Density Based Projection Pursuit Clustering

Sotiris K. Tasoulis[*], Michael G. Epitropakis[†], Vassilis P. Plagianakos[*], and Dimitris K. Tasoulis[§]
[*]Department of Computer Science and Biomedical Informatics, University of Central Greece,
Papassiopoulou 2–4, Lamia, GR-35100, Greece. Email: {stas, vpp}@ucg.gr
[†]Department of Mathematics, University of Patras, GR-26110 Patras, Greece. Email: mikeagn@math.upatras.gr
[‡]Winton Capital Management, 1–5 St Mary Abbot's Place, London SW8 6LS, U.K. Email: d.tasoulis@wintoncapital.com

Abstract—**Clustering of high dimensional data is a very important task in Data Mining. In dealing with such data, we typically need to use methods like Principal Component Analysis and Projection Pursuit, to find interesting lower dimensional directions to project the data and hence reduce their dimensionality in a manageable size. In this work, we propose a new criterion of direction interestingness, which incorporates information from the density of the projected data. Subsequently, we utilize the Differential Evolution algorithm to perform optimization over the space of the projections and hence construct a new hierarchical clustering algorithmic scheme. The new algorithm shows promising performance over a series of real and simulated data.**

I. Introduction

Data clustering is a very important and challenging topic in machine learning. It aims to capture the structure of a dataset by creating clusters of similar elements based on a similarity measure. However applying clustering methodologies in high dimensional data is often a very difficult task due to the sparsity of the high dimensional spaces. This problem is known as the "curse of dimensionality" [1]. The most common technique to deal with such high dimensional datasets is to reduce their dimensionality by projecting the dataset onto a lower dimensional subspace. Projection pursuit [2] is the procedure of finding "interesting" projections for a dataset, i.e. directions that maintain its structure and at the same time reduce its dimensionality. This can be consider as an optimization task over the space of the projection directions, where one have to optimize the interestingness criterion.

Several measures of interestingness can be found in the literature. It has been argued by Huber [3] and by Jones and Sibson [4] that the direction in which the projected data are Gaussian distributed is the least interesting one, while the most interesting directions are those that exhibit the least Gaussian distribution. Classical measure of non-gaussianity are kurtosis and the fourth-order cumulant. The most widely used type of projection pursuit is the Principal Component Analysis (PCA) [5]. The interestingness criterion of the PCA is the variance of the projected data.

Motivated by a recently proposed hierarchical clustering technique developed in [6], we introduce a new interestingness criterion based on data cluster's separability. Its main characteristic is that incorporates information from the density of the projected data. In turn, we integrate the aforementioned interestingness criterion into a hierarchical clustering technique and create a new clustering algorithm. To efficiently tackle the optimization task for the projection pursuit procedure, we employ a stochastic optimization methodology, namely the Differential Evolution (DE) algorithm [7].

The remaining of the paper is structured as follows: In Section II, we give background material. Next, in Section III we examine the projection pursuit optimization problem and propose a new measure of interestingness. Section IV is devoted to the utilized optimization method. In Section V, we present a new hierarchical clustering algorithm and in Section VI we investigate the efficiency of the proposed technique. The paper ends with concluding remarks.

II. Background material

The "divisive" hierarchical clustering techniques produce a nested sequence of partitions, with a single, all-inclusive cluster at the top. Starting from this all-inclusive cluster the nested sequence of partitions is constructed by iteratively splitting clusters, until a termination criterion is satisfied. The operation of each algorithm can be understood, by the manner in which they answer the following questions:

Q_1: Which cluster to split further?
Q_2: How to split the selected cluster?
Q_3: When should the iteration terminate?

As this work is based on the recently proposed hierarchical clustering algorithm, dePDDP [6], for completeness purposes we will briefly describe it in the following subsection.

A. The dePDDP algorithm

The main characteristic of the dePDDP algorithm is that it incorporates information from the density of the projected data onto the first principal component. The dePDDP procedure suggests that the best we can do to avoid splitting clusters is to split the data based on the global minimizer of the estimated density of the projected data onto the first principal component. The cluster selection criterion and the termination criterion are guided by the same idea.

To formally describe how the high dimensional data are projected onto a lower dimensional space, let us assume the data at hand is represented by an $n \times a$ matrix D, in which each row represents a data sample $d_i, i = 1, \ldots, n,$ and a denotes the dimensionality. Let A be the matrix which columns are the vectors that denote the targeted subspace, then

$$D^P_{n\times k} = D_{n\times a} A_{a\times k},$$

is the projection of the data onto the lower k-dimensional subspace defined by the matrix A. In the case of projection pursuit procedure the targeted subspace is denoted by a $1 \times a$ vector a and $D^P_{n\times 1}$ is the projection of the data onto the one-dimensional direction which is determined by a.

In the case of dePDDP the projection method used is the PCA. If we define the vector b and matrix Σ to represent the mean vector and the covariance of the data respectively:

$$b = \frac{1}{n}\sum_{i=1}^{n} d_i, \quad \Sigma = \frac{1}{n}(D - be)^\top (D - be),$$

where e is a column vector of ones. The covariance matrix Σ is symmetric and positive semi-definite, so all its eigenvalues are real and non-negative. The eigenvectors u_j, $j = 1, \ldots, k$, corresponding to the k largest eigenvalues, are called the principal components or principal directions. The dePDDP algorithm use the projections p_i:

$$p_i = u_1(d_i - b), \quad i = 1, \ldots, n,$$

onto the first principal component u_1, to initially separate the entire data set into two partitions P_1 and P_2 based on a global minimizer defined in 2.1 as follows:

Definition 2.1: (**Global Minimiser**) A global minimizer x^* is a point of $\mathbb{R}$ such that $\hat{f}'(x^*;h') = \min_x \mathcal{X}$, where $\mathcal{X} = \{x \in R : \exists\delta > 0, \hat{f}'(x+\delta;h') > \hat{f}'(x;h')$ and $\hat{f}'(x-\delta;h') > \hat{f}'(x;h')\}$,
where $\hat{f}'(x;h')$ is the kernel density estimation of the density of the projected data onto the first principal component.

To demonstrate the intuition behind dePDDP, in Fig. 1 we illustrate a 2-dimensional example with the associated principal components and an approximation of the estimated density of the projection on the principal component.

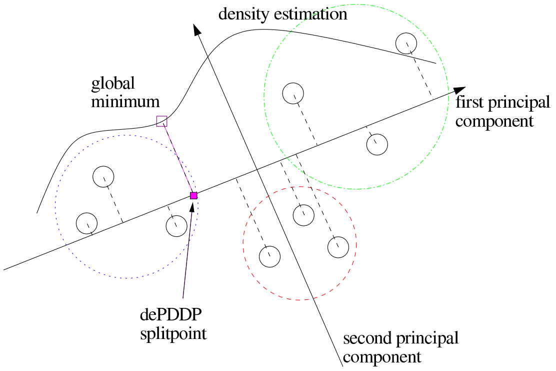

Fig. 1. An illustrative example of dePDDP.

III. Density-based projection pursuit

Before describing the proposed projection pursuit method, we need to define the optimization space. As we are interested of one-dimensional projections of the data, we can restrict the optimization space to the half part of a unit hyper-sphere.

For example, in the two dimensional space, the space of all possible directions can be first restricted to the vectors on the unity circle as vectors with different lengths produce the same direction. Also as the symmetric vectors define the same directions, the projection direction space can be further bounded in the half unit circle.

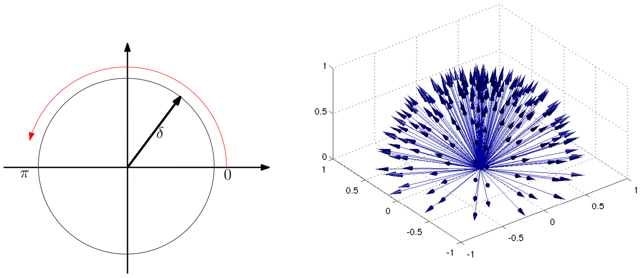

Fig. 2. Example of the projection direction space in the two (left) and three (right) dimensional cases.

As such the optimization space for the two dimensional case can be defined with the help of polar coordinates. Let θ be the angular coordinate such that $\theta \in [0, \pi]$ and the radial coordinate $\delta = 1$, then for any direction $[x, y]$ holds that $x = \delta\cos\theta$ and $y = \delta\sin\theta$ (see Figure 2 (left)). Similarly, in the three dimensional case the optimization space can be defined by the vectors on the surface of the half part of a unit sphere. To explain better the three dimensional case, we visualize the optimization space in Figure 2 (right).

Having defined the optimization space, the main goal is to find an optimizer, i.e. the best direction to project our data. PCA utilizes the variance of the projected data as a quality criterion, and assumes that the direction that maximizes it is the most appropriate. This turns out to be the principal direction of the data. Although PCA is a very effective technique, there are cases where the structure of the projected data onto the principal direction does not capture the data clustering structure. To illustrate such a case we employ a two dimensional dataset shown at Figure 3 (left). The projected data onto the principal component, as well as their corresponding density, are illustrated at Figure 3 (center). As expected, based on the dePDDP algorithm splitting criterion, we would be unable to appropriately split the data into clusters, because the density of the projected data has no minima.

To examine the optimization task of the PCA in this dataset, we can observe the quality criterion of the projected data for several directions (angular coordinate θ), in Figure 3 (right). The maximum variance direction is very evident and stable, but although the corresponding projection fails to capture the clustering structure.

Recently, in [8] we have introduced such a quality criterion guided by the minimizer of the density function of the projected data. A lower density value of the minimizer would determine a better direction [8]. However, no matter how coherent a dataset is, it is very common that there is a projection direction for which the projected data will contain outlying points. In those points, the density of the data will be very small and the particular direction would be recognised as

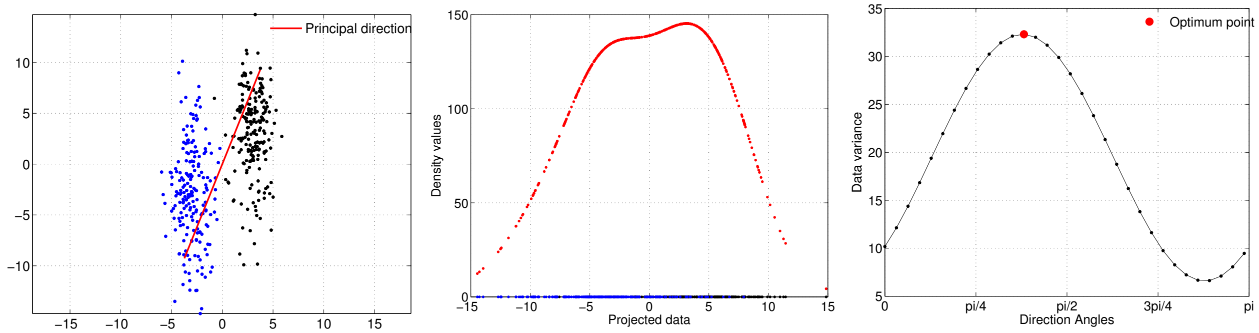

Fig. 3. The two–dimensional dataset with the principal direction (left). The projections onto the principal direction along with their densities (center) and The optimization quality criterion (right).

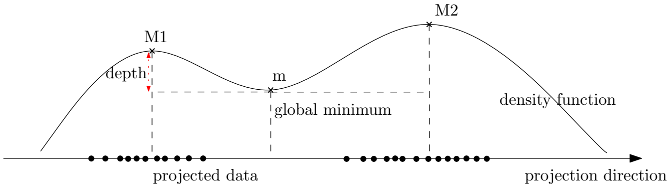

Fig. 4. The proposed quality criterion

a good one, irrespective of what happens to the bulk of the data. In such a case, the whole procedure could be guided by the outlying points. The aforementioned behavior leads to a hierarchical clustering algorithm that splits a dataset first to all the outlying points before it actually splits actual clusters.

In this work, we propose a new quality criterion, i.e. a new objective function that is able to avoid this problem. As long as we locate the minimizer x^* of the projected data onto a particular direction, we retrieve the maximum value of the density at the left ($M1$) and right ($M2$) side of the minimizer. The new quality criterion is defined as the difference between the density values of the splitting point and the minimum of $M1$ and $M2$ density values. Formally:

Definition 3.1: (**Quality Criterion**): Let u be any vector of R^a with $\|u\| = 1$, $\mathcal{P}$ be the set of projections p_i of the vectors d_i onto u, $\hat{f}'(x; h')$ be the kernel density estimation of the projections $p_i \in \mathcal{P}$, and $x^\star$ its global minimizer as defined in Definition 2.1. Let $M1$ and $M2$ be the maximum density value at the left and the right sides of $x^\star$ respectively and $M = max\{M1, M2\}$. The Quality Criterion is a function $QC : \mathbb{R}^a \to \mathbb{R}$ of u such that

$$QC(u) = \hat{f}'(M; h') - \hat{f}'(x^\star; h')$$

We refer to this quality criterion as *depth* (see Figure 4). Using the depth criterion the Projection Pursuit problem can be formally defined as follows.

Definition 3.2: (**Best Depth Projection Direction Problem**): Let $\mathcal{U} = \{u \in \mathbb{R}^a : \|u\| = 1\}$ represent the space of projection directions. Then the problem of finding the best depth projection direction resorts to finding the maximum u^{opt} of the projection directions space $\mathcal{U}$, i.e. the vector $u^{opt} \in \mathcal{U}$ such that:

$$QC(u^{opt}) \geqslant QC(u), \quad \forall u \in \mathcal{U}. \tag{1}$$

To exhibit the behavior of the proposed best depth projection direction methodology, we employ the two dimensional dataset used at the previous example. At the left side of Figure 5, we illustrate the best depth direction as well as the principal direction of the data set. The projected data and their corresponding density values onto the best depth direction are demonstrated at Figure 5 (center). As shown, this direction conveniently makes the projected data density to contain a minimum between the points of the two actual clusters. This is particularly very well suited for the dePDDP algorithm, as the splitting criterion used by that algorithm would effectively split the actual data clusters.

Similarly, for further visual understanding, we employ a three dimensional dataset constituted by two clusters of different sizes (see Figure 6). Figure 6 (left) illustrates the dataset along with the principal and the chosen best depth direction, while their projections and their corresponding densities are demonstrated at the center and right of Figure 6, respectively. Finally, Figure 7 (right) reports the landscape of the three dimensional optimization space and its optimal value. As show, the optimization landscape becomes more challenging as the dimensionality grows, since it is non-differentiable and highly multimodal. For this reason the utilization of a global optimization algorithm is essential.

IV. Differential Evolution

We attempt to tackle the aforementioned optimization problem using the Differential Evolution (DE) Algorithm [7]. DE is a stochastic parallel direct search method, which utilizes concepts borrowed from the broad class of Evolutionary Algorithms (EAs). DE is capable of handling non-differentiable, discontinuous, non-linear, noisy and highly multimodal objective functions, which makes it a suitable choice to handle the aforementioned landscapes.

More specifically, DE is a population–based stochastic algorithm that exploits a population of NP potential solutions, *individuals*, to effectively probe the optimization space. DE randomly initializes the population in the D–dimensional optimization domain through a uniform probability distribution.

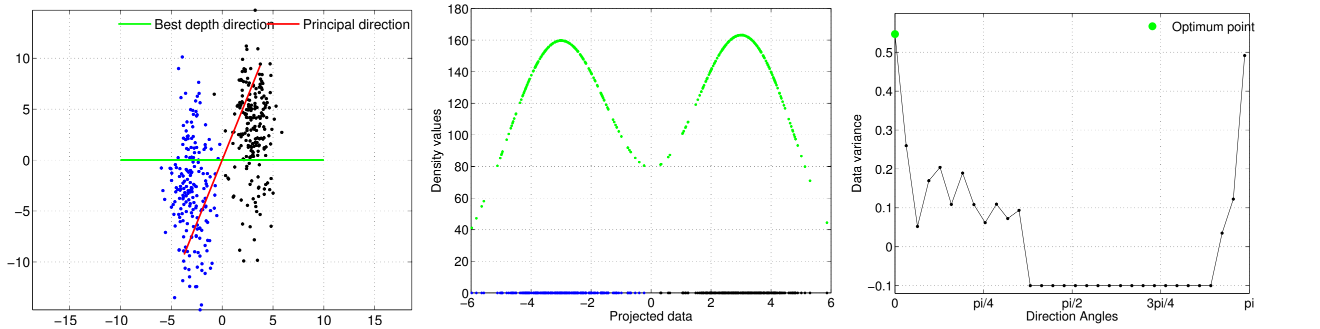

Fig. 5. The two–dimensional dataset with the principal and the best depth direction (left). The projections onto best depth direction along with their densities (center) and the depth quality criterion values (right) for each direction.

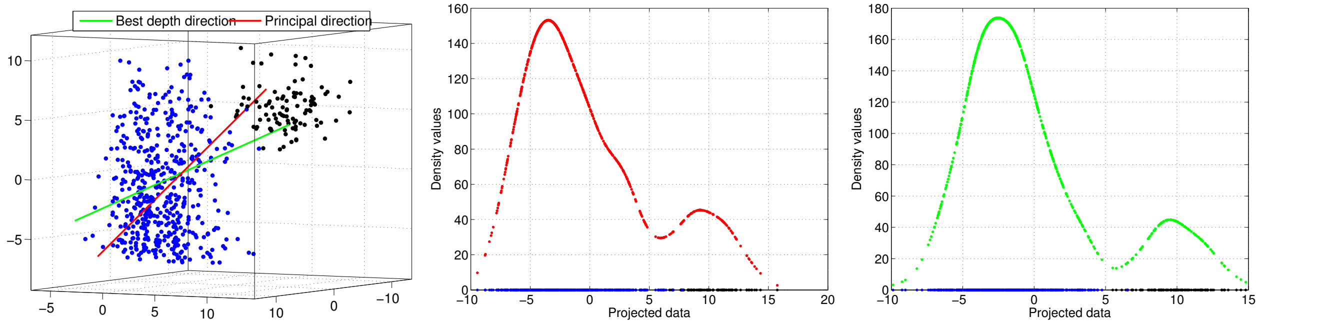

Fig. 6. The three–dimensional dataset with the principal and the best depth direction (left). The projections onto the principal direction (center) and the best depth direction (right) along with their densities.

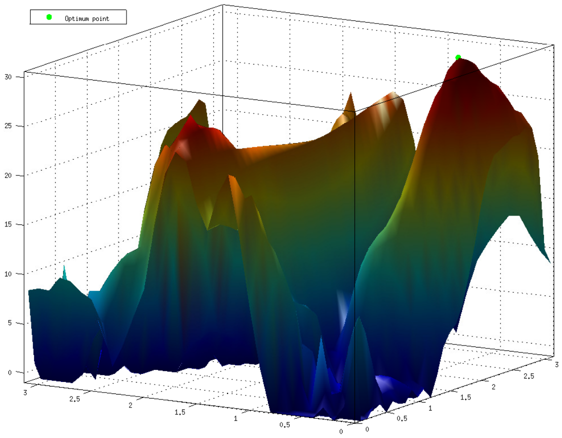

Fig. 7. The best depth optimization landscape of the three–dimensional dataset along with its optimum point.

Individuals evolve over successive steps to explore promising regions of the search space and locate the minima of the objective function. The user–defined population size, *NP*, is fixed throughout the evolution process. At each iteration, called *generation*, new vectors are derived by the combination of randomly chosen vectors from the current population. This operation in our context can be referred to as *mutation*, while the outcoming vectors as *mutant individuals*. Several mutation strategies have been proposed in the DE literature. The most common and widely used can be found in [7], [9]–[11]. Afterwards, each mutant individual is then mixed with another vector – the *target* vector – through an operation called *recombination* or *crossover*, which yields the so–called *trial* vector. The most well known and widely used variants of DE utilize two main crossover schemes; the *exponential* and the *binomial* crossover [7], [9], [10]. Finally, the trial vector undergoes the *selection* operator, according to which, it is accepted as a member of the population of the next generation only if it yields a reduction in the value of the objective function f relative to that of the target vector. Alternatively, the target vector is retained in the next generation. The search operators efficiently guide the population to search for an optimum and focus on the most promising regions of the solution space. A more comprehensive description of the DE can be found in [7], [9]–[11].

More specifically, for each individual x_g^i, $i = 1, 2, \ldots, NP$, where g denotes the current generation, the mutant individual v_{g+1}^i can be generated through several mutation strategies [11] with different characteristics. The most known and widely used mutation strategy acts in accordance with the following equation:

$$v_{g+1}^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}), \tag{2}$$

where $F > 0$ is a real parameter, called *mutation constant* and $r_1, r_2, r_3, r_4, r_5 \in \{1, 2, \ldots, i-1, i+1, \ldots, NP\}$, are random integers mutually different and not equal to the running index i. The *mutation constant*, controls the impact of the difference between the last two individuals and is mainly responsible for the convergence rate of the algorithm [9].

In turn, the recombination operator is applied to further increase the diversity of the population. The outcome of the recombination operation are trial vectors which are a combination of the mutant individuals with other predetermined individuals, called the target individuals. In detail, for each component l $(l = 1, 2, \dots, D)$ of the mutant individual v^i_{g+1}, we uniformly choose a real number r in the interval $[0, 1]$ and compare this number with the predefined *recombination constant*, CR. If $r \leqslant CR$, we select, as the l–th component of the trial individual u^i_{g+1}, the l–th component of the mutant individual v^i_{g+1}. Otherwise, the l–th component of the target vector x^i_g becomes the l–th component of the trial vector. Finally, the trial individual is accepted for the next generation only if it reduces the value of the objective function at hand (selection operator).

In this context, we try to tackle the optimization problem defined in Definition 3.2, where the optimization search space is the space of all possible projections, i.e. $\mathcal{U} \subset R^a$. It should be noticed that we do not constrain the individuals of the population to lie in $\mathcal{U}$. Instead, we let them lie in $\mathbb{R}^a$. However, when we evaluate the $QC(\cdot)$, we transform the trial individuals u^i_g to $\acute{u}^i_g = u^i_g / \|u^i_g\|$ and evaluate $QC(\acute{u}^i_g)$ instead.

V. The Proposed Clustering Algorithm

In this Section, we introduce a new algorithmic scheme based on the principles of the dePDDP algorithm. The new technique utilizes the depth quality criterion proposed in Section III to guide a projection pursuit method. As already mentioned, for finding the best depth direction over the space of all possible ones, we use the DE optimization algorithm described above. After projecting the data onto the direction of maximum *depth* the algorithm splits them based on the global minimizer x^*. More specifically, the new divisive hierarchical clustering algorithm, given the name DBPPC (Density Based Projection Pursuit Clustering) utilizes the following criteria:

- (Stopping Criterion) ST: Let $\Pi = \{\{\mathcal{C}_i, P_i\},\ i = 1, \dots, k\}$ a partition of the data set $\mathcal{D}$ into k sets $\mathcal{C}_i$, and the assorted projections P_i of them onto the direction of maximum *depth*. Let $\mathcal{X}$, be the set of minimizers x^*_i of the density estimates $\hat{f}(x^*_i; h)$ of the projection P_i of the data of each $C_i \in \Pi$, $i = 1, \dots, k$. Stop the procedure when the set $\mathcal{X}$ is empty.
- (Cluster Selection Criterion) CS: Let $\Pi = \{\{\mathcal{C}_i, P_i\},\ i = 1, \dots, k\}$ a partition of the data set $\mathcal{D}$ into k sets $\mathcal{C}_i$, and the assorted projections P_i of them onto the direction of maximum *depth*. Let $\mathcal{F}$ be the set of the density estimates $f_i = \hat{f}(x^*_i; h)$ of the minimizers x^*_i for the projection P_i of the data of each $C_i \in \Pi$, $i = 1, \dots, k$. The next set to split is C_j, with $j = \arg\min_i\{f_i : f_i \in \mathcal{F}\}$.
- (Splitting Criterion) SPC: Let $\hat{f}'(x; h')$ be the kernel density estimation of the density of the projections $p_i \in \mathcal{P}$, and $x^\star$ its global minimizer. Then construct $P_1 = \{d_i \in \mathcal{D} : p_i \leqslant x^\star\}$ and $P_2 = \{d_i \in \mathcal{D} : p_i > x^\star\}$.

Algorithm 1 The DBPPC algorithm summary.

```
Set $\Pi = \{\mathcal{D}\}$
repeat
    Select a set $\mathcal{C} \in \Pi$, using cluster selection criterion $CS$
    Split $\mathcal{C}$ into two sub-sets $\mathcal{C}_1$ and $\mathcal{C}_2$ using Splitting Criterion $SPC$
    Remove $\mathcal{C}$ from $\Pi$ and set $\Pi \to \Pi \cup \{\mathcal{C}_1, \mathcal{C}_2\}$
until Stopping Criterion $ST$ is not satisfied
Return $\Pi$ the partition of $\mathcal{D}$ into $|\Pi|$ clusters
```

Based on that criteria Algorithm 1 reports the complete algorithmic scheme.

VI. Experimental Results

In this section, we perform an experimental evaluation of the proposed clustering algorithm. At the first part of our experimental analysis, we employ a series of simulated datasets to examine the performance of the proposed methodology. This gives the opportunity to pre-design (and hence know beforehand) the structure of the data that the clustering procedure aims to recover. This kind of artificial cluster construction method is typically used in similar empirical evaluations [12], [13].

In an attempt to construct datasets which are constituted by clusters of random shapes, we employ the following procedure. The actual clusters are composed by independent univariate Beta distributions, one for each dimension, of which the shape parameters are drawn at random uniformly in a specified interval. After drawing 100 points for each cluster, the data of each one is rescaled by a random factor and subsequently randomly repositioned. This data generation mechanism generates clusters of random shapes depending on the values of the parameters of the Beta distributions. We will refer to this data generation mechanism as $DSET_{\mathrm{Beta}}$.

To assess the quality of a data partition, we use the class labels which are not available to the clustering algorithms. We measure the degree of correspondence between the resulting clusters and the classes of each object. In detail, let $\mathcal{L}$ be the set of class labels $l_i \in \mathcal{L}$, for each point $d_i \in \mathcal{D}$, $i = 1, \dots, n$, with l_i taking values in $\{1, \dots, L\}$ we define the purity of a k-cluster partitioning as $\Pi = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$. The purity of Π is defined by the following formula:

$$p(\Pi) = \frac{\sum_{j=1}^{k} \max\{|\{p_i \in \mathcal{C}_j : l_i = 1, \dots, L\}|\}}{n}, \qquad (3)$$

so that $0 \leq p(\Pi) \leq 1$. High values indicate that the majority of vectors in each cluster come from the same class, so in essence the partitioning is "pure" with respect to class labels.

However, cluster purity does not address the question of whether all members of a given class are included in a single cluster and therefore is expected to increase monotonically with the number of clusters in the result. For this reason, criteria like the V-measure [14] have been proposed. The V-measure tries to capture cluster homogeneity and completeness, which summarizes a clustering solution's success in

TABLE I
Mean purity for the generated datasets (with the observed standard deviation in parenthesis)

Dimension 5					
No. Of Cl.	dePDDP	k-means	DBSCAN	GMM	DBPPC
5	0.90 (0.23)	0.97 (0.05)	0.52 (0.23)	0.99 (0.02)	0.98 (0.05)
25	0.34 (0.15)	0.86 (0.03)	0.06 (0.03)	0.88 (0.04)	0.86 (0.04)
Dimension 25					
No. Of Cl.	dePDDP	k-means	DBSCAN	GMM	DBPPC
5	1.00 (0.17)	0.96 (0.08)	0.90 (0.10)	0.96 (0.07)	1.00 (0.00)
25	0.80 (0.39)	0.88 (0.03)	0.05 (0.02)	0.86 (0.04)	1.00 (0.00)

TABLE II
Mean V-measure for the generated datasets (with the observed standard deviation in parenthesis)

Dimension 5					
No. Of Cl.	dePDDP	k-means	DBSCAN	GMM	DBPPC
5	0.91 (0.19)	0.95 (0.06)	0.56 (0.33)	0.98 (0.04)	0.94 (0.05)
25	0.44 (0.21)	0.90 (0.02)	0.04 (0.07)	0.93 (0.02)	0.88 (0.03)
Dimension 25					
No. Of Cl.	dePDDP	k-means	DBSCAN	GMM	DBPPC
5	1.00 (0.22)	0.97 (0.05)	0.95 (0.05)	0.96 (0.06)	0.98 (0.02)
25	0.82 (0.42)	0.96 (0.01)	0.03 (0.05)	0.94 (0.01)	0.98 (0.01)

including every point of a single class and no others. Again, high values corresponds to better performance. For details on how these are calculated, the interested reader should refer to [14].

We compare the performance of the proposed clustering algorithm against four well known clustering algorithms, namely dePDDP [6], k-means [15], DBSCAN [16], GMM. Table I reports the purity and Table II the V-measure of the algorithms in 100 randomly generated datasets, using the described $DSET_{\mathrm{beta}}$ mechanism. The clustering algorithms have been implemented in the Matlab environment. For the k-means algorithm, we employ Matlab's k-means functions. For the DBSCAN algorithm the eps (neighborhood radius) parameter was set to the default value given in [16] and the k (number of objects in a neighborhood of an object) parameter was set to 5. The density estimation of the projected data in the dePDDP and the DBPPC algorithms is calculated using the Fast Gauss Transform [17]. As proposed in [6], the bandwidth parameter for the density was set by choosing a multiple of the h_{opt} bandwidth ("normal reference rule"), which is the bandwidth that minimizes the Mean Integrated Squared Error (MISE). This is given by:

$$h_{opt} = \sigma \left(\frac{4}{3n} \right)^{1/5},$$

where σ is the standard deviation of the data. The multiple was set to 4 for these experiments.

To facilitate a more direct understanding of the results, we will use two 2-dimensional datasets constructed with the $DSET_{\mathrm{beta}}$ mechanism and will resort to visual inspection (Figures 8 and 9). As shown for the dataset of Figure 8 only the DBPPC algorithm manages to retrieve all the actual clusters. Although more than 3 clusters have been retrieved, none of the sub-clusters contain elements that belong to more than one actual cluster. In Figure 9, the second dataset is a typical case, where projecting the data onto the principal direction is not considered to be a good projection and as expected dePDDP fails to split the data. On the other hand, the DBPPC algorithms manage to split the data effectively.

A. Real Data Application

In this section, we study the performance of the proposed method against the aforementioned clustering algorithms in real world applications. For this purpose, we employ two biomedical datasets from the UCI Machine Learning Repository [18], the Breast Canser dataset and the Vertebral dataset, and two microarray datasets, the Leukemia [19] and the Lymphoma [20] datasets. A brief description for each dataset is reported below.

- (BREAST CANCER): This breast cancer database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. There are 369 instances in this dataset, described though 10 features. Each instance has one of 2 possible classes: benign or malignant. There are 16 instances that contain a single missing (i.e. unavailable) attribute value that we arbitrary set to 0.
- (VERTEBRAL): This biomedical data set was built by Dr. Henrique da Mota during a medical residence period in the Group of Applied Research in Orthopaedics (GARO) of the Centre Médico-Chirurgical de Réadaptation des Massues, Lyon, France. There are 310 instances in this dataset that corresponds to patients, described though 6 biomechanical attributes. Each patient belongs to one out of three categories: Normal (100 patients), Disk Hernia (60 patients) or Spondylolisthesis (150 patients).
- (LEYKEMIA): The dataset is the one used by Handl *et al.* [19] in their survey of computational cluster validation to illustrate the use of some measures. It is a 38×100 data matrix, where each row corresponds to a patient with acute leukemia and each column to a gene. For this dataset, there are three actual clusters.
- (LYMPHOMA): The dataset comes from the study of Alizadeh *et al.* [20] on the three most common adult lymphoma tumors. It is an 80×100 matrix, where each row corresponds to a tissue sample and each column to a gene. There are three clusters in the dataset. The dataset has been obtained from the original microarray experiments as described by Dudoit and Fridlyand in [21].

For this experiment the actual number of clusters was also given as input to the dePDDP and DBPPC algorithms and the multiple value for the bandwidth parameter was set recursively to 4. If the algorithm cannot split the initial dataset, we decrease this parameter by $1/4$. As shown at Table III, the DBPPC algorithm's performance remains high in all cases.

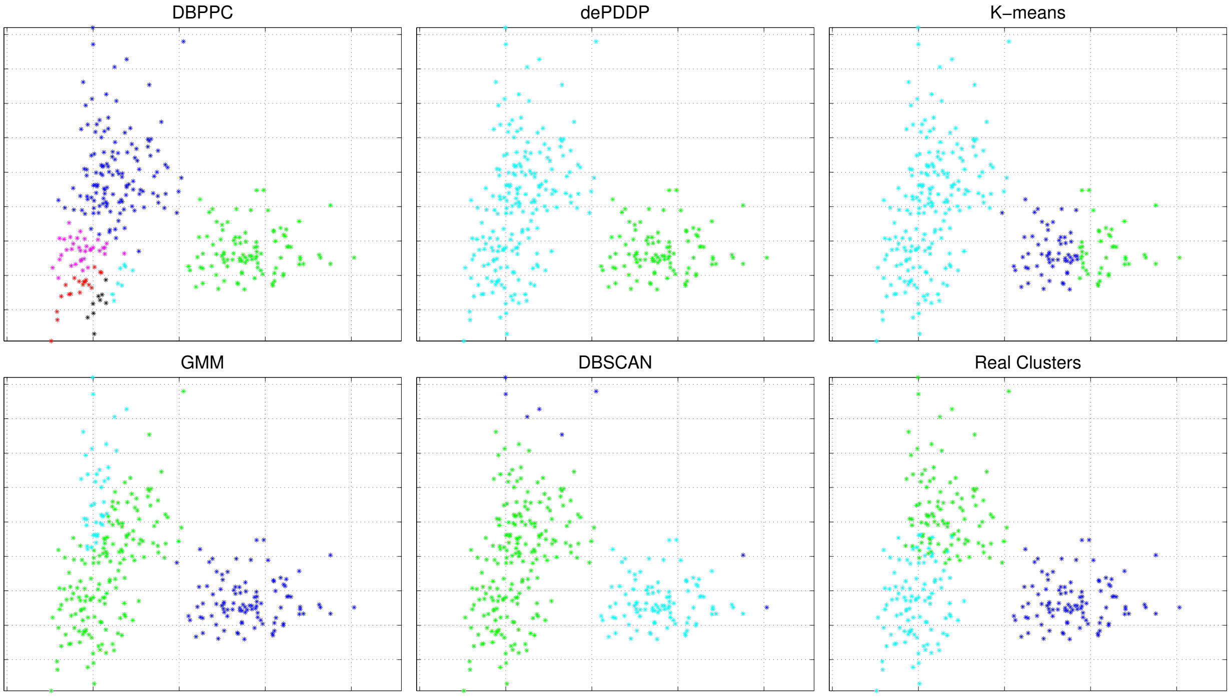

Fig. 8. Clustering Results for a two dimensional dataset

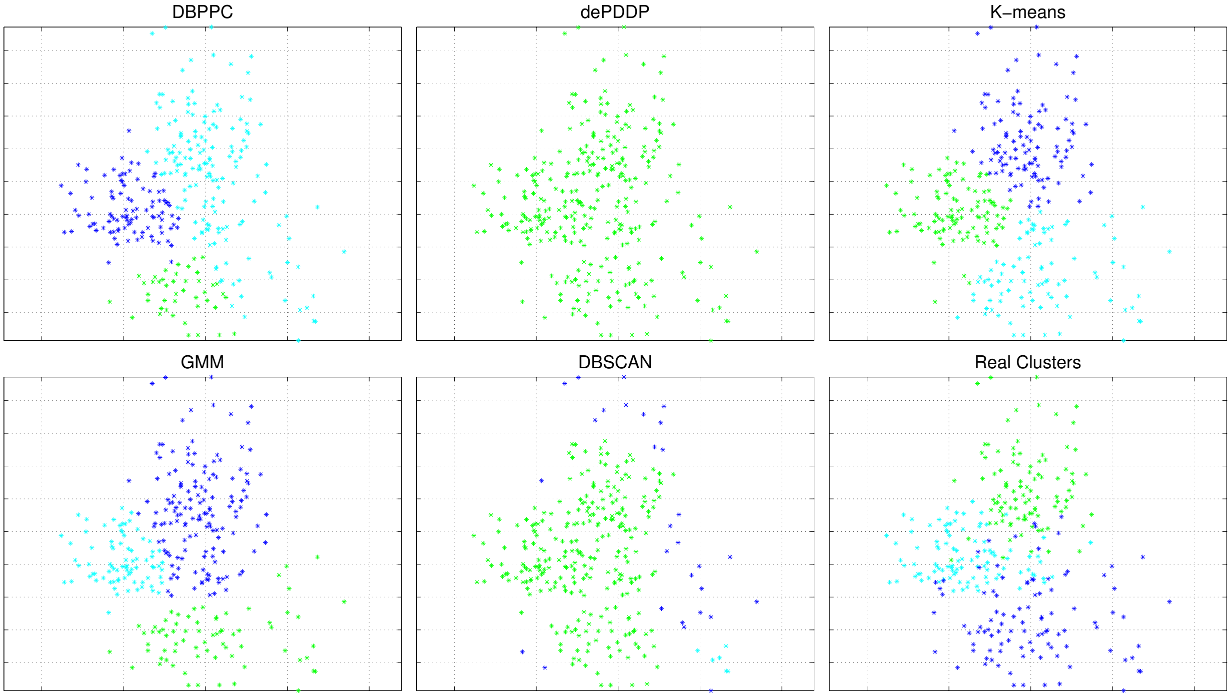

Fig. 9. Clustering Results for a two dimensional dataset

For the Vertebral and the Leukemia datasets, since the DBPPC splits only a few outliers at the first algorithmic steps, we let the algorithm retrieve a few more than the actual clusters. Note that this is not an uncommon procedure for this type of clustering algorithms. To have comparable results, we assign the same number of clusters as input to all methods. It is important to note that dePDDP algorithm although it is very effective for the first three cases, it does not manage to split the Vertebral dataset at all. GMM is producing good results as well, but it is unable to operate on the first two datasets, because of their dimensionality. On the other hand, DBPPC performs efficiently in comparison with the other methods. It

TABLE III
RESULTS WITH RESPECT TO THE MEAN CLUSTERING PURITY AND V-MEASURE (WITH THE OBSERVED STANDARD DEVIATION IN PARENTHESIS)

Dataset		Leukemia	Lymphoma	Breast-Cancer	Vertebral
Classes		3	3	2	3
dePDDP	Cl.	4	3	3	...
	Pur.	0.9737	0.8375	0.9714	...
	V-m.	0.8369	0.5885	0.8051	...
k-means	Cl.	4	3	2	5
	Pur.	0.9695(0.01)	0.8413(0.04)	0.9585(0.00)	0.7383(0.01)
	V-m.	0.8201(0.02)	0.5826(0.09)	0.7361(0.00)	0.4023(0.00)
DBSCAN	Cl.	2	2	2	2
	Pur.	0.6053	0.5250	0.7883	0.4839
	V-m.	0.2782	0.1088	0.2614	0.0044
GMM	Cl.	...	...	2	3
	Pur.	...	...	0.8741(0.00)	0.7678(0.00)
	V-m.	...	...	0.5553(0.00)	0.4540(0.03)
DBPPC	Cl.	4	3	2	5
	Pur.	0.9395(0.01)	0.8750(0.00)	0.9605(0.00)	0.7461(0.02)
	V-m.	0.7010(0.03)	0.6924(0.03)	0.7470(0.03)	0.4937(0.05)

is notable that the DBPPC algorithm's performance remains stable across all the different datasets, while all the other considered methods, at least one of the cases, fail to retrieve good results.

VII. CONCLUSION

Clustering of high dimensional data is a topic of main interest in several research areas such as Bioinformatics and Text Mining. Typically, algorithms in order to deal with such datasets, project the high dimensional data onto a lower dimensional space. To find suitable projections, methods like projection pursuit and PCA have been developed. In the same theme, here we introduce a new measure of interestingness (quality criterion) of projection directions and for each problem we use the Differential Evolution algorithm to optimize it. Finally, a new clustering algorithm is proposed with promising performance in simulated and real world clustering applications.

In a future research we intend to investigate the theoretical aspects of the proposed method and to extend the application domain of the new algorithm in additional real world problems.

ACKNOWLEDGMENT

The authors would like to thank the European Union (European Social Fund ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: "Heracleitus II. Investing in knowledge society through the European Social Fund" for financially supporting this work.

REFERENCES

[1] R. Bellman, *Adaptive control processes: A guided tour.* Princeton university press Princeton, NJ, 1961.

[2] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Trans. Comput.*, vol. 23, pp. 881–890, September 1974.

[3] P. J. Huber, "Projection pursuit," *Annals of Statistics*, vol. 13, no. 2, pp. 435–475, 1985.

[4] M. C. Jones and R. Sibson, "What is projection pursuit?" *Journal of the Royal Statistical Society. Series A (General)*, vol. 150, no. 1, pp. 1–37, 1987.

[5] A. Jain and R. Dubes, *Algorithms for Clustering Data.* Prentice Hall, 1988.

[6] S. Tasoulis, D. Tasoulis, and V. Plagianakos, "Enhancing Principal Direction Divisive Clustering," *Pattern Recognition*, vol. 43, pp. 3391–3411, 2010.

[7] R. Storn and K. Price, "Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[8] S. K. Tasoulis, D. K. Tasoulis, and V. P. Plagianakos, "Evolutionary principal direction divisive partitioning," in *IEEE World Congress on Computational Intelligence*, 2010.

[9] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series).* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[10] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[11] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 99–119, 2011.

[12] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data: Recent Advances in Clustering*, J. Kogan, C. Nicholas, and M. Teboulle, Eds. Berlin: Springer, 2006, pp. 25–72.

[13] M. Nilsson, "Hierarchical Clustering Using Non-Greedy Principal Direction Divisive Partitioning," *Information Retrieval*, vol. 5, no. 4, pp. 311–321, 2002.

[14] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 410–420.

[15] J. A. Hartigan and M. A. Wong, "A K-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100–108, 1979.

[16] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *KDD'96*, 1996, pp. 226–231.

[17] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis, "Improved fast gauss transform and efficient kernel density estimation," in *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, 2003, pp. 664–671.

[18] C. Blake and C. Merz, "UCI repository of machine learning databases," 1998.

[19] J. Handl, J. Knowles, and D. B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics*, vol. 21, pp. 3201–3212, August 2005.

[20] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, and et al., "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling." *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.

[21] S. Dudoit and J. Fridlyand, "A prediction-based resampling method for estimating the number of clusters in a dataset." *Genome Biology*, vol. 3, no. 7, p. RESEARCH0036, 2002.