Marek Walesiak[*]

CLUSTER ANALYSIS WITH clusterSim COMPUTER PROGRAM AND R ENVIRONMENT

ABSTRACT. The article presents auxiliary functions of clusterSim package (see Walesiak & Dudek (2006)) and selected functions of packages stats, cluster, and ade4, which are applied to solving clustering problems. In addition, the examples of the procedures for solving different clustering problems are presented. These procedures, which are not available in statistical packages (SPSS, Statistica, SAS), can help solving a broad range of classification problems.

Key words: cluster analysis, R, clusterSim, data analysis.

I. INTRODUCTION

In a typical cluster analysis study seven major steps are distinguished (see Milligan (1996), 342–343): selection of objects and variables, decisions concerning variable normalization, selection of a distance measure, selection of clustering method, determining the number of clusters, cluster validation, describing and profiling clusters. The article presents functions of clusterSim package and selected functions of packages stats, cluster, and ade4, which are applied to solving clustering problems.

II. THE PACKAGES AND FUNCTIONS OF R COMPUTER PROGRAM IN A TYPICAL CLUSTER ANALYSIS PROCEDURE

Table 1 contains selected packages and functions of R program applied on each step of typical cluster analysis study.

[*] Professor, Chair of Department of Econometrics and Computer Science, Wrocław University of Economics.

Table 1

The packages and functions of R computer program in a typical cluster analysis study

No.	Steps in a typical cluster analysis study	Selected packages	Functions
1	Selection of objects and variables	clusterSim	HINoV.Mod
2	Decisions concerning variable normalization	clusterSim	data.Normalization
3	Selection of a distance measure	clusterSim stats ade4	dist.BC, dist.GDM, dist.SM dist dist.binary
4	Selection of clustering method	cluster stats clusterSim	agnes, diana, pam kmeans, hclust initial.Centers
5	Determining the number of clusters	clusterSim	index.G1, index.G2, index.G3, index.S, index.KL, index.H, index.Gap
6	Cluster validation	clusterSim	replication.Mod
7	Describing and profiling clusters	clusterSim	cluster.Description

Source: own presentation.

Step 1. Selection of objects and variables. Carmone, Kara, and Maxwell (1999) proposed the Heuristic Identification of Noisy Variables (*HINoV*) method based on *k*-means cluster analysis on each variable and corrected Rand index for each resulting pair of partitions. The *HINoV* algorithm can identify noisy variables in a data set and yield better cluster recovery. As a result of this algorithm, we receive the contribution of each variable to cluster structure. Package clusterSim contains extended version of *HINoV* method for nonmetric data:

```
HINoV.Mod(x, type="metric", s=2, u, distance=NULL,
    method="kmeans", Index="cRAND")
```

where:

x – data matrix;

s – for metric data (1 – ratio; 2 – interval or mixed);

u – number of clusters (for metric data);

distance – NULL for kmeans and nonmetric data, for ratio data ("d1" – Manhattan, "d2" – Euclidean, "d3" – Chebychev (maximum), "d4" – squared Euclidean, "d5" – GDM1, "d6" – Canberra, "d7" – Bray & Curtis), for interval and mixed data ("d1", "d2", "d3", "d4", "d5");

method – classification method: "kmeans" (default) , "single", "complete", "average", "mcquitty", "median", "centroid", "Ward", "pam" (NULL for nonmetric data);
Index–"cRAND" – corrected Rand index, "RAND" – Rand index.

Step 2. Decisions concerning variable normalization. Function data.Normalization (x, type="n0") calculates normalization data using the formula of variable normalization n0 – n11 for data matrix x (n0 – without normalization, n1 – standardization, n2 – Weber standardization, n3 – unitization, n4 – unitization with zero minimum, n5 – normalization with range [–1; 1], n6–n11 – quotient transformations with different base) – details see Walesiak (2006).

Step 3. Selection of a distance measure. The packages clusteSim, stats and ade4 contain distance measures for metric and nonmetric data (see Table 2).

Table 2

Distance measures for metric and nonmetric data

Package	Syntax
clusterSim	dist.GDM (x, method="GDM1") – function calculates Generalized Distance Measure for variables measured on metric scale (GDM1) or ordinal scale (GDM2) dist.BC (x) – function calculates the Bray-Curtis distance measure for ratio data dist.SM(x) – function calculates the Sokal-Michener distance measure for nominal variables
stats	dist(x, method="euclidean", p = 2) x data matrix or "dist" object method distance measure: "euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski" p the power for the Minkowski distance
ade4	dist.binary(df, method = NULL) df a data frame with positive or zero values. Used with as.matrix(1*(df>0)) method an integer between 1 and 10 (distance measure $d = \sqrt{1-s}$): 1 = Jaccard, 2 = Sokal & Michener, 3 = Sokal & Sneath (1), 4 = Rogers & Tanimoto, 5 = Czekanowski, 6 = Gower & Legendre (1), 7 = Ochiai, 8 = Sokal & Sneath (2), 9 = Phi of Pearson, 10 = Gower & Legendre (2)

Source: own presentation.

Step 4. Selection of clustering method. The most frequently applied clustering methods are available in packages stats (hclust – hierarchical agglomerative methods; kmeans – *k*-means method) and cluster (pam – partitioning around medoids; agnes – hierarchical agglomerative methods; diana – hierarchical divisive method). Example syntax for function kmeans for clustering data:

```
kmeans(x, centers, iter.max = 10, nstart = 1,
  algorithm = c("Hartigan-Wong", "Lloyd", "Forgy",
  "MacQueen"))
```

where: x – data matrix; centers – either the number of clusters or a set of initial cluster centers; iter.max – the maximum number of iterations allowed; nstart – if centers is a number, how many random sets should be chosen?; algorithm – applied algorithm.

Function initial.Centers(x,k) of clusterSim package calculates initial cluster centers for *k*-means algorithm (x – data matrix, k – number of initial cluster centers).

Step 5. Determining the number of clusters. Package clusterSim contains seven cluster quality indices necessary in determination of the number of clusters in a data set (Calinski & Harabasz, Baker & Hubert, Hubert & Levine, Silhouette, Krzanowski & Lai, Hartigan, gap). For example function index.H (x,clall) calculates Hartigan index for data matrix x and two vectors of integers clall indicating the cluster to which each object is allocated in partition of n objects into u, and $u+1$ clusters (details and others indices see Walesiak (2007)).

Step 6. Cluster validation. In replication analysis (see Breckenridge (2000)) we compare the results of classification of two random samples obtained from a data set. The level of agreement between the two partitions (mean corrected Rand index) reflects the stability of the clustering in the data. Package clusterSim contains replication.Mod function:

```
replication.Mod(x, v="m", u=2,
    centrotypes="centroids",
    normalization=NULL, distance=NULL,
    method="kmeans",
    S=10, fixedAsample=NULL)
```

where: x – data matrix, v – type of data: metric ("r" – ratio, "i" – interval, "m" – mixed), nonmetric ("o" – ordinal, "n" – multistate nominal, "b" – binary), u – number of clusters, centrotypes – "centroids", "medoids"; normalization – normalization formula n1-n11 (see stage 2); distance – NULL for "kmeans", distance measure (see stage 3); method – classification method (see stage 4); S – number of simulations; fixedAsample – if NULL A sample is generated randomly, otherwise this parameter contains object numbers arbitrarily assigned to A sample.

Step 7. Describing and profiling clusters. Function cluster.Description (x,cl) of clusterSim package calculates descriptive statistics separately for each cluster and variable in classification cl: arithmetic mean and standard deviation, median and median absolute deviation, mode.

III. THE EXAMPLE PROCEDURES WITH SELECTED FUNCTIONS OF R PACKAGES

The 75 observations were generated from standard two-dimensional spherical normal distribution into five clusters of size 15 each with means: $\mu_1 = [0 \quad 0]^T$, $\mu_2 = [0 \quad 10]^T$, $\mu_3 = [5 \quad 5]^T$, $\mu_4 = [10 \quad 0]^T$, $\mu_5 = [10 \quad 10]^T$, and covariance matrices: $\Sigma_1 = \Sigma_2 = \Sigma_3 = \Sigma_4 = \Sigma_5 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. In addition, three noisy variables are included in the model to obscure the underlying clustering structure to be recovered. 75 observations for these variables were generated with means and covariance matrix: $\mu = [5 \quad 5 \quad 7{,}5]^T$, $\Sigma = \begin{bmatrix} 5 & 2 & 6 \\ 2 & 1 & -5 \\ 6 & -5 & 2 \end{bmatrix}$.

Finally, the data were standardized via formula „n1”. To help isolate noisy variables HINoV.Mod procedure was applied (see example 1).

Example 1

```
> library(cluster)
> library(clusterSim)
> x<-read.csv2("C:/Data_75x5.csv",
header=TRUE,strip.white=TRUE,row.names=1)
> x<-as.matrix(x)
> z<-data.Normalization (x, type="n1")
> z<-as.data.frame(z)
> r1<-HINoV.Mod(z, type="metric", s=2, 5,
method="kmeans",Index="cRAND")
> options(OutDec = ",")
> plot(r1$stopri[,2],type="p", pch=0, xlab="Number of
variable", ylab="topri",xaxt="n")
```

```
> axis(1,at=c(1:max(r1$stopri[,1])),
labels=r1$stopri[,1])
```

The result of this procedure is shown in Figure 1.

Based on scree diagram (Figure 1) three noisy variables v_3, v_4, and v_5 were eliminated via *HINoV* method.

In procedure of example 2 the following assumptions is taken into account:

– for clustering of 75 objects in two-dimensional space (file `data_75x2.csv`) the *k*-means method was applied,

– the estimated number of clusters is the smallest $u \in [2;10]$ such that $H(u) \leq 10$,

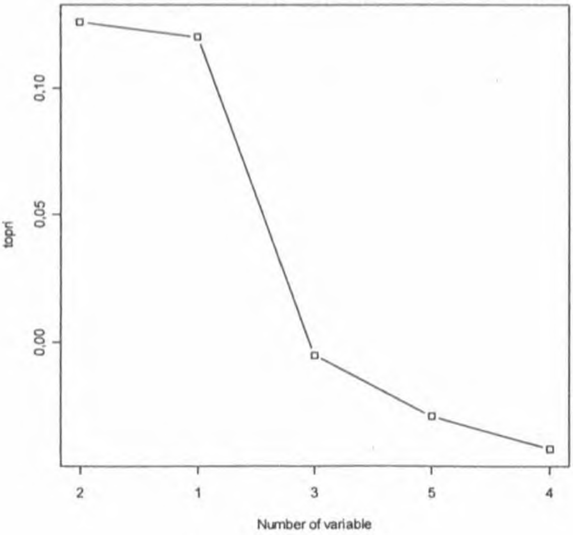

Figure 1. Scree diagram
Source: own research.

– `write.table` function allow to save results in files: values of index $H(u)$, a vector of integers indicating the cluster to which each object is allocated („cluster"), a matrix of cluster centers („centers"), the within-cluster sum of squares for each cluster(„withinss"), the number of objects in each cluster („size").

Example 2 (first six instructions from example 1).

```
> min_u=2
> max_u=10
> min <- 0
> results <- array(0,c(max_u-min_u+1, 2))
> results[,1] <- min_u:max_u
> find <- FALSE
> for (u in min_u:max_u)
> {
> cl1 <- kmeans(z, z[initial.Centers(z, u),])
> cl2 <- kmeans(z, z[initial.Centers(z, u+1),])
> clall<- cbind(cl1$cluster,cl2$cluster)
> results[u-min_u+1,2] <- H <- index.H(z,clall)
> if ((results[u-min_u+1,2]<10) &&(!find))
>  {
>    lk<-u
>    min<-H
>    clopt<-cl1
>    find<-TRUE
>  }
> }
> if (find)
> {
> print(paste("minimal u for H<=10 equals", lk, "for H
=",min))
> }else
> {
>    print("Classification not find")
> }
> write.table(results, file="C:/H_results.csv",
sep=";", dec=",", row.names=TRUE, col.names=FALSE)
> write.table(clopt$cluster, file="C:/cluster.csv",
sep=";", dec=",", row.names=TRUE, col.names=FALSE)
> write.table(clopt$centers, file="C:/centers.csv",
sep=";", dec=",",row.names=TRUE, col.names=FALSE)
> write.table(clopt$withinss, file="C:/withinss.csv",
sep=";", dec=",", row.names=TRUE, col.names=FALSE)
> write.table(clopt$size, file="C:/size.csv", sep=";",
dec=",", row.names=TRUE, col.names=FALSE)
> plot(results, type="p", pch=0, xlab="u",
ylab="H",xaxt="n")
> abline(h=10, untf = FALSE)
> axis(1,c(min_u:max_u))
```

The results of this procedure are following:

```
> [1] "minimal u for H<=10 equals 5 for H =
5,10784236355176"
```

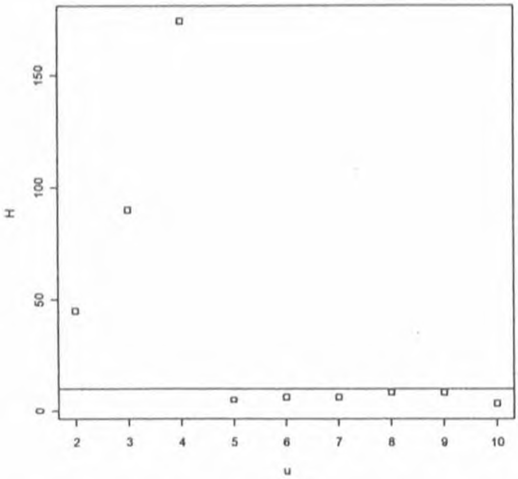

Figure 2. Graphical presentation of Hartigan H index

In example 3, the stability of the clustering in the data was done by replication analysis (function `replication.Mod` from `clusterSim` package).

Example 3.

```
> library(clusterSim)
> x<-read.csv2("C:/Data_75x2.csv",
header=TRUE,strip.white=TRUE,row.names=1)
> x <-as.matrix(x)
> x <-as.data.frame(x)
> options(OutDec = ",")
> w<-replication.Mod(x,v="m",u=5, centro-
types="centroids", normalization="n1",
method="kmeans",S=10, fixedAsample=NULL)
> print(w$cRand)
```

The result of this procedure is following:

```
> [1] 0,9794591
```

The high level of agreement between the two partitions reflects the stability of the clustering in the data.

IV. SUMMARY

In article, selected packages of R environment applied in seven major steps of cluster analysis study were presented. The selected functions of packages `clusterSim`, `stats`, `cluster`, and `ade4`, which are applied to solving clustering problems, were characterized. Additionally, the examples of the procedures for solving different clustering problems are presented which are not available in commercial statistical packages.

REFERENCES

Breckenridge J.N. (2000), *Validating cluster analysis: consistent replication and symmetry*, "Multivariate Behavioral Research", 35 (2), 261–285

Carmone F.J., Kara A., Maxwell S. (1999), *HINoV: a new method to improve market segment definition by identifying noisy variables*, "Journal of Marketing Research", November, vol. 36, 501–509.

Milligan G.W. (1996), *Clustering validation: results and implications for applied analyses*, W: P. Arabie, L.J. Hubert, G. de Soete (Eds.), *Clustering and classification*, World Scientific, Singapore, 341–375.

R Development Core Team (2007), *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, URL http://www. R-project.org.

Walesiak M. (2006), *Uogólniona miara odległości w statystycznej analizie wielowymiarowej*. Wydanie drugie rozszerzone. Wydawnictwo AE we Wrocławiu.

Walesiak M. (2007), *Wybrane zagadnienia klasyfikacji obiektów z wykorzystaniem programu komputerowego `clusterSim` dla środowiska R*, Prace Naukowe AE we Wrocławiu, nr 1169, 46–56.

Walesiak M., Dudek A. (2006), *Symulacyjna optymalizacja wyboru procedury klasyfikacyjnej dla danego typu danych – oprogramowanie komputerowe i wyniki badań*, Prace Naukowe AE we Wrocławiu nr 1126, 120–129.

Marek Walesiak

ZAGADNIENIA ANALIZY SKUPIEŃ Z WYKORZYSTANIEM PROGRAMU KOMPUTEROWEGO **clusterSim** I ŚRODOWISKA R

W artykule scharakteryzowano funkcje pomocnicze pakietu `clusterSim` oraz wybrane funkcje pakietów `stats`, `cluster` i `ade4` służące zagadnieniu analizy skupień. Ponadto zaprezentowano przykładowe procedury, wykorzystujące analizowane funkcje, ułatwiające potencjalnemu użytkownikowi realizację wielu zagadnień klasyfikacyjnych niedostępnych w podstawowych pakietach statystycznych (np. SPSS, Statistica, SAS).