Volume 6, Issue 2, February 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Method for Cluster Visualization using Geometrical Transformations

Malay K. Pakhira[*]

Department of computer Science and Engineering

Kalyani Government Engineering College, Nadia, West Bengal, India

Abstract— Data visualization is a technique that lets users to view data in order to help better understanding the information content within the data, and also to assist in any data related decision making process in the context of a related business area. Data mining processes should be suitably augmented with a desired type of data visualization technique. In this work, a novel approach is presented to visualize data, clustered by the k-means algorithm, in three dimensions by use of geometrical transformations over the concerned clusters. Relevance of the process is described with suitable example data sets.

Keywords— Clustering, data mining, data visualization, geometrical transformation.

I. INTRODUCTION

Data Visualization is an important technique in the fields of Data Analysis and Data Mining. An effort to visualize data may help users to determine significance of any data item, and correlation between collective data items. In Data mining applications visualization of clustered data is very important. This helps identifying any specific data item of importance, determining an outlier, or, sparse and dense areas of clusters etc. Data Visualization enables users to create graphical and often interactive representations of data sets which highly improve business analytic efforts and organization productivity. If it is not possible to visualize a very large data set at a time, a subset of the complete data can always be visually analysed. Therefore, data visualization is a process that helps people understand the significance of data in a visual context. Correlations among patterns in a pattern set, that might go undetected in text-based form, can be better exposed and recognized with a suitable data visualization technique.

A large number of clustering and classification methodologies [1–3] are used in the above said fields. Every methodology has its own characteristics and accordingly, outputs produced by them are also different. However, these differences may be desired, in some cases, because developers of the concerned algorithms desired to highlight certain features of the output clusters. Interrelationship between different features of the data sets and its clusters can be properly studied with a visual representation of the same. Selection of important and significant features, i.e., dimensionality reduction is also possible in this way.

Although, there exists a large number of clustering algorithms, the classical *k*-means algorithm [4] is still considered to be superior over almost all others. This is because of its simplicity, either in logic or in computational efficiency for moderate sized data sets. However, at present Data Mining applications are in focus of majority of clustering algorithms. Since data mining processes involve very large data sets, the classical *k*-means algorithm may not be very suitable under such scenario[5]-[11]. It is observed that the classical *k*-means algorithm has an approximate time complexity of $O(n^2)$. Therefore, for large data sets, this high time complexity debars the algorithm from efficient usage. Very recently a new version of the *k*-means has been developed, that can execute approximately in linear time, i.e., $O(n)$, under certain specific conditions [11]. This algorithm can handle large data sets quite efficiently.

In this work, we shall describe a cluster visualization technique which involves appropriate use of geometrical transformation techniques considering each clustered data point as an object. Specifically, clusters are translated for allowing sufficient separation among themselves (but without any distortion in data distribution within the cluster) in 3D representation over three selected features, clusters are rotated about a desired feature axis. The rotation process creates a number of circular boundaries for each clustered data point which clearly depicts the contribution of the point to the concerned cluster.

II. APPLYING GEOMETRICAL TRANSFORMATIONS OVER *K*-MEANS CLUSTERS

In this section, we shall show how we can apply geometrical transformations (basic transformations only, i.e. Translation or shifting, Scaling and Rotation) over the clusters generated by the k-means clustering algorithm, in order to improve the visibility of the concerned clusters. It is to be mentioned here that, although we are using results of *k*-means algorithm only for our experiments, results of any other clustering algorithm may be used without any problem.

The classical *k*-means algorithm iterates through the process of distribution of data items, based on the minimum distance to a pre-computed cluster representative (cluster center), and followed by re-computation of cluster centers.

After a certain number of iterations, the process terminates with a set (*K* in size, for a *K* cluster situation) of stable cluster centers.

Here, the *k*-means is augmented by a shifting phase. After completion of the *k*-means, we shall move the clusters, in the outward direction (away from the global data center), by a small distance. Fig. 1 illustrates the process of shifting and its effects visually. The original data has three different clusters having data labels '*****', '**+**' and '**o**' (shown in bold fonts). Cluster centers are labeled as A, B and C respectively. O is the global data centre. Cluster boundaries are represented by bold circles. After shifting in the outward direction we get A', B' and C' as new cluster centers. The shifted clusters and the corresponding data items are shown in lighter shade. Here, shift amounts are proportional to the original distance of a cluster from the global center. That is,

$$|\mathrm{AA}'| \infty |\mathrm{OA}|, \quad |\mathrm{BB}'| \infty |\mathrm{OB}|, \quad |\mathrm{CC}'| \infty |\mathrm{OC}|$$

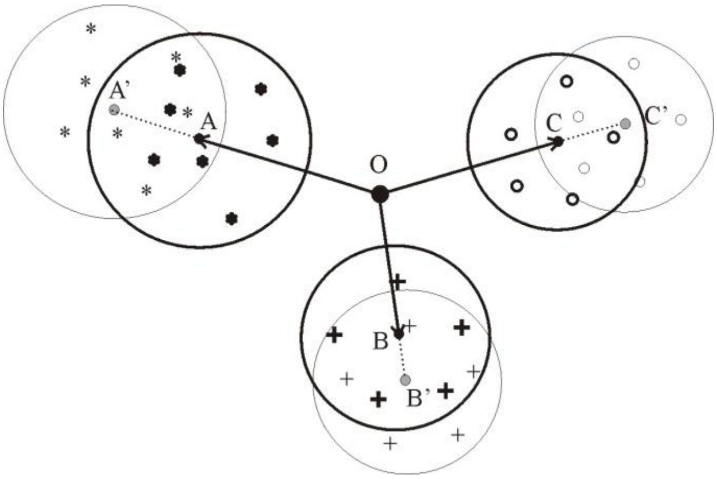


Fig. 1. Three *k*-means clusters in 2-dimension before and after shifting about the global center

This kind of directional shifting does not incorporate any deformation within the clusters, the between cluster separations are improved only. This following analysis may be found in [10], [11], and presented here for demonstration purposes.

For this purpose, we need to calculate direction of movement for each cluster, and to determine the amount of movement. Let us consider a data set **S** in *d*-dimension having *N* items in *K* number of clusters. Let, $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ be the *i*th data item for *i* = 1 to *N*; $\mathbf{z}_k = (z_{k1}, z_{k2}, \ldots, z_{kd})$ be the center of the *k*th cluster, for *k* = 1 to *K*. If $\mathbf{z} = (z_1, z_2, \cdots, z_d)$ be the global center of the complete data set, we can calculate direction vectors for movement of individual clusters as $\mathbf{g}_k = (g_{k1}, g_{k2}, \ldots, g_{kd})$ *k* = 1 to *K*, such that

$$\mathbf{g}_k = (g_{k1}, g_{k2}, \ldots, g_{kd}) = \mathbf{z}_k - \mathbf{z} = (z_{k1} - z_1, z_{k2} - z_2, \ldots, z_{kd} - z_d)$$

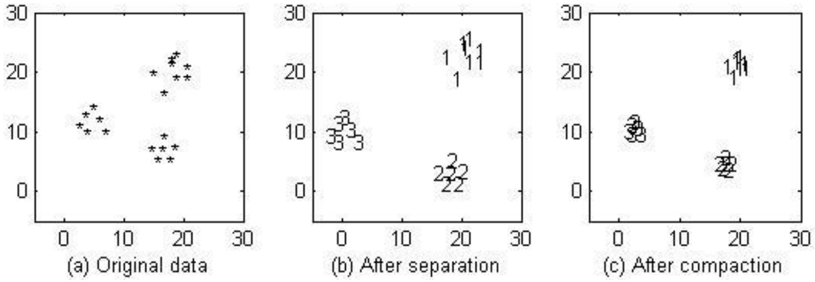
The amount of shift is considered a small fraction, say α, of this direction vector. So, during this phase, the *k*th cluster will be shifted by an amount **s** = α × **g**$_k$. The corresponding new center vector will become $\mathbf{z}_k^{'} = (z_{k1}^{'}, z_{k2}^{'}, \ldots, z_{kd}^{'}) = \mathbf{z}_k$ + α × **g**$_k$. Along with the centers, clustered elements are also shifted by the same amount. So, actual shape and compactness of the clusters are not degraded, but separation between them will be improved. Hence, we can achieve compact and separate clusters.

Let us now consider the situation of compaction of clusters after initial clustering. In this case, cluster centers will not be moved away from the global data centers, instead elements of individual clusters will approach the corresponding centers. Here also, $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ is the *i*th data item for *i* = 1 to *N*, and $\mathbf{z}_k = (z_{k1}, z_{k2}, \ldots, z_{kd})$ is the center of the *k*th cluster for *k* = 1 to *K*. Since $\mathbf{z} = (z_1, z_2, \cdots, z_d)$ is the global center of the complete data set, we can calculate direction vectors for movement of individual data elements of a particular cluster (say, *j*) as $\mathbf{h}_i = (h_{i1}, h_{i2}, \ldots, h_{id})$ for *i* = 1 to *N*, such that

$$\mathbf{h}_i = (h_{i1}, h_{i2}, \ldots, h_{id}) = \mathbf{z}_j - \mathbf{x}_i = (z_{j1} - x_{i1}, z_{j2} - x_{i2}, \ldots, z_{jd} - x_{id}).$$

Again, the amount of shift is considered a small fraction, say α, of this direction vector. So, at the end of the process, the *i*th element of the *j*th cluster will be shifted by an amount **s** = α × **h**$_i$ toward its own representative. The corresponding new pattern vector will become $\mathbf{x}_i^{'} = (x_{i1}^{'}, x_{i2}^{'}, \ldots, x_{id}^{'}) = \mathbf{x}_i +$ α × **h**$_i$. Here, clustered elements are shifted toward corresponding local centers. So, compactness values of the clusters are improved without changing separation between local centers. This shifting of data elements is done only once, after the *k*-means is over, resulting in compact and separate clusters.

In Fig. 2, an example with a very simple two dimensional data set *Sample* used for illustration. We have applied the above mentioned shifting approach to improve the compactness and separation of three clusters, and to observe the visual impact of the results.



(a) Original data (b) After separation (c) After compaction

Fig. 2. *Sample* data set: (a) before *k*-means, (b) after applying separation, and (c) after applying compaction

The original distribution of (clustered) data is shown in Fig. 3(a), and the effect of cluster shifting in Fig. 3(b), and that of data compaction in Fig. 3(c), applied over the clustered data. One threat with the compaction improvement technique is that data within a cluster itself may coincide with each other, and this may eventually lead to singleton clusters. Moreover, within cluster data distribution is also affected. To avoid this, separation improvement is a safer technique.

In the above, we have shown the translation (shifting) of clusters from each other using specific relations so that within cluster data distribution remains unchanged. Now for the purpose of visualization, we can rotate any 3D display of clusters about a particular axis. For the purpose of cluster rotation, we have used the concept of homogeneous geometrical transformation, and rotated each element of a cluster about a selected axis. The rotation operation forms a circle, for each of the data items, on a plane formed by the other two axes. For example, if we rotate about the Z-axis, circles will be formed on the X-Y plane. The circular representation of a data item seems more useful in revealing inherent structure of the concerned cluster. The 3D visual representation seems to be useful for selection of interesting data items such as elements that cause overlap between clusters, or an outlier element etc. Such application specific data selection is an important operation in Data Mining. While outlier data items are rejected in clustering, they may be very interesting to a data miner. An interactive visual platform is very useful for this purpose. In this work our objective is thus better visualization of clustered data instead of clustering quality.

III. EXPERIMENTAL RESULTS

For visualization of clusters using the present method, we have used three well-known real-life data sets, viz., *Iris*, *Wine* and *Vote* data sets. These data sets and their detailed descriptions are available in the UCI database [12], [13]. A brief tabular description is presented here in Table I. Experimental results are shown in Figures 3 through 10. For the *Iris* and the Wine data sets we have used shift factor value of 0.5 in each case. Rotated views of shifted clusters are shown about X, Y and Z axes. We can select any of the features as X, Y and Z axes for a data item having at least 3 features. For the highly overlapped Vote data set, we have shifted clusters by factors of 0.5 and 5.0 as shown in Figures 9 and 10. Rotation about only the Z axis is shown here.

Table I Brief Description of Data Sets Used

Data Set	**Number of Elements**	**Dimension**	**Expected no. of Clusters**	**No. of elements in clusters**
Iris	150	4	3	50, 50, 50
Wine	178	16	3	71, 69, 48
Vote	435	13	2	168, 267

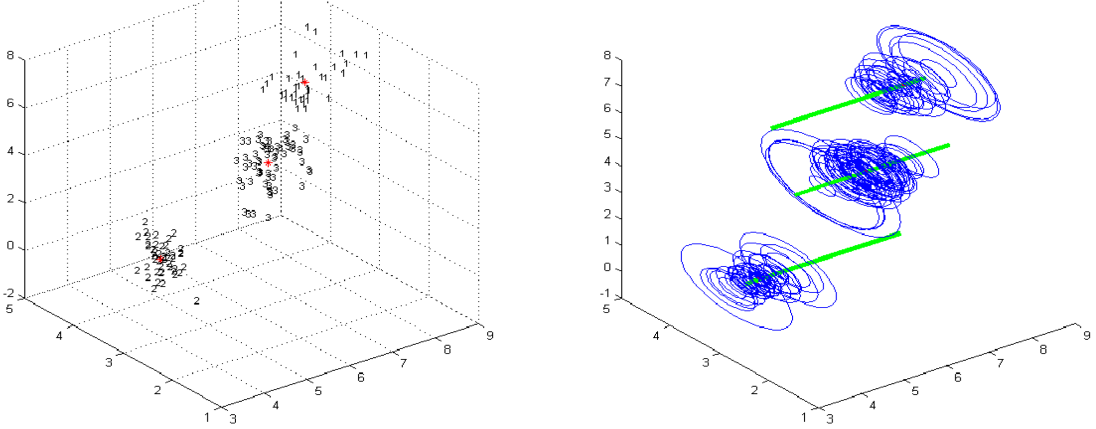


Fig. 3. *Iris* data plotted using features 1, 2, 3 as X, Y and Z axes, and rotated view of 3 clusters about X-axis for alpha = 0.5

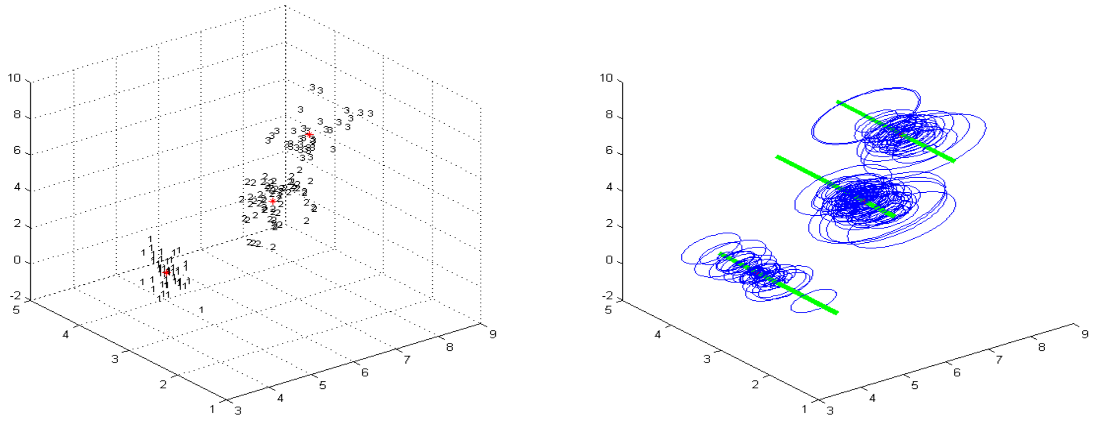


Fig. 4. *Iris* data plotted using features 1, 2, 3 as X, Y and Z axes, and rotated view of 3 clusters about Y-axis for alpha = 0.5

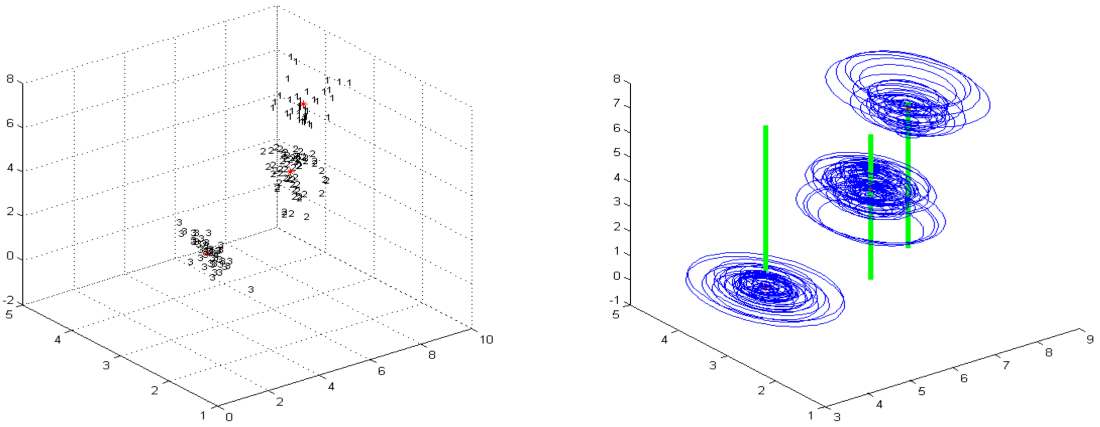


Fig. 5. *Iris* data plotted using features 1, 2, 3 as X, Y and Z axes, and rotated view of 3 clusters about Z-axis for alpha = 0.5

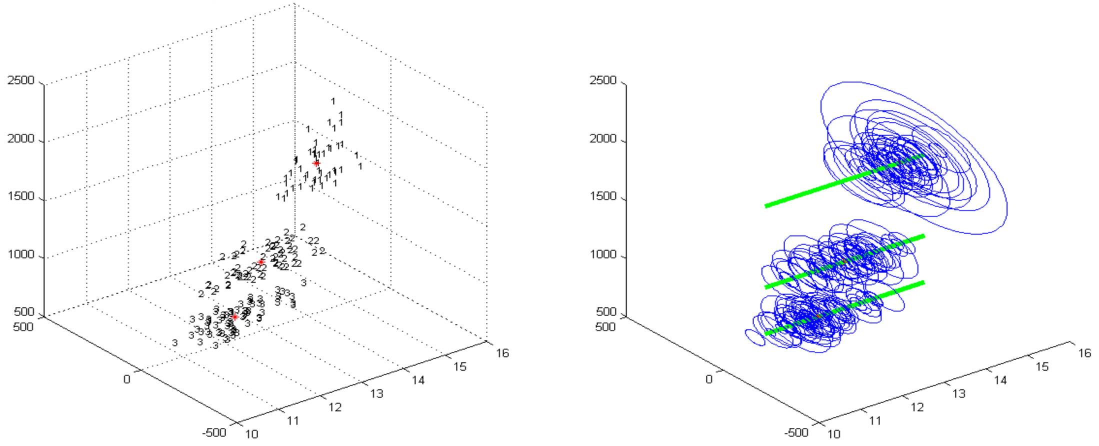


Fig. 6. Wine data plotted using features 1, 2, 13 as X, Y and Z axes, and rotated view of 3 clusters about X-axis for alpha = 0.5

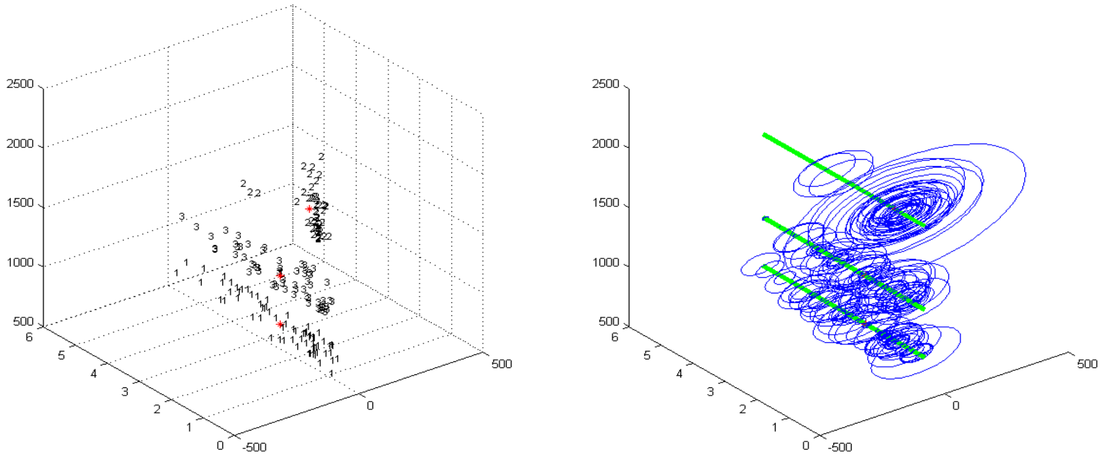


Fig. 7. Wine data plotted using features 1, 2, 13 as X, Y and Z axes, and rotated view of 3 clusters about Y-axis for alpha = 0.5

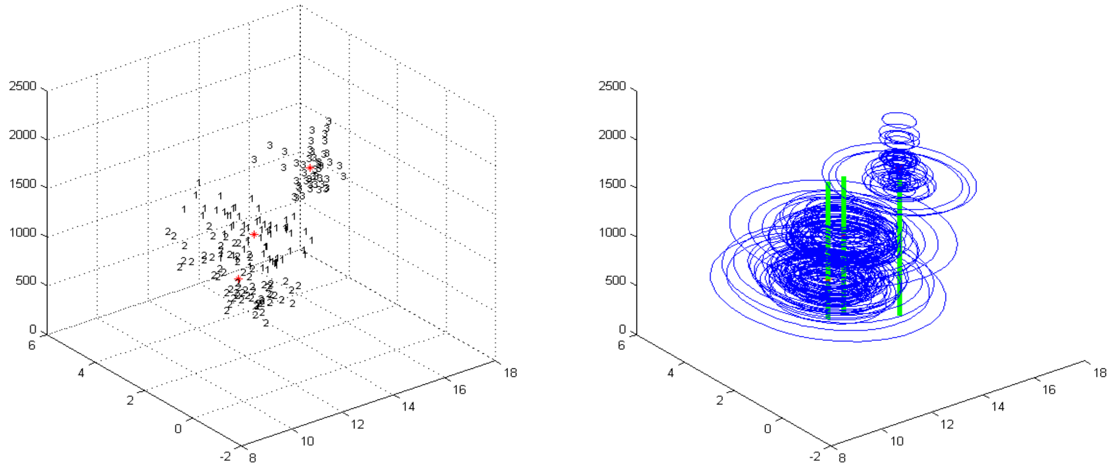


Fig. 8. Wine data plotted using features 1, 2, 13 as X, Y and Z axes, and rotated view of 3 clusters about Z-axis for alpha = 0.5

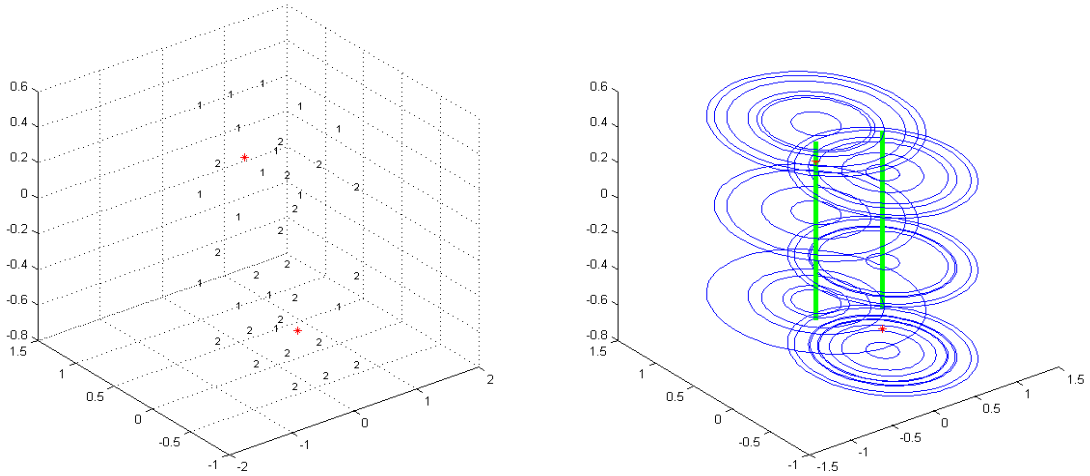


Fig. 9. Vote data plotted using features 1, 2, 12 as X, Y and Z axes, and rotated view of 3 clusters about Z-axis for alpha = 0.5

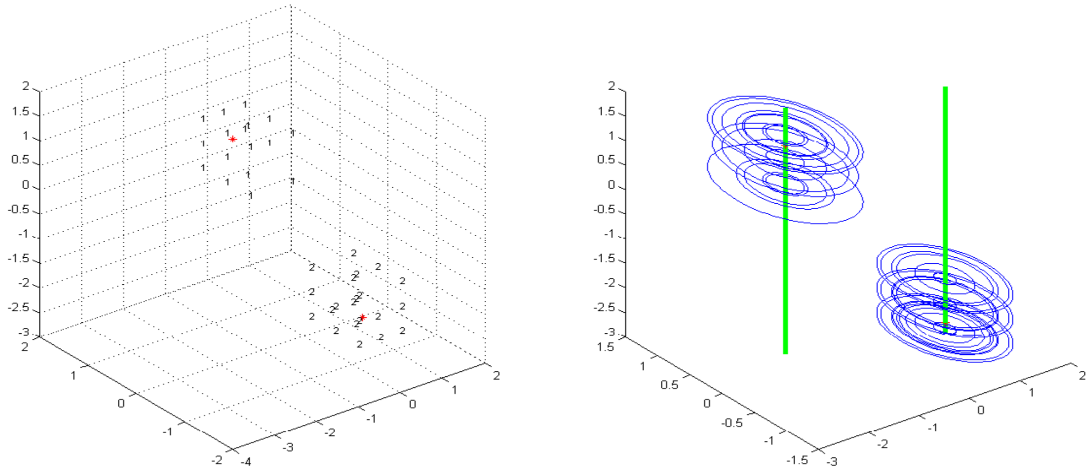


Fig. 10. Vote data plotted using features 1, 2, 12 as X, Y and Z axes, and rotated view of 3 clusters about Z-axis for alpha = 5.0

For the Vote data set, we have shown rotated views of clusters because this data set has two highly overlapped clusters. It is observed that by increasing the shift factor values, we can get distinct views of individual clusters.

In Figures 11 to 13, we have shown results of another well-known data visualization technique called X-DAT (parallel axis display of data). X-DAT stands for X-dimensional visualization of data. In the well-known KDnuggets database [14], a number of such visualization techniques are present. Some other techniques are Miner3D (3D View of Data Mine), CViz (Cluster Visualization) etc. Almost all of the techniques present 2D or 3D plots of selected features and samples. We have used the X-DAT results for comparison purposes. The X-DAT displays all the dimensions directly, but is able to correlate only two of the dimensions simultaneously. It is observed that, while results of X-DAT enables us to select proper attributes for some data mining applications, the present algorithm provides a true 3D appearance on a 2D surface, and helps in selection of outlier elements or interesting information of some aspect of the data mining application. It is notable that, in data mining we generally try to view only a limited number of features of the data set concerned. In our case, only 3 dimensions can be viewed at a time. We may select 3 arbitrary features and the axis of rotation earlier. In each of the figures, from Fig. 3 to Fig. 10, there are two components: one left and one right. The left components are nothing but a 3D scatter plot of the clustered and shifted data. The right components are rotated views of the clusters about a selected axis.

An interesting difference between results of X-DAT and the present algorithm is illustrated for the Vote data set. Here, result of the former method fails to provide any useful information regarding inter-relationship between any two attributes, but the present algorithm clearly describes features of the two concerned clusters. In Fig. 12, result of Wine data by the X-DAT algorithm is truncated in order to visually treat some of the important features only.

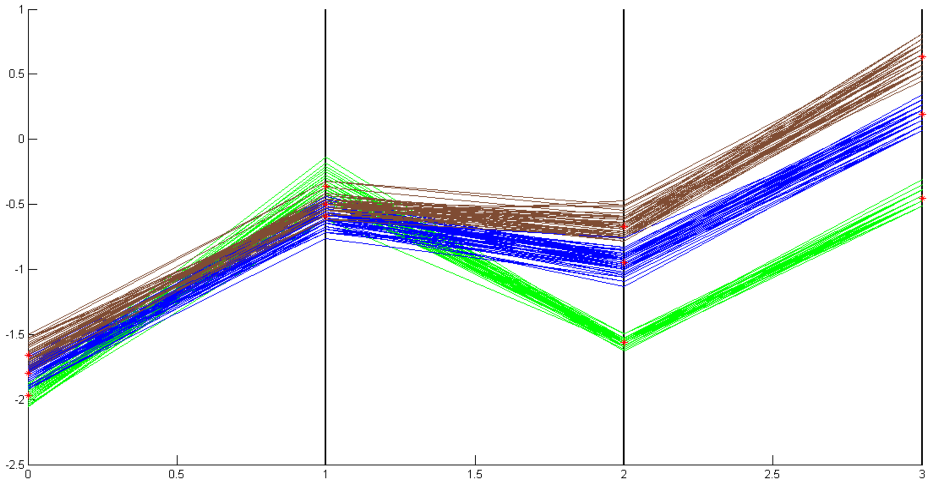


Fig. 11. Clustered *Iris* data plotted using all 4 features as different axes, for shift factor alpha = 5.0

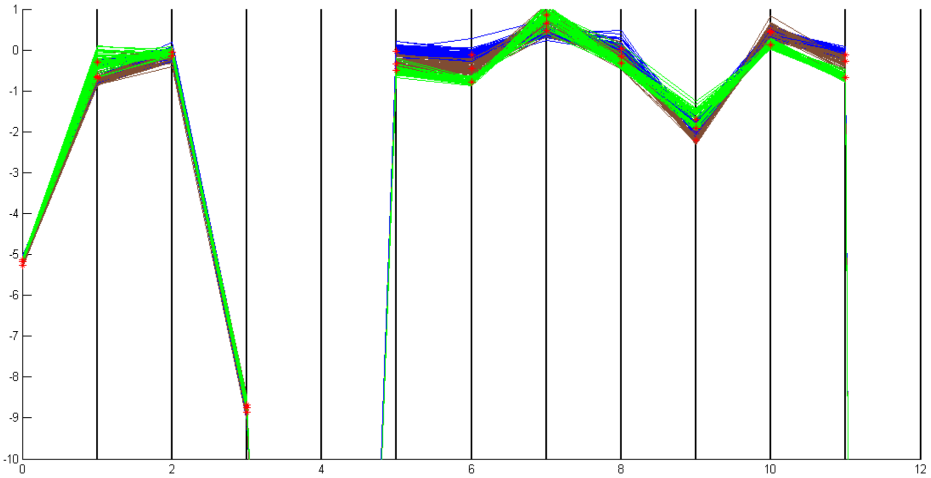


Fig. 12. Clustered *Wine* data plotted using all 13 features as different axes, for shift factor alpha = 5.0

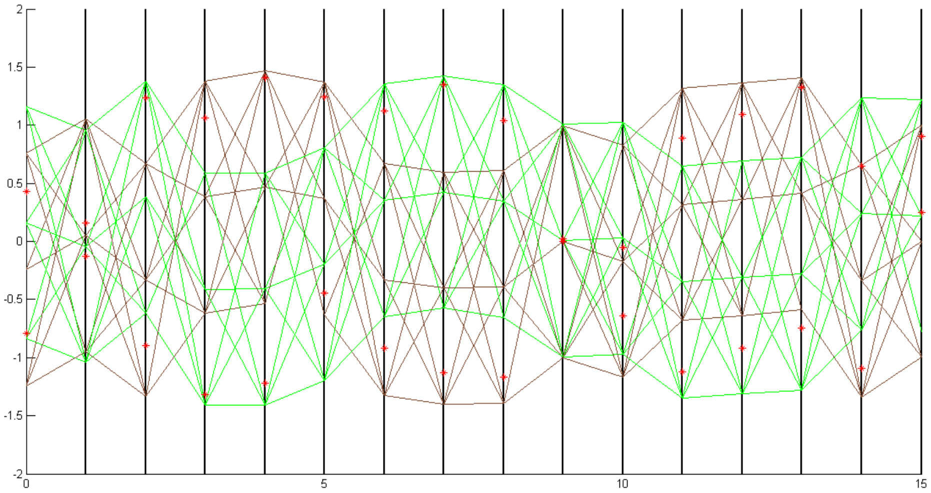
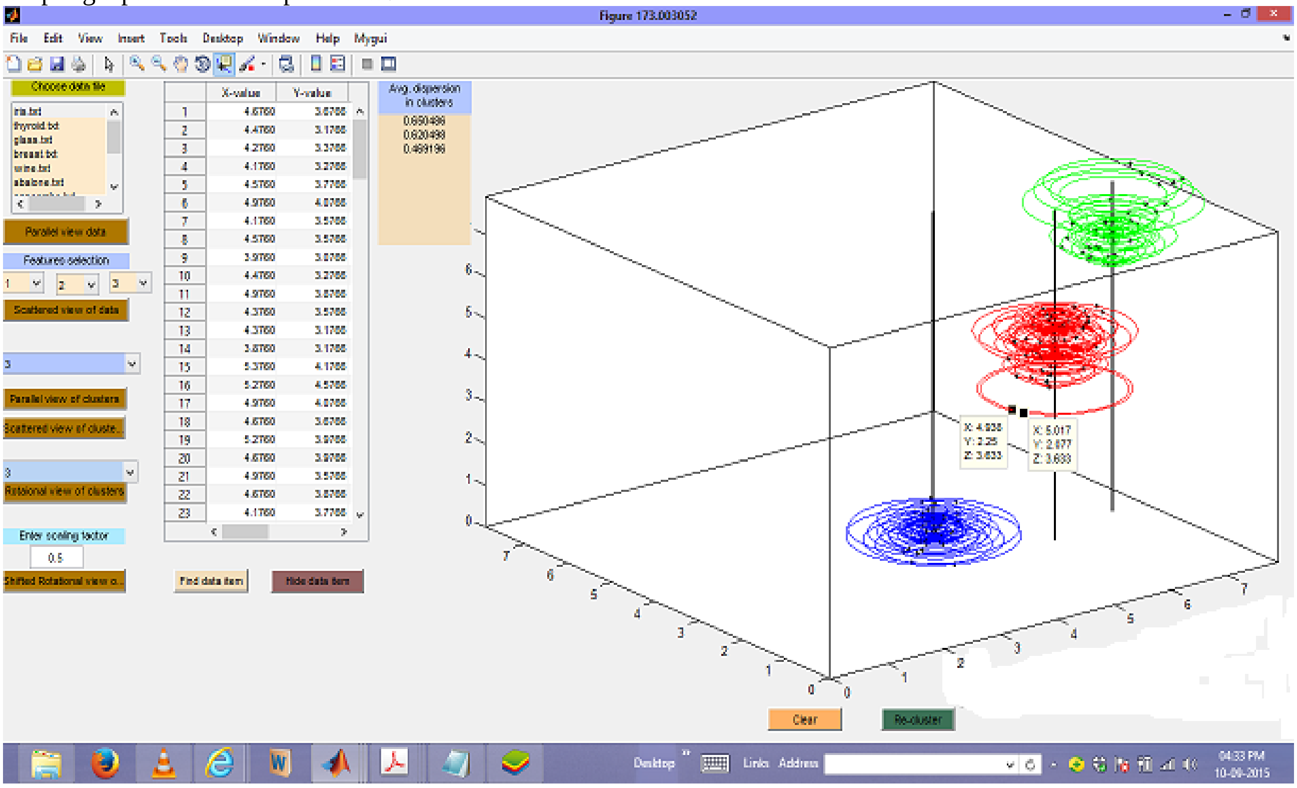


Fig. 13. Clustered *Vote* data plotted using all 16 features as different axes, for shift factor alpha = 5.0

The X-DAT plot is a very common approach to visualize clustered data. But from the results obtained in our experiments, we see that special characteristics of a particular data item cannot be distinguishingly determined from the X-DAT outputs. While outputs of the present method clearly identify the chance of a data item to be an outlier, constituents of the denser part of a cluster etc. However, X-DAT plot is advantageous in the sense that, here all the features can be plotted at a time in a two dimensional graph. We have plotted the X-DAT graph using data for shifted clusters for comparison purposes with our present method.

A snapshot of our visual system is shown in Fig. 14. Here, the *Iris* data (clustered) is plotted along with their circular boundaries when rotated about the Z-axis. We can select any three of the features as three axes. In the red cluster we have shown how possible outlier elements can be selected and deleted from the data set. After elimination we can re-display the clusters or can re-cluster the reduced data set. One option is presented here, to calculate the changes in the within cluster dispersions after such elimination operation. As shown in Fig. 14, we may claim that true 3D behaviour of a clustered data item is exactly realized, with regard to clustering, in this new form. Since knowledge representation in a suitable visual manner is highly desirable in Data Mining, our process may be treated as a valuable visual tool for cluster visualization. However, there is a limitation for displaying very large data sets, and this problem can be overcome by sampling a part of the complete data.



	X-value	Y-value
1	4.6760	3.6766
2	4.4760	3.1766
3	4.2760	3.3766
4	4.1760	3.2766
5	4.5760	3.7766
6	4.9760	4.0766
7	4.1760	3.5766
8	4.5760	3.5766
9	3.9760	3.0766
10	4.4760	3.2766
11	4.9760	3.8766
12	4.3760	3.5766
13	4.3760	3.1766
14	3.8760	3.1766
15	5.3760	4.1766
16	5.2760	4.5766
17	4.9760	4.0766
18	4.6760	3.6766
19	5.2760	3.9766
20	4.6760	3.9766
21	4.9760	3.5766
22	4.6760	3.8766
23	4.1760	3.7766

Fig. 14. Clustered *Iris* plotted, with elements shifted and rotated about the Z-axis in interactive visual platform, for alpha = 0.5

IV. CONCLUSIONS

In this work, an effective 3D data visualization tool is presented. It is clear that, conventional 2D visual products like bar charts, pie charts or other kind of charts like the X-DAT, are unable to properly display important features of data or clusters efficiently. A proper 3D view, such as the present one, may reveal many such interesting aspects from the data without ambiguity. Data Mining or Business Analytic processes may, therefore, be highly benefited using this approach. From the experimental results we see that, shifted and rotated view of clustered data is superior representation compared to 3D scatter plot of the same. As an extension of the process, we can now develop a proper user interaction platform which is essentially needed.

REFERENCES

[1] J. T. Tou and R. C. Gonzalez, Pattern Recognition Principles. Reading: Addison-Wesley, 1974.

[2] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[3] A. K. Jain, R. P. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," in *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, No. 1, 2000, pp. 04 - 37.

[4] J. B. McQueen, "Some methods of classification and analysis in multivariate observations," in *Proc. of fifth Barkley symposium on mathematical statistics and probability*, 1967, pp. 281 - 297.

[5] R. Jin, A. Goswami and G. Agarwal, "Fast and Exact Out-of-Core and Distributed K-Means Clustering," in *Knowledge Information Systems*, vol. 10, 2006, pp. 17 - 40.

[6] P. S. Bradley and U. M. Fayyad, "Refining Initial Points for K-means Clustering," in *Technical Report of Microsoft Research Center, Redmond*, USA, 1998.

[7] M. K. Pakhira, "A Modified *k*-means Algorithm to Avoid Empty Clusters," in *International Journal of Recent Trends in Engineering*, vol. 1, 2009, pp.220-226.

[8] M. K. Pakhira, "Efficient Parallel *k*-means Algorithm on a Cyclic Network," in *International Journal of Information Processing* , vol. 4, 2010, pp. 15-30.

[9] D. Arthur, and S. Vassilvitskii, "*k*-means++: the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2007, pp. 1027–1035.

[10] M. K. Pakhira, "A Fast *k*-means Algorithm using cluster shifting to generate compact and separate clusters", *International Journal of Engineering*, vol. 28(1), 2015, pp. 35-43 .

[11] M.K. Pakhira, "A linear time-complexity *k*-means Algorithm using cluster shifting", in Proc. of *International conference ICCICN*-2014, pp. 1047 - 1051 .

[12] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 3, 1936, pp. 179–188.

[13] Machine Learning Database at http://kdd.ics.uci.edu.

[14] KDnuggetts database at http://kdnuggetts.com