# R Ultimate Multilabel Dataset Repository

Francisco Charte, David Charte, Antonio Rivera, María José del Jesus, Francisco Herrera
Dep. of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain.
Dep. of Computer Science, University of Jaén, Jaén, Spain

**Abstract**

Multilabeled data is everywhere on the Internet. From news on digital media and entries published in blogs, to videos hosted in Youtube, every object is usually tagged with a set of labels. This way they can be categorized into several non-exclusive groups. However, publicly available multilabel datasets (MLDs) are not so common. There is a handful of websites providing a few of them, using disparate file formats. Finding proper MLDs, converting them into the correct format and locating the appropriate bibliographic data to cite them are some of the difficulties usually confronted by researchers and practitioners.

In this paper RUMDR (*R Ultimate Multilabel Dataset Repository*), a new multilabel dataset repository aimed to fuse all public MLDs, is introduced, along with mldr.datasets, an R package which eases the process of retrieving MLDs and their bibliographic information, exporting them to the desired file formats and partitioning them.

# Introduction

Multilabel classification (MLC) is a sort of machine learning technique characterized by the fact that each data sample is associated to a group of labels or tags. MLC is useful in many different fields, including protein classification [EW01], image labeling [CTH+09], tag suggestion [CRdJH15c], and text categorization [LYRL04]. Several dozens of multilabel datasets (MLDs) have been produced from these areas in late years, and some of them are publicly available in web repositories.

The research in MLC algorithms [GV14, ZZ14], as well as in preprocessing methods [CRdJH15b, CRdJH15a] has been extraordinary, with hundreds of proposals already published. These development efforts rely on the availability of MLDs in order to test their behavior and performance. In the initial stages, a decade ago, MLDs were not publicly available, so most authors produced them by themselves. Some of those MLDs are now sparsely hosted in several web repositories and disparate file formats, a situation that does not ease the work in new developments.

In addition to the data itself, in the adequate format to work with it, the corresponding bibliographic information to properly give attribution to who produced the MLD is also needed. Sometimes finding this piece of data can be very time consuming. Also, to conduct empirical experiments the MLDs usually have to be partitioned, obtaining training and test folds. Therefore, new research attempts have to fulfill the process of locating the MLDs along with the bibliographic data, converting them to the desired format and partitioning them. Additional steps would be obtaining basic characterization metrics from these MLDs, determining how many labels they have, how frequent each label is, their complexity, etc.

Aiming to ease most of the steps in the described process, two new proposals are made on this paper:

- **RUMDR:** The *R Ultimate Multilabel Dataset Repository* is a GitHub[1] Repository that collects the MLDs publicly available and provides them under an unified file format.
- **mldr.datasets**: It is an R package that automates the use of RUMDR, providing functions to download the MLDs, recover their bibliographic information, export them to several file formats and partition them, among other tasks.

This paper is structured as follows. In Section  the previous works aimed to provide multilabel datasets repositories are enumerated and their main characteristics are shown. Section  describes the content of RUMDR, our ultimate multilabel dataset repository, and how it has been structured. How to use the mldr.datasets package to download MLDs, obtain information about them, partitioning and exporting them is the goal of Section . Lastly, some conclusions are provided in Section .

# Related Work

Many early multilabel works produced original MLDs, and some authors published them in their own web page or other publicly accessible web sites, allowing third parties to use them. For instance, the datasets created by the authors of [CRdJH15c] are available at `http://figshare.com/articles/Multilabel_datasets_from_Stack_Exchange_forums/1385315`. Many of these MLDs were compiled, mainly by developers of MLC tools, giving rise to public repositories such as the following:

- **LibSVM:** This is a well-known library for Support Vector Machines (SVM) [CL11], and their authors maintain a repository with all kind of datasets, including binary, multiclass and multilabel[2] ones. The

[1] `http://GitHub.com`
[2] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multilabel.html`

file format is similar to CSV (*Comma-Separated Values*), but with a sparse representation and locating the labels at the beginning of each row.

- **MEKA:** It is a multilabel software tool designed by Read [RR] and founded on WEKA. Its associated repository[3] provides over 20 MLDs. The file format is ARFF, but with a special header that indicates how many attributes are labels. Those are always located at the beginning.

- **Mulan:** This is probably the leading multilabel software [TXVV11], including reference implementation for many of the published MLC algorithms. The Mulan repository[4] is also the largest one, with more than 25 MLDs available at this moment. The base file format is ARFF, but the labels can be any position since their names are supplied in a separate XML file.

- **KEEL:** It is a general purpose data mining software tool [AFFL$^+$11], similar to WEKA, and it also has an associated dataset repository[5] which includes some MLDs. The file format is also ARFF-based, but with specific header fields that indicate which attributes are labels.

As can be seen, the goal of each one of these repositories is to provide the MLDs in the correct file format for the tool they are interested in. To use the MLDs which are exclusively available in one of the repositories, for instance some of the Mulan datasets, with another tools, such as LibSVM, MEKA or KEEL, a previous conversion step is mandatory.

# The R Ultimate Multilabel Dataset Repository

Although some specific multilabel software programs, such as Mulan, are actively used to work in new multilabel developments, none of them stand out for their ease of use as exploratory data analysis tools, or for their ability to be interactively used to develop algorithm prototypes. On the contrary, these are tasks in which the use of R [R C14] is prominent. Its large collection of packages makes data analysis and visualization easier, as well as the creation of proofs of concept for new methods.

Regarding the processing of multilabeled data, the main exploratory analysis tasks can be accomplished by means of the mldr package [CC15]. It only includes three typical MLDs (genbase, emotions and birds), but it is able to load any other MLD in Mulan and MEKA formats. We have used the functionality in mldr, along with some custom R code, to build RUMDR. This new repository holds all the publicly available MLDs, taken from the previously mentioned repositories and translated to a common format.

In this section the file format of RUMDR MLDs and the structure of the repository are detailed. Furthermore, the list of MLDs initially provided by RUMDR is also enumerated.

## RUMDR structure and file format

RUMDR is a multilabel dataset repository hosted at `https://github.com/fcharte/mldr.datasets`. It is a public GitHub repository which also hosts the source code of the mldr.datasets package described in Section . The repository has two folders containing MLDs:

- **data:** It holds 10 small and medium sized MLDs, such as birds, emotions, genbase, medical and slashdot. They are among the most used in the literature.

[3] `http://sourceforge.net/projects/meka/files/Datasets/`
[4] `http://mulan.sourceforge.net/datasets-mlc.html`
[5] `http://sci2s.ugr.es/keel/multilabel.php`

- **additional-data:** It holds 52 additional MLDs. Some of them are subsets of bigger datasets, such as rcv1sub1 to rcv1sub5 which are five subsets of the well-known RCV1 text corpus from Reuters.

The purpose of grouping the MLDs into two different sets is to ease the functionality of the mldr.datasets package, as will be further explained. All the files stored into these two folders have the same file format. They are standard .rda[6] R files, thus can be loaded into the current workspace by simply typing `load('filename.rda')` at the R console. This would get into memory an R object with the same name but without the .rda extension.

Once the object is in memory, it can be queried to access the data it contains through the syntax `object$dataset`. In addition, information about the labels, labelsets and other metrics can be obtained by querying `object$labels`, `object$labelsets` and `object$measures`, respectively.

## Datasets in RUMDR

At launch time RUMDR holds more than forty distinct MLDs, including several subsets of large datasets and also some pre-partitioned datasets. They sum 62 individual .rda files in total. 41 of them originate from the text field, 15 more from the image field, 3 from the sound/music field, 2 from the protein/genetics area, and the last one from the video field. The name of each one, the number of individual objects, where they come from and the folder where they are stored within the repository are shown in Table 1.

Most cases consist in only one MLD, but there are some special situations. These must be detailed to be able to use them:

- **corel16k:** These MLDs, there are 10 in total with names from corel16k001 to corel16k010, are subsets of the Corel image database [BDF+03]. Each one holds almost 14 000 instances and the same set of input features, but there are slight differences in the sets of labels.

- **EUR-Lex:** There are six files in this set, named eurlexdc_tra, eurlexdc_test, eurlexev_tra, eurlexev_test, eurlexsm_tra and eurlexsm_test. They correspond to the train and test partitions of the directory codes (dc), EUROVOC descriptors (ev), and subject matters (sm) of the EUR-Lex dataset [MF08]. Each object consist in a list of 10 folds.

- **nus-wide:** The original NUS-WIDE dataset [CTH+09] used a set of 500 input features to represent each image. An alternative version, with 128 input features, is also available. The former is accessible as nuswide_BoW, while the latter is named nuswide_VLAD.

- **rcv1v2** and **reuters:** The Reuters corpus, best known as RCV1-v2 (*Reuters Corpus Volume 1 version 2*) [LYRL04], is a large text corpus generated from news published by Reuters. The .rda files rcv1sub1 to rcv1sub5 consist in five MLDs which are subsets of RCV1-v2, containing 6 000 instances each one of them while preserving all the input features. The reuters MLD [Rea10] is a reduced version with only 500 input features.

- **stackexchange:** It is a collection of six MLDs generated from text collected in a selection of Stack Exchange forums [CRdJH15c]. The individual files are stackex_chess, stackex_chemistry, stackex_coffee, stackex_cooking, stackex_cs and stackex_philosophy.

- **yahoo:** As the previous case, this also consists of a collection of MLDs. They were produced from text extracted from the web, specifically from the Yahoo! web directory. There are a total of 11 MLDs, named yahoo_arts, yahoo_business, yahoo_computers, yahoo_education, yahoo_entertainment, yahoo_health, yahoo_recreation, yahoo_reference, yahoo_science, yahoo_social and yahoo_society.

[6] This is a compressed file format of the representation of R objects in memory.

Table 1: MLDs initially included in RUMDR

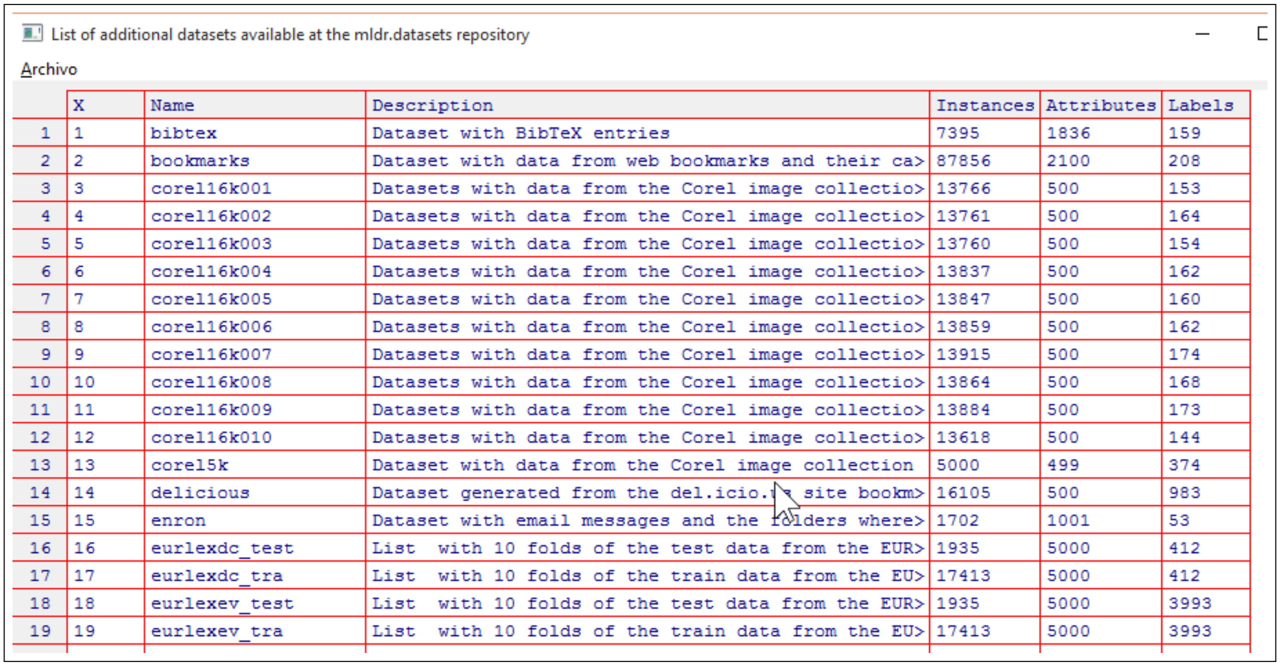
| Field | Name | # MLDs | Reference | Folder |
|---|---|---|---|---|
| Text | 20ng | 1 | [Lan95] | data |
| | bibtex | 1 | [KTV08] | additional-data |
| | bookmarks | 1 | [KTV08] | additional-data |
| | delicious | 1 | [TKV08] | additional-data |
| | enron | 1 | [KY04] | additional-data |
| | EUR-Lex | 6 | [MF08] | additional-data |
| | imdb | 1 | [RPHF11] | additional-data |
| | langlog | 1 | [Rea10] | data |
| | medical | 1 | [CDG+07] | data |
| | ohsumed | 1 | [Joa98] | additiona-data |
| | rcv1v2 | 5 | [LYRL04] | additional-data |
| | reuters | 1 | [Rea10] | additional-data |
| | slashdot | 1 | [RPHF11] | data |
| | stackexchange | 6 | [CRdJH15c] | data & additional-data |
| | tmc2007 | 2 | [SZU05] | additional-data |
| | yahoo | 11 | [US02] | additional-data |
| Sound/Music | birds | 1 | [BLN+12] | data |
| | cal500 | 1 | [TBTL08] | data |
| | emotions | 1 | [WSR06] | data |
| Image | corel16k | 10 | [BDF+03] | additional-data |
| | corel5k | 1 | [DBdFF02] | additional-data |
| | flags | 1 | [GPF13] | data |
| | nus-wide | 2 | [CTH+09] | additional-data |
| | scene | 1 | [BLSB04] | additional-data |
| Video | mediamill | 1 | [SWvG+06] | additional-data |
| Protein/Genetics | genbase | 1 | [DTMV05] | data |
| | yeast | 1 | [EW01] | additional-data |

## The mldr.datasets Package

Although RUMDR is valuable by itself, as an unified multilabel repository from where almost all available MLDs can be download in a common file format, we aimed to increase its usefulness by pairing it with a specific software package. This is an R package named mldr.datasets, and it is already available on CRAN (*The Comprehensive R Archive Network*). Therefore, it can be installed from the R command line by simply typing `install.packages('mldr.datasets')`, then loaded into memory with `library(mldr.datasets)`.

Once loaded, the ten MLDs hosted in the data folder of RUMDR will be immediately available as they are embedded into the package. This means that queries such as `emotions$measures`, `genbase$labels` or `flags$labelsets` can be entered to get general information about emotions, label information from genbase and data related to the labelsets in flags. The list of the remainder datasets, those that can be downloaded at any time, is retrieved with the `mldrs()` function. This returns a table as the shown in Fig. 1.

To load any of the additional MLDs, those that are not embedded into the package, there are two alternatives. The first one is calling the function that shares the same name that the desired MLD, for instance


List of additional datasets available at the mldr.datasets repository

Archivo

| | X | Name | Description | Instances | Attributes | Labels |
|---|---|---|---|---|---|---|
| 1 | 1 | bibtex | Dataset with BibTeX entries | 7395 | 1836 | 159 |
| 2 | 2 | bookmarks | Dataset with data from web bookmarks and their ca> | 87856 | 2100 | 208 |
| 3 | 3 | corel16k001 | Datasets with data from the Corel image collectio> | 13766 | 500 | 153 |
| 4 | 4 | corel16k002 | Datasets with data from the Corel image collectio> | 13761 | 500 | 164 |
| 5 | 5 | corel16k003 | Datasets with data from the Corel image collectio> | 13760 | 500 | 154 |
| 6 | 6 | corel16k004 | Datasets with data from the Corel image collectio> | 13837 | 500 | 162 |
| 7 | 7 | corel16k005 | Datasets with data from the Corel image collectio> | 13847 | 500 | 160 |
| 8 | 8 | corel16k006 | Datasets with data from the Corel image collectio> | 13859 | 500 | 162 |
| 9 | 9 | corel16k007 | Datasets with data from the Corel image collectio> | 13915 | 500 | 174 |
| 10 | 10 | corel16k008 | Datasets with data from the Corel image collectio> | 13864 | 500 | 168 |
| 11 | 11 | corel16k009 | Datasets with data from the Corel image collectio> | 13884 | 500 | 173 |
| 12 | 12 | corel16k010 | Datasets with data from the Corel image collectio> | 13618 | 500 | 144 |
| 13 | 13 | corel5k | Dataset with data from the Corel image collection | 5000 | 499 | 374 |
| 14 | 14 | delicious | Dataset generated from the del.icio.us site bookm> | 16105 | 500 | 983 |
| 15 | 15 | enron | Dataset with email messages and the folders where> | 1702 | 1001 | 53 |
| 16 | 16 | eurlexdc_test | List with 10 folds of the test data from the EUR> | 1935 | 5000 | 412 |
| 17 | 17 | eurlexdc_tra | List with 10 folds of the train data from the EU> | 17413 | 5000 | 412 |
| 18 | 18 | eurlexev_test | List with 10 folds of the test data from the EUR> | 1935 | 5000 | 3993 |
| 19 | 19 | eurlexev_tra | List with 10 folds of the train data from the EU> | 17413 | 5000 | 3993 |

Figure 1: List of additional mldrs.

`bibtex()`, `corel5k()` or `scene()`. The second way consists in using the function `check_n_load.mldr()`, providing as argument the name of the MLD. Assuming that a new MLD called `newmld` existed at RUMDR, entering `check_n_load.mldr('newmld')` in the R command line would download and install it in our system. Either way, regardless of the function used, firstly whether the needed MLD is locally available will be checked. If so, it is simply loaded into memory. Otherwise, the corresponding file is downloaded from RUMDR, copied to the local installation of mldr.datasets, and then loaded.

The MLDs in memory are R objects with class `mldr`, and the mldr.datasets package includes several functions to deal with this type of objects, easing the process of obtaining bibliographic information, partitioning them and exporting them. They are explained below.

## Obtaining BibTeX Entries

All MLDs hosted in RUMDR hold bibliographic information, specifically BibTeX entries that can be used while writing new works with LaTeX. The entry associated to an MLD can be retrieved by means of the generic function `toBibtex()`, delivering the mldr object as parameter. The raw entry can be copied to the clipboard or displayed in the R console, as shown in Fig. 2.

The BibTeX entries of all MLDs in RUMDR are also available in the README[7] file stored in the additional-data folder of the repository.

[7] `https://github.com/fcharte/mldr.datasets/blob/master/additional-data/README.md`

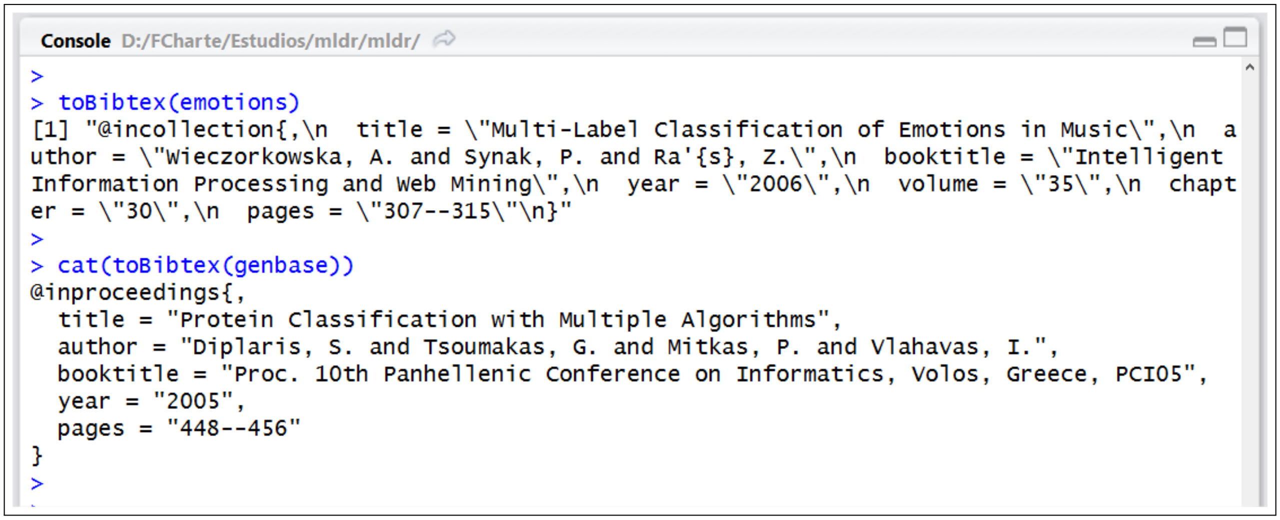



Figure 2: Obtaining bibliographic information from MLDs.

## Theoretical Complexity Score

All mldr objects have a member called `measures` containing disparate characterization metrics, such as the number of instances, attributes and labels, imbalance levels, etc. mldr.datasets adds a data item to this member, named `tcs`, with the value of a metric introduced in [CRdJH16], TCS (*Theoretical Complexity Score*). It is computed as shown in (1), $f$ being the number of input features and $k$ the number of labels of the dataset $D$.

$$TCS(D) = \log(f \times k \times |unique\ labelsets|) \tag{1}$$

The goal of this new metric is to give a glimpse of how hard would be for a classifier to learn from each MLD. Theoretically, the bigger the input and output spaces are, the more complex the generated model will be, making it more difficult to properly adjust. The number of different combinations of labelsets is also a factor to be considered while working in the multilabel field.

Relying on the precalculated TCS values of the MLDs, it is easy to sort them according to their theoretical complexity, as shown in Fig. 3. This information can be useful to chose the MLDs to be used in a new experimentation.

## Partitioning MLDs

With few exceptions, such as the MLDs generated from the EUR-Lex datasets, all files available in RUMDR contain full datasets. Prior to their usage in any experiment they have to be partitioned, obtaining training and test partitions with subsets of the instances they contain. The mldr.datasets package provides two functions to accomplish this task:

- `random.kfolds()`: Randomly samples the instances in the dataset until the desired number of folds is produced.

```
Console D:/FCharte/Estudios/mldr/mldr/
>
> as.matrix(sort(sapply(data(package = "mldr.datasets")$result[,3], function(mld) get(mld
)$measures$tcs)))
                 [,1]
flags        8.879333
emotions     9.364262
scene       10.183389
birds       13.395470
genbase     13.839914
ng20        13.916803
slashdot    15.124688
cal500      15.597163
medical     15.628586
langlog     16.946263
stackex_chess 18.779425
>
```

Figure 3: Embedded MLDs in mldr.datasets sorted according to their TCS value.

- `stratified.kfolds()`: It was introduced in [CRdJH16]. First groups the instances into strata attending to the frequency of the labels appearing in them, then each strata is randomly sampled to divide it into the number of desired folds. This way the distribution of rare samples among the folds results more balanced.

Both functions take as input the same parameters, the MLD to be partitioned, the number of folds and the seed for the random generator. The last two arguments have default values, 5 for the number of folds and 10 as seed. The obtained result is a list with as many items as folds the user asked for. Each item consists of a `train` and a `test` element, both mldr objects that can be used in the same way as the full MLD. The example shown in Fig. 4 demonstrates the use of both functions.

## Exporting MLDs to Other File Formats

Even though R is an environment from which the MLDs can be explored, analyzed and given as input to different preprocessing and learning algorithms, it could be interesting to use them with other software tools, such as the aforementioned Mulan, MEKA, etc. This is a task that the mldr.datasets package considerably simplifies through the `write.mldr()` function. It is able to export any MLD to Mulan, MEKA, KEEL, LibSVM and CSV formats, using dense or sparse representation in the first three cases.

An invocation to `write.mldr()` needs at least one parameter, but it can take three additional ones. The names and goal of each parameter are the following:

- `mld`: It can be an mldr object or the result returned by the `random.kfolds()` and `stratified.kfolds()` functions. In the latter case write operation is performed for each training and test partition.

- `format`: This parameter can be a string or a vector of strings, stating the formats the MLD have to be exported to. The valid values are `'MULAN'`, `'MEKA'`, `'KEEL'`, `'LIBSVM'` and `'CSV'`. By default `c('MULAN', 'MEKA')` is used. That means that the MLD will be written in these two formats.

```
Console D:/FCharte/Estudios/mldr/mldr/
>
> emotions.folds <- random.kfolds(emotions)
> summary(emotions.folds[[1]]$train)
  num.attributes num.instances num.inputs num.labels num.labelsets num.single.labelsets
1             78           474         72          6            25                    2
  max.frequency cardinality   density   meanIR    scumble scumble.cv      tcs
1            65    1.848101 0.3080169 1.488775 0.01166691   1.359773 9.287301
>
> emotions.folds <- stratified.kfolds(emotions, k = 10)
> summary(emotions.folds[[4]]$test)
  num.attributes num.instances num.inputs num.labels num.labelsets num.single.labelsets
1             78            60         72          6            14                    2
  max.frequency cardinality   density   meanIR    scumble scumble.cv      tcs
1             9    1.883333 0.3138889 1.555556 0.01600574   1.029966 8.707483
>
>
```

Figure 4: Partitioning the emotions MLD randomly and stratifiedly

- `sparse`: It is applicable only with Mulan, MEKA and KEEL file formats, all of them based on the WEKA ARFF format. By default it gets the `FALSE` value, thus dense representation of features is used. Assigning it the `TRUE` value a disperse representation will be used.

- `basename`: One call to `write.mldr()` can create several files. The Mulan format produces one .arff and one .xml file. The CSV format generates two, one with the data and another one with the labels. In addition, if a partitioned datasets is given as input two files, one for test and one for training, will be written for each partition. With this parameter the base name for all these files is established. By default, mldr.datasets looks for a `name` member in the mldr object. If it is present, it is used as basename; otherwise the value `'unnamed_mldr'` is used.

A single call as the shown below will partition the given MLD into 10 subsets and write it in the five file formats supported by the package, generating 120 files in total:

```
write.mldr(stratified.kfolds(emotions, k = 10),
    format = c('MULAN', 'MEKA', 'KEEL', 'LIBSVM', 'CSV'),
    basename = 'emotions')
```

# Conclusions

The process of analyzing, researching and putting into practice new multilabel solutions demands the availability of enough datasets, in the proper format and with the corresponding bibliographic information. Additional data about its structure and complexity are also quite useful. The automation of these MLDs manipulation tasks is the goal of RUMDR and the associated mldr.datasets R package.

RUMDR is a new repository in which almost all publicly available multilabel datasets have been collected, being stored in a common file format. The mldr.datasets package is a software tool that eases the loading of these MLDs from R, as well as the retrieval of metrics and bibliographic information and the partitioning and exporting to several file formats.

The main goal behind the development of RUMDR and mldr.datasets has been to make easier the work of researchers and practitioners interested in multilabeled data. The repository will be maintained by incorporating new MLDs that may be published in the future.

**Acknowledgments**: This work was partially supported by the Spanish Ministry of Science and Technology under projects TIN2014-57251-P and TIN2012-33856, and the Andalusian regional projects P10-TIC-06858 and P11-TIC-7765.

# Bibliography


[AFFL$^+$11] J. Alcala-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. Keel data-mining software tool: Data set repository and integration of algorithms and experimental analysis framework. *J. of Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287, 2011.

[BDF$^+$03] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan. Matching words and pictures. *J. Mach. Learn. Res.*, 3:1107–1135, 2003.

[BLN$^+$12] F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, R. Raich, S. J. K. Hadley, A. S. Hadley, and M. G. Betts. Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. *The Journal of the Acoustical Society of America*, 131(6):4640–4650, 2012.

[BLSB04] M. Boutell, J. Luo, X. Shen, and C. Brown. Learning multi-label scene classification. *Pattern Recognit.*, 37(9):1757–1771, 2004.

[CC15] Francisco Charte and David Charte. Working with multilabel datasets in R: The mldr package. *The R Journal*, 7(2):149–162, December 2015.

[CDG$^+$07] Koby Crammer, Mark Dredze, Kuzman Ganchev, Partha P. Talukdar, and Steven Carroll. Automatic Code Assignment to Medical Text. In *Proc. Workshop on Biological, Translational, and Clinical Language Processing, Prague, Czech Republic, BioNLP'07*, pages 129–136, 2007.

[CL11] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[CRdJH15a] Francisco Charte, Antonio J. Rivera, Mara J. del Jesus, and Francisco Herrera. MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation. *Knowledge-Based Systems*, 89:385–397, 2015.

[CRdJH15b] Francisco Charte, Antonio J. Rivera, María J. del Jesus, and Francisco Herrera. Addressing imbalance in multilabel classification: Measures and random resampling algorithms. *Neurocomputing*, 163(0):3–16, 2015.

[CRdJH15c] Francisco Charte, Antonio J. Rivera, Maria J. del Jesus, and Francisco Herrera. QUINTA: A question tagging assistant to improve the answering ratio in electronic forums. In *EUROCON 2015 - International Conference on Computer as a Tool (EUROCON), IEEE*, pages 1–6, Sept 2015.

[CRdJH16] Francisco Charte, Antonio Rivera, Mara Jos del Jesus, and Francisco Herrera. On the impact of dataset complexity and sampling strategy in multilabel classifiers performance. In *Hybrid Artificial Intelligence Systems*, LNCS. Springer International Publishing, 2016.

[CTH+09] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. NUS-WIDE: a real-world web image database from National University of Singapore. In *Proc. of the ACM international conference on image and video retrieval*, page 48. ACM, 2009.

[DBdFF02] P. Duygulu, K. Barnard, J. de Freitas, and D. Forsyth. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In *Proc. 7th European Conf. on Computer Vision-Part IV, Copenhagen, Denmark, ECCV'02*, pages 97–112, 2002.

[DTMV05] Sotiris Diplaris, Grigorios Tsoumakas, Pericles Mitkas, and Ioannis Vlahavas. Protein Classification with Multiple Algorithms. In *Proc. 10th Panhellenic Conference on Informatics, Volos, Greece, PCI'05*, pages 448–456, 2005.

[EW01] André Elisseeff and Jason Weston. A Kernel Method for Multi-Labelled Classification. In *Advances in Neural Information Processing Systems 14*, volume 14, pages 681–687. MIT Press, 2001.

[GPF13] E. C. Gonçalves, A. Plastino, and A. A. Freitas. A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In *Proc. 25th IEEE International Conference on Tools with Artificial Intelligence (ICTAI13)*, pages 469–476, 2013.

[GV14] E. Gibaja and S. Ventura. Multi-label learning: a review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(6):411–444, 2014.

[Joa98] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Proc. 10th European Conference on Machine Learning*, pages 137–142. Springer-Verlag, 1998.

[KTV08] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. Multilabel Text Classification for Automated Tag Suggestion. In *Proc. ECML PKDD'08 Discovery Challenge, Antwerp, Belgium*, pages 75–83, 2008.

[KY04] Bryan Klimt and Yiming Yang. The Enron Corpus: A New Dataset for Email Classification Research. In *Proc. ECML'04, Pisa, Italy*, pages 217–226. 2004.

[Lan95] Ken Lang. Newsweeder: Learning to filter netnews. In *Proc. 12th International Conference on Machine Learning*, pages 331–339, 1995.

[LYRL04] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

[MF08] E. L. Mencia and J. Fürnkranz. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Machine Learning and Knowledge Discovery in Databases*, pages 50–65. Springer, 2008.

[R C14] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2014.

[Rea10] Jesse Read. *Scalable multi-label classification.* PhD thesis, University of Waikato, 2010.

[RPHF11] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Mach. Learn.*, 85:333–359, 2011.

[RR] J. Read and P. Reutemann. MEKA multi-label dataset repository.

[SWvG+06] Cees G. M. Snoek, Marcel Worring, Jan C. van Gemert, Jan M. Geusebroek, and Arnold W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proc. 14th Annu. ACM Int. Conf. on Multimedia, Santa Barbara, CA, USA, MULTIMEDIA'06*, pages 421–430, 2006.

[SZU05] Ashok N Srivastava and Brett Zane-Ulman. Discovering recurring anomalies in text reports regarding complex space systems. In *Aerospace Conference*, pages 3853–3862. IEEE, 2005.

[TBTL08] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Semantic Annotation and Retrieval of Music and Sound Effects. *IEEE Audio, Speech, Language Process.*, 16(2):467–476, 2008.

[TKV08] G. Tsoumakas, I. Katakis, and I. Vlahavas. Effective and Efficient Multilabel Classification in Domains with Large Number of Labels. In *Proc. ECML/PKDD Workshop on Mining Multidimensional Data, Antwerp, Belgium, MMD'08*, pages 30–44, 2008.

[TXVV11] Grigorios Tsoumakas, Eleftherios S. Xioufis, Jozef Vilcek, and Ioannis Vlahavas. MULAN: A Java Library for Multi-Label Learning. *J. Mach. Learn. Res.*, 12:2411–2414, 2011.

[US02] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In *Advances in neural information processing systems*, pages 721–728, 2002.

[WSR06] Alicja Wieczorkowska, Piotr Synak, and Zbigniew Raś. Multi-Label Classification of Emotions in Music. In *Intelligent Information Processing and Web Mining*, volume 35 of *AISC*, chapter 30, pages 307–315. 2006.

[ZZ14] M. Zhang and Z. Zhou. A review on multi-label learning algorithms. *IEEE Trans. Knowl. Data Eng.*, 26(8):1819–1837, Aug 2014.