Finding Non-Redundant, Statistically Significant Regions in High Dimensional Data: a Novel Approach to Projected and Subspace Clustering

Gabriela Moise
Dept. of Computing Science
University of Alberta
Edmonton, AB, T6G 2E8, Canada
gabi@cs.ualberta.ca

Jörg Sander
Dept. of Computing Science
University of Alberta
Edmonton, AB, T6G 2E8, Canada
joerg@cs.ualberta.ca

ABSTRACT

Projected and subspace clustering algorithms search for clusters of points in subsets of attributes. Projected clustering computes several disjoint clusters, plus outliers, so that each cluster exists in its own subset of attributes. Subspace clustering enumerates clusters of points in all subsets of attributes, typically producing many overlapping clusters. One problem of existing approaches is that their objectives are stated in a way that is not independent of the particular algorithm proposed to detect such clusters. A second problem is the definition of cluster density based on user-defined parameters, which makes it hard to assess whether the reported clusters are an artifact of the algorithm or whether they actually stand out in the data in a statistical sense.

We propose a novel problem formulation that aims at extracting axis-parallel regions that stand out in the data in a statistical sense. The set of axis-parallel, statistically significant regions that exist in a given data set is typically highly redundant. Therefore, we formulate the problem of representing this set through a reduced, non-redundant set of axis-parallel, statistically significant regions as an optimization problem. Exhaustive search is not a viable solution due to computational infeasibility, and we propose the approximation algorithm STATPC. Our comprehensive experimental evaluation shows that STATPC significantly outperforms existing projected and subspace clustering algorithms in terms of accuracy.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval—*Clustering*

General Terms

Algorithms

Keywords

Projected clustering, Subspace clustering

1. INTRODUCTION

Seminal research [9] has shown that increasing data dimensionality results in the loss of contrast in distances between data points. Thus, clustering algorithms measuring the similarity between points based on all features/attributes of the data tend to break down in high dimensional spaces.

It is hypothesized [21] that data points may form clusters only when a subset of attributes, i.e., a *subspace*, is considered. Furthermore, points may belong to clusters in different subspaces. Global dimensionality reduction techniques cluster data only in a particular subspace in which it may not be possible to recover all clusters, and information concerning points clustered differently in different subspaces is lost [21].

Therefore, several algorithms for discovering clusters of points in subsets of attributes have been proposed in the literature. They can be classified into two categories: *subspace clustering* algorithms, and *projected clustering* algorithms.

Subspace clustering algorithms search for all clusters of points in *all* subspaces of a data set according to their respective cluster definition. A large number of overlapping clusters is typically reported. To avoid an exhaustive search through all possible subspaces, the cluster definition is typically based on a global density threshold that ensures anti-monotonic properties necessary for an Apriori style search. However, the cluster definition ignores 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.

Projected clustering algorithms 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 on the attributes in Y, but they are "not close" when projected on the remaining attributes. Projected clustering algorithms 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 an algorithm considers. If only k projected clusters are desired, the algorithms typically use an objective function to define what the optimal set of k projected clusters is.

Based on our analysis, we argue that a first problem for both projected and subspace clustering is that their objectives are stated in a way that is not independent of the particular algorithm that is proposed to detect such clusters in the data - often leaving the practical relevance of the detected clusters unclear, particularly since their performance also depends critically on difficult to set parameter values.

A second problem for the most previous approaches is that they assume, explicitly or implicitly, that clusters have some point density controlled by user-defined parameters, and they will (in most cases) report some clusters. However, we have to judge if these clusters "stand out" in the data in some way, or, if, in fact, there are many structures alike in the data. Therefore, a density criterion for selecting clusters should be based on statistical principles.

Contributions and Outline of the paper. Motivated by these observations, we propose a novel problem formulation that aims at extracting from the data axis-parallel regions that "stand out" in a statistical sense. Intuitively, a *statistically significant* region is a region that contains significantly more points than expected. In this paper, we consider the expectation under uniform distribution. The set of statistically significant regions R that exist in a data set is typically highly redundant in the sense that regions that overlap with, contain, or are contained in other statistically significant regions may themselves be statistically significant. Therefore, we propose to represent the set R through a reduced, non-redundant set of (axis-parallel) statistically significant regions that in a statistically meaningful sense "explain" the existence of all the regions in R. We will formalize these notions and formulate the task of finding a minimal set of statistically significant "explaining" regions as an optimization problem. Exhaustive search is not a viable solution because of computational infeasibility. We propose STATPC - an algorithm for 1) selecting a suitable set $R^{reduced} \subset R$ in which we can efficiently search for 2) a smallest set P^* that explains (at least) all elements in $R^{reduced}$. Our comprehensive experimental evaluation shows that STATPC significantly outperforms previously proposed projected and subspace clustering algorithms in the accuracy of both cluster points and relevant attributes found.

The paper is organized as follows. Section 2 surveys related work. Section 3 describes our problem definition. The algorithm STATPC is presented in section 4. Section 5 contains an experimental evaluation of $STATPC$. Conclusions and directions for future work are presented in section 6.

2. RELATED WORK

CLIQUE [4], ENCLUS [11], MAFIA [19], nCluster [17] are grid-based subspace clustering algorithms that use global density thresholds in a bottom-up, Apriori style [5] discovery of clusters. Grid-based subspace clustering algorithms are sensitive to the resolution of the grid, and they may miss clusters inadequately oriented or shaped due to the positioning of the grid. SCHISM [23] uses a variable, statistically-aware, density threshold in order to detect dense regions in a grid-based discretization of the data. However, for the largest part of the search space, the variable threshold equals a global density threshold. SUBCLU [15] is a grid-free approach that can detect subspace clusters with more general orientation and shape than grid-based approaches, but it is also based on a global density threshold. DiSH [1] computes hierarchies of subspace clusters in which multiple inheritance is possible. Algorithmically, DiSH resembles both subspace and projected clustering algorithms. DiSH uses a bottom-up search based on a global density threshold to compute a subspace dimensionality for each data point. These subspace dimensionalities are used to derive a distance between points, which is then used in a top-down computation of clusters. In DUSC [6], a point is initially considered a core point if its density measure is F times larger than the expected value of the density measure under uniform distribution, which does not have anti-monotonic properties, and thus cannot be used for pruning the search space. As a solution, DUSC modifies the definition of a core point so that it is anti-monotonic, which, however, introduces a global density threshold.

Several subspace clustering algorithms attempt to compute a succinct representation of the numerous subspace clusters that they produce, by reporting only the highest dimensional subspace clusters [19], merge similar subspace clusters [23], or organize them hierarchically [1].

Projected clustering algorithms can be classified into 1) k-medoid-like algorithms: PROCLUS [3], SSPC [26]; 2) hypercube based approaches: DOC/FASTDOC [22], MINECLUS [27]; 3) hierarchical: HARP [25]; 4) DBSCAN-like approach: PREDECON [10]; and 5) algorithms based on the assumption that clusters stand out in low dimensional projections: EPCH [20], FIRES [16], P3C [18]. For the algorithms in categories 1) and 2), the problem is defined using an objective function. However, these objective functions are restrictive and/or require critical parameters that are hard to set. The other algorithms do not define what a projected cluster is independent of the method that finds it. P3C takes into account statistical principles for deciding whether two $1D$ projections belong to the same cluster. Many of these algorithms show unsatisfactory performance for discovering low-dimensional projected clusters.

Related to our work is also the work on Scan Statistics [2], in which the goal is to detect spatial regions with significantly higher counts relative to some underlying baseline. The methods in Scan Statistics are applicable to full-dimensional data, whereas our problem formulation concerns statistically significant regions in all subspaces of a data set. Also related is the method PRIM [13], which shares some similarities with DOC and its variants, since it computes one dense axis-aligned box at a time, where the density of the box is controlled by a user-defined parameter. PRIM does not take into account the statistical significance of the computed boxes and often reports many redundant boxes for the same high-density region.

3. PROBLEM DEFINITION

3.1 Preliminary Definitions

Let $D = \{(x_{i1}, \ldots, x_{id}) | 1 \leq i \leq n\}$ be a data set of n d-dimensional data points. Let $A = \{Attr_1, \ldots, Attr_d\}$ be the set of the d attributes of the points in D so that $x_{ij} \in dom(Attr_j)$, where $dom(Attr_j)$ denotes the domain of the attribute $Attr_j$, $1 \leq j \leq d$. Without restricting the generality, we assume that all attributes have normalized domains, i.e., $dom(Attr_j) = [0, 1]$, and we also refer to projections of a point $x_i \in D$ using dot-notation, i.e., if $x_i = (x_{i1}, \ldots, x_{id})$ then $x_i.Attr_j = x_{ij}$.

A ***subspace*** S is a non-empty subset of attributes, $S \subseteq A$. The dimensionality of S, $dim(S)$, is cardinality of S.

An ***interval*** $I = [v_l, v_u]$ on an attribute $a \in A$ is defined as all real values $x \in dom(a)$ so that $v_l \leq x \leq v_u$. The width of interval I is defined as $width(I) := v_u - v_l$. The associated attribute of an interval I is denoted by $attr(I)$.

A ***hyper-rectangle*** H is an axis-aligned box of intervals on different attributes in A, $H = I_1 \times \ldots \times I_p$, $1 \leq p \leq d$, and $attr(I_i) \neq attr(I_j)$ for $i \neq j$. $S = \{attr(I_1), \ldots, attr(I_p)\}$ is the subspace of H, denoted by $subspace(H)$.

Let $H = I_1 \times \ldots \times I_p$ be a hyper-rectangle, $1 \leq p \leq d$. The **volume** of H, denoted by $vol(H)$, is defined as the hyper-volume occupied by H in $subspace(H)$, which is computed as $vol(H) = \prod_{i=1}^{p} width(I_i)$. The *support set* of H, denoted by $SuppSet(H)$, represents the set of database points whose coordinate values fall within the intervals of H for the corresponding attributes in $subspace(H)$, i.e., $SuppSet(H) := \{x \in D | x.attr(I_i) \in I_i, \forall i : 1 \leq i \leq p\}$. The ***actual support*** of H, denoted by $\mathbf{AS(H)}$, represents the cardinality of its support set, i.e., $AS(H) := |SuppSet(H)|$.

3.2 Statistical Significance

Let H be a hyper-rectangle in a subspace S. We use the methodology of statistical hypothesis testing to determine the probability that H contains $AS(H)$ data points under the null hypothesis that the n data points are uniformly distributed in subspace S. The distribution of the test statistic, $AS(H)$, under the null hypothesis is the Binomial distribution with parameters n and $vol(H)$ [7] [1], i.e.,

$$AS(H) \sim Binomial(n, vol(H)) \qquad (1)$$

The significance level α of a statistical hypothesis test is a fixed probability of wrongly rejecting the null hypothesis, when in fact it is true. α is also called the rate of false positives or the probability of type I error.

The critical value of a statistical hypothesis test is a threshold to which the value of the test statistic is compared to determine whether or not the null hypothesis is rejected. For a one-sided test, the critical value θ_α is computed based on

$$\alpha = probab(AS(H) \geq \theta_\alpha) \qquad (2)$$

for a two-sided test, the right critical value θ_α^R is computed by (1), and the left critical value θ_α^L is computed based on

$$\alpha = probab(AS(H) \leq \theta_\alpha^L) \qquad (3)$$

where the probability is computed in each case using the distribution of the test statistic under the null hypothesis.

Definition 1. Let H be a hyper-rectangle in a subspace S. Let α_0 be a significance level. Let θ_{α_0} be the critical value computed at significance level α_0 based on (2), where the probability is computed using $Binomial(n, vol(H))$. H is a ***statistically significant*** hyper-rectangle if $AS(H) > \theta_{\alpha_0}$.

A statistically significant hyper-rectangle H contains significantly more points than what is expected under uniform distribution, i.e., the probability of observing $AS(H)$ many points in H, when the n data points are uniformly distributed in subspace S is less than α_0.

Let α_0 be an initial significance level. A value of $\alpha_0 = 0.001$ is quite common in statistical tests when a single and typically well-conceived hypothesis (i.e., one that has a high chance of being true) is tested; however, the value should be much smaller if the number of possible tests is very large, and we are actually searching for hypotheses that will pass a test; otherwise, a considerable number of false positives will be eventually reported. We will test for statistical significance hyper-rectangles in each subspace of the data set. Thus, the number of false positives increases proportionally to the number of subspaces tested. We can use a conservative Bonferroni approach and adjust the significance level α_0 for testing hyper-rectangles in a subspace of dimensionality p by the total number of subspaces of dimensionality p as $\alpha = \frac{\alpha_0}{choose(d,p)}$, where $choose(d, p)$ is the binomial coefficient $choose(d, p) = \frac{d!}{p!*(d-p)!}$, or we can use the FDR method [8].

An important property of statistical significance is that it is *not* anti-monotonic, i.e., if $H = I_1 \times \ldots \times I_p$ is a statistically significant hyper-rectangle, then a hyper-rectangle $H' = I_{i_1} \times \ldots \times I_{i_k}$, $1 \leq i_j \leq p$, $1 \leq j \leq k$, formed by a subset of the intervals of H, is not necessarily statistically significant. Therefore, Apriori-like bottom-up constructions of statistically significant hyper-rectangles is not possible.

3.3 Relevant vs. irrelevant attributes

Let H be a hyper-rectangle in a subspace S. As the dimensionality of S increases, $vol(H)$ decreases towards 0, and, consequently, the critical value θ_α decreases towards 1. Thus, in high dimensional subspaces, hyper-rectangles H with very few points may be statistically significant.

Also, assume H is a statistically significant hyper-rectangle in a subspace S, and assume that there is another attribute $a \notin S$ where the coordinates of the points in $SuppSet(H)$ are uniformly distributed in $dom(a)$. We could then add the smallest interval $I' = [l, u]$ to H that satisfies $attr(I') = a$ and $SuppSet(I') = H$, i.e., $l = min\{x.a | x \in SuppSet(H)\}$, and $u = max\{x.a | x \in SuppSet(H)\}$. The resulting hyper-rectangle H' will then be statistically significant in subspace $S' = S \cup \{a\}$. We prove this result as follows.

For simplicity of notation, let $p = vol(H)$, $0 \leq p \leq 1$, and let A be the right critical value of the distribution $Binomial(n, p)$ at significance level α_0, as in equation (2). By definition 1, $AS(H) > A$. Let $q = vol(H')$, $0 \leq q \leq 1$. Let B be the right critical value of the distribution $Binomial(n, q)$ at significance level α_0. By the construction of H', it holds that $AS(H) = AS(H')$ and $vol(H') = q \leq p = vol(H)$.

Let X be a Binomial distributed variable with parameters n and p. Then, the probability $Pr(X \geq k)$, $k \in \{0, 1, \ldots, n\}$,

[1] Note that if the attributes are not normalized to $[0, 1]$, we have to replace $vol(H)$ by $vol(H)/vol(S)$.

of obtaining k or more successes in n independent "yes/no" experiments, where each experiment has the probability of success p, is $Pr(X \geq k) = \sum_{i=k}^{n} choose(n,i) * p^i * (1-p)^{n-i}$. Let X' be a Binomial distributed variable with parameters n and q. Then, the probability $Pr(X' \geq k)$, $k \in \{0, 1, \ldots, n\}$, of obtaining k or more successes in n independent "yes/no" experiments, where each experiment has the probability of success q, is $Pr(X' \geq k) = \sum_{i=k}^{n} choose(n,i) * q^i * (1-q)^{n-i}$. Since $q \leq p$, it holds that $Pr(X \geq k) \geq Pr(X' \geq k)$, $\forall k \in \{0, 1, \ldots, n\}$ [12].

From equation (2), it follows that $Pr(X \geq A) = \alpha_0$. But $Pr(X \geq k) \geq Pr(X' \geq k), \forall k \in \{0, 1, \ldots, n\}$; thus, $Pr(X \geq A) \geq Pr(X' \geq A)$. It follows that $Pr(X' \geq A) \leq \alpha_0$. From the definition of the right critical value of a Binomial distribution with parameters n and q at significance level α_0, it holds that $Pr(X' \geq B) = \alpha_0$. Based on $Pr(X' \geq A) \leq \alpha_0$ and $Pr(X' \geq B) = \alpha_0$, it must follow that $B \leq A$.

Since $B \leq A$, it holds that $AS(H') = AS(H) > A \geq B$; thus, by definition 1, H' is also a statistically significant hyper-rectangle at significance level α_0.

Clearly, reporting statistically significant hyper-rectangles such as H' does not add any information, since their existence is "caused" by the existence of other statistically significant hyper-rectangles to which intervals have been added in which the points are uniformly distributed along the whole range of the corresponding attributes.

To deal with these problems, we introduce the concept of "relevant" attributes versus "irrelevant" attributes.

Definition 2. Let H be a hyper-rectangle in a subspace S. An attribute $a \in S$, is called ***relevant*** for H if points in $SuppSet(H)$ are *not* uniformly distributed in $dom(a)$; otherwise it is called ***irrelevant*** for H.

To test whether points in $SuppSet(H)$ are uniformly distributed in the whole range of an attribute a we use the Kolmogorov-Smirnov goodness of fit test for the uniform distribution [24] with a significance level of the test of α_K.

3.4 Redundancy-oblivious problem definition

Given a data set D of n d-dimensional points, we would like to find in each subspace all hyper-rectangles that satisfy definitions 1 and 2. The number of hyper-rectangles in a certain subspace can be infinite. However, we consider, for each subspace, all unique Minimum Bounding Rectangles (MBRs) formed with data points instead of all possible hyper-rectangles. The reason is that adding empty space to an MBR keeps its support constant, but it increases its volume; thus, it only decreases its statistical significance.

Definition 3. Given a data set D of n d-dimensional points. We define a ***subspace cluster*** as an MBR formed with data points in some subspace so that the MBR 1) is statistically significant, and 2) has only relevant attributes.

Redundancy-oblivious problem definition. Find all unique subspace clusters in a set of n d-dimensional points.

For any non-trivial values of n and d, the *size of the search space* for the redundancy-oblivious problem definition is obviously very large. There are $2^d - 1$ subspaces, and the number of unique MBRs in each subspace S, that contain at least 2 points, assuming all coordinates of n points to be distinct in S, is at least $choose(n, 2)$ and upper bounded by $choose(n, 2) + choose(n, 3) + \ldots + choose(n, 2 \times dim(S))$.

The *size of the solution* to the redundancy-oblivious problem definition can be quite large as well, even if the overall distribution is generated by only a few "true" subspace clusters $\{T_1, \ldots, T_k\}$, $k \geq 1$, plus uniform background noise: 1) for each T_i, $1 \leq i \leq k$, other subspace clusters may exist around it in $subspace(T_i)$, formed by subsets of points in $SuppSet(T_i)$ plus possibly neighboring points in $subspace(T_i)$ - Figures 1(a), 1(b) and 1(c) illustrate some of these cases; 2) subspace clusters may also exist in lower or higher-dimensional subspaces of $subspace(T_i)$ due to the existence of T_i - Figure 1(d) illustrates for a true 2-dimensional subspace cluster in the xy-plane an induced 3-dimensional subspace cluster and two 1-dimensional subspace clusters; 3) additional subspace clusters may also exist whose points or attributes belong to different T_i - Figure 1(e) illustrates a subspace cluster induced by two subspace clusters from the same subspace, 1(f) illustrates a subspace cluster induced by two subspace clusters from different subspaces; 4) combinations of all these cases are possible as well, and the number of subspace clusters that exist only because of the "true" subspace clusters is typically very large. For instance, the total number of subspace clusters in even the simple data set depicted in Figure 1(g) —with 50 points and two 2-dimensional subspace clusters, which are embedded in a 3-dimensional space— is 656.

Conceptually, the solution R to the redundancy-oblivious problem definition contains three types of elements: 1) a set of subspace clusters T representing the "true" subspace clusters, 2) a set ϵ representing the false positives reported by the statistical tests, and 3) a set of subspace clusters F representing subspace clusters that exist only because of the subspace clusters in T and possibly ϵ, i.e.,

$$R = T \cup \epsilon \cup F \tag{4}$$

We argue that reporting the entire set R is not only computationally expensive, but it will also overwhelm the user with a highly ***redundant*** amount of information, because of the large number of elements in F.

3.5 Explain relationship

Our goal is to represent the set R of all subspace clusters in a given data set by a reduced set P^{opt} of subspace clusters such that the existence of each subspace cluster $H \in R$ can be *explained* by the existence of the subspace clusters P^{opt}, and P^{opt} should be a smallest set of subspace clusters with that property. Ideally, $P^{opt} = T \cup \epsilon$.

To achieve this goal, we have to define an appropriate *Explain* relationship, which is based on the following intuition. We can think of the overall data distribution as being generated by the "true" subspace clusters, which we hope to capture in the set P^{opt}, plus background noise. We can say

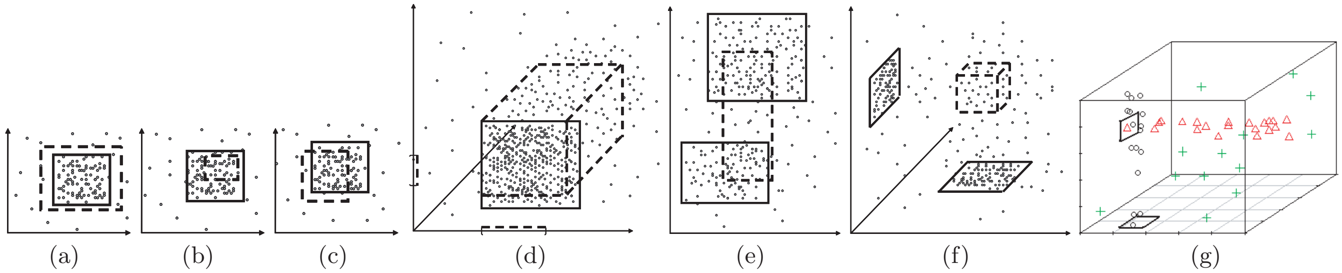


Figure 1: (a)-(f) Redundancy in R (solid/dotted lines for true/"induced" subspace clusters) (g) Example data

that the actual support $AS(H)$ of a subspace cluster H can be *explained* by a set of subspace clusters P, if $AS(H)$ is consistent with the assumption that the data was generated by only the subspace clusters in P and background noise.

More formally, if we have a set $P = \{P_1, ..., P_K\}$ of subspace clusters that should explain all subspace clusters in R, we assume that the overall distribution is a distribution mixture of $K+1$ components, K components corresponding to (derived from) the K elements in P and the $K+1$ component corresponding to background noise, i.e.,

$$f(x;\Theta) = \sum_{k=1}^{K+1} \mu_k f_k(x;\theta_k) \tag{5}$$

where θ_k are the parameters of each component density, and μ_k are the proportions of the mixture.

Conceptually, to justify that an observed subspace cluster H is explained by P, we have to test that the actual support $AS(H)$ of H is *not significantly* larger or smaller than what can be expected under the given model. Again, this can in theory be done using a statistical test, if we can determine left and right critical values for the distribution of the test statistics $AS(H)$, given a desired significance level.

Practically, there are limitations to what can be done analytically to apply such a statistical test. An analytical solution requires to first estimate the parameters and mixing proportions of the model, using the data and information that can be derived from the set P; and then, an equation for the distribution of $AS(H)$ has to be derived from the equation for the mixture model. Obviously, this is challenging (if not impossible) for more complex forms of distributions.

In the following, we show how to define the *Explain* relationship assuming that all component densities are Uniform distributions. Let the $K+1$ component be the uniform background noise in the whole space, i.e.,

$$f_{K+1}(x) \sim Uniform([0,1] \times \ldots \times [0,1]) \tag{6}$$

For the other components, corresponding to $P_k \in P$, we assume that data is generated such that in $subspace(P_k)$, $1 \leq k \leq K$, the points are uniformly distributed in the corresponding intervals of P_k (and uniformly distributed in the whole domain in the remaining attributes, since these are the irrelevant attributes for P_k). Formally, if P_k has m_k relevant attributes, i.e., $P_k = I_1^k \times \ldots \times I_{m_k}^k$, and the d attributes are ordered as $(attr(I_1^k), \ldots, attr(I_{m_k}^k), [0,1] \ldots, [0,1])$, the k-th component density is given by

$$f_k(x) \sim Uniform(I_1^k \times \ldots \times I_{m_k}^k \times [0,1] \times \ldots \times [0,1]) \tag{7}$$

To determine whether the existence of a subspace cluster $H = I_1^H \times \ldots \times I_{m_H}^H$ is consistent with such a model, we have to estimate the possible contribution of each component density to H. For a component density f_k, that contribution is proportional to the volume of the intersection between f_k and H in the subspace of H, i.e., we have to determine the part of f_k that lies in H. This intersection is —like H— an m_H-dimensional hyper-rectangle $\pi_H(P_k)$ that can be computed as following. For f_k, $1 \leq k \leq K$, let $P_k = I_1^k \times \ldots \times I_{m_k}^k$, and for f_{K+1}, i.e. background noise, let $P_{K+1} = [0,1] \times \ldots \times [0,1]$:

$$\pi_H(P_k) = I_1^{\pi_H} \times \ldots \times I_{m_H}^{\pi_H}, \tag{8}$$

where

$$I_i^{\pi_H} = \begin{cases} I_i^H \cap I_j^k & \text{if } \exists j : attr(I_j^k) = attr(I_i^H) \\ I_i^H & \text{else} \end{cases}$$

Because f_k is a uniform distribution, the number of points in $\pi_H(P_k)$ generated by f_k follows a Binomial distribution

$$Binomial(n_k, \frac{vol(\pi_H(P_k))}{vol(P_k)}) \tag{9}$$

with expected value $n_k * \frac{vol(\pi_H(P_k))}{vol(P_k)}$, where n_k is the total number of points generated by f_k, and $\frac{vol(\pi_H(P_k))}{vol(P_k)}$ is the fraction of P_k that intersects H.

The numbers n_k can easily (under our assumptions) be estimated using the total number of points n and the information about the actual supports $AS(P_i)$ of the subspace clusters $P_i \in P$ in the data set. For any of the components f_i, $1 \leq i \leq K+1$, the number of points generated by that component is, according to the data model, equal to the observed number of points in P_i, $AS(P_i)$, minus the contributions n_j of the other components f_j, $j \neq i$, and $P_{K+1} = [0,1] \times \ldots \times [0,1]$ (for the background noise f_{K+1}):

$$n_i = AS(P_i) - \sum_{\substack{1 \leq j \leq K+1 \\ j \neq i}} \frac{vol(\pi_{P_i}(P_j))}{vol(P_j)} * n_j \tag{10}$$

where $\pi_{P_i}(P_j)$ is the intersection of hyper-rectangle P_j with hyper-rectangle P_i as defined in equation (8). The equations in (10) can easily be solved for n_i since (10) is a system of $K + 1$ linear equations in $K + 1$ variables. [2]

We want to say that a set of subspace clusters P, plus background noise, explains a subspace cluster H if the observed number of points in H is *consistent* with this assumption and not *significantly* larger or smaller that expected. From the Binomial distributions (9), we can derive a lower and an upper bound on the number of points in H that could be generated by component density f_k, without this number being statistically significant; these are the left $\theta^L_{\alpha_0}(k)$, respectively right $\theta^R_{\alpha_0}(k)$, critical values of this Binomial distribution, with significance level α_0.

By summing up these bounds for each component density, we obtain a range $[ES^L_H, ES^U_H]$ of the number of points in H that could be accounted for just by the presence of the subspace clusters in P, plus background noise, i.e.,

$$ES^L_H = \sum_{k=1}^{K+1} \theta^L_{\alpha_0}(k) \tag{11}$$

$$ES^U_H = \sum_{k=1}^{K+1} \theta^R_{\alpha_0}(k) \tag{12}$$

If $AS(H)$ falls into this range, we can say that $AS(H)$ is consistent with P, or that P is in fact sufficient to explain the observed number of points in H.

Definition 4. Let $P \cup \{H\}$ be a set of subspace clusters. P *explains* H if and only if $AS(H) \in [ES^L_H, ES^U_H]$.

Property 1. $\{H\} \cup P$ explains H.

Proof. Based on (10), it follows that:

$$AS(H) = n_H + \sum_{1 \le j \le K+1} \frac{vol(\pi_H(P_j))}{vol(P_j)} * n_j \tag{13}$$

Thus, $AS(H)$ is the sum of the expected values of the Binomial distributions given in (9), $\forall k = \{1, \ldots, K+1\}$, plus the expected value of the Binomial distribution $Binomial(H, 1)$, which represents the component H. Since the expected value of a Binomial distribution is between the left and right critical values of the Binomial distribution, it follows that $AS(H) \in [ES^L_H, ES^U_H]$, i.e., $\{H\} \cup P$ explains H.

3.6 Redundancy-aware problem definition

The problem of representing R via a smallest (in this sense non-redundant) set of subspace clusters can now be defined.

Redundancy-aware problem definition. Given a data set D of n d-dimensional points. Let R be the set of all subspace clusters. Find a non-empty set $P^{opt} \subseteq R$ with smallest cardinality $|P^{opt}|$ so that P^{opt} explains H for all $H \in R$.

Note that the optimization problem has always a solution, since R explains H for all $H \in R$, because of Property 1.

We emphasize the fact that the redundancy-aware problem definition avoids shortcomings of existing problem definitions in the literature. First, our objective is formulated through an optimization problem, which is independent of a particular algorithm used to solve it. Second, our definition of subspace cluster is based on statistical principles; thus, we can trust that P^{opt} stands out in the data in a statistical way, and is not simply an artifact of the method.

Enumerating all elements in R in an exhaustive way is computationally infeasible for larger values of n and d. Finding a smallest set of explaining subspace clusters by testing all possible subsets of R has complexity $2^{|R|}$, which is in turn computationally infeasible for typical sizes of R. We ran an exhaustive search on several small data sets where some low-dimensional subspace clusters were embedded into higher dimensional spaces, similar to and including the data set depicted in Figure 1(g). The result set P^{opt} found for these data sets was always containing only the embedded subspace clusters (i.e., we did not even have any false positives in these cases); In Figure 1(g), the two depicted 2-dimensional rectangles indicating the embedded subspace clusters represent in fact the subspace clusters found by the exhaustive search.

4. APPROXIMATION ALGORITHM

Let $P^{opt} = \{T_1, \ldots, T_k\}$ be the solution to the redundancy-aware problem definition. We refer to the subspace clusters in P^{opt} as the "true" subspace clusters. The approximation algorithm STATPC constructs a set $R^{reduced}$ by trying to find true subspace clusters around data points. These data points are called *anchor points*. Ideally, $P^{opt} \subseteq R^{reduced}$. Second, we solve heuristically the optimization problem on $R^{reduced}$ through a greedy optimization strategy and we obtain the solution P^{sol}.

The pseudo-code of STATPC is given in figure 2.

4.1 Finding a true subspace cluster around an anchor point

We want to determine if a true subspace cluster could exist around an anchor point Q. Towards this goal, we select up to 3 *candidate* subspaces around Q, so that, when a true subspace cluster exists around Q, the probability that at least one of the candidate subspaces is relevant for the true subspace cluster is high. Second, for each candidate subspace, we find a locally optimal subspace cluster around Q. Up to 3 locally optimal subspace clusters around Q are detected, and from those, we select the "best" one as the locally optimal subspace cluster between them.

4.1.1 Detecting candidate subspaces

To determine a candidate subspace around Q, we start by building a hyper-rectangle of side width $2 * \delta$ around Q in each $2D$ subspace of the data set. δ is a real number, $\delta \in [0, 0.5]$. These hyper-rectangles are built by taking intervals

[2] When the solution to (10) is not unique or has negative values for some n_i this indicates redundancy in the set P, respectively inconsistency with our assumptions, and we can later use this fact to prune such a candidate set P early.

```
Input: Data set $D = (x_{ij})_{i=\overline{1,n},j=\overline{1,d}}$, parameters $\alpha_0$, $\alpha_K$, $\alpha_H$.
Output: Several, possibly overlapping, subspace clusters, and outliers.
Method:
  1. Build a set $R^{reduced}$:
     (a) Select an anchor point $Q$: either randomly from the set of data points
         that are eligible as anchor points, or based on a previous recommendation, if it exists.
     (b) Detect up to 3 candidate subspaces around $Q$ (figure 4).
     (c) For each candidate subspace, detect a locally optimal subspace cluster around $Q$ (figure 5).
     (d) Between the (up to) 3 subspace clusters detected in steps 1.b) and 1.c), detect a locally
         optimal subspace cluster, store it in $R^{reduced}$, and mark its points as not being
         eligible as anchor points.
     (e) Repeat steps 1.a), 1.b), 1.c), and 1.d) until no data points eligible as anchor points are left.
  2. Solve greedily the redundancy-aware problem on $R^{reduced}$, and obtain a solution $P^{sol}$ (figure 6).
     Points that do not belong to any of the subspace clusters in $P^{sol}$ are declared outliers.
```

Figure 2: Pseudo-code of STATPC

of width δ to the right and width δ to the left of Q on each attribute. If it is not possible to take an interval of length δ to the left, respectively, right of Q, we take the maximum possible interval to the left, respectively, right of Q, and we compensate the difference in length with an interval to the opposite side of Q, i.e., the right, respectively, left of Q.

Subsequently, we rank the $2D$ subspaces in decreasing order of the actual support of these $2D$ hyper-rectangles. Let *Rank* denote this ranking.

Let M be a positive integer, $1 \leq M \leq choose(d, 2)$. Let a_j, $j = \overline{1,d}$ be an attribute of the data set D. Let us assume that we select M pairs of attributes *randomly* from the set of all pairs of attributes. Let X be the random variable that represents the number of occurrences of attribute a_j in the selected M pairs of attributes. There are $choose(d, 2) = \frac{d*(d-1)}{2}$ pairs of attributes, and the total number of pairs of attributes that contain attribute a_j is $(d-1)$. Thus, X is a hyper-geometric distributed variables with parameters $\frac{d*(d-1)}{2}$, $d-1$, M, i.e., the probability that attribute a_j occurs k times in the selected M pairs of attributes, $k \in \{0, 1, \ldots, M\}$ is:

$$Pr(X = k) = \frac{choose(d-1, k) * choose(\frac{d*(d-1)}{2} - (d-1), M-k)}{choose(\frac{d*(d-1)}{2}, M)}$$

This is because there are $choose(\frac{d*(d-1)}{2}, M)$ different samples of size M in the set of all pairs of attributes, and the number of such samples with exactly k pairs of attributes that contain a_j is obtained by multiplying the number of ways of choosing k pairs of attributes that contain a_j from the set of $(d-1)$ pairs of attributes that contain a_j and the number of way of choosing $M-k$ pairs of attributes that do not contain a_j from the set of $\frac{d*(d-1)}{2} - (d-1)$ pairs of attributes that do not contain a_j.

Definition 5. Let d be the dimensionality of the data set D. Let M be a positive integer, $1 \leq M \leq choose(d, 2)$. Let α_H be a significance level. Let θ_{α_H} be the right critical value of the Hyper-geometric distribution with parameters $\frac{d*(d-1)}{2}$, $d-1$, and M, at significance level α_H. An attribute a_j, $j = \overline{1,d}$, is said to *occur more often than expected* in the top M pairs of attributes in the ranking *Rank*, if its number of occurrences in the top M pairs of attributes in the ranking *Rank* is larger than θ_{α_H}.

When one or more true subspace clusters exist around Q, the actual support of the 2-dimensional projections that involve attributes of the true subspace clusters may not be statistically significant, nor higher than the support of some 2-dimensional rectangles formed by uniformly distributed attributes. However, the actual support is likely to be at least in the higher range of possible support values under uniform distribution. This does not mean that the top M pairs consist mostly of relevant attributes, but it means that the frequency with which individual relevant attributes are involved in the top M pairs is likely to be significantly higher than the frequency of a randomly chosen attribute.

We have to decide a value for M. M should take relatively small values, because, as we go down the ranking, eventually all attributes will appear as often as expected.

We observe that, given a fixed significance level value α_H, different values of M result in the same right critical values θ_{α_H} for $Hyper-geometric(\frac{d*(d-1)}{2}, d-1, M)$. For instance, if $\alpha_H = 0.001$, $d = 50$, then $\theta_{\alpha_H} = 2$ for $M \in M^{val} = \{2, 3, 4, 5\}$. This means that, for any value of M in the set M^{val}, we will conclude that an attribute a_j, $j = \overline{1,d}$, occurs more often than expected in the top M pairs of attributes in *Rank*, if it occurs at least θ_{α_H} times in the top M pairs of attributes in *Rank*. Therefore, we shall choose M as the largest value in M^{val}.

In STATPC, in order to be robust to the value of M, and in order to position M at the top of the ranking, we consider three sets of values M_1^{val}, M_2^{val}, and M_3^{val} for M that result in three consecutive critical values $\theta_{\alpha_B} \in \{2, 3, 4\}$. In each case, we set M to the largest value in $(M_i^{val})_{i=\overline{1,3}}$, and we obtain three values $(M_i)_{i=\overline{1,3}}$ for M. In our example, the three values for M are 5, 12, and 20, because $\forall M \in \{2, 3, 4, 5\}$, we obtain $\theta_{\alpha_H} = 2$; $\forall M \in \{6, 7, 8, 9, 10, 11, 12\}$, we obtain $\theta_{\alpha_H} = 3$; and $\forall M \in \{13, 14, 15, 16, 17, 18, 19, 20\}$, we obtain $\theta_{\alpha_H} = 4$.

Definition 5.6 For each $(M_i)_{i=\overline{1,3}}$, let $(A_i)_{i=\overline{1,3}}$ be sets of attributes that occur more often than expected in the top $(M_i)_{i=\overline{1,3}}$ pairs of attributes in the ranking *Rank*. We define the *signaled* attributes as being the attributes most frequent in $(A_i)_{i=\overline{1,3}}$.

For example, if we obtain $A_1 = \{a_1, a_2, a_3\}$ for $M_1 = 5$; $A_2 = \{a_1, a_2, a_3\}$ for $M_2 = 12$; and $A_3 = \{a_1, a_3\}$ for $M_3 = 20$, then the set of signaled attributes is $\{a_1, a_3\}$.

By taking the signaled attributes to be the most frequently occurring attributes in $(A_i)_{i=\overline{1,3}}$, we decrease the probability that a signaled attribute is irrelevant for all true subspace clusters, and we increase the probability that a signaled attribute is relevant for the true subspace cluster to which Q belongs.

Iterative refinement of signaled attributes. Let S^0 be the set of signaled attributes. S^0 may be a subset of a set of relevant attributes for a true subspace cluster around Q, in which case, we would like to extend S^0 with all relevant attributes for the true subspace cluster around Q.

We observe that, if S^0 is a subset of a set of relevant attributes for a true subspace cluster around Q, then, by building a hyper-rectangle W of side width $2*\delta$ around Q in subspace S^0, we capture a certain fraction of the true subspace cluster's points. Based on W, we can determine the attributes a_j, $j = \overline{1,d}$, where the data points in $SuppSet(W)$ are *not* uniformly distributed. Let S^1 be the set of these attributes. The more cluster points W captures, the more likely that S^1 consists of attributes relevant for the true subspace cluster around Q.

Based on this observation, we obtain a *candidate subspace* around Q through an iterative refinement of S^0, as follows. If S^0 is *not* included in S^1, the candidate subspace is the empty set $\emptyset$. If $S^0 = S^1$, the candidate subspace is S^0. Otherwise, we repeat the same procedure for S^1 as for S^0 until no more attributes can be added.

Commit to a candidate subspace or recommend the next anchor point. Let $S = S^{iter}$, $iter \geq 1$, $S \neq \emptyset$, be the candidate subspace determined by the iterative refinement of a set of signaled attributes S^0. Let us consider a hyper-rectangle W of side width $2*\delta$ around Q in subspace S^{iter-1}. Clearly, by the construction of the set S, the data points in $SuppSet(W)$ are not uniformly distributed on each attribute $a_j \in S$.

For each attribute $a_j \in S$, we would like to detect the 1D regions that are responsible for the fact that the data points in $SuppSet(W)$ are not uniformly distributed on a_j. There is at least one such 1D region on each attribute $a_j \in S$, due to the construction of S. In addition, Q may or may not belong to these 1D regions. Figure 3 illustrates such a case.

In figure 3, Q is an anchor point, and the set of signaled attributes S^0 for Q is $S^0 = \{a_1\}$. W is a hyper-rectangle of side width $2*\delta$ around Q in subspace $S^0 = \{a_1\}$. Through the iterative refinement, S^0 is extended into the candidate subspace $S = \{a_1, a_2\}$, because the points in $SuppSet(W)$ are not uniformly distributed on a_1 and on a_2. The 1D regions that are responsible for the fact that the points in $SuppSet(W)$ are not uniformly distributed on a_1 and on a_2 are depicted as bold lines on a_1 and on a_2. Q belongs to such a 1D region on the signaled attribute a_1, but it does not belong to such a 1D region on the attribute a_2.

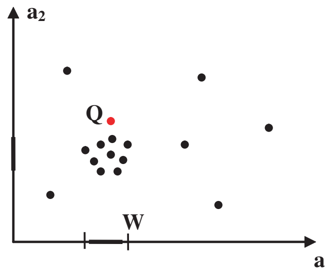


Figure 3: The issue of committing to a candidate subspace

As exemplified in figure 3, there are cases when the candidate subspace contains a subspace cluster of interest, i.e., a possible true subspace cluster, and Q is placed in this subspace in the vicinity of this subspace cluster. If we keep this subspace as a candidate subspace, then, in the next step, STATPC computes a locally optimal subspace cluster around Q in this subspace. Because of the positioning of Q with respect to the subspace cluster, the locally optimal subspace cluster around Q in this subspace will be a subspace cluster that includes Q and some of the cluster points.

As explained in the next subsections, the locally optimal subspace cluster around Q in this subspace may be stored in $R^{reduced}$, in which case, its points cannot be selected as eligible anchor points anymore. This means that the chance of selecting an anchor point that is centrally located in the subspace cluster of interest has decreased. This also means that the chance of having the subspace cluster of interest in $R^{reduced}$ has decreased.

Thus, if a situation like the one illustrated in figure 3 is detected, we *do not commit* to the candidate subspace, i.e., we do not keep this subspace as a candidate subspace. Furthermore, we are able to *recommend* the next anchor point as an eligible data point located centrally in the subspace cluster of interest. In this way, we increase the chance to put in $R^{reduced}$ the subspace cluster of interest when considering the next anchor point.

Concretely, we decide whether to commit or not to a candidate subspace as follows. Let a_j be an attribute so that the data points in $SuppSet(W)$ are not uniformly distributed on a_j. We detect the 1D region(s) where the data points in $SuppSet(W)$ are not uniformly distributed on a_j using a methodology similar to the one used in P3C to detect cluster projections.

First, we divide attribute a_j into $\lfloor 1 + log_2(AS(W)) \rfloor$ bins, by the Sturge's rule [24]. Second, we compute, for each bin, how many points from $SuppSet(W)$ it contains, and compare this number with the expected number of points in a bin if the points in $SuppSet(W)$ were uniformly distributed across all these bins. If a bin has more points than expected, then the bin is marked. Finally, adjacent marked bins are merged

```
*Input:* Data set $D = (x_{ij})_{i=\overline{1,n}, j=\overline{1,d}}$, parameter $\alpha_H$, anchor point $Q$.
*Output:* Up to 3 candidate subspaces around $Q$.
*Method:*

  1. For each $\delta \in \{0.05, 0.1, 0.15\}$:
     (a) Build a hyper-rectangle of side width $2 * \delta$ around $Q$ in each 2D subspace of the data set.
     (b) Rank the 2D subspaces in decreasing order of the actual support of the 2D hyper-rectangles built at step 1.a).
     (c) Let $\theta_{\alpha_H}$ be the right critical value of distribution $Hyper - geometric(\frac{d*(d-1)}{2}, d-1, M)$ at significance level $\alpha_H$. Determine the largest values $M_1, M_2, M_3$ for which the corresponding critical values $\theta_{\alpha_H}$ are, respectively, 2, 3 and 4.
     (d) For each $(M_i)_{i=\overline{1,3}}$, determine a set of attributes $(A_i)_{i=\overline{1,3}}$ that *occur more often than expected* in the top $M_i$ pairs in the ranking computed in step 1.b), i.e., attributes that occur more than the corresponding $\theta_{\alpha_H}$ times in the top $M_i$ pairs in the ranking computed in step 1.b).
     (e) Compute the set of *signaled* attributes $S^0$ as the attributes most frequent in $(A_i)_{i=\overline{1,3}}$.
     (f) Refine iteratively $S^0$ and obtain a candidate subspace $S$.
     (g) Decide whether to commit to candidate subspace $S$, and if not, recommend the next anchor point.
```

Figure 4: Pseudo-code of detecting candidate subspaces around an anchor point Q

into 1D regions.

If there exists at least one attribute a_j in the candidate subspace S so that Q does *not* belong to one 1D region, computed as above, on this attribute, then, we conclude that, if there were a subspace cluster of interest in the candidate subspace, then Q is placed in its vicinity, and thus we *do not commit* to this candidate subspace.

When we do not commit to a candidate subspace, we can recommend the next anchor point. The 1D regions identified as described above form a multi-dimensional region in the candidate subspace. We recommend as the next anchor point the eligible data point that is closest, in terms of Manhattan distance in this subspace, to the centroid of the multi-dimensional region formed with 1D regions.

So far, we have detected a candidate subspace around Q given a certain value for δ. There is no "best" value for δ, and to improve our chances of detecting a true subspace cluster if it exists around an anchor point, we simply try the 3 values 0.05, 0.1, 0.15 for δ. The candidate subspaces detected for different values of δ may be identical or they may be the empty set $\emptyset$; thus, we detect *up to* 3 candidate subspaces.

4.1.2 Detecting a locally optimal subspace cluster

Let S be a candidate subspace determined in the previous step. We want to determine if a true subspace cluster around Q exists in S. There could be many MBRs in S that include Q; however, if there were a true subspace cluster around Q in S, we would like to capture it with the MBRs around Q.

Thus, starting from Q, we build a series of at most $(n-1)$ MBRs in S, by adding one point at a time to the current MBR, i.e., the data point with smallest MINDIST to the current MBR. MINDIST [3] is the popular distance between a data point and an MBR used in index structures [14]. For efficiency reasons, and because a cluster contains typically only a fraction of the total number of points, we build $0.3*n$ MBRs around Q in subspace S.

Let R^{local} be the set of MBRs built in this way that are also subspace clusters. If $R^{local} = \emptyset$, no true subspace cluster around Q in S could be found.

Out of all subspace clusters in R^{local}, we search for the subspace cluster that is *locally optimal* in the sense that it explains more subspace clusters in R^{local} than any other subspace cluster in R^{local}.

We compute for each subspace cluster in R^{local}, the number of subspace clusters in R^{local} that it explains. However, assuming that there exists a true subspace cluster around Q in S, there may be several subspace clusters in R^{local}, which may differ only in a few points, and which explain the same maximum number of subspace clusters in R^{local}.

We would like to differentiate between such subspace clusters. For this, we need, instead of the current binary *Explain* relationship, an *Explain* relationship that can distinguish various degrees of being explained.

This can be easily achieved when the explaining set P consists of just one subspace cluster, as in the current context. Let P_1 be a subspace cluster that represents the explaining set and let H be a subspace cluster to be explained. Under the same data model assumptions as for the *Explain* relationship, we can define the *expected support* of a subspace cluster H assuming a single subspace cluster P_1, as:

$$ES(H|P_1) = n_{P_1} * \frac{vol(H \cap P_1)}{vol(P_1)} + (n - n_{P_1}) * vol(H) \quad (14)$$

where n_{P_1} is the number of points generated by the density component associated with P_1, obtained by solving equation (10) for P_1 and background noise.

Then, we introduce the notion of *quality of explanation* $QualityExplain : R^{local} \text{ x } R^{local} \to [0,1]$, so that:

$$QualityExplain(P_1, H) := 1 - \frac{|AS(H) - ES(H|P_1)|}{max(AS(H), ES(H|P_1))}, \; P_1, H \in R^{local}$$

$QualityExplain$ represents the relative difference between

[3] If $l = (l_1, \dots, l_d)$ and $u = (u_1, \dots, u_d)$ are the left-most, respectively, right-most corners of a d-dimensional MBR M, then MINDIST between a d-dimensional point $P = (p_1, \dots, p_d)$ and the MBR M is the square of $\sum_{i=1}^{d}(p_i - r_i)^2$, where $r_i = l_i$, if $p_i < l_i$; $r_i = u_i$, if $p_i > u_i$; $r_i = p_i$, else.

```
Input: Data set $D = (x_{ij})_{i=\overline{1,n},j=\overline{1,d}}$, parameter $\alpha_0$, $\alpha_K$,
anchor point $Q$, candidate subspace $S$.
Output: A subspace cluster around $Q$ in subspace $S$.
Method:
  1. Build $0.3 * n$ MBRs around $Q$ in subspace $S$ by
     adding, one at a time, the data point with smallest
     MINDIST to the current MBR.
  2. Build $R^{local}$ as the set of MBRs constructed in step
     1) that are subspace clusters.
  3. Choose as the locally optimal subspace
     cluster around $Q$ in $S$ the subspace
     cluster $P^{local} \in R^{local}$ that maximizes
     $\sum_{H \in R^{local}} QualityExplain(P^{local}, H)$.
```

Figure 5: Pseudo-code of detecting a locally optimal subspace cluster around an anchor point Q in a candidate subspace S.

the actual support $AS(H)$ of H and the estimated support $ES(H|P_1)$ of H given the subspace cluster P_1 and the background noise. Equivalently, $QualityExplain(P_1, H)$ can be written as:

$$QualityExplain(P_1, H) = \frac{AS(H)}{ES(H|P_1)}, \, if \; AS(H) < ES(H|P_1)$$

$$QualityExplain(P_1, H) = \frac{ES(H|P_1)}{AS(H)}, \, if \; AS(H) \geq ES(H|P_1)$$

The closer $AS(H)$ and $ES(H|P_1)$ are, the closer $QualityExplain$ is to 1, and the better the quality of explanation.

Consequently, we choose as the locally optimal cluster around Q in S the subspace cluster $P^{local} \in R^{local}$ that maximizes $\sum_{H \in R^{local}} QualityExplain(P^{local}, H)$. Ties may be possible with the $QualityExplain$ relationship too, but with a much lower probability. If there are however ties, we choose one of the subspace clusters in the tie at random.

4.1.3 *Detecting a locally optimal subspace cluster between locally optimal subspace clusters in candidate subspaces*

For a given anchor point Q, let R^{all_local} be the set of all locally optimal subspace clusters around Q detected in previous steps. Since we determine up to 3 candidate subspaces around Q, and in each one of these candidates subspaces, we determine up to 1 locally optimal subspace cluster around Q, it holds that $|R^{all\_local}| \leq 3$.

We regard as the “true” subspace cluster around Q the subspace cluster $P^{all\_local} \in R^{all\_local}$ that explains more subspace clusters in R^{all_local} than any other subspace cluster in R^{all_local}. Formally, the “true” subspace cluster around Q is the subspace cluster $P^{all\_local} \in R^{all\_local}$ that maximizes $\sum_{H \in R^{all\_local}} QualityExplain(P^{all\_local}, H)$.

4.2 Greedy optimization

```
Input: Data set $D = (x_{ij})_{i=\overline{1,n},j=\overline{1,d}}$, parameter $\alpha_0$, set
$R^{reduced}$.
Output: A solution $P^{sol} \subseteq R^{reduced}$ so that
$Explain(P^{sol}, H) = 1, \forall H \in R^{reduced}$.
Method:
1. Initialization: $P^{sol} := \emptyset$; $Cand := R^{reduced}$.
2. Greedy optimization:
While ($Cand \neq \emptyset$)
Choose $H_{i_*} \in Cand$ so that $NrExplained(P^{sol} \cup
\{H_{i_*}\}) = max_{H_i \in Cand} NrExplained(P^{sol} \cup H_i)$.
$P^{sol} := P^{sol} \cup \{H_{i_*}\}$.
$Cand := R^{reduced} \setminus Explained(P^{sol})$.
End while
```

Figure 6: Pseudo-code of detecting greedily a solution P^{sol} on $R^{reduced}$.

Section 4.1 of STATPC tries to find true subspace clusters around anchor points. The points of a “true” subspace cluster already found around an anchor point cannot be selected as anchor points for finding future true subspace clusters. The first anchor point is selected randomly. Subsequent anchor points are selected randomly from the eligible anchor points left. Building $R^{reduced}$ terminates when no data point can be selected as the next anchor point.

Although $|R^{reduced}| < |R|$, solving the optimization problem on $R^{reduced}$ by testing all its possible subsets is still unacceptable for practical purposes. Thus, we construct *greedily* a set P^{sol} that explains all subspace clusters in $R^{reduced}$, but may not be the smallest set with this property.

We build P^{sol} by adding one subspace cluster at a time from $R^{reduced}$. At each step, let $Cand$ be the set of subspace clusters in $R^{reduced}$ that are not explained by the current P^{sol}. Thus, subspace clusters in $Cand$ can be used to extend P^{sol} furthermore, until P^{sol} explains all subspace clusters in $R^{reduced}$. Initially, $P^{sol} = \emptyset$ and $Cand = R^{reduced}$. At each step, we select in P^{sol} greedily the subspace cluster $H_{i_*} \in Cand$ for which $P^{sol} \cup \{H_{i_*}\}$ explains more subspace clusters in $R^{reduced}$ than any other $P^{sol} \cup \{H_i\}$, for $H_i \in Cand$. We stop when $Cand$ is void.

Because of Property 1, set $Cand$ cannot include a subspace cluster that has been already selected in P^{sol}. Thus, $Cand$ is guaranteed to become void, and the optimization strategy is guaranteed to end.

The pseudo-code of the greedy optimization is given in figure 6. For $\forall P \in PowerSet(R^{reduced})$, we define $Explained(P)$ as the set of subspace clusters in $R^{reduced}$ that are explained by P, i.e., $Explained(P) := \{H \in R^{reduced} | Explain(P, H) = 1\}$. We also define $NrExplained(P)$ as be the number of subspace clusters in $R^{reduced}$ that are explained by P, i.e., $NrExplained(P) := |Explained(P)|$.

5. EXPERIMENTAL EVALUATION

The experiments reported in this section were conducted on a Linux machine with 3 GHz CPU and 2 GB RAM.

Synthetic Data.[4] We generated data with $n = 300$ points, $d = 50$ attributes, $k = 5$ subspace clusters (with sizes 60, 50, 40, 40, and 50 points), plus 60 uniformly distributed noise points. The distribution of points in subspace clusters can be 1) uniform or 2) Gaussian. Subspace clusters can have 1) an equal or 2) a different number of relevant attributes. Thus, we obtained 4 categories of data for which we generated data sets with average numbers of relevant attributes 2, 4, 6, 8, 10, 15, and 20. The width of clusters in relevant attributes was 10% – 30% of the attribute range. Clusters did not overlap in common relevant attributes.

Real Data. We test the performance of the compared algorithms on the following data sets from the UCI machine learning repository [5]: Pima Indians Diabetes (768 points, 8 attributes, 2 classes); Liver Disorders (345 points, 6 attributes, 2 classes); and Wisconsin Breast Cancer Prognostic (WPBC)(198 points, 34 attributes, 2 classes).

Experimental setup. We evaluate STATPC against projected clustering algorithms PROCLUS, ORCLUS, HARP, SSPC, MINECLUS, P3C; subspace clustering algorithm MAFIA; and the method PRIM. For MINECLUS, HARP, SSPC and P3C we used the original implementations. PROCLUS and ORCLUS were provided by the SemBiosphere project[6]. PRIM is available as a package[7] for the R statistical software. MAFIA was implemented by us.

We also compared STATPC against a representative set of full dimensional clustering algorithms: KMeans, EM, CLARANS, agglomerative (BAHC) and divisive (DIANA) hierarchical clustering, and DBSCAN.

On synthetic data, we set the target number of clusters to the number of implanted clusters for PROCLUS, ORCLUS, HARP, SSPC, and MINECLUS. PROCLUS and ORCLUS require the average cluster dimensionality, which was set to the known average cluster dimensionality. HARP requires the maximum percentage of outliers, which was set to the known percentage of outliers. For algorithms that require other parameter settings, we set these parameters as recommended by their authors: for PROCLUS: $A = 20$, $B = 5$; for ORCLUS: $\alpha = 0.5$, $k_0 = 30$, $srr = 10$; for SSPC: $m = 0.5$; for MINECLUS: $w = 0.3$, $\alpha = 0.1$, $\beta = 0.25$, $maxout = 20$; 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 PRIM: $peel\_alpha = 0.05$, $paste\_alpha = 0.01$, $mass\_min = 0.1$. SSPC was run without the supervision option. Except HARP, P3C, and MAFIA, all algorithms are non-deterministic; thus, each of them is run 5 times, and the results are averaged. STATPC, P3C, and MAFIA allow data points to belong to more than one cluster; the other algorithms compute disjoint clusters.

STATPC requires 3 significance levels: α_0, α_K, and α_H. After testing the sensitivity of STATPC to these parameters (see Figure 7), we set $\alpha_0 = 1.0E-10$, $\alpha_K = \alpha_H = 0.001$.

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

The real data sets used were collected for classification purposes. We use such real data sets because a systematic evaluation of the compared algorithms on unlabeled data is cumbersome. 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 algorithms to find subspace clusters, we add 5, 10, 20, respectively 50 attributes to each real data set where the data points are uniformly distributed. 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.

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 points. The *precision* of output cluster i is defined as the number of points common to i and j^i divided by the total number of points in i. The *recall* of output cluster i is defined as the number of points common to i and j^i divided by the total number of 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 (except for ORCLUS, since it generates general sets of orthogonal vectors).

Statistical significance of results. STATPC computes subspace clusters that are statistically significant. The other algorithms sometimes compute statistically significant subspace clusters, other times they do not, depending on parameter values and on the density of the implanted clusters (denser clusters are easier to detect). The classes in the real data sets form statistically significant clusters, and these clusters stay statistically significant when adding uniform attributes, as shown in section 3.3.

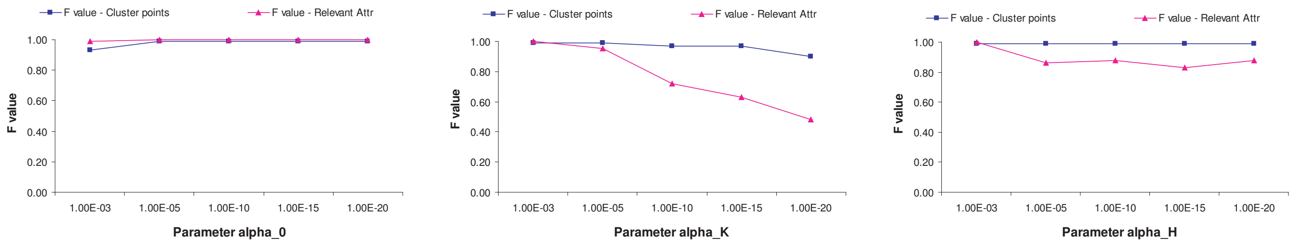
Accuracy results. Figure 8 shows the accuracy of the compared algorithms as a function of increased average cluster dimensionality for the category $Uniform\_Equal$, where the cluster points are *uniformly* distributed in their relevant subspace, and the clusters have an *equal* number of relevant attributes. Figures 9, 10, 11 illustrate the accuracy of the compared algorithms as a function of increased average cluster dimensionality for the categories $Uniform\_Different$, $Gaussian\_Equal$, and $Gaussian\_Different$. We observe that STATPC significantly and consistently outperforms the competing algorithms, both in terms of clustering accuracy and in terms of accuracy of the found relevant attributes. STATPC obtains an improvement in clustering accuracy and accuracy of the found relevant attributes over the best other

[4] Available at `http://www.cs.ualberta.ca/~gabi/SData/`

[5] `http://archive.ics.uci.edu/ml/`

[6] `yeasthub2.gersteinlab.org/sembiosphere`

[7] `http://cran.r-project.org/web/packages/prim/`



(a) (b) (c)

Figure 7: Sensitivity to (a) α_0 (b) α_K (c) α_H

competing algorithm of up to 30%, and respectively, 34%. The difference in accuracy between STATPC and previous algorithms is more pronounced for the more difficult case of data sets with low-dimensional subspace clusters. As the number of relevant attributes increases, the accuracy of competing algorithms increases as well, since the clusters become more easily recognizable in full-dimensional space.

Equal versus different number of relevant attributes for subspace clusters does not have an impact on the accuracy of STATPC. The accuracy results of STATPC on data sets where cluster points are uniformly distributed in their relevant subspace is slightly higher than when the cluster points are Gaussian distributed in their relevant subspace. This is because in the later case, STATPC may miss some points at the clusters' boundaries. We observe that the full dimensional clustering algorithms do not perform well for the task of extracting subspace clusters.

We have also studied systematically the accuracy of STATPC as a function of different data generation parameters.

Figure 12 shows accuracy results for increasing database dimensionality d on synthetic data sets from category $Uniform\_Equal$ with $n = 300$, $k = 5$ (60, 50, 40, 40, 50 cluster points, and 60 uniformly distributed noise points), and 4 relevant attributes per cluster. STATPC is unaffected by increasing database dimensionality, whereas the accuracy of the other algorithms decreases. This is because the clusters become increasingly lower dimensional as d increases, and the competing algorithms have difficulties with low dimensional projected clusters.

Figure 13 shows accuracy results for increasing database size n on synthetic data sets from category $Uniform\_Equal$ with $d = 50$, $k = 5$, 4 relevant attributes per cluster. 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 $n = 1000$: 200, 170, 140, 140, 170 cluster points and 180 noise points; for $n = 2000$, 400, 340, 280, 280, 340 cluster points and 360 noise points. The accuracies of all algorithms increases as n increases, because clusters become more easily discernible.

Figure 14 shows accuracy results for increasing number of clusters on synthetic data sets from category $Uniform\_Equal$ with $n = 300$, $d = 50$, 4 relevant attributes per cluster. Cluster sizes and number of noise points are as follows: for $k = 1$: 125, 125 cluster points and 50 noise points; for $k = 3$: 83, 83, 84 cluster points and 50 noise points; for $k = 4$: 62, 63, 63, 62 cluster points and 50 noise points; for $k = 5$; 50, 50, 50, 50, 50 cluster points and 50 noise points. The accuracy of STATPC is unaffected by increasing number of clusters. In contrast, the accuracy of competing algorithms decreases, because these algorithms are not effective, and this fact is even more pronounced when more clusters have to be discovered.

Figure 15 shows accuracy results for increasing cluster sizes (and consequently, decreasing number of noise points) on synthetic data sets from category $Uniform\_Equal$ with $n = 300$, $d = 50$, $k = 5$, 4 relevant attributes per cluster. Cluster sizes and number of noise points are as follows: 1) 40, 30, 20, 20, 30 cluster points and 160 noise points; 2) 50, 40, 30, 30, 40 cluster points and 110 noise points; 3) 60, 50, 40, 40, 50 cluster points and 50 noise points; 4) 65, 55, 45, 45, 55 cluster points and 35 noise points; 5) 72, 62, 52, 52, 62 cluster points and 0 noise points. The accuracy of all algorithms increases with increasing cluster sizes because the clusters become denser, and thus, more easily recognizable.

Figure 16 shows accuracy results for increasing extent in relevant attributes on synthetic data sets from category $Uniform\_Equal$ with $n = 300$, $d = 50$, $k = 5$ (60, 50, 40, 40, 50 cluster points and 50 noise points), 4 relevant attributes per cluster. The data sets are characterized by clusters with 0.1, 0.2, 0.3, respectively 0.4 extent in the relevant attributes. The accuracy of all algorithms decreases with increasing extent in relevant attributes because the clusters become sparser, and thus less statistically significant.

Figure 17 shows accuracy results for increasing overlap between clusters on common relevant attributes on synthetic data sets from category $Uniform\_Equal$ with $n = 300$, $d = 50$, $k = 2$ (125, 125 cluster points and 50 noise points), 2 relevant attributes per cluster. The two clusters are characterized by an overlap of 0, 0.1, 0.2, respectively 0.3 overlap in common relevant attributes. The accuracy all algorithms decreases with increasing overlap in common relevant attributes because the clusters become more and more identical.

Figure 18 shows the accuracy of the compared algorithms on the 3 real data sets, 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 at-

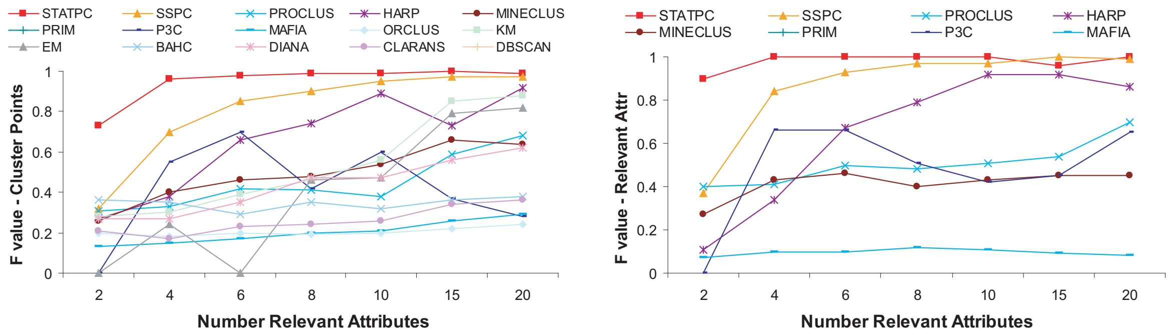


Figure 8: Category Uniform_Equal

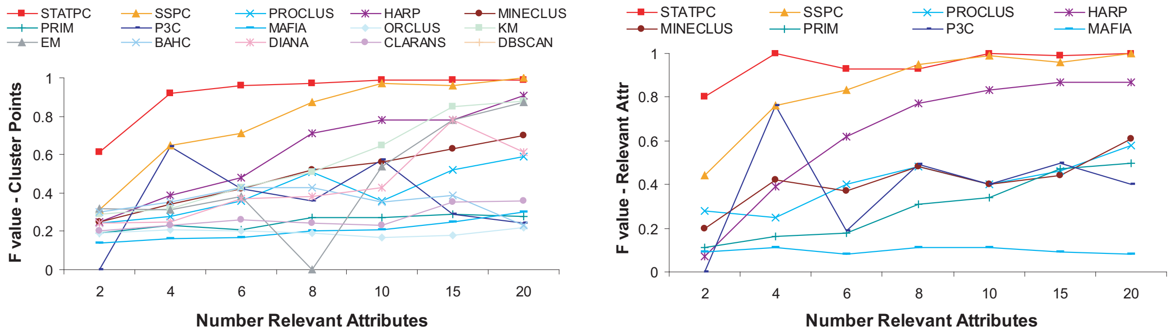


Figure 9: Category Uniform_Different

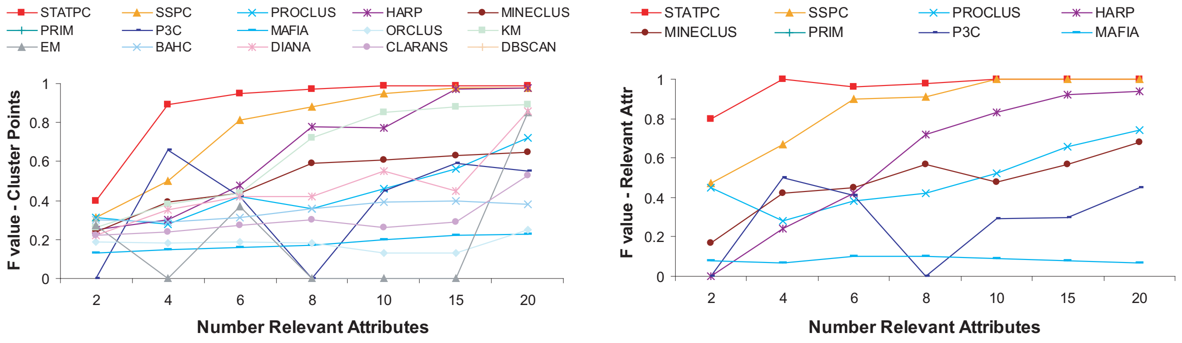


Figure 10: Category Gaussian_Equal

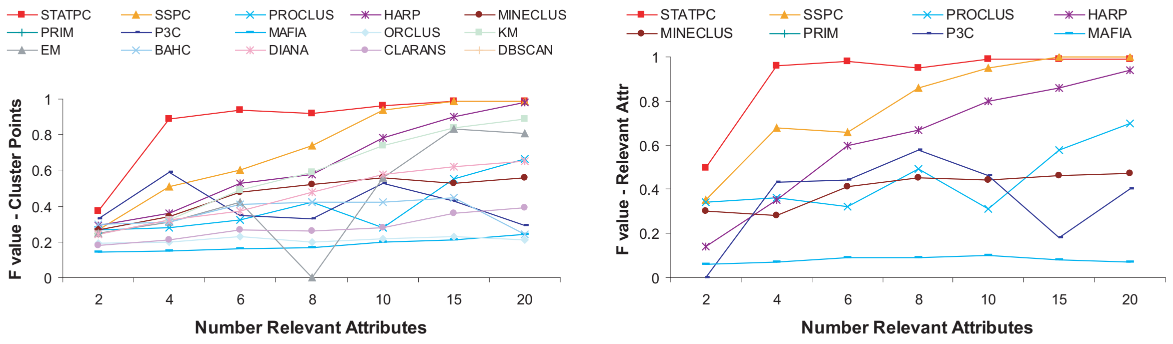


Figure 11: Category Gaussian_Different

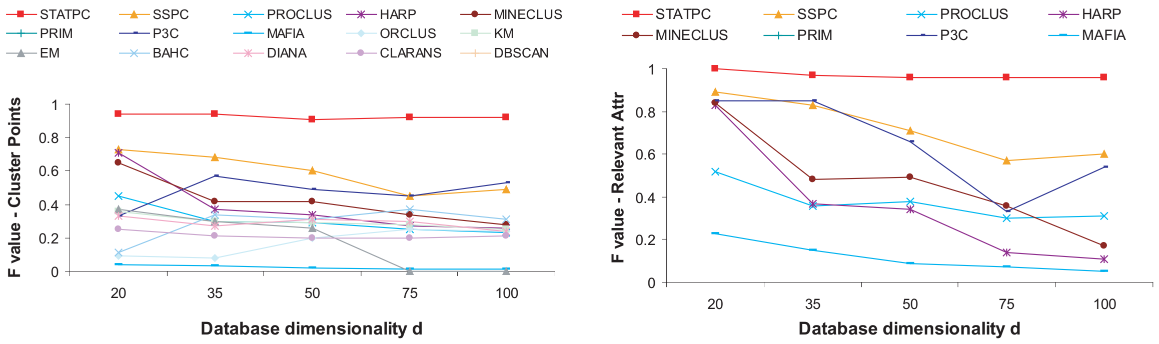


Figure 12: Effect d

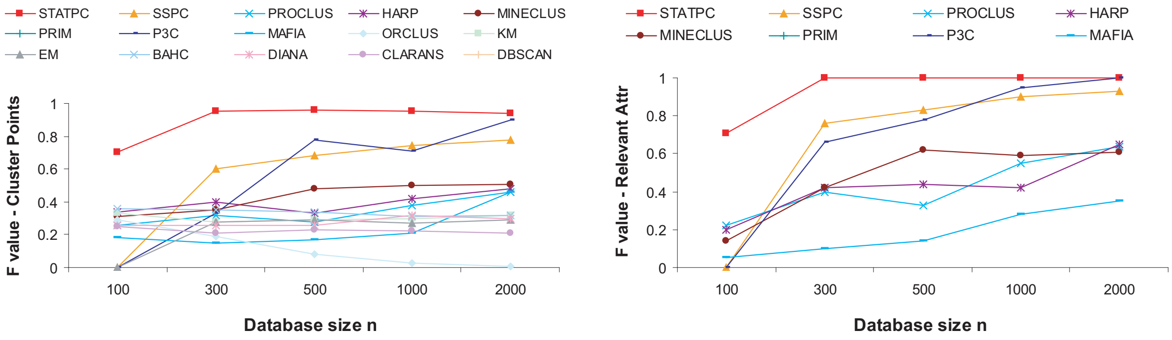


Figure 13: Effect n

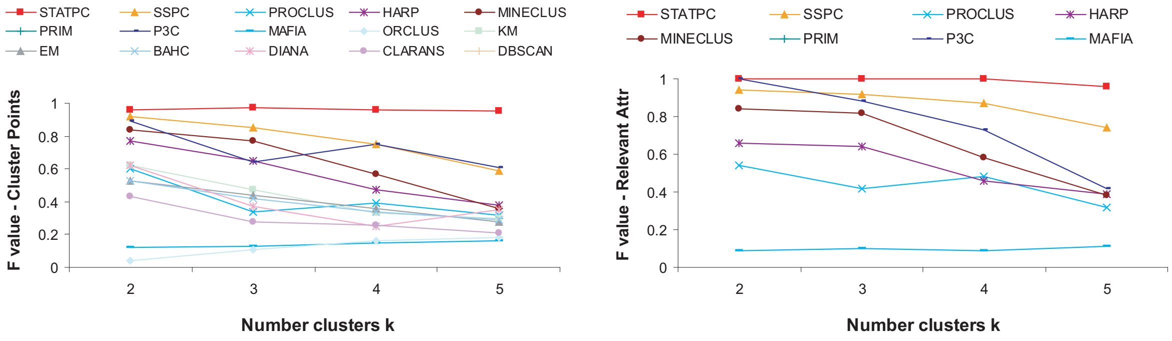


Figure 14: Effect k

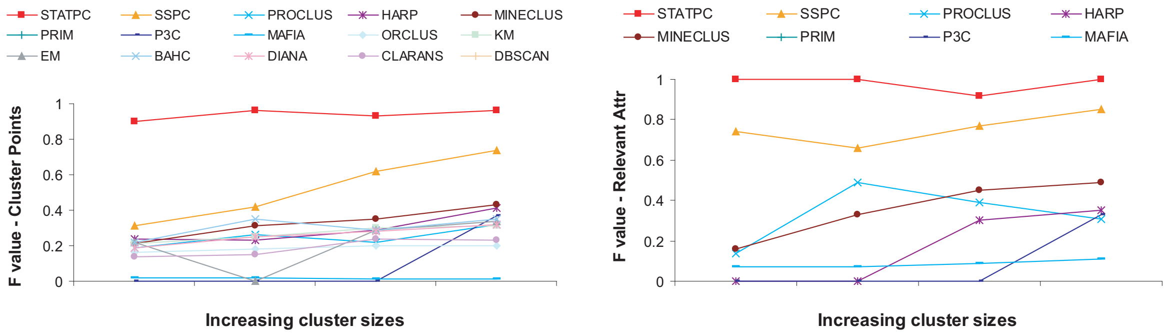


Figure 15: Effect Cluster Sizes

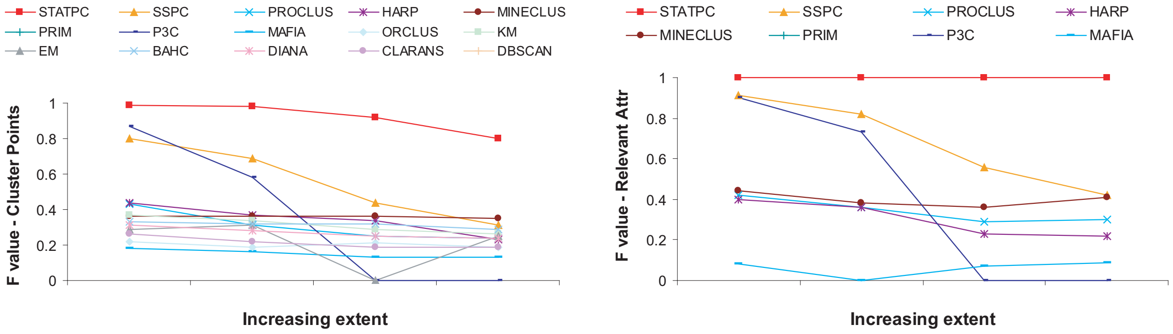


Figure 16: Effect Extent

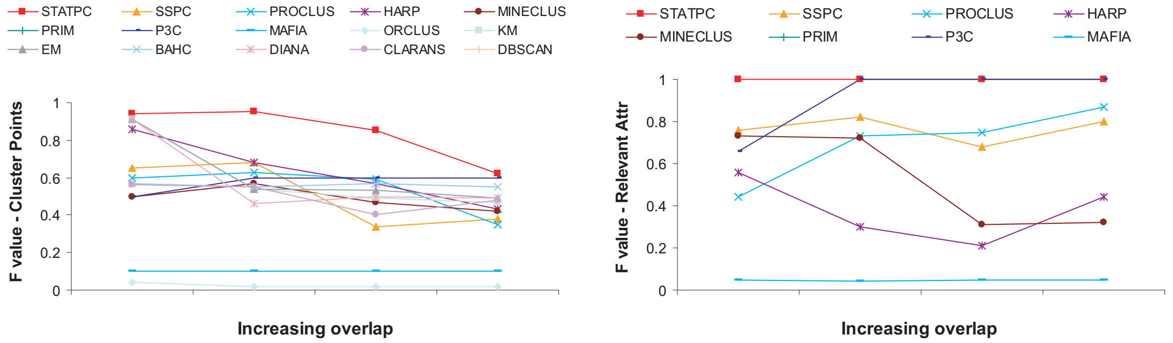


Figure 17: Effect Overlap

tributes added. STATPC consistently finds 7 8-dimensional subspace clusters, 5 6-dimensional subspace clusters, and 2 33-dimensional subspace clusters on Pima Indians Diabetes, Liver Disorders, respectively WPBC data sets and their extensions, outperforming the other algorithms by at least a margin of 20%, 10%, respectively 10%.

Scalability experiments. In all scalability figures, the time is represented on a log10 scale. Figure 19(a) 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. Although STATPC has a larger runtime than previous algorithms, it is still acceptable and, as we believe, worth the trade-off for much better effectiveness in finding subspace clusters. Figure 19(b) 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. The scalability of STATPC is comparable to that of the other algorithms. Figure 19(c) shows that STATPC is unaffected by increasing average cluster dimensionalities on the data sets from category $Uniform\_Equal$.

6. CONCLUSIONS AND FUTURE WORK

In this work, we identified important shortcomings in the existing projected and subspace clustering literature and proposed a novel problem formulation that ensures that found subspace clusters actually stand out in the data in a statistical sense. We also proposed an approximation algorithm STATPC for the problem, which in our experimental evaluation clearly outperforms state-of-the art projected and subspace clustering algorithms.

There are many opportunities for future work such as the study of other approximation algorithms, and the investigation of distribution assumptions other than Uniform.

7. ACKNOWLEDGMENTS

We thank the authors who provided us the code for some projected clustering algorithms. This research was supported by the Alberta Ingenuity Fund, iCORE, and NSERC.

8. REFERENCES

[1] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, I. Müller-Gorman, and A. Zimek. Detection and visualization of subspace clusters hierarchies. In *DASFAA*, 2007.

[2] D. Agarwal, A. McGregor, J. Phillips, S. Venkatasubramanian, and Z. Zhu. Spatial scan statistics: approximations and performance study. In *KDD*, 2006.

[3] C. C. Aggarwal, C. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *SIGMOD*, 1999.

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

[5] R. Agrawal and R. Srikan. Fast algorithms for mining association rules. In *VLDB*, 1994.

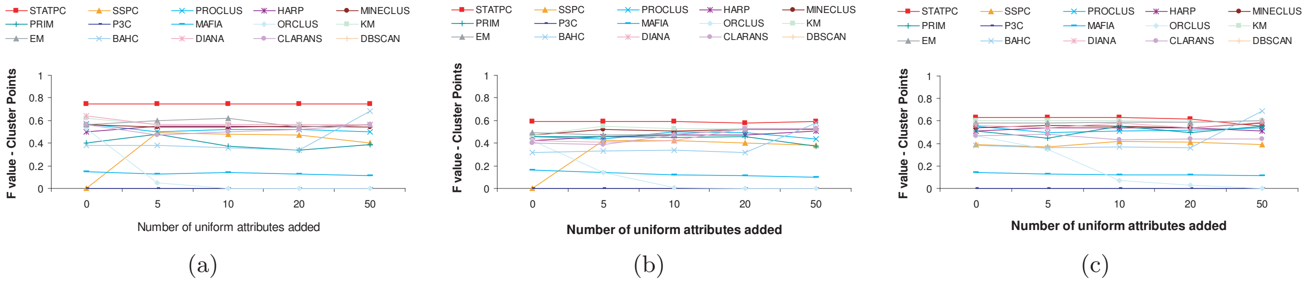


Figure 18: Accuracy on (a) Pima Indians Diabetes (b) Liver Disorders (c) Wisconsin Breast Cancer Prognostic

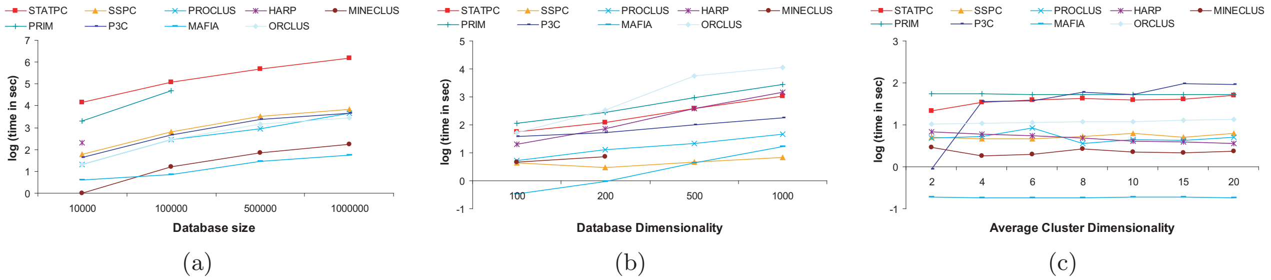


Figure 19: Scalability with increasing (a) db size (b) db dimensionality (c) avg. cluster dimensionality

[6] I. Assent, R. Krieger, E. Müller, and T. Seidl. DUSC: Dimensionality unbiased subspace clustering. In *ICDM*, 2007.

[7] A. Baddeley. Spatial point processes and their applications. *Lecture Notes in Mathematics*, 1892:1–75, 2007.

[8] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *JRSS-B*, 57:289–200, 1995.

[9] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is nearest neighbor meaningful? *LNCS*, 1540:217–235, 1999.

[10] C. Böhm, K. Kailing, H.-P. Kriegel, and P. Kröger. Density connected clustering with local subspace preferences. In *ICDM*, 2004.

[11] C. H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *KDD*, 1999.

[12] W. Feller. *An Introduction to Probability Theory and Its Applications*. John Wiley and Sons, NY, 1950.

[13] J. Friedman and N. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9:123–143, 1999.

[14] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *SIGMOD*, 1984.

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

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

[17] G. Liu, J. Li, K. Sim, and L. Wong. Distance based subspace clustering with flexible dimension partitioning. In *ICDE*, 2007.

[18] G. Moise, J. Sander, and M. Ester. P3C: A robust projected clustering algorithm. In *ICDM*, 2006.

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

[20] K. Ng, A. Fu, and C.-W. Wong. Projective clustering by histograms. *IEEE TKDE*, 17(3):369–383, 2005.

[21] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explorations Newsletter*, 6(1):90–105, 2004.

[22] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. Murali. A Monte Carlo algorithm for fast projective clustering. In *SIGMOD*, 2002.

[23] K. Sequeira and M. Zaki. SCHISM: a new approach for interesting subspace mining. In *ICDM*, 2004.

[24] G. W. Snedecor and W. G. Cochran. *Statistical Methods*. Iowa State University Press, 1989.

[25] K. Yip, D. Cheung, and M. Ng. HARP: a practical projected clustering algorithm. *IEEE TKDE*, 16(11):1387–1397, 2004.

[26] K. Yip, D. Cheung, and M. Ng. On discovery of extremely low-dimensional clusters using semi-supervised projected clustering. In *ICDE*, 2005.

[27] M. L. Yiu and N. Mamoulis. Iterative projected clustering by subspace mining. *IEEE TKDE*, 17(2):176–189, 2005.