Package: ROCnReg (via r-universe)

October 22, 2026

Type Package

Title ROC Curve Inference with and without Covariates

Version 1.0-9

Date 2024-05-09

Imports stats, grDevices, graphics, splines, np, Matrix, moments, nor1mix, spatstat.univar, lattice, MASS, pbivnorm, parallel

Description Estimates the pooled (unadjusted) Receiver Operating Characteristic (ROC) curve, the covariate-adjusted ROC (AROC) curve, and the covariate-specific/conditional ROC (cROC) curve by different methods, both Bayesian and frequentist. Also, it provides functions to obtain ROC-based optimal cutpoints utilizing several criteria. Based on Erkanli, A. et al. (2006) <doi:10.1002/sim.2496>; Faraggi, D. (2003) <doi:10.1111/1467-9884.00350>; Gu, J. et al. (2008) <doi:10.1002/sim.3366>; Inacio de Carvalho, V. et al. (2013) <doi:10.1214/13-BA825>; Inacio de Carvalho, V., and Rodriguez-Alvarez, M.X. (2022) <doi:10.1214/21-STS839>; Janes, H., and Pepe, M.S. (2009) <doi:10.1093/biomet/asp002>; Pepe, M.S. (1998) <http://www.jstor.org/stable/2534001?seq=1>; Rodriguez-Alvarez, M.X. et al. (2011a) <doi:10.1016/j.csda.2010.07.018>; Rodriguez-Alvarez, M.X. et al. (2011a) <doi:10.1007/s11222-010-9184-1>. Please see Rodriguez-Alvarez, M.X. and Inacio, V. (2021) <doi:10.32614/RJ-2021-066> for more details.

License GPL

NeedsCompilation no

Author Maria Xose Rodriguez-Alvarez [aut, cre] (<https://orcid.org/0000-0002-1329-9238>), Vanda Inacio [aut] (<https://orcid.org/0000-0001-8084-1616>)

Maintainer Maria Xose Rodriguez-Alvarez <mxrodriguez@uvigo.es>

Config/pak/sysreqs make

Repository https://mxrodriguezuvigo.r-universe.dev

Date/Publication 2024-06-01 02:44:54 UTC

RemoteUrl https://github.com/cran/ROCnReg

RemoteRef HEAD

RemoteSha 9afa291b8ac265da36fb97043c032512020d502f

Contents

ROCnReg-package . . . 3
AROC.bnp . . . 4
AROC.kernel . . . 9
AROC.sp . . . 13
compute.threshold.AROC . . . 16
compute.threshold.cROC . . . 19
compute.threshold.pooledROC . . . 22
cROC.bnp . . . 25
cROC.kernel . . . 31
cROC.sp . . . 35
cROCData . . . 39
densitycontrol . . . 40
densitycontrol.aroc . . . 41
endosyn . . . 42
mcmccontrol . . . 44
pauccontrol . . . 45
plot.AROC . . . 46
plot.cROC . . . 47
plot.pooledROC . . . 48
pooledROC.BB . . . 49
pooledROC.dpm . . . 52
pooledROC.emp . . . 56
pooledROC.kernel . . . 59
predictive.checks . . . 63
print.AROC . . . 64
print.cROC . . . 65
print.pooledROC . . . 66
priorcontrol.bnp . . . 67
priorcontrol.dpm . . . 69
psa . . . 70
summary.AROC . . . 71
summary.cROC . . . 72
summary.pooledROC . . . 73

Index **75**

ROCnReg-package *ROC Curve Inference with and without Covariates*

Description

Estimates the pooled (unadjusted) Receiver Operating Characteristic (ROC) curve, the covariate-adjusted ROC (AROC) curve, and the covariate-specific/conditional ROC (cROC) curve by different methods, both Bayesian and frequentist. Also, it provides functions to obtain ROC-based optimal cutpoints utilizing several criteria. Based on Erkanli, A. et al. (2006) <doi:10.1002/sim.2496>; Faraggi, D. (2003) <doi:10.1111/1467-9884.00350>; Gu, J. et al. (2008) <doi:10.1002/sim.3366>; Inacio de Carvalho, V. et al. (2013) <doi:10.1214/13-BA825>; Inacio de Carvalho, V., and Rodriguez-Alvarez, M.X. (2022) <doi:10.1214/21-STS839>; Janes, H., and Pepe, M.S. (2009) <doi:10.1093/biomet/asp002>; Pepe, M.S. (1998) <http://www.jstor.org/stable/2534001?seq=1>; Rodriguez-Alvarez, M.X. et al. (2011a) <doi:10.1016/j.csda.2010.07.018>; Rodriguez-Alvarez, M.X. et al. (2011a) <doi:10.1007/s11222-010-9184-1>. Please see Rodriguez-Alvarez, M.X. and Inacio, V. (2021) <doi:10.32614/RJ-2021-066> for more details.

Details

Package:	ROCnReg
Type:	Package
Version:	1.0-9
Date:	2024-05-30
License:	GPL

Author(s)

Maria Xose Rodriguez-Alvarez and Vanda Inacio Maintainer: Maria Xose Rodriguez-Alvarez <mxrodriguez@uvigo.es>

References

Erkanli, A., Sung M., Jane Costello, E., and Angold, A. (2006). Bayesian semi-parametric ROC analysis. *Statistics in Medicine*, **25**, 3905–3928.

Faraggi, D. (2003). Adjusting receiver operating characteristic curves and related indices for covariates. *The Statistician* **52**, 179–192.

Gu, J., Ghosal, S., and Roy, A. (2008). Bayesian bootstrap estimation of ROC curve. *Statistics in Medicine*, **27**, 5407–5420.

Inacio de Carvalho, V., Jara, A., Hanson, T. E., and de Carvalho, M. (2013). Bayesian nonparametric ROC regression modeling. *Bayesian Analysis*, **8**, 623–646.

Inacio de Carvalho, V., and Rodriguez-Alvarez, M. X. (2022). The Covariate-Adjusted ROC Curve: The Concept and Its Importance, Review of Inferential Methods, and a New Bayesian Estimator. *Statistical Science*, **37**, 541 -561

Janes, H., and Pepe, M.S. (2009). Adjusting for covariate effects on classification accuracy using the covariate-adjusted receiver operating characteristic curve. Biometrika, **96**, 371–382.

Pepe, M.S. (1998). Three approaches to regression analysis of receiver operating characteristic curves for continuous test results. *Biometrics* **54**, 124–135.

Rodriguez-Alvarez, M. X. and Inacio, V., and (2021). ROCnReg: An R Package for Receiver Operating Characteristic Curve Inference with and without Covariate Information. *The R Journal*

Rodriguez-Alvarez, M.X., Tahoces, P. G., Cadarso-Suarez, C., and Lado, M.J. (2011). Comparative study of ROC regression techniques–Applications for the computer-aided diagnostic system in breast cancer detection. *Computational Statistics and Data Analysis*, **55**, 888–902.

Rodriguez-Alvarez, M.X., Roca-Pardinas, J., and Cadarso-Suarez, C. (2011). ROC curve and covariates: extending induced methodology to the non-parametric framework. *Statistics and Computing*, **21**, 483–499.

`AROC.bnp` *Nonparametric Bayesian inference of the covariate-adjusted ROC curve (AROC).*

Description

This function estimates the covariate-adjusted ROC curve (AROC) using the nonparametric Bayesian approach proposed by Inacio de Carvalho and Rodriguez-Alvarez (2018).

Usage

```
AROC.bnp(formula.h, group, tag.h, data, standardise = TRUE,
 p = seq(0, 1, l = 101), ci.level = 0.95, compute.lpml = FALSE, compute.WAIC = FALSE,
  compute.DIC = FALSE, pauc = pauccontrol(), density = densitycontrol.aroc(),
  prior.h = priorcontrol.bnp(), mcmc = mcmccontrol(),
  parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

`formula.h`	A `formula` object specifying the regression function associated to each component of the single-weights linear dependent Dirichlet process mixture model used to estimate the conditional distribution function of the diagnostic test outcome in the healthy population. Regarding the modelling of continuous covariates, both linear and nonlinear effects are allowed, with nonlinear effects being modelled through B-spline basis expansions (see Note).
`group`	A character string with the name of the variable that distinguishes healthy/nondiseased from diseased individuals.
`tag.h`	The value codifying healthy individuals in the variable `group`.
`data`	A data frame representing the data and containing all needed variables.
`standardise`	A logical value. If TRUE both the test outcomes and the continuous covariates assumed to have a linear effect are standardised (i.e., the resulting variables have mean zero and standard deviation of one). The default is TRUE.

p	Set of false positive fractions (FPF) at which to estimate the covariate-adjusted ROC curve.
ci.level	An integer value (between 0 and 1) specifying the level for the credible interval. The default is 0.95.
compute.lpml	A logical value. If TRUE, the log pseudo marginal likelihood (LPML, Geisser and Eddy, 1979) and the conditional predictive ordinates (CPO) are computed.
compute.WAIC	A logical value. If TRUE, the widely applicable information criterion (WAIC, Gelman et al., 2014; Watanabe, 2010) is computed.
compute.DIC	A logical value. If TRUE, the deviance information criterion (DIC)(Celeux et al., 2006, Spiegelhalter et al., 2002) is computed.
pauc	A list of control values to replace the default values returned by the function pauccontrol. This argument is used to indicate whether the partial area under the covariate-adjusted ROC curve (pAAUC) should be computed, and in case it is computed, whether the focus should be placed on restricted false positive fractions (FPFs) or on restricted true positive fractions (TPFs), and the upper bound for the FPF (if focus is FPF) or the lower bound for the TPF (if focus is TPF).
density	A list of control values to replace the default values returned by the function densitycontrol.aroc. This argument is used to indicate whether the conditional densities of the marker in the healthy population should be computed, and in case it is to be computed, at which grid of test outcomes the conditional densities should be evaluated, and at which covariate values they should be predicted.
prior.h	A list of control values to replace the default values returned by the function priorcontrol.bnp. See priorcontrol.bnp for details.
mcmc	A list of control values to replace the default values returned by the function mcmccontrol. See mcmccontrol for details.
parallel	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
ncpus	An integer with the number of processes to be used in parallel operation. Defaults to 1.
cl	An object inheriting from class cluster (from the parallel package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates the covariate-adjusted ROC curve (AROC) defined as

$$AROC\,(p) = Pr\{1 - F_{\bar{D}}(Y_D|\mathbf{X}_D) \leq p\},$$

where $F_{\bar{D}}(\cdot|\mathbf{X}_{\bar{D}})$ denotes the distribution function of $Y_{\bar{D}}$ conditional on the vector of covariates $\mathbf{X}_{\bar{D}}$.

The method implemented in this function combines a single-weights linear dependent Dirichlet process mixture model (De Iorio et al., 2009) to estimate $F_{\bar{D}}(\cdot|\mathbf{X}_{\bar{D}})$ and the Bayesian bootstrap (Rubin, 1981) to estimate the outside probability. More precisely, and letting $\{(\mathbf{x}_{\bar{D}i}, y_{\bar{D}i})\}_{i=1}^{n_{\bar{D}}}$

be a random sample from the nondiseased population, our postulated model for the conditional distribution function takes the following form

$$F_{\bar{D}}(y_{\bar{D}i}|\mathbf{X}_{\bar{D}} = \mathbf{x}_{\bar{D}i}) = \sum_{l=1}^{L} \omega_l \Phi(y_{\bar{D}i} \mid \mu_l(\mathbf{x}_{\bar{D}i}), \sigma_l^2),$$

where $\Phi(y|\mu, \sigma^2)$ denotes the cumulative distribution function of the normal distribution, evaluated at y, with mean μ and variance σ^2. The regression function $\mu_l(\mathbf{x}_{\bar{D}i})$ can incorporate both linear and nonlinear (through B-splines) effects of continuous covariates, categorical covariates (factors) as well as interactions. Interactions between categorical and (nonlinear) continuous covariates are also allowed (factor-by curve interactions). For the sake of simplicity we write $\mu_l(\mathbf{x}_{\bar{D}i}) = \mathbf{z}_{\bar{D}i}^T \beta_l$ ($l = 1, ..., L$), where $\mathbf{z}_{\bar{D}i}$ is the ith column of the design matrix (possibly containing a basis representation of some/all continuous covariates). Here L is a pre-specified upper bound on the number of mixture components. The ω_l's result from a truncated version of the stick-breaking construction ($\omega_1 = v_1$; $\omega_l = v_l \prod_{r<l}(1 - v_r)$, $l = 2, \ldots, L$; $v_1, \ldots, v_{L-1} \sim$ Beta $(1, \alpha)$; $v_L = 1$, $\alpha \sim \Gamma(a_\alpha, b_\alpha)$), $\beta_l \sim N_Q(\mathbf{m}, \mathbf{S})$, and $\sigma_l^{-2} \sim \Gamma(a, b)$. It is further assumed that $\mathbf{m} \sim N_Q(\mathbf{m}_0, \mathbf{S}_0)$ and $\mathbf{S}^{-1} \sim W(\nu, (\nu\Psi)^{-1})$. Here $\Gamma(a, b)$ denotes a Gamma distribution with shape parameter a and rate parameter b, $W(\nu, (\nu\Psi)^{-1})$ denotes a Wishart distribution with ν degrees of freedom and expectation Ψ^{-1}, and Q denotes the dimension of the vector $\mathbf{z}_{\bar{D}i}$. It is worth mentioning that when $L = 1$, the model for the conditional distribution of the test outcomes (in the healthy population) reduces to a normal regression model (where continuous covariates effects are modelled either parametrically or nonparametrically). For a detailed description, we refer to Inacio de Carvalho and Rodriguez-Alvarez (2018).

Regarding the area under the curve, we note that

$$AAUC = \int_0^1 AROC(p)dp = 1 - E\{U_D\},$$

where $U_D = 1 - F_{\bar{D}}(Y_D|\mathbf{X}_D)$. In our implementation, the expectation is computed using the Bayesian bootstrap (using the same weights as those used to estimate the AROC, see Inacio de Carvalho and Rodriguez-Alvarez (2018) for details). As far as the partial area under the curve is concerned, when `focus = "FPF"` and assuming an upper bound u_1 for the FPF, what it is computed is

$$pAAUC_{FPF}(u_1) = \int_0^{u_1} AROC(p)dp = u_1 - E\{U_{D,u_1}\},$$

where $U_{D,u_1} = min\{u_1, 1 - F_{\bar{D}}(Y_D|\mathbf{X}_D)\}$. Again, the expectation is computed using the Bayesian bootstrap. The returned value is the normalised pAAUC, $pAAUC_{FPF}(u_1)/u_1$ so that it ranges from $u_1/2$ (useless test) to 1 (perfect marker). Conversely, when `focus = "TPF"`, and assuming a lower bound for the TPF of u_2, the partial area corresponding to TPFs lying in the interval $(u_2, 1)$ is computed as

$$pAAUC_{TPF}(u_2) = \int_{AROC^{-1}(u_2)}^1 AROC(p)dp - \{1 - AROC^{-1}(u_2)\} \times u_2.$$

Here, the computation of the integral is done numerically. The returned value is the normalised pAAUC, $pAAUC_{TPF}(u_2)/(1 - u_2)$, so that it ranges from $(1 - u_2)/2$ (useless test) to 1 (perfect test).

Finally, it is important referring that with respect to the computation of the DIC, when $L = 1$, it is computed as in Spiegelhalter et al. (2002), and when $L > 1$, DIC3 as described in Celeux et al.

(2006) is computed. Also, for the computation of the conditional predictive ordinates (CPO) we follow the stable version proposed by Gelman et al. (2014).

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`data`	The original supplied data argument.
`missing.ind`	A logical value indicating whether for each pair of observations (test outcomes and covariates) missing values occur.
`marker`	The name of the diagnostic test variable in the dataframe.
`group`	The value of the argument `group` used in the call.
`tag.h`	The value of the argument `tag.h` used in the call.
`p`	Set of false positive fractions (FPF) at which the covariate-adjusted ROC curve (AROC) has been estimated.
`ci.level`	The value of the argument `ci.level` used in the call.
`prior`	A list returning the hyperparameter values.
`ROC`	Estimated covariate-adjusted ROC curve (AROC) (posterior mean) and `ci.level`*100% pointwise credible band.
`AUC`	Estimated area under the covariate-adjusted ROC curve (AAUC) (posterior mean), and `ci.level`*100% credible interval.
`pAUC`	If computed, estimated partial area under the covariate-adjusted ROC curve (pAAUC) (posterior mean) and `ci.level`*100% credible interval. Note that the returned values are normalised, so that the maximum value is one (see more on Details).
`newdata`	If `compute` is set to `TRUE` in the argument `density`, a data frame containing the values of the covariates at which the regression function and conditional densities were computed (see below).
`reg.fun.h`	If `compute` is set to `TRUE` in the argument `density`, a data frame containing the predicted regression function (posterior mean) and `ci.level`*100% pointwise credible band.
`dens`	If `compute` is set to `TRUE` in the argument `density`, a list with two components (only for the healthy population): `grid` (grid of test outcomes where the densities were evaluated) and `dens` (MCMC realisations of the corresponding conditional densities).
`lpml`	If computed, a list with two components: the log pseudo marginal likelihood (LPML) and the conditional predictive ordinates (CPO).
`WAIC`	If computed, widely applicable information criterion (WAIC) and associated complexity penalty (pW).
`DIC`	If computed, deviance information criterion (DIC) and associated complexity penalty (pD).

`fit` Results of the fitting process. A list with the following components: (1) `formula`: the value of the argument `formula.h` used in the call. (2) `mm`: information needed to construct the model design matrix associated with the single weights linear dependent Dirichlet process mixture model. (3) `beta`: array of dimension `nsavexLxQ` with the sampled regression coefficients. (4) `sd`: matrix of dimension `nsavexL` with the sampled variances. (4) `probs`: matrix of dimension `nsavexL` with the sampled components' weights. Here, `nsave` is the number of Gibbs sampler iterations saved, `L` is the upper bound on the number of mixture components, and `Q` is the dimension of vector $\mathbf{z}_{\bar{D}}$ (see also Details).

`data_model` List with the data used in the fit: observed diagnostic test outcome and design matrices, separately for the healthy and diseased groups.

Note

The input argument `formula.h` is similar to that used for the `glm` function, except that flexible specifications can be added by means of the function `f()`. For instance, specification $y \sim x1 + f(x2, K = 3)$ would assume a linear effect of `x1` (if `x1` continuous) and the effect of `x2` would be modeled using B-splines basis functions. The argument `K = 3` indicates that `3` internal knots will be used, with the quantiles of `x2` used for their location. Categorical variables (factors) can be also incorporated, as well as interaction terms. For example, to include the factor-by-curve interaction between `age` and `gender` we need to specify, e.g., $y \sim gender + f(age, by = gender, K = c(3, 5))$. Note that, in this case, the number of knots can be different for each level of the factor. The order of the vector `K` of knots should match the levels of the factor.

References

Celeux, G., Forbes, F., Robert C. P., and Titerrington, D. M. (2006). Deviance information criteria for missing data models. *Bayesian Analysis*, **1**, 651–674.

De Iorio, M., Johnson, W. O., Muller, P., and Rosner, G. L. (2009). Bayesian nonparametric nonproportional hazards survival modeling. *Biometrics*, **65**, 762–775.

Geisser, S. and Eddy, W.F. (1979) A Predictive Approach to Model Selection. *Journal of the American Statistical Association*, **74**, 153–160.

Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., and Rubin, D.B. (2014). Bayesian Data Analysis, 3rd ed. CRC Press: Boca Raton, FL.

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, **24**, 997–1010.

Inacio de Carvalho, V., and Rodriguez-Alvarez, M. X. (2022). The Covariate-Adjusted ROC Curve: The Concept and Its Importance, Review of Inferential Methods, and a New Bayesian Estimator. *Statistical Science*, **37**, 541 -561.

Rubin, D. B. (1981). The Bayesian bootstrap. *The Annals of Statistics*, **9**, 130–134.

Speigelhalter, D. J., Best, N. G., Carlin, B. P., and van der Linde, A. (2002). Bayesian measures of model comparison and fit. *Journal of the Royal Statistical Society, Ser. B*, **64**, 583–639.

Watanabe, S. (2010). Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory. *Journal of Machine Learning Research*, **11**, 3571–3594.

See Also

AROC.bnp, AROC.sp, AROC.kernel, pooledROC.BB, pooledROC.emp, pooledROC.kernel, pooledROC.dpm, cROC.bnp, cROC.sp or AROC.kernel.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

AROC_bnp <- AROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
              group = "status",
              tag.h = 0,
              data = newpsa,
              standardise = TRUE,
              p = seq(0,1,l=101),
              compute.lpml = TRUE,
              compute.WAIC = TRUE,
              compute.DIC = TRUE,
              pauc = pauccontrol(compute = TRUE, focus = "FPF", value = 0.5),
              density = densitycontrol.aroc(compute = TRUE, grid.h = NA, newdata = NA),
           prior.h = priorcontrol.bnp(m0 = rep(0, 4), S0 = 10*diag(4), nu = 6, Psi = diag(4),
              a = 2, b = 0.5, alpha = 1, L =10),
              mcmc = mcmccontrol(nsave = 500, nburn = 100, nskip = 1))

summary(AROC_bnp)

plot(AROC_bnp)
```

AROC.kernel *Nonparametric kernel-based estimation of the covariate-adjusted ROC curve (AROC).*

Description

This function estimates the covariate-adjusted ROC curve (AROC) using the nonparametric kernel-based method proposed by Rodriguez-Alvarez et al. (2011). The method, as it stands now, can only deal with one continuous covariate.

Usage

```
AROC.kernel(marker, covariate, group, tag.h,
    bw = c("LS", "AIC"),
```

```
regtype = c("LC", "LL"),
pauc = pauccontrol(),
data, p = seq(0, 1, l = 101), B = 1000, ci.level = 0.95,
parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

marker	A character string with the name of the diagnostic test variable.
covariate	A character string with the name of the continuous covariate.
group	A character string with the name of the variable that distinguishes healthy from diseased individuals.
tag.h	The value codifying healthy individuals in the variable group.
bw	A character string specifying which method to use to select the bandwidths. AIC specifies expected Kullback-Leibler cross-validation, and LS specifies least-squares cross-validation. Defaults to LS. For details see R-package np.
regtype	A character string specifying which type of kernel estimator to use for the regression function (see Details). LC specifies a local-constant estimator (Nadaraya-Watson) and LL specifies a local-linear estimator. Defaults to LC. For details see R-package np.
pauc	A list of control values to replace the default values returned by the function pauccontrol. This argument is used to indicate whether the partial area under the covariate-adjusted ROC curve (pAAUC) should be computed, and in case it is computed, whether the focus should be placed on restricted false positive fractions (FPFs) or on restricted true positive fractions (TPFs), and the upper bound for the FPF (if focus is FPF) or the lower bound for the TPF (if focus is TPF).
data	A data frame representing the data and containing all needed variables.
p	Set of false positive fractions (FPF) at which to estimate the covariate-adjusted ROC curve. This set is also used to compute the area under the covariate-adjusted ROC curve (AAUC) using Simpson's rule. Thus, the length of the set should be an odd number and it should be rich enough for an accurate estimation.
B	An integer value specifying the number of bootstrap resamples for the construction of the confidence intervals. The default is 1000.
ci.level	An integer value (between 0 and 1) specifying the confidence level. The default is 0.95.
parallel	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
ncpus	An integer with the number of processes to be used in parallel operation. Defaults to 1.
cl	An object inheriting from class cluster (from the parallel package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates the covariate-adjusted ROC curve (AROC) defined as

$$AROC\,(p) = Pr\{1 - F_{\bar{D}}(Y_D|X_D) \leq p\},$$

where $F_{\bar{D}}(y|x) = Pr\{Y_{\bar{D}} \leq y|X_{\bar{D}} = x\}$. In particular, the method implemented in this function estimates the outer probability empirically (see Janes and Pepe, 2009) and $F_{\bar{D}}(y|x)$ is estimated assuming a nonparametric location-scale regression model for $Y_{\bar{D}}$, i.e.,

$$Y_{\bar{D}} = \mu_{\bar{D}}(X_{\bar{D}}) + \sigma_{\bar{D}}(X_{\bar{D}})\varepsilon_{\bar{D}},$$

where $\mu_{\bar{D}}(x) = E(Y_{\bar{D}}|X_{\bar{D}} = x)$ is the regression funcion, $\sigma^2_{\bar{D}}(x) = Var(Y_{\bar{D}}|X_{\bar{D}} = x)$ is the variance function, and $\varepsilon_{\bar{D}}$ has zero mean, variance one, and distribution function $G_{\bar{D}}$. As a consequence,

$$F_{\bar{D}}(y|x) = G_{\bar{D}}\left(\frac{y - \mu_{\bar{D}}(x)}{\sigma_{\bar{D}}(x)}\right).$$

By default, both the regression and variance functions are estimated using the Nadaraya-Watson estimator (LC), and the bandwidths are selected using least-squares cross-validation (LS). Implementation relies on the R-package np. No assumption is made about $G_{\bar{D}}$, which is empirically estimated on the basis of the standardised residuals.

The area under the AROC curve is

$$AAUC = \int_0^1 AROC(p)dp,$$

and there exists a closed-form estimator. With regard to the partial area under the curve, when `focus = "FPF"` and assuming an upper bound u_1 for the FPF, what it is computed is

$$pAAUC_{FPF}(u_1) = \int_0^{u_1} AROC(p)dp,$$

where again there exists a closed-form estimator. The returned value is the normalised pAAUC, $pAAUC_{FPF}(u_1)/u_1$ so that it ranges from $u_1/2$ (useless test) to 1 (perfect marker). Conversely, when `focus = "TPF"`, and assuming a lower bound for the TPF of u_2, the partial area corresponding to TPFs lying in the interval $(u_2, 1)$ is computed as

$$pAAUC_{TPF}(u_2) = \int_{AROC^{-1}(u_2)}^1 AROC(p)dp - \{1 - AROC^{-1}(u_2)\} \times u_2.$$

Here, the computation of the integral is done numerically. The returned value is the normalised pAAUC, $pAAUC_{TPF}(u_2)/(1 - u_2)$, so that it ranges from $(1 - u_2)/2$ (useless test) to 1 (perfect test).

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`data`	The original supplied data argument.

missing.ind	A logical value indicating whether for each pair of observations (test outcomes and covariates) missing values occur.
marker	The name of the diagnostic test variable in the dataframe.
covariate	The value of the argument covariate used in the call.
group	The value of the argument group used in the call.
tag.h	The value of the argument tag.h used in the call.
p	Set of false positive fractions (FPF) at which the covariate-adjusted ROC curve has been estimated.
ci.level	The value of the argument ci.level used in the call.
ROC	Estimated covariate-adjusted ROC curve (AROC), and ci.level*100% pointwise confidence band (if computed).
AUC	Estimated area under the covariate-adjusted ROC curve (AAUC), and ci.level*100% confidence interval (if computed).
pAUC	If computed, estimated partial area under the covariate-adjusted ROC curve (pAAUC) and ci.level*100% confidence interval (if computed). Note that the returned values are normalised, so that the maximum value is one.
fit	List with the following components: (1) bw.mean: An object of class npregbw with the selected bandwidth for the nonparametric regression function. For further details, see R-package np. (2) bw.var: An object of class npregbw with the selected bandwidth for the nonparametric variance function. For further details, see R-package np. (3) fit.mean: An object of class npreg with the nonparametric regression function estimate. For further details, see R-package np. (4) fit.var: An object of class npreg with the nonparametric variance function estimate. For further details, see R-package np.

References

Hayfield, T., and Racine, J. S. (2008). Nonparametric Econometrics: The np Package. *Journal of Statistical Software*, 27(5). URL http://www.jstatsoft.org/v27/i05/.

Inacio de Carvalho, V., and Rodriguez-Alvarez, M. X. (2022). The Covariate-Adjusted ROC Curve: The Concept and Its Importance, Review of Inferential Methods, and a New Bayesian Estimator. *Statistical Science*, **37**, 541 -561.

Janes, H., and Pepe, M.S. (2009). Adjusting for covariate effects on classification accuracy using the covariate-adjusted receiver operating characteristic curve. *Biometrika*, **96**, 371–382.

Rodriguez-Alvarez, M. X., Roca-Pardinas, J., and Cadarso-Suarez, C. (2011). ROC curve and covariates: extending induced methodology to the non-parametric framework. *Statistics and Computing*, **21**, 483–499.

See Also

AROC.bnp, AROC.sp, AROC.kernel, pooledROC.BB, pooledROC.emp, pooledROC.kernel, pooledROC.dpm, cROC.bnp, cROC.sp or AROC.kernel.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

m2 <- AROC.kernel(marker = "l_marker1",
covariate = "age",
group = "status",
tag.h = 0,
data = newpsa,
bw = "LS",
regtype = "LC",
pauc = pauccontrol(compute = TRUE, focus = "FPF", value = 0.5),
B = 500)

summary(m2)

plot(m2)
```

`AROC.sp` *Semiparametric frequentist inference for the covariate-adjusted ROC curve (AROC).*

Description

This function estimates the covariate-adjusted ROC curve (AROC) using the semiparametric approach proposed by Janes and Pepe (2009).

Usage

```
AROC.sp(formula.h, group, tag.h, data,
    est.cdf.h = c("normal", "empirical"), pauc = pauccontrol(),
    p = seq(0, 1, l = 101), B = 1000, ci.level = 0.95,
   parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

`formula.h`	A `formula` object specifying the location regression model to be fitted in the healthy population (see Details).
`group`	A character string with the name of the variable that distinguishes healthy from diseased individuals.
`tag.h`	The value codifying healthy individuals in the variable `group`.

data	A data frame representing the data and containing all needed variables.
est.cdf.h	A character string. It indicates how the conditional distribution function of the diagnostic test in the healthy population is estimated. Options are "normal" and "empirical" (see Details). The default is "normal".
pauc	A list of control values to replace the default values returned by the function pauccontrol. This argument is used to indicate whether the partial area under the covariate-adjusted ROC curve (pAAUC) should be computed, and in case it is computed, whether the focus should be placed on restricted false positive fractions (FPFs) or on restricted true positive fractions (TPFs), and the upper bound for the FPF (if focus is FPF) or the lower bound for the TPF (if focus is TPF).
p	Set of false positive fractions (FPF) at which to estimate the covariate-adjusted ROC curve. This set is also used to compute the area under the covariate-adjusted ROC curve (AAUC) using Simpson's rule. Thus, the length of the set should be an odd number, and it should be rich enough for an accurate estimation.
B	An integer value specifying the number of bootstrap resamples for the construction of the confidence intervals. The default is 1000.
ci.level	An integer value (between 0 and 1) specifying the confidence level. The default is 0.95.
parallel	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
ncpus	An integer with the number of processes to be used in parallel operation. Defaults to 1.
cl	An object inheriting from class cluster (from the parallel package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates the covariate-adjusted ROC curve (AROC) defined as

$$AROC\,(p) = Pr\{1 - F_{\bar{D}}(Y_D|\mathbf{X}_D) \leq p\},$$

$$F_{\bar{D}}(y|\mathbf{x}) = Pr\{Y_{\bar{D}} \leq y|\mathbf{X}_{\bar{D}} = \mathbf{x}\}.$$

The method implemented in this function estimates the outer probability empirically (see Janes and Pepe, 2009) and $F_{\bar{D}}(\cdot|\mathbf{x})$ is estimated assuming a semiparametric location regression model for $Y_{\bar{D}}$, i.e.,

$$Y_{\bar{D}} = \mathbf{X}_{\bar{D}}^T \beta_{\bar{D}} + \sigma_{\bar{D}} \varepsilon_{\bar{D}},$$

where $\varepsilon_{\bar{D}}$ has zero mean, variance one, and distribution function $G_{\bar{D}}$. As a consequence, we have

$$F_{\bar{D}}(y|\mathbf{x}) = G_{\bar{D}}\left(\frac{y - \mathbf{x}^T \beta_{\bar{D}}}{\sigma_{\bar{D}}}\right).$$

In line with the assumptions made about the distribution of $\varepsilon_{\bar{D}}$, estimators will be referred to as: (a) "normal", where a standard Gaussian error is assumed, i.e., $G_{\bar{D}}(y) = \Phi(y)$; and, (b) "empirical",

where no assumption is made about the distribution (in this case, $G_{\bar{D}}$ is empirically estimated on the basis of standardised residuals).

The area under the AROC curve is

$$AAUC = \int_0^1 AROC(p)dp,$$

and there exists a closed-form estimator. With regard to the partial area under the AROC curve, when `focus = "FPF"` and assuming an upper bound u_1 for the FPF, what it is computed is

$$pAAUC_{FPF}(u_1) = \int_0^{u_1} AROC(p)dp,$$

where again there exists a closed-form estimator. The returned value is the normalised pAAUC, $pAAUC_{FPF}(u_1)/u_1$ so that it ranges from $u_1/2$ (useless test) to 1 (perfect marker). Conversely, when `focus = "TPF"`, and assuming a lower bound for the TPF of u_2, the partial area corresponding to TPFs lying in the interval $(u_2, 1)$ is computed as

$$pAAUC_{TPF}(u_2) = \int_{AROC^{-1}(u_2)}^1 AROC(p)dp - \{1 - AROC^{-1}(u_2)\} \times u_2.$$

Here, the computation of the integral is done numerically. The returned value is the normalised pAAUC, $pAAUC_{TPF}(u_2)/(1-u_2)$, so that it ranges from $(1-u_2)/2$ (useless test) to 1 (perfect test).

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`data`	The original supplied data argument.
`missing.ind`	A logical value indicating whether for each pair of observations (test outcomes and covariates) missing values occur.
`marker`	The name of the diagnostic test variable in the dataframe.
`group`	The value of the argument `group` used in the call.
`tag.h`	The value of the argument `tag.h` used in the call.
`formula`	The value of the argument `formula.h` used in the call.
`est.cdf.h`	The value of the argument `est.cdf.h` used in the call.
`p`	Set of false positive fractions (FPF) at which the covariate-adjusted ROC (AROC) curve has been estimated
`ci.level`	The value of the argument `ci.level` used in the call.
`ROC`	Estimated covariate-adjusted ROC curve (AROC), and `ci.level`*100% point-wise confidence bands (if computed)
`AUC`	Estimated area under the covariate-adjusted ROC curve (AAUC), and `ci.level`*100% confidence intervals (if required).
`pAUC`	If computed, estimated partial area under the covariate-adjusted ROC curve (pAAUC) and `ci.level`*100% confidence interval (if computed). Note that the returned values are normalised, so that the maximum value is one.

fit	Object of class lm with the fitted regression model in the healthy population.
coeff	Estimated regression coefficients (and ci.level*100% confidence interval if B greater than zero) from the fit of the linear model in the healthy population, as specified in formula.h.

References

Janes, H., and Pepe, M.S. (2009). Adjusting for covariate effects on classification accuracy using the covariate-adjusted receiver operating characteristic curve. Biometrika, 96(2), 371 - 382.

See Also

AROC.bnp, AROC.sp, AROC.kernel, pooledROC.BB, pooledROC.emp, pooledROC.kernel, pooledROC.dpm, cROC.bnp, cROC.sp or AROC.kernel.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

m3 <- AROC.sp(formula.h = l_marker1 ~ age,
group = "status",
tag.h = 0,
data = newpsa,
est.cdf.h = "normal",
pauc = pauccontrol(compute = TRUE, focus = "FPF", value = 0.5),
p = seq(0,1,l=101),
B = 500)

summary(m3)

plot(m3)
```

---

compute.threshold.AROC

AROC based threshold values.

Description

This function implements methods for estimating AROC-based threshold values.

Usage

```
compute.threshold.AROC(object, criterion = c("FPF", "YI"), FPF, newdata,
  ci.level = 0.95, parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

object	An object of class AROC as produced by AROC.bnp(), AROC.sp(), or AROC.kernel().
criterion	A character string indicating whether the covariate-adjusted threshold values should be computed based on the Youden index ("YI") or for a fixed set of false positive fractions ("FPF").
FPF	For criterion = FPF, a numeric vector with the FPF at which to calculate the AROC-based threshold values. Atomic values are also valid.
newdata	Optional data frame containing the values of the covariates at which the AROC-based threshold values will be computed. If not supplied, the function cROCData is used to build a default dataset.
ci.level	An integer value (between 0 and 1) specifying the confidence level. The default is 0.95.
parallel	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
ncpus	An integer with the number of processes to be used in parallel operation. Defaults to 1.
cl	An object inheriting from class cluster (from the parallel package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates AROC-based threshold values based on two different criteria, namely, the Youden index (YI) and the one that gives rise to a pre-specified FPF. Before proceeding, we would like to mention that when the accuracy of a test is not affected by covariates, this does not necessarily imply that the covariate-specific ROC curve (which in this case is the same for all covariate values) coincides with the pooled ROC curve. It does coincide, however, with the AROC curve. Consequently, in all cases where covariates affect the test, even though they might not affect its discriminatory capacity, inferences based on the pooled ROC curve might be misleading. In such cases the AROC curve should be used instead. This also applies to the selection of (optimal) threshold values, which, as will be seen, might be covariate-specific (i.e., possibly different for different covariate values).

For the AROC curve, the Youden Index is defined as

$$YI = \max_p \{AROC(p) - p\},$$

The value p^* (FPF) that achieves the maximum is then used to calculate the optimal (covariate-specific) YI threshold as follows

$$c^*_{\mathbf{x}} = F^{-1}_{\bar{D}}(1 - p^*|\mathbf{x}),$$

where

$$F_{\bar{D}}(y|\mathbf{x}) = Pr(Y_{\bar{D}} \leq y|\mathbf{X}_{\bar{D}} = \mathbf{x}).$$

In a similar way, when using the criterion for a fixed FPF, the covariate-specific threshold values are obtained as follows

$$c^*_{\mathbf{x}} = F^{-1}_{\bar{D}}(1 - FPF|\mathbf{x}).$$

In both cases, we use the notation $c^*_{\mathbf{x}}$ to emphasise that this value depends on covariate $\mathbf{x}$.

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`newdata`	Data frame containing the values of the covariates at which the AROC-based thresholds were computed.
`thresholds`	If method = "YI", the estimated AROC-based threshold corresponding to the Youden index, and if method = "FPF", AROC-based threshold corresponding to the specified FPF. For the Bayesian approach (`AROC.bnp`), in addition to the posterior mean, the `ci.level`*100% pointwise credible band is also returned.
`YI`	If criterion = "YI", the AROC-based Youden index. For the Bayesian approach (`AROC.bnp`), in addition to the posterior mean, the `ci.level`*100% pointwise credible band is also returned.
`FPF`	If criterion = "YI", the FPF where the Youden index is attained, and if criterion = "FPF", the supplied FPF argument. For the Bayesian approach (`AROC.bnp`), in addition to the posterior mean, the `ci.level`*100% pointwise credible band is also returned.

References

Inacio de Carvalho, V., and Rodriguez-Alvarez, M. X. (2022). The Covariate-Adjusted ROC Curve: The Concept and Its Importance, Review of Inferential Methods, and a New Bayesian Estimator. *Statistical Science*, **37**, 541 -561.

Rodriguez-Alvarez, M. X., Roca-Pardinas, J., and Cadarso-Suarez, C. (2011). ROC curve and covariates: extending induced methodology to the non-parametric framework. *Statistics and Computing*, **21**, 483–499.

Rutter, C.M. and Miglioretti, D. L. (2003). Estimating the Accuracy of Psychological Scales Using Longitudinal Data. *Biostatistics*, **4**, 97–107.

Youden, W. J. (1950). Index for rating diagnostic tests. Cancer, **3**, 32–35.

See Also

`AROC.bnp`, `AROC.kernel` or `AROC.sp`

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
```

```
newpsa$l_marker1 <- log(newpsa$marker1)

AROC_bnp <- AROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
              group = "status",
              tag.h = 0,
              data = newpsa,
              standardise = TRUE,
              p = seq(0,1,l=101),
              prior = priorcontrol.bnp(m0 = rep(0, 4),
              S0 = 10*diag(4), nu = 6, Psi = diag(4),
              a = 2, b = 0.5, alpha = 1, L =10),
              mcmc = mcmccontrol(nsave = 500, nburn = 100, nskip = 1))

### Threshold values based on the YI
th_AROC_bnp_yi <- compute.threshold.AROC(AROC_bnp, criterion = "YI")

# Plot results
plot(th_AROC_bnp_yi$newdata$age, th_AROC_bnp_yi$thresholds[,"est"],
 type = "l", xlab = "Age",
 ylab = "log(PSA)", ylim = c(0,3),
 main = "Threshold values based on the Youden Index")
lines(th_AROC_bnp_yi$newdata$age, th_AROC_bnp_yi$thresholds[,"qh"], lty = 2)
lines(th_AROC_bnp_yi$newdata$age, th_AROC_bnp_yi$thresholds[,"ql"], lty = 2)

### Threshold values for a fixed FPF
th_AROC_bnp_fpf <- compute.threshold.AROC(AROC_bnp, criterion = "FPF", FPF = 0.1)

# Plot results
plot(th_AROC_bnp_yi$newdata$age, th_AROC_bnp_fpf$thresholds[["0.1"]][,"est"],
 type = "l", xlab = "Age",
 ylab = "log(PSA)", ylim = c(0,3),
 main = "Threshold values for a FPF = 0.1")
lines(th_AROC_bnp_yi$newdata$age, th_AROC_bnp_fpf$thresholds[["0.1"]][,"qh"], lty = 2)
lines(th_AROC_bnp_yi$newdata$age, th_AROC_bnp_fpf$thresholds[["0.1"]][,"ql"], lty = 2)
```

`compute.threshold.cROC` *Covariate-specific ROC based threshold values.*

Description

This function implements methods for estimating covariate-specific ROC-based threshold values.

Usage

```
compute.threshold.cROC(object, criterion = c("FPF", "TPF", "YI"), FPF, TPF, newdata,
  ci.level = 0.95, parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

`object`	An object of class `cROC` as produced by `cROC.bnp()`, `cROC.sp()`, or `cROC.kernel()`.
`criterion`	A character string indicating whether the covariate-specific threshold values should be computed based on the Youden index ("YI") or for fixed false positive fractions ("FPF") or true positive fractions ("TPF").
`FPF`	For `criterion = "FPF"`, a numeric vector with the FPF at which to calculate the covariate-specific threshold values. Atomic values are also valid.
`TPF`	For `criterion = "TPF"`, a numeric vector with the TPF at which to calculate the covariate-specific threshold values. Atomic values are also valid.
`newdata`	Optional data frame containing the values of the covariates at which the covariate-specific threshold values will be computed. If not supplied, the function `cROCData` is used to build a default dataset.
`ci.level`	An integer value (between 0 and 1) specifying the confidence level. The default is 0.95.
`parallel`	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
`ncpus`	An integer with the number of processes to be used in parallel operation. Defaults to 1.
`cl`	An object inheriting from class `cluster` (from the `parallel` package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates covariate-specific ROC-based threshold values based on three different criteria, namely, the Youden index (YI), one that gives rise to a pre-specified FPF, and one that gives rise to a pre-specified TPF.

In the conditional case, the Youden index is defined as

$$YI(\mathbf{x}) = \max_c |TPF(c|\mathbf{x}) - FPF(c|\mathbf{x})| = \max_c |F_{\bar{D}}(c|\mathbf{x}) - F_D(c|\mathbf{x})|,$$

where

$$F_D(y|\mathbf{x}) = Pr(Y_D \leq y|\mathbf{X}_D = \mathbf{x}),$$

$$F_{\bar{D}}(y|\mathbf{x}) = Pr(Y_{\bar{D}} \leq y|\mathbf{X}_{\bar{D}} = \mathbf{x}).$$

The value $c^*_{\mathbf{x}}$ that achieves the maximum is called the optimal covariate-specific YI threshold. Regarding the criterion for a fixed FPF, the covariate-specific threshold values are obtained as follows

$$c^*_{\mathbf{x}} = F^{-1}_{\bar{D}}(1 - FPF|\mathbf{x}),$$

and for a fixed TPF we have

$$c^*_{\mathbf{x}} = F^{-1}_{D}(1 - TPF|\mathbf{x}),$$

In all cases, we use the notation $c^*_{\mathbf{x}}$ to emphasise that this value depends on covariate $\mathbf{x}$.

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`newdata`	Data frame containing the values of the covariates at which the covariate-specific thresholds were computed.
`thresholds`	If method = "YI", the estimated covariate-specific (optimal) threshold corresponding to the covariate-specific Youden index (the one that maximises TPF/sensitivity + TNF/specificity). If method = "FPF", the covariate-specific threshold corresponding to the specified FPF, and if method = "TPF", the covariate-specific threshold corresponding to the specified TPF. For the Bayesian approach (`cROC.bnp`), in addition to the posterior mean, the `ci.level`*100% pointwise credible band is also returned.
`YI`	If method = "YI", the estimated covariate-specific Youden index. For the Bayesian approach (`cROC.bnp`), in addition to the posterior mean, the `ci.level`*100% pointwise credible band is also returned.
`FPF`	If method = "YI" or method = "TPF", the FPF corresponding to the estimated (optimal) covariate-specific thresholds (for the Bayesian approach (`cROC.bnp`), in addition to the posterior mean, the `ci.level`*100% pointwise credible band is also returned.). If method = "FPF", the supplied FPF argument.
`TPF`	If method = "YI" or method = "FPF", the covariate-specific TPF/sensitivity corresponding to the estimated covariate-specific (optimal) threshold. For the Bayesian approach (`AROC.bnp`), in addition to the posterior mean, the `ci.level`*100% pointwise credible band is also returned. If method = "TPF", the supplied TPF argument.

References

Inacio de Carvalho, V., de Carvalho, M. and Branscum, A. J. (2017). Nonparametric Bayesian Covariate-Adjusted Estimation of the Youden Index. Biometrics, **73**, 1279-1288.

Rodriguez-Alvarez, M. X., Roca-Pardinas, J., and Cadarso-Suarez, C. (2011). ROC curve and covariates: extending induced methodology to the non-parametric framework. Statistics and Computing, **21**, 483–499.

Rutter, C.M. and Miglioretti, D. L. (2003). Estimating the Accuracy of Psychological Scales Using Longitudinal Data. Biostatistics, **4**, 97–107.

Youden, W. J. (1950). Index for rating diagnostic tests. Cancer, **3**, 32–35.

See Also

`cROC.bnp`, `cROC.kernel` or `cROC.sp`

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]
```

```
# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

cROC_bnp <- cROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
     formula.d = l_marker1 ~ f(age, K = 0),
              group = "status",
              tag.h = 0,
              data = newpsa,
              standardise = TRUE,
              p = seq(0,1,l=101),
              mcmc = mcmccontrol(nsave = 500, nburn = 100, nskip = 1))

### Threshold values based on the YI
th_cROC_bnp_yi <- compute.threshold.cROC(cROC_bnp, criterion = "YI")

# Plot results
 # Threshold values
 plot(th_cROC_bnp_yi$newdata$age, th_cROC_bnp_yi$thresholds[,"est"],
  type = "l", xlab = "Age",
  ylab = "log(PSA)", ylim = c(0,3),
  main = "Threshold values based on the Youden Index")
 lines(th_cROC_bnp_yi$newdata$age, th_cROC_bnp_yi$thresholds[,"qh"], lty = 2)
 lines(th_cROC_bnp_yi$newdata$age, th_cROC_bnp_yi$thresholds[,"ql"], lty = 2)

 # Youden Index
 plot(th_cROC_bnp_yi$newdata$age, th_cROC_bnp_yi$YI[,"est"],
  type = "l", xlab = "Age",
  ylab = "log(PSA)", ylim = c(0,1),
  main = "Threshold values based on the Youden Index")
 lines(th_cROC_bnp_yi$newdata$age, th_cROC_bnp_yi$YI[,"qh"], lty = 2)
 lines(th_cROC_bnp_yi$newdata$age, th_cROC_bnp_yi$YI[,"ql"], lty = 2)

### Threshold values for a fixed FPF
th_cROC_bnp_fpf <- compute.threshold.cROC(cROC_bnp, criterion = "FPF", FPF = 0.1)

# Plot results
 # Threshold values
 plot(th_cROC_bnp_yi$newdata$age, th_cROC_bnp_fpf$thresholds[["0.1"]][,"est"],
  type = "l", xlab = "Age",
  ylab = "log(PSA)", ylim = c(0,3), main = "Threshold values for a FPF = 0.1")
 lines(th_cROC_bnp_yi$newdata$age, th_cROC_bnp_fpf$thresholds[["0.1"]][,"qh"], lty = 2)
 lines(th_cROC_bnp_yi$newdata$age, th_cROC_bnp_fpf$thresholds[["0.1"]][,"ql"], lty = 2)
```

`compute.threshold.pooledROC`

Pooled ROC based threshold values.

Description

This function implements methods for estimating pooled ROC-based threshold values.

Usage

```
compute.threshold.pooledROC(object, criterion = c("FPF", "TPF", "YI"), FPF, TPF,
  ci.level = 0.95, parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

object	An object of class pooledROC as produced by pooledROC.BB, pooledROC.emp, pooledROC.kernel, or pooledROC.dpm functions.
criterion	A character string indicating if the threshold value should be computed based on the Youden index ("YI"), or for fixed false positive fractions ("FPF") or true positive fractions ("TPF").
FPF	For criterion = "FPF", a numeric vector with the FPF at which to calculate the threshold values. Atomic values are also valid.
TPF	For criterion = "TPF", a numeric vector with the TPF at which to calculate the threshold values. Atomic values are also valid.
ci.level	An integer value (between 0 and 1) specifying the confidence level. The default is 0.95.
parallel	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
ncpus	An integer with the number of processes to be used in parallel operation. Defaults to 1.
cl	An object inheriting from class cluster (from the parallel package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates pooled ROC-based threshold values based on three different criteria, namely, the Youden index (YI), one that gives rise to a pre-specified FPF, and one that gives rise to a pre-specified TPF.

The Youden Index is defined as

$$YI = \max_c\{TPF(c) - FPF(c)\} = \max_c\{F_{\bar{D}}(c) - F_D(c)\},$$

where

$$F_D(y) = Pr(Y_D \leq y),$$
$$F_{\bar{D}}(y) = Pr(Y_{\bar{D}} \leq y).$$

The value c^* that achieves the maximum is called the optimal YI threshold. Regarding the criterion for a fixed FPF, the threshold value is obtained as follows

$$c = F_{\bar{D}}^{-1}(1 - FPF).$$

and for a fixed TPF we have

$$c = F_D^{-1}(1 - TPF).$$

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`threshold`	If method = "YI", the estimated (optimal) threshold corresponding to the Youden index (the one that maximises TPF/sensitivity + TNF/specificity). If method = "FPF", the estimated threshold corresponding to the specified FPF, and if method = "TPF", the estimated threshold corresponding to the specified TPF. For the Bayesian approaches (`pooledROC.dpm` and `pooledROC.BB`), and in both cases, in addition to the posterior mean, the `ci.level`*100% credible interval is also returned.
`YI`	If method = "YI", the estimated Youden index. For the Bayesian approaches (`pooledROC.dpm` and `pooledROC.BB`), in addition to the posterior mean, the `ci.level`*100% credible interval is also returned.
`FPF`	If method = "YI" or method = "TPF", the FPF corresponding to the estimated (optimal) threshold (For the Bayesian approaches (`pooledROC.dpm` and `pooledROC.BB`), in addition to the posterior mean, the `ci.level`*100% credible interval is also returned). If method = "FPF", the supplied FPF argument.
`TPF`	If method = "YI" or method = "FPF", the TPF/sensitivity corresponding to the estimated (optimal) threshold. For the Bayesian approaches (`pooledROC.dpm` and `pooledROC.BB`), in addition to the posterior mean, the `ci.level`*100% credible interval is also returned. If method = "TPF", the supplied TPF argument.

References

Rutter, C.M. and Miglioretti, D. L. (2003). Estimating the Accuracy of Psychological Scales Using Longitudinal Data. Biostatistics, **4**, 97–107.

Youden, W. J. (1`ci.level`*1000). Index for rating diagnostic tests. Cancer, **3**, 32–35.

See Also

`pooledROC.BB`, `pooledROC.emp`, `pooledROC.kernel` or `pooledROC.dpm`.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

m0_dpm <- pooledROC.dpm(marker = "l_marker1", group = "status",
            tag.h = 0, data = newpsa, standardise = TRUE,
            p = seq(0,1,l=101), compute.WAIC = TRUE, compute.lpml = TRUE,
            compute.DIC = TRUE,
```

```
            prior.h = priorcontrol.dpm(m0 = 0, S0 = 10, a = 2, b = 0.5, alpha = 1,
            L =10),
            prior.d = priorcontrol.dpm(m0 = 0, S0 = 10, a = 2, b = 0.5, alpha = 1,
            L =10),
            mcmc = mcmccontrol(nsave = 400, nburn = 100, nskip = 1))


## Threshold values based on the YI
th_m0_dpm_yi <- compute.threshold.pooledROC(m0_dpm, criterion = "YI")

th_m0_dpm_yi$threshold
th_m0_dpm_yi$YI

### Threshold values for a fixed FPF
th_m0_dpm_fpf <- compute.threshold.pooledROC(m0_dpm, criterion = "FPF", FPF = 0.1)

th_m0_dpm_fpf$threshold
```

---

cROC.bnp *Nonparametric Bayesian inference for the covariate-specific ROC curve (cROC).*

Description

This function estimates the covariate-specific ROC curve (cROC) using the nonparametric Bayesian approach proposed by Inacio de Carvalho et al. (2013).

Usage

```
cROC.bnp(formula.h, formula.d, group, tag.h, data,
    newdata, standardise = TRUE, p = seq(0, 1, l = 101), ci.level = 0.95,
    compute.lpml = FALSE, compute.WAIC = FALSE, compute.DIC = FALSE,
    pauc = pauccontrol(), density = densitycontrol(),
    prior.h = priorcontrol.bnp(), prior.d = priorcontrol.bnp(),
    mcmc = mcmccontrol(),
    parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

formula.h	A formula object specifying the regression function associated to each component of the single-weights linear dependent Dirichlet process mixture of normals model used to estimate the conditional distribution function of the diagnostic test outcome in the healthy population. Regarding the modelling of continuous covariates, both linear and nonlinear effects are allowed, with nonlinear effects being modelled through B-spline basis expansions (see Note).

formula.d	A formula object specifying the regression function associated to each component of the single weights linear dependent Dirichlet process mixture model used to estimate the conditional distribution function of the diagnostic test outcome in the diseased population. Both linear and nonlinear (through the use of smooth functions approximated by linear combinations of cubic B-splines basis functions) covariate effects are allowed (see Note).
group	A character string with the name of the variable that distinguishes healthy from diseased individuals.
tag.h	The value codifying healthy individuals in the variable group.
data	A data frame representing the data and containing all needed variables.
newdata	Optional data frame containing the values of the covariates at which the covariate-specific ROC curve (AUC and pAUC, if computed) will be computed. If not supplied, the function cROCData is used to build a default dataset.
standardise	A logical value. If TRUE both the test outcomes and the continuous covariates assumed to have a linear effect are standardised (i.e., the resulting variables have mean zero and standard deviation of one). The default is TRUE.
p	Set of false positive fractions (FPF) at which to estimate the covariate-specific ROC curve.
ci.level	An integer value (between 0 and 1) specifying the level for the credible interval. The default is 0.95.
compute.lpml	A logical value. If TRUE, the log pseudo marginal likelihood (LPML, Geisser and Eddy, 1979) and the conditional predictive ordinates (CPO) are computed.
compute.WAIC	A logical value. If TRUE, the widely applicable information criterion (WAIC, Gelman et al., 2014; Watanabe, 2010) is computed.
compute.DIC	A logical value. If TRUE, the deviance information criterion (DIC)(Celeux et al., 2006, Spiegelhalter et al., 2002) is computed.
pauc	A list of control values to replace the default values returned by the function pauccontrol. This argument is used to indicate whether the partial area under the covariate-adjusted ROC curve should be computed, and in case it is computed, whether the focus should be placed on restricted false positive fractions (FPFs) or on restricted true positive fractions (TPFs), and the upper bound for the FPF (if focus is FPF) or the lower bound for the TPF (if focus is TPF).
density	A list of control values to replace the default values returned by the function densitycontrol. This argument is used to indicate whether the conditional densities of the marker in the healthy and diseased population should be computed, and in case it is to be computed, at which grid of test outcomes in each of the populations the conditional densities should be evaluated.
prior.h	Hyparameter specification for the healthy population. A list of control values to replace the default values returned by the function priorcontrol.bnp. See priorcontrol.bnp for details.
prior.d	Hyparameter specification for the diseased population. A list of control values to replace the default values returned by the function priorcontrol.bnp. See priorcontrol.bnp for details.
mcmc	A list of control values to replace the default values returned by the function mcmccontrol. See mcmccontrol for details.

`parallel` A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".

`ncpus` An integer with the number of processes to be used in parallel operation. Defaults to 1.

`cl` An object inheriting from class `cluster` (from the `parallel` package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates the covariate-specific ROC curve (cROC) defined as

$$ROC(p|\mathbf{x}) = 1 - F_D\{F_{\bar{D}}^{-1}(1-p|\mathbf{x})|\mathbf{x}\},$$

where

$$F_D(y|\mathbf{x}) = Pr(Y_D \leq y|\mathbf{X}_D = \mathbf{x}),$$
$$F_{\bar{D}}(y|\mathbf{x}) = Pr(Y_{\bar{D}} \leq y|\mathbf{X}_{\bar{D}} = \mathbf{x}).$$

Note that, for the sake of clarity, we assume that the covariates of interest are the same in both the healthy and diseased populations. The method implemented in this function estimates $F_D(\cdot|\mathbf{x})$ and $F_{\bar{D}}(\cdot|\mathbf{x})$ by means of a single-weights linear dependent Dirichlet process mixture of normals model (De Iorio et al., 2009). More precisely, and letting $\{(\mathbf{x}_{\bar{D}i}, y_{\bar{D}i})\}_{i=1}^{n_{\bar{D}}}$ and $\{(\mathbf{x}_{Dj}, y_{Dj})\}_{j=1}^{n_D}$ be two independent random samples from the nondiseased and diseased populations, respectively, our postulated model for the conditional distribution in each group function takes the following form

$$F_{\bar{D}}(y_{\bar{D}i}|\mathbf{X}_{\bar{D}} = \mathbf{x}_{\bar{D}i}) = \sum_{l=1}^{L_{\bar{D}}} \omega_{l\bar{D}} \Phi(y_{\bar{D}i} \mid \mu_{l\bar{D}}(\mathbf{x}_{\bar{D}i}), \sigma_{l\bar{D}}^2),$$

$$F_D(y_{Dj}|\mathbf{X}_D = \mathbf{x}_{Dj}) = \sum_{l=1}^{L_D} \omega_{lD} \Phi(y_{Dj} \mid \mu_{lD}(\mathbf{x}_{\bar{D}i}), \sigma_{lD}^2),$$

where $\Phi(y|\mu, \sigma^2)$ denotes the cumulative distribution function of the normal distribution, evaluated at y, with mean mu and variance σ^2. The regression function $\mu_{ld}(\mathbf{x}_{di})$ can incorportate both linear and nonlinear (through B-splines) effects of continuous covariates, categorical covariates (factors) as well as interactions. Interactions between categorical and (nonlinear) continuous covariates are also allowed (factor-by curve interactions). For the sake of simplicity we write $\mu_{ld}(\mathbf{x}_{di}) = \mathbf{z}_{di}^T \beta_{ld}$, where $\mathbf{z}_{di}$ is the ith column of the design matrix (possibly containing a basis representation of some/all continuous covariates), $d \in \{D, \bar{D}\}$. Here L_d is a pre-specified upper bound on the number of mixture components. The ω_{ld}'s result from a truncated version of the stick-breaking construction ($\omega_{1d} = v_{1d}$; $\omega_{ld} = v_{ld} \prod_{r<l}(1 - v_{dr})$, $l = 2, \ldots, L_d$; $v_{d1}, \ldots, v_{L_d - 1} \sim$ Beta $(1, \alpha_d)$; $v_{Ld} = 1$, $\alpha_d \sim \Gamma(a_{\alpha_d}, b_{\alpha_d})$), $\beta_{ld} \sim N_{Q_d}(\mathbf{m}_d, \mathbf{S}_d)$, and $\sigma_{ld}^{-2} \sim \Gamma(a_d, b_d)$. It is further assumed that $\mathbf{m}_d \sim N_{Q_k}(\mathbf{m}_{0d}, \mathbf{S}_{0d})$ and $\mathbf{S}_d^{-1} \sim W(\nu, (\nu_k \Psi_d)^{-1})$. Here $W(\nu, (\nu\Psi)^{-1})$ denotes a Wishart distribution with ν degrees of freedom and expectation Ψ^{-1}, Here $\Gamma(a, b)$ denotes a Gamma distribution with shape parameter a and rate parameter b, and Q_d denotes the dimension of the vector $\mathbf{z}_{di}$. It is worth mentioning that when $L_d = 1$, the model for the conditional distribution of the test outcomes reduces to a normal regression model (where continuous covariates effects are modelled either parametrically or nonparametrically). For a detailed description, we refer to Inacio de Carvalho et al. (2013).

The covariate-specific area under the curve is

$$AUC(\mathbf{x}) = \int_0^1 ROC(p|\mathbf{x})dp.$$

When the upper bound on the number of mixture components is 1, i.e., $L_d = 1$ ($d \in \{D, \bar{D}\}$), there is a closed-form expression for the covariate-specific AUC (binormal model), which is used in the package. In contrast, when $L_D > 1$ or $L_{\bar{D}} > 1$, the integral is computed numerically using Simpson's rule. With regard to the partial area under the curve, when `focus = "FPF"` and assuming an upper bound u_1 for the FPF, what it is computed is

$$pAUC_{FPF}(u_1|\mathbf{x}) = \int_0^{u_1} ROC(p|\mathbf{x})dp.$$

As for the AUC, when $L_d = 1$ ($d \in \{D, \bar{D}\}$), there is a closed-form expression for the $pAUC_{FPF}$ (Hillis and Metz, 2012), and when $L_D > 1$ or $L_{\bar{D}} > 1$ the integral is approximated numerically using Simpson's rule. The returned value is the normalised pAUC, $pAUC_{FPF}(u_1|\mathbf{x})/u_1$ so that it ranges from $u_1/2$ (useless test) to 1 (perfect marker). Conversely, when `focus = "TPF"`, and assuming a lower bound for the TPF of u_2, the partial area corresponding to TPFs lying in the interval $(u_2, 1)$ is computed as

$$pAUC_{TPF}(u_2|\mathbf{x}) = \int_{u_2}^1 ROC_{TNF}(p|\mathbf{x})dp,$$

where $ROC_{TNF}(p|\mathbf{x})$ is a 270° rotation of the ROC curve, and it can be expressed as $ROC_{TNF}(p|\mathbf{x}) = F_{\bar{D}}\{F_D^{-1}(1-p|\mathbf{x})|\mathbf{x}\}$. Again, when $L_d = 1$ ($d \in \{D, \bar{D}\}$), there is a closed-form expression for the $pAUC_{TNF}$ (Hillis and Metz, 2012), and when $L_D > 1$ or $L_{\bar{D}} > 1$ the integral is approximated numerically using Simpson's rule. The returned value is the normalised pAUC, $pAUC_{TPF}(u_2|\mathbf{x})/(1-u_2)$, so that it ranges from $(1-u_2)/2$ (useless test) to 1 (perfect test).

It is worth referring that with respect to the computation of the DIC, when $L = 1$, it is computed as in Spiegelhalter et al. (2002), and when $L > 1$, DIC3 as described in Celeux et al. (2006) is computed. Also, for the computation of the conditional predictive ordinates (CPO) we follow the stable version proposed by Gelman et al. (2014).

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`newdata`	A data frame containing the values of the covariates at which the covariate-specific ROC curve (as well as the AUC, pAUC, dens and reg.fun, if required) was computed.
`data`	The original supplied data argument.
`missing.ind`	A logical value indicating whether for each pair of observations (test outcomes and covariates) missing values occur.
`marker`	The name of the diagnostic test variable in the dataframe.
`group`	The value of the argument `group` used in the call.
`tag.h`	The value of the argument `tag.h` used in the call.

`p` Set of false positive fractions (FPF) at which the covariate-specific ROC curve has been estimated.

`ci.level` The value of the argument `ci.level` used in the call.

`prior` A list returning the hyperparameter values in the healthy and diseased populations.

`ROC` A list returning the `np` (length of the vector `p`) by npred predicted covariate-specific ROC curves (cROC) (posterior mean) and `ci.level`*100% pointwise posterior credible bands.

`AUC` Estimated area under the covariate-specific ROC curve (posterior mean) and `ci.level`*100% posterior credible band.

`pAUC` If computed, estimated partial area under the covariate-adjusted ROC curve (posterior mean) and `ci.level`*100% credible band. Note that the returned values are normalised, so that the maximum value is one (see more on Details).

`dens` Named list of length two, with components 'h' (healthy) and 'd' (diseased). Each component is a list with two components: `grid` (grid of test outcomes where the conditional densities were evaluated) and `dens` (MCMC realisations of the corresponding conditional densities).

`reg.fun` Named list of length two, with components 'h' (healthy) and 'd' (diseased). Each component is a data frame containing the predicted regression function (posterior mean) and `ci.level`*100% credible band.

`lpml` If computed, named list of length two, with components 'h' (healthy) and 'd' (diseased). Each component is a list with two components: the log pseudo marginal likelihood (LPML) and the conditional predictive ordinates (CPO).

`WAIC` If computed, named list of length two, with components 'h' (healthy) and 'd' (diseased). Each component is a list with two components: widely applicable information criterion (WAIC) and associated complexity penalty (pW).

`DIC` If computed, named list of length two, with components 'h' (healthy) and 'd' (diseased). Each component is a list with two components: deviance information criterion (DIC) and associated complexity penalty (pD).

`fit` Named list of length two, with components `h` (healthy) and `d` (diseased). Each component is a list with the following information: (1) `formula`: the value of the argument `formula.h` or `formula.d` used in the call. (2) `mm`: information needed to construct the model matrix associated with single-weights linear dependent Dirichlet process mixture of normals model. (3) `beta`: array of dimension `nsavexLxQ` with the sampled regression coefficients. (4) `sd`: matrix of dimension `nsavexL` with the sampled variances. (4) `probs`: matrix of dimension `nsavexL` with the sampled components' weights. Here, `nsave` is the number of Gibbs sampler iterations saved, `L` is the maximum number of mixture components, and `Q` is the dimension of vector $\mathbf{z}_d$, $d \in \{D, \bar{D}\}$. (see also Details). (see also Details).

`data_model` A list with the data used in the fit: observed diagnostic test outcome and design matrices, separately for the healthy and diseased groups.

Note

The input arguments `formula.h` and `formula.d` are similar to that used for the `glm` function, except that flexible specifications can be added by means of the function `f()`. For instance, specification $y \sim x1 + f(x2, K = 3)$ would assume a linear effect of `x1` (if `x1` continuous) and the effect of `x2` would be modeled using B-splines basis functions. The argument `K = 3` indicates that `3` internal knots will be used, with the quantiles of `x2` used for their location. Categorical variables (factors) can be also incorporated, as well as interaction terms. For example, to include the factor-by-curve interaction between `age` and `gender` we need to specify, e.g., $y \sim gender + f(age, by = gender, K = c(3, 5))$. Note that, in this case, the number of knots can be different for each level of the factor. The order of the vector `K` of knots should match the levels of the factor.

References

Celeux, G., Forbes, F., Robert C. P., and Titerrington, D. M. (2006). Deviance information criteria for missing data models. Bayesian Analysis, **1**, 651–674.

De Iorio, M., Johnson, W. O., Muller, P., and Rosner, G. L. (2009). Bayesian nonparametric nonproportional hazards survival modeling. Biometrics, **65**, 762–775.

Geisser, S. and Eddy, W.F. (1979) A Predictive Approach to Model Selection, Journal of the American Statistical Association, **74**, 153–160.

Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., and Rubin, D.B. (2014). Bayesian Data Analysis, 3rd ed. CRC Press: Boca Raton, FL.

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. Statistics and Computing, **24**, 997–1010.

Hillis, S. L. and Metz, C.E. (2012). An Analytic Expression for the Binormal Partial Area under the ROC Curve. Academic Radiology, **19**, 1491–1498.

Inacio de Carvalho, V., Jara, A., Hanson, T. E., and de Carvalho, M. (2013). Bayesian nonparametric ROC regression modeling. Bayesian Analysis, **8**, 623–646.

Speigelhalter, D. J., Best, N. G., Carlin, B. P., and van der Linde, A. (2002). Bayesian measures of model comparison and fit. Journal of the Royal Statistical Society, Ser. B, **64**, 583–639.

Watanabe, S. (2010). Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory. Journal of Machine Learning Research, **11**, 3571–3594.

See Also

`AROC.bnp`, `AROC.sp`, `AROC.kernel`, `pooledROC.BB`, `pooledROC.emp`, `pooledROC.kernel`, `pooledROC.dpm`, `cROC.bnp`, `cROC.sp` or `AROC.kernel`.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]
```

```
# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

cROC_bnp <- cROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
               formula.d = l_marker1 ~ f(age, K = 0),
               group = "status",
               tag.h = 0,
               data = newpsa,
               standardise = TRUE,
               p = seq(0, 1, len = 101),
               compute.lpml = TRUE,
               compute.WAIC = TRUE,
               compute.DIC = TRUE,
               pauc = pauccontrol(compute = TRUE, value = 0.5, focus = "FPF"),
               density = densitycontrol(compute = TRUE, grid.h = NA, grid.d = NA),
               mcmc = mcmccontrol(nsave = 500, nburn = 100, nskip = 1))
summary(cROC_bnp)

plot(cROC_bnp)
```

`cROC.kernel` *Nonparametric kernel-based estimation of the covariate-specific ROC curve (cROC).*

Description

This function estimates the covariate-specific ROC curve (cROC) using the nonparametric kernel-based method proposed by Rodriguez-Alvarez et al. (2011). The method, as it stands now, can only deal with one continuous covariate.

Usage

```
cROC.kernel(marker, covariate, group, tag.h,
  bw = c("LS", "AIC"), regtype = c("LC", "LL"),
  data, newdata, pauc = pauccontrol(),
  p = seq(0, 1, l = 101), B = 1000, ci.level = 0.95,
    parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

`marker`	A character string with the name of the diagnostic test variable.
`covariate`	A character string with the name of the continuous covariate.
`group`	A character string with the name of the variable that distinguishes healthy from diseased individuals.
`tag.h`	The value codifying healthy individuals in the variable `group`.

bw	A character string specifying which method to use to select the bandwidths. AIC specifies expected Kullback-Leibler cross-validation, and LS specifies least-squares cross-validation. Defaults to LS. For details see R-package np.
regtype	A character string specifying which type of kernel estimator to use for the regression function (see Details). LC specifies a local-constant estimator (Nadaraya-Watson) and LL specifies a local-linear estimator. Defaults to LC. For details see R-package np.
data	Data frame representing the data and containing all needed variables.
newdata	Optional data frame containing the values of the covariates at which the covariate-specific ROC curve (AUC and pAUC, if computed) will be computed. If not supplied, the function cROCData is used to build a default dataset.
pauc	A list of control values to replace the default values returned by the function pauccontrol. This argument is used to indicate whether the partial area under the covariate-adjusted ROC curve should be computed, and in case it is computed, , whether the focus should be placed on restricted false positive fractions (FPFs) or on restricted true positive fractions (TPFs), and the upper bound for the FPF (if focus is FPF) or the lower bound for the TPF (if focus is TPF).
p	Set of false positive fractions (FPF) at which to estimate the covariate-specific ROC curve. This set is also used to compute the area under the covariate-specific ROC curve using Simpson's rule. Thus, the length of the set should be an odd number, and it should be rich enough for an accurate estimation.
B	An integer value specifying the number of bootstrap resamples for the construction of the confidence intervals. The default is 1000.
ci.level	An integer value (between 0 and 1) specifying the confidence level. The default is 0.95.
parallel	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
ncpus	An integer with the number of processes to be used in parallel operation. Defaults to 1.
cl	An object inheriting from class cluster (from the parallel package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates the covariate-specific ROC curve (cROC) defined as

$$ROC(p|x) = 1 - F_D\{F_{\bar{D}}^{-1}(1-p|x)|x\},$$

where

$$F_D(y|x) = Pr(Y_D \leq y|X_D = x),$$

$$F_{\bar{D}}(y|x) = Pr(Y_{\bar{D}} \leq y|X_{\bar{D}} = x).$$

Note that, for the sake of clarity, we assume that the covariate of interest is the same in both healthy and diseased populations. In particular, the method implemented in this function estimates $F_D(\cdot|x)$

and $F_{\bar{D}}(\cdot|x)$ assuming a nonparametric location-scale regression model for Y in each population separately, i.e.,

$$Y_D = \mu_D(X_D) + \sigma_D(X_D)\varepsilon_D,$$

$$Y_{\bar{D}} = \mu_{\bar{D}}(X_{\bar{D}}) + \sigma_{\bar{D}}(X_{\bar{D}})\varepsilon_{\bar{D}},$$

where $\mu_D(x) = E(Y_D|X_D = x)$, $\mu_{\bar{D}}(x) = E(Y_{\bar{D}}|X_{\bar{D}} = x)$ (regression function), $\sigma^2_D(x) = Var(Y_D|X_D = x)$, $\sigma^2_{\bar{D}}(x) = Var(Y_{\bar{D}}|X_{\bar{D}} = x)$ (variance functions), and ε_D and $\varepsilon_{\bar{D}}$ have zero mean, variance one, and distribution functions G_D and $G_{\bar{D}}$, respectively. In this case, the covariate-specific ROC curve can be expressed as

$$ROC(p|x) = 1 - G_D\{a(\mathbf{x}) + b(\mathbf{x})G^{-1}_{\bar{D}}(1-p)\},$$

where $a(x) = \frac{\mu_{\bar{D}}(x)-\mu_D(x)}{\sigma_D(x)}$, $b(x) = \frac{\sigma_{\bar{D}}(x)}{\sigma_D(x)}$, and G_D and $G_{\bar{D}}$ are the distribution functions of ε_D and $\varepsilon_{\bar{D}}$, respectively. By default, for both the healthy and diseased population, both the regression and variance functions are estimated using the Nadaraya-Watson estimator (LC), and the bandwidth are selected using least-squares cross-validation (LS). Implementation relies on the R-package np. No assumptions are made about G_D and $G_{\bar{D}}$, which are empirically estimated on the basis of standardised residuals.

The covariate-specific area under the curve is

$$AUC(\mathbf{x}) = \int_0^1 ROC(p|\mathbf{x})dp,$$

and is computed numerically (using Simpson's rule). With regard to the partial area under the curve, when `focus = "FPF"` and assuming an upper bound u_1 for the FPF, what it is computed is

$$pAUC_{FPF}(u_1|\mathbf{x}) = \int_0^{u_1} ROC(p|\mathbf{x})dp,$$

where again the integral is approximated numerically (Simpson's rule). The returned value is the normalised pAUC, $pAUC_{FPF}(u_1|\mathbf{x})/u_1$ so that it ranges from $u_1/2$ (useless test) to 1 (perfect marker). Conversely, when `focus = "TPF"`, and assuming a lower bound for the TPF of u_2, the partial area corresponding to TPFs lying in the interval $(u_2, 1)$ is computed as

$$pAUC_{TPF}(u_2|\mathbf{x}) = \int_{u_2}^1 ROC_{TNF}(p|\mathbf{x})dp,$$

where $ROC_{TNF}(p|\mathbf{x})$ is a 270° rotation of the ROC curve, and it can be expressed as $ROC_{TNF}(p|\mathbf{x}) = F_{\bar{D}}\{F^{-1}_D(1-p|\mathbf{x})|\mathbf{x}\} = G_{\bar{D}}\{\frac{\mu_D(x)-\mu_{\bar{D}}(x)}{\sigma_{\bar{D}}(x)} + G^{-1}_D(1-p)\frac{\sigma_D(x)}{\sigma_{\bar{D}}(x)}\}$. Again, the computation of the integral is done via Simpson's rule. The returned value is the normalised pAUC, $pAUC_{TPF}(u_2|\mathbf{x})/(1-u_2)$, so that it ranges from $(1-u_2)/2$ (useless test) to 1 (perfect test).

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`newdata`	A data frame containing the values of the covariates at which the covariate-specific ROC curve (AUC and pAUC, if required) was computed.

data	The original supplied data argument.
missing.ind	A logical value indicating whether for each pair of observations (test outcomes and covariates) missing values occur.
marker	The name of the diagnostic test variable in the dataframe.
group	The value of the argument group used in the call.
tag.h	The value of the argument tag.h used in the call.
covariate	The value of the argument covariate used in the call.
p	Set of false positive fractions (FPF) at which the covariate-specific ROC curve has been estimated.
ci.level	The value of the argument ci.level used in the call.
ROC	Estimated covariate-specific ROC curve (AROC), and ci.level*100% pointwise confidence band (if computed).
AUC	Estimated area under the covariate-specific ROC curve, and ci.level*100% confidence interval (if computed).
pAUC	If computed, estimated partial area under the covariate-adjusted ROC curve and ci.level*100% confidence interval (if computed). Note that the returned values are normalised, so that the maximum value is one.
fit	Named list of length two, with components 'h' (healthy) and 'd' (diseased). Each component of the list contains the following information: (1) bw.mean: An object of class npregbw with the selected bandwidth for the nonparametric regression function. For further details, see R-package np. (2) bw.var: An object of class npregbw with the selected bandwidth for the nonparametric variance function. For further details, see R-package np. (3) fit.mean: An object of class npreg with the nonparametric regression function estimate. For further details, see R-package np. (4) fit.var: An object of class npreg with the nonparametric variance function estimate. For further details, see R-package np.

References

Hayfield, T., and Racine, J. S.(2008). Nonparametric Econometrics: The np Package. Journal of Statistical Software 27(5). URL http://www.jstatsoft.org/v27/i05/.

Rodriguez-Alvarez, M. X., Roca-Pardinas, J., and Cadarso-Suarez, C. (2011). ROC curve and covariates: extending induced methodology to the non-parametric framework. Statistics and Computing, **21**, 483–499.

See Also

AROC.bnp, AROC.sp, AROC.kernel, pooledROC.BB, pooledROC.emp, pooledROC.kernel, pooledROC.dpm, cROC.kernel or cROC.sp.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
```

```
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

cROC_kernel <- cROC.kernel(marker = "l_marker1",
               covariate = "age",
               group = "status",
               tag.h = 0,
               data = newpsa,
               bw = "LS",
               regtype = "LC",
               p = seq(0, 1, len = 101),
               pauc = pauccontrol(compute = TRUE, value = 0.5, focus = "FPF"),
               B = 500)

plot(cROC_kernel)

summary(cROC_kernel )
```

---

cROC.sp *Parametric and semiparametric frequentist inference of the covariate-specific ROC curve (cROC).*

Description

This function estimates the covariate-specific ROC curve (cROC) using the parametric approach proposed by Faraggi (2003) and the semiparametric approach proposed by Pepe (1998).

Usage

```
cROC.sp(formula.h, formula.d, group, tag.h, data,
  newdata, est.cdf = c("normal", "empirical"),
  pauc = pauccontrol(), p = seq(0, 1, l = 101), B = 1000, ci.level = 0.95,
  parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

formula.h	A formula object specifying the location regression model to be fitted in the healthy population (see Details).
formula.d	A formula object specifying the location regression model to be fitted in the diseased population (see Details).
group	A character string with the name of the variable that distinguishes healthy from diseased individuals.
tag.h	The value codifying the healthy individuals in the variable group.
data	Data frame representing the data and containing all needed variables.

newdata	Optional data frame containing the values of the covariates at which the covariate-specific ROC curve (AUC and pAUC, if required) will be computed. If not supplied, the function cROCData is used to build a default dataset.
est.cdf	A character string. It indicates how the conditional distribution functions of the diagnostic test in healthy and diseased populations are estimated. Options are "normal" and "empirical" (see Details). The default is "normal".
pauc	A list of control values to replace the default values returned by the function pauccontrol. This argument is used to indicate whether the partial area under the covariate-adjusted ROC curve should be computed, and in case it is computed, whether the focus should be placed on restricted false positive fractions (FPFs) or on restricted true positive fractions (TPFs), and the upper bound for the FPF (if focus is FPF) or the lower bound for the TPF (if focus is TPF).
p	Set of false positive fractions (FPF) at which to estimate the covariate-specific ROC curve. This set is also used to compute the area under the covariate-specific ROC curve using Simpson's rule. Thus, the length of the set should be an odd number, and it should be rich enough for an accurate estimation.
B	An integer value specifying the number of bootstrap resamples for the construction of the confidence intervals. By default 1000.
ci.level	An integer value (between 0 and 1) specifying the confidence level. The default is 0.95.
parallel	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
ncpus	An integer with the number of processes to be used in parallel operation. Defaults to 1.
cl	An object inheriting from class cluster (from the parallel package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates the covariate-specific ROC curve (cROC) defined as

$$ROC(p|\mathbf{x}) = 1 - F_D\{F_{\bar{D}}^{-1}(1-p|\mathbf{x})|\mathbf{x}\},$$

where

$$F_D(y|\mathbf{x}) = Pr(Y_D \leq y|\mathbf{X}_D = \mathbf{x}),$$
$$F_{\bar{D}}(y|\mathbf{x}) = Pr(Y_{\bar{D}} \leq y|\mathbf{X}_{\bar{D}} = \mathbf{x}).$$

Note that, for the sake of clarity, we assume that the covariates of interest are the same in both healthy and diseased populations. In particular, the method implemented in this function estimates $F_D(\cdot|\mathbf{x})$ and $F_{\bar{D}}(\cdot|\mathbf{x})$ assuming a (semiparametric) location regression model for Y in each population separately, i.e.,

$$Y_D = \mathbf{X}_D^T\beta_D + \sigma_D\varepsilon_D,$$
$$Y_{\bar{D}} = \mathbf{X}_{\bar{D}}^T\beta_{\bar{D}} + \sigma_{\bar{D}}\varepsilon_{\bar{D}},$$

such that the covariate-specific ROC curve can be expressed as

$$ROC(p|\mathbf{x}) = 1 - G_D\{a(\mathbf{x}) + bG_{\bar{D}}^{-1}(1-p)\},$$

where $a(\mathbf{x}) = \mathbf{x}^T \frac{\beta_{\bar{D}} - \beta_D}{\sigma_D}$, $b = \frac{\sigma_{\bar{D}}}{\sigma_D}$, and G_D and $G_{\bar{D}}$ are the distribution functions of ε_D and $\varepsilon_{\bar{D}}$, respectively. In line with the assumptions made about the distributions of ε_D and $\varepsilon_{\bar{D}}$, estimators will be referred to as: (a) "normal", where Gaussian errors are assumed, i.e., $G_D(y) = G_{\bar{D}}(y) = \Phi(y)$ (Faraggi, 2003); and, (b) "empirical", where no assumptios are made about the distribution (in this case, G_D and $G_{\bar{D}}$ are empirically estimated on the basis of standardised residuals (Pepe, 1998)).

The covariate-specific area under the curve is

$$AUC(\mathbf{x}) = \int_0^1 ROC(p|\mathbf{x})dp.$$

When Gaussian errors are assumed, there is a closed-form expression for the covariate-specific AUC, which is used in the package. In contrast, when no assumptios are made about the distributionis of the errors, the integral is computed numerically using Simpson's rule. With regard to the partial area under the curve, when `focus = "FPF"` and assuming an upper bound u_1 for the FPF, what it is computed is

$$pAUC_{FPF}(u_1|\mathbf{x}) = \int_0^{u_1} ROC(p|\mathbf{x})dp.$$

Again, when Gaussian errors are assumed, there is a closed-form expression (Hillis and Metz, 2012). Otherwise, the integral is approximated numerically (Simpson's rule). The returned value is the normalised pAUC, $pAUC_{FPF}(u_1|\mathbf{x})/u_1$ so that it ranges from $u_1/2$ (useless test) to 1 (perfect marker). Conversely, when `focus = "TPF"`, and assuming a lower bound for the TPF of u_2, the partial area corresponding to TPFs lying in the interval $(u_2, 1)$ is computed as

$$pAUC_{TPF}(u_2|\mathbf{x}) = \int_{u_2}^1 ROC_{TNF}(p|\mathbf{x})dp,$$

where $ROC_{TNF}(p|\mathbf{x})$ is a 270° rotation of the ROC curve, and it can be expressed as$ROC_{TNF}(p|\mathbf{x}) = F_{\bar{D}}\{F_D^{-1}(1-p|\mathbf{x})|\mathbf{x}\} = G_{\bar{D}}\{\frac{\mu_D(\mathbf{x}) - \mu_{\bar{D}}(\mathbf{x})}{\sigma_{\bar{D}}(\mathbf{x})} + G_D^{-1}(1-p)\frac{\sigma_D(\mathbf{x})}{\sigma_{\bar{D}}(\mathbf{x})}\}$. Again, when Gaussian errors are assumed, there is a closed-form expression (Hillis and Metz, 2012). Otherwise, the integral is approximated numerically (Simpson's rule). The returned value is the normalised pAUC, $pAUC_{TPF}(u_2|\mathbf{x})/(1-u_2)$, so that it ranges from $(1-u_2)/2$ (useless test) to 1 (perfect test).

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`newdata`	A data frame containing the values of the covariates at which the covariate-specific ROC curve (AUC and pAUC, if required) was computed.
`data`	The original supplied data argument.
`missing.ind`	A logical value indicating whether for each pair of observations (test outcomes and covariates) missing values occur.
`marker`	The name of the diagnostic test variable in the dataframe.
`group`	The value of the argument `group` used in the call.
`tag.h`	The value of the argument `tag.h` used in the call.
`formula`	Named list of length two with the value of the arguments `formula.h` and `formula.d` used in the call.

est.cdf	The value of the argument est.cdf used in the call.
p	Set of false positive fractions (FPF) at which the covariate-specific ROC curves have been estimated.
ci.level	The value of the argument ci.level used in the call.
ROC	Estimated covariate-specific ROC curve, and ci.level*100% pointwise confidence intervals (if computed).
AUC	Estimated area under the covariate-specific ROC curve, and ci.level*100% confidence interval (if computed).
pAUC	If computed, estimated partial area under the covariate-adjusted ROC curve and ci.level*100% confidence interval (if computed). Note that the returned values are normalised, so that the maximum value is one.
fit	Named list of length two, with components 'h' (healthy) and 'd' (diseased). Each component contains an object of class lm with the fitted regression model.
coeff	Estimated regression coefficients (and ci.level*100% confidence interval if B greater than zero) from the fit of the linear model in the healthy and diseased population, as specified in formula.h and formula.d, respectively.

References

Faraggi, D. (2003). Adjusting receiver operating characteristic curves and related indices for covariates. *The Statistician* **52**, 179–192.

Hillis, S. L. and Metz, C.E. (2012). An Analytic Expression for the Binormal Partial Area under the ROC Curve. Academic Radiology, **19**, 1491–1498.

Pepe, M.S. (1998). Three approaches to regression analysis of receiver operating characteristic curves for continuous test results. *Biometrics* **54**, 124–135.

See Also

AROC.bnp, AROC.sp, AROC.kernel, pooledROC.BB, pooledROC.emp, pooledROC.kernel, pooledROC.dpm, cROC.bnp, cROC.sp or AROC.kernel.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

# Covariate for prediction
agep <- seq(min(newpsa$age), max(newpsa$age), length = 50)
df.pred <- data.frame(age = agep)


cROC_sp_normal <- cROC.sp(formula.h = l_marker1 ~ age,
                          formula.d = l_marker1 ~ age,
```

```
                          group = "status",
                          tag.h = 0,
                          data = newpsa,
                          newdata = df.pred,
                          est.cdf = "normal",
                          pauc = list(compute = TRUE, value = 0.5, focus = "FPF"),
                          p = seq(0, 1, l = 101),
                          B = 500)
summary(cROC_sp_normal)

plot(cROC_sp_normal)
```

`cROCData` *Selects an adequate set of points from a data set for obtaining predictions.*

Description

Selects an adequate set of points from a data set for obtaining predictions

Usage

```
cROCData(data, names.cov, group)
```

Arguments

`data`	Data set from which the new set of covariate values is obtained.
`names.cov`	Character vector with the names of the covariates to be included in the new data set.
`group`	A character string with the name of the variable in the original data set that distinguishes healthy from diseased individuals.

Value

A data frame containing selected values of all needed covariates. For those that are continuous, 30 different values are selected.

See Also

`AROC.bnp`, `cROC.bnp`, `cROC.sp`, `cROC.kernel`, `compute.threshold.cROC` or `compute.threshold.AROC`.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

newdf <- cROCData(newpsa, "age", "status")

summary(newdf)
```

`densitycontrol` *(Conditional) density estimates of test outcomes*

Description

This function is used to set various parameters controlling the estimation of the (conditional) density (densities) of test outcomes in both the healthy and diseased groups.

Usage

```
densitycontrol(compute = FALSE, grid.h = NA, grid.d = NA)
```

Arguments

`compute`	Logical value. If TRUE the (conditional) density (densities) of test outcomes in each group, healthy and diseased, are estimated.
`grid.h`	Grid of test outcomes in the healthy group where the (conditional) density (densities) estimates are to be evaluated. Value `NA` signals autoinitialization, with default a vector of length 200 in the range of test outcomes in the healthy group.
`grid.d`	Grid of test outcomes in the diseased group where the (conditional) density (densities) estimates are to be evaluated. Value `NA` signals autoinitialization, with default a vector of length 200 in the range of test outcomes in the diseased group.

Details

The value returned by this function is used as a control argument of the `cROC.bnp` and `pooledROC.dpm` functions.

Value

A list with components for each of the possible arguments.

See Also

`cROC.bnp` and `pooledROC.dpm`

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

cROC_bnp <- cROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
                formula.d = l_marker1 ~ f(age, K = 0),
                group = "status",
                tag.h = 0,
                data = newpsa,
                standardise = TRUE,
                p = seq(0, 1, len = 101),
                compute.lpml = TRUE,
                compute.WAIC = TRUE,
                compute.DIC = TRUE,
                pauc = pauccontrol(compute = TRUE, value = 0.5, focus = "FPF"),
                density = densitycontrol(compute = TRUE, grid.h = NA, grid.d = NA),
                mcmc = mcmccontrol(nsave = 500, nburn = 100, nskip = 1))
```

`densitycontrol.aroc` *Conditional density estimates of test outcomes in the healthy population*

Description

This function is used to set various parameters controlling the estimation of the conditional densities of test outcomes in the healthy group.

Usage

```
densitycontrol.aroc(compute = FALSE, grid.h = NA, newdata = NA)
```

Arguments

`compute`	Logical value. If TRUE the conditional densities of test outcomes in the healthy group are estimated.
`grid.h`	Grid of test outcomes in the healthy group where the conditional density estimates are to be evaluated. Value `NA` signals autoinitialization, with default a vector of length 200 in the range of test outcomes in the healthy group.
`newdata`	Data frame containing the values of the covariates at which the conditional density estimates are computed.

Details

The value returned by this function is used as a control argument of the AROC.bnp function.

Value

A list with components for each of the possible arguments.

See Also

AROC.bnp

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

# Covariate for prediction
agep <- seq(min(newpsa$age), max(newpsa$age), length = 5)
df.pred <- data.frame(age = agep)


AROC_bnp <- AROC.bnp(formula.h =  l_marker1 ~ f(age, K = 0),
                     group = "status",
                     tag.h = 0,
                     data = newpsa,
                     standardise = TRUE,
                     p = seq(0, 1, len = 101),
                     compute.lpml = TRUE,
                     compute.WAIC = TRUE,
                     compute.DIC = TRUE,
                     pauc = pauccontrol(compute = TRUE, value = 0.5, focus = "FPF"),
               density = densitycontrol.aroc(compute = TRUE, grid.h = NA, newdata = df.pred),
                     mcmc = mcmccontrol(nsave = 500, nburn = 100, nskip = 1)
)
```

---

endosyn *Simulated endocrine data.*

Description

The endosyn data set was simulated based on the data analysed in Rodriguez-Alvarez et al. (2011a,b) and Inacio de Carvalho and Rodriguez-Alvarez (2018); and presented in Botana et al. (2007) and Tome et al. (2008). The aim of these studies was to use the body mass index (BMI) to detect patients having a higher risk of cardiovascular problems, ascertaining the possible effect of age and gender on the accuracy of this measure.

Usage

```
data("endosyn")
```

Format

A data frame with 2840 observations on the following 4 variables.

`gender` patient's gender. Factor with `Men` and `Women` levels.

`age` patient's age.

`cvd_idf` true disease status (presence/absence of two of more cardiovascular risk factors according to the International Diabetes Federation). Numerical vector (0 = absence, 1 = presence).

`bmi` patient's body mass index.

Source

Botana, M.A., Mato, J.A., Cadarso-Suarez, C., Tome, M.A., Perez-Fernandez, R., Fernandez-Mario, A., Rego-Iraeta, A., Solache, I. (2007). Overweight, obesity and central obesity prevalences in the region of Galicia in Northwest Spain. Obesity and Metabolism, 3, 106–115.

Tome, M.A., Botana, M.A., Cadarso-Suarez, C., Rego-Iraeta, A., Fernandez-Mario, A., Mato, J.A, Solache, I., Perez-Fernandez, R. (2008). Prevalence of metabolic syndrome in Galicia (NW Spain) on four alternative definitions and association with insulin resistance. Journal of Endocrinological Investigation, 32, 505–511.

References

Inacio de Carvalho, V., and Rodriguez-Alvarez, M. X. (2022). The Covariate-Adjusted ROC Curve: The Concept and Its Importance, Review of Inferential Methods, and a New Bayesian Estimator. *Statistical Science*, **37**, 541 -561.

Rodriguez-Alvarez, M.X., Roca-Pardinas, J. and Cadarso-Suarez, C. (2011a). ROC curve and covariates: extending induced methodology to the non-parametric framework. *Statistics and Computing*, **21(4)**, 483–49.

Rodriguez- Alvarez, M.X., Roca-Pardinas, J. and Cadarso-Suarez, C. (2011b). A new flexible direct ROC regression model - Application to the detection of cardiovascular risk factors by anthropometric measures. *Computational Statistics and Data Analysis*, **55(12)**, 3257–3270.

Examples

```
data(endosyn)
summary(endosyn)
```

mcmccontrol *Markov chain Monte Carlo (MCMC) parameters*

Description

This function is used to set various parameters controlling the Markov chain Monte Carlo (MCMC) parameters.

Usage

```
mcmccontrol(nsave = 8000, nburn = 2000, nskip = 1)
```

Arguments

nsave	An integer giving the total number of scans to be saved (does not include the burn-in and thinning iterations).
nburn	An integer giving the number of burn-in scans.
nskip	An integer giving the thinning interval,

Details

The value returned by this function is used as a control argument of the AROC.bnp, cROC.bnp, and pooledROC.dpm functions.

Value

A list with components for each of the possible arguments.

See Also

pooledROC.dpm, AROC.bnp and cROC.bnp

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

cROC_bnp <- cROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
               formula.d = l_marker1 ~ f(age, K = 0),
               group = "status",
               tag.h = 0,
               data = newpsa,
               standardise = TRUE,
```

```
p = seq(0, 1, len = 101),
compute.lpml = TRUE,
compute.WAIC = TRUE,
compute.DIC = TRUE,
pauc = pauccontrol(compute = TRUE, value = 0.5, focus = "FPF"),
density = densitycontrol(compute = TRUE, grid.h = NA, grid.d = NA),
mcmc = mcmccontrol(nsave = 500, nburn = 100, nskip = 1))
```

`pauccontrol` *Partial area under the covariate-adjusted/covariate-specific/pooled ROC curve*

Description

Used to set various parameters controlling the estimation of the partial area under the covariate-adjusted/covariate-specific/pooled ROC curve .

Usage

```
pauccontrol(compute = FALSE, focus = c("FPF", "TPF"), value = 1)
```

Arguments

`compute`	Logical value. If TRUE the partial area under the covariate-adjusted/covariate-specific/pooled ROC curve is estimated.
`focus`	Whether computation should be done over a restricted range of false positive fractions (FPF) or a restricted range of true positive fractions (TPF).
`value`	Numeric value. Pre-specified upper bound for the FPF (if focus = "FPF") or lower bound for the TPF (if focus = "TPF").

Details

The value returned by this function is used as a control argument of the `AROC.bnp`, `AROC.sp`, `AROC.kernel`, `cROC.bnp`, `cROC.sp`, `cROC.kernel` functions.

Value

A list with components for each of the possible arguments.

See Also

`AROC.bnp`

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

# Covariate for prediction
agep <- seq(min(newpsa$age), max(newpsa$age), length = 5)
df.pred <- data.frame(age = agep)


cROC_sp_normal <- cROC.sp(formula.h = l_marker1 ~ age,
                          formula.d = l_marker1 ~ age,
                          group = "status",
                          tag.h = 0,
                          data = newpsa,
                          newdata = df.pred,
                          est.cdf = "normal",
                          pauc = list(compute = TRUE, value = 0.5, focus = "FPF"),
                          p = seq(0, 1, l = 101),
                          B = 10)
```

---

plot.AROC *Default AROC plotting*

Description

Takes a fitted AROC object produced by AROC.bnp(), AROC.sp(), or AROC.kernel() and plots the covariate-adjusted ROC curve (AROC) and associated area under the AROC (AAUC).

Usage

```
## S3 method for class 'AROC'
plot(x, main = NULL, ...)
```

Arguments

x	An object of class AROC as produced by AROC.bnp(), AROC.sp(), or AROC.kernel().
main	Character string with the overall title for the plot. If NULL, the default, the method used to estimate the AROC curve is depicted.
...	Further arguments passed to or from other methods.

See Also

AROC.bnp, AROC.sp or AROC.kernel

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

AROC_bnp <- AROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
group = "status", tag.h = 0, data = newpsa, standardise = TRUE,
p = seq(0,1,l=101), compute.lpml = TRUE, compute.WAIC = TRUE,
compute.DIC = TRUE)

plot(AROC_bnp)
```

---

plot.cROC *Default cROC plotting*

Description

Takes a fitted cROC object produced by cROC.bnp(), cROC.sp(), or cROC.kernel() and plots the covariate-specific ROC curve (cROC) and associated area under the cROC curve. The suitable type of graphic is chosen according to the number and nature of the covariates.

Usage

```
## S3 method for class 'cROC'
plot(x, ask = TRUE, ...)
```

Arguments

x	An object of class cROC as produced by cROC.bnp(), cROC.sp(), or cROC.kernel().
ask	A logical value. If TRUE, the default, the user is asked for confirmation, before a new figure is drawn.
...	Further arguments passed to or from other methods.

See Also

cROC.bnp, cROC.sp or cROC.kernel

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

cROC_bnp <- cROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
                formula.d = l_marker1 ~ f(age, K = 0),
                group = "status",
                tag.h = 0,
                data = newpsa,
                standardise = TRUE,
                p = seq(0, 1, len = 101),
                compute.lpml = TRUE,
                compute.WAIC = TRUE,
                compute.DIC = TRUE,
                pauc = pauccontrol(compute = TRUE, value = 0.5, focus = "FPF"),
                density = densitycontrol(compute = TRUE, grid.h = NA, grid.d = NA),
                mcmc = mcmccontrol(nsave = 500, nburn = 100, nskip = 1))

plot(cROC_bnp)
```

`plot.pooledROC` *Default pooledROC plotting*

Description

Takes a fitted `pooledROC` object produced by `pooledROC.BB`, `pooledROC.emp`, `pooledROC.kernel`, or `pooledROC.dpm` and plots the pooled ROC curve and associated area under the ROC curve (AUC).

Usage

```
## S3 method for class 'pooledROC'
plot(x, main = NULL, ...)
```

Arguments

`x`	An object of class `pooledROC` as produced by `pooledROC.BB`, `pooledROC.emp`, `pooledROC.kernel` or `pooledROC.dpm`.
`main`	Character string with the overall title for the plot. If NULL, the default, the method used to estimate the pooled ROC curve is depicted.
`...`	Further arguments passed to or from other methods.

See Also

pooledROC.BB, pooledROC.emp, pooledROC.kernel or pooledROC.dpm

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

m0_emp <- pooledROC.emp(marker = "l_marker1", group = "status",
tag.h = 0, data = newpsa, p = seq(0,1,l=101), B = 500)

summary(m0_emp)

plot(m0_emp)
```

---

pooledROC.BB *Bayesian bootstrap estimation of the pooled ROC curve.*

Description

This function estimates the pooled ROC curve using the Bayesian bootstrap estimator proposed by Gu et al. (2008).

Usage

```
pooledROC.BB(marker, group, tag.h, data,
 p = seq(0, 1, l = 101), B = 5000, ci.level = 0.95, pauc = pauccontrol(),
   parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

marker	A character string with the name of the diagnostic test variable.
group	A character string with the name of the variable that distinguishes healthy from diseased individuals.
tag.h	The value codifying healthy individuals in the variable group.
data	Data frame representing the data and containing all needed variables.
p	Set of false positive fractions (FPF) at which to estimate the pooled ROC curve.
B	An integer value specifying the number of Bayesian bootstrap resamples. By default 5000.

`ci.level` An integer value (between 0 and 1) specifying the level for the credible interval. The default is 0.95.

`pauc` A list of control values to replace the default values returned by the function `pauccontrol`. This argument is used to indicate whether the partial area under the pooled ROC curve should be computed, and in case it is computed, whether the focus should be placed on restricted false positive fractions (FPFs) or on restricted true positive fractions (TPFs), and the upper bound for the FPF (if focus is FPF) or the lower bound for the TPF (if focus is TPF).

`parallel` A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".

`ncpus` An integer with the number of processes to be used in parallel operation. Defaults to 1.

`cl` An object inheriting from class `cluster` (from the `parallel` package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates the pooled ROC curve (ROC) defined as

$$ROC(p) = 1 - F_D\{F_{\bar{D}}^{-1}(1-p)\},$$

where

$$F_D(y) = Pr(Y_D \leq y),$$

$$F_{\bar{D}}(y) = Pr(Y_{\bar{D}} \leq y).$$

The method implemented in this function makes use of the equivalence (see Gu et al., 2008)

$$ROC(p) = 1 - F_D\{F_{\bar{D}}^{-1}(1-p)\} = 1 - Pr(1 - F_{\bar{D}}(Y_D) \leq p),$$

and estimates both $F_{\bar{D}}$ and the outer probability using the Bayesian bootstrap resampling distribution.

Regarding the area under the curve, we note that

$$AUC = \int_0^1 ROC(p)dp = 1 - E\{U_D\},$$

where $U_D = 1 - F_{\bar{D}}(Y_D)$. In our implementation, the expectation is computed using the Bayesian bootstrap (using the same weights as those used to estimate the pooled ROC). As far as the partial area under the curve is concerned, when `focus = "FPF"` and assuming an upper bound u_1 for the FPF, what it is computed is

$$pAUC_{FPF}(u_1) = \int_0^{u_1} ROC(p)dp = u_1 - E\{U_{D,u_1}\},$$

where $U_{D,u_1} = \min\{u_1, 1 - F_{\bar{D}}(Y_D)\}$. Again, the expectation is computed using the Bayesian bootstrap. The returned value is the normalised pAUC, $pAUC_{FPF}(u_1)/u_1$ so that it ranges from $u_1/2$ (useless test) to 1 (perfect marker). Conversely, when `focus = "TPF"`, and assuming a lower

bound for the TPF of u_2, the partial area corresponding to TPFs lying in the interval $(u_2, 1)$ is computed as

$$pAUC_{TPF}(u_2) = \int_{u_2}^{1} ROC_{TNF}(p)dp,$$

where $ROC_{TNF}(p)$ is a 270° rotation of the ROC curve, and it can be expressed as $ROC_{TNF}(p) = F_{\bar{D}}\{F_D^{-1}(1-p)\}$. Thus

$$pAUC_{TPF}(u_2) = \int_{u_2}^{1} ROC_{TNF}(p)dp = E\{U_{\bar{D},u_2} - u_2\},$$

where $U_{\bar{D},u_2} = \max\{u_2, 1 - F_D(Y_{\bar{D}})\}$, and the expectation is computed using the Bayesian bootstrap. The returned value is the normalised pAUC, $pAUC_{TPF}(u_2)/(1-u_2)$, so that it ranges from $(1-u_2)/2$ (useless test) to 1 (perfect test).

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`marker`	A list with the diagnostic test outcomes in the healthy (h) and diseased (d) groups.
`missing.ind`	A logical value indicating whether missing values occur.
`p`	Set of false positive fractions (FPF) at which the pooled ROC curve has been estimated.
`ci.level`	The value of the argument `ci.level` used in the call.
`ROC`	Estimated pooled ROC curve, and corresponding `ci.level`*100% pointwise credible band.
`AUC`	Estimated pooled AUC, and corresponding `ci.level`*100% credible interval.
`pAUC`	If computed, estimated partial area under the pooled ROC curve (posterior mean) and `ci.level`*100% credible interval. Note that the returned values are normalised, so that the maximum value is one (see more on Details).
`weights`	list with the Dirichlet weights (involved in the estimation) in the healthy (h) and diseased (d) groups. These are matrices of dimension n0 x B and n1 x B, where n0 is the number of healthy individuals and n1 is the number of diseased individuals.

References

Gu, J., Ghosal, S., and Roy, A. (2008). Bayesian bootstrap estimation of ROC curve. Statistics in Medicine, **27**, 5407–5420.

See Also

`AROC.bnp`, `AROC.sp`, `AROC.kernel`, `pooledROC.BB`, `pooledROC.emp`, `pooledROC.kernel`, `pooledROC.dpm`, `cROC.bnp`, `cROC.sp` or `AROC.kernel`.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

m0_BB <- pooledROC.BB(marker = "l_marker1", group = "status",
tag.h = 0, data = newpsa, p = seq(0,1,l=101), B = 5000)

summary(m0_BB)

plot(m0_BB)
```

`pooledROC.dpm` *Nonparametric Bayesian inference of the pooled ROC curve*

Description

This function estimates the pooled ROC curve using a Dirichlet process mixture of normals model as proposed by Erkanli et al. (2006).

Usage

```
pooledROC.dpm(marker, group, tag.h, data,
  standardise = TRUE, p = seq(0, 1, l = 101), ci.level = 0.95,
  compute.lpml = FALSE, compute.WAIC = FALSE, compute.DIC = FALSE,
  pauc = pauccontrol(), density = densitycontrol(),
  prior.h = priorcontrol.dpm(), prior.d = priorcontrol.dpm(),
  mcmc = mcmccontrol(),
  parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

`marker`	A character string with the name of the diagnostic test variable.
`group`	A character string with the name of the variable that distinguishes healthy from diseased individuals.
`tag.h`	The value codifying healthy individuals in the variable `group`.
`data`	Data frame representing the data and containing all needed variables.
`standardise`	A logical value. If TRUE the test outcomes are standardised (so that the resulting test outcomes have mean zero and standard deviation of one). The default is TRUE.

p	Set of false positive fractions (FPF) at which to estimate the pooled ROC curve. This set is also used to compute the area under the ROC curve (AUC) using Simpson's rule. Thus, the length of the set should be an odd number, and it should be rich enough for an accurate estimation.
ci.level	An integer value (between 0 and 1) specifying the level for the credible interval. The default is 0.95.
compute.lpml	A logical value. If TRUE, the log pseudo marginal likelihood (LPML, Geisser and Eddy, 1979) and the conditional predictive ordinates (CPO) are computed.
compute.WAIC	A logical value. If TRUE, the widely applicable information criterion (WAIC, Gelman et al., 2014; Watanabe, 2010) is computed.
compute.DIC	A logical value. If TRUE, the deviance information criterion is computed.
pauc	A list of control values to replace the default values returned by the function pauccontrol. This argument is used to indicate whether the partial area under the pooled ROC curve should be computed, and in case it is computed, whether the focus should be placed on restricted false positive fractions (FPFs) or on restricted true positive fractions (TPFs), and the upper bound for the FPF (if focus is FPF) or the lower bound for the TPF (if focus is TPF).
density	A list of control values to replace the default values returned by the function densitycontrol. This argument is used to indicate whether the densities of the marker in the healthy and diseased population should be computed, and in case it is to be computed, at which grid of test outcomes in each of the populations.
prior.h	Hyparameter specification for the healthy population. A list of control values to replace the default values returned by the function priorcontrol.dpm. See priorcontrol.dpm for details.
prior.d	Hyparameter specification for the diseased population. A list of control values to replace the default values returned by the function priorcontrol.dpm. See priorcontrol.dpm for details.
mcmc	A list of control values to replace the default values returned by the function mcmccontrol. See mcmccontrol for details.
parallel	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
ncpus	An integer with the number of processes to be used in parallel operation. Defaults to 1.
cl	An object inheriting from class cluster (from the parallel package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates the pooled ROC curve (ROC) defined as

$$ROC(p) = 1 - F_D\{F_{\bar{D}}^{-1}(1-p)\},$$

where

$$F_D(y) = Pr(Y_D \leq y),$$

$$F_{\bar{D}}(y) = Pr(Y_{\bar{D}} \leq y).$$

The method implemented in this function estimates $F_D(\cdot)$ and $F_{\bar{D}}(\cdot)$ by means of a Dirichlet process mixture of normals model. More precisely, and letting $\{y_{\bar{D}i}\}_{i=1}^{n_{\bar{D}}}$ and $\{y_{Dj}\}_{j=1}^{n_D}$ be two independent random samples from the nondiseased and diseased populations, respectively, the model postulated for the distribution function is as follows

$$F_{\bar{D}}(y_{\bar{D}i}) = \sum_{l=1}^{L_{\bar{D}}} \omega_{l\bar{D}} \Phi(y_{\bar{D}i} \mid \mu_{l\bar{D}}, \sigma^2_{l\bar{D}}),$$

$$F_D(y_{Dj}) = \sum_{l=1}^{L_D} \omega_{lD} \Phi(y_{Dj} \mid \mu_{lD}, \sigma^2_{lD}),$$

where L_d is pre-specified is a pre-specified upper bound on the number of mixture components ($d \in \{D, \bar{D}\}$). The ω_{ld}'s result from a truncated version of the stick-breaking construction ($\omega_{1d} = v_{1d}$; $\omega_{ld} = v_{ld} \prod_{r<l}(1 - v_{dr})$, $l = 2, \ldots, L_d$; $v_{d1}, \ldots, v_{L_d - 1} \sim$ Beta $(1, \alpha_d)$; $v_{Ld} = 1$, $\alpha_d \sim \Gamma(a_{\alpha_d}, b_{\alpha_d})$), $\beta_{ld} \sim N(m_{0d}, S_{0d})$, and $\sigma_{ld}^{-2} \sim \Gamma(a_d, b_d)$.

The area under the curve is

$$AUC = \int_0^1 ROC(p)dp.$$

When the upper bound on the number of mixture components is 1, i.e., $L_d = 1$ ($d \in \{D, \bar{D}\}$), there is a closed-form expression for the AUC (binormal model), which is used in the package. In contrast, when $L_D > 1$ or $L_{\bar{D}} > 1$, the AUC is computed using results presented in Erkanli et al. (2006). With regard to the partial area under the curve, when `focus = "FPF"` and assuming an upper bound u_1 for the FPF, what it is computed is

$$pAUC_{FPF}(u_1) = \int_0^{u_1} ROC(p)dp.$$

As for the AUC, when $L_d = 1$ ($d \in \{D, \bar{D}\}$), there is a closed-form expression for the $pAUC_{FPF}$ (Hillis and Metz, 2012), and when $L_D > 1$ or $L_{\bar{D}} > 1$ the integral is approximated numerically using Simpson's rule. The returned value is the normalised pAUC, $pAUC_{FPF}(u_1)/u_1$ so that it ranges from $u_1/2$ (useless test) to 1 (perfect marker). Conversely, when `focus = "TPF"`, and assuming a lower bound for the TPF of u_2, the partial area corresponding to TPFs lying in the interval $(u_2, 1)$ is computed as

$$pAUC_{TPF}(u_2) = \int_{u_2}^1 ROC_{TNF}(p)dp,$$

where $ROC_{TNF}(p)$ is a 270° rotation of the ROC curve, and it can be expressed as $ROC_{TNF}(p) = F_{\bar{D}}\{F_D^{-1}(1 - p)\}$. Again, when $L_d = 1$ ($d \in \{D, \bar{D}\}$), there is a closed-form expression for the $pAUC_{TNF}$ (Hillis and Metz, 2012), and when $L_D > 1$ or $L_{\bar{D}} > 1$ the integral is approximated numerically using Simpson's rule. The returned value is the normalised pAUC, $pAUC_{TPF}(u_2)/(1 - u_2)$, so that it ranges from $(1 - u_2)/2$ (useless test) to 1 (perfect test).

It is worth referring that with respect to the computation of the DIC, when $L = 1$, it is computed as in Spiegelhalter et al. (2002), and when $L > 1$, DIC3 as described in Celeux et al. (2006) is computed. Also, for the computation of the conditional predictive ordinates (CPO) we follow the stable version proposed by Gelman et al. (2014).

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`marker`	A list with the diagnostic test outcomes in the healthy (h) and diseased (d) groups.
`missing.ind`	A logical value indicating whether missing values occur.
`p`	Set of false positive fractions (FPF) at which the pooled ROC curve has been estimated.
`ci.level`	The value of the argument `ci.level` used in the call.
`prior`	A list returning the hyperparameter values in the healthy and diseased populations.
`ROC`	Estimated pooled ROC curve, and corresponding `ci.level`*100% pointwise credible band.
`AUC`	Estimated pooled AUC, and corresponding `ci.level`*100% credible interval.
`pAUC`	If computed, estimated partial area under the pooled ROC curve (posterior mean) and `ci.level`*100% pointwise credible band. Note that the returned values are normalised, so that the maximum value is one (see more on Details).
`dens`	Named list of length two, with components 'h' (healthy) and 'd' (diseased). Each component is a list with two components: `grid` (grid of test outcomes where the densities are evaluated) and `dens` (MCMC realisations of the corresponding densities).
`lpml`	If computed, named list of length two, with components 'h' (healthy) and 'd' (diseased). Each component is a list with two components: the log pseudo marginal likelihood (LPML) and the conditional predictive ordinates (CPO).
`WAIC`	If computed, named list of length two, with components 'h' (healthy) and 'd' (diseased). Each component is a list with two components: widely applicable information criterion (WAIC) and associated complexity penalty (pW).
`DIC`	If computed, named list of length two, with components 'h' (healthy) and 'd' (diseased). Each component is a list with two components: deviance information criterion (DIC) and associated complexity penalty (pD).
`fit`	Named list of length two, with components 'h' (healthy) and 'd' (diseased). Each component is a list with the following information: (1)`probs`: matrix of dimension nsave x L with the sampled components' weights; (2) `mu`: matrix of dimension nsave x L with the sampled means; and (3) `sd`: matrix of dimension nsave x L with the sampled standard deviations. Here, nsave is the number of Gibbs sampler iterates saved, and L is the upper bound on the number of mixture components.

References

Erkanli, A., Sung M., Jane Costello, E., and Angold, A. (2006). Bayesian semi-parametric ROC analysis. Statistics in Medicine, **25**, 3905–3928.

Geisser, S. and Eddy, W.F. (1979) A Predictive Approach to Model Selection, Journal of the American Statistical Association, **74**, 153–160.

Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., and Rubin, D.B. (2014). Bayesian Data Analysis, 3rd ed. CRC Press: Boca Raton, FL.

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. Statistics and Computing, **24**, 997–1010.

Hillis, S. L. and Metz, C.E. (2012). An Analytic Expression for the Binormal Partial Area under the ROC Curve. Academic Radiology, **19**, 1491–1498.

Watanabe, S. (2010). Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory. Journal of Machine Learning Research, **11**, 3571–3594.

See Also

AROC.bnp, AROC.sp, AROC.kernel, pooledROC.BB, pooledROC.emp, pooledROC.kernel, pooledROC.dpm, cROC.bnp, cROC.sp or AROC.kernel.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

m0_dpm <- pooledROC.dpm(marker = "l_marker1", group = "status",
            tag.h = 0, data = newpsa, standardise = TRUE,
            p = seq(0,1,l=101), compute.WAIC = TRUE, compute.lpml = TRUE,
            compute.DIC = TRUE,
            prior.h = priorcontrol.dpm(m0 = 0, S0 = 10, a = 2, b = 0.5, alpha = 1,
            L =10),
            prior.d = priorcontrol.dpm(m0 = 0, S0 = 10, a = 2, b = 0.5, alpha = 1,
            L =10),
            mcmc = mcmccontrol(nsave = 400, nburn = 100, nskip = 1))

summary(m0_dpm)

plot(m0_dpm)
```

---

pooledROC.emp *Empirical estimation of the pooled ROC curve.*

Description

This function estimates the pooled ROC curve using the empirical estimator proposed by Hsieh and Turnbull (1996).

Usage

```
pooledROC.emp(marker, group, tag.h, data,
 p = seq(0, 1, l = 101), B = 1000, ci.level = 0.95,
    method = c("ncoutcome", "coutcome"), pauc = pauccontrol(),
   parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

marker	A character string with the name of the diagnostic test variable.
group	A character string with the name of the variable that distinguishes healthy from diseased individuals.
tag.h	The value codifying healthy individuals in the variable group.
data	Data frame representing the data and containing all needed variables.
p	Set of false positive fractions (FPF) at which to estimate the pooled ROC curve.
B	An integer value specifying the number of bootstrap resamples for the construction of the confidence intervals. The default is 1000.
ci.level	An integer value (between 0 and 1) specifying the confidence level. The default is 0.95.
method	A character string specifying if bootstrap resampling (for the confidence intervals) should be done with or without regard to the disease status ("coutcome" or "noutcome"). In both cases, a naive bootstrap is used. By default, the resampling is done conditionally on the disease status.
pauc	A list of control values to replace the default values returned by the function pauccontrol. This argument is used to indicate whether the partial area under the pooled ROC curve should be computed, and in case it is computed, whether the focus should be placed on restricted false positive fractions (FPFs) or on restricted true positive fractions (TPFs), and the upper bound for the FPF (if focus is FPF) or the lower bound for the TPF (if focus is TPF).
parallel	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
ncpus	An integer with the number of processes to be used in parallel operation. Defaults to 1.
cl	An object inheriting from class cluster (from the parallel package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates the pooled ROC curve (ROC) defined as

$$ROC(p) = 1 - F_D\{F_{\bar{D}}^{-1}(1-p)\},$$

where

$$F_D(y) = Pr(Y_D \leq y),$$
$$F_{\bar{D}}(y) = Pr(Y_{\bar{D}} \leq y).$$

The method implemented in this function estimates $F_D(\cdot)$ and $F_{\bar{D}}(\cdot)$ by means of the empirical dsitributions. More precisely, and letting $\{y_{\bar{D}i}\}_{i=1}^{n_{\bar{D}}}$ and $\{y_{Dj}\}_{j=1}^{n_D}$ be two independent random samples from the nondiseased and diseased populations, respectively, the distribution functions in each group take the form

$$\widehat{F}_D(y) = \frac{1}{n_D}\sum_{j=1}^{n_D} I(y_{Dj} \leq y),$$

$$\widehat{F}_{\bar{D}}(y) = \frac{1}{n_{\bar{D}}}\sum_{i=1}^{n_{\bar{D}}} I(y_{\bar{D}i} \leq y).$$

The area under the curve is

$$AUC = \int_0^1 ROC(p)dp$$

and is estimated empirically by means of the Mann-Whitney U-statistic. With regard to the partial area under the curve, when `focus = "FPF"` and assuming an upper bound u_1 for the FPF, what it is computed is

$$pAUC_{FPF}(u_1) = \int_0^{u_1} ROC(p)dp,$$

where again is estimated empirically. The returned value is the normalised pAUC, $pAUC_{FPF}(u_1)/u_1$ so that it ranges from $u_1/2$ (useless test) to 1 (perfect marker). Conversely, when `focus = "TPF"`, and assuming a lower bound for the TPF of u_2, the partial area corresponding to TPFs lying in the interval $(u_2, 1)$ is computed as

$$pAUC_{TPF}(u_2) = \int_{u_2}^1 ROC_{TNF}(p)dp,$$

where $ROC_{TNF}(p)$ is a 270° rotation of the ROC curve, and it can be expressed as $ROC_{TNF}(p) = F_{\bar{D}}\{F_D^{-1}(1-p)\}$. Again, $ROC_{TNF}(p)$ is estimated empirically. The returned value is the normalised pAUC, $pAUC_{TPF}(u_2)/(1-u_2)$, so that it ranges from $(1-u_2)/2$ (useless test) to 1 (perfect test).

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`marker`	A list with the diagnostic test outcomes in the healthy (h) and diseased (d) groups.
`missing.ind`	A logical value indicating whether missing values occur.
`p`	Set of false positive fractions (FPF) at which the pooled ROC curve has been estimated.
`ci.level`	The value of the argument `ci.level` used in the call.
`ROC`	Estimated pooled ROC curve, and corresponding `ci.level`*100% pointwise confidence band (if computed).
`AUC`	Estimated pooled AUC, and corresponding `ci.level`*100% confidence interval (if computed).

pAUC — If computed, estimated partial area under the pooled ROC curve along with its ci.level*100% confidence interval (if B greater than zero). Note that the returned values are normalised, so that the maximum value is one (see more on Details).

References

Hsieh, F., and Turnbull, B.W. (1996). Nonparametric and semiparametric estimation of the receiver operating characteristic curve, The Annals of Statistics, **24**, 25–40.

See Also

AROC.bnp, AROC.sp, AROC.kernel, pooledROC.BB, pooledROC.emp, pooledROC.kernel, pooledROC.dpm, cROC.bnp, cROC.sp or AROC.kernel.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

m0_emp <- pooledROC.emp(marker = "l_marker1", group = "status",
tag.h = 0, data = newpsa, p = seq(0,1,l=101), B = 10,
method = "coutcome", pauc = pauccontrol(compute = TRUE, value = 0.5, focus = "FPF"))

summary(m0_emp)

plot(m0_emp)
```

---

pooledROC.kernel *Kernel-based estimation of the pooled ROC curve.*

Description

This function estimates the pooled ROC curve using the kernel-based density estimator proposed by Zhou et al. (1997).

Usage

```
pooledROC.kernel(marker, group, tag.h, data,
 p = seq(0, 1, l = 101),
 bw = c("SRT", "UCV"), B = 1000, ci.level = 0.95,
 method = c("ncoutcome", "coutcome"), pauc = pauccontrol(),
  parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL)
```

Arguments

marker	A character string with the name of the diagnostic test variable.
group	A character string with the name of the variable that distinguishes healthy from diseased individuals.
tag.h	The value codifying healthy individuals in the variable group.
data	Data frame representing the data and containing all needed variables.
p	Set of false positive fractions (FPF) at which to estimate the pooled ROC curve. This set is also used to compute the area under the ROC curve (AUC) using Simpson's rule. Thus, the length of the set should be an odd number, and it should be rich enough for an accurate estimation.
bw	A character string specifying the density bandwidth selection method. "SRT": Silverman's rule-of-thumb; "UCV": unbiased cross-validation. The default is "SRT".
B	An integer value specifying the number of bootstrap resamples for the construction of the confidence intervals. The default is 1000.
ci.level	An integer value (between 0 and 1) specifying the confidence level. The default is 0.95.
method	A character string specifying if bootstrap resampling (for the confidence intervals) should be done with or without regard to the disease status ("coutcome" or "noutcome"). In both cases, a naive bootstrap is used. By default, the resampling is done conditionally on the disease status.
pauc	A list of control values to replace the default values returned by the function pauccontrol. This argument is used to indicate whether the partial area under the pooled ROC curve should be computed, and in case it is computed, whether the focus should be placed on restricted false positive fractions (FPFs) or on restricted true positive fractions (TPFs), and the upper bound for the FPF (if focus is FPF) or the lower bound for the TPF (if focus is TPF).
parallel	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
ncpus	An integer with the number of processes to be used in parallel operation. Defaults to 1.
cl	An object inheriting from class cluster (from the parallel package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Estimates the pooled ROC curve (ROC) defined as

$$ROC(p) = 1 - F_D\{F_{\bar{D}}^{-1}(1-p)\},$$

where

$$F_D(y) = Pr(Y_D \leq y),$$
$$F_{\bar{D}}(y) = Pr(Y_{\bar{D}} \leq y).$$

The method implemented in this function estimates $F_D(\cdot)$ and $F_{\bar{D}}(\cdot)$ by means of kernel methods. More precisely, and letting $\{y_{\bar{D}i}\}_{i=1}^{n_{\bar{D}}}$ and $\{y_{Dj}\}_{j=1}^{n_D}$ be two independent random samples from the nondiseased and diseased populations, respectively, the distribution functions in each group take the form

$$F_D(y) = \frac{1}{n_D}\sum_{j=1}^{n_D}\Phi\left(\frac{y - y_{Dj}}{h_D}\right).$$

$$F_{\bar{D}}(y) = \frac{1}{n_{\bar{D}}}\sum_{i=1}^{n_D}\Phi\left(\frac{y - y_{\bar{D}i}}{h_{\bar{D}}}\right).$$

where $\Phi(y)$ stands for the standard normal distribution function evaluated at y. For the bandwidth h_d, $d \in \{D, \bar{D}\}$ which controls the amount of smoothing, two options are available. When `bw = "SRT"`,

$$h_d = 0.9\min\{SD(\mathbf{y}_d), IQR(\mathbf{y}_d)/1.34\}n_d^{-0.2},$$

where $SD(\mathbf{y}_d)$ and $IQR(\mathbf{y}_d)$ are the standard deviation and interquantile range, respectively, of $\mathbf{y}_d = (y_{d1}, \ldots, y_{dn_d})$. In turn, when `bw = "UCV"`, the bandwidth is selected via unbiased cross-validation, for further details we refer to `bw.ucv` from the base package `stats`.

The area under the curve is

$$AUC = \int_0^1 ROC(p)dp$$

and is computed numerically (using Simpson's rule). With regard to the partial area under the curve, when `focus = "FPF"` and assuming an upper bound u_1 for the FPF, what it is computed is

$$pAUC_{FPF}(u_1) = \int_0^{u_1} ROC(p)dp,$$

where again the integral is approximated numerically (Simpson's rule). The returned value is the normalised pAUC, $pAUC_{FPF}(u_1)/u_1$ so that it ranges from $u_1/2$ (useless test) to 1 (perfect marker). Conversely, when `focus = "TPF"`, and assuming a lower bound for the TPF of u_2, the partial area corresponding to TPFs lying in the interval $(u_2, 1)$ is computed as

$$pAUC_{TPF}(u_2) = \int_{u_2}^1 ROC_{TNF}(p)dp,$$

where $ROC_{TNF}(p)$ is a 270° rotation of the ROC curve, and it can be expressed as $ROC_{TNF}(p) = F_{\bar{D}}\{F_D^{-1}(1-p)\}$. Again, the computation of the integral is done via Simpson's rule. The returned value is the normalised pAUC, $pAUC_{TPF}(u_2)/(1-u_2)$, so that it ranges from $(1-u_2)/2$ (useless test) to 1 (perfect test).

Value

As a result, the function provides a list with the following components:

`call`	The matched call.
`marker`	A list with the diagnostic test outcomes in the healthy (h) and diseased (d) groups.
`missing.ind`	A logical value indicating whether missing values occur.
`bws`	Named list of length two with components 'h' (healthy) and 'd' (diseased). Each component is a numeric value with the selected bandwidth.

bw	The value of the argument bw used in the call.
p	Set of false positive fractions (FPF) at which the pooled ROC curve has been estimated.
ci.level	The value of the argument ci.level used in the call.
ROC	Estimated pooled ROC curve, and corresponding ci.level*100% pointwise confidence band (if computed).
AUC	Estimated pooled AUC, and corresponding ci.level*100% confidence interval (if computed).
pAUC	If computed, estimated partial area under the pooled ROC curve along with its ci.level*100% confidence interval (if B greater than zero). Note that the returned values are normalised, so that the maximum value is one (see more on Details).

References

Zou, K.H., Hall, W.J., Shapiro, D.E. (1997) Smooth non-parametric receiver operating characteristic (ROC) curves for continuous diagnostic tests. Statistics in Medicine, **16**, 2143–2156.

See Also

AROC.bnp, AROC.sp, AROC.kernel, pooledROC.BB, pooledROC.emp, pooledROC.kernel, pooledROC.dpm, cROC.bnp, cROC.sp or AROC.kernel.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

m0_kernel <- pooledROC.kernel(marker = "l_marker1", group = "status",
tag.h = 0, data = newpsa, p = seq(0,1,l=101), bw = "SRT",
B = 500, method = "coutcome",
pauc = pauccontrol(compute = TRUE, value = 0.5, focus = "FPF"))

summary(m0_kernel)

plot(m0_kernel)
```

predictive.checks *Posterior predictive checks.*

Description

Implements posterior predictive checks for objects of AROC or cROC as produced by AROC.bnp or cROC.bnp.

Usage

```
predictive.checks(object,
 statistics = c("min", "max", "kurtosis", "skewness"),
 ndensity = 512, devnew = TRUE)
```

Arguments

object	An object of class AROC or cROC as produced by AROC.bnp or cROC.bnp.
statistics	Character vector. Statistics to be used for the posterior predictive checking. By default, "min", "max", "kurtosis" and "skewness"
ndensity	An integer giving the number of equally spaced points at which the density of the test outcomes and of the simulated datasets from the posterior predictive distribution (see more on Details) is to be estimated (for more details see the help of the function density in the stats package).
devnew	A logical value. If TRUE, each plot is depicted in a new graphic device.

Details

Compares a selected test statistic computed based on the observed diagnostic test outcome (either in the nondiseased group, AROC object, or in both the nondiseased and diseased groups, cROC object) against the same test statistics computed based on simulated data from the posterior predictive distribution of the diagnostic test outcome obtained using a single-weights linear dependent Dirichlet process mixture of normals model. The following graphics are depicted: (1) histograms of the desired statistics computed from simulated datasets (nsave of them) from the posterior predictive distribution of the diagnostic test outcome. In these plots, the estimated statistics from the observed diagnostic test outcome are also depicted. (2) Kernel density estimates computed computed from simulated datasets (nsave of them) from the posterior predictive distribution of the diagnostic test outcome. In these plots, the kernel density estimate of the observed diagnostic test outcome is also depicted. In the case of an object of class AROC, the abovementioned graphics are depicted only for the diagnostic test outcome in the nondiseased group. However, for an object of class cROC, the graphics are depicted, separately, for both the nondiseased and diseased groups. For a detailed discussion about predictive checks, see Gabry et al. (2019).

Value

As a result, the function provides a list with the following components:

yrep — List of matrices associated with the diseased (d) and nondiseased (h) groups. Each column of the matrix (there are nsave of them) corresponds to a dataset generated from the posterior predictive distribution of the diagnostic test outcomes. For AROC objects, the list only contains results for the nondiseased group.

y — List of numeric vectors associated with the diseased (d) and nondiseased (h) groups. The vector contains the observed diagnostic test outcomes. For AROC objects, the list only contains results for the nondiseased group.

References

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., and Gelman, A. (2019). Visualization in Bayesian workflow. *Journal of the Royal Statistical Society, Series A*, **182**, 1–14.

Inacio de Carvalho, V., and Rodriguez-Alvarez, M. X. (2022). The Covariate-Adjusted ROC Curve: The Concept and Its Importance, Review of Inferential Methods, and a New Bayesian Estimator. *Statistical Science*, **37**, 541 -561.

See Also

AROC.bnp, cROC.bnp

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

AROC_bnp <- AROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
group = "status", tag.h = 0, data = newpsa, standardise = TRUE,
p = seq(0,1,l=101), compute.lpml = TRUE, compute.WAIC = TRUE)

predictive.checks(AROC_bnp, statistics = "skewness")
```

---

print.AROC *Print method for* AROC *objects*

Description

Default print method for objects fitted with AROC.bnp(), AROC.sp(), and AROC.kernel() functions.

Usage

```
## S3 method for class 'AROC'
print(x, ...)
```

Arguments

x	An object of class AROC as produced by AROC.bnp(), AROC.sp(), or AROC.kernel().
...	Further arguments passed to or from other methods. Not yet implemented.

Details

A short summary is printed including the area under the covariate-adjusted ROC curve (AAUC), and if required, the partial area under the covariate-adjusted ROC curve (pAAUC).

See Also

AROC.bnp, AROC.sp or AROC.kernel.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

AROC_bnp <- AROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
group = "status", tag.h = 0, data = newpsa, standardise = TRUE,
p = seq(0,1,l=101), compute.lpml = TRUE, compute.WAIC = TRUE)

AROC_bnp
```

---

print.cROC *Print method for* cROC *objects*

Description

Default print method for objects fitted with cROC.bnp(), cROC.sp() and cROC.kernel() functions.

Usage

```
## S3 method for class 'cROC'
print(x, ...)
```

Arguments

x	An object of class cROC as produced by cROC.bnp(), cROC.sp() or cROC.kernel().
...	Further arguments passed to or from other methods. Not yet implemented.

Details

A short summary is printed.

See Also

cROC.bnp, cROC.sp or cROC.kernel.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

cROC_bnp <- cROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
                formula.d = l_marker1 ~ f(age, K = 0),
                group = "status",
                tag.h = 0,
                data = newpsa,
                standardise = TRUE,
                p = seq(0, 1, len = 101),
                compute.lpml = TRUE,
                compute.WAIC = TRUE,
                compute.DIC = TRUE,
                pauc = pauccontrol(compute = TRUE, value = 0.5, focus = "FPF"),
                density = densitycontrol(compute = TRUE, grid.h = NA, grid.d = NA),
                mcmc = mcmccontrol(nsave = 500, nburn = 100, nskip = 1))

cROC_bnp
```

---

print.pooledROC *Print method for* pooledROC *objects*

Description

Default print method for objects fitted with pooledROC.BB, pooledROC.emp, pooledROC.kernel or pooledROC.dpm functions.

Usage

```
## S3 method for class 'pooledROC'
print(x, ...)
```

Arguments

x	An object of class pooledROC as produced by pooledROC.BB, pooledROC.emp, pooledROC.kernel or pooledROC.dpm functions.
...	Further arguments passed to or from other methods. Not yet implemented.

Details

A short summary is printed including the area under the pooled ROC curve (AUC).

See Also

pooledROC.BB, pooledROC.emp, pooledROC.kernel or pooledROC.dpm.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

m0_emp <- pooledROC.emp(marker = "l_marker1", group = "status",
tag.h = 0, data = newpsa, p = seq(0,1,l=101), B = 500)

summary(m0_emp)

plot(m0_emp)
```

---

priorcontrol.bnp *Prior information for the* AROC.bnp *and* cROC.bnp

---

Description

This function is used to set various parameters controlling the prior information to be used in the AROC.bnp and cROC.bnp functions.

Usage

```
priorcontrol.bnp(m0 = NA, S0 = NA, nu = NA, Psi = NA, a = 2, b = NA,
 alpha = 1, L = 10)
```

Arguments

m0	A numeric vector. Hyperparameter; mean vector of the (multivariate) normal prior distribution for the mean of the normal component of the centring distribution. NA signals autoinitialization, with defaults: a vector, of length Q, of zeros, if the data are standardised and the least squares estimates of the regression coefficients if the data are not standardised.
S0	A numeric matrix. Hyperparameter; covariance matrix of the (multivariate) normal prior distribution for the mean of the normal component of the centring distribution. NA signals autoinitialization, with defaults: $10I_{Q\times Q}$ if the data are standardised and $\hat{\mathbf{\Sigma}}$ if the data are not standardised, where $\hat{\mathbf{\Sigma}}$ is the estimated covariance matrix of the regression coefficients obtained by fitting a linear model to the data.
nu	A numeric value. Hyperparameter; degrees of freedom of the Wishart prior distribution for the precision matrix of the the normal component of the centring distribution.NA signals autoinitialization, with default: $Q + 2$ where Q is the number of columns of the design matrix.
Psi	A numeric matrix. Hyperparameter; scale matrix of the Wishart distribution for the precision matrix of the the normal component of the centring distribution. NA signals autoinitialization, with defaults: $I_{Q\times Q}$ if the data are standardised and to $30\hat{\mathbf{\Sigma}}$ if the data are not standardised, where $\hat{\mathbf{\Sigma}}$ is the estimated covariance matrix of the regression coefficients obtained by fitting a linear model to the data.
a	A numeric value. Hyperparameter; shape parameter of the gamma prior distribution for the precisions (inverse variances) of each component. The default is 2.
b	A numeric value. Hyperparameter; shape parameter of the gamma prior distribution for the precisions (inverse variances) of each component. NA signals autoinitialization, with defaults: 0.5 if the data are standardised and $\frac{\hat{\sigma}^2}{2}$ if the data are not standardised
alpha	A numeric value. Precision parameter of the Dirichlet Process. The default is 1.
L	A numeric value. Upper bound on the number of mixture components. Setting L = 1 corresponds to a normal model. The default is 10.

Value

A list with components for each of the possible arguments.

See Also

AROC.bnp and cROC.bnp

priorcontrol.dpm *Prior information for the* pooledROC.dpm

Description

This function is used to set various parameters controlling the prior information to be used in the pooledROC.dpm function.

Usage

```
priorcontrol.dpm(m0 = NA, S0 = NA, a = 2, b = NA, alpha = 1, L = 10)
```

Arguments

m0	A numeric value. Hyperparameter; mean of the normal prior distribution for the means of each component. NA signals autoinitialization, with defaults: 0 if the data are standardised and $\bar{y}_d$ ($d \in \{D, \bar{D}\}$ if the data are not standardised.
S0	A numeric value. Hyperparameter; variance of the normal prior distribution for the means of each component. NA signals autoinitialization, with defaults: 10 if the data are standardised and $100 * \frac{s_d^2}{n_d}$ ($d \in \{D, \bar{D}\}$ if the data are not standardised, where s_d denotes the sample standard deviation.
a	A numeric value. Hyperparameter; shape parameter of the gamma prior distribution for the precisions (inverse variances) of each component. The default is 2.
b	A numeric value. Hyperparameter; rate parameter of the gamma prior distribution for the precisions (inverse variances) of each component. NA signals autoinitialization, with defaults: 0.5 if the data are standardised and $\frac{s_d^2}{2}$ ($d \in \{D, \bar{D}\}$ if the data are not standardised.
alpha	A numeric value. Precision parameter of the Dirichlet Process. The default is 1.
L	A numeric value. Upper bound on the number of mixture components. Setting L=1 corresponds to a normal model. The default is 10.

Value

A list with components for each of the possible arguments.

See Also

pooledROC.dpm

psa *Prostate specific antigen (PSA) biomarker study.*

Description

The dataset contains 71 prostate cases and 71 controls who participated in a lung cancer prevention trial (CARET, Beta-carotene and retinol trial). For details, see Etzioni et al. (1999) and Pepe (2003).

Usage

```
data("psa")
```

Format

A data frame with 683 observations on the following 6 variables.

`id` Patient identifier.

`marker1` Total prostate specific antigen (PSA).

`marker2` Free prostate specific antigen (PSA)

`status` Presence/absence of prostate cancer. The non-cancer patients are controls matched to cases on age and number of serum samples available for analysis (see Details).

`age` Patient age at blood draw (serum sample).

`t` Time (years) relative to prostate cancer diagnosis.

Details

The CARET enrolled 12000 men, aged between 50 and 65 years, at high risk of lung cancer. For each subject on the study, serum samples were drawn at baseline and at two-year intervals after that. The data presented here represent a subsample of the original sample, and it was reported by Etzioni et al. (1999). It contains 71 cases of prostate cancer that occurred during the study. All these cases had, at least, three and up to eight serum samples. As far as controls are concerned, they were selected from the participants of the CARET study verifying that they had not been diagnosed with prostate cancer by the time of the original study, and the selection was done by matching to cases on date of birth and number of serum samples available for analysis.

Source

The dataset can be downloaded from https://research.fredhutch.org/diagnostic-biomarkers-center/en/datasets.html.

References

Pepe, M. S. (2003). The Statistical Evaluation of Medical Tests for Classification and Prediction. Oxford Statistical Science Series. Oxford University Press, New York.

Etzioni, R., Pepe, M. S., Longton, G., Hu. C., and Goodman, G. (1999). Incorporating the time dimension in receiver operating characteristic curves: A case study of prostate cancer. Medical Decision Making, 19(3), 242-251.

Examples

```
data(psa)
summary(psa)
```

---

summary.AROC *Summary method for* AROC *objects*

Description

Default summary method for objects fitted with AROC.bnp(), AROC.sp(), or AROC.kernel() functions.

Usage

```
## S3 method for class 'AROC'
summary(object, ...)
```

Arguments

object	An object of class AROC as produced by AROC.bnp(), AROC.sp(), or AROC.kernel().
...	Further arguments passed to or from other methods. Not yet implemented.

Details

The information printed depends on the method. In all cases, the call to the function, the method, the area under the covariate-adjusted ROC curve (AAUC), the partial area under the covariate-adjusted ROC curve (if required) (AAUC), and the sample sizes are printed. For the semiparametric approach (AROC.sp()), the estimated coefficients (and 95% confidence intervals, if required) of the model for the healthy population are printed. In addition, the function provides the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). For the nonparametric Bayesian approach (AROC.bnp()), and if required, the function provides the log pseudo marginal likelihood (LPML), the widely applicable information criterion (WAIC) and/or the deviance information criterion (DIC). For the kernel-based approach (AROC.kernel()), information regarding the selected bandwidth and the type of kernel estimator (for both regression and variance functions) is printed.

See Also

AROC.bnp, AROC.sp or AROC.kernel.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)
```

```
m0 <- AROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
group = "status", tag.h = 0, data = newpsa, standardise = TRUE,
p = seq(0,1,l=101), compute.lpml = TRUE, compute.WAIC = TRUE)

summary(m0)
```

---

summary.cROC *Summary method for* cROC *objects*

Description

Default summary method for objects fitted with cROC.bnp(), cROC.sp(), or cROC.kernel() functions.

Usage

```
## S3 method for class 'cROC'
summary(object, ...)
```

Arguments

object	An object of class cROC as produced by cROC.bnp(), cROC.sp(), or cROC.kernel().
...	Further arguments passed to or from other methods. Not yet implemented.

Details

The information printed depends on the method. In all cases, the call to the function, the method, and the sample sizes are printed. For the semiparametric approach (cROC.sp()), the estimated coefficients (and 95% confidence intervals, if required) of the model for the healthy population, the diseased population and the conditional ROC curve, are printed. In addition, the function provides the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). For the nonparametric Bayesian approach (cROC.bnp()), and if required, the function provides the log pseudo marginal likelihood (LPML), the widely applicable information criterion (WAIC) and/or the deviance information criterion (DIC) (for both healthy and diseased populations). For the kernel-based approach (cROC.kernel()), information regarding the selected bandwidths and the type of kernel estimator(for both healthy and diseased populations and for both regression and variance functions) is printed.

References

Rodriguez-Alvarez, M.X., Tahoces, P.G., Cadarso-Suarez, C. and Lado, M.J. (2011). Comparative study of ROC regression techniques. Applications for the computer-aided diagnostic system in breast cancer detection. *Computational Statistics and Data Analysis*, **55**, 888–902.

See Also

cROC.bnp, cROC.sp or cROC.kernel.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

cROC_bnp <- cROC.bnp(formula.h = l_marker1 ~ f(age, K = 0),
                formula.d = l_marker1 ~ f(age, K = 0),
                group = "status",
                tag.h = 0,
                data = newpsa,
                standardise = TRUE,
                p = seq(0, 1, len = 101),
                compute.lpml = TRUE,
                compute.WAIC = TRUE,
                compute.DIC = TRUE,
                pauc = pauccontrol(compute = TRUE, value = 0.5, focus = "FPF"),
                density = densitycontrol(compute = TRUE, grid.h = NA, grid.d = NA),
                mcmc = mcmccontrol(nsave = 500, nburn = 100, nskip = 1))

summary(cROC_bnp)
```

---

summary.pooledROC *Summary method for* pooledROC *objects*

Description

Default summary method for objects fitted with pooledROC.BB, pooledROC.emp, pooledROC.kernel, or pooledROC.dpm functions.

Usage

```
## S3 method for class 'pooledROC'
summary(object, ...)
```

Arguments

object	An object of class pooledROC as produced by pooledROC.BB, pooledROC.emp, pooledROC.kernel, or pooledROC.dpm.
...	Further arguments passed to or from other methods. Not yet implemented.

Details

A short summary is printed including the call to the function, the method, samples sizes, the area under the pooled ROC curve (AUC), and if required, the partial area under the pooled ROC curve. For the nonparametric Bayesian approach (`pooledROC.dpm()`), and if required, the function provides the log pseudo marginal likelihood (LPML), the widely applicable information criterion (WAIC) and/or the deviance information criterion (DIC). For the kernel-based approach (`pooledROC.dpm()`), information regarding the selected bandwidths and the density bandwidth selection method is presented.

See Also

`pooledROC.BB`, `pooledROC.emp`, `pooledROC.kernel` or `pooledROC.dpm`.

Examples

```
library(ROCnReg)
data(psa)
# Select the last measurement
newpsa <- psa[!duplicated(psa$id, fromLast = TRUE),]

# Log-transform the biomarker
newpsa$l_marker1 <- log(newpsa$marker1)

m0_emp <- pooledROC.emp(marker = "l_marker1", group = "status",
            tag.h = 0, data = newpsa, p = seq(0,1,l=101), B = 500)

summary(m0_emp)

plot(m0_emp)
```

Index

∗ **datasets**
endosyn, 42
psa, 70
∗ **package**
ROCnReg-package, 3

AROC.bnp, 4, *9*, *12*, *16*, *18*, *30*, *34*, *38, 39*, *42*, *44, 45*, *47*, *51*, *56*, *59*, *62*, *64, 65*, *68*, *71*
AROC.kernel, 9, *9*, *12*, *16*, *18*, *30*, *34*, *38*, *45*, *47*, *51*, *56*, *59*, *62*, *65*, *71*
AROC.sp, *9*, *12*, 13, *16*, *18*, *30*, *34*, *38*, *45*, *47*, *51*, *56*, *59*, *62*, *65*, *71*

compute.threshold.AROC, 16, *39*
compute.threshold.cROC, 19, *39*
compute.threshold.pooledROC, 22
cROC.bnp, *9*, *12*, *16*, *21*, 25, *30*, *38–40*, *44, 45*, *47*, *51*, *56*, *59*, *62*, *64*, *66*, *68*, *73*
cROC.kernel, *21*, 31, *34*, *39*, *45*, *47*, *66*, *73*
cROC.sp, *9*, *12*, *16*, *21*, *30*, *34*, 35, *38, 39*, *45*, *47*, *51*, *56*, *59*, *62*, *66*, *73*
cROCData, *17*, *20*, *26*, *32*, *36*, 39

densitycontrol, *26*, 40, *53*
densitycontrol.aroc, *5*, 41

endosyn, 42

formula, *4*, *13*, *25, 26*, *35*

glm, *8*, *30*

lm, *16*, *38*

mcmccontrol, *5*, *26*, 44, *53*

np, *10–12*, *32–34*
npreg, *12*, *34*
npregbw, *12*, *34*

pauccontrol, *5*, *10*, *14*, *26*, *32*, *36*, 45, *50*, *53*, *57*, *60*
plot.AROC, 46
plot.cROC, 47
plot.pooledROC, 48
pooledROC.BB, *9*, *12*, *16*, *23, 24*, *30*, *34*, *38*, *48*, 49, *49*, *51*, *56*, *59*, *62*, *67*, *73, 74*
pooledROC.dpm, *9*, *12*, *16*, *23, 24*, *30*, *34*, *38*, *40*, *44*, *48, 49*, *51*, 52, *56*, *59*, *62*, *67*, *69*, *73, 74*
pooledROC.emp, *9*, *12*, *16*, *23, 24*, *30*, *34*, *38*, *48, 49*, *51*, 56, *56*, *59*, *62*, *67*, *73, 74*
pooledROC.kernel, *9*, *12*, *16*, *23, 24*, *30*, *34*, *38*, *48, 49*, *51*, *56*, 59, *59*, *62*, *67*, *73*, *74*
predictive.checks, 63
print.AROC, 64
print.cROC, 65
print.pooledROC, 66
priorcontrol.bnp, *5*, *26*, 67
priorcontrol.dpm, *53*, 69
psa, 70

ROCnReg *(*ROCnReg-package*)*, 3
ROCnReg-package, 3

summary.AROC, 71
summary.cROC, 72
summary.pooledROC, 73