Journal of Classification 32:414-442 (2015)
DOI: 10.1007/s00357-015-9186-y

Bisecting K-Means and 1D Projection Divisive Clustering: A Unified Framework and Experimental Comparison

EkaterinaV. Kovaleva

National Research University Higher School of Economics, Russia

Boris G. Mirkin

National Research University Higher School of Economics, Russia
Birkbeck University of London, UK

Abstract: The paper presents a least squares framework for divisive clustering. Two popular divisive clustering methods, Bisecting K-Means and Principal Direction Division, appear to be versions of the same least squares approach. The PDD recently has been enhanced with a stopping criterion taking into account the minima of the corresponding one-dimensional density function (dePDDP method). We extend this approach to Bisecting K-Means by projecting the data onto random directions and compare thus modified methods. It appears the dePDDP method is superior at datasets with relatively small numbers of clusters, whatever cluster intermix, whereas our version of Bisecting K-Means is superior at greater cluster numbers with noise entities added to the cluster structure.

Keywords: Divisive clustering; Bisecting k-means; Split base decomposition; Up-hierarchy; Principal directions; Random directions; Computational experiment; Cluster structure generator/

This paper has been prepared in the framework of a research grant 11-04-0019 "Teacher–Students: Development and adaptation of cluster analysis methods to automate the analysis of textual information by using a domain's ontology" awarded by the Research Fund of NRU HSE for years 2011-2012. The authors thank the anonymous reviewers for their comments that helped us to reshape the paper in a much more focused style.

Authors' Addresses: E.V. Kovaleva and B. Mirkin, Department of Data Analysis and Machine Intelligence, National Research University Higher School of Economics, 20 Miasnitskaya, Moscow RF, emails: bermud.tri@gmail.com, bmirkin@hse.com; B. Mirkin, Department of Computer Science and Information Systems, Birkbeck University of London, London UK, email: mirkin@dcs.bbk.ac.uk.

Published online: 2 October 2015

1. Introduction

Divisive clustering is a powerful technique for recursively building a hierarchy of clusters in a top-to-bottom manner: starting from the entire dataset, clusters are split in two parts until a stopping condition is met. The cluster hierarchy can be used as a device for obtaining a good partition of the dataset or as a structure of its own. The latter is especially popular as a tool for building “conceptual clusters” and phylogenetic trees.

Divisive methods were introduced at the very dawn of clustering research in two forms, monothetic clustering (Sonquist, Baker, and Morgan 1973) and numerical taxonomy (Sneath and Sokal 1973). Monothetic clustering is a divisive procedure at which each cluster is split over a single feature. This was further developed, in 80s, as the so-called conceptual clustering (Michalsky and Stepp 1983; Fisher 1987), after which the approach somewhat faded. It is re-emerging as a tool for ontology developing (see, for example, Hazman, El-Beltagy, and Rafea 2011). The numerical taxonomy approach uses the concept of (dis)similarity between entities and divides each cluster in “most distant” parts. The least squares divisive clustering approach was proposed rather early, too, see Edwards and Cavalli-Sforza (1965). However, researchers did not immediately pick it up, probably because of a controversy pointed out by Gower (1967). The controversy concerns a property of the square-error clustering criterion that favors splitting clusters into evenly-sized parts over those of unbalanced sizes; this sometimes overrides the structure of distances between them (see also Mirkin 2012, p. 151).

The situation changed after the method was experimentally approved by Steinbach, Karypis and Kumar (2000) in the form of the so-called Bisecting K-Means that uses K-Means clustering approach, at K=2, as the splitting method. This method, in fact, maximizes the Ward distance between split parts’ centroids as shown by Mirkin (1996), who referred to the method as two-splitting. Simultaneously, Boley (1998) proposed the so-called Principal Direction Projection method: the method projects all the data points to the principal axis and uses this for splitting. This latter method was substantially enhanced by Tasoulis, Tasoulis and Plagianakos (2010) who proposed an effective stopping condition for the cluster division process. According to their approach, referred to as dePDDP, it is the minima of data density functions over the within-cluster principal axes that dictate the next split location; the splits stop when the density functions have no minima anymore. Divisive clustering techniques were also applied for similarity and network clustering by using the normalized cut splitting criterion and an associated spectral approach (Shi and Malik 2000) or the modularity partitioning approach by Girvan and Newman (see

Newman 2006). Currently, divisive clustering is increasingly used in unstructured text analysis, bioinformatics and network structure analysis.

The goal of this paper is to present a mathematical framework for the least-squares divisive clustering in which both the PDDP and Bisecting K-Means are but different options for minimizing the same square-error criterion. No polynomial time algorithm for finding its global optimum has been proposed so far. Therefore, suboptimal approaches are appropriate. This criterion is equivalent to the weighted squared Euclidean distance between cluster centers used in the so-called Ward method for agglomerative clustering. We propose a version of the Bisecting K-Means enhanced with a 1D density function based stopping condition, BiKM-R, as a natural counterpart to dePDDP in divisive clustering, and experimentally compare the two methods. Besides, to initialize the Bisecting K-Means, we use the Anomalous Cluster method (see, for example, Mirkin 2012) which proved successful in a number of experimental studies (Chiang and Mirkin 2010; Amorim and Mirkin 2012). To choose a cluster to split, we further utilize a greedy version of the least-squares computation leading to a computationally intensive rule: that cluster is to be split in which the found Ward distance between the split part centroids is maximum. Yet the least-squares framework suggests little to specify a stopping rule. Therefore, we extend the successful heuristic by Tasoulis, Tasoulis, and Plagianakos (2010) to the multivariate clustering case by generating a number of random axes and using a threshold for the proportion of those at which no minimum has occurred as the stopping rule. Thus specified Bisecting K-Means divisive clustering method is referred to as BiKM-R.

Then we proceed to the issue of comparing the two methods. We consider 2D datasets from Tasoulis, Tasoulis, and Plagianakos (2010) and, also, synthetic datasets generated with a specially developed Gaussian cluster structure data generator. In contrast to other synthetic data generators (see, for example, Steinley and Brusco 2007; Mirkin and Chiang 2010), this data generator allows modeling various features of cluster structure, including that of cluster sizes and intermix, by using just one parameter.

According to our experiments, the winner much depends on a combination of the number of clusters, cluster intermix and noise in data. When the number of clusters is relatively small, the dePDDP is the winner, even when clusters are well intermixed, whereas the BiKM-R method becomes superior at the intermixed noisy data while dismally failing at intermixed data with no noise. Also, we discuss some possible variations of the methods regarding the choice of the cluster to split and/or stopping condition – none of those is superior to the two methods under consideration, including a classical statistics criterion for distinguishing between a single Gaussian and mix of two Gaussians in one dimension.

The paper is organized as follows. Section 2 presents the least-squares framework for divisive clustering demonstrating a clear-cut relation between the PDDP and Bisecting K-Means methods. In section 3, we describe those specific versions of dePDDP and Bisecting K-Means that are going to be compared. The latter is specified by using two subroutines proposed by the authors: (i) Anomalous clusters for initialization of the splitting process, and (ii) Random projections for stopping the division process. Section 4 describes the setting of our experiments. Section 5 presents results of the experiments including some unexpected effects of the increasing number of clusters, noise and the cluster intermix on clustering results. Section 6 discusses possible use of some other versions of the least-squares divisive clustering. Section 7 concludes the paper.

2. Split Base Vector Framework and The Least Squares Splitting Criterion

This section presents a mathematical framework, which allows us to formulate and, thus, compare such different approaches to hierarchical clustering as Bisecting K-Means and PDDP, for divisive clustering, and Ward's method for agglomerative clustering.

Consider a set **S** of subsets S_w of an entity set I satisfying the following conditions:

(i) $I \in \mathbf{S}$;

(ii) If $S_{w1} \in \mathbf{S}$ and $S_{w2} \in \mathbf{S}$ then $S_{w1} \cap S_{w2}$ either is empty or coincides with one of S_{w1} or S_{w2};

(iii) Any non-terminal subset $S_w \in \mathbf{S}$ is equal to $S_w = S_{w1} \cup S_{w2}$ for some S_{w1}, $S_{w2} \in \mathbf{S}$. A subset $S_w \in \mathbf{S}$ is called terminal, or a leaf, if it does not contain any other subset from **S**. Sets S_{w1} and S_{w2} form split parts of S_w in the hierarchy.

Such a set **S** will be referred to as a cluster up-hierarchy and its subsets as clusters. A cluster up-hierarchy can be obtained as a result of a divisive clustering procedure and represented by a rooted binary tree whose nodes are the **S** clusters. For example, take I to consist of nine numerals, 1, 2, …, 9 and clusters in **S** forming the binary tree shown on Figure 1. Its terminal clusters correspond to the leaves in the tree.

Each non-terminal cluster S_w of an up-hierarchy **S** is split in two clusters, $S_w = S_{w1} \cup S_{w2}$, according to item (iii) in the definition. This allows

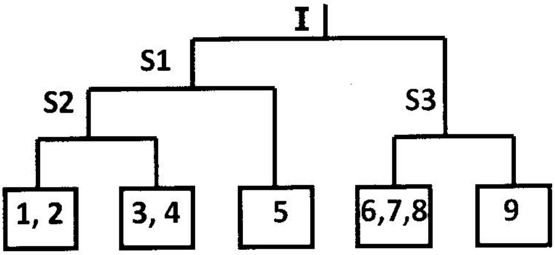

Figure 1. An up-hierarchy represented by a binary rooted tree. Its terminal clusters, {1,2}, {3,4}, {5}, {6,7,8}, and {9} correspond to leaves in the tree, whereas its non-terminal clusters S1, S2, S3 correspond to the interior nodes, and the universal cluster I, to the root.

to define an |I|-dimensional three-valued vector $\varphi_w=(\varphi_{iw})$ at which φ_{iw} =0 for all i$\notin$S$_w$ and $\varphi_{iw}=\alpha_w$ for $i\in S_{w1}$ and $\varphi_{iw}=-\beta_w$ for $i\in S_{w2}$. The positive values α_w and β_w are defined in such a way that vector φ_w is centered and normed. To satisfy these two conditions values, α_w and β_w are to be uniquely defined by formulas:

$$\alpha_w=\sqrt{\frac{N_{w2}}{N_w N_{w1}}}, \qquad \beta_w=\sqrt{\frac{N_{w1}}{N_w N_{w2}}}.$$

where N_w, N_{w1}, and N_{w2} are cardinalities of clusters S_w, S_{w1}, and S_{w2}, respectively (Mirkin 1997). This vector φ_w uniquely characterizes the split, up to the signs at α_w and β_w. Vectors φ_w are referred to as split base vectors because they are mutually orthogonal and, therefore, form an orthonormal base of a linear subspace of the set of all centered |I|-dimensional vectors (Mirkin 1997). To see how this may happen, let us take a look at the vectors corresponding to the four interior nodes, I, S1, S2, S3, of the tree on Figure 1 as displayed in Table 1.

For example, elements of column S1 are defined as follows. They are set to zero outside of cluster S1, that is, at entities 6, 7, 8, and 9. Those, corresponding to the left split part, entities 1 to 4, get the square root of 1/(4*5), that is, 0.298, and that remaining on the right, 5, gets minus square root of 4/(1*5), that is, -0.373, according to the definition. Of course, the selection of positive or negative signs is arbitrary. This corresponds to the arbitrariness in drawing children of a node, one to the left, the other to the right.

It is not difficult to see that the column vectors in Table 1 are normed. Indeed, the sum of squared components of vector φ_w is equal to $N_{w1}\alpha_w^2+N_{w2}\beta_w^2 = N_{w1}N_{w2}/(N_wN_{w1})+N_{w2}N_{w1}/(N_wN_{w2})$, that is, $N_{w1}/N_w+N_{w2}/N_w$

Table 1. Split base vectors corresponding to interior nodes in the hierarchy of Figure 1. After slash, cardinalities of the cluster and its children are displayed.

Node Leaf	I/9-5-4	S1/5-4-1	S2/4-2-2	S3/4-3-1
1	0.298	0.224	0.500	0
2	0.298	0.224	0.500	0
3	0.298	0.224	-0.500	0
4	0.298	0.224	-0.500	0
5	0.298	-0. 894	0	0
6	-0.373	0	0	0.289
7	-0.373	0	0	0.289
8	-0.373	0	0	0.289
9	-0.373	0	0	-0. 866

=1. Also, the sum of components in each split base vector is 0, that is, the vector is centered.

Now one can see why these vectors are mutually orthogonal. Those assigned to unrelated nodes, such as split base vectors corresponding to S2 and S3, are orthogonal because there is a zero at each component in one or two of them. Those assigned to related nodes, such as S1 and S2, are orthogonal because the vector corresponding to the larger of them has the same value for all the components corresponding to the smaller cluster. Say, φ_{S1} has the same value, 0.224, assigned to all non-zero components of φ_{S2}. Therefore, the inner product of φ_{S1} and φ_{S2} is zero because φ_{S2} is centered so that its first 4 components sum to 0, whereas its other components are zeros leading to 0 in the inner product.

Since the interior nodes of an up-hierarchy correspond to splits in two parts, their number is always the number of leaves short one. Given an up-hierarchy on *I* with *K* interior nodes, an N×K matrix Φ can be defined so that its columns are split base vectors φ_w defined above. Since the columns are normed and mutually orthogonal, equation $\Phi^T\Phi=U_K$ holds, where U_K is *K*-dimensional identity matrix, and Φ^T the transpose of Φ. Given an $N\times V$ data matrix $Y=(y_{iv})$, it can be decomposed over base matrix Φ, according to equation

$$Y=\Phi A+E\,, \tag{1}$$

or, in the element notation,

$$y_{iv} = \sum_{w=1}^{K} \varphi_{iw} a_{wv} + e_{iv}, \qquad (1')$$

where $A=(a_{wv})$ is a $K\times V$ "loading" matrix and $E=(e_{iv})$ the matrix of residuals. Of course, columns in matrix Y are centered because all the columns in Φ are.

To fit model (1), at a givenΦ, let us use the least-squares approach so that the problem is to find a matrix A minimizing the sum of squared residuals

$$L = \sum_{i=1}^{N} \sum_{v=1}^{V} e_{iv}^{\ 2} = tr[(Y - \Phi A)^T (Y - \Phi A)], \qquad (2)$$

where $tr(C)$ is the sum of diagonal entries in a square matrix C.
Let us use the first-order optimality condition. First of all, transform L to

$$L=tr(Y^TY)-2tr(Y^T\Phi A)+tr(A^TA)\ ,$$

which is true because $tr(Y^T\Phi A)= tr(A^T\Phi^TY)$ and $\Phi^T\Phi=U_K$. Therefore, according to the rules of matrix differentiation, the derivative of L over A is equal to

$$\frac{\partial L}{\partial C} = 0 - 2Y^T\Phi + 2A\,.$$

Equating this to 0, one obtains an equation determining the optimal A:

$$A^T= Y^T\Phi. \qquad (3)$$

Therefore, the minimum value of L must be equal to

$$L=tr(Y^TY)-tr(A^TA)\ , \qquad (4)$$

because of (3) put into the formula for L above.

The entries of the "loading'' matrix A can be expressed from (3) as

$$a_{wv} = \sum_{i\in I} \varphi_{iw} y_{iv} = \sum_{i\in S_{w1}} \sqrt{\frac{N_{w2}}{N_w N_{w1}}} y_{iv} - \sum_{i\in S_{w2}} \sqrt{\frac{N_{w1}}{N_w N_{w2}}} y_{iv}$$

$$= \sqrt{\frac{N_{w1}N_{w2}}{N_w}} \big(\sum_{i\in S_{w1}} y_{iv}/N_{w1} - \sum_{i\in S_{w2}} y_{iv}/N_{w2}\big) \ .$$

where S_{w1} and S_{w2} are split parts of w-th cluster in the hierarchy ($w=1,\ldots, K$), N_{w1} and N_{w2} are their respective cardinalities and $N_w = N_{w1}+N_{w2}$ the cardinality of the parental w-th cluster. Let us denote the gravity centers of clusters S_{w1} and S_{w2} by c_{w1} and c_{w2}, respectively. Then the formula above means that

$$a_{wv} = \sqrt{\frac{N_{w1}N_{w2}}{N_w}}(c_{w1v} - c_{w2v})\,. \qquad (5)$$

Therefore, the squared norm of vector $a_w = (a_{wv})$ is

$$\mu_w{}^2 = < a_w, a_w > = \frac{N_{w1}N_{w2}}{N_w}\sum_{v=1}^{V}(c_{w1v} - c_{w2v})^2 = \frac{N_{w1}N_{w2}}{N_w} d^2(c_{w1}, c_{w2})\,, \quad (6)$$

where d is Euclidean distance. This expression is not unknown in clustering: it is Ward's distance between clusters S_{w1} and S_{w2} expressing the increase in the value of the square error K-Means criterion when these clusters are merged together (Ward 1963). Or, equivalently, this is the decrease of the value of the square error clustering criterion when w-th cluster is split in these parts (Mirkin 2011).

Since $tr(A^TA)$ is but the sum of squared entries of A, that is, the sum of all $<a_w,a_w>$, the expression (4) can be reformulated as follows:

$$L = \sum_{i=1}^{N}\sum_{v=1}^{V} y_{iv}^2 - \sum_{w=1}^{K}\mu_w^2 = \sum_{i=1}^{N}\sum_{v=1}^{V} y_{iv}^2 - \sum_{w=1}^{K}\frac{N_{w1}N_{w2}}{N_w} d^2(c_{w1}, c_{w2})\,. \quad (7)$$

This sets the least-squares criterion for building an up-hierarchy. The criterion is to maximize the right hand item in (7), the total contribution of the hierarchy to the data scatter, the sum of all K $\mu_w{}^2$ values $M_K = \Sigma_w \mu_w{}^2$ ($w=1,2,\dots, K$):

$$M_K = \sum_{w=1}^{K}\frac{N_{w1}N_{w2}}{N_w} d^2(c_{w1}, c_{w2}). \qquad (8)$$

The authors are not aware of any research on this optimization problem. Of course, one may propose to try first a relaxed problem by finding first K singular vectors of matrix Y, but this strategy has never been applied to divisive clustering either, except in the case of the input data being a square similarity data matrix, although with mixed results (Ng, Jordan and Weiss 2001).

A practical way to go regarding the maximization of M_K in (8) is to use a greedy-wise heuristic by doing one split w at a time. The setting is as follows. At each step consider the model and least squares algorithm for K=1 only:

$$y_{iv} = \varphi_{iw} a_{wv} + e_{iv}\,. \qquad (9)$$

As $\varphi_{iw}=0$ for $i \in I$ out of w-th cluster, this equation makes sense only within the cluster so that values y_{iv} are centered within the cluster because the split base vector φ_w is centered.

The heuristic can be formulated as a divisive clustering method that recursively splits a cluster in two parts starting from the universal cluster consisting of all the entities. A cluster to be split is selected according to a pre-specified selection rule. The splitting process does not apply if a pre-specified stopping rule holds at the selected cluster. The splitting process is governed by the least-squares criterion applied to model (9).

The model requires to find a split, or equivalently, a split base vector φ_w, and a corresponding vector a_w that minimize the sum of the squared residuals e_{iv}^2. According to equation (3), or (5), vector a_w is determined by the vector φ_w as $a_w = Y^T\varphi_w$.Therefore, the problem would be to find such a split that the Ward's distance μ_w^2 (6) between its centroids is maximized. This is equivalent to applying the square-error criterion of K-Means clustering at K=2 (Mirkin 1996) and, therefore, can be done by using an alternating maximization/minimization procedure of K-Means. Independently, the usage of K-Means in divisive clustering, as just a heuristic under the name of Bisecting K-Means, has been advocated by Steinbach, Karypis, and Kumar (2000).

Another way of suboptimal minimization of (2) according to model (9) would be through relaxation of the condition that φ_w must be a split base vector to find an optimal arbitrary vector z minimizing the least squares criterion for (9) and then approximate it with an appropriate split base vector. Let us denote the cluster to be split by S and drop the index at φ_w. To minimize the square error criterion

$$D = \sum_{i\in S_w}\sum_{v=1}^{M}(y_{iv} - \varphi_i a_v)^2$$

with respect to arbitrary φ_i and a_v, one should apply the first order optimality conditions $\partial D/\partial\varphi_i = 0,\ \partial D/\partial a_v = 0$, leading to equations

$$\sum_{v=1}^{V} y_{iv}a_v = \varphi_i \sum_{v=1}^{V} a_v^{\ 2}, \qquad \sum_{i\in S_w} y_{iv}\varphi_i = a_v \sum_{v=1}^{V} \varphi_i^{\ 2}.$$

Note that $\sum_{v=1}^{V} a_v^{\ 2} = \| a \|^2$ and $\sum_{i\in S_w} \varphi_i^{\ 2} = \| \varphi \|^2$ are squared norms of vectors a and φ, respectively. Denote λ=||a||||φ||, then the equations above can be rewritten as matrix equations

$$Y_w a = \lambda\varphi, \qquad Y_w^{\ T}\varphi = \lambda a, \tag{10}$$

where Y_w is part of Y restricted to $i\in S_w$, a and φ are normed, and λ is arbitrary satisfying equation $D=Tr(Y_w^T Y_w)-\lambda^2$. This latter equation shows that the minimum of D is reached at the maximum λ. This proves that the solution to the relaxed problem is the first singular triplet of matrix Y_w, that is, optimal a and φ satisfy (10) at the maximum possible λ the singular value. Because Y_w has been centered, the solution coincides with the principal component; the optimal φ is frequently referred to as the principal direction.

Therefore, to pursue the way of relaxing the problem constraints, one arrives at the need to make a split over the principal direction. Boley (1998) proposed such a method, based on heuristic considerations, under the name of Principal Direction Divisive Partitioning (PDDP). In the original PDDP, the splits are made over the sign: entities i with positive φ_i go to one split part and those with negative φ_i go to the other one. Savaresi and Boley (2004) tried to address the issue of why PDDP and Bisecting K-Means divisive clustering approaches lead to similar results. What is said above suggests a simple answer: because both of them realize the same greedy-wise procedure for fitting a bilinear clustering model. They differ just by the way they suboptimize the same criterion: Bisecting K-Means approximates the solution directly, whereas PDDP by first relaxing the constraint and imposing it afterwards. These two basic splitting options will be considered in a greater detail in the next section.

3. Two Algorithms for Divisive Least Squares Clustering

3.1. dePDDP Algorithm

In the Principal Direction approach, the original splitting rule (Boley 1998) is rather simplistic – split the projection axis in the positive and negative parts. Recently, PDDP has been updated by a rule taking into account the distribution of the data. Tasoulis, Tasoulis, and Plagianakos (2010) use the shape of a density function estimated from the sample over the principal direction: their rule states that the split should be made over the deepest minimum of the estimated density function. What is nice in this is that the same rule can be used for the other parts of the triad: choosing the cluster to split and stopping the process. That cluster is split in which the deepest minimum is the deepest over all the leaf clusters. That cluster is not split anymore at which the density function is either monotone or concave, thus having no local minima. Tasoulis, Tasoulis, and Plagianakos (2010) demonstrated that the Principal Direction method with this rule, referred to as dePDDP, is quite effective on a number of real and synthetic datasets.

Let us add a few specifying remarks about dePDDP. Given a cluster S of data points, the data features are centered within the cluster and then the principal direction is found as the singular vector corresponding to the maximum singular value of the centered data matrix. Then all the cluster points are projected onto the axis of the principal direction and a Parzen-type estimate of the density function is found. This function is defined by formula

$$\hat{f}\left(x_j\right)=n^{-1}h^{-1}\sum_{i=1}^{N}K\left(\left(x_j-x_{n(j)}\right)/h\right),$$

where *K(x)* is the Gaussian density

$$K\left(x\right)=\left(2\pi\right)^{-1/2}e^{-0.5x^2},$$

and h>0 is the Parzen window parameter computed according to equation $h=\sigma\left(\frac{4}{3n}\right)^{1.5}$, where σ is the data standard deviation, as specified in Tasoulis, Tasoulis, and Plagianakos (2010). This function is used for both deciding whether the cluster should be left as is, undivided, or, if not, then what split should be made. A minimum of *f* is defined by a triplet of point projections such that the density value in the middle point is smaller than those at the boundary points of the triplet. The deepest minima are compared over all admissible clusters and the cluster with the deepest minimum is chosen to be split over the minimum so that all points, whose projections are less than the minimum point, go to one part, and those whose projections are greater than the minimum point go to the other part. The splitting process halts when the set of all minima is empty, that is, if the density functions of all the clusters are either concave or monotone.

3.2. A Version of Bisecting K-Means

3.2.1. K-Means Splitting Using Anomalous Clusters

The Bisecting K-Means method applies to a cluster S to be split and starts at some initial centroids, c_1 and c_2, of the two clusters to be built, and proceeds in iterations, consisting of two steps each. The first step, the clusters update, divides S in two clusters, each consisting of those entities that are nearer to either of the centroids. The second step, the centroids update, computes new centroids. The process terminates when the new centroids coincide with the previous ones.

The result much depends on the initial centroids. Usually, the random choice is recommended. In this next section, a more intuitive procedure to choose the centroids is described.

As a prequel to K-Means, Anomalous Pattern (AP) one-by-one clustering procedure has shown good results in experiments reported by Chiang and Mirkin (2010), Amorim and Mirkin (2012). To apply the AP method, the data are standardized by putting the origin into the gravity center (mean point) of *S* and normalizing to balance the feature contributions. Then an Anomalous Pattern cluster is built starting from an entity that is farthest away from the origin, as the initial centroid *c*. After that, a one-cluster version of the generic K-Means is utilized. The current AP cluster *P* is defined as the set of all those entities that are closer to *c* than to the origin, and the next centroid *c* is defined as the center of gravity of *P*. This process is iterated until convergence. The final *P*, along with its centroid *c* is the AP cluster. After it is removed from the data set, the process of extracting of AP clusters is reiterated without ever changing the origin, until no unclustered entity remains. Centroids of the two largest of AP clusters are used to initialize Bisecting K-Means. The combined splitting method is referred to as Bisecting iK-Means clustering. As the stopping criterion with Bisecting K-Means clustering, we propose a "random direction projections" procedure described in the next section.

The least squares greedy one-by-one splitting procedure above requires maximizing Ward's distance between split parts at any division. Therefore, this rule requires to test splitting at every leaf cluster after which only that with the maximum Ward's distance between split parts should be actually performed.

3.2.2. Random Direction Projections

To extend the minimum of one-dimensional density function approach to Bisecting K-means framework, we propose extending the no-minima stopping rule by Tasoulis, Tasoulis, and Plagianakos (2010) to projections of the dataset onto a number of randomly generated directions, rather than just one principal direction. Given a set of clusters, S_k, the proposed procedure works as follows. First, generate a number, *s*, of random vectors p_i, $i=1,..., s$. Currently p_i is generated from the spherical Gaussian distribution with its mean at the feature space origin, 0, and $\sigma^2=1/V$ where *V* is the number of features. Then, every element *x* of each cluster S_k ($k=1,...K$) is projected onto each of the directions p_t so that its coordinate over the direction is computed as $x_t =<x, p_t >$ and the value of density function f_k^t over p_t is determined as described above ($t=1,..., s$). Then, at a given *k*, the proportion ε_k of those f_k^t that have at least one minimum is calculated. If $\varepsilon_k < \varepsilon$, where ε is a pre-specified threshold value, cluster S_k is not split at this step, yet to be tested again at the next step (option (a)) or not split anymore at all. If condition $\varepsilon_k < \varepsilon$ is true for all $k=1, . . ., K$, the process stops and the binary split tree made so far is output. This approach can be applied both to select a cluster to split and to do the splitting itself. Specifically, cluster S_k with the maximum $\varepsilon_k \geq \varepsilon$ is selected to be split and then S_k can be split over its deepest minimum. Note that the fact that projections are randomly taken after each splitting implies a new situation for the stopping condition: a cluster can satisfy the stopping condition at some random directions and not satisfy it at the next generated random directions. This leads to two different options defined above: "once not-to-be-split, next time may be split" (option (a)) and "once not-to-be split, always not-to-be split".

It should be noted that the idea of random projections is not new. In Vempala (2005), Tasoulis, Tasoulis, and Plagianakos (2013) and many others it applies to project data onto random subspaces to both reduce computations and find a subspace at which the data manifest interesting patterns. We apply this approach to neither of these but rather just as a score counter for our stopping criterion, which makes it different. For example, our random directions are independent, thus not necessarily mutually orthogonal, in contrast to random axes in Tasoulis, Tasoulis, and Plagianakos (2013).

To implement this procedure computationally, one needs to specify: (a) the number of random directions s, and (b) the threshold ε. In the former, we take as many directions as the number of variables, which seems a reasonable compromise between the thoroughness of the decision and the computational intensity. As to the latter, we take $\varepsilon=0.32$ because of the following considerations.

Let the data be a union of samples from two m-dimensional Gaussian distributions with the same covariance matrix Σ and differing centers μ_1 and μ_2. Let us draw some rationale for deriving the probability that these are confused when projected at a random direction. Assume that the centers are separated by a distance satisfying condition $\|d\| \geq c\sqrt{trace(\Sigma)}$, where $d=\mu_1 - \mu_2$. Consider a random vector $p \in N(0, I_m/m)$. For every m-dimensional x, the projection of x onto direction p is equal to the inner product $<p,x>$. The probability that the centers of projections of the clusters onto the direction p satisfy the same condition is

$$P\left(\left|\langle p,d\rangle\right| \geq c\sqrt{p^T\Sigma p}\right) = 1 - P\left(-c\sqrt{p^T\Sigma p} \leq \langle p,d\rangle \leq c\sqrt{p^T\Sigma p}\right).$$

Since components of p are independent, the inner product $<p,d>$ is a Gaussian random variable with the mean equal to 0 and the variance $||d||^2/m$, so that

$$\frac{\sqrt{m}\langle d,p\rangle}{\|d\|} \in N(0,1).$$

Also, the definition of p implies that:

$$E\left(\|p\|^2\right) = E\left(\sum_i p_i^2\right) = \sum_i Ep_i^2 = \sum_i \sigma_i^2 = \sum_i 1/m = m/m = 1,$$

$$E\left(p^T\Sigma p\right) = E\left(\sum_{i,j}\Sigma_{ij}p_ip_j\right) = \sum_{i,j}\Sigma_{ij}E\left(p_ip_j\right) = \sum_i \Sigma_{ii}E\left(p_i^2\right) = \frac{1}{m}\sum_i \Sigma_{ii} = \frac{trace(\Sigma)}{m}.$$

By substituting the variables with their expectations into the inequality, we obtain

$$P\left(\left|\left\langle p,d\right\rangle\right| \geq c\sqrt{p^T\Sigma p}\right) = 1 - P\left(-c\sqrt{p^T\Sigma p} \leq \left\langle p,d\right\rangle \leq c\sqrt{p^T\Sigma p}\right) \approx$$

$$\approx 1 - P\left(-c\sqrt{\frac{trace(\Sigma)}{m}} \leq \left\langle p,d\right\rangle \leq c\sqrt{\frac{trace(\Sigma)}{m}}\right) =$$

$$= 1 - P\left(-c\sqrt{\frac{trace(\Sigma)}{m}}\frac{\sqrt{m}}{\|d\|} \leq \left\langle p,d\right\rangle\frac{\sqrt{m}}{\|d\|} \leq c\sqrt{\frac{trace(\Sigma)}{m}}\frac{\sqrt{m}}{\|d\|}\right) =$$

$$= 1 - \left[\Phi\left(c\sqrt{\frac{trace(\Sigma)}{m}}\frac{\sqrt{m}}{\|d\|}\right) - \Phi\left(-c\sqrt{\frac{trace(\Sigma)}{m}}\frac{\sqrt{m}}{\|d\|}\right)\right] =$$

$$= 1 - \left[\Phi\left(c\frac{\sqrt{trace(\Sigma)}}{\|d\|}\right) - \Phi\left(-c\frac{\sqrt{trace(\Sigma)}}{\|d\|}\right)\right],$$

where Φ is the normalized Gaussian *N(0,1)* cumulative distribution function.

The assumption $c\sqrt{trace(\Sigma)} \leq \|d\|$ implies $\frac{c\sqrt{trace(\Sigma)}}{\|d\|} \leq 1$. Therefore,

$$\Phi\left(\frac{c\sqrt{trace(\Sigma)}}{\|d\|}\right) \leq \Phi(1),$$

because Φ is monotone. Similarly,

$$\Phi\left(-\frac{c\sqrt{trace(\Sigma)}}{\|d\|}\right) \geq \Phi(-1).$$

Therefore,

$$1 - \left[\Phi\left(c\frac{\sqrt{trace(\Sigma)}}{\|d\|}\right) - \Phi\left(-c\frac{\sqrt{trace(\Sigma)}}{\|d\|}\right)\right] \geq 1 - \left(\Phi(1) - \Phi(-1)\right) = 0.3173.$$

Of course, the derivation refers not to the distributions but to the means, so that the threshold derived, ε=0.3173, should hold "on average", not always. Yet it gives us a reasonable estimate of the threshold value, which has been confirmed in a series of computational experiments with generated two-Gaussian data. Therefore, in our computations we use the threshold value ε=0.32. The BiKM method with the random projections stopping rule will be denoted as BiKM-R. One should notice that, at this method, any cluster may be subject to splitting even in the case when it was declared "not-to-be split" at a previous step – what is referred to as option (a) in this paper. This may happen because of the randomness of the generated directions and can never happen with the principal direction because it is constant. Therefore, one can either permit splitting those leaf clusters that have been declared final on previous steps, or not. In our experiments, using (a) outperforms the case of not using it, so that further on the random directions stopping condition applies to all leaf clusters, even those that have been qualified as not-to-split on a previous step.

4. Experiment Setting

We apply both methods to two 2D examples considered by Tasoulis, Tasoulis, and Plagianakos (2010), see Figure 2, as well as to synthetic datasets generated with controllable levels of cluster intermix and noise objects.

Given a data set with K^* generated clusters C_k $(k=1,\ldots, K^*)$, a divisive clustering method with an innate stopping criterion produces a partition with K leaf clusters S_l $(l=1,\ldots, K)$. We utilize the following popular criteria to score the similarity between found partition $\{S_l\}$ and given partition $\{C_k\}$:

(1) Number of found clusters K, so that the difference between K and K^* can be assessed;

(2) The Purity Index, an average proportion of the overlap between obtained clusters and those generated:

$$p(C\mid S)=\frac{\sum_{k=1}^{K^*}\max\{|C_k\cap S_1|,\ldots.,|C_k\cap S_K|\}}{N}. \tag{3}$$

(3) Adjusted Rand Index (ARI). It takes the average proportion of pairs that are similar between the partitions (two entities are similar in C and S if the pair belongs to the same cluster, or it does not, in both) and rescales it so that the expected mean is zero and the boundaries are -1 and +1. ARI is computed over the $K^*\times K$ confusion matrix elements of which are cardinalities N_{kl} of intersections $C_k \cap S_l$, according to formula (Hubert and Arabie 1985):

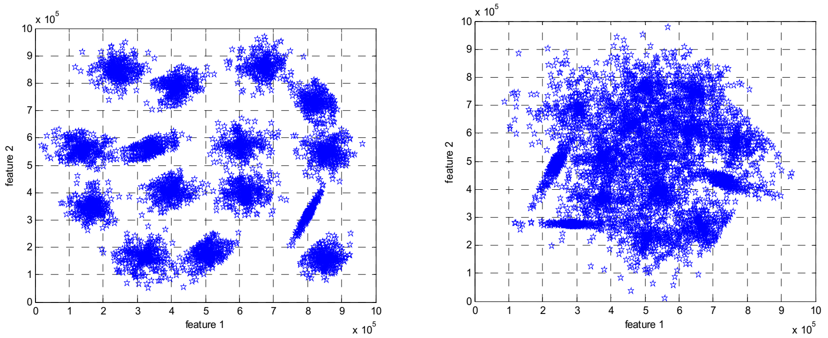

Figure 2. Two-dimensional sets from Tasoulis, Tasoulis, and Plagianakos (2010) with 15 Gaussian clusters, *S1* (on the left) and *S2* (on the right).

$$ARI = \frac{\sum_{k=1}^{K}\sum_{l=1}^{L}\binom{N_{kl}}{2} - \left[\sum_{k=1}^{K}\binom{N_{k+}}{2}\sum_{l=1}^{L}\binom{N_{+l}}{2}\right] / \binom{N}{2}}{\frac{1}{2}\left[\sum_{k=1}^{K}\binom{N_{k+}}{2} + \sum_{l=1}^{L}\binom{N_{+l}}{2}\right] - \left[\sum_{k=1}^{K}\binom{N_{k+}}{2}\sum_{l=1}^{L}\binom{N_{+l}}{2}\right] / \binom{N}{2}}.$$

Here N_{k+} is the sum of k-th row in the matrix, and N_{+l}, the sum of its l-th column.

Let us point out a property of *ARI*: *ARI*=0 if one of the partitions consists of just one universal cluster containing the entire entity set. Indeed, if say L=1, then the first item in the numerator gets equal to

$$\sum_{k=1}^{K}\binom{N_k}{2},$$

while the second, subtracted, item gets the factor on the right equal to 1, so that it is also equal to

$$\sum_{k=1}^{K}\binom{N_k}{2},$$

which makes the difference between them 0.

This brings forth the question of whether *ARI* can ever be negative; we did not see it raised before. Yes indeed, it can. Given a partition *C*, one may define the so called dual partitions (Schreider and Sharov 1982, see also Mirkin 1996, p. 237–238). Each class of a dual partition *S* consists of single representatives of different classes of *C*, so that no intersection $C_k \cap S_l$ may consist of more than one entity, thus making the denominator in *ARI* negative and so the *ARI* itself. In our experiments the negative values of *ARI* fluctuated around –0.10-0.15. This leaves the question of minimum *ARI* value open.

Synthetic datasets are usually generated using relatively simple cluster structures with rather well separated clusters (Milligan 1996). To use changeable cluster intermix parameters, more specific data generation models were proposed recently (see, for example, Steinley and Brusco 2007, Chiang and Mirkin 2010). Yet these data generators involve too many user-specified parameters which may obscure the relation between the generated data structure and the accuracy of an algorithm. Therefore we propose one more data generator of Gaussian clusters. In contrast to other approaches, this data generator allows to control spreading both within and between clusters by using just a single parameter. As previous experiments show rather convincingly, neither the cluster cardinalities nor dimensions of the data table play important role in cluster recovery when using K-Means and similar algorithms (Steinley and Brusco 2007, Chiang and Mirkin 2010, Amorim and Mirkin 2012). Therefore, we specify each generated dataset to consist of N=1500 M-dimensional entities at M=15. Each entity is generated independently from a Gaussian distribution modelling a cluster. Given a number of clusters K, the cardinality of each cluster is defined as a pre-specified minimum value, in this case 60, plus a uniformly random value selected so that the total of cardinalities is equal to N. Having rather large numbers of entities in the generated clusters makes applicable the procedure for building one-dimensional density functions in dePDDP and BiKM-R methods. For example, at K=5 clusters, the cardinality of each cluster is defined as 60 plus the corresponding proportion of the 1200 entities remaining to be distributed. To this end, we generate four uniformly (pseudo) random numbers, for example, (0.6324, 0.0975, 0.2785, 0.5469), sort them in the ascending order, and take the proportions of 1200, corresponding to the differences between the neighboring values and the complement to 1200 of the total of the four. Finally, by adding the minimum 60 to these, we obtain the final distribution of cluster cardinalities totalling to 1500. We also generate datasets of the same dimension with 9, 15, and 25 clusters. Of course, in the latter case, all clusters have been of the same cardinality, 60. We recognise the limited scope of the datasets generated, yet this suffices to see whether the two methods under investigation have different areas of strength.

Given a cluster's cardinality, we generate parameters of the corresponding Gaussian distribution. We specify $d_1=1$, $d_2=-1$ and generate the cluster's center, the mean, as a uniformly random vector from the M-dimensional box $a[d_1, d_2]^M$ where $[d_1, d_2]$ is interval of reals between d_1 and

d_2 and *a* parameter controlling the level of intermix among the clusters: the smaller the *a*, the greater the intermix. The covariance matrix is generated as a diagonal matrix with uniformly random variances on the diagonal generated within the interval $[0.025, 0.05](d_2 - d_1)$. In doing this we follow conclusions of a comprehensive experiment by Chiang and Mirkin (2010) that the extent of "oblique elongation" of clusters more or less uniformly affects different square-error based clustering algorithms, and in fact, much less than the cluster intermix. Therefore, for the purposes of comparison of dePDDP and K-MRad over the cluster intermix, the structures generated should suffice. After this, a pre-specified number of points is generated according to the Gaussian distribution with thus specified center and covariance matrix.

At *a=1* clusters are generated as rather clearly distinguishable from each other: the number of intermixed cases at which the Gaussian density at a generated point is smaller than the density of another cluster at this point (Chiang and Mirkin, 2010) is rather small. When value *a* decreases, the number of intermixed cases grows. The effect of decreasing the value of *a* is illustrated on Figure 3 – from well separated clusters on the left at *a*=0.75, through less separated clusters at *a*=0.50 to a rather messy cloud on the right (*a*=0.25). In our experiments, these three values of *a* are taken to compare the effects of the level of cluster intermix on the algorithms' performances.

Another effect to be tested is the presence of noise data, that is, points generated as uniformly random within the box defined by the maxima and minima of features on the generated dataset. At each experimental setting, we generated 20 datasets so that all the results reported further on are averages over these generated data sets. To include noise data, each of the datasets was replicated and 20% (i.e., 300) noise points were added to it.

5. Experimental Comparison of dePDDP and BiKM-R Methods

5.1. Results at Two-Dimensional Sets S1 and S4

Two-dimensional datasets S1 and S4 have been generated by Tasoulis, Tasoulis, and Plagianakos (2010) as a layman test bed for clustering algorithms (see Fig. 2 above). The former is of a relatively clear cluster structure, whereas the latter is somewhat messy. Table 2 presents clustering results for these datasets found with the selected algorithms.

The ARI columns in Table 2 show that dePDDP is the clear winner at both datasets, with BiKM-R with no option (a) involved, a close runner-up. Also, one can clearly see that the Purity index values are not sensitive to the numbers of clusters found. Using cardinality 5 as a split stopping criterion (see columns on the right) does not affect the results that

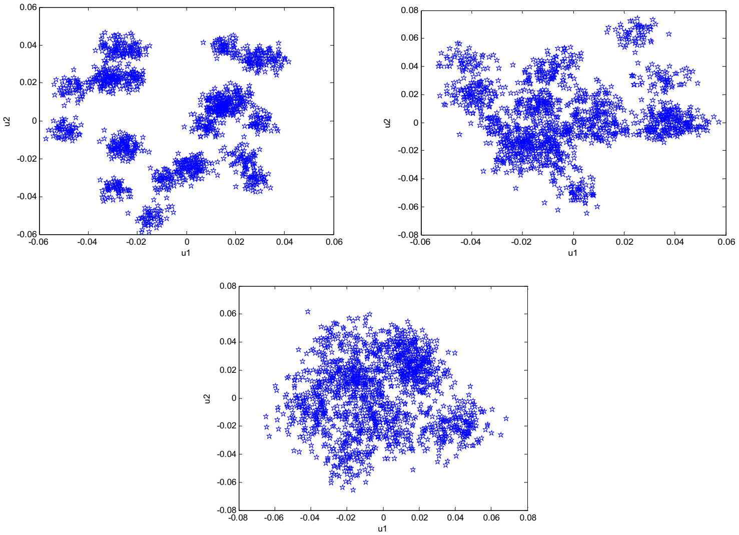

Figure 3. Projection of a 25-cluster generated dataset to the plane of the first two principal components. The values of α are 0.75 (on the left), 0.50 (in the middle) and 0.25 (on the right).

Table 2. Clustering results at 2D datasets S1, S4 (K*=15).

Method	Dataset S1 (Figure 2., left)				Dataset S4 (Figure 2, right)			
Index	Purity	ARI	K	K_adjust	Purity	ARI	K	K_adjust*
dePDDP	0.9930	0.969	23	19	0.7534	0.5693	31	30
BiKM-R	0.9708	0.929	25	25	0.7332	0.5061	27	26
BiKM-R (a)	0.9709	0.730	65	59	0.7444	0.3770	144	135

*K_adjust is the number of clusters after those with 5 or fewer elements have been excluded.

much, probably because the generated clusters cardinalities are much greater, 60 or more entities.

5.2. Accuracy at Generated Gaussian Cluster Datasets Regarding the Intermix

We present the accuracy of the algorithms on synthetic datasets, generated with the generator described above, in three tables corresponding to different values of the intermix parameter: *a*=0.75 (low intermix), *a*=0.50 (medium intermix), *a*=0.25 (high intermix), see Tables 3, 4, 5. The results in these tables concern both (i) the situation with no noise in the dataset and (ii) 20% of random noise points added to the generated dataset.

The original hypothesis, at these experiments, was that all the algorithms should do well at clear-cut cluster structures and be worse at noisy intermixed data. The former structure types, with no noise, should be a niche for dePDDP to win, whereas BiKM-R should be better at intermixed and/or noisy data. The expected reason for this is that the principal axis direction, the dePDDP base, should get confused at greater intermixes and noise. We did not expect BiKM-R(a) to be competitive because of its propensity to cut too many clusters, as clearly seen in Table 2.

The Tables 3, 4, 5 do support the view of the overall tendency to produce less accurate results with the intermix and noise growing. Yet they show somewhat more versatile picture, even in this regard, not to say of comparative performances of the algorithms.

First of all, one can see that BiKM-R(a), the version of Bisecting iK-Means at which the decision to stop splitting a cluster is reconsidered at each step of the algorithm, consistently shows a better performance than the generic BiKM-R method. Second, dePDDP is the best, at clear-cut structures, only when the number of clusters is relatively small (see Tables 3, 4). In these cases, dePDDP is the best at messier cluster structures as well. On the contrary, BiKM-R(a) is the best at larger numbers of generated clusters, K*=15, 25, with or without noise.

However, dePDDP's performance at greater cluster numbers gets better than that of its BiKM-R counterparts when the intermix grows further. According to Table 5, at *a*=0.25 with no noise objects added, dePDDP is better than both BiKM-R and BiKM-R(a), at all cluster numbers tested, from 5 to 25. Yet one cannot help but notice that, in fact, the method collapses at K*=15 or K*= 25. However, BiKM methods collapse even more comprehensively. At this level of intermix, *a*=0.75, they are unable to distinguish more than two clusters at K*=5, 9 or even no clusters at all, at K*=15, 25; see Table 5 on the left. Of course the ARI values must be about 0 when an algorithm builds no clusters in the data!

The tendencies are similar at the situations at which 20% noise objects are added to datasets, except for the cases of the large number of generated clusters, K*=25. The distance based BiKM-R(a) method significantly outperforms the dePDDP method at K*=25, whatever the cluster intermix under consideration, as clearly seen in Tables 3, 4, 5. Moreover, at the greatest intermix *a*=0.25, BiKM-R(a) outperforms dePDDP even at a smaller number of generated clusters, K*=15 (see Table

Table 3. Comparative accuracy of the methods at *a*=0.75 at different generated numbers of clusters K* with or without noise. The entry 5.10(0.31) in column K reads: the average number of clusters K is 5.10 with the standard deviation 0.31. The best results are highlighted using bold font.

Method	K*	No noise			20% noise added		
		K	ARI	Purity	K	ARI	Purity
dePDDP	5	**5.10(0.31)**	**1.00(0.00)**	**1.00(0.00)**	**5.05(0.22)**	**1.00(0.00)**	**1.00(0.00)**
BiKM- R(a)		**4.90(0.31)**	0.96(0.11)	0.99(0.02)	5.20(0.41)	**1.00(0.00)**	**1.00(0.00)**
BiKM- R		4.80(0.41)	0.95(0.11)	0.99(0.03)	5.15(0.49)	0.98(0.08)	1.00(0.02)
dePDDP	9	9.65(0.88)	**1.00(0.01)**	**1.00(0.00)**	9.30(0.57)	**1.00(0.00)**	**1.00(0.00)**
BiKM- R(a)		8.80(0.41)	0.98(0.05)	0.99(0.02)	9.30(0.66)	1.00(0.02)	1.00(0.01)
BiKM- R		**8.85(0.59)**	0.99(0.02)	0.99(0.02)	**9.25(0.72)**	0.99(0.03)	1.00(0.01)
dePDDP	15	16.65(1.04)	0.98(0.01)	**1.00(0.00)**	16.90(1.29)	**1.00(0.00)**	**1.00(0.01)**
BiKM- R(a)		**15.15(0.49)**	**1.00(0.01)**	1.00(0.01)	15.80(1.11)	1.00(0.01)	**1.00(0.01)**
BiKM- R		14.25(1.16)	0.95(0.06)	0.97(0.04)	**15.10(1.29)**	0.95(0.05)	0.98(0.03)
dePDDP	25	29.20(2.02)	**0.97(0.02)**	**1.00(0.00)**	28.35(6.75)	0.93(0.22)	0.95(0.21)
BiKM- R(a)		**25.10(1.52)**	**0.97(0.03)**	0.99(0.01)	26.75(1.45)	**0.99(0.02)**	**0.99(0.01)**
BiKM- R		22.60(1.50)	0.89(0.04)	0.94(0.03)	**25.40(1.60)**	0.95(0.05)	0.97(0.03)

5). This is probably because of greater distortions, in this case, of the principal direction on which the dePDDP method is based. The explanation can be supported with the evidence of larger standard deviations of the number of clusters found with dePDDP at different generated datasets at K*=25: they are several times larger than those at BiKM-R and BiKM-R(a) methods in Tables 3 and 4.

5.3. Accuracy of The Algorithms Regarding the Noise

In this section, we give a more detailed account of performances of the algorithms dePDDP and BiKM-R(a) in a situation of intermixed clusters (*a*=0.25) at different levels of noise (see Figures 4 and 5). At each of the generated ten data sets corresponding to the same setting, K*=15 and *a*=0.25, 11 levels of noise have been introduced by adding 0, 150, 300, 450, 600, 750, 900, 1050, 1200, 1350, and 1500 of random noise points to the fifteen hundred generated entities.

Table 4. Comparative accuracy of the methods at *a*=0.5 at different generated numbers of clusters K* with or without noise. The entry 5.30(0.47) in column K reads: the average number of clusters K is 5.30 with the standard deviation 0.47. The best results are highlighted using bold font.

Method	K*	No noise			20% noise added		
		K	ARI	Purity	K	ARI	Purity
dePDDP	5	**5.30(0.47)**	**1.00(0.01)**	**1.00(0.00)**	**5.15(0.37)**	**1.00(0.00)**	**1.00(0.00)**
BiKM- R(a)		4.35(0.88)	0.83(0.20)	0.93(0.10)	5.70(0.73)	**1.00(0.00)**	**1.00(0.00)**
BiKM- R		4.55(0.60)	0.92(0.16)	0.96(0.06)	5.55(0.76)	**1.00(0.00)**	1.00(0.01)
dePDDP	9	9.70(0.66)	**1.00(0.01)**	**1.00(0.00)**	10.45(1.32)	0.99(0.01)	**1.00(0.00)**
BiKM-R(a)		**8.60(0.88)**	0.95(0.05)	0.98(0.03)	10.40(1.47)	0.99(0.01)	1.00(0.01)
BiKM-R		8.00(0.86)	0.89(0.11)	0.95(0.04)	**9.90(1.07)**	**1.00(0.01)**	1.00(0.01)
dePDDP	15	17.65(3.10)	0.93(0.18)	**0.99(0.03)**	19.25(2.31)	**0.98(0.02)**	**0.99(0.01)**
BiKM- R(a)		**14.40(0.88)**	**0.95(0.06)**	0.98(0.03)	16.60(1.35)	0.95(0.05)	0.97(0.02)
BiKM- R		12.35(1.39)	0.80(0.12)	0.92(0.04)	**15.75(1.25)**	0.93(0.04)	0.96(0.02)
dePDDP	25	30.70(8.81)	0.86(0.28)	0.93(0.21)	25.95(12.19)	0.72(0.40)	0.83(0.34)
BiKM- R(a)		**25.70(2.11)**	**0.95(0.05)**	**0.98(0.03)**	29.35(2.58)	**0.94(0.05)**	**0.98(0.02)**
BiKM- R		19.45(2.74)	0.71(0.11)	0.86(0.06)	**24.70(1.69)**	0.84(0.06)	0.92(0.03)

One can see that the BiKM-R(a) method, which is hopeless at the situation of no noise, becomes the undisputed winner at the noise of the order of 10%-35% of the original sample. When we double the number of genenerated entities to N=3000, at the same cluster setting, the accuracy of both algorithms show a remarkable growth in the cluster recovery levels, although with similar patterns of behavior, as clearly seen from the graphs of Figure 5. The accuracy of dePDDP steadily decreases from ARI=0.8, at the situation of no noise, to ARI=0.3 at the situation at which the amount of noise is comparable to the number of entities. In contrast, the accuracy of BiKM-R(a) is 0 at the situation of no noise, but then it shoots up to almost ARI=0.9 at the proportion of noise between 0.1-0.5. A similar effect is observed when the number of entities grows further to 5000 (not shown).

As noted above, the zero ARI value corresponds to the situation at which BiKM-R is not able to cluster the dataset, so that K=1. The effect of better clustering at the larger numbers of entities, probably can be

Table 5. Comparative accuracy of the methods at the most intermixed data structure, *a*=0.25, at different generated numbers of clusters K* with or without noise. The entry 6.35(1.18) at column K reads: the average number of clusters K is 6.35 with the standard deviation of 1.18. The best results are highlighted with bold font.

Method	K*	No noise			20% noise added		
		K	ARI	Purity	K	ARI	Purity
dePDDP	5	**6.35(1.18)**	**0.98(0.03)**	**0.99(0.02)**	**7.40(1.70)**	**0.99(0.01)**	**0.99(0.02)**
BiKM- R(a)		2.10(1.33)	0.31(0.36)	0.57(0.26)	13.25(2.02)	0.72(0.15)	0.97(0.03)
BiKM- R		1.75(0.85)	0.23(0.25)	0.53(0.20)	10.40(1.82)	0.77(0.15)	0.97(0.03)
dePDDP	9	**14.45(1.85)**	**0.95(0.02)**	**0.97(0.02)**	13.35(3.08)	**0.90(0.20)**	**0.96(0.04)**
BiKM- R(a)		2.25(2.05)	0.16(0.27)	0.39(0.28)	16.40(4.07)	0.84(0.21)	0.90(0.17)
BiKM- R		1.60(1.05)	0.09(0.15)	0.27(0.20)	**11.95(2.61)**	0.83(0.17)	0.91(0.06)
dePDDP	15	**16.85(8.74)**	**0.50(0.34)**	**0.82(0.26)**	**14.25(8.57)**	0.49(0.38)	0.77(0.32)
BiKM- R (a)		1.20(0.52)	0.01(0.03)	0.11(0.11)	18.55(7.75)	**0.73(0.32)**	**0.79(0.31)**
BiKM- R		1.10(0.31)	0.01(0.02)	0.10(0.10)	7.95(4.11)	0.38(0.27)	0.65(0.27)
dePDDP	25	**7.80(9.49)**	**0.06(0.07)**	**0.42(0.39)**	5.05(7.42)	0.05(0.09)	0.25(0.36)
BiKM- R (a)		1.00(0.00)	0.00(0.00)	0.04(0.00)	**9.80(11.04)**	**0.28(0.32)**	**0.35(0.37)**
BiKM- R		1.00(0.00)	0.00(0.00)	0.04(0.00)	3.85(2.32)	0.09(0.08)	0.26(0.19)

attributed to the fact that at the given parameters of the data model, especially the modest space dimensionality, doubling the cluster cardinalities makes them more concentrated and, thus, better "visible" by the K-Means process. The effect of steadily declining of the accuracy of dePDDP when noise increases can be attributed to the growing instability of the principal axis of the data cloud in this case. In contrast, adding moderate levels of noise to the cluster intermix probably puts the intermix on par with the random noise so that the cluster interiors become more "visible"in the multidimensional space, leading to improving similarity between generated clusters and those found using BiKM-R(a).

Let us summarize our findings. Using our intuition on the least-squares criterion "simplicity", we expected that the two methods would broadly differ over intermixed and noisy data structures: the more separated the clusters, the better dePDDD, and, in contrast, the greater the intermix and noise, the better the BiKM-R. Yet we found somewhat different dividing lines:

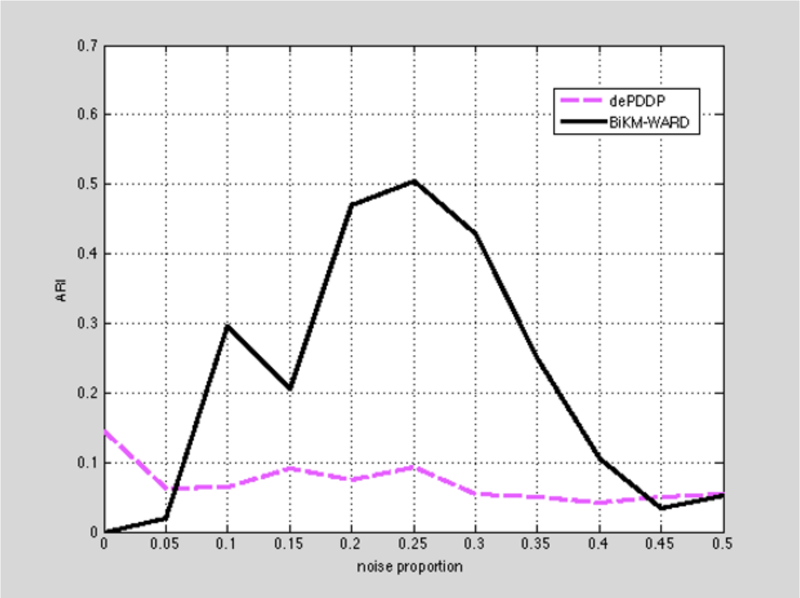

Figure 4. Accuracy of dePDDP (dashed) and BiKM-R(a) (solid) methods with respect to the noise added at N=1500, K*=15 and *a*=0.25. The *x*-axis represents proportion of the added noise, and *y*-axis, the value of ARI, the similarity index between the generated partition and that found at K=K*.

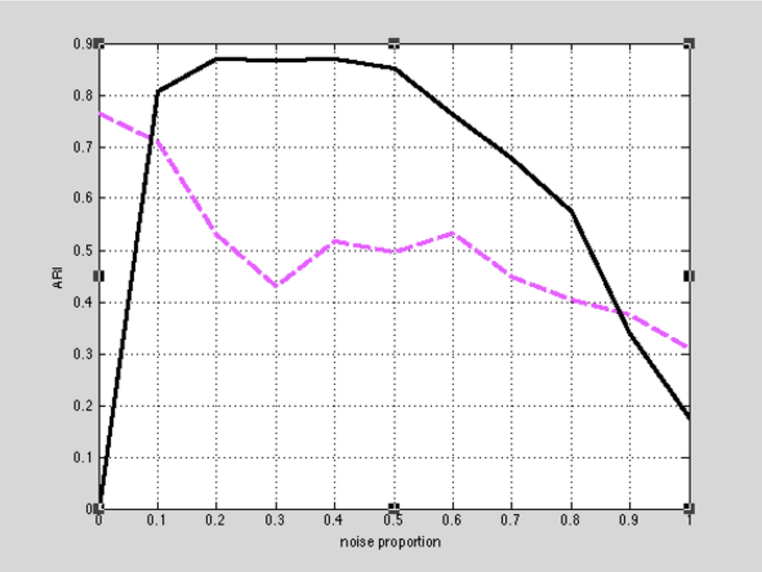

Figure 5. Accuracy of dePDDP and BiKM-R(a) (black solid) methods with respect to the noise added at N=3000, K*=15 and *a*=0.25. The *x*-axis represents proportions of the added noise, and *y*-axis, the values of ARI, the similarity index between the generated partition and that found at K=K*.

(a) dePDDP method is quite steady in the intermixed clusters data, but it is sensitive to increasing the number of clusters: while being a better bet at relatively small cluster numbers, dePDDP's performance is weaker at the larger number of clusters even at clear-cut clusters;
(b) BiKM-R dismally fails at greatly intermixed clusters, but it greatly improves when the data is affected with a moderate noise, of the order of 30% of the number of entities.

6. Discussion

In this paper, the two most popular divisive clustering methods, Division over Principal Directions and Bisecting K-Means, are shown to be different versions of the same least-squares greedy splitting strategy. Building on the success of the one-dimensional "minima-of-the-density-function" strategy by Tasoulis, Tasoulis, and Plagianakos (2010) in PDDP, we decided to implement a similar stopping rule strategy to the multivariate Bisecting K-Means by using projections of the data onto random directions and see whether it works at all. It appears it does, as described in the end of the previous section.

It should be mentioned, as well, that there is a wide scope for developing many more versions of divisive clustering, even within the same least-squares framework. There are three points that can be changed independently of each other: (i) the stopping criterion, (ii) the choice of cluster to split, (iii) the choice of algorithm for splitting.

Specifically, with the random projections, one should check the effect of increasing the number of random projections and changes in the threshold ε. We have found (not reported here) that the former has no visible effects. Changing the threshold ε may bring changes indeed. For example, we may resort to the minimum threshold value so that a cluster is to be split if its projection onto at least one random direction has a minimum. This rule appears by far inferior in almost all the situations except for the case of very noisy data with high levels of both cluster intermix and noisy entities. We also experimented with the option of non-splitting those clusters that have too few, say 5 or less, elements; the conclusions are similar for both using the option and not using it.

We also tested a statistical criterion applied to the one-dimensional Parzen-type density function above. There has been a great deal of research devoted to theoretical derivations for testing the hypothesis that the data is a sample from a mixture of two Gaussian distributions against the hypothesis that the data come from a single Gaussian distribution (see, for example, Bock 1996). One of the best decision rules is based on the ratio of likelihood statistics. The model of a mixture of two Gaussian distributions can be operationally specified by dividing all the projection points in two classes by using the Bisecting K-Means; then the clusters can be modelled as Gaussian distributions with parameters *(m_k, σ_k), $k=1, 2$*. The ratio of the log-likelihood statistic, in this case, is

$$2LR = 2\sum_{i=1}^{N} \frac{log\left(p_1 \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x_i - m_1)^2}{2\sigma_1^2}\right) + p_2 \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{(x_i - m_2)^2}{2\sigma_2^2}\right)\right)}{\log\left(\frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{(x_i - m_0)^2}{2\sigma_0^2}\right)\right)}$$

The numerator is the log-likelihood of a mixture of two Gaussian distributions. The p_1 and p_2 are mixing parameters and *(m_k, σ_k)* are Gaussian parameters, the mean and standard deviation (*$k=1,2$*). Of course in real-world computations the parameters are substituted by their sample-based estimates. Analogously, m_0 and σ_0 in the denominator stand for parameters of the only Gaussian at the alternative hypothesis of a single Gaussian distribution. The estimates of means in the numerator are taken as centers of clusters found with Bisecting K-Means algorithm; the standard deviations are computed as those within the clusters, and the mix probabilities as the proportions of points in the clusters. Parameters for the alternative hypothesis are estimated as the mean and standard deviation on the whole set. The stopping condition for dePDDP is $2LR<0$. In the setting of random projections, we use the averaged value of *2LR* in the same way. Our experiments show that these stopping rules work well at clear-cut structures with few clusters, a=0.75 and K*=5 or 9. However they do not work, and in fact dismally fail, at structures with a greater intermix, that is, at a=0.5, 0.25, and/or a greater number of clusters, K*=15, 25.

7. Conclusion

The prime goal of the research project resulted in this paper was to develop a viable alternative to the dePDDP method of divisive clustering by implementing its one-dimensional “no-minima” stopping rule in the multivariate framework of Bisection K-Means method. The reported results contribute to the body of literature on clustering in each of the three aspects: theoretic, algorithmic, and experimental ones. In the theoretic aspect, the paper presents a mathematical framework unifying two popular divisive clustering approaches, Bisecting K-Means and Principal Direction method. In this framework, divisive clustering is considered as a technique to grow a binary tree approximately representing the data. We have shown that the two seemingly much different methods are just slightly different approaches to locally minimize the least-squares criterion. Moreover, a “non-greedy” summary splitting criterion (8) is derived as a proper representation of the least-squares approximation. Maximizing this criterion is a problem which is yet to be addressed.

In the algorithmic aspect, the main innovation is the proposed Random directions approach that brings the efficiency of one-dimensional statistics, well exploited in dePDDP method, to the multivariate Bisecting K-Means. This involves some user-defined parameters such as the number of random projections to generate and majority threshold. The number of random projections is probably not that important (see Tasoulis, Tasoulis, and Plagianakos 2013), yet the results are rather sensitive to the value of the majority threshold in some cases. Although our analysis of the issue of thresholding "on average" leads us to a reasonable threshold value of about one third, it is likely that the parameter could be used as a tool for fine tuning the method, which is yet to be explored. In our experiments both methods took approximately similar computational times at a run: the computational cost of deriving a principal direction in dePDDP is balanced by the need to estimate a greater number of the density functions and their minima in BiKM-R. Yet this is not a big deal because both methods are oriented at moderate size data.

In the experimental aspect, we developed a multivariate Gaussian cluster structure data generator in which the extent of cluster intermix can be easily controlled by changing just one parameter. By using this generator we arrive at the following conclusions. Method dePDDP (Tasoulis, Tasoulis, and Plagianakos 2010) works quite well at any number of clusters, even when the cluster intermix is on increase; but it fails at larger numbers of clusters, when random objects are added as noise. At more or less "regular" conditions our BiKM method is not as good as dePDDP, especially at a high cluster intermix, when it fails comprehensively, by collapsing all the clusters in one or two. This happens because the proportion of random directions at which the data density function has minima cannot reach as high a value as ε=0.32 specified above. Our preliminary experiments have shown that BiKM-R results greatly improve, in this case, if the proportion threshold is decreased, up to the value 0. This brings forth the issue of properly selecting the threshold value which should be adjusted depending on the dataset structure. Furthermore, when supplementary random noise entities are added, the method recovers and beats dePDDP dramatically, even with the unchanged threshold ε=0.32. This happens, probably, because with added noise objects the cluster intermix is perceived as a greater noise, and cluster cores become more visible to the splitting procedure. A feature of BiKM-R, an option that allows reconsidering split-stopping decisions at individual clusters from iteration to iteration, proves competitive when cluster intermix, noise and the number of clusters grow. Also, applying the classical one-dimensional statistics criterion for distinguishing between the cases of one versus two clusters, we found that this criterion works well only on data in which there are just a few rather well separated clusters.

References

ALBATINEH, A.N., NIEWIADOMSKA-BUGAJ, M., and MIHALKO, D. (2006), "On Similarity Indices and Correction for Chance Agreement", *Journal of Classification*, *23*, 301–313.

BOCK, H.H. (1996), "Probability Models and Hypothesis Testing in Partitioning Cluster Analysis", in *Clustering and Classification*, eds. P. Arabie, C.D. Carroll and G. De Soete, River Edge NJ: World Scientific Publishing, pp. 377–453.

BOLEY, D. (1998), "Principal Direction Divisive Partitioning", *Data Mining and Knowledge Discovery*, *2(4),* 325–344.

FENG, Y., and HAMERLY, G. (2006), "PG-Means: Learning the Number of Clusters in Data", in *Advances in Neural Information Processing Systems*, *19 (NIPS 2006),* eds. B. Schölkopf, J.C. Platt and T. Hoffman, MIT Press, pp. 393–400.

FRAYLEY, C., and RAFTERY, A. (1998), "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis", *The Computer Journal*, *41(8),* 578–588.

DASGUPTA, S. (1999), "Learning Mixtures of Gaussians", *IEEE Symposium on Foundations of Computer Science*, 634–644.

DASGUPTA, S. (2000), "Experiments with Random Projection", in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, San Francisco: Morgan Kaufmann, p. 143–151.

DEMPSTER, A.P., LAIRD, N.M., and RUBIN, D.B. (1977), "Maximum Likelihood from Incomplete Data Via the EM Algorithm", *Journal of the Royal Statistical Society. Series B (Methodological), 39(1),* 1–38.

EDWARDS, A.W.F., and CAVALLI-SFORZA, L.L. (1965), "A Method for Cluster Analysis", *Biometrics*, *21*, 362–375.

FISHER, D.W. (1987), "Knowledge Acquisition Via Incremental Conceptual Clustering", *Machine Learning, 2*, 139–172.

GOWER, J.C. (1967), "A Comparison of Some Methods of Cluster Analysis", *Biometrics*, *23*, 623–637.

HAZMAN, M., EL-BELTAGY, S.R., and RAFEA, A. (2011), "A Survey of Ontology Learning Approaches", *International Journal of Computer Applications*, *22(9),* 36–43.

HUBERT, L.J., and ARABIE, P. (1985), "Comparing Partitions", *Journal of Classification*, *2*, 193–218.

JOLLIFFE, I.T. (2002), *Principal Component Analysis* (2nd ed.), Springer Series in Statistics, New York: Springer.

JUNG, Y., PARK, H., DING-ZHU, D., and BARRY, L. (2003), "A Decision Criterion for the Optimal Number of Clusters in Hierarchical Clustering", *Journal of Global Optimization, 25*, 91–111.

MEILA, M. (2007), "Comparing Clusterings—An Information Based Distance", *Journal of Multivariate Analysis, 98(5),* 873–895.

MENDELL, R., RUBIN, D., and LO, Y. (2001), "Testing the Number of Components in a Normal Mixture", *Biometrika*, *88(3),* 767–778.

MICHALSKI, R.S., and STEPP, R.E. (1983), "Learning from Observation: Conceptual Clustering", in *Machine Learning: An Artificial Intelligence Approach*, eds. R.S. Michalski, J.G. Carbonell, T.M. Mitchell, San Mateo CA: Morgan Kauffmann, pp. 331–363.

MILLIGAN, G.W. (1996), "Clustering Validation: Results and Implications for Applied Analyses", in *Clustering and Classification*, eds. P. Arabie, C.D. Carroll and G. De Soete, River Edge NJ: World Scientific Publishing, pp. 341–375.

MIRKIN, B. (1996), *Mathematical Classification and Clustering*, Dordrecht: Kluwer.

MIRKIN, B. (2011), “Choosing the Number of Clusters”, *WIRE Data Mining and Knowledge Discovery*, *1*, 252–260.

MIRKIN, B. (2012), *Clustering: A Data Recovery Approach*, London: CRC Press/ Chapman and Hall.

MIRKIN, B., and MING-TSO CHIANG, M. (2010), “Intelligent Choice of the Number of Clusters in K-Means Clustering: An Experimental Study with Different Cluster Spreads”, *Journal of Classification*, *27*, 3–40.

NEWMAN, M.E.J. (2006), “Modularity and Community Structure in Networks”, *PNAS*, *103(23),* 8577–8582.

NG, A.Y., JORDAN, M.I., and WEISS, Y. (2001), “On Spectral Clustering: Analysis and an Algorithm”, *Advances in Neural Information Processing Systems, 2*, 849–856.

RAND, W.M. (1971), “Objective Criteria for the Evaluation of Clustering Methods”, *Journal of the American Statistical Association*, *66*, 846–850.

SCHREIDER, Y.A., and SHAROV, A.A. (1982), *Systems and Models* (in Russian), Moscow: Radio i Sviaz'.

SHI, J., and MALIK, J. (2000), “Normalized Cuts and Image Segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22(8),* 888–905.

SNEATH, P.H.A., and SOKAL, R.R. (1973), *Numerical Taxonomy*, San Francisco: W.H. Freeman.

SONQUIST J.A., BAKER E.L., and MORGAN J.N. (1973), *Searching for Structure*, Ann Arbor: Institute for Social Research, University of Michigan.

STEINBACH, M., KARYPIS, G., and KUMAR, V. (2000), “A Comparison of Document Clustering Techniques”, *KDD Workshop on Text Mining*, *400(1),* 525–526.

STEINLEY, D., and BRUSCO, M. (2007), “Initializing K-Means Batch Clustering: A Critical Evaluation of Several Techniques”, *Journal of Classification*, *24,* 99–121.

TASOULIS, S.K., and TASOULIS, D.K. (2008), “Improving Principal Direction Divisive Clustering”, in: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008), Workshop on Data Mining using Matrices and Tensors.*

TASOULIS, S.K., TASOULIS, D.K., and PLAGIANAKOS, V.P. (2010), “Enhancing Principal Direction Divisive Clustering”, *Pattern Recognition*, *43*, 3391–3411.

TASOULIS, S. K., TASOULIS, D. K., and PLAGIANAKOS, V.P. (2013), “Random Direction Divisive Clustering”, *Pattern Recognition Letters, 34(2),*131–139.

TEICHER, H. (1960), “On the Mixture of Distributions”, *Annals of Mathematical Statist*istics, *31(1),* 55–73.

VEMPALA, S. (2005), *The Random Projection Method*, DIMACS Series in Discrete Mathematics (Vol. 65), American Mathematical Society.

YEUNG, K.Y., FRALEY, C., MURUA, A., RAFTERY, A.E., and RUZZO, W.L. (2001), “Model-Based Clustering and Data Transformations for Gene Expression Data”, *Bioinformatics*, *17(10),* 977–987.