STATISTICS IN TRANSITION new series, Autumn 2013
Vol. 14, No. 3, pp. 487–494

EMPIRICAL EVALUATION OF OCLUS AND GENRANDOMCLUST ALGORITHMS OF GENERATING CLUSTER STRUCTURES

Jerzy Korzeniewski[1]

ABSTRACT

The OCLUS algorithm and genRandomClust algorithm are newest proposals of generating multivariate cluster structures. Both methods have the capacity of controlling cluster overlap, but both do it quite differently. It seems that OCLUS method has much easier, intuitive interpretation. In order to verify this opinion a comparative assessment of both algorithms was carried out. For both methods multiple cluster structures were generated and each of them was grouped into the proper number of clusters using *k*-means. The groupings were assessed by means of divisions similarity index (modified Rand index) referring to the classification resulting from the generation. The comparison criterion is the behaviour of the overlap parameters of structures. The monotonicity of the overlap parameters with respect to the similarity index is assessed as well as the variability of the similarity index for the fixed value of overlap parameters. Moreover, particular attention is given to checking the existence of an overlap parameter limit for the classical grouping procedures as well as uniform nature of overlap control with respect to all clusters.

Key words: cluster analysis, cluster structure generation, OCLUS algorithm, genRandomClust algorithm.

1. Introduction

In cluster analysis it is not common that one can achieve far reaching theoretical results, therefore numerical simulations are a popular way of research. Therefore, generating cluster structures resembling real world data sets is a vital task. Across the decades researchers were trying to find better and better generating algorithms. Many proposals were made, however, none seems to be effective enough. In the early days, the inability to control the final degree of the overlap between clusters was the main obstacle. Kuiper and Fisher (1975) generated clusters from multivariate normal distributions, changing their means

[1] University of Lodz, Poland. E-mail: jurkor@wp.pl.

and covariance matrices but the final overlap was unknown. Gold and Hoffman (1976), also generated clusters from multivariate normal distributions adding observations drawn from distributions with different means. The resulting overlap between clusters was unknown. Blashfield (1976) generated clusters from possibly correlated distributions adding outlying observations from the uniform distribution. Also, the final overlap is unknown because some parameters are drawn at random. McIntyre and Bashfield (1980) changed the dispersion of distributions from which clusters were generated but the precise overlap remained undefined. Milligan (1985) generated very well separated distributions from truncated normal distributions adding outlying observations to blur the structures. The final overlap was unknown and some coordinates were distinguished with more stringent conditions imposed on them. Price (1993) controlled the overlap by means of trial and error, shifting the mean vectors until the desired overlap was achieved. However, this method is very time-consuming and not all values of possible overlaps can be assessed in this way. Atlas and Overall (1994) tried to keep overlap control, however, it remained unknown for the whole data set structure. Waller et al. (1999) fund a method which allows one to control overlap but only for low dimensional spaces and the control is visual.

2. OCLUS algorithm

Probably the first algorithm for generating cluster structures in which, in some cases, one has full control over the overlap between particular clusters as well as for the whole data set structure was the OCLUS algorithm (Steinley and Henson, 2005). Every cluster is generated from a predefined distribution and the overlap of each pair of two adjacent clusters also has to be predetermined for every dimension. If we simplify things slightly and assume that every pair of adjacent clusters has the same overlap $p_{j-1,j}$ then the overlap for the whole *i*-th dimension (*marginal overlap*) will be equal to

$$overlap_i = \frac{1}{k-1}\sum_{j=1}^{k-1} p_{j-1,j} \tag{1}$$

where *k* is the number of clusters. If we further assume that the dimensions are independent we can write the overall extent of overlap (*joint overlap*) as

$$overlap = \prod_{i=1}^{T} overlap_i \tag{2}$$

where *T* is the number of dimensions. Figure 1 presents an example of the scheme of generating three clusters with similar number of objects. The overlap between each two adjacent clusters is equal as well as the number of objects in each class.

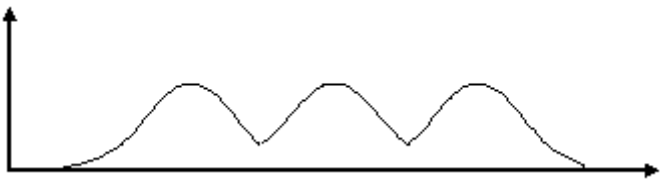

Figure 1. The marginal distribution for generating three clusters with roughly equal number of objects in each cluster.

Table 1. Distances between two adjacent clusters in dependence on overlap and the number T of dimensions.

overlap	T=2	T=4	T=6
0	6	6	6
.1	2	1.16	.82
.2	1.52	.86	.60
.3	1.2	.66	.46
.4	.96	.51	.35

Source: own calculations.

From formula (2) it follows that for the fixed marginal overlap the joint overlap tends to 0 with the growing number of dimensions. From formula (2) it also follows that for the fixed joint overlap the marginal overlap tends to 1 with the growing number of dimensions. We generated cluster structures according to the OCLUS algorithm assuming identical normal distributions with unit variance for every cluster and every dimension. The means of the normal distributions are determined by the overlap and the number of dimensions. The distances between the means of adjacent clusters are given in Table 1. The natural frontier of the overlap parameter for the classical grouping methods assigning every object to unique class is the number 0.5. If this value is exceeded then the clusters are indistinguishable from each other.

3. GenRandomClust algorithm

Quite different way of generating cluster structures can be found in the GenRandomClust algorithm (*Qiu and Joe*, 2006). The user is not obliged to determine the parameters of clusters distributions. Instead, one defines only (so the authors claim) one overlap parameter and the remaining parameters for every cluster (concerning shape and direction) are drawn at random from a defined interval.

When two clusters are generated from the normal distributions with means and covariance matrices, respectively, $\theta_i, \Sigma_i \quad i=1,2$, then their separation measures the authors use is:

$$J_{12}^{*} = \frac{a^T(\theta_2 - \theta_1) - q_{\alpha/2}\left(\sqrt{a^T\Sigma_1 a} + \sqrt{a^T\Sigma_2 a}\right)}{a^T(\theta_2 - \theta_1) + q_{\alpha/2}\left(\sqrt{a^T\Sigma_1 a} + \sqrt{a^T\Sigma_2 a}\right)} \tag{3}$$

where $\alpha \in (0; 0.5)$ is a parameter defining the percentage of outlying observations, $q_{\alpha/2}$

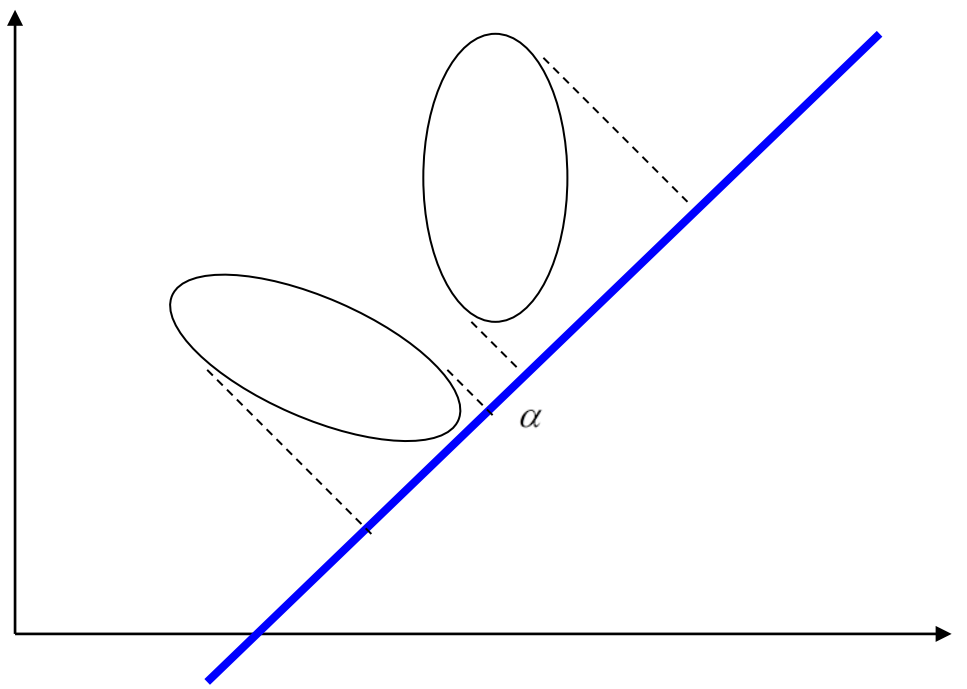

Figure 2. Graphical interpretation of the optimal projection and coefficient *α*.

is a quantile of order $\alpha/2$ of the marginal distribution (after projection see Fig. 2), vector ***a*** is the projection direction maximizing the value of measure J_{12}^{*}.

The measure thus defined takes into account the possibly different dispersions of objects in both clusters. However, the user has to specify the percentage *α.* of outlying observations. Typically, the parameter is assumed to be equal to 0.05. The algorithm works in the following way:

- cluster centers are placed in the vertices of a simplex and covariance matrices are randomly generated (by means of generating eigenvalues from a defined interval) so that the adjacent clusters would have the prespecified value of measure J_{12}^{*} - working as overlap measure.

- cluster centers and covariance matrices are randomly rotated by means of multiplication by an orthogonal matrix.
- objects are generated from the received distributions.

The algorithm thus defined does not allow one to generate clusters from other distributions type than the elliptical one. Neither there is a natural value of overlap parameter which would constitute a frontier between cluster structures meant only for traditional grouping methods and those which allow for objects belonging to more than one cluster. The authors provide function *genRandomClust* available in *R* language, which allows one to generate a vast palette of different cluster structure. However, it is probably impossible to use this function to generate big number of cluster structures with predefined means and covariance matrices. Theoretically, it is possible – the authors provide an algorithm to find the optimal projection direction, but we do not know if this algorithm has found that direction or got stuck in a local solution. Therefore, the value of the overlap parameter may be uncertain for the predefined normal clusters. The authors acknowledge also that only the separation between two closest clusters is controlled but other clusters are probably much better separated from the closest two.

4. Simulation experiment

In order to assess the quality of the cluster structures generated by both algorithms the following experiment was carried out. Data sets with similar cluster structures with respect to the number of clusters, number of dimensions, numbers of objects in each cluster and crossing the whole range of overlap parameter values characteristic for each algorithm were generated. Then, in order to get some numerical ground for the assessment, the objects were grouped into the proper number of clusters and into the number of clusters distorted by 1. When contaminating the number of clusters the number 3 was used instead of 2, number 3 instead of 4 and number 5 instead of 6. The grouping method was the traditional *k*-means with the random choice of starting points, repeated 50 times with the best grouping remembered (see *Steinley and Brusco*, 2007). The quality of the groupings was assessed by means of the corrected Rand index (*Hubert and Arabie*, 1985) of the similarity of the division resulting from the grouping with the one resulting from clusters generation.

For the Joe and Qiu algorithm, the function *genRandomClust* available in the R language was used in two variants: first, for equal cluster sizes (adding up to roughly 200 objects), second, for the size of each cluster drawn randomly from the set {20, …,120}. The value of the overlap control parameter, in the function named *sepval*, had eight values -0.35, -0.25, -0.15, -0.05, 0.05, 0.15, 0.25, 0.35. These values did not cover the whole range of possible variability of *sepval*, i.e. the interval (-1, 1), but the results were clearly predictable for the values above 0.35 and less interesting for the values below -0.35.

For the OCLUS algorithm, the clusters were generated (also in the two variants mentioned above) from the normal distribution with unit variance and the means on each dimension far away from the adjacent means by the values given in Table 1. The clusters were subsequently randomly and independently permutated on every dimension. The value of the overlap control parameter, in this case *joint overlap,* had five values 0.4, 0.3, 0.2, 0.10, 0. The two cases of equal and unequal cluster sizes were considered.

For both algorithms the possible numbers of clusters and dimensions were identical, number of clusters had three variants {2,4,6} and the number of dimensions also had three variants {2,4,6}. Every parameter set up was repeated 5 times for both algorithms.

5. Results and conclusions

The values of the Rand index presented in Tables 2 and 3 are arithmetic means from all 5 repetitions of the same set-ups. The results of the first case of equal cluster sizes are only presented as the second case of unequal cluster sizes gave very similar results for both algorithms. Apart from the mean values of the Rand index the dispersion of these values was also investigated. The examination showed that the standard deviation of the 5 values of the Rand index averaged through the whole table was similar for both algorithms (equal to about 0.1).

There are considerable differences between the two algorithms with respect to the number of clusters and the number of dimensions. The OCLUS algorithm generates structures which are more obscure for smaller number of clusters. There is also some dependence on the number of dimensions. The separability of clusters growing with the number of clusters may be the effect of the grouping method used – it is easier for the *k*-means method to hit (at least partially) the cluster centers when there are more of them than when there are just two clusters. The dependence on the number of dimensions is not considerable. There are no such correlations for the genRandomClust algorithm. All structures generated by this algorithm keep roughly the same quality for all numbers of clusters and dimensions considered.

The greatest difference between the two algorithms, however, was revealed for distorted number of clusters. It is very common that while examining the real world data sets we do not know the true number of clusters, therefore it is very vital to check the efficiency of cluster analysis methods in this case. The structures generated by the OCLUS algorithm did not suffer very much – there was a 5% or smaller drop (or rise) for the distorted number of clusters. The structures generated by the genRandomClust algorithm turned out to be very non-robust to these distortions – there always was a 20%-30% drop of the values of the Rand index. This feature seems to have some serious consequences. It corroborates the authors surmise of the lack of overlap control over the rest of all clusters apart from the two closest. The consequences depend on what kind of

efficiency investigation one will use the cluster structures for. If one wants to test a number of clusters index, then it is highly unadvisable to use the genRandomClust to generate cluster structures because the index may tend to be satisfied with the smaller number of well separated clusters. The interpretation is not clear when one wants to test a grouping method, however, since the case of the distorted number of clusters should be considered, it seems that the results may be seriously affected. A good example is that of the model based approaches to cluster analysis in which both the number of clusters assessment as well as the classification of objects are done simultaneously.

Table 2. Values of the Rand index for the *genRandomClust* algorithm

sepval		-.35	-.25	-.15	-.05	.05	.15	.25	.35
Number of clusters	Number of dimen.								
2	2	.29	.49	.68	.86	.91	.97	1.00	1.00
	4	.32	.53	.60	.84	.92	.96	1.00	1.00
	6	.30	.51	.62	.82	.89	.94	.99	1.00
4	2	.27	.47	.62	.79	.91	.98	.99	0.99
	4	.30	.41	.62	.80	.88	.94	.99	1.00
	6	.31	.46	.61	.83	.87	.95	.99	1.00
6	2	.30	.48	.69	.83	.91	.93	.99	1.00
	4	.28	.49	.65	.81	.88	.96	.98	1.00
	6	.29	.50	0.70	.81	.87	.92	1.00	1.00

Source: Own calculations.

Table 3. Values of the Rand index for the *OCLUS* algorithm

Overlap		.4	.3	.2	.1	0
Number of clusters	Number of dimen.					
2	2	.35	.43	.56	.81	1.00
	4	.21	.36	.46	.68	1.00
	6	.20	.35	.45	.62	1.00
4	2	.47	.62	.66	.83	1.00
	4	.68	.71	.85	.91	1.00
	6	.70	.74	.86	.92	1.00
6	2	.56	.61	.71	.76	.91
	4	.72	.77	.80	.85	.89
	6	.86	.89	.89	.95	1.00

Source: Own calculations.

REFERENCES

ATLAS, R., OVERALL J., (1994). Comparative Evaluation of Two Superior Stopping Rules for Hierarchical Cluster Analysis, Psychometrika, 59, 581–591.

BLASHFIELD, R. K., (1976). „Mixture Model Tests of Cluster Analysis: Accuracy of Four Agglomerative Hierarchical Methods", Psychological Bulletin, 83, 377–388.

GOLD, E., HOFFMAN P., (1976). Flange Detection Cluster Analysis, Multivariate Behavioral Research, 11, 217–235.

HUBERT, L., ARABIE, P., (1985). Comparing Partitions, Journal of Classification 2.

KUIPER, F. K., FISHER, L., (1975). A Monte Carlo Comparison for Six Clustering Procedures, Biometrics, 31, 777–784.

MCINTYRE, R., BLASHFIELD, R., (1980). A Nearest-Centroid Technique for Evaluating the Minimum Variance Clustering Procedure, Multivariate Behavioral Research, 15, 225–238.

MILLIGAN, G., (1985). An Algorithm for Generating Artificial Test Clusters, Psychometrika, 50, 123–127.

PRICE, L., (1993). Identifying Cluster Overlap with NORMIX Population Membership Probabilities, Multivariate Behavioral Research, 28, 235–262.

QIU, W., JOE H., (2006). Generation of random clusters with specified degree of separation, Journal of Classification 23, 315–334.

STEINLEY, D., BRUSCO, M., (2007). Initializing k-means batch clustering: A critical evaluation of several techniques, Journal of Classification 24, 99–121.

STEINLEY, D., HENSON, R., (2005). OCLUS: An Analytic Method for Generating Clusters with Known Overlap, Journal of Classification 22, 221–250.

WALLER, N., UNDERHILL, J, KAISER, H., (1999). A Method for Generating Simulated Plasmodes and Artificial Test Clusters with User-Defined Shape, Size and Orientation. Multivariate Behavioral Research, 34, 123–142.