Package 'ganDataModel'

July 16, 2022

Type Package

Title Create a Hierarchical, Categorical Data Model for a Data Source

Version 1.0.2

Date 2022-07-14

Author Werner Mueller

Maintainer Werner Mueller <werner.mueller5@chello.at>

Description Neural networks are applied to create a density value function which approximates density values for a data source. The trained neural network is analysed for different levels. For each level subspaces with density values above a level are determined. The obtained set of subspaces categorizes the data source hierarchically. A prerequisite is the definition of a data source, the generation of generative data and the calculation of density values. These tasks are executed using package 'ganGenerativeData' <https://cran.r-project.org/package=ganGenerativeData>.

License GPL (>= 2)

Imports Rcpp (>= 1.0.3), tensorflow (>= 2.0.0)

LinkingTo Rcpp

RoxygenNote 7.0.2

SystemRequirements TensorFlow (https://www.tensorflow.org)

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-07-16 08:20:02 UTC

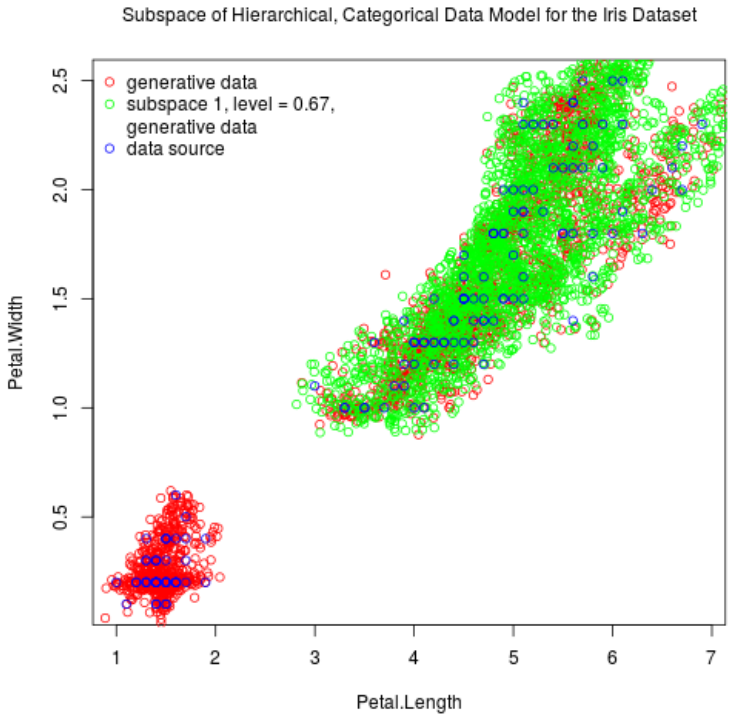
R topics documented:

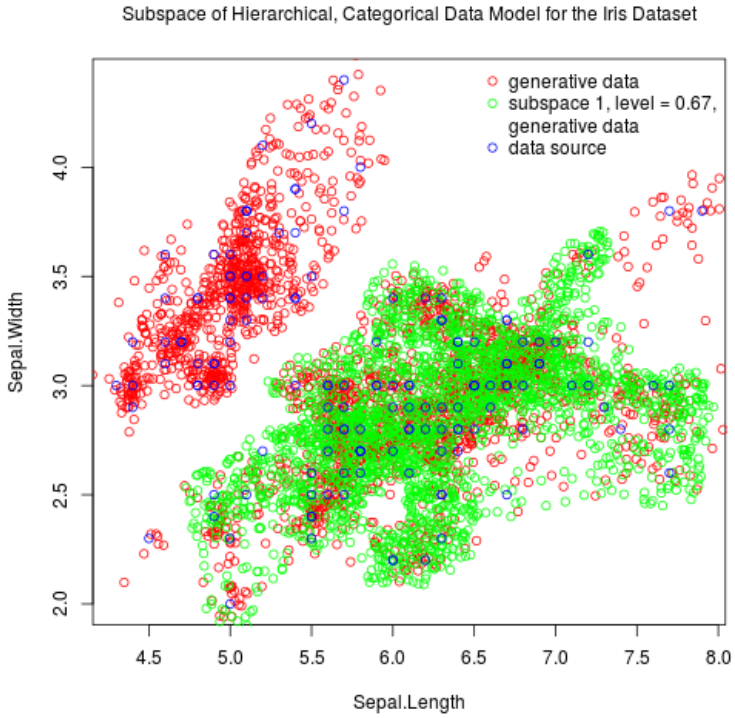
`ganDataModel-package` *Create a Hierarchical, Categorical Data Model for a Data Source*

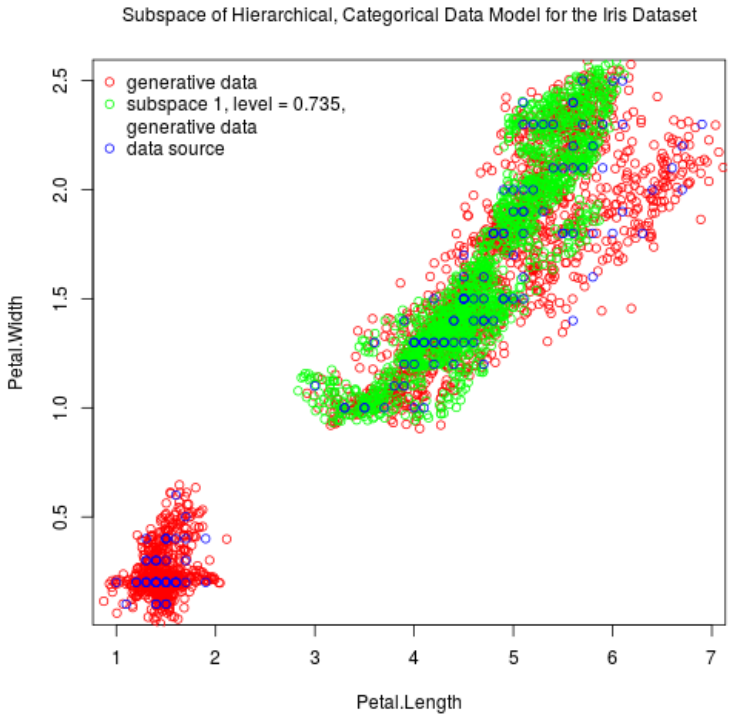
Description

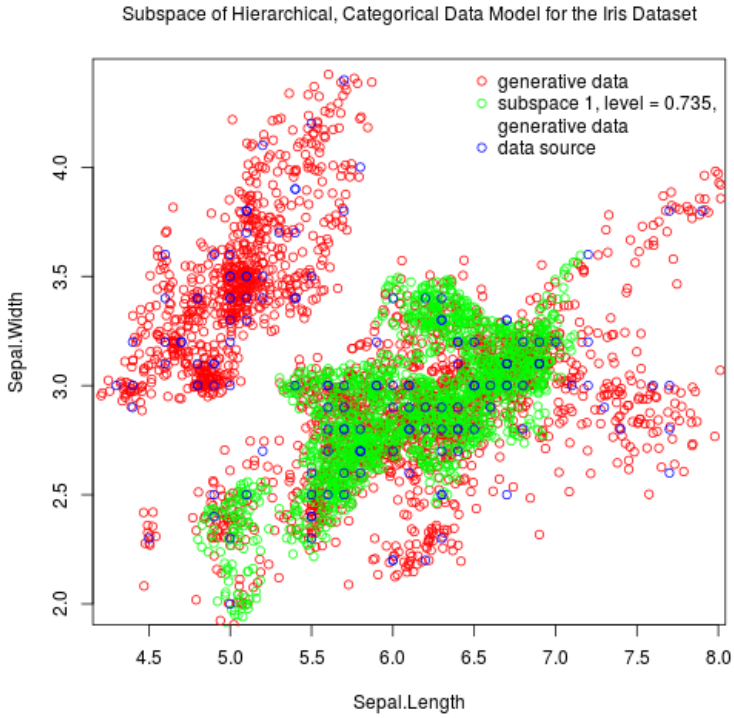
Neural networks are applied to create a density value function which approximates density values for a data source. The trained neural network is analysed for different levels. For each level subspaces with density values above a level are determined. The obtained set of subspaces categorizes the data source hierarchically. A prerequisite is the definition of a data source, the generation of generative data and the calculation of density values. These tasks are executed using package 'ganGenerativeData' <https://cran.r-project.org/package=ganGenerativeData>.

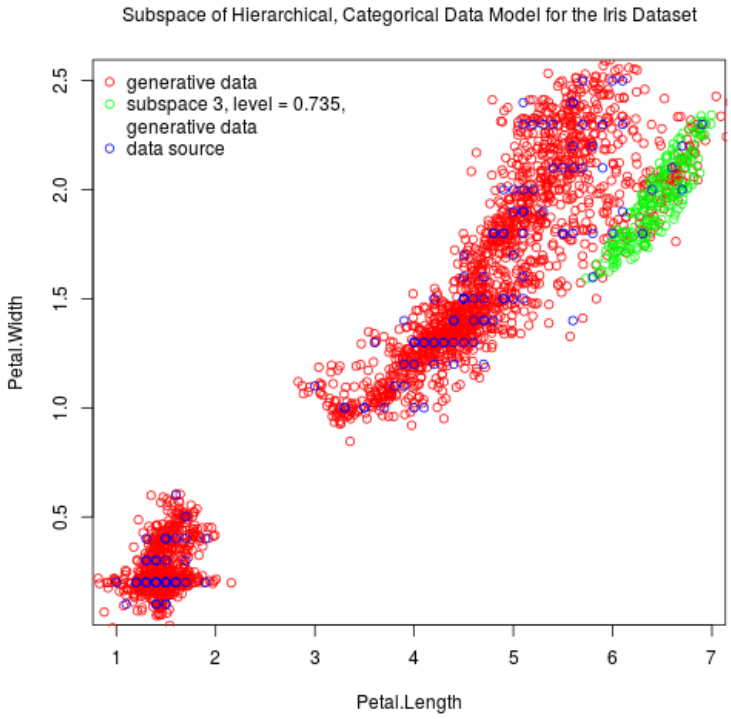
The inserted images show two-dimensional projections of generative data for the iris dataset for selected subspaces. Generative data assigned to subspaces is displayed in green colour. Subspace 1 for lower level 0.67 contains subspaces 1, 3 and 4 for higher level 0.735.

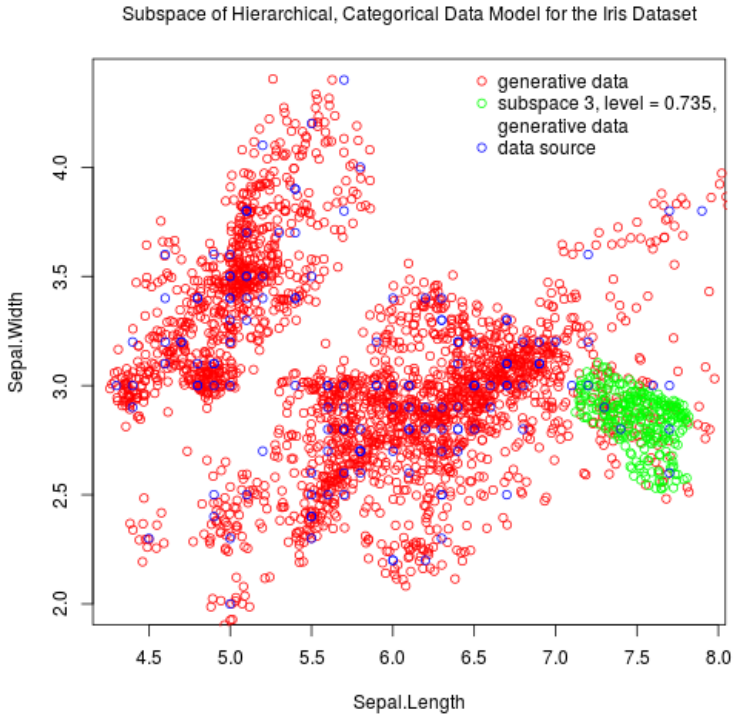

Subspace of Hierarchical, Categorical Data Model for the Iris Dataset
generative data
subspace 1, level = 0.67,
generative data
data source
Petal.Width
Petal.Length
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7

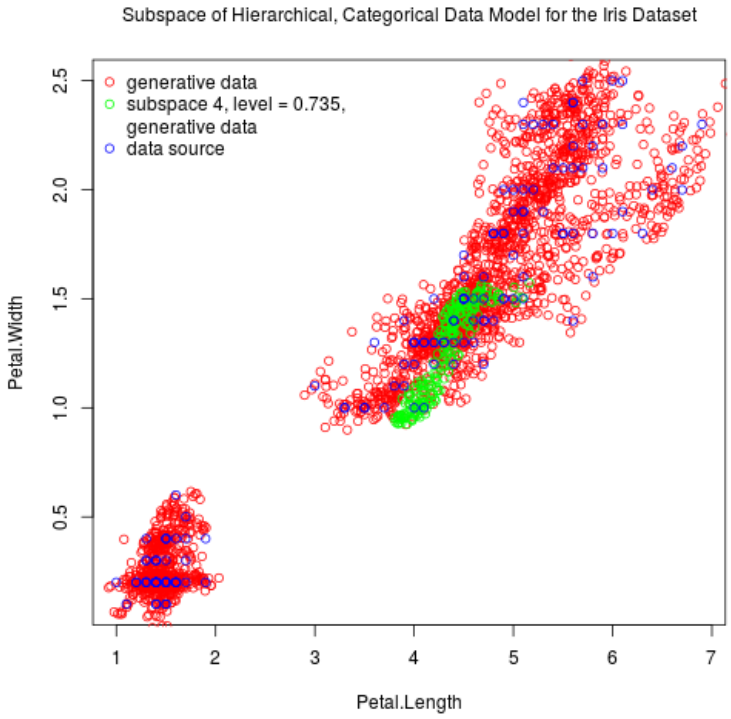

Subspace of Hierarchical, Categorical Data Model for the Iris Dataset
generative data
subspace 1, level = 0.67,
generative data
data source
Sepal.Width
Sepal.Length
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

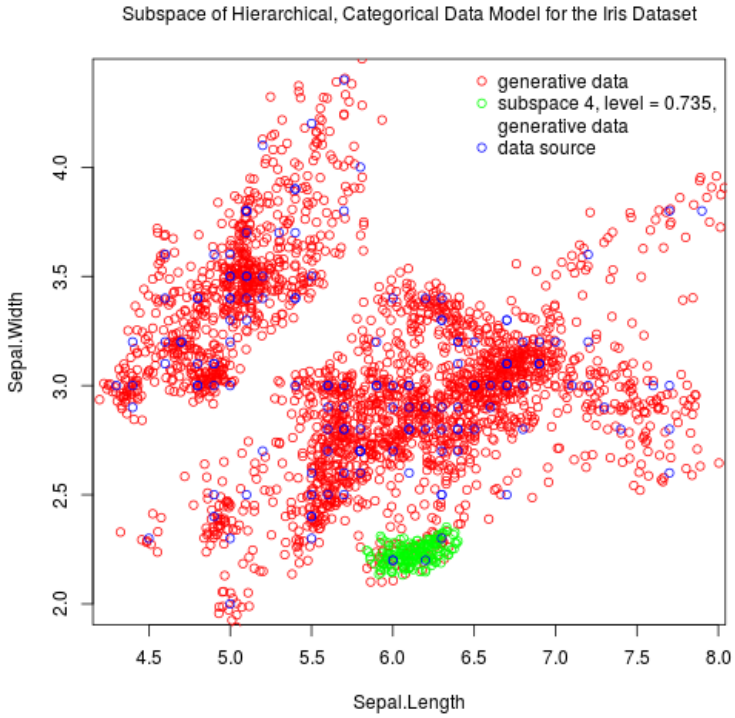

Subspace of Hierarchical, Categorical Data Model for the Iris Dataset
generative data
subspace 1, level = 0.735,
generative data
data source
Petal.Width
Petal.Length
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7


Subspace of Hierarchical, Categorical Data Model for the Iris Dataset
generative data
subspace 1, level = 0.735,
generative data
data source
Sepal.Width
Sepal.Length
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


Subspace of Hierarchical, Categorical Data Model for the Iris Dataset
generative data
subspace 3, level = 0.735,
generative data
data source
Petal.Width
Petal.Length
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7


Subspace of Hierarchical, Categorical Data Model for the Iris Dataset
generative data
subspace 3, level = 0.735,
generative data
data source
Sepal.Width
Sepal.Length
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


Subspace of Hierarchical, Categorical Data Model for the Iris Dataset
generative data
subspace 4, level = 0.735,
generative data
data source
Petal.Width
Petal.Length
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7


Subspace of Hierarchical, Categorical Data Model for the Iris Dataset
generative data
subspace 4, level = 0.735,
generative data
data source
Sepal.Width
Sepal.Length
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

Details

The API includes main functions `dmTrain()` and `dmBuildSubspaces()`. `dmTrain()` trains a neural network that approximates density values for a data source. `dmBuildSubspaces()` analyses the trained neural network for a level and determines subspaces with density values above the level. The API is used as follows:

1. Prerequisite for creating a hierarchical, categorical data model: Create a data source, generate generative data and calculate density values using package ganGenerativeData

`dsCreateWithDataFrame()` Create a data source with passed data frame.

`dsDeactivateColumns()` Deactivate columns of a data source in order to exclude them in generation of generative data. In current version only columns with values of type double or float can be used in generation of generative data. All columns with values of other type have to be deactivated.

`dsWrite()` Write created data source including settings of active columns to a file in binary format.

`gdGenerate()` Read a data source from a file, generate generative data for the data source in iterative training steps and write generated data to a file in binary format.

`gdCalculateDensityValues()` Read generative data from a file, calculate density values and write generative data with assigned density values to original file.

2. Create a hierarchical, categorical data model

`dmTrain()` Read a data source and generative data from files, train a neural network which approximates density values for a data source in iterative training steps, create a data model with the trained neural network and write it to a file in binary format.

`dmBuildSubspaces()` Read a data model and generative data from files, analyse the trained neural network in the data model for a level, determine subspaces with density values above the level, add obtained subspaces with assigned level to the data model and write it to original file.

`dmRemoveSubspaces()` Remove subspaces from a data model for a level.

`dmRead()` Read a data model and generative data from files.

`dmGetLevels()` Get levels for subspaces in a data model.

`dmGetNumberOfSubspaces()` Get number of subspaces in a data model for a level.

`dmGetAssignedSubspaces()` Get subspaces in a data model to which a data record is assigned

`dmPlotGenerativeDataParameters()` Specify plot parameters for generative data.

`dmPlotSubspaceParameters()` Soecify plot parameters for subspaces.

`dmPlotEvaluateDataSourceParameters()` Specify plot parameters for a data source.

`dmPlotSubspaces()` Create an image file containing two-dimensional projections of generative data, generative data assigned to subspaces and optionally an evaluated data source.

`dmEvaluateDataRecord()` Calculate density value for a data record by evaluating trained neural network in a data model.

`dmPlotEvaluate()` Create an image file containing two-dimensional projections of generative data and an evaluated data source.

`dmReset()` Reset API.

Author(s)

Werner Mueller

Maintainer: Werner Mueller <werner.mueller5@chello.at>

References

Package 'ganGenerativeData' <https://cran.r-project.org/package=ganGenerativeData>

Examples

```
# Packages ganGenerativeData and ganDataModel import package tensorflow
# which provides an interface to machine learning framework TensorFlow.
# It is used for training of neural networks and must be installed by the user.
## Not run:
library(tensorflow)
install_tensorflow()
## End(Not run)

# 1. Prerequisite for creating a hierarchical, categorical data model for the iris dataset:
# Create a data source, generate generative data and calculate density values for the iris dataset.

# Load library
## Not run:
library(ganGenerativeData)
## End(Not run)

# Create a data source with passed iris data frame.
## Not run:
dsCreateWithDataFrame(iris)
## End(Not run)
```

```
# Deactivate the column with name Species and index 5 in order to exclude it in
# generation of generative data.
## Not run:
dsDeactivateColumns(c(5))
## End(Not run)

# Write the data source including settings of active columns to file "ds.bin" in binary format.
## Not run:
dsWrite("ds.bin")
## End(Not run)

# Read data source from file "ds.bin",
# generate generative data in iterative training steps (in practise 50000 iterations are used))
# and write generated generative data to file "gd.bin".
## Not run:
gdGenerate("ds.bin", "gd.bin", 2500, 0.95, c(1, 2))
## End(Not run)

# Read generative data from file "gd.bin", calculate density values and
# write generative data with density values to original file.
## Not run:
gdCalculateDensityValues("gd.bin")
## End(Not run)

# 2. Create a hierarchical, categorical data model for the iris data set

# Load  library
## Not run:
library(ganDataModel)
## End(Not run)

# Read a data source and generative data from files "ds.bin" and "gd.bin",
# train a neural network which approximates density values for a data source
# in iterative training steps (in practise 250000 iterations are used),
# create a data model with trained neural network
# and write it to a file "dm.bin" in binary format.
## Not run:
dmTrain("dm.bin", "ds.bin", "gd.bin", 10000)
## End(Not run)

# Read a data model and generative data from files "dm.bin" and "gd.bin",
# build subspaces for level 0.73,
# add obtained subspaces with assigned level to data model
# and write it to original file.
## Not run:
dmBuildSubspaces("dm.bin", 0.73, "gd.bin")
## End(Not run)

# Read a data model and generative data from files "dm.bin" and "gd,bin".
# Read in data is accessed in function dmPlotSubspaces.
## Not run:
dmRead("dm.bin", "gd.bin")
```

```
## End(Not run)

# Create an image showing a two-dimensional projection of generative data
# for column indices 3, 4 for subspace 1 and write it to file "s1d34.png"
## Not run:
dmPlotSubspaces(0.73,
  c(1),
  "s1d34.png",
  "Subspace of Hierarchical, Categorical Data Model for the Iris Dataset",
  c(3, 4),
  dmPlotGenerativeDataParameters(10),
  dmPlotSubspaceParameters(100),
  "ds.bin",
  dmPlotEvaluateDataSourceParameters()
)
## End(Not run)

# Create an image showing a two-dimensional projection of generative data
# for column indices 3, 4 for subspace 2 and write it to file "s2d34.png"
## Not run:
dmPlotSubspaces(0.73,
  c(2),
  "s2d34.png",
  "Subspace of Hierarchical, Categorical Data Model for the Iris Dataset",
  c(3, 4),
  dmPlotGenerativeDataParameters(10),
  dmPlotSubspaceParameters(100),
  "ds.bin",
  dmPlotEvaluateDataSourceParameters()
)
## End(Not run)
```

`dmBuildSubspaces` *Build subspaces for a data model for a level*

Description

Read a data model and generative data from files, analyse the trained neural network in a data model for a level, determine subspaces with density values above the level, add obtained subspaces with assigned level to the data model and write it to original file.

Usage

```
dmBuildSubspaces(dataModelFileName, level, generativeDataFileName)
```

Arguments

`dataModelFileName`	Name of data model file
`level`	Level

generativeDataFileName
: Name of generative data file

Value

None

Examples

```
## Not run:
dmBuildSubspaces("dm.bin", 0.73, "gd.bin")
## End(Not run)
```

---

dmCalculateDensityValue

Calculate density value for a data record

Description

Calculate density value for a data record by evaluating the trained neural network in a data model.

Usage

```
dmCalculateDensityValue(dataRecord)
```

Arguments

dataRecord	List containing a data record

Value

Normalised density value

Examples

```
## Not run:
dmRead("dm.bin", "gd.bin")
dmCalculateDensityValue(list(4.4, 2.9, 1.4, 0.3))
## End(Not run)
```

dmGetAssignedSubspaces *Get subspaces in a data model to which a data record is assigned*

Description

Determine to which pairs of level and enumerated subspace a data record is assigned.

Usage

```
dmGetAssignedSubspaces(dataRecord)
```

Arguments

dataRecord	List of a data record

Value

List of pairs of level and enumerated subspace

Examples

```
## Not run:
dmRead("dm.bin", "gd.bin")
dmGetAssignedSubspaces(list(4.4, 2.9, 1.4, 0.3))
## End(Not run)
```

dmGetLevels *Get levels for subspaces*

Description

Get levels for subspaces in a data model.

Usage

```
dmGetLevels()
```

Value

Vector of levels

Examples

```
## Not run:
dmRead("dm.bin", "gd.bin")
dmGetLevels()
## End(Not run)
```

dmGetNumberOfSubspaces *Get number of subspaces for a level*

Description

Get number of subspaces in a data model for a level.

Usage

```
dmGetNumberOfSubspaces(level)
```

Arguments

level	Level

Value

Number of subspaces

Examples

```
## Not run:
dmRead("dm.bin", "gd.bin")
dmGetNumberOfSubspaces(0.73)
## End(Not run)
```

dmPlotEvaluate *Create an image file for an evaluated data source*

Description

Create an image file containing two-dimensional projections of generative data and an evaluated data source. Plot parameters are passed by functions dmPlotGenerativeDataParameters() and dmPlotEvaluateDataSourceParameters(). Data points are drwan in the order generative data, evaluated data source.

Usage

```
dmPlotEvaluate(
  evaluateDataSourceFileName,
  level,
  imageFileName,
  title,
  columnIndices,
 plotGenerativeDataParameters = dmPlotGenerativeDataParameters(percent = 10, colour =
```

```
    "red"),
 plotEvaluateDataSourceParameters = dmPlotEvaluateDataSourceParameters(colour = "blue")
)
```

Arguments

evaluateDataSourceFileName	Name of evaluated data source file'
level	Level for subspaces
imageFileName	Name of image file
title	Title of image
columnIndices	Vector of two column indices that are used for the two-dimensional projection. Indices refer to indices of active columns of data source.
plotGenerativeDataParameters	Plot generative data parameters, see dmPlotGenerativeDataParameters().
plotEvaluateDataSourceParameters	Plot parameters for evaluated data source, see dmPlotEvaluateDataSourceParameters().

Value

None

Examples

```
## Not run:
dmRead("dm.bin", "gd.bin")
dmRead("dm.bin", "gd.bin")
dmPlotEvaluate("ds.bin",
  0.73,
  "s1d34.png",
  "Subspace of Hierarchical, Categorical Data Model for the Iris Dataset",
  c(3, 4),
  dmPlotGenerativeDataParameters(10),
  dmPlotEvaluateDataSourceParameters())
## End(Not run)
```

---

dmPlotEvaluateDataSourceParameters

Specify plot parameters for evaluated data source

Description

Specify plot parameters for evaluated data source passed to function dmPlotSubspaces().

Usage

```
dmPlotEvaluateDataSourceParameters(colour = "blue")
```

Arguments

colour	Colour for data points of evaluaded data source

Value

List of plot parameters for evaluated data source

Examples

```
## Not run:
dmPlotEvaluateDataSourceParameters()
## End(Not run)
```

---

dmPlotGenerativeDataParameters

Specify plot parameters for generative data

Description

Specify plot parameters for generative data passed to function dmPlotSubspaces().

Usage

```
dmPlotGenerativeDataParameters(percent = 10, colour = "red")
```

Arguments

percent	Percentr of randomly selected data points of generative data
colour	Colour for data points of generative data

Value

List of plot parameters for generative data

Examples

```
## Not run:
dmPlotGenerativeDataParameters(5)
## End(Not run)
```

dmPlotSubspaceParameters *Specify plot parameters for subspaces in a data model*

Description

Specify plot parameters for subspaces in a data model passed to function dmPlotSubspaces().

Usage

```
dmPlotSubspaceParameters(percent = 100, boundary = TRUE, colour = "green")
```

Arguments

percent	Percent of randomly selected data points of generative data assigned to subspaces
boundary	Boolean value indicating if only data points of subspace boundaries should be selected
colour	Colour for data points of generative data assigned to subspaces

Value

List of plot parameters for subspaces

Examples

```
## Not run:
dmPlotSubspaceParameters(50)
## End(Not run)
```

dmPlotSubspaces *Create an image file for built subspaces*

Description

Create an image file containing two-dimensional projections of generative data, gernerative data assigned to subspaces and optionally an evaluated data source. Plot parameters are passed by functions dmPlotGenerativeDataParameters(), dmPlotSubspaceParameters(), dmPlotEvaluateDataSourceParameters(). Data points are drwan in the order generative data, generative data assigned to built subspaces and evaluated data source.

Usage

```
dmPlotSubspaces(
  level,
  enumeratedSubspaces,
  imageFileName,
  title,
  columnIndices,
 plotGenerativeDataParameters = dmPlotGenerativeDataParameters(percent = 10, colour =
    "red"),
 plotSubspaceParameters = dmPlotSubspaceParameters(percent = 100, colour = "green"),
  evaluateDataSourceFileName = "",
  plotEvaluateDataSourceParameters = NULL
)
```

Arguments

`level`	Level for subspaces
`enumeratedSubspaces`	Vector of enumerated subspaces
`imageFileName`	Name of image file
`title`	Title of image
`columnIndices`	Vector of two column indices that are used for the two-dimensional projection. Indices refer to indices of active columns of data source.
`plotGenerativeDataParameters`	Plot generative data parameters, see dmPlotGenerativeDataParameters().
`plotSubspaceParameters`	Plot parameters for subspaces, see dmPlotSubspaceParameters().
`evaluateDataSourceFileName`	Name of evaluated data source file
`plotEvaluateDataSourceParameters`	Plot parameters for evaluated data source, see dmPlotEvaluateDataSourceParameters().

Value

None

Examples

```
## Not run:
dmRead("dm.bin", "gd.bin")
dmPlotSubspaces(0.73,
  c(1),
  "s1d34.png",
  "Subspace of Hierarchical, Categorical Data Model for the Iris Dataset",
  c(3, 4),
  dmPlotGenerativeDataParameters(10),
  dmPlotSubspaceParameters(100),
```

```
  "ds.bin",
  dmPlotEvaluateDataSourceParameters())
## End(Not run)
```

---

dmRead *Read a data model and generative data from files*

Description

Read a data model and generative data from files. This function has to be called before calling API functions to which file names for a data model and gernerative data are not passed directly.

Usage

```
dmRead(dataModelFileName, generativeDataFileName)
```

Arguments

dataModelFileName	Name of data model file
generativeDataFileName	Name of generative data file

Value

None

Examples

```
## Not run:
dmRead("dm.bin", "gd.bin")
## End(Not run)
```

---

dmRemoveSubspaces *Remove subspaces from a data model for a level*

Description

Read a data model from file, remove subspaces from the data model for a level and write it to original file.

Usage

```
dmRemoveSubspaces(dataModelFileName, level)
```

Arguments

dataModelFileName	Name of data model file
level	Level

Value

None

Examples

```
## Not run:
dmRead("dm.bin", "gd.bin")
dmRemoveSubspaces("dm.bin", 0.73)
## End(Not run)
```

---

dmReset *Reset API*

Description

Reset API

Usage

```
dmReset()
```

Value

None

Examples

```
## Not run:
dmReset()
## End(Not run)
```

dmTrain *Train a neural network which approximates density values for a data source*

Description

Read a data source and generative data from files, train a neural network which approximates density values for a data source in iterative training steps, create a data model with the trained neural network and write it to a file in binary format.

Usage

```
dmTrain(
  dataModelFileName,
  dataSourceFileName,
  generativeDataFileName,
  numberOfIterations
)
```

Arguments

dataModelFileName	Name of data model file
dataSourceFileName	Name of data source file
generativeDataFileName	Name of generative data file
numberOfIterations	Number of iterations.

Value

None

Examples

```
## Not run:
dmGTrain("dm.bin", "iris4d.bin", "gd.bin", 10000)
## End(Not run)
```

Index