ClusROC: An R Package for ROC Analysis in Three-Class Classification Problems for Clustered Data

by Duc-Khanh To, Gianfranco Adimari, and Monica Chiogna

Abstract This paper introduces an R package for ROC analysis in three-class classification problems, for clustered data in the presence of covariates, named ClusROC. The clustered data that we address have some hierarchical structure, i.e., dependent data deriving, for example, from longitudinal studies or repeated measurements. This package implements point and interval covariate-specific estimation of the true class fractions at a fixed pair of thresholds, the ROC surface, the volume under the ROC surface, and the optimal pairs of thresholds. We illustrate the usage of the implemented functions through two practical examples from different fields of research.

1 Introduction

In clinical studies, receiver operating characteristic (ROC) surface analysis is widely used to evaluate the accuracy of a diagnostic test (or biomarker) when there are three ordinal disease classes (or diagnostic groups). See Nakas (2014) for a comprehensive review. Although the clinical context is where ROC surface analysis finds its natural application, other contexts, such as economics and engineering, very often face the problem of evaluating the accuracy of a classifier.

Within the ROC surface analysis framework, the following quantities are typically objects of interest: ROC surface, VUS (volume under ROC surface), and optimal pair of thresholds. Statistical methods for evaluating such quantities have been widely discussed in the statistical literature. We cite, among others, papers by Nakas and Yiannoutsos (2004); Xiong et al. (2006); Nakas et al. (2010); Attwood et al. (2014); Bantis et al. (2017); To Duc et al. (2016). Moreover, on the Comprehensive R Archive Network (CRAN), there are several packages implementing estimation methods for ROC surface analysis, for instance, **trinROC** (Noll and Reinhard, 2022), **ThresholdROC** (Sara et al., 2021) and **bcROCsurface** (To, 2021).

Most of the existing methods for ROC surface analysis have focused on a standard setting, in which measurements on statistical units are realizations of independent random variables, and the diagnostic test, or, more broadly, the classifier, is not influenced by any covariate. In some studies, however, not only the classifier can be affected by some covariates that characterize the units themselves (e.g. age, gender), but statistical units can be enrolled in clusters (e.g., families, genotype, communities, etc.). When statistical units are drawn from clusters, they can no longer be treated as independent. Indeed, units from the same cluster are typically more similar to each other than they will be to statistical units from other clusters. Therefore, unobserved variables may induce statistical dependence between observations within clusters that may be uncaptured by covariates. For such kinds of clustered data, which have a hierarchical structure and are dependent, Xiong et al. (2018) proposed the use of a standard linear mixed-effects model (McCulloch and Searle, 2001) to account for clusters and covariates' effects on the classifier. Then, the authors developed an approach to estimate the VUS, under an assumption of normality. Based on the model in Xiong et al. (2018), To et al. (2022) developed an estimation procedure for the ROC surface and methods for choosing the optimal pair of thresholds. The authors also discussed a variant of their approach, based on the Box-Cox transformation, useful when the normality assumption is (not heavily) violated.

In this paper, we introduce our R package for ROC surface analysis with clustered data, named **ClusROC**. The package (with related details) is available on CRAN at `http://CRAN.Rproject.org/package=ClusROC`. In the package, we implement procedures for estimating the parameters of the models (with and without the Box-Cox transformation), and for making inferences about the ROC surface, and the optimal pair of thresholds, by following methods outlined in To et al. (2022). In addition, we also implement a procedure for estimating the VUS, as discussed in Xiong et al. (2018).

In the following sections, we first briefly present the reference model and the inferential procedures for ROC surface analysis with clustered data. Then, we describe the **ClusROC** package and illustrate its use through two real datasets. The last section provides a brief conclusion.

2 The ROC surface analysis for clustered data

Here we briefly review methods proposed by Xiong et al. (2018) and To et al. (2022). Details and theoretical results can be found in the original articles. For convenience, as in the quoted papers, we refer to some clinical studies in the presentation and use appropriate language for that area.

The models

Let Y be the diagnostic test result, on a continuous scale, and let Y_1, Y_2, Y_3 be the test results for subjects in classes 1, 2, and 3, respectively. We assume that higher values of test results are associated with higher severity of the disease, and the severity of the disease grows with the class (i.e., class 3 is the worst). Let $X_1, \dots, X_p$ be p covariates, possibly associated with the test Y.

Let c be the total number of clusters, randomly selected from the population. For the k-th cluster, $k = 1, \dots, c$, let n_{ki} be the total number of subjects belonging to class i, $i = 1, 2, 3$ and let $n_k = n_{k1} + n_{k2} + n_{k3}$ be the total sample size within the cluster. Note that n_{ki} might be equal to 0 for some clusters. The linear mixed-effects model for the clustering effect on the test result Y, as well as for covariates' effects, is written as follows (Xiong et al., 2018; To et al., 2022):

$$\begin{aligned} Y_1 &= \alpha_{k_1} + z_1^\top \boldsymbol{\beta}_1 + \varepsilon_1, \\ Y_2 &= \alpha_{k_2} + z_2^\top \boldsymbol{\beta}_2 + \varepsilon_2, \\ Y_3 &= \alpha_{k_3} + z_3^\top \boldsymbol{\beta}_3 + \varepsilon_3, \end{aligned} \tag{1}$$

where (Y_1, Y_2, Y_3) is a triplet of test scores from three randomly sampled subjects from the three disease classes, (k_1, k_2, k_3), $k_i \in \{1, \dots, c\}$, are cluster memberships indicating the clusters from which Y_1, Y_2, Y_3 are observed, $z_i = (1, x_{1i}, \dots, x_{pi})^\top$ are fixed (i.e., not random) covariates values, and $\boldsymbol{\beta}_i = (\beta_{0i}, \beta_{1i}, \dots, \beta_{pi})^\top$, $i = 1, 2, 3$, are vectors of parameters representing covariates effects. In model (1), α_k are random effects accounting for the presence of clusters, and ε_i are subject-level random errors. We assume that: (i) the random effects α_k and the subject-level random errors ε_i follow a normal distribution, i.e., $\alpha_k \sim \mathcal{N}(0, \sigma_c^2)$ and $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ with $i = 1, 2, 3$; (ii) $\alpha_1, \alpha_2, \dots, \alpha_c$ and $\varepsilon_1, \varepsilon_2, \varepsilon_3$ are all independent (see also, McCulloch and Searle, 2001).

Let $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \boldsymbol{\beta}_2^\top, \boldsymbol{\beta}_3^\top)^\top$ with $\boldsymbol{\beta}_i = (\beta_{0i}, \beta_{1i}, \dots, \beta_{pi})^\top$, and $\boldsymbol{\theta} = (\sigma_c, \sigma_1, \sigma_2, \sigma_3)^\top$ be the unknown parameters in model (1). By using a restricted (or residual) maximum likelihood (REML) estimation approach, a consistent estimator $\widehat{\gamma} = (\widehat{\boldsymbol{\beta}}^\top, \widehat{\boldsymbol{\theta}}^\top)^\top$ of $\gamma = (\boldsymbol{\beta}^\top, \boldsymbol{\theta}^\top)^\top$ can be obtained. Under some regularity conditions, we have $\widehat{\gamma} \,\dot\sim\, \mathcal{N}(\gamma, \boldsymbol{\Lambda})$. The asymptotic covariance matrix $\boldsymbol{\Lambda}$ can be consistently estimated by using the sandwich formula (Liang and Zeger, 1986; Kauermann and Carroll, 2001; Mancl and DeRouen, 2001).

In some practical situations, data distributions may be skewed and the normality assumption might be violated. For such cases, To et al. (2022) considered the application of Box-Cox transformation for linear mixed-effects models (Lipsitz et al., 2000; Gurka et al., 2006):

$$\begin{aligned} Y_1^{(\lambda)} &= \alpha_{k_1} + z_1^\top \boldsymbol{\beta}_1 + \varepsilon_1, \\ Y_2^{(\lambda)} &= \alpha_{k_2} + z_2^\top \boldsymbol{\beta}_2 + \varepsilon_2, \\ Y_3^{(\lambda)} &= \alpha_{k_3} + z_3^\top \boldsymbol{\beta}_3 + \varepsilon_3, \end{aligned} \tag{2}$$

where $Y_i^{(\lambda)}$ is the Box-Cox transformed response, $Y_i^{(\lambda)} = (Y_i^\lambda - 1)/\lambda$ if $\lambda \neq 0$ and $Y_i^{(\lambda)} = \log(Y_i)$ if $\lambda = 0$, with $i = 1, 2, 3$, $Y_i > 0$, and λ is the transformation parameter (Box and Cox, 1964). Assumptions about the random effects α_k and the subject-level random errors ε_i are the same as in model (1). To obtain $\widehat{\lambda}$ and the REML estimator $\widehat{\gamma}$, To et al. (2022) applied the method proposed by Gurka and Edwards (2011) which is based on the scaled Box-Cox transformation model (Gurka et al., 2006). The estimator of the variance-covariance matrix of the REML estimator $\widehat{\gamma}$ is obtained again by applying the sandwich formula.

ROC surface analysis

According to the model (1), at a given vector z of covariates' values, $Y_i \sim \mathcal{N}(z^\top \boldsymbol{\beta}_i, \sigma_c^2 + \sigma_i^2)$ with $z = (1, x_1, \dots, x_p)^\top$ and $i = 1, 2, 3$. To et al. (2022) further assume that $z^\top \boldsymbol{\beta}_1 < z^\top \boldsymbol{\beta}_2 < z^\top \boldsymbol{\beta}_3$, i.e., that the stochastic dominance for the three classes holds at z.

For given thresholds t_1 and t_2 ($t_1 < t_2$), the covariate-specific true class fractions (TCFs) are written as $\mathrm{TCF}_1(t_1; z) = \Phi\left(\frac{t_1 - z^\top \boldsymbol{\beta}_1}{\sqrt{\sigma_c^2+\sigma_1^2}}\right)$, $\mathrm{TCF}_2(t_1, t_2; z) = \Phi\left(\frac{t_2 - z^\top \boldsymbol{\beta}_2}{\sqrt{\sigma_c^2+\sigma_2^2}}\right) - \Phi\left(\frac{t_1 - z^\top \boldsymbol{\beta}_2}{\sqrt{\sigma_c^2+\sigma_2^2}}\right)$ and $\mathrm{TCF}_3(t_2; z) = 1 - \Phi\left(\frac{t_2 - z^\top \boldsymbol{\beta}_3}{\sqrt{\sigma_c^2+\sigma_3^2}}\right)$, where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. Moreover, by setting $\mathrm{TCF}_1(t_1; z) = p_1$ and $\mathrm{TCF}_3(t_2; z) = p_3$, the covariate-specific ROC surface can be defined as a function of (p_1, p_3), i.e.,

$$\begin{aligned}\mathrm{ROCs}(p_1, p_3; z) \quad &= \quad \Phi\left(\frac{\Phi^{-1}(1-p_3)\sqrt{\sigma_c^2+\sigma_3^2} + z^\top \boldsymbol{\beta}_3 - z^\top \boldsymbol{\beta}_2}{\sqrt{\sigma_c^2+\sigma_2^2}}\right) \\ &\quad - \Phi\left(\frac{\Phi^{-1}(p_1)\sqrt{\sigma_c^2+\sigma_1^2} + z^\top \boldsymbol{\beta}_1 - z^\top \boldsymbol{\beta}_2}{\sqrt{\sigma_c^2+\sigma_2^2}}\right), \end{aligned} \tag{3}$$

if $\Phi^{-1}(p_1) < \frac{\Phi^{-1}(1-p_3)\sqrt{\sigma_c^2+\sigma_3^2} + z^\top \boldsymbol{\beta}_3 - z^\top \boldsymbol{\beta}_1}{\sqrt{\sigma_c^2+\sigma_1^2}}$; otherwise, $\mathrm{ROCs}(p_1, p_3; z) = 0$.

Based on the above expressions, for clustered data, To et al. (2022) considered covariate-specific estimation of the TCFs at a fixed pair of thresholds (t_1, t_2) and estimation of the ROC surface for each pair (p_1, p_3). Moreover, the authors proposed methods to estimate covariate-specific optimal pairs of thresholds (t_1^+, t_2^+) by considering three different criteria: (i) maximization of the covariate-specific generalized Youden index (GYI); (ii) minimization of the covariate-specific Euclidean distance between the ideal point $(1, 1, 1)$ and the point $(\mathrm{TCF}_1(t_1; z), \mathrm{TCF}_2(t_1, t_2; z), \mathrm{TCF}_3(t_2; z))$ (CtP); (iii) maximization of the covariate-specific volume of the cuboid under the covariate-specific ROC surface (MV). Resulting estimators are shown to be consistent and asymptotically normal, with asymptotic covariance matrices estimable by using the plug-in method. The normal approximation results can be used to construct suitable (joint) confidence regions.

Starting from model (1), covariate-specific inference on the VUS is discussed in Xiong et al. (2018), where maximum likelihood (ML) methods for point and interval estimation are proposed. In particular, Xiong et al. (2018) showed that covariate-specific ML VUS estimator $\hat{\theta}(z)$ is the summation of five components: $\hat{\theta}_1(z) = \widehat{\Pr}(Y_1 < Y_2 < Y_3, k_1 = k_2 = k_3 | z)$, $\hat{\theta}_2(z) = \widehat{\Pr}(Y_1 < Y_2 < Y_3, k_1 = k_2 \neq k_3 | z)$, $\hat{\theta}_3(z) = \widehat{\Pr}(Y_1 < Y_2 < Y_3, k_1 = k_3 \neq k_2 | z)$, $\hat{\theta}_4(z) = \widehat{\Pr}(Y_1 < Y_2 < Y_3, k_1 \neq k_2 = k_3 | z)$, and $\hat{\theta}_5(z) = \widehat{\Pr}(Y_1 < Y_2 < Y_3, k_1 \neq k_2 \neq k_3 | z)$.

In cases where the assumption of normality for Y_1, Y_2 and Y_3 is unrealistic, inferential procedures for covariate-specific TCFs, ROC surface, optimal pair of thresholds (and VUS) can be obtained starting from model (2); see Section 3.3 in To et al. (2022).

3 Overview of R package ClusROC

The **ClusROC** package implements techniques for ROC surface analysis, in case of clustered data and the presence of covariates. The package comprises five major functions:

- `clus_lme()`: This function fits the linear mixed-effects model (1) under the normality assumption, or the model (2) when the Box-Cox transformation is used.
- `clus_roc_surface()`: This function estimates and makes a 3D plot of covariate-specific ROC surfaces.
- `clus_opt_thres3()`: This function estimates the covariate-specific optimal pair of thresholds.
- `clus_vus()`: This function estimates the covariate-specific VUS.
- `clus_tcfs()`: This function estimates the covariate-specific TFCs at a specified pair of thresholds.

The **ClusROC** package can be installed directly from CRAN by using the code below:

```
install.packages("ClusROC")
```

Also, it can be installed from GitHub (“toduckhanh/ClusROC”), by using the function `install_github()` from **devtools** package:

```
library(devtools)
install_github("toduckhanh/ClusROC")
```

Description

The REML estimation in function `clus_lme()` is based on the function `lme()` of package **nlme** (Pinheiro et al., 2022). The function `clus_lme()` needs, firstly, specification of a `fixed_formula` which is a two-sided linear formula object describing the fixed-effects part of the model for three classes, with the response on the left of ~ operator and the terms, separated by + operators, on the right. Secondly, the arguments `name_class` and `name_clust` are needed to specify the name of the variables indicating the disease classes (or diagnostic groups) and the clusters in the data, respectively. To enable the Box-Cox transformation, users need to set the argument `boxcox` as `TRUE`. The Box-Cox parameter λ is estimated by a grid search on the interval $(-2, 2)$. This interval is suggested by Gurka and Edwards (2011), but users can change this range by setting the argument `interval_lambda`. Before fitting the model, `clus_lme()` determines the ordering of the disease groups based on the average values of test results in each disease group. If an ordering is provided by the user via the argument `levl_class`, the ordering of the mean values is still obtained to confirm the input ordering. In case of disagreement between the two orderings, the one based on the averages of test results is adopted. The function `plot()` provides three diagnostic plots for the model fitted by `clus_lme()`, namely, a Q-Q plot for residuals, a Fitted vs. Residuals plot, and a Q-Q plot for cluster effects. These plots exploit the **ggplot2** package (Hadley et al., 2022).

The functions `clus_roc_surface()`, `clus_opt_thres3()`, `clus_vus()` and `clus_tcfs()` are the main functions to perform the ROC surface analysis for clustered data. All of them require the output of `clus_lme()` as an argument. When one of the above functions is called, a check on the monotone ordering assumption is performed. That is, for a given value of the covariates, say z, the three predicted mean values of the test results in the three diagnostic groups, i.e., $z^\top \widehat{\boldsymbol{\beta}}_1$, $z^\top \widehat{\boldsymbol{\beta}}_2$ and $z^\top \widehat{\boldsymbol{\beta}}_3$, are computed and compared. If the assumption ($z^\top \widehat{\boldsymbol{\beta}}_1 < z^\top \widehat{\boldsymbol{\beta}}_2 < z^\top \widehat{\boldsymbol{\beta}}_3$) is not met, the ROC surface analysis is not performed at z.

The function `clus_roc_surface()` estimates a covariate-specific ROC surface at a single point for covariates and makes a 3D plot to display the estimated covariate-specific ROC surface by using **rgl** package (Duncan et al., 2021). This function also allows plotting an ellipsoidal confidence region for TCFs at a given pair of thresholds, in the ROC surface space. If the constructed confidence region is outside the unit cube, a probit transformation (Bantis et al., 2017) is automatically applied to obtain an appropriate confidence region, which is inside the unit cube.

The function `clus_opt_thres3()` gives the estimated covariate-specific optimal pair of thresholds as defined by the criteria GYI, CtP, and MV at multiple points of the covariates. The optimization is done by using the function `optim()` with the "L-BFGS-B" method; however, users can select other optimization methods, such as, "BFGS" or "Nelder-Mead". The function returns also the estimated asymptotic variance-covariance matrix of the (estimated) covariate-specific optimal pair of thresholds. Under the normality assumption, the asymptotic variance-covariance matrix is estimated by the Delta method. If the Box-Cox transformation is applied, a nonparametric bootstrap procedure for clustered data is used to estimate the asymptotic variance-covariance matrix. To speed up the bootstrap procedure, users can set the argument `parallel` as `TRUE` to enable parallel computing support. Users can also select the number of CPUs needed for the computation. After calling `clus_opt_thres3()`, the function `plot()` can be used to display confidence regions (and point estimates) of covariate-specific optimal pairs of thresholds.

The function `clus_vus()` estimates the covariate-specific VUS at multiple points of the covariates by using the `integrate()` routine. This function also performs the statistical test for the null hypothesis $H_0 : VUS = 1/6$ versus the alternative $H_A : VUS > 1/6$. This statistical test is a formal assessment of the adequacy of a diagnostic test in a three-class classification problem via its VUS at given covariates values. After calling `clus_vus()`, users can apply `ci_clus_vus()` to obtain confidence intervals for covariate-specific VUS. Three types of confidence intervals are computed: normal approximation-based; after logit and probit transformations.

The function `clus_tcfs()` estimates covariate-specific TCFs at a specified pair of thresholds, given one or multiple points for the covariates. This function also implements the Delta method to estimate the asymptotic variance-covariance matrix of the estimated covariate-specific TCFs. Note that, if the Box-Cox transformation is applied for the linear mixed-effects model, the pair of threshold values must be provided in the original scale.

Applications

To illustrate the use of the **ClusROC** package, we provide two examples with the `MouseNeurons` dataset and `EnergyEthiopia`, which are included in the package.

The glutamatergic neurons

To et al. (2022) used the MouseNeurons dataset to evaluate the ability of the Lysosomal Associated Membrane Protein Family Member 5 (Lamp5) gene to discriminate three types of glutamatergic neurons, namely Layer 2/3 Intratelencephalic (L2/3 IT), Layer 4 (L4) and Layer 5 Pyramidal Tract (L5 PT) neurons. A full version of the data is publicly available at http://portal.brain-map.org/atlases-and-data/rnaseq/mouse-v1-and-alm-smart-seq.

This dataset includes 860 observations (brain cells) and the following variables: the expression of the Lamp5 gene (diagnostic test/biomarker), the mouse genotype (which yields 23 clusters), the class labels (L2/3 IT, L4, and L5 PT), and the sex and age (in days) of the mouse. Below, we illustrate the use of the **ClusROC** package using this data.

Step 1: Load library and data

```
> library(ClusROC)
> data("MouseNeurons")
```

The above code loads the **ClusROC** package and the MouseNeurons data into R.

Step 2: Model fitting

In this step, we fit a linear mixed-effects model with Lamp5_cpm as a response and age_days as a covariate, under the Box-Cox transformation, by using function clus_lme():

```
> out_md <- clus_lme(fixed_formula = Lamp5_cpm ~ age_days,
+                    name_class = "subclass_label", name_clust = "genotype_id",
+                    data = MouseNeurons, boxcox = TRUE)
The ordered levels of classes are specified by the order of
 averages of the test values for each class:
L4 < L5 PT < L2/3 IT
```

As, in this case, the rank-ordered nature of the biomarker concerning the classes is not given, the monotone ordering was specified by ordering the classes according to the rank of the biomarker's sample means in the three groups. The results are shown as follows.

```
> print(out_md)

CALL: clus_lme(fixed_formula = Lamp5_cpm ~ age_days, name_class = "subclass_label",
    name_clust = "genotype_id", data = MouseNeurons, boxcox = TRUE)

Coefficients:
                                   Est. Std.Error z-value p-value
subclass_labelL4                0.78770   5.95328   0.132 0.89474
subclass_labelL4:age_days       0.45039   0.18193   2.476 0.01330 *
subclass_labelL5 PT            34.30543  12.27946   2.794 0.00521 **
subclass_labelL5 PT:age_days    0.20995   0.10652   1.971 0.04872 *
subclass_labelL2/3 IT          49.34642  20.88162   2.363 0.01812 *
subclass_labelL2/3 IT:age_days  0.08991   0.08306   1.082 0.27907
sigma_c                         6.78582   4.15973      --      --
sigma_1                        15.02492   7.12012      --      --
sigma_2                        11.24066   5.70949      --      --
sigma_3                        11.14321   6.07808      --      --
lambda                          0.44565        --      --      --
ICC                             0.22848        --      --      --
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Number of observations: 860
Number of clusters: 23
Sample size within cluster:
     Min      Max  Average
  1.0000 330.0000  37.3913
Box-Cox transformation: TRUE
```

The diagnostic plots for the fitted model are obtained by the following command (see Figure 1):

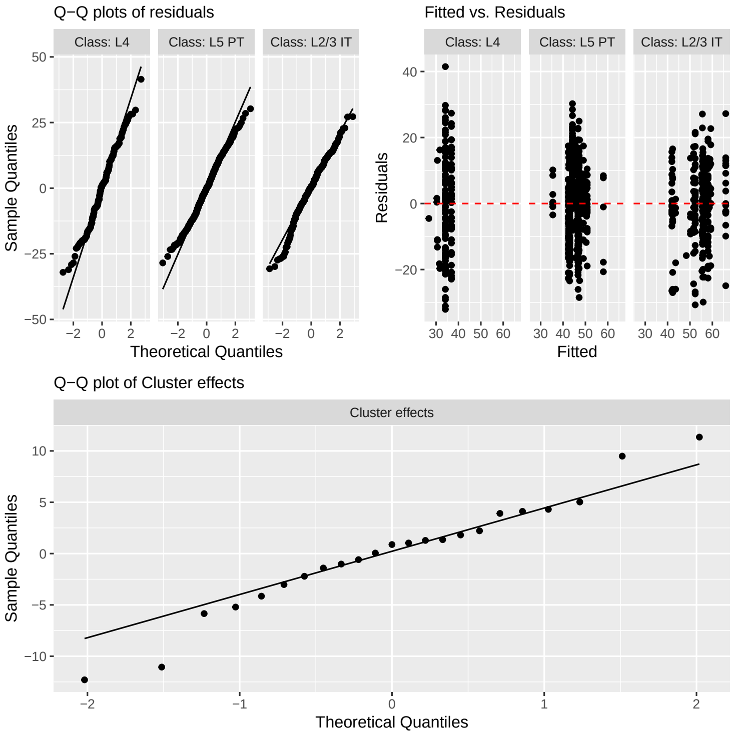


Figure 1: The diagnostic plots for the linear mixed-effects model.

```
> plot(out_md)
```

The plots suggest that the assumptions of normality and homogeneity of variances are satisfied. To help the reader evaluate the need for transforming the data, we also fitted the linear mixed-effects model without the Box-Cox transformation. The results are reported in Appendix, showing that the assumptions are violated.

Step 3: ROC surface analysis

After fitting the linear mixed-effects models of interest and verifying the normality assumption and the homogeneity of variances assumption, the ROC surface analysis can be performed. First, let us start with the estimation for covariate-specific VUS.

```
> out_vus <- clus_vus(out_clus_lme = out_md,
+                     newdata = data.frame(age_days = c(54, 60, 66)))
> print(out_vus)

CALL: clus_vus(out_clus_lme = out_md, newdata = data.frame(age_days = c(54,
    60, 66)))

Covariate-specific VUS:
 Covariates Values  Est. Std.Error z-value p-value
                54 0.541    0.0505    7.42  <0.001 ***
                60 0.514    0.0535    6.49  <0.001 ***
                66 0.485    0.0582    5.47  <0.001 ***
---
Signif. codes:   0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
z-value and p-value are for testing the null hypothesis H0: VUS = 1/6 vs HA: VUS > 1/6
```

In this case, we are interested in computing the (estimated) covariate-specific VUS at three different ages of mice, i.e., 54, 60, and 66 days. The 95% confidence intervals for covariate-specific VUS are obtained by:

```
> ci_clus_vus(out_vus)

The 95% confidence intervals for covariate-specific VUS:
   Covariates Values  Normal approximation  Logit transformation  Probit transformation
```

```
                 54       (0.442, 0.640)       (0.442, 0.637)        (0.442, 0.638)
                 60       (0.409, 0.618)       (0.410, 0.616)        (0.410, 0.617)
                 66       (0.371, 0.599)       (0.374, 0.598)        (0.373, 0.599)
```

As shown in the results, we can see three types of confidence intervals: normal approximation-based; after logit and probit transformations.

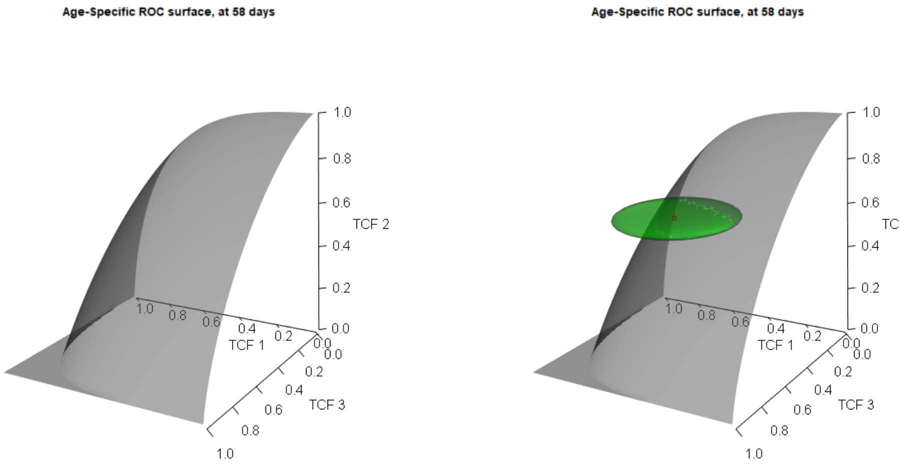
One obtains an estimate and a plot of the covariate-specific ROC surface, at, for example, the age of 58 days, as follows:

```
> clus_roc_surface(out_clus_lme = out_md, newdata = data.frame(age_days = 58),
+                  main = "Age-Specific ROC surface, at 58 days")
```

The result is displayed in Figure 2(a). A plot of a 95% ellipsoidal confidence region for TCFs at a fixed pair of threshold, for instance, $(t_1, t_2) = (350, 1350)$ is obtained as follows:

```
> clus_roc_surface(out_clus_lme = out_md, newdata = data.frame(age_days = 58),
+                  main = "Age-Specific ROC surface, at 58 days",
+                  ellips = TRUE, thresholds = c(350, 1350))
```

The 95% ellipsoidal confidence region for TCFs and covariate-specific ROC surface are displayed in 2(b), together.



(a) (b)

Figure 2: The plots of covariate-specific ROC surface at Age as 58 days: (a) without an ellipsoidal confidence region; (b) with an ellipsoidal confidence region for TCFs at $t_1 = 350$ and $t_2 = 1350$.

Estimates of covariate-specific TCFs at several values of the covariate, for a fixed pair of thresholds, are obtained by:

```
> clus_tcfs(out_clus_lme = out_md, newdata = data.frame(age_days = c(54, 58, 62)),
+           thresholds = c(350, 1350), ap_var = TRUE)

CALL: clus_tcfs(out_clus_lme = out_md, newdata = data.frame(age_days = c(54,
    58, 62)), thresholds = c(350, 1350), apVar = TRUE)

Covariate-specific TCFs at (350,1350) :
 Covariate(s) Values TCF 1 TCF 2 TCF 3 Se.TCF 1 Se.TCF 2 Se.TCF 3
                  54 0.576 0.632 0.522    0.113   0.0242   0.0630
                  58 0.533 0.620 0.533    0.119   0.0242   0.0604
                  62 0.490 0.607 0.544    0.124   0.0244   0.0592
```

The results consist of three-point estimates of covariate-specific TCFs at the fixed pair of thresholds (350, 1350), corresponding to three different values of age, 54, 58, and 62 (days), and the associated standard errors (Se). Note that, for the function `roc_surface()` and `clus_tcfs()`, if the Box-Cox transformation is applied, the pair of thresholds values must be provided in the original scale.

The following call performs the estimation of the covariate-specific optimal pair of thresholds, and the corresponding asymptotic variance-covariance matrices, at three different ages of mice: 55, 65 and 75 days. Here, we consider all criteria, i.e., GYI, CtP and MV.

```
> out_thresh <- clus_opt_thres3(method = c("GYI", "CtP", "MV"), out_clus_lme = out_md,
+                               newdata = data.frame(age_days = c(55, 65, 75)),
+                               ap_var = TRUE,
+                               control = list(n_boot = 1000, parallel = TRUE,
+                                              ncpus = 8))
```

In this case, a nonparametric bootstrap procedure for clustered data estimates the asymptotic variance-covariance matrix of the (estimated) covariate-specific optimal thresholds. Hence, we used a parallel computation with 8 CPUs (Windows 10 Pro 64-bit, Intel(R) Core(TM) i7-7700, 3.6 GHz) to speed up this process (the computation time for the parallel bootstrap process is about 15 minutes). As a general recommendation, we advise the user to switch to parallel computation not only in cases similar to the one tackled in this example but whenever the computational times appear to be too slow. The results are shown below.

```
> print(out_thresh)

CALL: clus_opt_thres3(method = c("GYI", "CtP", "MV"), out_clus_lme = out_md,
    newdata = data.frame(age_days = c(55, 65, 75)), ap_var = TRUE,
    control = list(n_boot = 1000, parallel = TRUE, ncpus = 8))

Covariate-specific optimal pair of thresholds:
 Covariate(s) Values                   Method Threshold 1 Threshold 2 TCF 1 TCF 2 TCF 3
                  55 Generalized Youden Index         530        1170 0.706 0.429 0.630
                  55    Closest to Perfection         446        1260 0.647 0.534 0.575
                  55               Max Volume         460        1260 0.658 0.525 0.575
                  65 Generalized Youden Index         627        1240 0.671 0.395 0.613
                  65    Closest to Perfection         538        1350 0.612 0.508 0.550
                  65               Max Volume         548        1350 0.619 0.502 0.550
                  75 Generalized Youden Index         731        1320 0.632 0.361 0.597
                  75    Closest to Perfection         639        1450 0.575 0.482 0.524
                  75               Max Volume         644        1450 0.578 0.480 0.523

Standard errors of Covariate-specific optimal pair of thresholds:
 Covariate(s) Values                   Method SE. Threshold 1 SE. Threshold 2
                  55 Generalized Youden Index            61.1           130.0
                  55    Closest to Perfection            73.8            85.7
                  55               Max Volume            66.7            85.8
                  65 Generalized Youden Index            98.6           120.0
                  65    Closest to Perfection           115.0            87.1
                  65               Max Volume            98.7            84.3
                  75 Generalized Youden Index           133.0           147.0
                  75    Closest to Perfection           146.0           140.0
                  75               Max Volume           126.0           121.0
```

The following call is needed to plot 95% confidence regions for the covariate-specific optimal pairs of thresholds, and the result is shown in Figure 3.

```
> plot(out_thresh, colors = c("forestgreen", "blue", "red"),
+      xlims = c(250, 1250), ylims = c(750, 1750), names.labels = "Age in days:",
+      size.point = 0.9)
```

House cooking fuel choice

We use the dataset EnergyEthiopia to illustrate the use of the package for evaluating a classifier in a three-class setting with panel data. The EnergyEthiopia data, included in the package, is a subset of the panel dataset discussed in Alem et al. (2016). The full dataset is collected in 4 cities of Ethiopia (Addis Ababa, Awassa, Dessie and Mekelle) within three different years: 2000, 2004 and 2009. Each household participated in the study at most three times (three years). For each time, the information on household energy choice and some covariates such as expenditure, demographic indicators, household size and educational status, are recorded. Alem et al. (2016) used the full dataset to investigate the determinants of household fuel cooking choice and energy transition in urban Ethiopia, based on a random-effects multinomial logistic model.

The dataset, named EnergyEthiopia, includes 2088 observations from 1123 households living in Addis Ababa and the following variables:

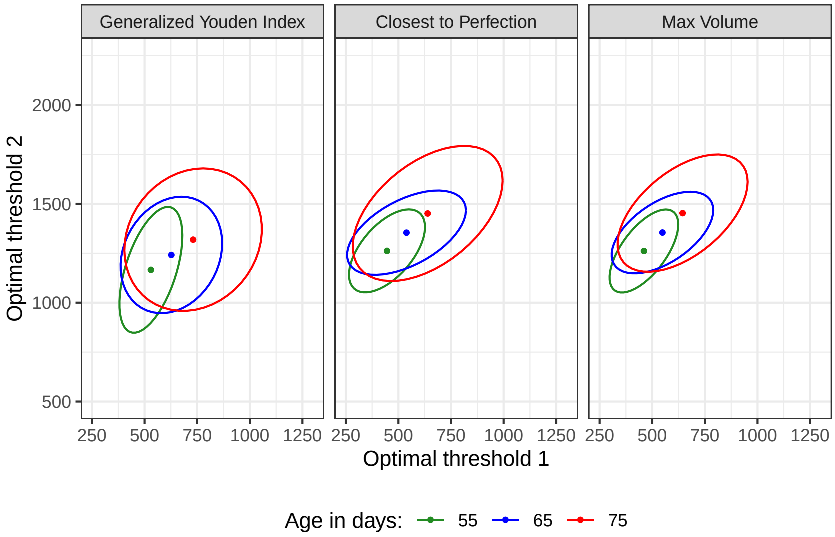


Figure 3: The 95% confidence regions for the age-specific optimal pairs of thresholds for three values of age: 55, 65, and 75.

- the cooking energy states of a household (energy2) with three categories: clean fuel only (denoted as 1), a mix of clean and biomass fuel (2) and biomass fuel only (1);
- log real consumption per adult equivalent units (lrconsaeu);
- the household size (hhs);
- the log of firewood price (lfirewood_pr);
- the log of charcoal price (lcharcol_pr);
- the log of kerosene price (lkerosene_pr);
- the log of electricity price (lelectric_pr).

We are interested in evaluating the ability of the *log real consumption per adult equivalent units* to discriminate three cooking energy states for a household, given the information provided by some covariates, such as the household size, the prices of firewood, charcoal and kerosene. Above, the term "clean fuel only" refers to clean energy sources such as electricity, gas and kerosene; whereas, "biomass fuel only" refers to energy sources such as firewood, charcoal, dung and crop residues. In our analysis, the 1123 households make up the clusters.

```
> library(ClusROC)
> data("EnergyEthiopia")
```

As the first step, we fit a linear mixed-effects model for the response variable lrconsaeu with the covariates: hhs_ft, lfirewood_pr, lcharcol_pr and lkerosene_pr. Here hhs_ft is a factor representing four levels of the household size: small ($1 \leq$ hhs ≤ 4), medium ($5 \leq$ hhs ≤ 8), large ($9 \leq$ hhs ≤ 12) and very large (hhs ≥ 13).

```
> out_md_enery <- clus_lme(
+   fixed_formula = lrconsaeu ~ hhs_ft + lfirewood_pr + lcharcol_pr + lkerosene_pr,
+   name_class = "energy2", name_clust = "uqid",
+   data = EnergyEthiopia, boxcox = FALSE
+ )
The ordered levels of classes are specified by the order of
 averages of the test values for each class:
3 < 2 < 1
```

As in the first application, we still have no information about the rank-ordered nature of the classifier (lrconsaeu) concerning the classes, so the monotone ordering was specified by ordering the classes according to the rank of the sample means of lrconsaeu inside the three groups. The results are shown as follows.

```
> print(out_md_enery)
```

```
CALL: clus_lme(fixed_formula = lrconsaeu ~ hhs_ft + lfirewood_pr +
    lcharcol_pr + lkerosene_pr, name_class = "energy2", name_clust = "uqid",
    data = EnergyEthiopia, boxcox = FALSE)

Coefficients:
                             Est. Std.Error z-value  p-value
energy23                  4.93925   0.11856  41.661  < 2e-16 ***
energy23:hhs_ftmedium    -0.48145   0.09126  -5.275 1.33e-07 ***
energy23:hhs_ftlarge     -0.77942   0.12139  -6.421 1.36e-10 ***
energy23:hhs_ftvery large -0.92139   0.14911  -6.179 6.44e-10 ***
energy23:lfirewood_pr    -0.09812   0.04806  -2.042  0.04119 *
energy23:lcharcol_pr     -0.01146   0.08398  -0.136  0.89150
energy23:lkerosene_pr    -0.13069   0.09731  -1.343  0.17927
energy22                  5.14705   0.08235  62.502  < 2e-16 ***
energy22:hhs_ftmedium    -0.39801   0.06730  -5.914 3.33e-09 ***
energy22:hhs_ftlarge     -0.60084   0.09475  -6.341 2.28e-10 ***
energy22:hhs_ftvery large -0.63877   0.18434  -3.465  0.00053 ***
energy22:lfirewood_pr     0.02983   0.04203   0.710  0.47793
energy22:lcharcol_pr     -0.10345   0.05570  -1.857  0.06329 .
energy22:lkerosene_pr    -0.24888   0.08953  -2.780  0.00544 **
energy21                  5.05581   0.05928  85.283  < 2e-16 ***
energy21:hhs_ftmedium    -0.40857   0.04612  -8.859  < 2e-16 ***
energy21:hhs_ftlarge     -0.57306   0.06944  -8.252  < 2e-16 ***
energy21:hhs_ftvery large -0.74813   0.17193  -4.351 1.35e-05 ***
energy21:lfirewood_pr     0.05490   0.02776   1.978  0.04793 *
energy21:lcharcol_pr     -0.11632   0.03957  -2.940  0.00328 **
energy21:lkerosene_pr     0.13104   0.06246   2.098  0.03590 *
sigma_c                   0.47835   0.02062      --       --
sigma_1                   0.55878   0.03409      --       --
sigma_2                   0.49135   0.02568      --       --
sigma_3                   0.57106   0.02103      --       --
ICC                       0.43933        --      --       --
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Number of observations: 2088
Number of clusters: 1123
Sample size within cluster:
     Min      Max  Average
1.000000 3.000000 1.859305
Box-Cox transformation: FALSE
```

The diagnostic plots shown in Figure 4 seem to suggest that the normality assumption and the homogeneity of variances are reliable.

```
> plot(out_md_enery)
```

We estimate the covariate-specific VUS at four different combinations of the covariates, when the value of hhs_ft changes from “small” to “very large”, the values of lfirewood_pr, lcharcol_pr and lkerosene_pr are fixed as 1, -1 and 2, respectively. Here, the values “1” and “2” of lfirewood_pr and lkerosene_pr refer to the very high price of the firewood, and the kerosene, respectively, while the value “-1” of lcharcol_pr refers to the very low price of the charcoal. The results of the covariate-specific VUS estimation procedure are displayed below.

```
> out_vus_enery <- clus_vus(
+   out_clus_lme = out_md_enery,
+   newdata = data.frame(hhs_ft = c("small", "medium", "large", "very large"),
+                        lfirewood_pr = c(1, 1, 1, 1),
+                        lcharcol_pr = c(-1, -1, -1, -1),
+                        lkerosene_pr = c(2, 2, 2, 2))
+ )
> print(out_vus_enery)

CALL: clus_vus(out_clus_lme = out_md_enery, newdata = data.frame(hhs_ft = c("small",
    "medium", "large", "very large"), lfirewood_pr = c(1, 1,
    1, 1), lcharcol_pr = c(-1, -1, -1, -1), lkerosene_pr = c(2,
```

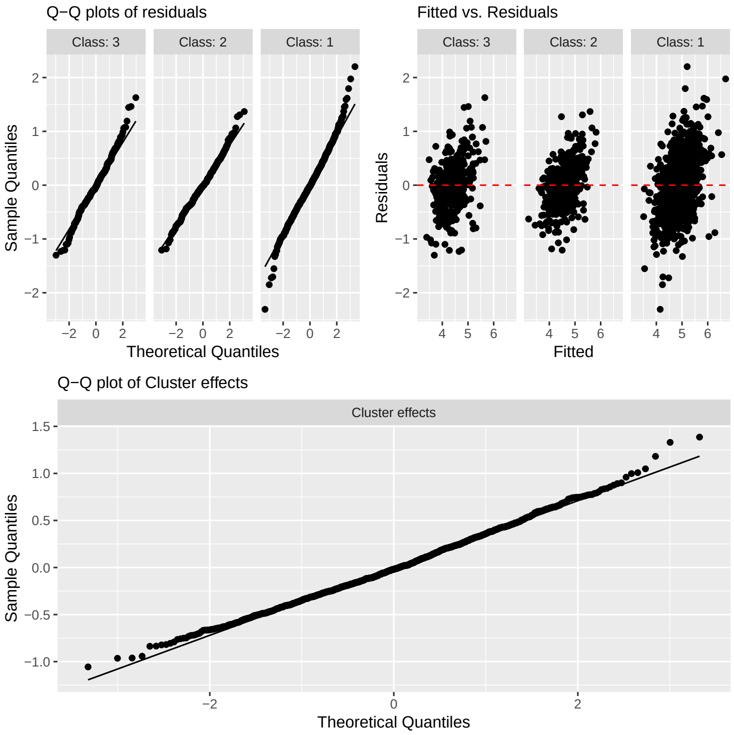


Figure 4: The diagnostic plots for the linear mixed-effects model (EnergyEthiopia data).

```
    2, 2, 2)))

Covariate-specific VUS:
      Covariates Values  Est. Std.Error z-value p-value
      (small, 1, -1, 2) 0.380    0.0674    3.17  <0.001 ***
     (medium, 1, -1, 2) 0.404    0.0639    3.72  <0.001 ***
      (large, 1, -1, 2) 0.443    0.0693    3.99  <0.001 ***
 (very large, 1, -1, 2) 0.440    0.0862    3.17  <0.001 ***
---
Signif. codes:    0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
z-value and p-value are for testing the null hypothesis H0: VUS = 1/6 vs HA: VUS > 1/6
```

The estimated covariate-specific VUSs suggest that the accuracy of lrconsaeu increases as the household size increases while the other covariates are fixed. The inference results for covariate-specific VUS indicate that the lrconsaeu can be considered as a classifier for distinguishing a household using biomass fuel only from others using either clean fuel only or a mixed fuel. However, its accuracy is not so high. We also compute the 95% confidence intervals for the covariate-specific VUS at four different points.

```
> ci_clus_vus(out_vus_enery)

The 95\% confidence intervals for covariate-specific VUS:
   Covariates Values  Normal approximation  Logit transformation  Probit transformation
    (small, 1, -1, 2)       (0.248, 0.512)       (0.259, 0.518)         (0.257, 0.517)
   (medium, 1, -1, 2)       (0.279, 0.529)       (0.287, 0.533)         (0.286, 0.532)
    (large, 1, -1, 2)       (0.307, 0.579)       (0.314, 0.580)         (0.313, 0.580)
(very large, 1, -1, 2)      (0.271, 0.609)       (0.284, 0.609)         (0.281, 0.609)
```

Since the covariate point ("large", 1, -1, 2) gives the highest covariate-specific VUS estimate among the considered points, we plot the covariate-specific ROC surface at this point, to visualize the covariate-specific TCFs at all possible pairs of thresholds. The result is displayed in Figure 5.

```
> clus_roc_surface(
+   out_clus_lme = out_md_enery,
+   newdata = data.frame(hhs_ft = "large", lfirewood_pr = 1,
+                        lcharcol_pr = -1, lkerosene_pr = 2),
+   file_name = "ROCS_Energy_ex1.png"
+ )
```

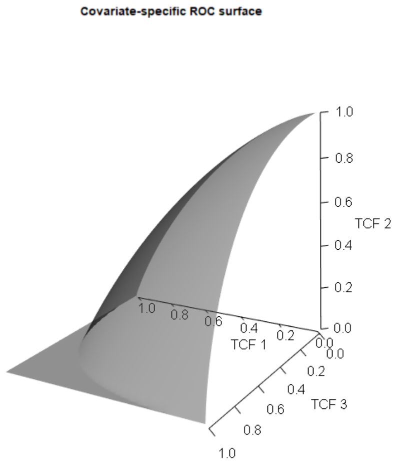


Figure 5: The plot of covariate-specific ROC surface at the covariate point `("large", 1, -1, 2)`.

Now, suppose we consider a pair of thresholds as $(t_1, t_2) = (3.75, 4.75)$. The results of the estimation of the covariate-specific TCFs at that pair, associated with four different covariate points, are obtained by:

```
> clus_tcfs(out_clus_lme = out_md_enery,
+           newdata = data.frame(
+             hhs_ft = c("small", "medium", "large", "very large"),
+             lfirewood_pr = c(1, 1, 1, 1), lcharcol_pr = c(-1, -1, -1, -1),
+             lkerosene_pr = c(2, 2, 2, 2)),
+           thresholds = c(3.75, 4.75), ap_var = TRUE)

CALL: clus_tcfs(out_clus_lme = out_md_enery, newdata = data.frame(hhs_ft = c("small",
    "medium", "large", "very large"), lfirewood_pr = c(1, 1,
    1, 1), lcharcol_pr = c(-1, -1, -1, -1), lkerosene_pr = c(2,
    2, 2, 2)), thresholds = c(3.75, 4.75), ap_var = TRUE)

Covariate-specific TCFs at (3.75,4.75) :
    Covariate(s) Values TCF 1 TCF 2 TCF 3 Se.TCF 1 Se.TCF 2 Se.TCF 3
      (small, 1, -1, 2) 0.126 0.415 0.839   0.0569   0.0617   0.0366
     (medium, 1, -1, 2) 0.312 0.526 0.671   0.0906   0.0247   0.0532
      (large, 1, -1, 2) 0.467 0.532 0.588   0.1100   0.0147   0.0612
 (very large, 1, -1, 2) 0.543 0.529 0.495   0.1230   0.0247   0.1080
```

Finally, we obtain the covariate-specific optimal pairs of thresholds at four different covariate points, based on all criteria, i.e., GYI, CtP and MV.

```
> out_thresh_enery <- clus_opt_thres3(
+   method = c("GYI", "CtP", "MV"), out_clus_lme = out_md_enery,
+   newdata = data.frame(hhs_ft = c("small", "medium", "large", "very large"),
+                        lfirewood_pr = c(1, 1, 1, 1),
+                        lcharcol_pr = c(-1, -1, -1, -1),
+                        lkerosene_pr = c(2, 2, 2, 2)),
+   ap_var = TRUE
+ )
> print(out_thresh_enery)

CALL: clus_opt_thres3(method = c("GYI", "CtP", "MV"), out_clus_lme = out_md_enery,
    newdata = data.frame(hhs_ft = c("small", "medium", "large",
        "very large"), lfirewood_pr = c(1, 1, 1, 1), lcharcol_pr = c(-1,
        -1, -1, -1), lkerosene_pr = c(2, 2, 2, 2)), ap_var = TRUE)

Covariate-specific optimal pair of thresholds:
    Covariate(s) Values                  Method Threshold 1 Threshold 2 TCF 1 TCF 2 TCF 3
      (small, 1, -1, 2) Generalized Youden Index        4.52        5.18 0.460 0.370 0.661
      (small, 1, -1, 2)     Closest to Perfection        4.51        5.35 0.455 0.453 0.572
      (small, 1, -1, 2)                Max Volume        4.51        5.35 0.454 0.452 0.575
     (medium, 1, -1, 2) Generalized Youden Index        4.13        4.78 0.509 0.363 0.657
     (medium, 1, -1, 2)     Closest to Perfection        4.08        4.94 0.481 0.465 0.574
     (medium, 1, -1, 2)                Max Volume        4.07        4.93 0.480 0.464 0.578
      (large, 1, -1, 2) Generalized Youden Index        3.91        4.59 0.553 0.379 0.669
      (large, 1, -1, 2)     Closest to Perfection        3.84        4.74 0.514 0.486 0.591
      (large, 1, -1, 2)                Max Volume        3.83        4.73 0.512 0.483 0.597
 (very large, 1, -1, 2) Generalized Youden Index        3.84        4.50 0.592 0.369 0.627
 (very large, 1, -1, 2)     Closest to Perfection        3.74        4.63 0.535 0.485 0.560
 (very large, 1, -1, 2)                Max Volume        3.74        4.63 0.535 0.483 0.562

Standard errors of Covariate-specific optimal pair of thresholds:
    Covariate(s) Values                  Method SE. Threshold 1 SE. Threshold 2
      (small, 1, -1, 2) Generalized Youden Index           0.234          0.1040
      (small, 1, -1, 2)     Closest to Perfection           0.112          0.0914
      (small, 1, -1, 2)                Max Volume           0.114          0.0932
     (medium, 1, -1, 2) Generalized Youden Index           0.166          0.0990
     (medium, 1, -1, 2)     Closest to Perfection           0.110          0.0868
     (medium, 1, -1, 2)                Max Volume           0.111          0.0891
      (large, 1, -1, 2) Generalized Youden Index           0.158          0.1090
      (large, 1, -1, 2)     Closest to Perfection           0.122          0.0954
      (large, 1, -1, 2)                Max Volume           0.123          0.0988
 (very large, 1, -1, 2) Generalized Youden Index           0.192          0.1640
 (very large, 1, -1, 2)     Closest to Perfection           0.154          0.1390
 (very large, 1, -1, 2)                Max Volume           0.154          0.1410
```

The 95

```
> plot(out_thresh_enery, colors = c("orange", "red", "blue", "forestgreen"),
+      file_name = "optThres_energy_ellip.pdf", nrow_legend = 2,
+      width = 6, height = 4)
```

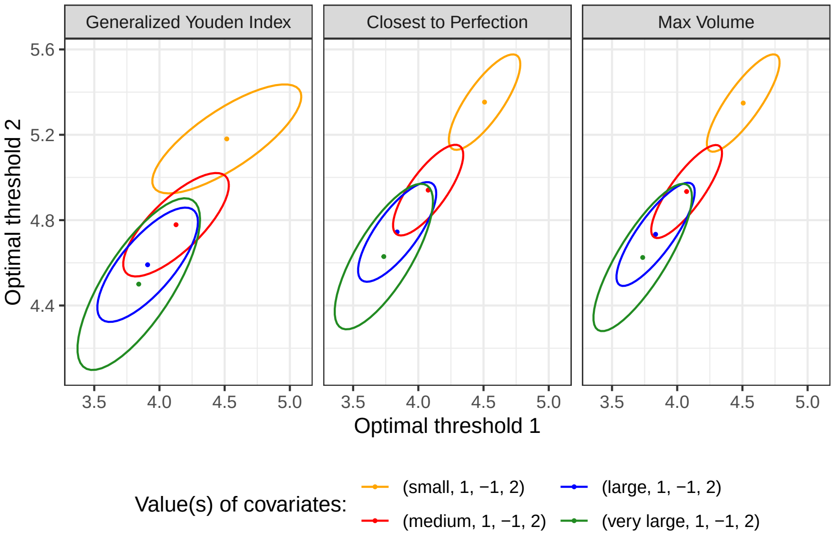


Figure 6: The 95% confidence regions for the age-specific optimal pairs of thresholds for four covariate points (in the house cooking fuel choice example).

4 Conclusion

This paper introduces the **ClusROC** package, the first R package for ROC surface analysis in three-class classification problems, for clustered data and in the presence of covariates. The package allows obtaining point estimates and confidence regions for true class fractions, ROC surface estimates and plots, point and interval estimates of VUS, and point estimates and confidence regions for optimal pair of thresholds. In the last case, three different criteria can be used: GYI, CtP, and MV. The package is available on Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org/package=ClusROC. The functions in the package are described and implemented using, as examples, the real datasets `MouseNeurons` and `EnergyEthiopia` provided in the package itself.

The linear mixed-effects model relies on the assumptions of normality and homoscedasticity. When such assumptions do not hold, the Box-Cox transformation can be employed. In these cases, a bootstrap procedure is needed to estimate the elliptical confidence regions for the optimal thresholds. This procedure can take a long computation time depending on the size of the data (either the number of clusters or the sample size within the clusters). For this reason, we recommend users enable the parallel computation option, which is already implemented within the function `clus_opt_thres3()`.

5 Acknowledgments

This research was supported by the Ministero dell'Istruzione, dell'Università e della Ricerca-Italy (grant number DIFO_ECCELLENZA18_01). The authors acknowledge the Editor and two anonymous reviewers whose valuable suggestions contributed to improving the presentation of the contents.

Bibliography

Y. Alem, A. D. Beyene, G. Köhlin, and A. Mekonnen. Modeling household cooking fuel choice: A panel multinomial logit approach. *Energy Economics*, 59:129–137, 2016. URL https://doi.org/10.1016/j.eneco.2016.06.025. [p261]

K. Attwood, L. Tian, and C. Xiong. Diagnostic thresholds with three ordinal groups. *Journal of Biopharmaceutical Statistics*, 24(3):608–633, 2014. URL https://doi.org/10.1080/10543406.2014.888437. [p254]

L. E. Bantis, C. T. Nakas, B. Reiser, D. Myall, and J. C. Dalrymple-Alford. Construction of joint confidence regions for the optimal true class fractions of Receiver Operating Characteristic (ROC) surfaces and manifolds. *Statistical Methods in Medical Research*, 26(3):1429–1442, 2017. URL https://doi.org/10.1177/0962280215581694. [p254, 257]

G. E. Box and D. R. Cox. An analysis of transformations (with discussion). *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–252, 1964. URL https://doi.org/10.1111/j.2517-6161.1964.tb00553.x. [p255]

M. Duncan, A. Daniel, N. Oleg, U. Simon, C. Ming, G. Albrecht, B. Ben, C. Gabor, S. Adam, S. Alexander, T. R. C. Team, E. Dirk, T. authors of Shiny, T. authors of knitr, O. Jeroen, D. Yohann, U. Joshua, i. M. Xavier, Fernandez, H. George, K. Ivan, and S. Michael. *rgl: 3D Visualization Using OpenGL*, 2021. URL https://CRAN.R-project.org/package=rgl. R package version 0.108.3. [p257]

M. J. Gurka and L. J. Edwards. Estimating variance components and random effects using the box-cox transformation in the linear mixed model. *Communications in Statistics—Theory and Methods*, 40(3):515–531, 2011. URL https://doi.org/10.1080/03610920903411192. [p255, 257]

M. J. Gurka, L. J. Edwards, K. E. Muller, and L. L. Kupper. Extending the Box–Cox transformation to the linear mixed model. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 169(2):273–288, 2006. URL https://doi.org/10.1111/j.1467-985X.2005.00391.x. [p255]

W. Hadley, C. Winston, H. Lionel, L. P. Thomas, T. Kohske, W. Claus, W. Kara, Y. Hiroaki, D. Dewey, and RStudio. *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*, 2022. URL https://CRAN.R-project.org/package=ggplot2. R package version 3.3.6. [p257]

G. Kauermann and R. J. Carroll. A note on the efficiency of sandwich covariance matrix estimation. *Journal of the American Statistical Association*, 96(456):1387–1396, 2001. URL https://doi.org/10.1198/016214501753382309. [p255]

K.-Y. Liang and S. L. Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22, 1986. URL https://doi.org/10.1093/biomet/73.1.13. [p255]

S. R. Lipsitz, J. Ibrahim, and G. Molenberghs. Using a Box–Cox transformation in the analysis of longitudinal data with incomplete responses. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 49(3):287–296, 2000. URL https://doi.org/10.1111/1467-9876.00192. [p255]

L. A. Mancl and T. A. DeRouen. A covariance estimator for GEE with improved small-sample properties. *Biometrics*, 57(1):126–134, 2001. URL https://doi.org/10.1111/j.0006-341X.2001.00126.x. [p255]

C. E. McCulloch and S. R. Searle. *Generalized, linear, and mixed Models*. USA: John Wiley & Sons, 2001. ISBN 0-471-19364-X. [p254, 255]

C. T. Nakas. Developments in ROC surface analysis and assessment of diagnostic markers in three-class classification problems. *REVSTAT-Statistical Journal*, 12(1):43–65, 2014. URL https://doi.org/10.57805/revstat.v12i1.143. [p254]

C. T. Nakas and C. T. Yiannoutsos. Ordered multiple-class ROC analysis with continuous measurements. *Statistics in Medicine*, 23(22):3437–3449, 2004. URL https://doi.org/10.1002/sim.1917. [p254]

C. T. Nakas, T. A. Alonzo, and C. T. Yiannoutsos. Accuracy and cut-off point selection in three-class classification problems using a generalization of the Youden index. *Statistics in Medicine*, 29(28): 2946–2955, 2010. URL https://doi.org/10.1002/sim.4044. [p254]

S. Noll and F. Reinhard. *trinROC: Statistical Tests for Assessing Trinormal ROC Data*, 2022. URL https://CRAN.R-project.org/package=trinROC. R package version 0.6. [p254]

J. Pinheiro, D. Bates, and R Core Team. *nlme: Linear and Nonlinear Mixed Effects Models*, 2022. URL https://CRAN.R-project.org/package=nlme. R package version 3.1-159. [p257]

P.-J. Sara, P. Natalia, and S. Konstantina. *ThresholdROC: Optimum Threshold Estimation*, 2021. URL https://CRAN.R-project.org/package=ThresholdROC. R package version 2.9.0. [p254]

D.-K. To. *bcROCsurface: Bias-Corrected Methods for Estimating the ROC Surface of Continuous Diagnostic Tests*, 2021. URL https://CRAN.R-project.org/package=bcROCsurface. R package version 1.0-5. [p254]

D.-K. To, G. Adimari, M. Chiogna, and D. Risso. Receiver operating characteristic estimation and threshold selection criteria in three-class classification problems for clustered data. *Statistical Methods in Medical Research*, 31(7):1325–1341, 2022. URL https://doi.org/10.1177/09622802221089029. [p254, 255, 256, 258]

K. To Duc, M. Chiogna, and G. Adimari. Bias–corrected methods for estimating the receiver operating characteristic surface of continuous diagnostic tests. *Electronic Journal of Statistics*, 10(2):3063–3113, 2016. URL https://doi.org/10.1214/16-EJS1202. [p254]

C. Xiong, G. van Belle, J. P. Miller, and J. C. Morris. Measuring and estimating diagnostic accuracy when there are three ordinal diagnostic groups. *Statistics in Medicine*, 25(7):1251–1273, 2006. URL https://doi.org/10.1002/sim.2433. [p254]

C. Xiong, J. Luo, L. Chen, F. Gao, J. Liu, G. Wang, R. Bateman, and J. C. Morris. Estimating diagnostic accuracy for clustered ordinal diagnostic groups in the three-class case–Application to the early diagnosis of Alzheimer disease. *Statistical Methods in Medical Research*, 27(3):701–714, 2018. URL https://doi.org/10.1177/0962280217742539. [p254, 255, 256]

6 Appendix

To help the reader evaluate the need for transforming the data, we report here the results of the analysis based on the linear mixed-effects model without the Box-Cox transformation. The results show that the assumptions of normality and homoscedasticity are violated.

```
> out_md_0 <- clus_lme(fixed_formula = Lamp5_cpm ~ age_days,
+                       name_class = "subclass_label", name_clust = "genotype_id",
+                       data = MouseNeurons)
The ordered levels of classes are specified by the order of
 averages of the test values for each class:
L4 < L5 PT < L2/3 IT
```

```
> print(out_md_0)

CALL: clus_lme(fixed_formula = Lamp5_cpm ~ age_days, name_class = "subclass_label",
    name_clust = "genotype_id", data = MouseNeurons)

Coefficients:
                                    Est. Std.Error z-value  p-value
subclass_labelL4               -378.9740  208.6613  -1.816   0.0693 .
subclass_labelL4:age_days        11.3966    1.4166   8.045 8.62e-16 ***
subclass_labelL5 PT             558.0245   89.6372   6.225 4.80e-10 ***
subclass_labelL5 PT:age_days      8.6915    0.6581  13.207  < 2e-16 ***
subclass_labelL2/3 IT          1227.4463  251.7121   4.876 1.08e-06 ***
subclass_labelL2/3 IT:age_days    5.0032    3.4950   1.432   0.1523
sigma_c                         330.1655   68.9777      --       --
sigma_1                         529.2924    9.1052      --       --
sigma_2                         514.0238   12.5033      --       --
sigma_3                         611.5288   30.7857      --       --
ICC                               0.2638        --      --       --
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Number of observations: 860
Number of clusters: 23
Sample size within cluster:
     Min      Max  Average
  1.0000 330.0000  37.3913
Box-Cox transformation: FALSE

> plot(out_md_0)
```

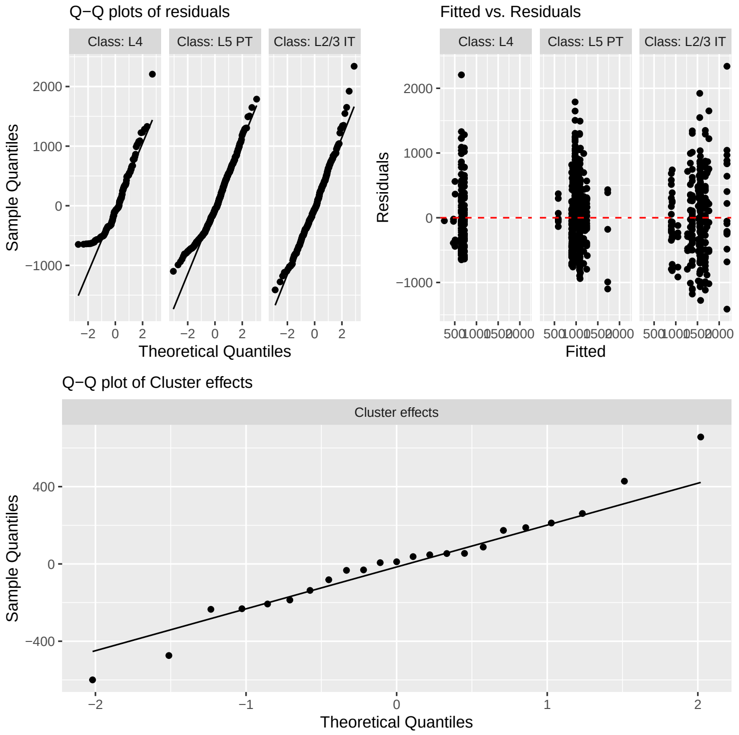


Figure 7: The diagnostic plots for the linear mixed-effects model for MouseNeurons data, without Box-Cox transformation.

Duc-Khanh To
University of Padova
Department of Statistical Sciences
Via C. Battisti, 241; I-35121 Padova, Italy

and
Department of Information and Engineering
Via Gradenigo, 6/b; 35131 Padova, Italy
ORCiD: 0000-0002-4641-0764
duckhanh.to@unipd.it

Gianfranco Adimari
University of Padova
Department of Statistical Sciences
Via C. Battisti, 241; I-35121 Padova, Italy
ORCiD: 0000-0002-7811-912X
gianfranco.adimari@unipd.it

Monica Chiogna
University of Bologna
Department of Statistical Sciences "Paolo Fortunati' '
Via Belle Arti, 41; 40126 Bologna, Italy
ORCiD: 0000-0002-7238-3739
monica.chiogna2@unibo.it