Package 'GraphFactor'

October 17, 2016

Type Package

Title Network Topology of Intravariable Clusters with Intervariable Links

Version 1.1

Date 2016-10-10

Author Matthew C. Bascom

Maintainer Matthew C. Bascom <matt@graphfactor.com>

Description A Network Implementation of Fuzzy Sets: Build Network Objects from Multivariate Flat Files. For more information on fuzzy sets, refer to: Zadeh, L.A. (1964) <DOI:10.1016/S0019-9958(65)90241-X>.

Imports igraph

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2016-10-17 11:09:21

R topics documented:

`GraphFactor-package` *Network Topology of Intravariable Clusters with Intervariable Links*

Description

A Network Implementation of Fuzzy Sets: Build Network Objects from Multivariate Flat Files. For more information on fuzzy sets, refer to: Zadeh, L.A. (1964) <DOI:10.1016/S0019-9958(65)90241-X>.

Details

GraphFactor is for exploring the qualitative features of a flat file. In a sense, by clustering like values in each variable, the graphFactor function reduces the degrees of freedom of the data. An edge between two clusters indicates that a threshold criteria is met between two variables. Each node (each clustered set of indices in the variable) shares an intersection of indices in an adjacent node, as defined in the parameters by the useR. Nodes within the same variable are never adjacent: Network construction starts from a variable-wise bipartite graph.

Since smaller clusters may lack the criteria for connectedness to any other node in the graph, they won't show in the visualization. It is because of this that sometimes, though each variable receives a color assignment, an event variable's respective node may be absent. So in that sense, events that are statistical leverage points may be apparent by their absense! Further, an interesting artifact of the graphFactor function is that potentially influential observations may be evident because the point's cluster assignments are uncommon with respect to the graph at large: Nodes of an event may appear yet remain disjoint.

Does this tool add value to the analyst's toolbox? Time will tell. Feedback is appreciated. Thank you for using GraphFactor.

Author(s)

Matthew C. Bascom

Maintainer: Matthew C. Bascom <matt@graphfactor.com>

References

Zadeh, L.A. (1964) _Fuzzy Sets_ Electronics Research Laboratory, University of California, Berkeley, Report No. 64-44

Examples

```
# Subset the data.
# Ensure the data is of data.frame class.

library(GraphFactor)
library(igraph)
library(datasets)

state77 <- state.x77[,c(1,2,3,5,7)]
state77 <- data.frame(state77) # # Make sure it's data.frame!

seex <- c(4,5,6,7,8) # Data set indices to inspect with graphFactor.

graphFactor(dat = state77, datName = 'stateX77', showSequence = TRUE,
buildSVD = FALSE, plotToFile = FALSE, minimumEventThreshold = 3, seqDex = seex)
```

graphFactor *Network Topology of Intravariable Clusters with Intervariable Links*

Description

A Network Implementation of Fuzzy Sets: Build Network Objects from Multivariate Flat Files.

Usage

```
graphFactor(dat = dat, datName = NULL, howmanyvariables = dim(dat)[2L],
 howmanyclusters = 10, buildSVD = FALSE,
  catDirectoryPrefixName = "cat.expressions.dir.",
   catDirectorySuffixName = 1.1, minimumEventThreshold = 2,
  seed = 20150523, ubFlag = FALSE, maximumEventThreshold = 10,
                   YourNameHere = 'Your Name Here',
                  YourLocationHere = 'Your Location Here',
 plotToFile = FALSE, showSequence = FALSE, seqDex = NULL)
```

Arguments

dat	The data to model. The data must be a flat, numeric, data.frame class object.
datName	Name the data so it shows up on the network visualization. (character)
howmanyvariables	The number of variables to place in the model. (integer)
howmanyclusters	The number of clusters to construct in each variable. (integer)
buildSVD	Normalize the data using SVD. (logical)
catDirectoryPrefixName	Directory to place edgelist and kmeans cluster assignments. Default:'cat.expressions.dir.' (character)
catDirectorySuffixName	Default: 1.1 (numeric)
minimumEventThreshold	When placing edges between clusters, this is the minimum number of events shared across constituent variables of a multivariate event. Default: 2 (integer)
seed	Seed for replicating results. (integer)
ubFlag	Upper Bound Flag. Should maximumEventThreshold be considered when constructing the model? Default = FALSE (logical)
maximumEventThreshold	The maximum number of shared event indices between clusters across any two variables to generate an edge.
YourNameHere	Character. Used to personalize the network viz.
YourLocationHere	Character. Used to personalize the network viz.

plotToFile	The function will print the viz to file and won't print the visualization to the terminal. Default: FALSE (logical)
showSequence	Logical. If TRUE, generate a sequence of plots for the indices of the seqDex ("sequence index") parameter. Default value is FALSE.
seqDex	An event index vector used if showSequence is TRUE.

Details

GraphFactor is for exploring the qualitative features of a flat file. In a sense, by clustering like values in each variable, the graphFactor function reduces the degrees of freedom of the data. An edge between two clusters indicates that a threshold criteria is met between two variables. Each node (each clustered set of indices in the variable) shares an intersection of indices in an adjacent node, as defined in the parameters by the useR. Nodes within the same variable are never adjacent: Network construction starts from a variable-wise bipartite graph.

Since smaller clusters may lack the criteria for connectedness to any other node in the graph, they won't show in the visualization. It is because of this that sometimes, though each variable receives a color assignment, an event variable's respective node may be absent. So in that sense, events that are statistical leverage points may be apparent by their absense! Further, an interesting artifact of the graphFactor function is that potentially influential observations may be evident because the point's cluster assignments are uncommon with respect to the graph at large: Nodes of an event may appear yet remain disjoint.

Does this tool add value to the analyst's toolbox? Time will tell. Feedback is appreciated. Thank you for using GraphFactor.

Value

A list with components:

consecutive characters: Clustered event indices for a variable. "PC.clusters[[1]] <- list(a = c(19,27,31,44,48), ..." etc.

A file folder with files:

kmeans.cluster.subset.expressions.R: The cluster list.

edgelist.R: an edgelist

Both files are saved to the current working directory in a folder named when calling graphFactor.

Plots are saved to png file or printed to the on screen graphics device.

Author(s)

Matthew C. Bascom

References

Zadeh, L.A. (1964) _Fuzzy Sets_ Electronics Research Laboratory, University of California, Berkeley, Report No. 64-44

Examples

```
library(GraphFactor)
library(igraph)
library(datasets)

### Help files: ###

# # help(package = GraphFactor)
# # ?GraphFactor::GraphFactor
# # ?GraphFactor::graphFactor

data(USArrests)

graphFactor(dat = datasets::USArrests, datName = 'USArrests')
dev.new()
graphFactor(dat = datasets::USArrests, buildSVD = TRUE)
dev.new()
graphFactor(dat = datasets::USArrests, buildSVD = TRUE, minimumEventThreshold = 3)
graphFactor(dat = datasets::USArrests,
 datName = 'USArrests', buildSVD = FALSE, minimumEventThreshold = 3)
# Not very exciting so far...

# Show a sequence of events.

# Put one value in seqDex.
graphFactor(dat = datasets::USArrests, showSequence = TRUE, seqDex = 1)
# Put a few values in seqDex.
graphFactor(dat = datasets::USArrests, showSequence = TRUE, seqDex = c(1,2,3))
graphFactor(dat = datasets::USArrests, showSequence = TRUE, seqDex = c(1,2,3), buildSVD = TRUE)

seex <- c(4,5,6)
graphFactor(dat = datasets::USArrests, showSequence = TRUE, seqDex = seex,
 buildSVD = FALSE, datName = 'USArrests', plotToFile = FALSE,
minimumEventThreshold = 3)

# Try another data set.

data(state)

state77 <- state.x77[,c(1,2,3,5,7)]

head(state77)
class(state77) # matrix.

state77 <- data.frame(state77) # # Make sure it's data.frame!

graphFactor(dat = state77, datName = 'stateX77', showSequence = FALSE,
buildSVD = FALSE, plotToFile = FALSE, minimumEventThreshold = 2)

# Plot to file.
graphFactor(dat = state77, datName = 'stateX77', showSequence = TRUE,
buildSVD = FALSE, plotToFile = TRUE, minimumEventThreshold = 2, seqDex = c(1,2,3))
```

```
# Increase the minimum threshold to place an edge between any two nodes.

# Don't plot to file...
graphFactor(dat = state77, datName = 'stateX77', showSequence = TRUE,
buildSVD = FALSE, plotToFile = FALSE, minimumEventThreshold = 3, seqDex = seex)

# ...plot to file.
graphFactor(dat = state77, datName = 'stateX77', showSequence = TRUE,
buildSVD = FALSE, plotToFile = TRUE, minimumEventThreshold = 3, seqDex = seex)

#X# graphFactor(dat = state77, datName = 'stateX77', showSequence = T,
#X#  buildSVD = F, plotToFile = T, minimumEventThreshold = 2)
#X# #  Error. (no print...needs seqDex!)

graphFactor(dat = state77, datName = 'stateX77', showSequence = TRUE,
buildSVD = FALSE, plotToFile = TRUE, minimumEventThreshold = 2, seqDex = seex)

# Pick a different subset of variables to inspect.
state76 <- state.x77[,c(1,2,3,5)]
state76 <- data.frame(state76)

#X# graphFactor(dat = state76, datName = 'stateX77', showSequence = F,
#X#  buildSVD = F, plotToFile = TRUE, minimumEventThreshold = 2)
#X# # 'plotToFile' failed because showSequence is set to FALSE!

# Note the presence of the sequence vector, seqDex, when showSequence is TRUE.
graphFactor(dat = state76, datName = 'stateX77', showSequence = TRUE,
buildSVD = FALSE, plotToFile = TRUE, minimumEventThreshold = 2,
seqDex = c(1,2,3))

state75 <- data.frame(state.x77[,c(1,2,3)])
class(state75) # data.frame.
graphFactor(dat = state75, datName = 'stateX77', showSequence = TRUE,
buildSVD = FALSE, plotToFile = FALSE, minimumEventThreshold = 2,
seqDex = c(1,2,3))

graphFactor(dat = state75, datName = 'stateX77', showSequence = TRUE,
buildSVD = FALSE, plotToFile = TRUE, minimumEventThreshold = 2,
seqDex = c(10,11,12))
```

Index