International Journal of Computer Theory and Engineering, Vol. 2, No. 3, June, 2010
1793-8201

Incremental Clustering in Data Mining using Genetic Algorithm

Atul Kamble

***Abstract*—Data warehouses provide a great deal of opportunities for performing data mining tasks such as classification and clustering. Typically, updates are collected and applied to the data warehouse periodically. Then, all patterns derived from the warehouse by some data mining algorithm have to be updated as well. Due to the very large size of the databases, it is highly desirable to perform these updates incrementally. In this paper, we present the new approach/algorithm based on Genetic algorithm. Our algorithm is applicable to any database containing data from a metric space, e.g., to a spatial database. Based on the formal definition of clusters, it can be proven that the incremental algorithm yields the same result as any other algorithm. A performance evaluation of algorithm Incremental Clustering using Genetic Algorithm (ICGA) on a spatial database is presented, demonstrating the efficiency of the proposed algorithm. ICGA yields significant speed-up factors over other clustering algorithms.**

***Index Terms*—Data Mining, Clustering, Genetic Algorithm.**

I. Introduction

Many organizations have recognized the importance of the knowledge hidden in their large databases and, therefore, have built data warehouses. When speaking of a data warehousing environment, we work on two characteristics namely, analysis and multiple updates. These characteristics or requirements tend to new approach called Data Mining. Data mining has been defined as the application of data analysis and discovery algorithms that - under acceptable computational efficiency limitations - produce a particular enumeration of patterns over the data. Several data mining tasks have been identified, e.g., clustering, classification and summarization. Our area of concentration is clustering. In data warehouse, data is not updated immediately when insertions and deletions on the operational databases occur. Updates are collected and applied to the data warehouse periodically in a batch mode, e.g., each night. Due to the very large size of the databases, it is unfeasible to cluster entire data for every updates. Hence, it is highly desirable to perform these updates incrementally.

In this paper we are concentrating on new way of clustering using biological inspired Genetic algorithm. This algorithm clusters data in dynamic form. The database is assumed to be clustered initially, and every new element is added as without need of changing existing clustered database. Our algorithm (ICGA) is an efficient clustering algorithm for metric databases (that is, databases with a distance function for pairs of objects) for mining in a data warehousing environment. ICGA is density-based in nature. Current approach works only for insertion operation. We demonstrate the high efficiency of incremental clustering on a spatial database.

Manuscript received September 19, 2009.

Atul Kamble is with the D.K.T.E.S. Textile and Engineering Institute, Ichalkaranji-416115, India. (Mobile phone: +91-9673274518; e-mail: atulbkamble@yahoo.com).

II. Initiative to Proposed Work

The proposed work initiates with some requirements.

A. Some prerequisites

Proposed work uses GA for clustering but does requires a special file which holds information regarding clusters. We call this file as meta file. Meta file record format looks like.

VALID_BIT
CLUSTER_ID
NO_OF_RECORDS
TRUE_ELEMENT_ATTRIBUTES_VALUE

Where, VALID_BIT indicates whether this cluster is valid (for example all elements deleted, then value equals zero '0'), CLUSTER_ID is a numerical value indicating cluster identity, NO_OF_RECORDS indicates the total records which are currently part of that cluster, TRUE_ELEMENT_ATTRIBUTES_VALUE indicates values of all attributes which forms first element in cluster, hence name TRUE.

Initially, we initialize this meta_file with one record called INIT_RECORD. This record has all its fields as zero. The INIT_RECORD is shown in figure 1.

B. Genetic Algorithm

Genetic algorithm [1] is a biologically inspired search algorithm. The GA uses and manipulates a population of potential solutions to find the optimal solutions. A generation is completed after each individual in the population has performed the genetic operators. The individuals in the population will be better adapted to the objective/fitness function, as they have to survive in the subsequent generations. At each step, the GA selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generation, the population evolves toward an optimal solution. This advantage of GA is used to find the suitable cluster for new data to be inserted in database.

Figure 1. INIT_RECORD in *meta file*

Database records are considered or referred as objects.

The fitness function in GA is represented in the following way.

$$f(new_{obj}) = \sqrt{\sum_{i=0}^{n}(meta_{obj}(i) - new_{obj})^2} \quad (1)$$

Where, n is number of attributes in database, metaobj is object from meta-file, newobj is object to be added.

The key idea of clustering is that for each element of a cluster is in neighborhood of a given radius (gR), i.e. the cardinality of the neighborhood has to exceed some threshold. Then after $f(new_{obj})$ is compared with threshold tR. If $f(new_{obj})$ is less than threshold tR, $f(new_{obj})$ is returned back else MAX_THRESHOLD is returned. MAX_THRESHOLD is any largest decimal value.

Algorithmic Settings for GA - The GA operators selection, crossover, mutation and Pc (Crossover Probability) and Pm (Mutation Probability) are listed in Table 1. Initial Population of GA is meta_file. Individuals (meta objects properties) goes through fitness calculation (as per equation 1). After fitness calculation the GA uses operators Crossover, Reproduction and mutation with parameters setting shown in Table 1. The GA undergoes 50 generations.

The after GAs execution, fittest object's CLUSTER_ID will be Identity (ID) for new object.

TABLE 1. PARAMETER SETTING FOR GA

Setting Type	Value
Encoding Scheme	Binary Encoding
Population size	200
Evolution generation	50
Selection	Roulette Wheel
Crossover	One point
Mutation	Uniform
Pc	0.6
Pm	0.01
Elitism	Yes
Generations	50

III. THE ALGORITHM ICGA

Some terms:
Object is referred as a database record with attributes.
meta_file is meta file having information about clusters.
gR is radius or threshold.
new_object is object to be inserted.

Algorithm ICGA (*meta_file, gR, new_object*)

1. Initialize GA and all its parameters.

2. Get population from *meta_file*.

3. Calculate fitness of each individual

4. Apply GA operators to population

5. Generate new set of population.

6. If No. of Generations > MAX_GEN
   a. Goto Step 3

7. Else
   a. Get fitness value of fittest individual after MAX_GEN generations

   b. If FINAL_FITNESS == MAX_THRESHOLD
      i. Get new CLUSTER_ID (One greater than highest present ID)
      ii. Create new entry in *meta_file* with attributes of *new_object.*

   c. Else
      i. Get CLUSTER_ID of the fittest individual
      ii. Set that CLUSTER_ID to new object.

8. Return.

This is how ICGA works.

IV. PERFORMANCE ANALYSIS

For performance analysis we have used raw cotton fibre characteristics from a Textile mill. These characteristics have numerical values which are tested from instrument called HVI 9000 (High Volume Instrument - Model 9000). These characteristics are shown in Table 2.

TABLE 2. COTTON FIBRE CHARACTERISTICS

SL	UR	GTEX	MIC	SFI	CG
28.88	48.30	20.68	3.47	8.90	21.3
29.67	48.12	21.52	3.42	8.18	32.1
29.03	48.70	20.58	3.48	8.48	11.4
28.95	47.10	21.60	3.55	9.62	32.1
28.62	46.28	21.40	3.41	10.56	11.4

Similar kinds of ten thousand records were considered for clustering. gR was set to 1.

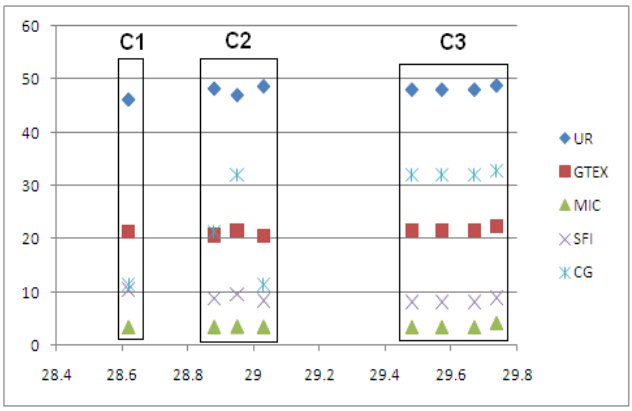

Figure 2. Results for clustering.

Typically, the number comparisons are used as a cost measure for database algorithms because the I/O time heavily dominates CPU time. Performance for any other clustering algorithm is at the least log(n2) i.e. it requires n2 comparisons.

Were as in proposed algorithm, the comparison of one element is done with elements in meta_file (mE) for nG number of Genetic Algorithm generations.

Hence, performance can be given as

$$Perf(ICGA) = mE * nG \quad (2)$$

As GA can be stopped after some n number of generations when it finds stable results, this number of comparisons can still be reduced. Hence, performance in equation 2 can be considered as Worst Case Performance.

An example demonstrating clustering results is shown in Figure 2.

V. Conclusions

Data warehouses provide a great deal of opportunities for performing data mining tasks such as classification and clustering. Typically, updates are collected and applied to the data warehouse periodically. In this paper, we presented the new approach incremental clustering using genetic algorithm - ICGA - for mining in a data warehousing environment. ICGA requires distance function and, therefore, it is applicable to any database containing data from a metric space. In the future, deletions will be considered to further improve the efficiency of ICGA.

Acknowledgment

My thanks to Prof. L. S. Admuthe, Prof. D. V. Kodawade, Prof. S. K. Shirgave and Prof. K. S. Kadam for their valuable suggestions.

References

[1] David. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Publication Addison-Wesley Professional
[2] Ester M., Kriegel H.-P., Sander J., Xu X.: “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, Proc. 2nd Int. Conf. on Data Mining.
[3] DBRS: A Density-Based Spatial Clustering Method with Random Sampling Xin Wang and Howard J. Hamilton Technical Report CS-2003-13 November, 2003
[4] Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, 1996, pp. 226-231.
[5] Jiawei H., Kamber M.: “Data Mining: concepts and techniques”, Academic Press, San Diego, 2001.
[6] Ramze Rezaee, B.P.F. Lelieveldt, J.H.C Reiber. "A new cluster validity index for the fuzzy c-mean", Pattern Recognition Letters, 19, pp237-246, 1998.
[7] C. Sheikholeslami, S. Chatterjee, A. Zhang. "WaveCluster: A-MultiResolution Clustering Approach for Very Large Spatial Database". Proceedings of 24th VLDB Conference, New York, USA, 1998.
[8] M. Halkidi, M. Vazirgiannis, Y. Batistakis. "Quality scheme assessment in the clustering process", In Proceedings of PKDD, Lyon, France, 2000.
[9] M. Halkidi, M. Vazirgiannis, “Clustering Validity Assessment: Finding the optimal partitioning of a data set”, In the Proceedings of ICDM Conference,California, USA, November 2001.
[10] X. Wang and H. J. Hamilton, “DBRS: A Density-Based Spatial Clustering Method with Random Sampling”, Proc. of the 7th PAKDD, Seoul, Korea, 2003, pp. 563 – 575.
[11] S. Shekhar and S. Chawla, Spatial Databases: A Tour, Prentice Hall, 2003.