Journal of Statistical Software

July 2011, Volume 43, Code Snippet 1. *http://www.jstatsoft.org/*

`fmixed`: A SAS Macro for Smoothing-Spline-Based Functional Mixed Effects Models

Ziyue Liu
Indiana University, Indianapolis

Wensheng Guo
University of Pennsylvania

Abstract

In this article we implement the smoothing-spline-based functional mixed effects models (Guo 2002) by a SAS macro by exploiting the connection between mixed effects models and smoothing splines. The macro can handle flexible design matrices and is easy to use. Input parameters and output results are described and explained. A numeric example and a real data example are used for illustration.

Keywords: functional mixed effects models, smoothing spline, SAS macro.

1. Introduction

Functional mixed effects models (see Guo 2002; Morris and Carroll 2006, among others) are extremely powerful tools in modeling longitudinal profiles. Among them, the smoothing-spline-based functional mixed effects model (SSFMM; Guo 2002) has been gaining attentions in recent years. For example Stacey *et al.* (2009) used SSFMM to fit the subject specific curves of cytokines in human immunodeficiency virus infections, and Fagiolini *et al.* (2009) used SSFMM to evaluate the temporal differences of treatment effects in bipolar disorder patients. However, the lack of a corresponding software package still limits the general accessibility of SSFMM. In this article, we implement SSFMM in SAS and demonstrate how to use the resultant macro `fmixed` by a numeric example and a data example.

In literature, there are several approaches in functional mixed effects modeling, and some of them have public-accessible softwares implemented. Rice and Wu (2001) used B-spline basis and the mixed effects modeling is on the B-spline coefficients. James *et al.* (2000) used a combination of B-spline and functional principal components to reduce the number of parameters required for the mixed effect modeling of the B-spline coefficients. Guo (2002) modeled both the functional fixed effects and random effects as smoothing splines, which only requires limited number of parameters. Wu and Zhang (2002) used local polynomial, the mixed effects modeling is on the polynomial coefficients and this method has been implemented in the companion MATLAB softwares to Wu and Zhang (2006). Morris and Carroll (2006) used wavelets which can handle abrupt and spiky signals, Bayesian approach was adopted for the mixed effects modeling on the wavelet coefficients and free software is available as indicated in their paper. Di *et al.* (2009) and Aston *et al.* (2010) used functional principal components as the basis and this approach has been implemented in Crainiceanu and Goldsmith (2010). The goal of this article is to implement Guo (2002) by a SAS macro, which adds one alternative among the large selection of different tools. A thorough comparison of those different approaches in functional mixed effects modeling, however, is beyond the scope of this article.

The key of the implementation is the equivalence between a smoothing spline and its Bayesian model (Wahba 1978). By utilizing this equivalence, Wang (1998) showed that a smoothing spline can be represented as a mixed effects model and consequently it can be implemented by SAS `PROC MIXED`. He also showed that the restricted maximum likelihood (REML) based on the Bayesian model outperforms generalized cross validation in selecting the smoothing parameters. Following this approach, Guo (2002) showed that SSFMM can also be represented as a mixed effects model and subsequently inference can be carried out based on REML such as likelihood ratio test. The implementation in this article follows the approach of Guo (2002). The SAS macro `fmixed` is written using SAS/STAT and SAS/IML (SAS Institute Inc. 2008a,b), and can handle flexible design matrices. The items needed for fitting the mixed effects model representation of SSFMM are calculated by SAS/IML, the model fitting is done by `PROC MIXED`, and the estimates and 95% confidence intervals of the fixed functions and the individual fittings are obtained by rearranging the `PROC MIXED` output using SAS data steps.

This article is organized as follows. In Section 2 we review the methodological background of SSFMM. In Section 3 we introduce the input parameters for `fmixed` and its output data sets. In Section 4 we illustrate `fmixed` by a numeric example. In Section 5 we apply `fmixed` to a data example. And in Section 6 we give some discussions on the issue of computational burdens.

2. Methodological background

In this section we briefly review SSFMM proposed by Guo (2002).

2.1. The general model

For subject $i = 1, \ldots, n$ and time point $j = 1, \ldots, m_i$, the observed value y_{ij} is modeled as

$$y_{ij} = X_{ij}\boldsymbol{\beta}\left(t_{ij}\right) + Z_{ij}\boldsymbol{\alpha}_i\left(t_{ij}\right) + e_{ij}, \qquad e_{ij} \sim \mathrm{N}\left(0, \sigma_e^2\right), \tag{1}$$

where $\boldsymbol{\beta}(t) = \{\beta_1(t), \ldots, \beta_p(t)\}^\top$ is a $p\times 1$ vector of fixed functions, $\boldsymbol{\alpha}_i(t) = \{\alpha_{1i}(t), \ldots, \alpha_{qi}(t)\}^\top$ is a $q \times 1$ vector of random functions that are modeled as realizations from the same stochastic processes, X_{ij} and Z_{ij} are design matrices which contain the information of experimental designs and covariates, ande_{ij} is the independent and identically distributed error term.

Each element of $\boldsymbol{\beta}(t)$ is modeled as a cubic smoothing spline with the equivalent Bayesian model as

$$\beta_k\left(t\right) = b_{1k} + b_{2k}t + \lambda_{bk}^{-1/2}\int_0^t W_k\left(s\right)\, ds, \qquad k = 1, \ldots, p, \tag{2}$$

where $\{b_{1k}, b_{2k}\}^\top \sim \mathrm{N}(0, \tau I)$, $\tau \to \infty$ and $W_k(s)$ is a standard Wiener process. Each element of $\boldsymbol{\alpha}_i(t)$ is modeled as independent across subjects and as realizations from the same stochastic process

$$\alpha_l(t) = a_{1l} + a_{2l}t + \lambda_{\alpha l}^{-1/2} \int_0^t W_l(s)\, s\, ds, \qquad l = 1, \dots, q, \tag{3}$$

where $\{a_{1l}, a_{2l}\}^\top \sim \mathrm{N}(0, D)$ and $W_l(s)$ is also a standard Wiener process. And $\beta_k(t)$ and $\alpha_{il}(t)$ are mutually independent and are independent across k and l.

The reproducing kernels corresponding to (2) and (3) are usually difficult to calculate and we will use an alternative Hilbert space on $[0, 1]$, where the reproducing kernels are specified in terms of Bernoulli polynomials (Gu 2002, Section 2.3.3). These kernels are easier to calculate and more numerically stable in the equivalent mixed effects model estimation.

We define the corresponding reproducing kernel through direct sum: $R = R^0 \oplus R^1$ (Gu 2002, Section 2.3.3). We first define the Bernoulli polynomials and then the reproducing kernels:

$$\begin{aligned}
k_1(t) &= t - 0.5,, \qquad k_2(t) = \frac{1}{2}\{k_1^2(t) - \frac{1}{12}\}, \\
k_4(t) &= \frac{1}{24}\{k_1^4(t) - \frac{1}{2}k_1^2(t) + \frac{7}{240}\}, \\
R^0(t_1, t_2) &= 1 + k_1(t_1)\, k_2(t_2), \\
R^1(t_1, t_2) &= k_2(t_1)\, k_2(t_2) - k_4(t_1 - t_2). \qquad (4)
\end{aligned}$$

A cubic spline $f(t)$ can be expanded as the following(Gu 2002, Section 2.3.3)

$$f(t) = \int_0^1 f(s)\; ds + k_1(t) \int_0^1 f^{(1)}(s)\; ds + \int_0^1 (k_2(t) - k_2(t - s))\, f^{(2)}(s)\; ds. \tag{5}$$

Using this expansion, we can remodel the functional fixed effects and random effects in (2) and (3) as

$$\beta_k(t) = b_{k1} + b_{2k}(t - 0.5) + \lambda_{bk}^{-1/2} \int_0^1 (k_2(t) - k_2(t - s))\, W_k(s)\; ds, \tag{6}$$

$$\alpha_l(t) = a_{1l} + a_{l2}(t - 0.5) + \lambda_{\alpha l}^{-1/2} \int_0^1 (k_2(t) - k_2(t - s))\, W_l(s)\; ds, \tag{7}$$

where $W_k(s)$, $W_l(s)$ and other coefficients are the same as before.

2.2. The mixed effects model representation

Similar to Wang (1998), Equation 6 and Equation 7 can be represented as equivalent mixed effects model at the design points $\boldsymbol{t}_i = \{t_{i1}, \dots, t_{i,m_i}\}$. We define T_i as an $n \times 2$ matrix with the first column as one and the second as $\boldsymbol{t}_i - 0.5$, $\boldsymbol{d}_k = \{b_{1k}, b_{2k}\}^\top$, and $\boldsymbol{c}_{li}$ is a realization of $\{a_{1l}, a_{2l}\}^\top \sim \mathrm{N}(0, D)$. Each element of $\boldsymbol{\beta}(t)$ and $\boldsymbol{\alpha}_i(t)$ can be represented as mixed effects models

$$\begin{aligned}
\beta_k(\boldsymbol{t}_i) &= T_i \boldsymbol{d}_k + u_k(\boldsymbol{t}_i), \quad u_k(\boldsymbol{t}_i) \sim \mathrm{N}\left(0, R^1(\boldsymbol{t}_i, \boldsymbol{t}_i)/\lambda_{bk}\right), \qquad (8) \\
\alpha_{li}(\boldsymbol{t}_i) &= T_i \boldsymbol{c}_{li} + v_{li}(\boldsymbol{t}_i), \quad v_{li}(\boldsymbol{t}_i) \sim \mathrm{N}\left(0, R^1(\boldsymbol{t}_i, \boldsymbol{t}_i)/\lambda_{\alpha k}\right). \qquad (9)
\end{aligned}$$

Collecting items in $\boldsymbol{\beta}(t)$ and $\boldsymbol{\alpha}_i(t)$, we have the mixed effects model representation for Equation 1

$$\boldsymbol{y}_i = \tilde{X}_i\tilde{\boldsymbol{d}} + \tilde{Z}_i\tilde{\boldsymbol{\alpha}}_i + \boldsymbol{e}_i, \qquad i = 1, \ldots, n, \tag{10}$$

where

$$\begin{aligned}
\boldsymbol{y}_i &= \{y_{i1}, \ldots, y_{i,m_i}\}^\top, \\
\tilde{X}_i &= \left[\{X_{i1}^\top, \ldots, X_{i,m_i}^\top\}^\top\right](I_p \otimes T_i), \\
\tilde{Z}_i &= \{\{X_{i1}^\top, \ldots, X_{i,m_i}^\top\}^\top, \left[\{Z_{i1}^\top, \ldots, Z_{i,m_i}^\top\}^\top\right](I_q \otimes T_i), \{Z_{i1}^\top, \ldots, Z_{i,m_i}^\top\}^\top\}, \\
\tilde{\boldsymbol{d}} &= \{\boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_p^\top\}^\top, \\
\tilde{\boldsymbol{\alpha}}_i &= \{u_1(\boldsymbol{t}_i)^\top, \ldots, u_p(\boldsymbol{t}_i)^\top, \boldsymbol{c}_{1i}^\top, \ldots, \boldsymbol{c}_{qi}^\top, v_{1i}(\boldsymbol{t}_i)^\top, \ldots, v_{qi}(\boldsymbol{t}_i)^\top\}^\top, \\
\boldsymbol{e}_i &= \{e_{i1}, \ldots, e_{i,m_i}\}^\top,
\end{aligned}$$

and $\otimes$ denotes the Kronecker product. In this representation, $\tilde{\boldsymbol{d}}$ are the fixed effects and $\tilde{\boldsymbol{\alpha}}_i$ are the random effects as in the conventional sense, and this model can be fitted with existing softwares such as SAS `PROC MIXED`.

2.3. Estimation and inference

The smoothing parameters and other variance components are estimated by REML. This approach is justified by two results: for correlated errors Wang (1998) showed that REML outperforms other methods such as generalized cross validation in selecting the smoothing parameters and in estimating other variance components, for independent errors Kohn *et al.* (1991) had a similar result. Based on REML, inference can be performed by likelihood ratio test and the null distribution of the test statistic can be approximated by simulation.

The spline estimates on the design points are the empirical best linear unbiased predictions (EBLUPs) conditional on the REML estimates. The function values other than at the design points can also be estimated as EBLUPs

$$\hat{\beta}_k(s) = \begin{pmatrix} 1 & s - 0.5 \end{pmatrix} \begin{pmatrix} \hat{b}_{1k} \\ \hat{b}_{2k} \end{pmatrix} + \hat{u}_k(\boldsymbol{t})^\top R^1(\boldsymbol{t}, \boldsymbol{t})^- R^1(\boldsymbol{t}, s), \tag{11}$$

$$\hat{\alpha}_{li}(s) = \begin{pmatrix} 1 & s - 0.5 \end{pmatrix} \begin{pmatrix} \hat{a}_{1li} \\ \hat{a}_{2li} \end{pmatrix} + \hat{v}_{li}(\boldsymbol{t}_i)^\top R^1(\boldsymbol{t}_i, \boldsymbol{t}_i)^- R^1(\boldsymbol{t}_i, s), \tag{12}$$

where $\boldsymbol{t} = \{\boldsymbol{t}_1^\top, \ldots, \boldsymbol{t}_n^\top\}^\top$, $\hat{u}_k(t)$ and $\hat{v}_{li}(t)$ are EBLUPs of $u_k(t)$ and $v_{li}(t)$, A^- is the Moore Penrose inverse of A, and $s \in [0, 1]$. Their confidence intervals can be constructed similarly.

3. The input parameters and the output data sets

3.1. Input parameters

The following is a list of the input parameters for `fmixed`.

```
%fmixed(data = , response_var = , t_var = , sid = ,
  fixed_effects = %bquote(), random_effects = %bquote(),
  random_int_slope_type = %bquote(), save_location = %bquote())
```

where

- `data`: Specifies the working data set with the same format as for `PROC MIXED`, except that all the variables are required to be numeric. Character variable is not allowed. Hence, preprocessing is necessary by the user to convert character variables to numeric.
- `response_var`: Specifies the dependent variable, or the "y" variable.
- `t_var`: Specifies the independent variable, or the "time" variable.
- `sid`: Specifies the variable that identifies the subjects in the model.
- `fixed_effects = %bquote()`: Specifies the design matrix for the functional fixed effects, or the "X" matrix. There are three scenarios:
  - `fixed_effects = %bquote(0)`: A fixed linear trend will be fitted.
  - `fixed_effects = %bquote(1)`: A fixed cubic smoothing spline will fitted for all the subjects.
  - `fixed_effects = %bquote(x1,x2)`: One fixed cubic smoothing spline for effect `x1` and one for `x2`. `x1` and `x2` are separated by a comma.
- `random_effects = %bquote()`: Specifies the design matrix for the functional random effects, or the "Z" matrix. There are three scenarios:
  - `random_effects = %bquote(0)`: A random linear trend will be used.
  - `random_effects = %bquote(1)`: A random cubic smoothing spline will be fitted with subjects defined by `sid`.
  - `random_effects = %bquote(x1,x2)`: One random cubic smoothing spline will be fitted for effect `x1` and one for `x2`. `x1` and `x2` are separated by a comma.
- `random_int_slope_type = %bquote()`: Specifies the covariance structure for the random intercept and slope of the random splines. This parameter can be either "`vc`", which stands for variance component, or "`un`", which stands for unstructured. If this parameter is left blank, the macro will use the default setting as "`vc`". Note that by using "`vc`", the random intercept and the random slope will be independent and each will have its own variance.
- `save_location = %bquote()`: Specifies a library name to save the output data sets. If this parameter is left blank, the macro will save the output data sets to the default library "`work`".

3.2. Output data sets

Besides the default output from `PROC MIXED` and the intermediate data sets, macro `fmixed` generates four data sets containing the results. In the following names of the data sets, "`response_var`" will be replaced by the actual name of the dependent variable. Those output data sets will be stored at the directory specified by `save_location = %bquote()` or at the "`work`" library if no save location is given.

- `fit_stat_response_var`: Contains the fit statistics such as `-2 Res Log Likelihood`.
- `cov_parms_response_var`: Contains the estimated covariance parameters as follows.
  - `Fixed Spline:$xk$`: The estimated smoothing parameter $1/\lambda_{bk}$ for the fixed spline corresponding to variable $xk$.
  - `Random Spline:intercept`: The estimated variance for the intercept part in the random spline.
  - `Random Spline:slope`: The estimated variance for the slope part in the random spline.
  - `Random Spline:spline`: The estimated variance, $1/\lambda_{\alpha}$, for the smooth part in the random spline.
  - `Residual`: The estimated variance for the error terms.
- `blup_response_var`: Contains BLUPs for all the data points and the 95% confidence intervals with variables
  - `pred`: Point estimates.
  - `l_bound`: Lower bounds.
  - `u_bound`: Upper bounds.
- `fun_fixed_response_var`: Contains the estimated functional fixed effects. Corresponding to the three scenarios in specifying the functional fixed effects, this data set can have variables as the following.
  - `fixed_effects = %bquote(0)`: Data `fun_fixed_response_var` is not created. The results about the linear trend are available in the default output.
  - `fixed_effects = %bquote(1)`: There are three variables.
    - `t_var`: The independent variable, or the "time" variable.
    - `fun1`: The estimated function values.
    - `StdErr_fun1`: The standard errors of the estimated function values.
  - `fixed_effects = %bquote(x1,x2)`: There are three types of variables.
    - `t_var`: The independent variable, or the "time" variable.
    - `fun_xi`: The estimated function values for effect `xi` where `i = 1,2`.
    - `StdErr_fun_xi`: The standard errors of the above estimated function values.

4. A numeric example

In this section we demonstrate how to use `fmixed` by a numeric example. For subject $i = 1, \ldots, 20$, and equally spaced time points $j = 1, \ldots, 30$ on $[0, 1]$, we generate data as

$$\boldsymbol{y}_i\left(\boldsymbol{t}\right) = 5x_{i1}\sin\left(2\pi\boldsymbol{t}\right) + 5x_{i2}\sin\left(\pi\boldsymbol{t}\right) + \boldsymbol{r}_i\left(\boldsymbol{t}\right) + \boldsymbol{e}_i,$$

where $\boldsymbol{t} = \{t_1, \ldots, t_{30}\}$. For each subject, x_{i1} is assigned by a draw from a uniform distribution on $[-0.5, 0.5]$ and x_{i2} is assigned as 0 or 1 equally likely, $\boldsymbol{r}_i(\boldsymbol{t})$ is a draw from $\mathrm{N}(\boldsymbol{0}, \sigma_1^2\Sigma(\boldsymbol{t}))$

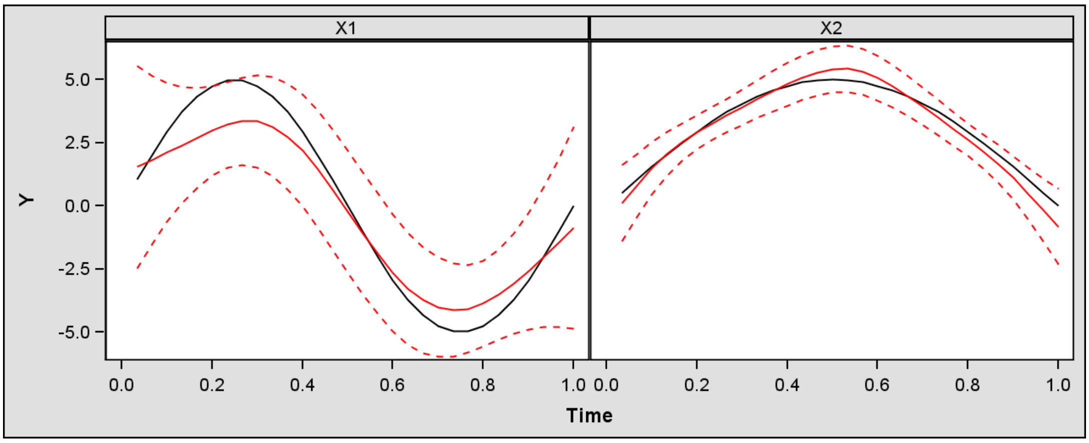

Figure 1: Numeric example; estimated functional effects. The left panel is for `x1` and the right panel is for `x2`. The black solid lines are for the true values and the red lines are for the estimated values with their 95% confidence intervals.

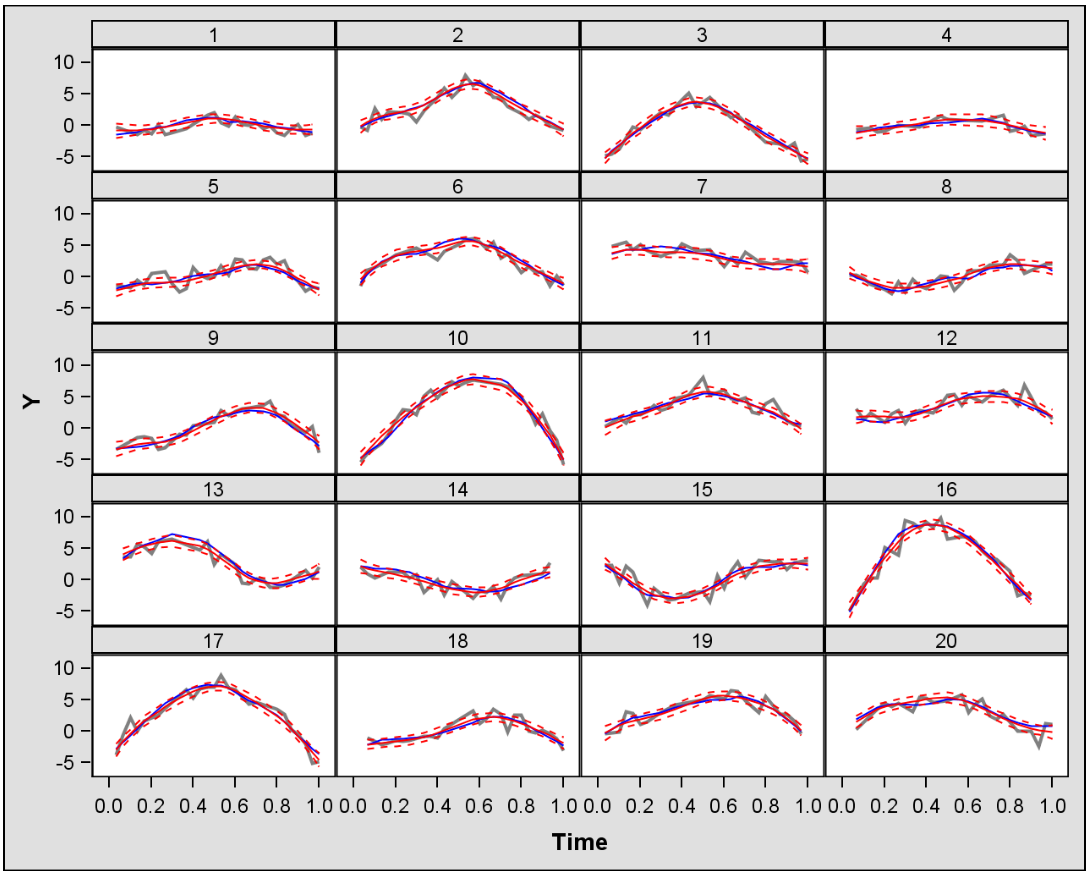

Figure 2: Numeric example; individual fittings. The gray lines are for the data points, the blue lines are for the true values without observational errors, and the red lines are for BLUPs with their 95% confidence intervals.

with $\sigma_1^2 = 900$ and $\Sigma = R^1(\boldsymbol{t}, \boldsymbol{t})$, $\boldsymbol{r}_i(\boldsymbol{t})$s are independent across different subjects, and $\boldsymbol{e}_i$ is a draw from $\mathrm{N}(\boldsymbol{0}, \sigma_2^2 I)$ with $\sigma_2^2 = 1$. To mimic real applications that subjects usually do not have the same number of observations, the generated data points are then subject to random missing with a probability of 0.2.

We call `fmixed` as

```
%fmixed(data = dat_gen, response_var = y, t_var = visit, sid = sid,
  fixed_effects = %bquote(x1,x2), random_effects = %bquote(1),
  random_int_slope_type = %bquote(), save_location = %bquote());
```

The estimated variance components are $\hat{\sigma}_1^2 = 718.80$ and $\hat{\sigma}_2^2 = 0.85$, both of which are slightly under-estimated. The estimated population effects are shown in Figure 1, and the individual fittings are shown in Figure 2. Those two figures show that both the population effects estimations and the individual fittings are reasonably good.

5. Hormone dynamic example

In this section we illustrate `fmixed` by modeling the dynamics of cortisol in patients with both chronic fatigue syndrome and fibromyalgia syndrome (see Crofford *et al.* 2004, and references therein). Chronic fatigue syndrome is characterized with unexplained fatigue which leads to substantial reduction in daily performance, and fibromyalgia syndrome is characterized with persistent chronic pain conditions. Both syndromes are stress related and therefore by studying the dynamics of cortisol, a stress related hormone, we may gain a better understanding of the pathophysiological pathways. Guo (2002) compared the cortisol levels between fibromyalgia patients and healthy controls and he found that the fibromyalgia patients have significantly higher cortisol levels. In this section we concentrate on patients diagnosed with both syndromes compared to healthy controls, and we are interested in comparing the group averages and in obtaining subject specific fittings.

Figure 3 displays the raw data. There are 12 subjects for each of the patient group and the control group, and each subject has hourly cortisol measurements for a 24-hour period. As shown in the figure, both groups exhibit 24-hour cycles which are difficult to capture by parametric models, and the subject specific deviations also need to be flexibly accounted. Therefore we adopt the same approach as in Guo (2002) and fit the following SSFMM to the data:

$$y_{ij} = X_{ij}\boldsymbol{\beta}\left(t_{ij}\right) + \alpha_i\left(t_{ij}\right) + e_{ij}, \quad e_{ij} \sim \mathrm{N}\left(0, \sigma_e^2\right), \tag{13}$$

where y_{ij} is the cortisol concentration for the ith subject and jth hour and $i = 1, \ldots, 24$, $j = 1, \ldots, 24$; $X_{ij} = \{x_{ij1}, x_{ij2}\} = \begin{pmatrix} 1 & 0 \end{pmatrix}$ for the patient group and $\begin{pmatrix} 0 & 1 \end{pmatrix}$ for the control group; $\boldsymbol{\beta}(t) = \{\beta_1(t), \beta_2(t)\}^\top$ where $\beta_1(t)$ is the functional group-average for the patient group and $\beta_2(t)$ for the control group; $\alpha_i(t)$ is the functional subject specific deviation where the random intercept and slope are modeled with a diagonal covariance matrix; and e_{ij} is the error term. The model fitting is done by calling `fmixed` as

```
%fmixed(data = both, response_var = cort, t_var = time, sid = subject,
  fixed_effects = %bquote(x1,x2), random_effects = %bquote(1),
  random_int_slope_type = %bquote(), save_location = %bquote());
```

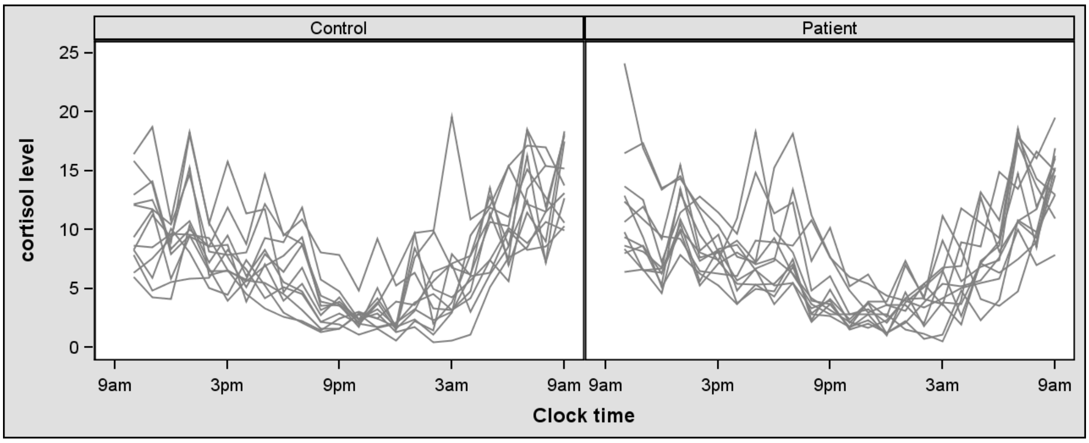

Figure 3: Hormone example; raw data. The left panel is the control group and the right panel is the patient group. The vertical axis indicate cortisol levels as $\mu g/dl$. Each group has 12 subjects, and each subject has hourly cortisol measurements for 24 hours.

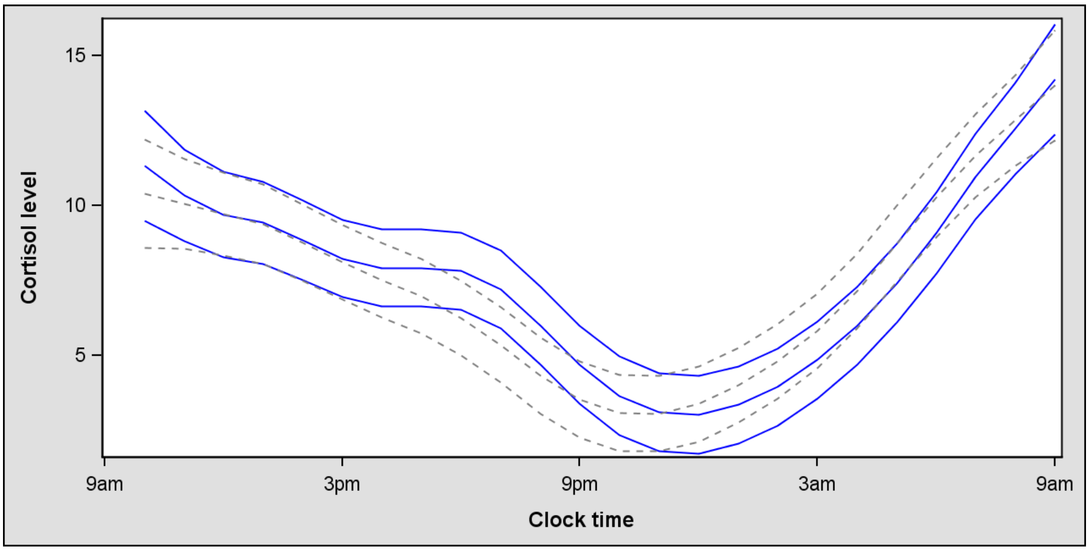

Figure 4: Hormone example; estimated group average curves with their 95% confidence intervals. The gray dashed lines are for the control group, and the blue solid lines are for the patient group.

The estimated group average curves are displayed in Figure 4. The figure shows that the patient group has higher cortisol levels from 3pm to 10pm and lower cortisol levels from 11pm to 8am compared to the control group. This suggests that patients tend to have higher cortisol levels from middle afternoon until early night and lower cortisol levels during sleep.

To test if the group averages are the same, we reformulate the models so that the reduced model is nested in the full model. The full model is

$$y_{ij} = f_1(t_{ij}) + f_2(t_{ij}) + \alpha_i(t_{ij}) + e_{ij}, \qquad \text{for patients,} \tag{14}$$
$$y_{ij} = f_1(t_{ij}) + \alpha_i(t_{ij}) + e_{ij}, \qquad \text{for controls,} \tag{15}$$

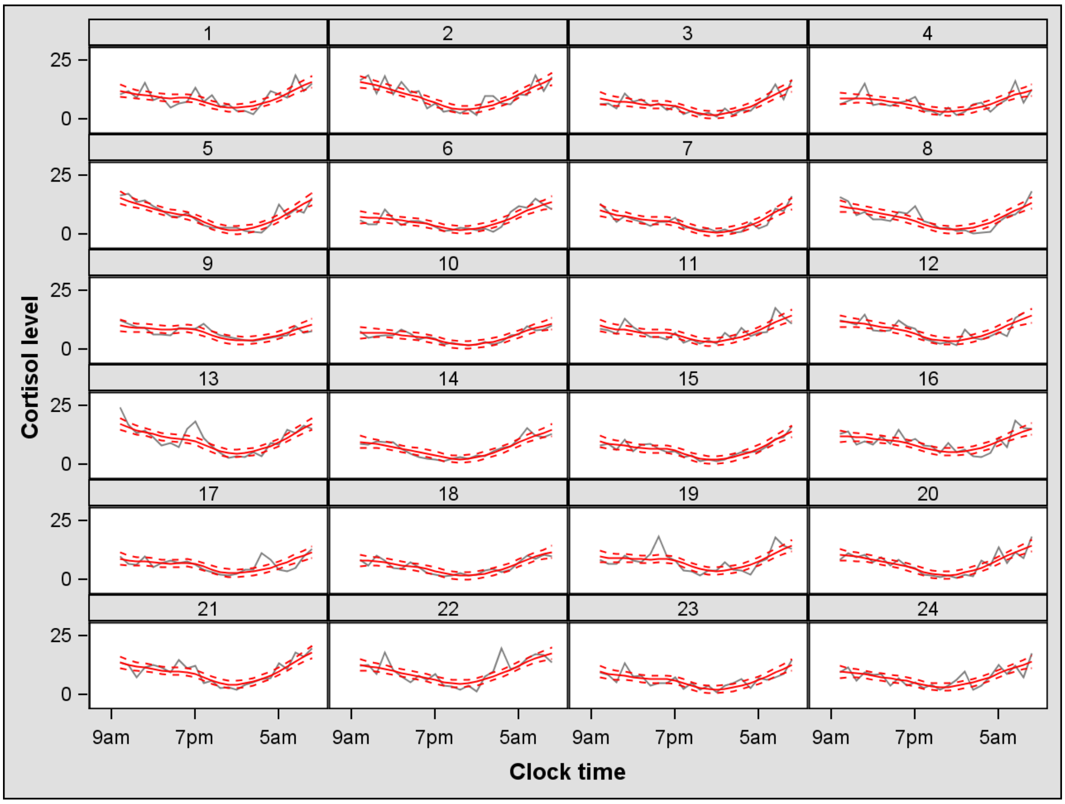

Figure 5: Hormone example: Individual fittings. The gray lines are for the observed data, and the red lines are for BLUPs with their 95% confidence intervals. 1–12 are the patients, and 13–24 are the controls.

where both $f_1(t)$ and $f_2(t)$ are smoothing splines. The reduced model is thus that $f_2(t) \equiv 0$. Note that under the mixed effects model representations the working covariance structure of the model defined in Equation 13, which has zero blocks for the covariance between the patient group and the control group, does not include this reduced model's covariance structure, which has nonzero blocks for the covariance between the patient group and the control group, as a reduced form. With the reformulation, however, the reduced model is special case of the full model defined in Equation 14 and Equation 15 by setting $\boldsymbol{d}_2 = 0$ and $\lambda_{b2}^{-1} = 0$.

The full model is fitted as

```
%fmixed(data = both, response_var = cort, t_var = time, sid = subject,
  fixed_effects = %bquote(x1,x3), random_effects = %bquote(1),
  random_int_slope_type = %bquote(), save_location = %bquote());
```

where $x3 \equiv 1$ for all the subjects, and the reduced model is fitted as

```
%fmixed(data = both, response_var = cort, t_var = time, sid = subject,
  fixed_effects = %bquote(x3), random_effects = %bquote(1),
  random_int_slope_type = %bquote(), save_location = %bquote());
```

The difference of -2 log REML between the full model and the reduced model is 5.92. For the null distribution of the test statistic, Crainiceanu and Ruppert (2004) showed that more weight should be put on the Chi-square distribution with smaller degrees of freedom rather than the conventional $50:50$ mixture of χ_2^2 and χ_3^2. But even with a $100:0$ mixture, the p-value 0.0518 is still not significant at the 0.05 level. The null distributions of the test statistics, on the other hand, can be approximated by simulation. Implementation of the simulation based approach, however, is beyond the scope of this article.

The individual fittings are displayed in Figure 5 which shows that the individual profiles are captured pretty well. To test if a functional random deviation is necessary, we set $\lambda_a^{-1} = 0$ by

```
%fmixed(data = both, response_var = cort, t_var = time, sid = subject,
  fixed_effects = %bquote(x1,x2), random_effects = %bquote(0),
  random_int_slope_type = %bquote(), save_location = %bquote());
```

to fit a reduced model. The difference of -2 log REML between the full model and the reduced model is 17.18, which is highly significant even under the conservative $50:50$ mixture of χ_1^2 and zero. Therefore the smooth random component cannot be dropped from the model.

6. Discussion

By utilizing the mixed effects model representations, we have implemented SSFMM in a SAS macro. This macro is easy to use and it can handle complex experimental designs and include covariates. The computational burden of this approach, however, is heavy. The mixed effects model representations of the functional fixed effects essentially introduce correlations among all the data points, both within and across subjects. Therefore we need to work with matrices with dimension nm for n subjects and m time points. This requires $O(n^2m^2)$ for storage and $O(n^3m^3)$ for calculating the inverse and the determinant. When the number of total observations is in the hundreds such as the numeric example and the hormone data example, it only takes 10 to 15 minutes for the model fitting on a personal computer with 2.40 GHz Intel i5 CPU and 3.42 GB RAM. But the computational time increases quickly as the number of total observations increases. For example it takes about 8 hours for $nm = 1500$ for the same model as the numeric example. Grid computing or mainframe computer are able to handle larger data sets, but the computing resources will still be quickly consumed because of the cubic relationship. An alternative is to adopt the state space approach as suggested in Guo (2002). This approach only requires $O(n^2m)$ for storage and $O(n^2m)$ for calculating the inverse and the determinant. Qin and Guo (2006) implemented this approach for periodic SSFMM without covariates. Unfortunately a general software package for SSFMM based on the state space approach is still unavailable.

References

Aston JAD, Chiou JM, Evans JP (2010). "Linguistic Pitch Analysis Using Functional Principal Component Mixed Effects Models." *Applied Statistics*, **59**, 297–317.

Crainiceanu CM, Goldsmith AJ (2010). "Bayesian Functional Data Analysis Using WinBUGS." *Journal of Statistical Software*, **32**(11), 1–33. URL http://www.jstatsoft.org/v32/i11/.

Crainiceanu CM, Ruppert D (2004). "Likelihood Ratio Tests in Linear Mixed Models with One Variance Component." *Journal of the Royal Statistical Society B*, **66**, 165–185.

Crofford LJ, Young EA, Engleberg NC, Korszun A, Brucksch CB, McClure LA, Brown MB, Demitrack MA (2004). "Basal Circadian and Pulsatile ACTH and Cortisol Secretion in Patients with Fibromyalgia and/or Chronic Fatigue Syndrome." *Brain, Behavior, and Immunity*, **18**, 314–325.

Di CZ, Crainiceanu CM, Caffo BS, Punjabi NM (2009). "Multilevel Functional Principal Component Analysis." *The Annals of Applied Statistics*, **3**, 458–488.

Fagiolini A, Frank E, Axelson DA, Birmaher B, Cheng Y, Curet DE, Friedman ES, Gildengers AG, Goldstein T, Grochocinski VJ, Houck PR, Stofko MG, Thase ME, Thompson WK, Turkin SR, Kupfer DJ (2009). "Enhancing Outcomes in Patients with Bipolar Disorder: Results from the Bipolar Disorder Center for Pennsylvanians Study." *Bipolar Disorder*, **11**, 382–390.

Gu C (2002). *Smoothing Spline ANOVA Models*. Springer-Verlag, New York.

Guo W (2002). "Functional Mixed Effects Models." *Biometrics*, **58**, 121–128.

James GM, Hastie TJ, Sugar CA (2000). "Principal Component Models for Sparse Functional Data." *Biometrika*, pp. 587–602.

Kohn R, Ansley CF, Tharm D (1991). "The Performance of Cross-Validation and Maximum Likelihood Estimators of Spline Smoothing Parameters." *Journal of the American Statistical Association*, **86**, 1042–1050.

Morris JS, Carroll RJ (2006). "Wavelet-Based Functional Mixed Models." *Journal of the Royal Statistical Society B*, **68**, 179–199.

Qin L, Guo W (2006). "Functional Mixed-Effects Model for Periodic Data." *Biostatistics*, **7**, 225–234.

Rice JA, Wu CO (2001). "Nonparametric Mixed Effects Models for Unequally Sampled Noisy Curves." *Biometrics*, **57**, 253–259.

SAS Institute Inc (2008a). *SAS/IML Software, Version 9.2*. Cary, NC. URL http://www.sas.com/.

SAS Institute Inc (2008b). *SAS/STAT Software, Version 9.2*. Cary, NC. URL http://www.sas.com/.

Stacey AR, Norris PJ, Qin L, Haygreen EA, Taylor E, Heitman J, Lebedeva M, DeCamp A, Li D, Grove D, Self SG, Borrow P (2009). "Induction of a Striking Systemic Cytokine Cascade Prior to Peak Viremia in Acute Human Immunodeficiency Virus Type 1 Infection, in Contrast to More Modest and Delayed Responses in Acute Hepatitis B and C Virus Infections." *Journal of Virology*, **83**, 3719–3733.

Wahba G (1978). "Improper Priors, Spline Smoothing and the Problem of Guarding against Model Errors in Regression." *Journal of the Royal Statistical Society B*, **40**, 364–372.

Wang Y (1998). "Smoothing Spline Models with Correlated Random Errors." *Journal of the American Statistical Association*, **93**, 341–348.

Wu H, Zhang JT (2002). "Local Polynomial Mixed-Effects Models for Longitudinal Data." *Journal of the American Statistical Association*, **97**, 883–897.

Wu H, Zhang JT (2006). *Nonparametric Regression Methods for Longitudinal Data Analysis: Mixed-Effects Modeling Approaches*. John Wiley & Sons.

Affiliation:

Ziyue Liu
Division of Biostatistics
Indiana University School of Medicine
410 West 10th Street, Suite 3000
Indianapolis, IN 46202-3002, United States of Ameria
E-mail: ziliu@iupui.edu

Wensheng Guo
Department of Biostatistics and Epidemiology
University of Pennsylvania School of Medicine
423 Guardian Drive, 613 Blockley Hall
Philadelphia, PA 19104-6021, United States of Ameria
E-mail: wguo@mail.med.upenn.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 43, Code Snippet 1 *Submitted:* 2010-09-13
July 2011 *Accepted:* 2011-05-27