ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

A Review of Data Clustering Approaches

Vaishali Aggarwal, Anil Kumar Ahlawat, B.N Panday

Abstract- Fast retrieval of the relevant information from the databases has always been a significant issue. Different techniques have been developed for this purpose; one of them is data clustering. Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). The clustering problem has always been a focus of many researchers in many fields and disciplines and has a broad attraction and usefulness as one of the steps in exploratory data analysis. Many problems in business, science, industry, and medicine can be treated as clustering problems. Some of the examples include bankruptcy prediction, credit scoring, medical diagnosis, quality control, handwritten character recognition, image processing and speech recognition. This paper presents and discusses some of the advancements in competitive and self organization learning algorithms for data clustering and presents their suitable applications in different fields.

Keywords- **Clustering, Competitive Learning, Dead Units, K-Means Clustering, Self Organization.**

I. INTRODUCTION

Clustering is a main task of explorative data mining, and a common approach for analysis of statistical data used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics etc. Cluster analysis is not an algorithm but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of constituting a cluster and how to efficiently find them. Popular notions of clusters include groups with low distances among the cluster members, grouping of unlabeled data, dense areas of the data space and multivariate normal distributions. Main focus of clustering analysis is to determine the cluster number, explore the properties of each cluster and find a structure in a collection of unlabeled data. During the last twenty years several algorithms have been developed. One of the algorithms developed is k-means algorithm [1] which starts with K random cluster center and divides a collection of objects into K subsets. But it has a problem of "dead units" i.e. if a centre is inappropriately chosen, it may never be updated, thus it may never represent a class. Another algorithm developed was frequency sensitive competitive learning [11] which tries to remove the problem of "dead units" but it also has a same problem as k-means i.e. selecting the number k in advance means it also needs to know the exact number of clusters. Thus the Rival penalized competitive learning RPCL [8] was introduced which was based on idea that, for each input, not only the winner among the seed points is updated to adapt to the input, but also its nearest rival (i.e., the second winner) is de-learned by a smaller learning rate (also called de-learning rate). But being sensitive to pre-assigned de-learning rate, it fails to perform better and correct clustering. After RPCL some of its variants were developed like DPRCL [14] and RPCCL [15]. After competitive learning algorithms a concept of self organization was introduced. Based on this concept Self Organization Maps [17] were developed to perform clustering. In SOM, the similar data in the input space under a measurement are placed physically close to each other on the map. But the problem with SOM was the initialization of learning rate in order to achieve the convergence. At last the concept of rival penalization self organization maps RPSOM [21] was developed which utilizes the constant learning rate and hence achieves better convergence in clustering.

II. APPROACHES DESCRIPTION

This section describes the various clustering approaches that have been discussed in this paper. These approaches are shown in figure 1.

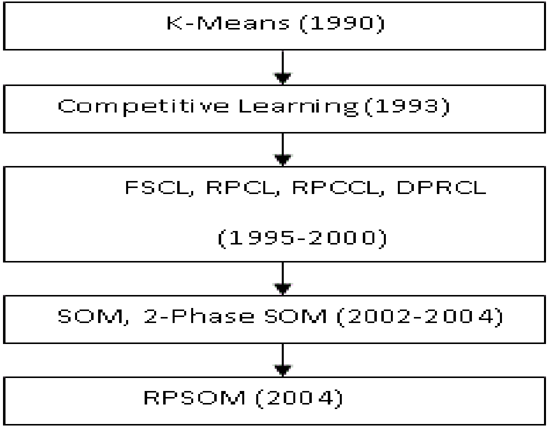

figure 1. Evolution of data clustering approaches

A. K-MEANS CLUSTERING ALGORITHM AND ITS VARIANTS

The term "*k*-means" was first used by James MacQueen in 1967 **[1]** and was published in 1991. *K*-means algorithm is one of the most popular clustering algorithms used in variety of domains. It is a typical competitive learning algorithm which partitions the input data set into *k* categories (called clusters) each finally represented by its centre, that change adaptively, starting from some initial values called seed points **[2].** The basic idea behind K-means algorithm is to choose *K* patterns as initial centers firstly (k is the user set parameter and is the number of final pattern cluster). This algorithm assigns each point to its closest center to form *K* clusters, then re-computes the center of each cluster, repeats the assignment and compute until no clusters change, or, until the center remain the same **[3].** The K-means algorithm has higher efficiency and scalability and converges fast when dealing with large data sets; therefore it is widely used in cluster analysis. Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum i.e. the algorithm ends in local minimum **[4]**. Another problem of the *k*-means algorithm is that it needs to know the exact number of clusters *k*, before performing data clustering. Otherwise, it will lead to a poor clustering performance. Unfortunately, it is often hard to determine *k* in advance in many practical problems. The *k*-means algorithm has also the "dead units" problem, which means that if a centre is inappropriately chosen, it may never be updated, thus it may never represent a class.

Variants of the k-means algorithm are Lloyd's k-means clustering algorithm and Progressive Greedy k-means clustering algorithm. Being relatively faster and fairly straight forward algorithm, the Lloyd's algorithm often converges to a local minimum of the squared error distortion rather than the global minimum [5]. One of the variant of the *k*-means algorithm, the *k*-means incremental algorithm performs clustering without knowing the clusters. It gradually increases the number of clusters under the control of a threshold parameter, which is however difficult to be decided. Some successful applications of K-means algorithm are: color quantization, data compression, and image segmentation.

B. FREQUENCY SENSITIVE LEARNING ALGORITHM

The frequency sensitive competitive learning (FSCL) introduced in 1990 [6] is an extension of the k-means algorithm to remove the problem of "dead units". It does so by introducing the parameter called the relative winning frequency or "conscience" into the similarity measurement between an input and the seed points (or centers). The chance of the center to win the competition is directly proportional to the conscience. The FSCL algorithm reduces the chance of frequent winner winning the competition by reducing their learning rate i.e. the larger the winning frequency, the larger is the chance of being penalized. So, in training process all the units have the opportunity to be updated. Although the FSCL algorithm works well and can almost successfully assign one or more seed points to a cluster without the "dead units" problem, but it suffers from the same problem as of k-means. It also needs to know the exact number of clusters i.e. its performance deteriorates rapidly if k is not well specified. Some successful applications of the FSCL algorithm are feature extraction [6] and image compression [7].

C. RIVAL PENALIZED COMPETITIVE LEARNING

The adaptive version of FSCL called rival penalized competitive learning algorithm (RPCL) was proposed by Xu et al. in 1993 [8]. RPCL can be regarded as an unsupervised extension of Kohonen's supervised LVQ2. RPCL performs appropriate clustering without knowing the clusters number and it also solves the "dead units" problem. The basic idea behind this algorithm is that, for each input, not only the winner of the seed points is updated to adapt to the input, but also its nearest rival (i.e., the second winner) is delearned by a smaller learning rate(also called de-learning rate). It performs the rival penalization for each input without considering the distance of the rival from the winning unit [8]. In fact, the rival should be more penalized if its distance to the winner is closer than the one between the winner and the input. The idea of RPCL is similar to the social scenario in our daily life. For example, the competition between two candidates called X and Y (we assume that X is the final winner and Y is therefore the rival) in an election campaign will become more intense if their public opinion polls are closer. Otherwise, X will be almost sure to win the election with little attack against (i.e., little penalizing) Y during the election campaign [9].

The algorithm is quite simple and provides a better convergence than the *k*-means and the FSCL algorithms and introduced some speedup to the learning process. But being sensitive to pre-assigned de-learning rate, it fails to perform better and correct clustering. One of the variant of RPCL algorithm is Stochastic RPCL (S-RPCL) algorithm, which penalizes the rivals by using the same rule as the RPCL, but the penalization is performed stochastically. Other variants of RPCL are Rival Penalization Controlled Competitive Learning (RPCCL) [9] and Distance sensitive RPCL (DSRPCL) [10]. Some applications of the RPCL algorithm are nonlinear channel equalization [11], color image segmentation [12], images features extraction [13].

D. THE DYNAMICALLY PENALIZED RIVAL COMPETITIVE LEARNING ALGORITHM

The DPRCL [14] is a variant of the RPCL algorithm [8]. It performs appropriate clustering without knowing the clusters number, by automatically driving the extra seed points far away from the input data set. It dynamically controls the selection of de-learning rate and introduces a new term called penalization strength. If the distance between the winning centre and its first rival is smaller than the distance between the winning centre and the input then the penalization strength will be maximum, of value 1; otherwise the penalization will be gradually attenuated up to zero, as the distance, between the winner and its first rival increases. The DRPCL algorithm is actually a generalization of the RPCL algorithm, which moves away the undesired centers much faster than the RPCL algorithm i.e. having the fast and better convergence, because its de-learning rate is greater. DPRCL finds its application in adaptive clustering of fast varying signals corrupted by noise.

E. RIVAL PENALIZED CONTROLLED COMPETITIVE LEARNING

Like DPRCL, rival penalized controlled competitive learning (RPCCL) [15] is also a variant of RPCL [8] introduced by Yiu-ming Cheung. In order to control the rival penalization, RPCCL fully penalizes the rival if its distance to the winner is closer than the distance between the winner and the input otherwise the penalization strength will be gradually attenuated when the distance between the winner and the rival increases. In RPCCL for each input, not only the winner seed point is modified to adapt to the input, but also its rival (the 2nd winner) is de-learned by a smaller learning rate. The de-learning rate needs to be re-selected appropriately not only for different clustering problems, but also for different initial positions of the seed points. The problem with RPCCL is the instability of clustering results i.e. if the initialized clustering centers are close to the actual ones, it can cluster accurately and fast and if not then it will perform more slowly and even lead to local minima. Another problem is of the updating process. The rivals' and winners' displacements are not only in relation to the distances between input point and the two seed points, but also in relation to the input point location. That is if the input point is close to a clustering center, it should attract winners more tightly to make it converge as soon as possible. Also it should repel the rival more greatly so that the rival leaves the clustering center more quickly. If the input point is located at a cluster margin, the repelling force is smaller to keep the winner moving away from the clustering center [15]. To overcome the problems in original RPCCL, a new algorithm was proposed in [16]. While original RPCCL works well when the number of initialized seed point is same as the actual number, but it needs more computing costs, this new algorithm takes advantage of influence of sample distribution into account, and perform correct clustering faster and more accurately, and the clustering results are more reliable.

F. SELF ORGANIZING MAPS

Self-Organizing maps were developed by Prof. Teuvo Kohonen in the early 1980's. SOM is used to categorize and interpret large, high- dimensional data sets. Self-organizing map (SOM) [17] and its variant [18][19], are one of the popular data visualization techniques that provide a topological mapping from the input space to the output space. Typically, an SOM map possesses a regular one or two-dimensional (2-D) grid of nodes. Each node (also called neurons) in the grid is associated with a parametric real vector called model or weight that has the same dimension as the input vectors. The task of SOM is to learn those models so that the similar high-dimensional input data are mapped into one-dimensional (1-D) or 2-D output space with the topology as unchanged as possible. That is, the similar data in the input space under a measurement are placed physically close to each other on the map. However the SOM algorithm spends lots of time to learn because of many factors such as large map size, large quantity of input data, and many dimensions in data, etc. The topology preservation feature i.e. the input vectors which are placed near in input domain are placed near in map of SOM. This makes it a good cluster analyzing tool for high dimensional data. When SOM is applied to clustering problems, the classification results sometimes depend on the initial learning rate and initial weight vectors, even the sequence of the input samples when the input training samples are not more enough. SOM needs to initialize a learning rate whose value decreases over time to ensure the convergence of the map. Usually, a small initial value of learning rate is prone to make the models stabilized at some locations of input space in an early training stage. As a result, the map is not well established.

If we reduce the learning rate very slowly, the map can learn the topology of inputs well with the small quantization error, but the map convergence needs a large number of iterations and becomes quite time-consuming. On the other hand, if we reduce the learning rate too quickly, the map will be likely trapped into a local suboptimal solution and finally led to the large quantization error. To circumvent the problem of selection of appropriate learning rate and it's decreasing monotonically decreasing function, a variant of SOM is introduced i.e. 2-phase SOM [20] in which training is done in 2 phases uses 2 learning rates. In the first phase, it keeps a large learning to obtain the rough topological structure of the training data quickly. In the second phase, a much smaller learning rate is utilized to the trained map from the first phase, which achieves the fine-tuning topological map to ensure the map convergence. But the performance of the training algorithm is still sensitive to the time-varied learning rate [21].Some applications of SOM are visualization [22], image analysis [23], data mining [24], and so forth.

G. RIVAL PENALIZED SELF ORGANIZING MAPS

RPSOM [21] is inspired by the idea of RPCL [8] and RPCCL [15]. To elude the problems of selecting learning rate and decreasing function in SOM, for each input, the RPSOM adaptively chooses several rivals of the best-matching unit (BMU) and penalizes their associated models a little far away from the input. Instead of specifying the decreasing function of learning rate, RPSOM utilizes the constant learning rate to elude the selection of monotonically decreased function for the learning rate and achieves good convergence of map. Several experiments have shown that RPSOM has better convergence, neuron utilization and has less quantization error [25].

III. CONCLUSION

This paper discusses the several algorithms based on the competitive learning and self organization learning for clustering problems. On the basis of study of several algorithms it was found that the quality of results obtained by clustering method depends on the similarity measures, by its ability to discover some or all of hidden patterns, on the definition and representation of clusters chosen i.e. predefined number of clusters and properties of each cluster, learning and de-learning rate, neighborhood considered, topology preservation etc. The main focus of all the algorithms is to determine how well the input feature is matched with the already existing features in several clusters and then how that input is placed within respective cluster. Among all the algorithms presented it has been observed that if parameters like learning rate, neighborhood function, distance measures are suitably chosen and implemented then the performance achieved in clustering the dataset has better results in case of RPSOM algorithm. Some points of focus:

1. K-means focuses on choosing K initial clusters and hence does not give better performance due to the existence of 'dead units' and hence leads to local minima problem..
2. FSCL tries to remove problem of dead units by selecting the parameter called the 'conscience' but again suffers from the same problem as k-means i.e. selection of appropriate centers.
3. RPCL exploits the concept of rival penalization and hence speeds up the learning process therefore providing better convergence than k-means and FSCL. But the problem is of selecting appropriate learning rate.
4. DPRCL exploits the penalization strength without knowing exact number of clusters and hence achieves better convergence.
5. RPCCL enhances the convergence and provides better clustering results by setting the de-learning rate for rival smaller than winner's learning rate. But the problem faced is instability of cluster results.
6. To be a good cluster analyzing tool for high dimensional data SOM focuses on appropriate initial learning rate and initial weight vectors. Being sensitive to initial values of learning rate and weight vectors SOM may lead to local suboptimum and quantization error.
7. RPSOM utilizes the constant learning rate and hence achieves better convergence and lesser quantization error and also achieves greater neuron utilization.

REFERENCES

[1] Mac Queen JE. Some methods for classification and analysis of multivariate observation, Proceedings of the Fifth Berkley Symposium Math. Stat Prob, 1967, pp, 281-297.

[2] Hecht-Nielsen, Neurocomputing. New York: Addison-Wesley Publishing Company, 1990.

[3] M. Steinbach, G. Karypis, V. Kumar, "A Comparision of Document Clustering Techniques," Proc.the 6th International Conference on Knowledge Discovery and Data Mining, Boston, 2000.

[4] J Dong, M. Qi. K-means Optimization Algorithm for Solving Clustering Problem. Knowledge Discovery and Data Mining. 2009:52-55.

[5] N. C. Jones and P. A. Pevsner, An Introduction to Bioinformatics Algorithms, the MIT Press, 2004.

[6] H. C. Card and S. Kamasu, "Competitive learning and vector quantization in digital vlsi," Neurocomputing, vol. 18, pp. 195–227, Jan 1998.

[7] C. H. Chang, P. Xu, R.Xiao, and T. Srikanthan, "New adaptive quantization method based on self-organization," IEEE Trans. on Neural Networks, vol. 16, pp. 237–249, Jan. 2005.

[8] L. Xu, A. Krzyzak, and A. E. Oja, "Rival penalized competitive learning for clustering analysis RBF net and curve detection," IEEE Trans. on Neural Networks, no. 4, pp. 636–64, 1993.

[9] Yiu-ming Cheung : Rival Penalization Controlled Competitive Learning For Data Clustering with Unknown Cluster Number[C], IEEE Trans. Proceedings of the 9th International conference on Neural Information Processing ,Singapore,2002-11-18~22 467-471.

[10] W. Ma, T.J Wang. A cost-function approach to rival penalized competitive learning (RPCL) IEEE Trans. Systems, Man, and Cybernetics, 2006, 36(4):722-737.

[11] S.C. Ahalt, A.K. Krishnamurthy, P. Chen, and D.E. Melton, "Competitive Learning Algorithms for Vector Quantization," Neural Networks, vol. 3, pp. 277-291, 1990.

[12] B. Fritzke, "Growing Cell Structures—A Self-Organizing Network for Unsupervised and Supervised Learning," Neural Networks, vol. 7, no. 9, pp. 1441-1460, 1994.

[13] P. Guo, C.L. Philip Chen, and M.R. Lye, "Cluster Number Selection for a Small Set of Samples Using the Bayesian Ying-Yang Model," IEEE Trans. Neural Networks, vol. 13, no. 3, pp. 757-763, 2002.

[14] Corina Botoca, Georgeta Budura, "Complex data clustering using a new competitive learning algorithm", ELEC. ENERG. vol. 19, no. 2, pp. 261-269, 2006.

[15] Yiu-ming Cheung: On Rival Penalization Controlled Competitive Learning for clustering with automatic Cluster Number selection" [J], IEEE Trans. Knowledge and Data Engineering, VOL.17, NO.11, 1583- 1588, NOV.2005.

[16] Sanfeng Chen,Tao Mei, Minzhou Luo,, Huawei Liang, "Study on a New RPCCL Clustering algorithm", Proceedings of the 2007 IEEE International Conference on mechanical electronics and Automation, pp. 5 - 8, 2007.

[17] Self-Organizing Maps, 3rd Ed. New York: Springer-Verlag, 2001.

[18] S. Kaski, J. Kangas, and T. Kohonen, "Bibliography of self-organizing map (SOM) papers: 1981–1997," Neural Comput. Surv., vol. 1, pp 102–350, 1998.

[19] T.Kohonen, "Things you haven't heard about the self-organizing map," in Proc. IEEE Int. Conf. neural Netw. 1993, vol. 3, pp. 1147–1156.

[20] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, "Self-organizing map in Matlab: The SOM toolbox," in Proc. Matlab DSP Conf., Espoo, Finland, 1999, pp. 35–40.

[21] L. T. Law and Y. M. Cheung, "Rival penalized self-organizing map," in Proc. IASTED Int. n Conf. Neural Netw. Comput. Intell. (NCI'2004),n Grindelwald, Switzerland, Feb. 23–25, 2004, pp. 142–145.

[22] J. Vesanto, “SOM-based data visualization methods,” Intell. Data Anal., vol. 3, pp. 111– 126, 1999.

[23] J. Laaksonen, M. Koskela, and E. Oja, “Application of tree structured self-organizing maps in content-based image retrieval,” in Proc. Int. Conf. Artif. Neural Netw. (ICANN’99), Edinburgh, U.K., 1999, pp. 174–179.

[24] “Using SOM in data mining,” Licentiate’s thesis, Helsinki Univ. Technology, Espoo, Finland, 2000.

[25] Yiu-ming Cheung and Lap-tak Law, “Rival-Model penalized Self-Organizing Map” IEEE Transactions on Neural Networks, vol. 18, no. 1, PP. 289-295, 2007.

AUTHOR BIOGRAPHY

Was born in Ghaziabad. Did her schooling from DAV Public School, Sahibabad. Did her B.Tech in Information Technology from R.K.G.I.T, Ghaziabad, institute affiliated to Uttar Pradesh Technical University, Lucknow. Student of M.Tech in Computer Science and Engineering AKGEC, Ghaziabad. Have 1 year of teaching experience in RKGITW, Ghaziabad. Her area of interest includes Computer Organization, Artificial Neural Networks, and Unsupervised Learning.

Was born in Muzaffar Nagar District of Uttar Pradesh, India, in 1975. He got his education from CCS University Meerut, IIT Roorkee and M.Tech in Computer Science and Engineering from Kurukshetra University. He has completed his Ph.D degree from University School of Engineering and Technology, Guru Gobind Singh Indraprastha University, New Delhi, India. He has published more than 30 research papers in International/National Journals/Conferences. He is a Head of Department in Masters of Computers Application department in KIET Ghaziabad. His present research interests include Artificial Neural Networks, Artificial Intelligence, Algorithm Design, Device Modeling and simulation of HEMT.

Did his B.Tech from Dr. KNMIET, Modinagar affiliated to UP Technical University her in International conference. He has 3 years of teaching experience. He is an Associate Professor in the Department of Computer Science and Engineering of AKGEC Ghaziabad affiliated to UP Technical University, Lucknow. His present research interests include pattern recognition, software engineering algorithms.