IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 1, Ver. I (Jan.-Feb. 2017), PP 19-24
www.iosrjournals.org

Recent Trends in Incremental Clustering: A Review

Neha Chopade[1], Jitendra Sheetlani[2]

[1]*(Department of Computer Application, SIES College of Management Studies/ University of Mumbai, India)*
[2]*(Department of Computer Science of Engineering/Sri Satya Sai Universityof Technology and Medical Sciences India)*

Abstract: *This paper presents a review on recent trends in incremental clustering algorithms. It tries to focus on both clustering based on similarity measure and clustering not based on similarity measure. In this context, the paper is devoted to various typical incremental clustering algorithms. Mainly optimization, genetic and fuzzy approaches of these algorithms is covered in the paper. The paper is original with respect to one aspect that is, it provides a complete overview that is fully devoted to evolutionary algorithms for incremental clustering. A number of references are provided that describe applications of evolutionary algorithms for incremental clustering in different domains, such as human activity detection, online fault detection, information security, track an object consistently throughout the network solving boundary problem etc.*

Keywords: *fuzzy logic, incremental clustering, natural genetics, optimization, similarity measure.*

I. Introduction

With the rapid development of World Wide Web, it becomes more and more difficult for users to find online information they need. Therefore, there is a huge potential to collect valuable information and structure behind Web documents to improve the intelligence and efficiency of internet. Over the past few years, Web mining techniques have shown rapid progress and been widely used. One of these techniques is document clustering, which tries to identify inherent groupings of text documents so that a set of clusters is produced in which clusters exhibit high intra-cluster similarity and low inter-cluster similarity. It is particularly useful in many applications such as grouping search results, clustering Web documents, and others.There is a large body of work that investigates clustering methods. Based on accumulation rules of data objects in clustering and methods employing these rules, clustering methods could be divided into four types: hierarchical clustering, partitional clustering, density & grid-based clustering and others. Hierarchical and partitional clustering methods are used in many applications, and their representative algorithms include agglomerative hierarchical clustering algorithm and *K*-Means algorithm, respectively. Currently, to meet the demands of online applications where documents are generated in the course of using, such as blog and wiki, incremental clustering becomes a research focus. Incremental clustering algorithms work by processing data objects one at a time, incrementally assigning data objects to their respective clusters while they progress. Typical examples include Single-Pass Clustering, *K*Nearest Neighbor clustering (*K*NN). This paper will analyze recent representative incremental clustering algorithms from the aspect of algorithm thinking, key techniques, advantages and disadvantages, and compare their clustering quality by experiments on some datasets. The rest of this paper is organized as follows: Section 2 briefly introduces the existing typical incremental clustering algorithms. Section 3 expatiates on the experiments carried out and discusses the experimental results with conclusion.

II. Incremental Clustering Algorithms

In current Web applications, there is much User-Generated Content (UGC) which covers a range of media content available in modern communications technologies, such as question-answer databases, digital video, blogging, podcasting, forums, review-sites, social networking, mobile phone photography and wikis . User-Generated Content has also been characterized as 'Conversational Media', which is a key characteristic of so-called Web 2.0 which encourages the publishing of one's own content and commenting on other people's. In above situations, the dataset is dynamic, so it is impossible to collect all data objects before starting clustering. When new data comes, non-incremental clustering will have to re-cluster all the data, which certainly decreases efficiency and wastes computing resource. On the contrary, incremental clustering just need to group new data and update new clusters to previous clustering results. The strategy optimizes clustering process and especially adapts to those applications where time is a critical factor for usability. However, incremental clustering faces several challenges.The number of pioneer documents is small, so it is difficult to obtain high clustering quality in the early stage. When more and more documents come, we will find that some previous documents were put into wrong groups, so it might be necessary to reassign some documents to new clusters. In another word, documents with different order could result in different clustering results. It can be seen from above analysis that there is still much work to do, though incremental clustering seems simple. Generally, a typical documents clustering process mainly includes two components, namely a similarity measure and a clustering algorithm. There are many methods for measuring document similarity, such as the *Cosine* measure, the *Jaccard* measure, the *Dice* measure and the *Overlap* measure. One clustering method could choose different similarity measures, and clustering results generated by the same method under various similarity measures could even have significant difference. Therefore, it is very crucial to select an appropriate similarity measure.However, there are some clustering methods that rely on the original document vectors not the pair-wise document similarity , such as Suffix Tree Clustering (STC) and DC-Tree Clustering. In this section, we will briefly discuss following five incremental clustering methods, covering above two categories.

2.1 Clustering based on Similarity

Reviews existing similarity measures, including the measures in the vector space model, the information theoretic measure, the measures derived from popular retrieval functions and the OM-based measures, and also proposes a novel measure based on the earth mover's distance to evaluate document similarity by allowing many-to-many matching between subtopics. When designing a clustering algorithm, similarity measure choice is sensitive to the specific representation, which determines whether the algorithm can accurately reflect the structure of various components in high dimensional data. However, there are many methods to measure pair-wise document similarity and no selection criteria so that too often it is an arbitrary choice to choose similarity measure although it is very important to a clustering algorithm.

1.1.1 Single-Pass Clustering

It is a very simple clustering method. The method makes the first object as the centroid for the first cluster, and then calculates the similarity between the next object and each existing cluster centroid using some similarity coefficient. If the highest value of the similarity is greater than the threshold value appointed beforehand, the new object is added to the corresponding cluster and the centroid is updated; otherwise, the object is put into a new cluster. The method is very efficient, but suffers from that the resulting clusters are not independent of the object insertion order.

1.1.2 K-Nearest Neighbor Clustering

*K*NN is an approach to classifying objects based on closest training data in pattern recognition, and it can also be used as a clustering method. According to this algorithm, an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *K* nearest neighbors .

2.2 Clustering NOT based on Similarity

Some clustering algorithms are designed without obvious similarity measure. For example, in Suffix Tree Clustering, documents are combined based on base clusters that indirectly measure the distance between documents; DC-Tree Clustering is based on the B+-tree structure, and inserting a new document just involves comparison of the document feature vector with the cluster vectors. Though there are no obvious similarity measures in these algorithms, they actually include steps for measuring similarity indirectly. In another word, the similarity measures of this type of clustering methods are different mainly in manifestations from methods based on similarity. We briefly review the following three algorithms.

2.2.1Suffix Tree Clustering

Its kernel is a suffix tree which is a data structure keeping track of all *n*-grams of any length in a set of word strings, while allowing strings to be inserted incrementally in time linear to the number of words in each string . STC is composed of the following three steps: document cleaning, identifying base clusters, and combining base clusters. The algorithm is theoretically fast, but it is just useful when there is enough memory available to create the suffix tree. When creating a suffix tree from a large sequence database, the memory requirement is difficult to be ensured, and the effectiveness of STC is decreased.

2.2.1Incremental DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a data clustering algorithm proposed by proposed a new document clustering approach based on graph model and enhanced incremental DBSCAN. In their approach, a graph-based model is used for document representation instead of traditional vector-based model. It is an effective incremental clustering algorithm suitable for mining in dynamically changing databases, and we can easily update graph structure when a new document is added to database by using graph model.

2.2.2 ICIB

ICIB (Incremental Clustering based on Information Bottleneck theory) is presented in 2011. The method is designed to improve the accuracy and efficiency of document clustering, and resolve the issue that an arbitrary choice of document similarity measure could produce an inaccurate clustering result. ICIB measures document similarity with information bottleneck theory. A first document is selected randomly and classified as one cluster, and then each remaining document is processed incrementally according to the mutual information loss introduced by the merger of the document and each existing cluster. If the minimum value of mutual information loss is below a certain threshold, the document will be added to its closest cluster; otherwise it will be classified as a new cluster. The incremental clustering process is low-precision and order-dependent, which cannot guarantee accurate clustering results. Therefore, an improved sequential clustering algorithm (SIB) is proposed to adjust the intermediate clustering results [1].

III. Various Techniques of Clustering

1.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) was originally designed and introduced by Eberhart and Kennedy. The PSO is a population based search algorithm based on the simulation of the social behavior of birds, bees or a school of fishes. PSO originally intends to graphically simulate the graceful and unpredictable choreography of a bird folk. Each individual within the swarm is represented by a vector in multidimensional search space. This vector has also one assigned vector which determines the next movement of the particle and is called the velocity vector.The PSO also determines how to update the velocity of a particle. Each particle updates its velocity based on current velocity and the best position it has explored so far; and also based on the global best position explored by swarm. The PSO process then is iterated a fixed number of times or until a minimum error based on desired performance index is achieved. It has been shown that this simple model can deal with difficult optimization problems efficiently. The PSO was originally developed for real valued spaces but many problems are, however, defined for discrete valued spaces where the domain of the variables is finite. Classical examples of such problems are: integer programming, scheduling and routing. In 1997, Kennedy and Eberhart introduced a discrete binary version of PSO for discrete optimization problems. In binary PSO, each particle represents its position in binary values which are 0 or 1. Each particle's value can then be changed (or better say mutate) from one to zero or vice versa. In binary PSO the velocity of a particle defined as the probability that a particle might change its state to one. Upon introduction of PSO algorithm, it was used in number of engineering applications. Using binary PSO, proposed a high quality splitting criterion for codebooks of tree-structured vector quantizers (TSVQ). Using binary PSO, they reduced the computation time too. Binary PSO is used to train the structure of a Bayesian network. Although PSO is successfully used in number of engineering applications, but this algorithm still has some shortcomings. In novel binary PSO, the velocity of a particle is its probability to change its state from its previous state to its complement value, rather than the probability of change to 1. In this new definition the velocity of particle and also its parameters has the same role as in real valued version of the PSO. There are also other versions of binary PSO. authors add birth and mortality to the ordinary PSO. Also fuzzy system can be used to improve the capability of the binary PSO [2].

1.2 Ant colony optimization

An Ant Colony Optimization algorithm (ACO) is essentially a system based on agents which simulate the natural behavior of ants, including mechanisms of cooperation and adaptation. The use of this kind of system as a new metaheuristic was proposed in order to solve combinatorial optimization problems. This new metaheuristic has been shown to be both robust and versatile – in the sense that it has been successfully applied to a range of different combinatorial optimization problems. ACO algorithms are based on the following ideas: Each path followed by an ant is associated with a candidate solution for a given problem. When an ant follows a path, the amount of pheromone deposited on that path is proportional to the quality of the corresponding candidate solution for the target problem. When an ant has to choose between two or more paths, the path(s) with a larger amount of pheromone have a greater probability of being chosen by the ant. As a result, the ants eventually converge to a short path, hopefully the optimum or a near-optimum solution for the target problem, as explained before for the case of natural ants. In essence, the design of an ACO algorithm involves the specification of [2]: An appropriate representation of the problem, which allows the ants to incrementally construct/modify solutions through the use of a probabilistic transition rule, based on the amount of pheromone in the trail and on a local, problem-dependent heuristic. A method to enforce the construction of valid solutions, that is, solutions those are legal in the real-world situation corresponding to the problem definition.A problem-dependent heuristic function (h) that measures the quality of items that can be added to the current partial solution. A rule for pheromone updating, which specifies how to modify the pheromone trail (t).A probabilistic transition rule based on the value of the heuristic function (h) and on the contents of the pheromone trail (t) that is used to iteratively construct a solution. Artificial ants have several characteristics similar to real ants, namely: Artificial ants have a probabilistic preference for paths with a larger amount of pheromone. 5 Shorter paths tend to have larger rates of growth in their amount of pheromone. The ants use an indirect communication system based on the amount of pheromone deposited on each path [3].

1.3 Genetic algorithm

Genetic algorithms (GAs) are search and optimization procedures that are motivated by the principles of natural selection and natural genetics. Some fundamental ideas of genetics are borrowed and used artificially to construct search algorithms that are robust and required minimum problem information. In GAs, the role of selection and recombination operators is very well defined. Selection operator controls the direction of search and recombination operator generates new regions for search. Genetic algorithms are having a large amount of implicit parallelism. GAs perform search in complex, large and multimodal landscapes, and provide near-optimal solutions for objective or fitness function of an optimization problem. In GAs, the parameters of the search space are encoded in the form of strings (called chromosomes) and collection of such strings called a population. Initially, a random population is created, which represents different points in the search space. An objective and fittness function is associated with each string that represents the degree of goodness of the string. Based on the principle of survival of the fittest, a few of the strings are selected and each is assigned a number of copies that go into the mating pool. Biologically inspired operators like crossover and mutation are applied on these strings to yield a new generation of strings. The process of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is satisfied.

1.3.1Applications of GA based clustering

In previous section, various GA based clustering algorithms are studied. This section provides discussion on few of applications of GA based clustering algorithms. Paper shows application of a Genetic Algorithm to production simulation. The simulation is treated as a detailed, stochastic, multi-modal function that describes a performance statistic. Authors tried to optimize (or at least improve) the performance of the system. A model of a real-world production line for printed circuit boards that has many products and must often be retooled or reconfigured was used. Since the product line is always changing, with half of the products turning over within a year, the job of configuring and fine-tuning the production line is never ending. This paper has shown that a Genetic Algorithm when attached to the simulation model can provide excellent support for this process. This combination can be used to obtain quick and stable results that do indeed indicate the direction to improved production. In microarray data analysis, clustering is a method that groups thousands of genes by their similarities of expression levels, helping to analyze gene expression profiles. This method has been used for identifying unknown functions of genes. The fuzzy clustering method assigns one sample to multiple groups according to their degrees of membership. This method was more appropriate for analyzing gene expression profiles, because a single gene might be involved in multiple functions. An evolutionary fuzzy clustering method with knowledge-based evaluation was proposed.The segmentation problems were formulated upon such images as an optimization problem and adopt evolutionary strategy of Genetic Algorithms for the clustering of small regions in colors feature space. The present approach uses k-Means unsupervised clustering methods into Genetic Algorithms, namely for guiding this last Evolutionary Algorithm in his search for finding the optimal or sub-optimal data partition, task that as we know, requires a non-trivial search because of its intrinsic NP-complete nature. To solve this task, the appropriate genetic coding was also discussed. The image compression problem using genetic clustering algorithms based on the pixels of the image was proposed. GA was used to obtain an ordered representation of the image and then the clustering was performed to obtain the compression. In order to solve the problem of error estimation of real life data set the optimization technique of genetic algorithm (GA) was applied to the new adaptive cluster validity index, which is called the Gene Index (GI). The algorithm applies GA to adjust the weighting factors of adaptive cluster validity index to train an optimal cluster validity index. Paper presents a genetic algorithm that deals with document clustering. This algorithm calculates an approximation of the optimum k value, and solves the best grouping of the documents into these k clusters. This algorithm was evaluated with sets of documents that were the output of a query in a search engine. The modified variable string length genetic algorithm (MVGA) was proposed in for text clustering. This algorithm has been exploited for automatically evolving the optimal number of clusters as well as providing proper data set clustering. The chromosome was encoded by a string of real numbers with special indices to indicate the location of each gene. More effective versions of operators for selection, crossover, and mutation were introduced in MVGA which can also automatically adjust the influence between the diversity of the population and selective pressure during generations. GA was applied to enhance the performance of clustering algorithms in mobile ad hoc networks. Authors have optimized recently proposed weighted clustering algorithm (WCA). The proposed technique was such that each clusterhead handles the maximum possible number of mobile nodes in its cluster in order to facilitate the optimal operation of the medium access control (MAC) protocol. Consequently, it results in the minimum number of clusters and hence cluster heads. Clustering the Gene Ontology is a difficult combinatorial problem and can be modeled as a graph partitioning problem. Additionally, deciding the number k of clusters to use is not easily perceived and is a hard algorithmic problem. Therefore, an approach for solving the automatic clustering of the Gene Ontology was proposed by incorporating cohesion-and-coupling metric into a hybrid algorithm consisting of a genetic algorithm and a split-and-merge algorithm. To solve task clustering problem on an unbounded number of processors, a new genetic algorithm approach which has nice properties called Genetic Convex Clustering Algorithm (GCCA) was proposed. The main idea was to assign tasks to locations in convex groups. Here arbitrary execution time was considered and a novel crossover operator in the context of the task clustering problem was developed. Genetic algorithm with species and sexual selection (GAS3) was proposed to solve unimodal and multi-modal function of various difficulties. GAS3 uses sex determination method to determine the sex (male or female) of members in population. Each female member was considered as a niche in the population and the species (cluster) formation takes place around these niches. Species formation was based on Euclidean distance between female and male members. Each species contains one female member and zero or more male members. Parameter-less cluster formation technique was used. Merging of clusters takes place based on performance evaluation criterion [4].

1.4 Fuzzy Logic

Lets consider an example, a bank wants to classify their customers into two grups as rich and poor customers. Obviously there is no crisp separation between rich and poor in a way that customers own less than 2 million SEK are poor while they own a fortune of 2million SEK or more are rich. So according to the above explanation a person with a fortune of 2.1 million SEK is considered as reasonably rich but still a little bit poor. The indicator for similarity in fuzzy sets is called membership degree µ ={0,.....,1}. A membership degree µ =1 indicates that an object completely belongs to set and a membership degree µ =0 shows a total dissimilarity between an object and a set. In our example, the customer with 2.1million SEK may have membership degrees of μ_{rich} 1.2(millionSEK) = 65.0 to the set rich and 1.2(millionSEK) = 35.0 μ_{poor} to the set poor. This indicates that the customer is rich but not extremely wealthy. However a customer having 30billion SEK would surely have memberships of 1(billionSEK) = 0.1 μ_{rich} and 0.0 μ_{poor} 1(billionSEK) = o.o. There is probabilistic uncertainity neither about the fortune of the customer who is having 2.1million SEK nor about the rules for how to classify him into one or other of the two sets [5].

IV. Related Work and Conclusion

Xizhe Yin et al. [2016] This paper presents our recent work on human activity detection based on smart phone sensors and incremental clustering algorithms. The proposed unsupervised (clustering) activity detection scheme works in an incremental manner, which contains two stages. In the first stage, streamed sensor data will be processed. A single-pass clustering algorithm is used in order to generate pre-clustered results for the next stage. In the second stage,pre-clustered results will be refined to form the final clusters, which means the clusters are built incrementally adding one cluster at a time [6].

Ming Hou et al. [2016] In this paper, we present a computationally efficient online tensor regression algorithm, namely incremental higher-order partial least squares (IHOPLS), for adapting HOPLS to the setting of infinite time-dependent tensor streams. By incrementally clustering the projected latent variables in latent space and summarizing the previous data, IHOPLS is able to recursively update the projection matrices and core tensors over time, resulting in greatly reduced costs in terms of both memory and running time while maintaining high prediction accuracy. To show the effectiveness and scalability of our approach for large databases, we apply IHOPLS to two real-life applications as reconstruction of 3D motion trajectories from video and ECoG streaming signals [7].

Theodora Kontogianni et al. [2016] In this paper, we address the problem of object discovery in time-varying, large-scale image collections. A core part of our approach is a novel Limited Horizon Minimum Spanning Tree (LH-MST) structure that closely approximates the Minimum Spanning Tree at a small fraction of the latter’s computational cost. Our proposed tree structure can be created in a local neighborhood of the matching graph during image retrieval and can be efficiently updated whenever the image database is extended. We show how the LH-MST can be used within both single-link hierarchical agglomerative clustering and the Iconoid Shift framework for object discovery in image collections, resulting in significant efficiency gains and making both approaches capable of incremental clustering with online updates [8].

Jueun Kwak et al. [2015] In this paper, we propose an online fault detection algorithm based on incremental clustering. The algorithm accurately finds wafer faults even in severe class distribution skews and efficiently processes massive sensor data in terms of reductions in the required storage. We tested our algorithm on illustrative examples and an industrial example. The algorithm performed well with the illustrative examples that included imbalanced class distributions of Gaussian and non Gaussian types and process drifts. In the industrial example, which simulated real data from a plasma etcher, we verified that the performance of the algorithm was better than that of the standard support vector machine (SVM), one-class SVM and three instance-based fault detection algorithms that are typically used in the literature. Most standard classification algorithms, such as support vector machines, can handle moderate sizes of training data and assume balanced class distributions. When the class sizes are highly imbalanced, the standard algorithms tend to strongly favor the majority class and provide a notably low detection of the minority class as a result [9].

Mahmuda Akter et al. [2015] In this paper, an Incremental Clustering Algorithm is proposed in conjunction with Static Clustering Technique to track an object consistently throughout the network solving boundary problem. The proposed research follows a Gaussian Adaptive Resonance Theory (GART) based Incremental Clustering that creates and updates clusters incrementally to incorporate incessant motion pattern without defiling the previously learned clusters. The objective of this research is to continue tracking at the boundary region in an energy-efficient way as well as to ensure robust and consistent object tracking throughout the network. The network lifetime performance metric has shown significant improvements for Incremental Static Clustering at the boundary regions than that of existing clustering techniques. Emerging significance of moving object tracking has been actively pursued in the Wireless Sensor Network (WSN) community for the past decade. As a consequence, a number of methods from different angle of assessment have been developed while relatively satisfying performance. Amongst those, clustering based object tracking has shown significant results, which in term provides the network to be scalable and energy efficient for large-scale WSNs. As of now, static cluster based object tracking is the most common approach for large-scale WSN. However, as static clusters are restricted to share information globally, tracking can be lost at the boundary region of static clusters [10].Teena Ajayan et al. [2015] in this paper, the key idea is to find multiple representative sequences like medoids to represent a cluster in a chunk and final DNA analysis is carried out based on those identified medoids from all the chunks. The main advantage of this incremental clustering is that it uses multiple medoids to represent each cluster in each chunk which captures the pattern structure more accurately. Not only that it overcomes the disadvantages of existing techniques but also has the mechanism to make use of DNA sequence relationship among those identified medoids that serves as a side information to help the final DNA sequence clustering. The proposed incremental approach outperforms existing clustering approaches in terms of clustering accuracy [11].

Jafar Boostanpour et al. [2015] In this paper, to overcome the aforementioned obstacles, a two-step algorithm is proposed which in first step, a fast spherical interpolation (SI) method is utilized to prepare an appropriate initial guess for the MLE algorithm. In the second step, a clustering-based network is described to attain less power consumption across the WSN. Furthermore, to increase the probability of convergence, a cooperative incremental clusterbased estimation strategy is proposed. In addition, major issues that can affect the performance of the proposed method are investigated. Simulation results prove the capability of this method and support the claims [12]. Puma Chandra Sethi et al. [2014] In this paper, EHMA is implemented in two tiers, but this paper considers the implementation of EHMA in three tiers. It follows incremental clustering algorithm for grouping clusters according to their impact factors. This three tier implementation of EHMA improves the security of the information as it uses SHA-256 for security. - In the present scenario, most of the computing operations are performed over the Internet. Many companies provide their services using Internet, so networking services becomes more important these days. Hence, there is high demand for secured management of information, along with faster processing of operations. Due to increased demand for network services, there is a need to increase the performance of these services [13].

References

[1]. Yongli Liu, Qianqian Guo, Lishen Yang, Yingying Li "Research on Incremental Clustering" 978-1-4577-1415-3/12/$26.00 ©2012 IEEE.

[2]. Amreen Khan, Prof. Dr. N.G.Bawane, Prof. Sonali Bodkhe "An Analysis of Particle Swarm Optimization with Data Clustering-Technique for Optimization in Data Mining." International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1363-1366.

[3]. Rafael S. Parpinelli, Heitor S. Lopes, and Alex A. Freitas "Data Mining with an Ant Colony Optimization Algorithm"2010.

[4]. Rahila H. Sheikh, M. M.Raghuwanshi, Anil N. Jaiswal "Genetic Algorithm Based Clustering: A Survey" First International Conference on Emerging Trends in Engineering and Technology, 2008 IEEE.

[5]. Md. Ehsanul Karim Feng Yun Sri Phani Venkata Siva Krishna Madani "Fuzzy Clustering Analysis" April 2010.

[6]. Xizhe Yin, Weiming Shen, Xianbin Wang "Incremental Clustering for Human Activity Detection Based on Phone Sensor Data" 2016 IEEE.

[7]. Ming Hou, Brahim Chaib-draa "ONLINE INCREMENTAL HIGHER-ORDER PARTIAL LEAST SQUARES REGRESSION FOR FAST RECONSTRUCTION OF MOTION TRAJECTORIES FROM TENSOR STREAMS"2016 IEEE.

[8]. Theodora Kontogianni, Markus Mathias, Bastian Leibe "Incremental Object Discovery in Time-Varying Image Collections" 2016 IEEE.

[9]. Jueun Kwak, Taehyung Lee and Chang Ouk Kim "An Incremental Clustering-Based Fault Detection Algorithm for Class-Imbalanced Process Data", IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING, VOL. 28, NO. 3 , 2015 IEEE.

[10]. Mahmuda Akter, Md. Obaidur Rahman, Md. Nazrul Islam and Md. Ahsan Habib "Incremental Clustering-based Object Tracking in Wireless Sensor Networks" 2015 IEEE.

[11]. Teena Ajayan, Sony P, Janu R Panicker, Shailesh S "Clustering DNA Sequences Of Aspergillus fumigatus Using Incremental Multiple Medoids" 2015 IEEE Fifth International Conference on Advances in Computing and Communications.

[12]. Jafar Boostanpour, Sayed Mostafa Taheri, Hossein Zamiri-Jafarian "Cluster-Based Two-Step Target Localization with Incremental Cooperation" 2015 IEEE.

[13]. Puma Chandra Sethi, Prafulla Kumar Behera "Secured Packet Inspection with Hierarchical Pattern Matching implemented using Incremental Clustering Algorithm" 978-1-4799-5958-7/14/$31.00 ©20 14 IEEE.