IMDC: An Image-Mapped Data Clustering Technique for Large Datasets

Faruq A. Al-Omari, and Nabeel I. Al-Fayoumi

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 publications.waset.org/7113.pdf

Abstract**—** In this paper, we present a new algorithm for clustering data in large datasets using image processing approaches. First the dataset is mapped into a binary image plane. The synthesized image is then processed utilizing efficient image processing techniques to cluster the data in the dataset. Henceforth, the algorithm avoids exhaustive search to identify clusters. The algorithm considers only a small set of the data that contains critical boundary information sufficient to identify contained clusters. Compared to available data clustering techniques, the proposed algorithm produces similar quality results and outperforms them in execution time and storage requirements.

Keywords**—** Data clustering, Data mining, Image-mapping, Pattern discovery, Predictive analysis.

I. Introduction

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information that can be used more productively to increase revenue, cuts costs, or both. Its concerned with finding correlations and patterns in different fields in large relational databases. Many applications including spatial analysis, credit card fraud detection, network intrusion detection, market basket analysis, financial portfolio analysis, medical diagnosis and many others rely heavily on data mining for classification, pattern detection, and predictive analysis. N-dimensional (N-D) data tuples in a large database are studied extensively to gain knowledge about relations and correlation between them. However, in many applications the N-D data set is reduced to 2-D data set depending on the attributes of interest; accordingly data clustering categorizes the 2-D data into clusters which assist in asserting some type of relationship between the data points within a cluster and hence drawing range of relationships between the two dimensions under consideration.

Several clustering algorithms have been developed in the past years. Conventionally, two major approaches where used, distance-based clustering, and density-based clustering. These techniques mostly rely on exhaustive iterative techniques to identify clusters. K-means-based algorithms are typical examples of distance-based clustering techniques [1,2,5,8,14]. However, K-means clustering algorithms are easily affected by noise and can easily miss ill-shaped clusters. While density-based techniques such as nearest-neighborhood can easily handle noise and can find arbitrarily-shaped clusters, they suffer from exhaustive search, which severely affects performance [4, 13, 15, 18, 22, 25]. New techniques have immerged recently, which hybridize both distance and density based algorithms to enhance accuracy and performance, BRIDGE [16] which merges BIRCH [22] with DBSCAN [23] is a good example. Intelligent algorithms such as genetic and fuzzy clustering algorithms have been used, however, performance of such algorithms severely affects their practicality and usability for real applications [4,6,13,17].

The developed technique uses a completely different approach. It applies image segmentation techniques instead of using classical distance and density-based approaches. The 2-D dataset of interest is mapped to a two-dimensional bitmap image. Then, object boundary detection techniques are used efficiently and effectively to detect the clusters. Very few data points that are highly scattered, independent, are finally clustered together as one noise cluster. The novelty of the developed algorithm stems from the fact that it uses a small percentage of the data points to identify clusters. This, in turn, saves large on execution time and required storage space. By looking only at boundary data points and overlooking the major point population that lay within, the algorithm can quickly and efficiently identify most of the data clusters. The algorithm eventually needs a single pass over the complete dataset to categorize and label the data points.

The rest of the paper is organized as follows: Section II introduces the methodology of the proposed algorithm. Experimental results and discussion are presented in Section III, and we conclude in Section IV.

II. Image-mapped data clustering

The developed algorithm is portrayed in Figure 1. As can be seen, the algorithm consists primarily of three major stages. The first is a data and image association stage, which intuitively transforms between the original dataset and the mapped image plane. The second is an image preprocessing stage to identify the boundary points of image objects in the image plane and to represent their shapes. The third stage is cluster localization and identification, which integrates knowledge gained in the second stage to localize clusters in

Manuscript received October 20, 2004.

Faruq. A. Al-Omari (email: fomari@yu.edu.jo, telephone: 962 2 7211111 ext. 4439) and Nabeel Al-Fayoumi are with the Computer Engineering Department, Hijjawi Faculty for Engineering Technology, Yarmouk University, Irbid, Jordan.

the dataset. The following subsections elaborate in more details on these major steps.

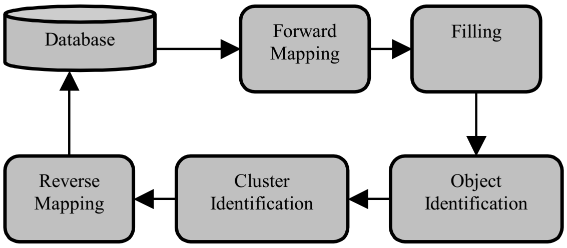


Figure 1. A block diagram of the developed algorithm.

A. Forward and Reverse Data Mapping

The data points in the dataset can be either numeric or textual taking values in a certain range, called vector space. A forward mapping function is developed to translate the data points from the 2-D vector space into a binary image. Primarily, a binary image plane is constructed with dimensions sufficient to accommodate all data points in the vector space. such that, the size of the constructed image plane is

$$\begin{aligned}\text{Image Size} &= \text{Rows x Columns} \\ &= (P_{Xmax} - P_{Xmin} + 2k_1) \text{ x } (P_{Ymax} - P_{Ymin} + 2k_2)\end{aligned} \quad (1)$$

where P_{Xmax}, P_{Xmin}, P_{Ymax}, and P_{Ymin} are the maximum and minimum values in the x- and y-directions. k_1 and k_2 are small constants added to introduce background around the data points in the image plane. Pixel values at image coordinates that correspond to data point coordinates are turned ON. The rest of the pixel values in the image plane are turned OFF. To fit the data points in the integer image grid, rounding is performed on the vector space coordinates of each data point. Henceforth, an image pixel at coordinates (x,y) that correspond to data point (P_{xi},P_{yi}) is turned ON, such that

$$\begin{aligned} &x = k_1 + round\left((P_{xi} - P_{X\min}) \cdot \frac{P_{X\min}}{P_{Y\min}} \right) \text{ and} \\ &y = k_2 + round\left((P_{yi} - P_{Y\min}) \cdot \frac{Columns}{Rows} \right) \end{aligned} \quad (2)$$

In this manner, the huge dataset is represented in a binary image which relatively requires lower memory resources as compared to the original dataset.

The forward mapping module eliminates the need to revisit the dataset in later stages of the process. However, to map each data point to its corresponding cluster after successful clustering, a reverse mapping module is developed to associate the data points to their corresponding cluster. This module is only needed as a final step in the clustering process. The rest of the modules in Figure 1 operate on the constructed binary image.

B. Object Filling

Clusters correspond to dense regions, called an object. An object appears as a bright spot in the image plane. However, this spot does not necessarily correspond to a uniformly filled object in the image plane, but rather several holes could be found in that object. Therefore, to be able to completely and sufficiently detect such an object, a filling step is performed prior to any processing step to purge holes in the scattered point region that correspond to the dense region. For this, a 3x3 majority mask is utilized to decide whether a particular point in the image plane belongs to the object or not. Empirical results indicate that a threshold value of 6 provides a satisfactory filling result that works well for filling holes without affecting the general shape of the cluster regions.

C. Object Identification and Boundary Detection

Objects in the image plane correspond to clusters in the dataset. These objects can be completely represented by their boundaries. In order to localize the boundary points that correspond to data clusters, a cleaning step is required. The cleaning process is mainly performed to first, smooth the boundaries of the objects to avoid any undesired notches that won't affect the identification process. Second, to eliminate small objects that correspond to few scattered points in the image plane. Erosion and dilation are among several techniques that are used to perform this job [24]. To localize the boundary points, an eroded image is subtracted from the original cleaned image. Chain encoding is utilized after that to derive a syntactic description of the boundary points from a neighborhood matrix.

In this manner each connected boundary is represented by a chain of consecutive boundary points. These points can represent a single cluster, called simple object, or more than one cluster, called complex object. The contour of a complex object experiences skirting. That means the contour shape dilutes down at certain point in shape forming a *neck*. An illustration of a simple and a complex object is depicted in Figure 2.

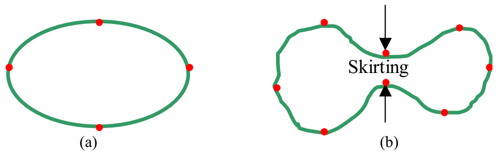


Figure 2. A typical contour line that corresponds to a (a) simple, (b) complex object.

A further reduction of the number of points processed to identify data clusters is accomplished through traversing critical boundary points. Critical boundary points are the set of boundary points that sufficiently represent the contour shape. Curvature Scale Space (CSS) principles [20, 21] suggest that a contour could be sufficiently and completely described by the set of boundary points that correspond to local maxima and minima in the 2-D spatial domain containing the object shape. Consequently, a boundary point *p* is considered critical if and only if the first derivative of the contour experiences a change of sign at that point. The rest of the boundary points are ignored and marked as unnecessary. The developed algorithm traverses the critical boundary of each object searching for possible necks. Necks are identified by the Euclidean distance between the two boundary points comprising them such that the distance is below certain threshold value, T_n. The threshold value is determined as a percentage of the maximum distance encountered between two critical boundary points in the contour. This value is tuned to achieve results that best fit the dataset.

Accordingly, a neck is encountered between two non-adjacent critical points, Pi and Pj if only if Pi is the closest to Pj and Pj is the closest to Pi. This process is repeated for all critical boundary points detected for all objects localized in the data-mapped image. As a result each cluster is identified by a closed contour in the image space.

As a result of erosion and other preprocessing steps, some points will be none clustered at this stage. For that, these points can be grouped together in a noise cluster.

D. Complexity Analysis

Due to article size limitation, detailed analysis of the complexity analysis are omitted. However, it has been found that the order of computations required to identify clusters and associate data points to these clusters is $O(5N+M^2)$, where N is the number of data points and M is the number of identified clusters.

On the other hand, the memory requirement to load the entire dataset, after mapping it into an image plane, was found to be $3N/8$ bytes which resembles 95% reduction in memory usage.

III. Results and Discussion

A synthetic dataset generated via a developed generator by Zhang et. al. [22] was used to test and validate the IMDC proposed technique. The dataset consists of 100,000 points distributed evenly in 100 clusters. The dataset was first mapped into a binary image plane. The resultant image size was set to 700x100 as shown in Figure 3.

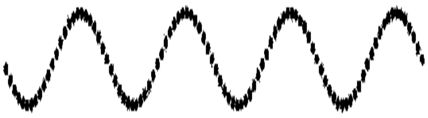
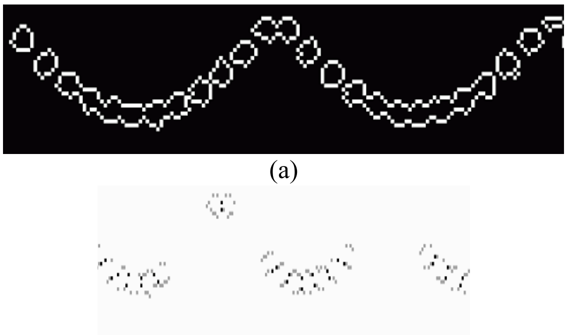


Figure 3. An data-mapped image representing the synthesized dataset generated by Zhang et. Al [22].

The data-mapped image was then processed according to the processes described earlier. Figure 4 shows the resultant image after finding the boundary points and critical points to localize necks.



(a)

(b)

Figure 4. (a) represents a zoomed portion of the data-mapped image after finding the boundary points, and (b) represents a zoomed portion of the resultant image after localizing critical boundary points and necks.

The result of clustering was finding exactly 100 clusters. The number of points located in each cluster is summarized in Figure 5. The number of boundary points located was 1470 points and the number of critical points was 417 points. Only the 417 critical points were processed in the segmentation and clustering module to localize the data clusters which is a great reduction in the complexity of the proposed technique.

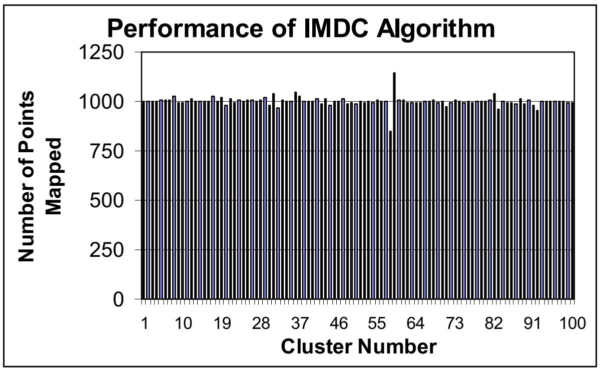


Figure 5. Distribution of data points.

IV. Conclusions

The paper presented a unique approach to cluster large datasets. The main contribution lies in utilizing existing image processing techniques to identify clusters in an image-mapped 2-D dataset. The proposed algorithm produced quality results that are at least as good as the available algorithms, furthermore, the presented algorithm outperforms the other techniques in performance and storage requirements. This is actually due to the fact that the algorithm identifies a comparatively small number of critical boundary data points, which are used to cluster the data. Complexity analysis show that the run time of the proposed algorithm is of O(5N+M2) whereas best performance of available clustering algorithms is of O(N log N). On the other hand, the proposed algorithm achieves a major reduction in storage requirement compared to other techniques. In fact, 95% saving in memory I/O operations was accomplished.

REFERENCES

[1] Biernacki, G. Celeux, and G. Govaert, “Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood,” IEEE Trans on pattern analysis and Machine Intelligence,22(7), pp. 719-725, 2000.

[2] R. Ostrovsky and Y. Rabani, “Polynomial time approximation schemes for geometric k-clustering,” Proceedings of the 41st Annual Symposium on Foundations of Computer Science, pp.349, Telcordia Technologies, Morristown, NJ, USA, 2000..

[3] S. Guha, R. Rastogi and K. Shim,“ROCK: A Robust Clustering Algorithm for Categorical Attributes,” Proceedings of the 15th International Conference on Data Engineering, pp.512, Sydney, Australia, March 1999.

[4] L.O. Hall and L.O.; B. Ozyurt,“Scaling genetically guided fuzzy clustering,” Proceedings of the 3rd International Symposium on Uncertainty Modeling and Analysis, pp.328, College Park, Maryland, March1995.

[5] D.E. Tamir, C.Y. Park; W.S. Yoo, “Vector quantization and clustering: a pyramid approach,” Proceedings of the Data Compression Conference(DCC’95), pp.482, Utah, USA, March 1995.

[6] N.K. Ratha, A.K Jain, and M.J. Chung Editor(s): Cantoni, V., Lombardi, L., Mosconi, M., Savini, M., Setti, A. “Clustering using a coarse-grained parallel genetic algorithm: a preliminary study,” International Conference on Computer Architectures for Machine Perception, pp.331, Como, Italy, Sept. 1995.

[7] Lee and V. Estivill-Castro, “Effective and Efficient Boundary-based Clustering for Three-dimensional Geoinformation Studies,” Proceedings of the Third International Symposium on Cooperative Database Systems for Advanced Applications (codas), pp.82, Beijing, China, April 2001.

[8] S. Guha, N. Mishra, R. Motwani, L. O'Callaghan, “Clustering data streams,” Proceedings of the 41st Annual Symposium on Foundations of Computer Science, pp.359, Redondo Beach, California, USA, 2000.

[9] Ching-Huang Yun and Kun-Ta Chuang and Ming-Syan Chen “An Efficient Clustering Algorithm for Market Basket Data Based on Small Large Ratios,” 25th Annual International Computer Software and Applications Conference (COMPSAC'01), pp.505, Chicago, Illinois, USA, October 2001.

[10] Bouguettaya, “On-Line Clustering,” IEEE Transactions on Knowledge and Data Engineering, pp. 333-339, April 1996.

[11] H. Nagesh and A. Choudhary “A Scalable Parallel Subspace Clustering Algorithm for Massive Data Sets,” Proceedings of the 2000 International Conference on Parallel Processing, pp.447, August 2000.

[12] Judd, P. McKinley, and A. Jain, “Large-Scale Parallel Data Clustering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.871-876, 1998.

[13] Petridis, V. and Kaburlasos, V.G. “Clustering and Classification in Structured Data Domains Using Fuzzy Lattice Neurocomputing (FLN),” IEEE Transactions on Knowledge and Data Engineering, pp. 245-260, March 2001.

[14] Mu-Chun Su, Chien-Hsing Chou, “Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry", Patterns Analysis and Machine Intelligence, 23(6): pp.674-680, June 2001.

[15] Cheng-Fa Tsai, Han-Chang Wu, Chun-Wei Tsai, “A New Data Clustering Approach for Data Mining in Large Databases,” In Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks, 2002. I-SPAN '02. , pp.315, Makati City, Metro Manila, Philippines, May, 2002.

[16] M. Dash H. Liu X. Xiaowei, “Merging distance and density based clustering,” Proceedings of the Seventh International Conference on Database Systems for Advanced Applications, pp. 332-39, Hong Kong, China, 2001.

[17] Sarafls, A.M.S. Zalzala, and P.W. Trinder “A genetic rule-based data clustering toolkit,” Proceedings of the 2002 Congress on Evolutionary Computation, CEC '02., Volume: 2 , pp.1238 -1243, 2002.

[18] C. Ordonez, E. Omiecinski, and N. Ezquerra, “A fast algorithm to cluster high dimensional basket data,” Proceedings of the IEEE International Conference on Data Mining (ICDM 2001), 633 – 636, San Jose, CA, USA, Nov. 2001.

[19] S. E. Umbaugh, “Computer Vision and Image Processing A Practical Approach Using CVIPtools”, Prentice Hall, 1998.

[20] Yoke Khim Ung and Mokhtarian, F., “Multi-scale spline-based contour data compression and reconstruction through curvature scale space”, In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 4, pp. 2123 – 2126, 2000.

[21] Pinheiro, A.M.G.; Izquierdo, E.; Ghanhari, M., “Shape matching using a curvature based polygonal approximation in scale-space”, In Proceedings of the International Conference on Image Processing, 2000, Vol. 2, pp. 538 –541, 2000.

[22] T. Zhang, R. Ramakrishanan, M. Livny, “BIRCH: an efficient clustering method for very large databases,” In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 103-114, 1996.

[23] V. Ganti, R. Ramakrishanan, and J. Gehrke, “Clustering large datasets in arbitrary metric spaces”, In Proceedings of the 15th Int. Conference On Data Engineering, pp. 502-511, March 1999.

[24] R. J. Schalkoff, “Digital Image Processing and Computer Vision”, John Wiley and Sons Inc.,1989.

[25] M. Goebel, and L. Gruenwald, “A survey of data mining and knowledge discovery software tools”, ACMKDD, Explorations, 1(1): pp. 20-33, 1999.

[26] Mokhtarian, F. and A. K. Mackworth, “Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes," IEEE Trans. PAMI, vol. 8, no. 1, pp. 34-43, 1986.