

Package ‘ramidst’

October 21, 2016

Type Package

Title An Interface to the AMIDST Toolbox for Data Stream Processing

Version 0.1.0

Date 2016-10-19

Author Antonio Salmeron, Dario Ramos-Lopez, Helge Langseth, Thomas D. Nielsen, Anders L. Madsen, Andres Masegosa, Ana M. Martinez, Rafael Cabanas

Maintainer Antonio Salmeron <antonio.salmeron@ual.es>

Description Offers a link to some of the functionality of the AMIDST toolbox <<http://www.amidsttoolbox.com>> for handling data streams. More precisely, the package provides inference and concept drift detection using hybrid Bayesian networks.

Depends R (>= 3.3.1), rJava (>= 0.9-8)

SystemRequirements JDK (>= 1.8)

License Apache License 2.0

LazyData TRUE

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-10-21 14:09:43

R topics documented:

amidst_data_stream	2
amidst_ds_iterator	2
dbn_generator	3
dynamic_importance_sampling	3
generate_stream_from_dbn	4
generate_stream_of_sequences	5
importance_sampling_from_stream	5
load_amidst_bn	7
load_dynamic_amidst_bn	7

load_dynamic_sequence	8
map_inference_from_stream	8
mpe_inference_from_stream	9
nb_concept_drift_detector_from_stream	10
print_amidst_bn	12
ramidst	12
save_dynamic_amidst_bn	13

Index 14

amidst_data_stream *Creates an AMIDST data stream from an arff file*

Description

Creates an AMIDST data stream from an arff file

Usage

amidst_data_stream(file_name)

Arguments

file_name the name of the data file in arff format

Value

a Java object of class DataStream

amidst_ds_iterator *Creates an iterator over an AMIDST data stream*

Description

Creates an iterator over an AMIDST data stream

Usage

amidst_ds_iterator(data_on_disk_reader)

Arguments

data_on_disk_reader
 an AMIDST data stream

Value

an iterator over the stream

dbn_generator	<i>Generates a random dynamic Bayesian network</i>
---------------	----------------------------------------------------

Description

Generates a random dynamic Bayesian network

Usage

```
dbn_generator(n_c_vars, n_d_vars, n_s)
```

Arguments

n_c_vars	the number of Gaussian variables
n_d_vars	the number of discrete variables
n_s	the number of states of the discrete variables

Value

a Java object of class DynamicBayesianNetwork

Examples

```
d <- dbn_generator(3,2,2)
print_amidst_bn(d)
```

dynamic_importance_sampling	<i>Runs belief update from a piece of dynamic evidence over a dynamic Bayesian network</i>
-----------------------------	--------------------------------------------------------------------------------------------

Description

Runs belief update from a piece of dynamic evidence over a dynamic Bayesian network

Usage

```
dynamic_importance_sampling(dyn_network, target_variable, dyn_evidence,
  sample_size = 50)
```

Arguments

dyn_network a java object of class DynamicBayesianNetwork over which the computations will be carried out
target_variable the name of the variable over which the posterior distribution will be computed
dyn_evidence the observations over the dynamic Bayesian network.
sample_size the size of the sample used to estimate the posterior distribution.

Value

a data.frame with the posterior distribution over the target variable on the different time slices.

Note

The function uses importance sampling and is based on the factored frontier method, see the AMIDST toolbox documentation.

Examples

```

network <- dbn_generator(1,2,2)
print_amidst_bn(network)
stream <- generate_stream_from_dbn(network,1,10,"ClassVar")
resultsIS <- dynamic_importance_sampling(network,"ClassVar",stream)
plot(resultsIS[,2],type="l",ylim = c(0,1),col="red",xlab="Time slice",ylab="Prob. ClassVar = 1")

```

generate_stream_from_dbn

Generates a stream from a DBN. The DBN is assumed to be a classifier, and therefore there is a class variable.

Description

Generates a stream from a DBN. The DBN is assumed to be a classifier, and therefore there is a class variable.

Usage

```
generate_stream_from_dbn(net, sequences, seq_length, class_var)
```

Arguments

net the network that will be sampled
sequences the number of random sequences to generate. Each one will be an element of the stream.
seq_length the length of the sequences
class_var the class variable

Value

a Java object of class DataStream with the generated sequences

generate_stream_of_sequences

Generates a stream where each item is a full sequence from some dynamic evidence.

Description

Generates a stream where each item is a full sequence from some dynamic evidence.

Usage

generate_stream_of_sequences(dynamic_evidence)

Arguments

dynamic_evidence
some dynamic observation over s DBN

Value

a data stream out of the input evidence

importance_sampling_from_stream

Runs Importance sampling evidence updating from an AMIDST data stream

Description

Runs Importance sampling evidence updating from an AMIDST data stream

Usage

importance_sampling_from_stream(network, target_variable, evidence_variables, input_stream, sample_size, parallel = T, seed = 3L)

Arguments

<code>network</code>	a java object of class <code>BayesianNetwork</code> over which the computations will be carried out
<code>target_variable</code>	a string representing the name of variable whose posterior distribution will be computed
<code>evidence_variables</code>	a vector with the names of the observed variables
<code>input_stream</code>	and AMIDST data stream
<code>sample_size</code>	the sample size to be used during the simulation
<code>parallel</code>	a boolean indicating whether or not the items in the sample will be generated in parallel (when allowed by the system)
<code>seed</code>	the seed for the generation of random numbers

Value

a `data.frame` with the posterior distribution of the target variable for each item in the stream

Note

The function computes the posterior distribution given some evidence for all the items in the input stream.

References

A. Salmeron, D. Ramos-Lopez, H. Borchani, A.M. Martinez, A.R. Masegosa, A. Fernandez, H. Langseth, A.L. Madsen, T.D. Nielsen (2015) Parallel importance sampling in conditional linear Gaussian networks. CAEPIA'2015. Lecture Notes in Artificial Intelligence 9422, 36-46.

Examples

```
network <- load_amidst_bn(system.file("extdata", "WasteIncinerator.bn",
package="ramidst"))
sample_stream <- amidst_data_stream(system.file("extdata",
"WasteIncineratorSample.arff", package="ramidst"))
posterior <- importance_sampling_from_stream(network, "B", c("F", "E"),
sample_stream, 50L)
posterior
posterior <- importance_sampling_from_stream(network, "L", c("F", "E"),
sample_stream, 50L)
posterior
```

load_amidst_bn	<i>Loads an AMIDST BN from a fle</i>
----------------	--------------------------------------

Description

Loads an AMIDST BN from a fle

Usage

```
load_amidst_bn(file_name)
```

Arguments

file_name the name of the file containing the network in .net format

Value

a java object of class BayesianNetwork

Examples

```
network <- load_amidst_bn(system.file("extdata", "WasteIncinerator.bn",  
package="ramidst"))  
print_amidst_bn(network)
```

load_dynamic_amidst_bn	<i>Loads an AMIDST dynamic BN from a fle</i>
------------------------	----------------------------------------------

Description

Loads an AMIDST dynamic BN from a fle

Usage

```
load_dynamic_amidst_bn(file_name)
```

Arguments

file_name the name of the file containing the network in .net format

Value

a java object of class DynamicBayesianNetwork

load_dynamic_sequence *Loads an AMIDST dynamic data sequence from a file*

Description

Loads an AMIDST dynamic data sequence from a file

Usage

```
load_dynamic_sequence(file_name)
```

Arguments

file_name the name of the file containing the data sequence

Value

a Java object of class DataStream

map_inference_from_stream

Runs MAP inference from an AMIDST data stream

Description

Runs MAP inference from an AMIDST data stream

Usage

```
map_inference_from_stream(network, map_variables, evidence_variables,
    input_stream, sample_size, parallel = T, seed = 3L)
```

Arguments

network a java object of class BayesianNetwork over which the computations will be carried out

map_variables a vector with the name of the variables over which the MAP configuration will be computed

evidence_variables a vector with the names of the observed variables

input_stream and AMIDST data stream

sample_size the sample size to be used for estimating marginals

parallel a boolean indicating whether or not the items in the sample will be generated in parallel (when allowed by the system)

seed the seed for the generation of random numbers

Value

a data.frame with the MAP configuration of the variables of interest for each item in the stream

Note

The function computes the MAP configuration of the variables of interest given some evidence for all the items in the input stream.

References

D. Ramos-Lopez, A. Salmeron, R. Rumi, A.M. Martinez, T.D. Nielsen, A.R. Masegosa, H. Langseth, A.L. Madsen (2016) Scalable MAP inference in Bayesian networks based on a Map-Reduce approach. PGM'2016. JMLR: Workshop and Conference Proceedings, vol. 52: 415-425.

Examples

```
## Not run:
network <- load_amidst_bn(system.file("extdata", "WasteIncinerator.bn",
package="ramidst"))
sample_stream <- amidst_data_stream(system.file("extdata",
"WasteIncineratorSample.arff", package="ramidst"))
map_configurations <- map_inference_from_stream(network, c("D", "B"), c("W"),
sample_stream, 5L)
map_configurations

## End(Not run)
```

mpe_inference_from_stream

Runs MPE inference from an AMIDST data stream

Description

Runs MPE inference from an AMIDST data stream

Usage

```
mpe_inference_from_stream(network, evidence_variables, input_stream,
parallel = T, seed = 3L)
```

Arguments

network	a java object of class BayesianNetwork over which the computations will be carried out
evidence_variables	a vector with the names of the observed variables
input_stream	and AMIDST data stream

parallel	a boolean indicating whether or not the items in the sample will be generated in parallel (when allowed by the system)
seed	the seed for the generation of random numbers

Value

a data.frame with the MPE configuration for each item in the stream

Note

The function computes the most probable explanation of the evidence in all the items in the input stream.

References

D. Ramos-Lopez, A. Salmeron, R. Rumi, A.M. Martinez, T.D. Nielsen, A.R. Masegosa, H. Langseth, A.L. Madsen (2016) Scalable MAP inference in Bayesian networks based on a Map-Reduce approach. PGM'2016. JMLR: Workshop and Conference Proceedings, vol. 52: 415-425.

Examples

```
## Not run:
network <- load_amidst_bn(system.file("extdata", "WasteIncinerator.bn",
package="ramidst"))
sample_stream <- amidst_data_stream(system.file("extdata",
"WasteIncineratorSample.arff", package="ramidst"))
mpe_configurations <- mpe_inference_from_stream(network, c("E"), sample_stream)
mpe_configurations

## End(Not run)
```

nb_concept_drift_detector_from_stream

Naive Bayes concept drift detector from an AMIDST data stream

Description

Naive Bayes concept drift detector from an AMIDST data stream

Usage

```
nb_concept_drift_detector_from_stream(input_stream, class_index = -1L,
window_size, transition_variance = 0.1, hidden_vars = 1L)
```

Arguments

input_stream	an AMIDST input stream
class_index	the index of the class variable in the list of variables
window_size	the number of items in the stream to be analysed simultaneously
transition_variance	the variance of the transition distribution
hidden_vars	the number of global hidden variables to include in the model

Value

the value of the hidden variables for each window

Note

The function builds a dynamic naive Bayes model with a Gaussian hidden variable which is aimed at capturing an underlying unobserved process.

References

H. Borchani, A.M. Martinez, A.R. Masegosa, H. Langseth, T.D. Nielsen, A. Salmeron, A. Fernandez, A.L. Madsen, R.Saez (2015) Modeling concept drift: A probabilistic graphical model based approach. IDA'2015. Lecture Notes in Computer Science 9385, 72-83.

Examples

```
## Not run:
data <- amidst_data_stream(system.file("extdata", "sea.arff",
package="ramidst"))
results <- nb_concept_drift_detector_from_stream(data, class_index = -1L,
window_size=1000L, transition_variance=0.1, hidden_vars=1L)
re <- 0
for (k in 1:length(results)) re[k] <- results[[k]]
ymin = min(re)-0.05
ymax = max(re)+0.05
plot(re, type="l", ylim=c(ymin,ymax), ylab="Hidden variable",
xlab="Instance number (x 1000)")
abline(v=15, col="red")
abline(v=30, col="red")
abline(v=45, col="red")

## End(Not run)
```

print_amidst_bn	<i>Prints an AMIDST model</i>
-----------------	-------------------------------

Description

Prints an AMIDST model

Usage

```
print_amidst_bn(network)
```

Arguments

network the network to print out.

Examples

```
network <- load_amidst_bn(system.file("extdata", "WasteIncinerator.bn",  
package="ramidst"))  
print_amidst_bn(network)
```

ramidst	<i>ramidst: A package for accessing inference and concept drift detection features provided by the AMIDST Toolbox for handling massive data streams. Visit http://www.amidsttoolbox.com for additional information on the AMIDST Toolbox. This package was partially developed as part of the AMIDST and PGMs-SDA projects. AMIDST has received funding from the European Union's 7th Framework Programme for research, technological development and demonstration under grant no. 619209. PGMs-SDA has received funding from the Spanish Ministry of Economy and Competitiveness and FEDER funds under grant TIN2013-46638-C3-1-P.</i>
---------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Description

ramidst: A package for accessing inference and concept drift detection features provided by the AMIDST Toolbox for handling massive data streams. Visit <http://www.amidsttoolbox.com> for additional information on the AMIDST Toolbox. This package was partially developed as part of the AMIDST and PGMs-SDA projects. AMIDST has received funding from the European Union's 7th Framework Programme for research, technological development and demonstration under grant no. 619209. PGMs-SDA has received funding from the Spanish Ministry of Economy and Competitiveness and FEDER funds under grant TIN2013-46638-C3-1-P.

`save_dynamic_amidst_bn`*Saves an AMIDST dynamic BN to a file*

Description

Saves an AMIDST dynamic BN to a file

Usage

```
save_dynamic_amidst_bn(network, file_name)
```

Arguments

<code>network</code>	a java object of class <code>DynamicBayesianNetwork</code>
<code>file_name</code>	the name of the file that will contain the network

Examples

```
network <- dbn_generator(1,2,2)
save_dynamic_amidst_bn(network, "sample_net.bn")
new_network <- load_dynamic_amidst_bn("sample_net.bn")
print_amidst_bn(network)
```

Index

[amidst_data_stream](#), 2
[amidst_ds_iterator](#), 2

[dbn_generator](#), 3
[dynamic_importance_sampling](#), 3

[generate_stream_from_dbn](#), 4
[generate_stream_of_sequences](#), 5

[importance_sampling_from_stream](#), 5

[load_amidst_bn](#), 7
[load_dynamic_amidst_bn](#), 7
[load_dynamic_sequence](#), 8

[map_inference_from_stream](#), 8
[mpe_inference_from_stream](#), 9

[nb_concept_drift_detector_from_stream](#),
10

[print_amidst_bn](#), 12

[ramidst](#), 12
[ramidst-package \(ramidst\)](#), 12

[save_dynamic_amidst_bn](#), 13