ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

PARALLEL AND MULTIPLE DISTRIBUTED PROCESS WITH PROGRESSIVE DUPLICATE DETECTION MODEL

[1] S. Thiruvenkatasamy, [2] R. Pradeepa, [3] R. Priyadharshini
[1] Assistant Professor, Computer Science and Engineering, Nandha College of Technology, Erode.
[2,3] UG Scholar, Computer Science and Engineering, Nandha College of technology,Erode.

ABSTRACT

In present, duplicate detection methods need to process ever larger datasets in ever shorter time, maintaining the quality of a dataset becomes increasingly difficult. This project presents two novel, progressive duplicate detection algorithms that significantly increase the efficiency of finding duplicates if the execution time is limited. They maximize the gain of the overall process within the time available by reporting most results much earlier than traditional approaches Comprehensive experiments show that progressive algorithms can double the efficiency over time of traditional duplicate detection and significantly improve upon related work. Data are among the most important assets of a company. But due to data changes and sloppy data entry, errors such as duplicate entries might occur, making data cleansing and in particular duplicate detection indispensable. As independent persons change the product portfolio, duplicates arise. Although there is an obvious need for de duplication, online shops without downtime cannot afford traditional de duplication. Progressive duplicate detection identifies most duplicate pairs early in the detection process. Instead of reducing the overall time needed to finish the entire process, progressive approaches try to reduce the average time after which a duplicate is found. Early terminations, in particular, then yields more complete results on a progressive algorithm than on any traditional approach.

1. INTRODUCTION

1.1 DATA MINING

Data mining, or knowledge discovery, is the computer-assisted process of digging through and analyzing enormous sets of data and then extracting the meaning of the data. Data mining tools predict behaviors and future trends, allowing businesses to make proactive, knowledge-driven decisions. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

Data mining derives its name from the similarities between searching for valuable information in a large database and mining a mountain for a vein of valuable ore. Both processes require either sifting through an immense amount of material, or intelligently probing it to find where the value resides. Although data mining is still in its infancy, companies in a wide range of industries - including retail, finance, heath care, manufacturing transportation, and aerospace - are already using data mining tools and techniques to take advantage of historical data. By using pattern recognition technologies and statistical and mathematical techniques to sift through warehoused information, data mining helps analysts recognize significant facts, relationships, trends, patterns, exceptions and anomalies that might otherwise go unnoticed. For businesses, data mining is used to discover patterns and relationships in the data in order to help make better business decisions. Data mining can help spot sales trends, develop smarter marketing campaigns, and accurately predict customer loyalty.Data mining technology can generate new business opportunities by:

1.1.1Automated Prediction Of Trends And Behaviors:

Data mining automates the process of finding predictive information in a large database. Questions that traditionally required extensive hands-on analysis can now be directly answered from the data. A typical example of a predictive problem is targeted marketing. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings. Other predictive problems include forecasting bankruptcy and other forms of default, and identifying segments of a population likely to respond similarly to given events.

1.1.2Automated discovery of previously unknown patterns:

Data mining tools sweep through databases and identify previously hidden patterns. An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions and identifying anomalous data that could represent data entry keying errors.

While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks. Generally, any of four types of relationships are sought:

- **Classes**: Stored data is used to locate data in predetermined groups.
- **Clusters**: Data items are grouped according to logical relationships or consumer preferences.
- **Associations**: Data can be mined to identify associations.
- **Sequential patterns**: Data is mined to anticipate behavior patterns and trends.

2. PROPOSED SYSTEM

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 10, March 2016

The proposed system contains the existing system proposes also. In addition, the overall records are kept in multiple resources after splitting. The intermediate duplication results are intimated immediately after fount in any resources and are returned to the main application in proposed system. So the time consumption is reduced. Like wise the resource consumption is split across the resources.

2.1. ADVANTAGES

The proposed system has following advantages.

- Concurrent approach is used. i.e., all the records are taken and checked as a parallel processes.
- Execution time is reduced.
- Resource consumption is same as existing system but the data is kept in multiple resource memories.

3. MODULE DESCRIPTION

The following modules are present in the project.

Record addition

Attribute selection for duplication finding

Input parameter settings for psnm/pb

Progressive sorted neighborhood method algorithm

Progressive blocking

Proposed system (psnm and progressive blocking)

3.1. RECORD ADDITION

In this module, the records are added for the given columns (Employees/Attendance). The records may contain any data. The records are saved in ‘Employees’ and ‘Attendance’ table.

3.2. ATTRIBUTE SELECTION FOR DUPLICATION FINDING

In this module, which columns are selected for finding duplicates in records.

3.3. INPUT PARAMETER SETTINGS FOR PSNM/PB

INPUT PARAMETERS (D, K, W, I, N)

In this module, input for the Algorithm PSNM is selected. The algorithm takes five input parameters: D is a reference to the data, which has not been loaded from disk yet. The sorting key K

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 10, March 2016

defines the attribute or attribute combination that should be used in the sorting step. W specifies the maximum window size, which corresponds to the window size of the traditional sorted neighborhood method. When using early termination, this parameter can be set to an optimistically high default value. Parameter I defines the enlargement interval for the progressive iterations. N is the number of records.

INPUT PARAMETERS (D, K, R, S, N)

In this module, input for the Algorithm PB is selected. The algorithm takes five input parameters: D is a reference to the data, which has not been loaded from disk yet. The sorting key K defines the attribute or attribute combination that should be used in the sorting step. R specifies the maximum block range, S Block Size and N Total No. of Records. When using early termination, this parameter can be set to an optimistically high default value.

3.4.PROGRESSIVE SORTED NEIGHBORHOOD METHOD ALGORITHM

The PSNM algorithm calculates an **appropriate partition size** pSize, i.e., the maximum number of records that fit in memory, using the pessimistic sampling function **calcPartitionSize**(D) in Line 2: If the data is read from a database, the function can calculate the size of a record from the data types and match this to the available main memory. Otherwise, it takes a sample of records and estimates the size of a record with the largest values for each field.

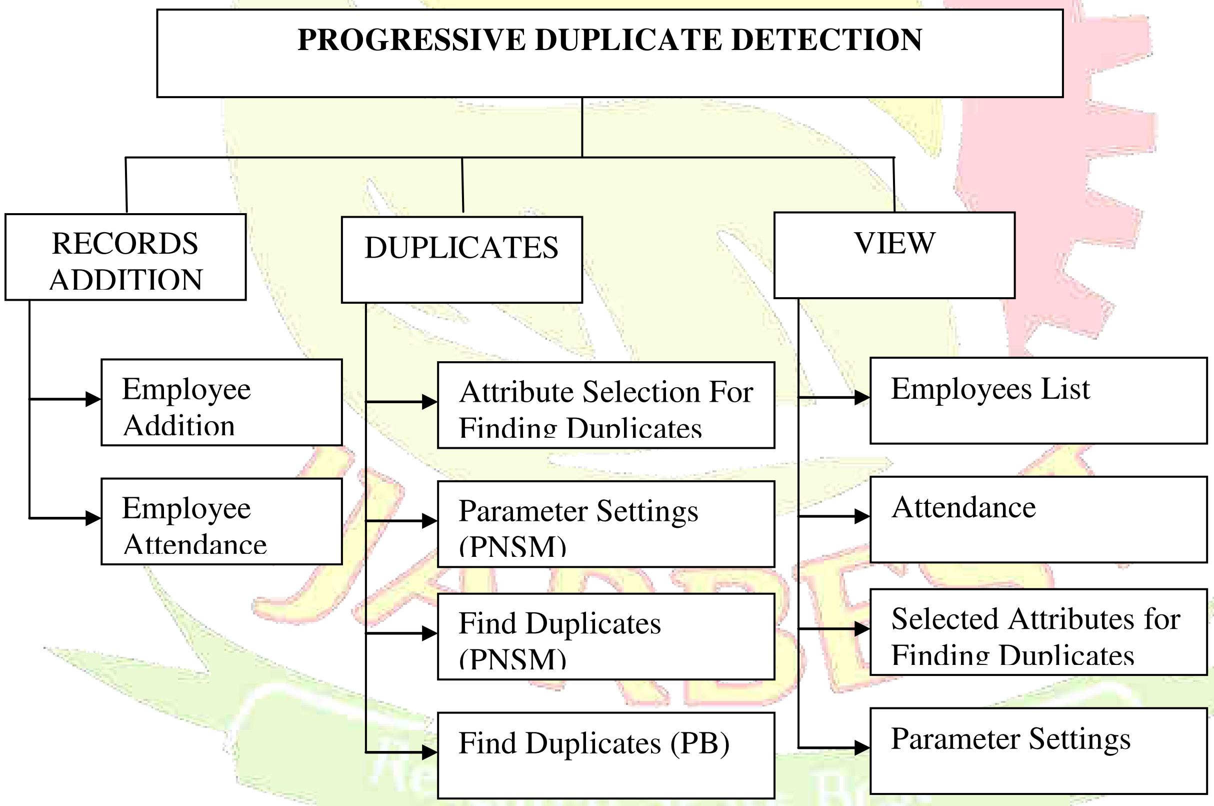
The algorithm calculates the number of necessary partitions pNum, while considering a partition overlap of W - 1 records to slide the window across their boundaries. Line 4 defines the order-array, which stores the order of records with regard to the given key K. By storing only record IDs in this array, we assume that it can be kept in memory. To hold the actual records of a current partition, PSNM declares the recs-array.

PSNM sorts the dataset D by key K. The sorting is done using progressive sorting algorithm. Afterwards, PSNM linearly increases the window size from 2 to the maximum window size W in steps of I. In this way, promising close neighbors are selected first and less promising far-away neighbors later on. For each of these progressive iterations, PSNM reads the entire dataset once. Since the load process is done partition-wise, PSNM sequentially iterates and loads all partitions. To process a loaded partition, PSNM first iterates overall record rank-distances dist that are within the current window interval current I. For I= 1 this is only one distance, namely the record rank-distance of the current main-iteration. In PSNM then iterates all records in the current partition to compare them to their dist-neighbor. **If this function returns "true", a duplicate has been found and can be emitted.**

3.5. PROGRESSIVE BLOCKING

In this module, dataset reference D, key attribute K, maximum block range R, block size S and record number N are given as input.

The algorithm accepts five input parameters: The dataset reference D specifies the dataset to be cleaned and the key attribute or key attribute combination K defines the sorting. The parameter R limits the maximum block range, which is the maximum rank-distance of two blocks in a block pair, and S specifies the size of the blocks. Finally, N is the size of the input dataset. At first, PB calculates the number of records per partition pSize by using a pessimistic sampling function in Line 2. The algorithm also calculates the number of loadable blocks per partition bPerP, the total number of blocks bNum, and the total number of partitions pNum. In the Lines 6 to 8, PB then defines thethree main data structures: the order-array, which stores the ordered list of record IDs, the blocks-array, which holds the current partition of blocked records, and the bPairs-list, which stores all recently evaluated block pairs. Thereby, a block pair is represented as a triple of (blockNr1, blockNr2, duplicates Per Comparison).



The bPairs-list is implemented as a priority queue, because the algorithm frequently reads the top elements from this list. The PB algorithm sorts the dataset using the progressive Magpie Sort algorithm. Afterwards, load all blocks partition-wise from disk to execute the comparisons within each block.

3.6. PROPOSED SYSTEM (PSNM AND PROGRESSIVE BLOCKING)

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

In this module, the algorithms are same as module 4 and 5 but the records are split and each window partition is given to different systems (resources i.e., computers) so that the process is executed in parallel.

4. CONCLUSION

Through this project, the efficiency of duplication detection is increased. This project introduced the progressive sorted neighborhood method and progressive blocking. Both algorithms increase the efficiency of duplicate detection for situations with limited execution time; they dynamically change the ranking of comparison candidates based on intermediate results to execute promising comparisons first and less promising comparisons later. The project proposed a novel quality measure for progressiveness that integrates seamlessly with existing measures.

It uses multiple sort keys concurrently to interleave their progressive iterations. By analyzing intermediate results, both approaches dynamically rank the different sort keys at runtime, drastically easing the key selection problem. It is believed that almost all the system objectives that have been planned at the commencements of the software development have been net with and the implementation process of the project is completed.

A trial run of the system has been made and is giving good results the procedures for processing is simple and regular order. The process of preparing plans been missed out which might be considered for further modification of the application.

REFERENCES

[1] 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.

[2] 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.

[3] F. Naumann and M. Herschel, An Introduction to Duplicate Detection.San Rafael, CA, USA: Morgan & Claypool, 2010.

[4] H. B. Newcombe and J. M. Kennedy, "Record linkage: Making maximum use of the discriminating power of identifying
information," Commun. ACM, vol. 5, no. 11, pp. 563–566, 1962.

[5] M. A. Hernandez and S. J. Stolfo, "Real-world data is dirty: Data
cleansing and the merge/purge problem," Data Mining Knowl. Discovery, vol. 2, no. 1, pp. 9–37, 1998.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 10, March 2016

[6] X. Dong, A. Halevy, and J. Madhavan, “Reference reconciliation in complex information spaces,” in Proc. Int. Conf. Manage. Data,2005, pp. 85–96.

[7] O. Hassanzadeh, F. Chiang, H. C. Lee, and R. J. Miller,“Framework for evaluating clustering algorithms in duplicate detection,” Proc. Very Large Databases Endowment, vol. 2, pp. 1282–1293, 2009.

[8] O. Hassanzadeh and R. J. Miller, “Creating probabilistic databases from duplicated data,” VLDB J., vol. 18, no. 5, pp. 1141–1166, 2009.

[9] U. Draisbach, F. Naumann, S. Szott, and O. Wonneberg,“Adaptive windows for duplicate detection,” in Proc. IEEE 28th Int. Conf. Data Eng., 2012, pp. 1073–1083.

[10] S. Yan, D. Lee, M.-Y. Kan, and L. C. Giles, “Adaptive sorted neighborhood methods for efficient record linkage,” in Proc. 7th ACM/IEEE Joint Int. Conf. Digit. Libraries, 2007, pp. 185–194.

[11] J. Madhavan, S. R. Jeffery, S. Cohen, X. Dong, D. Ko, C. Yu, and A. Halevy, “Web-scale data integration: You can only afford to pay as you go,” in Proc. Conf. Innovative Data Syst.Res., 2007.

[12] S. R. Jeffery, M. J. Franklin, and A. Y. Halevy, “Pay-as-you-go user feedback for dataspace systems,” in Proc. Int. Conf. Manage. Data,2008, pp. 847–860.

[13] C. Xiao, W. Wang, X. Lin, and H. Shang, “Top-k set similarity joins,” in Proc. IEEE Int. Conf. Data Eng., 2009, pp. 916–927.

[14] P. Indyk, “A small approximately min-wise independent family of hash functions,” in Proc. 10th Annu. ACM-SIAM Symp. Discrete Algorithms, 1999, pp. 454–456.Fig. 10. Duplicates found in the plista-dataset.1328 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 5, MAY 2015

[15] U. Draisbach and F. Naumann, “A generalization of blocking and windowing algorithms for duplicate detection,” in Proc. Int. Conf.Data Knowl. Eng., 2011, pp. 18–24.