Linkspotter: an R package for correlations analysis and visualization

Alassane Samba

"Les Sixièmes Rencontres R", Anglet, June 30th, 2017

Schedule

- Motivation
- Description
- Getting started
- Demonstration

Motivation

Dataset variables relationships exploration

An important phase to understand the data before any modelization.

Prior methods

- correlation matrix or static graph
- several packages with limited coefficients

Linkspotter

- interactive customizable graph plotting
- variable clustering
- user interface for interactive customization and an easy comprehensive analysis
- several coefficients, including 'Maximal Normalized Mutual Information' that is suitable for all types of links (quantitative-quantitative, quantitative-qualitative and qualitative-qualitative)

Linkspotter Description: Calculation Aspects (1/2)

- Perform a supervised discretization of one or a couple of quantitative variables using a supervized discretization method
- Compute a table containing all the bivariate correlations of the dataset using several link coefficients
- Extract the correlation matrices from the table of bivariate correlations
- Transform a correlation matrix into correlation couples and vice versa
- Perform a clustering of the variables using an unsupervised learning method on a correlation matrix

Linkspotter Description: Calculation Aspects (2/2)

Available correlation coefficients

	measured type of relationship
Pearson'r	quantitative-quantitative, linear
Spearman's rho, Kendall's tau	quantitative-quantitative, monotonic
Distance correlation, MIC	quantitative-quantitative, several types
MaxNMI	quanti-quanti, quali-quali,quanti-quali, several types

Linkspotter Description: Visualization Aspects

- Plot a customized graph: the nodes reprensent the variables and the edges represent the bivariate links.
- Generate a user interface
  - interactively customize the graph
  - visualize the distribution of each variable using an histogram, barplot, etc.
  - visualize the link between two variables using scatterplot, boxplot, etc.
  - show the correlations matrices
  - show the variable clustering result

Getting Started

Installing and loading the package

The package is for now available from its Github repository.

```
devtools::install_github("sambaala/linkspotter")
```

```
library(linkspotter)
```

The following examples are carried out on "iris" and "mtcars" mdata.

Calculate the MaxNMI between two variables

```
maxNMI(iris$Petal.Width,iris$Petal.Length)
```

```
## [1] 0.8351786
```

```
maxNMI(iris$Sepal.Length,iris$Petal.Length)
```

```
## [1] 0.6992338
```

```
maxNMI(iris$Sepal.Length,iris$Species)
```

```
## [1] 0.4873895
```

Calculate all of the link coefficients for all of the variable couples

```
corCouples<-multiBivariateCorrelation(iris)
print(corCouples, digits = 2, row.names = F)
```

```
##  id           X1           X2 typeOfCouple pearson spearman kendall  mic MaxNMI correlationType
##   1 Sepal.Length  Sepal.Width      num.num   -0.12    -0.17  -0.077 0.28   0.20        negative
##   2 Sepal.Length Petal.Length      num.num    0.87     0.88   0.719 0.77   0.70        positive
##   3 Sepal.Length  Petal.Width      num.num    0.82     0.83   0.655 0.67   0.61        positive
##   4 Sepal.Length      Species     num.fact      NA       NA      NA   NA   0.49         nominal
##   5  Sepal.Width Petal.Length      num.num   -0.43    -0.31  -0.186 0.44   0.38        negative
##   6  Sepal.Width  Petal.Width      num.num   -0.37    -0.29  -0.157 0.44   0.39        negative
##   7  Sepal.Width      Species     num.fact      NA       NA      NA   NA   0.26         nominal
##   8 Petal.Length  Petal.Width      num.num    0.96     0.94   0.807 0.92   0.84        positive
##   9 Petal.Length      Species     num.fact      NA       NA      NA   NA   0.87         nominal
##  10  Petal.Width      Species     num.fact      NA       NA      NA   NA   0.89         nominal
```

Extract the Pearson correlation matrix from the correlation dataframe

```
corMatrixPearson<-corCouplesToMatrix(x1_x2_val = corCouples[,c('X1','X2',"pearson")])
print(corMatrixPearson, digits = 2)
```

```
##              Petal.Length Petal.Width Sepal.Length Sepal.Width Species
## Petal.Length         1.00        0.96         0.87       -0.43      NA
## Petal.Width          0.96        1.00         0.82       -0.37      NA
## Sepal.Length         0.87        0.82         1.00       -0.12      NA
## Sepal.Width         -0.43       -0.37        -0.12        1.00      NA
## Species               NA          NA           NA          NA       1
```

Extract the MaxNMI correlation matrix from the correlation dataframe

```
corMatrixMaxNMI<-corCouplesToMatrix(x1_x2_val = corCouples[,c('X1','X2',"MaxNMI")])
print(corMatrixMaxNMI, digits = 2)
```

```
##              Petal.Length Petal.Width Sepal.Length Sepal.Width Species
## Petal.Length         1.00        0.84         0.70        0.38    0.87
## Petal.Width          0.84        1.00         0.61        0.39    0.89
## Sepal.Length         0.70        0.61         1.00        0.20    0.49
## Sepal.Width          0.38        0.39         0.20        1.00    0.26
## Species              0.87        0.89         0.49        0.26    1.00
```

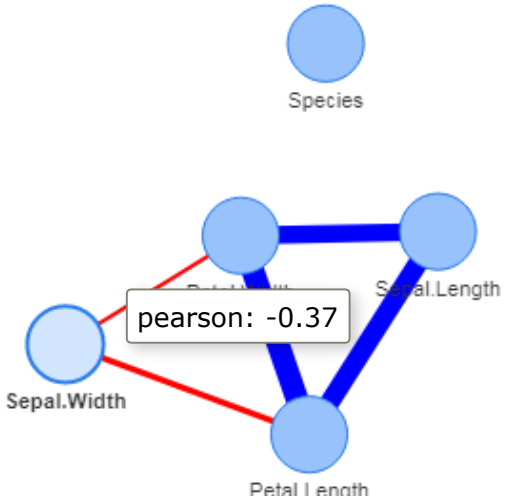
Clustering of variables using a correlation matrix

```
cl=clusterVariables(correlationMatrix = corMatrixMaxNMI)
print(cl, row.names = F)
```

```
##           var group
##  Petal.Length     1
##   Petal.Width     2
##  Sepal.Length     3
##   Sepal.Width     4
##       Species     2
```

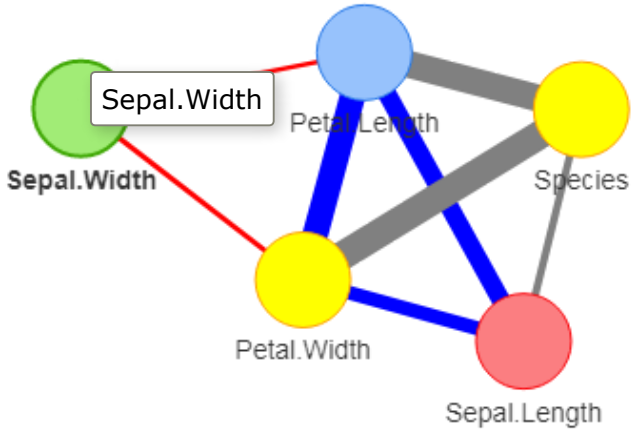
Visualize the graph using Pearson's r

```
linkspotterGraph(corDF = corCouples, corMethod = "pearson",
                 minCor = 0.3, colorEdgesByCorDirection=T,
                 smoothEdges = FALSE, dynamicNodes = FALSE)
```



Visualize the graph using MaxNMI

```
linkspotterGraph(corDF = corCouples, corMethod = "MaxNMI",
                 minCor = 0.3, colorEdgesByCorDirection=T,
                 smoothEdges = FALSE, dynamicNodes = TRUE,
                 variablesClustering = cl)
```



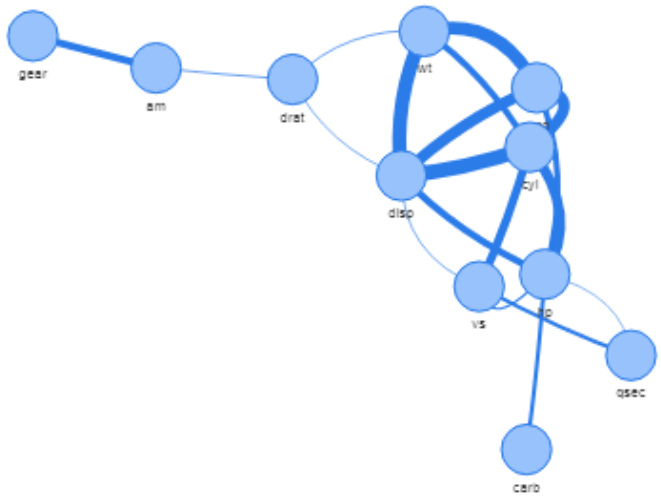
Visualize the graph of any other correlation/distance matrix (1/3)

```
corMatrix=cor(mtcars,method ='pearson')
print(corMatrix, digits = 2)
```

```
##        mpg   cyl  disp    hp   drat    wt   qsec    vs     am  gear   carb
## mpg   1.00 -0.85 -0.85 -0.78  0.681 -0.87  0.419  0.66  0.600  0.48 -0.551
## cyl  -0.85  1.00  0.90  0.83 -0.700  0.78 -0.591 -0.81 -0.523 -0.49  0.527
## disp -0.85  0.90  1.00  0.79 -0.710  0.89 -0.434 -0.71 -0.591 -0.56  0.395
## hp   -0.78  0.83  0.79  1.00 -0.449  0.66 -0.708 -0.72 -0.243 -0.13  0.750
## drat  0.68 -0.70 -0.71 -0.45  1.000 -0.71  0.091  0.44  0.713  0.70 -0.091
## wt   -0.87  0.78  0.89  0.66 -0.712  1.00 -0.175 -0.55 -0.692 -0.58  0.428
## qsec  0.42 -0.59 -0.43 -0.71  0.091 -0.17  1.000  0.74 -0.230 -0.21 -0.656
## vs    0.66 -0.81 -0.71 -0.72  0.440 -0.55  0.745  1.00  0.168  0.21 -0.570
## am    0.60 -0.52 -0.59 -0.24  0.713 -0.69 -0.230  0.17  1.000  0.79  0.058
## gear  0.48 -0.49 -0.56 -0.13  0.700 -0.58 -0.213  0.21  0.794  1.00  0.274
## carb -0.55  0.53  0.39  0.75 -0.091  0.43 -0.656 -0.57  0.058  0.27  1.000
```

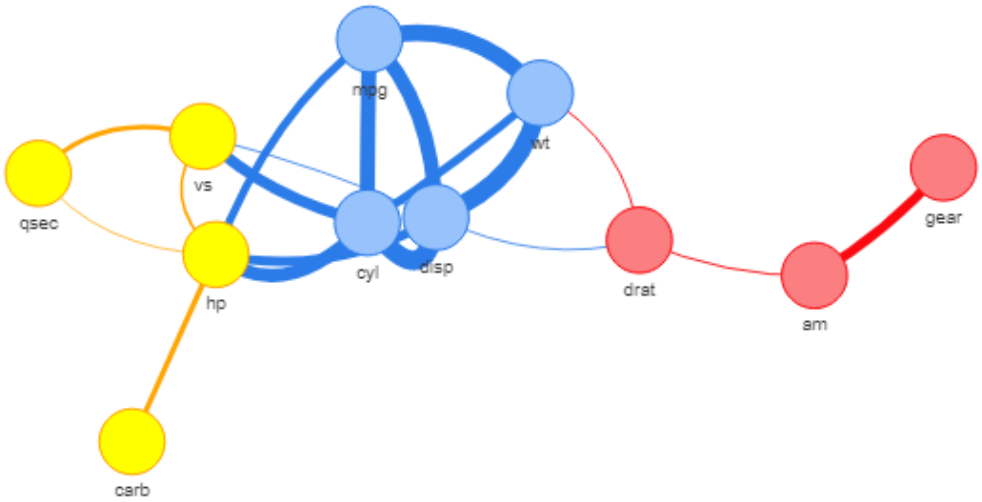
Visualize the graph of any other correlation/distance matrix (2/3)

```
linkspotterGraphOnMatrix(corMatrix, minCor = 0.7)
```



Visualize the graph of any other correlation/distance matrix (3/3)

```
linkspotterGraphOnMatrix(corMatrix, cluster = T, nbCluster = 3,
                         minCor = 0.7)
```



Launch the costumizable user interface

```
linkspotterUI(dataset = iris, corDF = corCouples,
              variablesClustering = cl,
              appTitle = "Linkspotter iris example")
```

Demo

Additional features (1/4)

Complete Linkspotter computation:

```
lsiris<-linkspotterComplete(iris)
```

```
## [1] "Number of variables: 5"
## [1] "Number of couples: 10"
## [1] "Number of observations: 150"
## [1] "Start time: 2017-06-01 14:04:04"
## [1] "Correlations computation finished: 2017-06-01 14:04:05"
## [1] "Clustering computation finished: 2017-06-01 14:04:05"
## [1] "Total Computation time: 1.227 secs"
```

Complete Linkspotter computation from an external file:

```
lsiris<-linkspotterOnFile("iris.csv")
```

Additional features (2/4)

Results:

```
summary(lsiris)
```

```
##                     Length Class        Mode
## computationTime      1     -none-       character
## run_it               5     shiny.appobj list
## dataset              5     data.frame   list
## corDF               10     data.frame   list
## corMatrices          5     -none-       list
## corGroups            2     data.frame   list
## clusteringCorMethod  1     -none-       character
## defaultMinCor        1     -none-       numeric
## defaultCorMethod     1     -none-       character
## corMethods           5     -none-       character
```

Additional features (3/4)

The user interface can be launched this way:

```
lsiris$run_it
```

Demo

Additional features (4/4)

Generate a ready-for-deployment shiny app folder

```
createShinyAppFolder(linkspotterShinyAppObject = lsiris$run_it,
                                    folderName = "myLinkspotterApp")
```

Next Steps

- Add the use of C++/parallel computing for big datasets
- Offer more possibilities for managing missing values
- CRAN submission …

Any help is welcome and feel free to contribute via Github

Links:

- https://github.com/sambaala/linkspotter
- http://linkspotter.sigmant.net

Thanks for your attention