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

Thomas Laloë, Rémi Servien

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

Thomas Laloë
Université de Nice Sophia-Antipolis,
Nice, France

Rémi Servien
UMR Toxalim, INRA Toulouse,
Toulouse, France

Abstract

Using quantization techniques, Laloë (2010) defined a new clustering algorithm called Alter. This L^1-based algorithm is proved to be convergent, but suffers two major flaws. The number of clusters K has to be supplied by the user and the computational cost is high. In this article, we adapt the X-means algorithm [Pelleg and Moore, 2000] to solve both problems.

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

AMS Subject Classification : 62H30 ; 68T10.

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 [1990]. The origin of clustering goes back to 45 years ago, when some biologists and sociologists began to search 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 [Hartigan and Wong, 1979, MacQueen, 1967]. Its attractiveness lies in its symplicity and its fast execution. It has however two main drawbacks. On the one hand, the number of clusters K has to be supplied by the user. Thus, different ways to determine K have been studied in the litterature [Li et al, 2008, Pham et al, 2005]. On the other hand, 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. After each run of 2-means, local decisions are done whether subsets of the current centroid should be splitted or not. The splitting decision is done by computing the Bayesian Information Criterion (BIC). In a different approach, Laloë [2010] 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 algorithms. The complexity of the Alter algorithm decreases and an automatic selection of the number of clusters simultaneously performed. Moreover, the convergence properties of the Alter algorithm will overcome the local optimality problem of the X-means algorithm, inherited from the K-means one.

The paper is organized as follows: in the first section the different algorithms are presented. Performances of X-Alter, X-means and other methods are compared in the second section.

Methodology

The Alter algorithm

Let us detail the Alter algorithm. All the theoretical results presented in this section come from Laloë [2010]. The method is based on quantization. It is a commonly used technique in signal compression [Graf and Luschgy, 2000, Linder, 2002].
Consider $(\mathcal{H}, \|.\|)$ a normed space. We let X be a $\mathcal{H}$-valued random variable with distribution μ such as $\mathbb{E}\|X\| < \infty$.

Given a set $\mathcal{C}$ of points in $\mathcal{H}^k$, any Borel function $q : \mathcal{H} \to \mathcal{C}$ is called a quantizer. The set $\mathcal{C}$ is called a codebook, and the error made by replacing X by $q(X)$ is measured by the distortion:

$$D(\mu, q) = E\, d(X, q(X)) = \int_{\mathcal{H}} \|x - q(x)\| \mu(dx).$$

Note that $D(\mu, q) < \infty$ since $\mathbb{E}\|X\| < \infty$. For a given k, the aim is to minimize $D(\mu, .)$ among the set $\mathcal{Q}_k$ of all possible k-quantizers. The optimal distortion is then defined by

$$D_k^*(\mu) = \inf_{q \in \mathcal{Q}_k} D(\mu, q).$$

When it exists, a quantizer q^* satisfying $D(\mu, q^*) = D_k^*(\mu)$ is said to be an optimal quantizer.

From Laloë [2010] we know that we can consider only nearest neighbor quantizers. That is a quantizer q will be characterized by its codebook $\mathcal{C} = \{y_i\}_{i=1}^k$ and the rule

$$q(x) = y_i \iff \forall 1 \leq j \leq k, j \neq i, \|x - y_i\| \leq \|x - y_j\|.$$

Thus, a quantizer can be defined by its codebook only. Moreover the aim is to minimize the distortion among all possible nearest neighbor 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 the L^1-based distortion to obtain more robust estimators (see Kemperman [1987] for a discussion on this fact). Then, clustering is done by regrouping the observations that have the same image by q. More precisely, we define a cluster $\mathcal{C}$ by $\mathcal{C} = \{X_i : q(X_i) = \hat{x}_{\mathcal{C}}\}$, $\hat{x}_{\mathcal{C}}$ being the representant of cluster $\mathcal{C}$.

Theoritical results of consistency and rate of convergence have been proved in Laloë [2010]. In particular, it is stated that the rate of convergence is closely related to the metric entropy. However, the minimization of the empirical distortion is not possible in practice and an alternative proposed in Laloë [2010] is to perform the Alter algorithm.
The idea is to select an optimal codebook among the data set. More precisely the outline of the algoritm is:

1. List all possible codebooks , i.e., all possible K-tuples of data;
2. Compute the empirical distortion associated to the first codebook. Each observation X_i is associated with its closed center;
3. For each successive codebook, compute 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 that 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 the theoretical method described 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.

The X-Means algorithm
In a different approach, Pelleg and Moore [2000] define the X-means algorithm which is adapted from K-means one. It goes into action after each run of K-means, making local decisions about which subset of the current centers should split themselves in order to better fit the data. The splitting decision is done by computing the BIC criterion. This new approach proposes an efficient solution to one major drawbacks of K-means : the search of the number of clusters K. Moreover, X-means has a low computational

cost. But results suffer from the non-convergence property of the K-means algorithm. The outline of this algorithm is :

1. Perform 2-means. This gives us clustering C;
2. Evaluate the relevance of the classification C with a BIC Criterion;
3. Iterate step one and two in each cell of C. Keep going until there is no more relevant discrimination.

The X-Alter Algorithm

Following the idea of X-means, a recursive use of Alter with $K = 2$ can simultaneously allow us to combine both advantages of these two methods : estimation of K/low computational cost for X-means and convergence/parameter-free character for Alter. We also add an aggregation step at the end of our algorithm to prevent the creation of too many clusters.

Note that no parameter is needed by the algorithm. Though, the user can specify a range in which the true K reasonably lies if he wishes to (this is $[2, +\infty[$ if one had no information).
More precisely, the outline of the algorithm is the following:

1. Perform Alter with $K = 2$. This gives us clustering C;
2. Evaluate the relevance of the classification C (Figure 1) with a BIC Criterion;
3. Iterate step one and two in each cell of C (Figure 2). Keep going until there is no more relevant discrimination (Figure 3);
4. Final step of aggregation: aggregation can be considered if $BIC(K = 1) > BIC(K = 2)$. The aggregations are successively made according to the decreasing values of $BIC(K = 1) - BIC(K = 2)$ (Figure 4).

The algorithm starts by performing Alter with $K = 2$ centers. At this point, a model selection criterion (BIC, detailed below) is performed on all the data set. Using this criterion, we check the suitability of the discrimination by comparing $BIC(K = 1)$ and $BIC(K = 2)$. In another way, the criterion asks if the model with the two clusters is better than the one with only one. If the answer is yes, the iterative procedure occurs in the two subsets.

The structure improvement operation begins by splitting each cluster into two subsets. The procedure is local on that the children are fighting each other for the points in the parent's region, no others. When the discrimination is not validated by BIC criterion, the algorithm ends in this region. Up to there, the only difference with X-means is that we use Alter instead of 2-means because the consistent property of Alter must improve results. Finally, when all regions are asleep and no more clusters are needed, the aggregative step starts to prevent the creation of too many clusters or the presence of splitted clusters (as in Figure 2).

The complexity of this algorithm in the worst case scenario (that is when it creates n clusters with one data set) is $O(n^4)$, which is less than the inital Alter algorithm. However, the computational cost is still higher than for X-means. For several thousand points, this complexity is not an important practical concern. But, if the database exceeds several tens of thousand points, it could still be too high.

The BIC criterion
We use here the same criterion than Pelleg and Moore Pelleg and Moore [2000], that is the formula from Kass and Wasserman [1995]. 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. We will see in the empirical study that it performs well on real data.

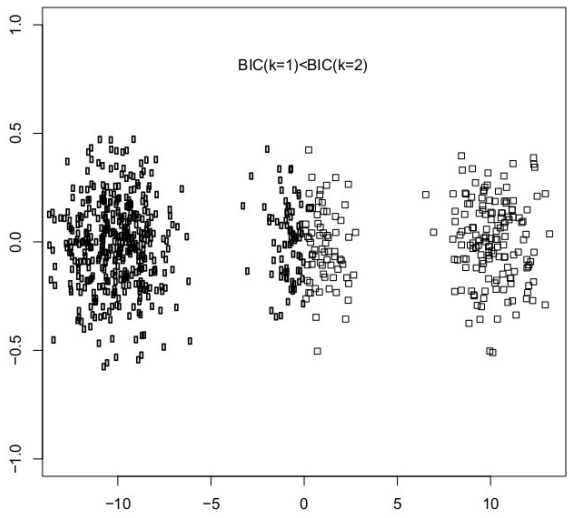

Figure 1: First iteration of X-Alter. The discrimination in 2 clusters (Step 1) is validated by BIC criterion (Step 2). In each cluster, observations are represented by a different symbol.

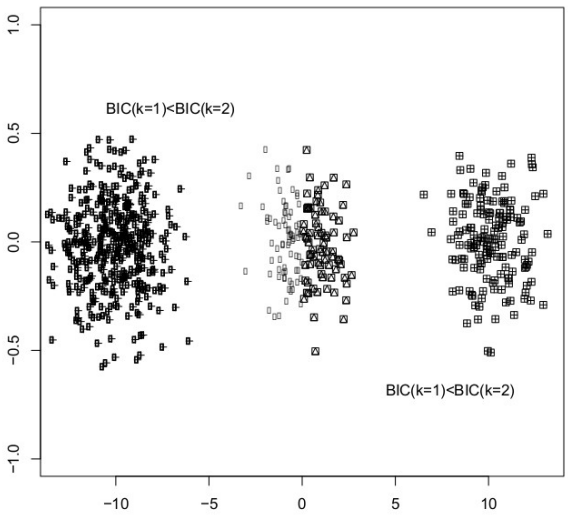

Figure 2: Second iteration of X-Alter: the sub-classification is done in the two relevant clusters (Step 1). Sub-classifications are validated by BIC (Step 2) so we obtain four clusters.

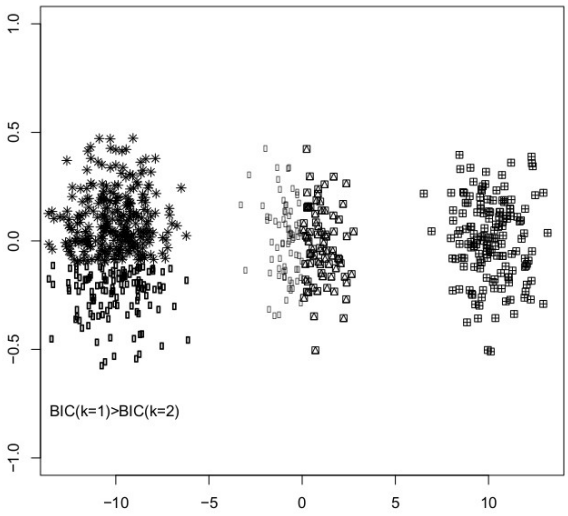

Figure 3: No relevant sub-classification in the left cluster according to BIC. In the three other clusters, we obtain the same rejection of sub-classification (Step 3).

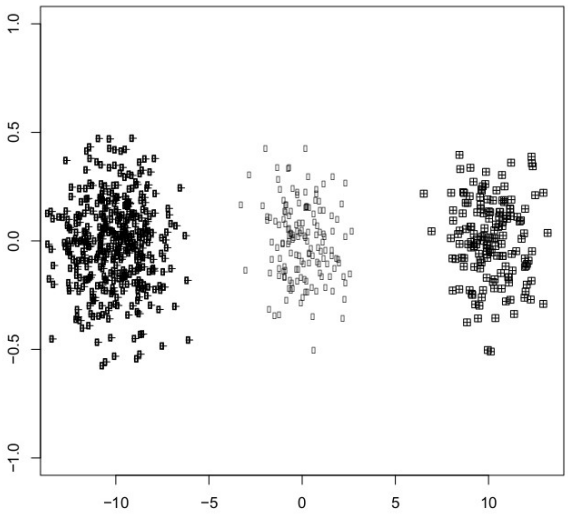

Figure 4: Final discrimination. The two middle clusters have been aggregated in Step 4.

Empirical results

In this section, an empirical study is performed to show 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 detected clusters, the Adjusted Rand Index (A.R.I.) [Rand, 1971, Hubert and Arabie, 1985] and the Dunn index [Dunn, 1974, Handl et al, 2005]. The Rand Index is a measure of the similarity between two clusters. A problem with the Rand index is that the expected value of the Rand index of two random partitions does not take a constant value (say zero). So, Hubert and Arabie [1985] defined the A.R.I. which is the corrected-for-chance version of the Rand index. Studies have shown the need and usefulness of the adjusted measures [Nguyen et al, 2009]. 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. The goal is to identify sets of clusters that are compact, with a small variance between individuals in the same cluster, and well separated, where the centers of different clusters are sufficiently far apart, as compared to the within cluster variance. Higher is the Dunn Index, better the clustering is. For more details on this classical cluster validation indexes we refer the reader to Dunn [1974] or Handl et al [2005].

Pelleg and Moore show that X-means 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. That is we obtain a clustering using X-means and we compute the aggregation procedure (Step 4 on Section) on this clustering. It allows us to assess the usefulness and the computational time of the aggregation step.

Simulated data

A simple case

We simulate here clusters of gaussian vectors 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 25 vectors (in $\mathbb{R}^{20}$) with $\mu_1 = -\mu_2 = 15$ and $\sigma_1^2 = \sigma_2^2 = 100$. That is the covariance matrices are given by $\Sigma^2 = 100 I_{20}$ where I_{20} is the identity $20 * 20$ matrix and the mean vectors by

$$\mathcal{M}_1 = -\mathcal{M}_2 = 15 \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.$$

So, we have $X_1, \ldots, X_{25} \sim \mathcal{N}(M_1, \Sigma)$ and $X_{25}, \ldots, X_{50} \sim \mathcal{N}(M_2, \Sigma)$. The results are averaged on 300 simulations.

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

Algorithm	% of correct 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

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 will often cut the middle cluster in its first iteration. More precisely we simulate two clusters of 20 vectors (in $\mathbb{R}^5$) with $\mu_1 = -\mu_2 = 20$ and $\sigma_1^2 = \sigma_2^2 = 100$; and one cluster of 20 vectors with $\mu_3 = 0$ and $\sigma_3^2 = 100$. The results are averaged on 300 simulations and gathered in Table 2.

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

Algorithm	% of correct 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

We see here the influence of the aggregation step, 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 inherited convergence property of Alter clearly improves the result.

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.

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

Algorithm	% of correct 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

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

Functional case

Now we want to consider functional data. Here, we must also compare computing times. When the dimension is smaller (as in the previous examples), these CPU time were sensibly the same. We consider two configurations:

First, we take functions $\sqrt{x}+\cos(10x+\pi/2-10)/5$, $x+\cos(10x+\pi/2-10)/5$ and $x^2+\cos(10x+\pi/2-10)/5$ in $[0,1]$ discretized 20 times. The term $\cos(10x+\pi/2-10)/5$ is added to disturb functions $\sqrt{x}$, x and x^2. 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 of the functions that we want to classify. 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).

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

Algorithm	% of correct 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

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

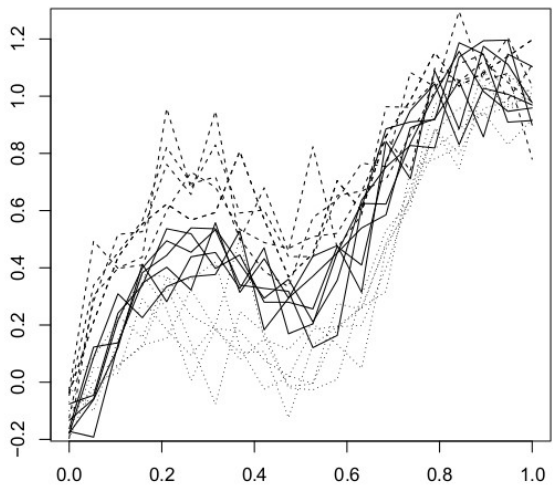

Figure 5: Example of functions. Functions based on $\sqrt{x}$ are on dashed lines, ones based on x are on solid lines and ones based on x^2 are on dotted lines.

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. Results are gathered in Table 5.

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

Algorithm	% of correct 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

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

Robustness study

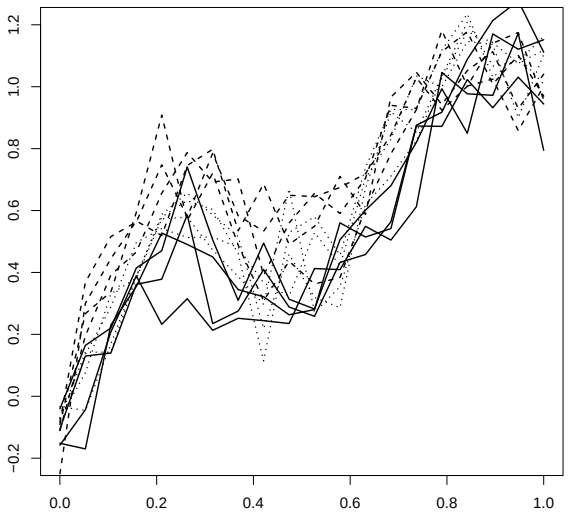

Figure 6: Example of functions. Functions based on $\sqrt{x}$ are on dashed lines, ones based on x are on solid lines and ones based on $x^{3/4}$ are on dotted lines.

In this paragraph, we illustrate the robustness properties of the L_1 distance. We consider as a starting point the first functionnal configuration above used in Figure 5. To obtain noisy 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.

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

Algorithm	% of correct 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

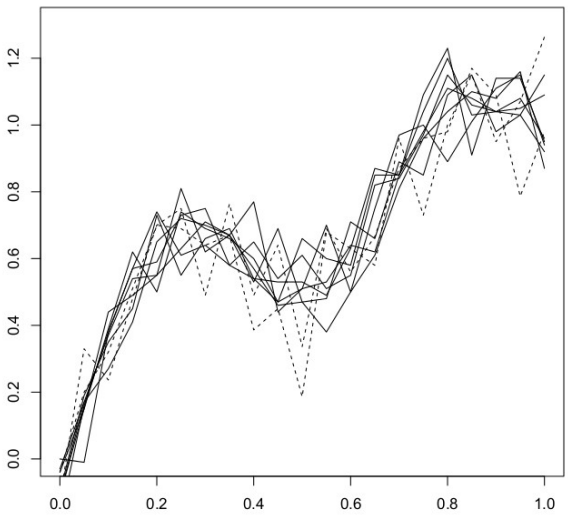

Figure 7: Example of the results of the pertubation of $\sqrt{x} + \cos(10x + \pi/2 - 10)/5$. Affected functions are on dashed lines.

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 correct number of clusters 95% of the time while X-means and X-means-R do not (a loss of respectively 4% and 6%).

Real data

In this section, we confront our method to two conventional data sets from the UCI Machine Learning Repository [Frank and Asuncion, 2010]: the wine and iris ones. In this case, we do not know if the spherical gaussian assumption of the BIC criterion is verified. So it is an important test to make sure that this hypothesis is reasonable. 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). So, 3-means have a significant advantage over others methods by knowing the number of clusters. In these two real 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.

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.

Table 7: Results for the wine data set.

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	-	0.76 (var=0.03)	0.163 (var=0.0002)
X-Alter	3	0.76	0.142

We can see that our method retrieves the real number of clusters, and that we get the same adjusted rand index than 3-means and slighty less than the 2 others. On the other hand, we do not have a good Dunn Index because one extreme instance is bad classified. We can also compare X-Alter to other methods used on this data set and listed on the UCI Machine Learning [Frank and Asuncion, 2010]. For example, we better estimate the number of clusters than Dy and Brodley [2004] with their different methods.

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 two; the latter are not linearly separable from each other which makes it more difficult to classify. The results are gathered in Table 8.

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 can not be improved. Indeed, as we consider here the Adjusted Rand Index (and not

Table 8: Results for Iris data set.

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	-	0.46 (var=0.0036)	0.04 (var=0)
X-Alter	6	1	0.402

the Rand Index), it does not mean that our classification is perfect. However the high value of the A.R.I. informs us that the great majority of iris plant are well-classified, the 3 additional clusters are in fact very small and do not affect the A.R.I and the global quality of the obtained clustering. In Dy and Brodley [2004], the estimation of the number of clusters is slighty better but, as discussed above, the quality of our clustering seems (as we don't use the same criterions) to be better. 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 acceptable and that X-Alter can be used with every data set.

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 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.

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 : 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 decreases (with respect to the Alter algorithm) it is still too important for the method to be applied on really big data sets. A possible way to overcome this problem could be the utilisation of Alter-Fast algorithm [Laloë, 2010] instead of Alter.

Alter-Fast runs several times Alter in several randomly chosen partitions of the data set. It can help to save computational time but lose efficiency. 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

Dunn J (1974) Well separated clusters and fuzzy partitions. Journal on Cybernetics 4:95–104

Dy J, Brodley C (2004) Feature selection for unsupervised learning. Journal of Machine Learning Research 5:845–889

Frank A, Asuncion A (2010) UCI machine learning repository. http://archive.ics.uci.edu/ml

Graf S, Luschgy H (2000) Foundations of Quantization for Probability Distributions, Lecture Notes in Mathematics, vol 1730. Springer-Verlag, Berlin

Handl J, Knowles K, Kell D (2005) Computational cluster validation in post-genomic data analysis. Bioinformatics 21:3201–3212

Hartigan J, Wong M (1979) A k-means clustering algorithm. Journal of the Royal Statistical Society 28:100–108

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

Kass R, 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, Rousseeuw P (1990) Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, New-York

Kemperman JHB (1987) The median of a finite measure on a Banach space. In: Statistical Data Analysis Based on the L_1-norm and Related Methods (Neuchâtel, 1987), North-Holland, Amsterdam, pp 217–230

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

Li M, Ng M, Cheung YM, 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. In: Principles of Nonparametric Learning (Udine, 2001), CISM Courses and Lectures, vol 434, Springer, Vienna, pp 163–210

MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, pp 281–297

Nguyen X, Epps J, Bailey J (2009) Information theoretic measures for clustering comparison: Is a correction for chance necessary ? ICML'09: Proceedings of the 26th Annual International Conference on Machine Learning pp 1073–1080

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

Pham T, Dimov S, 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

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