simm.fda package
A short description

Niels A. L. Olsen

June 2016

The `simm.fda` package is based on the article `http://arxiv.org/abs/1606.03295` (Olsen et al. 2016), which will be general reference for modelling & methods. This document is meant as reference for the software.

The `simm.fda` package is based on the `pavpop` package by L.L. Raket (link: `https://github.com/larslau/pavpop`), which largely does the same thing as `simm.fda` in a one-dimensional setting. Some of the functions are taken directly from this package, but the two packages are not compatible and have significant differences in their internal structures.

The statistical setting for `simm.fda` is the model given by

$$\boldsymbol{y}_i(t_k) = \boldsymbol{\theta}_i(v_i(t_k)) + \boldsymbol{x}_i(t_k) + \boldsymbol{\varepsilon}_{ik} \tag{1}$$

where v_i is the unknown warp function, $\theta_i : [0, 1] \to \mathbb{R}^d$ the expected trajetory for subject i and $\boldsymbol{x}_i$ and $\boldsymbol{\varepsilon}_i$ represents amplitude covariance and noise. θ_i can either be common to all subjects or subject to a design; $\theta_i(t) = A_i\beta(t)$.

It is assumed that $t \in [0, 1]$, although the model easily generalise to any domain $[a, b]$. However the software only accepts the domain $[0, 1]$, and hence the user must scale her time points to be within the domain $[0, 1]$. This may be changed in the future.

$\boldsymbol{\theta}$ (or the β's) is modeled as a spline, whose basis is assumed to be the same for all coordinates (and βs). `simm.fda` provides functions for making (increasing) b-splines.

Estimation Estimation in (1) is done using localized iterative linearisation. The inner iteration consits of maximizing the posterior likelihood, i.e. predicting the warp given the current parameters, and estimating spline coefficients. The computationally intensive outer iteration consits of maximizing variance parameters in the linearized model. For the inner iterations warp change is used as criterion for convergence. The outer iterations will continue until the likelihood no longer decreases or maximum number of iterations has been reached.

Warp We have $v_i(t_k) := v(t, \boldsymbol{w}_i))$, where $w \in \mathbb{R}^d$ is a latent variable describing the warp. We shall refer to $v(\cdot, \cdot)$ as the *warp function*, and w as the (latent) *warp values.*

Data The y's have to be provided as list of $m_i \times K$ matrices, corresponding to a list of t's. y may contain missing values as long as they are missing in all coordinates.

The amplitude covariances should be provided as function `function(t, param)` with the possibility of multiplying by a scale parameter, as a common scale parameter for warp and amplitude is always assumed. The result is a a $Km_i \times Km_i$ matrix. The following order is used: The first indices correspond to the values/entries of the first coordinate, followed by values/entries of the second coordinate etc. Be aware of overparametrization, especially when using a dynamical covarince structure.

The warp covariances should be provided as functions `function(t, param)`. Do not use a scale parameter (unless the amplitude covariance is non-scalable), as a scale parameter is already a part of the model. Also warp covariances must have an attribute `param` that contains default parameter values. This requirement will be relaxed in the future.

Warp covariance may be null. In that case, all aspects in model that arise due to warp variation are set to zero.

Object returned by ppMulti ppMulti will return a list containing

- w: Estimated warp values
- `amp_cov_par`

  Estimated amplitude covariance parameters.
- `warp_cov_par`

  Estimated warp covariance parameters.
- like: Negative log-likelihood as estimated by the linearization.
- `sigma` Estimated σ in the model.
- c: Estimated spline coefficients. A matrix of dimension length(design)$*K\times$ df(spline). The estimated coefficients can be found using $(I_K \otimes \text{design}_i) * c$.

Dynamical covariance structure

Referring to Proposition 1 of (Olsen et al. 2016), dynamical covariance structure is a general framework for constructing covariance matrices using few parameters. It is given by the covariance function K constructed below:

Let f be a temporal covariance function, let $0 = s_1 < \cdots < s_\ell = 1$ be anchor points, let $A_1, \ldots, A_\ell \in \mathbb{R}^{q \times q}$ be a set of positive definite matrices. For each $t \in [0, 1]$ define $B_t \in \mathbb{R}^{q \times q}$ as the unique positive definite matrix satisfying

$$B_t^\top B_t = \frac{s_{k+1} - t}{s_{k+1} - s_k} A_k + \frac{t - s_k}{s_{k+1} - s_k} A_{k+1} \quad \text{for } t \in [s_k, s_{k+1}]. \tag{2}$$

For all $s, t \in [0, 1]$, define $K(s, t) = f(s, t) B_s^\top B_t \in \mathbb{R}^{q \times q}$.

Dynamical covariance structure is implemented in `simm.fda` using `kovMat`.

Timefunction concept A `timefunction` is a temporal covariance function on the domain [0,1]. `simm.fda` provides Matern covariance (including OU processes) and Brownian motion and bridge.

The user can provide her own time function. It has to be a function on the form `myfct(s,t, parameters)` $= \mathrm{Cov}(X(s), X(t))$ that can be vectorized by `outer`.

Other details `simm.fda` uses the unprotected R abbrevations T/F for TRUE/FALSE. Please do not redefine these variables in your code. This issue will be changed in the future.

Data Example

The package includes a small data example. Data consits of four sequences of foot trajectories in x/y/z-space. The sequences are four repetions of the same individual walking at a normal pace. It is part of much larger data set available from `http://mocap.cs.cmu.edu/`.

The variable `MCD.data` contains observation values, and `MCD.time` contains time points scaled to be within [0.1, 0.9]. A fairly simple model is fitted in the help page of `ppMulti`. To be honest I'm not satisfied with the data example as it is now. Since fitting the model will generally take a long time, the output is available from `MCD.res`.