Fitting a simple tvem

John J. Dziak

2020-06-05

In this example we simulate a longitudinal dataset and fit a simple time-varying coefficient model to it.

First we load the tvem package.

```
library(tvem)
#> Loading required package: mgcv
#> Loading required package: nlme
#> This is mgcv 1.8-31. For overview type 'help("mgcv-package")'.
```

The tvem library has a function for simulating a dataset. It is good to start by specifying a random seed.

```
set.seed(123);
the_data <- simulate_tvem_example();
```

When analyzing any dataset, it is important to examine it descriptively first.

```
print(head(the_data));
#>   subject_id time  x1  x2   y
#> 1          1 0.00 5.3 5.8 1.9
#> 2          1 0.05 6.4 5.3 0.5
#> 3          1 0.10 5.4 5.5  NA
#> 4          1 0.15 5.7 3.2  NA
#> 5          1 0.20 5.8 2.8  NA
#> 6          1 0.25 8.7 3.2 2.9
print(summary(the_data));
#>    subject_id          time            x1              x2
#>  Min.   :  1.00   Min.   :0.00   Min.   : 0.00   Min.   : 0.000
#>  1st Qu.: 75.75   1st Qu.:1.75   1st Qu.: 3.60   1st Qu.: 1.900
#>  Median :150.50   Median :3.50   Median : 5.00   Median : 3.300
#>  Mean   :150.50   Mean   :3.50   Mean   : 5.02   Mean   : 3.326
#>  3rd Qu.:225.25   3rd Qu.:5.25   3rd Qu.: 6.40   3rd Qu.: 4.700
#>  Max.   :300.00   Max.   :7.00   Max.   :10.00   Max.   :10.000
#>
#>        y
#>  Min.   : 0.000
#>  1st Qu.: 2.000
#>  Median : 3.200
#>  Mean   : 3.249
#>  3rd Qu.: 4.400
#>  Max.   :10.000
#>  NA's   :29647
```

The dataset is in long form (one row per observation, with multiple observation times for each participant). There are 300 participants. Observation time ranges from 0 to 7. There is a single response variable y, and two predictor variables (covariates), x_1 and x_2. In the context of an intensive longitudinal study with

human participants, these variables might be ratings of different feelings, symptoms or behaviors, reported a few times per day at random times, during seven days following an event (such as the beginning of an intervention, treatment, lifestyle change, etc.). The values of the covariates and response vary over time within subject. However, we don't know yet whether their means change systematically over time, whether they are interrelated, or whether this relationship, if it exists, changes over time.

One easy thing to do is to investigate whether and how the response changes over time on average. This is simply curve fitting, similar to polynomial regression, but can be fit using the TVEM function, in an approach sometimes called 'intercept-only TVEM.' This approach uses a spline function to approximate the change in average y over time.

By default, the tvem function will fit a penalized B-spline, that is, a P-spline following Eilers and Marx (1996). This approach uses an automatic tuning penalty to choose the level of smoothness versus flexibility of the fitted function. It is similar, though not identical, to the P-splines used in the Methodology Center's %TVEM SAS macro, which are penalized truncated power splines.

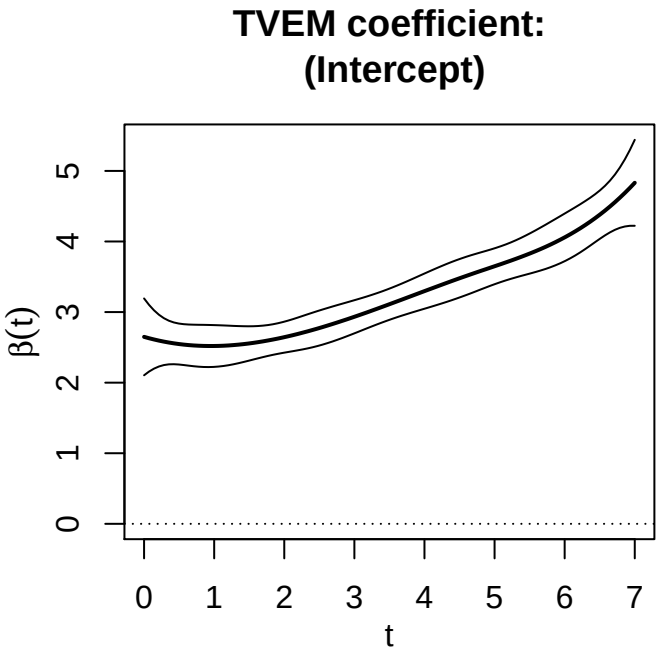
```
model1 <- tvem(data=the_data,
               formula=y~1,
               id=subject_id,
               time=time);
```

You also have the option to turn off the penalty and control the smoothness yourself, by specifying the number of interior knots, here 2.

```
model1 <- tvem(data=the_data,
               formula=y~1,
               id=subject_id,
               num_knots=2,
               penalize=FALSE,
               time=time);
```

The implied mean model is $E(y|t) = \beta_0(t)$ Where t is time in days. After fitting the model, you can print a summary and plot the estimated coefficient.

```
print(model1);
#> ========================================================
#> Time-Varying Effects Modeling (TVEM) Function Output
#> ========================================================
#> Response variable:   y
#> Time interval:   0 to 7
#> Number of subjects:  300
#> Effects specified as time-varying:  (Intercept)
#> You can use the plot_tvem function to view their plots.
#> ========================================================
#> Back-end model fitted in mgcv::bam function:
#> Method REML
#> Formula:
#> y ~ s(time, bs = "ps", by = NA, pc = 0, k = 6, fx = TRUE)
#> Pseudolikelihood AIC: 47682.1
#> Pseudolikelihood BIC: 47708.03
#> Note: Used listwise deletion for missing data.
#> ========================================================
plot(model1);
```



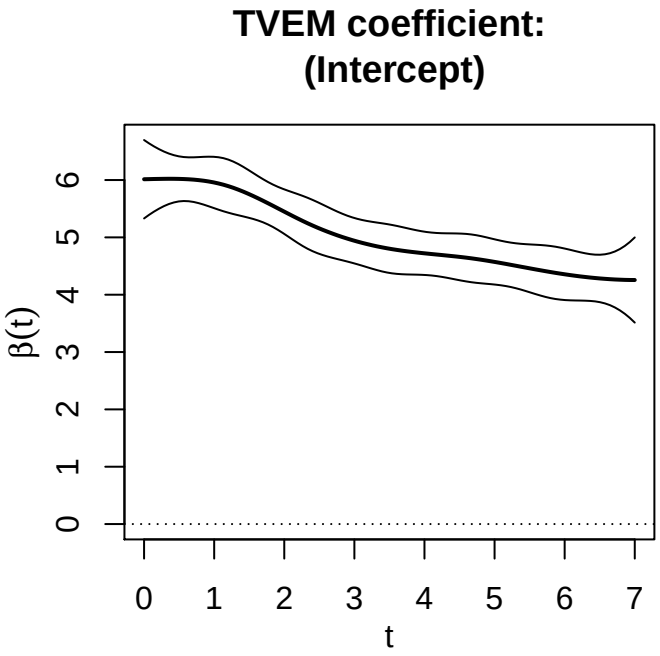
The plot shows the estimated coefficient function, and approximate estimates for 95% pointwise confidence intervals (not corrected for potential multiple comparisons) for the value of the function at each time.

We don't provide for random effects in the current version of this package. Instead, we use a (possibly penalized) form of generalized estimating equations with working independence, and adjust the standard errors for within-subject correlation using a sandwich formula.

It is a very good idea to examine how the covariate means change over time also. That is, we should fit intercept-only TVEM's for x_1 and $x_2, not just y. This gives us an opportunity to explore yet another way to fit a model using the tvem function, by using the select_tvem function to choose the number of knots by an pseudolikelihood equivalent to an AIC or BIC criterion. "Pseudolikelihood" here means that the information crtierion doesn't take within-subject correlation into account, because we are trying to fit a marginal model agnostic to the exact correlation structure.

```
model2 <- select_tvem(data=the_data,
                      formula = x1~1,
                      id=subject_id,
                      time=time);
#> [1] "Selected 5 interior knots."
print(model2);
#> ====================================================
#> Time-Varying Effects Modeling (TVEM) Function Output
#> ====================================================
#> Response variable:   x1
#> Time interval:   0 to 7
#> Number of subjects:  300
#> Effects specified as time-varying:  (Intercept)
#> You can use the plot_tvem function to view their plots.
#> ====================================================
#> Back-end model fitted in mgcv::bam function:
#> Method fREML
```

```
#> Formula:
#> x1 ~ s(time, bs = "ps", by = NA, pc = 0, k = 9, fx = FALSE)
#> Pseudolikelihood AIC: 177448.48
#> Pseudolikelihood BIC: 177478.73
#> ======================================================
plot(model2);
```

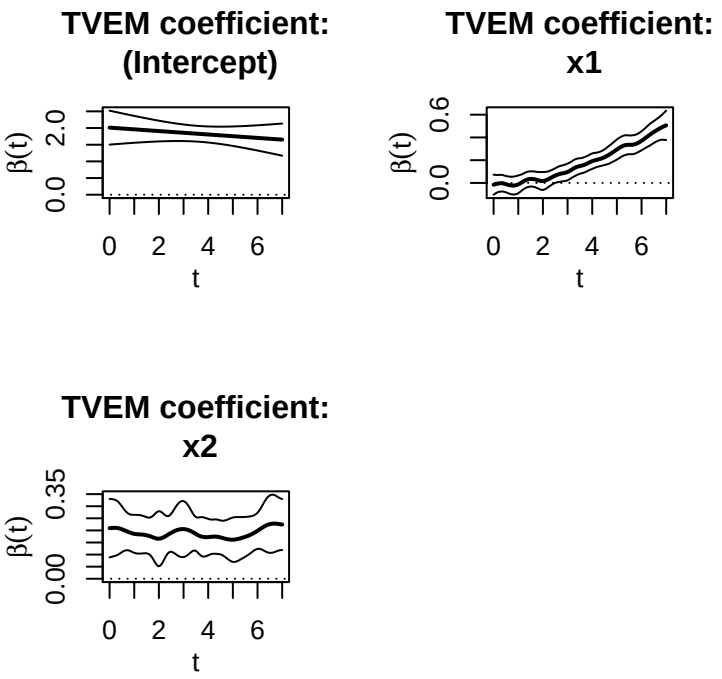


It would be good to do the same thing for x_2 also.

After this and other data exploration, we go ahead to fit a nontrivial TVEM model, with covariates. We allow both x_1 and x_2 to potentially have "time-varying effects" (regression relationships with the response that change over time, that is, a potential interaction between time and the covariate specified without the assumption of linearity). The implied mean model is $E(y|t, x_1(t), x_2(t)) = \beta_0(t) + \beta_1(t)x_1(t) + \beta 2_(t)x_2(t)$ where t is time in days.

```
model3 <- tvem(data=the_data,
               formula=y~x1+x2,
               id=subject_id,
               time=time);
print(model3);
#> ======================================================
#> Time-Varying Effects Modeling (TVEM) Function Output
#> ======================================================
#> Response variable:   y
#> Time interval:   0 to 7
#> Number of subjects:  300
#> Effects specified as time-varying:  (Intercept), x1, x2
#> You can use the plot_tvem function to view their plots.
#> ======================================================
#> Back-end model fitted in mgcv::bam function:
```

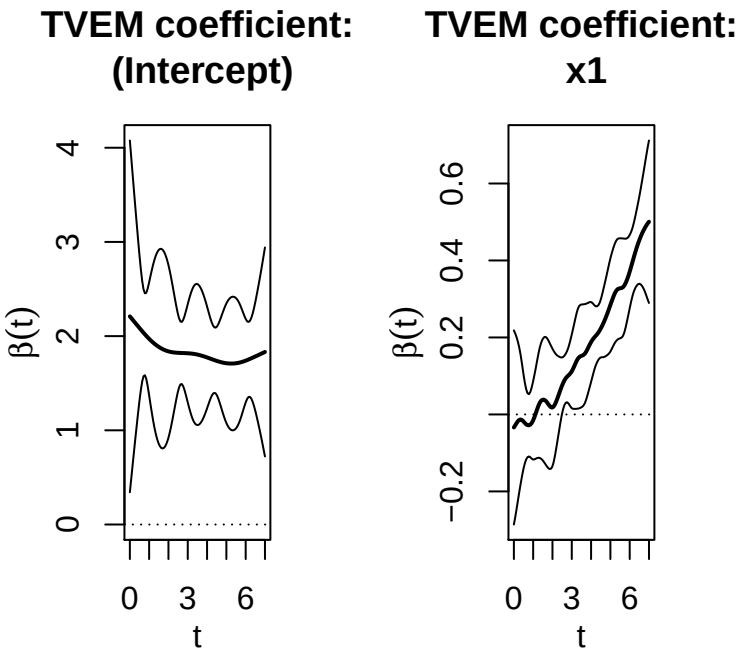
```
#> Method fREML
#> Formula:
#> y ~ x1 + x2 + s(time, bs = "ps", by = NA, pc = 0, k = 24, fx = FALSE) +
#>     s(time, bs = "ps", by = x1, pc = 0, m = c(2, 1), k = 24,
#>         fx = FALSE) + s(time, bs = "ps", by = x2, pc = 0, m = c(2,
#>     1), k = 24, fx = FALSE)
#> Pseudolikelihood AIC: 45761.6
#> Pseudolikelihood BIC: 45861.01
#> Note: Used listwise deletion for missing data.
#> ======================================================
plot(model3);
```



Holding x_1 and x_2 constant, the mean of y seems to decline over time. The penalty function estimates the relationship as linear for lack of any sign of nonlinearity. From the results, x_1 appears to have an increasingly positive relationship with y over time. x_2 also seems to predict y, but the strength of the relationship does not change over time. So we could fit a similar model, but with x_2 having a non-time-varying effect, even though it has time-varying values.

```
model4 <- tvem(data=the_data,
               formula=y~x1,
               invar_effect=~x2,
               id=subject_id,
               time=time);
print(model4);
#> ======================================================
#> Time-Varying Effects Modeling (TVEM) Function Output
#> ======================================================
#> Response variable:   y
#> Time interval:   0 to 7
```

```
#> Number of subjects:  300
#> Effects specified as time-varying:  (Intercept), x1
#> You can use the plot_tvem function to view their plots.
#> ========================================================
#> Effects specified as non-time-varying:
#>     estimate standard_error
#> x2 0.1885078     0.01877894
#> ========================================================
#> Back-end model fitted in mgcv::bam function:
#> Method fREML
#> Formula:
#> y ~ x1 + x2 + s(time, bs = "ps", by = NA, pc = 0, k = 24, fx = FALSE) +
#>     s(time, bs = "ps", by = x1, pc = 0, m = c(2, 1), k = 24,
#>         fx = FALSE)
#> Pseudolikelihood AIC: 45773.24
#> Pseudolikelihood BIC: 45864.5
#> Note: Used listwise deletion for missing data.
#> ========================================================
plot(model4);
```



The implied mean model is $E(y|t) = \beta_0(t) + \beta_1(t)x_1(t) + \beta_2 x_2(t)$.

The seeming oscillations in the confidence interval width do not have any particular interpretation; they are just an accident of the placement of the knots. The takeaway message is the increasing $\beta_1(t)$ over time, suggesting an increasing association between x_1 and y, that is, some kind of interaction between t and x_1 in predicting y.