The X-Alter algorithm : a parameter-free method to perform unsupervised clustering

Thomas Laloë, Rémi Servien

The X-Alter algorithm : a parameter-free method to perform unsupervised clustering

Thomas Laloë[1], Rémi Servien[2]

Abstract

Using quantization techniques, Laloë (2010) defined a new algorithm called Alter. This L^1-based algorithm is proved to be convergent, but suffers two shortcomings. First, the number of clusters K has to be supplied by the user. Second, it has high complexity. In this article, we adapt the idea of X-means algorithm by Pelleg and Moore (2000) to offer solutions for these problems. This fast algorithm is used as a building-block which quickly estimates K by optimizing locally the Bayesian Information Criterion (BIC). Our algorithm combines advantages of X-means (calculation of K and speed) and Alter (convergence and parameter-free). Finally, an aggregative step is performed to adjust the relevance of the final clustering according to BIC criterion. We also confront our algorithm to different simulated and real data sets, which shows its relevance.

Keywords: Clustering, Quantization, K-means, Free-parameter algorithm

1. Introduction

Clustering consists in partitioning a data set into subsets (or clusters), so that the data in each subset share some common trait. Proximity is determined according to some distance measure. For a thorough introduction to the subject, we refer to the book by Kaufman and Rousseeuw [8]. The origin of clustering goes back to 45 years ago, when some biologists and sociologists began to search

[1]Corresponding author, Université de Nice Sophia-Antipolis, Laboratoire J-A Dieudonné, Parc Valrose, 06108 Nice Cedex 02. Thomas.Laloe@unice.fr, http://math.unice.fr/~laloe/

[2]Montpellier SupAgro-INRA, UMR MISTEA 729, 2 place Pierre Viala, 34060 Montpellier Cedex 2, France. remi.servien@supagro.inra.fr

for automatics methods to build different groups with their data. Today, clustering is used in many fields. For example, in medical imaging, it can be used to differentiate between types of tissue and blood in a three dimensional image. Market researchers use it to partition the general population of consumers into market segments and to better understand the relationships between different groups of consumers/potential customers. There are also many different applications in artificial intelligence, sociology, medical research, or political sciences.

The K-means clustering is the most popular method of clustering [5, 13]. Its attractiveness lies in its symplicity and its fast execution. It has however two main shortcomings. One, the number of clusters K has to be supplied by the user. Thus, different ways to determine K have been studied in the litterature [11, 15]. Two, the algorithm strongly depends on the initialisation and can easily converge to a local minimum. Pelleg and Moore [14] offer a solution for the first problem with a building-block algorithm called X-means which quickly estimates K. It goes into action after each run of 2-means, making local decisions about which subsets of the current centroid should split themselves in order to better fit the data. The splitting decision is done by computing the Bayesian Information Criterion (BIC). On another hand, Laloë [10] proposes a consistent algorithm, called Alter, which also needs the specification of K.

The purpose of this paper is to combine the X-means and the Alter algorithm in order to overcome the drawbacks of both the algorithms. Besides decreasing the complexity of the Alter algorithm, it allows an automatic selection of the number of clusters. Moreover, thanks to the convergence properties of the Alter algorithm, we can also hope it will overcome the local optimality problem of the X-means algorithm, inherited from the K-means one.

The paper is organized as follows: section 2 recalls the Alter algorithm. Section 3 presents the X-Alter algorithm. Performances of X-Alter, X-means and another algorithm are compared in Section 4.

2. The Alter algorithm

Let us first recall the background of the Alter algorithm. All the theoretical results presented in this section come from Laloë [10]. The theoretical method which supports this algorithm is the quantization. The quantization is a commonly used technique in signal compression [3, 12]. Given a normed space $(\mathcal{H}, \|.\|)$, a codebook (of size K) is defined by a subset $\mathcal{C} \subset \mathcal{H}$ with cardinality K. Then, each $x \in \mathcal{H}$ is represented by a unique $\hat{x} \in \mathcal{C}$ via the function q,

$$\begin{aligned} q: \quad \mathcal{H} &\to \mathcal{C} \\ x &\to \hat{x}, \end{aligned}$$

which is called a quantizer. Here we come back to the clustering, as we create clusters in the data by regrouping the observations which have the same image by q. More precisely, these images by q are the representants of the clusters.

Denote by d the distance induced by the norm L^1 on $\mathcal{H}$:

$$\begin{aligned} d: \mathcal{H} \times \mathcal{H} &\to \mathbb{R}^+ \\ (x, y) &\to \|x - y\|. \end{aligned}$$

Considering a random variable X on $\mathcal{H}$, with distribution μ, the quality of the approximation of X by $q(X)$ is then given by the distortion $\mathbb{E}\, d\big(X, q(X)\big)$. Thus the aim is to minimize $\mathbb{E}\, d\big(X, q(X)\big)$ among all possible quantizers. However, in practice, the distribution μ of the observations is unknown, and we only have at hand n independent observations $X_1, \ldots, X_n$ with the same distribution than X. The goal is then to minimize the empirical distortion:

$$\frac{1}{n} \sum_{i=1}^{n} d(X_i, q(X_i)).$$

We choose here L^1-based distortion to lead to more robusts estimators. For a discussion of the advantage of the L^1-distortion we refer the reader to Kemperman [9].

Theoritical results of consistency and rate of convergence of this method have been proved in Laloë [10]. In particular, it is stated that the rate of convergence is closely related to the metric entropy of the space of the data. However, the minimization of the empirical distortion is not possible in practice. A possible alternative is to perform the Alter algorithm. The idea is to select an optimal codebook among the data. More precisely the outline of the algoritm is:

1. List all possible codebooks (set of the K centers of the clusters), i.e., all possible K-tuples of data;
2. Calculate the empirical distortion associated to the first codebook;
3. For each successive codebook, calculate the associated empirical distortion. Each time a codebook has an associated empirical distortion smaller than the previous smallest one, store the codebook;
4. Return the codebook which has the smallest distortion.

Again, theoritical results of consistency and rate of convergence have been proved for the Alter algorithm. In particular it is stated that the convergence rate is of the same order than for the theoretical methods above. Moreover, this algorithm does not depend on initial conditions (unlike the K-means algorithm) and it converges to the optimal distortion. Unfortunately its complexity is $o(n^{K+1})$ and it is impossible to use it for high values of n or K.

Following the idea of the X-means algorithm, a recursive utilisation with $K = 2$ could allow us to bring down the time consuming of the algorithm and give us an estimation of K.

3. The *X*-Alter Algorithm

Remember that the idea of the X-means algorithm lies on the recursive utilisation of 2-means. After each run of 2-means, the splitting decision is done using the BIC criterion. Then, the algorithm runs in each generated subset since there is no split available. We couple here this idea with the Alter algorithm and we

add an aggregation final step to prevent the creation of too much clusters.

Note that no parameter is needed by the algorithm. Though, the user can specify a range in which the true K reasonably lies (which is $[2, +\infty[$ if we had no information).

The algorithm starts by performing Alter with $K = 2$ centroids. After this, the structure improvement operation begins by splitting each cluster into two children. The procedure is local on that the children are fighting each other for the points in the parent's region, no others. At this point, a model selection test is performed on all pairs of children. The test aks if the model with the two offsprings is better than the one with his parent. If the answer is yes, the iterative procedure occurs in the two children. If not, the region is asleep and the algorithm tries to investigate other regions where more clusters are needed.

More precisely, the outline of the algorithm is the following:

1. Clustering in 2 clusters using Alter (Figure 1). That is we list all possible pairs of data and take the one who minimize the empirical distortion;
2. Then we perform a new discrimination in two clusters within each cluster previously obtained (Figure 2). This gives us clustering C;
3. We use the following formula from Kass and Wasserman [7] for the BIC criterion. It evaluates the relevance of the classification C with

$$BIC(C) = l - \frac{p}{2} \log n$$

where l is the log-likelihood of the data according to the clustering C and taken at the maximum likelihood point, and p is the number of parameters in C. The number of free parameters p is simply the sum of $K - 1$ class probabilities, $d*K$ centroids coordinates, and one variance estimate. Note that we suppose here that in each cluster, the data are normally distributed around the center. Using this criterion, we check the suitability of the

discrimination (Figure 2) by comparing $BIC(K = 1)$ and $BIC(K = 2)$ on each subset;

4. We iterate step two and three until there are no more relevant discrimination (Figure 3);
5. Final step of aggregation: All pairs of clusters are tested and aggregated according to the value of the criterion BIC (Figure 4): aggregation can be considered if $BIC(K = 1) > BIC(K = 2)$. The bigger value of $BIC(K = 1) - BIC(K = 2)$ gives the first aggregation. Next aggregations are performed according to the decreasing values of the BIC differences until it runs out of relevants.

The complexity of this algorithm in the worst case scenario (that is when it creates n clusters with one data) is $o(n^4)$, what makes it more easily usable than the inital Alter algorithm. However, it is still bigger than the complexity of the X-means algorithm. But the consistent property of Alter must improve results and the aggregation step must prevent the presence of truncated clusters (as in Figure 2).

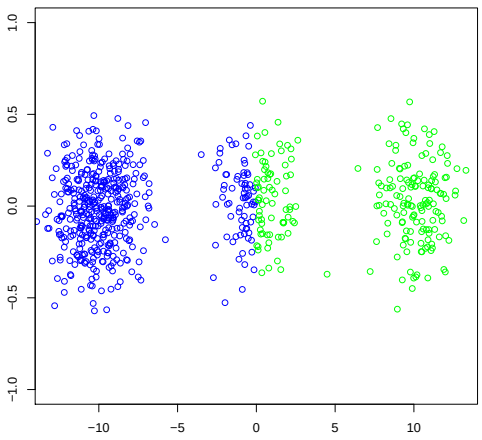

Figure 1: First iteration of X-Alter algorithm (Step 1.)

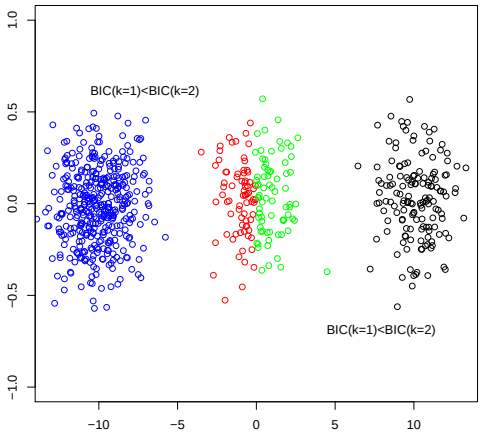

Figure 2: Second sub-classification in the two relevant clusters (Step 2.). Sub-classifications are validated by BIC (Step 3.) so we obtain four clusters.

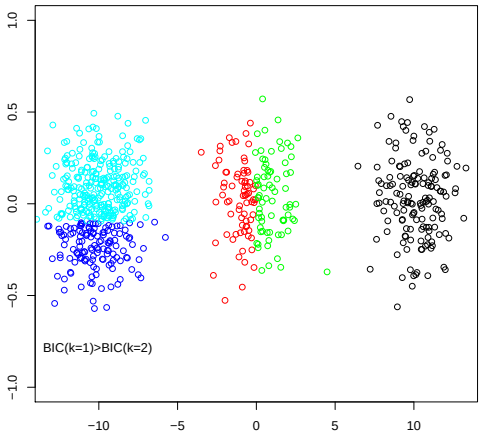

Figure 3: No relevant sub-classification in the left cluster according to BIC. In the three other clusters, we obtain the same rejection of sub-classification (Step 4.).

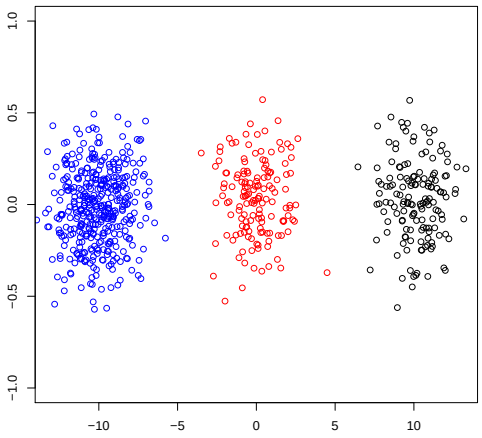

Figure 4: Final discrimination. The two middle clusters have been aggregated in Step 5.

4. Empirical study

In this section, we perform an empirical study to establish the relevance of our method. We confront our method to various simulated data sets, but also on classical real data sets. We consider three criterion: the number of clusters found, the Adjusted Rand Index (A.R.I.) [6, 16] and the Dunn index [1, 4]. The A.R.I. is the corrected-for-chance version of the Rand index which is a measure of the similarity between two clusters. More clusters are similars (respectively dissimilars), closer to 1 (respectively 0) the A.R.I. is. On another way, the Dunn Index measures the "compactness" of the clusters and is a sort of the worst case indicator. Higher is the Dunn Index, better the clustering is. For more details on this classical cluster validation indexes we refer the reader to the given references.

Pelleg and Moore shown that the X-means algorithm performs better and faster than repeateadly using accelerated K-means for different values of K. So, we compare our X-Alter algorithm to X-means and to X-means with the aggregation step, called X-means-R.

4.1. Simulated data

4.1.1. A simple case

We simulate here clusters of gaussian vectors in $\mathbb{R}^d$ (d can be different in each following case).

First, in Table 1 we consider two clusters well identified in $\mathbb{R}^{20}$. More precisely we simulate two clusters of 25 vectors (in $\mathbb{R}^{20}$) with $\sigma_1^2 = \sigma_2^2 = 100$ and $\mu_1 = -\mu_2 = 15$. The results are averaged on 300 simulations.

Algorithm	% of good number of clusters	A.R.I.	Dunn Index
X-means	99	1	1.62
X-means-R	100	1	1.64
X-Alter	100	1	1.64

Table 1: Results of the three algorithms for the two well-defined clusters.

As expected, we note that the three methods perform well on this very simple case.

Now we consider three simulated clusters well identified in $\mathbb{R}^5$. This allows us to see the relevance of the aggregation step, as X-means should often cut the middle cluster in its first iteration. More precisely we simulate two clusters of 20 vectors (in $\mathbb{R}^5$) with $\sigma_1^2 = \sigma_2^2 = 100$ and $\mu_1 = -\mu_2 = 20$; and one cluster of 20 vectors with $\sigma_3^2 = 100$ and $\mu_3 = 0$. The results are averaged on 300 simulations and gathered in Table 2.

Algorithm	% of good number of clusters	A.R.I.	Dunn Index
X-means	55	0.82	0.22
X-means-R	76	0.82	0.22
X-Alter	86	0.84	0.22

Table 2: Results for the three algorithms on the three clusters.

We see here the influence of the aggregation steps, since X-means-R find the good number of cluster almost fourty percent time more often than X-means. Moreover, we note that our algorithm obtains better results than the other two: the convergence property of Alter clearly improves results.

Finally we perform tests with random values for the numbers of clusters, the mean, standard deviation and number of data in each clusters. The μ_i are randomly selected between -50 and 50, the σ_i between 5 and 15, the number of clusters between 2 and 10, the number of vectors in each cluster between 8 and 25. The dimension of the data is fixed to 10. Table 3 summaries the results averaged on 300 simulations.

Again, we see that our algorithm obtains better results than the other two for the number of clusters and that A.R.I. and Dunn Index are slightly the same.

Algorithm	% of good number of clusters	A.R.I.	Dunn Index
X-means	63	0.96	0.60
X-means-R	71	0.97	0.60
X-Alter	91	0.96	0.59

Table 3: Results for the three algorithms on the random clusters.

4.1.2. Functionnal case

Now we consider functionnal data. Here, we must also compare times of executions which were slightly the same previously. We consider two configurations:

First, we take functions $\sqrt{x}$, x and x^2 in $[0, 1]$ discretized 20 times. We disturb each of these functions by adding $\cos(10x + \pi/2 - 10)/5$. Each data in $\mathbb{R}^{20}$ is noised with a vector composed by twenty gaussian law $N(0, \sigma)$ where the value of σ is selected for each data using $\sigma \sim N(0.1, 0.02)$. Figure 5 shows examples of some functions to classify.

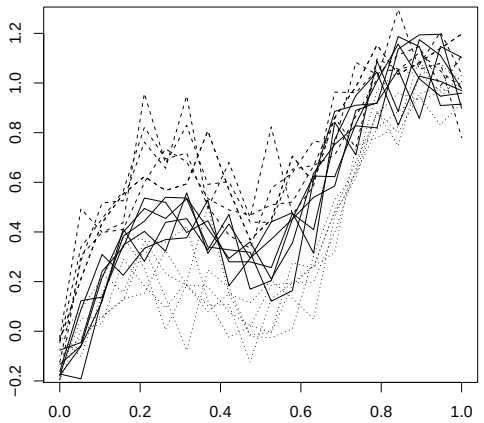

Figure 5: Example of functions. Functions based on $\sqrt{x}$ are on dashed lines, ones based on x are on solid lines and ones based on x^2 are on dotted lines.

Three clusters of size randomly chosen between 15 and 25 are simulated 300 times. Results are presented in Table 4 (time is given in seconds).

Algorithm	% of good number of clusters	A.R.I.	Dunn	Time
X-means	81	0.88	0.63	2.0
X-means-R	85	0.88	0.63	3.5
X-Alter	95	0.89	0.63	27.6

Table 4: Results for the three algorithms on the functionnal data.

We can see that our method gives better results, mostly on the search of the number of clusters.

Second, we consider a slightly more difficult case. We construct this configuration on the same model than the first, but based on functions $\sqrt{x}$, $x^{3/4}$ and x which are closer than previous ones as we can see in Figure 6.

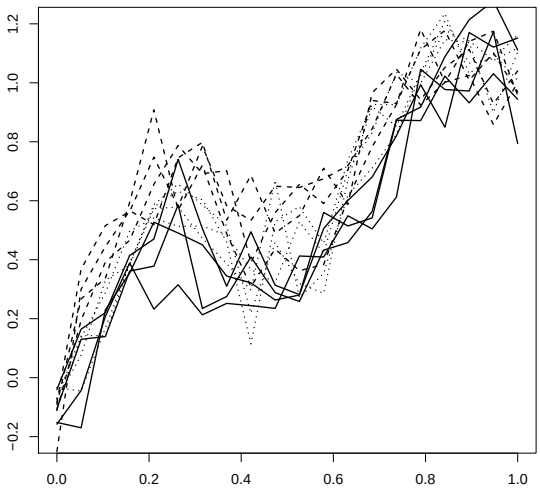

Figure 6: Example of functions. Functions based on $\sqrt{x}$ are on dashed lines, ones based on x are on solid lines and ones based on $x^{3/4}$ are on dotted lines.

Results are gathered in Table 5.

Algorithm	% of good number of clusters	A.R.I.	Dunn	Time
X-means	26	0.75	0.43	2.4
X-means-R	31	0.75	0.46	3.2
X-Alter	40	0.77	0.46	28.7

Table 5: Results for the three algorithms on the functionnal data.

Again, we see that our method retrieves more often the good number of clusters. Note that if the complexity of our algorithm is bigger than the X-means one, it is much smaller than the Alter one. Indeed Alter algorithm does not estimate the number of clusters.

Robustness study

In this paragraph, we illustrate the robustness properties of the L_1 distance. We consider as a starting point the first functionnal configuration above : $\sqrt{x}$, x and x^2 in $[0,1]$ noised with $\cos(10x+\pi/2-10)/5$ (Figure 5). To perturb data we use the following protocol : we add a value $x \in [-0.30;-0.15] \cup [0.15;0.30]$ to $a \in [10;25]$ percent of points (randomly chosen) of $b \in [10;25]$ percent of data (randomly chosen). An example is given in Figure 7. We repeat this 300 times and give averaged results in Table 6.

The relevance of the L^1-based distance error, which is much more robust to extrem values, is shown here. Indeed, if we compare to the results gathered in Table 4 we still find the good number of clusters 95% of the time while X-means and X-means-R suffer from a loss of respectively 4% and 6%.

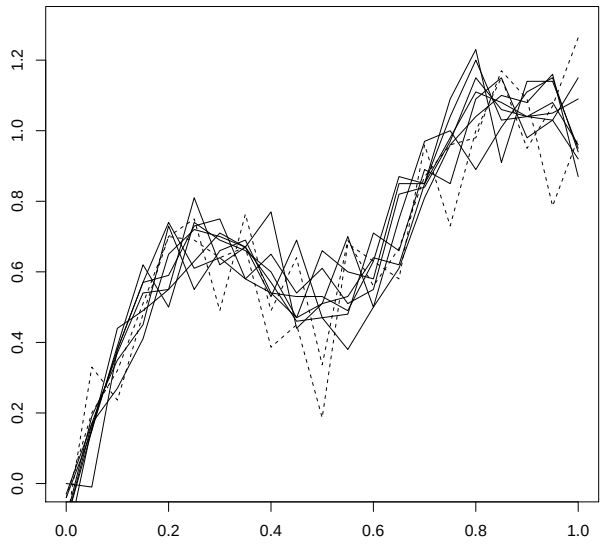

Figure 7: Example of the results of the pertubation of $\sqrt{x}\cos(10x + \pi/2 - 10)/5$. Affected functions are on dashed lines.

Algorithm	% of good number of clusters	A.R.I.	Dunn	Time
X-means	77	0.87	0.52	2.6
X-means-R	79	0.87	0.52	3.8
X-Alter	95	0.88	0.53	29.4

Table 6: Results for the three algorithms on the perturbated functionnal data sets.

4.2. Real data

In this section, we confront our method to two conventional data sets from the UCI Machine Learning Repository [2]: the wine and iris ones. In this case, we do not know if the spherical gaussian assumption of the BIC criterion is verified. We compare our method to the X-means algorithm but also to the K-means algorithm with K known to be 3 (the real number of clusters here). In these two

cases, as suggested in the description of the data sets, we center and standardize each variable before performing clustering.

Since K-means, X-means and X-means-R depends on the initialisation, we give averaged results (over 50 runnings) for these methods.

4.2.1. Wine data set

We consider first the wine data set. We have 178 instances and 13 variables found in each of the three types of wines. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. In a classification context, this is a well posed problem with "well behaved" class structures. The results for the 4 methods are presented in Table 7.

Algorithm	Number of clusters	A.R.I.	Dunn
X-means	8.67 (var=6.92)	0.78 (var=0.03)	0.162 (var=2.10^{-4})
X-means-R	8.54 (var=6.01)	0.78 (var=0.03)	0.165 (var=10^{-4})
3-means	$\emptyset$	0.76 (var=0.03)	0.163 (var=0.0002)
X-Alter	3	0.76	0.142

Table 7: Results for the wine data set.

We can see that our method retrieve the real number of clusters, and that we get the same adjusted rand index than 3-means and slighty less than the 2 others.

4.2.2. Iris data set

We consider now the Iris data set. We have 150 instances and 4 variables of 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other which makes it more difficult to classify. The results are gathered in Table 8.

Algorithm	Number of clusters	A.R.I.	Dunn
X-means	13.7 (var=6.2)	0.46 (var=0.07)	0.0405 (var=6.10^{-5})
X-means-R	8 (var=1.56)	0.57 (var=0.03)	0.0398 (var=0)
3-means	$\emptyset$	0.46 (var=0.0036)	0.04 (var=0)
X-Alter	6	1	0.402

Table 8: Results for Iris data set.

It appears that our method do not find the real number of clusters but gets closer to it than others. While Adjusted Rand Index were previously very close for all methods, X-Alter is here significantly better and is maximum. Indeed, as we consider here the Adjusted Rand Index (and not the Rand Index), it doesn't mean that our classification is perfect. Moreover, we observe the interest of the aggregation step in X-means-R and it seems to appear that the spherical gaussian assumption required for the BIC is reasonable.

Finally, we see that in all cases (simulated or real data sets) our method performs better than others to estimate the number of clusters. This confirms that we avoid the local convergence property of X-means, which is inherited from K-means. Furthermore, according to Adjusted Rand Index and to Dunn Index, quality of clustering is either equal or significantly better than other methods.

5. Conclusion

We have presented a simple new algorithm to perform clustering. The main advantage of this method is that it is parameter-free. So, it can be easily used without an expertise knowledge of the data. This algorithm combines Alter and X-means algorithm in order to benefit of qualities of both (respectively the convergence and the automatic selection of the number of clusters). Moreover, we avoid the main drawbacks of these two methods which are the high complexity for Alter and the dependence on initials conditions for X-means. A confrontation on both simulated and real data sets shows the relevance of this method.

However, even if the complexity is lowered (with respect to the Alter algorithm) it is still too important to apply the method on really big data sets. So as a future work, it could be interesting to look for another way to accelerate Alter while preserving (as much as possible) its properties of convergence.

References

[1] Dunn, J. [1974]. Well separated clusters and fuzzy partitions, *Journal on Cybernetics* **4**: 95–104.

[2] Frank, A. and Asuncion, A. [2010]. UCI machine learning repository. **URL:** *http://archive.ics.uci.edu/ml*

[3] Graf, S. and Luschgy, H. [2000]. *Foundations of Quantization for Probability Distributions*, Vol. 1730 of *Lecture Notes in Mathematics*, Springer-Verlag, Berlin.

[4] Handl, J., Knowles, K. and Kell, D. [2005]. Computational cluster validation in post-genomic data analysis, *Bioinformatics* **21**: 3201–3212.

[5] Hartigan, J. and Wong, M. [1979]. A k-means clustering algorithm, *Journal of the Royal Statistical Society* **28**: 100–108.

[6] Hubert, L. and Arabie, P. [1985]. Comparing partitions, *Journal of Classification* **2**(1): 193–218.

[7] Kass, R. and Wasserman, L. [1995]. A reference bayesian test for nested hypotheses and its relationship to the schwarz criterion, *Journal of the american statistical association* **90**: 928–934.

[8] Kaufman, L. and Rousseeuw, P. [1990]. *Finding Groups in Data: an Introduction to Cluster Analysis*, John Wiley & Sons, New-York.

[9] Kemperman, J. H. B. [1987]. The median of a finite measure on a Banach space, *Statistical Data Analysis Based on the L_1-norm and Related Methods (Neuchâtel, 1987)*, North-Holland, Amsterdam, pp. 217–230.

[10] Laloë, T. [2010]. L_1 quantizationand clustering in banach spaces, *Mathematical Methods of Statistics* **19**(2): 136–150.

[11] Li, M., Ng, M., Cheung, Y.-M. and Huang, J. [2008]. Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters, *IEEE transactions on knowledge and data engineering* **20**: 1519–1534.

[12] Linder, T. [2002]. Learning-theoretic methods in vector quantization, *Principles of Nonparametric Learning (Udine, 2001)*, Vol. 434 of *CISM Courses and Lectures*, Springer, Vienna, pp. 163–210.

[13] MacQueen, J. [1967]. Some methods for classification and analysis of multivariate observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press.

[14] Pelleg, D. and Moore, A. [2000]. X-means: Extending k-means with efficient estimation of the number of clusters, *Proceedings of the Seventeenth International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, pp. 727–734.

[15] Pham, T., Dimov, S. and Nguyen, C. [2005]. Selection of K in K-means clustering, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* **219**: 103–119.

[16] Rand, W. [1971]. Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical Association* **66**(336): 846–850.