - \frac{s_j^2}{(\sigma^2)^2}\right) - \mathbf{tr}\,\tilde{W}_j^{-1}(H_j - \tilde{W}_j)\tilde{W}_j^{-1}\right\},$$

$$\frac{\partial\mathscr{L}_F}{\partial\xi_g} = -\sum_{j=1}^m \mathbf{tr}\,\tilde{W}_j^{-1}(H_j - \tilde{W}_j)\tilde{W}_j^{-1}\tilde{C}_{jg},$$

and for the expected values of the second derivatives

$$\mathbf{E}\left[\frac{\partial^2\mathscr{L}_F}{\partial\sigma^2\partial\sigma^2}\right]=\sum_{j=1}^m\left\{\frac{n_j-r_j}{(\sigma^2)^2}+\mathbf{tr}\,\tilde{W}_j^{-2}\right\},$$

$$\mathbf{E}\left[\frac{\partial^2\mathscr{L}_F}{\partial\sigma^2\partial\xi_g}\right]=\sum_{j=1}^m\tilde{W}_j^{-1}\tilde{C}_{jg}\tilde{W}_j^{-1},$$

$$\mathbf{E}\left[\frac{\partial^2\mathscr{L}_F}{\partial\xi_g\partial\xi_h}\right]=\sum_{j=1}^m\tilde{W}_j^{-1}\tilde{C}_{jg}\tilde{W}_j^{-1}\tilde{C}_{jh}.$$

For REML we have

$$\frac{\partial\mathscr{L}_R}{\partial\sigma^2}=\frac{\partial\mathscr{L}_F}{\partial\sigma^2}-\mathbf{tr}\,A\Lambda,$$

$$\frac{\partial\mathscr{L}_R}{\partial\xi_g}=\frac{\partial\mathscr{L}_F}{\partial\xi_g}-\mathbf{tr}\,A\Psi_g,$$

where

$$A=\left(\sum_{j=1}^m\tilde{Z}_j'\tilde{W}_j^{-1}\tilde{Z}_j\right)^{-1},$$

and

$$\Lambda=\sum_{j=1}^m\tilde{Z}_j'\tilde{W}_j^{-2}\tilde{Z}_j,$$

$$\Psi_g=\sum_{j=1}^m\tilde{Z}_j'\tilde{W}_j^{-1}\tilde{C}_{jg}\tilde{W}_j^{-1}\tilde{Z}_j.$$

Also

$$\mathbf{E}\left[\frac{\partial^2\mathscr{L}_R}{\partial\sigma^2\partial\sigma^2}\right]=\mathbf{E}\left[\frac{\partial^2\mathscr{L}_F}{\partial\sigma^2\partial\sigma^2}\right]-\mathbf{tr}\,\Lambda A\Lambda A,$$

$$\mathbf{E}\left[\frac{\partial^2\mathscr{L}_R}{\partial\sigma^2\partial\xi_g}\right]=\mathbf{E}\left[\frac{\partial^2\mathscr{L}_F}{\partial\sigma^2\partial\xi_g}\right]-\mathbf{tr}\,\Lambda A\Psi_gA,$$

$$\mathbf{E}\left[\frac{\partial^2\mathscr{L}_R}{\partial\xi_g\partial\xi_h}\right]=\mathbf{E}\left[\frac{\partial^2\mathscr{L}_F}{\partial\xi_g\partial\xi_h}\right]-\mathbf{tr}\,\Psi_gA\Psi_hA,$$

3.6. **Post-processing.** If we have estimates of the parameters, we can use these to compute estimates of the regression coefficients and the predicted values, and we can use the expected values of the second derivatives associated with the scoring method to compute standard errors.

Other quantities that are often of interest are the best linear unbiased estimates of the random regression coefficients (the means of the conditional distribution of the regression coefficients given the data). These are [De Leeuw and Meijer, 2008b, p. 26-28]

$$\hat{b}_j = \tilde{\Omega}_j \tilde{W}_j^{-1} b_j + (I - \tilde{\Omega}_j \tilde{W}_j^{-1}) \tilde{Z}_j \gamma,$$

i.e. they are a matrix weighted mean of the ordinary least squares regression coefficients b_j and the maximum likelihood estimates $\tilde{Z}_j \gamma$. The $\hat{b}_j$ are also known as the *shrinkage estimates*.

3.7. **Current R Implementation.**

The complete `R` code for the multilevel function is in Section **??**. In our implementation we use various data structures implemented as lists of lists. It would be better, in many respects, to use `S3` or `S4` objects, but that is on the agenda of future improvements. The first data structure is `allData`, which is a list of m lists. In list j we store X_j, Z_j and y_j, where the y_j may have missing data. `allData` is the input to the `multilevel` program.

The data are used to compute `allMulti`, which is another list of m lists. In these we store copies of X_j, Z_j and y_j, but cleaned up. Missing data are eliminated, together with the corresponding rows of the X_j. If the resulting X_j have zero columns, then these are elimated as well, together with the corresponding rows of the Z_j. In addition `allMulti` stores $X_j'X_j$, $(X_j'X_j)^{-1}$, $X_j'y_j$, b_j, $y_j - X_j b_j$, and s_j^2. One possible improvement of the algorithm is to be a bit less generous with reserving storage for local data.

Two more lists are needed to start the iterations. `omStruc` is a list with the matrices C_g, while `parStruc` holds the current copies of the parameter estimatesγ, σ^2, and ξ. The program `makeIniEst` computes initial estimates of γ and σ^2 by either the one-step or the two-step ordinary least squares method discussed in Subsection 3.1. Initial consistent estimates of ξ are computed with the method similar in Subsection 3.4.

The program `omMake` makes Ω from the C_g. In a future version there will be a similar program `sgMake`, which will create $\Sigma = \mathbf{E}(\underline{\epsilon}_j \underline{\epsilon}_j')$ from a number of parameters θ, with options for example to handle auto-regressive error structures.

During the iterations auxilary quantities such as the negative log-likelihood, the first derivatives, and the expected values of the second derivatives are stored in a list `auxStruct`. If the `multilevel` program is called with `verbose=TRUE` then intermediate function values and gradient norms for each iteration are printed out.

After convergence we fill another list of lists with post-processing results. `allPost` stores, for each group, the least squares estimate b_j, the maximum likelihood estimate $Z_j\gamma$, and the BLUP estimates $\hat{b}_j$ of the regression coefficients. For each of these three different estimates of the regression coefficients we also compute the predicted values. There is enough information available to also easily compute the standard errors of all these parameter estimates, regression coefficients, and predicted values, because `multilevel` returns a list with `parStruc`, `auxStruc`, `allPost`, and `allMulti`.

The program `makePredY` is used for repeated measure data, in which groups are individuals, with measurements on a single variable at different time points. Once we have computed our estimates, we can plot the predicted values as a continuous curve, with continuous confidence bounds.

4. Additional Regressors

5. Multivariate Extensions

5.1. Dispersion Parametrization. The matrices V_H, V_D and V_M are assumed to have circumplex structure. This means, for example, that the covariance between hours i and j only depends on their circular distance, i.e. the minimum of the clockwise and the counter-clockwise distance. For the seven days of the week this means, for example, that

$$V_D = \begin{array}{|r|ccccccc|}
\hline
 & Mo & Tu & We & Th & Fr & Sa & Su \\
\hline
Mo & v_0 & v_1 & v_2 & v_3 & v_3 & v_2 & v_1 \\
Tu & v_1 & v_0 & v_1 & v_2 & v_3 & v_3 & v_2 \\
We & v_2 & v_1 & v_0 & v_1 & v_2 & v_3 & v_3 \\
Th & v_3 & v_2 & v_1 & v_0 & v_1 & v_3 & v_3 \\
Fr & v_3 & v_3 & v_2 & v_1 & v_0 & v_1 & v_2 \\
Sa & v_2 & v_3 & v_3 & v_2 & v_1 & v_0 & v_1 \\
Su & v_1 & v_2 & v_3 & v_3 & v_2 & v_1 & v_0 \\
\hline
\end{array}$$

Thus we need $13+4+7+1=25$ parameters to model the dispersion of $\underline{y}$, and $43-2=41$ parameters to model its expected value. A total of 66 parameters is used to model the 8760 observations.

If necessary we can also restrict the dispersion matrices, for instance by requiring in V_D that $v_k = v_0\rho^k$. Then each of the three matrices V_H, V_D and V_M is described by only two parameters, one for the variance and one for the correlation. Thus, for example,

$$V_D = \omega_D^2 \begin{array}{|c|ccccccc|}
\hline
 & Mo & Tu & We & Th & Fr & Sa & Su \\
\hline
Mo & \rho^0 & \rho^1 & \rho^2 & \rho^3 & \rho^3 & \rho^2 & \rho^1 \\
Tu & \rho^1 & \rho^0 & \rho^1 & \rho^2 & \rho^3 & \rho^3 & \rho^2 \\
We & \rho^2 & \rho^1 & \rho^0 & \rho^1 & \rho^2 & \rho^3 & \rho^3 \\
Th & \rho^3 & \rho^2 & \rho^1 & \rho^0 & \rho^1 & \rho^2 & \rho^3 \\
Fr & \rho^3 & \rho^3 & \rho^2 & \rho^1 & \rho^0 & \rho^1 & \rho^2 \\
Sa & \rho^2 & \rho^3 & \rho^3 & \rho^2 & \rho^1 & \rho^0 & \rho^1 \\
Su & \rho^1 & \rho^2 & \rho^3 & \rho^3 & \rho^2 & \rho^1 & \rho^0 \\
\hline
\end{array}$$

6. The Lebec Data

7. Discussion

7.1. **OLS.**

In a previous paper we analyzed the Lebec data using the model with $Z = I_H \oplus I_D \oplus I_M$ and with $V_H = V_D = V_M = 0$.

7.2. **ARIMA.**

7.3. **Structural TS Models.**

7.4. **Tucker Models.**

References

J. De Leeuw and E. Meijer, editors. *Handbook of Multilevel Analysis*. Springer, 2008a.

J. De Leeuw and E. Meijer. Introduction to Multilevel Analysis. In J. De Leeuw and E. Meijer, editors, *Handbook of Multilevel Analysis*, chapter 1, pages 1–75. Springer Verlag, 2008b.

J. Hemelrijk. Underlining Random Variables. *Statistica Neerlandica*, 20:1–7, 1966.

DEPARTMENT OF STATISTICS, UNIVERSITY OF CALIFORNIA, LOS ANGELES, CA 90095-1554

E-mail address, Jan de Leeuw: deleeuw@stat.ucla.edu

URL, Jan de Leeuw: http://gifi.stat.ucla.edu