Package 'DistributionOptimization'

October 12, 2022

Type Package

Title Distribution Optimization

Version 1.2.6

Author Florian Lerch, Jorn Lotsch, Alfred Ultsch

Maintainer Florian Lerch <lerch@mathematik.uni-marburg.de>

Description Fits Gaussian Mixtures by applying evolution. As fitness function a mixture of the chi square test for distributions and a novel measure for approximating the common area under curves between multiple Gaussians is used. The package presents an alternative to the commonly used Likelihood Maximization as is used in Expectation Maximization. The algorithm and applications of this package are published under: Lerch, F., Ultsch, A., Lotsch, J. (2020) <doi:10.1038/s41598-020-57432-w>. The evolution is based on the 'GA' package: Scrucca, L. (2013) <doi:10.18637/jss.v053.i04> while the Gaussian Mixture Logic stems from 'AdaptGauss': Ultsch, A, et al. (2015) <doi:10.3390/ijms161025897>.

Imports ggplot2, GA, AdaptGauss, graphics, stats, utils, pracma

Suggests parallelDist

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-02-12 15:40:02 UTC

R topics documented:

DistributionOptimization-package 2
BinProb4Mixtures 2
DistributionOptimization 3
MixedDistributionError 5
OverlapErrorByDensity 6

Index **7**

DistributionOptimization-package *Distribution Optimization*

Description

Distribution Optimization fits gaussian mixture models on to one dimensional samples by minimizing the Chi Squared Error by evolutional optimization. It is an alternative to likelihood maximizers like expection maximization. Through the included "Overlapping" Methods, single gaussians can be forced to be separated, achieving various signifcant models to choose from. The evolutionary part is done through the "GA" Package. The Gaussian Mixture Logic is based on the "AdaptGauss" Package.

Author(s)

Florian Lerch, Jorn Lotsch, Alfred Ultsch

References

Luca Scrucca (2013). GA: A Package for Genetic Algorithms in R. Journal of Statistical Software, 53(4), 1-37. URL http://www.jstatsoft.org/v53/i04/

BinProb4Mixtures *Bin Probabilities*

Description

Calculates the probability of bins/intervals within the dataspace defined by given breaks between them.

Usage

```
BinProb4Mixtures(Means, SDs, Weights, Breaks, IsLogDistribution = rep(F,
  length(Means)), LimitsAreFinite = T)
```

Arguments

Means	Means of the GMM Components
SDs	Standard Deviations of the GMM Components
Weights	Weights of the GMM Components
Breaks	Breaks Defining c-1 or c+1 bins (depending on LimitsAreFinite)
IsLogDistribution	If True, the GMM is interpreted as a logarithmic
LimitsAreFinite	If True, there are c+1 Bins, where the first and last bin are of inifinite size

Value

Probabalities of either c-1 or c+1 bins/intervals (depending on LimitsAreFinite)

Author(s)

Florian Lerch

Examples

```
Data = c(rnorm(50,1,2), rnorm(50,3,4))
NoBins = 20
breaks = seq(min(Data),max(Data), length.out=length(NoBins)+1)
BinProb4Mixtures(c(1,3), c(2,4), c(0.5,0.5), breaks)
```

DistributionOptimization

Distribution Fitting

Description

Fits a Gaussian Mixture Model onto a Dataset by minimizing a fitting error through evolutionary optimization. Every individual encodes one GMM. Details over the evolutionary process itself can be taken from the 'GA' package. ga

Usage

```
DistributionOptimization(Data, Modes, Monitor = 1,
  SelectionMethod = "UnbiasedTournament",
  MutationMethod = "Uniform+Focused",
  CrossoverMethod = "WholeArithmetic", PopulationSize = Modes * 3 * 25,
  MutationRate = 0.7, Elitism = 0.05, CrossoverRate = 0.2,
  Iter = Modes * 3 * 200, OverlapTolerance = NULL,
  IsLogDistribution = rep(F, Modes), ErrorMethod = "chisquare",
  NoBins = NULL, Seed = NULL, ConcurrentInit = F, ParetoRad = NULL)
```

Arguments

Data	Data to be modelled
Modes	Number of expected Modes
Monitor	0:no monitoring, 1: status messages, 2: status messages and plots, 3: status messages, plots and calculated error-measures
SelectionMethod	1: LinearRank selection 4: UnbiasedTournament 5: FitnessProportional selection

MutationMethod	1: UniformRandom mutation 2: NonuniformRandom mutation 4: Focused mutation, alternative random mutation around solution 5: GaussMutationCust 6: TwoPhaseMutation - mutation is uniform random during the first half of iterations, and than focuses around current solution
CrossoverMethod	1: single point crossover 2: whole arithmetic crossover 3: local arithmetic crossover 4: blend crossover 5: GaussCrossover - exchange complete gaussian components 6: MultiPointCrossover - Random amount of information between mixtures get exchanged
PopulationSize	Size of the population
MutationRate	amount (0..1) of population that gets mutated
Elitism	amount of best individuals that will survive generation unchanged
CrossoverRate	amount of individuals that will be used for crossover
Iter	number of iterations of this algorithm
OverlapTolerance	ratio between Chi-Square and OverlapError (only if FitnessMethod = 4 (Chi2ValueWithOverlap))
IsLogDistribution	which gauss components should be considered as log gaussian
ErrorMethod	"pde": fitting is measured by pareto density estimation. "chisquare": fitting is measured by Chi-Square test
NoBins	Number of Bins that will be used for evaluating fitting
Seed	Random Seed for reproducible results
ConcurrentInit	If true, before initialization a number of short optimizations are done to find a good starting point for evolution
ParetoRad	Pareto Radius for Pareto Density Estimation and its plots

Value

The GA object containing the evolutionary training and a description of the final GMM consisting of means, sdevs and weights.

Author(s)

Florian Lerch

Jorn Lotsch

Alfred Ultsch

Examples

```
## Not run:
DistributionOptimization(c(rnorm(200),rnorm(200,3), 2))

## End(Not run)
```

MixedDistributionError
MixedDistributionError

Description

Calculates a fitting error as well as the overlapping measure for the mixtures. Combines them with ratio rho in favor of Overlapping.

Usage

```
MixedDistributionError(Means, SDs, Weights, Data, rho = 0.5,
  breaks = NULL, Kernels = NULL, ErrorMethod = "chisquare")
```

Arguments

Means	Means of the GMM Components
SDs	Standard Deviations of the GMM Components
Weights	Weights of the GMM Components
Data	Empirical Data based on which the GMM is build
rho	Ratio of OverlappingError vs Fitting Error
breaks	vector containing the breaks between bins
Kernels	positions at which density is to be compared
ErrorMethod	"pdeerror": fitting error is measured through Pareto Density Estimation. "chisquare": fitting error is measured through the Chi Square fitting error.

Value

Mixed Error

Author(s)

Florian Lerch

Examples

```
Data = c(rnorm(50,1,2), rnorm(50,3,4))
MixedDistributionError(c(1,3), c(2,4), c(0.5,0.5), Data = Data)
```

OverlapErrorByDensity *OverlapErrorByDensity*

Description

Similarity in GMM by Density

Usage

```
OverlapErrorByDensity(Means, SDs, Weights, Data = NULL, Kernels = NULL)
```

Arguments

Means	Means of the GMM Components
SDs	Standard Deviations of the GMM Components
Weights	Weights of the GMM Components
Data	Dataset that the GMM should be compared with
Kernels	if length(Kernels) = 1: amount of kernels if length(Kernels) > 1: kernels in dataspace at which the GMM Components will be compared with each other

Details

Calculates the similarity (overlap) between multiple modes in Gaussian Mixture Models. Kernels at equally distanced positions are used, if not explicitly given.

Value

List: OverlapError Error for estimating the maximal Overlap of AUC of PDFs of each pair of GMM Components Kernels Kernels that were used for comparing the GMM Components

Author(s)

Florian Lerch

Examples

```
Data = c(rnorm(50,1,2), rnorm(50,3,4))
V<-OverlapErrorByDensity(c(1,3), c(2,4), c(0.5,0.5), Data = Data)
AdaptGauss::PlotMixtures(Data, c(1,3), c(2,4), SingleGausses = TRUE)
print(V$OverlapError)
```

Index

∗ **EM**
DistributionOptimization-package, 2
∗ **GMM**
DistributionOptimization-package, 2
∗ **gaussian mixture model**
DistributionOptimization-package, 2
∗ **pareto density estimation**
DistributionOptimization-package, 2
∗ **pde**
DistributionOptimization-package, 2

BinProb4Mixtures, 2

DistributionOptimization, 3
DistributionOptimization-package, 2

ga, *3*

MixedDistributionError, 5

OverlapErrorByDensity, 6