Knowledge Discovery in High Dimensional Data

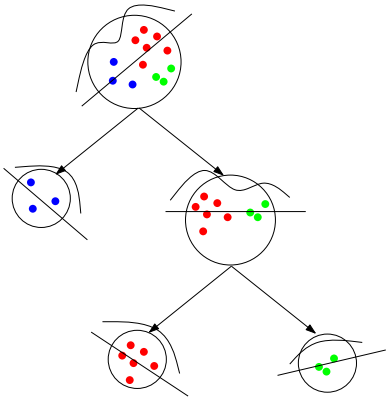
Sotiris Tasoulis

A thesis submitted for the degree of

Doctor of Philosophy

Department of Computer Science and Biomedical Informatics

University of Thessaly



Supervisor

Assistant Professor Vassilis Plagianakos

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

This Ph.D. thesis has been co-financed by 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".

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

This Ph.D. thesis has been prepared using the program LaTeX and the style provided by http://www.vincentgarcia.org/goodies_thesistemplate.php. Writing was made using the program "Kile" on the operating system "Linux Mint". The programming languages used for the development of the presented methods are MATLAB™, Octave and C++. The figures were created using the programs "Gnuplot", "Matlab", "Libreoffice" and "Ipe".

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Contents

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Acknowledgments

To this end I would like to thank the people that helped me completing this thesis. First of all I would like thank my supervisor Dr. Vassilis .P Plagianakos for his guidance, inspiration and understanding. The insistence of Vassilis to accomplish the most difficult tasks gave me strength and courage many times through these years. In addition I would like to thank my advisor Dr. Ilias Maglogiannis for his crucial input in my research work. A very special thanks goes out to Dr. Nial Adams, with whose motivation and encouragement I managed to set high standards in my research work.

My sincere thanks also goes to my strongest collaborator, best friend and brother Dr. Dimitris Tasoulis. He has been an inspiration for many scientists and it is my honour to be one amongst them. I must also express my gratitude to my friends and collaborators Dr. Michalis Epitropakis, Dr. Nicos Pavlidis, Charalambos Doukas and Spiros Georgakopoulos. Completing this work would have been much more difficult without their support and friendship.

Finally, I would also like to thank my family for the support they provided me all these years and in particular I would like to thank my wife Rania. Rania helped me overcome all the difficulties that came up, even those in Mathematics.

Sotiris K. Tasoulis
Lamia, 2013.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Abstract

The most exciting phrase to hear in science, the one that heralds new discoveries, is not "Eureka!" but "That's funny..."

—Isaac Asimov

While data clustering has a long history and a large amount of research has been devoted to the development of numerous clustering techniques, significant challenges still remain. One of the most important of them is associated with high data dimensionality. A particular class of clustering algorithms has been very successful in dealing with such datasets, utilizing information driven by dimensionality reduction techniques. Projection methods for dimension reduction have enabled the discovery of otherwise unattainable structure in ultra high dimensional data. In this thesis, we try to deepen our understanding on what can be achieved by this kind of approaches in an attempt to theoretically discover the relationship between true clusters in the data and the distribution of their projection. Based on such findings, we propose a series of new hierarchical divisive clustering algorithms. The proposed algorithms require minimal user-defined parameters and have the desirable feature of being able to provide approximations for the number of clusters present in the data. The experimental results indicate that the proposed techniques are effective in simulated data scenarios and as well in real world problems that are affected by high dimensionality.

This thesis is constituted by two major parts. First Part is devoted to the introductory material and the basic concepts, while the second Part contains an analytic overview of the proposed methodologies. More precisely in Chapter 1 an introduction to the Knowledge Discovery process is presented and Chapter 2 refers to the necessary background information. Subsequently in Chapter 3 the most well know clustering methodologies are presented, while Chapter 4 is devoted to the Knowledge Discovery of High Dimensional Data. In what follows, the new methods for clustering of high dimensional data are presented, while Chapter 5 is devoted to the devel-

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

opment of methods that can deal with ultra high dimensionality. Chapter 7 refers to the use of alternative methods for dimensionality reduction and finally Chapter 8 focus on the clustering of high dimensional data streams. The thesis ends with brief summary of the new algorithms and conclusions.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Part

Introduction and Basic Concepts

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

- Chapter 1 -

Introduction

I will not define time, space, place and motion, as being well known to all.

—*Sir Isaac Newton*

Knowledge Discovery in Databases (KDD) can be defined as the automatic analysis of data. More specifically KDD is the process used to identify patters from large datasets. The most important part of the KDD process is "Data Mining" which refers to the development of algorithms that discover unknown patterns in the data.

§ 1.1 Knowledge Discovery

The knowledge discovery process is iterative and interactive, consisting of the nine following steps [MR10]:

- Developing an understanding of the application domain.

  This is a preparation step where the goal and the environment of the knowledge discovery process need to be defined. During the KDD procedure this step may be revised.

- Selecting and creating a data set on which discovery will be performed.

  Having defined the KDD goals, the data preprocessing may begin. The available data have to be integrated into one dataset. Also the attributes that will be used in the process need to be determined.

- Preprocessing and cleansing.

  The data reliability need to be enhanced. For this reason the missing value issues are addressed and the noise or outlying points are

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

removed. To achieve this, usually data mining or statistical methods are employed.

- Data transformation.

  In this step, the data are transformed in order to be used better from the data mining method. The data transformation includes the dimensionality reduction through methods like feature selection and feature extraction, and the attribute transformation. This is a very important step for the overall success of the KDD process, but in most cases it is also very project specific. The steps that follows are focused on the data mining part of the process.

- Choosing the appropriate Data Mining task.

  In this stage, the choice of which type of data mining (e.g. regression, classification, clustering) to use, have to be made. This choice is mainly based on the goal of the KDD process.

- Choosing the Data Mining algorithm.

  As long as we have decided on the strategy, the decision for the specific method to be used have to be made.

- Employing the Data Mining algorithm.

  This step deals with the implementations of the data mining algorithm. The algorithm may need to be applied many times until the desired result is achieved.

- Evaluation.

  In this stage, the mined patterns are evaluated with respect to the defined goals

- Using the discovered knowledge.

  The knowledge now can be incorporated into another system for further action. Based on this knowledge we can make changes to the system and measure the effects. The success of this last step determines the efficiency of the KDD process.

§ 1.2 Data Mining

The increasingly availability of large amounts of data in the recent years, have made necessary the use of data mining methods for analysis. Data Mining (DM) can be defined as the extraction of knowledge from large amount of data. The goal of data mining is to find patterns in the data that was previously unknown. To achieve this data mining techniques employ the

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

use of analysis tools from statistical models, mathematical algorithms, and machine learning methods. Data mining methods can be divided into two categories.

- Supervised data mining

  These techniques predict a hidden function using training data. The training data is constituted by the by a set of input variables and their corresponding class labels. The results of the methods are the unknown class labels of the input data. The most well known examples of supervised data mining are classification and prediction.

- Unsupervised data mining

  These techniques make an attempt to identify patters in the data without the use of a training set. The most representative method of unsupervised data mining is clustering.

§ 1.3 Machine Learning

Learning is the process of developing a model, based on knowledge discovered from a data set. Machine Learning can be formally defined as the complex computation process of automatic pattern recognition and intelligent decision making based on input data. Machine learning is often confused with data mining as they both usually employ similar methods and in many cases overlap. In order to distinguish machine learning and data mining we can use the following definitions. Machine learning focuses on prediction, based on known properties learned from the training data. Data mining focuses on the discovery of (previously) unknown properties on the data. Data mining often uses machine learning methods, but by focusing on a different goal, while machine learning often employs methods from the area of data mining as unsupervised learning or as a preprocessing step in order to enhance the learner accuracy.

§ 1.4 Data Clustering

Data clustering is a central component of the knowledge discovery process. Formally it can be defined as *"the process of partitioning a set of data vectors into disjoint groups (clusters), so that objects of the same cluster are more similar to each other than objects in different clusters"*. The greater the similarity (or homogeneity) within a group and the greater the difference between groups, the better or more distinct the clustering. The modern roots of data clustering date back to 1939 [Try39], but there are references to it from antiquity. Clustering is related to Classification in the sense that

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

it creates a labelling of the data points , however it derives these labels only for the data. On the other hand, classification uses information from data with known class labels to assign a class label to unlabelled data. For this reason in some cases data clustering is referred as "supervised or automatic classification". In addition more rarely data clustering is also referred as "numerical taxonomy" and "typological analysis".

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

- Chapter 2 -

Background Information

A mind needs books as a sword needs a whetstone, if it is to keep its edge.

—*George R.R. Martin, A Game of Thrones*

In this chapter the background information is provided regarding data clustering which is the area that this thesis is focused on.

§ 2.1 Steps of the Clustering Process

A complete clustering process is constituted by the following steps:

1. Representation of data (may include the feature extraction or selection).
2. Definition of the similarity measure between the elements.
3. Clustering
4. Evaluation of the results.

In figure 2.1 we can see the connection between the first three steps of the clustering process. The feedback loop is also included, according to which, the result of the procedure affects the previous steps in order to improve the results.

The term "representation of data" refer to the number and the type of features that describe each on of the data elements. Feature selection called the process that tries to select the most important features. On the other hand, feature extraction refers to the transformation of features to others that are more important with respect to the further clustering procedure.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

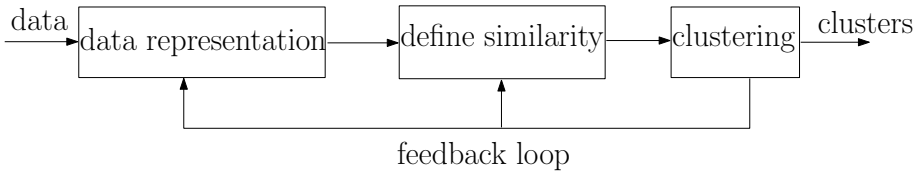


Figure 2.1: *The steps of the clustering process.*

A measure of similarity is being used to represent the similarity between the elements, and is usually defined between couples of elements. Several measures of similarity have been proposed that varies with respect to the corresponding application [WM97]. Measures of similarity will be further explained in Section 2.4.

The final clustering results can be exclusive where each data element is assigned to a single cluster, or fuzzy where every element belongs to every cluster with a membership weight. Moreover a clustering result can be a hierarchy, which is a set of nested clusters that are organised as a tree, or just as a division of the data set into non overlapping clusters. More details of the clustering techniques will be given in Chapter 3.

The evaluation of the result of a clustering algorithm is multifaceted. On one hand we have the evaluation of the data instead of the clustering result. A data set that does not contain a structure of clusters should not get processed by a clustering algorithm. The study of "Cluster tendency" where a data set is being examined before given to a clustering algorithm as input, does not get much attention in the recent years [JMF99]. For more information refer to [Dub87]. On the other hand, "the validity analysis" evaluates the clustering result. To achieve this, an optimality criterion is being used. However, in most cases these criteria are subjective, as such there are not any district rules. For more information regarding the evaluation of the results of a clustering algorithm refer to Section 2.7.

§ 2.2 Terminology

In this section we will refer to the terms that will be used in this thesis.

- an element (also referred as document, observation, pattern, or point) d is typically described as a a-dimensional vector, $d = (d^1, \ldots, d^a)$, where a is the number of features.
- The elements d^i of the observation d are called features, properties, variables or dimensions.
- A data set D is represented by an $n \times a$ data matrix whose each row represents a data sample d_i, for $i = 1, \ldots, n$.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

- A measure of similarity is a metric defined in the space of feature and is used to measure the similarity between the elements.

§ 2.3 Representation of data

In general there are not any particular specifications that suggest the optimal representation of data. Actually in most cases the data creation cannot be supervised. In that case the user is responsible to collect the data, and optionally, to choose or produce the features that will be used at the next steps of the procedure. Due to the difficulty of this task, the representation of the data is often consider to be known in advance. Although, this is a very important step of the procedure since the clustering result is heavily depended on the representation. For example in Figure 2.2 a simple two dimensional data set is illustrated where the elements are organised in one semicircular cluster. Choosing the Cartesian representation, many clustering algorithms would split the cluster due to its concave shape. On the other hand, if we choose to use the polar coordinate system it is easier to define that there is only one cluster in this data set.

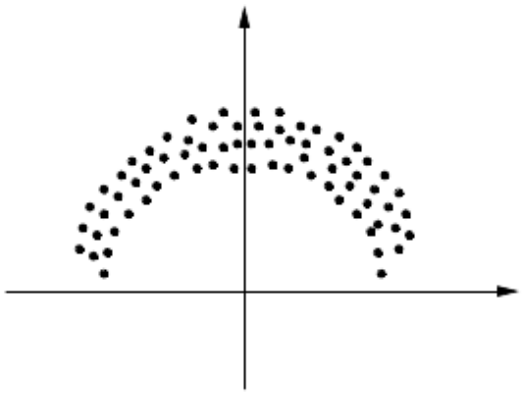


Figure 2.2: *A semi circular group of elements.*

An element from the data set can represent either a natural object (i.e. table), either an abstract concept (i.e. behaviour). The traditional representation of each data element is through a multidimensional vector, where each coordinate represents a particular feature [DH73]. These feature can be quantitative or qualitative. For example if the colour of the hair and the age are the features of a group of people, then the element (blond, 30), is the representation of a blond person of age 30. Feature can be separated in the following categories:

1. Quantitative feature:

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

(a) continues values (i.e. weight).

(b) district values (i.e. number of tables).

(c) values of spaces (i.e. length of a fact).

2. Qualitative features

(a) Unstructured (i.e. colour)

(b) Ordered (i.e. qualitative representation of age, young, middle aged, old).

The most important step of representation procedure is the selection of the most representative features. The feature selection techniques, select a subset of features, based on the optimisation of some proximity criterion of the clustering procedure. On the other hand, feature extraction techniques, like the Principal Component Analysis, transform the representation of data into a lower dimensional space.

§ 2.4 Measures of Similarity

As already mentioned, the definition of the similarity between the elements of the data set is fundamental for the clustering procedure. The choice of the similarity measure to use is a very difficult task, due to the variety of the type of features. Usually the similarity between two elements is measured through a distance function defined in the range values of the features.

The most widely used metric for continues features is the Euclidean distance,

$$dist_2(d_i, d_j) = \left(\sum_{k=1}^{a} |d_i^k - d_j^k|^2 \right)^{\frac{1}{2}} = \|d_i - d_j\|_2$$

which is a special case ($p = 2$) of the Minkowski distance,

$$dist_p(d_i, d_j) = \left(\sum_{k=1}^{a} |d_i^k - d_j^k|^p \right)^{\frac{1}{p}} = \|d_i - d_j\|_p$$

.

In general the algorithms that use the Euclidean distance perform better when the data set contains dense or isolated clusters [Jia94]. The disadvantage of the direct use of such techniques is that the similarity measure is dominated by the feature with the largest scale, however, this problem can be solved through normalisation of the data set or by attaching weights to the features. The linear correlation between the features can also alter such

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

measures. This can be avoided using a weight transformation to the data set or by employing the Mahalanobis metric:

$$dist_M(d_i, d_j) = (d_i - d_j)\sigma^{-1}(d_i - d_j)^\tau,$$

where τ is the covariance matrix of the data set, Mahalanobis distance is based on correlations between variables by which different patterns can be identified and analysed.

The computation of distances is not effective when some or all features are not continuous, since we cannot compare different types of features. However for this cause several metrics have been proposed.

§ 2.5 Types of Clusters

A cluster can be described in several ways based on the goal of data analysis. One of the most widely used type of clusters is the "center based clusters". The cluster is constituted by data points that are more close to the center of the cluster than to any of the centers of the other clusters. Next there is the "well separated clusters" type of clusters. In this case a cluster is constituted of data points that are more close to each other than to any data point not in the cluster. As expected this definition can only be satisfied when the actual clusters are sufficient far from each other. Another common type of clusters is the "graph based". In this case the cluster is constituted by data points that are connected to one another, but have no connection to data points of another cluster. An example of this type of clusters are the "contiguity based clusters", where two data points are consider connected only if their distance is lower than a specified value. This notion of a cluster can be very useful when dealing with cluster of irregular shapes but fails dramatically in the presence of noise. Finally there are the "density based" type of clusters where a cluster is defined by dense region of data points that is surrounded by a region of low density. This definition is widely used since it is not affected by the shape of the clusters or the existence of noise. At Figure 2.3 two dimensional data points are employed to visually illustrate examples of the described types of clusters.

§ 2.6 Hard Clustering and Fuzzy Clustering

In traditional (hard) clustering, data are divided into distinct clusters, where each data vector belongs to exactly one cluster. On the other hand, in fuzzy (soft) clustering, data points can belong to more than one cluster, and associated with each element is a set of membership levels. The membership level that defines the association of a data point with a cluster is between 0 (where the data point absolutely does not belong to the cluster) and 1 (where

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

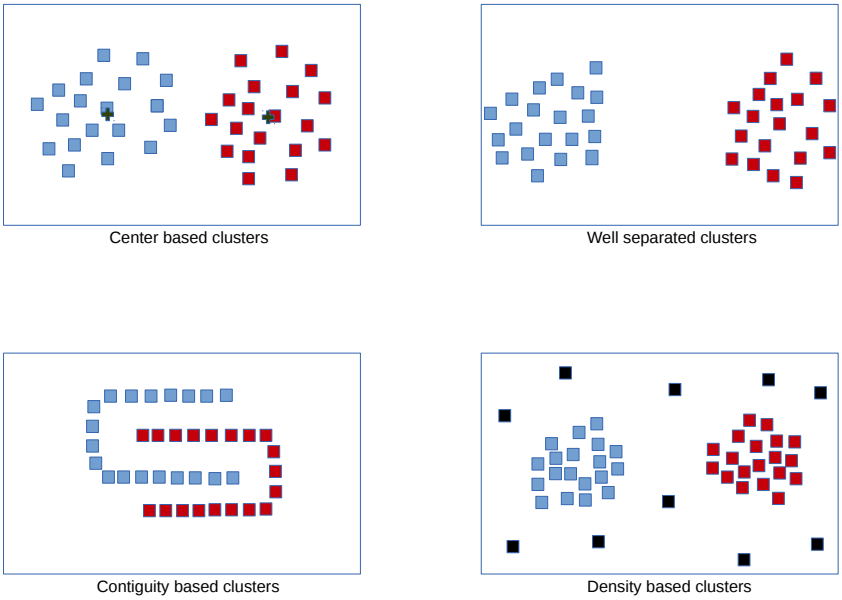


Figure 2.3: *Several cluster types.*

the data point absolutely belongs to the cluster) with the constraint that the sum of the weights for each data point must equal 1. These indicate the strength of the association between that data point and a particular cluster. Fuzzy clustering is a process of assigning these membership levels and then using them to assign data points to one or more clusters. In the same way, probabilistic clustering techniques can be used to compute the probability with which each data point belongs to each cluster with the similar constraint that the probabilities must sum to 1. Since both the probabilities and the membership weights must sum to 1, these types of clustering differs from the non-exclusive clustering, where a data point simultaneously belongs to multiple clusters. Instead, these approaches are mostly used to avoid the arbitrariness of assigning an object to only one cluster when it may be close to several. In many cases, fuzzy clustering is converted to hard clustering by assigning each data point to the cluster for which its membership weight is highest.

§ 2.7 Experiments and Evaluation

For the experimental evaluation of the algorithmic implementations presented in this following sections, a series of simulated datasets are employed. This procedure gives the opportunity to pre-design (and hence know beforehand) the structure of the data that the clustering procedure aims to recover.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

This kind of artificial cluster construction method is typically used in similar empirical evaluations [Nil02, KSI03, Ber06].

The datasets are constructed by drawing points from a finite mixture of k Gaussian distributions that represent the actual clusters in the data. The mean of each Gaussian is randomly placed in $[100, 200]^a$ and the covariance matrix is also randomly generated by an appropriate procedure, so as to ensure that it is symmetric and positive definite. Next, from each distribution 100 points are drawn, so the total number of points in each dataset is $k \times 100$. We will refer to this kind of data generation mechanism $DSET_{\text{Gausian}}$.

To extend the generality of the results, datasets has has also constructed in a similar manner using Beta distributions. In detail, 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 the interval $[1, 6]$. After drawing 100 points from each cluster, the data for each one is rescaled by a random factor (which is drawn uniformly in $[10, 20]$) and subsequently repositioned again randomly in $[100, 200]$. 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 $DSET_{\text{Beta}}$.

2.7.1 Evaluation

There are different way to assess the quality of a data partition [JD88, Kog07]. First, we could formulate quality as a function of the given data (internal criteria). In this case the clustering problem actually becomes an optimisation problem. A better method would be to use additional external information not available to the algorithm, such as class labels. In general there are three different techniques for evaluating the result of the clustering algorithms:

- External Criteria,
- Internal Criteria,
- Relative Criteria.

The internal and external criteria are based on statistical methods. The external validity methods evaluate the clustering based on some user specific intuition. The internal criteria are based on some metrics, which are based on data set and the clustering schema. Relative criteria are based on the comparison of the different clustering schema. The aim of the relative criteria is to choose the best clustering schema from the different results. For more information regarding validity analysis please refer to [Dub93, JD88].

When class labels are available a priori, external evaluation measures for clustering can be applied. The task of the clustering process then is to partition the data points to any number of clusters such that all data points

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

that belong to the same class constitute a single cluster. Since the actual classes are known it is trivial to determine whether a clustering algorithm has achieved a perfect clustering result or not. However, it is not straightforward to determine how far from perfect is an incorrect clustering result [Oak98].

Here we use external criteria to assess the quality of a data partition. As such, we can measure the degree of correspondence between the resulting clusters and the classes assigned a priori to each object. For a dataset $\mathcal{D}$, let $\mathcal{L}$ be a set of labels $l_i \in \mathcal{L}$, for each point $d_i \in \mathcal{D}$, $i = 1, \ldots, n$, with l_i taking values in $\{1, \ldots, L\}$. Let a k-cluster partitioning $\Pi = \{\mathcal{C}_1, \ldots, \ldots, \mathcal{C}_k\}$ with c_i talking values in $\{1, \ldots, C\}$. The purity of Π is defined as:

$$P(\Pi) = \frac{\sum_{j=1}^{k} \max\{|\{p_i \in \mathcal{C}_j : l_i = 1\}|, \ldots, |\{p_i \in \mathcal{C} : l_i = L\}|\}}{n}, \qquad (2.1)$$

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 [RH07] have been proposed.

The V-measure tries to capture cluster homogeneity and completeness, which summarises a clustering solution's success in including every point of a single class and no others. Homogeneity is satisfied when each one of the resulting clusters is constituted by data points of the same class. On the other hand completeness is satisfied when all data points that belong to a particular class are elements of the same cluster. Usually, for a clustering result, when homogeneity increases, the completeness decreases, For example the rare case where all data points are assigned into a single cluster, it is a case of perfect completeness, on the other hand, since this single cluster includes all classes, is consider to be totally inhomogeneous. In an another extreme case where each data point constitutes a cluster, the perfect homogeneity is achieved, but with respect to completeness this is a very poor result, with the exception of the case where each class contains only one element. The distance from a perfect clustering is measured as the weighted harmonic mean of the homogeneity and completeness measures.

Let A be the contingency table that represents the clustering result, such that $A = \{a_{ij}\}$ where a_{ij} is the number of data points that are members of class L_i and elements of the cluster C_j. Then homogeneity is defined as:

$$hm = \begin{cases} 1 & \text{if} \quad H(L,C) = 0 \\ 1 - \frac{H(L|C)}{H(L)} & \text{else} \end{cases} \qquad (2.2)$$

where

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

$$H(L|C) = -\sum_{c=1}^{|C|}\sum_{l=1}^{|L|}\frac{a_{lc}}{N}\log\frac{a_{lc}}{\sum_{l=1}^{|L|}a_{lc}}$$

and

$$H(L) = -\sum_{l=1}^{|L|}\frac{\sum_{c=1}^{|C|}a_{lc}}{n}\log\frac{\sum_{c=1}^{|C|}}{n}.$$

Completeness is symmetrical to homogeneity. As such completeness is defined as:

$$cm = \begin{cases} 1 & \text{if} \quad H(C,L) = 0 \\ 1 - \frac{H(C|L)}{H(C)} & \text{else} \end{cases}, \tag{2.3}$$

where

$$H(C|L) = -\sum_{l=1}^{|L|}\sum_{c=1}^{|C|}\frac{a_{lc}}{N}\log\frac{a_{lc}}{\sum_{C=1}^{|C|}a_{lc}}$$

and

$$H(C) = -\sum_{c=1}^{|C|}\frac{\sum_{l=1}^{|L|}a_{lc}}{n}\log\frac{\sum_{l=1}^{|L|}}{n}.$$

Based on these calculations the V-measure of Π is defined as:

$$V_\beta(\Pi) = \frac{(1+\beta)hc}{h\beta + c}, \tag{2.4}$$

where h and c are the homogeneity and the completeness of Π respectively and β is a positive real weight. If β is greater than 1, completeness is weighted more strongly in the calculation, while if β is less than 1, homogeneity is weighted more strongly. Again it holds that $0 \leq V_\beta(\Pi) \leq 1$, while higher values correspond to better performance. Here it is important to notice that the computation of V-measure is completely independent of the number of classes, the number of clusters and the size of the data set.

§ 2.8 Conclusions

Clustering deals with the separation of a data set into subclusters based on some similarity measure. In general clustering is considered to be a subjective process since the input data set needs to be clustered in different ways according to the goals of the corresponding application. This introduces several difficulties into the clustering process. A possible solution for this

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

matter is the representation of the subjective with knowledge. This knowledge is imported into various steps of the procedure in order to achieve the desired result. In this chapter, the various steps of the clustering process are presented.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

- Chapter 3 -

Clustering Algorithms

The roots of education are bitter, but the fruit is sweet.

—*Aristotle*

There exist many categorisations of clustering algorithms, but broadly we could divide them into three main categories:

- Hierarchical: Clustering algorithms that construct hierarchies of clusters in a top-down (agglomerative) or bottom-up (divisive) fashion. The former, start from n clusters, where n stands for the number of data points, each containing a single data point and iteratively merge the clusters satisfying certain measures of closeness. Divisive algorithms follow a reverse approach; starting with a single cluster containing all the data points and iteratively split existing clusters to subsets.
- Partitioning: Clustering algorithms [SKK00], start from an initial clustering (that may be formed at random) and subsequently create flat partitioning by iteratively adjusting the clusters based on the distance of the data points from a representative member of each cluster.
- Distance-Based: clustering algorithms create flat partitioning by considering neighbours of data points.

§ 3.1 Hierarchical Clustering Algorithms

A basic characteristic of the hierarchical methods is that the assignment of a data point to a cluster is permanent, as such it cannot be removed

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

from the cluster or merged with objects belonging to a different cluster. A hierarchical clustering can be displayed graphically using a tree diagram called dendrogram, which displays both the relationship between a cluster and its subclusters and the order in which the clusters were merged or split, in the agglomerative and divisive case, respectively. A simple example of the hierarchical procedure is illustrated in Figure 3.1.

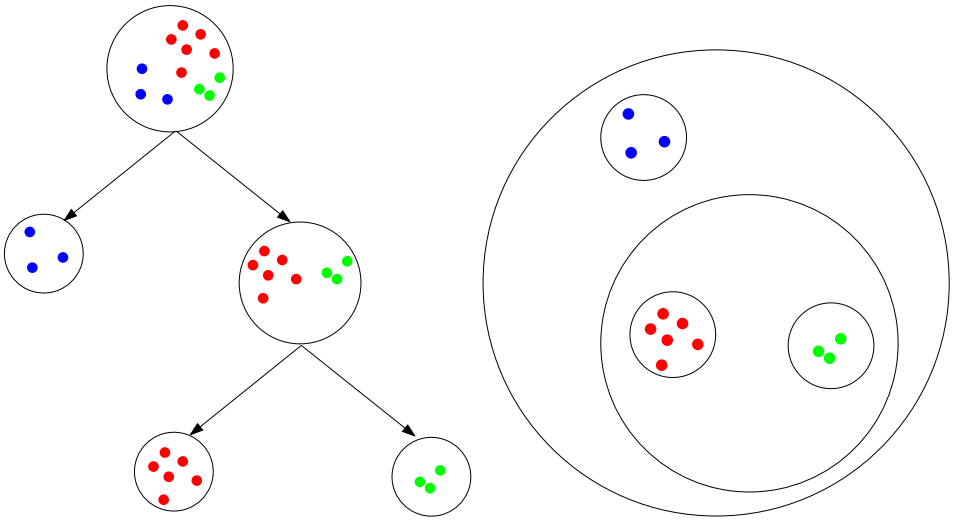


Figure 3.1: *A hierarchical clustering as a dendrogram (left) and as nested clusters (right).*

3.1.1 Hierarchical Agglomerative Methods

In general the steps of an agglomerative algorithm are the following:

1. Initially, the n objects are being split into n clusters and the symmetric proximity matrix $DIST = dist(d_i, d_j)$ is constructed.

2. We search $DIST$ for the closest clusters. Let K and L be the clusters that are more close to each other, then $min(DIST) = dist(d_K, d_L)$.

3. The clusters K and L are being merged and the new cluster is named KL. Also the matrix $DIST$ is updated by removing lines and columns that corresponds to K and L, and by adding a line and a column that displays the distances of the new cluster KL from the rest of the clusters.

4. Finally, repeat steps 2 and 3 $n-1$ times in total. At the final state of the algorithm all objects belongs to one clusters. The clusters created during the algorithmic process are being recorded along with the level of similarity of each merging.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Defining proximity between clusters The hierarchical agglomerative methods differentiates by the way they compute the proximity between two clusters. While the calculation of the distance between two elements is clear enough, the calculation of the distance between clusters that contain more than one elements can be done in many ways. Cluster proximity is typically defined with a particular cluster type in mind (see Section 2.5). For example many agglomerative hierarchical techniques, such as Single Link, Complete Link and Group Average, come from a graph based view of clusters. Single Link defines proximity between two clusters as the distance between the two closest points that are in different clusters. This yields contiguity-based clusters as shown in Section 2.5. On the other hand, Complete Link consider cluster proximity as the distance between the two farthest points that are in different clusters. One more graph based approach is the Group Average, which defines cluster proximity to be the average of all pairwise proximities of all points that belong to different cluster. In Figure 3.2 the above approaches are illustrated.

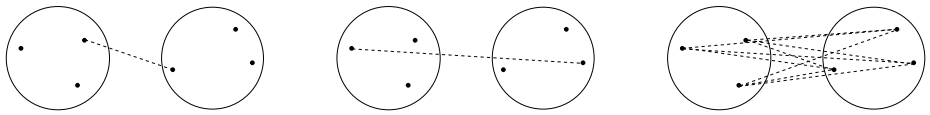


Figure 3.2: *The three graph based proximities described. Single Link (left), Complete Link (middle) and Group Averege (right).*

When each cluster is represented by a centroid (centroid method), the cluster proximity is defined as the proximity between cluster centroids.

The median method it was first proposed in order to alleviate some disadvantages of the centroid method. In the centroid method, if the sizes of the two groups to be merged are quite different, then the centroid of the new group will be very close to that of the larger group and may remain within that group. In the median method, the centroid of a new group is independent of the size of the groups that form the new group. On the other hand, the Ward's method, also assumes that a cluster is represented by a centroid, but it measure the proximity between two clusters in terms of the increase in the error sum of squares (ESS) that results from merging the two clusters.

The proximity calculation methods described above satisfy a recursive equation for the distance $d_{M(KL)}$ between the cluster M and the cluster KL which was created by merging cluster K and L at a previous step of the procedure. That is the following equation:

$$dist_{M(KL)} = \alpha_k dist_{MK} + \alpha_l dist_{ML} + \beta dist_{KL} + \gamma |dist_{MK} - dist_{ML}| \quad (3.1)$$

where $dist_{KL}$ the distance between the cluster K and L and α, β, γ the parameters that define equation 3.1.1 according to Table 3.1.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Single Link	$\alpha_K = \alpha_L = \frac{1}{2}$	$\beta = 0$	$\gamma = -\frac{1}{2}$
Complete Link	$\alpha_K = \alpha_L = \frac{1}{2}$	$\beta = 0$	$\gamma = \frac{1}{2}$
Group Average	$\alpha_K = \frac{n_K}{n_K+n_L}$, $\alpha_L = \frac{n_L}{n_K+n_L}$	$\beta = 0$	$\gamma = 0$
Centroid clust.	$\alpha_K = \frac{n_K}{n_K+n_L}$, $\alpha_L = \frac{n_L}{n_K+n_L}$	$\beta = -\alpha_K\alpha_L$	$\gamma = 0$
Median clust.	$\alpha_K = \alpha_L = \frac{1}{2}$	$\beta = -\frac{1}{4}$	$\gamma = 0$
Ward's method	$\alpha_K = \frac{n_M+n_K}{n_M+n_K+n_L}$, $\alpha_L = \frac{n_M+n_L}{n_M+n_K+n_L}$	$\beta = \frac{-n_M}{n_M+n_K+n_L}$	$\gamma = 0$

Table 3.1: *Parameters for Equation 3.1.1.*

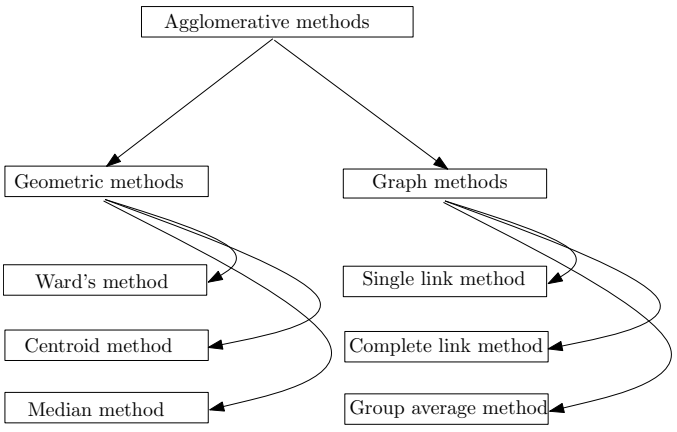


Figure 3.3: *Commonly used hierarchical agglomerative methods.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

3.1.1.1 BIRCH

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is a widely use agglomerative hierarchical algorithm proposed in [ZRL96] for clustering very large numerical data sets in Euclidean spaces. In the algorithmic procedure a Clustering Feature (CF) is used to summarise the information of each cluster. For a cluster C of an a-dimensional dataset, CF is defined as follows:

$$CF(C) = (|C|, S_1, S_2),$$

where $|C|$ is the number of instances in C and S_1,S_2 are a-dimensional vectors defined as:

$$S_1 = \sum_{d \in C} d = \left(\sum_{d \in C} x_1, \ldots, \sum_{d \in C} x_a \right),$$

$$S_2 = \sum_{d \in C} d^2 = \left(\sum_{d \in C} x_1^2, \ldots, \sum_{d \in C} x_a^2 \right),$$

where x_j $(1 \leq j \leq a)$ is the value of the j-th attribute of d. The algorithm initially builds a CF tree dynamically as the new data points are inserted. The parameters of the CF are the branching factor B, the leaf factor L and the threshold T. The nodes that are not leafs contain at most B subnodes of the form $[CF_i, child_i]$ while the leaf nodes contain a most L entries fo the form $[CF_i]$. Finally, the diameter of each entry in a leaf node has to be less than T. The determination of outliers is based on the density of each entry in a leaf node, entries of low density are considered to be outliers. Then the outliers are remove from the tree and are being stored to disk in order to reduce its size. Later during the process the outliers are scanned in order to examine if they can enter the tree form again without causing it to grow in size. Finally, when the CF tree is built, an agglomerative hierarchical clustering algorithm is applied to the nodes and a centroid is obtained for each cluster. The new clusters are formed by reassigning each data point to its nearest centroid.

The BIRCH algorithm has the advantage to perform very well in cases where clusters have spherical shape and uniform size, but it is not recommended for use with clusters of different sizes or irregular shapes.

3.1.1.2 CURE

Clustering Using Representatives (CURE) [KHK99] is an agglomerative hierarchical clustering algorithm that can deal with large data sets that contain nonspherical clusters of different sizes. The basic characteristic of this algorithm is that each cluster is represented by a set of points that is well

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

scattered in the cluster. CURE make use of random sampling and partitioning in order to handle large databases. The basic steps of the algorithmic procedure are the following:

1. Draw a random sample: To handle large databases a random sample is drawn. Here in order to analytically derive values for sample sizes, such that the probability of missing clusters is low, the Chernoff bounds [MR95] are used.

2. Partition the sample: In cases where the input sizes become large, a partitioning scheme is needed to speed up CURE. The samplespace is being partitioned into p partitions, each of size $\frac{s}{p}$, where s is the sample size.

3. Partially cluster the partitions: Each partition is clustered until the cluster number for each partition reduces to $\frac{s}{pq}$ for some constant $q \geq 1$.

4. Eliminate the outliers: The outliers do not merge with other points in the same rate as actual clusters due to the greater distances from other points and for this reason grow much slower in agglomerative hierarchical clustering.

5. Cluster the partial clusters: A second clustering pass is run on the $\frac{s}{p}$ partial clusters for all the partitions.

6. Label the data: Each of the remaining data points is assigned to the cluster containing the representative point closest to it.

The complexity of the cure algorithm is $O(n^2 \log n)$, but it can be reduced to $O(n^2)$ for the 2-dimensional case.

3.1.2 Hierarchical Divisive Methods

A divisive hierarchical method start the algorithmic procedure with one cluster that contains all the data points and splits this cluster. After the initial separation the data points move from one group to another or more sophisticated splits are executed on the already formed clusters [DM84]. There are two division techniques:

- Monothetic: A cluster is characterised monothetic when all of its elements have approximately the same value with respect to a particular variable. A monothetic method divides the data on the basis of the possession or otherwise of a single specified attribute.

- Polythetic: A cluster is characterised polythetic when all of its elements have approximately the same value with respect to all variables. A polythetic method divides the data based on the values taken by all attributes.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Hierarchical clustering algorithms have been shown to result in high quality partitions especially for applications involving clustering text collections. Nonetheless, their high computational requirements, usually prevents their usage in real-life applications, where the number of samples and their dimensionality is expected to be high (the computational cost is quadratic to the number of samples).

Amongst the class of divisive hierarchical clustering algorithms the Principal Direction Divisive (PDDP) algorithm is of great value due to its very low computational complexity, in comparison with other algorithms of the same class. We will further analyse PDDP algorithm in Section 4.1.5.

§ 3.2 Partitioning Clustering Algorithms

The partitioning clustering algorithm create a flat partitioning of the data instead of building hierarchies. The main advantage of this class of algorithms is that they can be applied on very big data sets.

The most popular of partitioning clustering algorithms is k-means, which starting from k centres iteratively assigns each data point to the cluster whose centroid minimises the Euclidean distance from the point [HW79]. The iteration terminates when none of the data points changes clusters, or equivalently, when the centroids do not change significantly. Spherical k-means [DM01] is a recently proposed modification of the algorithm that reduces k-means to the partitioning of the unit hypersphere by normalising the data points. Algorithms that belong to the same class as the k-means, can give adequate clustering results at low cost, since their running time is proportional to $k \cdot n$. However, their results depend heavily on their initialisation. Another similar approach is Gaussian Mixture Models (GMM) [MP00], where k multivariate normal density components are combined, by assuming that each component represents a cluster. Like k-means, an iterative algorithm is used, typically Expectation Maximization (EM), to fit the parameters of each density to the data. Then the posterior probabilities for each data point to each component of the model is indicative of the probability of the point belonging to each cluster. GMM may be more appropriate than the k-means clustering algorithm, when clusters have different sizes and there exists correlated variables, but more control parameters need to be estimated. This makes their application in high dimensions almost prohibitive.

3.2.1 The k-means algorithm

The basic concept of k-means algorithm is that each cluster is represented by a particular point named center. To split the data set into k clusters, the k centers $P_j, j = 1, \ldots, k$ are initialised randomly. Then, the algorithm

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

assigns each data point to the cluster whose center is more close to it. This assignment is based on on the following equation for every point d_i:

$$\mu_j(d_i) = \begin{cases} 1 & \text{if} \quad \|d_i - P_j\| \leq \|d_i - P_l\| \quad \forall l \neq j \\ 0 & \text{else} \end{cases} . \tag{3.2}$$

Usually for the calculation of the distance $\|d_i - P_l\|$ the Euclidian distance is being used as a proximity measure. When all data points are assigned to clusters the algorithm calculates the new centers based on the centroids of the data points of each cluster:

$$P_j = \frac{\sum_{i=1}^{n} \mu_j(d_i) d_i}{\sum_{i=1}^{n} \mu_j(d_i)}. \tag{3.3}$$

These steps are applied recursively until the membership of the clusters no longer changes or until the error function E (Equation 3.2.1) does not change significantly (converge).

$$E = \sum_{j=1}^{k} \sum_{i=1}^{n} \mu_j(d_i) \|d_i - P_j\|^2. \tag{3.4}$$

In brief, the k-means algorithm can be summarized as follows:

1. Initialize the k centers in the dataset.
2. Assign each data point to its closest center.
3. Calculate the new centers.
4. Repeat steps 2 and 3 until converged is achieved.

In general the algorithm can be described as a optimization procedure of the objective function:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \|d_i - P_j\|^2 \tag{3.5}$$

Although it can be proved that the process will always converge, the optimal partitioning does not necessary corresponds to the global minimum of the objective function.

Figure 3.4 displays a simple 2-dimensional example of the k-means algorithmic procedure. The actuals clusters are being found into four algorithmic steps.

The k-means algorithm is simple and quite efficient in most cases. There are also many variants that improve its performance and are less susceptible to initialization problems. However, k-means is not suitable in cases where clusters are not globular or vary in size and density. In addition k-means has troubles dealing with dataset that contain outliers. For this reason in such cases, outlier removal methods are employed.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

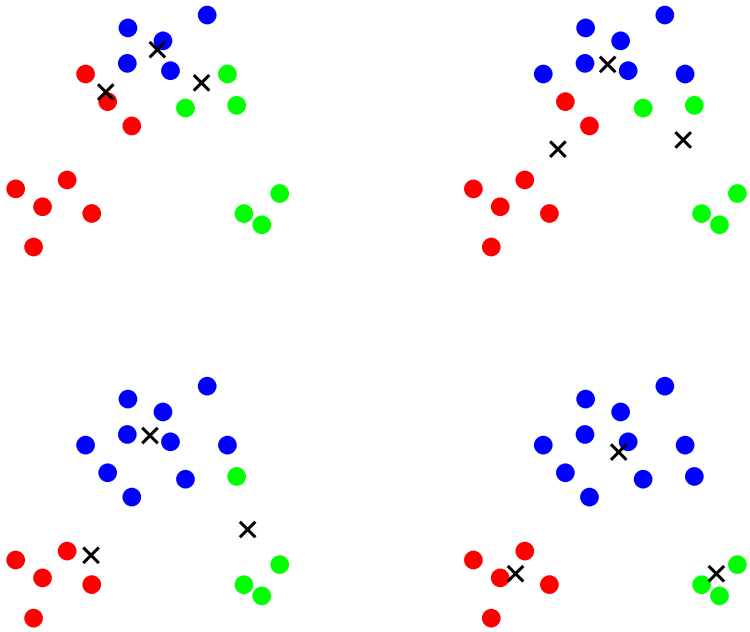


Figure 3.4: *Four iterations of the k-means algorithmic procedure.*

3.2.2 The Fuzzy c-means algorithm

One of the most widely used fuzzy clustering algorithms is the Fuzzy C-Means (FCM) algorithm. FCM is a clustering method that assigns each each data point to a cluster to some degree that is specified by a membership grade. This technique was originally introduced by Jim Bezdek in [Bez81] as an improvement on earlier clustering methods and attempts to partition a finite collection of data vectors into a collection of fuzzy clusters with respect to some given criterion. A theoretical discussion of FCM can be found in [Cox05].

Given a finite set of data, the algorithm returns a list of cluster centers and a partition matrix indicating the degree to which each element belongs to a given cluster. Like the k-means algorithm, the FCM aims to minimize an objective function, like the following:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \|d_i - P_j\|^2, \quad 1 \leq m \leq \infty, \tag{3.6}$$

where m is any real number greater than 1, u_{ij} is the degree of membership of d_i in the cluster j, d_i is the i-th of a-dimensional measured data, P_j is the a-dimension center of the cluster, and $\|\cdot\|$ is any norm expressing the similarity between any measured data and the center. Fuzzy partitioning is carried out through an iterative optimization of the objective function

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

shown above, with the update of membership u_{ij} and the cluster centers P_j by:

$$u_{ij} = \frac{1}{\sum_{l=1}^{C} (\frac{\|d_i - P_j\|}{\|d_i - P_l\|})^{\frac{2}{m-1}}}, \tag{3.7}$$

where

$$P_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot d_i}{\sum_{i=1}^{N} u_{ij}^m}.$$

This iteration stops when

$$max_{ij}\{|u_{ij}^{l+1} - |u_{ij}^l|\} \leq \epsilon, \tag{3.8}$$

where l is the iteration number and ϵ is a constant between 0 and 1 that controls the termination of algorithm. This procedure converges to a local minimum or a saddle point of J_m.

§ 3.3 Density Based Clustering Algorithms

The density based clustering algorithms define a cluster based on the hypothesis that the density (or the number of data points) in a neighbourhood is over a specified threshold, while the distance based algorithms, assign a data point to a cluster based on the distance between the data point and the center of the cluster. In contrast to distance based algorithms, the density based can also deal effectively with cluster of non spherical or random shapes. One of the most well known algorithms of this class is the DBSCAN algorithm.

3.3.1 DBSCAN

Density Based Spatial Clustering of Application with Noise [SEKX98] (DBSCAN) is a density based clustering algorithm that has been proved quite effective for spatial databases. Clusters are considered as high density neighbourhoods of data points that are separated by neighbourhoods of low density. Although the density parameter is critical for DBSCAN's success, recently proposed heuristics are able to give high quality results. The basic type of DBSCAN density is defined by the center based approach. The density of a data point is estimated by counting the number of points within a specified radius (R_d). As expected the density of any point is heavily depended on this radius. At the first step of the algorithmic procedure all data points are classified into three categories. These are:

- Core points,

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

- Border points,
- Noise points.

Core points are those points that contain at least a specified number of data points within the radius R_d, border points are points that are not core points, but a core point falls within their radius R_d and finally noise points are those points that do not belong to any of the other two classes. A two dimensional example of this classification is illustrated at Figure 3.5, where the minimum number of points within the radius R_d that defines a core point is five.

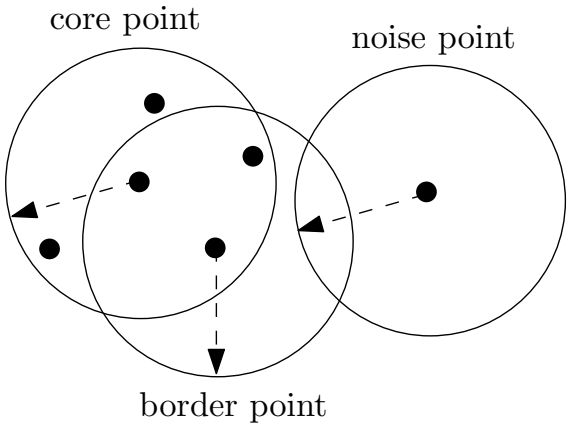


Figure 3.5: *An example of the data points classification for DBSCAN algorithm.*

At the next step of the algorithmic procedure, any two core points lying within the radius R_d of one another are put in the same cluster and any border point that is close enough to a core point is put in the same cluster as the core point. Finally all noise points are erased.

DBSCAN is relatively resistant to noise and, in contrast to k-means can handle clusters of arbitrary shapes and sizes. However, the algorithm has difficulties in dealing with high dimensional data because density is more difficult to define for such data.

§ 3.4 Model-Based Clustering

Model based clustering algorithms use certain models for clusters and try to optimize the fit between the data and the models. In model based clustering it is assumed that the data are generated by a mixture of probability distributions. Each component of the mixture corresponds to a cluster. We

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

expect a clustering method to perform accurately when the data conform to the model.

3.4.1 Gaussian Mixture Models

Gaussian Mixture Models (GMM) is a widely used clustering approach that is often used for understanding and developing clustering criteria. In GMM the data input D is considered to come from a random vector with density [CG93]:

$$f(d) = \sum_{j=1}^{k} m_j \Phi(d|\mu_j, \Sigma_j)$$

where m_j are the mixing proportions, $0 < m_j < 1$ and $\sum_{j=1}^{k} m_j = 1$, and $\Phi(d|\mu, \Sigma)$ is the density of the Gaussian distribution with mean vector μ and covariance matrix Σ:

$$\Phi(d|\mu, \Sigma) = \frac{exp[-\frac{1}{2}(d-\mu)^T \Sigma^- 1(d-\mu)]}{\sqrt{(2\pi)^d |\Sigma|}}. \tag{3.9}$$

There are two maximum likehood approaches for estimating the parameters in a mixture:

- Mixture approach

  The aim of this approach is to maximize the likehood over the mixture parameters. As such the parameters

  $$\theta = m_1, \ldots, m_{k-1}, \mu_1, \ldots, \mu_{k-1}, \Sigma_1, \ldots, \Sigma_k$$

  are chosen to maximize the loglikehood

  $$L(\theta|d_1, \ldots, d_n) = \sum_{i=1}^{n} \ln \left[\sum_{j=1}^{k} m_j \Phi(d_i|\mu_j, \Sigma_j) \right]. \tag{3.10}$$

  To find the parameters θ that maximize the above equation, the Expectation Maximization algorithm is used.

- Classification approach

  The aim of the classification approach is to maximize the likehood over the mixture parameters and also over the identifying labels of the mixture component origin for each data point. Therefore the indicator vectors $z_i = (z_{i1}, \ldots, z_{ik})$, for which $z_{ik} = 1$ if d_i has been drawn from the k-th component, and $z_{ik} = 0$ if d_i has been drawn from any other component, are treated as unknown parameters.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

3.4.2 The Expectation Maximization Algorithm

The Expectation Maximization (EM) algorithm is a statistical method of maximum likelihood estimation in the presence of incomplete data that can be used for the purpose of clustering. For the clustering purpose the data are defined as the set

$$\{(d_i, z_i), i = 1, \ldots, n\},$$

where $D = \{d_i, i = 1, \ldots, n\}$ is the input data set and $z_i = (z_{i1}, \ldots, z_{ik})$ are defined as

$$z_{ij} = \begin{cases} 1 & \text{if} \quad x_i \quad \text{belongs to the j-th cluster} \\ 0 & \text{else} \end{cases}.$$

There are two relevant assumptions. Firstly, each z_i is independent and identically distributed according to a multinomial distribution of one draw on k classes with probabilities $\tau_1, \ldots, \tau_2$. Secondly, we have that the density of d_i given z_i is given by

$$\prod_{j=1}^{k} f_j(d_i|\Theta_j)^{z_{ij}}.$$

Then the complete loglikehood has the form

$$l(\Theta_j, \tau_j, z_{ij}|D) = \sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij} \log[\tau_j f_j(d_i|\Theta_j)],$$

and the conditional expectation of z_ij for the data point d_i is the following

$$\hat{z}_{ij} = E[z_{ij}|d_i, \Theta_1, \ldots, \Theta_k].$$

In the case of the hard clustering task, the data point d_i is assigned to the cluster that corresponds to the highest membership. The algorithmic procedure of EM is consisted by two steps named the E-step and the M-step. During the E-step the values of $\hat{z}_{ij}$ are being calculated from the data input and the current parameter estimates. The M-step procedure consists the maximization of the complete data likehood, where each z_{ij} is replaced by its current conditional expectation $\hat{z}_{ij}$. For more information regarding the EM algorithm refer to [MK08].

§ 3.5 Conclusions

Although many clustering algorithms have been proposed in the recent years, the continuously increasing size of data bases and the new complex applications constantly arising, impose the necessity for development of new clustering algorithms. These new algorithms must have the ability to utilize the

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

possibilities of the modern data bases and incorporate the knowledge about the several application fields easily.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

- Chapter 4 -

Knowledge Discovery in High Dimensional Data

Do not say a little in many words, but a great deal in a few.

—*Pythagoras*

While clustering has a long history and a large number of clustering techniques have been developed in statistics, pattern recognition, data mining, and other fields, significant challenges still remain. One of the most important challenges encountered in clustering is associated with high data dimensionality ("curse of dimensionality") [Bel61].

In general terms, these problems result from the fact that a fixed number of data points become increasingly "sparse" as the dimensionality increases [SEK03]. For clustering purposes, the most relevant aspect of the curse of dimensionality concerns the effect that it has on distance or similarity.

More precisely, the measure of distance or similarity affects critically most clustering algorithms since it is required that the data points of one cluster to be closer to each other that any data point that belongs to another cluster. To visually analyse if a data set may contain clusters or not, we could plot the histogram of the pairwise distances of the data points. In the cases where clusters exist, the graph will typically show two peaks. One peak represents the distance between the data points in clusters and the other represents the average distance between data points. If there is only one peak visible it means that the clustering task based on distance measures is expected to be difficult.

In [BGRS99], for certain data distributions, it is shown that the relative difference of the distances of the closest and farthest data points of an independently selected point goes to 0 as the dimensionality increases.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

$$\lim_{a\to\infty} \frac{MaxDist - MinDist}{MinDist} = 0.$$

Thus, it is often said that *"in high dimensional spaces, distances between points become relatively uniform"* [SEK03].

The authors in [BGRS99], provide a theoretical analysis of several types of distributions along with results for high dimensional datasets. Although this work was oriented towards the problem of finding the nearest neighbors of points, many problematic situations have been exposed concerning clustering of high dimensional data.

The nearest neighbor (NN) problem involves the determination of the point x_{NN} from the dataset D that is the nearest to a query point $q \in \mathbb{R}^a$ (see Figure 4.1).

$$x_{NN} = \{x' \in D | \forall x \in D, x \neq x' : dist(x', q) \leq dist(x, q)\}.$$

Solving this problem is important for many applications such as similarity search in geometric or multimedia databases [MG95, KSF+96, FEF+94], data mining in fraud detection [BGMP99, HGY99] and information retrieval [AGM+90, Sal89]. Most of these fields concerns high dimensional data.

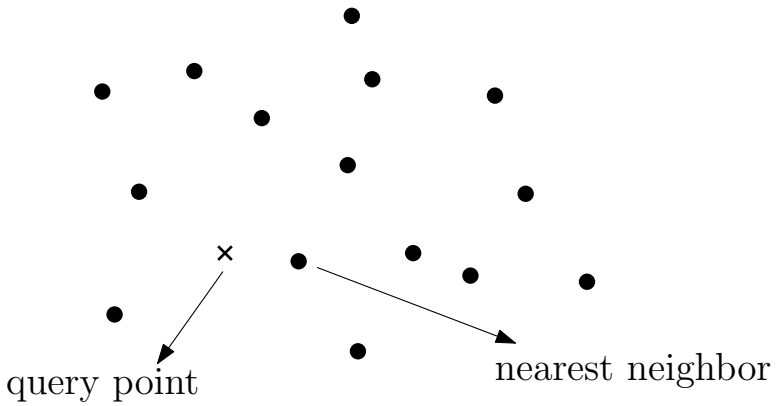


Figure 4.1: *A query point and its nearest neighbor.*

Although many other empirical studies have shown that traditional indexing methods fail in high dimensional spaces [BKK96, BGRS99, WSB98], most NN problems can be solved easily for low dimensional applications for which efficient index structures have been proposed. Many multidimensional indexes have been proposed that work efficiently for low dimensional data [GG98] .

However, as mentioned in [BGRS99] while it is natural to ask for the nearest neighbor, there is not always a meaningful answer for a wide range of distributions and distance functions as such questions arise as to weather the

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

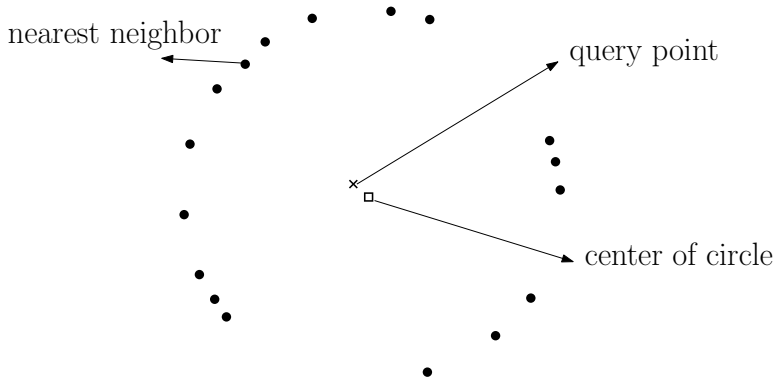


Figure 4.2: *A query point and its nearest neighbor.*

NN problem is actually meaningful. To explain this we consider the scenario illustrated in Figure 4.2. Although there is a well defined nearest neighbor, the distance difference between the nearest neighbor and any other point is very small, as such the solution of concrete problems is not of much use. Although this scenario is very contrived for any two dimensional application of nearest neighbor, in [BGRS99] is shown that this is the norm for a broad class of data distributions in high dimensionality. To establish this, the authors examine the number of points that fall into a query sphere enlarged by some factor ϵ (Figure 4.3). In the case where only few points fall into this sphere, it means that the nearest neighbor to the query point is separated from the rest of the data meaningfully. However, if most points fall into the sphere there is no meaning in differentiating the nearest neighbor from the rest data points if ϵ is small. The above observation is described by the following definition

Definition 1. *A nearest neighbor query is unstable for a given ϵ if the distance from the query point to most data points is less than $(1+\epsilon)$ times the distance from the query point to its nearest neighbor.*

The authors in [BGRS99] show that in many situations, for any $\epsilon > 0$, as dimensionality rises, the probability that a query is unstable converges to 1.

In [HAK00] the authors deal with the quality issue of nearest neighbor search and examine several theoretical and practical aspects of performing nearest neighbor queries in high dimensional space. There are many reasons for which there is not much meaning of the nearest neighbor search in high dimensional spaces. One of the most important reasons is the data sparsity. Here the authors extend the approach presented in [BGRS99] by focusing on the difference between the distance of the nearest and the farthest data point to a given query point (absolute difference, $MaxDist–MinDist$), instead

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

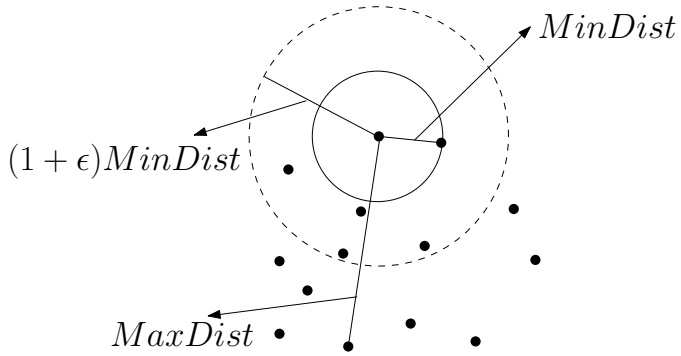


Figure 4.3: *The query region and the enlarged region. MinDist is the distance to the nearest neighbor and MaxDist the distance to the farthest data point.*

of the relative difference. It was shown that this value is depended on the distance measure. More precisely, for the L_1 distance it holds that *MaxDist–MinDist* increases with dimensionality, while for the L_2 distance remains almost constant and for L_d where $d \geq 3$ goes to 0. As such the L_d metric for $d \geq 3$ is meaningless for high dimensional data.

§ 4.1 Algorithms for High Dimensional Data Clustering

All the aforementioned results showed the difficulties that emerge in clustering of high dimensional data. However, things are sometimes not as bad as they might seem, since it is often possible to reduce the dimensionality of the data without significant loss of information. This can be accomplished by the feature selection procedure, i.e. discarding features showing little variation or having high correlation to other features. Feature selection is a complicated subject in its own and is out of the scope of the present study.

Another clustering approach is to project points to a suitable low dimensional space. Typically, this type of dimensionality reduction is accomplished by applying techniques from linear algebra or statistics such as Principal Component Analysis (PCA) [JD88] or Singular Value Decomposition (SVD) [Lax07]. This way, a powerful class of clustering algorithms has emerged that provides the opportunity for the deployment of these computational technologies from numerical linear algebra, an area that has seen enormous expansion in recent decades. In this class of algorithms, the Principal Direction Divisive Partitioning (PDDP) algorithm is of particular value [Bol98]. Compared to other similar techniques (like Latent Semantic Indexing [DDF+90] and Linear Least Square Fit [CY95]), PDDP has the

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

advantage of very low computational complexity.

4.1.1 CLIQUE

CLIQUE algorithm [AGGR98] is based on the idea that a dense region in a particular subspace must create dense regions when projected onto a lower dimensional subspace.

To explain how the algorithm proceeds a simple two dimensional dataset is employed. In Figure 4.4 we can see the one dimensional dense regions on the x (horizontal) and y (vertical) coordinates respectively, which indicate the projections of the two dimensional clusters. As shown the one dimensional dense regions denote the regions (the intersections of two dense one dimension dense regions) that probably contain clusters in a higher dimension. As such, by inspecting these one dimensional intervals one can find the actual clusters. This technique can be easily applied to any subspace for finding clusters efficiently. The disadvantages of CLIQUE is that in order to reduce the subset of the investigated dimensions the algorithm needs to use heuristics and also the computational complexity of the algorithm is not linear with the dimensionality of the dataset.

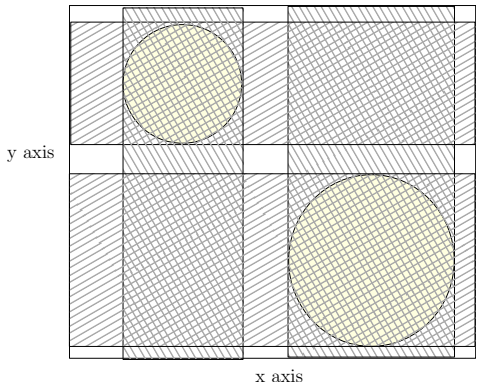


Figure 4.4: *A two dimensional example of the CLIQUE algorithm procedure.*

4.1.2 MAFIA

The Merging Adaptive Finite Intervals And is more than a clique (MAFIA) algorithm [GNC99] is an improvement over the CLIQUE approach. The boosted efficiency is achieved by using non-uniform grid cells. In more detail, in the MAFIA procedure we partition each dimension based on a variable number of adaptive intervals that represents better the distribution of the data set in that dimension, instead of splitting the data into

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

a predefined number of evenly spaced intervals. To better explain the algorithmic approach a two dimensional example is employed in Figure 4.5 for the comparison between the CLIQUE and the MAFIA algorithms. The grid illustrated in Figure 4.5 (left) that CLIQUE uses, breaks the dense one dimensional regions, as such there exists intervals at the edge that has lesser density. On the other hand, the MAFIA algorithm begins with a large number of intervals for each dimension and then combine those that have similar density. The results of the algorithm are shown in Figure 4.5 (right).

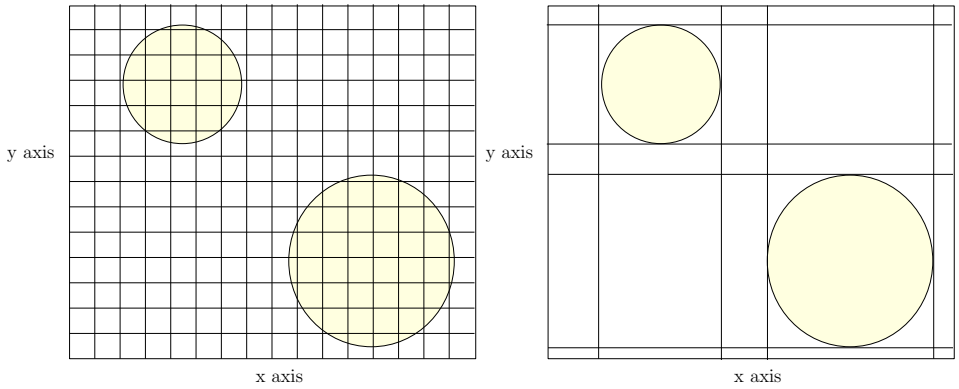


Figure 4.5: *A comparison between the CLIQUE algorithm procedure (left) and the MAFIA algorithmic procedure (right).*

4.1.3 DENCLUE

The DENsity CLUstering (DENCLUE) [HHK98] algorithm is a density based approach which can be considered as a generalization of other density approaches such as DBSCAN. In the DENCLUE approach the clusters are defined by the local peaks of the overall density. More specifically, for each data point a hill climping technique is applied in order to find its closest density peak. Then a set of data points assign to a particular peak define a cluster. In the case where the density of a local peak is extremely low, all elements assigned to the cluster defined by this peak are considered to be noise and are erased. In addition, if two local peaks are connected through a path of data points whose density is above a predefined threshold, then the clusters defined by those peaks are merged. Thus, based on that attribute, the algorithm can discover clusters of various shapes.

The DENCLUE procedure is based on the Kernel Density Estimation (KDE) [WJ95]. KDE is a statistical technique used to describe the distribution of a data set. In KDE the density is defined as the sum of a set of kernel functions that corresponds to each data point. More details regarding KDE will be given in Chapter 5.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

There are two basic steps in the DENCLUE algorithmic procedure. Firstly, there is a prepossessing step where the minimal bounding hyper rectangle is being divided in order to create a grid for the data. Then, the grid shell that contain data points are determined and numbered according to a particular origin. These grid shells are stored along with information about the number of points; the sum of the points in the cell and the connections they have to other cells. The next step of the algorithm is the clustering step, where only the high populated cells and the cells connected to them are being used. For each point the density is calculated based only on points that belong to cells that are close to it. Finally, the algorithm merges high density cells that are connected by a path of points having density greater than a predefined value.

4.1.4 OptiGrid

The OptiGrid algorithm [HK99] was created by the authors of DENCLUE in order to surpass the problems of the DENCLUE algorithm when the dimensionality or the noise increases. The OptiGrid algorithm is based on two observations made by the authors about the behaviour of points in high dimensional space. The first observation is that the data noise seems to correspond to uniformly distributed data in high dimensions, and the second observation is that interpoint distances become relatively uniform as dimensionality increases.

A simple representation of the algorithmic steps is given below:

1. Firstly for each dimension, a histogram of the data points is generated. Then, the noise is determined by examining the histogram, and the left and right maxima and the $c-1$ (where c the number of clusters input) maxima in between them are being found. Next, the c minima between these maxima that define all the possible cuts are being found and sorted according to a criterion (for example density).

2. Now select the best c cuts amongst all dimensions.

3. A grid that partitions the data is now created based on these cuts.

4. The high populated cell are being found and added to the list of clusters.

5. Refine the cluster list.

6. Steps 1-5 are repeated using each cluster.

The OptiGrid algorithm in summary simply looks for optimal cutting planes and then generates a grid based on that planes that is not likely to cut any real clusters. Then it locates the potential clusters in the set

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

of grid cells and further partitions them if possible. Although this is an effective technique, there are many parameters that need to be defined in the implementation of OptiGrid.

4.1.5 Principal Direction Divisive Partitioning

The Principal Component Divisive Partitioning algorithm (PDDP) is a "divisive" hierarchical clustering algorithm that operates directly on a $n \times a$ data matrix D. It's basic advantage is the very low computation complexity in comparison with other algorithms of the same category. The algorithm proceeds by separating the entire set of samples into two partitions P_1 and P_2, using the projections p_i onto the first principal component u_1, which is the direction on which the projections have the maximum variance (more details regarding Principal Component Analysis will be given in Section 4.2.1). Each of the two partitions will be separated into two subpartitions using the same process recursively. The result is a hierarchical structure of partitions arranged into a binary tree ("the PDDP tree") in which each partition is either a leaf node (meaning it has not been separated) or has been separated into two subpartitions forming its two children in the PDDP tree. Any divisive clustering algorithm can be characterized by the way it chooses to provide answers to the following three questions:

Q_1: Which cluster to split further?

Q_2: How to split the selected cluster?

Q_3: When should the iteration terminate?

To generate this partitioning, the original PDDP algorithm uses the sign of the projection of each data point (division point 0). This reveals the text mining origins of the algorithm, since based on a term-document data specification [Bol98], data points (documents) with positive projections should be more similar to each other than data points (documents) with negative ones. However, very often, the sign of the projection can lead to undesirable cluster splits [TT08, ZG03, Dhi01, DKN03]. More formally we can define this splitting criterion as follows:

- (Splitting Criterion SPC_1): $\forall d_i \in \mathcal{D}$, if $p_i \geqslant 0$, then the i-th data point belongs to the first partition $P_1 = P_1 \cup d_i$; otherwise, it belongs to the second partition $P_2 = P_2 \cup d_i$.

In [Nil02], the utilisation of the second, the third, etc. component (instead of the principal one) was proposed. A suitable projection was calculated based on a measure of coherence (scatter value, defined below) of the generated clusters. However, this criterion greatly increases the computational complexity of the algorithm. Similarly, in [ZG03] it is suggested to

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

extend this in a $2l$-way partitioning strategy, by splitting simultaneously in $2l$ clusters using the sign of the projections of the $l \geqslant 1$ singular vectors. Although, both these approaches have been shown to lead to improved cluster quality, the problem is not solved. Using alternative principal components does not guarantee improved partitioning, as the problem of where to split irrespective of which projection is used, still exists.

Another approach to determine the splitting point is to use a k-means steering procedure [Dhi01, DKN03, ZG07], with $k = 2$. In [ZG07] it is suggested to extensively search every splitting point and select the one with the best k-means objective function (for $k = 2$). The drawback of this approach lies in the fact that even if the best such splitting point is found, it does not guarantee a good splitting decision, as the k-means objective function is flawed in many cases (e.g. unbalanced clusters, more than two true clusters in the data, etc.) [JD88].

To answer the cluster selection question Q_2, the PDDP algorithm as well as all its variations [ZG03, Dhi01, DKN03, Nil02, ZG07], select the cluster with maximum scatter value SV, defined as:

$$SV = \|D - be\|_F, \tag{4.1}$$

where the vector b represents the mean vector of D and $\|\cdot\|_F$ is the Frobenious norm. This quantity can be a measure of coherence and it is the same as the one used by the k-means steered PDDP algorithm [ZG07].

Finally, most of the PDDP variants terminate the clustering procedure when a user defined number of clusters (c_{max}) has been retrieved. Little effort has been given to develop a method for the automatic cluster determination, which fits with the workings of the PDDP algorithm.

A detailed description of the algorithm is presented in Table 4.1. As shown the criteria to answer questions Q_1, Q_2 and Q_3 can be summarized as follows:

C_1: Select the cluster with the largest scatter value.

C_2: Split the cluster based on the sign of the data projection onto the first principal component.

C_3: Stop when a user defined number of clusters has been retrieved.

§ 4.2 Dimensionality Reduction Techniques

In this Section the already known techniques that are used for dimensionality reduction in this work are presented. These are the Principal Components Analysis (PCA), the Independent Component Analysis (ICA) and the Random Projection method.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Function PDDP (D, c_{max}) {

1. While number of leafs of the *pddp tree* smaller than c_{max}

2. Select the leaf P_k of the *pddp tree* with the largest scat value: $k = \arg\max_i\{scat(P_i)\}$

3. Split P_k according to Splitting Criterion SPC_1

4. Create sub-clusters of P_k and add them to the *pddp tree*

}

Table 4.1: *The PDDP algorithm Summary.*

4.2.1 Principal Component Analysis

Principal Components Analysis (PCA) is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. The central idea of Principal Component Analysis (PCA) is to reduce the dimensionality of a data set, while retaining as much as possible of the variation present in the data set.

To formally describe the manner in which PCA operates, let us assume the data is represented by an $n \times a$ matrix D, whose each row represents a data sample d_i, for $i = 1, \ldots, n$. Also define the vector b and matrix Σ to represent the mean vector and the covariance of the data respectively:

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

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.

Let U_k be the matrix with columns the eigenvectors that corresponds to the k largest eigenvalues and denote the targeted subspace. Then

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

are the projections of d_i onto the k-dimensional subspace.

A simple example of the steps of the PCA procedure is illustrated in Figure 4.6. For visual representation a two dimensional dataset is employed. In Figure 4.6 (top) we can see the initial dataset (left) and the centralized dataset (right), respectively. For PCA to work properly, you have to subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension, as such the resulted data set has a zero mean.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Next, we need to calculate the covariance matrix and the corresponding eigenvalues and eigenvectors. As expected, since this is a 2-dimensional data set there are two eigenvectors (the columns of EC) and two corresponding eigenvalues (EL) respectively.

$$EC = \begin{bmatrix} -0.8447 & 0.5353 \\ 0.5353 & 0.8447 \end{bmatrix}$$

$$EL = \begin{bmatrix} 2.7964 & 0 \\ 0 & 13.5504 \end{bmatrix}$$

In Figure 4.6 (bottom left) we can see the eigenvectors plotted on top of the data. The eigenvector that corresponds to the highest eigenvalue is the first principal component and is considered to be the most significant. To reduce the dimensionality of the data set with the minimum loss of information we project the data onto the first Principal Component. Figure 4.6 (bottom right) illustrates the projected data. An explanatory image of the procedure is presented in Figure 4.7.

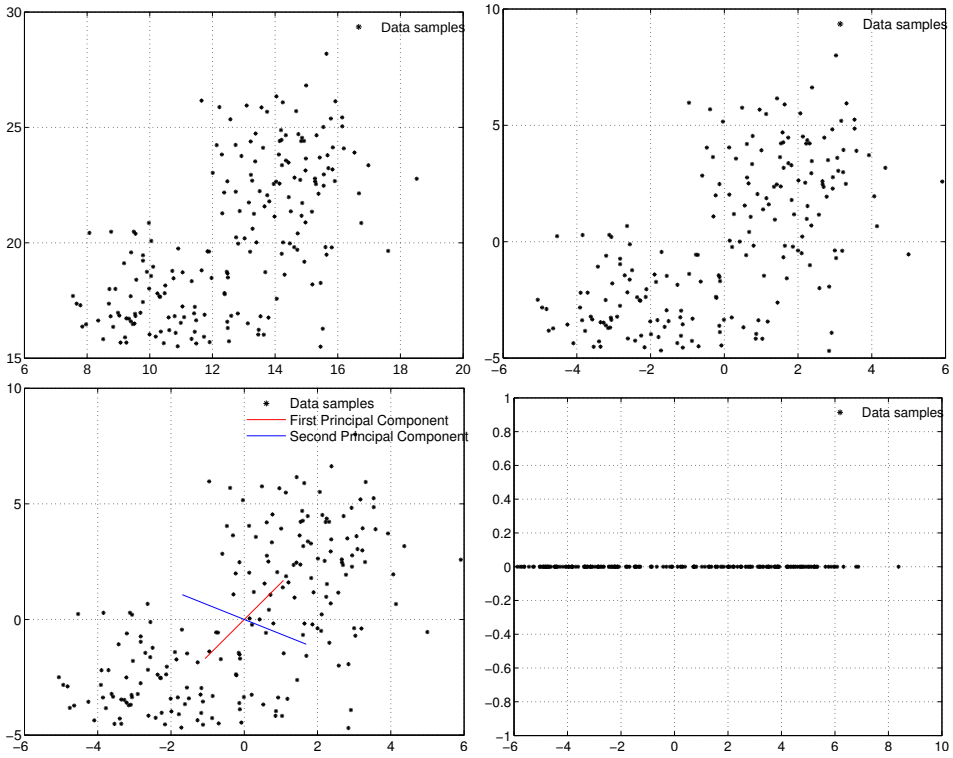


Figure 4.6: *A two dimensional example of various steps of the PCA method.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

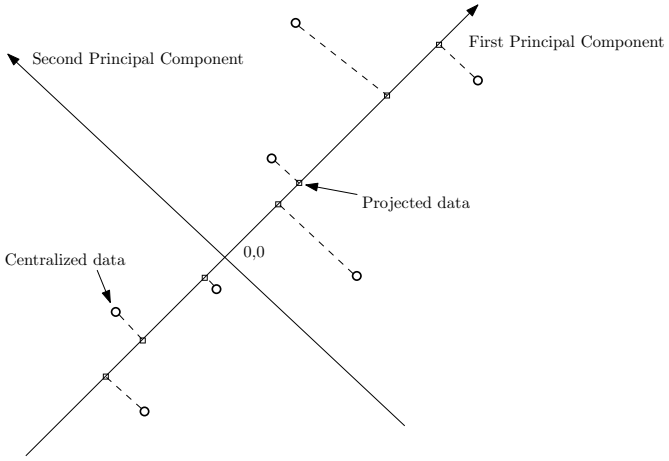


Figure 4.7: *A two dimensional example of various steps of the PCA method.*

4.2.2 Independent Component Analysis

Independent component analysis (ICA) [A. 01, Com94] is a technique that finds underlying factors or independent components from multivariate (multidimensional) statistical data by maximizing the statistical independence of the estimated components. ICA defines a generative model for the observed multivariate data, which is typically given as a large database of samples. In the model, the data variables are assumed to be linear or nonlinear mixtures of some unknown latent variables and the mixing system is also unknown. The latent variables are assumed non-gaussian and mutually independent and they are called the independent components of the observed data. These independent components can be found by ICA.

We can define ICA as follows. Let $x = (x_1, ..., x_n)$ be the random vector that represents the data and $s = (s_1, ..., s_n)$ be the random vector that represents the components. The task is to transform the data x, using a linear static transformation W as

$$s = Wx,$$

into maximally independent components s measured by some function of independence.

The most widely known definitions of independence for ICA are the maximization of nongaussianity and the minimization of mutual information. The classical measure of nongaussianity is kurtosis or the fourth-order cumulant. Kurtosis is zero for a gaussian random variable and nonzero for most non-gaussian random variables. Kurtosis, or rather its absolute value, has been widely used as a measure of nongaussianity in ICA and related

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

fields. The main reason is its simplicity, both computational and theoretical.

The second very important measure of nongaussianity is given by negentropy. Negentropy is based on the information-theoretic quantity of (differential) entropy. Entropy is the basic concept of information theory. The entropy of a random variable can be interpreted as the degree of information that the observation of the variable gives. The more random and unstructured the variable is, the larger its entropy. The minimization of mutual information is inspired by information theory.

Mutual information is equivalent to the well-known Kullback-Leibler divergence. It is always non-negative, and zero if and only if the variables are statistically independent. Thus, mutual information takes into account the whole dependence structure of the variables and not only the covariance, like the PCA and related methods do.

4.2.3 The Random Projection Method

In certain cases even Principal Component Analysis can be quite expensive to compute for ultra high dimension datasets. For this reason Random Projection (RP) [Ach01, BM01, PRTV98] emerged as an alternative method for dimension reduction. In RP, the data are projected onto a lower dimension subspace using a random matrix. RP is extremely computational efficient and has also the desirable property to preserve the structure of the data without introducing significant distortion. In [PRTV98], Papadimitriou et al. use random projection in the prepossessing of textual data, prior to applying Linear Discriminant Analysis (LDA) for document categorization and classification. Despite the fact that LDA is an accurate technique for solving these problems, it is also computational inefficient for large databases. The authors used RP to speed up LDA by reducing the dimensionality of the data. Kaski [Kas97] presented experimental results using RP in the context of a system for organizing textual documents using Self-Organising Maps (SOMs) (i.e. WEBSOM). His results illustrate that RP needs moderate number of dimensions for producing a good mapping. In this case, the results were as good as those obtained using PCA. Bingham and Manilla [BM01] compared several dimensionality reduction techniques on image and text data. In their results it was shown that RP preserves distances and has performance comparable to that of PCA, while being much faster. In [GBN], Goel et al. investigated the feasibility of RP for face recognition. The authors there performed a large number of experiments and comparisons using PCA. The experimental results illustrated that although RP represents faces in a random, low-dimensional subspace, its overall performance is comparable to that of PCA while having lower computational requirements. Dasgupta [Das99, Das00] also concludes that RP results in more spherical cluster than those in the original dimension.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

RP also performs better than PCA on eccentric data, where actually PCA might fail completely. Finally, Boutsidis et al. in [BZD10] use RP for dimensionality reduction for k-means clustering. The authors there provide theoretical results and present a fast and accurate method. The uses of RP are also abundant in other applications like molecular biology [LLTY97], and signal processing [CRT06, Don06].

4.2.3.1 Theoretical aspects of RP

In Random Projection method, the original a-dimensional data is projected to an r-dimensional subspace ($r < a$), using a random $a \times r$ orthogonal matrix R, whose rows have unit lengths. Using matrix notation where $D_{n\times a}$ is the original set of n a-dimensional observations,

$$D^{RP}_{n\times r} = D_{n\times a} R_{a\times r},$$

is the projection of the data onto the lower r-dimensional subspace spanned by $\mathbb{R}$.

The orthogonalization of R is computational expensive, but necessary in order to preserve the similarities between the original vectors in the low dimension space and to avoid distortions. However, in some cases, we can avoid orthogonalization. As shown in [HN94], in high dimension spaces, there exists a much larger number of almost orthogonal vectors than orthogonal directions. Thus, high-dimensional vectors having random directions are very likely to be close to orthogonal.

The selection of the elements of the matrix R is a matter of interest. In most cases the elements are drawn from a normal distribution, but this is not always necessary. In [Ach01] the author has proposed a much simpler algorithm for approximating the random matrix. That algorithm produces a sparse random matrix with elements in $\{-1, 0, 1\}$. In this way there are further computational savings. Random projection is computationally very simple, forming the random matrix R and projection the $n \times a$ data matrix D onto r dimensions is of order $O(nra)$ and if the data matrix D is sparse with about c non zero entries per column, the complexity is reduced to $O(cra)$ [PRTV98].

Random Projection is motivated by the Johnson - Lindenstrauss lemma (lemma 4.2.1) that states that a set of n points in a high dimension Euclidean space can be mapped down onto an $r < O(\log n/\varepsilon^2)$ dimension space such that the distances between the points are approximately preserved. In other words, the distance are not distorted more than a factor of $1 \pm \varepsilon$, for any $0 < \varepsilon < 1$.

Lemma 4.2.1. *(**Johnson and Lindenstrauss** [**JL84**]): Given $\varepsilon > 0$ and an integer n, let r be a positive integer such that $r \geq r_0 = O(\varepsilon^{-2} \log n)$. For*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

every set D of n points in $\mathbb{R}^a$, there exists $g : \mathbb{R}^a \to \mathbb{R}^r$ such that for all $x, y \epsilon D$

$$(1-\varepsilon)\|x-y\|^2 \leq \|g(x)-g(y)\|^2 \leq (1+\varepsilon)\|x-y\|^2$$

Proof of this lemma is given in [Ach01] and [DG99].

§ 4.3 High Dimensional Applications

In this Section we present the real life high dimensional clustering applications used in this work for the evaluation of the proposed clustering techniques.

4.3.1 Application on Microarray Datasets

The development of microarray technologies gives scientists the ability to examine, discover and monitor the mRNA transcript levels of thousands of genes in a single experiment. Discovering the patterns hidden in gene expression microarray data is a tremendous opportunity and challenge for functional genomics and proteomics. A promising approach to address this task is to utilize data mining techniques. Cluster analysis is a key step in understanding how the activity of genes varies during biological processes and is affected by disease states and cellular environments. In particular, clustering can be used either to identify sets of genes according to their expression in a set of samples, or to cluster samples into homogeneous groups that may correspond to particular macroscopic phenotypes. The latter is in general more difficult because of the high dimensionality that describes these datasets. For this reason they make a perfect candidate to explore the performance of the algorithms proposed in this paper.

4.3.2 Text Mining Application

In the recent years text mining is a topic of great interest. Due to the continuous evolution of the hardware and software platforms, web applications, social networks applications and other information based applications produce large amounts of text databases. As such there is a growing need for the development of methods that have been designed in order to be able to process these databases effectively. For this reason the text mining task constitutes a research field in data mining, machine learning and information retrieval. Text data are in their nature high dimensional and as a result very often dimensionality reduction techniques are employed by the algorithms.

4.3.3 Face Recognition

Facial recognition systems are becoming very popular in numerous applications from security systems to smart vending machines [TLL+10]. Face

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

is a highly non-rigid object. Detecting human faces from an image it is a difficult task mostly because of the variation of human faces such as races, illumination, facial expressions, face scales, head poses (off-plane rotations), face tilting (in-plane rotations), occlusions, etc. In most cases the problem is also affected by the environment where lighting conditions, image quality, and cluttered backgrounds emerge. For this ultra high dimensional task, PCA analysis or similar dimensionality reduction techniques are the most used tools for analysis.

§ 4.4 Conclusions

While a lot of work has been done on developing clustering methods, there are still significant challenges that need to be addressed. One of the most important of them is high data dimensionality. To deal with this problem new clustering methods have been developed based on the fact that it is possible to reduce the data dimensionality without significant loss of information. In this Section some of the most well known methods in this area a re presented along with the most effective tools for dimensionality reduction.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Part

New Algorimths

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

- Chapter 5 -

Enhancing Principal Direction Divisive Clustering

> Life is an unfoldment, ... To understand the things that are at our door is the best preparation for understanding those that lie beyond.
>
> —*Hypatia*

§ 5.1 Introduction

In this Section, we aim to understand what can be achieved by partitioning the data based on their projection on a particular direction. To this end, we attempt to theoretically discover the relationship between true clusters in the data and the distribution of their projection onto the principal components.

Obviously, this requires assumptions of what true clusters actually mean, which are called "inductive bias" [GvLW09]. Here, out of the large variety of such assumptions, we focus on two particular ones: the "compact" and the "peak" based clusters, respectively. Based on these assumptions, we propose appropriate criteria for the various steps involved in hierarchical divisive clustering (see Section 4.1.5).

Any divisive clustering algorithm can be characterised by the way it chooses to provide answers to the following three questions:

Q_1: Which cluster to split further?

Q_2: How to split the selected cluster?

Q_3: When should the iteration terminate?

In the sections that follow we are going to examine each of the three basic questions involved in the hierarchical divisive clustering procedure individually.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

§ 5.2 How to split the selected cluster?

This question is the most central one in a divisive clustering procedure. The reason for this lies in the fact that if a bad decision is made in the first steps on the procedure, little can be done in the subsequent steps to remedy the clustering result. This is the reason why a substantial amount of research has been devoted to find an optimal way to answer this question.

To illustrate the drawbacks of the approaches already in the literature, we can visually inspect the result of each of them in a simple example. In Fig. 5.1, we illustrate the clustering result of partitioning in two clusters a dataset composed of a mixture of 30 Gaussian distributions. Fig. 5.1 (top left) corresponds to the first step of an algorithm that splits based on the sign of the projections onto the first principal component (criterion SPC_1). Fig. 5.1 (top right) shows the equivalent result when the partitioning is performed based on the sign of the projections onto the second principal component, as proposed in [Nil02]. Finally, Fig. 5.1 (bottom) depicts the result of the partitioning using k-means steering, where the best splitting point is calculated exhaustively as proposed in [ZG07]. As shown, splitting utilising the sign results in sub-optimal decisions, since a large number of compact cluster are split in half in both cases. Also using k-means steering exhibits a marginal improvement.

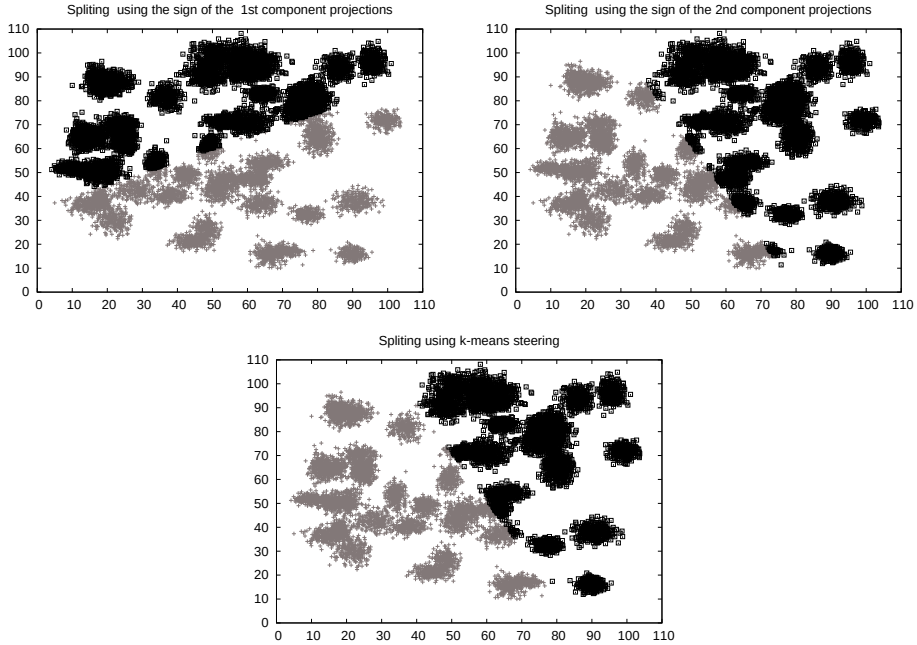


Figure 5.1: *The partitioning result of different splitting criteria.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

5.2.1 Contiguous Clusters

Trying to investigate the situations in which a criterion can be optimal, we first need to formally define a cluster. Different cluster definitions exist in the literature [SEK03]. The one employed here is used for *contiguous* clusters, where any point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster. Formally we define first the coherence of a set as follows:

Definition 2. *(***Set Coherence***): Let $D \in R^{n\times a}$ be a data matrix corresponding to a set of points $\mathcal{D}$. Then the coherence of the $\mathcal{D}$ is defined as $COH(\mathcal{D}) = max\{\|d_i - d_{i^*}\| : \forall i = 1,\ldots,n\}$, where d_i are the row vectors of D and $i^* = \arg\min_j\{\|d_i - d_j\| : \forall j = 1,\ldots,n,$ and $j \neq i\}$*

Now, we can formally define a cluster as a contiguous set:

Definition 3. *(***Contiguous Set***): Let $\mathcal{D}$ set of points d_i, for $i = 1,\ldots,n$, a subset $\mathcal{C} \subset \mathcal{D}$ is a Contiguous set, when $COH(\mathcal{C}) < COH(\mathcal{C} \cup \{d_i\})$, $\forall d_i \in \mathcal{D}/\mathcal{C}$.*

In the above definition, COH($\mathcal{C}$) is the coherence of the set $\mathcal{C}$ and $\mathcal{D}/\mathcal{C} = \{d \in \mathcal{D} : d \notin \mathcal{C}\}$. Using such definitions we can now define the clustering problem.

Definition 4. *(***Contiguous Clusterable Set***): A set $\mathcal{D}$ of points is k* **contiguous clusterable** *if there exists a partition Π of $\mathcal{D}$ into k subsets $\mathcal{C}_1,\ldots,\mathcal{C}_k$, such that any $\mathcal{C}_i$ is a Contiguous Set.*

Thus, we can prove the following Lemma, stating that if in a data projection the distance between two consecutive projections is at least M, then the points to the left of the interspace cannot lie in the same cluster with the points in the right, where M is the maximum of the cluster coherencies. As such, splitting on any point in that space is optimal in terms of the partition Π. In Fig. 5.2 such an example is illustrated. The distance L between the projections p_{t^*} and p_{l^*} is larger than both the cluster coherencies, so splitting the data on either p_{t^*} or p_{l^*} guarantees that each partition will contain points from different true clusters.

Lemma 5.2.1. *Let a k-contiguous-clusterable set $\mathcal{D}$ of points d_i, for $i = 1,\ldots,n$ and Π its partition into k Contiguous subsets $\mathcal{C}_1,\ldots,\mathcal{C}_k$. Let $u \in \mathbb{R}^a$, with $\|u\| = 1$. Let $\mathcal{P}$ be the set of projections p_i of the vectors d_i on u. Also, let $M = \max\{COH(\mathcal{C}) : \mathcal{C} \in \Pi\}$. If there is a $p_{l^*} \in \mathcal{P}$ such that $p_{t^*} - p_{l^*} > M$, where $p_{t^*} = \min\{p : p > p_{l^*}, p \in \mathcal{P}\}$, then all d_t, for which $p_t > p_{l^*}$, and all d_l, for which $p_l \leqslant p_{l^*}$, belong to different sets of Π.*

Proof. First note that for any d_l and d_t, with $l \neq t$, it holds that:

$$\|d_l - d_t\| \geqslant |p_l - p_t|. \tag{5.1}$$

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

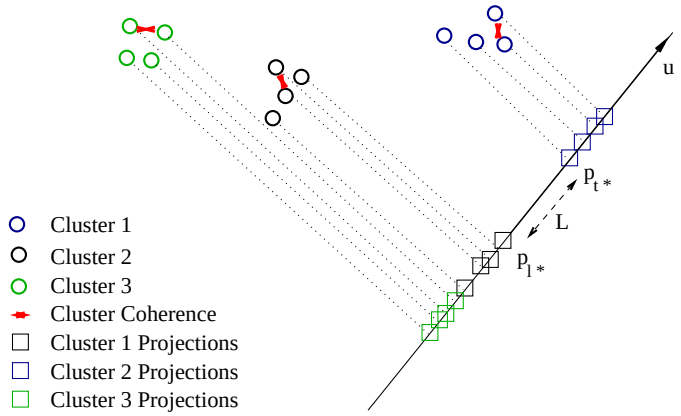


Figure 5.2: *An example application of Lemma 5.2.1.*

Since $p_{t^*} - p_{l^*} \geqslant M$, for any d_l with $p_l \leqslant p_{l^*}$ and for any d_t with $p_t > p_{l^*}$ it holds that:

$$p_t - p_l \geqslant p_{t^*} - p_{l^*} > M.$$

Thus from Eq. (5.1) we have:

$$\|d_l - d_t\| \geqslant |p_l - p_t| > M. \tag{5.2}$$

Since $\mathcal{D}$ is k-contiguous-clusterable, $\text{COH}(\mathcal{C}) \leqslant M$, for all $\mathcal{C} \in \Pi$. If there was a $\mathcal{C} \in \Pi$, such that $d_l, d_t \in \mathcal{C}$, then from Definition 2, it would have to hold $\|d_l - d_t\| \leqslant M$. This contradicts Eq. 5.2, thus d_l and d_t need to be in different clusters. Thus, the lemma is proved. □

In real life scenarios however cluster coherencies cannot be known in advance. Also even if they were known, it could be difficult to find a large enough empty space in a particular projection.

Even if we do not know the real cluster coherencies, we can still claim that by splitting on the largest distance between two consecutive projections is a good choice, given the particular projection, as it does not split any contiguous cluster. In the lemma that follows we show in more detail what can be achieved by this splitting criterion.

Lemma 5.2.2. *Let $\mathcal{D}$ be a set of n points $d_i \in \mathbb{R}^a$, for $i = 1, \ldots, n$ and $u \in \mathbb{R}^a$, with $\|u\| = 1$. Let $\mathcal{P}$ be the set of projections p_i, for $i = 1, \ldots, n$, of the vectors $d_i \in \mathcal{D}$ on u. Also, let for each $p_i \in \mathcal{P}$, $ne(i) = \arg\min_j\{p_j : p_j > p_i,\ p_j \in \mathcal{P}\}$ and define i_{sp} as follows:*

$$i_{sp} = \arg\max_i\{p_{ne(i)} - p_i, \forall p_i \in \mathcal{P}\}.$$

Then, if $\mathcal{D}_{i_{sp}} = \{d_i \in \mathcal{D} : p_i \leqslant p_{i_{sp}}\}$, $M = p_{i_{sp}} - p_{j^*}$, where $p_{j^*} = \min\{p_j \in \mathcal{P} : p_j > p_{i_{sp}}\}$, then for any subset $\mathcal{D}'_{i_{sp}} \subset \mathcal{D}_{i_{sp}}$ for which $COH(\mathcal{D}'_{i_{sp}}) \leqslant M$, it holds that:

$$COH(\mathcal{D}'_{i_{sp}}) \leqslant COH(\mathcal{D}'_{i_{sp}} \cup \{d_j\})$$

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

for any $d_j \in \mathcal{D}$ *for which* $p_j > p_{i_{sp}}$.

Proof. First note that for $d_i \in \mathcal{D}'_{i_{sp}}$, it holds that $p_i \leqslant p_{i_{sp}}$. This way, for any $d_j \in \mathcal{D}$ such that $p_j \geqslant p_{j^*}$ it holds that $p_i - p_j \geqslant M$. Subsequently, it also holds $\|d_i - d_j\| \geqslant M$. As such, from Definition 2, $\text{COH}(\mathcal{D}'_{i_{sp}} \cup \{d_j\}) \geqslant M$. Since $\text{COH}(\mathcal{D}'_{i_{sp}}) \leqslant M$, the Lemma is proved. □

An example of the application of this Lemma is illustrated in Fig. 5.3. In this example $\text{COH}(\mathcal{D}_{i_{sp}})$, is larger than M, but the coherence of the subsets $\text{COH}(\mathcal{D}_{i_{sp,1}})$, $\text{COH}(\mathcal{D}_{i_{sp,2}})$, and $\text{COH}(\mathcal{D}_{i_{sp,3}})$, is smaller than M. Adding any point d_j with $d_j \in \mathcal{D}$ for which $p_j > p_{i_{sp}}$, would only increase the coherence of any of those subsets, although the coherence of $\mathcal{D}_{i_{sp}}$ will decrease if we add p_{j^*}, since it holds that $L_1 < L_2, L_3$, and $L_1 = \text{COH}(\mathcal{D}_{i_{sp}})$.

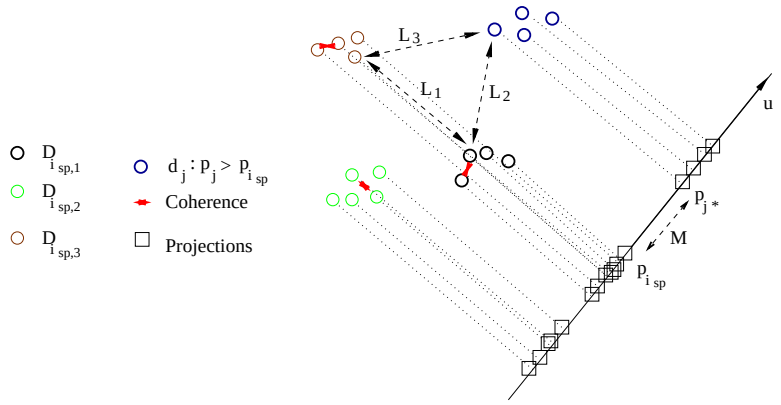


Figure 5.3: *An example application of Lemma 5.2.2.*

The above lemmas point out that partitioning data based on the largest distance M between consecutive projections is guaranteed not to split a cluster, if the maximum coherence of each cluster is less than M. In the case that such knowledge is not available, splitting in such a manner guarantees that no cluster with less than M coherence is split. In real life data sets, we do not how dense the real clusters are. Thus, one would like to discover clusters with coherence as large as possible. Given a particular projection, obviously the best strategy is to split on the largest distance between two consecutive projections. As proposed in [TT08], let us formally define the following splitting criterion:

- (Splitting Criterion SPC_2): Let $\forall p_i \in \mathcal{P}$, $\text{ne}(i) = \arg\min_j\{p_j : p_j > p_i,\ \forall p_j \in \mathcal{P}\}$ and define i_{sp} as follows $i_{sp} = \arg\max_i\{p_{\text{ne}(i)} - p_i, \forall p_i \in \mathcal{P}\}$. Then, $P_1 = \{d_i \in \mathcal{D} : p_i \leqslant p_{i_{sp}}\}$ and $P_2 = \{d_i \in \mathcal{D} : p_i > p_{i_{sp}}\}$.

Fig 5.4 illustrates the result of this criterion on the exemplary dataset used previously, where alternative criteria in the literature produce sub-optimal partitions. As shown, no cluster is split and only one of the points

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

of one cluster is attributed to a wrong partition. Note also, that the result is heavily unbalanced in terms of number of points in each partition, where the alternative criteria produce somewhat balanced partitions.

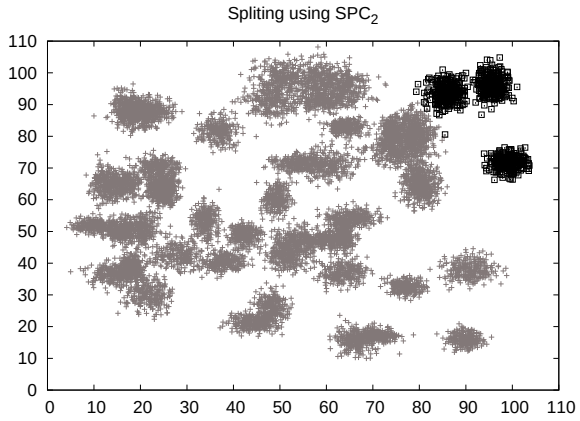


Figure 5.4: *The partitioning result of SPC_2 splitting criterion.*

5.2.2 Dense Convex Clusters

Although the SPC_2 criterion might seem appropriate in terms of cluster coherence, its main drawback is that it will operate sub-optimally in cases where there are outlying points in the data. In such situations all the distances between successive projections can be uniform and it could be impossible to make an informative splitting decision. However, this is not a problem of the particular criterion per se, but rather of the cluster definition that has been used. Nonetheless, we would like to be able to deal with such situations. In what follows, we will describe an alternative formulation of the clustering problem that allows us to design a splitting methodology able to deal with this situation. So, we are redefining clusters as convex subsets of the data space that are more dense than the regions around them.

Definition 5. *(***Dense Convex Set***): If we assume $f : \mathbb{R}^a \to \mathbb{R}$ to be the probability density function of $x \in \mathbb{R}^a$, then a set $\mathcal{C} \subset \mathbb{R}^a$ is dense and convex if there is a set $\mathcal{C}' \subset \mathbb{R}^a$ such that $f(x) > f(y)$, $\forall x \in \mathcal{C}$, and $\forall y \in \mathcal{C}'$ and $\forall x, y \in C$ then $(1-t)x + ty \in C$, $\forall t \in [0,1]$.*

The above definition determines clusters not in terms of the points they include, but their probability density function f. This way we can assume the data set $\mathcal{D}$, as a finite sample drawn from that distribution. Similarly, we can redefine the clustering problem as follows:

Definition 6. *(***Dense Convex Clusterable Set***): A set $\mathcal{D}$ of points is* **Dense Convex k-Clusterable** *if there exist k disjoint subsets $\mathcal{C}_1, \ldots, \mathcal{C}_k$, such that the convex hull $H(\mathcal{C}_i)$ of any $\mathcal{C}_i$ is a dense convex set.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

In practice, it is not possible to know in advance the real density of the data f, but we can construct estimates of it given the data set $\mathcal{D}$. There are approaches for such a task and among them *Kernel Density Estimation* [WJ95] is a well studied non-parametric estimation that does not assume any particular form for the real density f.

In the most simple forms for a multivariate kernel density estimator is:

$$\hat{f}(x;h) = n^{-1}h^{-a}\sum_{i=1}^{n} K((x-d_i)/h),$$

where $h \in \mathbb{R}^+$ is a real positive number called the bandwidth and $K : \mathbb{R}^a \to \mathbb{R}$ is a *kernel* function which needs to satisfy: $\int K(x)dx = 1$. There are several kernel functions that can be used, like uniform, triangular, Epanechnikov, normal, and others. However the quality of a kernel estimate depends less on the shape of the K than on the value of its bandwidth h. A usual choice for the kernel is the standard multi-variate normal density:

$$K(x) = (2\pi)^{-a/2}e^{-0.5\|x\|}.$$

For the estimation properties of such a method refer to [WJ95].

For any projection direction $u \in \mathbb{R}^a$, with $\|u\| = 1$, we can also estimate the univariate density of the projections $ux \in \mathbb{R}$, for $x \in \mathbb{R}^a$:

$$\hat{f}'(ux;h) = n^{-1}h^{-1}\sum_{i=1}^{n} K\left((ux-p_i)/h\right).$$

Now, the kernel function used is:

$$K'(ux) = (2\pi)^{-1/2}e^{-0.5|ux|}.$$

Note that $K'(ux) \geqslant K(x)$, since $\|x\| \geqslant |ux|$, and the normal density is a monotonically decreasing function. Moreover:

$$\hat{f}(x;h) \leqslant \hat{f}'(ux;h'), \tag{5.3}$$

as long as $h = h' > 1$ or $h' \leqslant h^a$.

Although kernel density estimation technique is a quite general and powerful method, has a significant disadvantage of being computationally intensive. In general, given n data samples and m points at which the density need to be evaluated, the computational complexity is $O(nm)$

However, it has been shown [YDGD03, GS91] that using techniques like the Fast Gauss Transform (FGT) we can achieve linear running time for the Kernel Density Estimation, especially for the one dimensional case at hand. The fast Gauss transform is an analysis based fast algorithm in the sense that it speeds up the computation by approximating the Gaussian function to achieve a desired precision.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

The performance of FGT degrades exponentially with increasing dimensionality, which makes it impractical for several applications in Machine Learning. While methods have been developed for improving FGT in order to be applicable in higher dimensions [RYDG05], in our benefit, in this work the FGT method is being applied on data samples of reduced dimensionality. Thus, there is no need to use such methods.

Using this estimation methodology, the following lemma can be proved:

Lemma 5.2.3. *Let a dense convex k-clusterable set $\mathcal{D}$ of points d_i, for $i = 1, \ldots, n$ and Π its partition into k dense convex subsets $\mathcal{C}_1, \ldots, \mathcal{C}_k$. Let $u \in R^a$, with $\|u\| = 1$. Let $\mathcal{P}$ be the set of projections p_i of the vectors d_i on u. Also, let $M = \min\{\hat{f}(x;h) : x \in \mathcal{C}, \mathcal{C} \in \Pi\}$. If there exists $x \in \mathbb{R}$, such that the univariate estimate density $\hat{f}'(x;h') < M$, with $h = h' > 1$ or $h' \leqslant h^a$, then all d_t for which $p_t > x$, and d_l for which $p_l \leqslant x$, belong to different sets of Π.*

Proof. Let us suppose that there exist $d_t, d_l \in \mathcal{D}, \mathcal{C} \in P$ such that $d_t, d_l \in \mathcal{C}$, and $p_t = ud_t$ and $p_l = ud_l$ with $p_l \leqslant x$ and $p_t > x$. Obviously, there exists $\lambda \in [0,1]$, such that $x = \lambda p_l + (1-\lambda)p_t$. Also for the vector $z \in \mathbb{R}^d$ with $z = \lambda d_l + (1-\lambda)d_t$, it holds that $z \in \mathcal{C}$ since $\mathcal{C}$ is convex (note that $uz = x$).

As such $\hat{f}'(uz;h') = \hat{f}'(x;h') < M$. However, from Eq. 5.3, we have that $\hat{f}(z;h) \leqslant \hat{f}'(uz;h') = \hat{f}'(x;h')$ and since $z \in \mathcal{C}$, $\hat{f}(z;h) > M$, this is a contradiction. Thus, the Lemma is proved. □

This lemma extends the result of Lemma 5.2.1 in the case of dense convex clusterable sets. Now, all we need to find is a point on the projection line with low enough density. If we split based on that point we are guaranteed not to split any clusters. Of course the above Lemma assumes that the minimum density of the points in each cluster is known. This assumption could be unattainable in real life scenarios. Nevertheless, given a particular projection we could find the global minimiser x^* of the projections density $\hat{f}'(x;h')$ and split based on that point. That helps not to split any clusters of density greater or equal to $\hat{f}'(x;h')$. However, x^* must be formally defined:

Definition 7. *(**Global Minimiser**): A global minimiser $x^* \in \mathbb{R}$, is the point that $\hat{f}'(x^*;h') \leqslant \hat{f}'(x';h')$, where $x' \in \mathcal{X} = \{x'' \in R : \exists \delta > 0, \hat{f}'(x'';h') < \hat{f}'(x;h'), \forall x$ for which $|x'' - x| < \delta\}$.*

Based on that definition the following splitting criterion is proposed:

- (Splitting Criterion SPC_3): Let $\hat{f}'(x;h')$ be the kernel density estimation of the density of the projections $p_i \in \mathcal{P}$ and x^* its global minimiser given by Definition 7. Then construct $P_1 = \{d_i \in \mathcal{D} : p_i \leqslant x^*\}$ and $P_2 = \{d_i \in \mathcal{D} : p_i > x^*\}$.

Note, that finding such a minimiser might seem more difficult than it actually is. The reason for that lies in the fact that valid splitting points lie

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

between consecutive data projections p_i. Also, between any two consecutive projections p_i and p_{i+1} the estimated density $\hat{f}'(x;h')$, for $p_i \leqslant x \leqslant p_{i+1}$, is monotonous, due to the manner that this is calculated. As such we can constrain the search for the global minimiser on the finite set $\mathcal{P}$ and set δ for each p_i, as the minimum of $|p_{i+1} - p_i|$ and $|p_{i-1} - p_i|$, where p_{i-1}, and p_{i+1} are the two nearest projections of p_i from either side.

Let us examine the result of this criterion on the exemplary dataset used before. Fig. 5.5(top) illustrates the result that is similar to the one obtained by SPC_2. The difference lies in the fact that the outlying point that was misplaced under SPC_2, does not affect the SPC_3 criterion and it is assigned to the correct sub-partition. It is interesting to inspect the estimated density shown in Fig. 5.5(bottom). The estimate density exhibits several minima. However, especially one of those minima has a very low density value (very close to 0) that provides a quite good indication that no cluster will be split.

This splitting criterion suggests that the minimiser of the estimated density is the best we can do to avoid splitting clusters. However, there always exists the problem of not being able to find such a minimiser. This could happen in two cases; either the data do not contain any density based convex clusters, or the selected bandwidth is very large [WJ95]. The first case can act as an indication when to stop the recursive splitting of the data, under the assumption that the density estimation is accurate enough. The second case, is a well studied problem, especially in the particular one dimensional situation [Tur93]. Later in Section 5.8, we investigate how different bandwidth values affect the performance of the clustering procedure.

§ 5.3 Which cluster to split?

As explained in Section 4.1.5, the second question that any divisive algorithm needs to face is which cluster to select from the pool of already retrieved partitions, to forego with the clustering procedure. It is obvious that this step of the procedure is quite important as well. Being able to split effectively has no merit, when the selected cluster to split corresponds to a coherent group. Imposing a clustering structure on a dataset, when such a structure does not exist, is a central issue in cluster analysis. In general, determining the presence or the absence of a clustering structure is called the "cluster tendency" problem. A series of specialised tests have been developed to judge the existence of a clustering structure, before the application of a clustering algorithm, like the scan test, quadrant analysis, moment structure and inter-point distances [TK06]. However, it still remains an open issue.

As already discussed in Section 4.1.5, the PDDP algorithm, as well as all its variations [ZG03, Dhi01, DKN03, Nil02, ZG07], select the cluster with maximum scatter value SV (Eq. 4.1). Another explanation of this selection

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

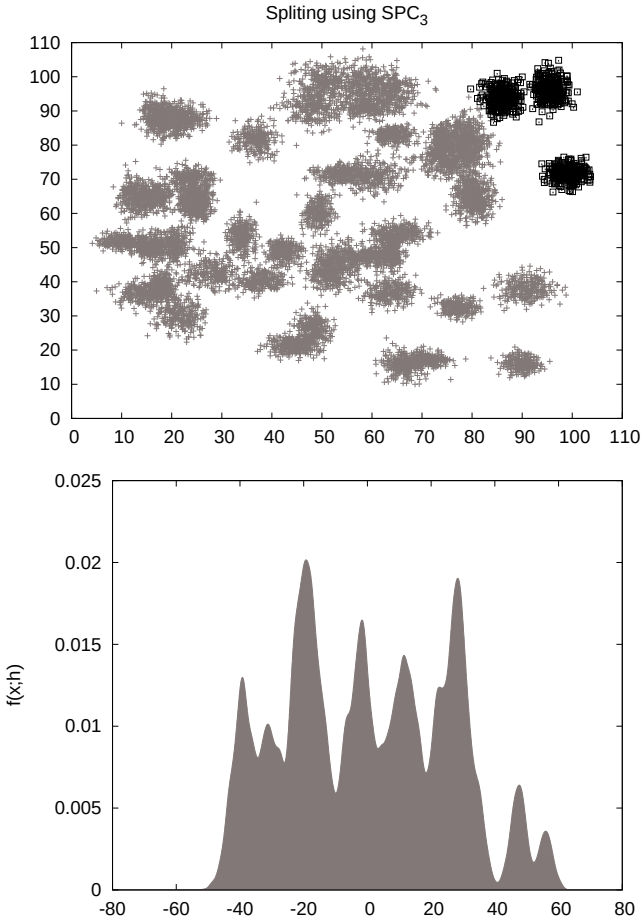


Figure 5.5: *The partitioning result of SPC_3 splitting criterion (top) and the corresponding density estimation of the projections (bottom).*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

method is that the cluster having in effect the widest (largest variance) projection on the first principal component is chosen ("shoot the biggest animal").

This can be very easily deemed unsuccessful through a simple example as shown in [TT08], which for completeness purposes is included here as well. Fig. 5.6 illustrates a case where the dataset has already been split into two clusters shown with different colours. In this case, the green large cluster has a SV value of 0.25414, which is 0.17945 larger than the SV value of the red cluster. So, in this case selecting the cluster with the larger SV value for further splitting would have no chance of producing a correct clustering.

The splitting criteria proposed in the previous section could also provide guidance for the cluster selection step. In the pool of already retrieved clusters, we expect the one with the largest distance among consecutive projections to probably contain more than one actual clusters.

The same can be said for the minimum estimated density criterion. A minimiser with very small density should be a good indicator of multimodality of the density function and consequently it lessens the chance to split an actual cluster. Thus we propose the following two selection criteria:

- (Cluster Selection Criterion CS_1): Let Π a partition of the data set $\mathcal{D}$ into k sets. Let $\mathcal{M} = \{\mathcal{M}_\rangle : \rangle = \infty, \ldots, \|\}$ be the set of the largest distances M_i among consecutive projections, for each $\mathcal{C}_i \in \Pi$, $i = 1, \ldots, k$. The next set to split is C_j, with $j = \arg\max_i\{M_i : M_i \in \mathcal{M}\}$.

- (Cluster Selection Criterion CS_2): Let Π a partition of the data set $\mathcal{D}$ into k sets. Let $\mathcal{F}$ be the set of the density estimates $f_i = \hat{f}(x_i^*; h)$ of the minimisers x_i^* for the projection of the data of each $C_i \in \Pi$, $i = 1, \ldots, k$. The next set to split is C_j, with $j = \arg\max_i\{f_i : f_i \in \mathcal{F}\}$.

§ 5.4 When should the iteration terminate?

Irrespectively of the method used, a fundamental issue in cluster analysis is the determination of the number of clusters present in a dataset. This issue also remains an open problem in cluster analysis. For instance, well-known and widely used iterative techniques, such as the k-means algorithm [HW79], require from the user a priori designation of the number of clusters present in the dataset. There also exist approaches that adjust the number of clusters during training. The ISODATA technique [BH67] is based on the same key idea as the k-means algorithm; starting with a typical number of initial clusters, it iteratively merges and splits existing clusters according to "within-group variability" and "closeness" thresholds. Another popular approach is to employ Akaike Information Criterion (AIC) and Bayesian Information

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

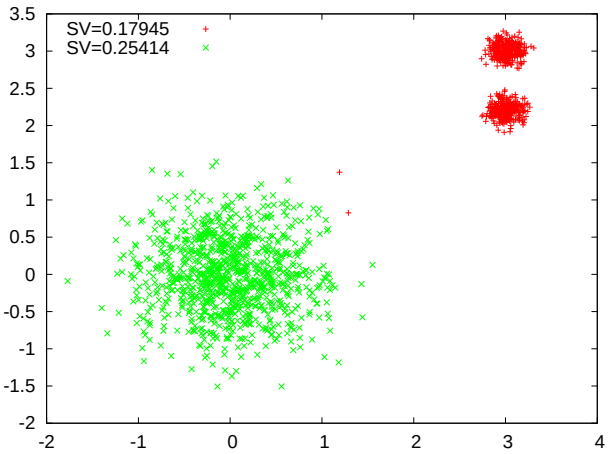


Figure 5.6: *A dataset of unbalanced clusters and the corresponding scatter values.*

Criterion (BIC) to choose among partitions with different number of clusters. In the same theme the Integrated Completed Likelihood (ICL) criterion has been proposed as more appropriate for clustering purposes [BCG00]. In this category of algorithms PG-means [FHE07] aims to learn the number of components of a Gaussian Mixture Modelling approach, using statistical hypothesis tests on one-dimensional projections of the data. The computationally efficient x-means algorithm, proposed in [PM00], is another popular approach from the class of partitioning algorithms that has the ability to approximate the number of clusters in the data.

One of the most promising approaches from the density based category of algorithms is DBSCAN [SEKX98]. DBSCAN is a density-based clustering algorithm that tries to recover clusters from spatial databases and automatically decides the number of clusters. Clusters are defined by means of neighbourhoods of objects. The density of each such neighbourhood has to exceed some threshold. The value of the threshold is critical for the execution of the algorithm and heuristics have been proposed to determine it. Finally, there exist some recent agglomerative hierarchical clustering algorithms that have been shown to be able to achieve high quality clustering results, such as BIRCH [ZRL96], CHAMELEON [GRS98] and CURE [KHK99]. However, these type of algorithms require higher user intervention to provide accurate estimations for the cluster number.

There also exist a few grid-based algorithms [HK99, AGGR05], that have been shown to be able to produce good clustering results. One of the most notable of these is CLIQUE [AGGR05]. Their biggest drawback is that they have a running time that is exponential to the data dimensionality. To be

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

more precise, they are exponential not to the actual data dimensionality, but to the dimensionality of the subspaces where the clusters reside and is possibly much smaller than the full data dimensionality. So they are more fitted to operate on cases where specific clusters lie on few dimensions and medium data dimensions (e.g. 20, 40), and not in cases where dimensions lie in the range of thousands.

Little has been done however to develop an efficient technique for PDDP based approaches. The crudest approach would be to stop the execution when all the discovered clusters have a scatter value that is smaller than a predefined value, but the tuning of this parameter can be difficult from a user perspective. The criteria used in other algorithms could also be employed in the PDDP case. For example in [KSI03] it is proposed to use BIC to determine if a further split would improve the clustering result or not. Additionally, we could use nearest neighbour statistics like the ones used in cluster tendency [TK06, You82].

In [TT08], a termination criterion based on the maximum distance between consecutive projections was proposed. More specifically, we propose to have a maximum number of allowed clusters k_{max} as an upper bound and subsequently continue splitting as long as there exists clusters with more than $MinPts$ points, where $MinPts$ is a user defined parameter to describe the minimum number of points that are allowed to constitute a valid cluster. Notice that this is not an uncommon procedure for algorithms that are designed to deal with noisy datasets [SEKX98]. In this case, it is indirectly assumed that the distances between two outliers are larger than any two points of a cluster. Formally the termination criterion is the following based on the two user defined parameters k_{max} and $MinPts$:

- (Stopping Criterion ST_1): Iteratively split the data set $\mathcal{D}$ into k_{max} subsets. Report as clusters the ones with more than $MinPts$ points. Designate the remaining points as outliers.

For the density based approach presented here, we could allow the existence of a minimiser to guide the termination of the procedure. We can stop the iteration as long as no minimiser exists for any of the retrieved clusters. This stopping criterion makes the assumption that all the retrieved clusters are uniformly dense, so they cannot be further split. Note however, that this depends on the bandwidth selection and automated bandwidth selection techniques could be employed to remove user intervention. Formally:

- (Stopping Criterion ST_2): Let Π a partition of the data set $\mathcal{D}$ into k sets. Let $\mathcal{X}$ be the set of minimisers x_i^* of the density estimates $\hat{f}(x_i^*; h)$ of the projection of the data of each $C_i \in \Pi$, $i = 1, \ldots, k$. Stop the procedure when the set $\mathcal{X}$ is empty.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

§ 5.5 Algorithms

In this section the algorithmic constructions are presented based on the different proposed criteria in the earlier sections. These particular constructions are just employed to investigate (through experimental analysis) the effectiveness of these criteria. In principle, more algorithms having different characteristics can be constructed using any combination of criteria. The particular implementations do not try to mix different approaches, but are pure in the sense that they employ criteria from a particular methodology. Thus, we have two algorithmic implementations. The first implementation (iPDDP) is based on the idea of splitting based on the largest distance between any two consecutive projections. The second implementation (de-PDDP) is a compilation of the criteria that are based on the minimiser of the estimated density of the projections.

The iPDDP implementation is shown in Table 5.1. This algorithmic schema is build around the stopping criterion ST_1, the cluster selection criterion CS_1, and the spitting criterion SPC_2. This has the drawback that the user needs to specify the maximum number of clusters k_{max} and the $MinPts$ parameter to describe the minimum number of points that are allowed to constitute a valid cluster. Although the selection of $MinPts$ parameter is easy, the selection of k_{max} is not that straightforward. If k_{max} is selected to obtain a much larger value than the actual number of clusters the algorithm will be forced to iteratively strip points from the clusters. In some extreme cases, a cluster could be totally decomposed into smaller sets of less than $MinPts$ points, eventually ending up in the outlier set. The effect of these parameters is further investigated in Section 5.7.

Function iPDDP $(\mathcal{D}, k_{max}, MinPts)$ {
1. Set $\Pi = \{\mathcal{D}\}$
2. While $|\Pi| < k_{max}$, do
3. Select a set $\mathcal{C} \in \Pi$, using cluster selection criterion CS_1
4. Split $\mathcal{C}$ into two sub-sets $\mathcal{C}_1$ and $\mathcal{C}_2$, using Splitting Criterion SPC_2
5. Remove $\mathcal{C}$ from Π and set $\Pi \rightarrow \Pi \cup \{\mathcal{C}_1, \mathcal{C}_2\}$
6. Set $\mathcal{O} = \emptyset$
7. For any $\mathcal{C}$ in Π with $|\mathcal{C}| < MinPts$, do
8. Remove $\mathcal{C}$ from Π
9. Set $\mathcal{C}$ with $|\mathcal{C}| < MinPts$ and set $\mathcal{O} \rightarrow \mathcal{O} \cup \mathcal{C}$,
10. Return Π the partition of $\mathcal{D}$ into $|\Pi|$ clusters and $\mathcal{O}$ the set of outliers.
}

Table 5.1: *The iPDDP algorithm summary.*

The computational complexity of the iPDDP implementation is mostly

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

influenced by the computation of the principal vectors as in the original PDDP algorithm. To compute this, the Singular Value Decomposition of the data matrix D is employed. This introduces a total worst case complexity of $\mathrm{O}(k_{max}(2+k_{SVD})s_{nz}\ n\ a)$, where k_{SVD} are the iterations needed by the Lanczos SVD computation algorithm and s_{nz} is the fraction of non-zero entries in D (for more details refer to [Bol98]). In the iPDDP case, the additional computation steps that are required, increase the complexity to $\mathrm{O}(k_{max}(2+k_{SVD})(s_{nz}n\ a+n\log(n)))$, which although increased is still on par with the most clustering algorithms. Notice that the additional cost is not influenced by the data dimensionality. Thus, the ability of the algorithms to deal with ultra high dimensional data is maintained.

The dePDDP implementation is shown in Table 5.2. This is just a compilation of the criteria SPC_3 for the cluster splitting, CS_2 for the cluster selection, and ST_2 for the termination of the algorithm. The computational complexity of this approach, using a brute force technique, would be quadratic in the number of samples. However, as shown in Section 5.2.2 using techniques like the Fast Gauss Transform we achieve linear running time for the Kernel Density Estimation. To find the minimiser we only need to evaluate the density at n positions, in between the projected data points, since those are the only places we can have valid splitting points. Thus, the total complexity of the algorithm remains $\mathrm{O}\left(k_{max}(2+k_{SVD})(s_{nz}n\ a)\right)$.

Function dePDDP ($\mathcal{D}$) {
1. Set $\Pi = \{\mathcal{D}\}$
2. Do
3. Select a set $\mathcal{C} \in \Pi$, using cluster selection criterion CS_2
3. Split $\mathcal{C}$ into two sub-sets $\mathcal{C}_1$ and $\mathcal{C}_2$ using Splitting Criterion SPC_3
4. Remove $\mathcal{C}$ from Π and set $\Pi \rightarrow \Pi \cup \{\mathcal{C}_1, \mathcal{C}_2\}$
5. While Stopping Criterion ST_2 is not satisfied
6. Return Π the partition of $\mathcal{D}$ into $|\Pi|$ clusters
}

Table 5.2: *The dePDDP algorithm summary.*

§ 5.6 Experimental Analysis

This section is devoted to the experimental evaluation of the algorithmic implementations presented in the previous section.

Table 5.3 reports the purity and V-measure (see Section 2.7) of the PDDP, iPDDP, dePDDP, KM-PDDP [ZG07], Gaussian Mixture Model (GMM), and k-means algorithms in 100 randomly generated datasets, using the

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

$DSET_{\text{Gausian}}$ mechanism (see Section 2.7). Each entry of the table is the mean observed value of the corresponding measure obtained over the 100 different datasets and the number in parentheses is the observed standard deviation. A standard deviation of 0 corresponds to a observed variance of less than 10^{-2}.

Dimension 2						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.88 (0.04)	0.89 (0.03)	0.83 (0.03)	0.86 (0.02)	0.79 (0.02)	0.86 (0.01)
iPDDP	0.89 (0.03)	0.88 (0.06)	0.89 (0.02)	0.77 (0.15)	0.93 (0.04)	0.38 (0.24)
dePDDP	0.95 (0.03)	0.96 (0.02)	0.93 (0.03)	0.93 (0.02)	0.86 (0.02)	0.91 (0.01)
KM-PDDP	0.91 (0.04)	0.93 (0.03)	0.88 (0.03)	0.79 (0.22)	0.86 (0.03)	0.71 (0.06)
GMM	0.93 (0.03)	0.94 (0.03)	0.87 (0.02)	0.91 (0.02)	0.80 (0.01)	0.90 (0.00)
k-means	0.92 (0.04)	0.93 (0.03)	0.90 (0.03)	0.92 (0.02)	0.84 (0.00)	0.90 (0.00)
Dimension 5						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.94 (0.03)	0.95 (0.02)	0.92 (0.03)	0.94 (0.02)	0.89 (0.02)	0.92 (0.02)
iPDDP	1.00 (0.01)	1.00 (0.01)	0.98 (0.02)	0.98 (0.03)	0.97 (0.01)	0.66 (0.25)
dePDDP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.00)
KM-PDDP	0.98 (0.02)	0.99 (0.01)	0.96 (0.03)	0.97 (0.02)	0.91 (0.05)	0.76 (0.33)
GMM	0.96 (0.02)	0.97 (0.01)	0.92 (0.02)	0.96 (0.01)	0.90 (0.02)	0.96 (0.01)
k-means	0.93 (0.02)	0.94 (0.02)	0.93 (0.02)	0.95 (0.01)	0.93 (0.01)	0.96 (0.01)
Dimension 20						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.97 (0.02)	0.97 (0.02)	0.96 (0.02)	0.95 (0.04)	0.96 (0.01)	0.94 (0.03)
iPDDP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.01)	0.92 (0.20)
dePDDP	1.00 (0.00)	0.99 (0.01)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.00)
KM-PDDP	0.98 (0.03)	0.99 (0.02)	0.96 (0.03)	0.98 (0.01)	0.90 (0.03)	0.92 (0.12)
GMM	0.93 (0.03)	0.94 (0.02)	0.90 (0.02)	0.94 (0.01)	0.90 (0.01)	0.95 (0.01)
k-means	0.94 (0.02)	0.96 (0.02)	0.93 (0.02)	0.95 (0.01)	0.92 (0.01)	0.95 (0.01)
Dimension 50						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.97 (0.02)	0.97 (0.03)	0.97 (0.02)	0.97 (0.02)	0.97 (0.01)	0.94 (0.02)
iPDDP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.01)	1.00 (0.01)
dePDDP	1.00 (0.00)	0.99 (0.01)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.00)
KM-PDDP	0.95 (0.03)	0.97 (0.02)	0.93 (0.03)	0.96 (0.01)	0.90 (0.02)	0.96 (0.01)
GMM	0.89 (0.03)	0.89 (0.03)	0.88 (0.02)	0.91 (0.02)	0.87 (0.02)	0.91 (0.01)
k-means	0.94 (0.02)	0.95 (0.01)	0.93 (0.02)	0.95 (0.01)	0.92 (0.01)	0.95 (0.01)

Table 5.3: *Mean purity and V-measure (with the observed standard deviation in parenthesis), for* $DSET_{\text{Gausian}}$ *generated data of different algorithms, over 100 experiments.*

As shown, the performance of PDDP, KM-PDDP, and GMM algorithms deteriorates as the number of clusters increases. Regarding the V-measure, the same is true only for that low dimensional cases. The reason for this lies in the fact that in low dimensions a kind of "chaining" effect [SEKX98] can appear between clusters. On the other hand, the dePDDP algorithm is marginally affected and produces high quality results in all cases. Note that

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

the $MinPts$ parameters for iPDDP was set to 5 and k_{max} was set to the actual number of clusters across all experiments. The actual cluster number was also given as input to all algorithms except dePDDP. The bandwidth parameter of dePDDP was set to twice the optimal value given by Eq. 5.8. Further explanation for this value is given in Section 5.8. Notice that this does not guarantee that the algorithm will stop to the actual cluster number, but this issue will be further explored below (see Section 5.6.2).

Next Table 5.4 reports the purity and V-measure of the algorithms in 100 randomly generated datasets, using the described (see Section 2.7) $DSET_{\text{Beta}}$ mechanism. Again, the reported values are the mean observed measure values in 100 different experiments and the number in parenthesis is the standard deviation. The results are similar to the $DSET_{\text{Gausian}}$ case. The exception is GMM as it shows worse results, especially in the low dimensional high cluster number case. This is expected since the assumption of a Gaussian mixture on which GMM operates does not hold, so cluster recovery is hindered. However, all the PDDP variants show similar results, with dePDDP to produce very accurate clusterings. We have also followed the same procedure for the t and log-normal distributions but the results are very similar so they are not included for brevity.

To facilitate a more direct understanding of the results of each algorithm, we will use two 2-dimensional datasets that are publicly available at http://cs.joensuu.fi/sipu/datasets/ and will resort to visual inspection of the results. These datasets S1 and S4, contain 5000 data points in 15 Gaussian clusters with different degree of cluster overlapping. In this case, we will test two additional algorithms. The first one is the CLIQUE algorithm [AGGR05], that were unable to use it in the previous experiments, as we only had a 2-dimensional implementation. The second algorithm is the PG-means [FHE07], which is considered a more sophisticated GMM approach.

The results are illustrated in Figs 5.7 and 5.8, for S1 and S4, respectively. In the S1 case iPDDP retrieves only 4 of the clusters and due to the chaining effect fails to retrieve any more. In the S4 case it does not manage to produce sensible clustering, for the same reason. The high degree of cluster overlapping also hinders the KM-PDDP and GMM algorithms, both of them producing bad results in the case of S4 dataset. PG-means seems to overestimate the number of clusters, splitting most clusters in 2 or 3 parts. However, it can clearly identify the dense core of each cluster, but quite often this happens by more than two overlapping components which make the partitions quite confusing. Although this is a GMM based approach it shows much better results in the S4 case, substantially outperforming the standard GMM approach, but still there seem to exist one or two overlapping components. On the other hand, the high degree of cluster overlapping seems to help the k-means algorithm, resulting in a very good result for the S4 dataset, improving on the result of S1 dataset where one of the clusters is

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Dimension 2						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.87 (0.04)	0.88 (0.03)	0.83 (0.03)	0.86 (0.02)	0.76 (0.02)	0.84 (0.01)
iPDDP	0.90 (0.04)	0.89 (0.05)	0.89 (0.03)	0.80 (0.13)	0.91 (0.04)	0.58 (0.18)
dePDDP	0.95 (0.02)	0.94 (0.02)	0.91 (0.04)	0.92 (0.02)	0.84 (0.03)	0.88 (0.01)
KM-PDDP	0.8 9(0.03)	0.89 (0.04)	0.86 (0.04)	0.76 (0.23)	0.86 (0.04)	0.63 (0.19)
GMM	0.92 (0.02)	0.92 (0.02)	0.87 (0.03)	0.91 (0.02)	0.23 (0.10)	0.61 (0.13)
k-means	0.91 (0.03)	0.92 (0.02)	0.88 (0.02)	0.91 (0.02)	0.81 (0.04)	0.88 (0.01)
Dimension 5						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.94 (0.03)	0.94 (0.03)	0.91 (0.03)	0.93 (0.02)	0.88 (0.02)	0.91 (0.02)
iPDDP	1.00 (0.01)	1.00 (0.01)	0.98 (0.02)	0.98 (0.03)	0.98 (0.01)	0.50 (0.24)
dePDDP	1.00 (0.00)	0.99 (0.01)	1.00 (0.00)	0.99 (0.01)	1.00 (0.00)	0.99 (0.00)
KM-PDDP	0.94 (0.04)	0.96 (0.03)	0.93 (0.02)	0.96 (0.02)	0.90 (0.04)	0.68 (0.33)
GMM	0.95 (0.03)	0.96 (0.01)	0.93 (0.02)	0.96 (0.01)	0.91 (0.01)	0.96 (0.01)
k-means	0.94 (0.03)	0.95 (0.02)	0.93 (0.02)	0.95 (0.01)	0.92 (0.01)	0.96 (0.01)
Dimension 20						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.96 (0.02)	0.96 (0.03)	0.96 (0.02)	0.96 (0.02)	0.95 (0.01)	0.92 (0.02)
iPDDP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.98 (0.01)	0.80 (0.27)
dePDDP	1.00 (0.00)	0.99 (0.01)	1.00 (0.00)	0.99 (0.01)	1.00 (0.00)	0.99 (0.00)
KM-PDDP	0.92 (0.04)	0.95 (0.03)	0.92 (0.03)	0.96 (0.01)	0.87 (0.03)	0.92 (0.06)
GMM	0.94 (0.03)	0.96 (0.01)	0.92 (0.02)	0.95 (0.01)	0.91 (0.02)	0.96 (0.01)
k-means	0.94 (0.02)	0.95 (0.02)	0.93 (0.02)	0.95 (0.01)	0.93 (0.01)	0.95 (0.01)
Dimension 50						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.97 (0.02)	0.97 (0.02)	0.97 (0.01)	0.95 (0.03)	0.97 (0.01)	0.94 (0.02)
iPDDP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.01)	0.97 (0.07)
dePDDP	1.00 (0.00)	0.99 (0.01)	1.00 (0.00)	0.99 (0.00)	1.00 (0.00)	0.99 (0.00)
KM-PDDP	0.88 (0.03)	0.93 (0.02)	0.89 (0.03)	0.95 (0.02)	0.88 (0.02)	0.95 (0.02)
GMM	0.88 (0.03)	0.87 (0.03)	0.88 (0.03)	0.91 (0.02)	0.87 (0.02)	0.92 (0.01)
k-means	0.94 (0.02)	0.95 (0.02)	0.93 (0.02)	0.95 (0.01)	0.92 (0.01)	0.95 (0.01)

Table 5.4: *Mean purity and V-measure (with the observed standard deviation in parenthesis), for $DSET_{\text{Beta}}$ generated data of different algorithms, over 100 experiments.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

split and two others are merged. CLIQUE in both cases manages to retrieve the core of the clusters, but one or two clusters are totally missed. This probably happens because CLIQUE is very sensitive to user intervention. It is very difficult to choose parameters that retrieve all the clusters without merging any of them (especially in the S4 case). Finally, dePDDP produces very good results in both S1 and S4. In the S4 case, dePDDP retrieves 2 or 3 more clusters than the 15 actual ones, but this is due to the large amount of noise points.

5.6.1 Datasets with Noise

In real life datasets we are expecting to find cases where the data are contaminated with noise. To examine the performance of the different algorithms in such cases we are going to construct appropriate simulation settings. For this reason, we first employ the $DSET_{\text{Gausian}}$ data generation mechanism and in each dataset we include a number of uniform randomly drawn points in the data space to represent noise. First, we are going to examine the case of 1000 noise points.

The results are shown in Table 5.5 with respect to cluster purity and V-measure, respectively. The reported results are again the mean values over 100 experiments, with the observed standard deviation shown in parenthesis. In this case, in low dimensions all the algorithms perform poorly. It seems that the noise contamination is so high that it makes it impossible for the algorithms to retrieve the cluster structure (especially the iPDDP and the KM-PDDP algorithms). Even in this case, the dePDDP algorithm produces the best results. However, as the dimensionality of the data increases, the clustering structure becomes more prominent and the algorithms manage to successfully retrieve it. The dePDDP algorithm stands out in this case, as it results in very pure clusterings and is the least affected by noise, irrespective of the number of clusters in the data. The iPDDP algorithm, as expected, manages to provide pure partitions when either the dimensionality is very large or the number noise points is small with respect to the total number of points in the data (cases with large number of clusters). However, in those cases PDDP and KM-PDDP perform similarly well.

To better illustrate the effect of noise contamination, in Figure 5.9 we plot the purity and the V-measure of different algorithms for an increasing percentage of noisy points in the data set. These plots refer to a particular case of 5 dimensions and 15 classes. The x-axis of this plot corresponds to the percentage of the noisy points in the dataset. For each algorithm the mean obtained purity over 100 experiments is plotted. The vertical lines designate the standard deviation in each experiment. It is evident that the performance of both iPDDP and KM-PDDP heavily deteriorates with the increase of noise contamination, while the rest algorithms are only gradually affected. dePDDP is the least affected algorithm and only when more than

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

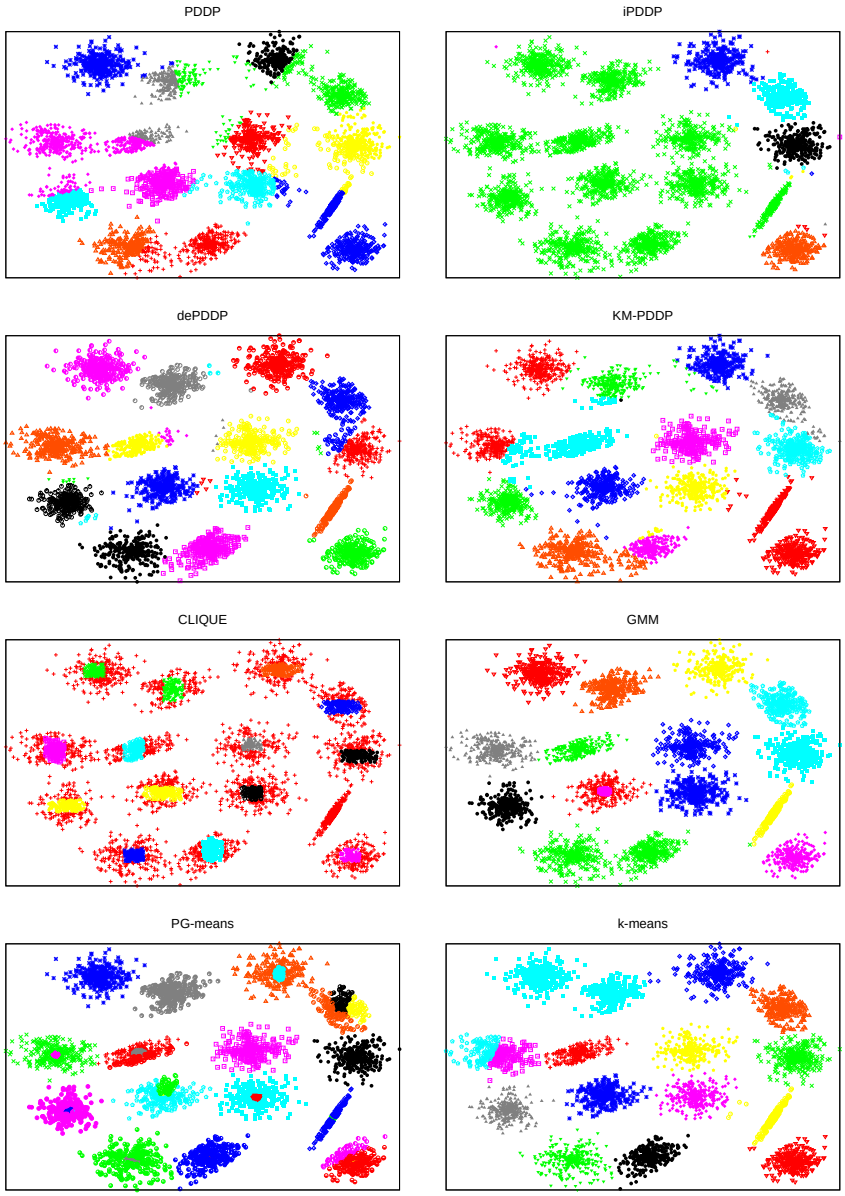


Figure 5.7: *Visual comparison of different algorithms on the S1 example dataset. Different colours and point types correspond to different clusters.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

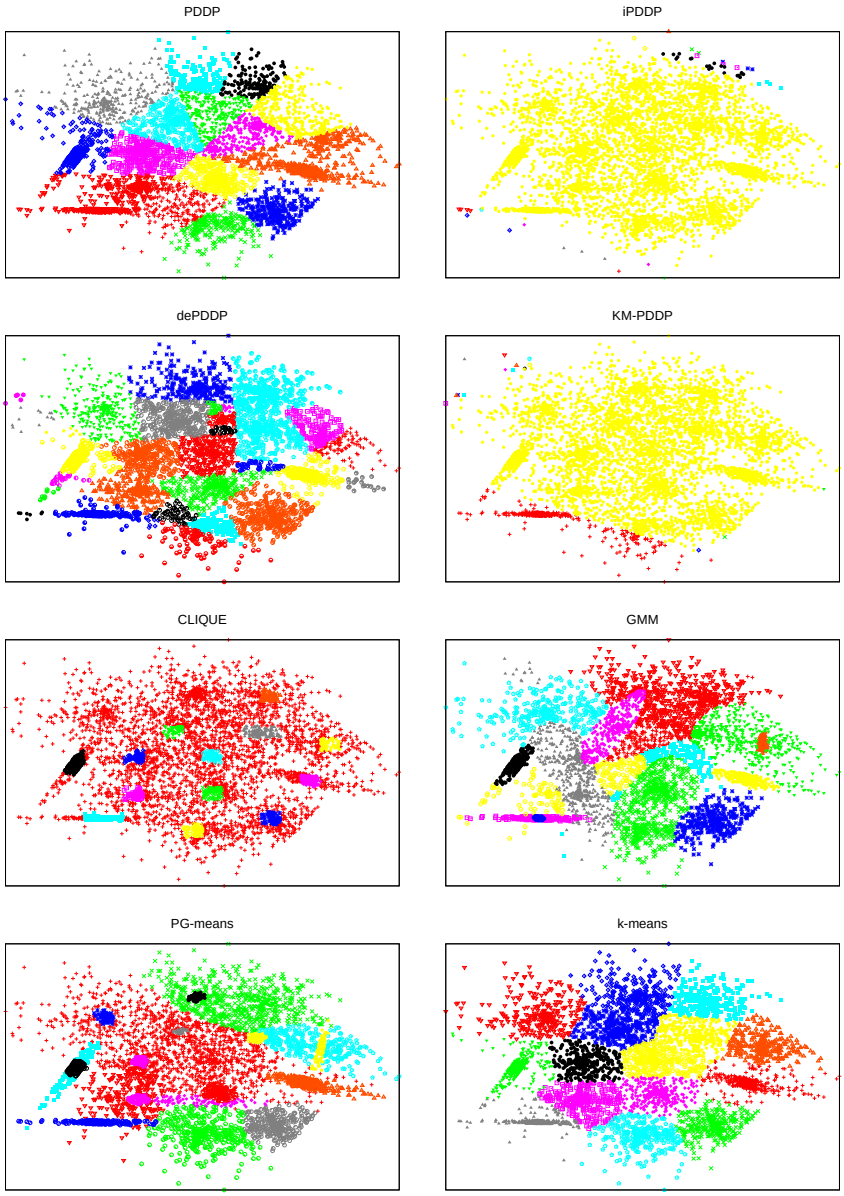


Figure 5.8: *Visual comparison of different algorithms on the S4 example dataset. Different colours and point types correspond to different clusters.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Dimension 2						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.70 (0.05)	0.80 (0.04)	0.66 (0.05)	0.79 (0.03)	0.63 (0.04)	0.78 (0.02)
iPDDP	0.11 (0.14)	0.01 (0.04)	0.21 (0.23)	0.01 (0.03)	0.67 (0.29)	0.05 (0.04)
dePDDP	0.87 (0.08)	0.89 (0.12)	0.88 (0.06)	0.92 (0.03)	0.84 (0.03)	0.88 (0.02)
KM-PDDP	0.29 (0.17)	0.32 (0.23)	0.38 (0.22)	0.44 (0.24)	0.71 (0.25)	0.38 (0.21)
GMM	0.71 (0.04)	0.84 (0.02)	0.70 (0.04)	0.84 (0.02)	0.64 (0.03)	0.83 (0.02)
k-means	0.79 (0.05)	0.90 (0.03)	0.79 (0.05)	0.91 (0.02)	0.80 (0.02)	0.90 (0.01)
Dimension 5						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.78 (0.06)	0.84 (0.05)	0.77 (0.06)	0.85 (0.04)	0.78 (0.04)	0.87 (0.01)
iPDDP	0.22 (0.23)	0.05 (0.08)	0.56 (0.27)	0.15 (0.09)	0.85 (0.11)	0.16 (0.09)
dePDDP	0.98 (0.02)	0.99 (0.01)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.00)
KM-PDDP	0.65 (0.17)	0.65 (0.28)	0.62 (0.22)	0.48 (0.32)	0.87 (0.09)	0.25 (0.25)
GMM	0.74 (0.05)	0.88 (0.01)	0.75 (0.04)	0.90 (0.01)	0.75 (0.03)	0.92 (0.01)
k-means	0.87 (0.04)	0.94 (0.01)	0.90 (0.03)	0.96 (0.01)	0.88 (0.02)	0.96 (0.01)
Dimension 20						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.92 (0.03)	0.88 (0.11)	0.92 (0.03)	0.79 (0.10)	0.92 (0.04)	0.68 (0.07)
iPDDP	0.80 (0.04)	0.47 (0.12)	0.87 (0.04)	0.37 (0.10)	0.95 (0.02)	0.29 (0.13)
dePDDP	0.95 (0.20)	0.95 (0.22)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
KM-PDDP	0.87 (0.04)	0.93 (0.03)	0.84 (0.04)	0.91 (0.03)	0.87 (0.06)	0.55 (0.32)
GMM	0.77 (0.06)	0.90 (0.02)	0.75 (0.03)	0.91 (0.01)	0.80 (0.03)	0.93 (0.01)
k-means	0.86 (0.04)	0.93 (0.02)	0.88 (0.03)	0.95 (0.01)	0.88 (0.02)	0.96 (0.00)
Dimension 50						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.89 (0.03)	0.82 (0.06)	0.93 (0.01)	0.74 (0.07)	0.92 (0.04)	0.65 (0.07)
iPDDP	0.93 (0.03)	0.88 (0.11)	0.95 (0.01)	0.82 (0.10)	0.97 (0.01)	0.57 (0.21)
dePDDP	1.00 (0.00)	1.00 (0.01)	1.00 (0.00)	1.00 (0.01)	1.00 (0.00)	1.00 (0.00)
KM-PDDP	0.89 (0.05)	0.95 (0.02)	0.86 (0.05)	0.94 (0.02)	0.87 (0.03)	0.86 (0.11)
GMM	0.64 (0.08)	0.82 (0.04)	0.70 (0.06)	0.87 (0.02)	0.77 (0.03)	0.91 (0.01)
k-means	0.87 (0.04)	0.93 (0.02)	0.86 (0.03)	0.94 (0.01)	0.88 (0.02)	0.95 (0.01)

Table 5.5: *Mean purity and V-measure (with the observed standard deviation in parenthesis), for $DSET_{\text{Gausian}}$ generated data contaminated with 1000 noise points of different algorithms, over 100 experiments.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

50% of the points in the data correspond to noise.

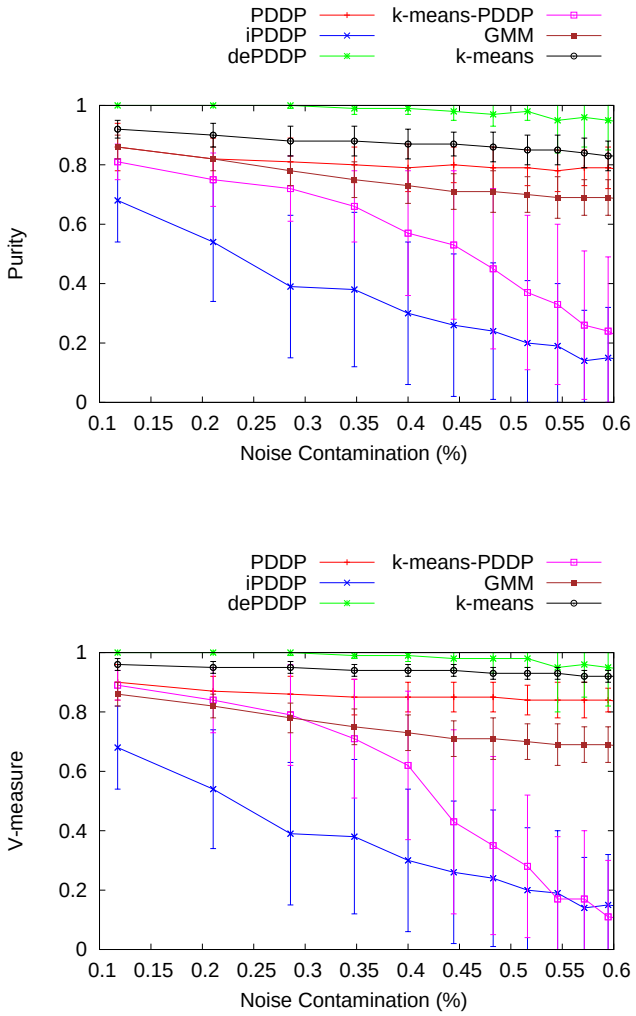


Figure 5.9: *The performance of the algorithms, when the noise contamination increases with respect to cluster purity (top) and the V-measure (bottom).*

5.6.2 Automatic cluster number determination

As described above, the iPDDP and dePDDP algorithms can automatically determine the number of clusters in a data set. To measure the efficiency of the algorithms, we resort to a similar experimental procedure to the one previously described. 100 datasets are artificially constructed using the $DSET_{\text{Gausian}}$ data generation mechanism.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

In this case, the use of PDDP and KM-PDDP algorithms is not possible, since these algorithms do not have an automated stopping criterion. On the other hand, we additionally include two very popular algorithms: the DBSCAN algorithm [SEKX98] as a representative of the density-based class and the x-means algorithm [PM00] as a representative of the partitioning class. Regarding the GMM algorithm, we executed the algorithm for a varying number of clusters (up to a maximum k_{max}) and then selected the best amongst them using three model selection criteria, namely BIC, AIC and ICL [BCG00]. Moreover, we also tested the PG-means algorithm [FHE07], as a more sophisticated GMM approach.

Similarly to the previous experimental settings, a number of different dimensions and number of clusters were tested. The $MinPts$ parameter for iPDDP was set to 5 and k_{max} was set twice the actual number of clusters across all experiments. The bandwidth parameter of dePDDP was set to the optimal one given by Eq. 5.8. For DBSCAN algorithm, the $MinPts$ was set to 5 and Eps to $5a$, where a is the data dimensionality.

The results from this experimental setting are exhibited in Table 5.6, with respect to cluster purity and V-measure, respectively. Furthermore, Table 5.7 reports the number of retrieved clusters for each algorithm.

The iPDDP algorithm in this case seems very efficient, especially as the dimensionality grows and the clusters are more clearly separated. It produces very good results with respect to both the purity and the V-measure, and gives almost perfect estimations for the number of clusters in high dimensional cases. Similarly, dePDDP performs very well in all dimensions examined and always produces accurate estimations for the number of clusters. DBSCAN manages to correctly retrieve the number of clusters, with very pure partitions. However, note that DBSCAN identifies many points as outliers including them in one big cluster and that this cluster is removed for the calculation of the purity and the V-measure (i.e. not shown in the tables). GMM-BIC and GMM-ICL perform similarly, producing good partitions in low dimensions, but their performance degrades as the dimensionality increases. The same holds for the number of retrieved clusters. As dimensionality increases the number of clusters is underestimated. GMM-AIC performs better with respect to both cluster purity and V-measure. In low dimensions it retrieves almost accurate cluster numbers, but in high dimensions it greatly overestimates the number of clusters. Out of all the mixture modelling approaches PG-means managed to produce the best results, very close to the ones achieved by dePDDP and iPDDP. The problem with this approach is that it cannot handle the 50 dimensional cases, resulting in the majority of runs in trivial partitions of one cluster. For this reason we have excluded it from the particular part of Tables 5.6 and 5.7. Finally, the x-means algorithm exhibits very poor performance across all experiments.

Next, we repeat the same experimental procedure, but with noise con-

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Dimension 2						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
iPDDP	0.88 (0.05)	0.92 (0.04)	0.84 (0.04)	0.86 (0.10)	0.75 (0.05)	0.65 (0.15)
dePDDP	0.94 (0.04)	0.95 (0.02)	0.92 (0.03)	0.93 (0.02)	0.84 (0.04)	0.89 (0.02)
GMM-BIC	0.88 (0.04)	0.93 (0.02)	0.72 (0.13)	0.87 (0.06)	0.18 (0.09)	0.58 (0.15)
GMM-AIC	0.89 (0.07)	0.93 (0.03)	0.68 (0.22)	0.85 (0.10)	0.24 (0.10)	0.63 (0.13)
GMM-ICL	0.85 (0.10)	0.92 (0.04)	0.72 (0.15)	0.87 (0.07)	0.25 (0.13)	0.64 (0.12)
PG-means	0.94 (0.02)	0.95 (0.01)	0.92 (0.02)	0.93 (0.02)	0.82 (0.03)	0.90 (0.01)
x-means	0.20 (0.10)	0.44 (0.09)	0.10 (0.04)	0.35 (0.06)	0.04 (0.01)	0.29 (0.03)
DBSCAN	0.80 (0.07)	0.88 (0.05)	0.73 (0.06)	0.84 (0.04)	0.56 (0.09)	0.63 (0.07)
Dimension 5						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
iPDDP	1.00 (0.01)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.95 (0.05)	0.79 (0.27)
dePDDP	1.00 (0.00)	0.99 (0.01)	1.00 (0.00)	0.99 (0.00)	1.00 (0.00)	0.99 (0.00)
GMM-BIC	0.99 (0.01)	0.98 (0.01)	0.98 (0.01)	0.98 (0.00)	0.94 (0.06)	0.97 (0.01)
GMM-AIC	0.97 (0.03)	0.95 (0.01)	0.97 (0.01)	0.96 (0.01)	0.89 (0.21)	0.92 (0.18)
GMM-ICL	0.98 (0.02)	0.98 (0.01)	0.96 (0.07)	0.98 (0.02)	0.95 (0.02)	0.97 (0.00)
PG-means	1.00 (0.00)	1.00 (0.01)	1.00 (0.01)	1.00 (0.01)	0.95 (0.02)	0.98 (0.01)
x-means	0.28 (0.15)	0.44 (0.09)	0.10 (0.04)	0.36 (0.05)	0.04 (0.00)	0.29 (0.01)
DBSCAN	1.00 (0.00)	1.00 (0.00)	1.00 (0.01)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Dimension 20						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
iPDDP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
dePDDP	1.00 (0.00)	1.00 (0.01)	1.00 (0.00)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)
GMM-BIC	0.80 (0.07)	0.92 (0.02)	0.74 (0.09)	0.92 (0.02)	0.73 (0.00)	0.93 (0.00)
GMM-AIC	0.98 (0.01)	0.92 (0.01)	0.98 (0.01)	0.93 (0.01)	0.96 (0.00)	0.96 (0.00)
GMM-ICL	0.83 (0.07)	0.93 (0.02)	0.77 (0.09)	0.93 (0.02)	0.66 (0.00)	0.92 (0.00)
PG-means	0.99 (0.02)	0.99 (0.01)	0.97 (0.02)	0.99 (0.01)	0.98 (0.01)	0.99 (0.00)
x-means	0.32 (0.07)	0.61 (0.09)	0.16 (0.05)	0.44 (0.09)	0.05 (0.02)	0.32 (0.05)
DBSCAN	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Dimension 50						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
iPDDP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
dePDDP	1.00 (0.00)	1.00 (0.01)	1.00 (0.00)	0.99 (0.00)	1.00 (0.00)	1.00 (0.00)
GMM-BIC	0.31 (0.06)	0.69 (0.03)	0.28 (0.03)	0.74 (0.02)	0.27 (0.02)	0.78 (0.01)
GMM-AIC	0.98 (0.02)	0.88 (0.02)	0.98 (0.01)	0.90 (0.01)	0.96 (0.01)	0.91 (0.01)
GMM-ICL	0.31 (0.04)	0.70 (0.04)	0.29 (0.03)	0.74 (0.02)	0.30 (0.03)	0.80 (0.01)
x-means	0.34 (0.09)	0.64 (0.04)	0.20 (0.07)	0.57 (0.03)	0.09 (0.03)	0.46 (0.08)
DBSCAN	1.00 (0.00)	0.98 (0.01)	1.00 (0.00)	0.98 (0.01)	1.00 (0.00)	0.99 (0.00)

Table 5.6: *Mean purity and V-measure (with the observed standard deviation in parenthesis), for $DSET_{\text{Gausian}}$ generated data of different algorithms, with automated cluster determination over 100 experiments.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Dimension 2			
	15 Clusters	25 Clusters	50 Clusters
iPDDP	12.25(1.48)	16.55(2.91)	18.10 (3.63)
dePDDP	15.10(1.41)	25.45(2.84)	46.70 (5.26)
GMM-BIC	13.15(1.39)	15.65(3.80)	7.50 (3.64)
GMM-AIC	14.30(2.61)	15.40(5.53)	9.00 (3.49)
GMM-ICL	12.30(2.39)	15.95(4.42)	9.35 (4.36)
PG-means	14.00(1.10)	26.20(2.48)	37.80(4.07)
x-means	2.43(0.72)	2.13(0.34)	2.03(0.18)
DBSCAN	9.93(1.41)	13.03(1.72)	10.90(2.20)
Dimension 5			
	15 Clusters	25 Clusters	50 Clusters
iPDDP	14.95 (0.22)	25.05 (0.38)	34.75 (16.15)
dePDDP	15.80 (1.08)	26.65 (1.01)	56.44 (2.96)
GMM-BIC	18.05 (1.56)	30.55 (2.31)	61.25 (10.19)
GMM-AIC	24.40 (4.13)	45.30 (4.12)	73.44 (25.53)
GMM-ICL	17.95 (2.01)	30.50 (4.57)	68.00 (6.50)
PG-means	15.60 (0.80)	25.20 (0.40)	45.00 (1.79)
x-means	2.67 (0.54)	2.17 (0.45)	2.00 (0.00)
DBSCAN	15.00 (0.00)	24.83 (0.37)	49.63 (0.48)
Dimension 20			
	15 Clusters	25 Clusters	50 Clusters
iPDDP	15.05 (0.22)	25.00 (0.00)	50.00 (0.00)
dePDDP	15.65 (0.73)	26.80 (1.50)	56.00 (0.00)
GMM-BIC	10.90 (1.14)	16.70 (2.17)	33.00 (0.00)
GMM-AIC	28.95 (1.86)	48.15 (5.12)	63.00 (0.00)
GMM-ICL	11.35 (1.28)	17.65 (2.35)	31.00 (0.00)
PG-means	14.80 (0.40)	23.80 (0.75)	48.20 (0.75)
x-means	3.77 (0.42)	2.97 (0.66)	2.27 (0.44)
DBSCAN	15.30 (0.59)	25.30 (0.46)	50.57 (0.76)
Dimension 50			
	15 Clusters	25 Clusters	50 Clusters
iPDDP	15.15 (0.36)	25.05 (0.22)	50.00 (0.00)
dePDDP	15.70 (0.95)	26.60 (1.16)	54.50 (0.50)
GMM-BIC	4.30 (0.56)	6.70 (0.64)	12.50 (0.50)
GMM-AIC	30.75 (0.54)	50.60 (0.66)	101.00 (0.00)
GMM-ICL	4.50 (0.59)	6.80 (0.60)	14.00 (1.00)
x-means	4.00 (0.00)	4.00 (0.00)	3.60 (0.55)
DBSCAN	16.23 (1.99)	28.33 (3.67)	53.40 (4.84)

Table 5.7: *Mean number of retrieved clusters (with the observed standard deviation in parenthesis), for $DSET_{\text{Gausian}}$ generated data of different algorithms, over 100 experiments.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

taminated data as before, by including 1000 noise points. The results are shown in Table 5.8 with respect to cluster purity and V-measure. Similarly, Table 5.9 reports the number of retrieved clusters for each algorithm. In this case, the results of the iPDDP algorithm are heavily affected by the inclusion of noise, except for the high dimensional cases. In low dimensions, the number of retrieved clusters is very small due the chaining effect. As expected, dePDDP is not affected by the inclusion of noise and again produces high quality results with respect to the cluster purity and V-measure, while simultaneously producing accurate estimations for the number of clusters. The same is true for the DBSCAN algorithm. However, the DBSCAN algorithm is also influenced by the chaining effect in the 2-dimensional case. The GMM-BIC and GMM-ICL are also affected by the noise points and they produce bad results in all cases (especially in high dimensions). On the other hand, GMM-AIC can retrieve good results only in high dimensions. It seems that the inclusion of noise improved the performance of GMM-AIC in high dimensions with respect to the number of retrieved clusters. The PG-means algorithm in this case finds it more difficult to approximate the number of clusters, underestimating their number significantly. This translates to lower values for both cluster purity and V-measure but it still outperforms all the GMM alternatives. The results of the x-means algorithm remain pretty bad.

§ 5.7 Selection of the k_{max} and $MinPts$ of the iPDDP algorithm

As we have seen, the iPDDP algorithm can be used to automatically determine the number of clusters in the dataset. To achieve this we need to force the algorithm to find more than the actual clusters, by selecting an appropriate value for the k_{max} parameter. However, in real life problems, the number of the actual clusters is not know and in effect the selection of the k_{max} parameter is not straightforward.

To examine the effect of this parameter on the performance of the proposed iPDDP algorithm, we employ datasets constructed as in Section 5.6, using $DSET_{\text{Gausian}}$ with and without noise. In both cases, the datasets contain 25 clusters in 5 dimensions, while the noisy datasets contain additionally 1000 uniformly distributed random points to represent noise.

The results with respect to the number of retrieved clusters, as well as the purity and V-measure are presented in Figure 5.10 (top) and (bottom), respectively. In the noiseless case, the algorithm consistently results in high values for both the purity and the V-measure. However, when k_{max} obtains very high values (higher than 300) the number of clusters are gradually underestimated. This occurs because when the value of k_{max} is too high the

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Dimension 2						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
iPDDP	0.19 (0.20)	0.05 (0.09)	0.28 (0.23)	0.12 (0.10)	0.55 (0.21)	0.19 (0.11)
dePDDP	0.89 (0.07)	0.92 (0.05)	0.90 (0.05)	0.91 (0.05)	0.84 (0.03)	0.89 (0.02)
GMM-BIC	0.42 (0.10)	0.66 (0.11)	0.32 (0.12)	0.60 (0.13)	0.15 (0.06)	0.50 (0.13)
GMM-AIC	0.41 (0.15)	0.64 (0.13)	0.27 (0.08)	0.55 (0.10)	0.13 (0.05)	0.45 (0.15)
GMM-ICL	0.34 (0.13)	0.59 (0.16)	0.26 (0.11)	0.51 (0.17)	0.12 (0.06)	0.41 (0.13)
PG-means	0.83 (0.07)	0.91 (0.03)	0.82 (0.04)	0.90 (0.01)	0.71 (0.05)	0.87 (0.02)
x-means	0.20 (0.10)	0.44 (0.09)	0.10 (0.04)	0.35 (0.06)	0.04 (0.01)	0.29 (0.03)
DBSCAN	0.80 (0.07)	0.88 (0.05)	0.73 (0.06)	0.84 (0.04)	0.56 (0.09)	0.63 (0.07)
Dimension 5						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
iPDDP	0.54 (0.15)	0.21 (0.10)	0.67 (0.11)	0.26 (0.10)	0.70 (0.03)	0.16 (0.14)
dePDDP	0.99 (0.02)	0.99 (0.01)	1.00 (0.00)	1.00 (0.00)	1.00 (0.01)	0.99 (0.00)
GMM-BIC	0.63 (0.10)	0.85 (0.03)	0.55 (0.09)	0.84 (0.04)	0.34 (0.20)	0.63 (0.29)
GMM-AIC	0.67 (0.06)	0.85 (0.03)	0.58 (0.07)	0.85 (0.02)	0.54 (0.05)	0.87 (0.01)
GMM-ICL	0.66 (0.11)	0.85 (0.04)	0.54 (0.07)	0.84 (0.03)	0.42 (0.04)	0.83 (0.01)
PG-means	0.76 (0.20)	0.89 (0.09)	0.78 (0.12)	0.91 (0.04)	0.77 (0.12)	0.92 (0.05)
x-means	0.28 (0.15)	0.44 (0.09)	0.10 (0.04)	0.36 (0.05)	0.04 (0.00)	0.29 (0.01)
DBSCAN	1.00 (0.00)	1.00 (0.00)	1.00 (0.01)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Dimension 20						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
iPDDP	0.87 (0.04)	0.70 (0.10)	0.89 (0.04)	0.58 (0.14)	0.89 (0.05)	0.27 (0.15)
dePDDP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
GMM-BIC	0.63 (0.12)	0.85 (0.04)	0.63 (0.08)	0.88 (0.02)	0.57 (0.16)	0.89 (0.03)
GMM-AIC	0.82 (0.12)	0.92 (0.04)	0.84 (0.12)	0.94 (0.04)	0.95 (0.02)	0.97 (0.00)
GMM-ICL	0.64 (0.08)	0.86 (0.03)	0.65 (0.07)	0.88 (0.02)	0.67 (0.05)	0.90 (0.01)
PG-means	0.85 (0.16)	0.73 (0.29)	0.93 (0.02)	0.83 (0.10)	0.90 (0.03)	0.76 (0.04)
x-means	0.32 (0.07)	0.61 (0.09)	0.16 (0.05)	0.44 (0.09)	0.05 (0.02)	0.32 (0.05)
DBSCAN	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
Dimension 50						
	15 Clusters		25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.	Pur.	V-meas.
iPDDP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.76 (0.24)	0.98 (0.02)	0.76 (0.24)
dePDDP	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.00)	1.00 (0.00)	0.99 (0.00)
GMM-BIC	0.26 (0.06)	0.62 (0.09)	0.26 (0.06)	0.74 (0.03)	0.24 (0.04)	0.74 (0.03)
GMM-AIC	0.74 (0.14)	0.86 (0.06)	0.74 (0.14)	0.87 (0.09)	0.69 (0.25)	0.87 (0.09)
GMM-ICL	0.29 (0.06)	0.65 (0.05)	0.29 (0.06)	0.77 (0.01)	0.31 (0.01)	0.77 (0.01)
x-means	0.34 (0.09)	0.64 (0.04)	0.20 (0.07)	0.57 (0.03)	0.09 (0.03)	0.46 (0.08)
DBSCAN	1.00 (0.00)	0.98 (0.01)	1.00 (0.00)	0.98 (0.01)	1.00 (0.00)	0.99 (0.00)

Table 5.8: *Mean purity and V-measure (with the observed standard deviation in parenthesis), for $DSET_{\text{Gausian}}$ generated data contaminated with 1000 noise points of different algorithms with automated number of clusters determination, over 100 experiments.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Dimension 2			
	15 Clusters	25 Clusters	50 Clusters
iPDDP	1.30 (0.46)	1.90 (0.77)	4.12 (1.45)
dePDDP	13.25 (2.84)	23.90 (3.74)	45.50 (5.41)
GMM-BIC	5.95 (1.88)	6.45 (2.73)	5.38 (1.93)
GMM-AIC	5.95 (2.62)	5.20 (1.81)	4.62 (1.87)
GMM-ICL	5.00 (2.37)	4.95 (2.71)	4.00 (1.87)
PG-means	14.60 (2.87)	20.80 (1.33)	32.40 (2.58)
x-means	2.00 (0.00)	2.00 (0.00)	2.00 (0.00)
DBSCAN	35.20 (3.25)	28.60 (3.76)	12.05 (4.15)
Dimension 5			
	15 Clusters	25 Clusters	50 Clusters
iPDDP	2.30 (0.56)	3.65 (1.01)	4.33 (1.89)
dePDDP	14.90 (0.89)	25.85 (0.96)	57.00 (0.00)
GMM-BIC	8.90 (1.55)	12.95 (2.18)	15.00 (9.63)
GMM-AIC	9.10 (0.94)	13.75 (1.73)	25.00 (1.41)
GMM-ICL	9.00 (1.38)	12.60 (1.50)	20.00 (2.16)
PG-means	11.00 (2.90)	17.60 (3.44)	33.60 (7.81)
x-means	2.00 (0.00)	2.00 (0.00)	2.00 (0.00)
DBSCAN	15.00 (0.00)	24.95 (0.22)	49.50 (0.67)
Dimension 20			
	15 Clusters	25 Clusters	50 Clusters
iPDDP	7.50 (1.53)	9.55 (2.73)	9.00 (4.00)
dePDDP	15.20 (0.40)	26.10 (1.30)	55.50 (1.50)
GMM-BIC	8.10 (1.45)	13.70 (1.76)	25.50 (6.50)
GMM-AIC	11.70 (2.12)	21.75 (5.31)	63.50 (0.50)
GMM-ICL	8.30 (0.90)	14.00 (1.61)	28.50 (1.50)
PG-means	9.00 (3.74)	16.20 (3.54)	23.60 (2.42)
x-means	2.65 (0.57)	2.35 (0.48)	2.00 (0.00)
DBSCAN	15.20 (0.51)	25.20 (0.40)	50.45 (0.74)
Dimension 50			
	15 Clusters	25 Clusters	50 Clusters
iPDDP	15.00 (0.00)	24.85 (0.91)	34.00 (5.00)
dePDDP	15.60 (0.80)	26.65 (1.31)	54.00 (1.00)
GMM-BIC	3.65 (0.91)	5.75 (1.04)	10.50 (1.50)
GMM-AIC	10.00 (2.37)	21.60 (1.91)	35.50 (1.50)
GMM-ICL	3.90 (0.54)	6.20 (1.29)	12.50 (0.50)
x-means	3.90 (0.30)	3.50 (0.59)	2.95 (0.59)
DBSCAN	17.15 (2.67)	27.85 (3.48)	53.85 (4.25)

Table 5.9: *Mean number of retrieved clusters (with the observed standard deviation in parenthesis), for $DSET_{\text{Gausian}}$ generated data contaminated with 1000 noise points of different algorithms, over 100 experiments.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

algorithm splits even the actual clusters in very small sets, incorrectly identifying them as outliers. So, in general, as along as we do not set extremely high values to k_{max}, the algorithm is expected to result in good partitions.

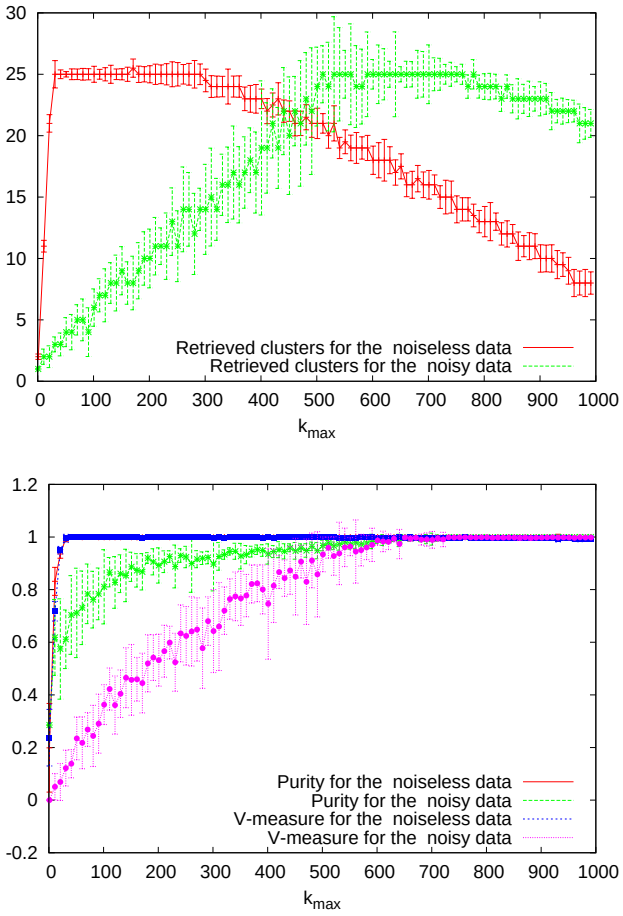


Figure 5.10: *The performance of the iPDDP algorithm with respect to the number of retrieved clusters (top) and the purity and V-measure values (bottom) for a range of values for the k_{max} parameter.*

The presence of noise in the dataset makes the problem much more difficult. This is expected as the iPDDP algorithm has failed to result in good partitions in noisy cases (see results in Section 5.6.2). The algorithm manages to achieve high values for the purity and V-measure only when k_{max} is set high enough (over 500). However, setting k_{max} to a very large value is not a solution. As shown, there exists a range of k_{max} values (i.e. 500 to 800) where the algorithm is able to retrieve the correct number of clusters. That interval is connected to the number of noisy points in the

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

data, which of course in real life problems is not known beforehand. As such, it is obvious that in a noisy dataset the determination of appropriate k_{max} values can be problematic and, as shown in Section 5.6.2, the algorithm is expected to exhibit inferior performance.

To examine the effect of the $MinPts$ parameter on the performance of the iPDDP algorithm we employ the same noiseless dataset used above. Now the k_{max} parameter is set to two times the actual number of clusters. In Figure 5.11 (top) and (bottom), we present the results with respect to the number of retrieved clusters, and the purity and V-measure, respectively, for several values of the $MinPts$ parameter. As shown, the performance of the algorithm is not affected, as long as, the $MinPts$ parameter has value smaller than the size of the clusters. Thus, in the case that the user cannot make an informative selection of the value of $MinPts$, setting it to a small number is a sensible choice. The results for the noisy dataset are similar and are excluded for brevity.

§ 5.8 Bandwidth Selection for the dePDDP algorithm

As already discussed, the dePDDP algorithm relies on the density of the projected points to guide the clustering procedure. This density is approximated though a non-parametric kernel density estimation mechanism, a standard technique in explorative data analysis, for which there is still a big dispute on how to assess the quality of the estimate and which choice of bandwidth is optimal [Tur93].

The bandwidth of the kernel is a free parameter which exhibits a strong influence on the resulting estimate. The way that the bandwidth selection influences the estimation can be explained by examining the Asymptotic Mean Squared Error (AMISE). Small values of h increase the (asymptotic) variance and thus the resulting estimate $\hat{f}(x;h)$ seems "wiggly" with many spurious features if graphically checked. On the other hand, big values of h reduce the (asymptotic) variance of $\hat{f}(x;h)$, but also increase the (asymptotic) bias, probably "smoothing away" the features of the true density f. An example of this behaviour is illustrated in Figure 5.12.

The most common optimality criterion used to select the bandwidth parameter is the mean integrated squared error

$$MISE(h) = E \int (\hat{f}(x;h) - f(x))^2 dx. \tag{5.4}$$

Under weak assumptions on f and K, $MISE(h) = AMISE(h) + o(1/nh + h^4)$, where o is the little o notation. The AMISE is the Asymptotic MISE which consists of the two leading terms

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

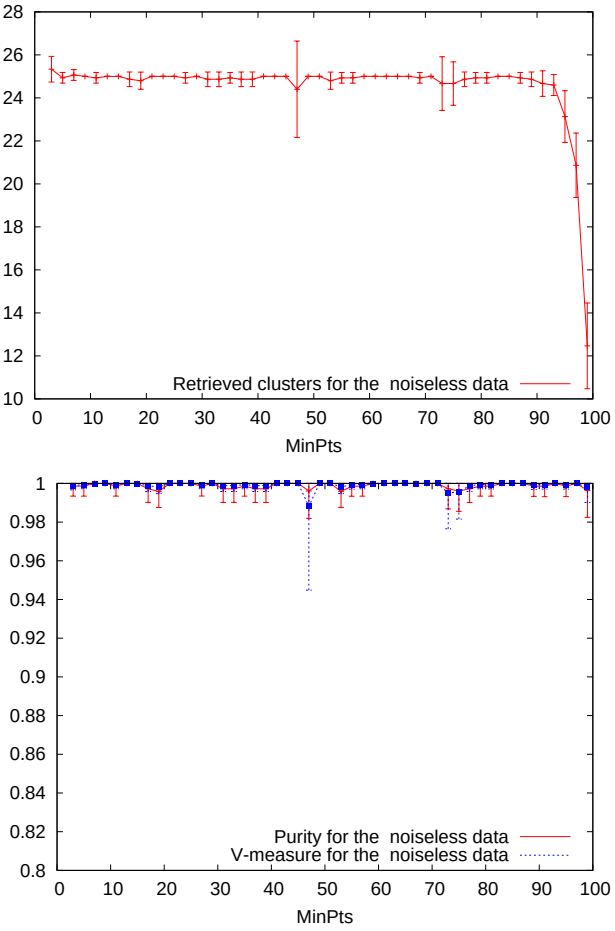


Figure 5.11: *The performance of the iPDDP algorithm with respect to the number of retrieved clusters (top) and the purity and V-measure values (bottom) for a range of values for the MinPts parameter.*

$$AMISE(h) = \frac{R(K)}{nh} + \frac{1}{4} m_2(K)^2 h^4 R(f''), \tag{5.5}$$

where $R(g) = \int g(x)^2 dx$ for a function g, $m_2(K) = \int x^2 K(x)$ and f'' is the second derivative of f. The minimum of this $AMISE$ is the solution to this differential equation

$$\frac{\partial}{\partial h} AMISE(h) = \frac{R(K)}{nh^2} + m_2(K)^2 h^3 R(f'') = 0 \tag{5.6}$$

or

$$h_A MISE = \frac{R(K)^{1/5}}{m_2(K)^{2/5} R(f'')^{1/5} n1/5}. \tag{5.7}$$

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

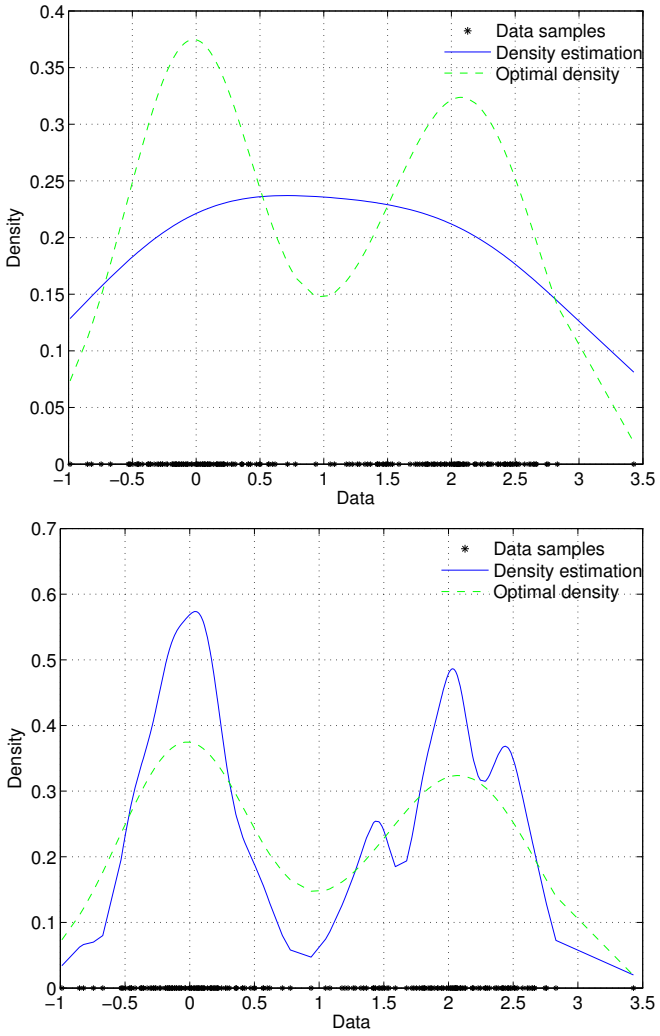


Figure 5.12: *An example of the KDE for a big bandwidth value (top) and for a small bandwidth value (bottom) respectivelly.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Neither the $AMISE$ nor the h_{AMISE} formulas can be used directly, since they involve the unknown density function f, so numerous methods have been proposed to automatically select the bandwidth. For more details refer to [Tur93], where various techniques to automatically tune the bandwidth are discussed and their particular characteristics are explained. A cross-validation selection method was proposed in [Sar93], while [AL95] proposed a plug-in estimate which minimises an estimate of the mean weighted integrated squared error, using the density function as a weight function. In [Han04] it was proposed to choose the bandwidth depending on the roughness of the first derivative of the density, which is straightforward to estimate. Here, we just follow the "normal reference rule", which suggests to use a bandwidth h_{opt} that minimises the Mean Integrated Squared Error (MISE). This is given by:

$$h_{opt} = \sigma \left(\frac{4}{3n} \right)^{1/5}, \tag{5.8}$$

where σ is standard deviation of the data.

Throughout the experiments in the paper we have set the bandwidth parameter to the value h_{opt} given by Eq. 5.8. In this section, we experimentally examine how different bandwidth selection schemes would affect the performance of dePDDP algorithm.

First we would like to investigate how an "improper" bandwidth would affect the performance of the algorithm. As it is not trivial to specify what an improper bandwidth is, since even that is data dependent, we are going to use a wide range of multiples of the h_{opt} value and examine the clustering performance of each one.

For this reason, we employ the $DSET_{\text{Gausian}}$ data generation mechanism and for various values of the bandwidth multiplier we examine obtained purity and V-measures. Further, to examine the effect with respect to different dimensions and number of clusters, we use datasets of 2 and 15 dimensions, with 15 and 25 clusters. The results are illustrated in Figure 5.13. It is evident that there exists a wide range of multiplier values between 1.5 and 3 that the dePDDP algorithm exhibits very good results with respect both to purity and V-measure. However, for small values of the multiplier, the results show high purity, the V-measure is much smaller as it quite possible that many actual clusters are split. For large values of the multiplier, both the purity and V-measure obtain small values, indicative of bad partitions. Note also that the results are similar for all of the combinations of dimensions and clusters examined, shows that the bandwidth selection is independent of the data dimensionality or the number of clusters in the data. In conclusion, we can see that any value of the multiplier less than 2 gives good results. Large bandwidths should be avoided, since the algorithm is expected to perform poorly. On the other hand, small bandwidths are not

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

such a big concern, since the algorithm is expected to return an increased number of (pure) clusters.

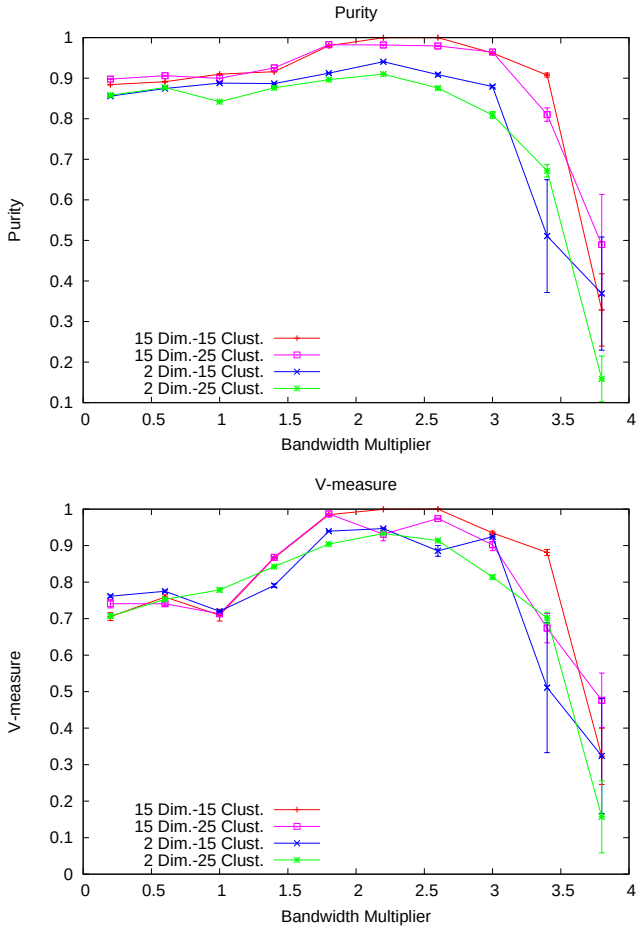


Figure 5.13: *The performance of the dePDDP algorithm with respect to cluster purity (top) and the V-measure (bottom), when the bandwidth h_{opt} is multiplied by a range of different values.*

§ 5.9 Running time Analysis

In this section we study the running time of the presented methods. We compare them against the running time of the original PDDP algorithm and the time required by iterative algorithms. The iterative algorithms used are the standard k-means (where the correct number of clusters is given a priori to the algorithm) and the GMM with the combination of the AIC to automatically estimate the number of clusters. Note that GMM is

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

executed for a varying number of clusters up to a maximum k_{max}, where k_{max} was set twice the actual number of clusters and that the AIC is used to select amongst them. For the dePDDP we are using the Fast Gauss Transform [YDGD03] to estimate the density of the projected points.

In detail, we are going to test the running time of the different algorithms initially for an increasing number of points and next for an increasing dimensionality. In the first case we use datasets generated from a mixture of 10 Gaussian 10 dimensional distributions that represent the actual clusters in the data and iteratively increase the data set size by sampling an increasing number of points from each Gaussian distribution. The results are illustrated in Figure 5.14 (top) (notice the log-scale of the y axis). For the second experiment, we again use a mixture of 10 Gaussian distributions, where the dimensionality of the distributions is iteratively increased. The results are illustrated in Figure 5.14 (bottom). The running time of each algorithm is measured in seconds and the CPU used is an Intel(R) Xeon(R) E5405, operating at $2.00GHz$, having 4 gigabytes of RAM.

As shown the k-means algorithm is the fastest algorithm with respect to the number of points in the data. PDDP, iPDDP and dePDDP have similar performance and they seem quite competitive to k-means. GMM-AIC on the other hand needs many orders of magnitude large running times. Note that all the algorithms show a similar increasing trend. The results for the increasing dimensionality case are reversed. In this case PDDP turns out to be the fastest algorithm, closely followed by iPDDP and dePDDP. The k-means algorithm in this case is heavily affected by the increasing dimensionality, resulting in very large running times. The running time of GMM was enormous and after 200 dimensions the algorithm exhibited numerical stability problems. Since it was unable to converge, it was not included in the plot. In general, we observe that the performance of the proposed methods are quite competitive even to the simplest of iterative methods and show very good performance when the dimensionality is increasing.

§ 5.10 Benchmark Datasets

In this section we test the performance of the algorithms on benchmark datasets from the UCI Machine Learning Repository [BM98]. These datasets have been used for the evaluation of a number of similar algorithms [TV07, Hua98, Fis87, MS83], so easy comparison to other approaches is facilitated. In particular we use the following datasets:

- (VOTES): This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA. The CQA lists nine different types of votes: voted for, paired for, and announced for (these three simplified to yea), voted against,

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

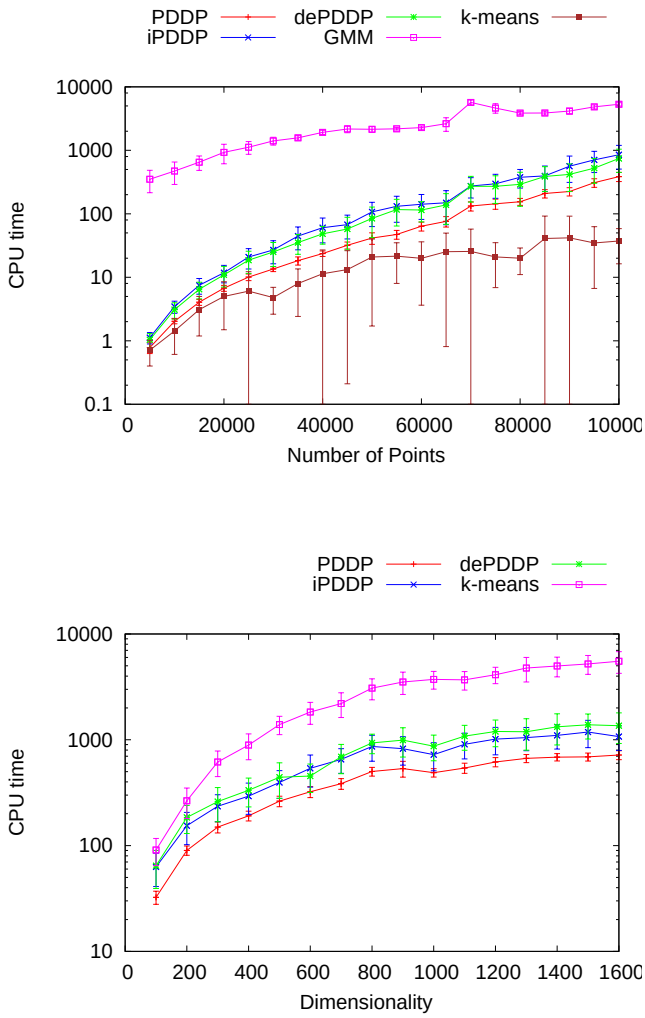


Figure 5.14: *The CPU time cost of the algorithms as the size of the dataset grows (top). The CPU time cost of the algorithms as the dimensionality grows (bottom).*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

paired against, and announced against (these three simplified to nay), voted present, voted present to avoid conflict of interest, and did not vote or otherwise made a position known (these three simplified to an unknown disposition). Thus, the data are described by 16 attributes. The total number of objects is 435; 267 of which are labelled as democrats, while the remaining 168 as republicans.

- (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 datasets, 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.

- (SYNTHETIC CONTROL): This data consists of 600 examples of control charts synthetically generated by the process in [AM99]. There are six different classes of control charts: Normal, Cyclic, Increasing trend, Decreasing trend, Upward shift, and Downward shift.

The results in terms of cluster purity and V-measured are shown in Table 5.10. Here, the PDDP algorithm manages to achieve very good results for the BREAST-CANCER dataset, but its performance deteriorates in the other two datasets. On the other hand, the iPDDP algorithm seems able to provide high purity partitions, but the associated V-measures are very low, indicating that some of the actual classes in the data are not split at all. This translates to possible chaining effects, since this is the main reason for the iPDDP failure, as shown by the experimental analysis on simulated data. The KM-PDDP algorithm, in the case of the SYNTHETIC CONTROL datasets, fails to provide good results in terms of V-measure. Possible because of the structure of this dataset which contains uneven classes that the algorithm fails to capture. The GMM algorithm seems to have difficulties to deal with the VOTES dataset, but manages to achieve good results in the other two cases. On the other hand, k-means provides very good results in all cases, supporting its popularity in low dimensional data. PG-means provides good results amongst all GMM based approaches and, in general, is competitively. Finally, the dePDDP algorithm achieves highly pure partitions in all tested datasets. It is of equal importance that in the considered problems, the dePDDP algorithm never underestimates the number of clusters, providing approximations that in two out of the three cases are very close to the actual number of classes, as well. Notice that the number of clusters is given as input to the rest of the clustering algorithms.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Dataset		BREAST-CAN.	VOTES	SYNTH. CONT.
Classes		2	2	6
	Clusters	2	2	6
PDDP	Pur.	0.9639	0.8432	0.5766
	V-meas.	0.8103	0.4097	0.5780
	Clusters	2	2	6
iPDDP	Pur.	0.8338	0.8065	0.7726
	V-meas.	0.0547	0.0033	0.7517
	Clusters	11	3	11
dePDDP	Pur.	0.9174	0.7837	0.9076
	V-meas.	0.7634	0.3784	0.7600
	Clusters	2	2	6
KM-PDDP	Pur.	0.9617	0.8489	0.8613
	V-meas.	0.7634	0.3944	0.0163
	Clusters	2	2	6
GMM	Pur.	0.8663	0.6252	0.7479
	V-meas.	0.5553	0.0086	0.5219
	Clusters	5	3	2
GMM-BIC	Pur.	0.9463	0.6565	0.5888
	V-meas.	0.4544	0.0539	0.1203
	Clusters	5	2	13
GMM-AIC	Pur.	0.8670	0.6302	0.8745
	V-meas.	0.4882	0.0151	0.6597
	Clusters	5	4	2
GMM-ICL	Pur.	0.9466	0.7123	0.3931
	V-meas.	0.4467	0.0368	0.3511
	Clusters	6	6	5
PG-means	Pur.	0.9472	0.8140	0.8154
	V-meas.	0.5034	0.3402	0.2698
	Clusters	2	2	6
k-means	Pur.	0.9571	0.8434	0.7637
	V-meas.	0.7361	0.4037	0.7013

Table 5.10: *Results with respect to the clustering purity and V-measure for the three UCI repository data sets.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

§ 5.11 Text Mining

Text mining was the initial motivation behind the development of the original PDDP algorithm. To this end, to examine the performance of the proposed approaches in text datasets we use a subset of the original TDT2 corpus. The TDT2 corpus (Nist Topic Detection and Tracking corpus) consists of data collected during the first half of 1998 and taken from 6 sources, including 2 newswires (APW and NYT), 2 radio programs (VOA and PRI) and 2 television programs (CNN and ABC). In total it consists of 11201 on-topic documents, which are classified into 96 semantic categories. Here, we used 4 samples from the initial corpora, containing 2, 3, 4, and 5 categories, respectively. For each sample, 50 randomly generated sub-samples were used, provided in http://www.cs.uiuc.edu/homes/dengcai2/Data/TextData.html.

The results are illustrated in Fig 5.15. Each boxplot depicts the obtained values for the clustering purity, in each of the random sub-samples. The box depicts the interquantile range of the data and contains a bold line at the data median. The lines extending from each end of the box indicate the range covered by the remaining data.

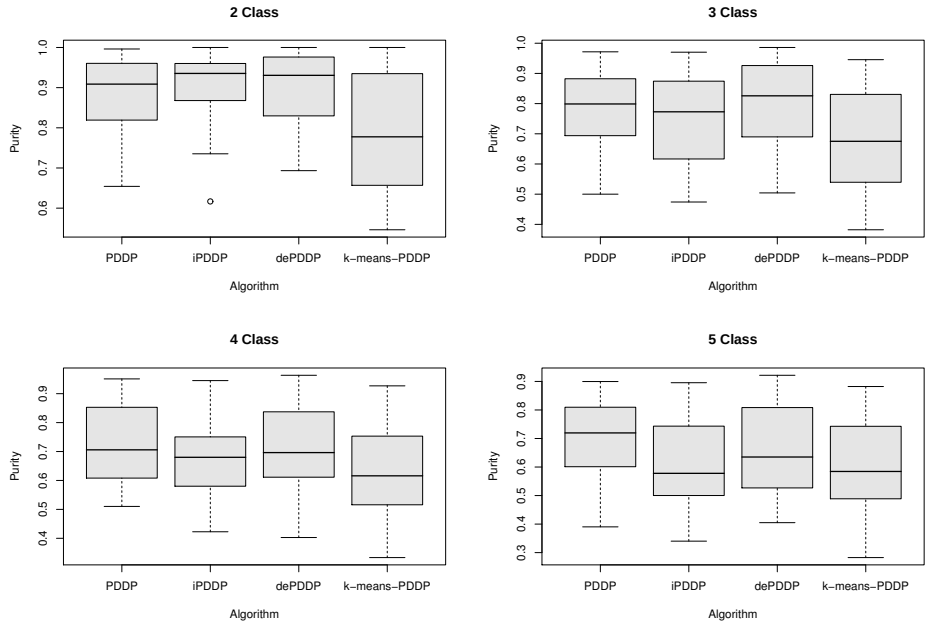


Figure 5.15: *The purity of clustering results for the TDT2 corpus subsets.*

As the plots indicate, the PDDP algorithm is quite effective in recognising the structure of the data, especially in the 2 and 3 class cases. In

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

those particular cases, the efficiency the iPDDP and dePDDP is always on par and marginally improved. The 4 class and 5 class case is a bit different. Here, the PDDP accuracy has a median value around 70%, which points out to the fact that the data are quite complex so deriving the class structure is quite difficult. For these cases, the iPDDP algorithm produces results with reduced partitioning purity. The KM-PDDP algorithm provides the worst results on all cases.

§ 5.12 Concluding Remarks

In this Chapter, we try to deepen our understanding on what can be achieved by approaches based on Principal Direction Divisive Clustering. We first make assumptions on the nature of the true clusters in the data ("inductive bias") and then attempt to theoretically discover the relationship between the true clusters and the distribution of their projection onto the principal components. Based on that, appropriate criteria for the various steps involved in hierarchical divisive clustering have been developed. At a next step, these criteria are combined into new algorithms and their effectiveness is investigated.

The presented algorithms require minimal user-defined parameters and have the desirable feature of being able to provide approximations for the number of clusters present in a dataset. This in itself is an open problem in cluster analysis and little has been done in the past for the high dimensional data case that we are mostly concerned with here.

The included experimental results indicate that the presented techniques are effective both in terms of partitioning the data and determining the true number of clusters in the dataset. Their performance is very good, even compared against the popular density based methods. Finally, to stress test the presented methods a series of experiments on real world and test datasets are included. In this case, density based techniques cannot be directly applied and complicated feature selection methods are required to actually get results [TPV06]. However, the presented algorithms exhibited promising results even in these hard and extremely interesting problems.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

- Chapter 6 -

Clustering of Ultra High Dimensional Data

There are things which seem incredible to most men who have not studied Mathematics.

—*Archimedes*

§ 6.1 Introduction

In this Chapter, we combine the RP method, (see Section 4.2.3) with the dePDDP technique to create a new effective and computationally efficient clustering methodology. While constructing a full PCA representation has a complexity that is polynomial with respect to the data dimension, RP's complexity is linear. This suggests that by applying RP first to get a reduced dimension representation and using PCA later could also reduce the total complexity significantly.

§ 6.2 Random Direction Divisive Clustering

In this section two new lemmas are introduced that lead as into constructing new algorithmic frameworks that compine RP and PCA. First based on the Johnson and Lindenstrauss lemma (see Section 4.2.3.1), utilizing the definitions of Section 5.1 and without making any assumptions for the data distribution, we can show the following:

Lemma 6.2.1. *Given $0 < \varepsilon < 1$ and an integer n, let r be a positive integer such that $r \geq r_0 = O(\varepsilon^{-2} \log n)$. For every set D of n points in $\mathbb{R}^a$ there exists $g : \mathbb{R}^a \to \mathbb{R}^r$ such that*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

$$\hat{f}^r(g(x);h) \leq \frac{e^{\sqrt{1-\epsilon}}}{(2\pi)^{(a-r)/2}h^{r-a}}\hat{f}(x;h),$$

where

$$\hat{f}^r(g(x);h) = n^{-1}h^{-r}\sum_{i=1}^{n} K\left((g(x)-g(x_i))/h\right),$$

with kernel function

$$K^r(g(x)) = (2\pi)^{-r/2}e^{-0.5\|g(x)\|}.$$

Proof. Applying the Johnson and Lindenstrauss lemma, we know that for all $x, x_i \in \mathbb{R}^a$ holds that

$$\begin{aligned}
\|g(x)-g(x_i)\|^2 &\geqslant (1-\epsilon)\|x-x_i\|^2 \Leftrightarrow \\
\|g(x)-g(x_i)\| &\geqslant \sqrt{1-\epsilon}\|x-x_i\| \Leftrightarrow \\
-0.5\|g(x)-g(x_i)\| &\leqslant -0.5\sqrt{1-\epsilon}\|x-x_i\| \Leftrightarrow \\
e^{-0.5\|g(x)-g(x_i)\|} &\leqslant e^{-0.5\sqrt{1-\epsilon}\|x-x_i\|}.
\end{aligned}$$

Since $e^{ab} \leq e^a e^b$ when a and b have different sign we have that

$$\begin{aligned}
e^{-0.5\|g(x)-g(x_i)\|} &\leqslant e^{\sqrt{1-\epsilon}}e^{-0.5\|x-x_i\|} \Leftrightarrow \\
(2\pi)^{(a-r)/2}(2\pi)^{(r/2)}e^{-0.5\|g(x)-g(x)_i\|} &\leqslant e^{\sqrt{1-\epsilon}}(2\pi)^{(a/2)}e^{-0.5\|x-x_i\|}.
\end{aligned}$$

Now, we can sum each part of the inequality, for all $i = 1, \ldots, n$. As such we have

$$(2\pi)^{(a-r)/2}\sum_{i=1}^{n}(2\pi)^{(r/2)}e^{-0.5\|g(x)-g(x_i)\|} \leqslant e^{\sqrt{1-\epsilon}}\sum_{i=1}^{n}(2\pi)^{(a/2)}e^{-0.5\|x-x_i\|}.$$

And by multiplying by n^{-1} and h^{-a} we have

$$\begin{aligned}
(2\pi)^{(a-r)/2}h^{r-a}\hat{f}^r(g(x);h) &\leqslant e^{\sqrt{1-\epsilon}}\hat{f}(x;h) \Leftrightarrow \\
\hat{f}^r(g(x);h) &\leqslant \frac{e^{\sqrt{1-\epsilon}}}{(2\pi)^{(a-r)/2}h^{r-a}}\hat{f}(x;h).
\end{aligned}$$

□

The above extends the result of Lemma 4.2.1, in terms of the kernel density estimate of the data projection. Using the above result, we extend Lemma 5.2.3 as follows:

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Lemma 6.2.2. *Let a dense convex k-clusterable set $\mathcal{D}$ of points $d_i \in R^a$, for $i = 1, \ldots, n$ and Π its partition into k dense convex subsets $\mathcal{C}_1, \ldots, \mathcal{C}_k$. Let D^{RP} be the set of projections Rd_i of the vectors d_i on the orthogonal matrix $R_{a\times r}$. Let $u \in R^r$, with $\|u\| = 1$. Let $\mathcal{P}$ be the set of projections p_i of the vectors Rd_i on u. Also, let $M = \min\{\hat{f}(x;h) : x \in \mathcal{C}, \mathcal{C} \in \Pi\}$ and $A = \frac{e^{\sqrt{1-\epsilon}}}{(2\pi)^{(a-r)/2}h^{r-a}}$ given $0 < \varepsilon < 1$. If there exists $x \in \mathbb{R}$, such that the univariate estimate density $\hat{f}'(x;h') < AM$, with $h = h' > 1$ or $h' \leqslant h^a$, then all d_t for which $p_t > x$, and d_l for which $p_l \leqslant x$, belong to different sets of Π.*

Proof. Let us suppose that there exist $d_t, d_l \in \mathcal{D}, \mathcal{C} \in P$ such that $d_t, d_l \in \mathcal{C}$, and $p_t = uRd_t$ and $p_l = uRd_l$ with $p_l \leqslant x$ and $p_t > x$. Obviously, there exists $\lambda \in [0,1]$, such that $x = \lambda p_l + (1-\lambda)p_t$. Also for the vector $z \in \mathbb{R}^a$ with $z = \lambda d_l + (1-\lambda)d_t$, it holds that $z \in \mathcal{C}$ since $\mathcal{C}$ is convex (note that $uRz = x$).

As such $\hat{f}'(uRz;h') = \hat{f}'(x;h') < AM$. However, since $\hat{f}(x;h) \leqslant \hat{f}'(ux;h')$, we have that $\hat{f}(Rz;h) \leqslant \hat{f}'(uRz;h') = \hat{f}'(x;h')$. Also, from lemma 6.2.1, we have

$$\begin{aligned} \hat{f}(Rz;h) &\geqslant A\hat{f}(z:h) \Leftrightarrow \\ A\hat{f}(z;h) &\leqslant \hat{f}'(x;h') \Leftrightarrow \\ A\hat{f}(z;h) &\leqslant AM \Leftrightarrow \\ \hat{f}(z;h) &\leqslant M \end{aligned}$$

and since $z \in \mathcal{C}$, $\hat{f}(z;h) > M$, this is a contradiction. Thus, the Lemma is proved. □

6.2.1 Constructing New RP Clustering Algorithms

The above theoretical results allow us to use RP and PCA in the dePDDP scheme to form new algorithms, with the promise of very low computational complexity. As this combination can be performed in two different ways, we formulate the following algorithms:

- Random Projection dePDDP *(rp-dePDDP)*:
  In this case the data are initially projected onto a random frame as explained in Section 4.2.3.1. Then the dePDDP algorithm is applied as before, but on that reduced dimension representation.

- Random Direction Divisive Partitioning *(RDDP)*:
  Here the RP method is used at each stage of the dePDDP process. As each subcluster is formed, the data belonging to just that cluster are projected onto a new random frame of reduced dimension, and then PCA is applied on it to get the final one dimensional projection.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Obviously, the RDDP is deemed to be computationally more expensive, and as will be shown in Section 6.3, without significant performance gains. However, for completeness purposes, it is included in this study.

6.2.1.1 Random Line RDDP

We can also formulate a simplified combination of the Random Projection method and dePDDP. In detail, we can create a random matrix R in the way described in Section 6.2, using $r = 1$. Thus, in essence:

$$D^{RP}_{n\times 1} = D_{n\times a} R_{a\times 1},$$

is the projection of the original set onto a random line. This kind of projection has been considered in a number of settings, including the approximation of nearest neighbors [Kle97, KOR98] and the learning intersections of halfspaces [Vem10]. Additionally, such projections have been used in learning mixture of Gaussians models [Das99, SK01].

This way we construct a algorithm, based on dePDDP that discards PCA altogether, and only uses projections of the data onto such a random line. We will refer to this method as Random Line RDDP *(rl-RDDP)*.

§ 6.3 Experimental Analysis

In this section, we perform a series of experiments to evaluate the effectiveness of the approaches presented in Section 6.2.1. Here, because of the strong relation between the algorithms, we also evaluate the statistical significance of the observed performance differences, for each algorithm we conducted three two-sample Wilcoxon rank sum tests between the results of that algorithm and the results of each of the three proposed methods rl-RDDP, rp-dePDDP and RDDP, respectively. The null hypothesis in each test is that the samples come from identical continuous distributions with equal medians, against the alternative that they do not have equal medians. In the tables summarizing the results, we report the result of the test as 1, when the null hypothesis is rejected at the 5% significance level and as 0 otherwise.

Table 6.1 reports the purity and the V-measure including the results of the statistical tests of the PDDP, dePDDP, km-PDDP [ZG07], rl-RDDP,rp-dePDDP, RDDP, k-means and rp-kmeans [BZD10] algorithms in 100 randomly generated datasets, using the previously described $DSET_{\text{Gausian}}$. The rp-kmeans algorithm is the standard k-means, applied to the dataset that is a priori projected onto a random frame. For k-means and rp-kmeans algorithms, we employ Matlab's k-means implementation. Each entry of Table 6.1 is the mean observed value of the corresponding measure obtained

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

over the 100 different datasets and the number in parentheses is the observed standard deviation. A standard deviation of 0 corresponds to a observed variance of less than 10^{-2}. The error rate of the RP method was set to 0.5 for all algorithms and the actual cluster number was given as input to PDDP, km-PDDP, k-means and rp-kmeans algorithms. To estimate the density of the projected data the algorithms are using the Fast Gauss Transform [YDGD03].

Dimension 100				
	25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.95 (0.03) (0/1/1)	0.98 (0.01) (0/1/1)	0.88 (0.04) (1/1/1)	0.95 (0.01) (1/1/1)
dePDDP	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.97 (0.15) (1/0/0)	0.97 (0.13) (1/0/0)
km-PDDP	0.74 (0.06) (1/1/1)	0.93 (0.01) (1/1/1)	0.78 (0.03) (1/1/1)	0.95 (0.00) (1/1/1)
rl-RDDP	0.93 (0.09) (0/1/1)	0.97 (0.04) (0/1/1)	0.96 (0.02) (0/1/1)	0.98 (0.00) (0/1/1)
rp-dePDDP	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)
RDDP	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)
rp-kmeans	0.85 (0.03) (1/1/1)	0.94 (0.01) (1/1/1)	0.82 (0.02) (1/1/1)	0.95 (0.00) (1/1/1)
k-means	0.86 (0.03) (1/1/1)	0.95 (0.01) (1/1/1)	0.84 (0.02) (1/1/1)	0.96 (0.01) (1/1/1)
Dimension 1000				
	25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.97 (0.03) (1/1/1)	0.99 (0.01) (1/0/0)	0.94 (0.04) (0/1/1)	0.98 (0.01) (0/1/1)
dePDDP	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.97 (0.05) (1/0/0)	0.98 (0.01) (1/0/0)
km-PDDP	0.74 (0.05) (1/1/1)	0.93 (0.01) (1/1/1)	0.76 (0.03) (1/1/1)	0.95 (0.00) (1/1/1)
rl-RDDP	0.96 (0.03) (0/1/1)	0.98 (0.01) (0/1/1)	0.95 (0.02) (0/1/1)	0.98 (0.00) (0/1/1)
rp-dePDDP	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)
RDDP	1.00 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.99 (0.03) (1/0/0)	0.99 (0.01) (1/0/0)
rp-kmeans	0.84 (0.03) (1/1/1)	0.94 (0.01) (1/1/1)	0.83 (0.04) (1/1/1)	0.95 (0.00) (1/1/1)
k-means	0.85 (0.05) (1/1/1)	0.95 (0.02) (1/1/1)	0.82 (0.02) (1/1/1)	0.95 (0.01) (1/1/1)
Dimension 5000				
	25 Clusters		50 Clusters	
	Pur.	V-meas.	Pur.	V-meas.
PDDP	0.98 (0.02) (1/1/1)	0.99 (0.00) (1/0/0)	0.97 (0.01) (1/1/1)	0.99 (0.00) (1/1/1)
dePDDP	0.94 (0.08) (1/1/1)	0.97 (0.02) (0/1/1)	0.93 (0.12) (1/0/0)	0.97 (0.06) (1/0/0)
km-PDDP	0.74 (0.04) (1/1/1)	0.93 (0.01) (1/1/1)	0.76 (0.03) (1/1/1)	0.95 (0.00) (1/1/1)
rl-RDDP	0.95 (0.05) (0/1/1)	0.98 (0.01) (0/1/1)	0.90 (0.22) (0/1/1)	0.92 (0.22) (0/1/1)
rp-dePDDP	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)
RDDP	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)	0.99 (0.00) (1/0/0)
rp-kmeans	0.84 (0.03) (1/1/1)	0.95 (0.01) (1/1/1)	0.82 (0.02) (1/1/1)	0.95 (0.00) (1/1/1)
k-means	-	-	-	-

Table 6.1: *Mean purity, V-measure (with the observed standard deviation in parenthesis) and statistical significance test for* $DSET_{\text{Gausian}}$ *generated data of different algorithms over 100 experiments.*

As shown, the rp-dePDDP and RDDP algorithms are as effective as dePDDP algorithm in all cases, while rl-RDDP, as expected, performs sightly worse. As we will see in the next section, the proposed algorithms are much more computationally efficient, so the fact that they maintain the accuracy of the original method is very promising.

Note that the results of the dePDDP based algorithms on the automatic

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

determination of the number of clusters is almost identical on this experiment and are not included for brevity. In all cases the algorithms retrieve no more than one or two extra clusters [TTP10b]. The results of k-means for 5000 dimensions are also excluded due to the extremely high computational time required.

6.3.1 Computational Cost

In this section, we study the running time of the proposed methods. Table 6.2 reports the running time in seconds of the previous experiment. As expected, the RP based methods are much faster especially for the high dimensional cases. In the case of 5000 dimensions, the rl-RDDP and RDDP algorithms perform as fast as rp-kmeans, while rp-dePDDP remains the fastest method in all cases.

Dimension 100		
	25 Clusters	50 Clusters
PDDP	1.17 (0.04) (1/1/1)	2.78 (0.06) (1/1/1)
dePDDP	1.69 (0.26) (1/1/1)	5.33 (1.01) (1/1/1)
km-PDDP	6.34 (1.13) (1/1/1)	21.79 (6.56) (1/1/1)
rl-RDDP	0.10 (0.01) (0/1/1)	0.36 (0.07) (0/1/1)
rp-dePDDP	0.72 (0.10) (1/0/0)	2.34 (0.41) (1/0/0)
RDDP	0.68 (0.09) (1/0/0)	2.25 (0.37) (1/0/0)
rp-kmeans	1.93 (0.37) (1/1/1)	12.04 (1.54) (1/1/1)
k-means	23.56 (3.47) (1/1/1)	96.37 (6.36) (1/1/1)
Dimension 1000		
	25 Clusters	50 Clusters
PDDP	11.34 (0.37) (1/1/1)	27.78 (0.51) (1/1/1)
dePDDP	17.14 (2.73) (1/1/1)	54.53 (12.61) (1/1/1)
km-PDDP	22.41 (4.19) (1/1/1)	82.43 (19.81) (1/1/1)
rl-RDDP	0.62 (0.11) (0/1/1)	2.22 (0.40) (0/1/1)
rp-dePDDP	0.79 (0.08) (1/0/1)	2.68 (0.47) (1/0/1)
RDDP	1.28 (0.22) (1/1/0)	4.09 (0.71) (1/1/0)
rp-kmeans	2.25 (0.25) (1/1/1)	13.99 (1.13) (1/1/1)
k-means	379.80(45.74) (1/1/1)	1678.61 (122.82) (1/1/1)
Dimension 5000		
	25 Clusters	50 Clusters
PDDP	64.29 (1.70) (1/1/1)	153.97 (3.58) (1/1/1)
dePDDP	85.36 (15.67) (1/1/1)	268.36 (56.94) (1/1/1)
km-PDDP	93.30 (16.58) (1/1/1)	385.85 (76.00) (1/1/1)
rl-RDDP	4.64 (0.92) (0/1/0)	13.78 (3.84) (0/1/1)
rp-dePDDP	1.04 (0.10) (1/0/1)	3.10 (1.48) (1/0/1)
RDDP	4.60 (0.75) (0/1/0)	16.48 (5.67) (1/1/0)
rp-kmeans	2.53 (0.28) (1/1/1)	15.02 (1.46) (0/1/1)
k-means	-	-

Table 6.2: *Mean computational time in seconds (with the observed standard deviation in parenthesis) and statistical significance test for* $DSET_{\text{Gausian}}$ *generated data of different algorithms over 100 experiments.*

In the second set of experiments, we test the running time of the algorithms that utilize RP, initially for an increasing dimensionality and next

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

for an increasing number of points. In the first case, we use datasets generated from a mixture of 25 Gaussian distributions, where the dimensionality of the distributions is iteratively increased. The results are illustrated in Figure 6.1 (top). Please notice the log-scale of the y axis. Subsequently, we use a mixture of 25 Gaussian distributions and iteratively increase the data set size by sampling an increasing number of points from each Gaussian distribution. The results are illustrated in Figure 6.1 (bottom). So, in general, we can observe that the RP based methods manage to achieve comparable performance to their PCA-based counterparts to a fraction of the computational time of the latter. Moreover, the simplest method of the lot (rp-dePDDP), seems to be the method of choice with regards to both computational performance and accuracy.

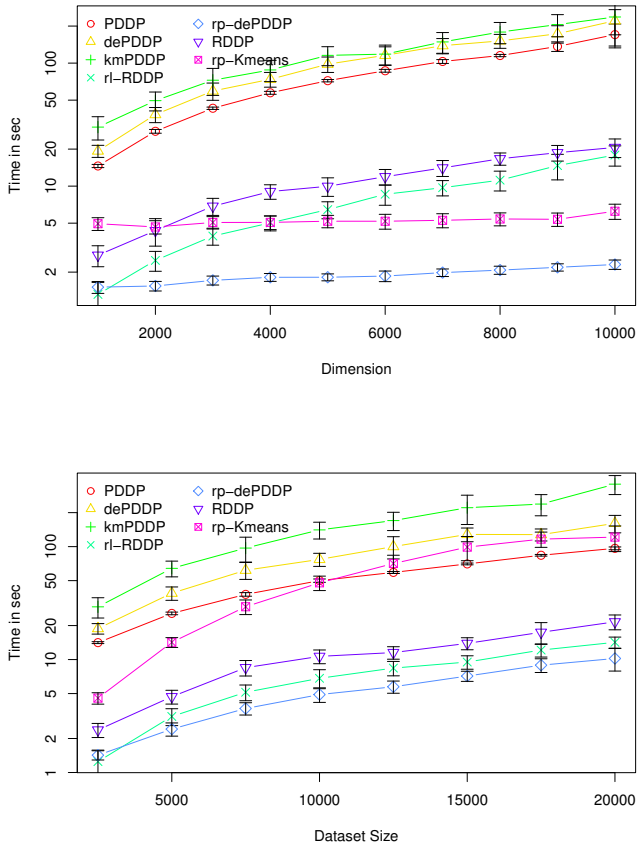


Figure 6.1: *The CPU time cost of the algorithms as the dimensionality grows (top). The CPU time cost of the algorithms as the size of the dataset grows (bottom).*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

The running time of each algorithm is measured in seconds and the system specifications are an Intel(R) Xeon(R) E5405 CPU, operating at $2.00GHz$, having 4 gigabyte of RAM.

6.3.2 Clustering Microarray data

In this Section we perform a series of experiments on the following microarray datasets (see also Section 4.3.1).

- COLON: 40 tumor and 22 normal colon tissues. 2000 gene expression level measurements.
  http://microarray.princeton.edu/oncology
- PROSTATE: 52 tumor and 50 non-tumor samples. 6033 gene expression level measurements.
  http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi
- CARCINOMA: 18 tumor and 18 normal samples. 7457 gene expression level measurements.
  http://microarray.princeton.edu/oncology/carcinoma.html
- LYMPHOMA: 62 of the 3 lymphoid malignancies samples types. 4026 gene expression levels.
  http://genome-www.stanford.edu
- SRBCT: 83 samples spanning 4 classes. 2308 gene expression level measurements.
  http://research.nhgri.nih.gov/microarray/Supplement
- LEUKEMIA: 72 samples of 2 types of acute leukemias. 7129 gene expression level measurements.
  http://www.genome.wi.mit.edu/MPR
- ALL: 248 samples of 6 subtypes of acute lymphoblastic leukemia. 12559 gene expression level measurements
  http://www.stjuderesearch.org/data/ALL1

In Table 6.3, the clustering results with respect to purity and V-measure of the algorithms for each dataset are illustrated. Again, the actual number of clusters in the data is given as input to the PDDP, KM-PDDP, k-means and rp-kmeans algorithms. The error rate of the RP method was set to 0.1 to achieve better clustering results. For rl-RDDP, RDDP, rp-dePDDP, k-means and rp-kmeans algorithms, we present the mean values and the standard deviation (in the parenthesis) over 100 experiments. Again, we also report the statistical significance of the results. As shown, both rp-dePDDP and RDDP algorithms performs as good as dePDDP and in some cases even better, while the rl-RDDP algorithm exhibits inferior performance. The

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

slight advantage of rp-dePDDP and RDDP could occur due to the fact that in some cases by projecting the data onto Random Projections, the clusters of the projected subspace are more spherical [Das99, Das00] than those in the original dimension and thus can be easier retrieved. The cases for which the proposed methods show inferior performance with respect to the V-measure are cases where the algorithms find a higher number of clusters than the actual one. Table 6.4 reports the running time of the algorithms in seconds for each dataset. Here, as expected due to the nature of microarray datasets that are being characterized by very small number of samples, there are not significant differences amongst the algorithms.

6.3.3 Face Recognition

In this experiment er examine the applicability of the methods presented in Section 6.2.1 in a unsupervised facial recognition task (see Section 4.3.3). Two different datasets are employed. The *ORL* dataset [SH94] contains ten different images of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). Each image has an analysis of 112×92 pixels, so the final size of the data matrix is 400×10304 pixels.

The second set was the *Yale* Face Database [GBK01], containing 5760 single light source images of 10 subjects each seen under 576 viewing conditions (9 poses x 64 illumination conditions). Each image has an analysis of 640×480 pixels. However, we have selected 400 images of them randomly belonging to 9 persons, to maintain an equivalence to the ORL dataset. Thus, the size of the data matrix is 400×307200.

The results are illustrated in Table 6.5. Note that only the RP based methods are used here since the computational complexity of the other algorithms is prohibitive for these datasets. As shown in the case of the ORL dataset, none of the algorithms is able to recognize the clustering structure of the data, as all of them produce clusterings with very low purity. For the case of the Yale dataset, the situation is reversed, contradicting our expectation of further performance reduction, as the dimensionality grows by a factor of 30. This indicates that increasing dimensionality is transformed from a “curse” to a “blessing”. The additional information of the high resolution images seems to be preserved by the Random Projection method, in a way that allows the algorithms to efficiently utilize it. In particular, the rp-dePDDP and RDDP methods produce almost totally pure clusterings, with an increased cluster number. However, the high purity of the results, combined with the moderate V-measure values, indicates that the resulting partitions also have a logical meaning. An example partition of a subset of

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Dataset		COL	PRO	CAR	LYM	SRB	LEY	ALL
Classes		2	2	2	3	4	2	6
	Cl.	2.00 (0.00)	2.00 (0.00)	2.00 (0.00)	3.00 (0.00)	4.00 (0.00)	2.00 (0.00)	6.00 (0.00)
	Pur.	0.65 (0.00)	0.58 (0.00)	0.75 (0.00)	0.97 (0.00)	0.62 (0.00)	0.90 (0.00)	0.33 (0.00)
PDDP		(1/1/1)	(0/1/1)	(1/1/1)	(1/1/1)	(1/1/1)	(1/1/1)	(1/1/1)
	V-m.	0.03 (0.00)	0.02 (0.00)	0.20 (0.00)	0.88 (0.00)	0.42 (0.00)	0.56 (0.00)	0.04 (0.00)
		(1/1/1)	(0/1/1)	(0/1/1)	(1/0/1)	(1/1/1)	(1/0/1)	(1/1/1)
	Cl.	7.00 (0.00)	12.00 (0.00)	4.25 (0.44)	9.00 (0.00)	10.00 (0.00)	2.00 (0.00)	21.00 (0.00)
	Pur.	0.79 (0.00)	0.80 (0.00)	0.87 (0.02)	1.00(0.00)	0.84 (0.00)	0.96 (0.00)	0.77 (0.00)
dePDDP		(1/0/0)	(1/1/1)	(1/0/0)	(1/1/1)	(1/0/0)	(1/0/0)	(1/0/0)
	V-m.	0.17 (0.00)	0.18 (0.00)	0.36 (0.04)	0.57 (0.00)	0.55 (0.00)	0.77 (0.00)	0.49 (0.00)
		(1/0/0)	(1/1/1)	(1/0/0)	(1/1/0)	(1/0/0)	(1/1/1)	(1/1/1)
	Cl.	2.00 (0.00)	2.00 (0.00)	2.00 (0.00)	3.00 (0.00)	4.00 (0.00)	2.00 (0.00)	6.00 (0.00)
	Pur.	0.65 (0.00)	0.59 (0.00)	0.66 (0.11)	0.85 (0.00)	0.49 (0.00)	0.96 (0.00)	0.33 (0.00)
KM-PDDP		(1/1/1)	(0/1/1)	(0/1/1)	(1/1/1)	(0/1/1)	(1/0/0)	(1/1/1)
	V-m.	0.02 (0.00)	0.05 (0.00)	0.14 (0.08)	0.65 (0.00)	0.28 (0.00)	0.77 (0.00)	0.03 (0.00)
		(1/1/1)	(0/1/1)	(0/1/1)	(1/0/0)	(1/1/1)	(1/1/1)	(1/1/1)
	Cl.	6.00 (3.48)	5.25 (4.34)	3.75 (1.55)	5.95 (2.39)	4.60 (3.10)	5.40 (3.57)	6.25 (2.81)
	Pur.	0.70 (0.05)	0.59 (0.07)	0.69 (0.11)	0.75 (0.05)	0.49 (0.08)	0.70 (0.05)	0.36 (0.03)
rl-RDDP		(0/1/1)	(0/1/1)	(0/1/1)	(0/1/1)	(0/1/1)	(0/1/1)	(0/1/1)
	V-m.	0.09 (0.05)	0.05 (0.04)	0.16 (0.13)	0.26 (0.09)	0.19 (0.10)	0.09 (0.07)	0.07 (0.05)
		(0/1/1)	(0/1/1)	(0/1/1)	(0/1/1)	(0/1/1)	(0/1/1)	(0/1/1)
	Cl.	5.45 (1.79)	10.05 (2.78)	3.80 (0.77)	5.05 (2.09)	9.30 (1.95)	6.60 (3.75)	17.35 (3.70)
	Pur.	0.77 (0.06)	0.72 (0.07)	0.88 (0.05)	0.99 (0.01)	0.84 (0.07)	0.96 (0.02)	0.77 (0.03)
rp-dePDDP		(1/0/0)	(1/0/1)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)
	V-m.	0.18 (0.07)	0.13 (0.04)	0.43 (0.11)	0.79 (0.19)	0.56 (0.06)	0.53 (0.14)	0.53 (0.02)
		(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)
	Cl.	5.60 (2.52)	7.90 (2.15)	4.20 (1.11)	6.65 (2.21)	8.40 (2.35)	6.40 (2.64)	16.60 (4.44)
	Pur.	0.77 (0.06)	0.67 (0.05)	0.88 (0.06)	0.99 (0.01)	0.81 (0.11)	0.96 (0.02)	0.77 (0.04)
RDDP		(1/0/0)	(1/1/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)
	V-m.	0.18 (0.04)	0.11 (0.03)	0.39 (0.09)	0.68 (0.15)	0.54 (0.10)	0.50 (0.12)	0.53 (0.03)
		(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)
	Cl.	2.00 (0.00)	2.00 (0.00)	2.00 (0.00)	3.00 (0.00)	4.00 (0.00)	2.00 (0.00)	6.00 (0.00)
	Pur.	0.68 (0.08)	0.58(0.00)	0.75(0.00)	0.94(0.07)	0.53(0.01)	0.97(0.00)	0.54(0.08)
k-means		(1/1/1)	(0/1/1)	(1/1/1)	(1/1/1)	(0/1/1)	(1/1/1)	(1/1/1)
	V-m.	0.07 (0.16)	0.02(0.00)	0.20(0.00)	0.82(0.16)	0.24(0.04)	0.81(0.00)	0.37(0.09)
		(1/1/1)	(0/1/1)	(0/1/1)	(1/0/1)	(1/1/1)	(1/1/1)	(1/1/1)
	Cl.	2.00 (0.00)	2.00 (0.00)	2.00 (0.00)	3.00 (0.00)	4.00 (0.00)	2.00 (0.00)	6.00 (0.00)
	Pur.	0.67 (0.07)	0.58 (0.01)	0.75 (0.07)	0.91 (0.07)	0.55 (0.03)	0.97 (0.01)	0.53 (0.08)
rp-kmeans		(1/1/1)	(0/1/1)	(1/1/1)	(1/1/1)	(1/1/1)	(1/0/0)	(1/1/1)
	V-m.	0.07 (0.14)	0.02 (0.00)	0.21 (0.13)	0.76 (0.17)	0.27 (0.04)	0.80 (0.08)	0.34 (0.09)
		(1/1/1)	(0/1/1)	(0/1/1)	(1/0/0)	(1/1/1)	(1/1/1)	(1/1/1)

Table 6.3: *Mean purity, V-measure, number of clusters discovered (with the observed standard deviation in parenthesis) and statistical significance test for the microarray datasets.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Dataset	COL	PRO	CAR	LYM	SRB	LEY	ALL
PDDP	0.56 (0.15)	1.74 (0.18)	1.79 (0.48)	1.71 (0.31)	1.27 (0.45)	1.08 (0.22)	12.10 (0.32)
	(1/1/1)	(1/1/1)	(1/1/0)	(1/1/1)	(1/1/0)	(1/1/1)	(1/1/1)
dePDDP	1.83 (0.04)	9.95 (0.09)	3.79 (0.23)	4.83 (0.26)	2.75 (0.07)	1.01 (0.05)	44.91 (1.78)
	(1/1/1)	(1/1/1)	(1/1/1)	(1/1/1)	(1/1/1)	(1/1/1)	(1/1/1)
KM-PDDP	0.47 (0.03)	1.72 (0.02)	1.53 (0.23)	1.81 (0.14)	1.16 (0.12)	1.03 (0.03)	11.12 (0.19)
	(1/0/1)	(1/1/1)	(1/1/0)	(1/1/0)	(1/1/1)	(1/1/1)	(1/1/1)
rl-RDDP	0.03 (0.01)	0.10 (0.07)	0.06 (0.02)	0.07 (0.02)	0.03 (0.01)	0.06 (0.02)	0.80 (0.29)
	(0/1/1)	(0/1/1)	(0/1/1)	(0/1/1)	(0/1/1)	(0/1/1)	(0/1/1)
rp-dePDD	0.47 (0.11)	1.09 (0.18)	0.52 (0.04)	0.59 (0.17)	0.74 (0.14)	0.71 (0.25)	2.83 (0.29)
	(1/0/1)	(1/0/1)	(1/0/1)	(1/0/1)	(1/0/1)	(1/0/1)	(1/0/1)
RDDP	0.86 (0.35)	3.39 (0.72)	1.68 (0.39)	2.04 (0.67)	1.34 (0.36)	1.68 (0.58)	16.32 (2.88)
	(1/1/0)	(1/1/0)	(1/1/0)	(1/1/0)	(1/1/0)	(1/1/0)	(1/1/0)
k-means	0.89 (0.19)	2.05 (0.16)	1.68 (0.23)	2.01 (0.35)	1.73 (0.33)	1.58 (0.47)	50.61 (15.66)
	(1/1/0)	(1/1/1)	(1/1/0)	(1/1/0)	(1/1/1)	(1/1/0)	(1/1/1)
rp-kmeans	0.31 (0.05)	0.56 (0.03)	0.48 (0.05)	0.49 (0.05)	0.51 (0.06)	0.52 (0.23)	2.72 (0.38)
	(1/1/1)	(1/1/1)	(1/1/1)	(1/0/1)	(1/1/1)	(1/1/1)	(1/0/1)

Table 6.4: *Mean computational time in seconds(with the observed standard deviation in parenthesis) and statistical significance test for the microarray datasets.*

the data is exhibited in Figure 6.2. In this sample, there exist 3 different persons in 10 different poses. Most of the algorithms in this example provide a purity of 1.0, which is a totally clear partition, but they estimate the number of clusters to 4. The images of one of the persons is split in two clusters, possibly because of the different luminosity in the two sets, as no image transformations are used to account for that. That is the reason we see reduced V-measure values.

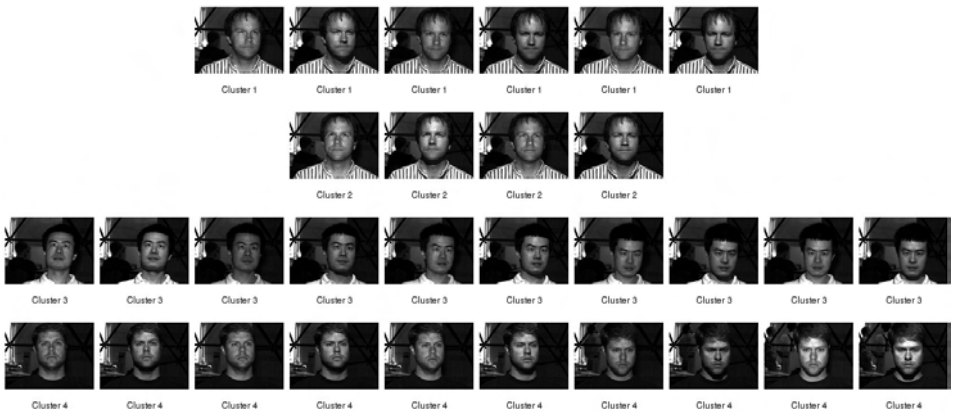


Figure 6.2: *Clustering result for 3 different persons in 10 different poses.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

ORL Dataset				
	rl-RDDP	rp-dePDDP	RDDP	rp-kmeans
Pu.	0.32 (0.12)	0.44 (0.11)	0.50 (0.07)	0.54 (0.02)
	(0,0,1)	(0,0,0)	(1,0,0)	(1,1,1)
Vm.	0.52 (0.19)	0.62 (0.08)	0.65 (0.03)	0.71 (0.01)
	(0,0,1)	(0,0,0)	(1,0,0)	(1,1,1)
Cl.	52.0 (23.4)	54.5 (15.9)	56.0 (10.3)	40.0 (0.0)
Ct.	1.30 (0.63)	0.87 (0.48)	2.48 (0.32)	0.60 (0.12)
	(0,0,1)	(0,0,1)	(1,1,0)	(1,1,1)
Yale Dataset				
	rl-RDDP	rp-dePDDP	RDDP	rp-kmeans
Pu.	0.63 (0.20)	0.96 (0.03)	0.94 (0.01)	0.76 (0.08)
	(0,1,1)	(1,0,0)	(1,0,0)	(1,1,1)
Vm.	0.51 (0.16)	0.66 (0.02)	0.67 (0.01)	0.76 (0.06)
	(0,1,1)	(1,0,0)	(1,0,0)	(1,1,1)
Cl.	42.5 (20.1)	65.0 (8.1)	63.0 (5.7)	9.0 (0.0)
Ct.	67.40 (27.38)	2.77 (0.24)	81.94 (8.17)	1.90 (0.07)
	(0,1,0)	(1,0,1)	(0,1,0)	(1,1,1)

Table 6.5: *Mean purity, V-measure, number of clusters discovered, computational time in seconds (with the observed standard deviation in parenthesis) and statistical significance test for ORL and Yale datasets over 10 experiments.*

§ 6.4 Concluding Remarks

For data sets with ultra high dimensionality even the application of Principal Component Analysis becomes problematic. For this type of problems, Random Projection has been proposed as a computational efficient alternative, with appealing theoretical characteristics. In this Chapter, we examine how Random Projection can be used into the dePDDP clustering framework. Additionally, we theoretically study the properties of the resulting framework.

In our analysis, we show that as the Random Projection method does not significantly alter the distribution of the data on the projected space, the theoretical characteristics of the clustering algorithms remain valid. This result suggests that the clustering framework would suffer minimal performance losses, while gaining all the computational savings of the Random Projection method.

Finally, using an experimental analysis of a combination of simulated and real world datasets, we show that the resulting algorithms can at least maintain the performance of the original Principal Component Analysis based algorithms in a fraction of the computation time.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

- Chapter 7 -

Clustering on Alternative Projection Directions

In mathematics the art of proposing a question must be held of higher value than solving it.

—*Geoge Cantor*

§ 7.1 Introduction

In this Chapter we examine the use of alternative methods for finding suitable projections directions. In Section 7.2 we utilize a projection pursuit method to create a hierarchical clustering framework that can be applied on gene expression microarray data. Next, in Section 7.3, a new criterion of direction interestingness is presented, which incorporates information from the density of the projected data.

§ 7.2 Independent Component Divisive Clustering

In this Section, we employ dePDDP [TTP10b] and substitute its projection methodology with Independent Component Analysis (ICA) (see Section 4.2.2). The ICA is a technique capable of finding the underlying factors or sources, when other classic methods fail. This allows the algorithm though the proposed modification to better use the density of the projected data to guide the clustering process.

The critical attribute of the ICA model is that we can use it to find directions for which the 1-dimensional projected data onto these directions show the least Gaussian distribution. It has been argued by Huber [Hub85] and by Jones and Sibson [JS87] that the Gaussian distribution is the least

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

interesting one, and that the most interesting directions are those that show the least Gaussian distribution. Interesting distribution can be a consider a distribution that captures the structure of the data. As such ICA can be considered as an 1-dimensional projection pursuit technique for finding directions of maximum nongaussianity. Since we need to find only one direction there is not any need to estimate all the independent components. The independent components are obtained in descending order of nongaussianity so the most interesting independent components are obtained first. This significantly reduces the computational cost of the method, especially in the case of high dimensional data.

A two dimensional example of such a case is shown at Figure 7.1. As shown, the non-Gaussian and the principal direction are almost identical for the dataset at Figure 7.1 (left). On the other hand, these two directions are almost vertical for the dataset shown at figure Figure 7.1 (right). In this case, although the two actual clusters are clearly visible, if we project the data onto the principal component would not be possible to distinguish them. However, the same does not happen when the data are projected onto the non-Gaussian direction. As such this is a direction with a higher level of interestingness.

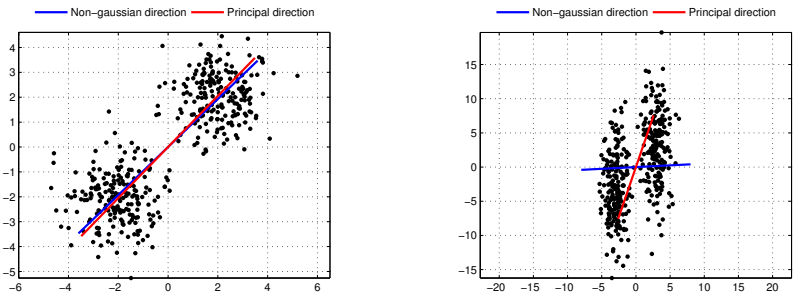


Figure 7.1: *Projection Pursuit: a two dimensional example.*

To find the direction of maximum non-gaussianity, we utilize a well known fixed point algorithm. The FastICA algorithm [Hyv99] is a very efficient method for maximizing the objective function with respect to the selected measure of non-gaussianity. For this task, it is assumed that the data is preprocessed. The most basic preprocessing is to center the data. Centering is performed to simplify the execution of the ICA algorithm. The next step of preprocessing is the data whitening. Before the application of ICA and after centering, the data matrix is transformed into a new one that is white, meaning that its components are uncorrelated and their variances are equal to unity. Whitening can be achieved with principal component analysis or singular value decomposition. Whitening ensures that all dimensions are treated equally before the algorithm is run. In this work, we only

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

make use of the FastICA algorithm for one unit. The FastICA for one unit finds a direction w such that the projection

$$D^P_{n\times k} = D_{n\times a} w_{a\times 1},$$

maximizes nongaussianity. Nongaussianity is here measured by the approximation of negentropy [HO00].

7.2.1 The Algorithmic Scheme

In this section, we will introduce a new algorithmic scheme based on the principles of dePDDP. The new technique is incorporating information from the ICA model to find optimal directions to project the data. To estimate the ICA model, we utilize the FastICA algorithm, described in the previous section.

A new hierarchical divisive clustering algorithm is presented, namely ICDC. More specifically, the ICDC algorithm utilizes the following criteria:

- (Stopping Criterion) ST_{ICDC}: Let $\Pi = \{\{\mathcal{C}_i, P_i\},\ i = 1,\ldots,k\}$ 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 nongaussionality. 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 $C_i \in \Pi$, $i = 1,\ldots,k$. Stop the procedure when the set $\mathcal{X}$ is empty.

- (Cluster Selection Criterion) CS_{ICDC}: Let $\Pi = \{\{\mathcal{C}_i, P_i\},\ i = 1,\ldots,k\}$ 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 nongaussionality. 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 $C_i \in \Pi$, $i = 1,\ldots,k$. The next set to split is C_j, with $j = \arg\max_i\{f_i : f_i \in \mathcal{F}\}$.

- (Splitting Criterion) SPC_{ICDC}: 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 $P_2 = \{d_i \in \mathcal{D} : p_i > x^\star\}$.

The ICDC implementation is shown in Table 7.1.

The advantage of the proposed method over the PCA based techniques can be illustrated by evaluating the 2-dimensional example of Figure 7.1. At Figure 7.2, we illustrate the projections of the data and their corresponding density for each direction. The top two plots of the figure correspond to the projected data onto the direction of maximum nongaussianity, while the bottom two correspond to the projected data onto the first principal direction for each dataset of Figure 7.1 (left and right), respectively. For the dataset presented in Figure 7.1 (left), there exists a minimizer of the

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Function ICDC ($\mathcal{D}$) {
1. Get u^p the direction of maximum nongaussionality of $\mathcal{D}$
2. Calculate $P = Du^p$ the projection of $\mathcal{D}$ to u^p
3. Set $\Pi = \{\{\mathcal{D}, P\}\}$
4. Do
5. Select an element $\{\mathcal{C}, \mathcal{P}^{\mathcal{C}}\} \in \Pi$ using Selection Criterion CS_{ICDC}
6. Split $\mathcal{C}$ into two sub-sets $\mathcal{C}_1, \mathcal{C}_2$, using Splitting Criterion SPC_{ICDC}
7. Remove $\{\mathcal{C}, P^C\}$ from Π and set $\Pi \rightarrow \Pi \cup \{\{\mathcal{C}_1, P^{\mathcal{C}_1}\}, \{\mathcal{C}_2, P^{\mathcal{C}_2}\}\}$, where $P^{\mathcal{C}_1}, P^{\mathcal{C}_2}$ are the projections of $\mathcal{C}_1, \mathcal{C}_2$ on the direction of maximum nongaussionality u^{p_1}, u^{p_2} of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively
8. While Stopping Criterion ST_{ICDC} is not satisfied
9. Return Π the partition of $\mathcal{D}$ into $|\Pi|$ clusters
}

Table 7.1: *The ICDC algorithm Summary.*

density function in both cases. However, this is not the case for the dataset of Figure 7.1 (right). A minimizer exists only for the density of the projected data onto the direction of maximum nongaussianity. Thus, in this case, the actual clusters cannot be distinguished, when the data are projected onto the first principal component.

7.2.2 Clustering Microarray data

In this Section the presented ICDC algorithm is compare against PCA driven methods on a series of microarray datasets (see Section 4.3.1. The datasets empoyed here are the follwing:

- The CARCINOMA [NASL01] data set that contains 18 tumor samples and 18 normal samples. There exist 7457 gene expression level measurements per sample. The data set is available at: http://microarray.princeton.edu/oncology/carcinoma.html

- The LYMPHOMA dataset [AED+00a] that contains 62 samples of the 3 lymphoid malignancies samples types. The samples are measured over 4026 gene expression levels. This dataset is available at: http://genome-www.stanford.edu

- The LEUKEMIA [DFS02, GST+99] data set that contains 72 samples of 2 types of acute leukemias.There exist 7129 gene expression level measurements per sample. It is available at: http://www.genome.wi.mit.edu/MPR

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

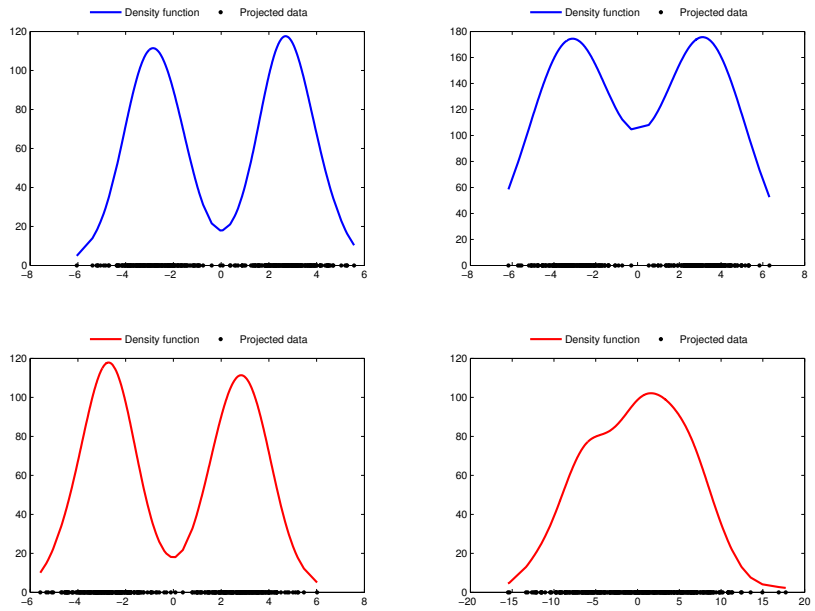


Figure 7.2: *Density estimation of the projected data onto the direction of maximum nongaussianity (Top). Density estimation of the projected data onto the first principal direction (bottom).*

- The Lung Cancer (Dana-Farber Cancer Institute, Harvard Medical School) dataset (LUNG-H) [BRS+01]. A total of 203 snap-frozen lung tumors and normal lung were analysized. The 203 specimens include 139 samples of lung adenocarcinomas, 21 samples of squamous cell lung carcinomas, 20 samples of pulmonary carcinoids, 6 samples of small-cell lung carcinomas and 17 normal lung samples. Each sample is described by 12600 genes. The data set is available at: http://www-genome.wi.mit.edu/mpr/lung/

- The Lung Cancer (University of Michigan) dataset (LUNG-M) [BKH+02] that contains 86 primary lung adenocarcinomas samples and 10 non-neoplastic lung samples. Each sample is described by 7129 genes. It is available at: http://www.camda.duke.edu/CAMDA03/contest.asp

- The DLBCL (Stanford) dataset (DLBCL-S) [AED+00b]. It contains distinct types of diffuse large B-cell lymphoma (DLBCL) using gene expression data. There are 47 samples, 24 of them are from germinal centre B-like group while 23 are activated B-like group. Each sample is described by 4026 genes. It is available at: http://llmpp.nih.gov/lymphoma/

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

In Table 7.2, the clustering results with respect to purity and V-measure of the algorithms for each dataset are illustrated. For ICDC algorithm 100 experiments for each dataset have been made and we present the mean values and standard deviation of purity, V-measure and number of found clusters (standard deviation of zero value is not reported in the table). Although both dePDDP and ICDC algorithms have the ability to automatically predict the actual cluster number, here we will not focus on this attribute of the algorithms; the number of clusters to be found is given as input to the algorithms. However, the termination criterion of the algorithms can still be satisfied at any time. In the case of LUNG-H, LUNG-M, and DLBCL-S datasets, to improve performance, the algorithms are forced to find more clusters than the true number of classes. This is a common procedure for this category of algorithms. The algorithms presented here in comparison with the proposed method are projection based method based on Principal Component Analysis. These are the original PDDP algorithm, the km-PDDP algorithm [ZG07], and the dePDDP. The value of the bandwidth parameter for the density function in the case of dePDDP and ICDC has been set based on the "normal reference rule" (see Section 5.8) using the multiplier value 2. As shown, in most of the cases, the ICDC algorithm has a significant performance advantage and in cases where dePDDP performs better there is only a slight difference compared to ICDC. Thus, the use of the proposed ICDC algorithm is highly recommended in the case of clustering of microarray datasets.

Dataset		LEY	LYM	CAR	LUNG-H	LUNG-M	DLBCL-S
Classes		2	3	2	5	2	2
PDDP	Cl.	2	3	2	10	10	10
	Pur.	0.90	0.96	0.75	0.78	**0.96**	0.74
	V-m.	0.56	0.87	0.19	0.33	**0.25**	0.23
dePDDP	Cl.	2	3	2	7	10	10
	Pur.	0.95	**1.00**	0.75	**0.83**	**0.96**	0.65
	V-m.	0.77	**1.00**	0.19	**0.43**	**0.25**	0.15
KM-PDDP	Cl.	2	3	2	10	10	10
	Pur.	0.96	0.85	0.75	0.78	0.93	0.65
	V-m.	0.77	0.64	0.19	0.33	0.14	0.10
ICDC	Cl.	2	3	2	9.7(0.9)	8.7(2.0)	9.9 (0.3)
	Pur.	**0.98**	**1.00**	**0.97**	0.82 (0.02)	**0.96 (0.01)**	**0.86 (0.05)**
	V-m.	**0.90**	**1.00**	**0.84**	0.37 (0.03)	0.21 (0.05)	**0.29 (0.04)**

Table 7.2: *Mean purity, V-measure, and number of clusters discovered for the microarray datasets.*

7.2.3 Concluding Remarks

In this Section, an effective hierarchical divisive technique, which utilizes the ICA model, is presented. The new clustering algorithm projects the data onto the direction of maximum interestingness and consequently incorporates information from the density of the projected data.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

The experimental results presented are promising, since the ICDC algorithm outperforms other algorithms of the same class in the case of clustering of microarray datasets.

§ 7.3 Density Based Projection Pursuit Clustering

Projection pursuit [FT74] 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 procedure 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 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 [Hub85, JS87]. 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) [JD88]. The interestingness criterion of the PCA is the variance of the projected data.

In this Section a new interestingness criterion is introduced, 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 [SP97].

7.3.1 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.

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

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

$[x, y]$ holds that $x = \delta \cos\theta$ and $y = \delta \sin\theta$ (see Figure 7.3 (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 7.3 (right).

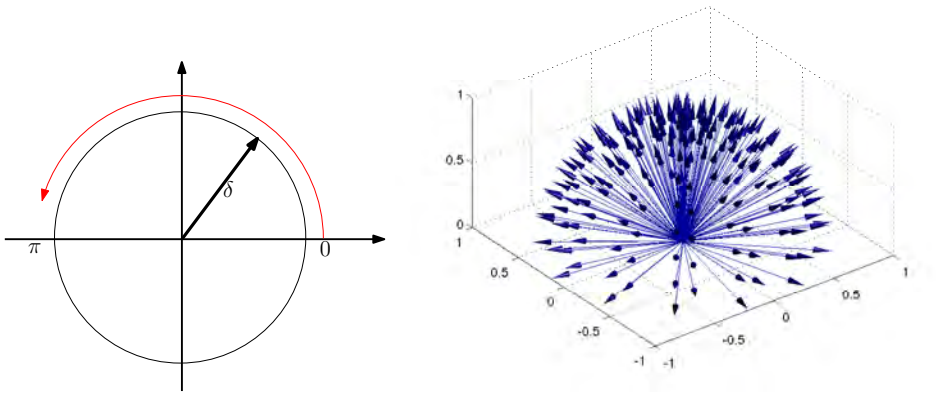


Figure 7.3: *Example of the projection direction space in the two (left) and three (right) dimensional cases.*

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 7.4 (top). The projected data onto the principal component, as well as their corresponding density, are illustrated at Figure 7.4 (bottom left). As expected, based on the de-PDDP 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 7.4 (bottom right). The maximum variance direction is very evident and stable, although the corresponding projection fails to capture the clustering structure.

In [TTP10a] 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 [TTP10a]. 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

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

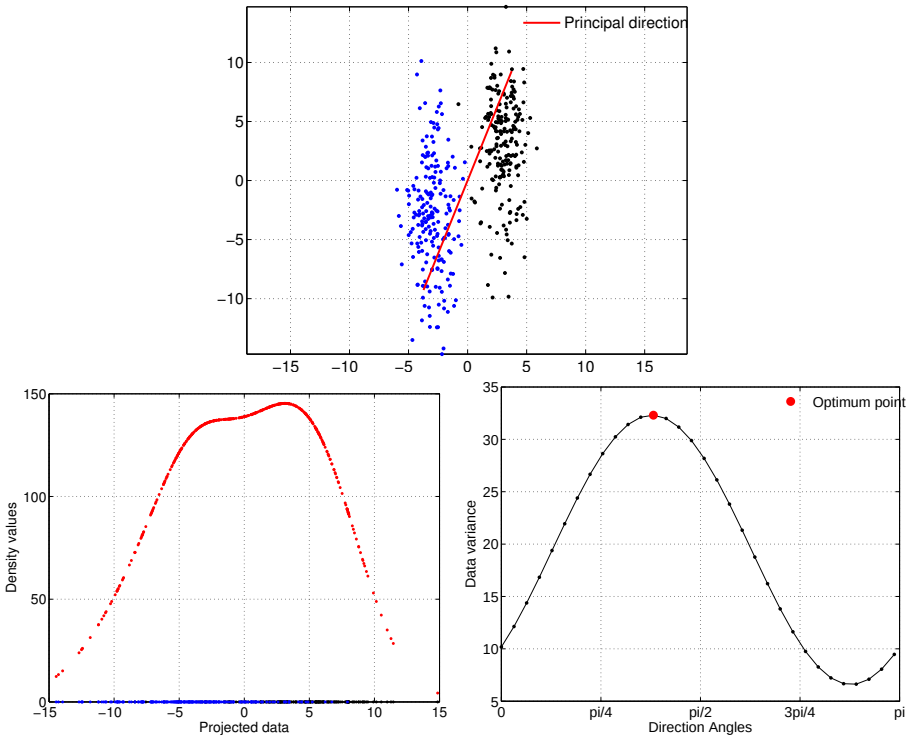


Figure 7.4: *The two–dimensional dataset with the principal direction (top). The projections onto the principal direction along with their densities (bottom left) and the optimization quality criterion (bottom right).*

points. In those points, the density of the data will be very small and the particular direction would be recognised as 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 Section, we present 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 7.3.1. *(**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 7. Let $M1$ and $M2$ be the maximum*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

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 7.5). Using the depth criterion the Projection Pursuit problem can be formally defined as follows.

Definition 7.3.2. *(***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:*

$$QC(u^{opt}) \geqslant QC(u), \quad \forall u \in \mathcal{U}. \tag{7.1}$$

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 top of Figure 7.6, 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 7.6 (bottom left). 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 7.7). Figure 7.7 (top) illustrates the dataset along with the principal and the chosen best depth direction, while their projections and their corresponding densities are demonstrated at the bottom left and bottom right of Figure 7.7, respectively. Finally, Figure 7.8 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

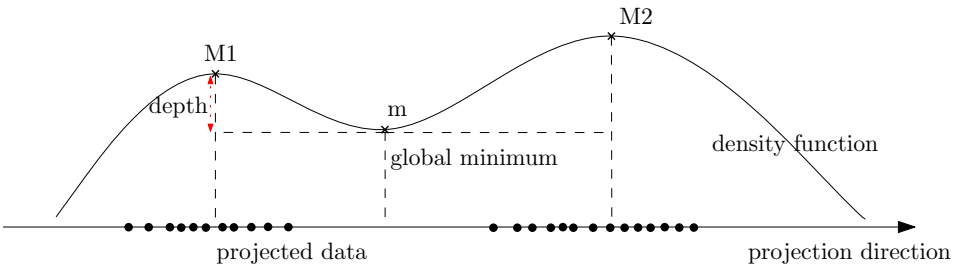


Figure 7.5: *The proposed quality criterion*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

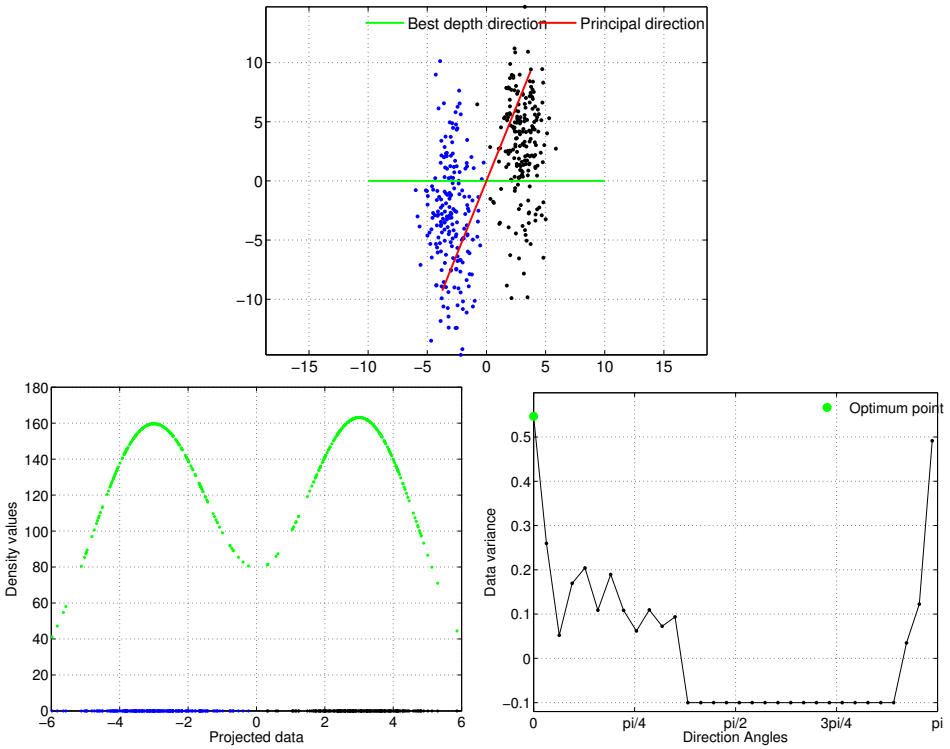


Figure 7.6: *The two–dimensional dataset with the principal and the best depth direction (top). The projections onto best depth direction along with their densities (bottom left) and the depth quality criterion values (bottom right) for each direction.*

non-differentiable and highly multimodal. For this reason the utilization of a global optimization algorithm is essential.

7.3.2 Differential Evolution

We attempt to tackle the aforementioned optimization problem using the Differential Evolution (DE) algorithm [SP97]. 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. Individuals evolve over successive steps to explore promising re-

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

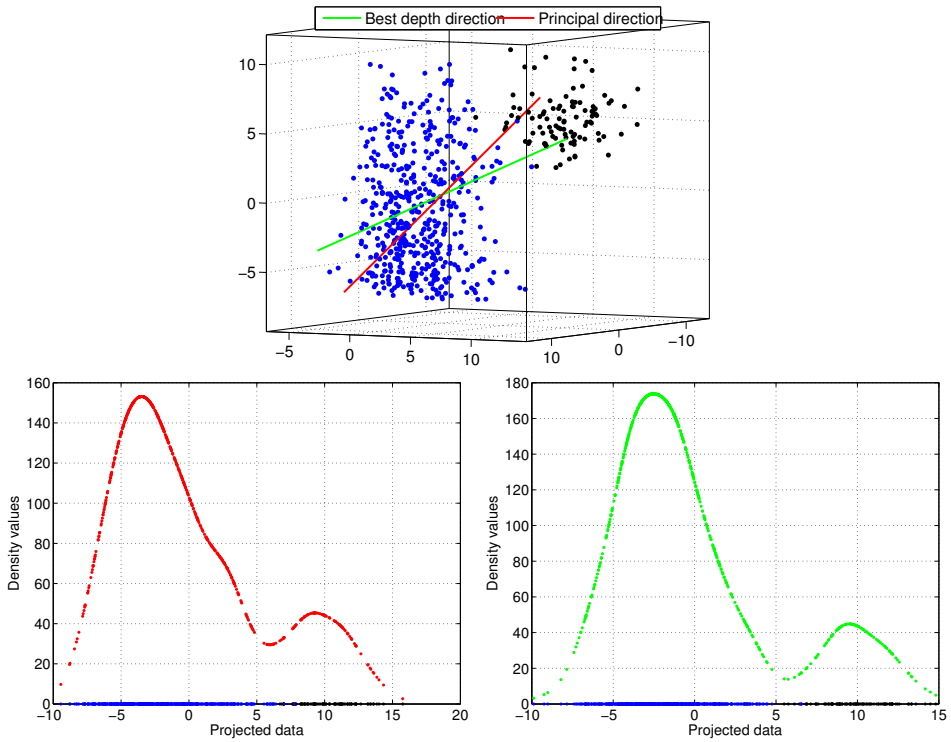


Figure 7.7: *The three–dimensional dataset with the principal and the best depth direction (top). The projections onto the principal direction (bottom left) and the best depth direction (bottom right) along with their densities.*

gions 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 [SP97, PSL05, DS11, ETP+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 [SP97, PSL05, DS11]. 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

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

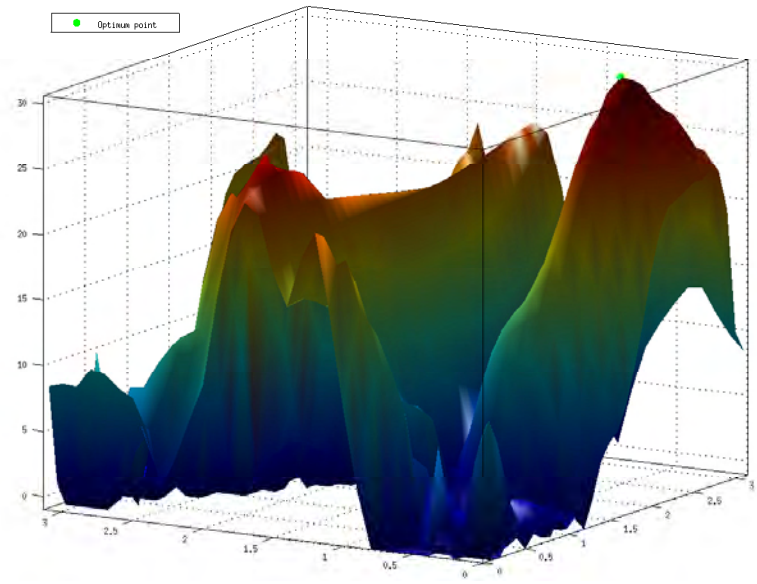


Figure 7.8: *The best depth optimization landscape of the three–dimensional dataset along with its optimum point.*

promising regions of the solution space. A more comprehensive description of the DE can be found in [SP97, PSL05, DS11, ETP+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 [ETP+11] with different characteristics. The most known and widely used mutation strategy acts in accordance with the following equation:

$$v_{g+1}^i = x_g^{r1} + F(x_g^{r2} - x_g^{r3}), \tag{7.2}$$

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 [PSL05].

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 ($l = 1, 2, \ldots, D$) of the mutant individual v_{g+1}^i, we uniformly

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

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 7.3.2, 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.

7.3.3 The Proposed Clustering Algorithm

In this Section, we present a new algorithmic scheme based on the principles of the dePDDP algorithm. The new technique utilizes the depth quality criterion proposed in Section 7.3.1 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_{DBPPC}: Let $\Pi = \{\{\mathcal{C}_i, P_i\},\ i = 1, \ldots, k\}$ 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 minimisers x^*_i of the density estimates $\hat{f}(x^*_i; h)$ of the projection P_i of the data of each $C_i \in \Pi$, $i = 1, \ldots, k$. Stop the procedure when the set $\mathcal{X}$ is empty.

- (Cluster Selection Criterion) CS_{DBPPC}: Let $\Pi = \{\{\mathcal{C}_i, P_i\},\ i = 1, \ldots, k\}$ 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 minimisers x^*_i for the projection P_i of the data of each $C_i \in \Pi$, $i = 1, \ldots, k$. The next set to split is C_j, with $j = \arg\min_i \{f_i : f_i \in \mathcal{F}\}$.

- (Splitting Criterion) SPC_{DBPPC}: 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 minimiser. Then construct $P_1 = \{d_i \in \mathcal{D} : p_i \leqslant x^\star\}$ and $P_2 = \{d_i \in \mathcal{D} : p_i > x^\star\}$.

Based on that criteria Table 7.3 reports the complete algorithmic scheme.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

7.3.4 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. We compare the performance of the proposed clustering algorithm against four well known clustering algorithms, namely dePDDP [TTP10b], k-means [HW79], DBSCAN [EKSX96], GMM. Table 7.4 reports the purity and Table 7.5 the V-measure of the algorithms in 100 randomly generated datasets, using the described $DSET_{\text{beta}}$ 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 [EKSX96] 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 [YDGD03]. As proposed in [TTP10b], 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 $DSET_{\text{beta}}$ mechanism and will resort to visual inspection (Figures 7.9 and 7.10). As shown for the dataset of Figure 7.9 only the DBPPC algorithm manages to retrieve all the actual clusters. Although more than 3 clusters have been retrieved, none of the

Function DBPPC ($\mathcal{D}$) {
1. Set $\Pi = \{\mathcal{D}\}$
2. Do
3. Select a set $\mathcal{C} \in \Pi$, using Selection Criterion CS_{DBPPC}
3. Split $\mathcal{C}$ into two sub-sets $\mathcal{C}_1$ and $\mathcal{C}_2$ using Splitting Crit. SPC_{DBPPC}
4. Remove $\mathcal{C}$ from Π and set $\Pi \rightarrow \Pi \cup \{\mathcal{C}_1, \mathcal{C}_2\}$
5. While Stopping Criterion ST_{DBPPC} is not satisfied
6. Return Π the partition of $\mathcal{D}$ into $|\Pi|$ clusters
}

Table 7.3: *The DBPPC algorithm summary.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

sub-clusters contain elements that belong to more than one actual cluster. In Figure 7.10, 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.

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 7.4: *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	0.91 (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)

Table 7.5: *Mean V-measure for the generated datasets (with the observed standard deviation in parenthesis)*

7.3.5 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 [BM98]; the Breast Canser dataset and the Vertebral dataset, and two microarray datasets, the Leukemia [HKK05] and the Lymphoma [AED+00a] 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.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

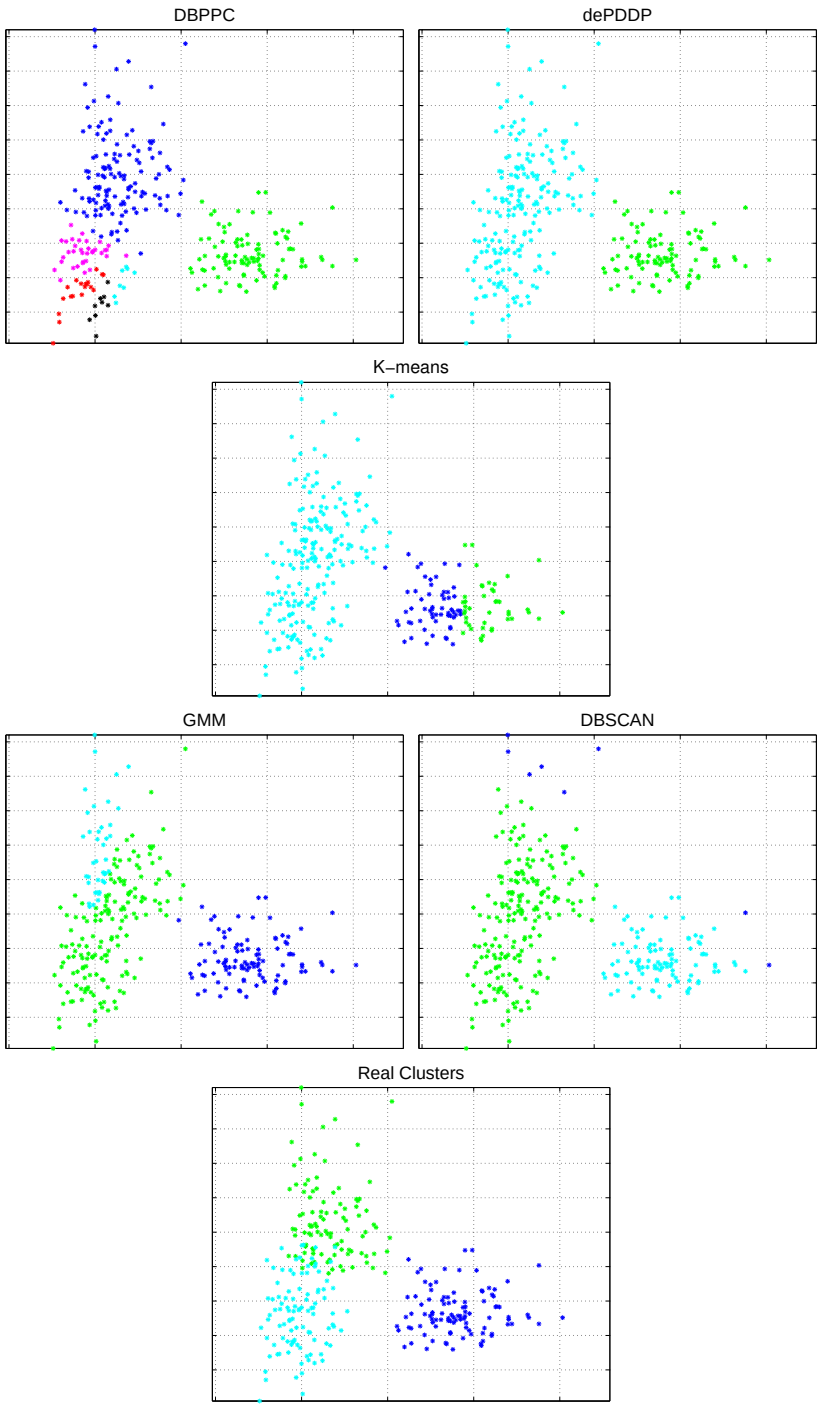


Figure 7.9: *Clustering results for a two dimensional dataset*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

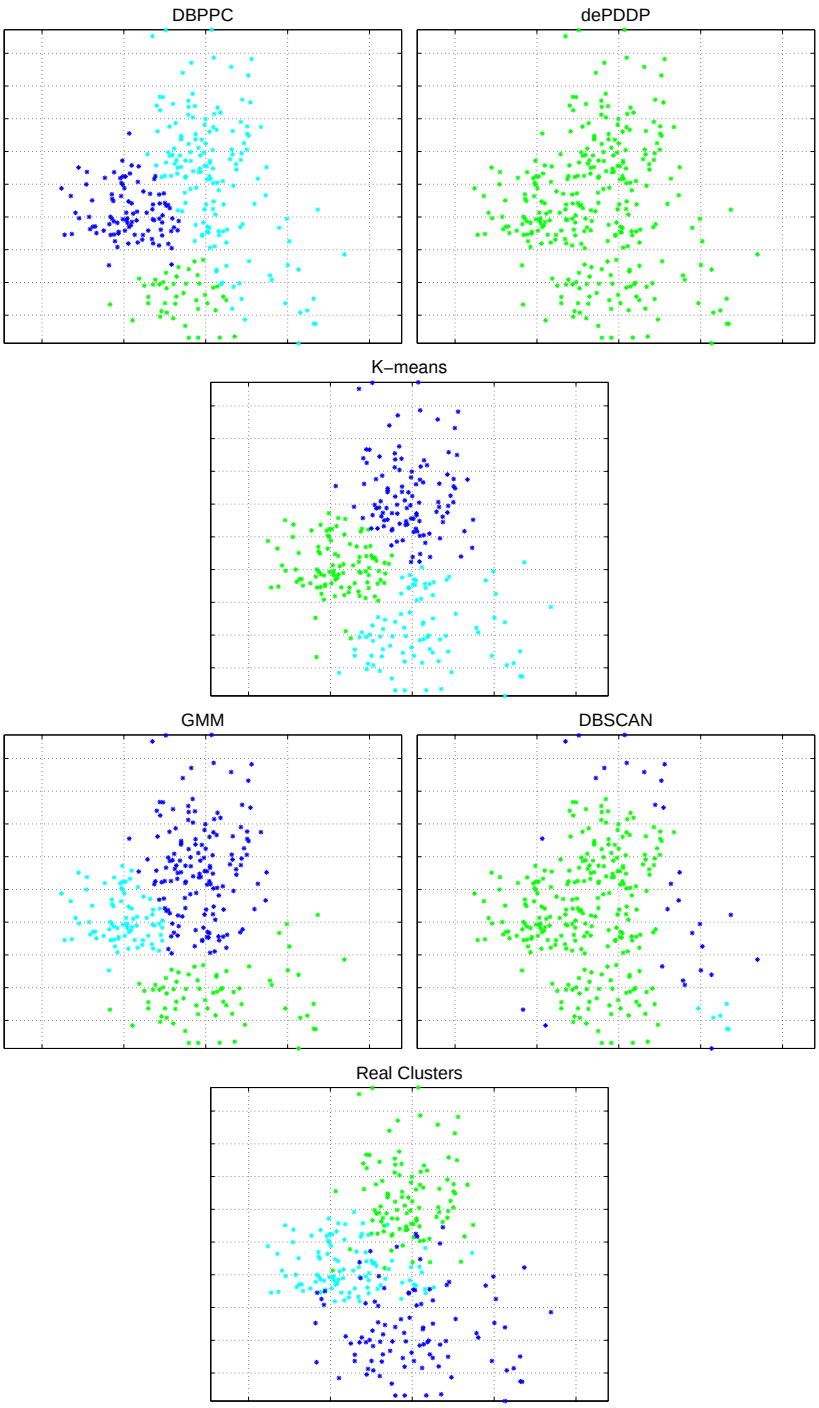


Figure 7.10: *Clustering results for a two dimensional dataset*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

- (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.* [HKK05] 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.* [AED+00a] 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 [DF02].

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 7.6, the DBPPC algorithm's performance remains high in all cases. 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 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.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Dataset		Leukemia	Lymphoma	Breast-Cancer	Vertebral
Classes		3	3	2	3
	Cl.	4	3	3	...
dePDDP	Pur.	0.9737	0.8375	0.9714	...
	V-m.	0.8369	0.5885	0.8051	...
	Cl.	4	3	2	5
k-means	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)
	Cl.	2	2	2	2
DBSCAN	Pur.	0.6053	0.5250	0.7883	0.4839
	V-m.	0.2782	0.1088	0.2614	0.0044
	Cl.	...	...	2	3
GMM	Pur.	...	...	0.8741(0.00)	0.7678(0.00)
	V-m.	...	...	0.5553(0.00)	0.4540(0.03)
	Cl.	4	3	2	5
DBPPC	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)

Table 7.6: *Results with respect to the mean clustering purity and V-measure (with the observed standard deviation in parenthesis)*

§ 7.4 Conclusion

In this Section, we presented a new algorithmic scheme based on the principles of the dePDDP algorithm. The new technique utilizes a new measure of interestingness (quality criterion) of projection directions to guide a projection pursuit method. For finding the best direction over the space of all possible ones, we use the DE optimization algorithm. The performance of the presented method is promising in simulated and real world clustering applications.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

- Chapter 8 -

Clustering of High Dimensional Data Streams

One should always generalize.

—*Carl Gustav Jacobi*

§ 8.1 Introduction

Recent technological advances have made the continues collection of data trivial. A series of everyday activities such as the use of a credit card, a phone call or web browsing require data storage. Usually interesting information can be mined from these large data collections, which is very useful in several applications. When the stored databases are very large, a series of computational and mining challenges arise.

The unlimited rate at which the data grow made impossible the passing of the database multiple times. This introduces a number of constraints on the implementation of the mining algorithms, as such the stream mining algorithms should be designed in such a way that they should not require more than one pass of the data. Another significant problem in mining of data streams is the evolution of the streams over time, a behaviour often referred as temporal locality. In that case, a straightforward adaptation of one pass mining algorithms does not solves the mining task effectively. For this reason the design of the stream mining algorithm must focus on the evolution of the data over time.

A streaming clustering process aims to continuously track the clustering structure of the data. Since stream data by nature imposes a one pass constraint on the design of the algorithms, this task becomes more difficult. In addition, in many cases streaming data are also high dimensional and in result more complex to cluster, due to the effect that high dimensionality has on distance or similarity [SEK03, BGRS99]. Recently, in [TTP10b], we

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

have proposed a density based hierarchical clustering approach (dePDDP) which can deal with high dimensional data, by projecting them onto a lower dimensional subspace.

Most clustering methods cannot be used for streaming data, since they rely on the assumption that the data are available in a permanent memory structure, from which global information can be obtained at any time. However in many cases a clustering algorithm can be extended to the concept of data streams. A k-means clustering model for data streams was proposed in [DHEE01] and more recently, [CEQZ06] developed the DENSTREAM algorithm, by extending the GDBSCAN algorithm.

§ 8.2 The Proposed Clustering Algorithm

In this Section, we extend the dePDDP framework and propose a new method for high dimensional data stream clustering. To extend the dePDDP approach to streaming data we need to update online the hierarchical structure of the algorithm at each data point arrival. Thus the proposed streaming modification will use the standard dePDDP methodology to assign the data entry to an already defined cluster or to create a new one one.

In more detail, at each time instant n for each new data point arrival d_n the proposed algorithm appropriately updates the hierarchical clustering structure recovered up to that time point. Starting from the root node the data points will be first projected on the u_1 Principal Component (PC) of all the points that have been up to that time assigned to that node. Subsequently they will be assigned to a sub-node as described in Section 5.1, based on the density estimate $\hat{f}(x;h)$ of all the points assigned up to that time point to that node.

However, this would imply that all the points assigned to each node need to be kept in memory to calculate both the PC u_1 and $\hat{f}(x;h)$, which is unrealistic in the data stream scenario. However, online methods have been developed that overcome this constraint for online adaptation of both u_1 and $\hat{f}(x;h)$, respectively. Here, for the principal component u_1, we use the candid covariance-free IPCA (CCIPCA) method [WZH03], which is based on the work of Oja and Karhunen [OK85] and Sanger [San89]. The CCIPCA method is described in Section 8.2.1. To calculate the density function over data stream efficiently we employ the method introduced in [ZCWQ03] based on the M-kernels concept. The main characteristic of this methodology is that it only uses a fix-sized main memory, which is irrespective of the total number of data points in the stream, and the time complexity is in linear with the size of the data stream (see Section 8.2.2).

The complete algorithmic scheme of the new SPDC (Streaming Principal Direction Clustering) algorithm is presented at Table 8.1. For each node of

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Function SPDC {
1. For each point arrival d_n
2. Do
3. $ID = RootNode$
3. Update u_1^{ID} and calculate $p_n^{ID} = u_1^{ID} d_n$
4. Update Density $\hat{f}(x;h)^{ID}$ with p_n^{ID}
5. If there is a minimiser $x*$ then
6. If $p_n^{ID} \geqslant x^*$ then $ID = Rkid^{ID}$ else $ID = Lkid^{ID}$
7. go to 4
5. End If
}

Table 8.1: *The SPDC algorithm summary.*

the hierarchical structure the algorithm keeps in memory the node identity ID, the PC u_1^{ID}, the Density Function $\hat{f}(x;h)^{ID}$, and the identity of the left and right kid $Rkid^{ID}$ and $Lkid^{ID}$ respectively. For each point arrival d_n, the PC is updated and d_n is projected onto the updated PC. Then the density function is updated and d is assigned to the left or right sub-node based on the minimizer x^* as explained in Section 5.1. The iteration stops when there is not a minimizer and the algorithms returns the identity ID.

8.2.1 Incremental PCA

Let $d_1, d_2, \ldots$ be the sample vectors that are acquired sequentially. Each $d_n, n = 1, 2, \ldots$, is a a-dimensional vector. Without loss of generality, we can assume that d_n has a zero mean (the mean may be incrementally estimated and subtracted out). Then the nth step estimate u_1^n of u_1 is given by

$$u_1^n = \frac{n-1-l}{n} u_1^{n-1} + \frac{1+l}{n} d_n d_n^T \frac{u_1^{n-1}}{\|u_1^{n-1}\|},$$

where $(n-1/n)$ is the weight for the last estimate and $1/n$ is the weight of the new data. The positive parameter l is called the amnesic parameter. With the presence of l, larger weight is given to new samples and the effect of old samples will fade out gradually. In this work we do not use an amnesic parameter and l is set to 0. Finally, to begin the iteration, we set $u_1^0 = d_1$, the first direction of data spread. A mathematical proof of the convergence of CCIPCA can be founded in [ZW01].

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

8.2.2 Density Estimation over Data Streams

To calculate the density of a point at the time frame of the stream n we need to calculate the sum of n elements as shown at Equation 5.3. To find a minimizer $x*$ we need to calculate the density over n points. As expected, when data comes in the form of data stream, the large volume and the endlessness of the data stream make it computationally impossible to keep them in memory.

To handle data stream efficiently, we can maintain a representative sample of points with appropriately weights in order to accurately approximate $\hat{f}(x;h)$, as described in [ZCWQ03].

Let m be the number of the sample of points that we keep in memory. When a new point arrives at time $m+1$ in order to update the density function based on this point without increasing computational complexity we merge two points based on a merging cost. As merging cost, we use the distance between two consecutive points. Let p_k and p_l be the two points with the smallest merging cost. Then p_k and p_l are substituted by $p_n = (p_k + p_l)/2$. The weight value of the kernel function that corresponds to p_n is the sum of the weight values of the kernel function of p_k and p_l respectively. Then, at time point n, the density estimation can be written as

$$\hat{f}_n^*(x;h) = n^{-1}h^{-1}\sum_{i=1}^{m}\rho_i^* K\left((x-p_i)/h\right).$$

where $\sum_{i=1}^{m}\rho_i^* = n$.

The bandwidth parameter is very important for the quality of the density estimation. Most well known bandwidth strategies [Sco92] often assign a global bandwidth to all kernels. However, these strategies depend on the complete sample, which is not known in the concept of data streams. To overcome this problem, we use an approximate solution that complies with the processing requirements of data streams similar to the one used in [HSia]. We use the "normal reference rule" bandwidth strategy, which is the bandwidth that minimizes the Mean Integrated Squared Error (MISE). For a sample with n samples this is given by $h_{opt}^n = \sigma\left(\frac{4}{3n}\right)^{1/5}$, where σ is the standard deviation of the data. The standard deviation here is computed incrementally in constant time.

§ 8.3 Experimental Analysis

In this Section, we perform an experimental evaluation of the proposed clustering method on streaming data. Firstly, we employ a series of simulated datasets. In particular, we construct datasets by randomly drawing points from a finite mixture of k Gaussian distributions that represent the actual clusters in the data. 5000 points are drawn in total for each dataset. The

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

mean of each Gaussian is randomly placed in $[100, 200]^a$ and the covariance matrix is also randomly generated by an appropriate procedure, so as to ensure that it is symmetric and positive definite.

Tables 8.2 and 8.3 report the clustering performance with respect to the clustering purity, the V-measure and the number of the found clusters for several types of datasets. For each dataset type, 50 experiments have been performed and the mean values with the standard deviation (in the parenthesis) are presented. The clustering performance is always measured at the last 100 points of the stream. The performance of the proposed method is compared against the well known DENSTREAM algorithm [CEQZ06]. For DENSTREAM the ϵ parameter was set to 100. In the case of SPDC (Streaming Principal Direction Clustering) 100 M-kernels where used in all experiments. The bandwidth parameter for the density estimation was set as explained in Section 8.2.2. As shown the SPDC clustering results yield superior Purity and V-measure in most cases.

	Dimension 2			Dimension 10		
No. Of Cl.	Purity	V-measure	Clusters	Purity	V-measure	Clusters
	SPDC			SPDC		
2	1.00 (0.02)	0.68 (0.09)	6.02 (1.59)	0.99 (0.06)	0.81 (0.16)	4.00 (1.43)
5	0.96 (0.07)	0.92 (0.05)	6.90 (1.68)	0.96 (0.07)	0.95 (0.05)	5.64 (1.16)
10	0.72 (0.14)	0.82 (0.09)	8.14 (2.06)	0.82 (0.11)	0.89 (0.06)	10.08 (2.26)
	DENSTREAM			DENSTREAM		
2	0.51 (0.00)	0.00 (0.00)	1.00 (0.00)	0.85 (0.05)	0.70 (0.23)	1.70 (0.23)
5	0.22 (0.00)	0.00 (0.00)	1.00 (0.00)	0.56 (0.07)	0.60 (0.12)	2.70 (1.78)
10	0.12 (0.00)	0.00 (0.00)	1.00 (0.00)	0.23 (0.00)	0.35 (0.02)	2.00 (0.22)

Table 8.2: *Mean purity, V-measure and number of the found clusters for the artificial datasets.*

	Dimension 50			Dimension 100		
No. Of Cl.	Purity	V-measure	Clusters	Purity	V-measure	Clusters
	SPDC			SPDC		
2	1.00 (0.00)	0.90 (0.10)	2.88 (0.82)	1.00 (0.00)	0.92 (0.08)	2.64 (0.62)
5	0.97 (0.07)	0.97 (0.05)	5.30 (0.78)	0.90 (0.13)	0.93 (0.08)	4.84 (0.86)
10	0.85 (0.11)	0.91 (0.06)	9.18 (1.73)	0.78 (0.13)	0.86 (0.10)	8.26 (1.81)
	DENSTREAM			DENSTREAM		
2	1.00 (0.00)	1.00 (0.00)	2.00 (0.00)	1.00 (0.00)	1.00 (0.00)	2.00 (0.00)
5	1.00 (0.00)	1.00 (0.00)	5.00 (0.00)	0.92 (0.01)	0.95 (0.00)	4.60 (0.26)
10	0.17 (0.00)	0.21 (0.05)	1.50 (0.27)	0.21 (0.00)	0.31 (0.04)	1.90 (0.54)

Table 8.3: *Mean purity, V-measure and number of the found clusters for the artificial datasets*

To examine the sensitivity of the number of M-kernels parameter used by SPDC algorithm, we perform a series of experiments for the 50-dimensional 5-cluster case. For each parameter value 50 experiments have been made. A shown at Figure 8.1, a parameter value higher than 10 is enough to achieve high quality results.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

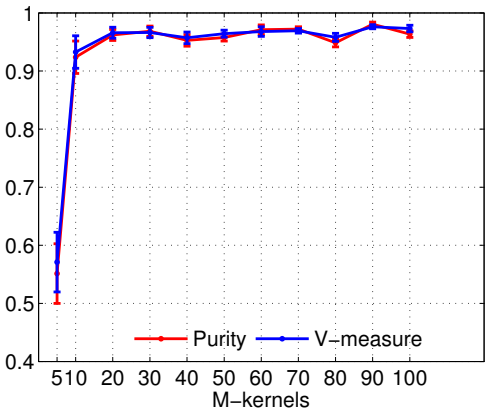


Figure 8.1: *Mean purity and V-measure with respect to the corresponding number of M-kernels*

Finally, we test the efficiency of SPDC and DENSTREAM at the Forest CoverType real world dataset, obtained from the UCI machine learning repository [BM98]. This dataset is comprised of 581012 observations characterized in 54 attributes, where each observation is labelled in one of seven forest cover classes. Here, we only use the 10 numerical attributes. In Figure 8.2, we can see boxplots of the Purity and the V-measure of the clustering result, obtained in the last 100 points for various time point of the data stream for SPDC and DENSTREAM, respectively. The ϵ parameter for DENSTREAM was set to 30 for better results. As shown the SPDC results are superior in both cases. Purity values are always high, but the V-measure obtain lower values since in most cases the algorithms tend to find more clusters than the actual.

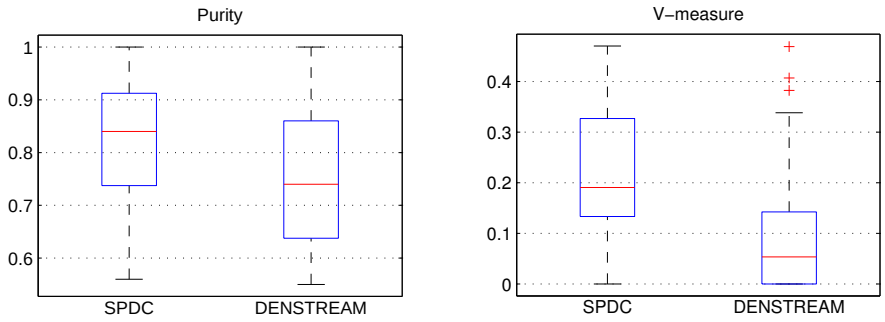


Figure 8.2: *Boxplots of Purity and V-measure for the Forest CoverType real world dataset.*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

§ 8.4 Concluding Remarks

Although many data stream clustering algorithms have been proposed in the literature, very few of them can actually deal with high dimensional data. Here, we present an algorithm that can effectively deal with such high dimensional data streams. The proposed method shows promising results in synthetic and real data scenarios.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Conclusion

The so-called Pythagoreans, who were the first to take up mathematics, not only advanced this subject, but saturated with it, they fancied that the principles of mathematics were the principles of all things.

—*Aristotle*

Clustering is a subjective problem in its nature [GvLW09]. Although it sounds easy to find similar objects in a database, the similarity in itself is only in the eyes of the beholder. Additionally, the introduction of high dimensionality datasets posed new problems to the data analyst. Surprisingly however, data clustering algorithms work and they provide meaningful results for a variety of problems ranging from text mining to microarray data analysis [ZG07, TPV06]. To tackle the challenging new problems, a new class of algorithms has been developed. These algorithms operate on the projection of the data into a lower dimensional subspace in order to be able to effectively discover clusters.

This thesis is devoted to the development of theoretically motivated clustering methods drawing on principles from non-parametric density estimation and recent dimensionality reduction approaches. This approach enable us to develop our understanding of the clustering problem and develop effective methods for real world problems that arise in text mining, face recognition and several biomedical applications.

The second Part of this thesis is devoted to the original work introduced here. More precisely, in Chapter 5, we try to deepen our understanding on what can be achieved by approaches that are based on dimensionality reduction techniques. We first make assumptions on the nature of the true clusters in the data ("inductive bias") and then attempt to theoretically discover the relationship between the true clusters and the distribution of their projection onto the principal components. Based on that, appropriate criteria for the various steps involved in hierarchical divisive clustering are proposed. At a next step, these criteria are combined into new algorithms

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

and their effectiveness is investigated.

The proposed algorithms require minimal user-defined parameters and have the desirable feature of being able to provide approximations for the number of clusters present in a dataset. This in itself is an open problem in cluster analysis and little has been done in the past for the high dimensional data case that we are mostly concerned with here.

The included experimental results indicate that the proposed techniques are effective both in terms of partitioning the data and determining the true number of clusters in the dataset. Their performance is very good, even compared against the popular density based methods. Finally, to stress test the proposed methods a series of experiments on real world microarray and test mining datasets are included. In this case, density based techniques cannot be directly applied and complicated feature selection methods are required to actually get results [TPV06]. However, the proposed algorithms exhibited promising results even in these hard and extremely interesting problems.

Later on, in Chapter 6, those approaches are extended to cases of ultra high data dimensionality. In such cases, even the application of Principal Component Analysis becomes problematic. For this type of problems, Random Projection has been proposed as a computational efficient alternative, with appealing theoretical characteristics. Thus, it is examined how Random Projection can be used into the recently proposed dePDDP clustering framework. Additionally, a theoretically study of the properties of the resulting framework is provided.

In the analysis, it is shown that as the Random Projection method does not significantly alter the distribution of the data on the projected space, the theoretical characteristics of the clustering algorithms remain valid. This result suggests that the resulting clustering framework would suffer minimal performance losses, while gaining all the computational savings of the Random Projection method. Finally, using an experimental analysis of a combination of simulated and real world datasets, it is shown that the resulting algorithms can at least maintain the performance of the original Principal Component Analysis based algorithms in a fraction of computation time.

Subsequently, in Chapter 7, new clustering algorithms are constructed based on the alternative methods for finding suitable projections directions. For this purpose a new measure of interestingness (quality criterion) of projection directions is introduced and for each problem the Differential Evolution algorithm is used to optimize it. In an additional attempt to retrieve the direction of maximum interestingness, the ICA model is utilized and incorporated into a hierarchical divisive clustering technique. The constructed clustering algorithms present promising performance in several simulated and real world clustering applications, since they outperform other algorithms of the same class.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Finally, in Chapter 8, a streaming clustering approach is constructed by extending the high dimensional clustering algorithm presented in Chapter 5. Although many data stream clustering algorithms have been proposed in the literature, very few of them can actually deal with high dimensionality. The algorithm presented here can effectively deal with high dimensional data streams. The proposed method shows promising results in synthetic and real data scenarios.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Index

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

List of Figures

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

List of Tables

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Bibliography

[A. 01] J. Karhunen E. Oja A. Hyvärinen. *Independent Component Analysis*. John Wiley & Sons., 2001. 44

[Ach01] D. Achlioptas. Database-friendly random projections. In *Proceedings of the Twentieth ACM Symposium on Principles of Database Systems, ACM Press*, pages 274–281, 2001. 45, 46, 47

[AED+00a] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, and X. Yu. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000. 108, 120, 123

[AED+00b] Ash A. Alizadeh, Michael B. Eisen, R. Eric Davis, Chi Ma, Izidore S. Lossos, Andreas Rosenwald, Jennifer C. Boldrick, Hajeer Sabet, Truc Tran, Xin Yu, John I. Powell, Liming Yang, Gerald E. Marti, Troy Moore, James Hudson, Lisheng Lu, David B. Lewis, Robert Tibshirani, Gavin Sherlock, Wing C. Chan, Timothy C. Greiner, Dennis D. Weisenburger, James O. Armitage, Roger Warnke, Ronald Levy, Wyndham Wilson, Michael R. Grever, John C. Byrd, David Botstein, Patrick O. Brown, and Louis M. Staudt. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000. 10.1038/35000501. 109

[AGGR98] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, SIGMOD '98, pages 94–105, New York, NY, USA, 1998. ACM. 37

[AGGR05] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Au-

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

tomatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery*, 11(1):5–33, 2005. 62, 67

[AGM+90] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403 – 410, 1990. 34

[AL95] N. Altman and C. Léger. Bandwidth selection for kernel distribution function estimation. *Journal of Statistical Planning and Inference*, 46(2):195–214, 1995. 84

[AM99] R. J. Alcock and Y. Manolopoulos. Time-series similarity queries employing a feature-based approach. In *7 th Hellenic Conference on Informatics, Ioannina, Greece*, 1999. 88

[BCG00] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(7):719–725, 2000. 62, 74

[Bel61] R. Bellman. *Adaptive control processes: A guided tour*. Princeton university press Princeton, NJ, 1961. 33

[Ber06] P. Berkhin. A survey of clustering data mining techniques. In J. Kogan, C. Nicholas, and M. Teboulle, editors, *Grouping Multidimensional Data: Recent Advances in Clustering*, pages 25–72. Springer, Berlin, 2006. 15

[Bez81] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981. 27

[BGMP99] Francesco Bonchi, Fosca Giannotti, Gianni Mainetto, and Dino Pedreschi. Using data mining techniques in fiscal fraud detection. In *Proceedings of the First International Conference on Data Warehousing and Knowledge Discovery*, DaWaK '99, pages 369–376, London, UK, UK, 1999. Springer-Verlag. 34

[BGRS99] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful. In *7th International Conference on Database Theory*, pages 217–235, 1999. 33, 34, 35, 125

[BH67] G. H. Ball and D. J. Hall. A clustering technique for summarizing multivariate data. *Behavioral Science*, 12:153–155, 1967. 61

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

[BKH+02] David G. Beer, Sharon L.R. Kardia, Chiang-Ching Huang, Thomas J. Giordano, Albert M. Levin, David E. Misek, Lin Lin, Guoan Chen, Tarek G. Gharib, Dafydd G. Thomas, Michelle L. Lizyness, Rork Kuick, Satoru Hayasaka, Jeremy M.G. Taylor, Mark D. Iannettoni, Mark B. Orringer, and Samir Hanash. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nature Medicine*, 8(8):816–823, 2002. 109

[BKK96] Stefan Berchtold, Daniel A. Keim, and Hans-Peter Kriegel. The x-tree: An index structure for high-dimensional data. pages 28–39, 1996. 34

[BM98] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. 86, 120, 130

[BM01] E. Bingham and H. Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press*, pages 245–250, 2001. 45

[Bol98] D. Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4):325–344, 1998. 36, 40, 65

[BRS+01] A. Bhattacharjee, W.G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E.J. Mark, E.S. Lander, W. Wong, B.E. Johnson, T.R. Golub, D.J. Sugarbaker, and M. Meyerson. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. volume 98, pages 13790–13795, 2001. 109

[BZD10] Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. Random projections for k-means clustering. *CoRR*, abs/1011.4632, 2010. 46, 96

[CEQZ06] Feng Cao, Martin Ester, Weining Qian, and Aoying Zhou. Density-based clustering over an evolving data stream with noise. In *In 2006 SIAM Conference on Data Mining*, pages 328–339, 2006. 126, 129

[CG93] Gilles Celeux and Gérard Govaert. Gaussian parsimonious clustering models. Rapport de recherche, INRIA, September 1993. 30

[Com94] Pierre Comon. Independent component analysis, a new concept? *Signal Process.*, 36:287–314, April 1994. 44

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

[Cox05] E. Cox. *Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration*. Elsevier Inc., USA, 2005. 27

[CRT06] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006. 46

[CY95] C.G. Chute and Y. Yang. An overview of statistical methods for the classification and retrieval of patient events. *Methods Inf Med*, 34(1-2):104–10, 1995. 36

[Das99] Sanjoy Dasgupta. Learning mixtures of gaussians. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:634, 1999. 45, 96, 101

[Das00] Sanjoy Dasgupta. Experiments with random projection. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, UAI '00, pages 143–151, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. 45, 101

[DDF+90] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. 36

[DF02] Sandrine Dudoit and Jane Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3(7):RESEARCH0036, 2002. 123

[DFS02] Sandrine Dudoit, Jane Fridlyand, and Terence P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002. 108

[DG99] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of the johnson-lindenstrauss lemma. Technical report, 1999. 47

[DH73] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1 edition, June 1973. 11

[DHEE01] Pedro Domingos, Geoff Hulten, Pedrod Cs. Washington Edu, and Cleoff Hulten Ghultencs. Washington. Edu. A general method for scaling up machine learning algorithms and its application to clustering. In *In Proceedings of the Eighteenth International Conference on Machine Learning*, pages 106–113. Morgan Kaufmann, 2001. 126

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

[Dhi01] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM New York, NY, USA, 2001. 40, 41, 59

[DKN03] I. Dhillon, J. Kogan, and C. Nicholas. Feature selection and document clustering. *A Comprehensive Survey of Text Mining*, pages 73–100, 2003. 40, 41, 59

[DM84] W. R. Dillon and Goldstein M. *Multivariate Analysis Methods and Applications*. John Wiley & Sons, 1984. 24

[DM01] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, Jan 2001. 25

[Don06] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. 46

[DS11] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2011. 116, 117

[Dub87] Richard C. Dubes. How many clusters are best? - an experiment. *Pattern Recognition*, 20(6):645 – 663, 1987. 10

[Dub93] Richard C. Dubes. Handbook of pattern recognition & computer vision. chapter Cluster analysis and related issues, pages 3–32. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1993. 15

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96*, pages 226–231, 1996. 119

[ETP+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*, 15(1):99–119, 2011. 116, 117

[FEF+94] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Petkovic, and R. Barber. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994. 34

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

[FHE07] Y. Feng, G. Hamerly, and C. Elkan. PG-means: learning the number of clusters in data. *Advances in Neural Information Processing Systems*, 19:393, 2007. 62, 67, 74

[Fis87] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Mach. Learn.*, 2(2):139–172, 1987. 86

[FT74] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput.*, 23:881–890, September 1974. 111

[GBK01] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001. 101

[GBN] N. Goel, G. Bebis, and A. Nefian. Face recognition experiments with random projection. 45

[GG98] Volker Gaede and Oliver Günther. Multidimensional access methods. *ACM Comput. Surv.*, 30(2):170–231, June 1998. 34

[GNC99] Sanjay Goil, Harsha Nagesh, and Alok Choudhary. Mafia: Efficient and scalable subspace clustering for very large data sets. Technical report, 1999. 37

[GRS98] S. Guha, R. Rastogi, and K. Shim. CURE: An efficient algorithm for clustering large databases. In *Proceedings of ACM-SIGMOD 1998 International Conference on Management of Data*, pages 73–84. Seattle, 1998. 62

[GS91] L. Greengard and J. Strain. The fast gauss transform. *SIAM J. Sci. Stat. Comput.*, 12(1):79–94, 1991. 57

[GST+99] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, and C. D. Bloomfield. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999. 108

[GvLW09] I. Guyon, U. von Luxburg, and R.C. Williamson. Clustering: Science or art? In *NIPS Workshop on Clustering Theory*, 2009. 51, 133

[HAK00] Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim. What is the nearest neighbor in high dimensional spaces? In *Proceedings of the 26th International Conference on Very Large Data Bases*, VLDB '00, pages 506–515, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. 35

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

[Han04] B. E. Hansen. *Bandwidth selection for nonparametric distribution estimation*. PhD thesis, University of Wisconsin, manuscript, 2004. 84

[HGY99] Hongxing He, Warwick Graco, and Xin Yao. Application of genetic algorithm and k-nearest neighbour method in medical fraud detection. In Bob McKay, Xin Yao, CharlesS. Newton, Jong-Hwan Kim, and Takeshi Furuhashi, editors, *Simulated Evolution and Learning*, volume 1585 of *Lecture Notes in Computer Science*, pages 74–81. Springer Berlin Heidelberg, 1999. 34

[HHK98] Alexander Hinneburg, Er Hinneburg, and Daniel A. Keim. An efficient approach to clustering in large multimedia databases with noise. pages 58–65. AAAI Press, 1998. 38

[HK99] A. Hinneburg and D.A. Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *Proceedings of the 25th International Conference on Very Large Data Bases*, pages 506–517, 1999. 39, 62

[HKK05] Julia Handl, Joshua Knowles, and Douglas B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21:3201–3212, August 2005. 120, 123

[HN94] R. Hecht-Nielsen. Context vectors: general purpose approximate meaning representations self-organized from raw data. *Computational Intelligence: Imitating Life, IEEE Press*, pages 43–56, 1994. 46

[HO00] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000. 107

[HSia] Christoph Heinz and Bernhard Seeger. Towards Kernel Density Estimation over Streaming Data. In *International Conference on Management of Data*. Computer Society of India, December COMAD 2006, Delhi, India. 128

[Hua98] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Min. Knowl. Discov.*, 2(3):283–304, 1998. 86

[Hub85] Peter J. Huber. Projection pursuit. *Annals of Statistics*, 13(2):435–475, 1985. 105, 111

[HW79] J.A. Hartigan and M.A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979. 25, 61, 119

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

[Hyv99] Aapo Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10:626–634, 1999. 106

[JD88] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988. 15, 36, 41, 111

[Jia94] *A self-organizing network for hyperellipsoidal clustering (HEC)*, volume 5, 1994. 12

[JL84] William Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemporary Mathematics*, pages 189–206. American Mathematical Society, 1984. 46

[JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, September 1999. 10

[JS87] M. C. Jones and Robin Sibson. What is projection pursuit? *Journal of the Royal Statistical Society. Series A (General)*, 150(1):1–37, 1987. 105, 111

[Kas97] Samuel Kaski. Data exploration using self-organizing maps, 1997. 45

[KHK99] G. Karypis, E.H. Han, and V. Kumar. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer*, 32(8):68–75, 1999. 23, 62

[Kle97] Jon M. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, STOC '97, pages 599–608, New York, NY, USA, 1997. ACM. 96

[Kog07] J. Kogan. *Introduction to Clustering Large and High-Dimensional Data*. 2007. 15

[KOR98] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 614–623, New York, NY, USA, 1998. ACM. 96

[KSF+96] Flip Korn, Nikolaos Sidiropoulos, Christos Faloutsos, Eliot Siegel, and Zenon Protopapas. Fast nearest neighbor search in medical image databases. In *In Proceedings of the Int. Conf. on Very Large Data Bases*, pages 215–226, 1996. 34

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

[KSI03] C. Kruengkrai, V. Sornlertlamvanich, and H. Isahara. Refining A Divisive Partitioning Algorithm for Unsupervised Clustering. *Proceedings of the 3rd International Conference on Hybrid Intelligent Systems*, pages 535–542, 2003. 15, 63

[Lax07] P. D. Lax. *Linear algebra and its applications*. Wiley-Interscience, 2007. 36

[LLTY97] M. Linial, N. Linial, N. Tishby, and G. Yona. Global self-organization of all known protein sequences reveals inherent biological signatures. *Journal of Molecular Biology*, 268(2):539–556, 1997. 46

[MG95] R. Mehrotra and J. Gary. Feature-index-based similar shape retrieval. In *Proceedings of the third IFIP WG2.6 working conference on Visual database systems 3 (VDB-3)*, pages 46–65, London, UK, UK, 1995. Chapman & Hall, Ltd. 34

[MK08] Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2 edition, March 2008. 31

[MP00] G. J. McLachlan and D. Peel. *Finite mixture models*. Wiley Series in Probability and Statistics, New York, 2000. 25

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge University Press, New York, NY, USA, 1995. 24

[MR10] Oded Maimon and Lior Rokach. Introduction to knowledge discovery and data mining. In *Data Mining and Knowledge Discovery Handbook*, pages 1–15. 2010. 5

[MS83] R. S. Michalski and R. E. Stepp. Automated construction of classifications: Conceptual clustering versus numerical taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(4):396 – 409, 1983. 86

[NASL01] D. A. Notterman, U. Alon, A. J. Sierk, and A. J. Levine. Transcriptional gene expression profiles of colorectal adenoma, adenocarcinoma, and normal tissue examined by oligonucleotide arrays. *Cancer Research*, 61:3124–3130, 2001. 108

[Nil02] M. Nilsson. Hierarchical Clustering Using Non-Greedy Principal Direction Divisive Partitioning. *Information Retrieval*, 5(4):311–321, 2002. 15, 40, 41, 52, 59

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

[Oak98] M. P. Oakes. *Statistics for Corpus Linguistics*. Edinburgh University Press, 1998. 16

[OK85] Erkki Oja and Juha Karhunen. On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix. *Journal of Mathematical Analysis and Applications*, 106:69–84, 1985. 126

[PM00] Dau Pelleg and Andrew Moore. X-means: Extending k-means with efficient estimation of the number of clusters. *IN PROCEEDINGS OF THE 17TH INTERNATIONAL CONF. ON MACHINE LEARNING*, pages 727—-734, 2000. 62, 74

[PRTV98] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. *Proc. 17th ACM Symp. on the Principles of Database Systems*, pages 159–168, 1998. 45, 46

[PSL05] Kenneth Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. 116, 117

[RH07] 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)*, pages 410–420, 2007. 16

[RYDG05] V. C. Raykar, C. Yang, R. Duraiswami, and N. Gumerov. Fast computation of sums of gaussians in high dimensions. Technical Report CS-TR-4767, Department of Computer Science, University of Maryland, CollegePark, 2005. 58

[Sal89] Gerard Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989. 34

[San89] Terence D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2(6):459–473, 1989. 126

[Sar93] P. Sarda. Smoothing parameter selection for smooth distribution functions. *Journal of statistical planning and inference*, 35(1):65–75, 1993. 84

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

[Sco92] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization (Wiley Series in Probability and Statistics)*. Wiley, September 1992. 128

[SEK03] M. Steinbach, L. Ertöz, and V. Kumar. The challenges of clustering high dimensional data. *New Vistas in Statistical Physics: Applications in Econophysics, Bioinformatics, and Pattern Recognition*, 2003. 33, 34, 53, 125

[SEKX98] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1998. 28, 62, 63, 66, 74

[SH94] F.S. Samaria and A.C. Harter. Parameterisation of a stochastic model for human face identification, 1994. 101

[SK01] Arora Sanjeev and Ravi Kannan. Learning mixtures of arbitrary gaussians. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, STOC '01, pages 247–257, New York, NY, USA, 2001. ACM. 96

[SKK00] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques, 2000. In KDD Workshop on Text Mining. 19

[SP97] R. Storn and K. Price. Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997. 111, 115, 116, 117

[TK06] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 2006. 59, 63

[TLL+10] Wen-Kwang Tsao, Anthony J.T. Lee, Ying-Ho Liu, Ting-Wei Chang, and Hsiu-Hui Lin. A data mining approach to face detection. *Pattern Recognition*, 43(3):1039 – 1049, 2010. 47

[TPV06] D.K. Tasoulis, V.P. Plagianakos, and M.N. Vrahatis. Unsupervised clustering in mRNA expression profiles. *Computers in Biology and Medicine*, 36:1126–1142, 2006. 91, 133, 134

[Try39] C. Tryon. *Cluster Analysis*. Ann Arbor, MI: Edward Brothers, 1939. 7

[TT08] S.K. Tasoulis and D.K. Tasoulis. Improving principal direction divisive clustering. In *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD*

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

2008), Workshop on Data Mining using Matrices and Tensors, Las Vegas, USA, 2008. 40, 55, 61, 63

[TTP10a] S. K. Tasoulis, D. K. Tasoulis, and V. P. Plagianakos. Evolutionary principal direction divisive partitioning. In *IEEE World Congress on Computational Intelligence*, 2010. 112

[TTP10b] S.K. Tasoulis, D.K. Tasoulis, and V.P. Plagianakos. Enhancing Principal Direction Divisive Clustering. *Pattern Recognition*, 43:3391–3411, 2010. 98, 105, 119, 125

[Tur93] B. A. Turlach. Bandwidth selection in kernel density estimation: A review. *CORE and Institut de Statistique*, pages 23–493, 1993. 59, 81, 84

[TV07] D. K. Tasoulis and M. N. Vrahatis. Generalizing the k-Windows clustering algorithm in metric spaces. *Mathematical and Computer Modelling*, 46(1-2):268–277, 2007. 86

[Vem10] Santosh S. Vempala. A random-sampling-based algorithm for learning intersections of halfspaces. *J. ACM*, 57:32:1–32:14, November 2010. 96

[WJ95] M. P. Wand and M. C. Jones. *Kernel smoothing*. Chapman & Hall/CRC, 1995. 38, 57, 59

[WM97] D. Randall Wilson and Tony R. Martinez. Improved heterogeneous distance functions. *J. Artif. Int. Res.*, 6(1):1–34, January 1997. 10

[WSB98] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 194–205, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. 34

[WZH03] J. Weng, Y. Zhang, and W. Hwang. Candid covariance-free incremental principal component analysis, 2003. 126

[YDGD03] 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*, pages 664–671, 2003. 57, 86, 97, 119

[You82] D.L. Young. The linear nearest neighbour statistic. *Biometrica*, 69(2):477–480, 1982. 63

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

[ZCWQ03] Aoying Zhou, Zhiyuan Cai, Li Wei, and Weining Qian. M-kernel merging: Towards density estimation over data streams. *Database Systems for Advanced Applications, International Conference on*, 0:285, 2003. 126, 128

[ZG03] D. Zeimpekis and E. Gallopoulos. PDDP(l): Towards a Flexing Principal Direction Divisive Partitioning Clustering Algorithms. In D. Boley, I. Dhillon, J. Ghosh, and J. Kogan, editors, *Proc. IEEE ICDM '03 Workshop on Clustering Large Data Sets*, pages 26–35, Melbourne, Florida, 2003. 40, 41, 59

[ZG07] D. Zeimpekis and E. Gallopoulos. Principal direction divisive partitioning with kernels and k-means steering. In *Survey of Text Mining II: Clustering, Classification, and Retrieval*, pages 45–64. 2007. 41, 52, 59, 65, 96, 110, 133

[ZRL96] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases. pages 103–114, 1996. 23, 62

[ZW01] Yilu Zhang and Juyang Weng. Convergence analysis of complementary candid incremental principal component analysis. 2001. 127

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

List of Publications

Publications in International Journal

[1] S. Tasoulis, D. Tasoulis, and V. Plagianakos, "Enhancing principal direction divisive clustering," *Pattern Recognition*, vol. 43, no. 10, pp. 3391–3411, 2010.

[2] S. Tasoulis, D. Tasoulis, and V. Plagianakos, "Random direction divisive clustering," *Pattern Recognition Letters*, vol. 34, no. 2, pp. 131 – 139, 2013.

[3] S. Tasoulis, C. Doukas, V. Plagianakos, and I. Maglogiannis, "Statistical data mining of streaming motion data for activity and fall recognition in assistive environments," *Neurocomputing*, vol. 107, no. 0, pp. 87–96, 2013.

[4] S. Tasoulis, I. Maglogiannis, and V. Plagianakos, "Fractal analysis and fuzzy c-means clustering for quantification of fibrotic microscopy images," *Artificial Intelligence Review (AIRE)*, pp. –, 2013 accepted.

Publications in International Conference Proccedings

[1] S. Tasoulis and D. Tasoulis, "Improving principal direction divisive clustering," *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008), Workshop on Data Mining using Matrices and Tensors, Las Vegas, USA*, 2008.

[2] S. Tasoulis, V. Plagianakos, and D. Tasoulis, "Projection based clustering at gene expression data," in *Sixth International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics (CIBB 2009)*, Genova, Italy, 2009.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

[3] S. Tasoulis, V. Plagianakos, and D. Tasoulis, "Independent component divisive clustering of gene expression data," in *Eighth International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics (CIBB 2011)*, Gargnano-Lago di Garda, Italy, 2011.

[4] S. Tasoulis, V. Plagianakos, and D. Tasoulis, "Projection based clustering of gene expression data," *Computational Intelligence Methods for Bioinformatics and Biostatistics*. Springer, 2010, pp. 228–239.

[5] S. Tasoulis, C. Doukas, I. Maglogiannis, and V. Plagianakos, "Classification of dermatological images using advanced clustering techniques," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 6721–6724.

[6] S. Tasoulis, C. Doukas, I. Maglogiannis, and V. Plagianakos, "Classification of apoptosis using advanced clustering techniques on digital microscopic images," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 5565–5568.

[7] S. Tasoulis, C. Doukas, I. Maglogiannis, and V. Plagianakos, "Skin lesions characterisation utilising clustering algorithms," in *Artificial Intelligence: Theories, Models and Applications*. Springer, 2010, pp. 243–253.

[8] S. Tasoulis, D. Tasoulis, and V. Plagianakos, "Evolutionary principal direction divisive partitioning," in *IEEE Congress on Evolutionary Computation (CEC)* . IEEE, 2010, pp. 1–7.

[9] S. Tasoulis, D. Tasoulis, and V. Plagianakos, "Clustering of high dimensional data streams," in *Artificial Intelligence: Theories and Applications*. Springer, 2012, pp. 223–230.

[10] S. Tasoulis, C. Doukas, I. Maglogiannis, and V. Plagianakos, "Statistical data mining of streaming motion data for fall detection in assistive environments," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 3720–3723.

[11] S. Tasoulis, C. Doukas, I. Maglogiannis, and V. Plagianakos, "Independent component clustering for skin lesions characterization," in *Artificial Intelligence Applications and Innovations*. Springer, 2011, pp. 472–482.

[12] S. Tasoulis, M. Epitropakis, V. Plagianakos, and D. Tasoulis, "Density based projection pursuit clustering," in *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2012, pp. 1–8.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138

[13] S. Tasoulis, I. Maglogiannis, and V. Plagianakos, "Unsupervised detection of fibrosis in microscopy images using fractals and fuzzy c-means clustering," in *Artificial Intelligence Applications and Innovations*. Springer Boston, 2012, pp. 385–394.

Institutional Repository - Library & Information Centre - University of Thessaly
23/09/2024 12:41:28 EEST - 3.144.112.138