Empirical orthogonal functions

James T. Thorson

```
library(tinyVAST)
library(fmesher)
set.seed(101)
```

`tinyVAST` is an R package for fitting vector autoregressive spatio-temporal (VAST) models. We here explore the capacity to specify a generalized linear latent variable model that is configured to generalize an empirical orthogonal function analysis.

Empirical Orthogonal Function (EOF) analysis

To start, we reformat data on September Sea ice concentrations:

```
data( sea_ice )
library(sf)
library(rnaturalearth)

# project data
sf_ice = st_as_sf( sea_ice, coords = c("lon","lat") )
st_crs(sf_ice) = "+proj=longlat +datum=WGS84"
sf_ice = st_transform( sf_ice,
                       crs=st_crs("+proj=laea +lat_0=90 +lon_0=-30 +units=km") )

#
sf_pole = st_point( c(0,90) )
sf_pole = st_sfc( sf_pole, crs="+proj=longlat +datum=WGS84" )
sf_pole = st_transform( sf_pole, crs=st_crs(sf_ice) )
sf_pole = st_buffer( sf_pole, dist=3000 )
sf_ice = st_intersection( sf_ice, sf_pole )
#> Warning: attribute variables are assumed to be spatially constant throughout all geometries

Data = data.frame( st_drop_geometry(sf_ice),
              st_coordinates(sf_ice),
              var = "Ice" )
```

Next, we construct the various inputs to *tinyVAST*

```
n_eof = 2
dsem = make_eof_ram( variables = "Ice",
                     times = sort(unique(Data[,'year'])),
                     n_eof = 2,
                     standard_deviations = 0 )
mesh = fm_mesh_2d( Data[,c('X','Y')], cutoff=1.5 )
```

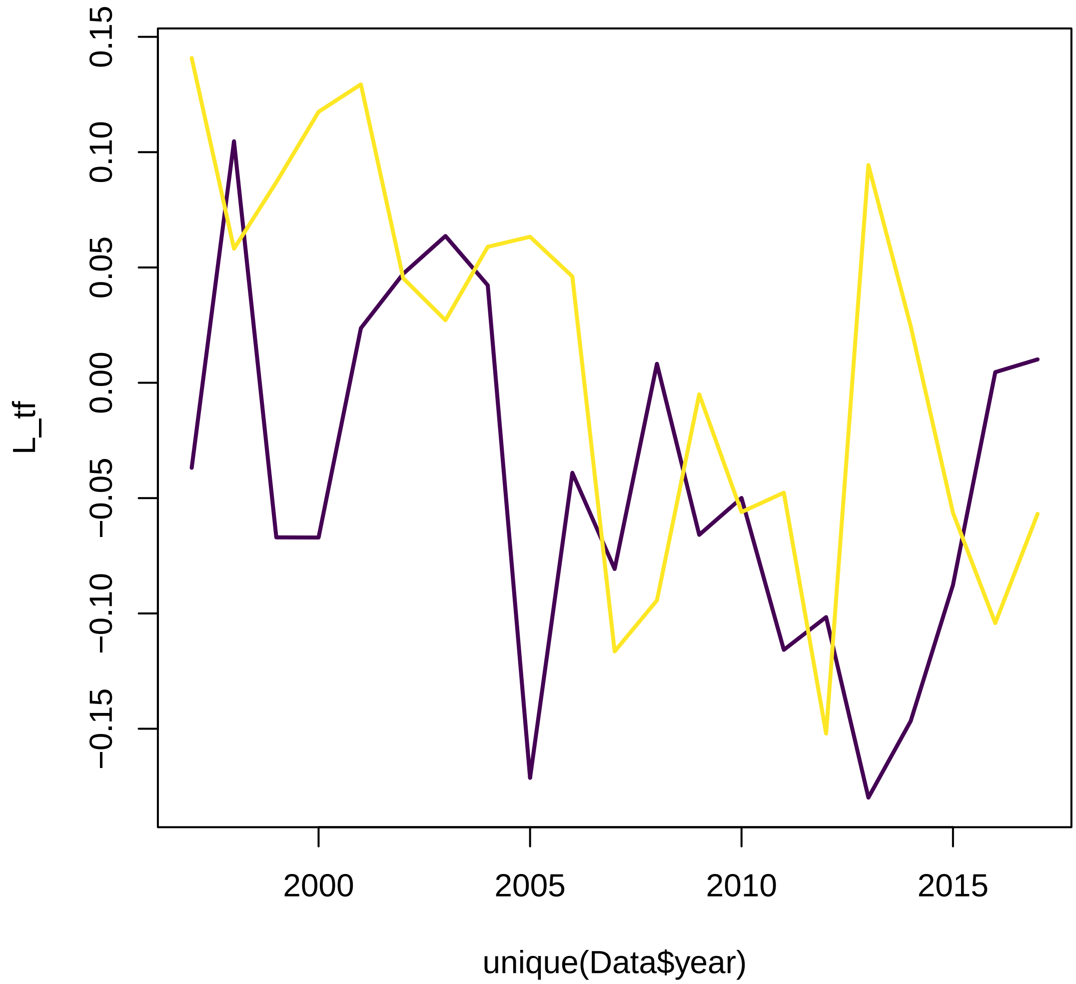
```r
#
family_link = matrix( 0,
                      nrow = length(unique(Data[,'var'])),
                      ncol = 2,
                      dimnames = list(unique(Data[,'var']),NULL) )

# fit model
out = fit( dsem = dsem,
           sem = "",
           data = as.data.frame(Data),
           formula = ice_concentration ~ 1,
           spatial_graph = mesh,
           family_link = family_link,
           data_colnames = list( "spatial"=c("X","Y"), "variable"="var",
                                 "time"="year", "distribution"="var"),
           times = c(paste0("EOF_",seq_len(n_eof)), sort(unique(Data[,'year']))),
           control = tinyVASTcontrol( quiet=TRUE, trace=0, profile="alpha_j",
                                      nlminb_loops=1, getsd=TRUE,
                                      gmrf_parameterization="projection") )
```

Finally, we can extract, rotate, and plot the dominant modes of variability

```r
# Visualize index
L_tf = matrix( 0, nrow=length(unique(Data$year)), ncol=2,
               dimnames=list(unique(Data$year),c("EOF_1","EOF_2")) )
L_tf[lower.tri(L_tf,diag=TRUE)] = out$opt$par[names(out$opt$par)=="beta_z"]
rotated_results = rotate_pca( L_tf=L_tf, x_sf=array(dim=c(0,n_eof)), order="decreasing" )
#> Warning in sqrt(Eigen$values): NaNs produced
L_tf = rotated_results$L_tf

# Line graph
matplot( y=L_tf, x=unique(Data$year), type="l",
         col=viridisLite::viridis(n_eof), lwd=2, lty="solid" )
```



0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
L_tf
2000
2005
2010
2015
unique(Data$year)