Package 'naspaclust'

October 13, 2022

Type Package

Title Nature-Inspired Spatial Clustering

Version 0.2.1

Suggests ppclust, spatialClust, cluster, ggplot2

Imports Rdpack, rdist, stabledist, beepr

RdMacros Rdpack

Author Bahrul Ilmi Nasution [aut, cre],
Robert Kurniawan [aut],
Rezzy Eko Caraka [aut]

Maintainer Bahrul Ilmi Nasution <bahrulnst@gmail.com>

Description Implement and enhance the performance of spatial fuzzy clustering using Fuzzy Geographically Weighted Clustering with various optimization algorithms, mainly from Xin She Yang (2014) <ISBN:9780124167438> with book entitled Nature-Inspired Optimization Algorithms. The optimization algorithm is useful to tackle the disadvantages of clustering inconsistency when using the traditional approach. The distance measurements option is also provided in order to increase the quality of clustering results. The Fuzzy Geographically Weighted Clustering with nature inspired optimisation algorithm was firstly developed by Arie Wahyu Wijayanto and Ayu Purwarianti (2014) <doi:10.1109/CITSM.2014.7042178> using Artificial Bee Colony algorithm.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2021-07-07 09:40:02 UTC

R topics documented:

`abcfgwc` *Fuzzy Geographicaly Weighted Clustering with Artificial Bee Colony Optimization*

Description

Fuzzy clustering with addition of spatial configuration of membership matrix with centroid optimization using Artificial Bee Colony

Usage

```
abcfgwc(
  data,
  pop = NA,
  distmat = NA,
  ncluster = 2,
  m = 2,
  distance = "euclidean",
  order = 2,
  alpha = 0.7,
  a = 1,
  b = 1,
  error = 1e-05,
  max.iter = 100,
  randomN = 0,
  vi.dist = "uniform",
  nfood = 10,
  n.onlooker = 5,
  limit = 4,
  pso = F,
  abc.same = 10
)
```

Arguments

data — an object of data with d>1. Can be matrix or data.frame. If your data is univariate, bind it with 1 to get a 2 columns.

pop — an n*1 vector contains population.

distmat — an n*n distance matrix between regions.

ncluster — an integer. The number of clusters.

m — degree of fuzziness or fuzzifier. Default is 2.

distance — the distance metric between data and centroid, the default is euclidean, see cdist for details.

order, — minkowski order. default is 2.

alpha — the old membership effect with [0,1], if alpha equals 1, it will be same as fuzzy C-Means, if 0, it equals to neighborhood effect.

a — spatial magnitude of distance. Default is 1.

b — spatial magnitude of population. Default is 1.

error — error tolerance. Default is 1e-5.

max.iter — maximum iteration. Default is 500.

randomN — random seed for initialisation (if uij or vi is NA). Default is 0.

vi.dist — a string of centroid population distribution between "uniform" (default) and "normal". Can be defined as vi.dist= in opt_param.

nfood — number of foods population. Can be defined as npar= in opt_param.

n.onlooker — number of onlooker bees, Can be defined as n.onlooker in opt_param.

limit — number of turns to eliminate food with no solutions. Can be defined as limit in opt_param.

pso — whether to add PSO term in bee's movement. Either TRUE or FALSE. Can be defined as pso in opt_param.

abc.same — number of consecutive unchange to stop the iteration. Can be defined as same= in opt_param.

Details

Fuzzy Geographically Weighted Clustering (FGWC) was developed by Mason and Jacobson (2007) by adding neighborhood effects and population to configure the membership matrix in Fuzzy C-Means. Furthermore, the Artificial Bee Colony (ABC) algorithm was developed by Karaboga and Basturk (2007) in order to get a more optimal solution of a certain complex function. FGWC using ABC has been implemented previously by Wijayanto and Purwarianti (2014) and Wijayanto et al. (2016).

Value

an object of class "fgwc".
An "fgwc" object contains as follows:

- converg - the process convergence of objective function

- f_obj - objective function value
- membership - membership matrix
- centroid - centroid matrix
- validation - validation indices (there are partition coefficient (PC), classification entropy (CE), SC index (SC), separation index (SI), Xie and Beni's index (XB), IFV index (IFV), and Kwon index (Kwon))
- max.iter - Maximum iteration
- cluster - the cluster of the data
- finaldata - The final data (with the cluster)
- call - the syntax called previously
- time - computational time.

References

Karaboga D, Basturk B (2007). "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm." *Journal of Global Optimization*, **39**(3), 459–471. doi: 10.1007/s108980079149x, https://doi.org/10.1007/s10898-007-9149-x.

Mason GA, Jacobson RD (2007). "Fuzzy Geographically Weighted Clustering." In *Proceedings of the 9th International Conference on Geocomputation*, 1–7.

Wijayanto AW, Purwarianti A (2014). "Improvement design of fuzzy geo-demographic clustering using Artificial Bee Colony optimization." In *2014 International Conference on Cyber and IT Service Management (CITSM)*, 69–74. ISBN 978-1-4799-7975-2.

Wijayanto AW, Purwarianti A, Son LH (2016). "Fuzzy geographically weighted clustering using artificial bee colony: An efficient geo-demographic analysis algorithm and applications to the analysis of crime behavior in population." *Applied Intelligence*, **44**(2), 377–398. ISSN 0924-669X.

See Also

fpafgwc gsafgwc

Examples

```
data('census2010')
data('census2010dist')
data('census2010pop')
# First way
res1 <- abcfgwc(census2010,census2010pop,census2010dist,3,2,'euclidean',4,nfood=10)
# Second way
# initiate parameter
param_fgwc <- c(kind='v',ncluster=3,m=2,distance='minkowski',order=3,
               alpha=0.5,a=1.2,b=1.2,max.iter=1000,error=1e-6,randomN=10)
## tune the ABC parameter
abc_param <- c(vi.dist='normal',npar=5,pso=FALSE,same=15,n.onlooker=5,limit=5)
##FGWC with ABC optimization algorithm
res2 <- fgwc(census2010,census2010pop,census2010dist,'abc',param_fgwc,abc_param)
```

census2010 *Indonesia 2010 Provincial Census data*

Description

A dataset containing the demographic characteristics of Indonesia on provincial level from BPS-Statistics Indonesia

Usage

```
census2010
```

Format

An object of class `data.frame` with 33 rows and 110 columns.

Details

A data frame with 33 rows and 110 demographic characteristics.
The dataset was used in Wijayanto (2015), Wijayanto and Purwarianti (2014), Wijayanto and Purwarianti (2014) and Wijayanto et al. (2016).
The details of each variable can be seen in Wijayanto (2015),

References

Wijayanto AW (2015). *Improvement of fuzzy geo-demographic clustering using metaheuristic optimization on Indonesia population census*. Master's thesis, Insitut Teknologi Bandung.

Wijayanto AW, Purwarianti A (2014). "Improvement of fuzzy geographically weighted clustering using particle swarm optimization." In *2014 International Conference on Information Technology Systems and Innovation (ICITSI)*, 7–12. ISBN 978-1-4799-6527-4.

Wijayanto AW, Purwarianti A (2014). "Improvement design of fuzzy geo-demographic clustering using Artificial Bee Colony optimization." In *2014 International Conference on Cyber and IT Service Management (CITSM)*, 69–74. ISBN 978-1-4799-7975-2.

Wijayanto AW, Purwarianti A, Son LH (2016). "Fuzzy geographically weighted clustering using artificial bee colony: An efficient geo-demographic analysis algorithm and applications to the analysis of crime behavior in population." *Applied Intelligence*, **44**(2), 377–398. ISSN 0924-669X.

`census2010dist` *Indonesia Provincial Matrix Distance*

Description

A dataset containing the calculated distance between provinces in Indonesia.

Usage

```
census2010dist
```

Format

An object of class `matrix` (inherits from `array`) with 33 rows and 33 columns.

Details

A matrix with 33x33 rows.
The dataset was firstly used in Wijayanto (2015), Wijayanto and Purwarianti (2014), Wijayanto and Purwarianti (2014), and Wijayanto et al. (2016).

References

Wijayanto AW (2015). *Improvement of fuzzy geo-demographic clustering using metaheuristic optimization on Indonesia population census*. Master's thesis, Insitut Teknologi Bandung.

Wijayanto AW, Purwarianti A (2014). "Improvement of fuzzy geographically weighted clustering using particle swarm optimization." In *2014 International Conference on Information Technology Systems and Innovation (ICITSI)*, 7–12. ISBN 978-1-4799-6527-4.

Wijayanto AW, Purwarianti A (2014). "Improvement design of fuzzy geo-demographic clustering using Artificial Bee Colony optimization." In *2014 International Conference on Cyber and IT Service Management (CITSM)*, 69–74. ISBN 978-1-4799-7975-2.

Wijayanto AW, Purwarianti A, Son LH (2016). "Fuzzy geographically weighted clustering using artificial bee colony: An efficient geo-demographic analysis algorithm and applications to the analysis of crime behavior in population." *Applied Intelligence*, **44**(2), 377–398. ISSN 0924-669X.

`census2010pop` *Indonesia 2010 Population*

Description

A dataset containing the 2010 population in Indonesia based on census result.

Usage

```
census2010pop
```

Format

An object of class `integer` of length 33.

Details

A vector with 33 elements.
The dataset was firstly used in Wijayanto (2015), Wijayanto and Purwarianti (2014), Wijayanto and Purwarianti (2014), and Wijayanto et al. (2016).

References

Wijayanto AW (2015). *Improvement of fuzzy geo-demographic clustering using metaheuristic optimization on Indonesia population census*. Master's thesis, Insitut Teknologi Bandung.

Wijayanto AW, Purwarianti A (2014). "Improvement of fuzzy geographically weighted clustering using particle swarm optimization." In *2014 International Conference on Information Technology Systems and Innovation (ICITSI)*, 7–12. ISBN 978-1-4799-6527-4.

Wijayanto AW, Purwarianti A (2014). "Improvement design of fuzzy geo-demographic clustering using Artificial Bee Colony optimization." In *2014 International Conference on Cyber and IT Service Management (CITSM)*, 69–74. ISBN 978-1-4799-7975-2.

Wijayanto AW, Purwarianti A, Son LH (2016). "Fuzzy geographically weighted clustering using artificial bee colony: An efficient geo-demographic analysis algorithm and applications to the analysis of crime behavior in population." *Applied Intelligence*, **44**(2), 377–398. ISSN 0924-669X.

`fgwc` *Fuzzy Geographicaly Weighted Clustering*

Description

Fuzzy clustering with addition of spatial configuration of membership matrix

Usage

```
fgwc(data, pop, distmat, algorithm = "classic", fgwc_param, opt_param)
```

Arguments

`data` an object of data with d>1. Can be `matrix` or `data.frame`. If your data is univariate, bind it with 1 to get a 2 columns.

`pop` an n*1 vector contains population.

`distmat` an n*n distance matrix between regions.

`algorithm`	algorithm used for FGWC
`fgwc_param`	a vector that consists of FGWC parameter (see `fgwcuv` for parameter details)
`opt_param`	a vector that consists of optimization algorithm parameter (see `fgwcuv` for parameter details)

Details

Fuzzy Geographically Weighted Clustering (FGWC) was developed by Mason and Jacobson (2007) by adding neighborhood effects and population to configure the membership matrix in Fuzzy C-Means. There are seven optimisation algorithms that currently provided in this package, mainly from the Yang (2014). The optimization algorithm uses the centroid as the parameter to be optimized. Here are the algorithm that can be used:

- `"classic"` - The classical algorithm of FGWC based on Mason and Jacobson (2007) for centroid optimisation and Runkler and Katz (2006) for membership optimization.
- `"abc"` - Optimization using artificial bee colony algorithm based on Karaboga and Basturk (2007) (see also Wijayanto and Purwarianti 2014 and Wijayanto et al. 2016 for FGWC implementation).
- `"fpa"` - Optimization using flower pollination algorithm based on (Yang 2012).
- `"gsa"` - Optimization using gravitational search algorithm based on Rashedi et al. (2009) and Li and Dong (2017) (see also Pamungkas and Pramana 2019 for FGWC implementation).
- `"hho"` - Optimization using harris-hawk optimization with `"heidari"` (Heidari et al. 2019) (default). and `"bairathi"` (Bairathi and Gopalani 2018).
- `"ifa"` - Optimization using intelligent firefly algorithm based on Yang (2009), as well as the intelligent improvement by Fateen and Bonilla-Petriciolet (2013) (see also Nasution et al. 2020 for FGWC implementation).
- `"pso"` - Optimization using particle swarm optimization based on Runkler and Katz (2006) and Bansal et al. (2011) for inertia option (see also Wijayanto and Purwarianti 2014; Putra and Kurniawan 2017; Abdussamad 2020 for FGWC implementation).
- `"tlbo"` - Optimization using teaching - learning based optimization based on Rao et al. (2012) and elitism improvement by Rao and Patel (2012).

Furthermore, there are 10 distance that can be used to calculate the membership (see `cdist` for details). the default parameter of FGWC (in case you do not want to tune anything) is
`c(kind='u',ncluster=2,m=2,distance='euclidean',order=2,alpha=0.7,a=1,b=1,`
`max.iter=500,error=1e-5,randomN=1)`.
There is also a universal parameter to the optimization algorithm as well as the details. The default parameter for the optimization algorithm is
`c(vi.dist='uniform',npar=10,par.no=2,par.dist='euclidean',par.order=2,pso=TRUE,`
`same=10,type='sim.annealing',ei.distr='normal',vmax=0.7,wmax=0.9,wmin=0.4,`
`chaos=4,x0='F',map=0.7,ind=1,skew=0,sca=1)`
If you do not define a certain parameter, the parameter will be set to its default value

Value

an object of class `"fgwc"`.
An `"fgwc"` object contains as follows:

- `converg` - the process convergence of objective function
- `f_obj` - objective function value
- `membership` - membership matrix
- `centroid` - centroid matrix
- `validation` - validation indices (there are partition coefficient (`PC`), classification entropy (`CE`), SC index (`SC`), separation index (`SI`), Xie and Beni's index (`XB`), IFV index (`IFV`), and Kwon index (`Kwon`))
- `max.iter` - Maximum iteration
- `cluster` - the cluster of the data
- `finaldata` - The final data (with the cluster)
- `call` - the syntax called previously
- `time` - computational time.

References

Abdussamad S (2020). "Evaluation of Implementation Context Based Clustering In Fuzzy Geographically Weighted Clustering-Particle Swarm Optimization Algorithm." *Jurnal EECCIS*, **14**(1), 10–15. ISSN 2460-8122, `https://jurnaleeccis.ub.ac.id/index.php/eeccis/article/view/609`.

Bairathi D, Gopalani D (2018). "A Novel Swarm Intelligence Based Optimization Method: Harris' Hawk Optimization." In *Advances in Intelligent Systems and Computing*, 832–842. Springer International Publishing. doi: 10.1007/9783030166601_81, `https://doi.org/10.1007/978-3-030-16660-1_81`.

Bansal JC, Singh PK, Saraswat M, Verma A, Jadon SS, Abraham A (2011). "Inertia Weight strategies in Particle Swarm Optimization." In *2011 Third World Congress on Nature and Biologically Inspired Computing*. doi: 10.1109/nabic.2011.6089659, `https://doi.org/10.1109/nabic.2011.6089659`.

Fateen SK, Bonilla-Petriciolet A (2013). "Intelligent Firefly Algorithm for Global Optimization." *Cuckoo Search and Firefly Algorithm: Theory and Applications*, **516**, 315–330.

Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019). "Harris hawks optimization: Algorithm and applications." *Future Generation Computer Systems*, **97**, 849–872. doi: 10.1016/j.future.2019.02.028, `https://doi.org/10.1016/j.future.2019.02.028`.

Karaboga D, Basturk B (2007). "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm." *Journal of Global Optimization*, **39**(3), 459–471. doi: 10.1007/s108980079149x, `https://doi.org/10.1007/s10898-007-9149-x`.

Li J, Dong N (2017). "Gravitational Search Algorithm with a New Technique." In *2017 13th International Conference on Computational Intelligence and Security (CIS)*, 516–519. doi: 10.1109/CIS.2017.00120, `https://doi.org/10.1109/CIS.2017.00120`.

Mason GA, Jacobson RD (2007). "Fuzzy Geographically Weighted Clustering." In *Proceedings* *of the 9th International Conference on Geocomputation*, 1–7.

Nasution BI, Kurniawan R, Siagian TH, Fudholi A (2020). “Revisiting social vulnerability analysis in Indonesia: An optimized spatial fuzzy clustering approach.” *International Journal of Disaster Risk Reduction*, **51**, 101801. doi: 10.1016/j.ijdrr.2020.101801, https://doi.org/10.1016/j.ijdrr.2020.101801.

Pamungkas IH, Pramana S (2019). “Improvement Method of Fuzzy Geographically Weighted Clustering using Gravitational Search Algorithm.” *Journal of Computer Science and Information*, **11**(1).

Putra FH, Kurniawan R (2017). “Clustering for Disaster Areas Endemic Dengue Hemorrhagic Fever Based on Factors had Caused in East Java Using Fuzzy Geographically Weighted Clustering - Particle Swarm Optimization.” *Jurnal Aplikasi Statistika \& Komputasi Statistik*, **8**(01), 27. ISSN 2615-1367.

Rao RV, Patel V (2012). “An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems.” *International Journal of Industrial Engineering Computations*, **3**(4), 535–560. ISSN 19232926, doi: 10.5267/j.ijiec.2012.03.007, https://doi.org/10.5267/j.ijiec.2012.03.007.

Rao RV, Savsani VJ, Balic J (2012). “Teaching\- learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems.” *Engineering Optimization*, **44**(12), 1447–1462. doi: 10.1080/0305215x.2011.652103, https://doi.org/10.1080/0305215x.2011.652103.

Rashedi E, Nezamabadi-pour H, Saryazdi S (2009). “GSA: A Gravitational Search Algorithm.” *Information Sciences*, **179**(13).

Runkler TA, Katz C (2006). “Fuzzy Clustering by Particle Swarm Optimization.” In *2006 IEEE International Conference on Fuzzy Systems*. doi: 10.1109/fuzzy.2006.1681773, https://doi.org/10.1109/fuzzy.2006.1681773.

Wijayanto AW, Purwarianti A (2014). “Improvement design of fuzzy geo-demographic clustering using Artificial Bee Colony optimization.” In *2014 International Conference on Cyber and IT Service Management (CITSM)*, 69–74. ISBN 978-1-4799-7975-2.

Wijayanto AW, Purwarianti A (2014). “Improvement of fuzzy geographically weighted clustering using particle swarm optimization.” In *2014 International Conference on Information Technology Systems and Innovation (ICITSI)*, 7–12. ISBN 978-1-4799-6527-4.

Wijayanto AW, Purwarianti A, Son LH (2016). “Fuzzy geographically weighted clustering using artificial bee colony: An efficient geo-demographic analysis algorithm and applications to the analysis of crime behavior in population.” *Applied Intelligence*, **44**(2), 377–398. ISSN 0924-669X.

Yang X (2014). *Nature-Inspired Optimization Algorithms*, Elsevier insights. Elsevier Science. ISBN 9780124167452.

Yang X (2012). “Flower Pollination Algorithm for Global Optimization.” In *Unconventional Com-*

putation and Natural Computation, 240–249. Springer Berlin Heidelberg. doi: 10.1007/9783642-328947_27, `https://doi.org/10.1007/978-3-642-32894-7_27`.

Yang X (2009). “Firefly Algorithms for Multimodal Optimization.” In *Stochastic Algorithms: Foundations and Applications*, 169–178. Springer Berlin Heidelberg. doi: 10.1007/978364204944-6_14, `https://doi.org/10.1007/978-3-642-04944-6_14`.

See Also

`fgwcuv`, `abcfgwc`, `fpafgwc`, `gsafgwc`, `hhofgwc`, `ifafgwc`, `psofgwc`, `tlbofgwc`

Examples

```
data('census2010')
data('census2010dist')
data('census2010pop')
# initiate parameter
param_fgwc <- c(kind='v',ncluster=3,m=2,distance='minkowski',order=3,
               alpha=0.5,a=1.2,b=1.2,max.iter=1000,error=1e-6,randomN=10)
## FGWC with classical algorithm
res1 <- fgwc(census2010,census2010pop,census2010dist,'classic',param_fgwc,1)
## tune the ABC parameter
abc_param <- c(vi.dist='normal',npar=5,pso=FALSE,same=15,n.onlooker=5,limit=5)
## FGWC with ABC optimization algorithm
res2 <- fgwc(census2010,census2010pop,census2010dist,'abc',param_fgwc,abc_param)
```

`fgwcuv` *Classical Fuzzy Geographicaly Weighted Clustering*

Description

Fuzzy clustering with addition of spatial configuration of membership matrix

Usage

```
fgwcuv(
  data,
  pop,
  distmat,
  kind = NA,
  ncluster = 2,
  m = 2,
  distance = "euclidean",
  order = 2,
  alpha = 0.7,
  a = 1,
  b = 1,
  max.iter = 500,
  error = 1e-05,
  randomN = 0,
  uij = NA,
  vi = NA
)
```

Arguments

`data` an object of data with d>1. Can be `matrix` or `data.frame`. If your data is univariate, bind it with 1 to get a 2 columns.

`pop` an n*1 vector contains population.

`distmat` an n*n distance matrix between regions.

`kind` use `'u'` if you want to use membership approach and `'v'` for centroid approach.

`ncluster` an integer. The number of clusters.

`m` degree of fuzziness or fuzzifier. Default is 2.

`distance` the distance metric between data and centroid, the default is euclidean, see `cdist` for details.

`order,` minkowski order. default is 2.

`alpha` the old membership effect with [0,1], if `alpha` equals 1, it will be same as fuzzy C-Means, if 0, it equals to neighborhood effect.

`a` spatial magnitude of distance. Default is 1.

`b` spatial magnitude of population. Default is 1.

`max.iter` maximum iteration. Default is 500.

`error` error tolerance. Default is 1e-5.

`randomN` random seed for initialisation (if uij or vi is NA). Default is 0.

`uij` membership matrix initialisation.

`vi` centroid matrix initialisation.

Details

Fuzzy Geographically Weighted Clustering (FGWC) was developed by Mason and Jacobson (2007) by adding neighborhood effects and population to configure the membership matrix in Fuzzy C-Means. There are two kinds of options in doing classical FGWC. The first is using `"u"` (Runkler and Katz 2006) (default) for membership optimization and `"v"` (Mason and Jacobson 2007) for centroid optimisation.

Value

an object of class `"fgwc"`.
An `"fgwc"` object contains as follows:

- `converg` - the process convergence of objective function
- `f_obj` - objective function value
- `membership` - membership matrix

- `centroid` - centroid matrix
- `validation` - validation indices (there are partition coefficient (`PC`), classification entropy (`CE`), SC index (`SC`), separation index (`SI`), Xie and Beni's index (`XB`), IFV index (`IFV`), and Kwon index (Kwon))
- `max.iter` - Maximum iteration
- `cluster` - the cluster of the data
- `finaldata` - The final data (with the cluster)
- `call` - the syntax called previously
- `time` - computational time.

References

Mason GA, Jacobson RD (2007). "Fuzzy Geographically Weighted Clustering." In *Proceedings of the 9th International Conference on Geocomputation*, 1–7.

Runkler TA, Katz C (2006). "Fuzzy Clustering by Particle Swarm Optimization." In *2006 IEEE International Conference on Fuzzy Systems*. doi: 10.1109/fuzzy.2006.1681773, `https://doi.org/10.1109/fuzzy.2006.1681773`.

See Also

`abcfgwc` `fpafgwc` `gsafgwc` `hhofgwc` `ifafgwc` `psofgwc` `tlbofgwc`

Examples

```
data('census2010')
data('census2010dist')
data('census2010pop')
res1 <- fgwcuv(census2010,census2010pop,census2010dist,'u',3,2,'euclidean',4)
```

`fpafgwc` *Fuzzy Geographicaly Weighted Clustering with Flower Pollination Algorithm*

Description

Fuzzy clustering with addition of spatial configuration of membership matrix with centroid optimization using Flower Pollination Algorithm

Usage

```
fpafgwc(
  data,
  pop = NA,
  distmat = NA,
  ncluster = 2,
  m = 2,
  distance = "euclidean",
  order = 2,
  alpha = 0.7,
  a = 1,
  b = 1,
  error = 1e-05,
  max.iter = 100,
  randomN = 0,
  vi.dist = "uniform",
  nflow = 10,
  p = 0.8,
  gamma = 1,
  lambda = 1.5,
  delta = 0,
  ei.distr = "normal",
  flow.same = 10,
  r = 4,
  m.chaotic = 0.7,
  skew = 0,
  sca = 1
)
```

Arguments

data	an object of data with d>1. Can be matrix or data.frame. If your data is univariate, bind it with 1 to get a 2 columns.
pop	an n*1 vector contains population.
distmat	an n*n distance matrix between regions.
ncluster	an integer. The number of clusters.
m	degree of fuzziness or fuzzifier. Default is 2.
distance	the distance metric between data and centroid, the default is euclidean, see cdist for details.
order,	minkowski order. default is 2.
alpha	the old membership effect with [0,1], if alpha equals 1, it will be same as fuzzy C-Means, if 0, it equals to neighborhood effect.
a	spatial magnitude of distance. Default is 1.
b	spatial magnitude of population. Default is 1.
error	error tolerance. Default is 1e-5.

`max.iter` maximum iteration. Default is 500.

`randomN` random seed for initialisation (if uij or vi is NA). Default is 0.

`vi.dist` a string of centroid population distribution between `"uniform"` (default) and `"normal"`. Can be defined as `vi.dist=` in `opt_param`.

`nflow` number of flowers population. Can be defined as `npar=` in `opt_param`. Default is 10.

`p` switch probability between global and local pollination, Can be defined as `p` in `opt_param`. default is 0.8.

`gamma` Step size scaling factor. Can be defined as `gamma` in `opt_param`. Default is 1.

`lambda` Levy flights index parameter between [0,2]. Can be defined as `lambda` in `opt_param`. Default is 1.5.

`delta` Levi flights shift. Can be defined as `delta` in `opt_param`. Default is 0.

`ei.distr` distribution of random walk parameter. Can be defined as `ei.distr` in `opt_param`.

`flow.same` number of consecutive unchange to stop the iteration. Can be defined as `same=` in `opt_param`.

`r` weight in logistic chaotic between [0,4]. Can be used when `ei.distr='logchaotic'`. Can be defined as `chaos` in `opt_param`.

`m.chaotic` mapping parameter in kent chaotic between [0,1]. Can be used when `ei.distr='kentchaotic'`. Can be defined as `map` in `opt_param`.

`skew` Levy distribution skewness for random walk. Can be used when `ei.distr='levy'`. Can be defined as `skew` in `opt_param`.

`sca` Levy distribution scale for random walk. Can be used when `ei.distr='levy'`. Can be defined as `sca` in `opt_param`.

Details

Fuzzy Geographically Weighted Clustering (FGWC) was developed by Mason and Jacobson (2007) by adding neighborhood effects and population to configure the membership matrix in Fuzzy C-Means. Furthermore, the Flower Pollination Algorithm was developed by Yang (2012) in order to get a more optimal solution of a certain complex function.

Value

an object of class `"fgwc"`.
An `"fgwc"` object contains as follows:

- `converg` - the process convergence of objective function
- `f_obj` - objective function value
- `membership` - membership matrix
- `centroid` - centroid matrix
- `validation` - validation indices (there are partition coefficient (`PC`), classification entropy (`CE`), SC index (`SC`), separation index (`SI`), Xie and Beni's index (`XB`), IFV index (`IFV`), and Kwon index (Kwon))
- `max.iter` - Maximum iteration

- cluster - the cluster of the data
- finaldata - The final data (with the cluster)
- call - the syntax called previously
- time - computational time.

See Also

fpafgwc gsafgwc

Examples

```
data('census2010')
data('census2010dist')
data('census2010pop')
# First way
res1 <- fpafgwc(census2010,census2010pop,census2010dist,3,2,'euclidean',4,nflow=10)
# Second way
# initiate parameter
param_fgwc <- c(kind='v',ncluster=3,m=2,distance='minkowski',order=3,
               alpha=0.5,a=1.2,b=1.2,max.iter=1000,error=1e-6,randomN=10)
## tune the FPA parameter
fpa_param <- c(vi.dist='normal',npar=5,same=15,p=0.7,
               gamma=1.2,lambda=1.5,ei.distr='logchaotic',chaos=3)
##FGWC with FPA
res2 <- fgwc(census2010,census2010pop,census2010dist,'fpa',param_fgwc,fpa_param)
```

gsafgwc *Fuzzy Geographicaly Weighted Clustering with Gravitational Search Algorithm*

Description

Fuzzy clustering with addition of spatial configuration of membership matrix with centroid optimization using Gravitational Search Algorithm

Usage

```
gsafgwc(
  data,
  pop = NA,
  distmat = NA,
  ncluster = 2,
  m = 2,
  distance = "euclidean",
  order = 2,
  alpha = 0.7,
  a = 1,
  b = 1,
  error = 1e-05,
  max.iter = 100,
  randomN = 0,
  vi.dist = "uniform",
  npar = 10,
  par.no = 2,
  par.dist = "euclidean",
  par.order = 2,
  gsa.same = 10,
  G = 1,
  vmax = 0.7,
  new = F
)
```

Arguments

`data` an object of data with d>1. Can be `matrix` or `data.frame`. If your data is univariate, bind it with `1` to get a 2 columns.

`pop` an n*1 vector contains population.

`distmat` an n*n distance matrix between regions.

`ncluster` an integer. The number of clusters.

`m` degree of fuzziness or fuzzifier. Default is 2.

`distance` the distance metric between data and centroid, the default is euclidean, see `cdist` for details.

`order,` minkowski order. default is 2.

`alpha` the old membership effect with [0,1], if `alpha` equals 1, it will be same as fuzzy C-Means, if 0, it equals to neighborhood effect.

`a` spatial magnitude of distance. Default is 1.

`b` spatial magnitude of population. Default is 1.

`error` error tolerance. Default is 1e-5.

`max.iter` maximum iteration. Default is 500.

`randomN` random seed for initialisation (if uij or vi is NA). Default is 0.

`vi.dist` a string of centroid population distribution between `'uniform'` (default) and `'normal'`. Can be defined as `vi.dist=` in `opt_param`.

`npar` number of particle. Can be defined as `npar=` in `opt_param`. Default is 10.

`par.no` The number of selected best particle. Can be defined as `par.no=` in `opt_param`. Default is 2

`par.dist` The distance between particles. Can be defined as `par.dist=` in `opt_param`. Default is `'euclidean'`,

`par.order` The minkowski order of the `par.dist` if `par.dist='minkowski'`. Can be defined as `par.order=` in `opt_param`. Default is 2

`gsa.same` number of consecutive unchange to stop the iteration. Can be defined as `same=` in `opt_param`.

`G` initial gravitatioal constant, Can be defined as `G` in `opt_param`. default is 1.

`vmax` maximum velocity to be tolerated. Can be defined as `vmax` in `opt_param`. Default is 0.7

`new` Boolean that represents whether to use the new algorithm by Li and Dong (2017). Can be defined as `new` in `opt_param`. Default is `FALSE`

Details

Fuzzy Geographically Weighted Clustering (FGWC) was developed by Mason and Jacobson (2007) by adding neighborhood effects and population to configure the membership matrix in Fuzzy C-Means. Furthermore, the Gravitational Search Algorithm was developed by Rashedi et al. (2009) and and the technique is also upgraded by Li and Dong (2017) in order to get a more optimal solution of a certain complex function. FGWC using GSA has been implemented previously by Pamungkas and Pramana (2019).

Value

an object of class `'fgwc'`.
An `'fgwc'` object contains as follows:

- `converg` - the process convergence of objective function
- `f_obj` - objective function value
- `membership` - membership matrix
- `centroid` - centroid matrix
- `validation` - validation indices (there are partition coefficient (`PC`), classification entropy (`CE`), SC index (`SC`), separation index (`SI`), Xie and Beni's index (`XB`), IFV index (`IFV`), and Kwon index (Kwon))
- `max.iter` - Maximum iteration
- `cluster` - the cluster of the data
- `finaldata` - The final data (with the cluster)
- `call` - the syntax called previously
- `time` - computational time.

References

Li J, Dong N (2017). "Gravitational Search Algorithm with a New Technique." In *2017 13th International Conference on Computational Intelligence and Security (CIS)*, 516–519. doi: 10.1109/CIS.2017.00120, `https://doi.org/10.1109/CIS.2017.00120`.

Pamungkas IH, Pramana S (2019). "Improvement Method of Fuzzy Geographically Weighted Clustering using Gravitational Search Algorithm." *Journal of Computer Science and Information*, **11**(1).

Rashedi E, Nezamabadi-pour H, Saryazdi S (2009). "GSA: A Gravitational Search Algorithm." *Information Sciences*, **179**(13).

See Also

fpafgwc gsafgwc

Examples

```
data('census2010')
data('census2010dist')
data('census2010pop')
# First way
res1 <- gsafgwc(census2010,census2010pop,census2010dist,3,2,'euclidean',4,npar=10)
# Second way
# initiate parameter
param_fgwc <- c(kind='v',ncluster=3,m=2,distance='minkowski',order=3,
               alpha=0.5,a=1.2,b=1.2,max.iter=1000,error=1e-6,randomN=10)
## tune the GSA parameter
gsa_param <- c(vi.dist='normal',npar=5,same=15,G=1,vmax=0.7,new=FALSE)
##FGWC with GSA
res2 <- fgwc(census2010,census2010pop,census2010dist,'gsa',param_fgwc,gsa_param)
```

hhofgwc *Fuzzy Geographicaly Weighted Clustering with Harris-Hawk Optimization*

Description

Fuzzy clustering with addition of spatial configuration of membership matrix with centroid optimization using Harris-Hawk Algorithm.

Usage

```
hhofgwc(
  data,
  pop = NA,
  distmat = NA,
  ncluster = 2,
  m = 2,
  distance = "euclidean",
  order = 2,
  alpha = 0.7,
  a = 1,
  b = 1,
  error = 1e-05,
  max.iter = 100,
  randomN = 0,
  vi.dist = "uniform",
  nhh = 10,
  hh.alg = "heidari",
  A = c(2, 1, 0.5),
  p = 0.5,
  hh.same = 10,
  levy.beta = 1.5,
  update.type = 5
)
```

Arguments

`data` an object of data with d>1. Can be `matrix` or `data.frame`. If your data is univariate, bind it with 1 to get a 2 columns.

`pop` an n*1 vector contains population.

`distmat` an n*n distance matrix between regions.

`ncluster` an integer. The number of clusters.

`m` degree of fuzziness or fuzzifier. Default is 2.

`distance` the distance metric between data and centroid, the default is euclidean, see `cdist` for details.

`order,` minkowski order. default is 2.

`alpha` the old membership effect with [0,1], if `alpha` equals 1, it will be same as fuzzy C-Means, if 0, it equals to neighborhood effect.

`a` spatial magnitude of distance. Default is 1.

`b` spatial magnitude of population. Default is 1.

`error` error tolerance. Default is 1e-5.

`max.iter` maximum iteration. Default is 500.

`randomN` random seed for initialisation (if uij or vi is NA). Default is 0.

`vi.dist` a string of centroid population distribution between `'uniform'` (default) and `'normal'`. Can be defined as `vi.dist=` in `opt_param`.

`nhh` number of harris-hawk eagles. Can be defined as `npar=` in `opt_param`. Default is 10.

`hh.alg` String between default is `'heidari'` and default is `'bairathi'`. The algorithm for HHO, Can be defined as `algo` in `opt_param`. default is `'heidari'`.

`A` a 3 vectors which represents initial energy and cut-off for exploitation and exploration. In `opt_param`, they can be defined as `'a1'` for initial energy, `'a2'` for exploitation cut-off and `'a3'` for exploration cut-off respectively. default is `c(”a1”=2,”a2”=1,”a3”=0.5)`.

`p` a real number between 0 and 1. The eagle’s movement probability

`hh.same` number of consecutive unchange to stop the iteration. Can be defined as `same=` in `opt_param`.

`levy.beta` The skewness of levy flight. Can be defined as `beta` in `opt_param`. Default is 1.5

`update.type` An integer. The type of energy `A[1]` update. Can be selected from 1 to 5. Can be defined as `update.type` in `opt_param`. Default is 5.

Details

Fuzzy Geographically Weighted Clustering (FGWC) was developed by Mason and Jacobson (2007) by adding neighborhood effects and population to configure the membership matrix in Fuzzy C-Means. Furthermore, the Harris-Hawk Optimization was developed by Bairathi and Gopalani (2018) and the technique is also upgraded by Heidari et al. (2019) by adding progressive rapid dives in order to get a more optimal solution of a certain complex function.

Value

an object of class `'fgwc'`.
An `'fgwc'` object contains as follows:

- `converg` - the process convergence of objective function
- `f_obj` - objective function value
- `membership` - membership matrix
- `centroid` - centroid matrix
- `validation` - validation indices (there are partition coefficient (`PC`), classification entropy (`CE`), SC index (`SC`), separation index (`SI`), Xie and Beni's index (`XB`), IFV index (`IFV`), and Kwon index (Kwon))
- `max.iter` - Maximum iteration
- `cluster` - the cluster of the data
- `finaldata` - The final data (with the cluster)
- `call` - the syntax called previously
- `time` - computational time.

References

Bairathi D, Gopalani D (2018). "A Novel Swarm Intelligence Based Optimization Method: Harris' Hawk Optimization." In *Advances in Intelligent Systems and Computing*, 832–842. Springer International Publishing. doi: 10.1007/9783030166601_81, `https://doi.org/10.1007/978-3-030-16660-1_81`.

Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019). "Harris hawks optimization: Algorithm and applications." *Future Generation Computer Systems*, **97**, 849–872. doi: 10.1016/j.future.2019.02.028, `https://doi.org/10.1016/j.future.2019.02.028`.

Mason GA, Jacobson RD (2007). "Fuzzy Geographically Weighted Clustering." In *Proceedings of the 9th International Conference on Geocomputation*, 1–7.

See Also

`fpafgwc` `gsafgwc`

Examples

```
data('census2010')
data('census2010dist')
data('census2010pop')
# First way
res1 <- hhofgwc(census2010,census2010pop,census2010dist,3,2,'euclidean',4,nhh=10)
# Second way
# initiate parameter
param_fgwc <- c(kind='v',ncluster=3,m=2,distance='minkowski',order=3,
               alpha=0.5,a=1.2,b=1.2,max.iter=1000,error=1e-6,randomN=10)
## tune the HHO parameter
hho_param <- c(vi.dist='normal',npar=5,same=15,algo='bairathi',a1=3,a2=1,a3=0.4)
##FGWC with HHO
res2 <- fgwc(census2010,census2010pop,census2010dist,'hho',param_fgwc,hho_param)
```

`ifafgwc` *Fuzzy Geographicaly Weighted Clustering with (Intelligent) Firefly Algorithm*

Description

Fuzzy clustering with addition of spatial configuration of membership matrix with centroid optimization using (Intelligent) Firefly Algorithm.

Usage

```
ifafgwc(
  data,
  pop = NA,
  distmat = NA,
  ncluster = 2,
  m = 2,
  distance = "euclidean",
  order = 2,
  alpha = 0.7,
  a = 1,
  b = 1,
  error = 1e-05,
  max.iter = 100,
  randomN = 0,
  vi.dist = "uniform",
  ei.distr = "normal",
  fa.same = 10,
  nfly = 10,
  ffly.no = 2,
  ffly.dist = "euclidean",
  ffly.order = 2,
  gamma = 1,
  ffly.beta = 1,
  ffly.alpha = 1,
  r.chaotic = 4,
  m.chaotic = 0.7,
  ind.levy = 1,
  skew.levy = 0,
  scale.levy = 1,
  ffly.alpha.type = 4
)
```

Arguments

data — an object of data with d>1. Can be matrix or data.frame. If your data is univariate, bind it with 1 to get a 2 columns.

pop — an n*1 vector contains population.

distmat — an n*n distance matrix between regions.

ncluster — an integer. The number of clusters.

m — degree of fuzziness or fuzzifier. Default is 2.

distance — the distance metric between data and centroid, the default is euclidean, see cdist for details.

order, — minkowski order. default is 2.

alpha — the old membership effect with [0,1], if alpha equals 1, it will be same as fuzzy C-Means, if 0, it equals to neighborhood effect.

a — spatial magnitude of distance. Default is 1.

b — spatial magnitude of population. Default is 1.

error — error tolerance. Default is 1e-5.

max.iter — maximum iteration. Default is 500.

randomN — random seed for initialisation (if uij or vi is NA). Default is 0.

vi.dist — a string of centroid population distribution between 'uniform' (default) and 'normal'. Can be defined as vi.dist= in opt_param.

ei.distr — distribution of random walk parameter. Can be defined as ei.distr in opt_param.

fa.same — number of consecutive unchange to stop the iteration. Can be defined as same= in opt_param.

nfly — number of fireflies. Can be defined as npar= in opt_param. Default is 10.

ffly.no — The number of selected best fireflies for intelligent firefly algorithm. Can be defined as par.no= in opt_param. Default is 2

ffly.dist — The distance between fireflies. Can be defined as par.dist= in opt_param. Default is 'euclidean',

ffly.order — The minkowski order of the par.dist if par.dist='minkowski'. Can be defined as par.order= in opt_param. Default is 2

gamma — distance scaling factor. Can be defined as gamma in opt_param. Default is 1.

`ffly.beta` Attractiveness constant. Can be defined as `beta` in `opt_param`. Default is 1.

`ffly.alpha` Randomisation constant. Can be defined as `alpha=` in `opt_param`.

`r.chaotic` weight in logistic chaotic between [0,4]. Can be used when `ei.distr='logchaotic'`. Can be defined as `chaos` in `opt_param`. Default is 4.

`m.chaotic` mapping parameter in kent chaotic between [0,1]. Can be used when `ei.distr='kentchaotic'`. Can be defined as `map` in `opt_param`. Default is 0.7.

`ind.levy` Levy distribution index for random walk. Can be used when `ei.distr='levy'`. Can be defined as `ind` in `opt_param`. Default is 1.

`skew.levy` Levy distribution skewness for random walk. Can be used when `ei.distr='levy'`. Can be defined as `skew` in `opt_param`. Default is 0.

`scale.levy` Levy distribution scale for random walk. Can be used when `ei.distr='levy'`. Can be defined as `sca` in `opt_param`. Default is 1.

`ffly.alpha.type`
An integer. The type of `ffly.alpha` update. Can be selected from 1 to 5. Can be defined as `update_type` in `opt_param`. Default is 4.

Details

Fuzzy Geographically Weighted Clustering (FGWC) was developed by Mason and Jacobson (2007) by adding neighborhood effects and population to configure the membership matrix in Fuzzy C-Means. Furthermore, the Firefly Algorithm was developed by Yang (2009) and the technique is also upgraded by Fateen and Bonilla-Petriciolet (2013) by adding the intelligent phase (choosing the best firefly based on the intensity) in order to get a more optimal solution of a certain complex function. FGWC using IFA has been implemented previously by Nasution et al. (2020).

Value

an object of class `'fgwc'`.
An `'fgwc'` object contains as follows:

- `converg` - the process convergence of objective function
- `f_obj` - objective function value
- `membership` - membership matrix
- `centroid` - centroid matrix
- `validation` - validation indices (there are partition coefficient (`PC`), classification entropy (`CE`), SC index (`SC`), separation index (`SI`), Xie and Beni's index (`XB`), IFV index (`IFV`), and Kwon index (Kwon))
- `max.iter` - Maximum iteration
- `cluster` - the cluster of the data
- `finaldata` - The final data (with the cluster)
- `call` - the syntax called previously
- `time` - computational time.

References

Fateen SK, Bonilla-Petriciolet A (2013). “Intelligent Firefly Algorithm for Global Optimization.” *Cuckoo Search and Firefly Algorithm: Theory and Applications*, **516**, 315–330.

Mason GA, Jacobson RD (2007). “Fuzzy Geographically Weighted Clustering.” In *Proceedings of the 9th International Conference on Geocomputation*, 1–7.

Nasution BI, Kurniawan R, Siagian TH, Fudholi A (2020). “Revisiting social vulnerability analysis in Indonesia: An optimized spatial fuzzy clustering approach.” *International Journal of Disaster Risk Reduction*, **51**, 101801. doi: 10.1016/j.ijdrr.2020.101801, `https://doi.org/10.1016/j.ijdrr.2020.101801`.

Yang X (2009). “Firefly Algorithms for Multimodal Optimization.” In *Stochastic Algorithms: Foundations and Applications*, 169–178. Springer Berlin Heidelberg. doi: 10.1007/978364204944-6_14, `https://doi.org/10.1007/978-3-642-04944-6_14`.

See Also

`fpafgwc` `gsafgwc`

Examples

```
data('census2010')
data('census2010dist')
data('census2010pop')
# First way
res1 <- ifafgwc(census2010,census2010pop,census2010dist,3,2,'minkowski',4,nfly=10)
# Second way
# initiate parameter
param_fgwc <- c(kind='v',ncluster=3,m=2,distance='minkowski',order=3,
               alpha=0.5,a=1.2,b=1.2,max.iter=1000,error=1e-6,randomN=10)
## tune the IFA parameter
ifa_param <- c(vi.dist='uniform', ei.distr='logchaotic',
fa.same=10, npar=15, par.no=3, par.dist='minkowski',
           par.order=4, gamma=1, beta=1.5,
           alpha=1, chaos=4,update_type=4)
##FGWC with IFA
res2 <- fgwc(census2010,census2010pop,census2010dist,'ifa',param_fgwc,ifa_param)
```

---

psofgwc *Fuzzy Geographicaly Weighted Clustering with Particle Swarm Optimization*

Description

Fuzzy clustering with addition of spatial configuration of membership matrix with centroid optimization using Particle Swarm Algorithm.

Usage

```
psofgwc(
  data,
  pop = NA,
  distmat = NA,
  ncluster = 2,
  m = 2,
  distance = "euclidean",
  order = 2,
  alpha = 0.7,
  a = 1,
  b = 1,
  error = 1e-05,
  max.iter = 100,
  randomN = 0,
  vi.dist = "uniform",
  npar = 10,
  vmax = 0.7,
  pso.same = 10,
  c1 = 0.49,
  c2 = 0.49,
  w.inert = "sim.annealing",
  wmax = 0.9,
  wmin = 0.4,
  map = 0.4
)
```

Arguments

`data` an object of data with d>1. Can be `matrix` or `data.frame`. If your data is univariate, bind it with 1 to get a 2 columns.

`pop` an n*1 vector contains population.

`distmat` an n*n distance matrix between regions.

`ncluster` an integer. The number of clusters.

`m` degree of fuzziness or fuzzifier. Default is 2.

`distance` the distance metric between data and centroid, the default is euclidean, see `cdist` for details.

`order,` minkowski order. default is 2.

`alpha` the old membership effect with [0,1], if `alpha` equals 1, it will be same as fuzzy C-Means, if 0, it equals to neighborhood effect.

`a` spatial magnitude of distance. Default is 1.

`b` spatial magnitude of population. Default is 1.

`error` error tolerance. Default is 1e-5.

`max.iter` maximum iteration. Default is 500.

`randomN` random seed for initialisation (if uij or vi is NA). Default is 0.

`vi.dist` a string of centroid population distribution between `'uniform'` (default) and `'normal'`. Can be defined as `vi.dist=` in `opt_param`.

`npar` number of particles. Can be defined as `npar=` in `opt_param`. Default is 10.

`vmax` maximum velocity to be tolerated. Can be defined as `vmax` in `opt_param`. Default is 0.7

`pso.same` number of consecutive unchange to stop the iteration. Can be defined as `same=` in `opt_param`.

`c1` Cognitive scaling parameters. Can be defined as `c1=` in `opt_param`. Default is 0.49

`c2` Social scaling parameters. Can be defined as `c2=` in `opt_param`. Default is 0.49,

`w.inert` The inertia weight update method between `"constant"`, `"chaotic"`, `"sim.annealing"`, `"nat.exponent1"`, `"nat.exponent2"` based on Bansal (2011). Can be defined as `type=` in `opt_param`. Default is `'sim.annealing'`

`wmax` Maximum inertia weight. Can be defined as `wmax` in `opt_param`. Default is 0.9.

`wmin` Minimum inertia weight. Can be defined as `wmin` in `opt_param`. Default is 0.4.

`map` Chaotic mapping parameter. Userful when `w.inert='chaotic'`. Can be defined as `map` in `opt_param`. Default is 0.4.

Details

Fuzzy Geographically Weighted Clustering (FGWC) was developed by Mason and Jacobson (2007) by adding neighborhood effects and population to configure the membership matrix in Fuzzy C-Means. Furthermore, the Particle Swarm Optimization was developed by Kennedy and Eberhart (1995) in order to get a more optimal solution of a certain complex function. PSO was also improved by Bansal et al. (2011) by modifying the inertia weight. FGWC using PSO has been implemented previously by some studies (Wijayanto and Purwarianti 2014; Putra and Kurniawan 2017; Abdussamad 2020).

Value

an object of class `'fgwc'`.
An `'fgwc'` object contains as follows:

- `converg` - the process convergence of objective function
- `f_obj` - objective function value
- `membership` - membership matrix
- `centroid` - centroid matrix
- `validation` - validation indices (there are partition coefficient (`PC`), classification entropy (`CE`), SC index (`SC`), separation index (`SI`), Xie and Beni's index (`XB`), IFV index (`IFV`), and Kwon index (Kwon))
- `max.iter` - Maximum iteration
- `cluster` - the cluster of the data
- `finaldata` - The final data (with the cluster)
- `call` - the syntax called previously
- `time` - computational time.

References

Abdussamad S (2020). "Evaluation of Implementation Context Based Clustering In Fuzzy Geographically Weighted Clustering-Particle Swarm Optimization Algorithm." *Jurnal EECCIS*, **14**(1), 10–15. ISSN 2460-8122, https://jurnaleeccis.ub.ac.id/index.php/eeccis/article/view/609.

Bansal JC, Singh PK, Saraswat M, Verma A, Jadon SS, Abraham A (2011). "Inertia Weight strategies in Particle Swarm Optimization." In *2011 Third World Congress on Nature and Biologically Inspired Computing*. doi: 10.1109/nabic.2011.6089659, https://doi.org/10.1109/nabic.2011.6089659.

Kennedy J, Eberhart R (1995). "Particle swarm optimization." In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, 1942–1948. doi: 10.1109/ICNN.1995.488968, https://doi.org/10.1109/ICNN.1995.488968.

Mason GA, Jacobson RD (2007). "Fuzzy Geographically Weighted Clustering." In *Proceedings of the 9th International Conference on Geocomputation*, 1–7.

Putra FH, Kurniawan R (2017). "Clustering for Disaster Areas Endemic Dengue Hemorrhagic Fever Based on Factors had Caused in East Java Using Fuzzy Geographically Weighted Clustering - Particle Swarm Optimization." *Jurnal Aplikasi Statistika \& Komputasi Statistik*, **8**(01), 27. ISSN 2615-1367.

Wijayanto AW, Purwarianti A (2014). "Improvement of fuzzy geographically weighted clustering using particle swarm optimization." In *2014 International Conference on Information Technology Systems and Innovation (ICITSI)*, 7–12. ISBN 978-1-4799-6527-4.

See Also

fpafgwc gsafgwc

Examples

```
data('census2010')
data('census2010dist')
data('census2010pop')
# First way
res1 <- psofgwc(census2010,census2010pop,census2010dist,3,2,'minkowski',4,npar=10)
# Second way
# initiate parameter
param_fgwc <- c(kind='v',ncluster=3,m=2,distance='minkowski',order=3,
               alpha=0.5,a=1.2,b=1.2,max.iter=1000,error=1e-6,randomN=10)
## tune the PSO parameter
pso_param <- c(vi.dist='uniform',npar=15,
         vmax=0.8, pso.same=10, c1=0.7, c2=0.6, type='chaotic',
                     wmax=0.8,wmin=0.3,map=0.3)
##FGWC with PSO
res2 <- fgwc(census2010,census2010pop,census2010dist,'pso',param_fgwc,pso_param)
```

tlbofgwc — *Fuzzy Geographicaly Weighted Clustering with Teaching - Learning Based Optimization*

Description

Fuzzy clustering with addition of spatial configuration of membership matrix with centroid optimization using Teaching - Learning Based Algorithm.

Usage

```
tlbofgwc(
  data,
  pop = NA,
  distmat = NA,
  ncluster = 2,
  m = 2,
  distance = "euclidean",
  order = 2,
  alpha = 0.7,
  a = 1,
  b = 1,
  error = 1e-05,
  max.iter = 100,
  randomN = 0,
  vi.dist = "uniform",
  nstud = 10,
  tlbo.same = 10,
  nselection = 10,
  elitism = F,
  n.elite = 2
)
```

Arguments

`data`	an object of data with d>1. Can be `matrix` or `data.frame`. If your data is univariate, bind it with 1 to get a 2 columns.
`pop`	an n*1 vector contains population.
`distmat`	an n*n distance matrix between regions.
`ncluster`	an integer. The number of clusters.
`m`	degree of fuzziness or fuzzifier. Default is 2.
`distance`	the distance metric between data and centroid, the default is euclidean, see `cdist` for details.
`order,`	minkowski order. default is 2.

`alpha` the old membership effect with [0,1], if `alpha` equals 1, it will be same as fuzzy C-Means, if 0, it equals to neighborhood effect.

`a` spatial magnitude of distance. Default is 1.

`b` spatial magnitude of population. Default is 1.

`error` error tolerance. Default is 1e-5.

`max.iter` maximum iteration. Default is 500.

`randomN` random seed for initialisation (if uij or vi is NA). Default is 0.

`vi.dist` a string of centroid population distribution between `'uniform'` (default) and `'normal'`. Can be defined as `vi.dist=` in `opt_param`.

`nstud` number of students. Can be defined as `npar=` in `opt_param`. Default is 10.

`tlbo.same` number of consecutive unchange to stop the iteration. Can be defined as `same=` in `opt_param`. Default is 10.

`nselection` number of teachers based on selected students. Can be defined as `nselection=` in `opt_param`. Default is equal to `nstud`.

`elitism` wheter to use elitism algorithm or not. Either `TRUE` or `FALSE`. Can be defined as `elitism=` in `opt_param`. Default is `FALSE`.

`n.elite` Number of elitist students. Can be defined as `n.elite=` in `opt_param`. Default is 2.

Details

Fuzzy Geographically Weighted Clustering (FGWC) was developed by Mason and Jacobson (2007) by adding neighborhood effects and population to configure the membership matrix in Fuzzy C-Means. Furthermore, the Teaching - Learning Based Optimization was developed by Rao et al. (2012) and Developed by Rao and Patel (2012) by adding the elitism algorithm in order to get a more optimal solution of a certain complex function.

Value

an object of class `'fgwc'`.
An `'fgwc'` object contains as follows:

- `converg` - the process convergence of objective function
- `f_obj` - objective function value
- `membership` - membership matrix
- `centroid` - centroid matrix
- `validation` - validation indices (there are partition coefficient (`PC`), classification entropy (`CE`), SC index (`SC`), separation index (`SI`), Xie and Beni's index (`XB`), IFV index (`IFV`), and Kwon index (Kwon))
- `max.iter` - Maximum iteration
- `cluster` - the cluster of the data
- `finaldata` - The final data (with the cluster)
- `call` - the syntax called previously
- `time` - computational time.

References

Mason GA, Jacobson RD (2007). "Fuzzy Geographically Weighted Clustering." In *Proceedings of the 9th International Conference on Geocomputation*, 1–7.

Rao RV, Patel V (2012). "An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems." *International Journal of Industrial Engineering Computations*, **3**(4), 535–560. ISSN 19232926, doi: 10.5267/j.ijiec.2012.03.007, `https://doi.org/10.5267/j.ijiec.2012.03.007`.

Rao RV, Savsani VJ, Balic J (2012). "Teaching\- learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems." *Engineering Optimization*, **44**(12), 1447–1462. doi: 10.1080/0305215x.2011.652103, `https://doi.org/10.1080/0305215x.2011.652103`.

See Also

`fpafgwc` `gsafgwc`

Examples

```
data('census2010')
data('census2010dist')
data('census2010pop')
# First way
res1 <- tlbofgwc(census2010,census2010pop,census2010dist,3,2,'minkowski',4,nstud=10)
# Second way
# initiate parameter
param_fgwc <- c(kind='v',ncluster=3,m=2,distance='minkowski',order=3,
               alpha=0.5,a=1.2,b=1.2,max.iter=1000,error=1e-6,randomN=10)
## tune the TLBO parameter
tlbo_param <- c(vi.dist="uniform",nstud=10, tlbo.same=10,
         nselection=10,elitism=FALSE,n.elite=2)
##FGWC with TLBO
res2 <- fgwc(census2010,census2010pop,census2010dist,'tlbo',param_fgwc,tlbo_param)
```

Index