Normalised prediction distribution errors in R: the npde library

Emmanuelle Comets [1]★, **Karl Brendel** [2] and **France Mentré** [1,3]

[1] *INSERM U738, Paris, France; Université Paris 7, UFR de Médecine, Paris, France*
[2] *Institut de recherches internationales Servier, Courbevoie, France*
[3] *AP-HP, Hôpital Bichat, UF de Biostatistiques, Paris, France*

Objectives

Normalised prediction distribution errors (npde) are a relatively new metric designed to allow the evaluation of non-linear mixed effect models [1, 2]. In this poster we report the development of **npde**, an add-on package for R, the open source language and environment for statistical computing and graphics [3], for the computation of npde. In addition, the npde package also optionally computes prediction discrepancies (pd) [1], which ignore correlations within an individual due to repeated measurements.

Computing npde with the library

Statistical methods

The computation of the npde has been described in [2], improving on the pd described in [1]. Both papers are available from the authors on request. Briefly, prediction discrepancies are obtained as the quantile of each observation within its predicted distribution. A model describes the data well when the predicted discrepancies are evenly distributed. Prediction distribution errors are obtained after decorrelation of both the observations and the simulated Y with respect to the empirical mean and variance obtained in the simulations. They are then normalised by using the inverse of the cumulative density function of $\mathcal{N}(0,1)$.

By construction npde follow the $\mathcal{N}(0,1)$ distribution without any approximation and are uncorrelated within an individual. By construction, pd follow the $\mathcal{U}(0,1)$ distribution when each subject only contributes one observation; repeated observations results in an increase in the type I error of the test [1].

Package description

The program is distributed as a add-on package or library for the free statistical software R. A guide for the installation of R and add-on packages such as **npde** can be found on the CRAN (Comprehensive R Archive Network) at the following url: http://www.r-project.org/ R is available free of charge and runs on all operating systems.

Illustrative examples

Data

To illustrate the use of the package, we simulated data based on the well known toy dataset recording the pharmacokinetics of the anti-asthmatic drug theophylline. This data was collected by Upton in 12 subjects given a single oral dose of theophylline who then contributed 11 blood samples over a period of 25 hours [4]. The data at t=0 was omitted from the dataset for all patients. The pharmacokinetic model was a one-compartment model with first-order absorption, and the variability was modelled using an exponential model for the interindividual variability and a combined error model for the residual variability.

We then simulated two datasets with the design of the real dataset. V_{true} was simulated using the parameters estimated with NONMEM using the FOCE INTERACTION estimation method. V_{false} was simulated assuming a bioavailability divided by 2 (corresponding to $V_c/F=0.92$ L). These datasets are stored in two files called respectively **vtrue.dat** and **vfalse.dat**. Figure 1 show plots of the concentration versus time profiles for the two datasets.
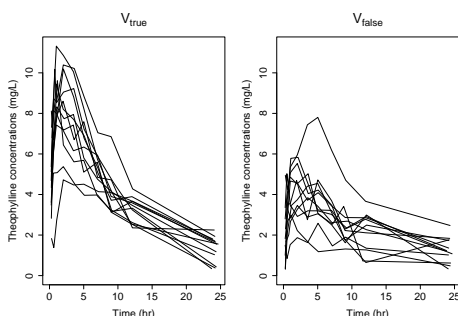

Figure 1: *spaghetti plots of the concentration versus time data for the two simulated datasets V_{true} (left) and V_{false} (right).*

In the following, the observed data will refer to either V_{true} or V_{false}. The observed data file must contain at least the following three columns: **id** (patient identification), **xobs** (independent variable such as time, X, ...), **yobs** (dependent variable such as DV, concentrations, effects...). Additional columns may be present but will not be used by the package.

Simulation setup

The package does not perform the simulations. The user must provide a file containing the K simulated datasets stacked one after the other. Within each simulated dataset, the order of the observations must be the same as within the observed dataset.

Simulations to compute the npde were performed using NONMEM. The control file used for the estimation was modified to set the values of the parameters (PK parameters, variability and error model) to those estimated, and the number of simulations was set to $K = 2000$. The simulated data were saved to a file called simdata.dat. The same set of simulated data was used for both datasets.

Note: The documentation shipped with the add-on package includes the NONMEM control files used to estimate parameters and to generate the simulated data, as well as detailed explanations.

Results

Computing npde for V_{true}

The function **npde** was used to compute the npde for the simulated dataset V_{true}, and the results were redirected to the R object **myres** with the following command:

```
myres<-npde()
```

Figure 2 shows the questions (in black) answered by the user (in green).

```
Name of the file containing the observed data: vtrue.dat
I'm assuming vtrue.dat has the following structure:
        ID X Y ...
To keep, press ENTER, to change, type any letter: n
        Column with ID information ? 1
        Column with X (eg time) information ? 3
        Column with Y (eg DV) information ? 4
Name of the file containing the simulated data: simdata.dat
Do you want results and graphs to be saved to files (y/Y) [default=yes] ? y
Different formats of graphs are possible:
        1. Postscript (extension eps)
        2. JPEG (extension jpeg)
        3. PNG (extension png)
        4. Acrobat PDF (extension pdf)
Which format would you like for the graph (1-4) ? 1
Name of the file (extension will be added, default=output): vtrue
Do you want to compute npde (y/Y) [default=yes] ? y
Do you want to compute pd (y/Y) [default=no] ? y
Do you want a message printed as the computation of npde
begins in a new subject (y/Y) [default=no] ? n
Do you want the function to return an object (y/Y) [default=yes] ? y
```

Figure 2: *using the interactive function npde to compute npde for V_{true}.*

The function produces the following results:

- an on-screen output (figure 3): first moments of the distribution of the npde as well as the results of 3 tests: (i) a t-test, to compare the mean of the npde to 0, (ii) a χ^2 test, to compare the variance of the npde to 1, and (iii) a Shapiro-Wilks test, to compare the distribution of the npde to a normal distribution. An adjusted p-value (equal to 3 times the minimum of the 3 previous p-values, as per a Bonferroni correction) is also shown when one of the tests is significant to provide a global test of the distribution

```
Computing npde
Saving graphs in file vtrue.eps
-------------------------------------------
Distribution of npde:
            mean= -0.09442
        variance= 1.006
        skewness= -0.1048
        kurtosis= -0.1783
-------------------------------------------

Statistical tests
    Wilcoxon signed rank test : 0.35
    Fisher variance test      : 0.931
    SW test of normality      : 0.839
Global adjusted p-value       : 1
-------------------------------------------
Computing pd
Saving results in file vtrue.npde
```

Figure 3: *output of the function npde applied to dataset V_{true}.*

- an R object (here, **myres**) containing the following elements is returned as the value of the function: (i) a data frame **obsdat** containing the observed data; (ii) ydobs: the decorrelated observed data; (iii) ydsim: the decorrelated simulated data; (iv) ypred: the empirical mean of the simulated predicted distribution for each observation; (v) xerr: an integer (0 if no error occurred during the computation); (vi) npde: the normalised prediction distribution errors; (vii) pd: the prediction discrepancies
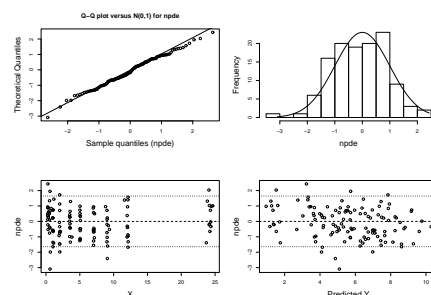- a plot illustrating the npde (shown in figure 4): this plot is also saved to disk

Figure 4: *graphs plotted by the package, for V_{true}. Quantile-quantile plot of the npde versus the expected standard normal distribution (upper left). Histogram of the npde with the density of the standard normal distribution overlayed (upper right). Scatterplot of npde versus observed X (lower left). Scatterplot of npde versus predicted Y (lower right).*

The results are saved in a text file with the following columns: id (patient ID), xobs (observed X), ypred (predicted Y), npde, pd. The name of the file is the same as the name of the file in which graphs are saved, with the extension .npde (here, vtrue.npde).

Options are available to prevent the numerical results and graphs to be saved to disk, and to prevent the function from returning a value.

Computing npde for V_{false}

A non-interactive function called **autonpde** is also available. Here we use it to compute the npde for the second dataset, V_{false}, setting the parameters as arguments to the function with the following command:

```
myres<-autonpde("vfalse.dat","simdata.dat",1,3,4,
    namesav="vfalse",calc.pd=T)
```

yielding the following results:

```
Computing npde
Saving graphs in file vfalse.eps
-------------------------------------------
Distribution of npde:
            mean= -0.4525
        variance= 1.748
        skewness= 0.3359
        kurtosis= -0.4629
-------------------------------------------

Statistical tests
    Wilcoxon signed rank test : 0.000285 **
    Fisher variance test      : 1.65e-06 **
    SW test of normality      : 0.141
Global adjusted p-value       : 4.95e-06 **
-------------------------------------------
Computing pd
Saving results in file vfalse.npde
```

Figure 5: *output of the function npde applied to dataset V_{false}.*

The graphs in figure 6 clearly illustrate a shift of the distribution of the npde when compared to the expected $\mathcal{N}(0,1)$ distribution. Although the normality assumption is not rejected, the npde have a non-zero mean and a larger variance than expected (see tests above). The lower plots in figure 6, showing the npde versus time (left) and predicted concentrations (right), illustrate trends in the npde.
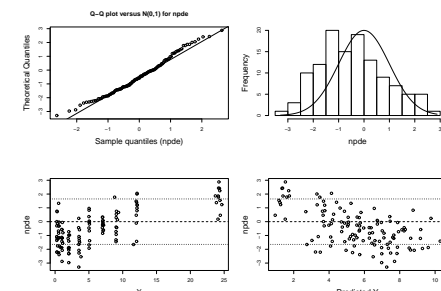

Figure 6: *graphs plotted by the package, for V_{false} (see legend of figure 4 for a description)*

Conclusion

Model evaluation is an important part of model building. Standardised prediction errors have been widely used until now but since they rely on model linearisation, they are intrinsically flawed. More sophisticated approaches include normalised prediction distribution errors which have better properties [2]. With the package **npde** we provide a tool to compute them easily, using the validation dataset and data simulated under the null hypothesis (model to be tested and design of the validation dataset). Default graphs and diagnostics are plotted to compare the distribution of npde to the theoretical distribution, thus checking model adequacy. Other diagnostic graphs can be plotted, against covariates for instance, using the npde returned by the package.

Perspectives

- Graphical interface
- Inclusion in other packages (Wings, Xpose?)

Download site: **www.npde.biostat.fr**

References

[1] F Mentré and S Escolano. Prediction discrepancies for the evaluation of nonlinear mixed-effects models. *J Pharmacokinet Biopharm*, 33:345–67, 2006.

[2] Karl Brendel, Emmanuelle Comets, Cline Laffont, Christian Laveille, and France Mentré. Metrics for external model evaluation with an application to the population pharmacokinetics of gliclazide. *Pharm Res*, 23:2036–49, 2006.

[3] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006.

[4] A Boeckmann, L Sheiner, and S Beal. *NONMEM Version 5.1*. University of California, NONMEM Project Group, San Francisco, 1998.

Inserm   ★ Presenting author
email: emmanuelle.comets@bichat.inserm.fr   PARIS DIDEROT