Evaluating Clustering in Subspace Projections of High Dimensional Data

Emmanuel Müller • Stephan Günnemann • Ira Assent ° Thomas Seidl •

•Data Management and Data Exploration Group
RWTH Aachen University, Germany
{mueller, guennemann, seidl}@cs.rwth-aachen.de

°Department of Computer Science
Aalborg University, Denmark
ira@cs.aau.dk

ABSTRACT

Clustering high dimensional data is an emerging research field. *Subspace clustering* or *projected clustering* group similar objects in subspaces, i.e. projections, of the full space. In the past decade, several clustering paradigms have been developed in parallel, without thorough evaluation and comparison between these paradigms on a common basis.

Conclusive evaluation and comparison is challenged by three major issues. First, there is no ground truth that describes the "true" clusters in real world data. Second, a large variety of evaluation measures have been used that reflect different aspects of the clustering result. Finally, in typical publications authors have limited their analysis to their favored paradigm only, while paying other paradigms little or no attention.

In this paper, we take a systematic approach to evaluate the major paradigms in a common framework. We study representative clustering algorithms to characterize the different aspects of each paradigm and give a detailed comparison of their properties. We provide a benchmark set of results on a large variety of real world and synthetic data sets. Using different evaluation measures, we broaden the scope of the experimental analysis and create a common baseline for future developments and comparable evaluations in the field. For repeatability, all implementations, data sets and evaluation measures are available on our website[1].

1. INTRODUCTION

Knowledge discovery in databases provides database owners with new information about patterns in their data. Clustering is a traditional data mining task for automatic grouping of objects [14]. Cluster detection is based on similarity between objects, typically measured with respect to distance functions. In high dimensional spaces, effects attributed to the "curse of dimensionality" are known to break traditional clustering algorithms [9]. Meaningful clusters cannot be detected as distances are increasingly similar for growing dimensionality. To detect patterns obscured by irrelevant dimensions, global dimensionality reduction techniques such as principle components analysis (PCA) are not sufficient [16]. By definition, they reduce the original high dimensional space to a single lower dimensional projection for all objects alike. In high dimensional spaces, however, dimensions might have locally varying relevance for different groups of objects. These cannot be detected by a global analysis of relevance. Recent research has introduced clustering in subspace projections, aiming at detecting locally relevant dimensions per cluster.

In several application scenarios like sensor networks, customer profiling, and bioinformatics high dimensional data is measured. Exemplary we highlight the requirement for cluster detection in gene expression analysis [11], as research on clustering in high dimensional data started with this application domain. High throughput experiments of gene expressions were available and opened questions like '*which genes have common functions and should be grouped*'. Databases consist of genes (objects) described by expression levels in different experimental conditions (attributes). High dimensional data occur as there are very many different experimental conditions to be analyzed. In general the problem can be abstracted to a huge number of objects with various attributes as depicted in Figure 1. Possible clusters in subspace projections are highlighted in gray. In many recent applications like sensor networks, objects are also described by very many attributes. As collecting and storing data is cheap, users tend to record everything without considering the relevance for their task. Clustering of such high dimensional data has become a general challenge for a broader range of data.

Recent research for clustering in high dimensional spaces has introduced a number of different approaches. They were named by the pioneers in this field *subspace clustering* [3] or *projected clustering* [1]. Both terms were used in parallel for development of further approaches. Their common goal

[1]http://dme.rwth-aachen.de/OpenSubspace/evaluation

VLDB '09, August 24-28, 2009, Lyon, France

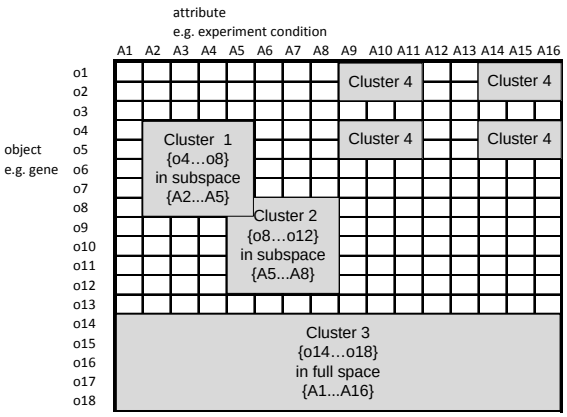


Figure 1: Example for subspace clustering

is to detect the most relevant subspace projections for any object in the database. Any cluster is then associated with a set of relevant dimensions in which this pattern has been discovered. These techniques have been successfully applied in a number of scenarios. We illustrate possible clusters with relevant subspace projections for an example database in Figure 1. Cluster 3 represents a traditional cluster in full space, while clusters 1, 2 and 4 appear only in a subset of relevant dimensions. Please note, that for both objects and dimensions an arbitrary subset can become a subspace cluster. Projected clustering algorithms are restricted to disjoint sets of objects, while subspace clustering algorithms might report several clusters for the same object in different subspace projections. Motivated by the gene expression analysis, a gene may have several function represented by clusters with different relevant attributes (cf. object 8 in Fig. 1). For simplicity of presentation, we choose *subspace clustering* as the preferred term in this publication.

For evaluation and comparison, of subspace clustering algorithms in general, the last decade has seen several paradigms, characterized by their underlying cluster models and their parametrization of the resulting clustering. In this young field, however, we lack a common ground for evaluation as a whole. Three major problems persist. First, there is no ground truth that describes the "true" clusters in real world data. Second, a large variety of evaluation measures have been used that reflect different aspects of the clustering result. Finally, in typical publications authors have limited their analysis to their favored paradigm only, while paying other paradigms little or no attention. This implies several problems for the advancement of the field. Properties of the different paradigms are not yet well understood, as cross-comparisons are not available. The same is true for evaluation measures. They reflect different aspects which are not yet fully understood. It is therefore not possible to compare the results that have been reported in different papers. As a consequence, there is no common basis for research in the area which implies misleading conclusions from reported experiments, and hence possibly wrong assumptions about the underlying properties of algorithms.

In this paper, we provide a systematic and thorough evaluation of subspace clustering paradigms. Results are analyzed using the measures that have been proposed by researchers in recent papers. We use a large collection of data sets, synthetic data with known hidden clusters and also publicly available real world data sets. Our work provides a meaningful characterization of the different paradigms and how these are reflected in the evaluation measures. We create a common ground on which future research in the field can build. Our analysis uses our own open source framework, which we recently presented to the community [7, 25, 23]. This framework extends the popular open source WEKA platform that has been successful for full space data mining algorithms [31]. Our work is therefore easily repeatable and extensible for future algorithms. Implementations, data sets and measures are all available for anyone interested in comparing their own algorithms or the ones evaluated in this paper.

This paper is structured as follows: in Section 2 we review existing subspace clustering paradigms and point out their characteristics. Section 3 introduces the different measures used to evaluate existing algorithms in the literature. Our experimental evaluation in Section 4 gives a detailed analysis of the different paradigms under these measures. Finally, we conclude with discussion of our findings and some pointers for future work in Section 5.

2. CLUSTERING IN SUBSPACE PROJECTIONS

Clustering in subspace projections aims at detecting groups of similar objects and a set of relevant dimensions for each object group. While there are two different names in the literature, *subspace clustering* [3] and *projective clustering* [1], we identify three major paradigms characterized by the underlying cluster definition and parametrization of the resulting clustering:

First, cell-based approaches search for sets of fixed or variable grid cells containing more than a certain threshold many objects. Subspaces are considered restrictions of a cell in a subset of the dimensions, while in the residual dimensions the cell spans the whole attribute domain. Cell-based approaches rely on counting objects in cells and with their discretization of data are similar to frequent itemset mining approaches.

Second, the density-based clustering paradigm defines clusters as dense regions separated by sparse regions. As density computation is based on the distances between objects, in subspace clustering one computes distances by taking only the relevant dimensions into account. Density-based approaches are thus dependent on the distance definition. They can be parametrized by specifying which objects should be grouped together due to their similarities / distances.

Finally, clustering-oriented approaches do not give a cluster definition like the previous paradigms. In contrast, they define properties of the entire set of clusters, like the number of clusters, their average dimensionality or more statistically oriented properties. As they do not rely on counting or density they are more flexible in handling different types of data distributions. However, they do not allow parametrization of each cluster.

For each of the three main paradigms we evaluate seminal approaches which defined these paradigms and evaluate also the most recent representatives. It is clearly not possible to include in this study all algorithms from all of these paradigms. Also, it is beyond the scope to this work to include more specialized algorithms like correlation clustering [2] which transform the data space based on detected cor-

paradigm	approach	properties
cell-based	CLIQUE [3]	fixed threshold, fixed grid, pruning by monotonicity property
cell-based	DOC [29]	fixed result size, fixed threshold, variable hypercubes, randomized, partitioning
cell-based	MINECLUS [32]	enhances DOC by FP-tree structure [15] resulting in more efficient mining
cell-based	SCHISM [30]	enhances CLIQUE by variable threshold, using heuristics for approximative pruning
density-based	SUBCLU [17]	fixed density threshold, pruning by monotonicity property
density-based	FIRES [19]	variable density threshold, using 1d histograms for approximative pruning
density-based	INSCY [6]	variable density threshold, reducing result size by redundancy elimination
clustering-oriented	PROCLUS [1]	fixed result size, iteratively improving result like k-means [20], partitioning
clustering-oriented	P3C [22]	statistical tests, using EM [12] clustering, pruning by monotonicity property
clustering-oriented	STATPC [21]	statistical tests, reducing result size by redundancy elimination, approximative

Table 1: Characteristics of three major paradigms

relations, or application dependent approaches popular e.g. in bioinformatics [11]. Furthermore, we consider subspace clustering only on continuous valued attributes. Subspace clustering of categorical attributes is a specialization of the frequent itemset mining task, and heterogeneous data is just a very recently upcoming topic in subspace clustering [26].

We consider an abstract high dimensional database with objects described by various attributes. As exemplified in Figure 1, a subspace projection is an arbitrary subset of attributes. Each cluster is described by a subset of objects (rows) and a subset of attributes (columns). Please note that in some approaches clusters in subspace projections may overlap in both objects and dimensions, as similarity between the objects is only evaluated with respect to the relevant dimensions.

To study different paradigms we use various evaluation measures that are described in more details in Section 3. Both efficiency in terms of runtime and also clustering quality in terms of different measures are analyzed. For our review on different subspace clustering paradigms we thus highlight both the effect of the cluster model on the quality, as well as the effect of algorithmic properties on the runtime. An overview of all paradigms, the used approaches and a summary of their important properties is given in Table 1.

Notations. For consistent notations in the following sections we abstract from the individual definitions in the literature. Every cluster C in a subspace projection is defined by a set of objects O that is a subset of the database DB and a set of relevant dimensions S out of the set of all dimensions D.

Definition 1. *A cluster C in a subspace projection S is*

$$C = (O, S) \text{ with } O \subseteq DB, S \subseteq D$$

A clustering result is a set of found clusters in the respective subspace projections.

Definition 2. *A clustering result R of k clusters is a set of clusters*

$$R = \{C_1, \ldots, C_k\}, C_i = (O_i, S_i) \text{ for } i = 1 \ldots k$$

We define several basic objective functions to describe the clustering result. The number of detected clusters is given by $numClusters(R) = k$. The average dimensionality of the clusters in the result is $avgDim(R) = \frac{1}{k} \cdot \sum_{i=1}^{k} |S_i|$. For ease of presentation and w.l.o.g. we assume each dimension has the same domain, thus, $domain(DB) = [0..v]^{|D|}$.

2.1 Cell-Based Approaches

First, we consider the cluster definition and its parameterization. Cell-based clustering is based on a cell approximation of the data space. Cells of width w are used to describe clusters. For all cell-based approaches, a cluster result R consists of a set of cells; each of them containing more than a threshold τ many objects ($|O_i| \geq \tau$ for $i = 1 \ldots k$). These cells describe the objects of the clusters either by a hypercube of variable width w [29, 32] or by a fixed grid of ξ cells per dimension [3, 30]. Fixed grids can be seen as discretization of the data space in pre-processing. In contrast, variable hypercubes are arbitrarily positioned to delimit a region with many objects.

Definition 3. ***Cell-Based Subspace Cluster.***

A cell-based subspace cluster (O, S) is defined w.r.t. minimum number of objects τ in cells CS of w width specified by intervals I_i per dimension $\forall i \in S$. Each interval is part of the common domain $I_i = [l_i \ldots u_i] \subseteq [0 \ldots v]$ with lower and upper bound l_i and u_i. For all non-relevant dimensions $\forall j \in D \setminus S$ the interval is the full domain $I_j = [0 \ldots v]$ i.e. the cell is not restricted in these dimensions. The clustered objects $O = \{o \mid o \in DB \cap CS\}$ fulfill $|O| \geq \tau$

The first approach for cell-based clustering was introduced by CLIQUE [3]. CLIQUE defines a cluster as a connection of grid cells with each more than τ many objects. Grid cells are defined by a fixed grid splitting each dimension in ξ equal width cells. Arbitrary dimensional cells are formed by simple intersection of the 1d cells. First enhancements of CLIQUE adapted the grid to a variable width of cells [27]. More recent approaches like DOC use flexible hypercubes of width w [29]. In MINECLUS such hypercube approaches are supported by FP-trees, known from fequent itemset mining to achieve better runtimes [15, 32]. SCHISM, improves the cluster definition by variable thresholds $\tau(|S|)$ adapting to the subspace dimensionality $|S|$ [30].

Second, we consider efficiency. As subspace clustering searches for clusters in arbitrary subspaces, naive search is exponential in the number of dimensions. CLIQUE proposes a pruning criterion for efficient subspace clustering based on a monotonicity property. A similar monotonicity property was introduced in the apriori algorithm [4] for efficient frequent itemset mining and has been adapted to subspace clustering [3]. Monotonicity is used by most subspace clustering algorithms, and states that each subspace cluster (O, S) appears in each lower dimensional projection

T, i.e. $\forall T \subset S : (O,T)$ is also a subspace cluster. The negation of this monotonicity can then be used for pruning in a bottom-up algorithm on the subspace lattice: If a set of objects O does not form a cluster in subspace T then all higher dimensional projects $S \supset T$ do not form a cluster either.

It is important to highlight two major characteristics of this pioneer work in clustering subspace projections: First, a monotonicity property is the most common way in pruning subspaces. It has been applied also in other paradigms for efficient computations of subspace clusters. And second, the cell-based processing of the data space has been used in several other techniques to efficiently compute regions with at least a minimal amount of objects.

Although there are some differences, cell-based approaches share a main common property. They all count the number of objects inside a cell to determine if this cell is part of a subspace cluster or not. This counting of objects is comparable to frequency counting in frequent itemset mining. Subspace clusters are sets of frequently occurring attribute value combinations in the discretized space. One abstracts from the original data distribution of continuous valued attributes and only takes the discretized (in or outside the cell) information into account. On the one side, this makes the computation more efficient, however, on the other side discretization may result in loss of information and possibly less accurate clustering results. Furthermore, quality of the result is highly dependent on cell properties like width and positioning.

Simple counting has further advantages as it is easy to parametrize. Giving a threshold for the number of objects in a cluster is very intuitive. However, as this is a property of a single cluster one has only little control on the overall clustering result. For example, the mentioned monotonicity of CLIQUE induces that for each detected subspace cluster all lower dimensional projections will also be clusters. This might result in a tremendously large clustering result R where $numClusters(R) \gg |DB|$ is possible.

2.2 Density-Based Approaches

Density-based approaches are based on the clustering paradigm proposed in DBSCAN [13]. They compute the density of each object by counting the number of objects in its ε-neighborhood without prior discretization. A cluster with respect to the density-based paradigm is defined as a set of dense objects having more than $minPoints$ many objects in their ε-neighborhood. Arbitrarily shaped clusters are formed by a chain of dense objects lying within ε distance of each other. To determine the neighborhood for each object, a distance function is used (typically Euclidean distance). By changing the underlying distance function and the ε parameter one can specify the range of similar objects to be grouped in one cluster. This parametrization of the similarity gives the approaches in this paradigm high flexibility, but requires knowledge about suitable choices for the data, often not available in unsupervised learning.

DEFINITION 4. ***Density-Based Subspace Cluster.***

A density-based subspace cluster (O,S) is defined w.r.t. a density threshold $minPoints$ and ε-neighborhood $N_\varepsilon(q) = \{p \in DB \mid dist^S(p,q) \leq \varepsilon\}$, where $dist^S$ denotes a distance function restricted to the relevant dimensions S:

All objects are dense: $\forall o \in O : |N_\varepsilon(o)| \geq minPoints$.
All objects are connected: $\forall o,p \in O : \exists\ q_1,\ldots,q_m \in O : q_1 = o \wedge q_m = p \wedge \forall i \in \{2,\ldots,m\}\ q_i \in N_\varepsilon(q_{i-1})$.
The cluster is maximal: $\forall o,p \in DB$: *o,p connected* $\Rightarrow (o \in O \Leftrightarrow p \in O)$

The first approach in this area was SUBCLU [17], an extension of the DBSCAN algorithm to subspace clustering, by restricting the density computation to only the relevant dimensions. Using a monotonicity property, SUBCLU reduces the search space by pruning higher dimensional projections like CLIQUE. In contrast to grid-based approaches, the density-based paradigm uses the original data and requires expensive database scans for each ε-neighborhood computation. This results in an inefficient computation. A more efficient, however, approximative solution is proposed by FIRES [19]. Instead of going through the subspaces bottom up, FIRES uses 1d histogram information to jump directly to interesting subspace regions. A non-approximative extension of SUBCLU is INSCY [6], which eliminates redundant low dimensional clusters, detected already in higher dimensional projections. In contrast to bottom up approaches, INSCY processes subspaces recursively and prunes low dimensional redundant subspace clusters. Thus, it achieves an efficient computation of density-based subspace clusters.

The overall quality of density-based subspace clusters is dependent on the similarity specification. Similar to the dependency of cell-based approaches to their grid properties, finding meaningful parameter settings for the neighborhood range ε is a challenging task. FIRES uses a heuristic to adapt its $\varepsilon(|S|)$ to the subspace dimensionality $|S|$, however, it still has to initialize $\varepsilon(1)$. Similarly, INSCY uses a normalization for the density threshold $minPoints(|S|)$ for arbitrary subspaces [5], keeping the ε parameter fixed. Both enhancements allow flexible parametrization, but, not totally eliminate the challenging task of finding an adequate similarity for arbitrary subspaces. Furthermore, like for grid-based approaches, individual cluster properties give almost no control over the final clustering result.

2.3 Clustering-oriented Approaches

In contrast to the previous paradigms, clustering-oriented approaches focus on the clustering result R by directly specifying objective functions like the number of clusters to be detected or the average dimensionality of the clusters as in PROCLUS [1], the first approach for this paradigm. PROCLUS partitions the data into k clusters with average dimensionality l, extending K-means [20]. Instead of a cluster definition, clustering oriented approaches define properties of the set of resulting clusters. Each object is assigned to the cluster it fits best. More statistically oriented, P3C uses χ^2 test and the expectation maximization algorithm to find a more sophisticated partitioning [22, 12]. Defining a statistically significant density, STATPC aims at choosing the best non-redundant clustering [21]. Although it defines cluster properties, it aims at an overall optimization of the clustering result R.

DEFINITION 5. ***Clustering Oriented Results.***

A clustering oriented result w.r.t. objective functions $f(R)$, which is based on the entire clustering result R and an optimal value parameter $optF$ (e.g. $numClusters(R) = k$ and $avgDim(R) = l$ in PROCLUS) is a result set R with: $f(R) = optF$.

The most important property for clustering-oriented approaches is their global optimization of the clustering. Thus, the occurrence of a cluster depends on the residual clusters in the result. Based on this idea, these approaches are parametrized by specifying objective functions for the resulting set of clusters. Some further constraints about the clusters like in STATPC are possible, but, the global optimization of the result is still the major goal.

Clustering-oriented approaches directly control the resulting clusters, e.g. the number of clusters. Other paradigms do not control such properties as they report every cluster that fulfills their cluster definition. Both cell-based and density-based paradigms provide a cluster definition; every set of objects O and set of dimensions S fulfilling this definition is reported as subspace cluster (O, S). There is no optimization process to select clusters. On the other side, clustering oriented approaches do not influence the individual clusters to be detected. For example, keeping the number of clusters fixed and partitioning the data, optimizes the overall coverage of the clustering like in PROCLUS or P3C, but, includes noise into the clusters. As these approaches optimize the overall clustering they try to assign each object to a cluster, resulting in clusters containing highly dissimilar objects (noise). Both approaches are aware of such effects and use outlier detection mechanisms to remove noise out of the detected clusters. As these mechanisms tackle noise after the optimization process, clustering quality is still affected by noisy data.

3. EVALUATION MEASURES

In this section we describe the measures used in our evaluation of the different subspace clustering algorithms. While the efficiency can easily be measured in terms of *runtime*, the quality is more difficult to determine. One problem is that there is usually no ground truth to which we can compare the clustering result $R = \{C_1, \ldots, C_k\}$. For the classification task in contrast one can easily use labeled data as ground truth and compare these labels with the predicted labels of any classifier to obtain a reproducible quality measure. For clustering two possibilities to determine the ground truth are used. On synthetic data the "true" clustering is known a priori and hence the ground truth is given. We refer to these "true" clusters as the *hidden* clusters $H = \{H_1, \ldots, H_m\}$ in contrast to the *found* clustering result $R = \{C_1, \ldots, C_k\}$. For each $H_i = (O_i, S_i) \in H$ we can use information about the grouped objects O_i and the relevant dimensions S_i of the cluster, known by the data generator.
On real world data this information is not given. Therefore the idea is to use labeled data with the assumption that the natural grouping of the objects is somehow reflected by the class labels. All objects O_i with the same class label i are grouped together to one cluster $H_i = (O_i, S_i)$. Disadvantageous for this method is that the relevant dimensions of the cluster H_i cannot be deduced by the labels. However, assuming all dimensions to be relevant for each cluster (i.e. $S_i = D$) one can define for real world data also the hidden clusters $H = \{H_1, \ldots, H_m\}$.

We categorize the measures in two types depending on the required information about the hidden clusters H. The measures in the first category use information on which objects should be grouped together i.e. form a cluster. Consequently only the information O_i out of each hidden cluster H_i is regarded. The second category of measures is based on the full information about the hidden subspace clusters. The objects O_i and the relevant dimensions S_i of each hidden cluster must be given to calculate these measures. The application of these measures to real world data is somehow restricted as typically the relevant dimensions are not available as ground truth for the hidden clusters. We constantly set the relevant dimensions for such data to D. By this, full-space clusters are preferred over potentially more meaningful subspace clusters. Nonetheless one can use these measures to judge the grouping of the data objects based on the information O_i.

3.1 Object-based measures

Entropy [5, 30]: The first approach is to measure the homogeneity of the found clusters with respect to the hidden clusters. A found cluster should mainly contain objects from one hidden cluster. The merging (splitting) of several hidden clusters to one (different) found cluster (clusters) is deemed to be a low quality cluster. The homogeneity can be measured by calculating the entropy an information theoretic measure. Based on the relative number $p(H_i|C) = \frac{|O_{H_i} \cap O|}{|O|}$ of objects from the hidden cluster $H_i = (O_{H_i}, S_{H_i})$ that are contained in the found cluster $C = (O, S)$, the entropy of C is defined as:

$$E(C) = -\sum_{i=1}^{m} p(H_i|C) \cdot \log(p(H_i|C))$$

The overall quality of the clustering is obtained as the average over all clusters $C_j \in R$ weighted by the number of objects per cluster. By normalizing with the maximal entropy $log(m)$ for m hidden clusters and taking the inverse the range is between 0 (low quality) and 1 (perfect):

$$1 - \frac{\sum_{j=1}^{k} |C_j| \cdot E(C_j)}{\log(m) \sum_{j=1}^{k} |C_j|}$$

Hence the entropy measures the purity of the found clusters with respect to the hidden clusters.

F1 [6, 24]: The next measure evaluates how well the hidden clusters are represented. The found clusters which represent a hidden cluster should cover many objects of the hidden cluster but few objects from other clusters. This idea can be formulated with the terms recall and precision. Let $mapped(H) = \{C_1, \ldots, C_l\}$ (later described) be the found clusters that represent the hidden cluster H. Let O_H be the objects of the cluster H and $O_{m(H)}$ the union of all objects from the clusters in $mapped(H)$. Recall and precision are formalized by:

$$recall(H) = \frac{|O_H \cap O_{m(H)}|}{|O_H|} \quad precision(H) = \frac{|O_H \cap O_{m(H)}|}{|O_{m(H)}|}$$

A high recall corresponds to a high coverage of objects from H, while a high precision denotes a low coverage of objects from other clusters. The harmonic mean of precision and recall is the F1-measure where a high F1-value corresponds to a good cluster quality. The average over the F1-values for all hidden clusters $\{H_1, \ldots, H_m\}$ is the F1-value of the clustering:

$$\frac{1}{m} \sum_{j=1}^{m} \frac{2 \cdot recall(H_j) \cdot precision(H_j)}{recall(H_j) + precision(H_j)}$$

Different mappings of the found clusters to the hidden clusters are used in the literature. In our evaluation each found cluster is mapped to the hidden cluster which is covered to the most part by this found cluster. Formally, $C_i \in mapped(H)$ iff

$$\frac{|O_i \cap O_H|}{|O_H|} \geq \frac{|O_i \cap O_{H_j}|}{|O_{H_j}|} \quad \forall j \in \{1, \ldots, m\}$$

Accuracy [24, 10]: Another measure uses the accuracy of classification specified by $\frac{|\text{correctly predicted objects}|}{|\text{all objects}|}$ to judge the clustering quality. The idea is to predict the hidden cluster of an object on the basis of the detected patterns (i.e. the found clusters). The higher the accuracy the better is the generalization of the data set by the found clusters. The found clusters are a good description of the hidden clusters and hence the clustering quality is high. For quality measurement in recent publications, a decision tree classifier is build and evaluated (C4.5 with 10-fold cross validation) [24, 10]. To train the classifier the 'extracted dimensions' out of the clustering are used. Each object o is therefore represented as a bitvector of length k if we found k clusters $\{C_1, \ldots, C_k\}$. The position j in the bitvector equals 1 if $o \in C_j$, otherwise 0.
Please note that typically the classification accuracy based on the original dimensions of the objects instead of the extracted dimensions is higher because the bitvectors contain only knowledge that is generated by an unsupervised learning task.

3.2 Object- and subspace-based measures

Up to now no measure accounts for the relevant dimensions of a subspace cluster. However for synthetic data sets this information is available. The basic idea used in the next two measures is to consider the subspaces as follows: Instead of regarding the original database objects for the evaluation, each object is partitioned into subobjects annotated with a dimension. In a d-dimensional database the object o is partitioned in d different objects $o_1, \ldots, o_d$. A subspace cluster is henceforward not a subset of objects and a subset of dimensions but only a subset of these new subobjects. As a consequence two subspace clusters that share original objects but have disjoint relevant dimensions do not share subobjects. On the basis of this new representation further measures can be described.

RNIA [28]: A first approach is the relative nonintersecting area (RNIA) which measures to which extent the hidden subobjects are covered by the found subobjects. For a good clustering it is desirable to cover all and only the hidden subobjects. Formally one determines the subobjects which are in a hidden or found cluster (union U of subobjects) and subtracts the number of subobjects which are both in a hidden and found cluster (intersection I of subobjects). The more equal I and U the more equal are the found and hidden clustering and hence the better the clustering quality. To normalize the measure the term $RNIA = (U - I)/U$ is calculated. In the evaluation, we plot the value $1.0 - RNIA$ so that the maximal value 1 corresponds to the best clustering.

CE [28]: An extended version of RNIA is the clustering error (CE). One problem for the RNIA measure is that one cannot distinguish if several found clusters cover a hidden cluster or exactly one found cluster matches the hidden cluster. The RNIA-value is in both cases the same even though the second case is usually preferable. Therefore the CE measure maps each found cluster to at most one hidden cluster and also each hidden cluster to at most one found cluster. For each such mapping of two clusters the intersection of the subobjects is determined. Summing up the individual values gives us a value $\bar{I}$. Substituting the value I by $\bar{I}$ in the RNIA formula results in the CE-value. In this way one penalizes clustering results which split up a cluster in several smaller ones (with respect to objects or dimensions).

Both measures, CE and RNIA, were implemented in versions that can handle also nondisjoint clusterings as described in [28].

We perform thorough evaluation on all measures of the two categories because there is no best solution so far. Each measure has its advantages and disadvantages depicted in the experimental section. Furthermore for some properties like redundancy-removal ([21, 6]) or the consideration of relevant subspaces in real world data there exist no measure yet.

4. EXPERIMENTS

In our thorough evaluation, we focus on the general properties of the clustering paradigms. For comparability, we implemented all algorithms in a common framework [23]. By extending the popular WEKA framework we base our work on a widely used data input format for repeatable and expandable experiments [31]. We used original implementations provided by the authors and best-effort reimplementations based on the original papers. The authors of SUBCLU, FIRES and MINECLUS provided us with original implementations, which we only adapted to our framework. For all other approaches, we re-implemented the approaches in our framework as no publicly available implementations exist so far. We ensure comparable evaluations and repeatability of experiments, as we deploy all implemented algorithms and parameter settings on our website http://dme.rwth-aachen.de/OpenSubspace/evaluation.

With INSCY, we include one of our own subspace clustering algorithms as part of the density-based clustering paradigm without compromising objectivity and independence in evaluation. However, as it is our own approach we have more knowledge about parameterizing it. Nonetheless, our analysis is independent, as we evaluated a broad range of parameter settings for each algorithm to find the best parameters on each data set. We thus claim to provide an independent evaluation for clustering in subspaces of high dimensional data.

For a fair evaluation we ran massive experiments with various parameter settings for each algorithm. For the sake of space, we show an aggregated view of the results. Due to the enormous amount of experiment runs (23 data sets $\times$ 10 algorithms $\times$ on average 100 parameter settings per algorithm), we had to restrict the runtime for each run to 30 minutes. Based on preliminary experiments we observed runtimes of several days for some algorithms, which are clearly impractical. Experiments were run on a compute cluster with compute nodes equipped with four quad core Opteron 2.3 GHz CPUs running Windows 2008 Server. Java 32-bit runtime environment has been limited to using 1.5GB of RAM for each experiment. For repeatability, please refer to our website for an exhaustive list of parameter settings for each experiment.

4.1 Clustering Output (Quality)

For a meaningful clustering result different aspects are important. First of all, clustering should detect only a small set of clusters, far less than the number of objects. Second, the detected clusters should represent the hidden structures in the data as closely as possible. In the following experiments we will give more insights in these two properties for each paradigm. We generated data with 10 hidden subspace clusters with a dimensionality of 50%, 60% and 80% of the five dimensional data space. In Figures 2, 3, 4, 5 and 6 we show dependencies between the number of found clusters and different quality measures as introduced in Section 3. Out of various parameter settings we picked the best five results for each of the presented measures. Thus, we realize comparability with publications using only one of these measures. In addition, we extend comparison to a broader set of measures to achieve objective results.

For ease of illustration we depict each paradigm in a separate figure but on identical scales. First, let us consider the number of clusters on the x-axis. For each paradigm we observe different characteristics: The basic approaches of cell-based and density-based clustering, CLIQUE and SUBCLU tend to produce a huge amount of clusters (> 1000) to achieve good results as shown in Figure 2. It is a common property for the more recent representatives of the two paradigms to achieve good quality with fewer clusters. In contrast, clustering oriented approaches in general produce only very few clusters. Their clustering optimization leads to high clustering quality already with 10 to 100 clusters.

The distribution in number of clusters can be observed throughout the following figures with different quality measures on the y-axis. In Figure 2 we observe that Accuracy shows increasing quality with more and more detected clusters. The measure is correlated with the number of clusters. In contrast, in Figures 5 and 6 the RNIA and CE measure show a peak around 10 clusters. This is exactly the number of clusters hidden in the synthetic data set. For more clusters both measures decrease as hidden structures are split up into more clusters. CLIQUE and SUBCLU have only low clustering quality w.r.t. RNIA and CE.

As a next aspect we want to analyze the cluster distribution w.r.t. average dimensionality of the detected clusters. In Figure 7 we show the result distribution for all parameter settings of the cell-based clustering approach DOC w.r.t. CE vs. average dimensionality. Please keep in mind that the CE measure takes not only objects but also the relevant dimensions into account. Thus, we see best results for three to four dimensional subspaces as we have three and four dimensional clusters hidden in the data. Due to space limitations we show only the CE measure and the cell-based paradigm. For RNIA and all other approaches a similar distribution has been observed. In contrast to other measures like F1 and Accuracy the CE measure highlights clustering quality if the right objects are detected as clusters in the right subspaces.

At last, we want to compare some measures among each other and point out advantages or disadvantages. First, we analyze the entropy and F1 measure. While the entropy is based on the found clusters, the F1 measure focuses on the hidden clusters. The problem by focusing on the found clusters is that in some sense the found clusters are regarded as the "true" clusters. This could lead to misleading results. For exemplification let us consider the case of an algorithm

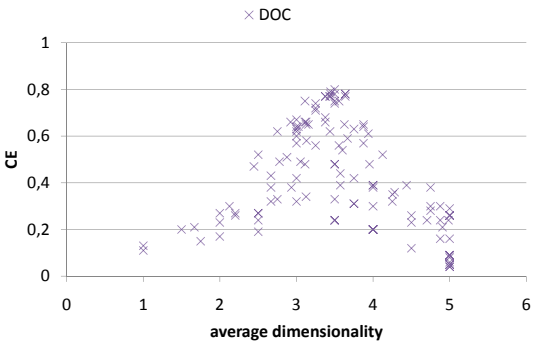


Figure 7: CE measure vs. average dimensionality

which detects only one out of several hidden clusters. If this found cluster is perfectly pure (i.e. only objects from one hidden cluster are contained), entropy reports optimal quality even though several other hidden clusters are not identified. Generally, entropy is biased towards high dimensional and therefore usually small and pure clusters (cf. Fig. 4(b) INSCY).

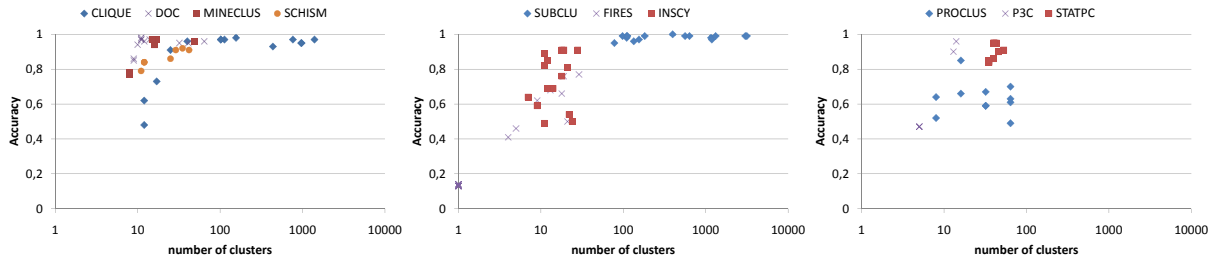
On the other hand, the F1 measure evaluates also detection of the hidden clusters. Not only the purity of the found clusters is important (resulting in a high precision), but also the detection of all hidden clusters (to get a high recall). Because both aspects are considered, the F1 measure is usually lower than the entropy measure (cf. Fig. 3 vs. 4), but also more meaningful.

The most important advantage of the last two measures RNIA and CE in comparison to Entropy, F1 and Accuracy was already mentioned in Section 3 and illustrated in Figure 7: Both measures consider the relevant dimensions of the clusters so that more meaningful quality values for synthetic data can result. Now we compare these two measure among each other. Looking at Figure 5 and 6 we see that the quality for RNIA is always larger than the CE quality. This is due to the fact that the RNIA measure do not penalize clusterings that distribute the found clusters over several hidden clusters. The difference in both measures however becomes smaller (e.g. for SUBCLU in Fig. 5(b) and 6(b)) if the result size if very large. Because of the large number of clusters the probability that a found cluster is mapped to a nearly identical hidden cluster increases and hence the difference in the intersection I in RNIA and $\bar{I}$ in CE (cf. Section 3.2) is small.
Another property of both measures is that for a huge result size the measures report poor clustering quality. This is in contrast to the accuracy measure which has usually improved quality w.r.t. the result size.

Because of the advantages of the CE measure on synthetic data, i.e. the consideration of the relevant subspaces, the penalization of high result sizes and the improvement over RNIA, we use this measure in the following experiments.

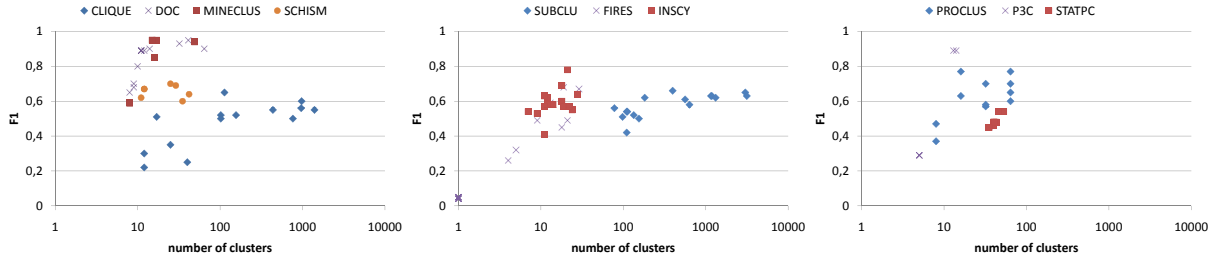
4.2 Scalability (Efficiency and Quality)

For scalability w.r.t. dimensionality of the database, we use synthetic data sets with 5-75 dimensional data. We generate data of different dimensionalities and hide 10 subspace clusters with a dimensionality of 50%, 60% and 80% of the data dimensionality. In Figure 8 we show clustering accuracy based on CE measure. We check quality both on object



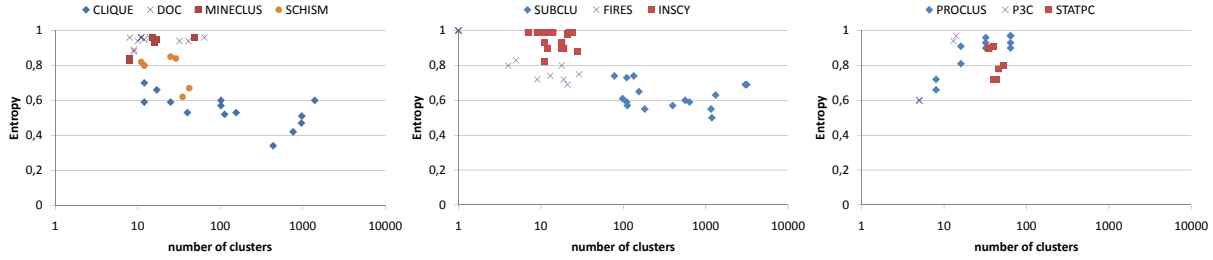
(a) cell-based (b) density-based (c) clustering-oriented

Figure 2: Accuracy measure vs. number of clusters



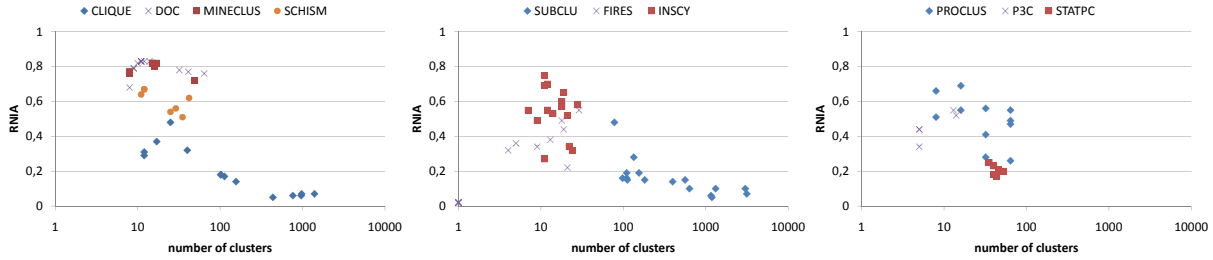
(a) cell-based (b) density-based (c) clustering-oriented

Figure 3: F1 measure vs. number of clusters



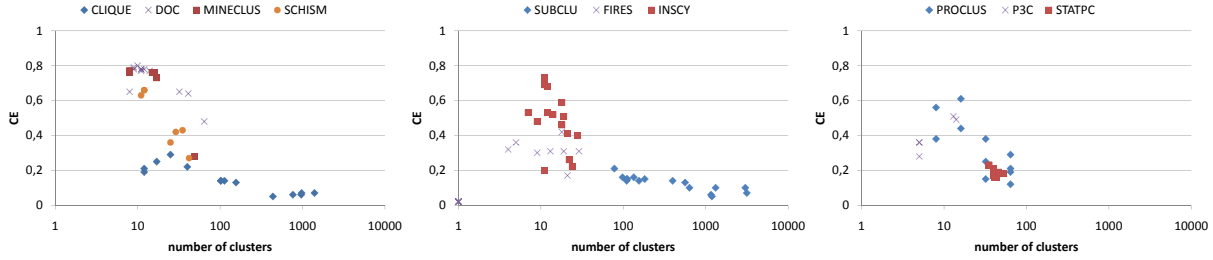
(a) cell-based (b) density-based (c) clustering-oriented

Figure 4: Entropy measure vs. number of clusters



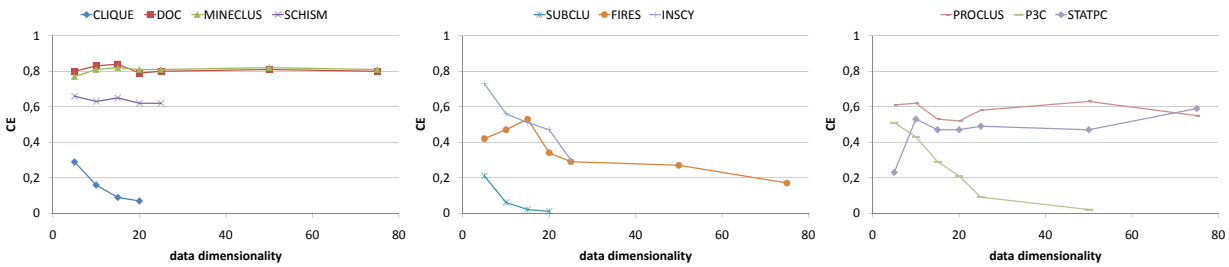
(a) cell-based (b) density-based (c) clustering-oriented

Figure 5: RNIA measure vs. number of clusters



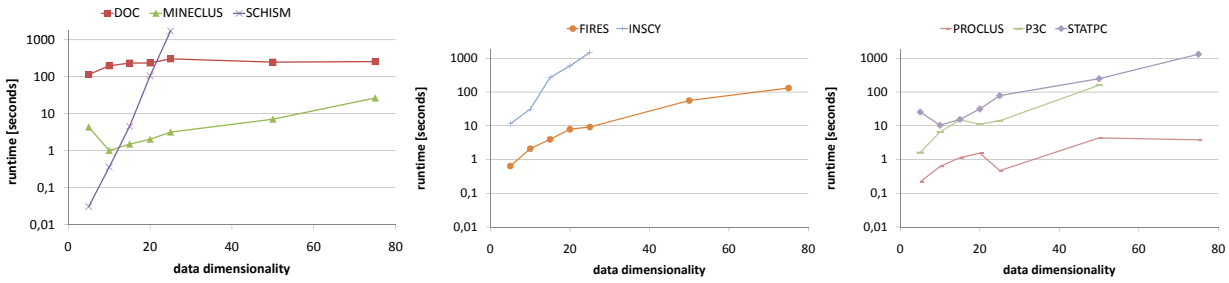
(a) cell-based (b) density-based (c) clustering-oriented

Figure 6: CE measure vs. number of clusters



(a) cell-based (b) density-based (c) clustering-oriented

Figure 8: Scalability: CE measure vs. database dimensionality



(a) cell-based (b) density-based (c) clustering-oriented

Figure 9: Scalability: runtime vs. database dimensionality

groups and detected subspaces as we know this for the synthetic data sets. Please note that for some algorithms due to extremely high runtimes we could not get meaningful results. In each paradigm at least one such algorithm is shown in Figure 8 where the quality measure decreases down to zero as there is not enough time to detect the hidden clusters. Preliminary experiments have shown that some algorithms result in runtimes of several days. The authors of SUBCLU published runtimes of up to six days [17]. Such extreme effects were observed especially for high dimensional data ($|D| \geq 25$) for several algorithms (CLIQUE, SUBCLU, INSCY, P3C). It is clearly impractical to evaluate algorithms with such high runtimes on high dimensionalities.

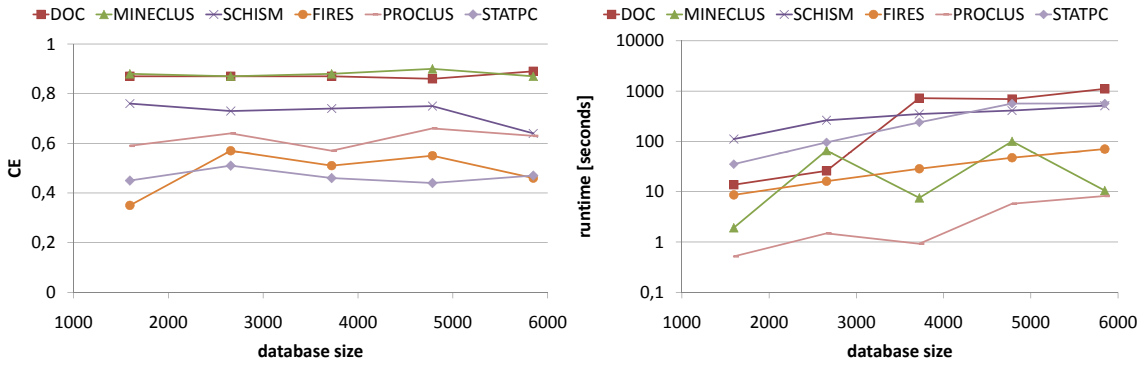
In general, cell-based approaches, except CLIQUE, show best results ($CE : 0.6 - 0.8$) for all dimensionalities. For clustering oriented approaches we observe medium to high quality ($CE : 0.4 - 0.7$). The density-based paradigm shows the worst quality measures with good results ($CE : 0.4 - 0.7$) only for low dimensional data ($|D| < 20$). Due to high runtimes all density-based approaches (and also CLIQUE and P3C in the other paradigms) are not scalable to higher dimensional data as they cannot detect the clusters anymore ($CE < 0.3$).

Efficiency is a major challenge in clustering of high dimensional data. In Figure 9 we depict the runtimes for the algorithms which detected the hidden clusters within the limit of 30 minutes. The runtimes of MINECLUS and PROCLUS show best runtimes for different dimensionalities. All other approaches show either significant increase of runtime for higher dimensional data sets or constantly high runtimes even for low dimensional data. Depending on the underlying clustering model, algorithms always have to tackle the trade-off between quality and runtime. Typically, high quality results have to be paid with high runtime. For cell-based approaches, DOC requires very many runs to achieve its very good clustering quality on our data sets, while SCHISM falls prey to the dramatically increasing number of fixed grid cells in high dimensional spaces. The density-based approaches in general do not scale to high dimensional data due to their expensive database scans for neighborhood density computation. Both their runtimes increase and thus high quality results are not achieved within the limited time in our evaluation. For clustering-oriented approaches statistic evaluations in both P3C and STATPC do not scale w.r.t. dimensionality.

Scalability w.r.t. database size is an important challenge especially for very large databases. In Figure 10(a) we show CE measure, while Figure 10(b) depicts the runtimes. We generated synthetic data by varying the number of objects per cluster, while keeping dimensionality fixed to 20d. We include only those algorithms which showed good quality results in this dimensionality in the previous experiments.

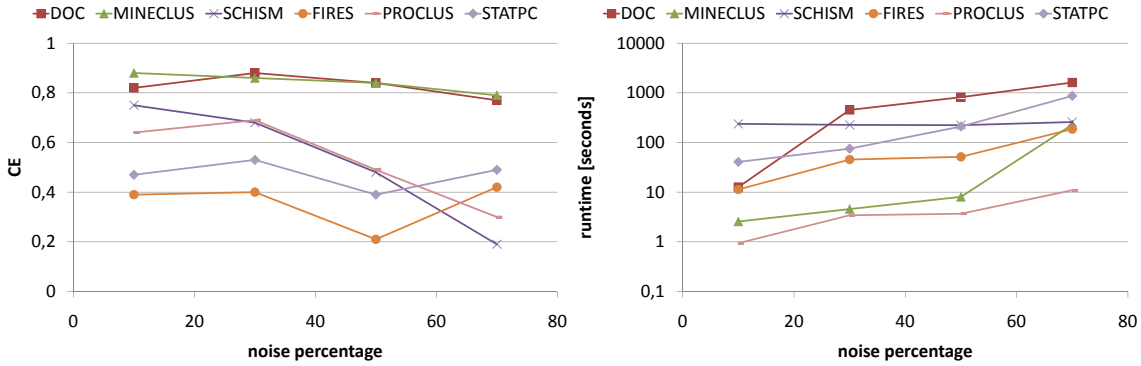
The cell-based paradigm outperforms the other paradigms also w.r.t. larger database sizes. It only slightly varies in clustering quality. Density-based and clustering oriented approaches show higher variance and overall significantly lower clustering quality. Considering runtime, the time to detect the clusters increases with the number of given objects for all approaches.

As clustering aims to detect clusters, it is challenging to cope with noisy data where some objects do not fit to any cluster. For the following experiment, we increase the percentage of noise objects in the database from 10% noise up to 70% noise, using the 20d database from the first scalability experiment. Figure 11 depicts quality and runtime results. As for database scalability we skip approaches that did not cluster in the given time. For clustering quality we see a significant decrease for almost all paradigms. Especially the clustering oriented approaches like PROCLUS are highly affected by the increase of noise. In terms of runtime we



(a) CE measure (b) time

Figure 10: Scalability: CE measure and runtime vs. database size



(a) CE measure (b) time

Figure 11: Scalability: CE measure and runtime vs. noise percentage

observe dramatically increasing runtimes for the cell-based approaches. Due to the increased amount of noise, more cells tend to have enough objects to form a cluster, but still DOC and MINCLUS achieve best quality results. For the residual algorithms, this leads to both a decrease in quality as clusters which mainly contain noise are detected, and an increase in runtime as overall more clusters are detected. In general, a partitioning of the data is not always meaningful as noise should not be assigned to any cluster.

4.3 Real World Data

We use benchmark real world data from the UCI archive [8]. These data sets, all of them or a subset, have been used for performance evaluations in recent publications [30, 6, 21]. In addition, we use 17-dimensional features as extracted [6] from the sequence data in [18]. In contrast to some of the data sets used in other publications, all of the data sets are publicly accessible. For repeatability we provide them along with the synthetic data sets on our website. For each of the data sets we have optimized parameters for each algorithm based on the resulting F1 and Accuracy measure. As the real world data sets are typically used for classification tasks, they all have class labels. However, there is no information about the relevance of dimensions per cluster. Thus, the measures CE and RNIA can only be applied on the object grouping. Therefore, we optimized F1 and Accuracy as these are the most meaningful measures where no information is given about the relevant subspaces (cf. Sec. 3 and Sec. 4.1).

For ease of illustration we picked one real world data set for discussion, while the residual data sets are given in Figure 13. Figure 12 shows the results for the *glass* data set. We show minimum and maximum values for various measures, starting with F1 and Accuracy which were used for parameter optimization. For these optimized results, we also show all other measures described in Section 3. For each measure we highlighted the best 95% results in gray. Additional information like the coverage of the resulting clustering, i.e. the proportion of objects which are contained in at least one cluster, the number of clusters, the average dimensionality (cf. Section 2), and the runtime are also included in the figures.

For F1 and Accuracy, the cell-based paradigm shows best results while also two density-based approaches have good accuracy values and one clustering oriented approach has the best result according to the F1 measure. However, going a little more into details, we observe that only DOC, MINECLUS and INSCY have also good values in CE and RNIA. Possible explanations can be derived from the basic measures: CLIQUE and SUBCLU achieve good F1 and Accuracy, but are punished by CE and RNIA for detection of very many clusters (far more than number of objects: 214 in the glass data set). Due to this excessive cluster detection covering 100% of the data (including noise as in most real world data sets) we also observe very high runtimes for both approaches. For our biggest real world data set, *pendigits*, SUBCLU did not even finish. Although SCHISM

Glass (size: 214; dim: 9)	F1		Accuracy		CE		RNIA		Entropy		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
CLIQUE	0,51	0,31	0,67	0,50	0,02	0,00	0,06	0,00	0,39	0,24	1,00	1,00	6169	175	5,4	3,1	411195	1375
DOC	0,74	0,50	0,63	0,50	0,23	0,13	0,93	0,33	0,72	0,50	0,93	0,91	64	11	9,0	3,3	23172	78
MINECLUS	0,76	0,40	0,52	0,50	0,24	0,19	0,78	0,45	0,72	0,46	1,00	0,87	64	6	7,0	4,3	907	15
SCHISM	0,46	0,39	0,63	0,47	0,11	0,04	0,33	0,20	0,44	0,38	1,00	0,79	158	30	3,9	2,1	313	31
SUBCLU	0,50	0,45	0,65	0,46	0,00	0,00	0,01	0,01	0,42	0,39	1,00	1,00	1648	831	4,9	4,3	14410	4250
FIRES	0,30	0,30	0,49	0,49	0,21	0,21	0,45	0,45	0,40	0,40	0,86	0,86	7	7	2,7	2,7	78	78
INSCY	0,57	0,41	0,65	0,47	0,23	0,09	0,54	0,26	0,67	0,47	0,86	0,79	72	30	5,9	2,7	4703	578
PROCLUS	0,60	0,56	0,60	0,57	0,13	0,05	0,51	0,17	0,76	0,68	0,79	0,57	29	26	8,0	2,0	375	250
P3C	0,28	0,23	0,47	0,39	0,14	0,13	0,30	0,27	0,43	0,38	0,89	0,81	3	2	3,0	3,0	32	31
STATPC	0,75	0,40	0,49	0,36	0,19	0,05	0,67	0,37	0,88	0,36	0,93	0,80	106	27	9,0	9,0	1265	390

Figure 12: Real world data set: *glass*

and STATPC have best results in Accuracy or F1 they show similar low values in CE and RNIA. Here the reason might be the detected relevant dimensions. While SCHISM tends to detect only very low dimensional projections, STATPC detects for all real world data sets full dimensional clusters (except on the *liver* data set).

The main properties we observed on synthetic data are validated by our real world scenarios. The cell-based paradigm overall shows best results with low runtimes in its recent representatives. The distance-based paradigm falls prey to its high runtimes and only finish on small data and only up to medium dimensional data (like e.g. glass with 10 dimensions). Clustering oriented approaches have shown reasonable runtimes and easy parametrization as they directly control the clustering result. However, as they do not define cluster properties explicitly like cell-based approaches they have lower quality measure values.

5. CONCLUSIONS

With this paper we provide a thorough evaluation and comparison of clustering in subspace projections of high dimensional data. We gave an overview of three major paradigms (cell-based, density-based and clustering oriented). We highlighted important properties for each of these paradigms and compared them in extensive evaluations. In a systematic evaluation we used several quality measures and provide results for a broad range of synthetic and real world data.

We provide the first comparison of different paradigm properties in a thorough evaluation. We could show that density-based approaches do not scale to very high dimensional data, while clustering oriented approaches are affected by noisy data resulting in low clustering quality. The recent cell-based approach MINECLUS outperformed, in most cases, the competitors in both efficiency and clustering quality. Surprisingly, the basic approach PROCLUS, in the clustering oriented paradigm, performs very well in our comparison. In contrast, the basic approaches CLIQUE and SUBCLU of the other two paradigms showed major drawback induced by the tremendously large result set. Recent approaches of these paradigms enhanced the quality and efficiency, however, could reach top results only in few cases. Summing up, we show that computing only a small set of relevant clusters like MINECLUS and PROCLUS and pruning most of the redundant subspace clusters achieves best results.

Our evaluation constitutes an important basis for subspace clustering research. We observe ongoing publications in this area for which our study gives a baseline for future evaluations. Our proposed baseline includes multiple aspects for a fair comparison not only in evaluation studies: First, a common open source framework with baseline implementations for a fair comparison of different algorithms. Second, a broad set of evaluation measures for clustering quality comparison. Third, a baseline of evaluation results for both real world and synthetic data sets with given parameter settings for repeatability. All of this can be downloaded from our website for further research, comparison or repeatability.

Acknowledgments

This research was funded in part by the cluster of excellence on Ultra-high speed Mobile Information and Communication (UMIC) of the DFG (German Research Foundation grant EXC 89).

Furthermore, we thank the authors of SUBCLU, FIRES and MINECLUS for providing us with their original implementations.

6. REFERENCES

[1] C. Aggarwal, J. Wolf, P. Yu, C. Procopiuc, and J. Park. Fast algorithms for projected clustering. In *SIGMOD*, pages 61–72, 1999.

[2] C. Aggarwal and P. Yu. Finding generalized projected clusters in high dimensional spaces. In *SIGMOD*, pages 70–81, 2000.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD*, pages 94–105, 1998.

[4] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *VLDB*, pages 487–499, 1994.

[5] I. Assent, R. Krieger, E. Müller, and T. Seidl. DUSC: Dimensionality unbiased subspace clustering. In *ICDM*, pages 409–414, 2007.

[6] I. Assent, R. Krieger, E. Müller, and T. Seidl. INSCY: Indexing subspace clusters with in-process-removal of redundancy. In *ICDM*, pages 719–724, 2008.

[7] I. Assent, E. Müller, R. Krieger, T. Jansen, and T. Seidl. Pleiades: Subspace clustering and evaluation. In *ECML PKDD*, pages 666–671, 2008.

[8] A. Asuncion and D. Newman. UCI Machine Learning Repository, 2007.

[9] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is nearest neighbors meaningful. In *IDBT*, pages 217–235, 1999.

Vowel (size: 990; dim: 10)

	F1		Accuracy		CE		RNIA		Entropy		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
CLIQUE	0,23	0,17	0,64	0,37	0,05	0,00	0,44	0,01	0,10	0,09	1,00	1,00	3062	267	4,9	1,9	523233	1953
DOC	0,49	0,49	0,44	0,44	0,14	0,14	0,85	0,85	0,58	0,58	0,86	0,86	64	64	10,0	10,0	120015	120015
MINECLUS	0,48	0,43	0,37	0,37	0,09	0,04	0,62	0,34	0,60	0,46	0,98	0,87	64	64	7,2	3,6	7734	5204
SCHISM	0,37	0,23	0,62	0,52	0,05	0,01	0,43	0,11	0,29	0,21	1,00	0,93	494	121	4,3	2,8	23031	391
SUBCLU	0,24	0,18	0,58	0,38	0,04	0,01	0,39	0,04	0,30	0,13	1,00	1,00	10881	709	3,6	2,0	26047	2250
FIRES	0,16	0,14	0,13	0,11	0,02	0,02	0,14	0,13	0,16	0,13	0,50	0,45	32	24	2,1	1,9	563	250
INSCY	0,82	0,33	0,61	0,15	0,09	0,07	0,75	0,26	0,94	0,21	0,90	0,81	163	74	9,5	4,3	75706	39390
PROCLUS	0,49	0,49	0,44	0,44	0,11	0,11	0,53	0,53	0,65	0,65	0,67	0,67	64	64	8,0	8,0	766	766
P3C	0,08	0,05	0,17	0,16	0,12	0,08	0,69	0,43	0,13	0,12	0,98	0,95	3	2	7,0	4,7	1610	625
STATPC	0,22	0,22	0,56	0,56	0,06	0,06	0,12	0,12	0,14	0,14	1,00	1,00	39	39	10,0	10,0	18485	16671

Diabetes (size: 768; dim: 8)

	F1		Accuracy		CE		RNIA		Entropy		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
CLIQUE	0,70	0,39	0,72	0,69	0,03	0,01	0,14	0,01	0,23	0,13	1,00	1,00	349	202	4,2	2,4	11953	203
DOC	0,71	0,71	0,72	0,69	0,31	0,26	0,92	0,79	0,31	0,24	1,00	0,93	64	17	8,0	5,1	1E+06	51640
MINECLUS	0,72	0,66	0,71	0,69	0,63	0,13	0,89	0,58	0,29	0,17	0,99	0,96	39	3	6,0	5,2	3578	62
SCHISM	0,70	0,62	0,73	0,68	0,08	0,01	0,36	0,09	0,34	0,20	1,00	0,79	270	21	4,2	3,9	35468	250
SUBCLU	0,74	0,45	0,71	0,68	0,01	0,01	0,01	0,01	0,14	0,11	1,00	1,00	1601	325	4,7	4,0	190122	58718
FIRES	0,52	0,03	0,65	0,64	0,12	0,00	0,27	0,00	0,68	0,00	0,81	0,03	17	1	2,5	1,0	4234	360
INSCY	0,65	0,39	0,70	0,65	0,37	0,11	0,45	0,42	0,44	0,15	0,83	0,73	132	3	6,7	5,7	112093	33531
PROCLUS	0,67	0,61	0,72	0,71	0,34	0,21	0,78	0,69	0,23	0,19	0,92	0,78	9	3	8,0	6,0	360	109
P3C	0,39	0,39	0,66	0,65	0,56	0,11	0,85	0,22	0,09	0,07	0,97	0,88	2	1	7,0	2,0	656	141
STATPC	0,73	0,59	0,70	0,65	0,06	0,00	0,63	0,17	0,72	0,28	0,97	0,75	363	27	8,0	8,0	27749	4657

Pendigits (size: 7494; dim: 16)

	F1		Accuracy		CE		RNIA		Entropy		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
CLIQUE	0,30	0,17	0,96	0,86	0,06	0,01	0,20	0,06	0,41	0,26	1,00	1,00	1890	36	3,1	1,5	67891	219
DOC	0,52	0,52	0,54	0,54	0,18	0,18	0,35	0,35	0,53	0,53	0,91	0,91	15	15	5,5	5,5	178358	178358
MINECLUS	0,87	0,87	0,86	0,86	0,48	0,48	0,89	0,89	0,82	0,82	1,00	1,00	64	64	12,1	12,1	780167	692651
SCHISM	0,45	0,26	0,93	0,71	0,05	0,01	0,30	0,08	0,50	0,45	1,00	0,93	1092	290	10,1	3,4	5E+08	21266
SUBCLU	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
FIRES	0,45	0,45	0,73	0,73	0,09	0,09	0,33	0,33	0,31	0,31	0,94	0,94	27	27	2,5	2,5	169999	169999
INSCY	0,65	0,48	0,78	0,68	0,07	0,07	0,30	0,28	0,77	0,69	0,91	0,82	262	106	5,3	4,6	2E+06	1E+06
PROCLUS	0,78	0,73	0,74	0,73	0,31	0,27	0,64	0,45	0,90	0,71	0,90	0,74	37	17	14,0	8,0	6045	4250
P3C	0,74	0,74	0,72	0,72	0,28	0,28	0,58	0,58	0,76	0,76	0,90	0,90	31	31	9,0	9,0	2E+06	2E+06
STATPC	0,91	0,32	0,92	0,10	0,09	0,00	0,67	0,11	1,00	0,53	0,99	0,84	4109	56	16,0	16,0	5E+07	3E+06

Liver-Dis. (size: 345; dim: 6)

	F1		Accuracy		CE		RNIA		Entropy		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
CLIQUE	0,68	0,65	0,67	0,58	0,08	0,02	0,38	0,03	0,10	0,02	1,00	1,00	1922	19	4,1	1,7	38281	15
DOC	0,67	0,64	0,68	0,58	0,11	0,07	0,51	0,35	0,18	0,11	0,99	0,90	45	13	3,0	1,9	625324	1625
MINECLUS	0,73	0,63	0,65	0,58	0,09	0,09	0,68	0,48	0,33	0,16	0,99	0,92	64	32	4,0	3,7	49563	1954
SCHISM	0,69	0,69	0,68	0,59	0,04	0,03	0,45	0,26	0,10	0,08	0,99	0,99	90	68	2,7	2,1	31	0
SUBCLU	0,68	0,68	0,64	0,58	0,11	0,02	0,68	0,05	0,07	0,02	1,00	1,00	334	64	3,4	1,3	1422	47
FIRES	0,58	0,04	0,58	0,56	0,14	0,00	0,39	0,01	0,37	0,00	0,84	0,03	10	1	3,0	1,0	531	46
INSCY	0,66	0,66	0,62	0,61	0,03	0,03	0,42	0,39	0,21	0,20	0,85	0,81	166	130	2,1	2,1	407	234
PROCLUS	0,53	0,39	0,63	0,63	0,26	0,11	0,66	0,25	0,05	0,05	0,83	0,46	6	2	5,0	3,0	78	31
P3C	0,36	0,35	0,58	0,58	0,55	0,27	0,96	0,47	0,02	0,01	0,98	0,94	2	1	6,0	3,0	172	32
STATPC	0,69	0,57	0,65	0,58	0,23	0,01	0,58	0,37	0,63	0,05	0,77	0,71	159	4	6,0	3,3	1890	781

Shape (size: 160; dim: 17)

	F1		Accuracy		CE		RNIA		Entropy		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
CLIQUE	0,31	0,31	0,76	0,76	0,01	0,01	0,07	0,07	0,66	0,66	1,00	1,00	486	486	3,3	3,3	235	235
DOC	0,90	0,83	0,79	0,54	0,56	0,38	0,90	0,82	0,93	0,86	1,00	1,00	53	29	13,8	12,8	2E+06	86500
MINECLUS	0,94	0,86	0,79	0,60	0,58	0,46	1,00	1,00	0,93	0,82	1,00	1,00	64	32	17,0	17,0	46703	3266
SCHISM	0,51	0,30	0,74	0,49	0,10	0,00	0,26	0,01	0,85	0,55	1,00	0,92	8835	90	6,0	3,9	712964	9031
SUBCLU	0,36	0,29	0,70	0,64	0,00	0,00	0,05	0,04	0,89	0,88	1,00	1,00	3468	3337	4,5	4,1	4063	1891
FIRES	0,36	0,36	0,51	0,44	0,20	0,13	0,25	0,20	0,88	0,82	0,45	0,39	10	5	7,6	5,3	63	47
INSCY	0,84	0,59	0,76	0,48	0,18	0,16	0,37	0,24	0,94	0,87	0,88	0,82	185	48	9,8	9,5	22578	11531
PROCLUS	0,84	0,81	0,72	0,71	0,25	0,18	0,61	0,37	0,93	0,91	0,89	0,79	34	34	13,0	7,0	593	469
P3C	0,51	0,51	0,61	0,61	0,14	0,14	0,17	0,17	0,80	0,80	0,66	0,66	9	9	4,1	4,1	140	140
STATPC	0,43	0,43	0,74	0,74	0,45	0,45	0,55	0,55	0,56	0,56	0,92	0,92	9	9	17,0	17,0	250	171

Breast (size: 198; dim: 33)

	F1		Accuracy		CE		RNIA		Entropy		Coverage		NumClusters		AvgDim		Runtime	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
CLIQUE	0,67	0,67	0,71	0,71	0,02	0,02	0,40	0,40	0,26	0,26	1,00	1,00	107	107	1,7	1,7	453	453
DOC	0,73	0,61	0,81	0,76	0,11	0,04	0,84	0,07	0,46	0,27	1,00	0,80	60	6	27,2	2,8	1E+06	37515
MINECLUS	0,78	0,69	0,78	0,76	0,19	0,18	1,00	1,00	0,56	0,37	1,00	1,00	64	32	33,0	33,0	40359	29437
SCHISM	0,67	0,67	0,75	0,69	0,01	0,01	0,36	0,34	0,35	0,34	1,00	0,99	248	197	2,3	2,2	158749	114609
SUBCLU	0,68	0,51	0,77	0,67	0,02	0,01	0,54	0,04	0,27	0,24	1,00	0,82	357	5	2,0	1,0	5265	16
FIRES	0,49	0,03	0,76	0,76	0,03	0,00	0,05	0,00	1,00	0,01	0,76	0,04	11	1	2,5	1,0	250	31
INSCY	0,74	0,55	0,77	0,76	0,02	0,00	0,24	0,11	0,60	0,39	0,97	0,74	2038	167	11,0	4,4	134373	63484
PROCLUS	0,57	0,52	0,80	0,74	0,51	0,11	0,65	0,43	0,32	0,23	0,89	0,69	9	2	24,0	18,0	703	141
P3C	0,63	0,63	0,77	0,77	0,04	0,04	0,19	0,19	0,36	0,36	0,85	0,85	28	28	6,9	6,9	6281	6281
STATPC	0,41	0,41	0,78	0,78	0,16	0,16	0,33	0,33	0,29	0,29	0,43	0,43	5	5	33,0	33,0	5187	4906

Figure 13: Residual real world data sets

[10] B. Bringmann and A. Zimmermann. The chosen few: On identifying valuable patterns. In *ICDM*, pages 63–72, 2007.

[11] Y. Cheng and G. M. Church. Biclustering of expression data. In *International Conference on Intelligent Systems for Molecular Biology*, pages 93–103, 2000.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. In *KDD*, pages 226–231, 1996.

[14] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.

[15] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *SIGMOD*, pages 1–12, 2000.

[16] I. Joliffe. *Principal Component Analysis*. Springer, New York, 1986.

[17] K. Kailing, H.-P. Kriegel, and P. Kröger. Density-connected subspace clustering for high-dimensional data. In *SDM*, pages 246–257, 2004.

[18] E. Keogh, L. Wei, X. Xi, S.-H. Lee, and M. Vlachos. LB_Keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures. In *VLDB*, pages 882–893, 2006.

[19] H.-P. Kriegel, P. Kröger, M. Renz, and S. Wurst. A generic framework for efficient subspace clustering of high-dimensional data. In *ICDM*, pages 250–257, 2005.

[20] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Berkeley Symp. Math. stat. & prob.*, pages 281–297, 1967.

[21] G. Moise and J. Sander. Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering. In *KDD*, pages 533–541, 2008.

[22] G. Moise, J. Sander, and M. Ester. P3C: A robust projected clustering algorithm. In *ICDM*, pages 414–425, 2006.

[23] E. Müller, I. Assent, S. Günnemann, T. Jansen, and T. Seidl. OpenSubspace: An open source framework for evaluation and exploration of subspace clustering algorithms in WEKA. In *Open Source in Data Mining Workshop at PAKDD*, pages 2–13, 2009.

[24] E. Müller, I. Assent, R. Krieger, S. Günnemann, and T. Seidl. DensEst: Density estimation for data mining in high dimensional spaces. In *SDM*, pages 173–184, 2009.

[25] E. Müller, I. Assent, R. Krieger, T. Jansen, and T. Seidl. Morpheus: Interactive exploration of subspace clustering. In *KDD*, pages 1089–1092, 2008.

[26] E. Müller, I. Assent, and T. Seidl. HSM: Heterogeneous subspace mining in high dimensional data. In *SSDBM*, pages 497–516, 2009.

[27] H. Nagesh, S. Goil, and A. Choudhary. Adaptive grids for clustering massive data sets. In *SDM*, 2001.

[28] A. Patrikainen and M. Meila. Comparing subspace clusterings. *TKDE*, 18(7):902–916, 2006.

[29] C. e. a. Procopiuc. A monte carlo algorithm for fast projective clustering. In *SIGMOD*, pages 418–427, 2002.

[30] K. Sequeira and M. Zaki. SCHISM: A new approach for interesting subspace mining. In *ICDM*, pages 186–193, 2004.

[31] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, USA, 2005.

[32] M. L. Yiu and N. Mamoulis. Frequent-pattern based iterative projected clustering. In *ICDM*, pages 689–692, 2003.