ORIE 6070 Report

Estimation of the Structural Mean of Curves

Wenyu Zhang (wz258)

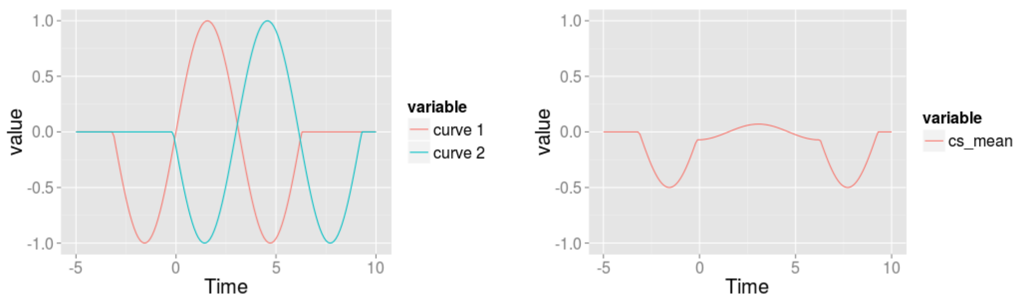
Abstract. Gervini and Gasser's paper [**2**] introduces curve samples as noisy realizations of a compound stochastic process $X(t) = (Z \circ W)(t)$, with amplitude variation $Z(t)$ and warping process $W(t)$. In many real-life applications where the 'mean' of the curves is wanted, the structural mean $\mu(t) = E[Z(t)]$ is more meaningful than the cross-sectional mean $E[X(t)]$. The paper proposes a MLE estimator for $\mu(t)$. Following simulations and real-data analysis to examine the claims in the paper, the proposed estimator is found to be robust to misspecification of the landmarks, and performs well on both simulated and real data.

1. Introduction

1.1. Background. This report is a study of the results of the paper *Non-parametric maximum likelihood estimation of the structural mean of a sample of curves* [**2**], published in 2005 on *Biometrika*, by Daniel Gervini and Theo Gasser.

The main research question the paper attempts to answer is:

- Given a sample of curves in the presence of time variability, how do we find the 'mean' of the curves?



(A) Raw curves (B) Cross-sectional mean

FIGURE 1. Sinusoidal curves with cross-sectional mean

This is a prevalent problem in many areas of research dealing with real-world time series or longitudinal data. Time variability is present when features of an underlying data-generating process are translated and/or stretched along the time-axis, due to differences across the subjects measured or across the instances the same subject is measured. Time variability cannot be easily controlled for in the experimental design. A frequently-used example of such a dataset is the Berkeley Growth Study data in the R {*fda*} package, containing the heights of boys and girls collected from age 1 to 18.

The 'mean' of curves can be considered in two ways:

- Cross-sectional mean,
- Structural mean.

The cross-sectional mean, although computationally simple, tends to underestimate the magnitude of peaks and troughs. Figure 1 shows an extreme example where the two sinusoidal curves have a phase difference, such that the cross-sectional mean is severely shrunk towards zero. Hence, care is needed to extract the structural mean instead.

1.2. Related Work. The structural mean of a sample of curves can be found by first aligning the curves, and then calculating the cross-sectional average. Common curve-alignment methods include:

- Landmark registration,
- Continuous monotone registration,
- Dynamic time warping.

However, these methods have various drawbacks. Landmark registration requires tedious labelling of landmarks on each curve, and the method fails if curves have missing landmarks. Continuous monotone registration aligns curves according to a target function. The method is dependent on the appropriateness of the target function defined and pre-smoothing of the data. Lastly, dynamic time warping entails calculating the distance of each pair of points between two curves, and hence is computationally intensive and memory and time consuming. [**3**]

1.3. Report Outline. In the following sections, the proposed MLE method will be introduced and studied. The theory is summarized in Section 2 and the computational aspects in Section 3. Further simulations and real data analysis are conducted and discussed in Sections 4 and 5. The code implementing the algorithm in R is included in the Appendix.

2. Review of Proposed MLE Method

2.1. Formulation. Suppose the dataset comprises of n curves, i.e $\{x_1, \ldots, x_n\}$, with $x_i \in \mathbb{R}^{m_i}$. The curves do not necessarily have the same number of observations and are not necessarily observed at the same time.

Assuming the data is generated from an underlying compound stochastic process $X(t) = (Z \circ W)(t)$ with Gaussian noise, we can formulate the model as follows:

$$\begin{aligned}
&x_{ij} = X_i(t_{ij}) + \epsilon_{ij} \ \ \forall i = 1, \dots, n, \ j = 1, \dots, m_i \\
&X_i : [a, b] \to \mathbb{R} \\
&\epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)
\end{aligned}$$

with amplitude variation as a truncation of the Karhunen-Loève decomposition:

$$Z(t) = \mu(t) + \sum_{k=1}^{q} \xi_k \phi_k(t)$$

$$\begin{aligned}
\mu, \{\phi_k\}_{k=1}^q \quad & \text{fixed, unknown functions} \\
\{\phi_k\}_{k=1}^q \quad & \text{pairwise orthogonal in } \mathcal{L}^2\left([a, b]\right) \\
\{\xi_k\}_{k=1}^q \quad & \text{pairwise uncorrelated with } E[\xi_k] = 0, \ var[\xi_k] < \infty
\end{aligned}$$

and warping process:

$$\begin{aligned}
&W(t) = g(t, \theta) \\
&\quad \theta \in \mathbb{R}^p \quad \text{unobservable} \\
&f(\theta) \propto \prod_{k=1}^{p} \frac{1}{\tau_k} \varphi\left(\frac{\theta_k - \theta_{0k}}{\tau_k}\right) \mathbb{1}\{a < \theta_1 < \ldots < \theta_p < b\} \\
&\quad g \quad \text{fixed, known, monotone increasing function in } t \\
&\qquad \text{where} \quad g(a, \theta) = a, g(b, \theta) = b, g(\theta_k, \theta) = \theta_{0k} \ \ \forall k = 1, \dots, p
\end{aligned}$$

θ can be viewed as the set of 'hidden landmarks'. The paper suggests estimating the parameters $\{\theta_{0k}\}_{k=1}^p$ and $\{\tau_k\}_{k=1}^p$ by picking out possible values by visually inspecting the data, and finalizing on the ones that evaluates to the highest likelihood. The paper uses the cubic interpolating polynomial for

g. In R, piecewise cubic Hermite interpolating polynomials in the $\{pracma\}$ package is used for implementation of the algorithm in Section 4 and 5.

The above formulation translates to the general variance-component model:

$$x_{ij} = \mu(g(t_{ij}, \theta_i)) + \sum_{k=1}^{q} \xi_k \phi_k(g(t_{ij}, \theta_i)) + \epsilon_{ij} \tag{2.1}$$

However, simultaneous MLE estimation of μ and $\{\phi_k\}_{k=1}^{q}$ is complicated. Assuming that Z has no variance components, the simpler mean-plus-error model is obtained:

$$x_{ij} = \mu(g(t_{ij}, \theta_i)) + \epsilon_{ij} \tag{2.2}$$

$$x_i|(\theta = \theta_i) \overset{iid}{\sim} N\left(\mu(g(t_i^*, \theta_i)), \sigma^2 I_{m_i}\right), \quad t_i^* = (t_{i1}, \ldots, t_{im_i})^T \tag{2.3}$$

2.2. Estimators. Under the mean-plus-error model, and further enforcing the continuity condition $\mu \in \mathcal{C}\left([a, b]\right)$, the paper attains the estimators:

$$\hat{\sigma}^2 = \left(\sum_{i=1}^{n} m_i\right)^{-1} \sum_{i=1}^{n} \int \|x_i - \hat{\mu}(g(t_i^*, \theta))\|^2 f(\theta|x_i; \hat{\mu}, \hat{\sigma}^2)\mathrm{d}\theta \tag{2.4}$$

$$\hat{\mu}(s) = \frac{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m_i} x_{ij} w_{ij}(s|x_i; \hat{\mu}, \hat{\sigma}^2)}{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m_i} w_{ij}(s|x_i; \hat{\mu}, \hat{\sigma}^2)} \quad \text{a.e. in } [a, b] \tag{2.5}$$

where $w_{ij}(s|x_i; \mu, \sigma^2)$ is the density of $g(t_{ij}, \theta)|x_i$.

The weighted average intuition for $\hat{\mu}$ can be extended to derive estimates of the aligned or registered curves:

$$\hat{Z}_i(t) = \frac{\sum\limits_{j=1}^{m_i} x_{ij} w_{ij}(s|x_i; \hat{\mu}, \hat{\sigma}^2)}{\sum\limits_{j=1}^{m_i} w_{ij}(s|x_i; \hat{\mu}, \hat{\sigma}^2)}$$

2.3. Derivation. The following outlines the theoretical derivation of the estimators presented in Section 2.2.

Under the mean-plus-error model,

$$x_i|(\theta = \theta_i) \overset{iid}{\sim} N\left(\mu(g(t_i^*, \theta_i)), \sigma^2 I_{m_i}\right), \quad t_i^* = (t_{i1}, \ldots, t_{im_i})^T$$

the log-likelihood function can be obtained as:

$$L(\mu, \sigma^2) = \sum_{i=1}^{n} \log \int f(x_i|\theta; \mu, \sigma^2) f(\theta) \mathrm{d}\theta \tag{2.6}$$

Since

$$\frac{\partial f(x_i|\theta_i; \mu, \sigma^2)}{\partial(\sigma^2)} = \left[-\frac{m_i}{2\sigma^2} + \frac{\|x_i - \mu\left(g\left(t_i^*, \theta_i\right)\right)\|^2}{2(\sigma^2)^2}\right] f(x_i|\theta_i; \mu, \sigma^2) \tag{2.7}$$

plugging Equation 2.7 into the log-likelihood function 2.6 and setting $\frac{\partial L(\mu,\sigma^2)}{\partial(\sigma^2)} = 0$, the estimating equation 2.4 for σ^2 can be obtained after straightforward algebra:

$$\hat{\sigma}^2 = \left(\sum_{i=1}^{n} m_i\right)^{-1} \sum_{i=1}^{n} \int \|x_i - \hat{\mu}(g(t_i^*, \theta))\|^2 f(\theta|x_i; \hat{\mu}, \hat{\sigma}^2) \mathrm{d}\theta$$

Due to the complexity of the expression, the estimating equation for μ cannot be simply attained by setting $\frac{\partial L(\mu,\sigma^2)}{\partial\mu} = 0$. Suppose $\mu \in \mathcal{M}$, the directional log-likelihood function:

$$L(\mu + th, \sigma^2), \quad h \in \mathcal{M}$$

needs to be employed.

Since $\{\hat{\mu}, \hat{\sigma}^2\}$ maximizes the log-likelihood function $L(\mu, \sigma^2)$, then it maximizes the directional log-likelihood function $L(\mu + th, \sigma^2)$ at $t = 0$ for every $h \in \mathcal{M}$. That is,

$$\frac{\partial}{\partial t} L(\hat{\mu} + th, \hat{\sigma}^2)\Big|_{t=0} = 0 \quad \forall h \in \mathcal{M}$$

where

$$\begin{aligned}
&\frac{\partial}{\partial t} L(\hat{\mu} + th, \hat{\sigma}^2)\Big|_{t=0} \\
&= \frac{1}{\hat{\sigma}^2} \sum_{i=1}^{n} \int \left[x_i - \hat{\mu}(g(t_i^*, \theta))\right]^T h(g(t_i^*, \theta)) f(\theta | x_i; \hat{\mu}, \hat{\sigma}^2) \mathrm{d}\theta \\
&= \frac{1}{\hat{\sigma}^2} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \int \{x_{ij} - \hat{\mu}(s)\} w_{ij}(s | x_i; \hat{\mu}, \hat{\sigma}^2) h(s) \mathrm{d}s
\end{aligned} \tag{2.8}$$

The estimating equations for μ (2.8) and σ^2 (2.4) are unbiased, which implies the existence of consistent and asymptotically unbiased estimators for μ and σ^2. However, this observation has limited usefulness since Equation 2.8 does not provide an explicit expression for μ that can be easily computed. By further restricting $\mathcal{M} = \mathcal{C}([a, b])$, the closed-form expression 2.5 can be obtained:

$$\hat{\mu}(s) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m_i} x_{ij} w_{ij}(s | x_i; \hat{\mu}, \hat{\sigma}^2)}{\sum_{i=1}^{n} \sum_{j=1}^{m_i} w_{ij}(s | x_i; \hat{\mu}, \hat{\sigma}^2)} \quad \text{a.e. in } [a, b]$$

3. Computational Considerations

3.1. Approximations. To evaluate the expressions in Section 2.2, Monte Carlo approximations are used:

$$\begin{aligned}
\hat{f}^{(N)}(x_i; \hat{\mu}, \hat{\sigma}^2) &= \frac{1}{N} \sum_{l=1}^{N} f\left(x_i | \theta^{(l)}; \hat{\mu}, \hat{\sigma}^2\right) \\
\hat{w}_{ij}^{(N,\lambda)}(s | x_i; \hat{\mu}, \hat{\sigma}^2) &= \frac{1}{N} \sum_{l=1}^{N} \frac{1}{\lambda} K\left(\frac{g\left(t_{ij}, \theta^{(l)}\right) - s}{\lambda}\right) \frac{f\left(x_i | \theta^{(l)}; \hat{\mu}, \hat{\sigma}^2\right)}{\hat{f}^{(N)}(x_i; \hat{\mu}, \hat{\sigma}^2)}
\end{aligned}$$

with the Epanechinikov kernel K and the average oversmoothing bandwidth λ suggested by Wand and Jones in 1995:

$$
\begin{aligned}
K(t) &= 0.75(1-t^2)\mathbb{1}\{|t| \leq 1\} \\
\lambda &= \left(\frac{243c_1}{35c_2^2 N}\right)^{\frac{1}{5}} \left(\sum_{i=1}^{n} m_i\right)^{-1} \left(\sum_{i=1}^{n}\sum_{j=1}^{m_i} s_{ij}\right) \\
c_1 &= \int K^2(t)\mathrm{d}t = \frac{3}{5} \\
c_2 &= \int t^2 K(t)\mathrm{d}t = \frac{1}{5} \\
\bar{g}^{(N)} &= \frac{1}{N}\sum_{l=1}^{N} g\left(t_{ij}, \theta^{(l)}\right) \\
s_{ij} &= \sqrt{\frac{1}{N-1}\sum_{l=1}^{N}\left(g\left(t_{ij}, \theta^{(l)}\right) - \bar{g}^{(N)}\right)^2}
\end{aligned}
$$

3.2. Algorithm. The algorithm is summarized as follows.

Algorithm 1: MLE for Structural Mean

Input: $\{x_i\}$, $\{t_i^*\}$, $\{\theta_{0k}\}$, $\{\tau_k\}$
Output: $\hat{\sigma}^2$, $\hat{\mu}$, $\{\hat{Z}_i\}$
initialization of $\hat{\sigma}_0^2$, $\hat{\mu}_0$
draw samples $\{\theta^{(l)}\}$ from $f(\theta)$
calculate λ, $\{g\left(t_{ij}, \theta^{(l)}\right)\}$
repeat
  calculate $\{\hat{\mu}\left(g\left(t_i^*, \theta^{(l)}\right)\right)\}$, $\{f\left(x_i|\theta^{(l)}; \hat{\mu}, \hat{\sigma}^2\right)\}$, $\{\hat{f}^{(N)}(x_i; \hat{\mu}, \hat{\sigma}^2)\}$, $\{\hat{w}_{ij}(s|x_i; \hat{\mu}, \hat{\sigma}^2)\}$
  update $\hat{\sigma}^2$, $\hat{\mu}$
until *Convergence*;
calculate $\{\hat{Z}_i\}$, log-likelihood $L(\hat{\mu}, \hat{\sigma}^2)$

The initialization of $\hat{\sigma}^2$, and $\hat{\mu}$ are:

$$
\begin{aligned}
\hat{\sigma}_0^2 &= \left(\sum_{i=1}^{n} m_i\right)^{-1} \sum_{i=1}^{n}\sum_{j=1}^{m_i}(x_{ij} - \bar{x})^2 \\
\hat{\mu}_0(t) &= \frac{1}{n}\sum_{i=1}^{n}\tilde{x}_i(t)
\end{aligned}
$$

where $\tilde{x}_i(t)$ is obtained from $x_{i1}, \ldots, x_{im_i}$ by piecewise cubic interpolation.

R code for the algorithm can be found in the Appendix.

4. Simulations

4.1. Robustness to Model Misspecification. This simulation tests the robustness of the algorithm when data is generated from the general variance-component model in Equation 2.1 instead of the mean-plus-error model in Equation 2.2.

A sinusoidal structural mean is defined $\mu(t) = \sin(t)\mathbb{1}\{-\pi \leq t \leq 2\pi\}$. For simplicity, to simulate the unaligned curves, only lateral shifts from the structural mean are considered.

Following the guidelines in the paper, raw curves are generated under the two models such that they have the same overall amplitude variance $E[\sum_{j=1}^{m_i} Z_i^2(t_{ij})]$. The specifications used under the general variance-component model are:

$$q = 1$$

$$\phi_1(t) = \text{third function in cubic B-spline basis in } [-\pi, 2\pi] \text{ with knots } \{-1, 0, 1\},$$

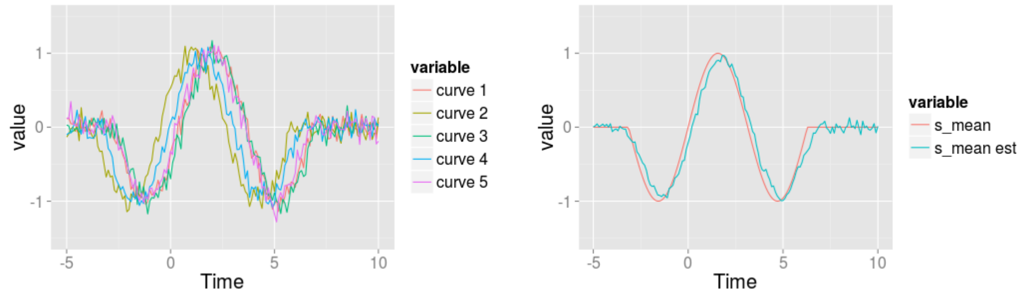
$$\text{standardized to } \int \phi_1^2(t)\mathrm{d}t = 1$$

$$\xi_1 \sim N(0, 0.75 \times 0.20^2)$$

$$\epsilon_{ij} \sim N(0, 0.25 \times 0.20^2)$$

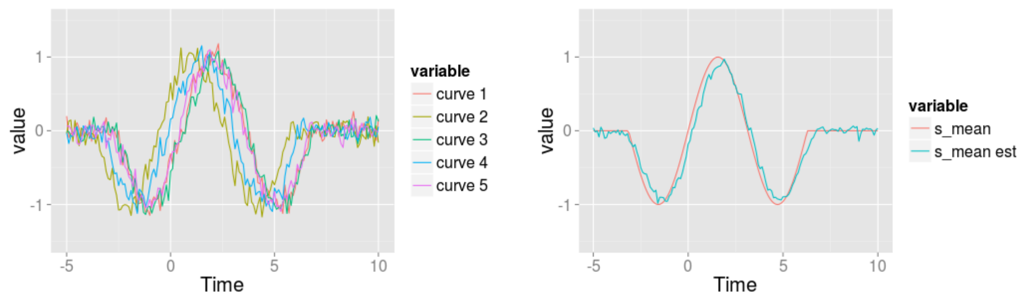
and the specifications used under the mean-plus-error model are:

$$\epsilon_{ij} \sim N(0, 0.20^2)$$



(A) Raw curves (B) Structural mean

FIGURE 2. Sinusoidal curves under general variance-component model



(A) Raw curves (B) Structural mean

FIGURE 3. Sinusoidal curves under mean-plus-error model

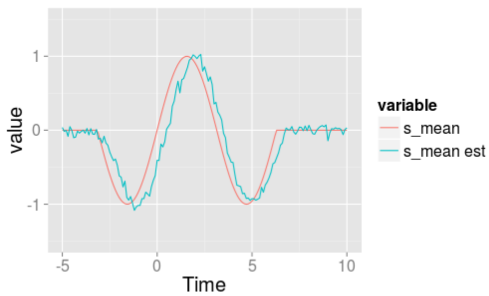
The parameters for the warping process are defined as:

$$\theta_0 = \left(-\frac{\pi}{2}, \frac{\pi}{2}, \frac{3\pi}{2}\right)$$

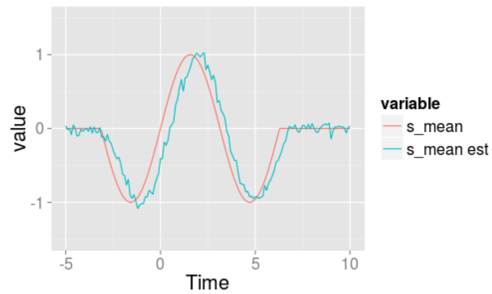
$$\tau = (0.05, 0.05, 0.05)$$

From plot (B) in Figure 2, the algorithm estimates the structural mean of data under the general variance-component model quite well, despite some lack of smoothness especially at the end of the interval.

The robustness may be due to the similarity of the raw data as plotted in Figures 2 and 3, such that the variance components can be effectively absorbed into the estimate for σ^2.



(A) Misspecified landmark location



(B) Overstatement of number of landmarks

FIGURE 4. Structural mean estimates with misspecified landmarks

4.2. Robustness to misspecification of landmarks. The paper demonstrated through simulation results that the algorithm is robust to an understatement of the number of landmarks. The following simulations further explores the cases:

- Locations of landmarks are misspecified,
- Number of landmarks is overstated.

The data are generated from the mean-plus-error model as in Section 4.1. The following parameters for the warping process are used, for the misspecification of landmark locations:

$$\theta_0 = (-3, 0, 3)$$
$$\tau = (0.05, 0.05, 0.05)$$

and for the overstatement of landmark count:

$$\theta_0 = \left(-\frac{\pi}{2}, 0, \frac{\pi}{2}, \frac{3\pi}{2}\right)$$
$$\tau = (0.05, 0.05, 0.05, 0.05)$$

As seen from Figure 4, the estimated structural means closely resemble the shape of the true structural mean even in the event of landmark misspecification. However, a slight lateral shift is observed as a result of the misspecification.

A possible way to correct for the shift is to incorporate the uncertainty of the landmarks into the model specification. One modification is to use larger values for $\{\tau_k\}_{k=1}^p$, the standard deviations of $\{\theta_{0k}\}_{k=1}^p$.

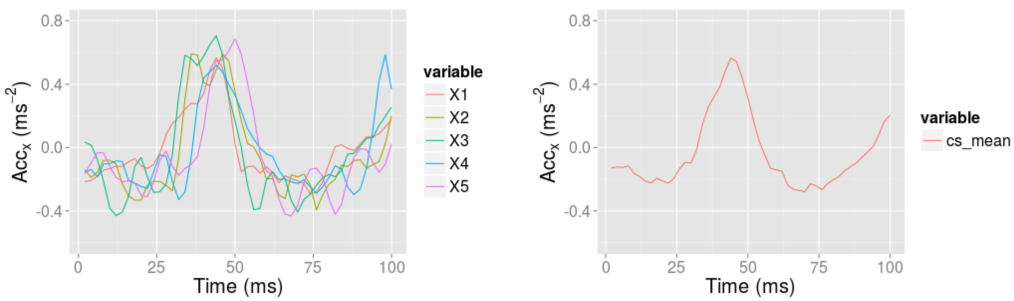
5. Application

The practicality of the proposed method is evaluated by an implementation on real data. The *Human Activity Recognition Using Smartphones Data Set* from the UCI repository is used for this purpose. The data set contains over 10000 samples of motion sensor readings, i.e. tri-axial accelerometer and gyroscope readings, of 30 subjects performing six activities: walking, walking upstairs, walking downstairs, standing, sitting, and lying. One reading is taken per $2ms$.

For demonstration purposes, the algorithm is run on only the first five samples of the first subject doing the 'walking downstairs' activity, with dimensionality restricted to acceleration in the x-axis and the length of the time-series set at $100ms$.

Figure 5 shows the raw curves and the cross-sectional mean. The cross-sectional mean is evidently sub-optimal as it is not very smooth, and underestimates the magnitude of the peak and troughs.

Since the paper already compared results from the proposed MLE method with those of landmark registration and continuous monotone registration, a comparison is conducted with dynamic time warping (DTW).



(A) Raw curves (B) Cross-sectional mean

FIGURE 5. Walking-downstairs acceleration in x-axis

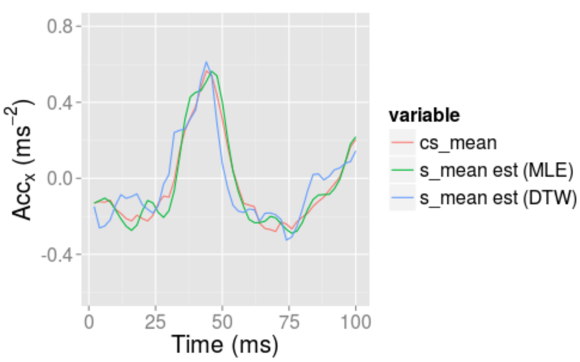


FIGURE 6. Mean Estimations

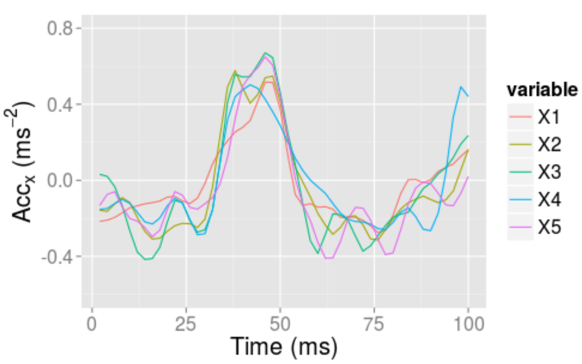


FIGURE 7. Registered Curves

Figure 6 plots the cross-sectional mean, along with the structural mean estimated using the proposed MLE method and DTW. The MLE estimate captures the width of the peak well, and reflects the presence of the two peaks as observed in the raw data.

Figure 7 plots the curves aligned using the proposed MLE method. The method seems to perform well based on visual inspection.

6. Conclusion

From the above analysis, the proposed MLE method can be concluded to be competitive with respect to existing methods. It does not require pre-smoothing as in continuous monotone registration. Although the parameters $\{\theta_{0k}\}_{k=1}^{p}$, $\{\tau_k\}_{k=1}^{p}$ need to specified by visual inspection of the data, this is much less labor intensive than the process of labelling the curves manually in landmark registration. Computation-wise, the MLE is less memory and computation intensive than DTW. However, this is dependent on several specifications of the algorithm, such as the number of Monte Carlo samples drawn to approximate the integrals, and the tolerance levels that determine the convergence of the algorithm. In the simulations and real-data application conducted, with tolerance set to 0.01, convergence is reached in three iterations.

From simulations in Section 4, the MLE method is robust to misspecification of the landmarks, which enables it practical for applications. An implementation of the algorithm on real data in Section 5 demonstrates this practicality.

However, plots of the estimated structural means are not smooth. Further work is required in determining the appropriated bandwidth parameter λ to enforce smoothness of the resulting estimate. One possible method is to enumerate possible values of λ in a grid search.

Appendix

```
require(tmvtnorm)
require(pracma)
require(mvtnorm)
require(pryr)

########################
# Defining functions #
########################

# Epanechnikov kernel
K = function(t){
  res = 0.75*(1-t^2)*(abs(t)<=1)
  return(res)
}

#############################
# Solving estimating eqns #
#############################

# evaluating expressions independent of mu and sigma2
intialize = function(data, t_index, theta0, tau, N=1000){
  n = dim(data)[1]
  m = dim(data)[2]
  p = length(theta0)
```

```
# initialization values for mu and sigma2
x_spline = apply(data, 1, splinefun, x=t_index) # length n
mu_0 = function(t){ # mu_0 to be evaluated at t (vector)
  x_t = matrix(nrow=n, ncol=length(t))
  for (i in 1:n){
    x_t[i,] = x_spline[[i]](t)
  }
  return(colMeans(x_t))
}
sigma2_0 = sum((data-mean(data))^2)/(n*m)

# return N random sample of theta
theta_samples = matrix(NA, N, p)
i = 0
while (i<N){
  # N-by-p
  samp = rtmvnorm(n=1, mean=theta0, sigma=diag(tau^2,p,p),
                  lower=rep(min(t_index),p),
                  upper=rep(max(t_index),p),
                  algorithm='rejection')
  samp_order = sum((order(samp)==1:p))
  if ((samp_order==p) &
      (min(samp)>min(t_index)) & (max(samp)<max(t_index))){
    i = i+1
```

```
      theta_samples[i,] = samp
    }
  }

  # g(t_ij, theta_l)
  # independent of i for standardized sampling time
  g_mat = t(apply(cbind(min(t_index),theta_samples,max(t_index)),
                  1, pchip,
                  yi=c(min(t_index),theta0,max(t_index)),
                  x=t_index)) # N-by-m
  # s_ij, independent of i for standardized sampling time
  s_vec = apply(g_mat, 2, sd)
  # lambda for kernel
  lambda = (((243*3/5)/(35*N/25))^0.2)*mean(s_vec)
  return(list(mu_0=mu_0, sigma2_0=sigma2_0,
              theta_samples=theta_samples, g_mat=g_mat,
              lambda=lambda))
}

# evaluating mu and sigma2 till convergence
eval = function(data, t_index,
                init_res, tol=10^-4, max_itr = 1000){
  mu_0 = init_res$mu_0
  sigma2_0 = init_res$sigma2_0
  theta_samples = init_res$theta_samples
```

```
g_mat = init_res$g_mat
lambda = init_res$lambda

tol_mu = rep(NA, max_itr)
tol_sigma2 = rep(NA, max_itr)
sigma2_track = rep(NA, max_itr)

n = dim(data)[1]
m = dim(data)[2]
N = dim(theta_samples)[1]
p = length(theta0)

itr = 0
# distance evaluated as L2 norm at t_index
mu_old = splinefun(x=t_index, y=t_index)
sigma2_old = 100
mu_eval_old = sapply(t_index, mu_old)
if (itr==0){
  mu_new = mu_0
  sigma2_new = sigma2_0
  mu_eval_new = sapply(t_index, mu_new)
}

while ((itr<max_itr) &
         ((abs(sigma2_new-sigma2_old)>tol) |
```

```
        (sqrt(sum((mu_eval_new-mu_eval_old)^2))>tol))){
itr = itr+1
tol_mu[itr] = sqrt(sum((mu_eval_new-mu_eval_old)^2))
tol_sigma2[itr] = abs(sigma2_new-sigma2_old)
sigma2_track[itr] = sigma2_new
print(itr)
mu_old = mu_new
sigma2_old = sigma2_new
mu_eval_old = mu_eval_new
# mu(g(t_i,theta_l))
# independent of i for standardized sampling time
mu_g = matrix(NA, N, m) # N-by-m
for (i in 1:N){
  mu_g[i,] = sapply(g_mat[i,], mu_old)
}
# f(x_i|theta_l)
fx_theta = matrix(NA, N, n) # N-by-n
for (i in 1:n){
  fx_theta[,i] = apply(mu_g, 1, dmvnorm, x=data[i,],
                       sigma=diag(sigma2_old,m,m))
}
# est f(x_i)
fx = colMeans(fx_theta) # length n
# est w_ij(s|x_i)
fg_x = function(s, i, j){
```

```
  summand = 0
  for (l in 1:N){
    summand = summand +
      K((g_mat[l,j]-s)/lambda)*fx_theta[l,i]/fx[i]
  }
  return(summand/(N*lambda))
}

# calculate mu_new
mu_new = function(a){
  w_mat=matrix(NA, n, m)
  for (i in 1:n){
    for (j in 1:m){
      w_mat[i,j] = fg_x(a, i, j)
    }
  }
  frac = sum(data*w_mat)/sum(w_mat)
  return(frac)
}
mu_eval_new = sapply(t_index, mu_new)

# calculate sigma2_new
sum_inner = matrix(NA, N, n)
for (l in 1:N){
  for (i in 1:n){
```

```
      sum_inner[l,i] = sum((data[i,]-mu_g[l,])^2)*
        fx_theta[l,i]/fx[i]
    }
  }
  sigma2_new = sum(colMeans(sum_inner))/(n*m)
}

# registered curves Z(t)
z = function(a){
  w_mat=matrix(NA, n, m)
  for (i in 1:n){
    for (j in 1:m){
      w_mat[i,j] = fg_x(a, i, j)
    }
  }
  frac = rowSums(data*w_mat)/rowSums(w_mat)
  return(frac)
}
z = sapply(t_index, z)

# log-likelihood
loglike = sum(log(fx))

return(list(mu=mu_new, mu_vec = mu_eval_new, sigma2=sigma2_new,
            loglike=loglike,
```

```
                    tol_mu = tol_mu, tol_sigma2 = tol_sigma2,
                    sigma2_track = sigma2_track,
                    z = z))
}
```

References

[1] J.-F. Dupuy, J.-M. Loubes, and E. Maza, *Non parametric estimation of the structural expectation of a stochastic increasing function*, Statistics and Computing **21** (2011), no. 1, 121–136 (English).

[2] Daniel Gervini and Theo Gasser, *Nonparametric maximum likelihood estimation of the structural mean of a sample of curves*, Biometrika **92** (2005), no. 4, 801–820.

[3] Isabella Morlini and Sergio Zani, *Estimation of the structural mean of a sample of curves by dynamic time warping*, Data Analysis, Classification and the Forward Search (Sergio Zani, Andrea Cerioli, Marco Riani, and Maurizio Vichi, eds.), Studies in Classification, Data Analysis, and Knowledge Organization, Springer Berlin Heidelberg, 2006, pp. 39–48 (English).

[4] Kongming Wang and Theo Gasser, *Synchronizing sample curves nonparametrically*, Ann. Statist. **27** (1999), no. 2, 439–460.