evgam

Generalised Additive Extreme Value Models

Ben Youngman

27/06/2021

Outline

- 20-minute intro to `evgam`
- 20-minute hands-on practical
- 10-minute outline of some of `evgam`'s less commonly used features
- 10-minute Q&A and discussion

Background

- `evgam` was added to CRAN in 2020
- Its purpose is to fit extreme value distributions (GEV, GPD, r-largest) with parameters of generalised additive model (GAM) form
- Past related works include penalised likelihood models for extremes (Pauli and Coles, 2001), GAMs for sample extremes (Chavez-Demoulin and Davison, 2005) and spline-based priors (Randell et al., 2016), amongst others
- `evgam` provides objective estimation of smoothing parameters (which control the wiggliness of GAMs) by implementing Laplace's method, as developed in (Wood, 2011; Wood et al., 2016)
- Calls to `evgam` are inspired by calls to Simon Wood's wonderful `mgcv` package (as is some code inside); see Wood (2017)

Getting started

- To fit a model with `evgam`, you just need the `evgam()` function
- Its basic setup is

```
> evgam(formula, data, family, ...)
```

- `formula` is a formula for each distribution parameter
  - for the GEV distribution, we'd want three formulae, which we'd supply as a list, such as

```
> list(response ~ location, ~ log_scale, ~ shape)
```

- `data` is a `data.frame`
- `family` is a character string specifying the EVD to fit
  - `family = 'gev'`, is the GEV distribution, and the default; `'gpd'` for GPD; `'rlarge'` for *r*-largest; `'exp'` for exponential; `'gauss'` for Gaussian; `'weibull'` for Weibull; `'ald'` for asymmetric Laplace distribution

Specifying your `formula`

- Smooth functions are put in `formula` with function `mgcv::s()`, but we can simply call `s()`
- Most of what's accepted by `s()` should work with `evgam` (but I've not checked every combination!)
- Suppose we've got explanatory variable `x1`; then we fit a smooth in `x1` with `s(x1, ...)`
  - if we've got explanatory variables `x1` and `x2` then `s(x1, x2)` gives a two-dimensional smooth in `x1` and `x2`, e.g. spatial model
- The default basis is the thin-plate regression spline (Wood, 2003), i.e. implicitly `s(..., bs = 'tp')`
  - other useful bases are cubic regression splines, `bs = 'cr'`, for one-dimensional smooths, Markov random fields, `bs = 'mrf'`, random effects, `bs = 're'`, and cyclic versions of some, e.g. `bs = 'cc'` for a cyclic cubic regression spline
  - `?mgcv::smooth.terms` gives a full list of what's available
- The basis dimension is controlled with `s(..., k = ...)`
  - higher basis dimensions allow the wigglier smooths; smoothing parameters then 'optimise' how wiggly smooths end up

Specifying your `formula`

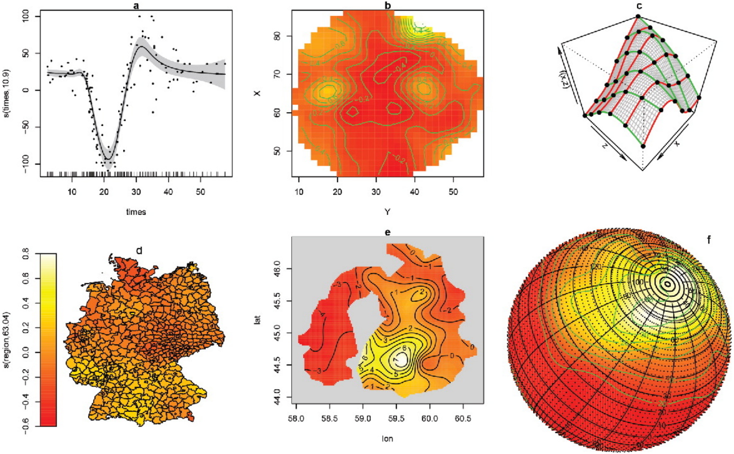


Figure 1: (Wood et al., 2016) Examples of smooths: (a) cubic regression spline; (b) 2D thin plate regression spline (c) tensor product of two 1D smooths; (d) Gaussian Markov random fields; (e) soap film; (f) splines on the sphere.

A starter example

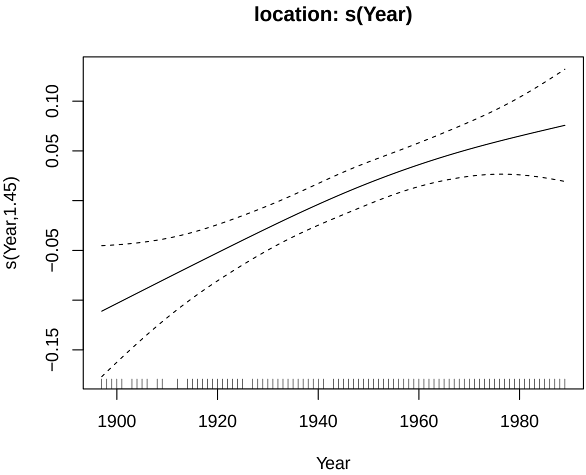
- Consider the Fremantle sea level data (see Coles, 2001, Example 1.3)
- Let $Y(t)$ denote annual maximum sea level in year t, for $t = 1887, ..., 1989$
- Assume $Y(t) \sim GEV(\mu(t), \psi, \xi)$
- Let $\mu(t)$ vary according to a cubic regression spline of rank 5

```
> library(evgam)
> data(fremantle)
> fmla <- list(SeaLevel                           # response variable
+              ~ s(Year, k = 5, bs = 'cr'), # location
+              ~ 1,                         # log scale
+              ~ 1)                         # shape
> m <- evgam(fmla, fremantle, family = 'gev')
```

Some generic functions

- Given a fit from `evgam`, e.g. `m` from before, we can plot a fitted object

```
> plot(m)
```



Some generic functions

- We can make predictions and get summaries

```
> predict(m, data.frame(Year = 1990), type = 'response')
```

```
##   location     scale      shape
## 1 1.556782 0.1228177 -0.1153052
```

```
> summary(m)
```

```
##
## ** Parametric terms **
##
## location
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.48       0.01  100.42   <2e-16
##
## logscale
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2.1       0.09  -24.54   <2e-16
##
## shape
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.12       0.07   -1.61    0.054
##
## ** Smooth terms **
##
```

Time to have a go yourself

You should have

- evgam_questions.pdf, which gives some questions to attempt and see some key features of `evgam`
- evgam_partial.pdf and evgam_partial.R, as above, but with some `R` code given for early questions, to help you get started (recommended)
- evgam_solution.pdf and evgam_solution.R, as above, but with all `R` code given, and resulting plots etc.

Some extra features

Tensor products for interactions

- Tensor-product smooths are particularly convenient for introducing interactions between smooths (Wood, 2006)
  - we can use `mgcv`'s `te()` function for these
- A spatio-temporal smooth, with coordinates `(lon, lat)`, and time component `time`, is a good example, so

```
> te(lon, lat, time, d = c(2, 1), bs = c('tp', 'cr'))
```

gives a smooth with a two-dimensional thin plate regression spline that varies with time according to a cubic regression spline.

- Here's a spatio-temporal example of mean US temperatures for Jan–Mar.

Some extra features

The asymmetric Laplace distribution

- It might not be an EVD, but the asymmetric Laplace distribution (ALD) can be used to perform quantile regression (Yu and Moyeed, 2001)
- ... and then be used to define the 'high threshold' needed to identify exceedances to model as GPD (Northrop and Jonathan, 2011)
- The following code fits an ALD model

```
> m1 <- evgam(formula1, data1, family = 'ald',
+             ald.args = list(tau = p))
```

and we'd need to specify p, the quantile we want to estimate. (Note that this implements the modified check function of (Oh et al., 2011), as it eases numerical optimisation)

```
> data1$threshold <- predict(m1)$location
> data1$excess <- data1$response - data1$threshold
> data2 <- subset(data1, excess > 0)
> m2 <- evgam(formula2, data2, family = 'gpd')
```

Some extra features

Return level estimation

- `evgam` is designed to make return level estimation a bit easier
- Suppose we've fitted a model, `m`, to annual maxima
- We can get $1/p$-year return level estimates for some `newdata` with

```
> predict(m, newdata, probs = 1 - p)
```

- `p` can be scalar or vector, depending on whether one or multiple return level estimate(s) is sought
- We can also numerically estimate return levels, such as for

$$F_{\text{ann}}(z) = \prod_{j=1}^{m} \left\{ F_{\text{GEV}}\left(z; \mu_j, \psi_j, \xi_j\right) \right\}^{m\alpha_j\theta_j},$$

  i.e. where the cdf of the annual maximum is a product of GEVs
- To solve $F_{\text{ann}}(z_p) = p$ we can use

```
> qev(p, loc, scale, shape, m = 1, alpha = 1, theta = 1, family)
```

where `loc`, `scale` and `shape` are an EVD's location, scale and shape parameters, respectively, and with scalars or vectors `m`, `alpha` and `theta` corresponding to m, α and θ, respectively.

Summary and the future

- `evgam` is designed for fitting EVDs with parameters that vary with GAM forms
- It heavily uses – and is heavily inspired by – `mgcv`, and hence uses similar calls
- GAM forms offer simple – yet flexible – spatial modelling, such as with thin-plate regression splines, trend estimation, such as with cubic regression splines, plus others
- I have some ideas for developing `evgam`...
  - more models, in addition to GEV, GPD, exponential, Weibull (they're usually straightforward to add)
  - conversion of some `R` code to `Rcpp`/`RcppArmadillo`, which should bring some speed-ups
  - better handling of sparsity, which is currently ignored
  - suggestions welcome and appreciated

References I

Chavez-Demoulin, V. and A. C. Davison (2005). Generalized additive modelling of sample extremes. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 54*(1), 207–222.

Coles, S. G. (2001). *An Introduction to Statistical Modeling of Extreme Values*. Springer-Verlag London.

Northrop, P. J. and P. Jonathan (2011). Threshold modelling of spatially dependent non-stationary extremes with application to hurricane-induced wave heights. *Environmetrics 22*(7), 799–809.

Oh, H.-S., T. C. M. Lee, and D. W. Nychka (2011). Fast Nonparametric Quantile Regression With Arbitrary Smoothing Methods. *Journal of Computational and Graphical Statistics 20*(2), 510–526.

Pauli, F. and S. G. Coles (2001). Penalized likelihood inference in extreme value analyses. *Journal of Applied Statistics 28*(5), 547–560.

Randell, D., K. Turnbull, K. Ewans, and P. Jonathan (2016). Bayesian inference for nonstationary marginal extremes. *Environmetrics 27*(7), 439–450.

References II

Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 65*(1), 95–114.

Wood, S. N. (2006). Low-Rank Scale-Invariant Tensor Product Smooths for Generalized Additive Mixed Models. *Biometrics 62*(4), 1025–1036.

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(1), 3–36.

Wood, S. N. (2017). *Generalized additive models: an introduction with R* (Second ed.). CRC press.

Wood, S. N., N. Pya, and B. Säfken (2016). Smoothing parameter and model selection for general smooth models. *Journal of the American Statistical Association 111*(516), 1548–1563.

Yu, K. and R. A. Moyeed (2001). Bayesian quantile regression. *Statistics & Probability Letters 54*(4), 437–447.