An Algorithm for the Removal of Redundant Dimensions to Find Clusters in N-Dimensional Data using Subspace Clustering

[1]Dr. Muhammad Shahbaz, [1]Dr Syed Muhammad Ahsen, [2]Ishtiaq Hussain, [1]Muhammad Shaheen, [3] Syed Athar Masood

[1]Department of Computer Science & Engineering University of Engineering & Technology, Lahore, Pakistan
[2]National Centre for Physics, Quaid e Azam University, Islamabad, Pakistan
[3]Department of Engineering Management, NUST College of E&ME, Rawalpindi Pakistan
[1] m.shahbaz@uet.edu.pk, ahsancs@hotmail.com, shaheen@uet.edu.pk, [3]atharmasood2000@hotmail.com

Abstract: The data mining has emerged as a powerful tool to extract knowledge from huge databases. Researchers have introduced several machine learning algorithms to explore the databases to discover information, hidden patterns, and rules from the data which were not known at the data recording time. Due to the remarkable developments in the storage capacities, processing and powerful algorithmic tools, practitioners are developing new and improved algorithms and techniques in several areas of data mining to discover the rules and relationship among the attributes in simple and complex higher dimensional databases. Furthermore data mining has its implementation in large variety of areas ranging from banking to marketing, engineering to bioinformatics and from investment to risk analysis and fraud detection. Practitioners are analyzing and implementing the techniques of artificial neural networks for classification and regression problems because of accuracy, efficiency. The aim of his short research project is to develop a way of identifying the clusters in high dimensional data as well as redundant dimensions which can create a noise in identifying the clusters in high dimensional data. Techniques used in this project utilizes the strength of the projections of the data points along the dimensions to identify the intensity of projection along each dimension in order to find cluster and redundant dimension in high dimensional data. [Dr. Muhammad Shahbaz, Dr Syed Ahsan, Ishtiaq Hussain, Muhammad Shaheen, Syed Athar Masood. An Algorithm for the Removal of Redundant Dimensions to Find Clusters in N-Dimensional Data Using Subspace Clustering. Journal of American Science 2011;7(6):956-964]. (ISSN: 1545-1003). http://www.americanscience.org.

Keywords: Classification, Regression, Clusters, Data mining, Algorithm

Introduction

In numerous scientific settings, engineering processes, and business applications ranging from experimental sensor data and process control data to telecommunication traffic observation and financial transaction monitoring, huge amounts of high-dimensional measurement data are produced and stored. Whereas sensor equipments as well as big storage devices are getting cheaper day by day, data analysis tools and techniques wrap behind. Clustering methods are common solutions to unsupervised learning problems where neither any expert knowledge nor some helpful annotation for the data is available. In general, clustering groups the data objects in a way that similar objects get together in clusters whereas objects from different clusters are of high dissimilarity. However it is observed that clustering disclose almost no structure even it is known there must be groups of similar objects. In many cases, the reason is that the cluster structure is stimulated by some subsets of the space's dimensions only, and the many additional dimensions contribute nothing other than making noise in the data that hinder the discovery of the clusters within that data. As a solution to this problem, clustering algorithms are applied to the relevant subspaces only. Immediately, the new question is how to determine the relevant subspaces among the dimensions of the full space. Being faced with the power set of the set of dimensions a brute force trial of all subsets is infeasible due to their exponential number with respect to the original dimensionality.

In high dimensional data, as dimensions are increasing, the visualization and representation of the data becomes more difficult and sometimes increase in the dimensions can create a bottleneck. More dimensions mean more visualization or representation problems in the data. As the dimensions are increased, the data within those dimensions seems dispersing towards the corners / dimensions. Subspace clustering solves this problem by identifying both problems in parallel. It solves the problem of relevant subspaces which can be marked as redundant in high dimensional data. It also solves the problem of finding the cluster structures within that dataset which become apparent in these subspaces. Subspace clustering is an extension to the traditional clustering which automatically finds the clusters present in the subspace of high dimensional data space that allows better clustering the data points than the original space and it works even when the *curse of dimensionality* occurs. The most of the clustering algorithms have been designed to discover clusters in full dimensional space so they are not effective in identifying the clusters that exists within subspace of the original data space. The most of the clustering algorithms produces clustering results based on the order in which the input records were processed [2].

Subspace clustering can identify the different cluster within subspaces which exists in the huge amount of sales data and through it we can find which of the different attributes are related. This can be useful in promoting the sales and in planning the inventory levels of different products. It can be used for finding the subspace clusters in spatial databases and some useful decisions can be taken based on the subspace clusters identified [1, 2]. The technique used here for indentifying the redundant dimensions which are creating noise in the data in order to identifying the clusters consist of drawing or plotting the data points in all dimensions. At second step the projection of all data points along each dimension are plotted. At the third step the unions of projections along each dimension are plotted using all possible combinations among all number of dimensions and finally the union of all projection along all dimensions and analyzed, it will show the contribution of each dimension in indentifying the cluster which will be represented by the weight of projection. If any of the given dimension is contributing very less in order to building the weight of projection, that dimension can be considered as redundant, which means this dimension is not so important to identify the clusters in given data. The details of this strategy will be covered in later chapters.

Subspace Clustering

Subspace clustering is a newer form of clustering which can find different clusters in subspaces within a dataset. Often in high dimensional data, many dimensions can be *redundant* and can create a noisy data for existing clusters. “Features selection eliminates the redundant and unrelated dimensions by analyzing the whole dataset. Subspace clustering algorithms searches for *relevant dimensions* allowing them to find clusters those exist in multiple overlapping dimensions. This is a particularly important challenge with high dimensional data where the *curse of dimensionality* occurs [3, 4]”.

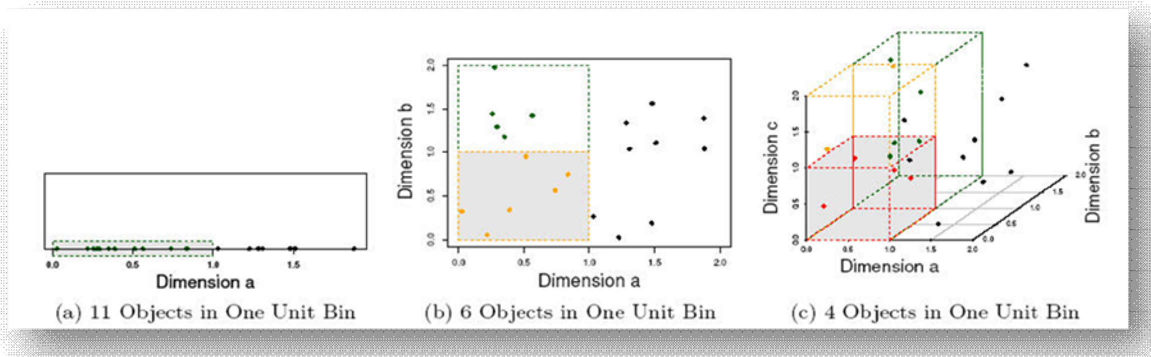
What is subspace clustering?

Automatically identifying clusters present in the subspace of a high dimensional data space that allows better clustering of the data points than the original space. Cluster analysis exposes the groups or clusters of similar groups. “Objects are normally shown as point in multidimensional space. Similarity between objects is often determined by distance measures over the various dimensions in dataset. Changes to existing algorithms are essential to keep up the cluster quality and speed since datasets have become larger and more varied [4]”.

Conventional clustering algorithm gives importance to all dimensions to learn about each object. In high dimensional data, often more dimensions are unimportant and can be considered as redundant. These irrelevant & redundant dimensions can confuse the clustering algorithms by hiding clustering in noisy data. In very high dimensions it is common for all of the objects in a dataset to be nearly equidistant from each other, completely masking the clusters. Feature selection methods have been working somewhat successfully to improve cluster quality. These algorithms find a subset of dimensions on which to perform clustering by removing irrelevant and redundant dimensions. Unlike feature selection methods which examine the dataset as a whole, subspace clustering algorithms localize their search and are able to uncover clusters that exist in multiple, possibly overlapping subspaces [5, 6].

“Another thing with which clustering algorithms fight is the *curse of dimensionality* [6]”. As the number of dimensions in a dataset increases, distance measures become increasingly worthless. Additional dimensions spread out the points until, in very high dimensions; they are almost equidistant from each other. Figure 4.1 illustrates how additional dimensions spread out the points in a sample dataset. The dataset contains 20 points arbitrarily placed between 0 and 2 in each of three dimensions. Figure 4.1(a) shows the data projected on one axis. The points are close together and are about half of them in a one unit sized area [6]. Figure 4.1(b) shows the same data in extended form into the second dimension. By adding another dimension, points are spread out along another axis, pulling them further apart. Now only about a quarter of the points fall into a unit sized area. In Figure 4.1(c) a third dimension is added which spreads the data further apart. A one unit sized bin now holds only about one eighth of the points. If we continue to add dimensions, the points will continue to spread out until they are all almost equally far apart and distance is no longer very important. The problem is made worse when objects are related in different ways in different subsets of dimensions [6]. It is this type of relationship that subspace clustering algorithms seek to uncover. In order to find such clusters, the irrelevant features must be removed to allow the clustering algorithm to focus on only the relevant dimensions. Clusters found in lower dimensional space also tend to be more easily interpretable, allowing the user to better direct further study [6].

Subspace clustering is also more general than feature selection in that each subspace is local to each cluster, instead of global to everyone. It also helps to get smaller descriptions of the clusters found since clusters are defined on fewer dimensions than the original number of dimensions. An example of subspace clustering can be in bioinformatics with DNA micro array data. One population of cells in a micro array experiment may be similar to another because they both produce chlorophyll, and thus be clustered together based on the expression levels of a certain set of genes related to chlorophyll. However, another population might be similar because the cells are regulated by the circadian clock mechanisms of the organism. In this case, they would be clustered on a different set of genes. These two relationships represent clusters in two distinct subsets of genes. These datasets present new challenges and goals for unsupervised learning. Subspace clustering algorithms are one answer to those challenges. They excel in situations like those described above, where objects are related in multiple, different ways.



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

Figure 1. Dimension

Why subspace clustering?

Clustering is a great data exploring methodology which is able to identify previously unknown patterns in data. Subspace clustering is an extension of conventional clustering, based on the observation that different clusters (groups of data points) may exist in different subspaces within a given dataset. This point is particularly important with respect to high dimensional data where the curse of dimensionality can occur and can reduce the worth of the results. Subspace clustering is important due to following points.

- Most of the clustering algorithms have been designed to identify clusters in the full dimensional space so they are not effective in identifying clusters that exist in subspaces of the original data space [2].
- Many times the data records contain some missing objects. Such missing objects are normally replaced with objects taken from given distribution [2].

The clustering results formed by most of the clustering algorithms rely a lot on the order in which input records are processed [2, 8].

Dimension Projection Theory

In this section a theory of dimension projection is described. Just for an example for understanding, only three dimensions are considered at first. In the first step, different data points are taken as input in which it is known that two clusters exist and is visible in all three dimensions and it is assumed that data is normalized in all of dimensions. The third cluster is visible in 2 dimensions but not visible in third dimension. For convenience the coordinates of all axis or dimensions are normalized.[7] The input data for the given example is described in the Table 1. The Table 2 shows the weights of the coordinates along all of the dimensions. Weights are calculated in such a way that the number of points are counted along each dimension on each and every coordinate and then total count becomes the weight of the coordinate along that dimension. Like Table 2 shows that the coordinate 5 (5th index along X-Axis) has 2 points so its intensity of weight is 2. The input data belongs to the three dimensions. There are three known clusters exist in the data. The two of them are visible in all three dimensions (Cluster 1 and Cluster 2). But the third cluster (Cluster 3) is visible only in two dimensions (X and Y-Axis) and is dispersed in third dimension (Z-Axis) and is not clearly visible in this dimension.

Drawing the points in 3 dimensional space XYZ plane will look something like as showing in Figure 2. The points in XY, YZ and XZ planes are also shown in Figure 3, Figure 4 and Figure 5 respectively. The Figure 3 which is showing points in

XY plane shows that three clusters exists but Figure 4 and Figure 5 are showing that there are two clusters only and remaining data points are scattered along Z-Axis which are not forming any cluster.

The shaded area in blue color is representing the projection of data points along X-Axis, the shaded area in the brown color is representing the projection of data points along Y-Axis and the shaded area in the red color is representing the projections of the data points along Z-Axis. The height of the shaded area shows the weight of the projection at that specific coordinate along that dimension. If the data is being visualized in three dimensions then it is observed that there are only two clusters present in the data as shown in Figure 2. The Figure 6 is also showing the overlapping of projections along X and Y-Axis but Z-Axis is has very less overlapping of projections of data points. The existence of more clusters needs to be checked since more clusters might be present but not visible due to the presence of some dimension(s). Dimensions will be removed one by one and then data will be visualized for having more clusters and process will end up until data is visualized by removing all dimensions for all possible combinations and the numbers of clusters are also stored in some data structure to analyze later. The visualization is assumed in this project as an automatic process of finding the number of clusters in that given dimension. If one dimension is removed which is X-Axis in this case then the projections of the data points along YZ dimensions as follows.

Table 1. Intensity of weight

Cluster 1			Cluster 2			Cluster 3		
X	Y	Z	X	Y	Z	X	Y	Z
5	21	29	19	5	4	13	18	4
5	23	27	19	7	2	13	19	1
6	21	28	19	9	1	14	16	16
6	22	28	20	6	3	14	17	30
6	24	26	20	8	3	14	18	25
7	20	30	21	5	4	14	20	10
7	21	27	21	7	4	15	17	27
7	23	28	21	8	2	15	19	12
8	21	30	21	9	1			
8	23	29	22	6	3			
8	24	28	22	8	3			
9	22	26						
9	24	26						

Table 2. Projection

X – Projection		Y – Projection		Z – Projection	
Coordinate	Weight	Coordinate	Weight	Coordinate	Weight
5	2	5	2	1	3
6	3	6	2	2	2
7	3	7	2	3	4
8	3	8	3	4	4
9	2	9	2	10	1
13	2	16	1	12	1
14	4	17	2	16	1
15	2	18	2	25	1
19	3	19	2	26	3
20	2	20	2	27	3
21	4	21	4	28	4
22	2	22	2	29	2
		23	3	30	3

		24	3		

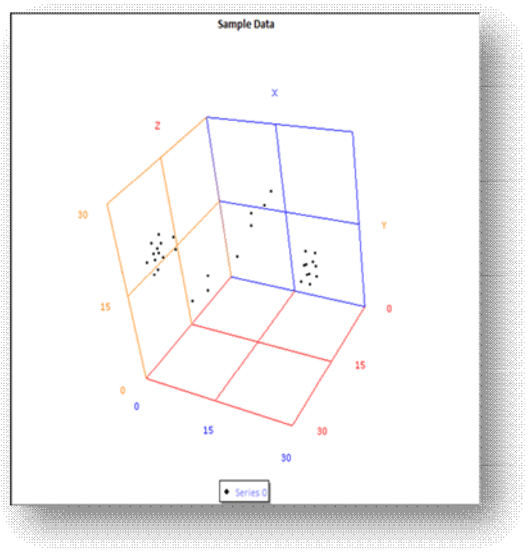


Figure 2

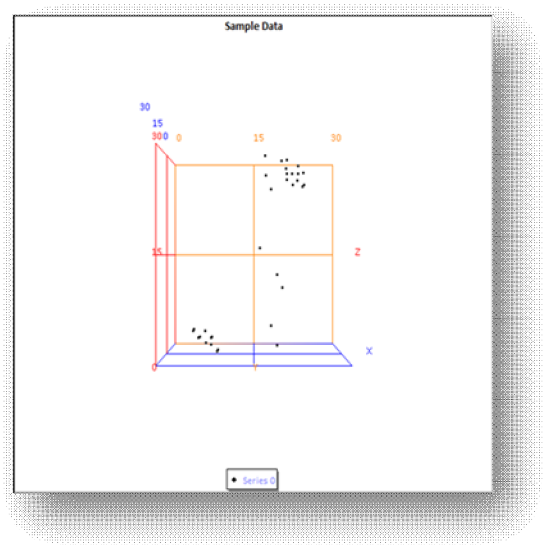


Figure 4

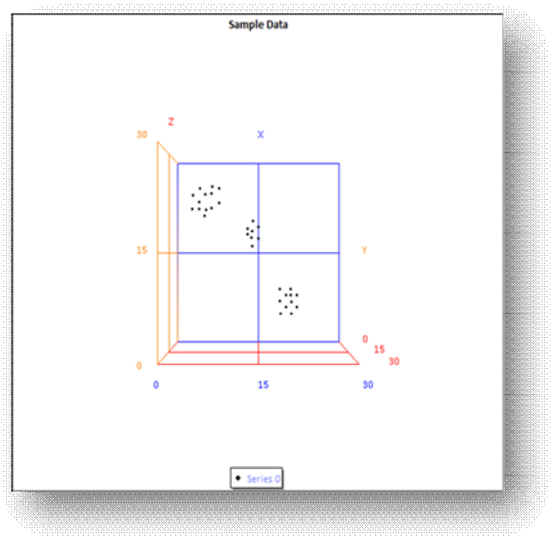


Figure 3

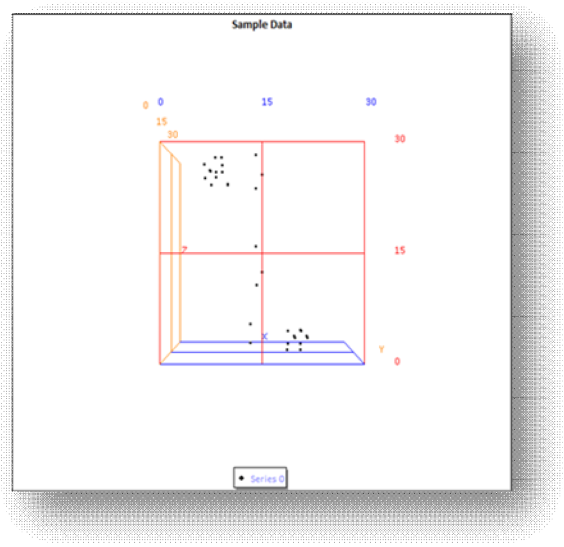


Figure 5

If the projections of data points are drawn along each dimension (in this case X-Axis, Y-Axis and Z-Axis) then accumulative projections of data points will be shown as in Figure 6.

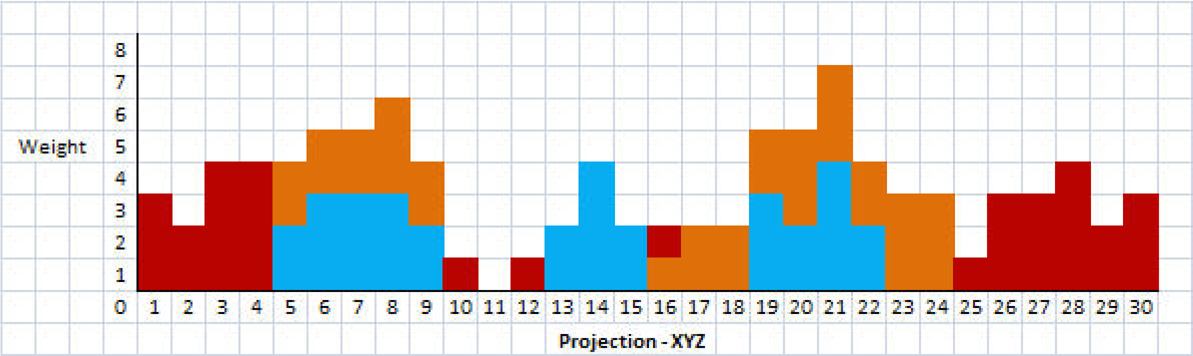


Figure 6

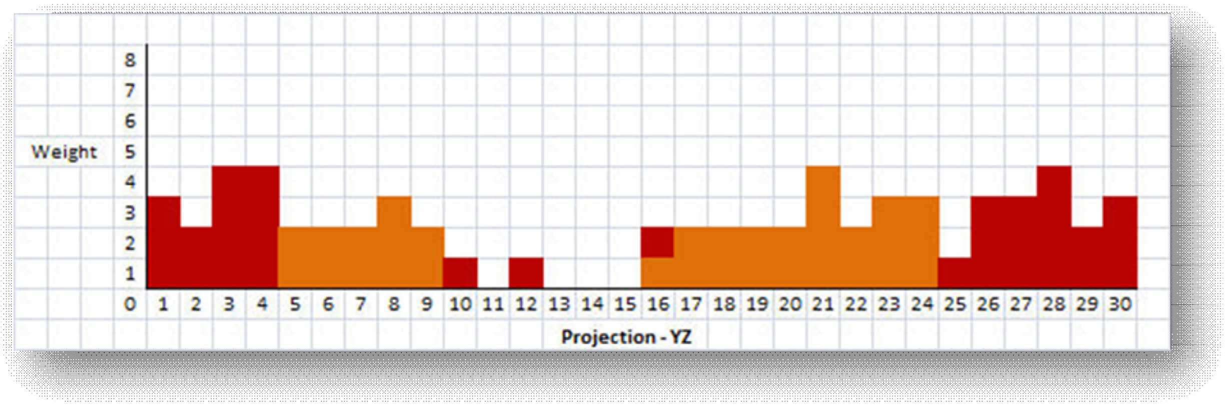


Figure 7

The Figure 7 shows that there is no overlapping of projection of data points when X-Axis is removed but only single point is overlapping which is at 16th coordinate. So removing X-Axis might not be that effective in identifying more clusters in the data since Figure 4 is also showing the two clusters. Now if another dimension which is Y-Axis is removed to check for more clusters by visualizing the data then the projections of data points along XZ dimensions will be as follows.

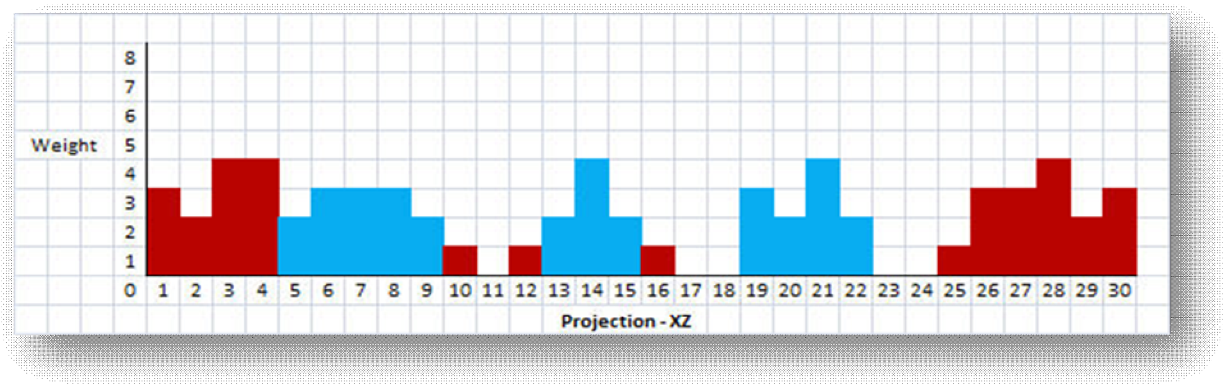


Figure 8

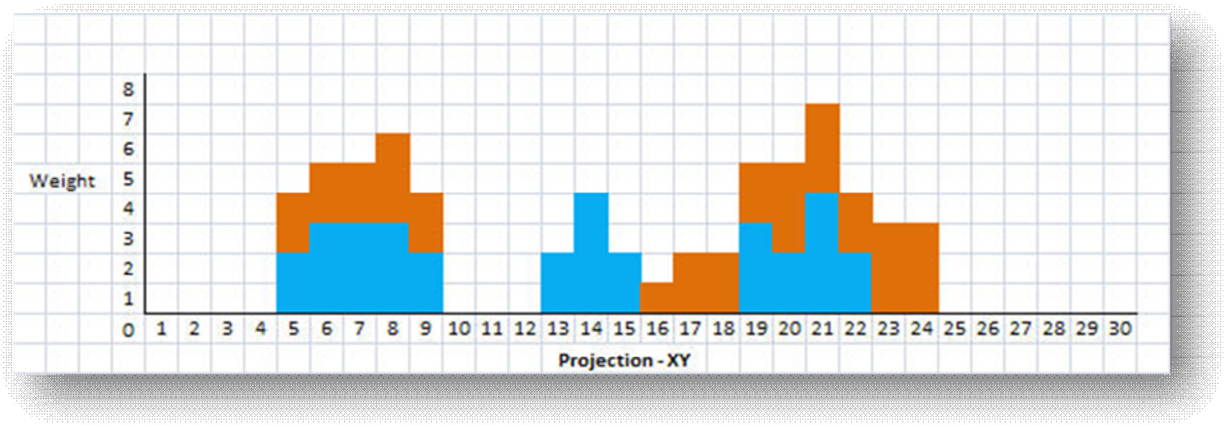


Figure 9

The above Figure 8 is showing that there is no projection overlapping exists by removing the Y-Axis and then plotting the projections of data points. Figure 5 is also plots the data points along XY dimensions and showing the existence of only two clusters. So removing Y-Axis is also not contributing in finding the more clusters in the given dataset. If we remove another single dimension which is Z-Axis and visualize the data, projections of the given data points will be shown as follows.

The Figure 9 is showing that that projection overlapping of the data points along XY dimension is larger than previous figures and most of the points are overlapping. If data is visualized in these two dimensions as in Figure 3, it is shown that another cluster becomes visible which was not present previously. So after removing the Z-Axis, another cluster becomes visible so removing Z-Axis is contributing a lot in finding more clusters in the given dataset.

The procedure doesn't stops here. Now combinations of the dimensions are going to be removed to check further existence of clusters. If the X-Axis and Y-Axis are removed then the projection of data points along Z-Axis will be drawn as follows in Figure 10.

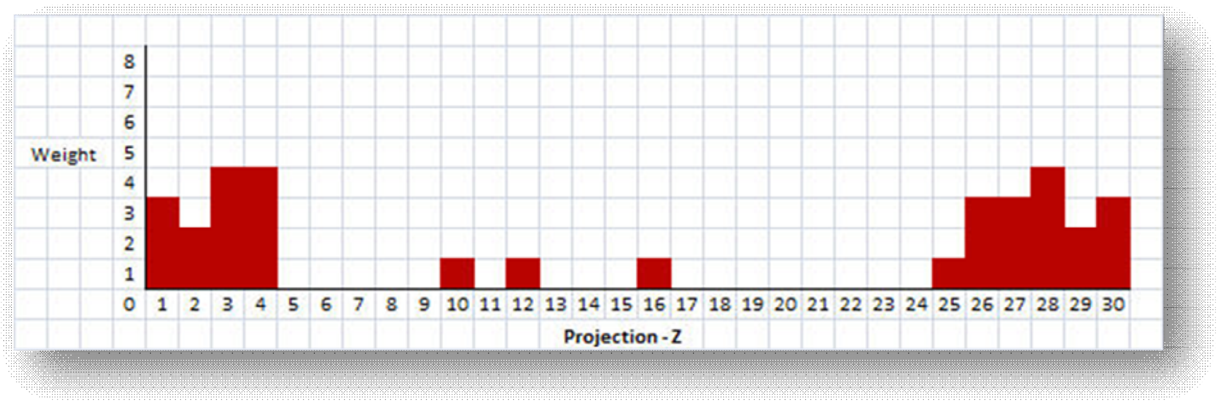


Figure 10

The Figure 10 is showing that data points are mostly scattered along Z-Axis and mostly not adjacent to each other. Visualizing the data in only Z-Axis is not yielding more clusters so this dimension is no so important and removing X-Axis and Y-Axis might not contribute in cluster identification. Now if another combination of dimensions which is X-Axis and Z-Axis is removed and projection of data points are drawn along Y-Axis then figure will be something like below.

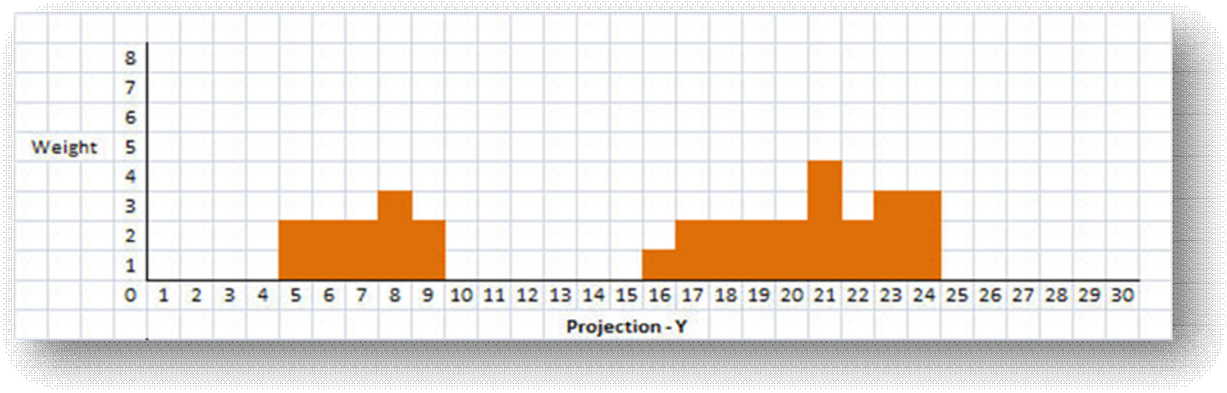


Figure 11

Figure 11 is showing the solid segment of projections which is representing the presence of clusters when data is visualized along Y-Axis only so this dimensions can be considered important in cluster identification. Finally if we remove Y-Axis and Z-Axis and projection are drawn only along X-Axis then it will be look like below in Figure 12.

In Figure 12, it is clear that along X-Axis projections of data points have more weight and data points are more adjacent and hence more chance of existence of clusters. If data is visualized along X-Axis only then it is clear that there are three clusters exist so this is also an important dimension in fining the clusters in the given data set. From the above analysis it is clear that the Z-Axis is the only dimension which seems redundant and can be removed to discover more clusters available in the given dataset. In original data there were only two clusters but by removing one dimension one more cluster is found and adding Z-Axis dimension in the data is causing to create a noise in the data and hindering the clusters identification. This procedure was used for three dimension data but it can be used as it is for n-dimensional data as well and the projection drawn in this procedure also helps to find out number of clusters within the data with respect to all dimensions.

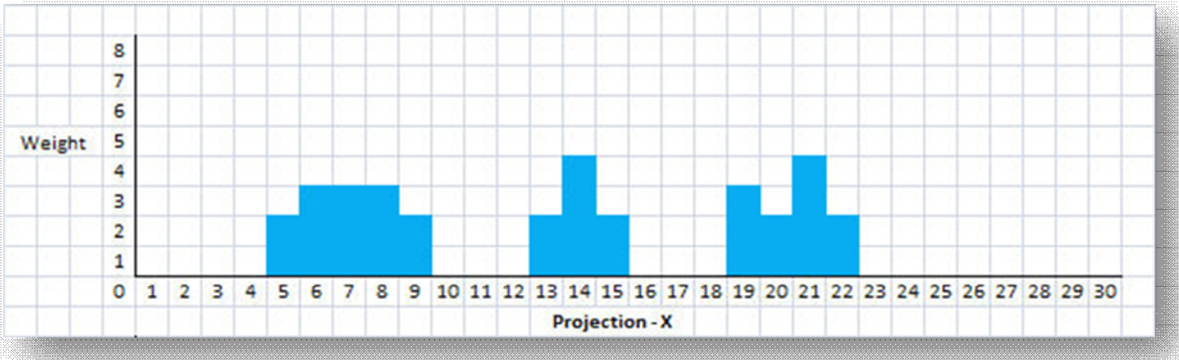


Figure 12

Algorithm

The above technique can be written in as an algorithm form.

1. Normalize the dimensions so that all dimensions should have same coordinates limit.
2. Generate the accumulative union of all the projections taken in all dimensions in such a way that weights of the final output projection on each coordinate will show the sum of all total number of data points at coordinate.
3. Visualize the data and identity number of clusters in the presence of all dimensions and store them in some data structure (visualization is assumed as an automatic process in this project)
4. Remove the dimensions one by one and identify the number of clusters available after removal and store them in data structure.
5. Remove all possible combination of dimensions which will 2^n and identify the number of clusters after removal store them in data structure.
6. Analyze the data structure with respect to the dimension combination and mark that dimension as redundant which has less number of clusters in its presence.

This algorithm is efficient for less number of dimensions. As long as the dimensions are increased in the data, the algorithm efficient will become poor. Visualization of the data is also the beyond of the scope of this project but implantation of the algorithm is based on this idea.

Conclusion

In high dimensional data there can be number of clusters exists and there are different techniques or methodologies introduced to identify them with respect to different dimension and in that techniques sometimes it is figured out that some of the dimensions might not be so important in identifying cluster because of the dispersed data in that dimension and creating a noise in the data for indentifying clusters. The above technique uses the Projection of data points along each dimension to determine the weight or intensity of the data points along each dimension and visualization process to identify the number of cluster present in the data along that particular dimension. Weight is calculated by counting the number of data points along that dimension. By removing the dimensions one by one and then all possible combinations of dimensions and then visualizing the data, it makes clear that which dimension is less important in order to find out the clusters in given subspace. The algorithm written has limitations of performance as it can work well with less number. of dimension and will become slow in high dimensions. Visualization process is considered as an automatic procedure which will return number of clusters given in the dataset. At the end the dimension which can be redundant can be removed which is creating noise and has less contribution in cluster identification in high dimensional data.

References

1. Removing Dimensionality Bias in Density-based Subspace Clustering, by Ira Assent, Ralph Krieger, Emmanuel Müller, Thomas Seidl Dept. Computer Science 9 (data management and data exploration) RWTH Aachen University, Germany
2. Deign & Analysis of subspace clustering algorithms and their applicability by Jyoti D. Pawar, Goa university India
3. Statistical subspace clustering, GRAppA - Charles de Gaulle University - Lille 3, Pertinence - 32 rue des Jeuneurs -75002 Paris
4. Evaluating Subspace Clustering Algorithms by Lance Parsons, Ehtesham Haque, Huan Liu, Department of Computer Science Engineering, Arizona State University, Tempe, AZ 85281
5. VISA: Visual Subspace Clustering Analysis by Ira Assent, Ralph Krieger, Emmanuel M¨uller, Thomas Seidl, Data management and data exploration group, RWTH Aachen University, Germany
6. Subspace Clustering for High Dimensional Data: A Review by Lance Parsons, Ehtesham Haque, Huan Liu, Department of Computer Science Engineering Arizona State University Tempe, AZ 85281
7. Data Mining Concepts and Techniques Second edition by Jiawei Han and Micheline Kamber
8. Predictive Data Mining a practical guide by Sholom M. Weiss and Nitin Indurkhya
9. Data Mining Practical Machine learning tools and techniques by Ian H. Witten and Eibe Frank
10. Web Mining Research: A Survey by Raymond Kosala, Hendrik Blockeel Department of Computer Science Katholieke Universiteit Leuven Celestijnenlaan 200A, B3001 Heverlee, Belgium

1/18/2011