Optimal Partitioning and Clustering: How Many Classes ?

Seminar, 12 May 2010

ABI, University Paris 6

Michel Petitjean

MTi, UMR-S 973, INSERM, University Paris 7

http://petitjeanmichel.free.fr/itoweb.petitjean.class.html

PARTITION

A partition of a set E is a collection of disjoint subsets such that the union of these subsets is E.

Each subset is called a class.

The relation "being in the same class" is an equivalence relation (reflexive, symmetric, transitive).

Finite case, n elements: the number of partitions $B(n)$ increases exponentially with n

k classes: $P(n,k) = P(n-1,k-1) + k \cdot P(n-1,k)$

$P(n,k) \geq C(n,k)$ ($C(n,k)$ is the binomial coefficient)

Bell number: $B(n) = P(n,1) + P(n,2) + ... + P(n,n)$

PARTITION MATRIX

A partition matrix Y is a symmetric square matrix of binary elements, of order n, such that
$Y_{ij} = 1$ when i and j fall in the same class, and $Y_{ij} = 0$ elsewhere.

Y is block diagonal up to a simultaneous renumbering of its lines and columns.

The n sums of the lines or of the columns are the sizes of the classes (the size k of a class appears k times)

EXAMPLES OF PARTITION MATRICES

$$Y = \left(\begin{array}{cccc|ccc|ccc} 1&1&1&1&0&0&0&0&0&0\\ 1&1&1&1&0&0&0&0&0&0\\ 1&1&1&1&0&0&0&0&0&0\\ 1&1&1&1&0&0&0&0&0&0\\ \hline 0&0&0&0&1&1&1&0&0&0\\ 0&0&0&0&1&1&1&0&0&0\\ 0&0&0&0&1&1&1&0&0&0\\ \hline 0&0&0&0&0&0&0&1&1&1\\ 0&0&0&0&0&0&0&1&1&1\\ 0&0&0&0&0&0&0&1&1&1 \end{array}\right)$$

$$Y = \left(\begin{array}{cc|cc|ccc} 1&1&0&0&0&0&0\\ 1&1&0&0&0&0&0\\ \hline 0&0&1&1&0&0&0\\ 0&0&1&1&0&0&0\\ \hline 0&0&0&0&1&1&1\\ 0&0&0&0&1&1&1\\ 0&0&0&0&1&1&1 \end{array}\right)$$

$$Y = \left(\begin{array}{c|ccc|ccc|cccc} 1&0&0&0&0&0&0&0&0&0&0\\ \hline 0&1&1&1&1&0&0&0&0&0&0&0\\ 0&1&1&1&1&0&0&0&0&0&0&0\\ 0&1&1&1&1&0&0&0&0&0&0&0\\ 0&1&1&1&1&0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&1&1&1&0&0&0&0\\ 0&0&0&0&0&1&1&1&0&0&0&0\\ 0&0&0&0&0&1&1&1&0&0&0&0\\ \hline 0&0&0&0&0&0&0&0&1&1&1&1\\ 0&0&0&0&0&0&0&0&1&1&1&1\\ 0&0&0&0&0&0&0&0&1&1&1&1\\ 0&0&0&0&0&0&0&0&1&1&1&1 \end{array}\right)$$

CLUSTERING ALGORITHMS

Clustering algorithms are expected to act as partitioning algorithms:
having an input set of n individuals, generate a partition of the set, optimal in some sense.

Many of these algorithms assume the prior knowledge of the number of classes !

(1) What number of classes should select the user ?

Hierarchical algorithms considers embedded partitions, from 1 to n classes (descending algorithms), or from n to 1 (ascending algorithms).

(2) Where to cut the resulting dendrogram ?

Many algorithms answer to (1) and (2) via the use of ***external*** parameters: these latter are critical values set by the programmer or/and by the user, and which are ***NOT*** part of the input data. Changing these parameters may induce dramatic changes on the resulting partition.

WE PRESENT TWO METHODS COMPUTING THE OPTIMAL NUMBER OF CLASSES WITHOUT *EXTERNAL* PARAMETERS

1. Similarity aggregation method (Marcotorchino, 1981)

Input: n individuals described by p qualitative variables,
or: a symmetric square array of signed dissimilarities.

2. Clustering gain (Jung et al., 2003; Meslamani et al., 2009)

Input: n individuals described by p numerical variables,
or: a symmetric square array of non negative dissimilarities.

QUALITATIVES VARIABLES

We generate the indicator matrix X (disjunctive coding):
to each category of each qualitative variable is associated a column of X:
the indicator variable of the category.

Example: ten individuals, three variables ($n = 10$, $p = 3$): (R,G,B), (+,-), (yes,no)

$$\left(\begin{array}{c|c|c} R & - & yes \\ G & - & no \\ G & - & yes \\ B & + & no \\ R & - & yes \\ B & + & no \\ R & - & yes \\ G & - & yes \\ R & - & yes \\ G & - & no \end{array}\right) \qquad X = \left(\begin{array}{ccc|cc|cc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{array}\right)$$

$X' \cdot X$ is the Burt table, used as input of the MCA (Multiple Correspondence Analysis)

$C = X \cdot X'$ is the sum of the p partition matrices associated to the variables.
(C/p is the mean of the partition matrices, but it is not itself a partition, in general)

C_{ij} is the number of times that the individuals i and j are classified together
(number of agreements between the p partitions).

$$X = \left(\begin{array}{ccc|cc|cc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{array}\right) \qquad C = \left(\begin{array}{cccccccccc} 3 & 1 & 2 & 0 & 3 & 0 & 3 & 2 & 3 & 1 \\ 1 & 3 & 2 & 1 & 1 & 1 & 1 & 2 & 1 & 3 \\ 2 & 2 & 3 & 0 & 2 & 0 & 2 & 3 & 2 & 2 \\ 0 & 1 & 0 & 3 & 0 & 3 & 0 & 0 & 0 & 1 \\ 3 & 1 & 2 & 0 & 3 & 0 & 3 & 2 & 3 & 1 \\ 0 & 1 & 0 & 3 & 0 & 3 & 0 & 0 & 0 & 1 \\ 3 & 1 & 2 & 0 & 3 & 0 & 3 & 2 & 3 & 1 \\ 2 & 2 & 3 & 0 & 2 & 0 & 2 & 3 & 2 & 2 \\ 3 & 1 & 2 & 0 & 3 & 0 & 3 & 2 & 3 & 1 \\ 1 & 3 & 2 & 1 & 1 & 1 & 1 & 2 & 1 & 3 \end{array}\right)$$

For each couple of individuals i,j,
the number of agreements C_{ij} plus the number of disagreements is equal to p.

Thus, for each couple i,j of individuals, $\quad S_{ij} = [2 \cdot C_{ij} - p] \quad$ is equal to the difference:
[number of agreements between i and j] $\;$ *minus* $\;$ [number of disagreements between i and j].

$$S = \begin{pmatrix} 3 & -1 & 1 & -3 & 3 & -3 & 3 & 1 & 3 & -1 \\ -1 & 3 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & 3 \\ 1 & 1 & 3 & -3 & 1 & -3 & 1 & 3 & 1 & 1 \\ -3 & -1 & -3 & 3 & -3 & 3 & -3 & -3 & -3 & -1 \\ 3 & -1 & 1 & -3 & 3 & -3 & 3 & 1 & 3 & -1 \\ -3 & -1 & -3 & 3 & -3 & 3 & -3 & -3 & -3 & -1 \\ 3 & -1 & 1 & -3 & 3 & -3 & 3 & 1 & 3 & -1 \\ 1 & 1 & 3 & -3 & 1 & -3 & 1 & 3 & 1 & 1 \\ 3 & -1 & 1 & -3 & 3 & -3 & 3 & 1 & 3 & -1 \\ -1 & 3 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & 3 \end{pmatrix}$$

Optimal partition: put together i and j
when there is a majority of variables judging them together.

We look for the partition Y maximizing the number of agreements

or, equivalently,

we look for the partition Y maximizing the difference between the number of agreements and the the number of disagreements.

Intuitive solution: put **"1"** where $S_{ij} > 0$ and put **"0"** where $S_{ij} < 0$.

Alas, in general, fails to satisfy the transitivity rule:

(i_1, i_2) together AND (i_2, i_3) together, but (i_1, i_3) NOT together !

CRITERIA OF MARCOTORCHINO (1981)

Solve: $\quad Max_{\{Y\}} \ [\ \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} Y_{ij} S_{ij} \ ]$

The above criteria is equivalent to minimize the Schur norm of the difference between the unknown partition Y and the mean partition C/p

$$p \cdot Min \| Y - C/p \|_s^2 = p \cdot Min \sum \sum (Y_{ij} - C_{ij}/p)^2$$

And because $Y_{ij}^2 = Y_{ij}$, it is equivalent to find the following minimum:

$$Min \sum \sum Y_{ij}(p - 2C_{ij}) = Min(- \sum \sum Y_{ij} S_{ij}) = Max(\sum \sum Y_{ij} S_{ij})$$

WEIGHTED VARIABLES (Petitjean, 2002)

We associate to the p variables the weights w_1, ..., w_p, with $W = w_1 + ... + w_p$.
The respective partition matrices are $Y^{(1)}$, ..., $Y^{(p)}$

Each agreement associated to the variable k is counted $+w_k$ rather than $+1$, and each disagreement associated to the variable k is counted $-w_k$ rather than -1.

$C = \sum_{k=1}^{k=p} w_k Y^{(k)}$ (mean partition matrix: $\bar{C} = C/W$, although $\bar{C}$ is still not a partition matrix)

$S_{ij} = \sum_{k=1}^{k=p} w_k (2Y_{ij}^{(k)} - 1) = 2C_{ij} - W$ [*The minimal Schur norm criteria is still satisfied*]

Coherence: when several categorical variables have the same partition matrix, they can be replaced by a single variable with a weight equal to the sum of the weights of the original variables.

GENERALIZATION

The criteria of Marcotorchino is applicable to any array of signed dissimilarities, discarding how it is generated.

NUMERICAL ASPECTS

Solve: $\quad Max_{\{Y\}} \ [\ \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} Y_{ij} \cdot S_{ij} \ ]$

Y and S are symmetric, and we do not care about the diagonal

$n(n-1)/2$ unknowns AND $n(n-1)(n-2)/2$ transitivity constraints !

The solution Y^* may be not unique.

The transitivity constraints are linear: ($Y_{ij} + Y_{ij} - Y_{ij} \leq 1$) and ($Y_{ij} = 0$ or $Y_{ij} = 1$)

We are left with the maximization of a linear function under linear constraints:
the problem can be solved by linear programming methods (Marcotorchino, 1981).

Time consuming AND space consuming !

(and usually not devoted to the ENUMERATION of all solutions).

SOLUTION BY DYNAMIC PROGRAMMING

A specialized version of the boolean algorithm of Faure and Malgrange is shown to require a spacework of only three matrices of order n (Petitjean, 2002).

Time consuming but NOT space consuming !

Enumerating all solutions is in general only slightly more consuming that finding one.

Approximate solution (fast):
perform an ascending hierarchical clustering as long as the dissimilarities are non-negative.

It is observed that the approximate solution is often optimal.

Open source (f77) and binaries: http://petitjeanmichel.free.fr/itoweb.petitjean.freeware.html

Available under **R**: http://cran.cict.fr/ (click on packages, then on amap, then on amap.pdf)

EXAMPLE

$$S = \begin{pmatrix} 3 & -1 & 1 & -3 & 3 & -3 & 3 & 1 & 3 & -1 \\ -1 & 3 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & 3 \\ 1 & 1 & 3 & -3 & 1 & -3 & 1 & 3 & 1 & 1 \\ -3 & -1 & -3 & 3 & -3 & 3 & -3 & -3 & -3 & -1 \\ 3 & -1 & 1 & -3 & 3 & -3 & 3 & 1 & 3 & -1 \\ -3 & -1 & -3 & 3 & -3 & 3 & -3 & -3 & -3 & -1 \\ 3 & -1 & 1 & -3 & 3 & -3 & 3 & 1 & 3 & -1 \\ 1 & 1 & 3 & -3 & 1 & -3 & 1 & 3 & 1 & 1 \\ 3 & -1 & 1 & -3 & 3 & -3 & 3 & 1 & 3 & -1 \\ -1 & 3 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & 3 \end{pmatrix} \qquad Y^* = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Only one optimal partition was found: $\{1, 3, 5, 7, 8, 9\}$, $\{2, 10\}$, $\{4, 6\}$

Cost of the optimal partition: 35 (upper bound of the cost: 39)

The initial partition found by hierachical clustering was the optimal one.

The branch and bound algorithm performed 131 forward moves and 131 backward moves
(variables initially sorted according algebraically decreasing contributions to the cost)

CATEGORICAL (QUALITATIVE) VARIABLES vs. NUMERICAL (QUANTITATIVE) VARIABLES

The method of Marcotorchino has nice properties.

But, in the real world, we more frequently encounter *numerical* variables rather than *categorical* variables and *non negative* dissimilarities (e.g. distances) rather than *signed* dissimilarities.

Converting data is always possible, but introduces arbitrariness !

We need a method working directly with numerical variables, or, more generally, working with non-negative dissimilarities.

Furthermore, we would like to define mean points in clusters, even if we heive neither Euclidean distances nor coordinates.

NUMERICAL VARIABLES IN E^p . NOTATIONS

n individuals. x_1, ..., x_n such that $g = \sum_{i=1}^{i=n} x_i/n = 0$

Cluster k ($k = 1, ..., K$) has n_k individuals: $x_1(k)$, ..., $x_{n_k}(k)$ with barycenter $g_k = \sum_{i=1}^{i=n_k} x_i(k)/n_k$

$$X = \begin{pmatrix} x_1' \\ \dots \\ x_n' \end{pmatrix} \qquad X_k = \begin{pmatrix} x_1'(k) \\ \dots \\ x_{n_k}'(k) \end{pmatrix}$$

Contribution of cluster k to the total inertia matrix T: $T_k = X_k' X_k$ ($T = X'X = \sum_{k=1}^{k=K} T_k$)

Inertia matrix of cluster k: $W_k = \sum_{i=1}^{i=n_k} (x_i(k) - g_k)(x_i(k) - g_k)'$

Within classes inertia matrix: $W = \sum_{k=1}^{k=K} W_k$

Contribution of cluster k to the between classes inertia matrix: $B_k = n_k g_k g_k'$

Between classes inertia matrix: $B = \sum_{k=1}^{k=K} B_k$

NUMERICAL VARIABLES IN E^p : RESULTS

$T_k = W_k + B_k \qquad T = W + B \qquad Tr(T_k) = Tr(W_k) + Tr(B_k) \qquad Tr(T) = Tr(W) + Tr(B)$

The traces are expressable from the squared Euclidean distances,
and the third equality above is an instance of the Huyghens theorem.

$$Tr(T_k) = \sum_{i=1}^{i=n_k} d^2(x_i(k), g) \qquad Tr(W_k) = \sum_{i=1}^{i=n_k} d^2(x_i(k), g_k) \qquad Tr(B_k) = n_k d^2(g_k, g)$$

Then, the following quantities are expressed only from the distances between individuals:

$$Tr(W_k) = [\sum_{i_1=1}^{i_1=n_k} \sum_{i_2=1}^{i_2=n_k} d^2(x_{i_1}(k), x_{i_2}(k))]/2n_k \qquad Tr(W) = \sum_{k=1}^{k=K} Tr(W_k)$$

$$Tr(T) = [\sum_{i_1=1}^{i_1=n} \sum_{i_2=1}^{i_2=n} d^2(x_{i_1}, x_{i_2})]/2n \qquad Tr(B) = Tr(T) - Tr(W)$$

Advantage: neither need of coordinates nor need of mean points.
Drawback: computations involve double summations.

CLUSTERING GAIN (Jung et al., 2003)

The clustering gain is computable as follows: $CG(K) = Tr(B) - Tr(\tilde{B})$

where $Tr(\tilde{B}) = \sum_{k=1}^{k=K} d^2(g_k, g)$ is the between classes inertia defined by the non weighted clusters, i.e. as if each cluster of size n_k $(k = 1, ..., K)$, had a weight equal to 1 rather than to n_k.

Thus: $CG(K) = \sum_{k=1}^{k=K} (n_k - 1) d^2(g_k, g)$ $\quad CG(K) \geq 0 \quad CG(1) = 0 \quad CG(n) = 0$

CG is a function of K which is ensured to reach a maximum.

The value of K for which this maximum is reached is the desired optimal number of classes.

Compare to Ward's method, based of the loss of inter clusters inertia:
grouping clusters k_1 and k_2 always generates a decrease of B, equal to $d^2(g_{k_1}, g_{k_2})[n_{k_1} n_{k_2}/(n_{k_1}+n_{k_2})]$
so that the maximum of B is always reached for $K = n$.

Remark 1: *The maximum of CG may be not unique.*

Remark 2: *Distances to mean points are required.*

COMPUTING MEAN POINTS FROM DISTANCES

We calculate: $Z^* = Min_{\{y \in \mathbb{R}^p\}} Z, \quad Z = \sum_{i=1}^{i=n} d^2(y, x_i) = \sum_{i=1}^{i=n} (x_i - y)'(x_i - y)$

$grad(Z) = -2 \sum_{i=1}^{i=n} x_i + 2 \sum_{i=1}^{i=n} y = 0 \quad$ thus, $\quad y = g = \sum_{i=1}^{i=n} x_i / n.$

The hessian $H = 2nI$ is positive definite, so that we rediscover that the minimum of the inertia is unique and is reached when y is the barycenter g of the set
(can be generalized to continuous distributions of finite inertia: Huyghens theorem).

Now we consider: $\hat{Z} = Min_{\{y \in \{x_1, ..., x_n\}\}} Z \quad$ (in other words, we minimize over a subset of $\mathbb{R}^p$)

This minimum $\hat{Z}$ is reached at least when $y = x_j, \quad j \in \{1, ..., n\}, \quad$ with $\quad \hat{Z} \geq Z^*$

The point x_j appears as the best approximation of the barycenter among the set $\{x_1, ..., x_n\}$.
It may be not unique, and the equality occurs if and only if g falls in the set $\{x_1, ..., x_n\}$.

The error is $\quad d^2(x_j, g) = [\sum_{i=1}^{i=n} d^2(x_j, x_i) - Tr(T)]/n \quad$ and the relative error is $\quad d^2(x_j, g)/[Tr(T)/n]$

where $Tr(T)$ is the inertia of the set (itself computable from the distances),
and $[Tr(T)/n]$ is the trace of the variance matrix of the set.

MODIFIED CLUSTERING GAIN (Meslamani et al., 2009)

The original clustering gain was defined as $CG(K) = \sum_{k=1}^{k=K} (n_k - 1) d^2(g_k, g)$

The modified clustering gain is defined as $MCG(K) = \sum_{k=1}^{k=K} (n_k - 1) d^2(\hat{g_k}, \hat{g})$

where $\hat{g_k}$ and $\hat{g}$ are the respective best approximations of g_k and g,
computed for each cluster k and for the whole set of n points.

$MCG(K)$ is computable from an array of distances.

We can compute it from an array of non-Euclidean distances, or from an array of non negative dissimilarities

Thus we have a defined a general stop criterion for hierarchical clustering

Open source (f77) and binaries: http://petitjeanmichel.free.fr/itoweb.petitjean.freeware.html

EXAMPLE: FISHER'S IRIS DATA

$n = 150$ 3 classes of 50 individuals each (Setosa, Versicolor, Virginica).

$p = 4$ (sepal length, sepal width, petal length, petal width)

Complete linkage: 5 classes. Sizes: $n_1, ..., n_5 = 29, 60, 12, 28, 21$
Setosa $\equiv$ { class 1 } + { class 5 }
Versicolor $\equiv$ { 23 ind. of class 2 } + { 27 ind. of class 4 }
Virginica $\equiv$ { 37 ind. of class 2 } + { class 3 } + { 1 ind. of class 4 }

Average linkage (quadratic): 7 classes. Sizes: $n_1, ..., n_7 = 49, 4, 37, 24, 23, 12, 1$
Setosa $\equiv$ { class 1 } + { class 7 }
Versicolor $\equiv$ { class 2 } + { 23 ind. of class 3 } + { 1 ind. of class 4 } + { 12 ind. of class 5 }
Virginica $\equiv$ { 14 ind. of class 3 } + { 23 ind. of class 4 } + { 1 ind. of class 5 } + { class 6 }

Single linkage: 5 classes. Sizes: $n_1, ..., n_5 = 50, 93, 2, 1, 4$
Setosa $\equiv$ { class 1 }
Versicolor $\equiv$ { 46 ind. of class 2 } + { class 5 }
Virginica $\equiv$ { 47 ind. of class 2 } + { class 3 } + { class 4 }

MCG produced too much clusters. Setosa was recognized, but not Versicolor and Virginica. Running CG on other data, Jung et al. observed that complete linkage gave good results.

CONCLUSIONS

No clustering method is universally efficient:
it depends on the data, and data can be transformed arbitrarily.
Thus, the optimal number of classes can be arbitrary, too.

Methods without *external* parameter are not proved to be more useful than methods with *external* parameters.

Methods without *external* parameter are just simpler to use:
no need to check the stability of the result for the *external* parameters.

The removal of *external* parameters is an illusion:
at least there is a $<<$ *choice of method* $>>$ parameter.

The user must anyway check the robustness of his conclusions.

SHORT BIBLIOGRAPHY

Marcotorchino F.
Agrégation des similarités en classification automatique.
Doctoral dissertation, University Paris 6, 25 June **1981**.

Petitjean M.,
Agrégation des similarités: une solution oubliée.
RAIRO Oper. Res. **2002**, 36[1], 101-108.
http://www.edpsciences.org/ro

Jung Y., Park H., Du D.Z., Drake B.L.
A Decision Criterion for the Optimal Number of Clusters in Hierarchical Clustering.
J. Glob. Opt. **2003**, 25[1], 91-111.

Meslamani J.E., André F., Petitjean M.,
Assessing the Geometric Diversity of Cytochrome P450 Ligand Conformers by Hierarchical Clustering with a Stop Criterion.
J. Chem. Inf. Model. **2009**, 49[2], 330-337.