Multilabel Classification in mlr

Quay Au

July 4th, 2018

LMU Munich
Working Group Computational Statistics

Table of contents

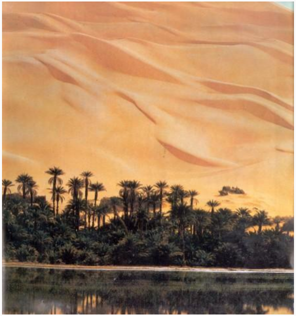
What is Multilabel Classification?

What is Multilabel Classification

- What labels are relevant in this picture?

Tree	Mountain	Water	Sunset	Desert
YES	YES	YES	YES	NO

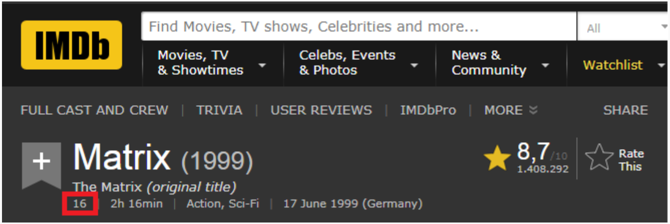
What is Multilabel Classification



- What labels are relevant in this picture?

Tree	Mountain	Water	Sunset	Desert
YES	NO	YES	NO	YES

Multilabel vs. Multiclass



- Age rating
  - Possible ratings: $\{0, 12, 16, 18\}$
  - Each movie can only be assigned **one** rating
  - **Multiclass** classification problem

Multilabel vs. Multiclass

- Genre classification
  - Possible genres: {Comedy, Sci-Fi, Horror, Romance, Action, ...}
  - Each movie can be categorized into **more than** one genre
  - **Multilabel** classification problem

Modeling Multilabel Problems

Modeling Multilabel Problems

- Algorithm adaptation methods
    - Directly handle multilabel data
    - E.g. **randomForestSRC**
- Problem transformation methods
    - Transform the multilabel problem into binary problems
        - Using label information as features
    - Many available binary classifiers

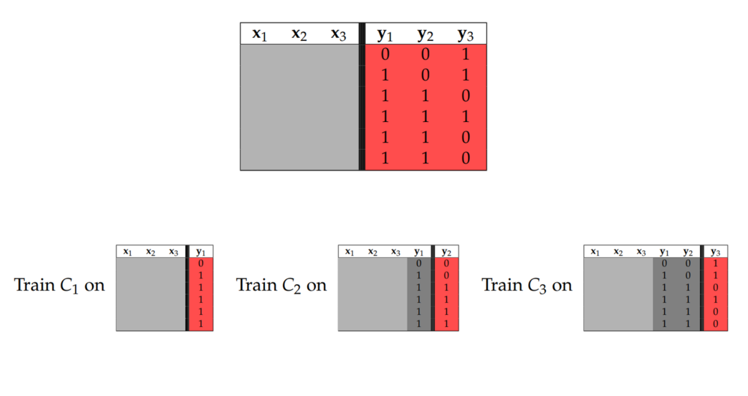
Problem Transformation Methods in mlr

Available problem transformation methods in mlr:

	True labels	Pred. labels
Partial cond.	Classifier chains	Nested stacking
Full cond.	Dependent binary relevance	Stacking

Benchmark paper: Multilabel Classification with R Package mlr

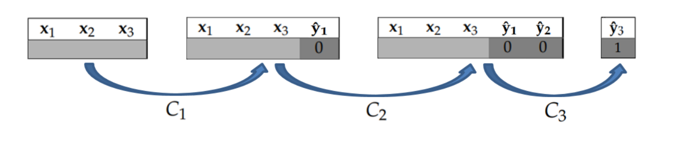
Example: Chaining



Example for order: $y_1 \rightarrow y_2 \rightarrow y_3$

Example: Chaining

- How to predict a new observation?
  - True label information is not available for a new observation
  - Label information is obtained by using classifiers along the chain



How to Measure Performance?

How to Measure Performance?

Performance can be measured on a *per instance*-basis:

- $\text{subset}_{0/1}(\mathbf{y}, \hat{\mathbf{y}}) = \mathbb{1}_{(\mathbf{y} \neq \hat{\mathbf{y}})}$
- $\text{HammingLoss}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{m} \sum_{k=1}^{m} \mathbb{1}_{(\mathbf{y}_k \neq \hat{\mathbf{y}}_k)}$
- Also F_1, precision and recall can be defined on a per instance basis

Also possible: label-based performance measures

Multilabel Classification in mlr

Example: yeast dataset (available with mlr)

- Gene expression data
- Each of $n = 2417$ genes is represented with 103 features
- $m = 14$ different labels can be assigned to a gene

Data Format

```
library(mlr)
yeast = getTaskData(yeast.task, target.extra = TRUE)
yeast$data[1:5, 1:5]
```

```
##          x1        x2        x3        x4        x5
## 1  0.093700  0.139771  0.062774  0.007698  0.083873
## 2 -0.022711 -0.050504 -0.035691 -0.065434 -0.084316
## 3 -0.090407  0.021198  0.208712  0.102752  0.119315
## 4 -0.085235  0.009540 -0.013228  0.094063 -0.013592
## 5 -0.088765 -0.026743  0.002075 -0.043819 -0.005465
```

Data Format

Targets must be logical vectors, indicating presence/absence of labels

```
yeast$target[1:5, 1:5]
```

```
##   label1 label2 label3 label4 label5
## 1  FALSE  FALSE   TRUE   TRUE  FALSE
## 2  FALSE  FALSE  FALSE  FALSE  FALSE
## 3  FALSE   TRUE   TRUE  FALSE  FALSE
## 4  FALSE  FALSE   TRUE   TRUE  FALSE
## 5   TRUE   TRUE  FALSE  FALSE  FALSE
```

Multilabel Task

```r
yeast.data = cbind(yeast$data, yeast$target)
y.task = makeMultilabelTask(data = yeast.data, target = names(yeast$target))
y.task
```

```
## Supervised task: yeast.data
## Type: multilabel
## Observations: 2417
## Features:
##    numerics     factors     ordered functionals
##         103           0           0           0
## Missings: FALSE
## Has weights: FALSE
## Has blocking: FALSE
## Has coordinates: FALSE
## Classes: 14
##  label1  label2  label3  label4  label5  label6  label7
##     762    1038     983     862     722     597     428
##  label8  label9 label10 label11 label12 label13 label14
##     480     178     253     289    1816    1799      34
```

Create Multilabel Learners

Algorithm adaptation method:

```
lrn.rfSRC = makeLearner("multilabel.randomForestSRC")
```

Problem transformation method:

```
lrn.rf = makeLearner("classif.ranger")
lrn.rf.cc = makeMultilabelClassifierChainsWrapper(lrn.rf)
```

Train and Predict

```r
n = getTaskSize(y.task)
train.set = seq(1, n, by = 2)
test.set = seq(2, n, by = 2)

mod.rfSRC = train(lrn.rfSRC, task = y.task, subset = train.set)
mod.rf.cc = train(lrn.rf.cc, task = y.task, subset = train.set)

pred.rfSRC = predict(mod.rfSRC, task = y.task, subset = test.set)
pred.rf.cc = predict(mod.rf.cc, task = y.task, subset = test.set)
```

Accessing Performance Values

```r
performance(pred.rfSRC,
  measures = list(multilabel.subset01, multilabel.hamloss))

## multilabel.subset01  multilabel.hamloss
##           0.8485099           0.1963103

performance(pred.rf.cc,
  measures = list(multilabel.subset01, multilabel.hamloss))

## multilabel.subset01  multilabel.hamloss
##           0.7913907           0.1922304
```

Outlook

- Multilabel classification is a subclass of the more generalized multi-output prediction problem, where targets can be of any kind
- This includes multivariate regression as well
- Implementation in mlr is planned

Links

- Tutorial: http://mlr-org.github.io/mlr/articles/tutorial/devel/multilabel.html
- Benchmark paper: https://journal.r-project.org/archive/2017/RJ-2017-012/RJ-2017-012.pdf