Incrementally Assessing Cluster Tendencies with a Maximum Variance Cluster Algorithm

Krzysztof Rządca[1] and Francesc J. Ferri[2,*]

[1] Institute of Computer Science. Warsaw University of Technology, Poland
krzadca@elka.pw.edu.pl
[2] Dept. Informàtica. Universitat de València. 46100 Burjassot, Spain
ferri@uv.es (* Contacting author)

Abstract. A straightforward and efficient way to discover clustering tendencies in data using a recently proposed Maximum Variance Clustering algorithm is proposed. The approach shares the benefits of the plain clustering algorithm with regard to other approaches for clustering. Experiments using both synthetic and real data have been performed in order to evaluate the differences between the proposed methodology and the plain use of the Maximum Variance algorithm. According to the results obtained, the proposal constitutes an efficient and accurate alternative.

1 Introduction

Clustering can be defined as the task of partitioning a given data set into groups based on *similarity*. Intuitively, members of each group should be more similar to each other than to the members of other groups. It is possible to view clustering as assigning labels to (unlabeled) data. Clustering is very important in a number of domains as document or text categorization, perceptual grouping, image segmentation and other applications in which is not possible or very difficult to assign appropriate labels to each object.

There is a variety of clustering algorithms and families [4]. On one hand, *hierarchical* approaches produce a hierarchy of possible clusters at each stage. On the other hand, *partitional* approaches usually deliver only one solution based on a certain criterion. In terms of the criterion used and the kind of representation used, clustering algorithms can be divided into *square error algorithms*, *graph theoretic*, *mixture resolving*, *mode seeking* and *nearest neighbors*. Additionally, the same search space can be scanned in a number of ways (deterministic, stochastic, using genetic algorithms, simulated annealing, neural networks etc.). Finally, the algorithms can be classified as *hard/crisp* or *fuzzy* according to the way the membership of objects to clusters is dealt with [4].

More formally, let $X = \{x_1, x_2, \ldots, x_N\}$ be a set of $N = |X|$ data points in a p-dimensional space. Clustering consists of finding the set of clusters $C = \{C_1, C_2, \ldots, C_M\}$ which minimizes a given criterion with given X and, usually but not necessarily, given M.

One of the simplest and most used methods to measure the quality of clustering is the square-error criterion:

Work partially supported by Spanish CICYT TIC2000-1703-C03-03.

$$J_e = \frac{\sum_{i=1}^{M} H(C_i)}{N} \tag{1}$$

where

$$H(Y) = \sum_{x \in Y} dist(x, \mu(Y))$$

is the cluster error ($dist$ is a distance measure function, e.g. Euclidean distance) and $\mu(Y) = \frac{1}{|Y|} \sum_{x \in Y} x$ is the cluster mean.

The straight minimization of equation 1 produces a trivial clustering where each data member is in its own cluster. Consequently, some constraints should be used in order to obtain meaningful results as in the (well-known) case of the k-means algorithm [5] in which the number of clusters, M, is fixed as a constraint. There are a number of algorithms [4,2] that share this feature with the k-means and all of them suffer from a common drawback: the difficulty of determining in advance the number of clusters. Most of the algorithms require trying different number of clusters and take a further stage to validate or assess which is the best result. The fact that the criterion used at each step cannot be used for validation makes the problem difficult [3,6].

2 Maximum Variance Cluster Algorithm

A straightforward clustering algorithm using a constraint based on variances of each cluster has been recently proposed [7]. This approach has a number of advantages. First, knowing cluster variances can be easier than the final number of clusters in some applications. Secondly, the same criterion can be used for the cluster validation. Additionally, as the number of clusters is modified, the algorithm seems to deal with outliers in a more natural way.

The so-called Maximum Variance Cluster (MVC) algorithm [7] requires that the variance of the union of any two clusters be greater than a given limit, σ^2_{max}:

$$\forall C_i, C_j, i \neq j : Var(C_i \cup C_j) \geq \sigma^2_{max} \tag{2}$$

where $Var(Y) = \frac{H(Y)}{|Y|}$. Clusters produced with such a constraint generally (but not necessarily) have variances below σ^2_{max}.

The way in which such a result is searched for consists of a stochastic optimization procedure in which the square error criterion in (1) is minimized (thus minimizing distances from the cluster centroids to cluster points) while holding the constraint on the cluster variance in (2). At each step, the algorithm moves points between neighboring clusters. In order to do this in an efficient way, the concepts of *inner* and *outer* borders of a cluster are introduced.

For a given point x, the qth order inner border, G_x, is a set of q furthest points belonging to the same cluster. The kth order outer border, F_x, is a set of k nearest points belonging to other clusters. The qth order inner border and kth order outer border of a cluster C_a can then be defined as the union of inner (outer) borders of all points in C_a,

$$I_a = \bigcup_{x \in C_a} G_x \ \text{ and } \ B_a = \bigcup_{x \in C_a} F_x$$

respectively. Borders defined in a such a way grow when clusters grow and the algorithm never ends up with empty borders.

The MVC algorithm starts with a cluster per data point and then repeats iterations in which the inner and outer borders of each cluster are the candidates to be moved from and to other clusters. To speed up the algorithm, only random subsets of sizes $i_a < |I_a|$ and $b_a < |B_a|$ are considered instead of the whole inner and outer borders, respectively. In particular, one of the three following operators is applied to each cluster (taken in random order) at each iteration:

- *isolation*: if the variance of the current cluster is higher than the predefined maximum, σ^2_{max}, the cluster is divided by isolating (in a new cluster) the furthest point (with regard to the cluster mean) among the i_a taken from the inner border.
- *union*: if the variance constraint is satisfied, the algorithm checks if the cluster can unite with one of the neighboring clusters which are found by looking at the b_a points taken from the outer border. Cluster union is performed only if the joint variance is lower than σ^2_{max}.
- *perturbation*: if none of the previous operators can be applied, the algorithm identifies the best candidate among the b_a taken from the outer border to be added to the cluster in terms of the gain this produces in the criterion J_e. The candidate is added to the cluster if the gain is positive. Otherwise, there is a small probability P_d (occasional defect) of adding the candidate regardless of the gain produced.

The algorithm in this form does not necessarily converge and a limited number of iterations E_{max} needs to be established in order to get a convenient result. After E_{max} iterations, isolation is no longer allowed and the probability of a random perturbation is set to 0. The clustering is considered as a final result when there is no change in the cluster arrangement for a certain number of iterations.

3 Cluster Tendency Assessing using Maximum Variance

The cluster tendency helps finding the appropriate values of the maximum variance parameter σ^2_{max} in (2). To explore all the possible values for σ^2_{max}, one possibility is to construct curves [7] showing the mean square error as a function of the maximum variance. *Plateaus* in this curve can be defined as the regions where the square error does not change while the maximum variance increases. The *strength* of the plateau ranging from σ^2_A to σ^2_B is defined as the ratio between both variances, $\frac{\sigma^2_B}{\sigma^2_A}$. A plateau is considered as *significant* if its strength is roughly above 2. This heuristic comes from the fact that the average distance to the new mean when two clusters are joined has to increase about 2 times in the worst case if one starts with two *real* clusters [7]. The significant plateaus in the mean square error curve have corresponding plateaus (with the same variance values) if the number of clusters, M, is plotted as a function of σ^2_{max}.

The most important drawback of directly using MVC to discover significant plateaus is the computational burden. One has to select the starting point and step size in order to be able to compute the curve in terms of σ^2_{max}. Moreover, the accurate detection of plateaus may depend on the above extra parameters of the algorithm. At the end, the MVC algorithm needs to be run hundreds or even thousand times in order to obtain the corresponding results.

4 Incrementally Assessing the Cluster Tendency

One of the properties of MVC is that it converges very quickly. Usually after less than 10 iterations the algorithm is able to find a solution very close to the finally obtained one. This happens because the algorithm works mainly by *uniting* clusters. For every value of σ^2_{max}, it starts by joining one-point clusters into groups of about 3 elements. Then it continues uniting such groups until the variance constraint is no longer satisfied. Isolation is performed occasionally and perturbation usually concerns a very limited number of points.

This behavior suggests a new strategy to discover significant plateaus without having to run MVC for each possible value of σ^2_{max}.

Let us suppose that we have a *stable* solution (i.e. a cluster-data points assignment) obtained by running the MVC with a value σ^2_A which corresponds to the beginning of a plateau. The goal consists of directly finding the value σ^2_B which corresponds to the end of the same plateau. Let us suppose that we know the value σ^2_B and we run the MVC algorithm with it, starting with the previous cluster assignment. As a consequence, we would not obtain any new isolation (if there was any, it would have occurred with the previous value σ^2_A and the initial solution would have been unstable). Perturbation would not occur neither, because it depends only on the error criterion. The only operator which could make profit from that increase is union which directly depends on σ^2_{max}.

Consequently, we can assume that the minimum value σ^2_B which leads to changes in the cluster assignment is the minimum value required to join any 2 clusters in the assignment corresponding to σ^2_A.

To directly obtain σ^2_B once σ^2_A is given, any two neighboring clusters (in terms of their outer borders) are tentatively merged and the corresponding joint variances are computed. The smallest joint variance is taken as σ^2_B. Three different cases are then possible:

1. If the MVC algorithm with variance σ^2_B converges to a solution with exactly one cluster less, we can conclude that the previous assumptions were correct. The value σ^2_B is the starting point of a new plateau and its corresponding cluster assignment can be used without having to fully run MVC starting with singletons.
2. If the MVC algorithm with variance σ^2_B converges to a solution with more than one cluster less, this implies that the true end of the plateau is smaller than σ^2_B. In such a case, our proposal runs again the MVC algorithm with σ^2_A but using the cluster assignment obtained for σ^2_B. With very high probability, the algorithm will increase the number of clusters but with an assignment different from the one originally

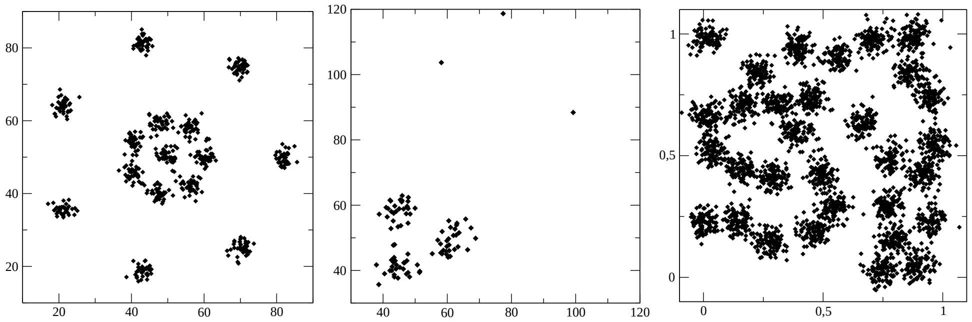
obtained with σ_A^2. This newly obtained stable solution can be used as explained above to compute the end of the sought plateau. It may happen that this produces an infinite loop if the original assignment is arrived at again. The proposed solution in this easily detectable case is to mark the whole zone as an unstable plateau and proceed from σ_B^2.

3. It is strictly possible but very unlikely that the MVC algorithm with variance σ_B^2 converges to a solution with the same (or even bigger) number of clusters. In this case, we proceed with the algorithm from this starting point but the whole zone has to be marked as unstable (in this case, even the σ_B^2 value cannot belong to any significant plateau).

The above introduced procedure which starts from a small value for σ_A^2 and proceeds by obtaining the corresponding ends of plateaus, will be referred to in this work as Incremental Maximum Variance Clustering (IMVC) algorithm. This procedure, obtains a list of variance values, $\{\sigma_i^2\}$ where some of them are marked as unstable. The algorithm always runs the original MVC algorithm with σ_i^2 starting from the cluster assignment obtained at σ_{i-1}^2. The corresponding computational burden is then certainly bounded by the cost of one run of the MVC algorithm times the number of plateaus.

5 Experiments and Results

Basically the same experiments reported in [7] using synthetic and real data have been repeated using MVC and the methodology of cluster validation proposed in this work. The parameter setting for the basic algorithm is also the same: $P_d = 0.001$, $E_{max} = 100$, $k = 3$ and $q = 1$. The number of points randomly selected from the inner and outer borders are fixed as the square root of the corresponding border sizes. The number of no change iterations needed to consider a cluster assignment as stable for the MVC algorithm is set to 10.



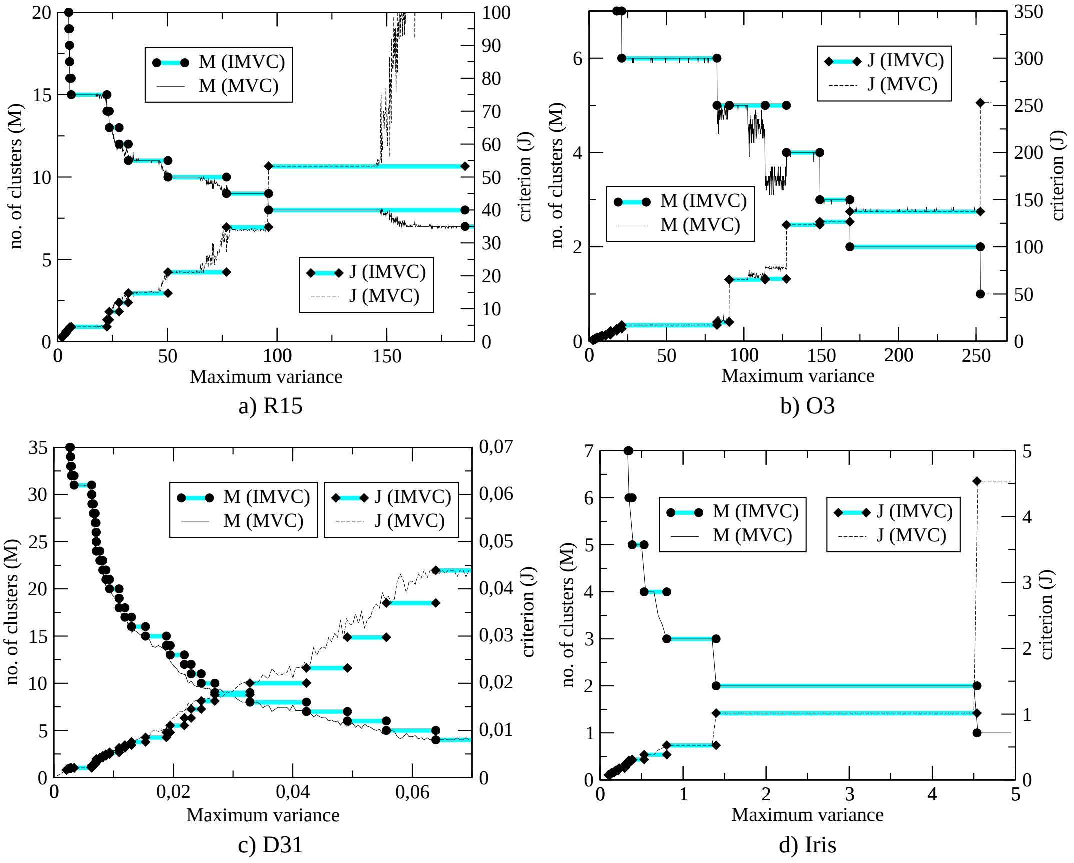
a) R15 b) O3 c) D31

Fig. 1. Scatter plot of the three synthetic data sets used in the experiments.

In particular, 3 artificial data sets (shown in Figure 1) consisting of spherically shaped bivariate Gaussian clusters have been considered. The R15 data set consists of

15 clusters of 40 points each positioned in two rings (7 each) around a central cluster. Two possible clustering results are possible: one with the 15 clusters, and the other with the 8 central clusters united in one big cluster. The O3 data set consists of 3 clusters of 30 points plus three outliers. A good solution for O3 consists of finding the three true clusters and isolate the outliers. The D31 data set consists of 31 randomly placed (non overlapping) clusters. As there are 100 points in each cluster, this can be considered as a large-scale clustering problem with regard to the previous ones.

Also the well-known Iris data set has been considered [1]. This consists of three dimensional data corresponding to three different classes of iris flowers. The goal consists of identifying these three classes in an unsupervised way.



a) R15

b) O3

c) D31

d) Iris

Fig. 2. Number of clusters, M, and criterion value, J_e, as a function of the maximum variance, σ^2_{max}, using the MVC algorithm and the incremental procedure IMVC.

The cluster tendency plots corresponding to the plain MVC and the incremental version are shown in Figure 2. In all cases the solid and dashed lines show the results obtained (number of clusters and squared error, respectively) by running the MVC using a fixed step size for the maximum variance parameter, σ^2_{max}. The algorithm has been run 10 times for each value of σ^2_{max} and the corresponding average value is plotted. Significant plateaus are identified by looking for approximately constant regions in this plots which are usually surrounded by oscillations.

The circles and diamonds show the exact values of σ^2_{max} used (once) by the IMVC algorithm. Horizontal wide grey lines represent the corresponding induced plateaus identified by the algorithm.

In the case of the R15 data set in Figure 2a, there is a significant plateau discovered by both approaches ($[6.23\ldots22.47]$) with strength 3.60 which corresponds to 15 clusters. The next plateau discovered by IMVC is located at $[96.14\ldots185.62]$ (8 clusters) with strength 1.93. In this case the plateau identified by MVC is slightly smaller but still is the second most important. In general, the plots induced by the IMVC algorithm closely follow the ones obtained directly with MVC for σ^2_{max} values below 150.

In the Figure 2b corresponding to the O3 data set, there is a significant plateau (strength 3.91) at $[21.12\ldots82.61]$ with 6 clusters discovered by both approaches. However, the plateau induced by IMVC at $[90.58\ldots113.77]$ corresponds to a region of big instabilities (switching among solutions with 5, 4 and 3 clusters) and consequently is not taken into account (This plateau is the only one marked as unstable in the presented figures). The only zone in which the plots induced by IMVC are different from the MVC plots is the above mentioned plateau. It is worth noting that besides this difference the IMVC algorithm does not identifies any significant plateau in the unstable zones.

The plots corresponding to D31 data set in Figure 2c has the most significant plateau (strength 1.87) identified by both approaches at $[0.0033\ldots0.0063]$ with 31 clusters. Apart from this, the MVC plots show a very unstable behavior and the plots induced by the IMVC differ significantly from them. From $\sigma^2_{max} = 0.02$, the IMVC produces one more cluster in average than the MVC which roughly corresponds to the standard deviation (in 10 runs) measured for the MVC curve in these regions. The IMVC results can be seen as an upper approximation (in terms of number of clusters) of the results obtained by MVC.

The Iris data set in Figure 2d gives rise to two most significant plateaus found by both approaches at $[0.80\ldots1.40]$ and $[1.40\ldots4.54]$ with strengths 1.74 and 3.25, respectively. In this case, the whole plots obtained by both approaches are very similar.

6 Concluding Remarks and Further Work

A straightforward and efficient way to discover appropriate values of the maximum variance parameter for the recently proposed MVC algorithm has been presented. One of the major benefits of this algorithm is the possibility of using it for exploratory data analysis and cluster validation. The algorithm presented constitutes an efficient and accurate alternative to the plain and exhaustive use of the MVC as proposed in [7].

We have found evidence about the ability of our proposal to quickly find the right clustering results. Only when the original algorithm exhibits severe instabilities (which means there is no real clustering result there) the approximation given by the proposed approach is not tight.

In our opinion, more experimentation is needed to properly assess the benefits of the original MVC algorithm with regard to other clustering approaches (which has been partially done in [7]) and also to fully test our approach to discover cluster tendencies in real data corresponding to challenging and nontrivial clustering problems. Nevertheless, the preliminary results obtained in this work give enough evidence to see the proposed methodology as very promising both because the good results obtained and the relatively small computational burden.

References

1. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
2. L.O. Hall, B. Ozyurt, and J.C. Bezdek. Clustering with a genetically optimized approach. *IEEE Transactions on Evolutionary Computation*, 3(2):103–112, 1999.
3. A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.
4. A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):265–323, 1999.
5. J. MacQueen. Some methods for classification and analysis of multivariate observations. In L.M. Le Cam and J. Neyman, editors, *Proc. Fifth Berkeley Symp. Math. Statistics and Probability*, volume 1, pages 281–297, 1967.
6. G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6, 1978.
7. Cor J. Veenman, Marcel J.T. Reinders, and Eric Backer. A maximum variance cluster algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1273–1280, 2002.