Package 'ccdrAlgorithm'

August 29, 2016

Title CCDr Algorithm for Learning Sparse Gaussian Bayesian Networks

Version 0.0.1

Date 2016-08-08

Maintainer Bryon Aragam <sparsebn@gmail.com>

Description Implementation of the CCDr (Concave penalized Coordinate Descent with reparametrization) structure learning algorithm as described in Aragam and Zhou (2015) <http://www.jmlr.org/papers/v16/aragam15a.html>. This is a fast, score-based method for learning Bayesian networks that uses sparse regularization and block-cyclic coordinate descent.

Depends R (>= 3.2.3)

Imports sparsebnUtils, Rcpp (>= 0.11.0)

LinkingTo Rcpp

Suggests testthat, graph

URL https://github.com/itsrainingdata/ccdrAlgorithm

BugReports https://github.com/itsrainingdata/ccdrAlgorithm/issues

License GPL (>= 2)

RoxygenNote 5.0.1

NeedsCompilation yes

Author Bryon Aragam [aut, cre]

Repository CRAN

Date/Publication 2016-08-10 14:16:05

R topics documented:

ccdr.run *Main CCDr Algorithm*

Description

Estimate a Bayesian network (directed acyclic graph) from observational data using the CCDr algorithm as described in Aragam and Zhou (2015).

Usage

```
ccdr.run(data, betas, lambdas = NULL, lambdas.length = NULL, gamma = 2,
  error.tol = 1e-04, max.iters = NULL, alpha = 10, verbose = FALSE)
```

Arguments

data	Data as sparsebnData. Must be numeric and contain no missing values.
betas	Initial guess for the algorithm. Represents the weighted adjacency matrix of a DAG where the algorithm will begin searching for an optimal structure.
lambdas	(optional) Numeric vector containing a grid of lambda values (i.e. regularization parameters) to use in the solution path. If missing, a default grid of values will be used based on a decreasing log-scale (see also generate.lambdas).
lambdas.length	Integer number of values to include in the solution path. If lambdas has also been specified, this value will be ignored. Note also that the final solution path may contain fewer estimates (see alpha).
gamma	Value of concavity parameter. If gamma > 0, then the MCP will be used with gamma as the concavity parameter. If gamma < 0, then the L1 penalty will be used and this value is otherwise ignored.
error.tol	Error tolerance for the algorithm, used to test for convergence.
max.iters	Maximum number of iterations for each internal sweep.
alpha	Threshold parameter used to terminate the algorithm whenever the number of edges in the current DAG estimate is > alpha * ncol(data).
verbose	TRUE / FALSE whether or not to print out progress and summary reports.

Details

Instead of producing a single estimate, this algorithm computes a solution path of estimates based on the values supplied to lambdas or lambdas.length. The CCDr algorithm approximates the solution to a nonconvex optimization problem using coordinate descent. Instead of AIC or BIC, CCDr uses continuous regularization based on concave penalties such as the minimax concave penalty (MCP).

This implementation includes two options for the penalty: (1) MCP, and (2) L1 (or Lasso). This option is controlled by the gamma argument.

Value

A sparsebnPath object.

Examples

```
## Not run:

### Generate some random data
dat <- matrix(rnorm(1000), nrow = 20)
dat <- sparsebnData(dat, type = "continuous")

# Run with default settings
ccdr.run(data = dat)

### Optional: Adjust settings
pp <- ncol(dat)

# Initialize algorithm with a random initial value
init.betas <- matrix(0, nrow = pp, ncol = pp)
init.betas[1,2] <- init.betas[1,3] <- init.betas[4,2] <- 1

# Run with adjusted settings
ccdr.run(data = dat, betas = init.betas, lambdas.length = 10, alpha = 10, verbose = TRUE)

## End(Not run)
```

---

ccdrAlgorithm *ccdrAlgorithm: CCDr Algorithm for Learning Sparse Gaussian Bayesian Networks*

Description

Implements the CCDr structure learning algorithm as described in Aragam and Zhou (2015).

Details

The CCDr algorithm uses sparse regularization (L1 or MCP) to produce a solution path of DAG estimates along a pre-determined grid of hyperparameters. This package implements a single function, ccdr.run that runs the main algorithm, and uses sparsebnUtils for the underlying data structures and methods.

Index