# Multi-label classification

Fridah Wanjala

5/25/2018

# Introduction

Figure 1

# What about this

Figure 2

# Multi-Class vs Multi-Label

We are used to carrying out supervised learning using single label classifcation :

- Binary Classification Problem
- Multiclass Classification Problem

Single label : there are multiple categories but each instance is assigned only one

Multi-Label : each instance can be assigned with multiple categories

# Pictogram

Table : Single-label $Y \in \{0, 1\}$

| $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $Y$ |
|---|---|---|---|---|---|
| 1 | 0.1 | 3 | 1 | 0 | 0 |
| 0 | 0.9 | 1 | 0 | 1 | 1 |
| 0 | 0.0 | 1 | 1 | 0 | 0 |
| 1 | 0.8 | 2 | 0 | 1 | 1 |
| 1 | 0.0 | 2 | 0 | 1 | 0 |
| 0 | 0.0 | 3 | 1 | 1 | ? |

Table : Multi-label $Y_1, \ldots, Y_L \in 2^L$

| $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.1 | 3 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0.9 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0.0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0.8 | 2 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0.0 | 2 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0.0 | 3 | 1 | 1 | ? | ? | ? | ? |

Figure 3

# Applications

- Text categorization Movie plot summaries can be associated with several genres

| example | *abandoned* $X_1$ | *accident* $X_2$ | ... | *violent* $X_{1000}$ | *wedding* $X_{1001}$ | $Y$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | ... | 1 | 0 | {romance, comedy } |
| 2 | 0 | 1 | ... | 0 | 1 | {horror} |
| 3 | 0 | 0 | ... | 1 | 0 | {romance} |
| 4 | 1 | 1 | ... | 0 | 1 | {horror, action} |
| 5 | 1 | 0 | ... | 0 | 1 | {action} |

Figure 4

# Applications

- Medical diagnosis Medical history and symptoms could be associated with different ailments

- Others:
- Image anootation
- Audio and video description
- Bioinformatics - classification of genes

# Methods

There are three methods to solve a multi-label classification problem,:

- Problem Transformation : try to transform the multilabel classification into binary or multiclass classification problems
- Adapted Algorithm : adapt multiclass algorithms so they can be applied directly to the problem. For example KNN, Rndom forests, SVM
- Ensemble approaches

# Problem Transformation

- Binary Relevance (BR): Basically treats each label as a separate single class classification problem.

| X | $Y_1$ |
|---|---|
| $x^{(1)}$ | 0 |
| $x^{(2)}$ | 1 |
| $x^{(3)}$ | 0 |
| $x^{(4)}$ | 1 |
| $x^{(5)}$ | 0 |

| X | $Y_2$ |
|---|---|
| $x^{(1)}$ | 1 |
| $x^{(2)}$ | 0 |
| $x^{(3)}$ | 1 |
| $x^{(4)}$ | 0 |
| $x^{(5)}$ | 0 |

| X | $Y_3$ |
|---|---|
| $x^{(1)}$ | 1 |
| $x^{(2)}$ | 0 |
| $x^{(3)}$ | 0 |
| $x^{(4)}$ | 0 |
| $x^{(5)}$ | 0 |

| X | $Y_4$ |
|---|---|
| $x^{(1)}$ | 0 |
| $x^{(2)}$ | 0 |
| $x^{(3)}$ | 0 |
| $x^{(4)}$ | 1 |
| $x^{(5)}$ | 1 |

Each of the binary classifiers votes separately to get the final result.

# Problem Transformation

- Classifier Chains (CC): Similar to BR, a ML problem is transformed into single label probems. Here the first classifier is trained just on the input data and then each next classifier is trained on the input data and all the previous classifiers in the chain.

| X | y1 |
|---|---|
| x1 | 0 |
| x2 | 1 |
| x3 | 0 |

Classifier 1

| X | y1 | y2 |
|---|---|---|
| x1 | 0 | 1 |
| x2 | 1 | 0 |
| x3 | 0 | 1 |

Classifier 2

| X | y1 | y2 | y3 |
|---|---|---|---|
| x1 | 0 | 1 | 1 |
| x2 | 1 | 0 | 0 |
| x3 | 0 | 1 | 0 |

Classifier 3

| X | y1 | y2 | y3 | y4 |
|---|---|---|---|---|
| x1 | 0 | 1 | 1 | 0 |
| x2 | 1 | 0 | 0 | 0 |
| x3 | 0 | 1 | 0 | 0 |

Classifier 4

- It achieves higher predictive performance than BR
- Preserves label correlation

# Problem Transformation

- Label Power set (LP) : Generates a new class for every combination of labels and then solves the problem using multiclass classification approaches.

| X | y1 | y2 | y3 | y4 |
|---|---|---|---|---|
| x1 | 0 | 1 | 1 | 0 |
| x2 | 1 | 0 | 0 | 0 |
| x3 | 0 | 1 | 0 | 0 |
| x4 | 0 | 1 | 1 | 0 |
| x5 | 1 | 1 | 1 | 1 |
| x6 | 0 | 1 | 0 | 0 |

# Performance Metrics

- Accuracy : proportion of correctly predicted labels with respect to the total number of labels for each instance
- Hamming-loss : symmetric difference between predicted and true labels and divided by the total number of labels in the MLD. The smaller the value of hamming loss, better the performance.

# Case study

Objective: The main objective of this study is to develop a classification rule that allows to correctly identify a patient with Chronic kidney disease (CKD) based on physical symptoms and data from blood analysis.

These algorithms allow to classify CKD, Hypertension and Diabetes as unified pathological entity since Hypertension and Diabetes could be underlying causes or complication of CKD.

# Case study

Study: In the study physical symptoms, clinical and blood test data were recorded from 401 patients. 250 patients had Chronic kidney disease (CKD), 147 patients were diagnosed with Hypertension and Diabetes has been diagnosed in 137 patients.

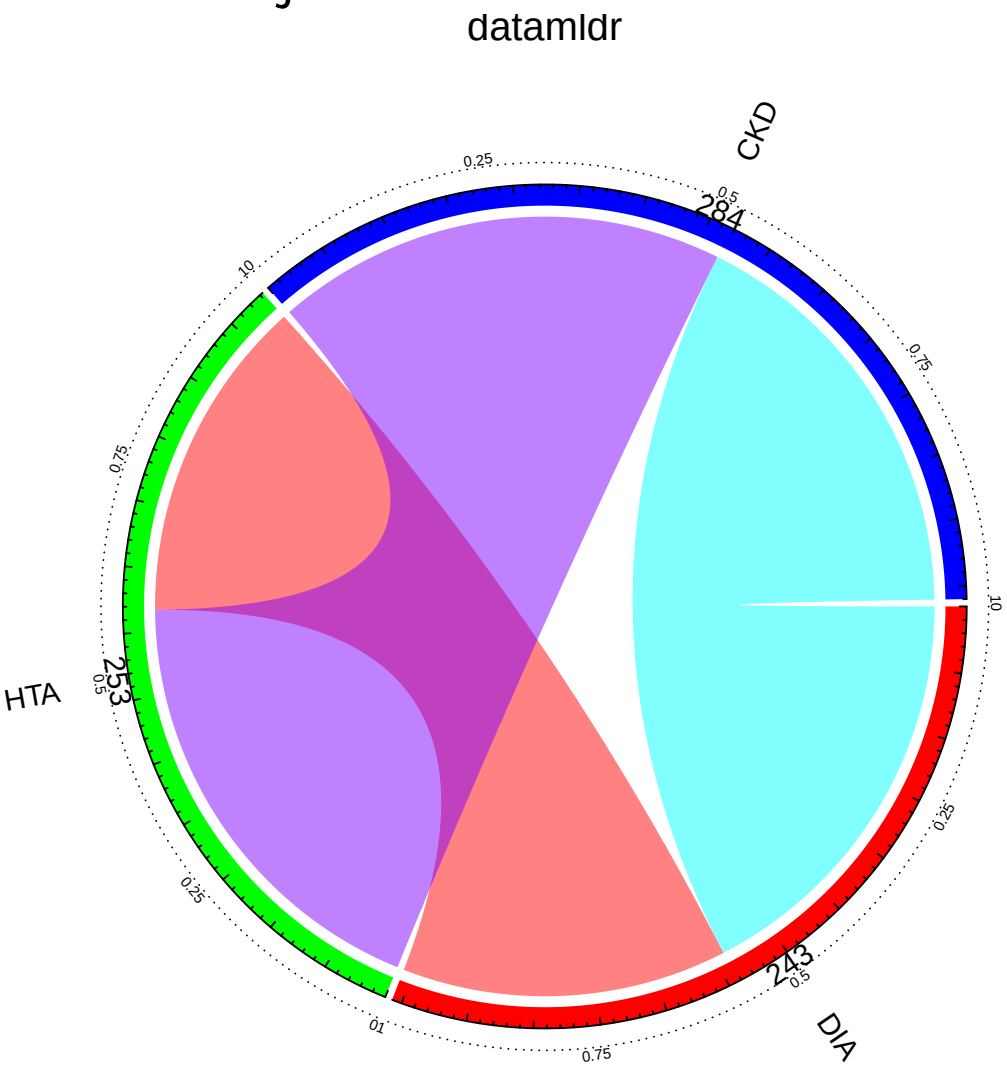
Data Source

| Age | BP | DIA | HTA | CKD |
|---|---|---|---|---|
| 38 | 80 | TRUE | TRUE | TRUE |
| 62 | 50 | FALSE | FALSE | TRUE |
| 54 | 80 | TRUE | FALSE | TRUE |
| 38 | 70 | FALSE | TRUE | TRUE |
| 42 | 80 | FALSE | FALSE | TRUE |
| 52 | 90 | TRUE | TRUE | TRUE |

# Data exploration

Data format : Each label should be coded as TRUE or FALSE

Using `mldr_from_dataframe` from the `mldr` package we generate an mldr object from a data.frame and a vector with label indices



# Creating a task

Create a MultilabelTask - specify a vector of targets which correspond to the names of logical variables in the data.frame

```
## Supervised task: multi
## Type: multilabel
## Target: DIA,HTA,CKD
## Observations: 397
## Features:
##    numerics     factors     ordered functionals
##          14           5           0           0
## Missings: TRUE
## Has weights: FALSE
## Has blocking: FALSE
## Has coordinates: FALSE
## Classes: 3
## DIA HTA CKD
## 137 147 250
```

# Create a learner

Use the `makeLearner` function to create a learner for your probelm. All classification learners start with `classif.` all regression learners with `regr.` all survival learners start with `surv.` all clustering learners with `cluster.`

```
## [1] "Algorithm adaptation methods: Randomforest"
```

```
## Learner multilabel.randomForestSRC from package randomFo
## Type: multilabel
## Name: Random Forest; Short name: rfsrc
## Class: multilabel.randomForestSRC
## Properties: missings,numerics,factors,prob,weights
## Predict-Type: prob
## Hyperparameters: na.action=na.impute
```

# Train and Predict

You can train a model as usual with a multilabel learner and a multilabel task as input.

Using `mlr`'s `predict` command, pass the trained model and either the task to the task argument or some new data to the newdata argument.

| | truth.DIA | truth.HTA | truth.CKD | prob.DIA | prob.HTA | prob |
|---|---|---|---|---|---|---|
| 4 | FALSE | TRUE | TRUE | 0.5596932 | 0.5799250 | 1.00 |
| 5 | FALSE | FALSE | TRUE | 0.2946445 | 0.2432667 | 0.99 |
| 9 | TRUE | TRUE | TRUE | 0.4915667 | 0.6765667 | 1.00 |
| 10 | TRUE | TRUE | TRUE | 0.3998833 | 0.6804833 | 1.00 |
| 11 | TRUE | TRUE | TRUE | 0.7312369 | 0.7638369 | 0.99 |

# Performance

| model | multilabel.acc | multilabel.hamloss | multilabel.ppv | m |
|---|---|---|---|---|
| RandomForest | 0.85 | 0.13 | 0.88 | |
| Binaryrelevance | 0.85 | 0.14 | 0.87 | |
| ClassifierChains | 0.86 | 0.13 | 0.89 | |

References [Multilabel example] (https://rpubs.com/ledongnhatnam/259348) [Multilabel documentation] (https://mlr-org.github.io/mlr-tutorial/devel/html/multilabel/index.html) [Intergrated learners] (http://mlr-org.github.io/mlr-tutorial/release/html/integrated_learners/)