Package ‘farff’

May 11, 2021

Title A Faster 'ARFF' File Reader and Writer

Version 1.1.1

Description Reads and writes 'ARFF' files. 'ARFF' (Attribute-Relation File Format) files are like 'CSV' files, with a little bit of added meta information in a header and standardized NA values. They are quite often used for machine learning data sets and were introduced for the 'WEKA' machine learning 'Java' toolbox. See <https://waikato.github.io/weka-wiki/formats_and_processing/arff_stable/> for further info on 'ARFF' and for <http://www.cs.waikato.ac.nz/ml/weka/> for more info on 'WEKA'. 'farff' gets rid of the 'Java' dependency that 'RWeka' enforces, and it is at least a faster reader (for bigger files). It uses 'readr' as parser back-end for the data section of the 'ARFF' file. Consistency with 'RWeka' is tested on 'Github' and 'Travis CI' with hundreds of 'ARFF' files from 'OpenML'.

License BSD_2_clause + file LICENSE

URL https://github.com/mlr-org/farff

BugReports https://github.com/mlr-org/farff/issues

Imports BBmisc, checkmate (>= 1.8.0), readr (>= 1.0.0), stringi

Suggests OpenML, testthat

ByteCompile yes

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation yes

Author Marc Becker [cre, aut] (<https://orcid.org/0000-0002-8115-0400>), Bernd Bischl [aut] (<https://orcid.org/0000-0001-6002-6980>), Jakob Bossek [aut]

Maintainer Marc Becker <marcbecker@posteo.de>

Repository CRAN

Date/Publication 2021-05-10 23:40:05 UTC

R topics documented:

readARFF *Read ARFF file into data.frame.*

Description

Implementation of a fast ARFF parser that produces consistent results compared to the reference implementation in **RWeka**. The "DATA" section is read with read_delim.

Usage

```
readARFF(
  path,
  data.reader = "readr",
  tmp.file = tempfile(),
  convert.to.logicals = TRUE,
  show.info = TRUE,
  ...
)
```

Arguments

path
[character(1)]
Path to ARFF file with read access.

data.reader
[character(1)]
Package back-end to parse ARFF data section with. At the moment only readr is supported. Default is "readr".

tmp.file
[character(1)]
The ARFF file must be preprocessed a bit, before it can be fed to the data.reader. Path to TEMP output file, where this result is stored. The file is deleted on exit. Default is tempfile().

convert.to.logicals
[logical(1)]
Should factors with values T or F be converted to logicals? (RWeka does this by default). Default is TRUE.

show.info
[logical(1)]
Default is TRUE

...
[any] Further parameters passed to read_delim.

Details

ARFF parsers are already available in package RWeka in `read.arff` and package `foreign` in `read.arff`. The RWeka parser requires `Java` and `rJava`, a dependency which is notoriously hard to configure for users in R. It is also quite slow. The parser in foreign in written in pure R, slow and not fully consistent with the reference implementation in `RWeka`.

Value

`data.frame` .

Note

- Integer feature columns in ARFF files are parsed as numeric columns into R.
- Sparse ARFF format is currently unsupported. The function will produce an informative error message in that case.
- ARFF attributes of type “relational”, e.g., for multi-instance data, are currently not supported.

Examples

```
path = tempfile()
writeARFF(iris, path = path)
d = readARFF(path)
```

`writeARFF` *Write ARFF data.frame to ARFF file.*

Description

Internally uses `write.table` and is therefore not much faster than RWeka’s `write.arff`. Moreover, for large data (> 1e6 rows) the date frame is written out in chunks of 1e6 lines to speed up the write process.

Usage

```
writeARFF(
  x,
  path,
  overwrite = FALSE,
  chunk.size = 1e+06,
  relation = deparse(substitute(x))
)
```

Arguments

`x`
: `[data.frame]`
Data to write to disk.

`path`
: `[character(1)]`
Path to ARFF file with write access. Existing files will not be overwritten unless `overwrite` is `TRUE`.

`overwrite`
: `[logical(1)]`
Should `path` be overwritten if it already exists? Default is `FALSE`.

`chunk.size`
: `[integer(1)]`
Large datesets are split before writing out to file into chunks of size `chunk.size`. Default is `1e6`.

`relation`
: `[character(1)]`
Name of the relation in the ARFF file. Default is to guess it from the object name.

Value

Nothing.

Note

Logical columns in R are converted to categorical attributes in ARFF with levels “TRUE” and “FALSE”.

Examples

```
# see readARFF
```

Index