'EffectiveDose'- Version 1.0-5
Estimation of the Effective Dose including Bootstrap confidence intervals in R

Regine Scheder

February 28, 2008

1 Introduction

To analyze the effects of a chemical drug, the dose-response relationship is studied in pharmacology or toxicology. The determination of the effective dose level is an important issue in this context. In quantal bioassay experiments m different subjects are treated with different drug doses, and it is observed if the subject reacts or not. So the response of these experiments is binary or quantal, which means either the drug takes effect and the subject has a reaction or not. The dose-response curve describes the probability of success or response of the analyzed drug. Traditionally, parametric models like the probit or logit model are used to estimate the dose response curve

$$p(x) = P(Y = 1|X = x) = 1 - P(Y = 0|X = x).$$

The effective dose level for a given α is obtained by

$$\mathrm{ED}_\alpha = p^{-1}(\alpha)$$

for an increasing probability function p. Parametric models suffer from the drawback that the specific parametric form has to be known. In many applications the specific structure is not evident, then nonparametric methods come into play.
The R-package 'EffectiveDose' is an implementation of the nonparametric estimate of the effective dose in quantal bioassay developed by Dette, Neumeyer, and Pilz (2005). In Section 2 the nonparametric estimate is introduced and the method for the bootstrap confidence intervals is explained. In the following section the usage and functionality of the package is discussed in detail. Finally in Section 4, the accuracy of the Bootstrap confidence intervals is scrutinized.

2 The nonparametric estimate and Bootstrap confidence intervals

Dette et al. (2005) proposed a strictly increasing estimator for the effective dose level ED_α, which is a combination of a regression and an integrate kernel density estimate. First of all, the dose-response curve is estimated by local linear techniques, i.e. the weighted sum of squares

$$\sum_{i=1}^{m}\{Y_i - \beta_0 - \beta_1(x_i - x)\}^2 K\left(\frac{x_i - x}{h}\right) \tag{1}$$

is minimized with respect to the parameters β_0 and β_1. Here K is a kernel function and h denotes a bandwidth, which converges to 0 with increasing sample size. The resulting estimate is given by $\hat{p}_{LL}(x) = \hat{\beta}_0$ if $(\hat{\beta}_0, \hat{\beta}_1)$ minimizes the equation (1). As in the last two sections this estimate is not necessarily monotone in x. The estimate of the effective dose level is obtained by applying a method to $\hat{p}_{LL}$ which deals simultaneously with this lack and the issue of inversion. Precisely, we define the effective dose level estimate for $\alpha \in (0,1)$ by

$$\hat{p}_I^{-1}(\alpha) = \hat{p}_I^{-1}(\alpha) := \frac{1}{Nh_d}\sum_{i=1}^{N}\int_{-\infty}^{\alpha} K_d\left(\frac{\hat{p}_{LL}(\frac{i}{N}) - u}{h_d}\right) du, \tag{2}$$

where the kernel K_d is positive, symmetric, twice continuously differentiable, and supported on $[-1,1]$. The corresponding bandwidth h_d converges to 0 with increasing sample size m. A detailed discussion of this estimate and its asymptotic behavior can be found in Dette et al. (2005).
In order to obtain confidence intervals for the effective dose level $\hat{p}_I^{-1}(\alpha)$, we suggest Bootstrap confidence intervals. Since the asymptotic distribution of $\hat{p}_I^{-1}$ contains unknown quantities like the second derivative of p, Bootstrap methods are often preferred. In this package, we apply a wild bootstrap method. We start to compute the residuals

$$\hat{\varepsilon}_j = Y_j - \hat{p}_I(x_j),$$

where $\hat{p}_I$ is the inverse of $\hat{p}_I^{-1}$ and by the way a strictly monotone estimate of the dose-response function p. We draw random variables ε_j^* with zero mean and variances $\hat{\varepsilon}_j^2$ for each iteration. Additionally, we assume that

$$P(\varepsilon_j^* = -\hat{\varepsilon}_j) = P(\varepsilon_j^* = +\hat{\varepsilon}_j) = 1/2.$$

With these randomly drawn ε_j^*, we construct a new data set by replacing Y_j with $\hat{p}_I(x_j) + \varepsilon_j^*$. Using this new data set, the effective dose level $\hat{p}_I^{*-1}(\alpha)$ is computed as described above. Then for each iteration, $|\hat{p}_I^{-1}(\alpha) - \hat{p}_I^{*-1}(\alpha)|$ is calculated. For a given confidence level β, the corresponding quantile t_β of

$|\hat{p}_I^{-1}(\alpha) - \hat{p}_I^{*-1}(\alpha)|$ over all iteration is used to give the following confidence interval

$$(\hat{p}_I^{-1}(\alpha) - t_\beta, \hat{p}_I^{-1}(\alpha) + t_\beta).$$

3 How to use 'EffectiveDose'

The package `EffectiveDose` contains functions to calculate the effective dose levels and Bootstrap confidence intervals. The function `ED` can be applied to `list`, `locfit`, and `locpoly` objects. The output of this function is an object of class `fitED`. The usage of this function is as follows. First of all, we start the package and generate data of a binary response model with a Normal distribution function as success probability.

```
> library(EffectiveDose)

locfit 1.5-4          2007-11-27

> ybin = function(x) {
+     n = length(x)
+     y = numeric(n)
+     p = pnorm(x, mean = 0.5, sd = 0.5)
+     for (i in 1:n) {
+         y[i] = rbinom(1, 1, prob = p[i])
+     }
+     return(y)
+ }
> x = seq(0, 1, length.out = 50)
> y = ybin(x)
```

The function `ED` can be used for a list object, where the data is combined. In this case within the function `ED` a local linear estimate is fitted to the orginial data. This interim estimator is used to obtain an estimate for the effective dose. The user can adjust the local linear estimator himself and fit a local linear estimate using the functions `locfit` or `locpoly`. In this way, the local linear method inside of the effective dose estimate can be controlled. This may be interesting for a more sophisticated user. We refer to the `locfit` package about the specific options of the `locfit` function.

```
> fit = locfit(y ~ lp(x, deg = 1, h = 0.1, nn = 0))
> fit2 = locpoly(x, y, degree = 1, bandwidth = 0.1)
> res = ED(list(x, y), alpha = seq(0.2, 0.8, length.out = 40))
> res2 = ED(fit, alpha = seq(0.2, 0.8, length.out = 40))
> res3 = ED(fit2, alpha = seq(0.2, 0.8, length.out = 40))
```

Beside the first argument which is the `list`, `locfit`, or `locpoly` object, the argument `alpha` specifies the α values where the effective dose is to be computed. Additionally, there are the arguments `bandwidth`, `N`, `mono`, and `type` for further specification. The arguments `bandwidth` and `N` are utilized for the monotonizing inversion [see (2)]. If the values are missing, the function `ED` uses a rule of thumb for the bandwidth and the default value 101 for `N`. The argument `mono` comes into play if the underlying model has a decreasing dose-response curve, e.g. in toxicology. For such setups the effective dose level EDα is in fact $p^{-1}(1-\alpha)$, such that the effective dose levels are again monotone increasing. The function `ED` returns $p^{-1}(1-\alpha)$ as effective dose level in this case. We demonstrate this in an example.

```
> toxdata = data.frame(conc = c(0, 0.22, 0.355, 0.444, 0.887),
+     matured = c(12, 7, 8, 2, 0), total = c(12, 8, 11, 11, 11))
> with(toxdata, ED(list(x = conc, y = matured/total), alpha = c(0.1,
+     0.2, 0.5, 0.8, 0.9), mono = "decreasing"))

Effective Dose level ED
CALL:
ED(fitprob = list(x = conc, y = matured/total), alpha = c(0.1,
    0.2, 0.5, 0.8, 0.9), mono = "decreasing")
alpha-values:
[1] 0.1 0.2 0.5 0.8 0.9
effective dose levels:
[1] 0.1812546 0.2804774 0.3887513 0.4586600 0.5363481
```

Using the argument `type`, it is possible to treat continuous data like frequencies. Default value is `cont`, but the value `prob` allows to compute $0 < \alpha < 1$ quantiles for continuous data.
For objects of the class `fitED`, it is possible to compute an Akaike's Information Criterion (aic), but only if the slot `fitold` is assigned with a `list` or `locfit` object. In this cases, the function `aic.ED.locfit` computes the Akaike's Information Criterion in a similar way as the function `aic` in the `locfit` package. This value gives a rough impression of the goodness of fit. The function `aic.ED.locfit` can help to adjust the arguments of the function `ED` and to define the local linear estimate.
The function `Boot.CI` is applied in a similar manner as the function `ED`, but the confidence bands are computed for only one α value per time. Furthermore the confidence level can be specified using the argument `level`. The argument `R` fixes the number of Bootstrap replications. The default value is R=100. Since the accuracy of the Bootstrap confidence interval might be problematic, the function `Boot.CI` gives a warning if the the estimate of the dose-response function is too flat. See the help page of `Boot.CI` for an example.

4 Simulation study about the accuracy of the Bootstrap confidence intervals

In this section, we present a small simulation study about the accuracy of the Bootstrap confidence intervals. A binary response model is considered with 8 different shapes for the dose-response function p. In particular, we investigate the functions

$$p_1(x) = \Phi\left(\frac{x-\mu}{\sigma}\right), \quad \mu = .5, \sigma = .5 \tag{3}$$

$$p_2(x) = \Phi\left(\frac{x-\mu}{\sigma}\right), \quad \mu = .5, \sigma = .1 \tag{4}$$

$$p_3(x) = 1 - \exp\{-x^\gamma\}, \gamma = .52876 \tag{5}$$

$$p_4(x) = \eta\Phi\left(\frac{x-\mu_1}{\tau}\right) + (1-\eta)\Phi\left(\frac{x-\mu_2}{\tau}\right), \quad \mu_1 = 0.4, \mu_2 = 1.0, \eta = .64946, \tau = .13546 \tag{6}$$

$$p_5(x) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x-\mu}{\sigma}\right), \quad \mu = 0.15, \sigma = 0.05 \tag{7}$$

$$p_6'(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}(1-x)^{\beta-1}x^{\alpha-1}, \qquad \alpha = 2, \beta = 3. \tag{8}$$

$$p_7(x) = (1+\exp(5-15x))^{-1} \tag{9}$$

$$p_8(x) = \begin{cases} 2x & \text{if } 0 \le x \le 0.3 \\ 0.4x + 0.48 & \text{if } 0.3 \le x \le 0.8 \\ x & \text{if } 0.8 \le x \le 1 \end{cases} \tag{10}$$

Exemplarily for the dose-response function p_1, we show how the simulation study can be done using the EffectiveDose Package. First, we simulate the data. Analyzing the different ways to apply the `EffectiveDose` package, the function `Boot.CI` is used to calculate the confidence intervals for 10 different α values for a `list`, `locfit`, and `locpoly` objects. At the end, we check for each interval if the true value is inside of the interval or not.

```
> m = 1000
> s = 10
> alpha = seq(0.1, 0.9, length.out = s)
> ybeob = function(x) {
+     p = pnorm(x, mean = 0.5, sd = 0.5)
+     y = rbinom(1, 1, prob = p)
```

```
+ }
> qinv = qnorm(alpha, mean = 0.5, sd = 0.5)
> res = matrix(nrow = m, ncol = s)
> res2 = matrix(nrow = m, ncol = s)
> res3 = matrix(nrow = m, ncol = s)
> for (r in 1:m) {
+     for (l in 1:s) {
+         x = seq(0, 1, length.out = 50)
+         n = length(x)
+         y = numeric(n)
+         for (i in 1:n) {
+             y[i] = ybeob(x[i])
+         }
+         fit = locfit(y ~ lp(x, deg = 1, h = 0.1, nn = 0))
+         fit2 = locpoly(x, y, degree = 1, bandwidth = 0.1)
+         W = Boot.CI(list(x, y), alpha = alpha[l], bandwidth = 0.1)
+         W2 = Boot.CI(fit, alpha = alpha[l], bandwidth = 0.1)
+         W3 = Boot.CI(fit2, alpha = alpha[l], bandwidth = 0.1)
+         res[r, l] = (qinv[l] > W@CI[1]) & (qinv[l] < W@CI[2])
+         res2[r, l] = (qinv[l] > W2@CI[1]) & (qinv[l] < W2@CI[2])
+         res3[r, l] = (qinv[l] > W3@CI[1]) & (qinv[l] < W3@CI[2])
+     }
+ }
> apply(res, 2, mean)
> apply(res2, 2, mean)
> apply(res3, 2, mean)
```

Acknowledgements The author would like to thank Christian Ritz and Holger Dette for helpful comments and suggestions for the implementation of this package.

References

H. Dette, K.F. Pilz, N. Neumeyer (2005). A note on nonparametric estimation of the effective doses in bioassay. J. Amer. Statist. Assoc., 100, no 470, 503-510.

H. Dette, R. Scheder (2008). Finite sample properties of nonparametric estimates of the effective dose in quantal bioassay. Technical report, Department of Mathematics.
http://www.ruhr-uni-bochum.de/mathematik3/preprint.htm

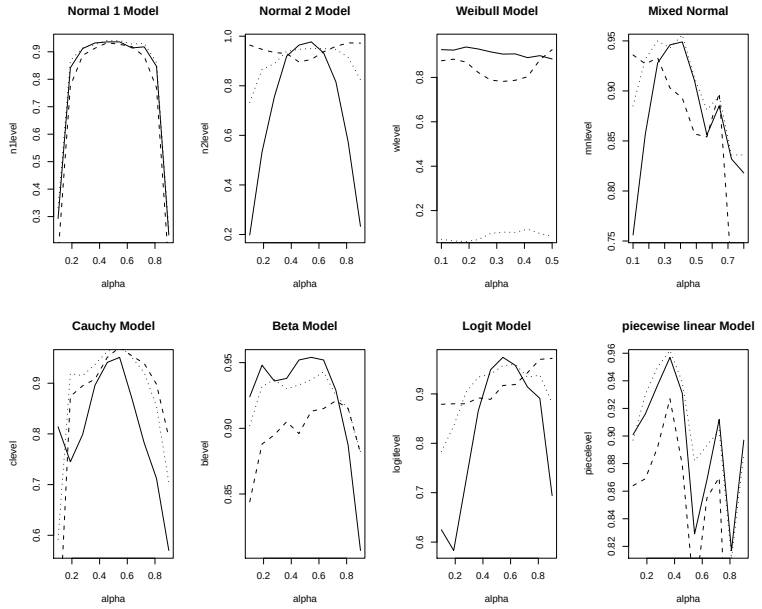

Figure 1: *Accuracy level for the Bootstrap confidence interval with confidence level .95 and 1000 simulation runs using the function* `Boot.CI` *for the dose-response functions (3)-(10). The straight line corresponds to the accuracy level of the fit with the list object, the dashed line to the locfit object, and the dotted line to the locpoly object.*