

Package: xplus (via r-universe)

September 27, 2026

Title Positive and Unlabeled Learning from Unbalanced Cases and Sparse Structures

Version 1.0.2

Description Provides PLUS-derived extensions for positive and unlabeled (PU) learning from unbalanced cases and sparse structures, based on Zhou et al. (2022) [doi:10.1371/journal.pcbi.1009956](https://doi.org/10.1371/journal.pcbi.1009956). Iteratively relabels unlabeled observations via penalised logistic regression and pseudo-label updates, then refits a final sparse model. Includes weighted bootstrap sampling, convergence diagnostics, prediction, coefficient extraction, and assessment utilities.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.1.0)

LazyData true

Imports glmnet (>= 4.1-8), Matrix, methods, stats, tibble, utils

Suggests covr, knitr, rmarkdown, survival, testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/alrobles/xplus>,
<https://alrobles.github.io/xplus/>

BugReports <https://github.com/alrobles/xplus/issues>

Config/roxygen2/version 8.0.0

Repository <https://alrobles.r-universe.dev>

Date/Publication 2026-09-27 05:09:01 UTC

RemoteUrl <https://github.com/alrobles/xplus>

RemoteRef HEAD

RemoteSha e9f545aa89552784f98694cb0fbfa6f4f65017ec

Contents

assess	2
auc	3
auc_matrix	4
binexample	5
coef.xplus	5
cutoff_example	6
fit_xplus_example	6
get_auc	6
get_predictions	7
lacs	8
lacsSample	8
predict.xplus	9
predicted_coefficients_example	10
predicted_y_example	10
print.summary.xplus	11
print.xplus	11
summary.xplus	12
xplus	13
xplus_object_example	15
Index	16

assess	<i>Assess predictive performance</i>
--------	--------------------------------------

Description

Assess predictive performance

Usage

```
assess(object, newx = NULL, newy, weights = NULL, ...)

## S3 method for class 'xplus'
assess(object, newx = NULL, newy, weights = NULL, ...)
```

Arguments

object	A model object.
newx	Optional feature matrix.
newy	Binary 0/1 labels or a two-column finite nonnegative matrix of negative and positive class masses; soft vectors are not accepted.
weights	Optional finite nonnegative numeric row weights without recycling; NULL means unit weights.
...	Additional arguments passed to <code>predict()</code> .

Value

A named list with deviance, class, auc, mse, and mae. For class metric, the threshold used is the model's cutoff (from `object$cutoff`), consistent with `predict(type = "class")`; MSE and MAE sum both class-column losses (twice the scalar loss). Undefined metrics return NA with a warning.

References

Zhou et al. (2022). doi:10.1371/journal.pcbi.1009956

See Also

`xplus()`, `get_auc()`

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 5), ncol = 5)
y <- c(rep(1, 20), rep(0, 80))
fit <- xplus(x, y, max_iter = 5)
assess(fit, newx = x, newy = y)
```

auc

Compute area under the ROC curve

Description

Compute area under the ROC curve

Usage

```
auc(y, prob, w = NULL)
```

Arguments

y	Binary 0/1 vector, including logical, character, or factor encodings.
prob	Finite numeric scores, one per label; values outside $[0, 1]$ are allowed.
w	Optional finite nonnegative numeric sample weights; NULL means unit weights.

Value

Numeric rank AUC with half credit for ties; NA with a warning if either effective class mass is zero.

References

Zhou et al. (2022). doi:10.1371/journal.pcbi.1009956

See Also

`auc_matrix()`, `get_auc()`

Examples

```
y <- c(0, 0, 1, 1)
p <- c(0.1, 0.3, 0.7, 0.9)
auc(y, p)
```

auc_matrix

Compute AUC from matrix labels

Description

Compute AUC from matrix labels

Usage

```
auc_matrix(y, prob, weights = NULL)
```

Arguments

y	Two-column finite nonnegative class masses (negative, positive); soft labels, counts, and zero-mass rows are supported.
prob	Finite numeric scores, one per row; values outside $[0, 1]$ are allowed.
weights	Optional finite nonnegative numeric row weights without recycling; NULL means unit weights.

Value

Weighted rank AUC with half credit for ties; NA with a warning if either effective class mass is zero.

References

Zhou et al. (2022). doi:10.1371/journal.pcbi.1009956

See Also

`auc()`, `assess()`

Examples

```
y <- cbind(c(1, 1, 0, 0), c(0, 0, 1, 1))
p <- c(0.2, 0.3, 0.7, 0.8)
auc_matrix(y, p)
```

binexample	<i>Example binary labels dataset</i>
------------	--------------------------------------

Description

A small binary example dataset used in package examples.

Usage

```
binexample
```

Format

A data frame.

coef.xplus	<i>Extract coefficients from an xplus model</i>
------------	---

Description

Extract coefficients from an xplus model

Usage

```
## S3 method for class 'xplus'  
coef(object, s = "lambda.min", ...)
```

Arguments

object	An xplus object.
s	Penalty value name or numeric lambda.
...	Additional arguments.

Value

A sparse coefficient matrix.

See Also

[summary.xplus\(\)](#), [print.xplus\(\)](#)

Examples

```
set.seed(1)  
x <- matrix(rnorm(100 * 5), ncol = 5)  
y <- c(rep(1, 20), rep(0, 80))  
fit <- xplus(x, y, max_iter = 5)  
coef(fit)
```

cutoff_example	<i>Example cutoff values</i>
----------------	------------------------------

Description

Example cutoff values used in demonstrations.

Usage

```
cutoff_example
```

Format

A numeric vector.

fit_xplus_example	<i>Example fitted xplus object</i>
-------------------	------------------------------------

Description

Example model fit used in package documentation.

Usage

```
fit_xplus_example
```

Format

An object of class xplus.

get_auc	<i>Compute AUC for predictions from a model</i>
---------	---

Description

Compute AUC for predictions from a model

Usage

```
get_auc(object, newx = NULL, newy = NULL, weights = NULL, ...)
```

```
## S3 method for class 'xplus'
```

```
get_auc(object, newx = NULL, newy = NULL, weights = NULL, ...)
```

Arguments

object	A model object.
newx	Feature matrix.
newy	True labels.
weights	Optional sample weights.
...	Arguments passed to <code>predict.xplus()</code> , such as <code>s</code> selecting a single lambda.

Value

Numeric AUC value.

See Also

`assess()`, `auc()`

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 5), ncol = 5)
y <- c(rep(1, 20), rep(0, 80))
fit <- xplus(x, y, max_iter = 5)
get_auc(fit, newx = x, newy = y)
```

get_predictions	<i>Build a tidy prediction table</i>
-----------------	--------------------------------------

Description

Build a tidy prediction table

Usage

```
get_predictions(object, newx, newy, use_cutoff = TRUE)
```

Arguments

object	An xplus model object.
newx	Feature matrix.
newy	True labels.
use_cutoff	Logical; if TRUE, classify with the model's cutoff. If FALSE, classify probabilities strictly greater than 0.5 as positive.

Value

A tibble with truth labels, probabilities and predicted classes; `Class1` is the positive label 1, classified by probability strictly greater than the cutoff.

References

Zhou et al. (2022). doi:10.1371/journal.pcbi.1009956

See Also

`predict.xplus()`, `assess()`

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 5), ncol = 5)
y <- c(rep(1, 20), rep(0, 80))
fit <- xplus(x, y, max_iter = 5)
get_predictions(fit, x, y)
```

lacs	<i>LACS dataset</i>
------	---------------------

Description

Example dataset from the original PLUS package.

Usage

lacs

Format

A data frame.

lacsSample	<i>LACS sample dataset</i>
------------	----------------------------

Description

Sample subset of the LACS data.

Usage

lacsSample

Format

A data frame.

predict.xplus	<i>Predict from an xplus model</i>
---------------	------------------------------------

Description

Predict from an xplus model

Usage

```
## S3 method for class 'xplus'  
predict(object, newx = NULL, s = "lambda.min", type = "response", ...)
```

Arguments

object	An xplus object.
newx	Optional finite numeric feature matrix; defaults to training data. Named training features must match and are reordered automatically.
s	Exact penalty name ("lambda.min", "lambda.1se") or finite nonnegative numeric lambda vector. Cache-only objects support only "lambda.min".
type	Prediction type: "response", "link", or "class".
...	Additional arguments are not supported and cause an error.

Value

Probabilities (type = "response") or log-odds (type = "link"); cached probabilities of 0 and 1 give infinite log-odds. Classes use a factor with fixed levels "0", "1" for one lambda, or a dimension-preserving 0/1 matrix for multiple lambdas.

References

Zhou et al. (2022). doi:10.1371/journal.pcbi.1009956

See Also

`xplus()`, `print.xplus()`

Examples

```
set.seed(1)  
x <- matrix(rnorm(100 * 5), ncol = 5)  
y <- c(rep(1, 20), rep(0, 80))  
fit <- xplus(x, y, max_iter = 5)  
predict(fit, newx = x, type = "response")  
predict(fit, newx = x, type = "link")
```

predicted_coefficients_example
<i>Example predicted coefficients</i>

Description

Sparse coefficients extracted from an xplus model.

Usage

predicted_coefficients_example

Format

A sparse matrix.

predicted_y_example	<i>Example predicted probabilities</i>
---------------------	--

Description

Predicted probabilities from an xplus model.

Usage

predicted_y_example

Format

A numeric matrix.

References

Zhou et al. (2022). doi:10.1371/journal.pcbi.1009956

print.summary.xplus	<i>Print method for summary.xplus objects</i>
---------------------	---

Description

Print method for summary.xplus objects

Usage

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

Arguments

x	A summary.xplus object.
...	Additional arguments.

Value

The input object x is returned invisibly (called for side effects).

print.xplus	<i>Print an xplus model</i>
-------------	-----------------------------

Description

Print an xplus model

Usage

```
## S3 method for class 'xplus'  
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	An xplus object.
digits	Number of significant digits.
...	Additional arguments.

Value

Invisibly returns x.

References

Zhou et al. (2022). doi:10.1371/journal.pcbi.1009956

See Also

[summary.xplus\(\)](#), [coef.xplus\(\)](#)

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 5), ncol = 5)
y <- c(rep(1, 20), rep(0, 80))
fit <- xplus(x, y, max_iter = 5)
print(fit)
```

summary.xplus

Summarize an xplus model

Description

Summarize an xplus model

Usage

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

Arguments

object	An xplus object.
...	Additional arguments.

Value

A list of class summary.xplus with model details.

References

Zhou et al. (2022). doi:10.1371/journal.pcbi.1009956

See Also

[print.xplus\(\)](#), [coef.xplus\(\)](#)

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 5), ncol = 5)
y <- c(rep(1, 20), rep(0, 80))
fit <- xplus(x, y, max_iter = 5)
summary(fit)
```

xplus

*Fit an xplus model***Description**

Fit a PLUS-derived model for positive-unlabeled learning.

Usage

```
xplus(
  x,
  y,
  alpha = 1,
  sample_use_time = 30,
  learning_rate = 1,
  qq = 0.1,
  verbose = FALSE,
  nfolds = 4,
  max_iter = 10000,
  convergence_threshold = 0.9,
  seed = NULL,
  sigmoid_scale = 10,
  min_iter = 5,
  stability_window = 5,
  min_coverage = 0.9,
  sampling = c("bootstrap", "unique"),
  cv_measure = c("deviance", "auc"),
  degenerate_threshold = 1e-06
)
```

Arguments

<code>x</code>	Finite numeric feature matrix with at least two columns.
<code>y</code>	Binary vector where 1 indicates known positives and 0 indicates unlabeled samples; factors are interpreted by their labels.
<code>alpha</code>	Elastic-net mixing parameter in $[0, 1]$, not the sigmoid scale in the PLUS paper.
<code>sample_use_time</code>	Unlabeled-sampling budget inherited from the reference implementation: maximum number of completed sampling rounds containing each unlabeled case, not the number of bootstrap copies within a round.
<code>learning_rate</code>	Pseudo-label smoothing rate in $(0, 1]$; values below one retain the package's global-update, hard-thresholded enhancement path.
<code>qq</code>	Quantile used to define the positive-reference cutoff.
<code>verbose</code>	Logical; print iterative progress messages.

<code>nfolds</code>	Requested CV folds, an integer at least three; reduced for small classes.
<code>max_iter</code>	Maximum number of pseudo-labeling iterations.
<code>convergence_threshold</code>	Required stability score in $(0, 1]$, evaluated before learning-rate damping.
<code>seed</code>	Integer or NULL. Random seed for reproducibility, applied via <code>set.seed()</code> . Default NULL (no seed).
<code>sigmoid_scale, degenerate_threshold</code>	Positive sigmoid scale and nonnegative residual-clamping tolerance.
<code>min_iter, stability_window</code>	Minimum iterations and consecutive stable iterations required before declaring convergence. <code>xplus</code> retains two iterative paths, distinct from the paper's pseudocode: <code>"current"</code> (<code>learning_rate = 1</code>) uses sampled Bernoulli labels, while <code>"continuous_enhancement"</code> (<code>learning_rate < 1</code>) smooths probabilities and updates all unlabeled hard labels. Both use soft final fitting targets. Stability compares the undamped mapped scores with the current pseudo-labels across all unlabeled cases and requires a full window and sampling coverage.
<code>min_coverage</code>	Minimum fraction of unlabeled cases sampled before convergence, in $[0, 1]$. Bootstrap multiplicities are represented as case weights, keeping duplicate copies together in CV. <code>sampling = "unique"</code> retains legacy deduplication. Final-fit fallback is based on effective class mass, not thresholded labels, and is recorded with the actual fitting targets and the iteration history.
<code>sampling, cv_measure</code>	Sampling convention (<code>"bootstrap"</code> or <code>"unique"</code>) and CV criterion (<code>"deviance"</code> or <code>"auc"</code>); deviance is the default for both fitting stages.

Details

Core PLUS behavior alternates between fitting penalized logistic models on known positives plus sampled unlabeled cases, anchoring predictions to a positive quantile cutoff, and iteratively relabeling unlabeled samples.

Value

An object of class `"xplus"` containing predictions, original and final labels, pseudo-labels, fallback metadata, sampling counts, and history.

References

Zhou et al. (2022). doi:10.1371/journal.pcbi.1009956

See Also

[predict.xplus\(\)](#), [summary.xplus\(\)](#), [assess.xplus\(\)](#)

Examples

```
set.seed(1)
x <- matrix(rnorm(200 * 10), ncol = 10)
y <- c(rep(1, 40), rep(0, 160))
fit <- xplus(x, y, max_iter = 20)
```

xplus_object_example *Example xplus object*

Description

Saved xplus object for examples and tests.

Usage

```
xplus_object_example
```

Format

An object of class xplus.

Index

* datasets

- binexample, [5](#)
- cutoff_example, [6](#)
- fit_xplus_example, [6](#)
- lacs, [8](#)
- lacsSample, [8](#)
- predicted_coefficients_example, [10](#)
- predicted_y_example, [10](#)
- xplus_object_example, [15](#)

[summary.xplus](#), [12](#)
[summary.xplus\(\)](#), [5](#), [12](#), [14](#)

[xplus](#), [13](#)
[xplus\(\)](#), [3](#), [9](#)
[xplus_object_example](#), [15](#)

[assess](#), [2](#)
[assess\(\)](#), [4](#), [7](#), [8](#)
[assess.xplus\(\)](#), [14](#)
[auc](#), [3](#)
[auc\(\)](#), [4](#), [7](#)
[auc_matrix](#), [4](#)
[auc_matrix\(\)](#), [4](#)

[binexample](#), [5](#)

[coef.xplus](#), [5](#)
[coef.xplus\(\)](#), [12](#)
[cutoff_example](#), [6](#)

[fit_xplus_example](#), [6](#)

[get_auc](#), [6](#)
[get_auc\(\)](#), [3](#), [4](#)
[get_predictions](#), [7](#)

[lacs](#), [8](#)
[lacsSample](#), [8](#)

[predict\(\)](#), [2](#)
[predict.xplus](#), [9](#)
[predict.xplus\(\)](#), [7](#), [8](#), [14](#)
[predicted_coefficients_example](#), [10](#)
[predicted_y_example](#), [10](#)
[print.summary.xplus](#), [11](#)
[print.xplus](#), [11](#)
[print.xplus\(\)](#), [5](#), [9](#), [12](#)