IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 04-06
www.iosrjournals.org

Progressive Algorithms for Efficient Duplicate Detection

Anusha Kenno[1], Bindhu J S[2]

[1]*(Department of Computer Science, College of Engineering Perumon, Kollam, India)*
[2]*(Department of Computer Science, College of Engineering Perumon, Kollam, India)*

Abstract: *Duplicate detection is the technique of identifying or detecting all group of record within a dataset that represent the same real world entity. Duplicate detection methods process large datasets in shorter time but maintaining the quality of the dataset becomes difficult, which is a major data quality concern in large databases. To address this progressive algorithm has been proposed that significantly increase the efficiency of finding duplicates if the execution time is limited and improve the quality of records. The overall gain of the process will be maximized by reporting results much earlier and can double the efficiency as compared to traditional approaches. This proposed system is less time consuming method with more accurate result as compared to the existing approaches. Experiments show that the progressive algorithm can double the efficiency of the traditional approaches with shorter execution time.*

Keywords: *Duplicate detection, blocking, progressive algorithm, windowing, data cleaning*

I. Introduction

Databases play an important role in IT and economy based industries. Many companies depend on the efficiency of databases to carry out all operations. Therefore, the qualities of records that are stored in the databases have significant cost indications to a system that relies on information to conduct business. Data has to be in integrity, and if exceeds the criteria then it is duplicate. Data is considered as an important asset of a company but due to data changes and sloppy data entries duplication arises.

One of the main facts of duplicate detection is that it detects duplicates early in the detection process. The pure size of the dataset makes duplicate detection an expensive process. So that the progressive algorithms reduce the average time after which a duplicate is found rather than reducing the average time to finish the entire process. The duplicate detection algorithm such as the incremental algorithm and pair selection techniques are used in the detection process. Certain problems occur in the detection process and have several use cases such as a user may have limited or unknown time for data cleansing, also user may have little knowledge about the given data. It is not possible to eliminate several factors of duplicate detection such as effectiveness and scalability due to database size.

There are two features in the problem of duplicate detection such as several representation are not same and have certain differences like misspelling, missing values, changed addresses which makes the detection of duplicates difficult. Secondly, the detection of duplicates is very challenging task due to the comparison among all possible pair is required. The challenges in the duplicate detection process are the vast amounts of records and that finding the duplicates is resource intensive. Duplicate detection plays an important role in customer relationship management, personal information management, and data mining.

Two approaches of progressive algorithms include the progressive sorted neighborhood method (PSNM) which performs best on small and clean dataset and the progressive blocking (PB) which performs best on large and dirty datasets. For the given fixed time slot in which data cleansing is possible, progressive algorithms try to maximize their efficiency for that given amount of time. Both algorithms adjust their behavior by dynamically choosing parameters such as block sizes, window sizes and sorting keys respectively. One of the main aspects of duplicate detection process is that multiple representations are not same and it contains differences mainly double data entries, change of same address and missing keys. As a result it is difficult to detect duplicates. Duplicate detection is considered as a complex operation which requires the comparison of all the available possible pair of duplicates using the complexity similarity calculation. Duplicate detection is an expensive operation considering the pure size of the dataset.

II. Existing System

In [1], Entity resolution can be considered as the problem of identifying records in a database that refers to the same entity. This paper specify how to maximize the progress of ER with the limited amount of work using hints, which provide information on records that refer to the same real world entity. It is possible to represent hints in various formats, ER uses this information to determine which records to compare first.

Various techniques are used for constructing hints which maximizes the number of records identified with limited amount of time.

In [2], Duplicate detection is a difficult task because representation differs slightly so similarity measures are defined to compare pair of records, and also the dataset may have high volume making comparison difficult. Here it propose the Duplicate count strategy, a contrast to SNM which uses a varying window size. It is based on the intuition that there might be regions of high similarity which requires a larger window size and regions of lower similarity which requires a smaller window size. A variant called DCS++ is better than the original SNM in terms of efficiency.

In [3], World Wide Web is witnessing an increase in the amount of structured content which is a vast heterogeneous collection of structured data which arise due to the Deep Web and sites like Google base. This paper highlights these challenges in two scenarios-the Deep Web and Google Base. It is contended that the traditional data integration techniques are no longer valid in heterogeneity and scale. A new data integration architecture is proposed PAYGO which is the concept of data spaces and emphasizes pay-as-you-go entity resolution, a concept of data management as means for achieving web scale data integration.

In [4], two approaches used are the blocking and windowing technique, where the blocking technique partition data into disjoint subsets and the windowing methods slide window over the sorted methods and compares records within the window. A new algorithm called sorting blocks in different variants which generalises both approaches, this new algorithm needed lesser number of comparisons to find the accurate number of duplicates. Therefore the challenge is to effectively and efficiently search for duplicates, an exhaustive duplicate detection process focuses on computing the similar feature of all record pairs which is very expensive for large datasets.

In [5], Similarity join is considered as a useful primitive operation for many applications such as near duplicate such as the detection of web pages, integrity of data and recognition of patterns, whereas the traditional similarity join needs a user to specify a similarity threshold. This paper uses a variant of the similarity join known as the top-k similarity join. This returns the top-k pairs of records ranked by their similarities which eliminate the users guess when the threshold is unknown. An algorithm known as the top-k joins is proposed for answering of the similarity join of top-k efficiently. It is focussed on the concept of prefix filtering principle and it computes answers in a progressive manner.

In [6], the presence of duplicate records is a major quality concern in large databases. For detecting duplicates, record linkage is used as a part of the data cleaning process for identifying records that refer to the same real word entity. Here it provides the Stringer system which is a framework towards the goal of truly scalable and general purpose duplicate detection algorithms. Stringer is used to evaluate the quality of clusters obtained from various unconstrained clustering algorithm. It reveals that some clustering algorithm that has never been considered for duplicate detection performs well in both accuracy and scalability.

In [7], the merging of information from multiple databases is encountered in KDD and decision support applications. The problem is called as the merge/purge problem which is difficult to solve both in accuracy and scale. Numerous duplicate information entries from large repositories of data are difficult to detect without an equational theory which identifies equivalent terms by a complex domain dependent matching process. Combing results of individual passes using transitive closuring over the not dependent results produces more accurate results at a faster rate.

In [8], the field of transitive relations mainly on dense, Boolean, undirected relations. With the arrival of the new area of intelligence retrieval, the sparse transitive fuzzy ordering relations are utilized. It is required that the existing theory and methodologies needs to be extended foe covering the newer needs. This paper proposes the incremental update of fuzzy binary relations focusing both on storage and computational complexity issues. It also proposes a transitive closure that has a low computational complexity for the average sparse relation such as observed in intelligent retrieval.

In [9], the problem of detecting duplicate records in a database is a crucial step of data cleaning and data integration process. It is possible that the real world entity can have one or more representation in databases. When considering large amount of data it is required that there is a well defined and tested mechanism for detecting duplicates. This paper analyzes the literature on duplicate record detection. This covers the similarity metric which is used to detect similar field entries and also increasing the efficiency and scalability and also discusses about use of coverage of various existing tools.

In [10], the major source of uncertainty in databases is due to the presence of duplicate items. Accurate de duplication is a difficult task and imperfect data cleaning result in considerable loss of information. The approach is to keep duplicates when the correct cleaning strategy is not uncertain and uses an efficient probabilistic query answering technique to return query results along with probabilities of each answer being correct. This paper presents a flexible modular framework for scalable creating probabilistic database out of dirty relation of duplicated data. It also considers the problem of associating probabilities with duplicates that are detected using state of art scalable approximate join methods.

III. Proposed System

The proposed system uses two types of efficient algorithms for progressive duplicate detection which are the PSNM, the progressive sorted neighbourhood method which operates on small and clean datasets and the PB, the progressive blocking algorithm which operates on large and dirty datasets. Both these algorithm increase the efficiency of finding duplicates with shorter execution time and satisfies two conditions such as improved early quality in which consider a target time at which results are to be produced, then the progressive algorithm discovers more number of duplicate pairs than the traditional approaches and it produces same eventual quality. It eases the complexity of duplicate detection and contributes to the development of various interactive applications. It introduces a multipass method and also adapts an incremental transitive closure algorithm which forms the compete basis for the progressive detection. One of the main advantage of the progressive duplicate detection is that it can detect duplicates early in the detection process

There are three stages in this workflow such as the pair selection, pair wise comparison and clustering. PSNM sorts the input data using a sorting key and compares the records within these blocks. It uses rank distance to calculate their matching strategy. Progressive blocking algorithm is based on the equidistant blocking technique and the successive enlargement of blocks. It assigns each block to a fixed set of similar records and compares the records within these blocks. Both these algorithms increase their efficiencies over huge datasets, and expose different strength and outperform current approaches. This approach is suitable for multipass method. To rank the performance, the progressive duplicate detection is measured using quality measures.

IV. Conclusion

This paper introduces two progressive algorithms such as the progressive sorted neighborhood method and progressive blocking. Both this algorithm increase the efficiency of finding duplicates if the execution time is limited, ranking of comparison candidates is changed based on intermediate results. The sorting key and the blocking key can be automatically adjusted based on parameter. Efficient duplicate detection is an important task in large datasets. This compares two approaches such as blocking and windowing for reducing the number of comparisons. This allows more comparisons if records have similar values. It evaluates strategies that group records with a high a chance of being duplicates in the same partition.

By analyzing intermediate results, both approaches dynamically rank the different keys that are sorted at execution time, which reduces the complexity of the key selection problem. To determine the performance gain of algorithms a novel quality measure is proposed for progressiveness that integrates with existing measures.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] M. A. Hern_andez 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.

[8] M. Wallace and S. Kollias, "Computationally efficient incremental transitive closure of sparse fuzzy binary relations," in Proc. IEEE Int. Conf. Fuzzy Syst., 2004, pp. 1561–1565.

[9] 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.

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