# Package ‘dplbnDE’

October 13, 2022

**Type** Package

**Title** Discriminative Parameter Learning of Bayesian Networks by Differential Evolution

**Version** 0.1.1

**Maintainer** Alejandro Platas-Lopez <alejandroplatasl@gmail.com>

**BugReports** https://github.com/alexplatasl/dplbnDE/issues

**Description** Implements Differential Evolution (DE) to train parameters of Bayesian Networks for optimizing the Conditional Log-Likelihood (Discriminative Learning) instead of the log-likelihood (Generative Learning). Any given Bayesian Network structure encodes assumptions about conditional independencies among the attributes and will result in an error if they do not hold in the data. Such an error includes the classification dimension. The main goal of Discriminative learning is to minimize this type of error. This package provides main variants of differential evolution described in Price & Storn (1996) <doi:10.1109/ICEC.1996.542711> and recent ones, described in Tanabe & Fukunaga (2014) <doi:10.1109/CEC.2014.6900380> and Zhang & Sanderson (2009) <doi:10.1109/TEVC.2009.2014613> with adaptation mechanism for factor mutarion and crossover rate.

**License** GPL (>= 2)

**Encoding** UTF-8

**Depends** R (>= 3.2.0)

**Imports** bnclassify (>= 0.4.5),

**LazyData** true

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Author** Alejandro Platas-Lopez [aut, cre] (<https://orcid.org/0000-0002-2584-343X>), Alejandro Guerra-Hernandez [ths] (<https://orcid.org/0000-0002-4856-4011>)

**Repository** CRAN

**Date/Publication** 2022-01-10 17:42:44 UTC

# R topics documented:



---

`car` *Car Evaluation Data Set.*

---

### Description

Data set from the UCI repository: https://archive.ics.uci.edu/ml/datasets/Car+Evaluation.

### Format

A `data.frame` with 7 columns and 1728 rows.

---

`DE` *Discriminative parameter learning of bayesian networks by differential evolution*

---

### Description

A list with Bayesian Networks, structure and parameters learned in a discriminative way by diferential evolution returned by `lshade`, `jade`, `DEbest`, or `DErand` functions.

### Examples

```
data(car)
dpl.lshade <- lshade(NP=40, G=50, data = car, class.name = names(car)[7], c = 0.1,
structure = "tan", pB=0.05, edgelist = NULL, verbose = 5)
dpl.lshade
```

DEbest *Discriminative parameter learning of BN by DE variant best/k/.*

## Description

Learn parameters of a Bayesian Network in a discriminative way by Differential Evolution with variant best/k/

## Usage

```
DEbest(
  NP = 40,
  G = 100,
  data,
  class.name,
  F = 0.5,
  CR = 0.7,
  mutation.pairs = c(1, 2),
  crossover = c("bin", "exp"),
  structure = c("nb", "tancl", "hc"),
  edgelist = NULL,
  verbose = 25,
  ...
)
```

## Arguments

| | |
|---|---|
| NP | positive integer giving the number of candidate solutions in the initial population. |
| G | positive integer specifying the maximum number of generations that may be performed before the algorithm is halted. |
| data | The data frame from which to learn the classifier. |
| class.name | A character. Name of the class variable. |
| F | A numeric. Mutation factor. Default is 0.5. |
| CR | A numeric. Cross over factor. Default is 0.7. |
| mutation.pairs | A positive integer giving the number of pairs (1 or 2) used in the mutation step. |
| crossover | A character. Crossover type among binomial (bin) or exponential (exp). |
| structure | A character. Name of the structure learning function. "tan" uses Tree Augmented Network. "nb" uses Naive Bayes. "hc" uses Hill Climbing. |
| edgelist | A matrix. An optional edge list to use a custom BN structure. that will replace de learned structure. |
| verbose | positive integer indicating the number of generations until the iteration progress should be printed. |
| ... | other structure learning options from tan_cl or tan_hc. |

## Value

An object of class `DE`, which is a list with the following components:

| | |
|---|---|
| `Best` | A `bnc_bn` object with the best individual in the final population, i.e., the bayesian network with the best fitness at the end of evolution. |
| `BestCLL` | A numeric specifying the Conditional Log-Likelihood of the best individual. |
| `pobFinal` | A list of `bnc_bn` objects with the final population, i.e., a set of bayesian networks with optimized parameters at the end of evolution. |
| `CLLPobFinal` | A numeric vector specifying the Conditional Log-Likelihood of the final population. |
| `N.evals` | An integer giving the total number of evaluations. |
| `convergence` | A numeric vector giving the maximum Conditional Log-Likelihood at each generation. |
| `evaluations` | An integer vector giving the total number of evaluations at each generation. |

## Examples

```
# Load data
data(car)
# Parameter learning with "best/1/bin" variant
dpl.best1bin <- DEbest(NP=25, G=35, data = car, class.name = names(car)[7], F = 0.5,
CR = 0.7, mutation.pairs = 1, crossover = "bin", structure = "tan", edgelist = NULL,
verbose = 10)
# Print results
print(dpl.best1bin)
## Not run: plot(dpl.best1bin)
```

---

`DErand` *Discriminative parameter learning of BN by DE variant rand/k/.*

---

## Description

Learn parameters of a Bayesian Network in a discriminative way by Differential Evolution with variant rand/k

## Usage

```
DErand(
  NP = 40,
  G = 100,
  data,
  class.name,
  F = 0.5,
  CR = 0.7,
  mutation.pairs = c(1, 2),
  crossover = c("bin", "exp"),
```

```
  structure = c("nb", "tancl", "hc"),
  edgelist = NULL,
  verbose = 25,
  ...
)
```

## Arguments

| | |
|---|---|
| NP | positive integer giving the number of candidate solutions in the initial population. |
| G | positive integer specifying the maximum number of generations that may be performed before the algorithm is halted. |
| data | The data frame from which to learn the classifier. |
| class.name | A character. Name of the class variable. |
| F | A numeric. Mutation factor. Default is 0.5. |
| CR | A numeric. Cross over factor. Default is 0.7. |
| mutation.pairs | A positive integer giving the number of pairs (1 or 2) used in the mutation step. |
| crossover | A character. Crossover type among binomial (bin) or exponential (exp). |
| structure | A character. Name of the structure learning function. "tan" uses Tree Augmented Network. "nb" uses Naive Bayes. "hc" uses Hill Climbing. |
| edgelist | A matrix. An optional edge list to use a custom BN structure. that will replace de learned structure. |
| verbose | positive integer indicating the number of generations until the iteration progress should be printed. |
| ... | other structure learning options from tan_cl or tan_hc. |

## Value

An object of class DE, which is a list with the following components:

| | |
|---|---|
| Best | A bnc_bn object with the best individual in the final population, i.e., the bayesian network with the best fitness at the end of evolution. |
| BestCLL | A numeric specifying the Conditional Log-Likelihood of the best individual. |
| pobFinal | A list of bnc_bn objects with the final population, i.e., a set of bayesian networks with optimized parameters at the end of evolution. |
| CLLPobFinal | A numeric vector specifying the Conditional Log-Likelihood of the final population. |
| N.evals | An integer giving the total number of evaluations. |
| convergence | A numeric vector giving the maximum Conditional Log-Likelihood at each generation. |
| evaluations | An integer vector giving the total number of evaluations at each generation. |

### Examples

```
# Load data
data(car)
# Parameter learning with "rand/2/bin" variant
dpl.rand2bin <- DErand(NP=25, G=40, data = car, class.name = names(car)[7], F = 0.5,
CR = 0.5, mutation.pairs = 2, crossover = "bin", structure = "tan",edgelist = NULL,
verbose = 10)
# Print results
print(dpl.rand2bin)
## Not run: plot(dpl.rand2bin)
```

---

## dplbnDE — *Discriminative Parameter Learning of Bayesian Networks by Differential Evolution*

---

### Description

Implements Differential Evolution (DE) to train parameters of Bayesian Networks (BN) for optimizing the Conditional Log-Likelihood (Discriminative Learning) instead of the log-likelihood (Generative Learning). Any given BN structure encodes assumptions about conditional independencies among the attributes and will result in error if they do not hold in the data. Such an error includes the classification dimension. The main goal of Discriminative learning is minimize this type of error.

### Details

DE variants: Based on different strategies followed by the operators of DE, there are different variants, which define the way in which the mutant and trial vectors are generated. The most popular variant is called `DE/rand/1/bin`, where "DE" means Differential Evolution, the word "rand" indicates that the so-called base vector is randomly chosen, "1" is the number of vector pairs (i.e., vector differences to be calculated) chosen, and finally "bin" means that a binomial recombination is chosen. The following is a list with the available variants:

- `DErand`: Implements `DE/rand/` variant with 1 or 2 pairs of vector differences, and binomial or exponential recombination. (Price and Storn, 1996)
- `DEbest`: Implements `DE/best/` variant with 1 or 2 pairs of vector differences and binomial or exponential recombination. (Price and Storn, 1996)
- `jade`: A variant that includes some mechanisms to decrease the dependence to its parameter values such as F and CR. JADE uses a mutation strategy called `DE/current-to-pbest`, where $p \in (0, 1]$. Base vectors are selected from the best $100p$ maintaining diversity, uses an optional external archive. (Zhang and Sanderson, 2009)
- `lshade`: An improved version of JADE, LSHADE incorporates a Linear Population Size Reduction (LPSR) in order to improve the performance. (Tanabe and Fukunaga, 2014)

### References

Price K and Storn R (1996), Minimizing the real functions of the icec'96 contest by differential evolution. In *Proc. of IEEE C. Evol. Computat.*, pp. 842–844.

Zhang J and Sanderson A (2009). Jade: adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.*, pp. 945–958.

Tanabe R and Fukunaga A (2014). Improving the search performance of shade using linear population size reduction. In *Proc. of IEEE C. Evol. Computat.*, pp. 1658–1665.

---

## jade — *Discriminative parameter learning of BN by JADE.*

---

### Description

Learn parameters of a Bayesian Network in a discriminative way by Adaptive Differential Evolution with optional external Archive

### Usage

```
jade(
  NP = 40,
  G = 100,
  data,
  class.name,
  c = 0.1,
  structure = c("nb", "tancl", "hc"),
  pB = 0.05,
  edgelist = NULL,
  archive = TRUE,
  verbose = 25,
  ...
)
```

### Arguments

| | |
|---|---|
| `NP` | positive integer giving the number of candidate solutions in the initial population. |
| `G` | positive integer specifying the maximum number of generations that may be performed before the algorithm is halted. |
| `data` | The data frame from which to learn the classifier. |
| `class.name` | A character. Name of the class variable. |
| `c` | A numeric. An adaptation parameter. Default is 0.1. |
| `structure` | A character. Name of the structure learning function. "tan" uses Tree Augmented Network. "nb" uses Naive Bayes. "hc" uses Hill Climbing. |
| `pB` | A numeric. JADE mutation strategy. |

| | |
|---|---|
| edgelist | A matrix. An optional edge list to use a custom BN structure that will replace de learned structure. |
| archive | A logical. If TRUE, trial vector r2 is randomly selected from the union of the current population and the external archive. |
| verbose | positive integer indicating the number of generations until the iteration progress should be printed. |
| ... | other structure learning options from tan_cl or tan_hc. |

## Value

An object of class DE, which is a list with the following components:

| | |
|---|---|
| Best | A bnc_bn object with the best individual in the final population, i.e., the bayesian network with the best fitness at the end of evolution. |
| BestCLL | A numeric specifying the Conditional Log-Likelihood of the best individual. |
| pobFinal | A list of bnc_bn objects with the final population, i.e., a set of bayesian networks with optimized parameters at the end of evolution. |
| CLLPobFinal | A numeric vector specifying the Conditional Log-Likelihood of the final population. |
| N.evals | An integer giving the total number of evaluations. |
| convergence | A numeric vector giving the maximum Conditional Log-Likelihood at each generation. |
| evaluations | An integer vector giving the total number of evaluations at each generation. |

## Examples

```
# Load data
data(car)
# Parameter learning with "JADE with Archive" variant, and structure with
# hill-climbing algorithm, so argument "k" must be provided.
dpl.jade <- jade(NP=40, G=50, data = car, class.name = names(car)[7], c = 0.1,
structure = "hc", pB=0.05, edgelist = NULL, archive = TRUE, verbose = 5, k = 3)
# Print results
print(dpl.jade)
## Not run: plot(dpl.jade)
```

---

lshade *Discriminative parameter learning of BN by L-SHADE.*

---

## Description

Learn parameters of a Bayesian Network in a discriminative way by Success-History based Adaptive Differential evolution with a Linear Population Size Reduction

## Usage

```
lshade(
  NP = 40,
  G = 100,
  data,
  class.name,
  c = 0.1,
  structure = c("nb", "tancl", "hc"),
  pB = 0.05,
  edgelist = NULL,
  verbose = 25,
  ...
)
```

## Arguments

| | |
|---|---|
| `NP` | positive integer giving the number of candidate solutions in the initial population. |
| `G` | positive integer specifying the maximum number of generations that may be performed before the algorithm is halted. |
| `data` | The data frame from which to learn the classifier. |
| `class.name` | A character. Name of the class variable. |
| `c` | Numeric. An adaptation parameter. Default is 0.1. |
| `structure` | A character. Name of the structure learning function. "tan" uses Tree Augmented Network. "nb" uses Naive Bayes. "hc" uses Hill Climbing. |
| `pB` | Numeric. JADE mutation strategy. |
| `edgelist` | A matrix. An optional edge list to use a custom BN structure that will replace de learned structure. |
| `verbose` | positive integer indicating the number of generations until the iteration progress should be printed. |
| `...` | other structure learning options from tan_cl or tan_hc. |

## Value

An object of class `DE`, which is a list with the following components:

| | |
|---|---|
| `Best` | A `bnc_bn` object with the best individual in the final population, i.e., the bayesian network with the best fitness at the end of evolution. |
| `BestCLL` | A numeric specifying the Conditional Log-Likelihood of the best individual. |
| `pobFinal` | A list of `bnc_bn` objects with the final population, i.e., a set of bayesian networks with optimized parameters at the end of evolution. |
| `CLLPobFinal` | A numeric vector specifying the Conditional Log-Likelihood of the final population. |
| `N.evals` | An integer giving the total number of evaluations. |

| | |
|---|---|
| convergence | A numeric vector giving the maximum Conditional Log-Likelihood at each generation. |
| evaluations | An integer vector giving the total number of evaluations at each generation. |

## Examples

```
# Load data
data(car)
# Parameter learning with "LSHADE" variant
dpl.lshade <- lshade(NP=40, G=50, data = car, class.name = names(car)[7], c = 0.1,
structure = "tan", pB=0.05, edgelist = NULL, verbose = 5)
# Print results
print(dpl.lshade)
## Not run: plot(dpl.lshade)
```

---

plot.DE *Plot main results of evolution*

---

## Description

Plot main results of evolution

## Usage

```
## S3 method for class 'DE'
plot(x, ...)
```

## Arguments

| | |
|---|---|
| x | A list of structures with parameters learned by DE. |
| ... | further arguments passed to plot. |

## Value

Nothing. Side-effect: plots graphs.

# Index