R Code Implementation

-

Variable Selection on Non-linear Manifolds

The code implements the Local Linear Selection Manifold (LLMS) and the Diagonal Auto-encoder Manifold Selection (DAMS), as presented in the PhD Thesis: *Desboulets, L. (2020). "Variable Selection on Non-linear Manifolds". AMSE Aix-Marseille University [unpublished].* It consists in two main functions:

- `LLMS.R` - for Local Linear Manifold Selection,
- `DAMS.R` - for Diagonal Autoencoders Manifold Selection,

and two sub-functions:

- `Simulate_Manifold.R` - for simulating data on a manifold.
- `Plot_Graph.R` - for plotting the sparse graphical model.

all written in the R language. There are no dependencies for the main functions. However, `Simulate_Manifold.R` is dependent on library `MASS`, and `Plot_Graph.R` on `viridis`. The former serves for simulating multivariate normal random variables, and the latter for customized colors.

Local Linear Manifold Selection

Description

This function implements the three versions of the Local Linear Manifold Selection estimator, as well as the graphical representation. In the first case, the output is a vector of selection probabilities, of the same length as the number of variables. In the second case, the output is a matrix of selection probabilities, that represents the edges of the graph.

Usage

```
LLMS(X, algo = "R-kNN", k = 0.2, theta = seq(0,2,length.out=100),
m = 5, q = 0.2, batch.size = 0.2, graph.model = FALSE)
```

Arguments

`X` The data matrix $n \times p$, or a dataframe.

`algo` The version of algorithm. Possible values are ”vanilla”, ”R-kNN”, ”q-kNN”, default is ”RkNN”.

`k` The size of neighbourhoods, as a fraction of the sample size n, default is 0.2.

`theta` An increasing vector of penalties θ, default is [0,2].

`m` Random subset size, default is 5. Only used if `algo = "R-kNN"`.

`q` Neighbourhood's neighbourhood size as a fraction of p, default is 0.2. Only used if `algo = "q-kNN"`.

`batch.size` Size of the random batch as a fraction of the sample size n. Default is 0.2. Because of replacement, it can be greater than 1.

`graph.model` Logical flag for computing the sparse graphical model. In that case the output is a matrix representing the edges of a graph, the diagonal is zero.

Value

`P` LLMS estimated vector of probabilities.

`Paths` Selection paths.

`G` 3D array of adjacency matrices, Each slice represents a sparse graphical model for a given penalty value. Only returned if `graph.model = TRUE`.

Examples

```r
## ---------------------------------------------------------------------------
## Clear Memory
rm(list=ls())

## ---------------------------------------------------------------------------
## Load the function to simulate data and perform LLMS
source("Simulate_Manifold.R")
source("LLMS.R")

## ---------------------------------------------------------------------------
## Simulate some data
X <- Simulate_Manifold()

## ---------------------------------------------------------------------------
## Invoke the function
result <- LLMS(X)

## ---------------------------------------------------------------------------
## Plot the Selection Paths
matplot(result$Paths,type="l",lwd=2)

## ---------------------------------------------------------------------------
## Plot the Sparse Graphical Model
result <- LLMS(X,graph.model=TRUE) # Rerun the algorithm with graph.model
source("Plot_Graph.R")             # Customized function for plotting
G <- result$Graph[,,100]           # The graph with the maximum penalty
Plot_Graph(G)
```

Diagonal Auto-Encoder Manifold Selection

Description

This function implements the three versions of the Diagonal Auto-Encoder Manifold Selection estimator. The output is a vector of probabilities that is the same length as number of variables in the data matrix.

Usage

```
DAMS(X, hidden.layers = c(7,2,7), max.iter = 2e3, batch.size = 0.2,
selection.start = 250, learning.rate = 0.05, ensemble.size = 100)
```

Arguments

`X` The data matrix $n \times p$.

`hidden.layers` Vector of length n_h specifying the size of each hidden layers. Default is $25 \times 5 \times 25$.

`max.iter` Number of iterations before the learning stops. Default is 10'000.

`batch.size` Size of the random batches as a fraction of the sample size n. Default is 0.2.

`start.selection` Iteration at which selection parameters start to be optimized. Default is 250.

`learning.rate` Normalized Learning rate controlling the speed of convergence. Default is 0.05.

`ensemble.size` Number of networks in the Ensemble. Default is 100.

Value

`P` DAMS estimated vector of probabilities (ensemble average).

`Paths` Selection paths.

`G` 3D array of adjacency matrices, Each slice represents a sparse graphical model for a given penalty value. Only returned if `graph.model = TRUE`.

Examples

```
## ---------------------------------------------------------------------------
## Clear Memory
rm(list=ls())

## ---------------------------------------------------------------------------
## Load the functions to simulate data and perform DAMS
source("Simulate_Manifold.R")
source("DAMS.R")

## ---------------------------------------------------------------------------
## Simulate some data
X <- Simulate_Manifold()

## ---------------------------------------------------------------------------
## Invoke the function
result <- DAMS(X)

## ---------------------------------------------------------------------------
## Plot Paths
matplot(result$Paths,type="l",lwd=2)

## ---------------------------------------------------------------------------
## Plot the Sparse Graphical Model
result <- DAMS(X,graph.model=TRUE) # Rerun the algorithm with graph.model
source("Plot_Graph.R")             # Customized function for plotting
G <- result$Graph[,,5]             # The graph with the halfway penalty
Plot_Graph(G)
```

Simulate_Manifold

Description

This function simulates a manifold, as defined in the numerical simulations design. C.f. section 2.6.1.

Usage

```
Simulate_Manifold(n = 1000, p = 20, active.set = 5, r = 1, sigma = 0,
max.corr = 0, linear = FALSE)
```

Arguments

`n`	Number of observations.
`p`	Number of candidate variables.
`active.set`	Number of active variables.
`r`	Dimension of the manifold.
`sigma`	Variance of the noise of the manifold.
`max.corr`	Maximum correlation between the noisy variables.
`linear`	Logical flag for the linearity/non-linearity of the manifold. Default is `FALSE`.

Value

`X`	The simulated data matrix $n \times p$, whose `active.set` first columns lie on a `r`-manifold.

Plot_Graph

Description

This function plot a sparse graphical model from adjacency matrix `G`, provided by `LLMS()` or `DAMS()`.

Usage

```
Plot_Graph(G, names= NULL, title = "Sparse Graphical Model",
circ.size = 5,text.size = 1)
```

Arguments

`G`	Adjacency matrix `G`.
`names`	Vector of names of the variables.
`title`	Title of the plot.
`circ.size`	Size of the vertices.
`text.size`	Size of the text inside the vertices.