The yaev package: Yet Another Extreme Value package?

Yves Deville deville.yves@alpestat.com

July 1, 2022

Contents

1 Probability functions of Extreme-Value distributions

The probability functions for the GPD and GEV distributions depend smoothly on the parameters: they are infinitely differentiable functions of the parameters. However these functions are not analytic functions of the parameters and a singularity exists for all the functions when the shape parameter say ξ is zero. In practice, the functions are given with different formulas depending on whether ξ is zero or not; the formulas for $\xi = 0$ relate to the exponential and Gumbel distributions and correspond to the limit for $\xi \to 0$ of the functions given by the formulas for $\xi \neq 0$. As an example consider the quantile function of the Generalized Pareto distribution with shape ξ and unit scale

$$q(p) = \begin{cases} [(1-p)^{-\xi} - 1]/\xi & \xi \neq 0 \\ -\log(1-p) & \xi = 0, \end{cases} \qquad 0 < p < 1. \tag{1}$$

It can be shown that for $\xi \approx 0$ whatever be p

$$q \approx -\log(1-p), \qquad \frac{\partial q}{\partial \xi} \approx \frac{1}{2}\log^2(1-p), \qquad \frac{\partial^2 q}{\partial \xi^2} \approx -\frac{1}{3}\log^2(1-p).$$

It is quite easy to obtain expressions for the the derivatives w.r.t. ξ using the definition (1). We can even rely on the symbolic differentiation method `D` available in R which, as opposed to me and many other humans, never makes any mistake when differentiating.

```r
qEx <- function(p, xi) ((1 - p)^(-xi) - 1) / xi
dqEx <- D(expression(((1 - p)^(-xi) - 1) / xi), name = "xi")
d2qEx <- D(dqEx, name = "xi")
```

```
p <- 0.99; q <- 1 - p
for (xi in c(1e-4, 1e-6, 1e-8)) {
    r <- rbind("ord 0" = c("lim" = -log(q), "der" = qEx(p = p, xi = xi)),
               "ord 1" = c("lim" = log(q)^2 / 2, "der" = eval(dqEx, list(p = p, xi = xi))),
               "ord 2" = c("lim" = -log(q)^3 / 3, "der" = eval(d2qEx, list(x = p, xi = xi))))
    cat("xi = ", xi, "\n")
    print(r)
}

## xi =  1e-04
##            lim       der
## ord 0  4.60517  4.606231
## ord 1 10.60380 10.607052
## ord 2 32.55486 32.566137
## xi =  1e-06
##            lim       der
## ord 0  4.60517  4.605181
## ord 1 10.60380 10.603811
## ord 2 32.55486 68.213867
## xi =  1e-08
##            lim           der
## ord 0  4.60517  4.605170e+00
## ord 1 10.60380  1.085129e+01
## ord 2 32.55486 -4.949912e+07
```

We see that the formula for the function works fine. However, the formula for the 2-nd order derivative can be completely wrong when ξ is about 1e-6 and the formula for the 1-st order derivative can also be wrong when ξ is about 1e-8. The reason is that the formulas for the derivatives involve difference and/or fractions or small quantities since ξ or ξ^2 comes at the denominator. As a general rule the derivatives with higher order are more difficult to evaluate numerically, since they involve more complex expressions. Note that using $\xi \leqslant$ 1e-6 is quite common in EVA because the values of ξ used in practice are often quite small, and moreover a very small value of ξ is often used in the initialisation of the Maximum-Likelihood (ML) optimization.

Although not yet widespread, the use of the exact formulas for the derivatives w.r.t. the parameters can be of great help in the optimization tasks required in EVA. These tasks of course involve the ML estimation but also profile-likelihood inference for models with covariates. Differential equations methods can be also used to derive confidence intervals. Note that using the formulas for the derivatives is *symbolic* differentiation, which differs from *automatic* differentiation as increasingly available.

Our strategy consists in fixing a small $\epsilon > 0$ and use the formulas only when $|\xi| > \epsilon$. When $|\xi| \leqslant \epsilon$, we use a few terms from the Taylor series at $\xi = 0$

$$q(p;\, \xi) \approx q(p;\, 0) + \partial_\xi q(p;\, \xi)\big|_{\xi=0} \times \xi + \frac{1}{2}\, \partial^2_{\xi,\xi} q(p;\, \xi)\big|_{\xi=0} \times \xi^2 + o(\xi^2).$$

In order to maintain the consistency between the derivatives, it seems good to use the same ϵ for all the derivatives and use consistent Taylor approximations, so use the order 1 for the derivative $\partial_\xi q$ and the order order 0 for $\partial^2_{\xi,\xi} q$ or a crossed derivative such as $\partial^2_{\sigma,\xi} q$. Since the 2-nd order derivatives can be required, we must take a value for ϵ which is not too small: 1e-4 or 1e-5, not much smaller. This method is used in some codes of the **revdbayes** package by Paul Northrop.

2 Deriving the formulas

The reports provided with **yaev** give the exact expressions for the first-order and the second-order derivatives of the probability functions w.r.t. the parameters and also provides workable approximations for the case $\xi \approx 0$. We used the Maxima Computer Algebra System along with the maxiplot package for LaTeX.

- The **raw expressions given by Maxima are reported in green**. The expressions can be regarded as exact, not being influenced by manual computations. However these formulas are usually difficult to use in a compiled code and require some manual transformation for this aim.
- The **simplified expressions are reported in red**. These expressions are derived by us from the raw expressions; they are influenced by manual computations hence could in principle contain errors, although they have been carefully checked. These formulas are used to write the compiled code. They often use auxiliary variables that are shared across several formulas.

3 Testing the derivatives

The **yaev** package comes with a series of tests in the format of the **testthat** package. The **numDeriv** package is used to compute the derivatives by numeric differentiation up to the order 2; these derivatives are compared to those provided by the formulas.

A quite difficult task when checking derivatives is to give a threshold used to decide if the difference between the numeric derivative and the symbolic derivative, say the “error”, is acceptable or not. This error has two sources: one is the numerical evaluation of the symbolic derivative or of its approximation for $\xi \approx 0$ (see example above), and the other is the approximation used in numeric differentiation where the limit defining the derivative is replaced by a finite difference. We should use small values of ξ with $|\xi| < \epsilon$ to check that the approximation for small ξ is correct, although we can only test the approximation at the first order by doing so. Then, with a good choice of ϵ the error should be mainly due to the numeric differentiation. But when the true derivative is small, the relative error may be large (think of a true derivative which is exactly zero). On the other hand, when a derivative is large in absolute value, the absolute error may also be quite large. Mind that the derivatives can in practice be very small or very large, and also that a gradient vector or a Hessian matrix often contain values that are not of the same order of magnitude.

We check that either the *absolute* error or the *relative* error is small. The idea is that none of these two things can come by chance, and if one holds, the formula used must be good even if the other criterion suggests an opposed conclusion. The test is made *elementwise* meaning that the relative error is computed for each element of a gradient vector or Hessian matrix ignoring the other elements.

4 POT to NHPP parameterizations

XXX To be completed

Package	GEV					GPD				
	Lang.	Vec. $\boldsymbol{\theta}$	Grad.	Hess.	$\xi = 0$	Lang.	Vec. $\boldsymbol{\theta}$	Grad.	Hess.	$\xi = 0$
evir	R	no	no	no	NT	R	no	no	no	NT
evd	R	no	no	no	0.0	R	no	no	no	0.0
ismev	R*	no	no	no	NT	R*	no	no	no	NT
Renext						R	no	yes	yes	ϵ/S
POT						R*	yes	no	no	0.0
SpatialExtremes	R	yes	no	no	0.0	R	yes	no	no	0.0
revdbayes	R	yes	no	no	ϵ/AI	R	yes	no	no	ϵ/AI
mev	R	yes	yes*	yes*	NT	R	yes	yes*	yes*	NT
yaev	C	yes	yes	yes	ϵ/AI	C	yes	yes	yes	ϵ/AI

Table 1: Features of some CRAN packages. *Lang.*: the implementation language, *Vec.* $\boldsymbol{\theta}$: vectorized w.r.t. the parameters. The columns *Grad.* and the *Hess.* indicate if the gradient and Hessian are provided, and the columns $\epsilon = 0$ indicate the strategy used to cope with a zero or small shape, as described in the text. A star $\star$ means that the functions are not exported.

5 EV distributions from other R packages

The EV distributions are implemented in many R packages. A variety of strategies regarding the problem $\xi = 0$ can be found. We now describe these strategies and provide for each of them a "code" ($\boxed{\text{NT}}$, ...) that is used in Table 1, columns $\xi = 0$. Each strategy is briefly discussed.

1. $\boxed{\text{NT}}$ Use only the formula for $\xi \neq 0$ i.e., "do nothing". In practice, an optimization or sampling algorithm will never come to the case $\xi = 0$ *exactly* and this can only happen when the user gives this value e.g. as an initial value. There will be some numerical problems when ξ is very small, say $\xi =$ 1e-14 or less. These problems are not so crucial for the usual probability functions: we get some wiggling when plotting the curves and zooming. Mind however that the random generation `rgev` or `rgpd` will produce silly results with very small ξ if they use `qgev` or `qgpd`.

2. $\boxed{0.0}$ Test the exact equality $\xi = 0$, and if this is true, *switch to the exponential/Gumbel formula*. This helps only when the user gives `xi = 0.0`, but we are essentially doing the same thing as in $\boxed{\text{NT}}$.

3. $\boxed{\epsilon/\text{S}}$ Test the equality $|\xi| \leqslant \epsilon$ where $\epsilon > 0$ is very small, and if this is true, *switch to the exponential/Gumbel formula*. So this produces a (very small) discontinuity. E.g., **Renext** uses $\epsilon \approx$ 2e-14.

4. $\boxed{\epsilon/\text{AI}}$ Test the equality $|\xi| \leqslant \epsilon$ where $\epsilon > 0$ is very small, and if this is true, use a dedicated *approximation or interpolation*. Several methods can be used including Taylor approximations. The discontinuity should then be undetectable. Mind the probability functions although not being *analytic* functions, are infinitely differentiable C^∞ w.r.t. the parameters.

Acknowledgments

We are grateful to the authors and contributors of **Maxima** and to the authors of the **maxiplot** LaTeX package (J.M. Planas and José Manuel Mira univ. de Murcia, Spain) which helped much for the tedious computations required by the package.