Clustering symbolic objects described by discrete distributions

Vladimir Batagelj
Simona Korenjak-Černe
Nataša Kejžar

University of Ljubljana

1187. Sredin seminar, Ljubljana, October 27, 2010

Outline

History

- basic theory and algorithm: Simona's PhD thesis (1998-
- some applications:
  - Simona, Vlado: clustering of foods (2002?)
  - Simona, Barbara Japelj Pavešić (2004): TIMSS – analysis of ego-centered networks
  - Simona, Nataša, Vlado (2008): Population pyramids, Informatica
  - first problem: the *center of cluster is not the cluster's distribution*; we solved this problem in 2009 by introducing an appropriate criterion function
  - second problem: is it possible to adapt the hierarchical clustering method also for examples when the *number of cases in the distributions are different for different variables*? The positive answer was found this summer.

Problem

In the usual vector description of a data unit each component of the vector corresponds to a descriptor – *variable*. Variables can be measured in different scales: nominal, ordinal, numerical. When all variables are not of the same type we are dealing with so called *mixed* data.

One of the possible solutions of the *clustering problem with mixed units* is to choose an uniform description for such units. We selected the description of units with *discrete distributions* because such a description has many advantages: it enables us to deal with all types of variables, can be used to reduce large data set, and also preserves more detailed information about the data than the mean values. The description with distributions is a special kind of *symbolic object* (Billard and Diday, 2006).

Symbolic objects

For the description based on distributions the domain of each variable $V^j (j = 1, \cdots, m)$ is partitioned into k_j subsets $\{V_i^j,\ i = 1, \ldots k_j\}$.

For a cluster C we denote with $f(i, C; V^j)$ the *frequency* and with

$$\pi(i, C; V^j) = \frac{f(i, C; V^j)}{\operatorname{card}(C)}$$

the *relative frequency* of the values of variable V^j in the i-th subset V_i^j.

For all variables $V^j (j = 1, \cdots, m)$ holds

$$\sum_{i=1}^{k_j} \pi(i, C; V^j) = 1.$$

The description $C(V^j)$ of the cluster C for variable V^j is a distribution, e.g. the vector of the relative frequencies on the subsets V_i^j $(i = 1, \cdots, k_j)$.

Symbolic objects

The cluster C is described with the vector $\boldsymbol{C}$ of the distributions:

$$\begin{aligned} \boldsymbol{C} &= [C(V^1), \ldots, C(V^m)], \\ C(V^j) &= [\pi(1, C; V^j), \ldots, \pi(k_j, C; V^j)]. \end{aligned}$$

A data unit is considered as a special cluster with only one element.

Such a description has the following important properties:

- it requires a *fixed space* per variable;
- it is *compatible with merging* of disjoint clusters – knowing the description of clusters C_1 and C_2, $C_1 \cap C_2 = \emptyset$, we can, without additional information, produce the description of their union

$$\pi(i, C_1 \cup C_2; V) = \frac{\mathrm{card}(C_1)\,\pi(i, C_1; V) + \mathrm{card}(C_2)\,\pi(i, C_2; V)}{\mathrm{card}(C_1 \cup C_2)};$$

- it produces an *uniform description* for all types of variables.

Example – Ego-centered networks

In the TIMSS study for each ego (teacher) values of his/her variables $T_1, T_2, T_3, \ldots$ are presented with singular values of the appropriate subset numbers in their domains' partitions, but for their alters (students) each alters' variable $S_1, S_2, \ldots$ is represented with frequency distribution of alters' answers.

For example, the symbolic object corresponding to teacher with id 4567 is

$$\begin{array}{lccccccc} SO_{4567} = & [\,4, & 6, & 3, & \ldots & [\,47, 16, 37, 0,0], & [0,0,100,0], & \ldots] \\ & \uparrow & \uparrow & \uparrow & & \uparrow & \uparrow & \\ & T_1 & T_2 & T_3 & \ldots & S_1 & S_2 & \ldots \end{array}$$

The values of the alter's variable S_1 are: 1 = strongly agree, 2 = agree, 3 = disagree, and 4 = strongly disagree. Therefore: 47 students among all students of the teacher with id 4567 strongly agree with the statement S_1, 16 of students agree with the statement S_1, 37 of students disagree with the statement S_1, and none of students strongly disagree with the statement S_1. The last number in the distribution represents the number of missing values. There were no missing values on this question.

Clustering as optimization problem

We shall deal with the following optimization problem:

find a clustering $\mathbf{C}^*$ in the set of feasible clusterings Φ for which

$$P(\mathbf{C}^*) = \min_{\mathbf{C}\in\Phi} P(\mathbf{C})$$

with the *criterion function*

$$P(\mathbf{C}) = \sum_{C\in\mathbf{C}} p(C) \qquad \text{where} \qquad p(C) = p(C, L_C) = \sum_{X\in C} d(X, L_C).$$

The set of feasible clusterings Φ is a *set of partitions into k clusters* of the finite set of units $\mathcal{U}$.

Optimal leaders

The optimal leader L_C of the cluster C has to satisfy the following relation

$$L_C = \operatorname*{argmin}_{L \in \Psi} p(C, L),$$

where Ψ is the set of feasible representations of the clusters. In our approach, the leader L (a representative element) of the cluster C is also described as a *vector of distributions*

$$\begin{aligned} \boldsymbol{L} &= [L(V^1), \ldots, L(V^m)], \\ L(V^j) &= [\lambda(1, L; V^j), \ldots, \lambda(k_j, L; V^j)]. \end{aligned}$$

Dissimilarity

The first two dissimilarities between clusters/SOs were defined as a weighted sum of the dissimilarities on each variable:

$$d(C_1, C_2) = \sum_{j=1}^{m} \alpha_j \, d(C_1, C_2; V^j), \qquad \sum_{j=1}^{m} \alpha_j = 1$$

where $d(C_1, C_2; V^j)$ is either

$$d_{abs}(C_1, C_2; V^j) = \frac{1}{2} \sum_{i=1}^{k_j} |\pi(i, C_1; V^j) - \pi(i, C_2; V^j)| \quad \text{or}$$

$$d_{sqr}(C_1, C_2; V^j) = \frac{1}{2} \sum_{i=1}^{k_j} (\pi(i, C_1; V^j) - \pi(i, C_2; V^j))^2.$$

Here, $\alpha_j \geq 0$ $(j = 1, \ldots, m)$ denote weights, which could be equal for all variables or different if we have same additional information about the *importance of the variables*. The weights α_j can also be used to consider associations among variables discovered with other statistical methods. The dissimilarity d_{sqr} has range $[0, 1]$.

In the applications we mainly used the dissimilarity d_{sqr}.

Optimal representative of a cluster

Theorem 1 *For the criterion function P_{sqr} with dissimilarity d_{sqr} the optimal leaders are uniquely determined with the averages of relative frequencies*

$$\lambda(i, L; V) = \frac{1}{\text{card}(C)} \sum_{X \in C} \pi(i, X; V).$$

$\lambda(i, L; V)$ *is also a distribution.*

As was already mentioned, the problem with this optimal distribution $\lambda(i, L; V)$ is that it is not also the distribution of the corresponding cluster; but a kind of *'average shape' of joint distributions*.

Another criterion function

Let n_{iu} denote the number of persons in the i-th age group in cluster C_u and $n_u = \sum_i n_{ui}$ the number of all persons in cluster C_u.

Assume that the cluster C_u is described by the distribution p_{ui}, where

$$p_{ui} = \frac{n_{ui}}{n_u} \tag{1}$$

We would like to use a clustering method such that for the distribution $p_{(uv)i}$ describing the cluster $C_{(uv)}$, obtained by joining clusters C_u and C_v, holds

$$p_{(uv)i} = \frac{n_{(uv)i}}{n_{(uv)}} \tag{2}$$

It turns out that the generalized Ward's method (Batagelj V., 1988) with criterion function of the form

$$p(C) = \min_l \sum_{X \in C} w_X (x - l)^2 \tag{3}$$

satisfies this requirement.

. . . Another criterion function

The minimum t_C of expression (3) has to satisfy the equation

$$\frac{\partial p(C)}{\partial t} = -2 \sum_{X \in C} w_X (x - t) = 0 \tag{4}$$

Therefore

$$t_C = \frac{\sum_{X \in C} w_X \cdot x}{\sum_{X \in C} w_X} \tag{5}$$

Let $x \equiv p_{iu}$ and $w_X \equiv n_u$. Then by equation (5) for clusters C_u and C_v holds

$$t_{i(uv)} = \frac{n_u p_{iu} + n_v p_{iv}}{n_u + n_v} = \frac{n_{iu} + n_{iv}}{n_{(uv)}} = \frac{n_{i(uv)}}{n_{(uv)}} = p_{i(uv)} \tag{6}$$

As we see the center $t_{i(uv)}$ has the required property (2).

In the following we shall use the cluster error function

$$p(C) = p(C, L_C) = \sum_{X \in C} \sum_{j=1}^{m} \alpha_j w(X, V^j) d_{sqr}(X, L_C; V^j)$$

because it allows us a general analysis of both cases. We get the *standard case* for $w(X, V^j) = 1$, and the *generalized case* for $w(X, V^j) = n(X, V^j)$.

Methods

Classical clustering methods face two *problems*: hierarchical methods are limited to *small number of units*; and nonhierarchical methods are mostly limited to *units described with numbers* and use for the cluster's representation only one value (usually the center of the cluster). Focusing on the problem of clustering large data sets, we adapted the well known leaders method for the selected uniform representations of units and clusters. To reveal the internal structure of the reduced data set we adapted also the agglomerative hierarchical clustering method. We proved that for the selected dissimilarity, the adapted method is an adapted version of the Ward's hierarchical clustering method (Ward 1963).

Leaders method and Ward's method are *compatible* – they are solving the same optimization problem.

Leaders method

One of the most popular clustering methods for large data sets is k-means method, which is a special version of the leaders method (Hartigan 1975, page 74). k-means method can be applied only on numerical variables. The leaders method as a variant of the dynamic clustering method (Diday 1979) can be described with the following procedure:

```
determine an initial clustering
repeat
    determine leaders of the clusters in the current clustering;
    assign each unit to the nearest new leader – producing a new clustering
until the leaders stabilize.
```

The *initial clustering* can be obtained randomly with selected number of clusters k or can be determined from the units by selection of the maximal allowed dissimilarity between the unit and the nearest leader.

Optimal representative in general case

We already saw that the optimal leaders of the clusters are the corresponding *weighted averages* (centers of gravity) of the units from the cluster.

Given a cluster C the corresponding leader T_C is the solution of the problem

$$p(C) = \min_L \sum_{X \in C} d(X, L) = \sum_j \alpha_j \min_{l_j} \sum_{X \in C} d(x_j, l_j)$$

therefore

$$l_{ij}^* = \operatorname*{argmin}_{l_{ij}} \sum_{X \in C} d(x_{ij}, l_{ij})$$

For the selected dissimilarity $d(x_j, l_j) = w_{jx}(p_{jx} - l_j)^2$ the solution is

$$l_j^* = \frac{\sum_{X \in C} w_{jx} \cdot p_{jx}}{\sum_{X \in C} w_{jx}}$$

Optimal clustering in general case

Given leaders $\mathbf{L}$ the corresponding optimal clustering $\mathbf{C}^*$ is determined from

$$P(\mathbf{C}^*) = \sum_{X \in \mathcal{U}} \min_{L \in \mathbf{L}} d(X, L) = \sum_{X \in \mathcal{U}} d(X, L_{c^*(X)}) \qquad (7)$$

where

$$c^*(X) = \operatorname*{argmin}_i d(X, L_i)$$

We assign each unit X to the closest leader $L_i \in \mathbf{L}$.

Note that $d(X, L_i)$ is a ”generalized” dissimilarity.

Agglomerative hierarchical clustering

The standard agglomerative hierarchical clustering method is described with the following procedure:

```
each unit forms a cluster: $\mathbf{C}_n = \{\{X\} \colon X \in \mathbf{U}\}$ ;
they are at level 0: $h(\{X\}) = 0,\ X \in \mathbf{U}$ ;
for $k = n - 1$ to $1$ do
  determine the closest pair of clusters
    $(p, q) = \mathrm{argmin}_{i,j:i \neq j}\{D(C_i, C_j) \colon C_i, C_j \in \mathbf{C}_{k+1}\}$ ;
  join the closest pair of clusters $C_{(pq)} = C_p \cup C_q$
    $\mathbf{C}_k = (\mathbf{C}_{k+1} \setminus \{C_p, C_q\}) \cup \{C_{(pq)}\}$ ;
    $h(C_{(pq)}) = D(C_p, C_q)$
    determine the dissimilarities $D(C_{(pq)}, C_s), C_s \in \mathbf{C}_k$
endfor
```

$\mathbf{C}_k$ is a partition of the finite set of units $\mathbf{U}$ into k clusters. The level $h(C)$ of the cluster $C_{(pq)} = C_p \cup C_q$ is determined by the dissimilarity between the joint clusters C_p and C_q by $h(C_{(pq)}) = D(C_p, C_q)$.

Dissimilarity between clusters

To obtain the compatibility with the adapted leaders method, we define the dissimilarity between clusters as

$$D(C_u, C_v) = p(C_u \cup C_v) - p(C_u) - p(C_v).$$

For disjoint clusters C_u and C_v, $C_u \cap C_v = \emptyset$, dissimilarity $D(C_u, C_v)$ can be calculated from the *weighted dissimilarity of clusters' representatives* with the formulae

$$D(C_u, C_v) = \sum_j \alpha_j \frac{A_j \cdot B_j}{A_j + B_j}(u_j - v_j)^2$$

where $A_j = \sum_{X \in C_u} w_{jx}$ and $B_j = \sum_{X \in C_v} w_{jx}$; and

$$u_j = \frac{\sum_{X \in C_u} w_{jx} \cdot p_{jx}}{A_j} \quad \text{and} \quad v_j = \frac{\sum_{X \in C_v} w_{jx} \cdot p_{jx}}{B_j}.$$

This is a generalization of Ward's relation.

Note: this relations holds also for singletons $D(\{X\}, \{Y\}),\ X, Y \in \mathcal{U}$.

Proof of the generalized Ward's relation

$$D(C_u, C_v) = p(C_u \cup C_v) - p(C_u) - p(C_v) =$$

$$\sum_j \alpha_j \sum_{X \in C_u \cup C_v} w_{jx}(p_{jx} - z_j)^2 - \sum_j \alpha_j \sum_{X \in C_u} w_{jx}(p_{jx} - u_j)^2 - \sum_j \alpha_j \sum_{X \in C_v} w_{jx}(p_{jx} - v_j)^2.$$

Since $C_u \cap C_v = \emptyset$ and finite sums, this is equal to

$$\sum_{X \in C_u} \sum_j \alpha_j w_{jx}((p_{jx} - z_j)^2 - (p_{jx} - u_j)^2) + \sum_{X \in C_v} \sum_j \alpha_j w_{jx}((p_{jx} - z_j)^2 - (p_{jx} - v_j)^2).$$

Let us evaluate the first term

$$\sum_{X \in C_u} \sum_j \alpha_j w_{jx}((p_{jx} - z_j)^2 - (p_{jx} - u_j)^2) = \sum_j \alpha_j \sum_{X \in C_u} w_{jx}(z_j^2 - 2p_{jx} z_j + 2p_{jx} u_j - u_j^2) =$$

From the definition of the u_j we have the relation $\sum_{X \in C_u} w_{jx} p_{jx} = A_j u_j$ and therefore

$$= \sum_j \alpha_j (A_j z_j^2 - 2A_j u_j (z_j - u_j) - A_j u_j^2) = \sum_j \alpha_j A_j (z_j - u_j)^2.$$

…Proof of the generalized Ward's relation

From here, for the sum of both terms holds

$$D(C_u, C_v) = \sum_j \alpha_j \left(A_j (z_j - u_j)^2 + B_j (z_j - v_j)^2 \right).$$

Further computation of dissimilarity yields the expression

$$\begin{aligned} D(C_u, C_v) &= \sum_j \alpha_j \left(A_j (z_j - u_j)^2 + B_j (z_j - v_j)^2 \right) \\ &= \sum_j \alpha_j \left(A_(z_j^2 - 2u_j z_j + u_j^2) + B_j (z_j^2 - 2v_j z_I + v_j^2) \right) \\ &= \sum_j \alpha_j \left(z_j^2 (A_j + B_j) - 2z_j (A_j u_j + B_j v_j) + A_j u_j^2 + B_j v_j^2 \right). \end{aligned}$$

We can express the new cluster leader z_j also in a different manner. Starting from equation (**??**) and since $C_u \cap C_v = \emptyset$ we have

$$\sum_{X \in C_z} w_{jx} \cdot z_j = \sum_{X \in (C_u \cup C_v)} w_{jx} \cdot p_{jx} = \sum_{X \in C_u} w_{jx} \cdot p_{jx} + \sum_{X \in C_v} w_{jx} \cdot p_{jx}$$

…Proof of the generalized Ward's relation

Therefore

$$\begin{aligned} z_j &= \frac{\sum_{X \in C_u} w_{jx} \cdot p_{jx} + \sum_{X \in C_v} w_{jx} \cdot p_{jx}}{\sum_{X \in (C_u \cup C_v)} w_{jx}} \\ &= \frac{A_j u_j + B_j v_j}{A_j + B_j}. \end{aligned}$$

This relation is used in the calculation of $D(C_u, C_v)$:

$$\begin{aligned} D(C_u, C_v) &= \sum_j \alpha_j \left(A_j u_j^2 + B_j v_j^2 - (A_j + B_j) z_j^2 \right) \\ &= \sum_j \alpha_j \left(A_j u_j^2 + B_j v_j^2 - (A_j + B_j) (\frac{A_j u_j + B_j v_j}{A_j + B_j})^2 \right) \\ &= \sum_j \alpha_j \frac{A_j \cdot B_j}{A_j + B_j} (u_j - v_j)^2. \end{aligned}$$

Special cases

1. General case `gDist`: Frequency distributions of different number of units by the unit X for each variable V_j: $w_{jx} = n_{jx}$

2. Simple frequencies `fDist`: Frequency distributions of the same number of units for all variables $V_j, j = 1, \ldots, m$: $w_{jx} = n_x$. The formulae can be simplified.

$$l_i^* = \frac{\sum_{X \in C} n_x \cdot p_{jx}}{A}$$

where $A = \sum_{X \in C} n_x$ is independent of j.

$$D(C_u, C_v) = \frac{A \cdot B}{A + B} \sum_j \alpha_j (u_j - v_j)^2 = \frac{A \cdot B}{A + B} d_0(u, v),$$

In the case of only one variable, $d_0(u, v)$ denotes original Euclidean distance.

3. Probabilities `pDist`: Probability distributions $w_{jx} = 1$. Further simplifications are possible.

$$l_j^* = \frac{1}{|C|} \sum_{X \in C} p_{jx}$$

$$D(C_u, C_v) = \frac{|C_u| \cdot |C_v|}{|C_u| + |C_v|} \sum_j \alpha_j (u_j - v_j)^2 = \frac{|C_u| \cdot |C_v|}{|C_u| + |C_v|} d_0(u, v).$$

Sheme of analysis

raw data

$\Downarrow$

ENCODE

$\Downarrow$

unified data

$\Downarrow$

MAKE SOs

$\Downarrow$

SOs - lists of distributions

$\Downarrow$

`leaderSO`

$\Downarrow$

clustering and cluster leaders

$\Downarrow$

`hclustSO`

$\Downarrow$

hierarchy and cluster leaders

$\Downarrow$

ANALYSIS

$\Downarrow$

dendrogram, reports

Encoding of numerical variables

```
# make "encWater" encoding
makeEnc(water,"Water",10,file="temp.R")
source("temp.R")
unlink("temp.R") # tidy up

encWater <- list(
  "[0]"           = function(x) x<=0,
  "(0,5.65]"      = function(x) x<=5.65,
  "(5.65,29.5]"   = function(x) x<=29.5,
  "(29.5,53.9)"   = function(x) x< 53.9,
  "[53.9,62.4)"   = function(x) x< 62.4,
  "[62.4,70.75)"  = function(x) x< 70.75,
  "[70.75,78.05]" = function(x) x<=78.05,
  "(78.05,88)"    = function(x) x< 88,
  "[88,100]"      = function(x) x<=100,
  "NA"            = function(x) TRUE )

encodeSO <- function(x,encoding,codeNA){
  if(is.na(x)) return(codeNA)
  for(i in 1:length(encoding)) if(encoding[[i]](x)) return(i)
}
```

SR22

Encoding of numerical variables

```
> source("C:\\...\\sr14\\encFood.R")
> b <- read.delim("./sr14/abbrev.txt",dec=",",row.names=2)
> colnames(b)
 [1] "NDB_No"      "Water"       "Energ_Kcal"  "Protein"     "Tot_Lipid"
 [6] "Carbohydrt"  "Fiber_TD"    "Ash"         "Calcium"     "Phosphorus"
[11] "Iron"        "Sodium"      "Potassium"   "Magnesium"   "Zinc"
[16] "Copper"      "Manganese"   "Selenium"    "Vit_A"       "Vit_E"
[21] "Thiamin"     "Ribolflavin" "Niacin"      "Panto_Acid"  "Vit_B6"
[26] "Folate"      "Vit_B12"     "Vit_C"       "FA_Sat"      "FA_Mono"
[31] "FA_Poly"     "Cholestrl"   "GmWt_1"      "GmWt_Desc1"  "GmWt_2"
[36] "GmWt_Desc2"  "Refuse_Pct"
> water <- b[[2]]
> water[1:10]
 [1] 15.87 15.87  0.24 42.41 41.11 48.42 51.80 39.28 36.75 37.65
> names(encWater)
 [1] "[0]"           "(0,5.65]"      "(5.65,29.5]"   "(29.5,53.9)"
 [5] "[53.9,62.4)"   "[62.4,70.75)"  "[70.75,78.05]" "(78.05,88)"
 [9] "[88,100]"      "NA"
> wat <- sapply(water,function(x) encodeSO(x,encWater,10))
> wat[1:10]
 [1] 3 3 2 4 4 4 4 4 4 4
%> varCats <- vector("list",31)
%> varCats[[1]]  <- names(encWater)
%   ...
%> varCats[[31]] <- names(encCholestr)
%> names(varCats) <- colnames(b)[2:32]
%> foodSO <- vector("list",nVar)
%> foodSO[[1]] <- wat
% ...
%> foodSO[[31]] <- cho
```

Encoding of numerical variables

```
ncat <- 10
# produce bins
wat <- sapply(water,function(x) encodeSO(x,encWater,10))
toVector<- function(cat,ncat)
                {x<-numeric(ncat);x[cat]<-1;return(x)}

# produce matrix
m <- sapply(wat,function(x) toVector(x,ncat+1))
m <- as.data.frame(t(m))
names(m) <- names("encWater") # set names of variables
```

Encoding of ordinal or nominal variables

```
cf <- read.table("./cars/cars.csv",header=TRUE,
  dec=".",row.names=1)
levType <- c("LI","KL","EN","KA","KB","RO","TE","KU")
typ <- factor(cf$type,levels=levType)
carsSO <- vector("list",26)
names(carsSO) <- colnames(cf)
carsSO$type <- as.integer(typ)
carsCats <- vector("list",26)
names(carsCats) <- colnames(cf)
carsCats$type <- c(levType,NA)
```

Data and metadata

```
setwd("C:/..../clamix.R")
source("C:\\....\\clamix.R\\clamix.R")
source("C:\\....\\clamix.R\\sr14\\encFood.R")
nVar <- 31
varCats <- vector("list",nVar)
varCats[[1]]  <- names(encWater)
varCats[[2]]  <- names(encEnergKC)
varCats[[3]]  <- names(encProtein)
varCats[[4]]  <- names(encTotLipi)
varCats[[5]]  <- names(encCarbohyd)
.......
varCats[[28]] <- names(encFaSat)
varCats[[29]] <- names(encFaMono)
varCats[[30]] <- names(encFaPoly)
varCats[[31]] <- names(encCholestr)
nVarP <- nVar+1
nCats <- sapply(varCats,length)
so <- emptySO(nCats)
b <- read.delim("./sr14/abbrev.txt",dec=",",row.names=2)
numSO <- nrow(b)
names(varCats) <- colnames(b)[2:32]
long <- rownames(b)
namedSO <- so; names(namedSO) <- c(names(varCats),"num")
for(i in 1:nVar) names(namedSO[[i]]) <- varCats[[i]]
save(nVar,nVarP,so,namedSO,numSO,long,varCats,file="./sr14/sr14.meta")
SOs <- vector("list",numSO)
for(i in 1:numSO){
  st <- so
  for(j in 1:nVar) st[[j]][foodSO[i,j]] <- 1
  st$num <- 1
  names(SOs)[[i]] <- rownames(foodSO)[[i]]
  SOs[[i]] <- st
}
save(nVar,nVarP,so,numSO,SOs,file="./sr14/sr14.so")
```

Transformation — classes

A `list` with a number of `data.frames` that consist of one variable described with distributions each can be transformed into a `symData` object. A *`symData` object* is an R class for description of histogram symbolic data. The class is represented as a list of:

`SOs`	a vector of symbolic objects (of *class `symObject`*); the transformed data set
`so`	an empty `symObject`
`namedSO`	an empty `symObject` with names for categories of each variable
`alpha`	a vector of weights α_j
`type`	a type of `symData`.

A `symObject` is an R class for description of a histogram symbolic object (one unit). It is represented as a list of variables. Each variable is represented as a vector. The last two components of the vector correspond to the number of `NA`s and the size (`num`) of the variable items.

The transformation of the data set is done via function `create.symData`.

Transformation — example

```
data(popul06f)
data(popul06m)
datalist <- list("M"=popul06f,"F"=popul06m)
dataset <- create.symData(datalist,"fDist")
summary(dataset) # description of the class
```

```
summary symData :

Dimension (units x variables): 224 x 2
Type of distributions: fDist
Outlook of symObject:
symObject :
Number of variables: 2
M
  0-4   5-9 10-14 15-19 20-24 25-29 30-34 35-39 40-44 45-49 50-54 55-59 60-64 65-6
    0     0     0     0     0     0     0     0     0     0     0     0     0
70-74 75-79   80+    NA   num
    0     0     0     0     0
F
  0-4   5-9 10-14 15-19 20-24 25-29 30-34 35-39 40-44 45-49 50-54 55-59 60-64 65-6
    0     0     0     0     0     0     0     0     0     0     0     0     0
70-74 75-79   80+    NA   num
    0     0     0     0     0
```

distSO

```
# computes weighted squared Euclidean dissimilarity between SOs

distSO <- function(X,Y,nVar,alpha){
  d <- numeric(nVar)
  for(i in 1:nVar) {
    ln <- length(X[[i]]);
    nX <- as.double(X[[i]][[ln]]); nY <- as.double(Y[[i]][[ln]])
    d[[i]] <- nX*sum((X[[i]][-ln]/nX-Y[[i]][-ln]/nY)**2)
  }
  dis <- as.numeric(d %*% alpha)/2
  if (is.na(dis)) dis <- Inf
  return(dis)
}
```

leaderSO

```
leaderSO <- function(dataset,maxL){
  #attach(dataset)
  so <- dataset$so; alpha <- dataset$alpha
  SOs <- dataset$SOs; nVar <- length(so)
  numSO <- length(SOs)
  L <- vector("list",maxL); Ro <- numeric(maxL)
# random partition into maxL clusters
  clust <- sample(1:maxL,numSO,replace=TRUE)
  tim <- 1; step <- 0
  repeat {
    step <- step+1
  # new leaders - determine the leaders of clusters in current partition
    for(k in 1:maxL){L[[k]] <- so; names(L)[[k]] <- paste("L",k,sep="")}
    for(i in 1:numSO){j <- clust[[i]];
      for(k in 1:nVar) L[[j]][[k]] <- L[[j]][[k]] + SOs[[i]][[k]] }
  # new partition - assign each unit to the nearest new leader
    clust <- integer(numSO)
    R <- numeric(maxL); p <- double(maxL)
    for(i in 1:numSO){d <- double(maxL)
      for(k in 1:maxL){d[[k]] <- distSO(SOs[[i]],L[[k]],nVar,alpha)}
      r <- min(d); j <- which(d==r)
      if(length(j)==0){
        cat("unit",i,"\n",d,"\n"); flush.console(); print(SOs[[i]]); flush.console()
        u <- which(is.na(d))[[1]]; cat("leader",u,"\n"); print(L[[u]])
        stop()}
      j <- which(d==r)[[1]];
      clust[[i]] <- j
      p[[j]] <- p[[j]] + r; if(R[[j]]<r) R[[j]]<- r
    }
  # report
    cat("\nStep",step,"\n")
    print(table(clust)); print(R); print(Ro-R); Ro <- R;
    print(p); print(sum(p)); flush.console()
    tim <- tim-1
    if(tim<1){
      ans <- readline("Times repeat = ")
      tim <- as.integer(ans); if (tim < 1)  break
    }
  # in the case of empty cluster put in the most distant SO
    repeat{
      t <- table(clust); em <- setdiff(1:maxL,as.integer(names(t)))
      if(length(em)==0) break
      j <- em[[1]]; rmax <- 0; imax <- 0
      for(i in 1:numSO){d <- double(maxL)
        for(k in 1:maxL){d[[k]] <- distSO(SOs[[i]],L[[k]],nVar,alpha)}
        r <- max(d);  if(rmax<r) {rmax <- r; imax <- i}
      }
      clust[[imax]] <- j; L[[j]] <- SOs[[imax]]
      cat("*** empty cluster",j,"- SO",imax,"transfered, rmax =",rmax,"\n")
      flush.console()
    }
  }
  #detach(dataset)
  leaders <- dataset
  leaders$SOs <- L
  return (list(clust=clust,leaders_symData=leaders,R=R,p=p))
}
```

hclustSO

```
hclustSO <- function(dataset){
# compute order of dendrogram
  orDendro <- function(i){if(i<0) return(-i)
    return(c(orDendro(m[i,1]),orDendro(m[i,2])))}
  #attach(dataset)
  so <- dataset$so; alpha <- dataset$alpha
  L <- dataset$SOs; nVar <- length(so)
  numL <- length(L); numLm <- numL-1
# each unit is a cluster; compute inter-cluster dissimilarity matrix
  D <- matrix(nrow=numL,ncol=numL); diag(D) <- Inf
  for(i in 1:numLm) for(j in (i+1):numL) {
      D[i,j] <- distCl(L[[i]],L[[j]],nVar,alpha); D[j,i] <- D[i,j]
  }
  active <- 1:numL; m <- matrix(nrow=numLm,ncol=2)
  node <- rep(0,numL); h <- numeric(numLm); LL <- vector("list",numLm)
  for(k in 1:numLm) LL[[k]] <- so
  for(k in 1:numLm){
  # determine the closest pair of clusters (p,q)
    ind <- active[sapply(active,function(i) which.min(D[i,active]))]
    dd <- sapply(active,function(i) min(D[i,active]))
    pq <- which.min(dd)
  # join the closest pair of clusters
    p<-active[pq]; q <- ind[pq]; h[k] <- D[p,q]
    if(node[p]==0){m[k,1] <- -p; Lp <- L[[p]]
      } else {m[k,1] <- node[p]; Lp <- LL[[node[p]]]}
    if(node[q]==0){m[k,2] <- -q; Lq <- L[[q]]
      } else {m[k,2] <- node[q]; Lq <- LL[[node[q]]]}
    for(t in 1:nVar) LL[[k]][[t]] <- Lp[[t]] + Lq[[t]]
    active <- setdiff(active,p)
    Lpq <- LL[[k]]
  # determine dissimilarities to the new cluster
    for(s in setdiff(active,q)){
      if(node[s]==0){Ls <- L[[s]]} else {Ls <- LL[[node[s]]]}
      D[q,s] <- distCl(Lpq,Ls,nVar,alpha); D[s,q] <- D[q,s]
    }
    node[[q]] <- k
  }
  hc <- list(merge=m,height=h,order=orDendro(numLm),labels=names(L),
    method="adapted ward",call=match.call(),dist.method="squared euclidean",leaders=LL)
  class(hc) <- "hclust"
  #detach(dataset)
  return(hc)
}
```

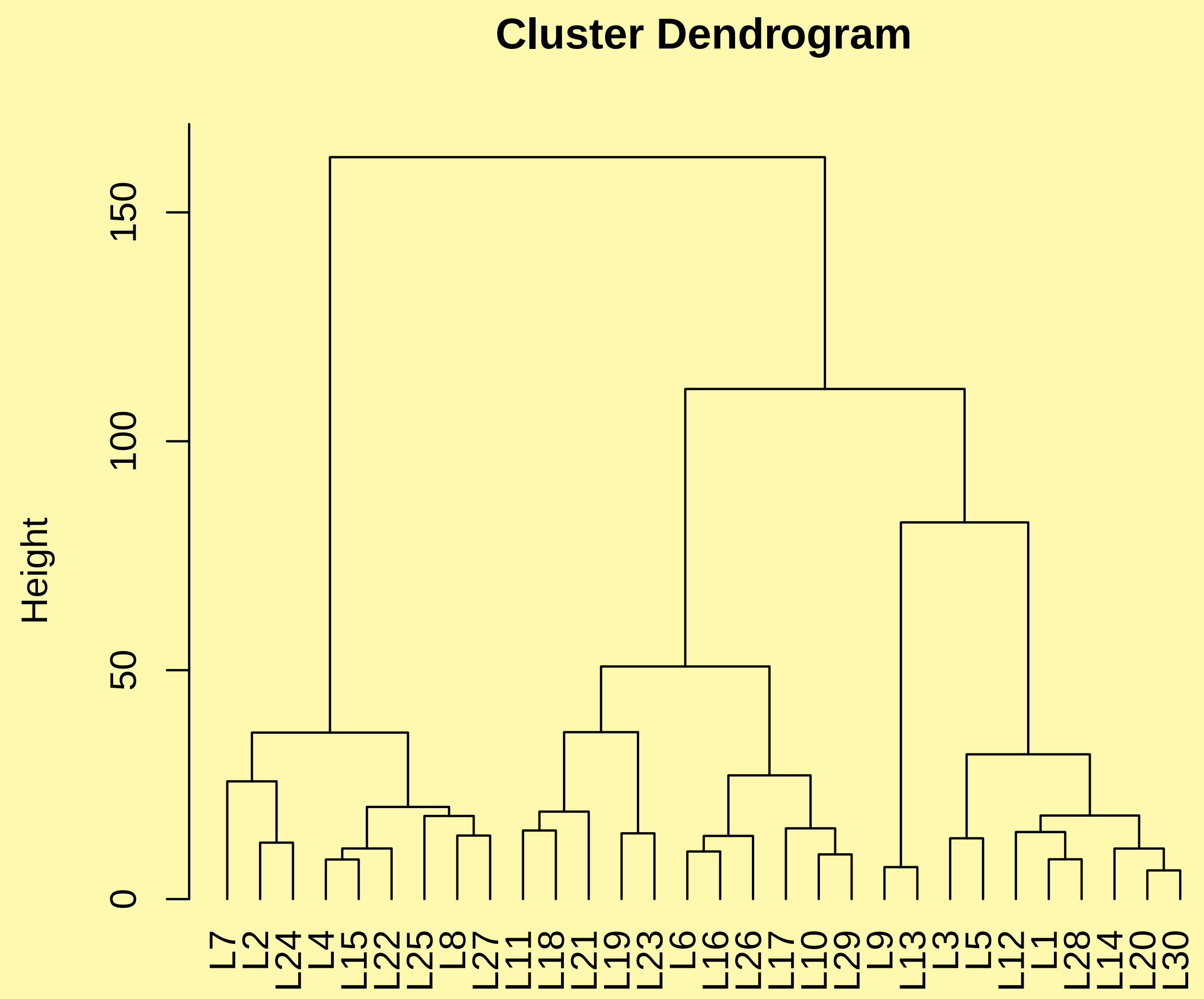
Application

```
> setwd("C:/.../clamix/clamix.R")
> source("C:\\...\\clamix.R\\Clamix.R")
> load("./sr14/sr14.so")
> objects()
[1] "dat"      "distSO"   "emptySO"  "encodeOS" "numSO"    "nVar"     "nVarP"
[8] "so"       "SOs"
> w <- rep(1/nVar,nVar)
> rez <- leaderSO(SOs,30,w,so)

Step 1
clust
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27
132 244 232 263 249 589 752 193 273  57 220 235 139  80  81 530  32 201 357  36 185  74 124 205  99  19  87
 28  29  30
178  25 148
 [1] 0.4466406 0.4443925 0.4568250 0.4571872 0.4660656 0.4721637 0.4488782 0.4399954 0.4803863 0.4419636
........
[21]  77.52889  31.03541  50.96077  80.89709  41.74524   8.06992  35.82662  73.81336  10.05692  59.96114
[1] 2483.908
Times repeat = 3
........

Step 42
clust
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27
116 179 367 215 299 186 275 122 324 255 179 165 165 168 206 276 221  57 308 105  73 161 290 126 230 133 360
 28  29  30
200 185  93
 [1] 0.3961897 0.4226399 0.4669776 0.4373342 0.4699742 0.4520931 0.4155581 0.3473073 0.4438606 0.4212916
[11] 0.4576676 0.4220990 0.3943517 0.4682425 0.4087065 0.4355381 0.4578319 0.3822814 0.4319199 0.4301251
[21] 0.2938638 0.4417081 0.4029236 0.2948455 0.4391676 0.4188643 0.4375495 0.4436210 0.4508061 0.4371343
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [1]  27.622636  37.895837 138.264393  64.037659 106.219657  61.910683  56.140880  28.080116  82.995619
[10]  74.473877  47.658137  44.289150  41.319062  53.270737  67.129032  95.597943  71.012699   6.274477
[19]  79.517072  27.804608   5.580203  55.304348  69.345717  19.494624  77.636606  37.387824 112.933781
[28]  59.985484  59.407323  26.572320
[1] 1735.163
Times repeat = 0
> save(rez,file="sr14.rez")
> hc <- hclustSO(rez$leaders,w)
> plot(hc, hang = -1)
```

Dendrogram


Cluster Dendrogram
Height
0
50
100
150
L7
L2
L24
L4
L15
L22
L25
L8
L27
L11
L18
L21
L19
L23
L6
L16
L26
L17
L10
L29
L9
L13
L3
L5
L12
L1
L28
L14
L20
L30

Analysis of results

```
> long[rez$clust==7]
   [1] "LUNCHEON MEAT,BF,THIN SLICED"
   [2] "BEEF,COMP OF RTL CUTS,LN,1/4\"FAT,ALL GRDS,CKD"
   [3] "BEEF,COMP OF RTL CUTS,LN,1/4\"FAT,CHOIC,CKD"
   [4] "BEEF,COMP OF RTL CUTS,LN,1/4\"FAT,SEL,CKD"
   [5] "BEEF,COMP OF RTL CUTS,LN,1/4\"FAT,PRIME,CKD"
   [6] "BEEF,BRISKET,WHL,LN,1/4\"FAT,ALL GRDS,CKD,BRSD"
   [7] "BEEF,BRISKET,FLAT HALF,LN,1/4\"FAT,ALL GRDS,CKD,BRSD"
   [8] "BEEF,BRISKET,POINT HALF,LN,1/4\"FAT,ALL GRDS,CKD,BRSD"
   [9] "BEEF,CHUCK,ARM POT RST,LN,1/4\"FAT,ALL GRDS,CKD,BRSD"
  [10] "BEEF,CHUCK,ARM POT RST,LN,1/4\"FAT,CHOIC,CKD,BRSD"
  [11] "BEEF,CHUCK,ARM POT RST,LN,1/4\"FAT,SEL,CKD,BRSD"
  [12] "BEEF,CHUCK,BLADE RST,LN,1/4\"FAT,ALL GRDS,CKD,BRSD"
  [13] "BEEF,CHUCK,BLADE RST,LN,1/4\"FAT,CHOIC,CKD,BRSD"
  [14] "BEEF,CHUCK,BLADE RST,LN,1/4\"FAT,SEL,CKD,BRSD"
  [15] "BEEF,FLANK,LN&FAT,0\"FAT,CHOIC,CKD,BRSD"
  [16] "BEEF,FLANK,LN&FAT,0\"FAT,CHOIC,CKD,BRLD"
  [17] "BEEF,FLANK,LN,0\"FAT,CHOIC,CKD,BRSD"
  [18] "BEEF,FLANK,LN,0\"FAT,CHOIC,CKD,BRLD"
  [19] "BEEF,RIB,WHL (RIBS 6-12),LN,1/4\"FAT,ALL GRDS,CKD,BRLD"
   ....
  [43] "BEEF,RIB,SML END (RIBS 10-12),LN,1/4\"FAT,PRIME,CKD,RSTD"
  [44] "BEEF,RIB,SHORTRIBS,LN,CHOIC,CKD,BRSD"
  [45] "BEEF,RND,FULL CUT,LN&FAT,1/4\"FAT,CHOIC,CKD,BRLD"
  [46] "BEEF,RND,FULL CUT,LN&FAT,1/4\"FAT,SEL,CKD,BRLD"
   ...........
  [79] "BEEF,SHANK CROSSCUTS,LN,1/4\"FAT,CHOIC,CKD,SIMMRD"
  [80] "BEEF,SHRT LOIN,PRTRHS STEAK,LN,1/4\"FAT,CHOIC,CKD,BRLD"
  [81] "BEEF,SHRT LOIN,T-BONE STEAK,LN,1/4\"FAT,CHOIC,CKD,BRLD"
  [82] "BEEF,TENDERLOIN,LN&FAT,1/4\"FAT,CHOIC,CKD,RSTD"
  [83] "BEEF,TENDERLOIN,LN&FAT,1/4\"FAT,SEL,CKD,BRLD"
   .....
 [102] "BEEF,TOP SIRLOIN,LN,1/4\"FAT,CHOIC,CKD,BRLD"
 [103] "BEEF,TOP SIRLOIN,LN,1/4\"FAT,CHOIC,CKD,PAN-FRIED"
 [104] "BEEF,TOP SIRLOIN,LN,1/4\"FAT,SEL,CKD,BRLD"
 [105] "BEEF,GROUND,EX LN,CKD,BKD,WELL DONE"
 [106] "BEEF,GROUND,EX LN,CKD,BRLD,MED"
 [107] "BEEF,GROUND,EX LN,CKD,BRLD,WELL DONE"
   .........
 [270] "BEEF,TOP SIRLOIN,LN&FAT,1/8\"FAT,CHOIC,CKD,BRLD"
 [271] "BEEF,TOP SIRLOIN,LN&FAT,1/8\"FAT,CHOIC,CKD,PAN-FRIED"
 [272] "BEEF,TOP SIRLOIN,LN&FAT,1/8\"FAT,SEL,CKD,BRLD"
 [273] "LAMB,DOM,COMP OF RTL CUTS,LN,1/4\"FAT,CHOIC,CKD"
 [274] "BEEF,SHRT LOIN,T-BONE STEAK,LN&FAT,1/8\"FAT,ALL GRDS,CKD,BRLD"
 [275] "BEEF,SHRT LOIN,T-BONE STEAK,LN&FAT,1/8\"FAT,SEL,CKD,BRLD"
```

Analysis of results

```
total <- namedSO
for(i in 1:numL) for(j in 1:nVarP) total[[j]] <- total[[j]] + L[[i]][[j]]

testSOvarP <- function(SO,var,total,a){
  pt <- total[[var]]/total$num
  pj <- SO[[var]]/SO$num
  te <- qnorm(1-a/2)*sqrt(pt*(1-pt)/SO$num)
  dif <- pj - pt; test <- abs(dif) > te
  for(k in 1:length(pt)) if(test[[k]]) {q <- pj[[k]]/pt[[k]]
    if(q > 1) cat(format(names(test)[[k]],width=15,justify="right"),
      format(c(pj[[k]],pt[[k]],q),
      width=12,justify="left",digits=5,nsmall=7),"\n")
  }
}
```

Analysis of results

```
> for(v in 1:nVar) {
+  cat("\n",v,". ",names(total)[[v]],"\n",sep="")
+  testSOvarP(L[[7]],v,total,0.001)}

1. Water
   [53.9,62.4)     0.7563636     0.1240272     6.0983712
2. Energ_Kcal
      (160,232]     0.5709091     0.1425733     4.0043206
      (232,312]     0.4036364     0.1400894     2.8812766
3. Protein
    [23.05,37]     1.0000000     0.1599603     6.2515528
4. Tot_Lipid
        [3.8,8)     0.2109091     0.1390959     1.5162857
       [8,13.7)     0.4654545     0.1394271     3.3383373
   [13.7,23.5)     0.3054545     0.1392615     2.1933888
5. Carbohydrt
            [0]     0.9890909     0.2702434     3.6600000
6. Fiber_TD
            [0]     1.0000000     0.3957609     2.5267782
7. Ash
       [1,1.24)     0.5600000     0.1697301     3.2993561
   [1.24,1.75)     0.3418182     0.1662527     2.0560159
8. Calcium
          (0,7]     0.5381818     0.1563173     3.4428814
         (7,12]     0.3854545     0.1592979     2.4197089
9. Phosphorus
      [208,268)     0.8181818     0.1318099     6.2072864
  .........
30. FA_Poly
  (0.115,0.335]     0.3309091     0.1563173     2.1169068
  (0.335,0.685]     0.5854545     0.1549925     3.7773077
31. Cholestrl
        (66,80]     0.3018182     0.0816360     3.6971197
        (80,94]     0.4363636     0.0864382     5.0482759
       (94,550]     0.1818182     0.0796489     2.2827443
```

Conclusions

It works.

`https://r-forge.r-project.org/projects/clamix/`

Still to do:

- functions to automatically create `data.frame` from encoded variable
- additional methods to classes `symData` and `symObject` ?
- additional options in `leaderSO`: limit on the radius of clusters, minimal number of units in cluster, silent mode
- additional support for analyses of results

References

Andreev L., Andreev M. (2004). Analysis of Population Pyramids by a New Method for Intelligent Pattern Recognition. *Matrixreasonong*, Equicom, Inc..

Batagelj, V (1988). *Generalized Ward and Related Clustering Problems*. Classification and Related Methods of Data Analysis. H.H. Bock (editor). North-Holland, Amsterdam, 1988. p. 67-74.
http://vlado.fmf.uni-lj.si/vlado/papers/Ward.pdf

Billard L., Diday E. (2006). *Symbolic data analysis. Conceptual statistics and data mining*. Wiley.

Bock H.H., Diday E. (2000). ”Symbolic Objects”, In *Analysis of Symbolic Data. Exploratory methods for extracting statistical information from complex data*. Springer, Heidelberg.

Diday E., Noirhomme-Fraiture M. (2008). ”Symbolic Data Analysis and the SODAS Software”, JohnWiley & Sons, Ltd..

Japelj Pavešić B.,Korenjak-Černe S. (2004). ”Differences in teaching and learning mathematics in classes over the world : the application of adapted leaders clustering method”, In

Proceedings of the IRC - 2004 : IEA International Research Conference. Nicosia: University of Cyprus, Department of Education, pp. 85–101.

Kaufman L., Rousseeuw P.J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York.

Korenjak-Černe S., Batagelj V. (2002). Symbolic Data Analysis Approach to Clustering Large Datasets. Jajuga etal. eds.: Classification, Clustering and Data Analysis. IFCS'02, Cracow, 2002. Springer, Berlin, p. 319-327.

Korenjak-Černe S., Batagelj V. (1998). Clustering Large Datasets of Mixed Units. IFCS'98, Rome.

Korenjak-Černe S., Kejžar N., Batagelj V. (2008). Clustering of population pyramids. *Informatica*, Jun. 2008, 32 (2), 157-167.

IDB: International Data Base. http://www.census.gov/ipc/www/idbnew.html.

R Development Core Team(2008) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org.