Subspace Clustering—A Survey

Bhagyashri A. Kelkar and Sunil F. Rodd

Abstract High-dimensional data clustering is gaining attention in recent years due to its widespread applications in many domains like social networking, biology, etc. As a result of the advances in the data gathering and data storage technologies, many a times a single data object is often represented by many attributes. Although more data may provide new insights, it may also hinder the knowledge discovery process by cluttering the interesting relations with redundant information. The traditional definition of similarity becomes meaningless in high-dimensional data. Hence, clustering methods based on similarity between objects fail to cope with increased dimensionality of data. A dataset with large dimensionality can be better described in its subspaces than as a whole. Subspace clustering algorithms identify clusters existing in multiple, overlapping subspaces. Subspace clustering methods are further classified as top-down and bottom-up algorithms depending on strategy applied to identify subspaces. Initial clustering in case of top-down algorithms is based on full set of dimensions and it then iterates to identify subset of dimensions which can better represent the subspaces by removing irrelevant dimensions. Bottom-up algorithms start with low dimensional space and merge dense regions by using Apriori-based hierarchical clustering methods. It has been observed that, the performance and quality of results of a subspace clustering algorithm is highly dependent on the parameter values input to the algorithm. This paper gives an overview of work done in the field of subspace clustering.

Keywords Clustering · Subspace clustering · High-dimensional data

---

B. A. Kelkar (✉)
KLS Gogte Institute of Technology, Belgaum, Karnataka, India
e-mail: kelkar.ba@sginstitute.in

B. A. Kelkar
Sanjay Ghodawat University, Atigre, Kolhapur, Maharashtra, India

S. F. Rodd
Department of Computer Science and Engineering, Gogte Institute of Technology, Belagavi, Karnataka, India
e-mail: sfroddgit@git.edu

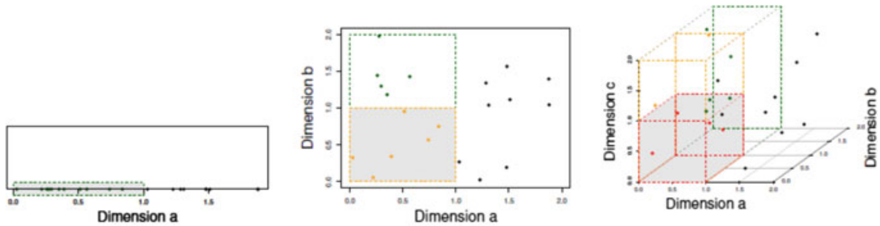
© Springer Nature Singapore Pte Ltd. 2019
V. E. Balas et al. (eds.), *Data Management, Analytics and Innovation*,
Advances in Intelligent Systems and Computing 808,
https://doi.org/10.1007/978-981-13-1402-5_16

1 Introduction

Clustering is an essential data mining task for summarization, learning, and segmentation of data. It has been applied for target marketing, machine learning, pattern recognition, and statistics. Clustering is an exploratory data analysis task and aims to discover groups of similar objects called as clusters from input data set. The objects belonging to the same cluster must be highly similar whereas objects from different clusters must be highly dissimilar. Desired properties of the clustering algorithm are completeness, stability, homogeneous and significant results and efficiency.

1.1 The Curse of Dimensionality

Data analytics and machine learning is an evolving area. The grand challenge in this research lies in dealing with ever-increasing amounts of high-dimensional data gathered from multiple sources and different modalities. Bellman [1] refers to the combinatorial explosion that is observed in a data mining task implied due to processing of large number of dimensions as curse of dimensionality. This is due to the fact that, high dimensionality increases the computational complexity and memory requirements. It can adversely degrade underlying algorithm's performance. Clustering is usually done based on distance notations like the Euclidean distance and due to increased dimensionality distance between data points become meaningless. Additional dimensions spread the data points further apart as shown in Fig. 1a. With one dimension, half of the points were in a unit bin. If second dimension is added, data gets stretched as shown in Fig. 1b and the points get spread out further, pulling them apart, resulting in only about a one fourth of the points into a unit bin. Further addition of a third dimension again spreads the data and a unit bin holds only a few points as shown in Fig. 1c. When the dimensionality of the data becomes too large, the points then are all almost equidistant [2, 3] and distance between the points tend to zero as shown in Fig. 2. Hence large amount of



(a) 11 Objects in One Unit Bin (b) 6 Objects in One Unit Bin (c) 4 Objects in One Unit Bin

Fig. 1 The curse of dimensionality [1]—data becomes extremely sparse with increasing dimensions

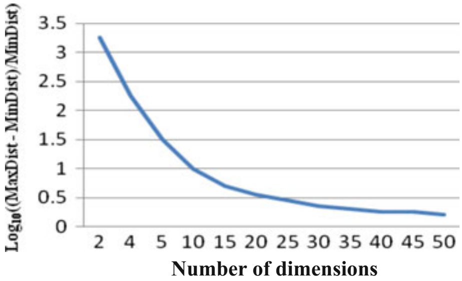


Fig. 2 Distance between data points is no longer meaningful with increased dimensions

data objects are required to satisfy a given density threshold. This fact badly affects performance of clustering algorithms as cluster membership is mainly determined based on distance between and density of data points.

The curse of dimensionality has many aspects. First, in a dataset, all attributes may not contribute to define a certain cluster. Rather the clusters may be present in subspaces. Second, a different subset of attributes may be involved in defining different subspace clusters. Hence a global feature selection procedure may not be applicable to identify attributes contributing to subspace clusters. Third, two subspace clusters might be overlapping, i.e., data point belonging to one subspace cluster $C1$ can be member of another subspace cluster $C2$. Hence subspace clustering requires appropriate feature selection methods which are different from the methods for traditional clustering based on density or partitioning of data.

In high-dimensional data, not all of the attributes are important for good clusters. Some of the attributes may be simply “noise”. The problem is further worsened by the fact that, objects may be related in different subsets of dimensions in different ways and also due to fact that, some of the attributes might also be correlated. Keeping these facts in view, approaches like feature transformation and feature selection have been suggested. Feature transformation methods uncover latent structure in datasets to create combinations of the original attributes and summarize given dataset in fewer dimensions. When the number of irrelevant attributes is large, these methods are rendered irrelevant as they preserve the distance between the objects. As the new features are combination of original features, it is difficult to interpret them. Feature selection is one of the dimensionality reduction techniques and is often applied as a preprocessing step to remove noisy features. It identifies most relevant attributes for the data mining task at hand. This is achieved by evaluating various feature subsets using some criterion. These methods are further classified as: (i) global versus local where global methods find features from complete dataset whereas local methods find features relevant for each individual cluster. (ii) wrapper (with feedback) versus filter (blind-without feedback) where the filter approach selects features based on criteria such as pair wise constraints, mutual information, Laplacian score, chi-square test, etc. then evaluate the attributes, rank them before applying selection criteria. Wrapper methods formulate the problem as a search problem. Different combinations of the features are prepared.

These combinations are evaluated and a comparison with other combinations is done. Combinations of features are scored based on model accuracy using a predictive model. Embedded methods like regularization methods for feature selection are based on learning which features best contribute to the accuracy of the model. The learning is done while the model is being created. However the feature selection methods have a critical limitation that, they cannot uncover relations between objects in multiple, overlapping sub-dimensional spaces.

2 Subspace Clustering

In subspace clustering, clusters are identified in subset of attributes. Subspace clustering algorithms can be considered as an extension to feature selection methods which identify most relevant attributes by evaluating various feature subsets using some criterion. The clustering process first identifies the projections in which clusters may reside and then applies a clustering algorithm in identified subspace. A search method is required to identify subsets of attributes and then they are evaluated based on certain criteria. In subspace clustering object similarity is measured based on the selected attribute subset. For given a database DB with a set Dim of dimensions, clustering result can be denoted as a set $C = \{(C_1, A_1), \ldots, (C_k, A_k)\}$ where $C_i \subseteq$ DB and $A_i \subseteq$ Dim. Figure 3 illustrates an example of subspace clustering.

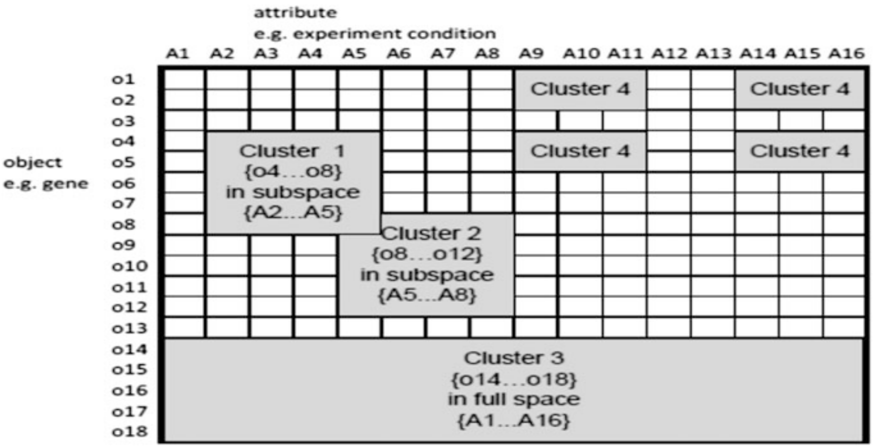


Fig. 3 Example for subspace clustering

3 Classification of Basic Subspace Clustering Approaches

The desirable property of any subspace clustering algorithm is that, it should identify all possible sets of subspace clusters. Also it must be ensured that, the outcome of clustering process must produce the same set of clusters during every run. Efficiency is another aspect of subspace clustering algorithms. The algorithms can be made to handle large data by applying proper heuristics to prune non-significant results. The results of subspace clustering should be easily interpretable. There are three major variants of subspace clustering, viz. the grid-based, window-based, and density-based. The homogeneity of attributes can be identified based on similarity between objects, density of objects, etc., depending on the criteria applied by clustering method.

3.1 Grid-Based Subspace Clustering

In this approach, data space is divided into axis-parallel cells [4]. Then the cells containing objects above a predefined threshold value given as a parameter are merged to form subspace clusters. Number of intervals is another input parameter which defines range of values in each grid. Apriori property is used to prune non-promising cells and to improve efficiency. If a unit is found to be dense in $k - 1$ dimension, then it is considered for finding dense unit in k dimensions. If grid boundaries are strictly followed to separate objects, accuracy of clustering result is hampered as it may miss neighboring objects which get separated by string grid boundary. Clustering quality is highly dependent on input parameters.

3.2 Window-Based Subspace Clustering

Window-based subspace clustering [5] overcomes drawbacks of cell-based subspace clustering that it may omit significant results. Here a window slides across attribute values and obtains overlapping intervals to be used to form subspace clusters. The size of the sliding window is one of the parameters. These algorithms generate axis-parallel subspace clusters.

3.3 Density-Based Subspace Clustering

A density-based subspace clustering approach—SUBCLU is proposed by Kailing et al. (2004). It drops use of grids to overcome drawbacks of grid based subspace clustering algorithms. A cluster is defined as a collection of objects forming a chain which fall within a given distance and exceed predefined threshold of object count. Then adjacent dense regions are merged to form bigger clusters. As no grids are used, these algorithms can find arbitrarily shaped subspace clusters. Clusters are built by joining together the objects from adjacent dense regions. These approaches are prone to values of distance parameters. The effect curse of dimensionality is overcome in density-based algorithms by utilizing a density measure which is adaptive to subspace size.

3.4 Other Prominent Approaches

Overlapping cluster algorithms like CLIQUE [4], ENCLUS (ENtropy based subspace CLUStering) [6], MAFIA [7], SUBCLU [8], FIRES [9] try to enumerate all possible subspace clusters. When a data object belongs to many subspace clusters, the clustering is called overlapping. When each data object is member of a unique cluster or marked as outlier, the clustering is non-overlapping. Some of the non-overlapping approaches are PROCLUS [10], DOC [11], PreDeCon [12], etc. Lance et al. [13] classify subspace clustering as top-down- and bottom-up algorithms based on the strategy used to identify cluster subspaces. In bottom-up approach cluster discovery starts from individual attributes and then the subspaces grow to higher dimensional space. For pruning the search space APRIORI property of density is used. Candidate subspaces for the next higher level of dimension sets is formed only from the lower level dense regions. CLIQUE, OPTIGRID [14], DENCOS [5], MAFIA, SUBCLU, FIRES are some of the bottom-up approaches.

In top-down subspace clustering approach, all dimensions are initially part of a cluster and are assumed to equally contribute to clustering. In the subsequent iterations, importance of each dimension is recalculated and clusters are regenerated. This requires multiple iterations over full set of dimensions. The performance can be improved by making use of sampling technique. Due to top-down partitioning of the data, each data object can be member of a unique cluster. Some of the algorithms additionally identify outliers as a separate group. For meaningful results, parameter tuning is necessary. Further classification of this approach is per cluster weighting methods and per instance weighting methods. Few of the Top-down Algorithms are FIND-IT, ORCLUS [15], PROCLUS [10], COSA [16], δ-CLUSTERS [17]. Figure 4 presents a hierarchy of these two prominent classes of subspace clustering algorithms. Clustering oriented subspace clustering relies on predefined parameters such as the expected number of clusters, average dimensionality of clusters, etc. These algorithms try to optimize the solution and hence each data point is assigned to a cluster which results in assigning noise objects to some clusters.

In a dataset, when an object belongs to a cluster, the variance of the occurring values is less compared range of all other attributes. This geometrical intuition lead to identification of a cluster which contains data points which are densely clustered along relevant attributes. The resulting cluster is an axis-parallel subspace cluster.

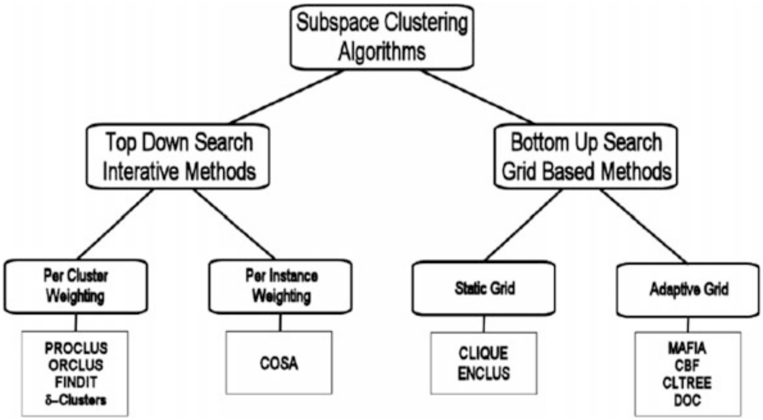


Fig. 4 Hierarchy of subspace clustering algorithms based on search strategy

Basic subspace clustering algorithms like CLIQUE, projected clustering algorithms fall under this category. The algorithms which result in arbitrary oriented subspaces, e.g., ORCLUS use the knowledge that, members of a subspace cluster are always close to the plane in which the subspace resides and use this information for cluster interpretation. Hard subspace clustering algorithms assume that all features have equal importance in forming a subspace whereas soft subspace clustering algorithms proceed by assigning a weight to each dimension based on its contribution to clustering.

4 Enhancements to Traditional Subspace Clustering

Although basic subspace clustering approaches look efficient in solving the clustering problem, they have certain major drawbacks. These algorithms can tackle quantitative 2-D data in format of object *X* attributes, but they are not customized to handle the data in 3-D format, i.e., having dimensions—objects, attributes and time stamp. Similarly, most of them cannot handle complex data such as categorical or streaming data. When the data is in 3-D format, it is rare that clusters can be found in every timestamp of the dataset when the data contains large number of timestamps and there is a need to develop efficient algorithms for mining 3D data. Distance measures applicable to numeric data cannot be applied directly to categorical data as they do not have natural order. Hence devising subspace clustering algorithms for categorical data is another challenge. Many real world data sets contain missing or erroneous values. Therefore, any subspace clustering algorithm working on real-world dataset must handle these datasets properly without affecting accuracy of the results.

Mostly all of the available subspace clustering algorithms work based on parameters values given by user at run time. It has been observed that, clustering output is very much sensitive to the input parameters and outcome varies drastically with minor changes in parameter values. Intuitively setting right values of parameters that will result in good clustering is very much difficult. Hence there is a need to overcome this parameter-sensitivity of subspace clustering algorithms. Domain knowledge or knowing data distribution can help for setting parameters. Sometimes, semi-supervised subspace clustering algorithms can be used to guide parameter setting process. When subspace clusters are overlapping, i.e., when an object may belong to multiple subspace clusters, it may result into explosion of clusters, i.e., too many subspace clusters may be enumerated. This is an undesirable solution as it may lead to too many interpretations of the same data. Hence it is desirable that, only significant subspace clusters which represent true and meaningful information of out the data should be enumerated. This can be achieved in two ways. First as a preprocessing step all significant subspaces can be mined, and then subspace clusters can be identified from these subspaces as in filter approach of feature selection [18]. In the second approach, what is significant in terms of subspace clusters is first defined and then the clustering algorithm mines these clusters directly.

5 Evaluation of Subspace Clustering

Evaluation of clustering output is a complex work. A clustering algorithm is evaluated in terms of execution time and quality of clustering results. Quality of clustering is defined in terms of compactness of a cluster and separation between different clusters and the same is true for subspace clusters. The motivation behind any clustering is to disclose the hidden information in the data as accurately as possible. Hence it is desirable to detect a minimum number of meaningful subspace clusters. There are various clustering quality indexes proposed in literature [19]. However there is lack of standardized guidelines for evaluation of clustering outcome. For a novice researcher, it is a dilemma which clustering quality index is to be used use for a particular dataset. Silhouette index, Simplified Silhouette index, Dunn index, Davies–Bouldin index, Isolation index, PBM index, Point-biserial index, RS index, Rand index are some of the indexes which can be used for the evaluation. In [2], the authors have analyzed some of the standard clustering quality measures and it reveals that, with increasing dimensionality different clustering quality indexes are affected in different ways and conclude that selecting a clustering quality index for high-dimensional data is nontrivial.

Liu et al. (2010) identify major criteria for evaluation of the clustering algorithms based on quality of the results produced namely, the results should be non-monotonous, the algorithm should be robust to noise, it should properly handle varying cluster density and skewed distributions of the data. Compactness of clusters and separation between the clusters are termed as internal clustering quality indexes. How well the data is partitioned is measured by external quality indexes. Müller et al. [20] present a common framework for evaluating major subspace clustering paradigms. Entropy, F1-measure and accuracy are some of the object based measures which mainly relate to (i) purity of clusters identified, (ii) an algorithm's power to discover hidden clusters and (iii) correctness of the algorithm in assigning objects to a cluster respectively. Relative non intersecting area (RNIA) is an object and subspace based measure to find the extent to which found sub-objects cover true sub-objects. Drawback of RNIA measure is that it cannot find if a true cluster is correctly covered by several found clusters or exactly one found cluster covers the true cluster. On the contrary, the clustering error (CE) is advancement over RNIA measure that maps each found cluster to at most one ground truth cluster and also each ground truth cluster to at most one found cluster. Intersection of sub-objects is determined for each such mapping of two clusters. After summing up the individual values give value I' which when substituted in place of I in the RNIA formula will give the CE-value. Thus CE-value penalizes the clustering results producing many smaller clusters. WEKA [21] is an open source framework containing various well known algorithms in clustering, classification, feature selection and association rule mining. It provides facility for visualization of the results. An open source framework OpenSubspace [22] can be used for evaluation of projected and subspace clustering algorithms in WEKA.

5.1 Results Obtained from Earlier Work

Müller et al. have systematically evaluated major paradigms of subspace clustering using OpenSubspace. The study highlights that, SUBCLU and CLIQUE have comparable F1 and Accuracy, but have to pay penalty in terms of RNIA and CE as they try to detect many clusters even more than the count of objects in the dataset as it tries to cover all of the data including noise. This also results into increased runtimes. SUBCLU does not even finish for the biggest real world data set, pendigits. The recent cell-based paradigms show best results with low runtimes. The distance-based approaches also face the problems of high runtimes. Clustering oriented approaches have easy parameterization as these settings decide on clustering output and they show reasonable runtimes. Cell-based approaches like CLIQUE and SUBCLU produce many more clusters in an attempt to achieve good results whereas clustering oriented approaches tend to produce comparatively few clusters.

Generally, high-quality results are paid with high runtime. But even in some algorithms meaningful results are not obtained within tolerable timeframe due to high runtimes even up to several days (for dimensionality >25). Hence practical application of such an algorithm with such high runtimes on high dimensionalities is infeasible. Hence a subspace clustering algorithm must have to find the trade-off between output quality and runtime. Also it is observed that cluster detection time increases with the number of objects. Several heuristics must be applied for having an efficient computation with acceptable accurate results. For neighborhood density computation, the density-based approaches have expensive database scans and hence they do not scale as dimensionality increases. DOC and MINCLUS are found to be good in handling noisy data. There are certain open issues in subspace clustering. Tuning parameter setting is a nontrivial task and usually guesswork is involved. Hence there is a need to have parameter-insensitive algorithms for subspace clustering. For time series data, there is need to identify proper search space pruning strategy. Appropriate post-processing methods for limiting output clusters, organizing the output clusters and formulating models to represent the output are necessary to uncover the information extracted from subspace clusters to useful knowledge.

6 Conclusion

High-dimensional data clustering is a challenging task which first requires formulating how a cluster needs to be represented. Many a times even though a dataset has lots of dimensions, only few of them are of importance for extracting knowledge and rest are noise. Subspace clustering algorithms solve this problem by finding clusters on subsets of attributes and objects. This has the advantage that, those patterns which may be missed by full dimensional clustering are also uncovered. A subspace clustering algorithm must ensure that the subspace projections must be dissimilar and at the same time must not be redundant. Performance of a subspace clustering algorithm is highly dependent on tuning parameters. It has been observed that, when dimensionality of the data increases, accuracy subspace clustering decreases with tremendous increase in runtime. Proper validation techniques must be applied to avoid spurious clusters. The quality evaluation of results obtained from subspace clustering algorithms is challenging as different subspace clustering approaches lead to different cluster characteristics and topologies. Fair and comparable evaluation based on objective evaluation measure of detected subspace clusters is of major importance. In synthetic datasets the best clustering is already known. But such a data might miss variations present in real-world data. Review of recent approaches for subspace clustering highlight that, cell-based approaches outperform in terms of efficiency and quality for low to medium dimensionality. It is also shown that instead of enumerating all subspace clusters which may contain many redundant clusters, outputting a few relevant clusters achieves best results. Further research direction in this field can be reducing database scans, automatic detection of clustering parameters based on data distribution, improving execution time and enhancements in existing algorithms to handle complex data.

References

1. Bellman, R. (1961). *Adaptive control processes*. Princeton: Princeton University Press.
2. Parsons, L., Haque, E., & Liu, H. (2004). Subspace clustering for high dimensional data: A review. *ACM SIGKDD Explorations, 6*(1), 90–105.
3. Francois, D., Wertz, V., & Verleysen, M. (2007). The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, *19*(7), 873–886.
4. Agrawal, R., Gehrke, J., & Gunopulos, D. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 94–105).
5. Liu, G., Sim, K., Li, J., & Wong, L. (2009). Efficient mining of distance-based subspace clusters. *Statistical Analysis and Data Mining, 2*(5–6), 427–444.
6. Cheng, C.-H., Fu, A. W., & Zhang, Y. (1999). Entropy-based subspace clustering for mining numerical data. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 84–93).
7. Goil, S., Nagesh, H., & Choudhary, A. (1999). Mafia: Efficient and scalable subspace clustering for very large data sets. Technical Report CPDC-TR-9906-010, Northwestern University.
8. Kröger, P., Kriegel, H.-P., & Kailing, K. (2004). Density-connected subspace clustering for high-dimensional data. In *Proceedings of SIAM International Conference on Data Mining* (pp. 246–257).
9. Kriegel, H.-P. H., Kroger, P., Renz, M., & Wurst, S. (2005). A generic framework for efficient subspace clustering of high-dimensional data. In *IEEE International Conference on Data Mining* (pp. 250–257), Washington, DC, USA.
10. Aggarwal, C. C., Procopiuc, C. M., Wolf, J. L., et al. (1999). Fast algorithms for projected clustering. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)* (pp. 61–72), Philadelphia, PA.
11. Procopiuc, C. M., Jones, M., Agarwal, P. K., & Murali, T. M. (2002). *A Monte Carlo algorithm for fast projective clustering in SIGMOD* (pp. 418–427). USA.
12. Bohm, C., Railing, K., Kriegel, H.-P., & Kroger, P. (2004). Density connected clustering with local subspace preferences. In *Fourth IEEE International Conference on Data Mining, ICDM* (pp. 27–34).
13. Lance, P., Haque, E., & Liu, H. (2004). Subspace clustering for high dimensional data: A review. *ACM SIGKDD Explorations Newsletter, 6*(1), 90–105.
14. Hinneburg, A., & Keim, D. A. (1999). Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *VLDB* (pp. 506–517).
15. Aggarwal, C. C., & Yu, P. S. (2000). Finding generalized projected clusters in high dimensional spaces. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 70–81).
16. Friedman, J. H., & Meulman, J. J. (2004). Clustering objects on subsets of attributes. *Journal of the Royal Statistical Society: Series B* (*Statistical Methodology*) (pp. 815–849).
17. Yang, J., Wang, W., Wang, H., & Yu, P. (2002). δ-Clusters: Capturing subspace correlation in a large data set. In *Proceedings of the 18th International Conference on Data Engineering* (pp. 517–528).
18. Dash, M., Choi, K., Scheuermann, P., & Liu, H. (2002). Feature selection for clustering – a filter solution. In *Proceedings of the IEEE International Conference on Data Mining* (ICDM02) (pp. 115–124).
19. Patrikainen, A., & Meila, M. (2006). Comparing subspace clusterings. *TKDE, 18*(7), 902–916.
20. Müller, E., Günnemann, S., Assent, I., & Seidl, T. (2009). Evaluating clustering in subspace projections of high dimensional data. *PVLDB, 2*(1), 1270–1281.
21. Weka 3: Data Mining Software in Java. (2014). Available: http://www.cs.waikato.ac.nz/ml/weka/.

22. OpenSubspace:Weka Subspace-Clustering Integration. (2014). Available: http://dme.rwth-aachen.de/OpenSubspace/.
23. Jaya Lakshmi, B., Shashi, M., & Madhuri, K. B. (2017). A rough set based subspace clustering technique for high dimensional data. *Journal of King Saud University-Computer and Information Sciences*.
24. Jaya Lakshmi, B., Madhuri, K. B., & Shashi, M. (2017). An efficient algorithm for density based subspace clustering with dynamic parameter setting. *International Journal of Information Technology and Computer Science, 9*(6), 27–33.
25. Tomašev, N., & Radovanović, M. (2016). Clustering evaluation in high-dimensional data. In *Unsupervised Learning Algorithms* (pp. 71–107). Berlin: Springer.
26. Zhu, B., Ordozgoiti, B., & Mozo, A. (2016). PSCEG: An unbiased parallel subspace clustering algorithm using exact grids. In *24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning ESSAN16* (pp. 27–29), Bruges (Belgium).
27. Peignier, S., Rigotti, C., & Beslon, G. (2015). Subspace clustering using evolvable genome structure. In *Proceedings of the ACM Genetic and Evolutionary Computation Conference (GECCO 2015)* (pp. 1–8).
28. Kaur, A., & Datta, A. (2015). A novel algorithm for fast and scalable subspace clustering of high-dimensional data. *Journal of Big Data, 2*(1), 1–24.
29. Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science, 2,* 165–193.
30. Sim, K., Gopalkrishnan, V., Zimek, A., & Cong, G. (2013). A survey on enhanced subspace clustering. *Data Mining and Knowledge Discovery, 26*(2), 332–397.
31. Liu, H. W., Sun, J., Liu, L., & Zhang, H. J. (2009). Feature selection with dynamic mutual information. *Pattern Recognition, 42*(7), 1330–1339.
32. Kriegel, H. P., Kröger, P., Zimek, A., & Oger, P. K. R. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery Data, 3*(1), 1–58.