Knowl Inf Syst (2009) 21:299–326
DOI 10.1007/s10115-009-0226-y

REGULAR PAPER

Subspace and projected clustering: experimental evaluation and analysis

Gabriela Moise · Arthur Zimek · Peer Kröger · Hans-Peter Kriegel · Jörg Sander

Received: 1 August 2008 / Revised: 27 March 2009 / Accepted: 17 May 2009 /
Published online: 8 July 2009

Abstract Subspace and projected clustering have emerged as a possible solution to the challenges associated with clustering in high-dimensional data. Numerous subspace and projected clustering techniques have been proposed in the literature. A comprehensive evaluation of their advantages and disadvantages is urgently needed. In this paper, we evaluate systematically state-of-the-art subspace and projected clustering techniques under a wide range of experimental settings. We discuss the observed performance of the compared techniques, and we make recommendations regarding what type of techniques are suitable for what kind of problems.

Keywords Subspace clustering · Projected clustering

1 Introduction

As a prolific research area in data mining, subspace clustering and related problem statements produced a plethora of proposed solutions for clustering high-dimensional data in about the last decade. However, in many publications, a new proposition is compared with one or two competitors, if at all, and in a small range of experimental settings only (typically, settings that are favorable to the new propositions). The interested reader has barely a chance to learn about the advantages and disadvantages of the different approaches.

In [23], a broad range of solutions to the subspace clustering problem is surveyed and different kinds of problems are identified and distinguished. A conceptual framework to understand different behavior of different algorithms is, thus, provided. However, in the same survey, the complete lack of a comprehensive experimental study, comparing different algorithms tackling similar problem statements, is deplored.

G. Moise (✉) · J. Sander
Department of Computing Science, University of Alberta, Edmonton, AB T6G 2E8, Canada
e-mail: gabi@cs.ualberta.ca

A. Zimek · P. Kröger · H.-P. Kriegel
Ludwig-Maximilians-Universität München, Munich, Germany

In this study, we take a first step in order to experimentally evaluate a selection of representative techniques for subspace and projected clustering and to analyze their behavior competitively under controlled conditions. There may be different points of view for such an analysis. First, there may be the expectation that the competition could reveal one winner among the analyzed approaches. However, this is probably too much to expect, since clustering techniques usually perform differently well for different problem settings. Thus, a second point is more realistic: to find clues about what kinds of techniques may be preferable for what kind of problems.

In the remainder of this study, we first present in Sect. 2 a brief overview of important approaches to subspace and projected clustering and give a short rationale for our selection of techniques for the experimental evaluation. The main part, Sect. 3, is dedicated to describe the experimental settings, the data generated and used, and the performance of selected techniques in the different experiments. Subsequently, in Sect. 4, we discuss the performance of different approaches observed in the experiments and sketch what possibly can be learned from these observations. Finally, we conclude the study in Sect. 5.

2 Algorithms

Classical approaches to clustering follow different paradigms. There are partitioning approaches like k-means [26], CLARANS [33], or EM-clustering [13]. There are hierarchical approaches, either agglomerative (e.g., BAHC [43]) or divisive (e.g., DIANA [21]). A third paradigm is the density-based approach proposed with DBSCAN [15] and DENCLUE [19]. Several recent advancements have been achieved in the field of clustering such as techniques based on matrix approximation [24], or techniques that, in the discovery process, aim at avoiding redundancies with existing knowledge [16].

Seminal research [3,10,18] has shown that increasing data dimensionality results in the loss of contrast in distances between data points. Thus, clustering techniques that measure the similarity between data points by considering all attributes of a data set tend to break down in high-dimensional spaces. In addition, because of automatic data collection facilities, not all attributes of a data set may be relevant for the clustering analysis. Motivated by these observations, it has been hypothesized that data points may form clusters only when a subset of the attributes, i.e., a *subspace*, is considered. Furthermore, data points may belong to different clusters in different subspaces. For an illustration of these problems, see also [34].

Techniques proposed in the literature for discovering clusters of points in subsets of attributes can be classified into *subspace clustering* techniques, and *projected clustering* techniques. Both types of techniques are similar in the sense that they detect clusters of points that exist in subspaces of a data set. However, they differ in their problem definition, and in their strengths and weaknesses.

From an algorithmic point of view, most subspace clustering techniques are based on an APRIORI-like, bottom-up discovery of clusters based on some global density thresholds. They typically report a large number of overlapping clusters. There is much more diversity in the existing projected clustering techniques that can be classified, just like the full dimensional algorithms, into partitional, hierarchical, and density-based techniques. Some projected clustering techniques compute disjoint clusters; others have the option to assign data points to more than one cluster. Clustering and outlier detection are closely related [40]. Most projected and subspace clustering techniques are able to compute outliers.

Subspace clustering techniques search for all clusters of points in *all* subspaces of a data set according to their respective cluster definition. Existing subspace clustering techniques

start with one-dimensional clusters, which are subsequently merged bottom-up, similarly to the Apriori algorithm for finding frequent itemsets [7], in order to compute clusters of higher dimensionality. To avoid an exhaustive search through all possible subspaces, the cluster definition is based on a global density threshold that ensures anti-monotonic properties necessary for an Apriori style search. However, the cluster definition ignores the fact that density decreases with dimensionality. Large values for the global density threshold will result in only low-dimensional clusters, whereas small values for the global density threshold will result in a large number of low-dimensional clusters (many of which are meaningless), in addition to the higher dimensional clusters. Some subspace clustering techniques use an axis-aligned grid for estimating the density of a region in the data space. These techniques are sensitive to the resolution of the grid used, and they may miss clusters that are inadequately oriented or shaped with respect to the grid positioning.

Projected clustering techniques define a projected cluster as a pair (X, Y), where X is a subset of data points, and Y is a subset of data attributes, so that the points in X are "close" when projected onto the attributes in Y, but they are "not close" when projected onto the remaining attributes. Projected clustering techniques have an explicit or implicit measure of "closeness" on relevant attributes (e.g., small range/variance), and a "non-closeness" measure on irrelevant attributes (e.g., uniform distribution/large variance). A search method will report all projected clusters in the particular search space that a technique considers. If only k projected clusters are desired, the techniques typically use an objective function to define what the optimal set of k projected clusters is.

Typically, projected clustering techniques require parameters that are difficult to set by users (e.g., the number of projected clusters or the average number of relevant attributes of projected clusters), and they are sensitive to the values of these parameters.

Moreover, projected clustering techniques are less effective for discovering projected clusters with few relevant attributes embedded in high-dimensional spaces, because these techniques are either based on the computation of tentative clusters in full-dimensional space—which often do not represent well the low-dimensional projected clusters in the data—or they tend to prefer clusters with many relevant attributes.

Finally, many projected clustering techniques restrict the membership of a data point to at most one projected cluster. Although this effect may be desirable in some applications, it is preferable to have techniques that leave the decision to the user irrespective of whether the computed clusters should be disjoint or not.

In the following, we review in some detail the prominent approaches according to these categories. We note that, recently, some surveys on the topic of subspace and projected clustering have been published in the literature. The survey of Parsons et al. [34] illustrates the problem of subspace clustering, summarizes some relevant techniques, and compares two representative algorithms, but it does not discuss the underlying differences in the corresponding problem definition. A more recent edition of the data mining textbook by Han and Kamber [17] sketches some techniques and discusses some problems. Several distance measures for comparing subspace clusterings have been introduced in [35]. For a rather complete presentation of these approaches and related problem formulations, we refer the reader to [23].

We note that we focus on the problem formulation in which a subspace is defined as subset of the original attributes of a data set. For this reason, the techniques in this category are sometimes called *axis-parallel* subspace clustering techniques. A related problem formulation is the one in which a subspace is defined as an arbitrary set of orthogonal vectors, such as in ORCLUS [5]. Another related problem formulation that has emerged with the bioinformatics community is biclustering. For a comprehensive survey of these related approaches, see [23].

So far, to the best of our knowledge, there exists no comprehensive empirical evaluation with respect to the effectiveness and efficiency of all or at least the most prominent approaches.

2.1 Projected clustering techniques

Partitional projected clustering techniques use an objective function to define what is the optimal set of k projected clusters. These techniques address the optimization problem in an iterative manner: first, k "tentative" clusters are computed in full-dimensional space; second, sets of relevant attributes for each tentative cluster are computed; and, finally, the tentative clusters are refined based on the relevant attributes just computed. This iterative process is repeated several times, and the solution corresponding to the best objective function value is kept. These techniques report disjoint clusters, and they use some heuristics to identify outliers. Partitional approaches to projected clustering are PROCLUS [4] and SSPC [42].

PROCLUS represents each cluster by one of its points, called a "medoid", together with its set of relevant attributes. PROCLUS minimizes the average within-cluster dispersion, which is defined as the average Manhattan segmental distance between the members of a cluster and the cluster medoid. PROCLUS requires two critical input parameters: k, the desired number of clusters, and l, the average cluster dimensionality.

In an initialization phase, the technique randomly samples $A * k$ data points, and from this sample, it greedily selects a set M of $B * k$ scattered medoids, with the goal of selecting at least one medoid from each cluster. A and B are user-defined parameters. In an iterative phase, PROCLUS first selects k arbitrary medoids from the set M. Second, for each of the k medoids, it computes a "tentative" cluster, as described above. The average within-cluster dispersion of the current clustering solution is computed, and the clustering solution is recorded if it is the solution with the minimum dispersion obtained so far. PROCLUS tries to improve iteratively the current solution by detecting a "bad" medoid, replacing it with a random medoid from M, and re-computing clusters around the medoids. A medoid is "bad" if its cluster has few points. If the current solution cannot be improved after a certain number of replacements have been tried, then the technique terminates, and this solution is fed into the refinement phase. In the refinement phase, the clustering solution obtained at the end of the iterative phase is used to re-compute relevant attributes for each cluster. Subsequently, cluster members and outliers are determined.

PROCLUS tends to compute clusters that are hyper-spherical in shape and which exist in subspaces of approximately equal dimensionality. Due to a sampling step, PROCLUS may miss clusters with a small number of points. The performance of PROCLUS crucially depends on two required input parameters k and l, whose appropriate values are difficult to guess. Another weakness is the strong dependency on the initial clustering which is hard to determine since it is performed in full dimensional space where the "true" distances will be distorted by noisy attributes.

SSPC [42] is similar in structure to PROCLUS, and it uses an objective function based on the relevance score of HARP [41] (see below). An attribute a is relevant for a set of data points X if the variance of the projections of the points in X along attribute a is m times smaller than the variance of the projections of all data points along attribute a. The parameter m is a user-defined.

SSPC starts by determining, for each cluster, a set of representative points and relevant attributes. If available, domain knowledge in the form of labeled data points and/or attributes is used to improve the quality of the representative points and their relevant attributes. In each clustering round, for each cluster, a representative point from the associated set of representative points is chosen, and each data point is assigned to the representative point that

gives the greatest improvement in the objective function. If a data point does not improve the quality score of any cluster, it is put in the outlier list. Subsequently, the relevant attributes for each cluster are determined so that the objective function is maximized. The quality of the current clustering solution is recorded if it has the best score obtained so far. The solution with the best score is restored, a "bad" representative point (i.e., a representative point from a small cluster) is identified and replaced with another representative point, and the process is repeated until the current best score has not changed for a user-defined number of consecutive iterations.

SSPC requires the number of clusters k as a parameter, and its performance depends on the selection threshold m used to determine relevance scores of attributes.

Hierarchical projected clustering techniques are guided in their computation of clusters by the idea that clusters with many relevant attributes are preferable to clusters with few relevant attributes. Like the partitional projected clustering techniques, these techniques require the desired number of clusters k as a parameter, compute disjoint clusters, and use some heuristics to identify outliers.

HARP [41] measures the quality of a cluster as the sum of the relevance scores of its relevant attributes. The relevance score of an attribute a with respect to a set of data points X is computed by comparing the variance of the projections of points in X along attribute a with the variance of the projections of all data points along attribute a.

HARP is an agglomerative, hierarchical clustering technique that starts by placing each data point in a cluster. Two clusters are allowed to merge if the resulting cluster has $d_{\min}$ or more relevant attributes, and an attribute is selected as relevant for the merged cluster if its relevance score is greater than $R_{\min}$. $d_{\min}$ and $R_{\min}$ are two internal thresholds that start at some harsh values so that only points belonging to the same real cluster are likely to be merged. Subsequently, as the clusters increase in size, and the relevant attributes are more reliably determined, the two thresholds are progressively decreased until they reach some base values or a certain number of clusters has been obtained. HARP detects outliers by removing small clusters in two stages, and it requires the maximum percentage of outliers as a parameter.

In comparison to partitional approaches to projected clustering, HARP avoids the computation of tentative clusters that may not be reasonable approximations of real clusters. However, HARP is still less effective in the case of low-dimensional clusters because of its quality measure. HARP also has the drawback that decisions regarding the clustering of points cannot be undone at a later stage in the algorithm.

Density-based approaches to projected clustering can be classified into (1) DBSCAN-like techniques: PreDeCon [11]; (2) hyper-cube-based techniques: DOC/FASTDOC [36], MINECLUS [44], and (3) techniques based on the assumption that clusters stand out in low-dimensional projections: EPCH [32], FIRES [22], and P3C [27]. These techniques do not require the desired number of clusters as a parameter, and some of them are able to compute overlapping clusters, i.e., clusters that share data points. However, these techniques require various other parameters.

PreDeCon [11] computes a vector of weights for each point: attributes for which the variance of the points in a full-dimensional ε-neighborhood of the point is smaller than a threshold δ receive weight $k \gg 1$; the other attributes receive weight 1. Attributes with weight k are considered relevant for the ε-neighborhood of the point. For each point, a modified ε-neighborhood is computed, based on a Euclidean distance weighted with the weights associated with the point. A core point is a point with at most λ relevant attributes, and whose modified ε-neighborhood contains at least μ points. Based on core points, clusters are defined as in the DBSCAN algorithm. Because of the parameter λ, PreDeCon tends to discover clusters

with approximately the same dimensionality, and it is sensitive to the numerous required parameters. The computation of relevant attributes is done in full dimensional space, and thus, it may be less effective for low dimensional clusters.

DOC [36] defines a projected cluster as a pair (X, Y), where X is a subset of points, and Y is a subset of attributes, such that X contains at least a fraction α of the total number of points, and Y consists of all the attributes on which the projection of X is contained within a segment of length w. DOC uses the function $\mu(|X|, |Y|) = |X| * (1/\beta)^{|Y|}$ to measure the quality of a projected cluster, where α, β are user-specified parameters.

DOC computes one projected cluster at a time. It starts by selecting an arbitrary pivot point p, and subsequently, it randomly selects some data points, different from p, to form a tentative cluster for p. An attribute is considered relevant if the projections of all the tentative cluster's members on this attribute are within distance w from the projection of p on this attribute. The members of the projected cluster are all data points that fall within distance w from p on all attributes deemed as relevant. The process is repeated for $2/\alpha$ pivot points, and for each pivot point, m tentative clusters are tried. Finally, the projected cluster with the highest quality is reported. Then, the technique will be repeated for the next projected cluster. DOC can compute disjoint or overlapping clusters, depending on whether once a cluster has been found, its points are discarded from the data set or not. Outliers are defined as the points that remain un-clustered. DOC computes values for m and for the size of a tentative cluster so that the method proposed can recover with some high-probability projected clusters in the data.

In order to reduce the time complexity of DOC, its authors introduce a variant, called FASTDOC [36], which uses three heuristics to reduce the search time, but the clustering accuracy is no longer guaranteed. The most important heuristic is that the number of tentative clusters tried for a pivot point is bound to $maxout$.

The performance of DOC is sensitive to the choice of the input parameters, whose values are difficult to determine for real-life data sets. In addition, the assumption that a projected cluster is a hyper-cube of equal side length in all attributes may not be appropriate in real applications.

MINECLUS [44] improves upon DOC by proposing a deterministic method to find the optimal projected cluster around a given pivot point p. Each data point is modeled as an itemset that includes the attributes in which the point is within distance w from the pivot point. The problem of finding the projected cluster around p with maximum μ value becomes the problem of mining the frequent itemset with the maximum μ value. Yet, the accuracy of MINECLUS still depends on the three parameters α, β, and w. To compensate for the effect of these parameters, several heuristic refinement strategies are proposed.

EPCH [32] computes 1D or 2D histograms, and "dense" regions are identified in each histogram, based on iteratively lowering a threshold that depends on a user-specified parameter. For each data point, a "signature" is derived, which consists of the identifiers of the dense regions the data point belongs to. The similarity between two points is measured by the matching coefficient of their signatures in which zero entries in both signatures are ignored. Points are grouped in decreasing order of similarity until at most a user-specified number of clusters is obtained. The performance of EPCH is sensitive to the values of its parameters.

FIRES [22] starts with 1D clusters, called *base* clusters, which can be obtained using any clustering algorithm of choice. These base clusters are used to construct a shared k-nearest neighbor graph: vertices correspond to base clusters, and an edge connects two vertices if each vertex is among the k-nearest neighbors of the other vertex. A modified DBSCAN algorithm similar to [14] is applied to this graph. This algorithm takes two user-defined parameters, ϵ and $MinPts$, and produces several sets of base clusters. Each set of base clusters is

used to compute a higher-dimensional cluster, as follows. In a "pruning" step, base clusters that produce low-quality higher-dimensional clusters are removed. The quality of a higher-dimensional cluster is a function of its size and dimensionality. Then, DBSCAN with parameters ϵ and $MinPts$ is applied on the union of the remaining base clusters. The performance of FIRES is very sensitive to its multiple parameters.

P3C [27,28] first computes 1D regions corresponding to projections of clusters onto individual attributes. Second, these 1D regions are aggregated bottom-up into "cluster cores", i.e., hyper-rectangular approximations of projected clusters. P3C aggregates 1D regions into a cluster core only if there is enough statistical evidence for doing so, where the statistical test is based on comparing the real and the expected number of points in a cluster core. The statistical significance of this test is controlled by the parameter $Poisson\_threshold$. Finally, cluster cores are refined using the EM algorithm, and outliers are identified. P3C can compute disjoint or overlapping clusters. P3C's performance is affected when the 1D cluster projections cannot be reliably identified, or when the clusters have low density so that there is not enough statistical evidence for the aggregation of 1D projections.

STATPC [29] proposes a problem formulation that aims at extracting axis-parallel regions that stand out in the data in a statistical sense. The set R of all axis-parallel, statistically significant regions that exist in a data set is typically highly redundant. Therefore, Moise and Sander [29] propose to represent the set R through a reduced set of axis-parallel, statistically significant regions that in a statistically meaningful sense *explains* the existence of all the regions in R. The task of representing R through a reduced set of "explaining" regions is formulated as an optimization problem. Since exhaustive search is not a viable solution for the optimization problem due to computational infeasibility, an approximation algorithm STATPC is introduced. STATPC guarantees that its solution stands out in the data in a statistical sense, and it is not just an artifact of the method. The parameters required by STATPC are error probabilities that the user is willing to accept, and thus, setting these parameters does not require prior knowledge about the data.

2.2 Subspace clustering techniques

One way of estimating the density of a region in a high-dimensional space is to build an axis-aligned grid that partitions the data space into disjoint hyper-rectangular regions, called *units*. A unit is called *dense* if it contains at least some fraction of the points.

In this setting, a subspace cluster is defined as a maximal set of connected dense units in a subset of attributes. Subsequently, the subspace clustering problem is equivalent to the task of automatically identifying subspaces of the original feature space that contain dense units.

Grid-based techniques for subspace clustering are CLIQUE, nCluster, ENCLUS, and MAFIA. Their performance is typically very sensitive to the resolution of the grid and the density threshold used. They may miss some clusters in cases when heuristic pruning strategies are used. They may miss clusters inadequately oriented or shaped relative to the positioning of the grid.

A fundamental problem that affects all subspace clustering techniques is the use of global density thresholds for detecting subspace clusters in subspaces of increasing dimensionality. The global density thresholds guarantee some anti-monotonic properties that are used to avoid an exhaustive search through all possible subspaces. However, no meaningful values for these parameters are likely to exist: large values will result in only low-dimensional subspace clusters, and small values will results in numerous, spurious low-dimensional subspace clusters in addition to higher-dimensional subspace clusters.

CLIQUE [6] overlays an axis-aligned grid over the data space by partitioning each attribute into ξ equi-width units. A unit is *dense* if it contains more than a fraction τ of the points. Both ξ and τ are input parameters.

First, CLIQUE identifies subspaces of the original feature space that contain dense units. Second, for each of the identified subspaces, clusters are computed as disjoint sets of connected dense units. The density of a unit is an anti-monotonic property, and it is used to prune effectively the search space. Finally, a description is generated for each cluster by computing its cover with maximal, possibly overlapping, axis-parallel hyper-rectangles. CLIQUE suffers from all the problems of subspace clustering techniques described above.

nCluster [25] differs from CLIQUE in that the 1D units are overlapping windows of length δ. It suffers from the same problems as CLIQUE.

ENCLUS [12] is a grid-based subspace clustering technique that differs from CLIQUE in the criterion used for subspace selection. ENCLUS is based on the observation that the entropy of a subspace is higher when the points are uniformly distributed in the subspace than when the points are closely located in the subspace. A subspace having its entropy below a certain threshold ω is considered "good" for clustering. The entropy of a subspace decreases as the dimensionality of the subspace decreases. Thus, if a k-dimensional subspace has its entropy smaller than ω, then all $(k-1)$-dimensional subspaces obtained by removing one attribute from the k-dimensional subspace have entropy smaller than ω. This anti-monotonic property of entropy is used to generate in a bottom-up, Apriori-like style subspaces that are good for clustering.

ENCLUS suffers essentially from the same problems as CLIQUE. In addition, setting the parameter ω is not very intuitive.

MAFIA [30] addresses some of the drawbacks of CLIQUE. MAFIA partitions each attribute into *adaptive* units that capture the data distribution on that attribute, as follows. Each attribute is divided into a large number of bins. For each bin, the *bin count*, i.e., the number of data points that belong to a bin on an attribute, is computed. Adjacent bins whose bin counts differ by less than a threshold percentage β are merged into units. A single unit on an attribute implies an attribute with uniform distribution. The domain of an attribute with uniform distribution is divided into a fixed number of equi-sized units. A unit on an attribute is dense if it contains α times more points than the expected number of points if the data were uniformly distributed on that attribute.

The number of 1D dense units generated by MAFIA is much smaller than those generated by CLIQUE, which results in a smaller search space than in CLIQUE. In addition, a cluster is represented by a cross-product of dense units, and, thus, MAFIA avoids computing cluster descriptions as in CLIQUE. MAFIA only reports "maximal" clusters with respect to this definition of density.

MAFIA still suffers from problems similar to CLIQUE, namely sensitivity to the input parameters and the usage of a global density threshold.

Another way of estimating the density of a region in a high-dimensional space is to generalize the definition of a density-connected cluster underlying the full dimensional clustering algorithm DBSCAN for the problem of subspace clustering. SUBCLU [20] follows this approach by showing that density-connectivity is an anti-monotonic property. The difference between SUBCLU and CLIQUE or MAFIA is that CLIQUE and MAFIA compute hyper-rectangular dense units, whereas SUBCLU computes hyper-spherical dense units, that can be subsequently merged as in DBSCAN. Consequently, SUBCLU can detect subspace clusters with more general orientation and shape than the grid-based approaches.

SCHISM [39] overlays an axis-aligned grid over the data set by partitioning each attribute into ξ intervals of equal width. It defines a "subspace" as an axis-parallel hyper-rectangle

formed with cells of the constructed grid. The paper introduces the notion of "interestingness" of a subspace, i.e., a subspace is "interesting" if it contains significantly more points than expected under uniform distribution.

This notion of interestingness is defined in the paper as follows. If a subspace has dimensionality greater than a certain threshold v, which depends on ξ and the total number of points n, then the subspace is interesting if it contains more points than a constant density threshold, which also depends on ξ and n. Thus, interesting subspaces with dimensionality larger than v can be computed using an Apriori-like search. If a subspace has smaller dimensionality than v, then the subspace is interesting if it contains more points than a variable threshold, which is the minimum between a global density threshold u and another threshold that depends on the dimensionality of the subspace, the total number of points n, and a user-specified significance level τ. Interesting subspaces in the latter category cannot be detected with an Apriori-like algorithm because the variable threshold does not guarantee the anti-monotonic property necessary for an Apriori-like search. The paper proposes a depth-first search heuristic with backtracking that starts from one-dimensional interesting subspaces. However, the method is not guaranteed to recover all interesting subspaces with dimensionality smaller than v. In addition, it is observed that the interesting subspaces are redundant, and thus, the paper proposes to merge similar interesting subspaces, where the similarity is controlled by a user-defined threshold ρ.

The notion of interestingness of a subspace based on statistical principles is valuable. However, for the largest part of the search space, the actual density threshold is a global density threshold, and for the remaining search space interesting subspace clusters may not be found due to the heuristic search. Also, the interesting subspace clusters found depend on the grid-based discretization of individual attributes.

DUSC [9] proposes a density definition based on statistical foundations. DUSC defines a subspace cluster similarly to SUBCLU, except that a point is associated with a density measure, and a point is considered a core point if its density measure is F times larger than the expected value of the density measure under uniform distribution. The definition of a subspace cluster used by DUSC has no anti-monotonic properties, and thus, it cannot be used for pruning the search space. DUSC modifies the definition of a core point so that it has anti-monotonic properties in order to introduce a pruning method.

DiSH [1] observes that subspace clusters may form hierarchies in which multiple inheritance is possible, i.e., a subspace cluster may be embedded in more than one other subspace cluster. DiSH computes for each point p the highest dimensional subspace in which p fits best. This is achieved by analyzing the ϵ-neighborhood of the point p in each attribute, and keep as "relevant" the attributes where this ϵ-neighborhood contains more than μ points. The relevant attributes are combined bottom-up, in the Apriori style, in order to determine a set of relevant attributes where the ϵ-neighborhood of p contains at least μ points. For efficiency reasons, a best-first search heuristic can be used instead of Apriori.

Subsequently, a distance measure between data points is defined that assigns 1, if both points share a common one-dimensional subspace cluster, 2, if both points share a common two-dimensional subspace cluster, etc. This distance measure is fed into the OPTICS algorithm [8] in order to compute clusters of points. The reachability plot of OPTICS is not suitable for illustrating hierarchies with multiple inclusions; thus, DiSH includes a method and a visualization tool for this task.

Algorithmically, DiSH is in fact a hybrid approach since the procedure of computing subspace dimensionality is bottom-up and relies on a global density threshold, but the derived distance measures associated with the single points are used to search clusters in a top-down procedure.

2.3 Selection of representative techniques

We select as representative techniques from the family of projected clustering, the partitional techniques PROCLUS and SSPC, the hierarchical technique HARP, and, from the density-based approaches, we select PreDeCon, MINECLUS, FIRES, and P3C. DOC and FASTDOC are not selected, since we selected MINECLUS, which is the latest improvement over DOC and FASTDOC. We intended to compare with EPCH too, but after consulting with its authors, and using the original implementation, we could not find a parameter setting that produces results with reasonable accuracy on our synthetic data sets.

For subspace clustering, we select MAFIA, which can be considered as representative for techniques that are based on Apriori-like schemes, i.e., CLIQUE, nCluster, ENCLUS, and SUBCLU. We select DiSH as an algorithmically hybrid approach. Although DUSC and SCHISM initially propose a definition of density that is based on statistical principles, they actually propose algorithms based on global density thresholds, similar to CLIQUE and its variants, and thus, we do not include DUSC and SCHISM in our evaluation.

We also include STATPC in the experiments, which proposes a statistical-based approach to projected and subspace clustering.

The list of compared projected and subspace clustering algorithms is as follows: STATPC, PROCLUS, SSPC, HARP, MINECLUS, P3C, MAFIA, DiSH, FIRES, and PreDeCon.

We also include in the evaluation a representative sample of full dimensional clustering algorithms: k-means, EM, BAHC, DIANA, CLARANS, and DBSCAN. The performance of these algorithms indicates how difficult it is for full-dimensional clustering algorithms to discover subspace clusters embedded in the full-dimensional space.

3 Experimental evaluation

3.1 Synthetic data

We have generated data according to the following criteria:

1. The distribution of cluster points in the relevant subspace: (1) uniform or (2) Gaussian;
2. Clusters that have an equal number of relevant attributes versus clusters that have a different number of relevant attributes;
3. The average number of relevant attributes per cluster;
4. The database dimensionality d;
5. Database size n;
6. Number of clusters k;
7. Extent of clusters in their relevant attributes;
8. Overlap between the extent of clusters in common relevant attributes;

By combining the first 2 criteria, we obtain 4 categories of synthetic data sets: *Uniform_Equal*, *Uniform_Different*, *Gaussian_Equal*, and *Gaussian_Different*. A data set in the category *Uniform_Equal* is a data set where the cluster points are *uniformly* distributed in their relevant subspace, and clusters have an *equal* number of relevant attributes. For each category, we study the effect of the 3^{rd} criterion in data generation over the performance of the compared techniques. For this purpose, in each category, we generate data sets with $n = 300$ data points, $d = 50$ attributes, $k = 5$ clusters (clusters sizes are 60, 50, 40, 40, and 50 points), 60 uniformly distributed noise points, and the average number of relevant attributes in {2, 4, 6, 8, 10, 15, 20}. The clusters have axis-parallel orientation, i.e., when the

cluster points are Gaussian distributed in their relevant subspace, the Gaussian distributions have diagonal covariance matrices, and when the cluster points are uniformly distributed in their relevant subspace, the clusters are axis-parallel hyper-rectangles. Cluster points are uniformly distributed on [0, 1] on the irrelevant attributes. The extent of clusters in their relevant attributes is between 0.1 and 0.4 of the attribute range. No overlap between extent of clusters in common relevant attributes is introduced.

To study the effects of the remaining criteria in data generation, we generate synthetic data sets where the cluster points are *uniformly* distributed in their relevant subspace, and clusters have an *equal* number of relevant attributes (i.e., 4 relevant attributes per cluster), and we vary the parameter of interest.

To study the effect of the database dimensionality (fourth criterion), for a database of size $n = 300$, $k = 5$ (60, 50, 40, 40, 50 cluster points, and 60 uniformly distributed noise points), 4 relevant attributes per cluster, we vary $d \in \{20, 35, 50, 75, 100\}$.

To study the effect of the database size in data generation (fifth criterion), we vary $n \in \{100, 300, 500, 1{,}000, 2{,}000, 5{,}000, 7{,}000, 10{,}000\}$. Cluster sizes and number of noise points are as follows: for $n = 100$: 20, 17, 14, 14, 17 cluster points and 18 noise points; for $n = 300$: 60, 50, 40, 40, 50 cluster points and 60 noise points; for $n = 500$: 100, 84, 67, 67, 84 cluster points and 98 noise points; for $n = 1{,}000$: 200, 170, 140, 140, 170 cluster points and 180 noise points; for $n = 2,000$: 400, 340, 280, 280, 340 cluster points and 360 noise points; for $n = 5{,}000$: 1,000, 834, 666, 666, 834 cluster points and 1000 noise points; for $n = 7{,}000$: 1,400, 1,167, 933, 933, 1,167 cluster points and 1,400 noise points; for $n = 10{,}000$: 2,000, 1,666, 1,334, 1,334, 1,666 cluster points, and 2,000 noise points.

To study the effect of the number of clusters in data generation (sixth criterion), we vary $k \in \{2, 5, 10, 15, 20\}$. Cluster sizes are fixed to 40 points per cluster, and the number of noise points is fixed to 100 points.

To study the effect of the extent of clusters in their relevant attributes in data generation (seventh criterion), we generate the clusters 0.1, 0.2, 0.3, respectively 0.4 extent in the relevant attributes.

To study the effect of the overlap between the extent of clusters in common relevant attributes in data generation (eighth criterion), we generate data sets with $k = 2$ (125, 125 cluster points and 50 noise points), so that the two clusters are characterized by an overlap of extent in their common relevant attributes of 0, 0.1, 0.2, respectively 0.3.

We summarize the data generation criteria in Tables 1, 2 and 3.

3.2 Real data

We choose to study the performance of the compared techniques extensively on synthetic data, because on synthetic data we can control all aspects of the implanted clusters, and we can study the criteria involved in data generation systematically by keeping all criteria fixed except the criterion on interest which is to be varied.

In addition, the evaluation of clustering results on synthetic data is straightforward, because the membership of the implanted clusters is known. In contrast, the evaluation of clustering results on real data is much more difficult. If the real data are unlabeled, then the evaluation of clustering results must be done based on domain knowledge or with the help of a domain expert. However, this scenario is impractical for the case when numerous clustering techniques that depend on various parameters need to be evaluated. If the real data are labeled for classification purposes, then we could use class labels as cluster labels. However, we have to be aware of the fact that, although class labels typically indicate some similarities between

Table 1 Criteria 1, 2 and 3 in data generation

	Unif_Equal	Unif_Diff	Gauss_Equal	Gauss_Diff
Distrib. of cluster points in relevant subspace	Unif	Unif	Gauss	Gauss
Equal versus diff. no. of relevant attr.	Equal	Diff	Equal	Diff
Avg. clust. dim.	2, 4, 6, 8 10, 15, 20	2, 4, 6, 8 10, 15, 20	2, 4, 6, 8 10, 15, 20	2, 4, 6, 8 10, 15, 20
Db. dim. d	50	50	50	50
Db. size n	300	300	300	300
No. clust. k	5	5	5	5
Extent clus. in relevant attr.	0.1–0.4	0.1–0.4	0.1–0.4	0.1–0.4
Overlap clus. in common relevant attr.	0	0	0	0
Clust. orientation	Axis-parallel	Axis-parallel	Axis-parallel	Axis-parallel
Distrib. of cluster points in irrelevant attr.	Unif [0,1]	Unif [0,1]	Unif [0,1]	Unif [0,1]

Table 2 Criteria 4, 5 and 6 in data generation

	Db. dim. d	Db. size n	No. clust. k
Distrib. of cluster points in relevant subspace	Unif	Unif	Unif
Equal vs. diff. no. of relevant attr.	Equal	Equal	Equal
Avg. clust. dim.	4	4	4
Db. dim. d	20, 35, 50 75, 100	50	50
Db. size n	300	100, 300, 500, 1000 2,000, 5,000, 7,000, 10,000	180, 300, 500 700, 900
No. clust. k	5	5	2, 5, 10, 15, 20
Extent clus. in relevant attr.	0.1–0.4	0.1–0.4	0.1–0.4
Overlap clus. in common relevant attr.	0	0	0
Clust. orientation	Axis-parallel	Axis-parallel	Axis-parallel
Distrib. of cluster points in irrelevant attr.	Unif [0,1]	Unif [0,1]	Unif [0,1]

Table 3 Criteria 7 and 8 in data generation

	Extent clust. in relevant attr.	Overlap clust. in common relevant attr.
Distrib. of cluster points in relevant subspace	Unif	Unif
Equal vs. diff. no. of relevant attr.	Equal	Equal
Avg. clust. dim.	4	4
Db. dim. d	50	50
Db. size n	300	300
No. clust. k	5	2
Extent clus.	0.1, 0.2	0, 0.1
in relevant attr.	0.3, 0.4	0.2, 0, 3
Overlap clus. in common relevant attr.	0	0
Clust. orientation	Axis-parallel	Axis-parallel
Distrib. of cluster points in irrelevant attr.	Unif [0,1]	Unif [0,1]

members of the same class, class labels are not the "perfect" ground truth in the sense that they do not correspond necessarily to potential subspace clusters in the data.

Keeping these issues in mind, we have tested the compared techniques on the following data sets from the UCI machine learning repository [38]: Pima Indians Diabetes (768 points, 8 attributes, 2 classes); Liver Disorders (345 points, 6 attributes, 2 classes); Iris (150 points, 4 attributes, 3 classes); and Glass (214 points, 9 attributes, 6 classes).

These real data sets were collected for classification purposes. However, in such real data sets, most of the attributes were selected in the first place because they were considered potentially relevant for the classification problems. Consequently, the real data sets may contain only full-dimensional subspace clusters or very high-dimensional subspace clusters. To actually verify the capability of the competing techniques to find subspace clusters, we add 5, 10, 20, and 50 attributes, respectively, to each real data set where the data points are uniformly distributed in [0, 1]. Subspace clusters that may exist in the data sets, full dimensional or not, will be subspace clusters of increasingly lower dimensionality as we add more uniform attributes to the data sets.

3.3 Experimental setup

MINECLUS, HARP, SSPC, FIRES, P3C, and DiSH are all tested with the original implementations. PROCLUS, BAHC, DIANA, and CLARANS are provided by the Biosphere project [43]. We implemented MAFIA ourselves. For k-means (denoted by KM) and EM, we used the implementations available in the R statistical software [37]. For PreDeCon and DBSCAN, the implementations in the ELKI framework [2] are used.

For synthetic data, we set the target number of clusters to the number of implanted clusters for techniques that require a target number of clusters. PROCLUS and PreDeCon require the

average or maximal cluster dimensionality, which is set to the known average or maximal cluster dimensionality, respectively. HARP requires the maximum percentage of outliers, which is set to the known percentage of outliers. For techniques that require other parameter settings, we set these parameters as recommended by their authors: for PROCLUS: $A = 20$, $B = 5$; for MINECLUS: $w = 0.3, \alpha = 0.1, \beta = 0.25, maxout = 20$; for SSPC: $m = 0.5$; for P3C: $Poisson\_threshold = 1.0E - 5$; for MAFIA: $\alpha = 1.5$, $\beta = 0.35$, $no\_tiny\_bins = 50$, $no\_intervals\_unif\_distrib = 5$; for CLARANS: $maxn = 250$, $numl = 5$; for DiSH: $\mu = 20$, $\varepsilon = 0.1$; for PreDeCon and DBSCAN, *minpts* is set to 10, and ε is set by approximating the density of the clusters based on *minpts* and the parameters involved in data generation. The remaining parameter of PreDeCon is $\delta = 0.2$. For FIRES, DBSCAN is used for preprocessing with $\varepsilon = 0.02$ and $minpts = 20$; for the merge-step, the parameters are $\mu = 3$, $k = 4$, and $minClu = 3$, as suggested in the paper; for the post-processing, pruning and refinement (again using DBSCAN) is performed as suggested by the authors.

SSPC is run without the supervision option. Some techniques are not deterministic; thus, each of them is run 5 times, and the results are averaged.

On real data, we use class labels as cluster labels. We set the target number of clusters to the number of classes. For the remaining parameters, such as the average cluster dimensionality, whose values are hard to determine, several values in a meaningful range are tried and the results with best accuracy are reported.

3.4 Performance measures

We use an $F_$value to measure the clustering accuracy. We refer to implanted clusters as *input* clusters, and to found clusters as *output* clusters. For each output cluster i, we determine the input cluster j^i with which it shares the largest number of data points. The *precision* of output cluster i is defined as the number of data points common to i and j^i divided by the total number of data points in i. The *recall* of output cluster i is defined as the number of data points common to i and j^i divided by the total number of data points in j^i. The $F_$value of output cluster i is the harmonic mean of its precision and recall. The $F_$value of a clustering solution is obtained by averaging the $F_$values of all its output clusters. Similarly, we use an $F_$value to measure the accuracy of found relevant attributes based on the matching between output and input clusters. This is, of course, not done for full-dimensional algorithms. Furthermore, the implementations of PreDeCon and FIRES do not report the set of attributes considered as relevant for a cluster.

3.5 Accuracy results

Effect of average cluster dimensionality. Figures 1, 2, 3 and 4 show the accuracy of the compared techniques as a function of increased average cluster dimensionality for the categories *Uniform_Equal*, *Uniform_Different*, *Gaussian_Equal*, and *Gaussian_Different*, respectively.

We observe that STATPC significantly and consistently outperforms the other techniques, both in terms of clustering accuracy and in terms of accuracy of the found relevant attributes. The difference in accuracy between STATPC and previous techniques is more pronounced for the more difficult case of data sets with low-dimensional subspace clusters.

We observe that the accuracies of SSPC, PROCLUS, HARP, and MINECLUS in terms of cluster points, as well as relevant attributes, increase as the average cluster dimensionality increases. PROCLUS depends strongly on an initial clustering in full-dimensional space, which is a better approximation of the implanted clusters as the average cluster dimensionality increases, because the implanted clusters become more easily recognizable in

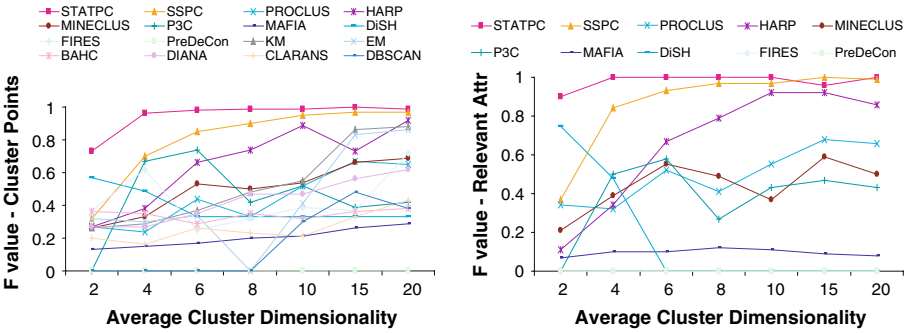

Fig. 1 Category Uniform_Equal

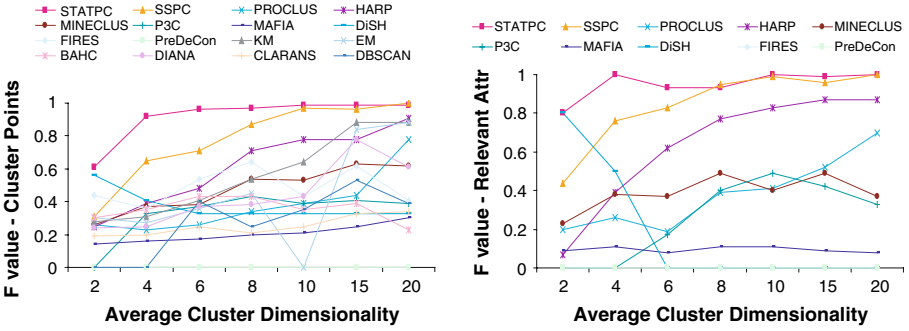

Fig. 2 Category Uniform_Different

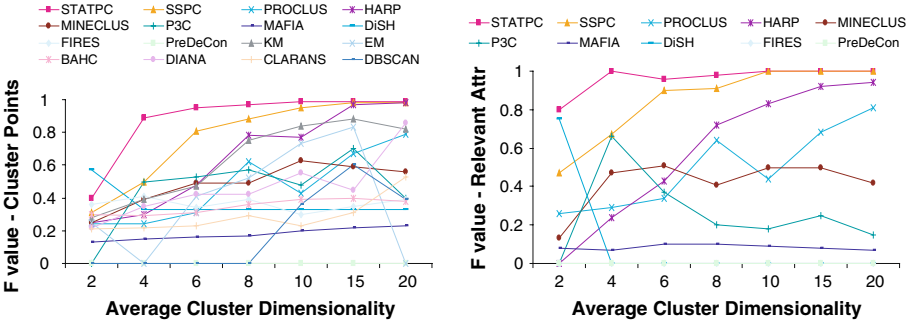

Fig. 3 Category Gaussian_Equal

fulldimensional space. The accuracies of SSPC, HARP, and MINECLUS increase with increasing average cluster dimensionality, as expected, since they are biased towards discovering clusters with as many attributes as possible.

The accuracy of MAFIA increases only slightly as the average cluster dimensionality increases, but this accuracy stays low overall.

P3C does not exhibit increasing accuracy as the average cluster dimensionality increases. The reason is that, on these data sets, the density of the implanted clusters is so low that P3C cannot detect all the 1D projections of the implanted clusters, and the statistical evidence

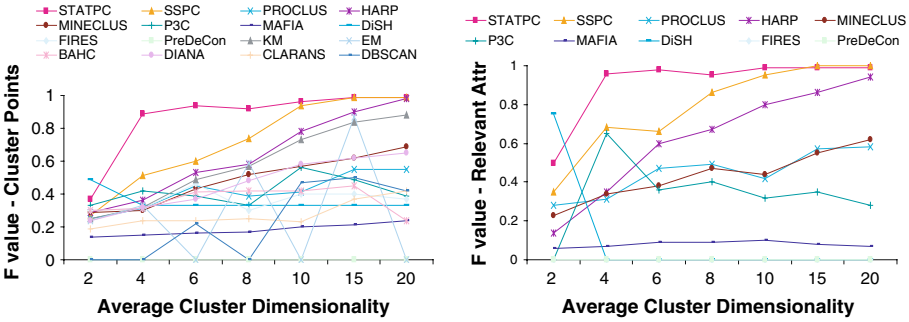

Fig. 4 Category Gaussian_Different

needed to aggregate 1D cluster projections is lacking. Similar effects may be responsible for the behavior or FIRES that shows no conclusive tendency.

PreDeCon did not find any clusters. The reason may be that the subset of relevant attributes is assessed based on the neighborhood of a point in full-dimensional space. Although this approach has shown useful in some data sets of moderate dimensionality, the experiments in this study use data sets of a dimensionality where the curse of dimensionality already seriously affects a meaningful assessment of neighborhood.

DiSH appears to be more reliable for very low-dimensional clusters and deteriorates with increasing cluster dimensionality. This could be an effect of the best-first search strategy to assign an appropriate subspace to each point, starting with single attributes. Accordingly, the reliability of DiSH in the detection of the correct subspaces [cf. Figs. 1(right), 2, 3, 4(right)] is significantly better than that of the other techniques for the experiment with 2 relevant out of 50 attributes but deteriorates rapidly to an F_value of 0 for higher-dimensional clusters.

The accuracies of the full-dimensional clustering algorithms increase as the average cluster dimensionality increases. The reason is that the higher the average cluster dimensionality, the more recognizable are the implanted clusters in full-dimensional space. EM may sometimes report an accuracy of 0 if it encounters in the computation singular or nearly-singular covariance matrices. We observe that the full-dimensional clustering algorithms are not effective for the task of retrieving the implanted clusters, especially when the average cluster dimensionality is small. However, some of the full-dimensional clustering algorithms outperform some of the projected and subspace clustering techniques, especially for higher average cluster dimensionalities.

Effect of distribution of cluster points in the relevant subspace. For all techniques, the accuracy results on data sets where cluster points are uniformly distributed in their relevant subspace are slightly higher than the accuracy results on data sets where cluster points are Gaussian distributed in their relevant subspace. The reason is that Gaussian-distributed clusters are denser in the center than at the boundaries, and all techniques tend to fail detecting good boundaries; thus, the decrease in accuracy.

Effect of equal versus different number of relevant attributes. For most techniques, the accuracy results on data sets where clusters have an equal number of relevant attributes are slightly higher than the accuracy results on data sets where clusters have a different number of relevant attributes, although the average cluster dimensionality is the same in both cases. This is because, in the latter case, there are implanted clusters with low dimensionality, which are more difficult to retrieve than the implanted clusters with higher dimensionality. STATPC is not affected by this criterion.

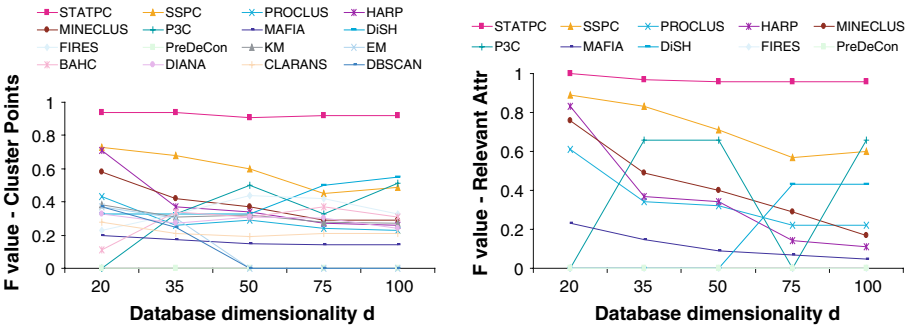

Fig. 5 Effect d

Effect of database dimensionality. Figure 5 shows the accuracy of the compared techniques as a function of increasing database dimensionality d. Varying the dimensionality of the data set has a similar effect as varying the average cluster dimensionality: the dimensionality of the clusters varies relatively to the database dimensionality. However, in this experiment we see the effects on a different range of d.

STATPC obtains consistently a high accuracy as the database dimensionality increases, and significantly higher accuracy than the accuracies of the other techniques.

The accuracy of PROCLUS declines as d increases, because its initialization in full dimensional space approximates increasingly worse the implanted clusters. Similarly, HARP's accuracy decreases, because HARP favors clusters with many relevant attributes. SSPC's accuracy decreases too, because it becomes increasingly difficult to initialize it with "good" seeds and relevant attributes. MINECLUS accuracy decreases because of the parameter β that fails to control effectively the trade-off between cluster sizes and number of relevant attributes.

The accuracy of MAFIA slightly decreases with increasing d, because MAFIA reports more low dimensional projections of the implanted clusters.

The accuracies of P3C and FIRES alternate between higher and lower values, depending on how successful these approaches are in detecting and aggregating cluster projections.

PreDeCon, again, does not find any clusters, probably for the same reasons as stated above.

DiSH shows better accuracy with increasing d. The relative dimensionality of the implanted clusters becomes increasingly low which favors DiSH in a similar way as the average cluster dimensionality.

The accuracies of KM, BAHC, DIANA, CLARANS, and DBSCAN decrease with increasing d, because the pair-wise distances between points become more and more similar. The accuracy of EM decreases too, because the quality of the 1-step KM initialization decreases, and because, as d increases, the covariance matrix of each cluster tends to over-fit the data more and more.

Effect of database size. Figure 6 shows the accuracy of the compared techniques as a function of increasing database size n.

The accuracy of STATPC has a consistently high value once the number of data points is at least 300. When the data set has only 100 data points, STATPC misses the implanted clusters with the least number of points (i.e., the two clusters with 14 points), which are only marginally statistically significant.

The accuracies of PROCLUS, HARP, SSPC, MINECLUS, and DiSH increase with increasing n, because the more points are in a cluster, the more reliable is the identification of relevant attributes.

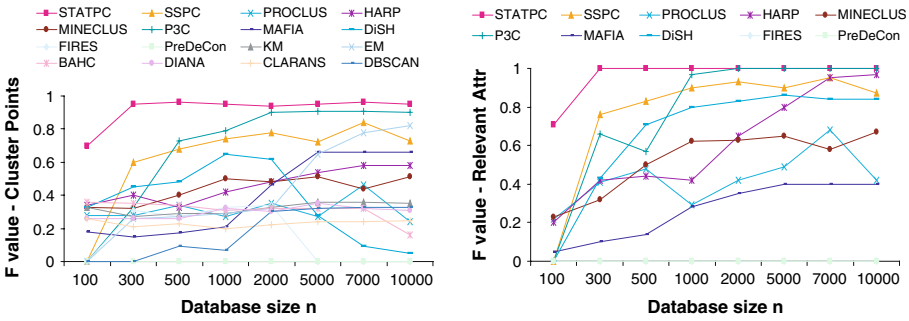

Fig. 6 Effect n

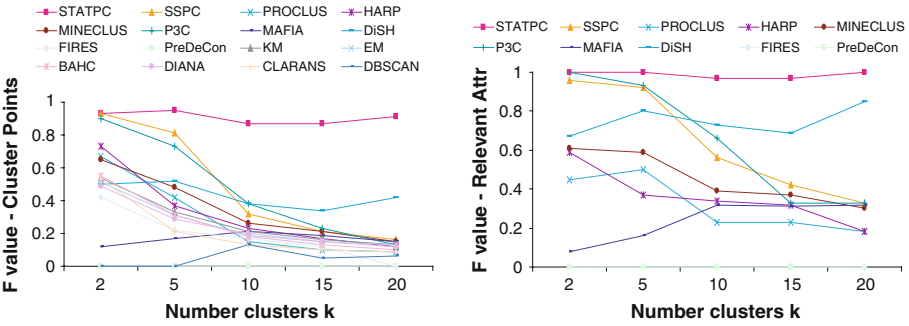

Fig. 7 Effect k

The accuracy of MAFIA also increases with increasing n, because more points in a cluster translate into less wrongly reported 1D clusters.

P3C's accuracy increases significantly with increasing n, because the 1D cluster projections become increasingly detectable, and more evidence is present for their aggregation.

The accuracies of FIRES and of the full-dimensional algorithms are relatively unaffected by increasing n. EM's accuracy slightly increases, because more points in a cluster mean a more reliable covariance matrix for the cluster.

Effect of number of clusters. Figure 7 shows the accuracy of the compared techniques as a function of increasing number of clusters k.

STATPC's accuracy remains constantly high as the number of implanted clusters increases.

The accuracies of the majority of techniques decrease as k increases, because these techniques cannot recover the clusters accurately, and the more clusters are in the data, the more visible this fact is. MAFIA is relatively unaffected but very inaccurate overall.

The full-dimensional algorithms perform approximately the same. However, the number k of true clusters is an input parameter for these algorithms (except DBSCAN).

Effect of extent in relevant attributes. Figure 8 shows the accuracy of the compared techniques as a function of increasing extent in relevant attributes.

The accuracies of all techniques decrease as the extent of clusters in their relevant attributes increases, because clusters will "stand out" less and less in comparison with the uniform background. The tendency of FIRES is again inconclusive.

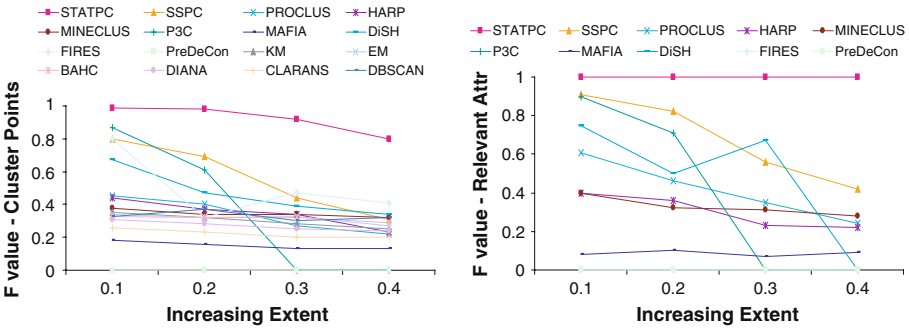

Fig. 8 Effect extent

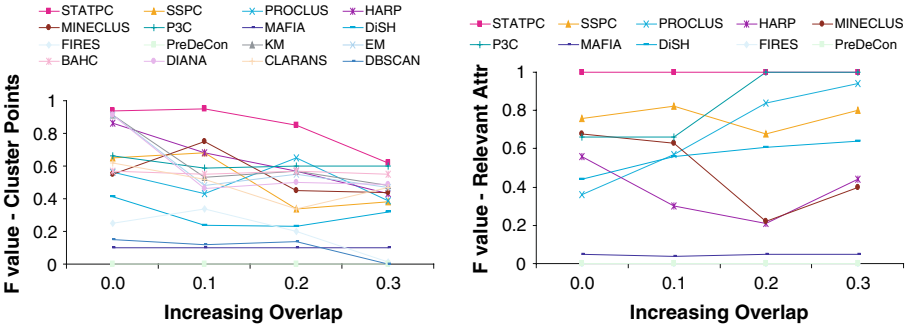

Fig. 9 Effect overlap

Effect of overlap in common relevant attributes. Figure 9 shows the accuracy of the compared techniques as a function of increasing overlap between clusters on common relevant attributes.

The accuracies in terms of cluster points for the majority of the compared techniques decrease as the overlap between clusters on common relevant attributes increases, because clusters become more and more identical. However, the accuracies in terms of relevant attributes of the compared techniques increase, because more overlap translates into more points per cluster, and thus, into a more reliable identification of relevant attributes. MAFIA is relatively unaffected but has very low-accuracy overall.

Accuracy results on real data sets. Figures 10, 11, 12, and 13 show the accuracy of the compared techniques on the Pima Indians Diabetes, Liver Disorders, Iris, and Glass data sets and their extensions, respectively, as a function of increased number of uniform attributes added to the data. The first point in the graphs corresponds to the original data sets with no uniform attributes added.

The accuracies of most competing techniques decrease as the number of uniform attributes added increases, because it becomes more difficult to detect increasingly lower-dimensional clusters. Some of the techniques are not affected by the added number of uniform attributes, such as STATPC, P3C and MAFIA.

The gaps in accuracy between the compared techniques is less pronounced in the case of these real data sets. The reason may be the fact that we evaluate the results on real data by using class labels as cluster labels. As mentioned earlier, class labels are not the "perfect" ground truth in the sense that they do not correspond necessarily to subspace clusters in the

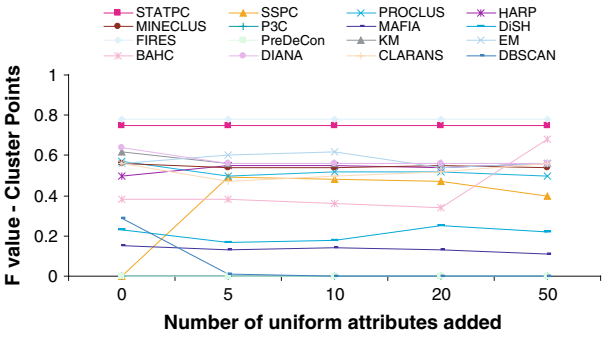

Fig. 10 Accuracy of competing techniques on Pima Indians Diabetes data set

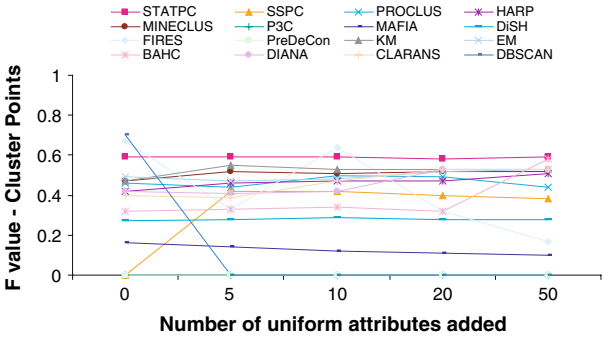

Fig. 11 Accuracy of competing techniques on Liver Disorders data set

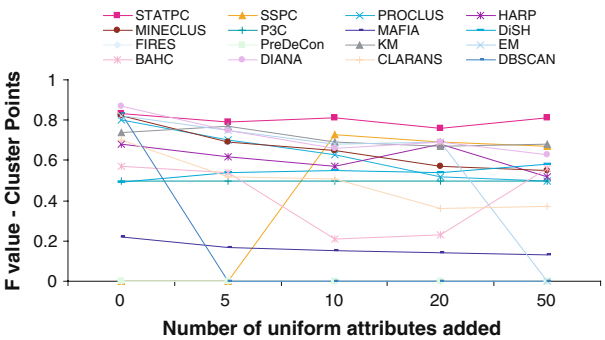

Fig. 12 Accuracy of competing techniques on Iris data set

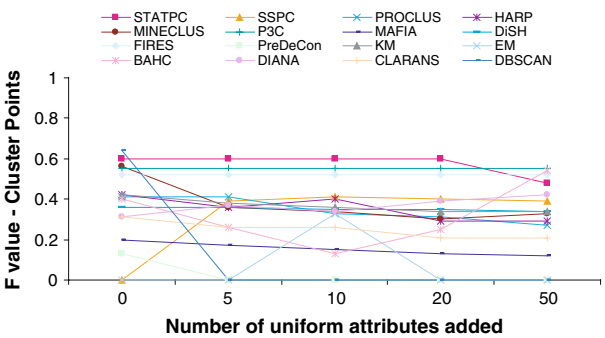

Fig. 13 Accuracy of competing techniques on Glass data set

data. If the class labels would correspond to clusters in the data, then that would translate into easy classification problems, but it is known that these data sets cannot be easily classified. Ideally, the results of a clustering algorithm should be evaluated based on domain knowledge or with the help of a domain expert. But, when the number of algorithms to be evaluated is large, and when many algorithms depend on parameters, such as the number of clusters, whose best values can only be determined through repeated trial-and-error procedures, the evaluation of clustering results becomes tedious. In these situations, class labels provide an acceptable "pseudo"-ground truth that can be used for evaluation.

3.6 Scalability experiments

We measure the scalability of the compared projected and subspace clustering techniques with respect to the database size n, database dimensionality d, and average cluster dimensionality, because these criteria have the largest impact on scalability. We do not need to study the scalability of the full dimensional algorithms because this issue has been studied carefully in the existing literature.

In all scalability figures, the time is represented on a log10 scale. We note that the absolute running times of the compared techniques are influenced by the specific implementations of the techniques and/or hardware used. Therefore, we are interested in the tendencies/slopes of the techniques rather than in a exact characterization of which techniques seem to be the fastest. Run times that differ by a small factor only will look similar in the log plot. However, if the run times differ by orders of magnitude, then there will be a significant difference between them in the plots, which suggests that there is a substantial difference between the techniques that may not be easy to overcome by pure implementation improvements.

Figure 14 shows scalability results for increasing database sizes on synthetic data sets from category *Uniform_Equal* with $d = 10$, $k = 2$, 2 relevant attributes per cluster. Due to memory consumption, HARP, DiSH, FIRES can be run only on the first data set with $n = 10{,}000$ points, and PreDeCon can be run only on the first two data sets with $n = 10{,}000$ and $n = 100{,}000$. We distinguish three groups of techniques based on their run times: STAPC shows the largest run time, followed by the group of SSPC, PROCLUS, and P3C, and then MINECLUS and MAFIA.

Figure 15 shows scalability results for increasing database dimensionality on synthetic data sets from category *Uniform_Equal* with $n = 300$, $k = 2$, 2 relevant attributes per cluster. MINECLUS cannot be run for the last two data sets with $d = 500$ and $d = 1000$ because of memory consumption. Based on their tendencies and the gap between them, the compared techniques can be partitioned into several groups from larger to smaller run times: STATPC, HARP, and FIRES in the first group, followed by P3C and DiSH in the second

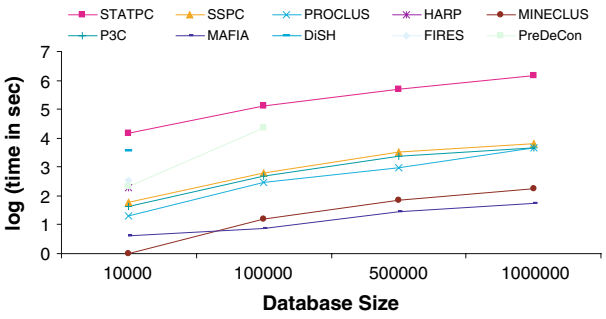

Fig. 14 Scalability with increasing database size

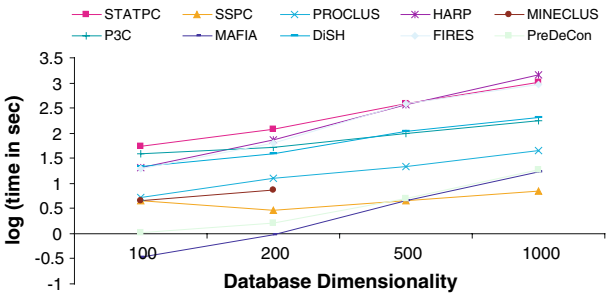

Fig. 15 Scalability with increasing database dimensionality

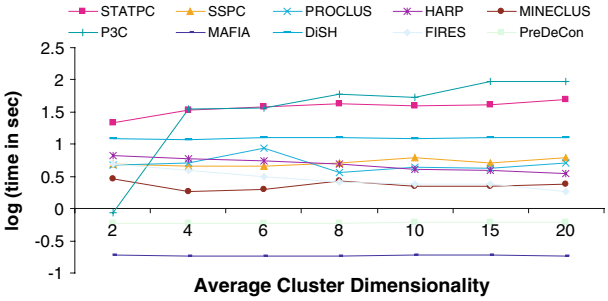

Fig. 16 Scalability with increasing average cluster dimensionality

group, followed by PROCLUS and SSPC in the third group, and finally, the group of MAFIA and PreDeCon.

Figure 16 shows that the majority of the techniques are unaffected by increasing average cluster dimensionality on the data sets from category *Uniform_Equal*. P3C has exponential complexity in the dimensionality of the largest subspace where clusters exist. MAFIA suffers theoretically from the same problem; however, in this case, MAFIA reports only some 1D cluster projections, and thus its run time does not show the expected behavior. MAFIA and PreDeCon form a group of techniques with smallest run times, and P3C and STATPC form a group of techniques with largest run time.

4 Discussion

We summarize our experimental results in Tables 4, 5 and 6. The rows in each table represent the compared techniques. The columns in each table represent criteria studied in data generation. Arrows indicate increase or decrease in accuracy, and $\sim$ indicates constant accuracy. For instance, the first cell in Table 4 should be read as "the accuracy of STATPC increases as the average cluster dimensionality increases". The notation $\uparrow_{Unif}$ signifies that the accuracy of the technique given in the corresponding row is higher when cluster points are uniformly distributed in their relevant subspace than when the cluster points are Gaussian-distributed in their relevant subspace. Similarly, the notation $\uparrow_{Eq}$ means that the accuracy of the technique given in the corresponding row is higher when clusters have an equal number of relevant attributes than when clusters have a different number of relevant attributes.

In general, we observe that the compared techniques show consistent tendencies with respect to the data generation effects studied. All techniques show higher accuracies when cluster points are uniformly distributed in their relevant subspace than when cluster points are

Table 4 Analysis of criteria 1, 2, and 3 in data generation

	Avg. cl. dim. ↑	Unit./Gauss.	Eq./Diff.
STATPC	↑	$\uparrow_{Unif}$	∼
PROCLUS	↑	$\uparrow_{Unif}$	$\uparrow_{Eq}$
PreDeCon	∼	∼	∼
MINECLUS	↑	$\uparrow_{Unif}$	$\uparrow_{Eq}$
HARP	↑	$\uparrow_{Unif}$	$\uparrow_{Eq}$
SSPC	↑	$\uparrow_{Unif}$	$\uparrow_{Eq}$
MAFIA	↑	$\uparrow_{Unif}$	$\uparrow_{Eq}$
P3C	∼	$\uparrow_{Unif}$	$\uparrow_{Eq}$
FIRES	↑	$\uparrow_{Unif}$	$\uparrow_{Eq}$
DiSH	↓	$\uparrow_{Unif}$	$\uparrow_{Eq}$
Full-dim	↑	$\uparrow_{Unif}$	$\uparrow_{Eq}$

Table 5 Analysis of criteria 4, 5, and 6 in data generation

	Db. dim. ↑	Db. size ↑	Numb. cl. ↑
STATPC	∼	↑	∼
PROCLUS	↓	↑	↓
PreDeCon	∼	∼	∼
MINECLUS	↓	↑	↓
HARP	↓	↑	↓
SSPC	↓	↑	↓
MAFIA	↓	↑	∼
P3C	∼	↑	↓
FIRES	∼	↑	↓
DiSH	↑	↑	↓
Full-dim	↓	∼	↓

Gaussian-distributed in their relevant subspace. Also, most techniques obtain higher accuracies when clusters have an equal number of relevant attributes than when clusters have a different number of relevant attributes. Most techniques have higher accuracies when database size or average cluster dimensionality increase. Most techniques have lower accuracies when number of clusters or extent of clusters in relevant attributes or overlap of clusters in common relevant attributes increase.

Our results indicate that the denser and/or the more relevant attributes subspace clusters have, the easier it is for the compared techniques to detect these clusters. If clusters are sufficiently dense and/or have enough relevant attributes, even most of the full-dimensional clustering algorithms can obtain a good accuracy. In such cases, we recommend that the selection of an algorithm should be driven by the trade-off between run time and accuracy.

With respect to the run times of the compared techniques, we have used implementations provided by their authors, or publicly available implementations provided in different packages. However, in this way, we are bound to compare different programming skills and programming languages, which is also why we use log plots to present the run times of the compared techniques. A refined evaluation of the efficiency of the compared techniques

Table 6 Analysis of criteria 7 and 8 in data generation

	Extent ↑	Overlap ↑
STATPC	↓	↓
PROCLUS	↓	↓
PreDeCon	∼	∼
MINECLUS	↓	↓
HARP	↓	↓
SSPC	↓	↓
MAFIA	↓	∼
P3C	↓	∼
FIRES	↓	↓
DiSH	↓	↓
Full-dim	↓	↓

would require implementing all algorithms within a unified framework. Steps in this direction are the Biosphere project [43] and the ELKI framework [2].

The run times of the competing techniques should be considered together with their effectiveness in finding subspace clusters. For instance, STATPC has often a larger run time than the other techniques, but it is also more effective in discovering subspace clusters. It is usually up to the user to specify an acceptable trade-off between run time and quality of the results.

One interesting question is what technique(s) should be selected for applying on real data sets. Obviously, in the case of real data sets, the user cannot know the density or the dimensionality of potential subspace clusters in the data. It may even be the case that the data set does not contain subspace clusters at all. In this case, we recommend that the selection of the technique(s) should be oriented towards technique(s) with less parameters that, in order to be set, do not require crucial knowledge about the data set (such as the number of clusters or the average cluster dimensionality). At the same time, the efficiency of the technique(s) must be taken into account, because some techniques may be inefficient for practical purposes. Finally, we believe that techniques that incorporate statistical principles are preferable to those that do not, because techniques in the first category can guarantee at least that the reported clusters are "unexpected" in the data in a certain way, and they are not just an artifact of the technique.

The current work studies an extensive list of data generation models by studying systematically numerous criteria involved in the data generation. We believe that many of these data generation models correspond to scenarios that can be found in real data, but we are convinced that some real data sets may correspond to models that we have not yet captured in our data generation models. Along these lines, the current study could be further extended by considering other distributions than uniform and Gaussian for clusters in their relevant subspace, or other distribution than uniform for the irrelevant attributes.

Our recommendations are that future research in the area of subspace and projected clustering should focus on techniques that can (1) discover clusters with relatively low density and/or low dimensionality; (2) limit the number of parameters required, as well as avoid parameters that could only be set if the clustering structure was already known, and (3) take into account the statistical significance of the reported clusters.

5 Conclusion

High-dimensional data are commonly encountered in many fields. In such data sets, full-dimensional clustering algorithms break down because of the curse of dimensionality. To deal with these challenges, numerous subspace and projected clustering techniques have been proposed, but no comprehensive evaluation of these techniques exists so far.

In this paper, we have evaluated and analyzed state-of-the-art subspace and projected clustering algorithms under a variety of experimental conditions on synthetic data sets, and we have identified their strengths and weaknesses. We believe that our systematic study can be used by data mining practitioners to decide which techniques are suitable, depending on the problem at hand.

Acknowledgments We thank Kevin Yip and Man Lung Yiu who provided us with the implementation of some of the compared algorithms. This research was supported by the Alberta Ingenuity Fund, the iCORE Circle of Research Excellence and NSERC.

References

1. Achtert E, Böhm C, Kriegel H-P, Kröger P, Müller-Gorman I, Zimek A (2007) Detection and visualization of subspace cluster hierarchies. In: Ramamohanarao K, Krishna P, Mohania M, Nantajeewarawat E (eds) Proceedings of the 12th international conference on database systems for advanced applications (DASFAA), Bangkok, Thailand, 2007, pp 152–163
2. Achtert E, Kriegel H-P, Zimek A (2008) ELKI: a software system for evaluation of subspace clustering algorithms. In: Proceedings of the 20th international conference on scientific and statistical database management (SSDBM), Hong Kong, China
3. Aggarwal C, Hinneburg A, Keim D (2001) On the surprising behavior of distance metrics in high dimensional space. In: Bussche J, Vianu V (eds) Proceedings of the eighth international conference on database theory (ICDT), London, UK, 2001, pp 420–434
4. Aggarwal C, Procopiuc C, Wolf J, Yu P, Park J (1999) Fast algorithms for projected clustering. In: Delis A, Faloutsos C, Ghandeharizadeh, S (eds) Proceedings of the ACM SIGMOD international conference on management of data, Philadelphia, PA, USA, 1999, pp 61–72
5. Aggarwal C, Yu P (2000) Finding generalized projected clusters in high dimensional spaces. In: Chen W, Naughton J, Bernstein P (eds) Proceedings of the ACM SIGMOD international conference on management of data, Dallas, TX, USA, 2000, pp 70–81
6. Agrawal R, Gehrke J, Gunopulos D, Raghavan P (1998) Automatic subspace clustering of high-dimensional data for data mining applications. In: Haas L, Tiwary A (eds) Proceedings of the ACM SIGMOD international conference on management of data, Seattle, WA, USA, 1998, pp 94–105
7. Agrawal R, Srikan R (1994) Fast algorithms for mining association rules in large databases. In: Bocca J, Jarke M, Zaniolo C (eds) Proceedings of the international conference on very large data bases VLDB, Santiago de Chile, Chile, 1994, pp 487–499
8. Ankerst M, Breunig M, Kriegel H-P, Sander J (1999) OPTICS: Ordering points to identify the clustering structure. In: Delis A, Faloutsos C, Ghandeharizadeh S (eds) Proceedings of the ACM international conference on management of data (SIGMOD), Philadelphia, PA, USA, 1999, pp 49–60
9. Assent I, Krieger R, Müller E, Seidl T (2007) DUSC: dimensionality unbiased subspace clustering. In: Proceedings of the seventh international conference on data mining (ICDM), Omaha, NE, USA, 2007, pp 409–414
10. Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) When is "nearest neighbor" meaningful?. In: Beeri C, Buneman P (eds) Proceedings of the seventh international conference on database theory (ICDT), Jerusalem, Israel, 1999, pp 217–235
11. Böhm C, Kailing K, Kriegel H-P, Kröger P (2004) Density connected clustering with local subspace preferences. In: Proceedings of the fourth international conference on data mining (ICDM), Brighton, UK, 2004, pp 27–34
12. Cheng C, Fu A, Zhang Y (1999) Entropy-based subspace clustering for mining numerical data. In: Proceedings of the fifth ACM international conference on knowledge discovery and data mining (SIGKDD), San Diego, CA, USA, 1999, pp 84–93

13. Dempster A, Laird N, Rubin D (1977) Maximum likelihood for incomplete data via the EM algorithm. J R Stat Soc Ser B 39(1):1–38
14. Ertöz L, Steinbach M, Kumar V (2003) Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In: Barbara D, Kamath C (eds) Proceedings of the third SIAM international conference on data mining (SDM), San Francisco, CA, USA, 2003
15. Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis E, Han J, Fayyad U (eds) Proceedings of the second ACM international conference on knowledge discovery and data mining (KDD), Portland, OR, USA, 1996, pp 226–231
16. Gondek D, Hofmann T (2007) Non-redundant data clustering. Knowl Inf Syst 12(1):1–116
17. Han J, Kamber M (2006) Data mining: concepts and techniques, 2nd edn. Academic Press, London
18. Hinneburg A, Aggarwal C, Keim D (2000) What is the nearest neighbor in high dimensional spaces? In: Abbadi A, Brodie M, Chakravarthy, Dayal U, Kamel N, Schlageter G, Whang K (eds) Proceedings of the 26th international conference on very large data bases (VLDB), Cairo, Egypt, 2000, pp 506–515
19. Hinneburg A, Keim D (2003) A general approach to clustering in large databases with noise. Knowl Inf Syst 5(4):387–415
20. Kailing K, Kriegel H-P, Kröger P (2004) Density-connected subspace clustering for high-dimensional data. In: Berry M, Dayal U, Kamath C, Skilicorn D (eds) Proceedings of the fourth SIAM international conference on data mining (SDM), Orlando, FL, USA, 2004, pp 1–11
21. Kaufman L, Rousseeuw P (1990) Finding groups in data: an introduction to cluster analyis. Wiley, New York
22. Kriegel H-P, Kröger P, Renz M, Wurst S (2005) A generic framework for efficient subspace clustering of high-dimensional data. In: Proceedings of the fifth international conference on data mining, Houston, TX, USA, 2005, pp 250–257
23. Kriegel H-P, Kröger P, Zimek A (2009) Clustering high dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Trans Knowl Discov Data 3(1)
24. Li T (2008) Clustering based on matrix approximation: a unifying view. Knowl Inf Syst 17(1):1–133
25. Liu G, Li J, Sim K, Wong L (2007) Distance based subspace clustering with flexible dimension partitioning. In: Proceedings of the 23rd international conference on data engineering (ICDE), Istanbul, Turkey, 2007, pp 1250–1254
26. McQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Fifth Berkeley symposium on mathematics, statistics, and probabilistics, vol 1, 1967, pp 281–297
27. Moise G, Sander J, Ester M (2006) P3C: A robust projected clustering algorithm. In: Proceedings of the sixth international conference on data mining (ICDM), Hong Kong, China, 2006, pp 414–425
28. Moise G, Sander J, Ester M (2008) Robust projected clustering. Knowl Inf Syst 14(3):273–298
29. Moise G, Sander J (2008) Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering. In: Proceedings of the 14th ACM international conference on knowledge discovery and data mining (SIGKDD), Las Vegas, NV, USA, 2008, pp 533–541
30. Nagesh H, Goil S, Choudhary A (2001) Adaptive grids for clustering massive data sets. In: Proceedings of the first SIAM international conference on data mining (SDM), Chicago, IL, USA, 2001, pp 1–17
31. Newman D, Hettich S, Blake C, Merz C (1998) UCI repository of machine learning databases 1998
32. Ng K, Fu A, Wong C (2005) Projective clustering by histograms. IEEE Trans Knowl Data Eng 17(3): 369–383
33. Ng R, Han J(1994) Efficient and effective clustering methods for spatial data mining. In: Bocca J, Jarke M, Zaniolo C (eds) Proceedings of the international conference on very large data bases VLDB, Santiago de Chile, Chile, 1994, pp 487–499
34. Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: a review. ACM SIGKDD Explor Newsl 6(1):90–105
35. Patrikainen A, Meila M (2006) Comparing subspace clusterings. IEEE Trans Knowl Data Eng 18(7): 902–916
36. Procopiuc C, Jones M, Agarwal P, Murali T (2002) A Monte Carlo algorithm for fast projective clustering. In: Franklin M, Moon B, Ailamaki, A (eds) Proceedings of the ACM international conference on management of data (SIGMOD, Madison, WI, USA, 2002, pp 418–427
37. R project http://www.r-project.org/
38. UCI Machine Learning Repository http://archive.ics.uci.edu/ml/
39. Sequeira K, Zaki M (2004) SCHISM: a new approach for interesting subspace mining. In: Proceedings of the fourth international conference on data mining (ICDM), Brighton, UK, 2004, pp 186–193
40. Tang J, Chen J, Fu A, Cheung W (2007) Capabilities of outlier detection schemes in large datasets, framework and methodologies. Knowl Inf Syst 11(1):45–84

41. Yip K, Cheung D, Ng M (2004) HARP: a practical projected clustering algorithm. IEEE Trans Knowl Data Eng 16(11):1387–1397
42. Yip K, Cheung D, Ng M (2005) On discovery of extremely low-dimensional clusters using semi-supervised projected clustering. In: Proceedings of the 21st international conference on data engineering (ICDE), Tokyo, Japan, 2005, pp 329–340
43. Yip K, Qi P, Schultz M, Cheung D, Cheung K (2006) SemBiosphere: a semantic web approach to recommending microarray clustering services. In: Proceedings of the 11th pacific symposium on biocomputing (PSB), Maui, HI, USA, 2006
44. Yiu M, Mamoulis N (2005) Iterative projected clustering by subspace mining. IEEE Trans Knowl Data Eng 17(2):176–189

Author Biographies

Gabriela Moise received her MSc in Computer Science in 2003, and her PhD in Computer Science in 2008, both from University of Alberta, Canada. Her PhD thesis was on subspace clustering in high dimensional data. She is currently an Oracle Systems Analyst at Epcor Utilities Inc., Canada. Her research interests include data mining, databases, bioinformatics, and machine learning.

Arthur Zimek is a postdoc in the database systems and data mining group of Hans-Peter Kriegel at the Ludwig-Maximilians-Universität München, Germany. He holds degrees in bioinformatics, philosophy, and theology, involving studies at universities in Munich, Mainz (Germany), and Innsbruck (Austria). Arthur Zimek presented several tutorials at major venues. He finished his PhD thesis in computer science on "Correlation Clustering" in summer 2008. He received the "Best Paper Honorable Mention Award" at the SIAM Int. Conf. on Data Mining (SDM) together with his co-authors in 2008 and has been selected as runner-up of the "SIGKDD Doctoral Dissertation Award" in 2009. His research interests include data mining for high-dimensional data and structured data especially for bioinformatics applications.

Peer Kröger is an assistant professor in the database and data mining group at the Ludwig-Maximilians-Universität München, Germany. He finished his PhD thesis on clustering moderate-to-high dimensional data in July 2004 and his Habilitation on similarity search and data mining in scientific and multimedia data in January 2009. Since 2001, he contributed more than 60 refereed conference and journal publications and presented several tutorials at major venues. In 2008, he received the "Best Paper Honorable Mention Award" at the SIAM International Conference on Data Mining (SDM) together with his co-authors. Peer Krger constantly serves as a program committee member for major database and data mining conferences like KDD, SSTD, and SSDBM as well as a referee for leading journals such as the VLDB Journal, KAIS, TKDE, and TKDD. His research generally deals with scalable solutions for similarity search and data mining in scientific and multimedia database applications.

Hans-Peter Kriegel is a full professor for database systems and data mining in the Department "Institute for Informatics" at the Ludwig-Maximilians-Universität München, Germany and has served as the department chair or vice chair over the last years. His research interests are in spatial and multimedia database systems, particularly in query processing, performance issues, similarity search, high-dimensional indexing as well as in knowledge discovery and data mining. Kriegel received his MS and PhD in 1973 and 1976, respectively, from the University of Karlsruhe, Germany. Hans-Peter Kriegel has been chairman and program committee member in many international database and data mining conferences. He has published over 250 refereed conference and journal papers, and he received the "SIGMOD Best Paper Award" 1997 and the "DASFAA Best Paper Award" 2006 together with members of his research team.

Jörg Sander received his MS in Computer Science in 1996, and his PhD in Computer Science in 1998, both from the University of Munich, Germany. He worked one year as a post-doctoral fellow at the University of British Columbia, Canada, and joined the University of Alberta, Canada, in 2001 as an Assistant Professor. His research interests include Knowledge Discovery in Databases, especially clustering and data mining–with particular interest in spatial, spatio-temporal and biological applications, and techniques and index structures that improve scalability of basic operations such as similarity search in large databases. (Current information can be found at http://www.cs.ualberta.ca/joerg.)