View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by MAT Journals

MAT JOURNALS
Journal of Data Mining and Management
Volume 2 Issue 1

A Fast Detection of Duplicates Using Progressive Methods

B. Bhagya Lakshmi
Department of Computer Science and Engineering, Gayatri Vidya Parishad College of Engineering (A), India
E-mail: bhagi.lucky7@gmail.com

Abstract

In any database large amount of data will be present and as different people use this data, there is a chance of occurring quality of data problems, representing similar objects in different forms called as 'duplicates' and identifying these duplicates is one of the major problems. In now-a-days, different methods of duplicate - detection need to process huge datasets in shorter amounts of time and at same time maintaining the quality of a dataset which is becoming difficult. In existing system, methods of duplicate - detection like Sorted Neighborhood Method (SNM) and Blocking Methods are used for increasing the efficiency of finding duplicate records. In this paper, two new Progressive duplicate - detection algorithms are used for increasing the efficiency of finding the duplicate records and to eliminate the identified duplicate records if there is a limited time for duplicate - detection process. These algorithms increase the overall process gain by delivering complete results faster. In this paper am comparing the two progressive algorithms and results are displayed.

***Keywords*:** *Attribute concurrency, data cleaning, duplicate detection, efficiency*

INTRODUCTION

In any database the datasets can be easily used by different users, so there is a chance of occurring errors like duplicate data and unsystematic data which makes the duplicate-detection and data cleansing compulsory. Duplicate - detection is the process of identifying different representations of same objects in a database [1]. Data cleansing is performed after duplicate - detection process to maintain clean and correct data in any database clearly [1]. So to perform data cleansing fast within the time limit on the dataset, two new progressive duplicate - detection algorithms are implemented here.

The main perspective of this paper is to enhance the duplicate - detection process, if there is less amount of time for delivering complete and fast results to the users.

In existing, two approaches called blocking and windowing are used for duplicate - detection process. Blocking method divides the records into different groups, and windowing method moves a window on the sorted data and after that comparing of records takes place only within the particular window by using static order. To avoid this problem in this project Progressive Sorted Neighborhood Method (PSNM) and Progressive Blocking (PB) uses the concurrent and parallel approaches for identifying the duplicate pair of records by using dynamic order.

The main disadvantage in previous algorithms is, until completion of total running process, the complete and accurate duplicates cannot be identified and cannot be eliminated if there is less time for duplicate - detection process. That is "Cost-Benefit" ratio value will be more.

Here, in this paper, two new duplicate - detection algorithms are: Progressive sorted neighborhood method (PSNM) works well

Page 1-5 © MAT Journals 2017. All Rights Reserved

on small and clean datasets. Progressive Blocking (PB) works well on large and unclean datasets.

Here, the efficiency of these algorithms is calculated by using Cost-Benefit Ratio: where algorithms runtime is taken as 'cost' and 'benefit' is the Number of duplicates recognized after running these new algorithms. And also by using parallel and concurrent approaches.

ADVANTAGES

1. These algorithms give more complete results in less span of time. For example: If SNM and Blocking takes 4671 milliseconds and 3006 milliseconds to process 18000 records containing duplicates, then PSNM and PB takes less than 4671 milliseconds and less than 3006 milliseconds to process same number of records.
2. These algorithms give fast results to the users.
3. When there is a time limit for duplicate identification, then start executing these algorithms and terminate it when required. These algorithms give almost all non-duplicate records as results.
4. Whenever user does not have complete idea of input taken for detection processing but still want to perform it, then by using these algorithms the output can be delivered correctly as these choose keys, block and window-sizes automatically.
5. Here progressive, incremental and concurrent process accessing takes place for identifying similar record pairs faster if there is a less time for execution time.

RELATED WORK

If A Similarity join called "*Top - k set Similarity join*" is proposed by Xiao et.al for identifying and eliminating the duplicate records [2]. Here the records are considered as sets and by using similarity functions [2] the duplicate records are identified. In this for every record an index is given and based on these indexes process is done. Here the process is one record is taken and based on it the same record is present or not is checked by the user and then that duplicate record is present will be eliminated [2]. Next second record is taken and so on process continues until all records complete duplicate - detection process. In this "top-k join" algorithm is implemented for identifying the top-k pair of records for duplicate – detection process [2]. First it returns the top-k pair of records [2] which are ranked based on their matching from input dataset and they are removed based on threshold of the user .So that, for next process it is easy to identify more duplicate records by considering less similar records. Here by using pruning and optimization techniques, similar records are identified [2]. It is progressive but disadvantage of this is it takes more number of comparisons and more time as it takes top – k records and compares with remaining records. If there is a time limit for executing it does not gives complete results to the user as it takes more time to complete processing entire records.

Next "*Pay-as-you-go Entity Resolution"* is used for duplicate - detection in a database if there a limit (i. e. for work , runtime) [4] .For example: (in real time system) there is a huge number of records related to persons in the web , if data cleansing is to be performed on that data within the time, then user(related to web) perform maximum possible duplicate - detection process to identify duplicates. So the concept called "hints" is used where it tries to increase the process of entity resolution if there is a time limit [3]. A "hint "can be represented in different forms [3] .Example: Grouping of records based on their matching. An ER uses 'hint' as a guideline for knowing which records to be compared first in order to identify duplicate records in database [4]. Here three different types of hints: a sorted

Page 1-5 © MAT Journals 2017. All Rights Reserved

list of record pairs, a hierarchy of record partitions and an ordered list of records for identifying the duplicate records in a dataset [4]. But here the disadvantage of ER is all the hints used for duplicate - detection processes presents static order and miss dynamic order for the comparisons at run time [4]. Here the duplicate - detection algorithm calculates a hint that is for only particular attribute which is having more number of records which can be fit into the memory .So that by finishing one partition consisting of records of a huge dataset after another then the overall duplicate detection process will be slower. It is only incremental.

The SNM sorts the input data based on sorting keys and moves a window called sliding window which is constant in an order on the sorted records [5]. And the records with in the window are paired with each other. The Windows and blocking methods are used for limiting the number of comparison of records [5]. And the Remaining records are eliminated and possible records are grouped, and final Non-duplicate records after performing duplicate – detection [1] are displayed.

The Blocking Method is used to group the records based on high similarity attribute values using keys [5]. Blocking Methods select a set of duplicate records out of possible records by assigning blocks and eliminates duplicate records [5].

In this paper DBLP dataset is taken as input and on that input the proposed algorithms are implemented. DBLP is a bibliographic database for computer sciences [6]. The main problem in DBLP is the assigning of papers to entities related to author. It provides bibliographical information of computer science proceedings and journals which stores the data related to authors, which are used in writing the book or article etc that a user might find useful for identifying and retrieving the particular relevant data [6]. Due to duplicate or missing information present in dataset, the output provided may results incorrect statistics and when taking data from different sources or when different users use same data, there is a chance of occurring duplicates.

MODULES

In this system there is only one module as this project C comes under Admin Analytics.

Admin

1. The admin is responsible for granting access rights to the users for accessing required data. The admin has the main access permission for maintaining the datasets over databases.
2. Admin Logins by giving Username and password .If correct details then admin can select the Dataset to perform Duplicate - detection process on it in order to maintain clean and reliable data in databases.
3. Admin can view the log details of the activities performed by different users on the data and perform operations like deleting unnecessary data and modify the data which is present in databases.
4. By applying different techniques admin will identify the duplicate data if present in dataset and removes that particular duplicate data and stores the Non- duplicate data of the particular dataset in databases.

ALGORITHM DESCRIPTION

EFFICIENT SORTED NEIGHBORHOOD METHOD (PSNM):

1. Load and Partition the input DBLP XML dataset.
2. Apply Attribute Concurrency for getting the keys in Sorted order.
3. Sort the partitioned data using sorting keys.
4. Update the partition after sorting.

Page 1-5 © MAT Journals 2017. All Rights Reserved

5. Compare sorted ordered records within partition and with remaining partitions using Windows.
6. Eliminate the identified Duplicate records and display the resultant Non-Duplicate Records.

PROGRESSIVE BLOCKING (PB):

1. Load and Partition the input DBLP XML dataset.
2. Apply Attribute Concurrency for getting the keys in Sorted order.
3. Sort the partitioned data using sorting keys.
4. Update the partition after sorting.
5. Compare sorted ordered records using blocks.
6. Eliminate the identified Duplicate records and display the resultant Non-Duplicate Records.

COMPARISON OF PSNM AND PB ALGORITHMS

Here the efficiency of proposed algorithms is given by time and the memory taken by the algorithms with different keys as shown in the below Figs.

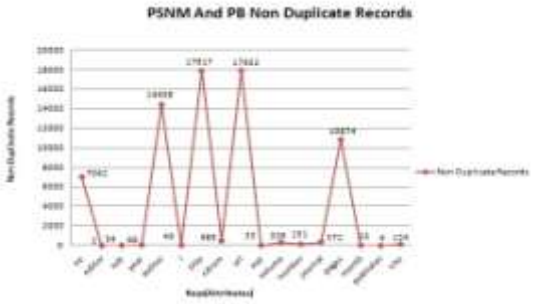

Fig. 1: *Non Duplicate Records Graph.*

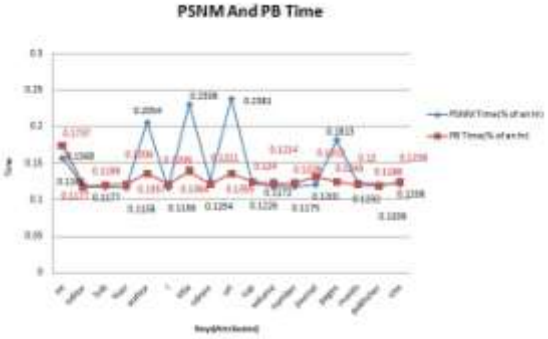

Fig. 2: *Time Taken Graph.*

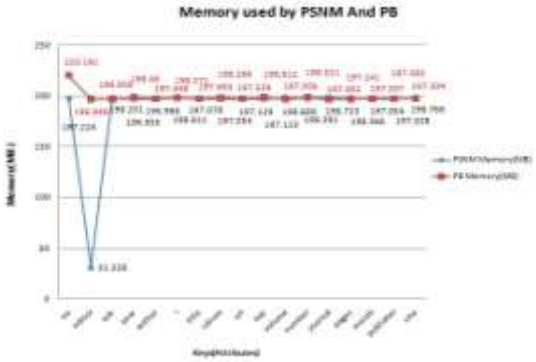

Fig. 3: *Memory Used Graph.*

CONCLUSION

In this paper when compared to PSNM, PB delivers fast results than PSNM when taking the DBLP as input dataset [7].

REFERENCES

1. A.K. Elmagarmid, P. G. Ipeirotis, and V.S. Verykios, "Duplicate record detection: A survey," IEEE Trans. Knowl. Data Eng., vol. 19, no. 1, pp. 1–16, Jan. 2007.
2. C. Xiao, W. Wang, X. Lin, and H. Shang, "Top-k set similarity joins," in Proc. IEEE Int. Conf. Data Eng., 2009, pp.
3. S. E. Whang, D. Marmaros, and H. Garcia-Molina "Pay-as-you-go entity resolution," IEEE Trans. Knowl. Data Eng., vol. 25, no. 5, pp. 1111–1124, May 2012.
   S. R. Jeffery, M. J. Franklin, and A. Y. Halevy, "Pay-as-you-go user feedback

Page 1-5 © MAT Journals 2017. All Rights Reserved

for data space systems," in Proc. Int. Conf. Manage. Data, 2008
4. http://dbs.uni-leipzig.de/file/multi_pass_sn with mr.pdf http://dblp.org/db/
5. https://hpi.de/fileadmin/user_upload/fachgebiete/ Naumann/publications/2014/Progressive Duplicate Detection. pdf.

Page 1-5 © MAT Journals 2017. All Rights Reserved