

# ISC – Intelligent Subspace Clustering, A Density based Clustering approach for High Dimensional Dataset

Sunita Jahirabdkar, and Parag Kulkarni

**Abstract**—Many real-world data sets consist of a very high dimensional feature space. Most clustering techniques use the distance or similarity between objects as a measure to build clusters. But in high dimensional spaces, distances between points become relatively uniform. In such cases, density based approaches may give better results. Subspace Clustering algorithms automatically identify lower dimensional subspaces of the higher dimensional feature space in which clusters exist. In this paper, we propose a new clustering algorithm, ISC – Intelligent Subspace Clustering, which tries to overcome three major limitations of the existing state-of-art techniques. ISC determines the input parameter such as  $\epsilon$  – distance at various levels of Subspace Clustering which helps in finding meaningful clusters. The uniform parameters approach is not suitable for different kind of databases. ISC implements dynamic and adaptive determination of Meaningful clustering parameters based on hierarchical filtering approach. Third and most important feature of ISC is the ability of incremental learning and dynamic inclusion and exclusions of subspaces which lead to better cluster formation.

**Keywords**—Density based Clustering, High Dimensional Data, Subspace Clustering, Dynamic Parameter Setting.

## I. INTRODUCTION

THE process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. The dissimilarities between objects are accessed based on the attribute values describing the objects. A cluster of data objects can be treated collectively as one group in many applications [4]. As a data mining function, cluster analysis can be used as a stand alone tool to gain insight into the distribution of data, to observe the characteristics of each cluster, and to focus on a particular set of clusters for further analysis.

Typical clustering methods compute similarities between objects based on an entire set of selected attributes. Many of the real world datasets consist of objects modeled by high dimensional data. Each object is described by hundreds of attributes. For instance, In many Computer Vision applications, such as motion segmentation, face clustering with varying illumination, Pattern Classification, Temporal Video Segmentation etc., image data is huge-dimensional.

S. Jahirabdkar is with the Computer Engineering Department of Cummins College of Engineering, Pune University, Pune (India) as Asst. professor (e-mail: sunita.jahirabdkar@cumminscollege.in).

P. Kulkarni is with Capsilon Research Labs, Pune (India) as Chief Scientist and Director – Research. He is Alumnus of IIT and IIM (e-mail: parag.kulkarni@capsilon.com).

Other examples for high-dimensional feature vectors representing complex objects can be found in the area of Molecular Biology [13], CAD databases etc. However, when the number of measured attributes is large, it may be the case that two given groups differ at only a subset of the measured attributes, and so only a subset of the attributes are “relevant” to the clustering. In such cases, traditional clustering methods may fail because the differences between any two groups, averaged over all the attributes, are small [1]. Subspace Clustering algorithms are clustering algorithms that look for and build clusters not necessarily in the whole space, but also in subspaces of the attributes. Formally, a subspace cluster can be defined as a pair (Subspace of the feature space, Subset of data points).

Generally, the subspace clusters may be hierarchically nested, i.e. several subspace clusters of low dimensionality may together form a subspace cluster of higher dimensionality. Detecting such relationships of subspace clusters is obviously a hierarchical problem. Fig. 1 illustrates a simple example of a hierarchy of subspace clusters in a 3-dimensional feature space: the 2-dimensional clusters  $C_1$  and  $C_2$  are embedded within cluster  $C_3$  which is a 3-dimensional cluster.

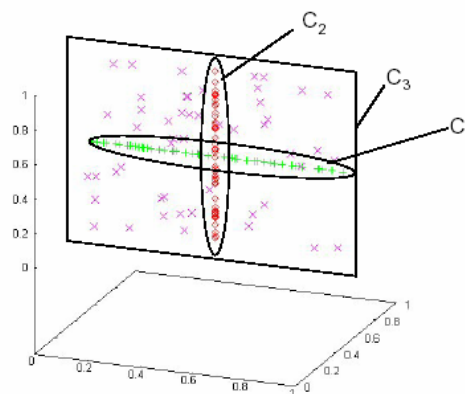


Fig. 1 Hierarchy of Subspace Clusters

The resulting hierarchy is different from the result of a conventional hierarchical clustering algorithm, e.g., a Dendrogram. In a Dendrogram, each object is placed in a singleton cluster at the leaf level, whereas the root node represents the cluster consisting of the entire data set.

This concept of hierarchy will be used at dimension level in ISC i.e. we find low dimensional Subspace Clusters first and

then try to combine these low dimensional Subspace Clusters to form a higher dimensional meaningful subspace cluster. At each dimension level, objects will be assigned to subspace clusters using the density notion of clustering. Thus research area of our paper, we call as, "Density based Hierarchical Subspace Clustering". ISC determines meaningful clustering parameters dynamically and adaptively based on hierarchical filtering approach. Thus the most important feature of ISC is the ability of incremental learning and dynamic inclusion and exclusions of subspaces which lead to better cluster formation.

The remainder of this paper is organized as follows. We start by reviewing the related work in Density based Subspace Clustering approaches for clustering High Dimensional Dataset in section II. In Section III, we detail our new clustering algorithm, ISC – Intelligent Subspace Clustering. Section IV discusses the results along with future research direction and the conclusion is in Section V.

## II. RELATED WORK

Subspace Clustering is a very important technique to seek clusters hidden in various subspaces (Dimensions) in a very high dimensional database. There are very few approaches to Subspace Clustering. These approaches can be classified by the type of results they produce. The first class of algorithms allows overlapping clusters, i.e., one data point or object may belong to different clusters in different projections e.g. CLIQUE [5], ENCLUS [6], MAFIA [7], SUBCLU [8] and FIRES [9]. The second class of subspace clustering algorithms generate non-overlapping clusters and assign each object to a unique cluster or noise e.g. DOC [10] and PreDeCon [11].

The first well-known Subspace Clustering algorithm is CLIQUE, CLUSTERING in QUEST [5]. CLIQUE is a grid-based algorithm, using an apriori-like method which recursively navigates through the set of possible subspaces. A slight modification of CLIQUE is the algorithm ENCLUS, Entropy based CLUSTERING [6]. A more significant modification to CLIQUE is MAFIA, Merging of Adaptive Finite IntervALS [7], which is also a grid-based but uses adaptive, variable sized grids in each dimension. The major disadvantage of all these techniques is caused by the use of grids. Grid-based approaches are based on positioning of grids. Thus clusters are always of fixed size and depend on orientation of grid. Density based Subspace Clustering is one more approach. The first of this kind, DOC proposes a mathematical formulation regarding the density of points in subspaces. But again, the density of subspaces is measured using a hypercube of fixed width  $w$ , so it has the similar problems [10].

Another approach SUBCLU (density connected SUBspace CLUSTERING) is able to effectively detect arbitrarily shaped and positioned clusters in subspaces [8]. Compared to the grid-based approaches SUBCLU achieves a better clustering quality but requires a higher runtime. SURFING is one more effective and efficient algorithm for feature selection in high dimensional data [12]. It finds all subspaces interesting for clustering and sorts them by relevance. But it just gives relevant subspaces for further clustering. The only approach which can find subspace cluster

hierarchies is HiSC [14]. However it uses the global parameters such as Density Threshold ( $\mu$ ) and Epsilon Distance ( $\epsilon$ ) at different levels of dimensionalities while finding subspace clusters. Thus its results are biased with respect to the dimensionality.

To find clusters those are hidden in various subspaces, parameters like  $\epsilon$  – distance (epsilon distance) has to be set depending upon number of dimensions considered for clustering. ISC determines  $\epsilon$  – distance dynamically and adaptively at various dimensionality levels, which helps in finding meaningful clusters. The most important feature of ISC is the ability of incremental learning and dynamic inclusion and exclusion of subspaces which lead to better cluster formation.

## III. INTELLIGENT SUBSPACE CLUSTERING ALGORITHM - ISC

Algorithm ISC is based on the density notion of hierarchical subspace clustering. Thus the aim of ISC is to detect clusters of lower dimensional subspaces contained in clusters of higher dimensional subspaces. Our general idea is to evaluate whether two points are contained in a common subspace cluster, using the density based clustering approach. For example, two points that are in a 1-d subspace cluster may be contained in a 2-d cluster that consists of the two 1-d projections.

Let  $D$  be a data set of  $n$ -normalized feature vectors of dimensionality  $d$ . Let  $A = \{A_1, \dots, A_d\}$  be the set of all attributes  $A_i$  of  $D$ . Any subset  $S \subseteq A$  is called a Subspace. The projection of an object  $p \in D$  into a subspace  $S \subseteq A$  is denoted by  $\pi_S(p)$ .

For any  $\epsilon \in \mathbb{R}^+$  the  $\epsilon$ -neighborhood of an object  $p \in DB$  is denoted by  $N_\epsilon(p)$ . It can be defined as all those objects, the distance between some object  $p$  and other object is less than  $\epsilon$ . The parameter  $\mu$  specifies the density threshold, initially as an input parameter. Based on these two parameters dense regions can be specified with the help of core objects. An object  $p \in DB$  is called core object in  $D$  if its  $\epsilon$ -neighborhood in  $D$  contains at least  $\mu$  objects.

Usually clusters contain several core objects located inside a cluster.

The aim of ISC is to detect clusters of lower dimensional subspaces contained in clusters of higher dimensional subspaces. The general idea is to evaluate whether two points are contained in a common subspace cluster. For example, two points that are in a 1-d subspace cluster may be contained in a 2-d cluster that consists of the two 1-d projections.

For example, in Fig. 2 each of the two lines forms a 1-dimensional subspace cluster. The plane is a 2-dimensional subspace cluster and it includes the two 1-dimensional subspace clusters. In order to detect the lines, a search for 1-dimensional subspace clusters has to be applied, whereas in order to detect the plane, a search for 2-dimensional subspace clusters has to be performed. Moreover, searching subspace clusters of different dimensionality is basically a hierarchical problem, because the information that a point belongs to some  $i$ -dimensional subspace cluster that is itself embedded into an  $j$ -dimensional subspace cluster where  $i < j$ , can only be uncovered by using a hierarchical approach.

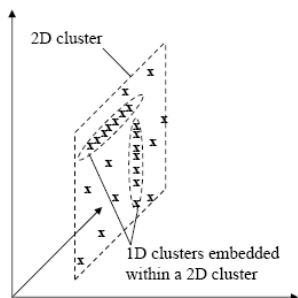


Fig. 2 1-D clusters embedded within 2-D cluster

The algorithm ISC starts at 1-dimension and will iterate till d-dimensions. At each dimension level, it applies a density based clustering. The dimensions are going to form a tree-like structure with single dimensional clusters at leaf nodes and d-dimensional clusters at root node. It uses the concept that several subspace clusters of low dimensionality may together form a larger subspace cluster of higher dimensionality [14]. The parameter such as  $\epsilon$  - Epsilon distance will be decided differentially and adaptively depending upon the subspace dimensionality and the minimum and maximum distances among the data points at that level. If this distance has to be measured between 2 subspace clusters where Subspace Dimensionality  $\neq 1$ , we use single-link linkage method which tries to find the minimum distance between two clusters  $C_i$  and  $C_j$  as the distance between the two closest objects within those clusters [2]. The strategy will be to merge those points into common clusters which will have Subspace Distances smaller than  $\epsilon$  - distance, adaptively calculated at that level. A hierarchy of subspace clusters will be build accordingly.

None of the previously proposed algorithms uses density based approach for subspace clustering to detect such hierarchies of nested Subspace Clusters using differential input parameters.

ISC can be divided into 4 major tasks.

1. Application of algorithm RANK to find dimensions in the descending order of relevance / interestingness.
2. Application of density based clustering on all 1-d data (starting with highest ranked attribute). {Algorithm DBSCAN}
3. Continue combining smaller dimensional clusters to form higher dimensional clusters by selecting next ranked dimensions to add with.
  - Density approach of clustering to detect clusters even at any Subspace Dimensionality  $> 1$
  - The  $\epsilon$  - distance will be the distance between clusters. It will be called as Subspace Distance.
  - Single-Link linkage method to find the distance between two clusters (set of points).
  - Combine those clusters whose Subspace Distance is less than  $\epsilon$  - distance.
4. At each dimension level -
  - Depending upon minimum and maximum distance between Subspace Distances, the parameters such as  $\epsilon$  - distance and threshold ( $\mu$ ) will be changed to accommodate correct

points in the clusters. This leads to adaptive parameter setting.

- Those Subspaces which do not contain any core points are directly removed which makes it more efficient.

#### A. Algorithm RANK

Algorithm RANK measures the “Interestingness” of a dimension with respect to no. of data points taking part in building subspace clusters along that dimension. For each object, we will first compute a Subspace Preference Vector ( $p\_vector$ ). This vector will contain “1” for all those dimensions for which the point can become part of the cluster along that dimension and “0” otherwise. For this we search  $\epsilon$  - neighborhood of the point in that dimension. If it contains no. of objects higher than  $\mu$ , the point can participate in cluster along that dimension. So  $p\_vector$  contains “1” in that dimension position. All these sets are then compared and the attributes with highest number of “1”s are ranked as most interesting. Subsequently in descending order we tag remaining dimensions with corresponding rankings. The pseudo code of algorithm is given in Fig. 3.

```

Algorithm RANK (Database D, real  $\mu$ )
// Finds dimensions in descending order of interestingness
For each  $p \in D$  DO
    Initialize  $p\_vector_{p(i)}(v^1, \dots, v^d)$ ; // preference vector
For dim = 1,d DO
    for each data point
         $N_{\epsilon}^{(dim)}(p) = \{x \mid DIST_{(dim)}(p, x) \leq \epsilon\}$ ; // find no. of objects
         $S = \text{sum}(x)$ ; // in  $\epsilon$ -neighborhood
        if  $s > \mu$  then
             $p\_vector_{p(i)} = p\_vector_{p(i)} \cup 1$ ;
        Else
             $p\_vector_{p(i)} = p\_vector_{p(i)} \cup 0$ ;
        End if
    End For // all dimensions
End For // all data points

For dim = 1,d DO
    For each  $p \in D$  DO
        Sum  $p\_vector_{p(i)}[d]$ ;
        Update vector  $p\_vector$ ; // vector with sorted dims
    End For
End
    
```

Fig. 3 Algorithm to find interestingness of dimensions

Once the vector  $p\_vector$  is generated, we check for any dimension which does not contain any core objects. These dimensions indicate least importance in forming Subspace Clusters along those dimensions. We remove such dimensions from the vector  $p\_vector$  to reduce the computations and thus to improve the efficiency of our algorithm.

#### B. Algorithm ISC

First we apply algorithm RANK which gives a list of dimensions in the descending order of Interestingness.

Then we apply DBSCAN [3], the robust density based clustering algorithm with input parameters  $\mu$  (Density Threshold) to one dimensional dataset. DBSCAN also needs  $\epsilon$  - distance as another parameter which we calculate by using

the dissimilarity matrix formed with that dimension and the minimum and maximum distance found there in. For this we will start with that dimension which is having highest rank given by algorithm RANK. DBSCAN is able to detect arbitrarily shaped clusters by one single pass over the data. DBSCAN checks the  $\epsilon$ -neighborhood of each point  $p$  in the database. If  $N_{\epsilon}(p)$  of an object  $p$  consists of at least  $\mu$  objects, i.e., if  $p$  is a core object, a new cluster  $C$  containing all objects of  $N_{\epsilon}(p)$  is created. Then, the  $\epsilon$  - neighborhood of all points  $q \in C$  which has not yet been processed is checked. If object  $q$  is also a core object, the neighbors of  $q$  which are not already assigned to cluster  $C$  are added to  $C$  and their  $\epsilon$  -neighborhood is checked in the next step. This procedure is repeated until no new point can be added to the current cluster  $C$ . Then the algorithm continues with a point which has not yet been processed, trying to expand a new cluster.

The algorithm ISC (see Fig. 4) will start at 1-dimension and will iterate till  $d$ -dimensions according to the ranks stored in vector  $p\_vector$ . At each dimension level, we first calculate  $\epsilon$  - distance to be used at that dimensionality level. Then DBSCAN will be applied considering these parameters. Again those Subspace Clusters with null core objects will be removed to reduce the computations.

We will need to define Subspace Distance which will be the distance between two subspace clusters to be combined in higher dimensions. For this Single-Link method [2] will be used to find Subspace Distance between two subspace clusters. It is the distance between any two clusters  $C_i$  and  $C_j$  as the minimum distance between two closest objects.

```

Algorithm ISC (Database D, real  $\mu$  )
RANK(D,  $\mu$  );
Remove dimensions with null core objects;
Apply DBSCAN on RANK(1); // on the highest ranked dim
For dim = 2,d DO // in vector p_vector
    Calculate  $\epsilon$  - distance;
    Apply DBSCAN on RANK(1)+dim(i);
    Remove Subspaces with null core objects;
End For;
End;
    
```

Fig. 4 Algorithm ISC

### C. Input Parameters

ISC applies DBSCAN which is a robust density based clustering at each level of dimensionality to find Subspace Clusters. DBSCAN needs two input parameters  $\epsilon$  - distance and  $\mu$  - Density Threshold used to define:

- 1) An  $\epsilon$  - neighborhood of a point  $p$
- 2) A core object (a point with a neighborhood consisting of more than  $\mu$  objects)
- 3) A concept of a point  $q$ , density-reachable from a core object  $p$  (a finite sequence of core objects between  $p$  and  $q$ , such that each next belongs to an  $\epsilon$  - neighborhood of its predecessor)
- 4) A density-connectivity of two points  $p, q$  (they should be density-reachable from a common core object) from different subspace clusters.

But ISC takes only  $\mu$  as input parameter. The parameter  $\epsilon$  specifies the locality of the neighborhood from which the local Subspace Distance of each point in Subspace Clusters is determined. Obviously, this parameter is rather critical because if it is chosen too large, the subspace image may be blurred by noise points, whereas if it is chosen too small, there may not be a clear subspace preference observable, although existing. Further, as the dimensions goes on increasing, the distance between data points already become larger. So we may need to increase it a little bit at higher level to accommodate high level subspace clusters. ISC successfully identifies this parameter at various levels of dimensions when tested with scientific data with nearly 30-40 dimensions. This gives the ability of incremental learning and dynamic inclusion and exclusions of subspaces which lead to better cluster formation.

## IV. EXPERIMENTAL EVALUATION

In this section, we present a broad evaluation of ISC. We implemented ISC as well as the two basic methods DBSCAN and RANK in JAVA. All experiments were run on Microsoft Windows XP platform with a 2.0 GHz CPU and min 2.0 GB RAM. We evaluated ISC using several synthetic datasets. We tried to vary the dimensions of the data sets from 8 to 40, the number of clusters from 2 to 5, the subspace dimensionality of the clusters from 2 to 8. The density of the clusters was chosen randomly. All attribute values were normalized between 0 and 10.

In all experiments, ISC could generate Subspace Clusters hidden in the data.

## V. CONCLUSION

In this paper, we first motivated the need for Hierarchical Subspace Clustering for very high dimensional dataset. Later we proposed a new approach ISC (Intelligent Subspace Clustering) which uses the density based clustering to find Subspace Clusters embedded in higher dimensional clusters. By determining the  $\epsilon$  - distance parameter dynamically and adaptively at each dimension level, it allows for incremental learning by allowing modifying parameters adaptively. This leads to better cluster formation at higher dimensionality. The experimental evaluation proved it to be a successful clustering approach. Thus, it will benefit large application domains such as DNA database and correspondingly in DNA analysis, It will help to identify customer groups, identify frauds or unusual transactions in financial database and lots of other application areas like information integration system, text-mining, CAD database etc. It will be a specialized, very effective Data Mining tool.

## REFERENCES

- [1] Michael Steinbach, Levent Ertöz and Vipin Kumar, "The Challenges of Clustering High Dimensional Data", [online]. Available : [http://www-users.cs.umn.edu/~kumar/papers/high\\_dim\\_clustering\\_19.pdf](http://www-users.cs.umn.edu/~kumar/papers/high_dim_clustering_19.pdf)
- [2] R. Sibson. SLINK, "An optimally efficient algorithm for the single-link cluster method", The Computer Journal, 16(1):30{34,1973.
- [3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with Noise", In Proceedings of the 2nd ACM International Conference on Knowledge Discovery and Data Mining (KDD), Portland, OR, 1996.
- [4] J. Han and M. Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufman, 2001.

- [5] R. Agrawal, J. Gehrke, D. Gunopulos, and Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications", In Proceedings of the SIGMOD Conference, Seattle, WA, 1998.
- [6] C. H. Cheng, A. W.-C. Fu, and Y. Zhang, "Entropy-based subspace clustering for mining numerical data", In Proceedings of the 5th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), San Diego, CA, pages 84-93, 1999.
- [7] S. Goil, H. Nagesh, and A. Choudhary, "MAFIA: Efficient and scalable subspace clustering for very large data sets", Technical Report CPDC-TR-9906-010, Northwestern University, 1999.
- [8] K. Kailing, H.P. Kriegel, and P. Kroger, "Density-connected subspace clustering for high-dimensional data", In Proceedings of the 4th SIAM International Conference on Data Mining (SDM), Orlando, FL, 2004.
- [9] H.P. Kriegel, P. Kroger, M. Renz, and S. Wurst, "A generic framework for efficient subspace clustering of high-dimensional data. In Proceedings of the 5th International Conference on Data Mining (ICDM), Houston, TX, 2005.
- [10] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali, "A Monte Carlo algorithm for fast projective clustering. In Proceedings of the SIGMOD Conference, Madison, WI, 2002.
- [11] C. Bohm, K. Kailing, H.P. Kriegel, and P. Kroger, "Density connected clustering with local subspace preferences", In Proceedings of the 4th International Conference on Data Mining (ICDM), Brighton, U.K., 2004.
- [12] C. Baumgartner, Plant C, Railing K, Kriegel H. -P, Kroger P, "Subspace Selection for Clustering High-Dimensional Data", In proceedings of 4th IEEE Int. Conference on Data Mining (ICDM 04), PP 11-18, Brighton, UK, 2004.
- [13] Daxin Jiang, Chun Tang , Aidong Zhang: "Cluster Analysis for Gene Expression Data: A Survey", IEEE Transactions on Knowledge and Data Engineering, Issue Date : November 2004, pp. 1370-1386.
- [14] Elke Achtert, Christian Bohm, Hans-Peter Kriegel, Peer Kroger, Ina Muller-Gorman, Arthur Zimek, "Finding Hierarchies of Subspace Clusters", In Proceedings of 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), Berlin, Germany, 2006.