



HAL
open science

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

Thomas Laloë, Rémi Servien

► **To cite this version:**

Thomas Laloë, Rémi Servien. The X-Alter algorithm : a parameter-free method to perform clustering. 2011. hal-00674407v2

HAL Id: hal-00674407

<https://hal.science/hal-00674407v2>

Preprint submitted on 27 Feb 2012 (v2), last revised 26 Jun 2013 (v5)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Thomas Laloë
Université de Nice Sophia Antipolis,
Laboratoire J-A Dieudonné,
Parc Valrose, 06108 Nice Cedex 2,
France

Rémi Servien
Montpellier SupAgro-INRA,
UMR MISTEA 729,
Bâtiment 29, 2 place Pierre Viala,
34060 Montpellier Cedex 2,
France

Abstract

Using quantization techniques, Laloë (2009) defined a new algorithm called Alter. This L^1 -based algorithm is proved to be convergent, but suffers two shortcomings. Firstly the number of clusters K has to be supplied by the user. Secondly it has a high complexity. In this article, we adapt the idea of X -means algorithm (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 confront here our algorithm to different simulated data sets, which show its relevance.

Key-words and phrases: 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 of the subject, we refer to the book by Kaufman and Rousseeuw (1990). The origin of clustering goes back to 45 years ago, when some biologists and sociologists began to search for automatic 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 (MacQueen; 1967; Hartigan and Wong; 1979). Its attractiveness lies in its simplicity, 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 literature (Pham et al.; 2005; Li et al.; 2008). Two, the algorithm strongly depends on the initialisation and can easily converge to a local minimum. Pelleg and Moore (2000) 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 centroids 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 (2009) 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 (2009). The theoretical method which supports this algorithm is the quantization. The quantization is a commonly used technique in signal compression (Graf and Luschgy; 2000; Linder; 2002). Given a normed space $(\mathcal{H}, \|\cdot\|)$, 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 ,

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

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} :

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

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 $\mathbb{E} d(X, q(X))$. Thus the aim is to minimize $\mathbb{E} d(X, q(X))$ 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, \dots, 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 robust estimators. For a discussion of the advantage of the L^1 -distortion we refer the reader to Kemperman (1987).

Theoretical results of consistency and rate of convergence of this method have been proved in Laloë (2009). 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 algorithm 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, theoretical results of consistency and rate of convergence have been proved for the Alter algorithm. 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.

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 according. 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 asks 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);
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 (1995) 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 . Using this criterion, we check the suitability of the discrimination (Figure 2) by comparing $BIC(K = 1)$ and $BIC(K = 2)$ on our 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 relevant.

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 initial 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).

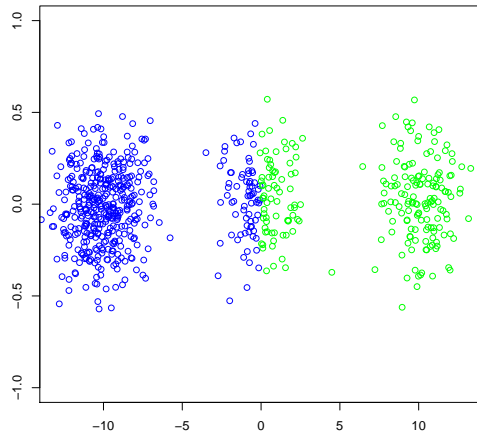


Figure 1: First iteration of X -Alter algorithm (Step 1.)

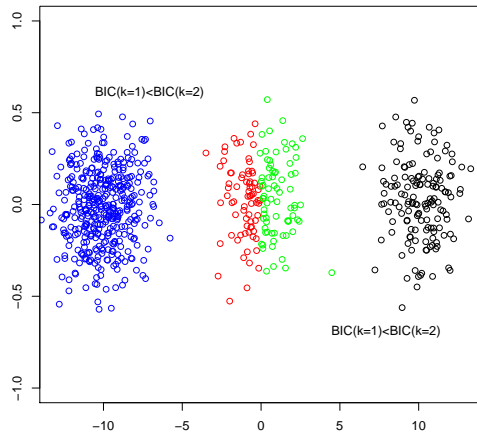


Figure 2: Second sub-classification in the two relevant clusters (Step 2.). Sub-classifications are validated by BIC (Step 3.) so we obtain four clusters.

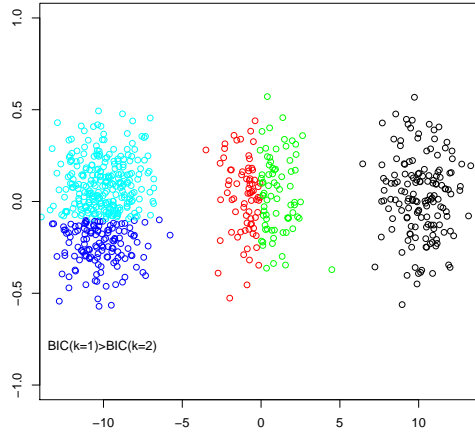


Figure 3: No relevant sub-classification in the left cluster according to BIC. In the three other clusters, we obtain the same rejection of sub-classification (Step 4.).

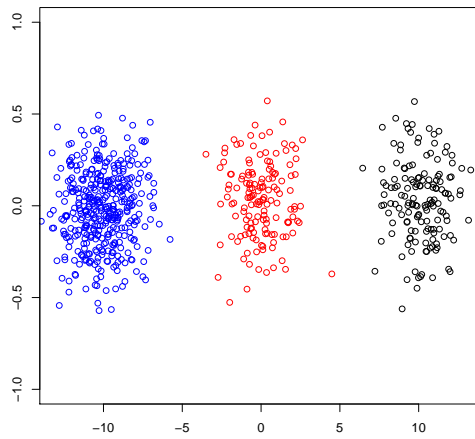


Figure 4: Final discrimination. The two middle clusters have been aggregated in Step 5.

4 Simulations

In this section, we perform an empirical study in order to establish the relevance of our method. To this aim, we consider two criterion: the number of clusters and the Correct Classification Rate (CCR) which is the percentage of data that are in the same cluster in both the real and estimated clustering (Genolini and Falissard; 2010). Pelleg and Moore shown that the X -means algorithm performs better and faster than repeatedly 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 first consider a simple case: we simulate clusters of normal vector in \mathbb{R}^d . First, in Table 1 we consider two clusters well identified in \mathbb{R}^{20} . More precisely we simulate two clusters of 20 data 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	CCR
X -means	99	0.99
X -means-R	100	1
X -Alter	100	1

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 data with $\sigma_1^2 = \sigma_2^2 = 100$ and $\mu_1 = -\mu_2 = 20$; and one cluster 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	CCR
X -means	56	0.88
X -means-R	78	0.91
X -Alter	86	0.91

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 two others: 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 data 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.

Algorithm	% of good number of clusters	CCR
X -means	71	0.93
X -means-R	77	0.94
X -Alter	97	0.96

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

Again, we see that our algorithm obtains better result than the two others mainly for the number of clusters.

4.1.2 Fonctionnal case

Now we consider fonctionnal 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 randomly selected for each data using $\sigma \sim N(0.1, 0.02)$. Figure 5 shows examples of some functions to classify. Three clusters of size randomly chosen between 15 and 25 are simulated 300 times. Results are gathered in Table 5. Then, in order to illustrate the robustness properties of the L_1 distance we add an extreme fonction to the data set. In this last case, we consider that the good number of clusters is found if the extreme value is in the nearest cluster with all the others values or alone in a cluster. Again the results are averaged on 300 simulations (Table 5).

In these two cases, our method gives better results on the search of the number of clusters and on the CCR. Further, our results are not pertubated by the extreme value unlike K -means-type algorithm. The complexity of our algorithm is still bigger than the X -means one, but is much smaller than the

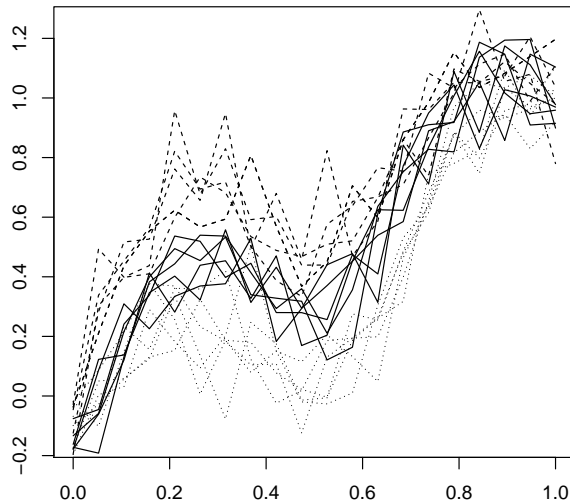


Figure 5: Example of functions. Functions based on \sqrt{x} are on dashed lines, ones based on x are on solid lines and ones based on x^2 are on dotted lines.

Algorithm	% of good number of clusters	CCR	Time (in seconds)
X -means	82	0.89	2.0
X -means-R	86	0.90	3.5
X -Alter	99	0.91	29.6

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

Alter one. Indeed Alter algorithm does not estimate 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. Again, in a second time, we add an extreme function. Results are gathered in Tables 6 and 7.

We see that our method retrieves more often the good number of clusters. It is particularly true for the second configuration. We see here the interest of the L^1 -based distance error which is much more robust to extrem values.

Algorithm	% of good number of clusters	CCR	Time (in seconds)
X -means	76	0.86	2.6
X -means-R	82	0.89	3.8
X -Alter	99	0.91	29.4

Table 5: Results for the three algorithms on the functional data with one extreme function.

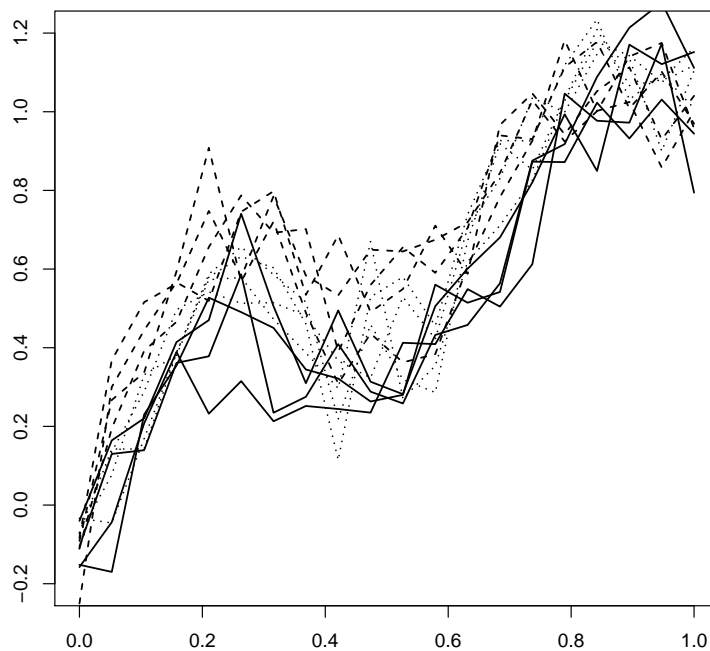


Figure 6: Example of functions. Functions based on \sqrt{x} are on dashed lines, ones based on x are on solid lines and ones based on $x^{3/4}$ are on dotted lines.

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 -

Algorithm	% of good number of clusters	CCR	Time (in seconds)
X-means	28	0.73	2.4
X-means-R	32	0.75	3.2
X-Alter	40	0.75	28.7

Table 6: Results for the three algorithms on the fonctionnal data.

Algorithm	% of good number of clusters	CCR	Time (in seconds)
X-means	24	0.67	2.2
X-means-R	27	0.70	3.9
X-Alter	47	0.74	31.5

Table 7: Results for the three algorithms on the fonctionnal data with one extreme function.

means. Results of simulation show the relevance of this method.

References

- Genolini, C. and Falissard, B. (2010). Kml: k -means for longitudinal data, *Computational Statistics* **25**: 317–328.
- Graf, S. and Luschgy, H. (2000). *Foundations of Quantization for Probability Distributions*, Vol. 1730 of *Lecture Notes in Mathematics*, Springer-Verlag, Berlin.
- Hartigan, J. and Wong, M. (1979). A k -means clustering algorithm, *Journal of the Royal Statistical Society* **28**: 100–108.
- 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.
- Kaufman, L. and Rousseeuw, P. (1990). *Finding Groups in Data: an Introduction to Cluster Analysis*, John Wiley & Sons, New-York.
- 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.
- Laloë, T. (2009). L_1 quantization and clustering in banach spaces, *Mathematical Methods of Statistics*.
- 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.

- 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.
- 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.
- Pelleg, D. and Moore, A. (2000). X -means: Extending K -means with efficient estimation of the number of clusters, *Proceedings of of the 17th International Conference on Machine Learning* .
- 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.