Package 'Ghost'

May 8, 2026

Type Package

Title Missing Data Segments Imputation in Multivariate Streams

Version 0.1.0

Author Siyavash Shabani, Reza Rawassizadeh

Maintainer Siyavash Shabani <s.shabani.aut@gmail.com>

Description Helper functions provide an accurate imputation algorithm for reconstructing the missing segment in a multi-variate data streams. Inspired by single-shot learning, it reconstructs the missing segment by identifying the first similar segment in the stream. Nevertheless, there should be one column of data available, i.e. a constraint column. The values of columns can be characters (A, B, C, etc.). The result of the imputed dataset will be returned a .csv file. For more details see Reza Rawassizadeh (2019) <doi:10.1109/TKDE.2019.2914653>.

URL https://www.researchgate.net/publication/332779980_Ghost_Imputation_Accurately_Reconstructing_Missing_Data_of_the_Off_Period

Depends R (>= 2.10)

License GPL-3

Encoding UTF-8

LazyData true

NeedsCompilation no

Imports R6

RoxygenNote 7.0.1

Repository CRAN

Date/Publication 2020-03-25 16:50:05 UTC

Contents

reconstruct *reconstruct: Missing Data Segments Imputation in Multivariate Streams*

Description

Ghost is an accurate imputation algorithm for reconstructing the missing segment in multi-variate data streams. Inspired by single-shot learning, it reconstructs the missing segment by identifying the first similar segment in the stream. Nevertheless, there should be one column of data available, i.e. a constraint column. The values of columns can be characters (A, B, C,etc.). The result of the imputed dataset will be returned a .csv file.

Usage

```
reconstruct(data_frame, constraintCol, wSize, direction_save,epsilon)
```

Arguments

data_frame	A data frame with missing values.
constraintCol	The column number that all of its fields have data (without missing values). This column is considered as a constraint and it can always produce data, even if the system is shut down.
wSize	Length of a window that is used for data reconstruction, before and after the missing row(s).
direction_save	A direction for saving the output .CSV file(see details).
epsilon	A similarity coefficient that is used for searching in the algorithm (see details).

Details

More information about operation of algorithm is prepared in algorithm's article: https://www.researchgate.net/publication/33

Epsilon: The algorithm searches the data for the closest similar segment. As the first step, the algorithm determines prior and posterior segments missing part (the size of the segment will be given by wSize). As the second step, the algorithm starts to find the similar segment that passes the segment size and constraint similiarty condition. Sometimes, finding windows with exact similarity is impossible in a dataset. To mitigate this issue, and finding windows with approximate similarities the user can define the minimum percentage of similarity for searching the dataset with Epsilon coefficient.

direction_save: If the user inserts the Direction_save, the output file will be saved in the specified folder. Contrarily, if the user does not insert the Direction_save, the output file will be saved in the Environment R.

Author(s)

Siyavash Shabani,s.shabani.aut@gmail.com Reza Rawassizadeh,rrawassizadeh@acm.org

References

Rawassizadeh, Reza, Hamidreza Keshavarz, and Michael Pazzani. "Ghost Imputation: Accurately Reconstructing Missing Data of the Off Period." IEEE Transactions on Knowledge and Data Engineering (to appear).

Examples

```
#An example of the operation of the Algorithm.

data(test_ghost_csv)

## sample dataset----------------------------------
#   S0 S1 S2 S3
#1   5  F  G  H
#2   5  B  N  T
#3   4     P  O
#4   1  X  C  B
#5   1  N     X
#6   1  R  R  R
#7   1  W     W
#8   1  W  W  W
#9   2
#10  2
#11  1  O  K  O
#12  1  B     O
#13  2     S  D
#14  1  W  W
#15  1  W  S  W
#16  2  P  I  M
#17  2  R  U
#18  1  O  K  O
#19  1  B     O
#20  1  R  R  R
#21  5  F  G  H
#22  5  B  N  T
#23  4
#24  1  X  C  B
#25  1  N     X


reconstruct(test_ghost_csv,1,2,epsilon=0.4)

### output-----------------------------------------------------
#   S0 S1 S2 S3
#1   5  F  G  H
#2   5  B  N  T
#3   4     P  O
#4   1  X  C  B
#5   1  N     X
#6   1  R  R  R
#7   1  W     W
#8   1  W  W  W
```

```
#9   2  P  I  M
#10  2  R  U
#11  1  O  K  O
#12  1  B     O
#13  2     S  D
#14  1  W  W
#15  1  W  S  W
#16  2  P  I  M
#17  2  R  U
#18  1  O  K  O
#19  1  B     O
#20  1  R  R  R
#21  5  F  G  H
#22  5  B  N  T
#23  4     P  O
#24  1  X  C  B
#25  1  N     X
```

---

saxTransform *saxTransform*

Description

This function is added to the package to enable users converting numeric data to discrete data. This is due to the fact that Ghost designed for discrete data and this function discretize numeric data and prepare them for the ghost algorithm.

Usage

```
saxTransform(data_frame, buckets,skipColumnVec,constraint_row)
```

Arguments

`data_frame`	A data frame with numeric values.
`buckets`	The Input data range is divided to this number.
`skipColumnVec`	Column number that is not used in the algorithm.
`constraint_row`	Column number that is considered for constant column.

Author(s)

Siyavash Shabani, s.shabani.aut@gmail.com, Reza Rawassizadeh, rrawassizadeh@acm.org

References

1- Rawassizadeh, Reza, Hamidreza Keshavarz, and Michael Pazzani. "Ghost Imputation: Accurately Reconstructing Missing Data of the Off Period." IEEE Transactions on Knowledge and Data Engineering (to appear).

2- Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery (pp. 2-11). ACM.

Examples

```
data(sax_test)

#### Input dataframe-----------------------
#   S0  S1  S2 S3
#1   1   2  54 65
#2   1  NA  21 54
#3   2  34  32 87
#4   1  23  58 52
#5   1  43  75 56
#6   2  12  20 95
#7   1  54  14 87
#8   3  -6  NA 30
#9   2   5 -60 32
#10  1 -85  58 25
#11  2  78  95 45
#12  3  52  52 62
#13  2  20  NA 58
#14  3  NA -62 78
#15  1  20 -10 96
#16  1  30  -6 NA
#17  1  12 -85 45
#18  1  NA  78 20
#19  1  23  95 NA

saxTransform(sax_test,buckets =10,skipColumnVec=1, constraint_row=1)

### Output data-----------------------------------------------
#     S0  S1  S2   S3
# [1,] "1" "2" "54" "65"
# [2,] "1" ""  "f"  "h"
# [3,] "2" "g" "g"  "j"
# [4,] "1" "g" "h"  "h"
# [5,] "1" "h" "i"  "h"
# [6,] "2" "f" "f"  "k"
# [7,] "1" "h" "f"  "j"
# [8,] "3" "e" ""   "g"
# [9,] "2" "f" "b"  "g"
#[10,] "1" "a" "h"  "g"
#[11,] "2" "j" "k"  "h"
#[12,] "3" "h" "h"  "i"
#[13,] "2" "f" ""   "h"
```

```
#[14,] "3" ""  "b"  "j"
#[15,] "1" "f" "e"  "k"
#[16,] "1" "g" "e"  ""
#[17,] "1" "f" "a"  "h"
#[18,] "1" ""  "j"  "f"
#[19,] "1" "g" "k"  ""
```

Index