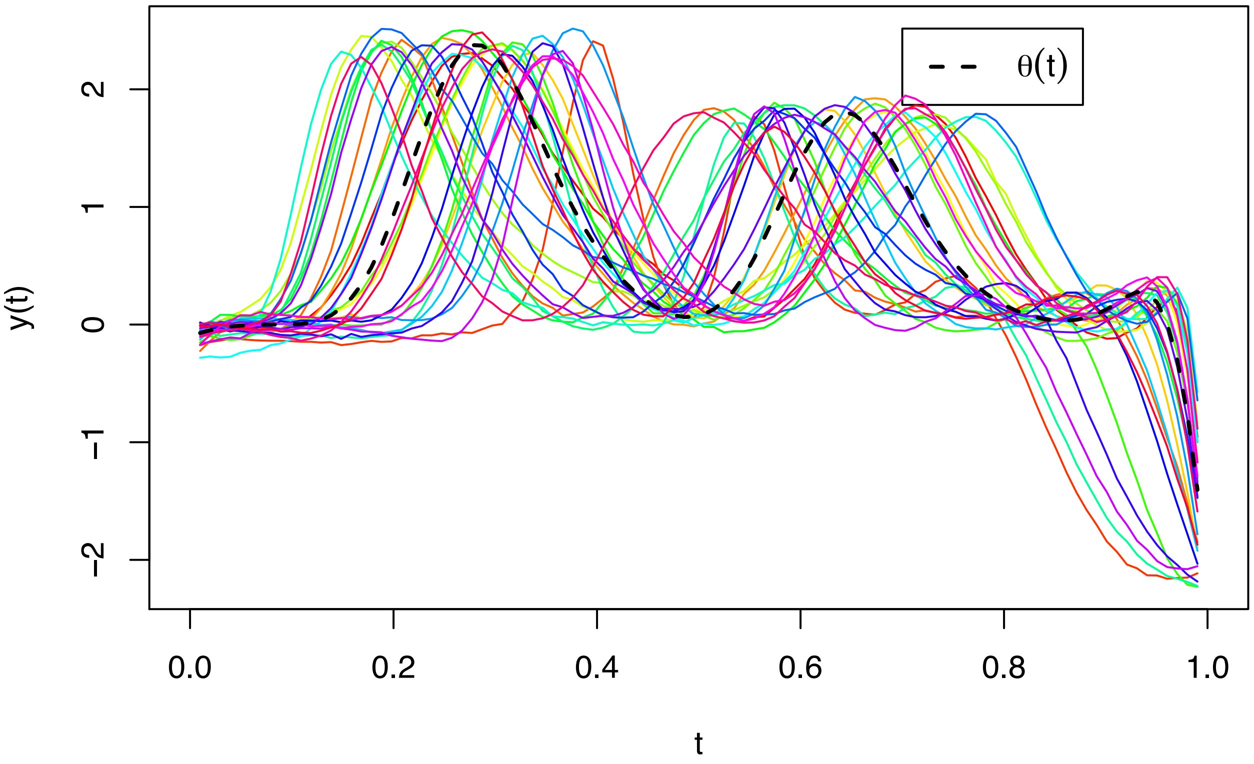
pavpop models with amplitude variation modeled by a functional basis

Lars Lau Raket

2016-01-08

Example 1

```r
# Number of samples
n <- 30
# Number of observation points
m <- 100

# Observation points
t <- seq(0, 1, length = m + 2)[2:(m + 1)]

# Common basis function (both mean and amplitude variation)
kts <- seq(0, 1, length = 12)[2:11]
basis_fct <- amp_fct <- make_basis_fct(kts = kts, intercept = TRUE)
df <- attr(basis_fct, 'df')

# Generate true mean weights
beta_t <- rexp(df, 0.5) * sample(-1:1, df, replace = TRUE, prob = c(0.2, 0.6, 0.2))

# Generate random variation weights
b_t <- replicate(n, rnorm(df, sd = 0.1))

# Generate warping function and random parameters
tw <- seq(0, 1, length = 4)
warp_fct <- make_warp_fct(type = 'smooth', tw = tw)
w_t <- replicate(n, rnorm(2, sd = 0.1))

# Generate data
sigma <- 0.01
y <- lapply(1:n, function(i) {as.numeric(basis_fct(warp_fct(w_t[, i], t)) %*% beta_t
                                         + amp_fct(t) %*% b_t[, i]
                                         + rnorm(m, sd = sigma))})
t <- lapply(1:n, function(x) t)

# Plot observations
plot(0, 0, xlim = c(0, 1), ylim = range(y), type = 'n',
     xlab = 't', ylab = 'y(t)')
legend(0.7, range(y)[2], legend = expression(theta(t)), lty = 2, lwd = 2)
for (i in 1:n) lines(t[[i]], y[[i]], col = rainbow(n)[i])
lines(t[[1]], basis_fct(t[[1]]) %*% beta_t , lwd = 2, lty = 2)
```

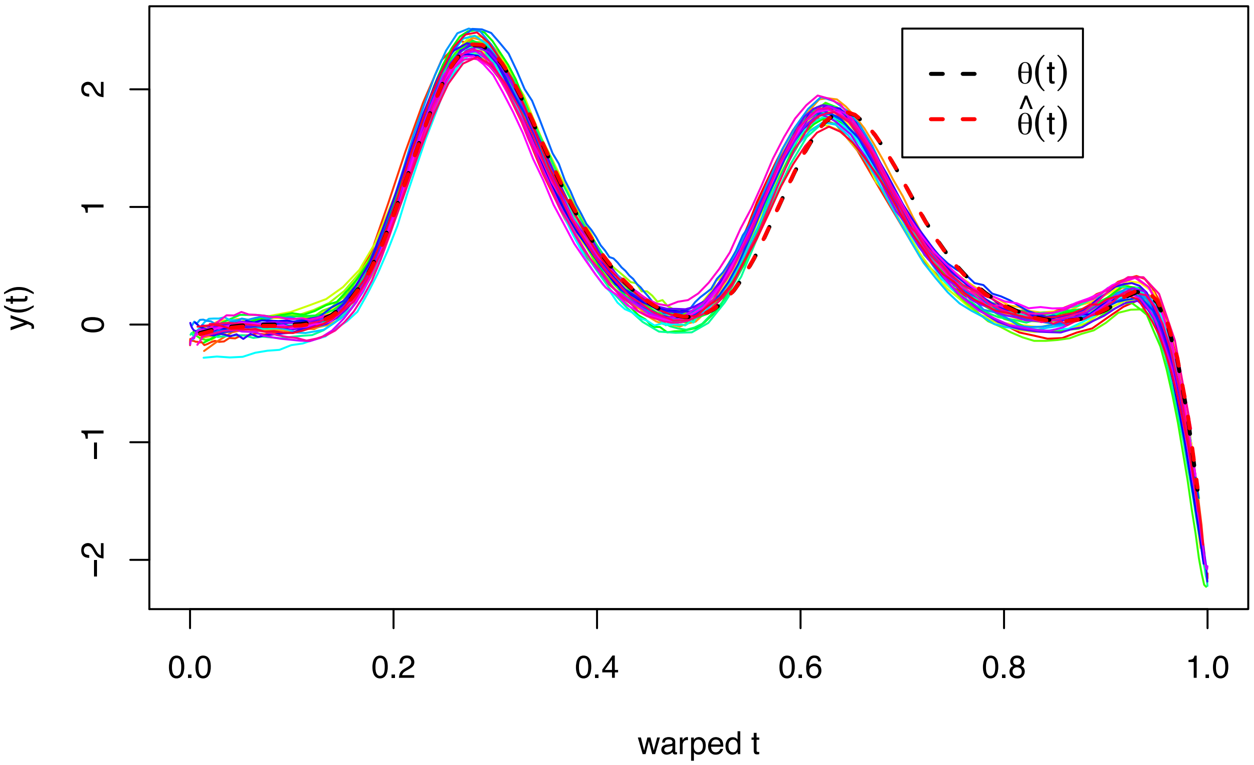


Now we set up pavpop to estimate in the model

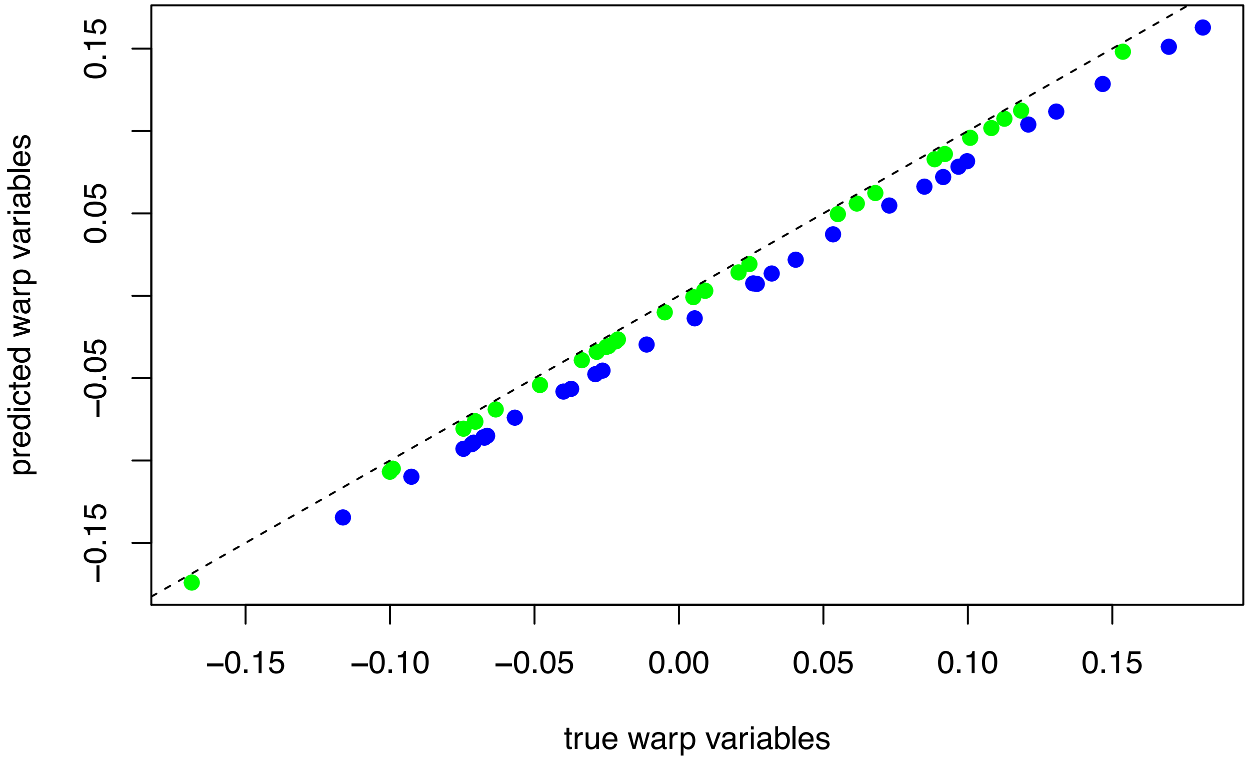
```
amp_cov <- make_cov_fct(id_cov, noise = FALSE)
warp_cov <- make_cov_fct(id_cov, noise = FALSE)

res <- pavpop(y, t, basis_fct, warp_fct, amp_cov, warp_cov, amp_fct, iter = c(10, 10))
```

```
#> Outer    :   Inner   :   Estimates
#> 1    :   1   2   3   4   5   6   7   8   9   10  :    104.9354 68.114
#> Linearized likelihood:    -24873.56
#> 2    :   1   :    104.7971 67.24256
#> Linearized likelihood:    -24930.24
#> 3    :   1   2   3   4   5   6   7   8   9   10  :    104.7483 69.44106
#> Linearized likelihood:    -24943.86
#> 4    :   1   :    104.6139 67.47894
#> Linearized likelihood:    -24945.77
#> 5    :   1   2   3   4   5   6   7   8   9   10  .
#> Likelihood not improved, returning best likelihood estimates.
```

```
# Plot aligned samples
plot(0, 0, xlim = c(0, 1), ylim = range(y), type = 'n',
     xlab = 'warped t', ylab = 'y(t)')
legend(0.7, range(y)[2], legend = c(expression(theta(t)), expression(hat(theta)(t))),
       lty = 2, lwd = 2, col = c('black', 'red'))
for (i in 1:n) lines(warp_fct(res$w[, i], t[[i]]), y[[i]], col = rainbow(n)[i])
lines(t[[1]], basis_fct(t[[1]]) %*% beta_t , lwd = 2, lty = 2)
lines(t[[1]], basis_fct(t[[1]]) %*% res$c, lwd = 2, lty = 2, col = 'red')
```

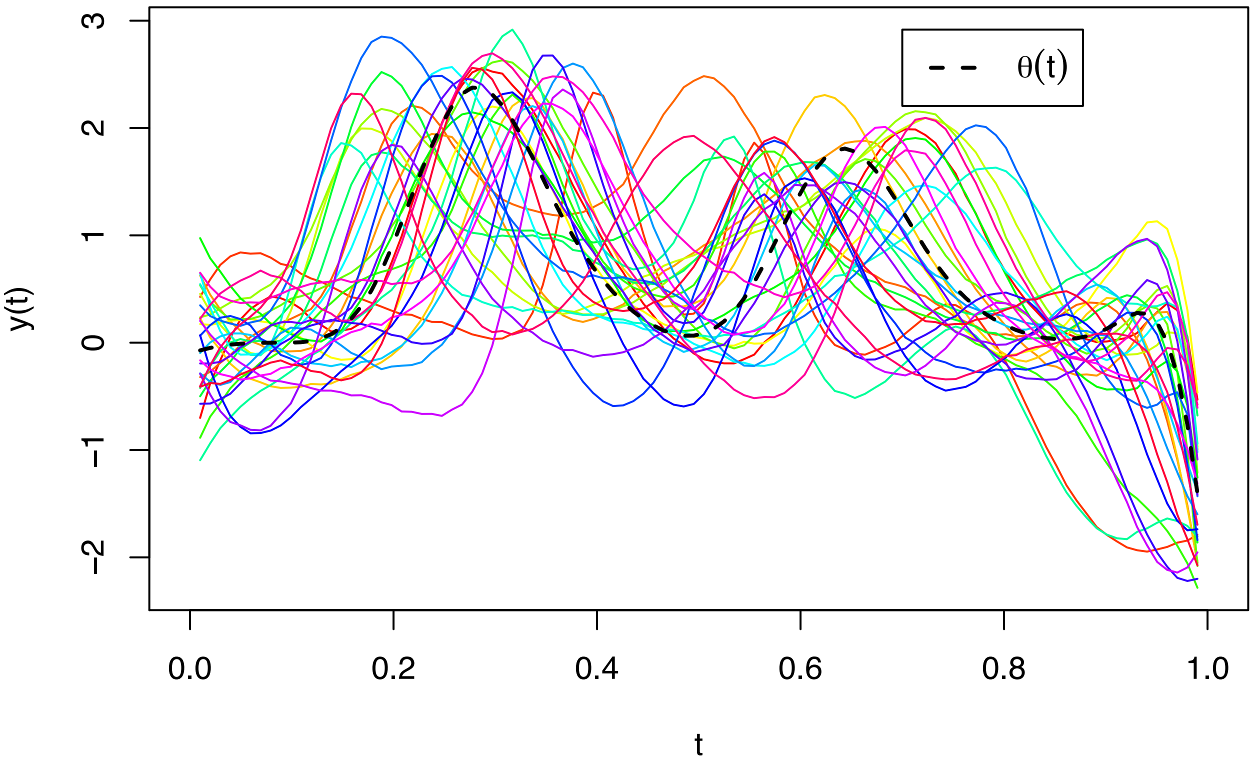


```r
plot(as.numeric(w_t), as.numeric(res$w), xlab = 'true warp variables',
     ylab = 'predicted warp variables', pch = 19, col = c('green', 'blue'))
abline(0, 1, lty = 2)
```



Example 2: more amplitude variation

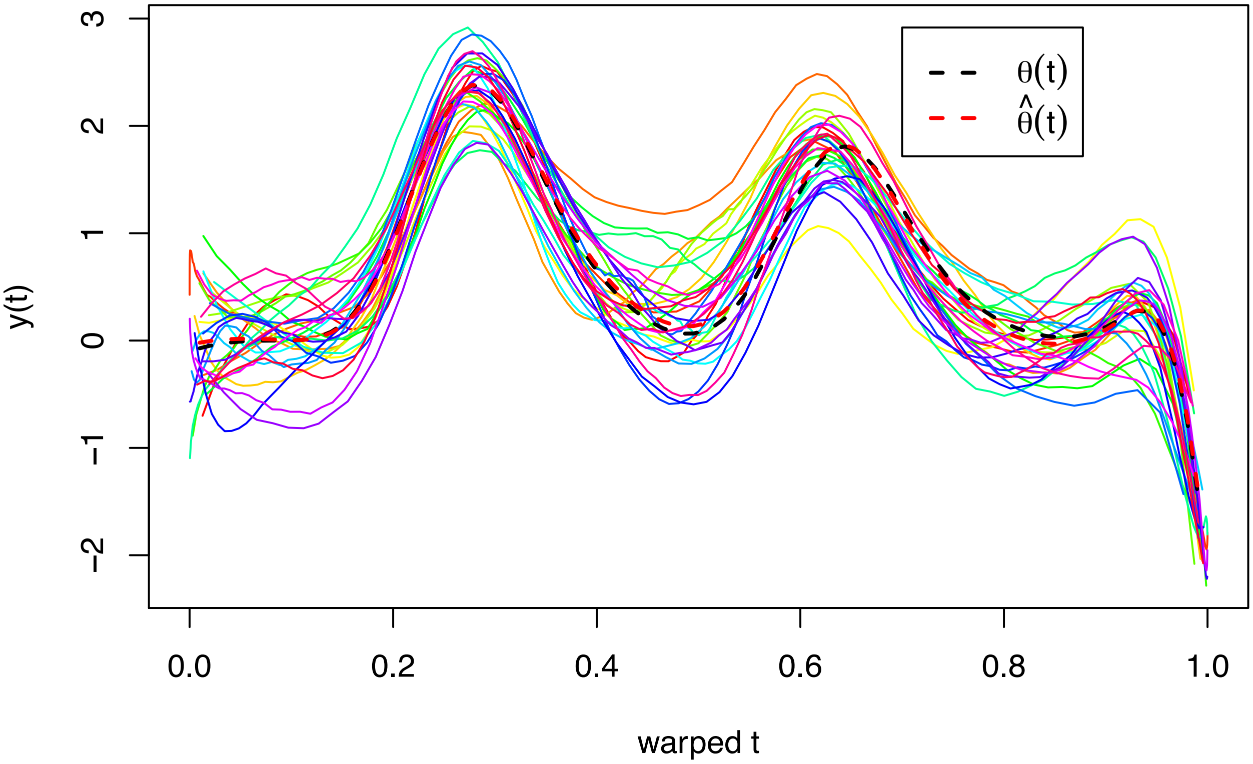
```
# Generate new random variation weights
b_t <- replicate(n, rnorm(df, sd = 0.5))

# Generate data
y <- lapply(1:n, function(i) {as.numeric(basis_fct(warp_fct(w_t[, i], t[[i]])) %*% beta_t
                                         + amp_fct(t[[i]]) %*% b_t[, i]
                                         + rnorm(m, sd = sigma))})
# Plot observations
plot(0, 0, xlim = c(0, 1), ylim = range(y), type = 'n',
     xlab = 't', ylab = 'y(t)')
legend(0.7, range(y)[2], legend = expression(theta(t)), lty = 2, lwd = 2)
for (i in 1:n) lines(t[[i]], y[[i]], col = rainbow(n)[i])
lines(t[[1]], basis_fct(t[[1]]) %*% beta_t , lwd = 2, lty = 2)
```



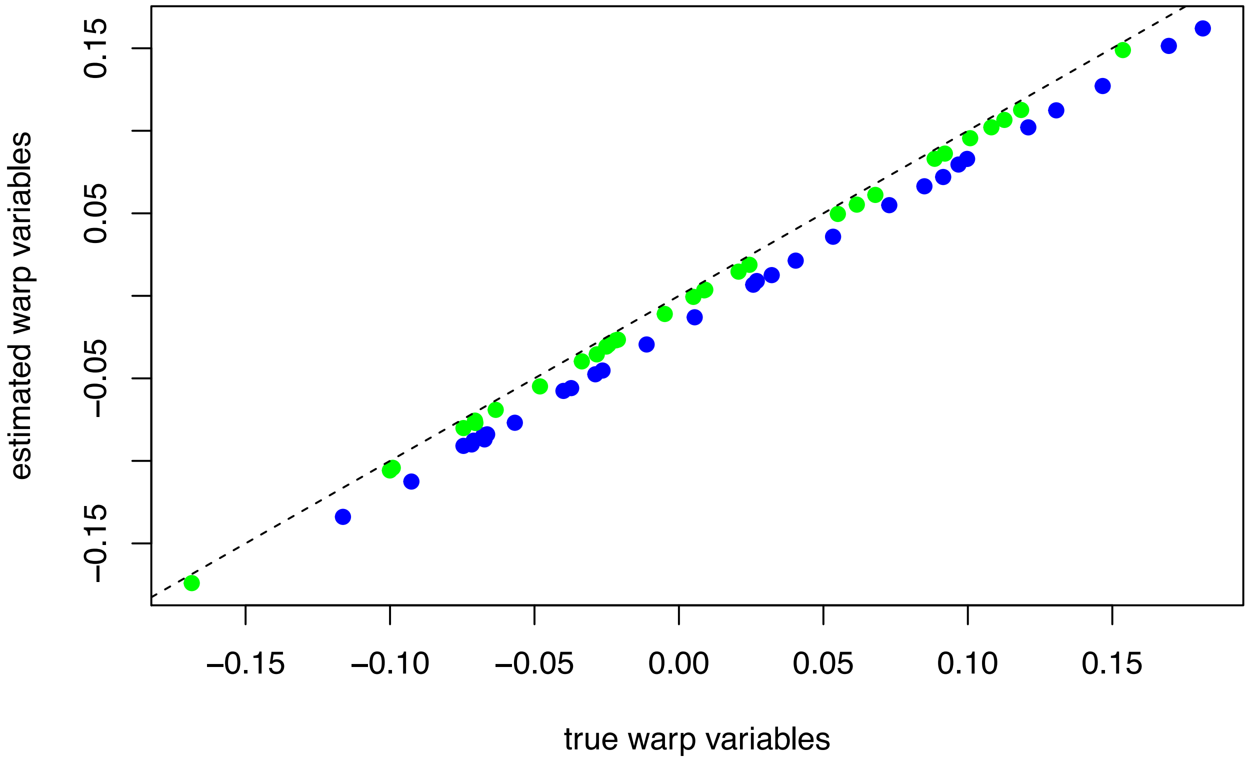
```
res <- pavpop(y, t, basis_fct, warp_fct, amp_cov, warp_cov, amp_fct, iter = c(10, 10))
```

```
#> Outer    :   Inner   :   Estimates
#> 1    :   1   2   3   4   5   6   7   8   9   10  :    1568.59 42.34475
#> Linearized likelihood:    -22351.31
#> 2    :   1   2   3   4   5   6   7   8   9   10  :    2576.439 67.36384
#> Linearized likelihood:    -23533.07
#> 3    :   1   :    2576.914 65.50864
#> Linearized likelihood:    -23534.55
#> 4    :   1   2   3   4   5   6   7   8   9   10  .
#> Likelihood not improved, returning best likelihood estimates.
```

```
# Plot aligned samples
plot(0, 0, xlim = c(0, 1), ylim = range(y), type = 'n',
     xlab = 'warped t', ylab = 'y(t)')
legend(0.7, range(y)[2], legend = c(expression(theta(t)), expression(hat(theta)(t))),
       lty = 2, lwd = 2, col = c('black', 'red'))
for (i in 1:n) lines(warp_fct(res$w[, i], t[[i]]), y[[i]], col = rainbow(n)[i])
lines(t[[1]], basis_fct(t[[1]]) %*% beta_t , lwd = 2, lty = 2)
lines(t[[1]], basis_fct(t[[1]]) %*% res$c, lwd = 2, lty = 2, col = 'red')
```



```
plot(as.numeric(w_t), as.numeric(res$w), xlab = 'true warp variables',
     ylab = 'estimated warp variables', pch = 19, col = c('green', 'blue'))
abline(0, 1, lty = 2)
```


estimated warp variables
true warp variables
−0.15
−0.05
0.05
0.15
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15