Package ‘gRapfa’

February 19, 2015

Type Package
Title Acyclic Probabilistic Finite Automata
Version 1.0
Date 2014-04-10
Author Smitha Ankinakatte <Smitha.AA@agrsci.dk> and David Edwards <David.Edwards@agrsci.dk>
Maintainer Smitha Ankinakatte <Smitha.AA@agrsci.dk>
Description gRapfa is for modelling discrete longitudinal data using acyclic probabilistic finite automata (APFA). The package contains functions for constructing APFA models from a given data using penalized likelihood methods. For graphical display of APFA models, gRapfa depends on ‘igraph package’. gRapfa also contains an interface function to Beagle software that implements an efficient model selection algorithm.
License GPL (>= 2)
Depends R (>= 3.0.2), igraph
NeedsCompilation no
Repository CRAN
Date/Publication 2014-04-11 22:47:57

R topics documented:
gRapfa-package ......................................................... 2
add.stats ............................................................. 2
apfa2NS .............................................................. 3
BioFam ................................................................. 4
contract.last.level .................................................. 5
cross.validate ....................................................... 5
dIC ...................................................................... 6
fit.APFA ............................................................... 8
getXY ................................................................. 9
KL ...................................................................... 9
LogLike.APFA ....................................................... 10
merge2nodes ......................................................... 11
Description

The package supports the use of models for discrete longitudinal data using acyclic probabilistic finite automata (APFA).

Details

Package: gRapfa
Type: Package
Version: 1.0
Date: 2014-04-10
License: GPL (>= 2) Depends: R (>= 3.0.2), igraph

Author(s)

Smitha Ankinakatte <Smitha.AA@agrsci.dk> and David Edwards <David.Edwards@agrsci.dk>
Maintainer: Smitha Ankinakatte <Smitha.AA@agrsci.dk>

References

Ankinakatte, S. and Edwards, D. Modelling discrete longitudinal data using acyclic probabilistic finite automata. Submitted to "Computational Statistics and Data Analysis".

Description

Adds edge probabilities, log-likelihood and dimension to an APFA igraph object.

Usage

```
add.stats(G)
```

Arguments

- `G`: An APFA object

Value

The input APFA with additional attributes - edge probabilities, log-likelihood and dimension.

Author(s)

Smitha Ankinakatte and David Edwards

Examples

```
library(grapfa)
data(Wheeze)
G <- st(Wheeze)
G.as1 <- add.stats(G)
G.c <- contract.last.level(G)
G.as2 <- add.stats(G.c)
```

---

### `apfa2NS`

### APFA to node symbol array

**Description**

Derives a node by symbol array from an APFA.

**Usage**

```
apfa2NS(G)
```

**Arguments**

- `G`: A graph object of class `igraph`

**Details**

A node by symbol array represents the APFA in a convenient form for computation. The rows correspond to nodes, the columns correspond to edge symbols (twice). The first $k$ columns contain the ids of the target nodes for an edge from the node with the corresponding symbol. Zeroes mean that there is no corresponding edge. The last $k$ columns contain the corresponding edge counts.
**Value**

output: Derived node by symbol array.

**Author(s)**

Smitha Ankinakatte and David Edwards

**Examples**

```r
library(grapfa)
data(wheeze)
G <- st(wheeze)
G.c <- contract.last.level(G)
ns.array <- apfa2NS(G.c)
```

**Description**

Family life sequences from the Swiss Household Panel biographical survey. Life course sequences of family status for 2000 individuals, recorded from age 15 to 30. The data were collected retrospectively in a survey carried out in 2002. The data set also contains two covariates, sex and religion, derived from the original data.

**Usage**

BioFam

**Format**

Data frame of 2000 rows and 16 columns

**Source**

Swiss Household Panel, www.swisspanel.ch

**References**


contract.last.level

Description

contract.last.level merges all nodes at the last level of a sample tree or other APFA igraph object.

Usage

contract.last.level(G)

Arguments

G G is a sample tree or other APFA igraph object.

Value

An APFA igraph object.

Author(s)

Smitha Ankinakatte and David Edwards

Examples

library(gRapfa)
data(Wheeze)
G <- st(Wheeze)
G.c <- contract.last.level(G)
plot(G.c)

cross.validate

K - cross validation

Description

The function measures the prediction efficiency of the model using K-fold cross-validation.

Usage

cross.validate(Data, K = 10, crit = NULL, beagle = TRUE, dir = '')
Arguments

- **Data**: A data frame
- **K**: Number of cross validations
- **crit**: The model selection criterion, either AIC or BIC, for penalised likelihood method.
- **beagle**: If beagle=TRUE, the function performs model selection using BEAGLE.
- **dir**: specifying the path for `beagle.jar` directory.

Details

The cross validation for a given data frame is done as follows:
1. The data is divided into K subsets of equal sizes.
2. At each cross validation step in k=1:K, the kth subset is taken as the test data and the rest as training data.
3. APFA model is fitted to the training data using a model selection method (AIC, BIC or Beagle), then using the edge probabilities of the fitted model, the loglikelihood and the per-symbol loglikelihood are calculated for the test data set.
4. The function returns the mean of the log-likelihood from K-cross validation and pzero.

Value

Returns per symbol loglikelihood of the K-cross validation.

Author(s)

Smitha Ankinakatte and David Edwards

References


Examples

```r
library(gRapfa)
data(Wheeze)
```

---

**dIC**

*Difference in information criteria*

Description

The function returns the difference in AIC or BIC associated with merging a node pair in an APFA igraph object.

Usage

```r
dIC(G, nodeset, crit = "BIC", NS=NULL)
```
**Arguments**

- **G**: APFA igraph object
- **nodeset**: vector of length two, contain the names of the nodes to be merged.
- **crit**: Information criterion, 'AIC' or 'BIC' or a positive numerical value for the tuning parameter.
- **NS**: Node symbol array

**Details**

\[ dIC \text{ is } \text{The penalized likelihood criterion, } IC(A) = -2(A) + \alpha \cdot \text{dim}(A), \text{ where dim}(A) \text{ is the number of free parameters under A, and } \alpha \text{ is a tuning parameter. For the AIC, } \alpha=2 \text{ and for the BIC, } \alpha= \log(N). \text{ BIC penalises the parameters more heavily and so selects simpler models.} \]

The difference in IC is

\[ d(IC) = IC(A_0) - IC(A) = G^2 - \alpha \cdot \text{df}, \text{ where } A_0 \text{ is the APFA obtained after merging the two nodes in } A, G^2 \text{ is the deviance statistic and } \text{d.f.} \text{ is the associated degrees of freedom.} \]

**Value**

A numerical vector of length three containing \( d(IC), G^2 \) and the degrees of freedom.

**Author(s)**

Smitha Ankinakatte and David Edwards.

**References**


**Examples**

```r
library(grapfa)
data(Wheeze)
G <- st(Wheeze)
G.c <- contract.last.level(G)
dic1 <- dIC(G.c, nodeset=c(5,3))
dic2 <- dIC(G.c, nodeset=c(6,4))
```
**fit.APFA**  
*Fitting an APFA igraph object to data*

**Description**

Fits the APFA igraph object $G$ to a commensurate dataset, i.e., the edge probabilities are calculated using the data.

**Usage**

```r
fit.APFA(G, dat)
```

**Arguments**

- **G**: An APFA object.
- **dat**: The data for which the APFA models from $G$ has to be fit.

**Details**

Any observations not in the sample space of $G$ are ignored.

**Value**

Returns fitted APFA igraph object for the given data.

**Author(s)**

Smitha Ankinakatte and David Edwards

**Examples**

```r
library(grapfa)
data(Wheeze)
samp <- sample(1:537, 250)
G <- select.IC(Wheeze[samp,])
G.fit <- fit.APFA(G, Wheeze[!samp,])
```
getXY

Description

Sets XY co-ordinates for the igraph plots.

Usage

getXY(G)

Arguments

G an apfa igraph object.

Details

The function uses the igraph layout.sugiyama function. For large graphs the layout.drl function is more efficient.

Value

An array containing a set of co-ordinates for the graph.

Author(s)

Smitha Ankinakatte and David Edwards

See Also

igraph.plot

---

KL

Kullback-Leibler divergence for APFA models

Description

The Kullback-Leibler divergence measures the similarity between two APFA models. If the two models are identical then it is zero.

Usage

KL(A, B)
Arguments

A

APFA igraph object

B

APFA igraph object

Details

A and B must be commensurate, i.e., defined on the same variable set. Note that the KL-divergence is not a true distance measure, since it is not symmetric in A and B. For large APFA the computation of the KL-divergence may be prohibitive in both time and memory.

Value

Returns the KL-divergence.

Author(s)

Smitha Akinakatte and David Edwards

References


Examples

library(grapfa)
data(Wheeze)
samp <- sample(1:537, 250)
G <- select.IC(Wheeze[samp])
G.fit <- fit.APFA(G, Wheeze[-samp])
kl <- KL(G, G.fit)

Description

Uses the edge probabilities from G to calculate the log likelihood of the model.

Usage

LogLike.APFA(G, dat, complete.cases=TRUE)

Arguments

G

a fitted APFA

dat

a data frame that contains the same variables that G is based on.

complete.cases

a Boolean that determines whether incomplete cases are included in the calculations (see Details).
Details

An observation in the data may not be in the sample space of the APFA, i.e. there may not a root-to-sink path in the APFA generating the observation. However, there will be a partial path, that is, generating the initial part of the observation. If complete.cases is true, such observations are excluded from the calculations, otherwise contributions from the partial path are included.

See the reference below for the per-symbol log-likelihood.

Value

Returns the log-likelihood and the per-symbol log-likelihood.

Author(s)

Smitha Ankinakatte and David Edwards

References


See Also

add.stats

Examples

library(gRapfa)
data(Wheeze)
G <- st(Wheeze)
samp <- sample(1:537, 250)
G <- select.IC(Wheeze[samp,])
G LL <- LogLike.APFA(G, Wheeze[-samp,])

merge2nodes

Merge two nodes

Description

Calculates various quantities in connection with merging two nodes in a level of a sample tree.

Usage

merge2nodes(NS, mnode, test = TRUE, get.map = FALSE, doMerge = FALSE)
merge2nodes

Arguments

NS is a node by symbol array, the 1st half of the columns are node ids, the 2nd half the edge counts. When the corresponding edge is absent, the edge id is set to 0.

mnode is a vector of nodes to be merged, specified as vertex ids (rather than names). Required to be of length two.

test If test=TRUE, the deviance and df associated with the merging are returned.

get.map If get.map=TRUE, a map is returned.

doMerge If doMerge=TRUE, NS returned is the node by symbol array after merging (used in MergeNodes)

Value

A list of computed quantities

mmat An integer matrix containing the nodes to be merged (the original and the induced).

map A integer vector of length vcount(G)) containing the vertex ids of the vertices after merging

devtest A numeric vector of length two containing the degrees of freedom and deviance associated with the merging

NS A node by symbol array representing the result of the merging

Author(s)

Smitha Ankinakatte and David Edwards

References


Examples

library(grapfa)
data(Wheeze)
G <- st(Wheeze)
G.c <- contract.last.level(G)
NS <- apfa2NS(G.c)
n2n <- merge2nodes(NS, c(5,3))
**MergeNodes**

**Merge given node set of an APFA**

**Description**

Merges two nodes (at the same level) in an APFA, returning the resulting APFA.

**Usage**

```r
MergeNodes(G, nodeset, NS = NULL, setLayout = TRUE)
```

**Arguments**

- `G`:
  - G is an APFA object.
- `nodeset`:
  - nodeset is a vector of vertex names or nodes of length two, which are to be merged.
- `NS`:
  - NS is the node by symbol array for G. Supplying this will speed the computations.
- `setLayout`:
  - If setLayout=TRUE sets XY coordinates for the graph

**Details**

If necessary, more details than the description above.

**Value**

Returns the APFA igraph object after merging.

**Author(s)**

Smitha Ankinakatte and David Edwards

**See Also**

- `merge2nodes`

**Examples**

```r
library(gRapfa)
data(Wheeze)
G <- st(Wheeze)
G.c <- contract.last.level(G)
G.m <- MergeNodes(G.c, c(5,3))
```
**MergeSelect**

*Selection of nodes to merged at a level*

**Description**

At a given level of an APFA object, the function performs a greedy search of node pairs to be merged.

**Usage**

```
MergeSelect(G, NS = NULL, this.level, crit = "BIC", verbose = FALSE)
```

**Arguments**

- **G**: an APFA igraph object
- **NS**: a node by symbol array. Supplying this instead of G will speed computations.
- **this.level**: The level in which nodes are searched to be merged.
- **crit**: The criterion for the model selection, either AIC or BIC.
- **verbose**: If verbose is TRUE, then the all the information on the merge selection of nodes at each level are printed in the output.

**Details**

The function performs greedy selection at the given level. That is to say, the delta ICs for all nodes pairs at the given level are computed, the pair leading to the greatest reduction are merged, the delta ICs are recomputed as necessary, and the process continues until no further reduction in IC can be made.

**Value**

Returns the list of G: resulting APFA and ns: node-symbol array.

**Author(s)**

Smitha Ankinakatte and David Edwards

**See Also**

dIC, MergeNodes
Examples

data(Wheezel)
G <- st(Wheezel)
G <- contract.last.level(G)
G1 <- MergeSelect(G, this.level=3)
G <- G1$G
G$layout <- getXY(G)
plot(G)

Description

The mildew.raw is stem from a cross between two isolates of the barley powdery Mildew fungus. For each of N = 70 offspring, p = 6 binary markers, each corresponding to a single locus, were recorded.

Usage

mildew.raw

Format

Data frame of 70 observations with 6 variables

Source


References

select.beagle  

Select APFA using Beagle

Description

select.beagle runs Beagle software selection method to build APFA model.

Usage

select.beagle(A, m=4, b=0.2, dir = '', row.marker = FALSE, col.hap = FALSE)

Arguments

A  
a data frame whose variables are factors. Any missing values are regarded as additional factor levels.

m  
is the scale parameter.

b  
the shift parameter

dir  
path of the directory to find the beagle.jar file. The destined path should end with either '\\' or '/'. If dir=' ', by default the function considers the beagle.jar is in the same directory as R is running.

row.marker  
For genotype data, to specify whether the markers are in rows or columns

col.hap  
For genotype data, to specify whether the haplotypes are in rows or in columns

Details

select.beagle is an interface in R to work with Beagle software. Beagle is a software package for analysis of large scale genetic data sets. Beagle is written in Java, hence the Java interpreter needs to needs to be installed. More details on downloading and installing Beagle can be found in http://faculty.washington.edu/browning/beagle/beagle.html.

Value

Returns APFA igraph object using Beagle software for the calculations.

Author(s)

Smitha Ankinakatte and David Edwards

See Also


select.IC

Examples

# you have to have 'beagle.jar' to run the code below,
#library(gRAPfa)
#data(Wheeze)
#G <- select.beagle(Wheeze)

select.IC  Selection of APFA model using penalised likelihood criteria

Description

Selects an APFA using the algorithm described in Ankinakatte and Edwards (2014)

Usage

select.IC(dat, crit = "BIC", verbose = FALSE)

Arguments

dat  a data frame whose variables are factors. Any missing values are regarded as additional factor levels.
crit  Model selection criteria, either AIC or BIC.
verbose  If verbose=TRUE, then the function prints calculations involved in merge selection method.

Value

Returns an APFA model

Author(s)

Smitha Ankinakatte and David Edwards

References


Examples

library(gRAPfa)
data(Wheeze)
G <- select.IC(Wheeze)
**simulateAPFA**

*Simulation of data from an APFA model*

**Description**

The function draws a number of independent samples from the given APFA model.

**Usage**

```
simulateAPFA(g, Nsim = 1000)
```

**Arguments**

- `g` is an APFA model from which the samples are drawn.
- `Nsim` is the number of simulations.

**Details**

The function simulates the data from an APFA model using the edge probabilities.

**Value**

A data frame containing the simulated data.

**Author(s)**

Smitha Akinakatte and David Edwards

**Examples**

```
library(gRapfa)
data(Wheeze)
G <- select.IC(Wheeze)
simWheeze <- simulateAPFA(G)
head(simWheeze)
```

---

**st**

*Sample tree*

**Description**

Builds the sample tree of a discrete longitudinal dataset.

**Usage**

```
st(iD)
```
Arguments

id

A data frame whose variables are factors. Any missing values are treated as additional factor levels.

Details

Sample trees are constructed as follows. Suppose $N$ observations of $p$ discrete variables $(x_1, \ldots, x_p)$ are given. Starting with $N$ observations at the root node, edges branch out to nodes at the first level. The number of branches corresponds to the number of distinct values of $x_1$, and the count on each edge corresponds to the frequency of occurrence of the respective value. From each node at level one, edges branch out to level two, based on the distinct values of $x_2$ given $x_1$. The process continues up to level $p$.

Value

An igraph object containing the sample tree.

Note

Further notes

Author(s)

Smitha Ankinakatte

References


Examples

data(Wheeze)
G <- st(Wheeze)
E(G)$arrow.size <- 0.6
V(G)$size <- 10
V(G)$label <- ''
E(G)$label <- E(G)$count
plot(G)

Wheeze

Wheeze in Steubenville children

Description

Wheeze is a longitudinal data set of binary indicator of the presence of wheeze at ages 7, 8, 9 and 10, among 537 Steubenville children.
Usage

Wheeze

Format

Data frame of 537 observations with 4 variables

Source


References

Index

*Topic R, longitudinal data, BioFam
  BioFam, 4
*Topic R, longitudinal data, mildew
  mildew.raw, 15
*Topic R, longitudinal data, wheeze
  Wheeze, 19

add.stats, 2, 11
apfa2NS, 3

BioFam, 4

contract.last.level, 5
cross.validate, 5

dIC, 6, 14

fit.APFA, 8

getXY, 9
gRapfa (gRapfa-package), 2
gRapfa-package, 2

KL, 9

LogLike.APFA, 10

merge2nodes, 11, 13
MergeNodes, 13, 14
MergeSelect, 14
mildew.raw, 15

select.beagle, 16
select.IC, 17
simulateAPFA, 18
st, 18

Wheeze, 19