

Package: mrfse (via r-universe)

August 21, 2024

Title Markov Random Field Structure Estimator

Version 0.4.1

Date 2022-11-10

Description Three algorithms for estimating a Markov random field structure. Two of them are an exact version and a simulated annealing version of a penalized maximum conditional likelihood method similar to the Bayesian Information Criterion. These algorithm are described in Frondana (2016) [doi:10.11606/T.45.2018.tde-02022018-151123](https://doi.org/10.11606/T.45.2018.tde-02022018-151123). The third one is a greedy algorithm, described in Bresler (2015) [doi:10.1145/2746539.2746631](https://doi.org/10.1145/2746539.2746631).

License GPL (>= 3)

NeedsCompilation yes

LinkingTo Rcpp

Imports Rcpp

Depends gtools, Rfast

Author Rodrigo Carvalho [aut, cre], Florencia Leonardi [rev, ths]

Maintainer Rodrigo Carvalho <rodrigorsdc@gmail.com>

Date/Publication 2022-11-16 14:40:15 UTC

Repository <https://rodrigorsdc.r-universe.dev>

RemoteUrl <https://github.com/cran/mrfse>

RemoteRef HEAD

RemoteSha ef461185fc0b73c664f4c6417b2e14dc13a13dce

Contents

mrfse.ci	2
mrfse.ci.con	3
mrfse.ci.ncon	4
mrfse.create.sampler	5
mrfse.exact	6

mrfse.exact.con	7
mrfse.exact.ncon	8
mrfse.sa	9
mrfse.sa.con	10
mrfse.sa.ncon	11
mrfse.sample	12

Index 13

mrfse.ci	<i>Bresler's non-binary Markov random field structure estimator</i>
----------	---

Description

A greedy algorithm to estimate Markovian neighborhoods.

Usage

```
mrfse.ci(a_size, sample, tau, max_degree=ncol(sample)-1)
```

Arguments

a_size	Size of the alphabet.
sample	A integer-valued matrix. Each value must belong range 0 and a_size - 1. Matrix has dimension n x V, where n is number of samples and V is number of nodes.
tau	A hyperparameter. See references.
max_degree	The maximum length of a candidate Markovian neighborhood. Must be non-negative and less than ncol(sample).

Value

A list filled with estimated Markov neighborhood for each graph vertex

Author(s)

Rodrigo Carvalho

References

Guy Bresler. 2015. Efficiently Learning Ising Models on Arbitrary Graphs. In Proceedings of the forty-seventh annual ACM symposium on Theory of Computing (STOC '15). Association for Computing Machinery, New York, NY, USA, 771–782. DOI:<https://doi.org/10.1145/2746539.2746631>

Examples

```
library(mrfse)
a_size = c(0, 1)
s = matrix(sample(a_size, size=1000, replace=TRUE), ncol=5)
mrfse.ci(length(a_size), s, 0.2)
```

Description

A greedy algorithm to estimate Markovian neighborhoods.

Usage

```
mrfse.ci.con(a_size, sample, tau, max_degree=ncol(sample)-1)
```

Arguments

<code>a_size</code>	Size of the alphabet.
<code>sample</code>	A integer-valued matrix. Each value must belong range 0 and <code>a_size - 1</code> . Matrix has dimension $n \times V$, where n is number of samples and V is number of nodes.
<code>tau</code>	A hyperparameter. See references.
<code>max_degree</code>	The maximum length of a candidate Markovian neighborhood. Must be non-negative and less than <code>ncol(sample)</code> .

Value

A adjacency matrix of the estimated Markov random field graph.

Author(s)

Rodrigo Carvalho

References

Guy Bresler. 2015. Efficiently Learning Ising Models on Arbitrary Graphs. In Proceedings of the forty-seventh annual ACM symposium on Theory of Computing (STOC '15). Association for Computing Machinery, New York, NY, USA, 771–782. DOI:<https://doi.org/10.1145/2746539.2746631>

Examples

```
library(mrfse)
a_size = c(0, 1)
s = matrix(sample(a_size, size=1000, replace=TRUE), ncol=5)
mrfse.ci.con(length(a_size), s, 0.2)
```

`mrfse.ci.ncon`*Non-conservative approach for Bresler's non-binary estimator*

Description

A greedy algorithm to estimate Markovian neighborhoods.

Usage

```
mrfse.ci.ncon(a_size, sample, tau, max_degree=ncol(sample)-1)
```

Arguments

<code>a_size</code>	Size of the alphabet.
<code>sample</code>	A integer-valued matrix. Each value must belong range 0 and <code>a_size - 1</code> . Matrix has dimension $n \times V$, where n is number of samples and V is number of nodes.
<code>tau</code>	A hyperparameter. See references.
<code>max_degree</code>	The maximum length of a candidate Markovian neighborhood. Must be non-negative and less than <code>ncol(sample)</code> .

Value

A adjacency matrix of the estimated Markov random field graph.

Author(s)

Rodrigo Carvalho

References

Guy Bresler. 2015. Efficiently Learning Ising Models on Arbitrary Graphs. In Proceedings of the forty-seventh annual ACM symposium on Theory of Computing (STOC '15). Association for Computing Machinery, New York, NY, USA, 771–782. DOI:<https://doi.org/10.1145/2746539.2746631>

Examples

```
library(mrfse)
a_size = c(0, 1)
s = matrix(sample(a_size, size=1000, replace=TRUE), ncol=5)
mrfse.ci.ncon(length(a_size), s, 0.2)
```

mrfse.create.sampler *Create a sampler for Markov random field.*

Description

Create a sampler for Markov random field from a DAG

Usage

```
mrfse.create.sampler(dag_adj, A)
```

Arguments

dag_adj	An direct acyclic graph adjacency matrix
A	Size of alphabet

Value

A list filled with the following components:

neigh: A list of neighborhood. For each i, neigh[[i]] is a markovian neighborhood of vertex i

probs: A list of probabilities. For each i, probs[[i]] is matrix of probabilities of vertex i given your markovian neighborhood. Those probabilities will be used to generate a sample.

moral_adj: moral graph of adj_dag

topol_sort: topological sort of adj_dag

num_nodes: number of nodes de adj_dag

A: alphabet size

Author(s)

Rodrigo Carvalho

Examples

```
library(mrfse)
adj = matrix(c(0, 1, 0, 0, 0, 0, 0, 1, 0), byrow=TRUE, ncol=3)
mrfse.create.sampler(adj, 3)
```

`mrfse.exact`*A Markov random field structure estimator*

Description

A penalized likelihood BIC-based to estimate Markovian neighborhoods.

Usage

```
mrfse.exact(a_size, sample, c, max_neigh= ncol(sample) - 1)
```

Arguments

<code>a_size</code>	Size of the alphabet.
<code>sample</code>	A integer-valued matrix. Each value must belong range 0 and <code>a_size - 1</code> . Matrix has dimension $n \times V$, where n is number of samples and V is number of nodes.
<code>c</code>	The penalization constant. Must be positive.
<code>max_neigh</code>	The maximum length of a candidate Markovian neighborhood. Must be non-negative and less than <code>ncol(sample)</code> .

Value

A list filled with estimated Markov neighborhood for each graph vertex

Author(s)

Rodrigo Carvalho

References

FRONDANA, Iara Moreira. *Model selection for discrete Markov random fields on graphs*. São Paulo : Instituto de Matemática e Estatística, University of São Paulo, 2016. Doctoral Thesis in Estatística. <doi:10.11606/T.45.2018.tde-02022018-151123> http://www.teses.usp.br/teses/disponiveis/45/45133/tde-02022018-151123/publico/tese_Iara_Frondana.pdf

Examples

```
library(mrfse)
a_size = c(0, 1)
s = matrix(sample(a_size, size=1000, replace=TRUE), ncol=5)
mrfse.exact(length(a_size), s, 1.0)
```

mrfse.exact.con *Conservative approach for Frondana's mrfse*

Description

Conservative construction of the estimated Markov random field graph.

Usage

```
mrfse.exact.con(a_size, sample, c, max_neigh = ncol(sample) - 1)
```

Arguments

a_size	Size of the alphabet.
sample	A integer-valued matrix. Each value must belong range 0 and a_size - 1. Matrix has dimension n x V, where n is number of samples and V is number of nodes.
c	The penalization constant. Must be positive.
max_neigh	The maximum length of a candidate Markovian neighborhood. Must be non-negative and less than ncol(sample).

Value

A adjacency matrix of the estimated Markov random field graph.

Author(s)

Rodrigo Carvalho

References

FRONDANA, Iara Moreira. *Model selection for discrete Markov random fields on graphs*. São Paulo : Instituto de Matemática e Estatística, University of São Paulo, 2016. Doctoral Thesis in Estatística. <doi:10.11606/T.45.2018.tde-02022018-151123> http://www.teses.usp.br/teses/disponiveis/45/45133/tde-02022018-151123/publico/tese_Iara_Frondana.pdf

Examples

```
library(mrfse)
a = c(0, 1)
s = matrix(sample(a, size=1000, replace=TRUE), ncol=5)
mrfse.exact.con(length(a), s, 1.0)
```

mrfse.exact.ncon *Non-conservative approach for Frondana's mrfse*

Description

Conservative construction of the estimated Markov random field graph.

Usage

```
mrfse.exact.ncon(a_size, sample, c, max_neigh = ncol(sample) - 1)
```

Arguments

a_size	Size of the alphabet.
sample	A integer-valued matrix. Each value must belong range 0 and a_size - 1. Matrix has dimension n x V, where n is number of samples and V is number of nodes.
c	The penalization constant. Must be positive.
max_neigh	The maximum length of a candidate Markovian neighborhood. Must be non-negative and less than ncol(sample).

Value

A adjacency matrix of the estimated Markov random field graph.

Author(s)

Rodrigo Carvalho

References

FRONDANA, Iara Moreira. *Model selection for discrete Markov random fields on graphs*. São Paulo : Instituto de Matemática e Estatística, University of São Paulo, 2016. Doctoral Thesis in Estatística. <doi:10.11606/T.45.2018.tde-02022018-151123> http://www.teses.usp.br/teses/disponiveis/45/45133/tde-02022018-151123/publico/tese_Iara_Frondana.pdf

Examples

```
library(mrfse)
a = c(0, 1)
s = matrix(sample(a, size=1000, replace=TRUE), ncol=5)
mrfse.exact.ncon(length(a), s, 1.0)
```

mrfse.sa	<i>A Markov random field structure estimator using simulated annealing approach</i>
----------	---

Description

A penalized likelihood BIC-based to estimate Markovian neighborhoods.

Usage

```
mrfse.sa(a_size, sample, c, t0, iterations=1000, max_neigh=ncol(sample)-1)
```

Arguments

a_size	Size of the alphabet.
sample	A integer-valued matrix. Each value must belong range 0 and a_size - 1. Matrix has dimension n x V, where n is number of samples and V is number of nodes.
c	The penalization constant. Must be positive.
t0	Initial temperature
iterations	Number of simulated annealing iterations
max_neigh	The maximum length of a candidate Markovian neighborhood. Must be non-negative and less than ncol(sample).

Value

A list filled with estimated Markov neighborhood for each graph vertex

Author(s)

Rodrigo Carvalho

References

FRONDANA, Iara Moreira. *Model selection for discrete Markov random fields on graphs*. São Paulo : Instituto de Matemática e Estatística, University of São Paulo, 2016. Doctoral Thesis in Estatística. <doi:10.11606/T.45.2018.tde-02022018-151123> http://www.teses.usp.br/teses/disponiveis/45/45133/tde-02022018-151123/publico/tese_Iara_Frondana.pdf

Examples

```
library(mrfse)
a_size = c(0, 1)
s = matrix(sample(a_size, size=1000, replace=TRUE), ncol=5)
mrfse.sa(length(a_size), s, 1.0, 500, 1000)
```

`mrfse.sa.con`*Conservative approach for Frondana's mrfse using simulated annealing*

Description

A penalized likelihood BIC-based to estimate Markovian neighborhoods.

Usage

```
mrfse.sa.con(a_size, sample, c, t0, iterations=1000, max_neigh=ncol(sample)-1)
```

Arguments

<code>a_size</code>	Size of the alphabet.
<code>sample</code>	A integer-valued matrix. Each value must belong range 0 and <code>a_size - 1</code> . Matrix has dimension $n \times V$, where n is number of samples and V is number of nodes.
<code>c</code>	The penalization constant. Must be positive.
<code>t0</code>	Initial temperature
<code>iterations</code>	Number of simulated annealing iterations
<code>max_neigh</code>	The maximum length of a candidate Markovian neighborhood. Must be non-negative and less than <code>ncol(sample)</code> .

Value

A adjacency matrix of the estimated Markov random field graph.

Author(s)

Rodrigo Carvalho

References

FRONDANA, Iara Moreira. *Model selection for discrete Markov random fields on graphs*. São Paulo : Instituto de Matemática e Estatística, University of São Paulo, 2016. Doctoral Thesis in Estatística. <doi:10.11606/T.45.2018.tde-02022018-151123> http://www.teses.usp.br/teses/disponiveis/45/45133/tde-02022018-151123/publico/tese_Iara_Frondana.pdf

Examples

```
library(mrfse)
a_size = c(0, 1)
s = matrix(sample(a_size, size=1000, replace=TRUE), ncol=5)
mrfse.sa.con(length(a_size), s, 1.0, 500, 1000)
```

mrfse.sa.ncon	<i>Non-conservative approach for Frondana's mrfse using simulated annealing</i>
---------------	---

Description

A penalized likelihood BIC-based to estimate Markovian neighborhoods.

Usage

```
mrfse.sa.ncon(a_size, sample, c, t0, iterations=1000, max_neigh=ncol(sample)-1)
```

Arguments

a_size	Size of the alphabet.
sample	A integer-valued matrix. Each value must belong range 0 and a_size - 1. Matrix has dimension n x V, where n is number of samples and V is number of nodes.
c	The penalization constant. Must be positive.
t0	Initial temperature
iterations	Number of simulated annealing iterations
max_neigh	The maximum length of a candidate Markovian neighborhood. Must be non-negative and less than ncol(sample).

Value

A adjacency matrix of the estimated Markov random field graph.

Author(s)

Rodrigo Carvalho

References

FRONDANA, Iara Moreira. *Model selection for discrete Markov random fields on graphs*. São Paulo : Instituto de Matemática e Estatística, University of São Paulo, 2016. Doctoral Thesis in Estatística. <doi:10.11606/T.45.2018.tde-02022018-151123> http://www.teses.usp.br/teses/disponiveis/45/45133/tde-02022018-151123/publico/tese_Iara_Frondana.pdf

Examples

```
library(mrfse)
a_size = c(0, 1)
s = matrix(sample(a_size, size=1000, replace=TRUE), ncol=5)
mrfse.sa.ncon(length(a_size), s, 1.0, 500, 1000)
```

`mrfse.sample`*Generate a independent sample of a Markov random field*

Description

Generate a independent sample of a Markov random field according to the probabilities of the sampler.

Usage

```
mrfse.sample(sampler, n)
```

Arguments

<code>sampler</code>	A sampler created by <code>mrfse.create.sampler</code> function
<code>n</code>	Size of sample

Value

A matrix whose number of columns is the number of nodes. Each line is a single independent sample of Markov random field given by the probabilities of sampler.

Author(s)

Rodrigo Carvalho

Examples

```
library(mrfse)
adj = matrix(c(0, 1, 0, 0, 0, 0, 0, 1, 0), byrow=TRUE, ncol=3)
sampler = mrfse.create.sampler(adj, 3)
mrfse.sample(sampler, 3000)
```

Index

`mrfse.ci`, [2](#)
`mrfse.ci.con`, [3](#)
`mrfse.ci.ncon`, [4](#)
`mrfse.create.sampler`, [5](#)
`mrfse.exact`, [6](#)
`mrfse.exact.con`, [7](#)
`mrfse.exact.ncon`, [8](#)
`mrfse.sa`, [9](#)
`mrfse.sa.con`, [10](#)
`mrfse.sa.ncon`, [11](#)
`mrfse.sample`, [12](#)