The **yaev** package: Yet Another Extreme Value package?

Yves Deville deville.yves@alpestat.com

June 30, 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 derivation 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 an the formula for the 1-st order derivative wan also be wrong whe 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, since they involve more complex expressions. Note that using $\xi \approx$ 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 $\xi = 0$. 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. 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.

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.
- 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.

3 Testing

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 up to order 2 by numeric differentiation. These derivatives can be 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. When a derivative is small, the relative error may be large. On the other side, when a derivative is large in absolute value, the absolute error may also be quite large. Mind that the derivatives can be quite large in practice 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 *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. 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

5 EV distributions from other R packages

XXX To be completed