The Benchmark Data Library Project: ein Metadaten-Archiv für Simulationsdaten

Rainer Dangl und Friedrich Leisch

Institut für Angewandte Statistik & EDV
Universität für Bodenkultur Wien

Kolloquium der Wiener Biometrischen Sektion der ROeS

Introduction

New methods in model validation in unsupervised learning are commonly tested on artificial data

Usual approach

- develop new test data from scratch $\rightarrow$ is this always necessary?

Alternative approach

- search for suitable datasets in previous studies
- advantage $\rightarrow$ compare performance of methods directly
- if no suitable dataset is available develop own setup

$\rightarrow$ where to conveniently get artificial data?

Existing Repositories

Problems

- targeted toward supervised learning
- real world data
- no clearly structured metadata information

Requirements for a Benchmarking Repository

- only for artificial data
- clearly structured metadata that is the same for all datasets
- data sets are summarized in experimental setups
- 'peer-reviewed data sets'
- three data types: metric/functional/ordinal
- clear documentation of source of data sets
- visualization of data sets preferable
- users can easily contribute to the library

The Benchmark Data Library

Combines the aforementioned requirements into an easy to use combination of an R package and a web application based on the `shiny` package for R.

- `shiny` enables reactive programming paradigms $\rightarrow$ dynamic UI generation possible
- there is no huge storage required - no actual data is stored, just the metadata
- convenient way of exploring available data sets in the library
- if a new setup is is uploaded it is immediately available for download
- users who download create their data set on the client - no server-side lags due to heavy load

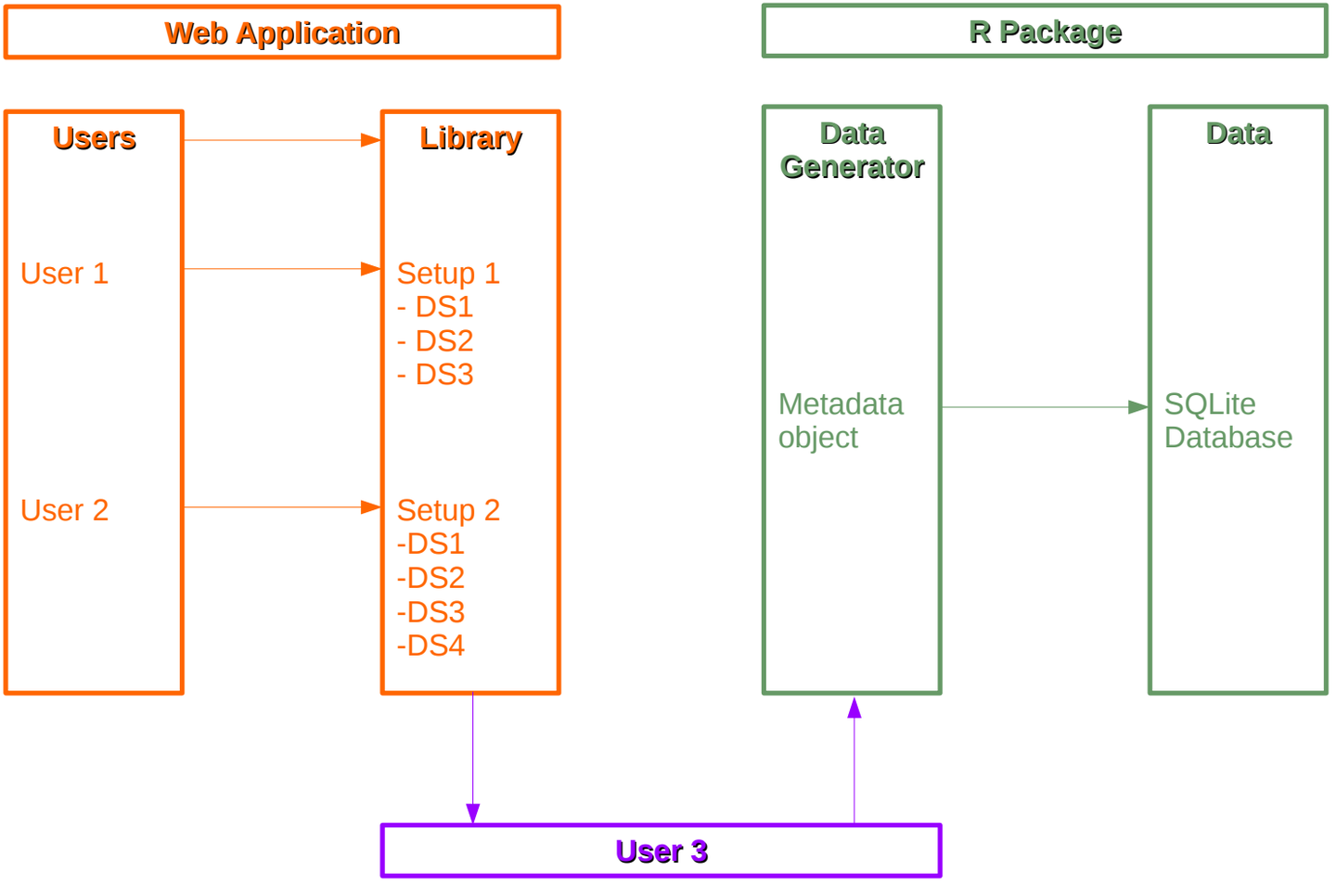
Reactive expressions in shiny

```
shinyServer(function(input, output) {
  output$result <- renderText({
    if(input$number == 1) return("One!")
    else return("Something else!")
  })
})
```

```
<input id="number" class="shiny-bound-input" type="number"></input>
<div id="result" class="shiny-html-output"></div>
```

The reactive function automatically updates the content of variable `result` depending on the value of `number`

The Benchmark Data Library

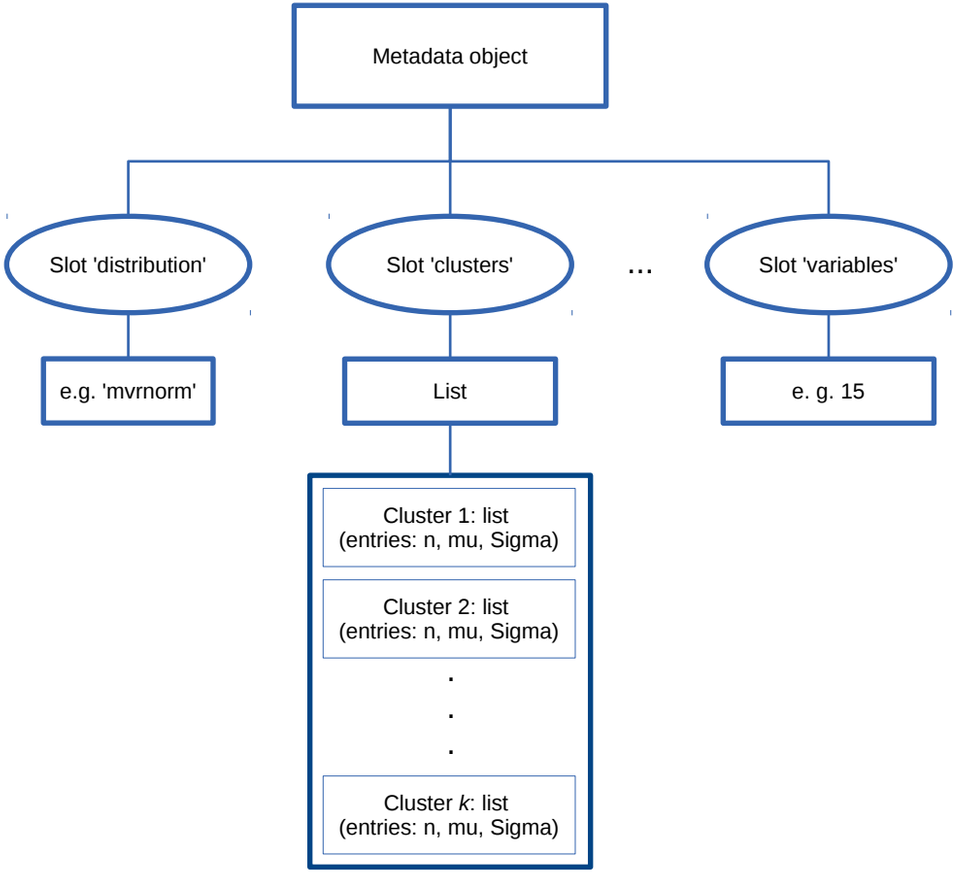


Metadata objects

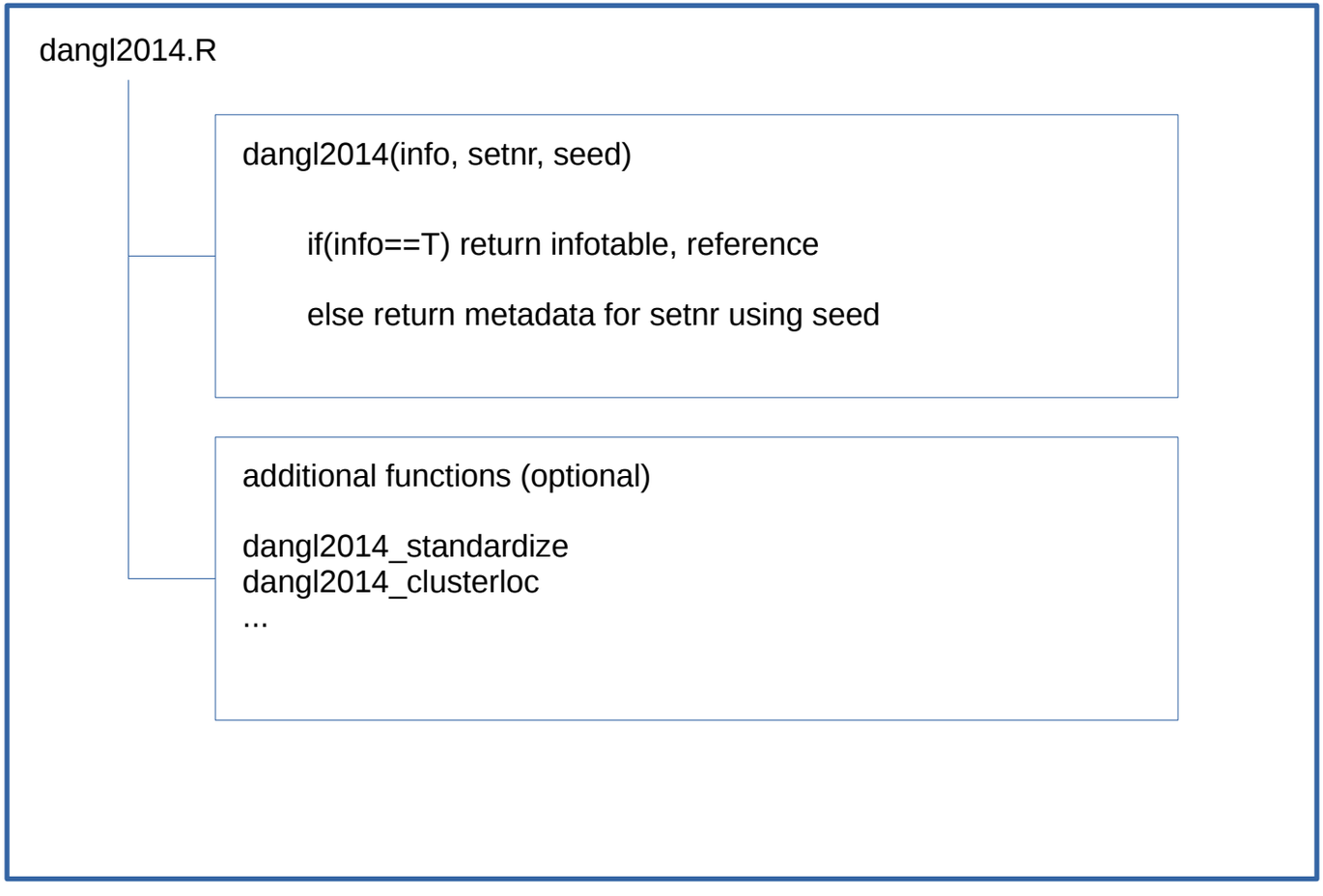
- each dataset in a setup is represented by a metadata object
- S4 objects that are processed by the data generator
- three types available: metric/functional/ordinal
- flexible enough to enable any data set but strict enough to provide a universal frame for all data sets
- the data generator only acts as an assembler that puts together the data set based on the metadata object

Example: it depends on the content of the slot 'distribution'(which data generating function) how the slot 'clusters' looks like - arguments must match $\rightarrow$ `mvrnorm` needs at least `n`, `mu` and `Sigma`, these arguments must be there for each cluster in the 'clusters' slot

Metadata objects



Input file


dangl2014.R
dangl2014(info, setnr, seed)
if(info==T) return infotable, reference
else return metadata for setnr using seed
additional functions (optional)
dangl2014_standardize
dangl2014_clusterloc
...

Input file

```
require(MASS)

dangl2014 <- function(setnr = NULL, seed = NULL, info = FALSE){

  tab <- data.frame(n = c(50, 40), k = c(2,2), shape = c("spherical", "spherical"))
  ref <- "Dangl R. (2014) A small simulation study. Journal of Simple Datasets 10(2), 1-10"
  if(info == T) return(list(summary = inf, reference = ref))

  if(setnr == 1) {
    return(new("metadata.metric",
               clusters = list(c1 = list(n = 250, mu = c(3,4), Sigma=diag(1,2)),
                               c2 = list(n = 250, mu = c(1,2), Sigma=diag(1,2))),
               distribution = "mvrnorm", seed = seed, variables = 2, total_n = 500, k = 2))
  }
  if(setnr == 2){
    return(new("metadata.metric",
               clusters = list(c1 = list(n = 200, mu = c(0,2), Sigma=diag(1,2)),
                               c2 = list(n = 200, mu = c(-1,-2), Sigma=diag(1,2))),
               distribution = "mvrnorm", seed = seed, variables = 2, total_n = 400, k = 2))
  }
}
```

Outlook

In the near future, the app will be used within the group $\rightarrow$ thorough testing before it can go public

Features still to be implemented:

- functional data implementation not satisfactory yet
- documentation
- other additional features/data types
- ...