Package 'torchMAUM'

September 30, 2025

Type Package

Title Multi-Class Area Under the Minimum in Torch

Version 2025.7.30

Encoding UTF-8

Description Torch code for computing multi-class Area Under The Minimum, <https://www.jmlr.org/papers/v24/21-0751.html>, Generalization. Useful for optimizing Area under the curve.

License LGPL-3

URL https://github.com/OGuenoun/torchMAUM

BugReports https://github.com/OGuenoun/torchMAUM/issues

Imports torch, ggplot2, data.table

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Omar Guenoun [aut, cre]

Maintainer Omar Guenoun <omargue31@gmail.com>

Repository CRAN

Date/Publication 2025-09-30 16:00:02 UTC

Contents

Draw_ROC_curve_macro *Draws multi-class ROC curve macro*

Description

This function draws K ROC curves using OvR approach, each time considering one class as the positive class. It assumes that all the inputs are torch tensors and labels are in [1,K] with K being the number of classes.

Usage

```
Draw_ROC_curve_macro(pred_tensor, label_tensor)
```

Arguments

pred_tensor	output of the model assuming it is of dimension NxK (or Nx1 for binary classification)
label_tensor	true labels , tensor of length N

Value

K ROC curves

Examples

```
# Small example with 3 classes and 10 samples
set.seed(1)
labels = torch::torch_randint(1, 4, size = 10, dtype = torch::torch_long())
Draw_ROC_curve_micro(torch::torch_randn(c(10, 3)), labels)
```

Draw_ROC_curve_micro *Draws multi-class ROC curve micro*

Description

This function draws one ROC curve using OvR approach and micro average. It assumes that all the inputs are torch tensors and labels are in [1,K] with K being the number of classes.

Usage

```
Draw_ROC_curve_micro(pred_tensor, label_tensor)
```

Arguments

pred_tensor	output of the model assuming it is of dimension NxK (or Nx1 for binary classification)
label_tensor	true labels , tensor of length N

Value

plot of the ROC curve

Examples

```
# Small example with 3 classes and 10 samples
set.seed(1)
labels = torch::torch_randint(1, 4, size = 10, dtype = torch::torch_long())
Draw_ROC_curve_micro(torch::torch_randn(c(10, 3)), labels)
```

ROC_AUC_macro *Compute multi-class ROC AUC macro averaged*

Description

This function computes the multi class ROC AUC using OvR approach and macro averaging. It assumes that all the inputs are torch tensors and labels are in [1,K] with K being the number of classes.

Usage

```
ROC_AUC_macro(pred_tensor, label_tensor)
```

Arguments

pred_tensor	output of the model assuming it is of dimension NxK (or Nx1 for binary classification)
label_tensor	true labels , tensor of length N

Value

ROC AUC macro averaged

Examples

```
# Small example with 3 classes and 10 samples
set.seed(1)
labels = torch::torch_randint(1, 4, size = 10, dtype = torch::torch_long())
Draw_ROC_curve_micro(torch::torch_randn(c(10, 3)), labels)
```

`ROC_AUC_micro` *Compute multi-class ROC AUC micro averaged*

Description

This function computes the multi class ROC AUC using OvR approach and micro averaging. It assumes that all the inputs are torch tensors and labels are in [1,K] with K being the number of classes.

Usage

```
ROC_AUC_micro(pred_tensor, label_tensor)
```

Arguments

`pred_tensor`	output of the model assuming it is of dimension NxK (or Nx1 for binary classification)
`label_tensor`	true labels , tensor of length N

Value

ROC AUC macro averaged

Examples

```
# Small example with 3 classes and 10 samples
set.seed(1)
labels = torch::torch_randint(1, 4, size = 10, dtype = torch::torch_long())
Draw_ROC_curve_micro(torch::torch_randn(c(10, 3)), labels)
```

ROC_AUM_macro *Compute multi-class ROC AUM macro averaged*

Description

This function computes the multi class ROC AUM using OvR approach and macro averaging. It assumes that all the inputs are torch tensors and labels are in [1,K] with K being the number of classes.

Usage

```
ROC_AUM_macro(pred_tensor, label_tensor)
```

Arguments

pred_tensor	output of the model assuming it is of dimension NxK (or Nx1 for binary classification)
label_tensor	true labels , tensor of length N

Value

ROC AUM macro averaged

Examples

```
# Small example with 3 classes and 10 samples
set.seed(1)
labels = torch::torch_randint(1, 4, size = 10, dtype = torch::torch_long())
Draw_ROC_curve_micro(torch::torch_randn(c(10, 3)), labels)
```

ROC_AUM_micro *Compute multi-class ROC AUM micro averaged*

Description

This function computes the multi class ROC AUM using OvR approach and micro averaging. It assumes that all the inputs are torch tensors and labels are in [1,K] with K being the number of classes.

Usage

```
ROC_AUM_micro(pred_tensor, label_tensor, counts = NULL)
```

Arguments

pred_tensor	output of the model assuming it is of dimension NxK (or Nx1 for binary classification)
label_tensor	true labels , tensor of length N
counts	(optional) the counts of each class , tensor of length K, used to compute weighted ROC AUM micro.

Value

ROC AUM micro averaged

Examples

```
# Small example with 3 classes and 10 samples
set.seed(1)
labels = torch::torch_randint(1, 4, size = 10, dtype = torch::torch_long())
Draw_ROC_curve_micro(torch::torch_randn(c(10, 3)), labels)
```

Index