Package 'orclus'

July 22, 2025

Version 0.2-6

Date 2018-03-17

Title Subspace Clustering Based on Arbitrarily Oriented Projected Cluster Generation

Author Gero Szepannek

Maintainer Gero Szepannek <gero.szepannek@web.de>

Description Functions to perform subspace clustering and classification.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2018-03-17 22:55:34 UTC

Contents

`orclass` *Subspace clustering based local classification using ORCLUS.*

Description

Function to perform local classification where the subclasses are concentrated in different subspaces of the data.

Usage

```
orclass(x, ...)
## Default S3 method:
orclass(x, grouping, k, l, k0, a = 0.5, prior = NULL, inner.loops = 1,
                          predict.train = "nearest", verbose = TRUE, ...)
## S3 method for class 'formula'
orclass(formula, data = NULL, ...)
```

Arguments

x	A matrix or data frame containing the explanatory variables. The method is restricted to numerical data.
grouping	A factor specifying the class for each observation.
formula	A formula of the form grouping ~ x1 + x2 + ... That is, the response is the grouping factor and the right hand side specifies the (non-factor) discriminators.
data	Data frame from which variables specified in formula are to be taken.
k	Prespecifies the final number of clusters.
l	Prespecifies the dimension of the final cluster-specific subspaces (equal for all clusters).
k0	Initial number of clusters (that are computed in the entire data space). Must be greater than k. The number of clusters is iteratively decreased by factor a until the final number of k clusters is reached.
a	Prespecified factor for the cluster number reduction in each iteration step of the algorithm.
prior	Argument for optional specification of class prior probabilities if different from the relative class frequencies.
inner.loops	Number of repetitive iterations (i.e. recomputation of clustering and cluster-specific subspaces) while the number of clusters and the subspace dimension are kept constant.
predict.train	Character pecifying whether prediction of training data should be pursued. If "nearest" the class distribution in orclus cluster assignment is used for classification.
verbose	Logical indicating whether the iteration process sould be displayed.
...	Currently not used.

Details

For each cluster the class distribution is computed.

Value

Returns an object of class orclass.

orclus.res	Object of class orclus containing the resulting clusters.

`cluster.posteriors`	Matrix of clusterwise class posterior probabilities where clusters are rows and classes are coloumns.
`cluster.priors`	Vector of relative cluster frequencies weighted by class priors.
`purity`	Statistics indicating the discriminability of the identified clusters.
`prior`	Vector of class prior probabilities.
`predict.train`	Prediction of training data if specified.
`orclass.call`	(Matched) function call.

Author(s)

Gero Szepannek

References

Aggarwal, C. and Yu, P. (2000): *Finding generalized projected clusters in high dimensional spaces*, Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 70-81.

See Also

`predict.orclass`, `orclus`, `predict.orclus`

Examples

```
# definition of a function for parameterized data simulation
sim.orclus <- function(k = 3, nk = 100, d = 10, l = 4,
                       sd.cl = 0.05, sd.rest = 1, locshift = 1){
  ### input parameters for data generation
  # k           number of clusters
  # nk          observations per cluster
  # d           original dimension of the data
  # l           subspace dimension where the clusters are concentrated
  # sd.cl       (within cluster subspace) standard deviations for data generation
  # sd.rest     standard deviations in the remaining space
  # locshift    parameter of a uniform distribution to sample different cluster means

  x <- NULL
  for(i in 1:k){
  # cluster centers
  apts <- locshift*matrix(runif(l*k), ncol = l)
  # sample points in original space
 xi.original <- cbind(matrix(rnorm(nk * l, sd = sd.cl), ncol=l) + matrix(rep(apts[i,], nk),
                              ncol = l, byrow = TRUE),
                      matrix(rnorm(nk * (d-l), sd = sd.rest), ncol = (d-l)))
  # subspace generation
  sym.mat <- matrix(nrow=d, ncol=d)
  for(m in 1:d){
    for(n in 1:m){
      sym.mat[m,n] <- sym.mat[n,m] <- runif(1)
      }
```

```
  }
  subspace <- eigen(sym.mat)$vectors
  # transformation
  xi.transformed <- xi.original %*% subspace
  x <- rbind(x, xi.transformed)
  }
  clids <- rep(1:k, each = nk)
  result <- list(x = x, cluster = clids)
  return(result)
  }

# simulate data of 2 classes where class 1 consists of 2 subclasses
simdata <- sim.orclus(k = 3, nk = 200, d = 15, l = 4,
                      sd.cl = 0.05, sd.rest = 1, locshift = 1)

x <- simdata$x
y <- c(rep(1,400), rep(2,200))

res <- orclass(x, y, k = 3, l = 4, k0 = 15, a = 0.75)
res

# compare results
table(res$predict.train$class, y)
```

`orclus` *Arbitrarily ORiented projected CLUSter generation*

Description

Function to perform subspace clustering where the clusters are concentrated in different cluster specific subspaces of the data.

Usage

```
orclus(x, ...)
## Default S3 method:
orclus(x, k, l, k0, a = 0.5, inner.loops = 1, verbose = TRUE, ...)
```

Arguments

`x`	A matrix or data frame containing the explanatory variables. The method is restricted to numerical data.
`k`	Prespecifies the final number of clusters.
`l`	Prespecifies the dimension of the final cluster-specific subspaces (equal for all clusters).
`k0`	Initial number of clusters (that are computed in the entire data space). Must be greater than `k`. The number of clusters is iteratively decreased by factor `a` until the final number of `k` clusters is reached.

a	Prespecified factor for the cluster number reduction in each iteration step of the algorithm.
inner.loops	Number of repetitive iterations (i.e. recomputation of clustering and cluster-specific subspaces) while the number of clusters and the subspace dimension are kept constant.
verbose	Logical indicating whether the iteration process sould be displayed.
...	Currently not used.

Details

The function performs ORCLUS subspace clustering (Aggarwal and Yu, 2000). Simultaneously both cluster assignments as well as cluster specific subspaces are computed. Cluster assignments have minimal euclidean distance from the cluster centers in the corresponding subspaces. As an extension to the originally proposed algorithm initialization in the full data space is done by calling kmeans for k0 clusters. Further, by inner.loops a number of repetitions during the iteration process for each number of clusters and subspace dimension can be specified. An outlier option has not been implemented. Even though increasing the initialzation parameter k0 most strongly effects the computation time it should be chosen as large as possible (at least several times greater then k).

Value

Returns an object of class orclus. Its structure is similar to objects resulting from calling kmeans.

cluster	Returns the final cluster labels.
centers	A matrix where each row corresponds to a cluster center (in the original space).
size	The final number of observations in each cluster.
subspaces	List of matrices for projection of the data onto the cluster-specific supspaces by post-multiplication.
subspace.dimension	Dimension of the final subspaces.
within.projenss	Corresponds to withinss of kmeans objects: projected within cluster energies for each cluster.
sparsity.coefficient	Sparsity coefficient of the clustering result. If its value is close to 1 the subspace dimension may have been chosen too large. A small value close to 0 can be interpreted as a hint that a strong cluster structure has been found.
orclus.call	(Matched) function call.

Author(s)

Gero Szepannek

References

Aggarwal, C. and Yu, P. (2000): *Finding generalized projected clusters in high dimensional spaces*, Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 70-81.

See Also

predict.orclus

Examples

```
# generate simple artificial example of two clusters
clus1.v1 <- runif(100)
clus2.v1 <- runif(100)
xample <- rbind(cbind(clus1.v1, 0.5 - clus1.v1), cbind(clus2.v1, -0.5 + clus2.v1))
plot(xample, col=rep(1:2, each=100))

# try standard kmeans clustering
kmeans.res <- kmeans(xample, 2)
plot(xample, col = kmeans.res$cluster)

# use orclus instead
orclus.res <- orclus(x = xample, k = 2, l = 1, k0 = 8, a = 0.5)
plot(xample, col = orclus.res$cluster)

# show data in cluster-specific subspaces
par(mfrow=c(1,2))
for(i in 1:length(orclus.res$size)) plot(xample %*% orclus.res$subspaces[[i]],
    col = orclus.res$cluster, ylab = paste("Identified subspace for cluster",i))


### second 'more multivariate' example to play with...

# definition of a function for parameterized data simulation
sim.orclus <- function(k = 3, nk = 100, d = 10, l = 4,
                       sd.cl = 0.05, sd.rest = 1, locshift = 1){
  ### input parameters for data generation
  # k           number of clusters
  # nk          observations per cluster
  # d           original dimension of the data
  # l           subspace dimension where the clusters are concentrated
  # sd.cl       (within cluster subspace) standard deviations for data generation
  # sd.rest     standard deviations in the remaining space
  # locshift    parameter of a uniform distribution to sample different cluster means

  x <- NULL
  for(i in 1:k){
  # cluster centers
  apts <- locshift*matrix(runif(l*k), ncol = l)
  # sample points in original space
 xi.original <- cbind(matrix(rnorm(nk * l, sd = sd.cl), ncol=l) + matrix(rep(apts[i,], nk),
                              ncol = l, byrow = TRUE),
                       matrix(rnorm(nk * (d-l), sd = sd.rest), ncol = (d-l)))
  # subspace generation
  sym.mat <- matrix(nrow=d, ncol=d)
  for(m in 1:d){
    for(n in 1:m){
      sym.mat[m,n] <- sym.mat[n,m] <- runif(1)
```

```
      }
    }
  subspace <- eigen(sym.mat)$vectors
  # transformation
  xi.transformed <- xi.original %*% subspace
  x <- rbind(x, xi.transformed)
  }
  clids <- rep(1:k, each = nk)
  result <- list(x = x, cluster = clids)
  return(result)
  }

# simulate data, you can play with different parameterizations...
simdata <- sim.orclus(k = 3, nk = 200, d = 15, l = 4,
                      sd.cl = 0.05, sd.rest = 1, locshift = 1)

# apply kmeans and orclus
kmeans.res2 <- kmeans(simdata$x, 3)
orclus.res2 <- orclus(x = simdata$x, k = 3, l = 4, k0 = 15, a = 0.75)
cat("SC: ", orclus.res2$sparsity.coefficient, "\n")

# compare results
table(kmeans.res2$cluster, simdata$cluster)
table(orclus.res2$cluster, simdata$cluster)
```

`predict.orclass` *Subspace clustering based local classification using ORCLUS.*

Description

Assigns clusters and distances and classes for new data according to the intrinsic subspace clusters of an `orclass` classification model.

Usage

```
## S3 method for class 'orclass'
predict(object, newdata, type = "nearest", ...)
```

Arguments

`object`	Model resulting from a call of `orclass`.
`newdata`	A matrix or data frame to be clustered by the given model.
`type`	Default `"nearest"` computes relative class frequencies of nearest cluster as class posterior probabilities.
`...`	Currently not used.

Details

For prediction the class distribution of the "nearest"" cluster is used. If type = "fuzzywts" cluster memberships (see e.g. Bezdek, 1981) are computed based on the cluster distances of cluster assignment by predict.orclus. For orclass prediction the class distributions of the clusters are weigthed using the cluster memberships of an observation.

Value

class	Vector of predicted class levels.
posterior	Matrix where coloumns contain class posterior probabilities.
distances	A matrix where coloumns are the distances to all cluster centers in the corresponding subspaces for the new data.
cluster	The resulting cluster labels for the new data.

Author(s)

Gero Szepannek

References

Aggarwal, C. and Yu, P. (2000): *Finding generalized projected clusters in high dimensional spaces*, Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 70-81.

Bezdek, J. (1981): *Pattern recognition with fuzzy objective function algorithms*, Kluwer Academic, Norwell, MA.

See Also

orclass, orclus, predict.orclus

Examples

```
# definition of a function for parameterized data simulation
sim.orclus <- function(k = 3, nk = 100, d = 10, l = 4,
                       sd.cl = 0.05, sd.rest = 1, locshift = 1){
  ### input parameters for data generation
  # k           number of clusters
  # nk          observations per cluster
  # d           original dimension of the data
  # l           subspace dimension where the clusters are concentrated
  # sd.cl       (within cluster subspace) standard deviations for data generation
  # sd.rest     standard deviations in the remaining space
  # locshift    parameter of a uniform distribution to sample different cluster means

  x <- NULL
  for(i in 1:k){
  # cluster centers
  apts <- locshift*matrix(runif(l*k), ncol = l)
  # sample points in original space
 xi.original <- cbind(matrix(rnorm(nk * l, sd = sd.cl), ncol=l) + matrix(rep(apts[i,], nk),
```

```
                              ncol = l, byrow = TRUE),
                        matrix(rnorm(nk * (d-l), sd = sd.rest), ncol = (d-l)))
  # subspace generation
  sym.mat <- matrix(nrow=d, ncol=d)
  for(m in 1:d){
    for(n in 1:m){
      sym.mat[m,n] <- sym.mat[n,m] <- runif(1)
      }
    }
  subspace <- eigen(sym.mat)$vectors
  # transformation
  xi.transformed <- xi.original %*% subspace
  x <- rbind(x, xi.transformed)
  }
  clids <- rep(1:k, each = nk)
  result <- list(x = x, cluster = clids)
  return(result)
  }

# simulate data of 2 classes where class 1 consists of 2 subclasses
simdata <- sim.orclus(k = 3, nk = 200, d = 15, l = 4,
                      sd.cl = 0.05, sd.rest = 1, locshift = 1)

x <- simdata$x
y <- c(rep(1,400), rep(2,200))

res <- orclass(x, y, k = 3, l = 4, k0 = 15, a = 0.75)
prediction <- predict(res, x)

# compare results
table(prediction$class, y)
```

`predict.orclus` *Arbitrarily ORiented projected CLUSter generation*

Description

Assigns clusters and distances to cluster centers in the corresponding subspaces for new data according to a subspace clustering model of class `orclus`.

Usage

```
## S3 method for class 'orclus'
predict(object, newdata, ...)
```

Arguments

`object`	Model resulting from a call of `orclus`.
`newdata`	A matrix or data frame to be clustered by the given model.
`...`	Currently not used.

Value

distances	A matrix where coloumns are the distances to all cluster centers in the corresponding subspaces for the new data.
cluster	The resulting cluster labels for the new data.

Author(s)

Gero Szepannek

References

Aggarwal, C. and Yu, P. (2000): *Finding generalized projected clusters in high dimensional spaces*, Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 70-81.

See Also

orclus

Examples

```
# generate simple artificial example of two clusters
clus1.v1 <- runif(100)
clus2.v1 <- runif(100)
xample <- rbind(cbind(clus1.v1, 0.5 - clus1.v1), cbind(clus2.v1, -0.5 + clus2.v1))

orclus.res <- orclus(x = xample, k = 2, l = 1, k0 = 8, a = 0.5)

# generate new data and predict it using the
newclus1.v1 <- runif(100)
newclus2.v1 <- runif(100)
true.clusterids  <- rep(1:2, each = 100)
xample2 <- rbind(cbind(newclus1.v1, 0.5 - newclus1.v1),
                 cbind(newclus2.v1, -0.5 + newclus2.v1))

orclus.prediction <- predict(orclus.res, xample2)
table(orclus.prediction$cluster, true.clusterids)
```

Index