International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) e-ISSN: 2395-5325
Volume 2, Issue 3 (March'2016)

CONCURRENT PROGRESSIVE APPROACH FOR DETECTING THE DUPLICATION

B.Praveen Kumar,
Assistant Professor,
Computer science and engineering,
Velammal Institute of Technology,Chennai,India.

M.R.Priyanka,
Student,
Computer science and engineering,
Velammal Institute of Technology,Chennai,India.

C.Radhika,
Student,
Computer Science and Engineering,
Velammal Institute of Technology,Chennai,India.

A.Subashini,
Student,
Computer science and engineering,
Velammal Institute of Technology,Chennai,India.

Abstract: Concurrency is the ability of a system to allow for several operations to be carried out simultaneously. The ability to offer concurrency is unique to databases. This is one of the main properties that separates a database from other forms of data storage like spreadsheets.Duplicate detection lets organizations set duplicate detection policies and create duplicate detection rules for business and custom entities. These rules can be applied across different record types which allows for detection of fraud that takes place while rating a product in websites. For example, an organization may define that a lead is a duplicate of a contact, if they have the same name, same IP address, same ID. Based on the duplicate detection rules set by the administrator, the system alerts the admin about potential duplicates. To maintain data quality, you can schedule a duplicate detection job to check for duplicates for all records that match a certain criteria. You can clean the data by deleting, deactivating or merging the duplicates reported by a duplicate detection job. The proposed system adds more strength for detection of duplication by using ranking methodology.

Keywords-Duplicate detection, Data cleaning, Record linkage, Data deduplication.

I.INTRODUCTION:

Whenever the duplicates has to be found from dataset we go for Data mining. The Data mining takes its' concepts from Knowledge Discovery in Database (KDD) in the field of computer science. In the recent past,duplication is becoming a major threat in almost all the domains. Because of this duplication the data received is more and thus memory limitation becomes arduous. Thus admin finds it difficult to manage the data sets.

The duplicate detection processes are expensive. The common people keep changing their portfolio despite retailers offering many product catalogs.Thus there occurs duplication in wide range and all the organisations cannot afford for the deduplication process as it is expensive.

The adaptive techniques improve the efficiency in detecting the duplication but these techniques cannot bare up to the level of progressive techniques. The Progressive techniques could process larger dataset in short span of time and the quality of data is also good comparatively [2]. The Progressive duplicate detection makes it different from the traditional approach by yielding more complex results during the early termination [1].

The algorithms of duplicate detection also computes the duplicates at an almost constant frequency but the progressive algorithms increase the overall time as it finds out the duplicates at the early stage itself. The candidate keys in the record pairs that are identical have to be first found out. The pair selection techniques of the duplicate detection process exhibits a trade-off between the amounts of time needed to run a duplicate detection algorithm and the completeness of the results [1]. This trade-off is made more efficient by the progressive detection techniques as it computes the results in shorter amount of time.

Sometimes the duplication could also be performed taking into account the window size.To avoid a prohibitively expensive comparison of all pairs of records, a common technique is to carefully partition the records into smaller subsets and thus fitting them to a particular window. If similar records appear in the same partition and within the same window, then the data is declared duplicate [4].

If the window size is selected too small, some duplicates might be missed. If the window size is selected large enough to find all duplicates even for the largest cluster, then there are a lot of unnecessary comparisons in the area of the smaller clusters.The variety of parameters that have to be set by a user is so complex. Due to space limitation, it can only be used for singleton datasets.

Progressive duplicate detection algorithms are Progressive Sorted Neighborhood Method (PSNM) and Progressive Blocking (PB) [1] .Progressive Sorted Neighborhood Method (PSNM) operates only on small datasets, whereas the Progressive blocking (PB) performs best on large datasets. PSNM sorts the input data using a predefined sorting key and only compares records that are within a window of records in the sorted order.PB sorts the input data and compares itself within the blocks.

The proposed system enhances the efficiency of duplicate detection even on very large datasets.The parameterization complexity for duplicate detection is made comfortable in general and contribute to the development of more user interactive applications. The quality of datasets is enhanced by using the proposed algorithm. The duplicates can be detected within a very short span of time thus providing faster access. The window size can be changed dynamically.

II.RELATED WORK

One entity can have several representations. There is no common key that the duplicate records share. Thus the duplicate detection is found to be a difficult task and they also contain errors that makes it difficult for duplicate matching. . Errors are introduced as the result of transcription errors, incomplete information, lack of standard formats, or any combination of these factors [2].

The duplicate detection could be difficult for various reasons like, the duplicate records might not have the same key and datasets might have high volume making the pairwise comparison of all record infeasible [6]. To avoid this algorithms have been suggested to partition the datasets in to smaller groups. The larger dataset has to be partitioned in to smaller ones and the comparison has to be made within those partitions alone. But this could not be possible for very larger datasets like online applications.

The duplicate detection techniques mainly focus on pair selection algorithms. These techniques mainly focus on reducing the overall runtime and identifying the pairwise keys. For the same reason, certain approaches use the windowing techniques. .In windowing technique the records are carefully partitioned into smaller subsets. Each subset has few data in to it.In these methods the data within each subset is compared with each other and if two data are found to be identical then it considers the data as duplicate copy of each other. But these approaches can be used only for certain window size and can't be further enhanced.

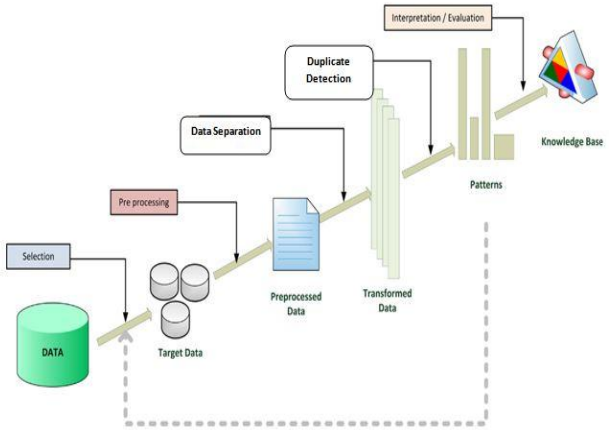

Figure1. Detection of Duplicate Datasets

Record linkage[16] (RL) refers to the task of finding records in a data set that refer to the same entity across different data sources (e.g., data files, books, websites, databases). Record linkage is necessary when joining data sets based on entities that may or may not share a common identifier (e.g., database key, URI, National identification number).When the records are matched on a single database it is called as deduplication. The duplicate detection is a crucial process when applied on a single database. The number of record pairs that ought to be compared in matching process is done by removing the nonmatching pairs and by thus ensuring the quality is maintained [3].

The partition caching is one of the technique used for detecting the duplicates. The previously used algorithms has to repeatedly reiterate the entire file while searching for the duplicate records [1]. In partition caching when a record has been read once it takes the records and stores it separately on to the cache. So when the record has to be computed again for detecting the duplication it can be taken from the cache thus time and cost could be less.

In Record matching [20] algorithms the existence of duplicate records constitutes a problem which is becoming increasingly alarming in networked environments, as the size of individual databases increases and new cooperative networks or consortia are created. Special algorithms are developed for this purpose. The integrity of bibliographic databases are maintained with the algorithms of record matching. The similar records could never be matched at any stage despite matching the bibliographic descriptions.

The progressive techniques like pay-as-you-go [13] algorithms were used for integration on large scale datasets. In pay-as-you-go, we theoretically order the candidate pairs by the chances of a match. The ER algorithms are used for performing this.Entity resolution (ER) is the problem of identifying which records in a database refer to the same entity. For real-time applications ER processing takes longer than a certain amount of time.The progress of ER can be maximised using hints that give information on records that are likely to refer to the same real-world entity. A hint can be represented in various formats and ER uses this information as a guideline for which records are computed first [5].

```
for i : 1 . . . N do {Initialization}
    R[i] ← (1 − α)/((α^T +1   )N) {Normalization}
    if T≥ 0 then                  {Normal PageRank or Truncated?}
        Score[i] ← 0
    else {Calculate normal PageRank}
        Score[i] ← R[i]
    end if
end for
distance = 1
while not converged do
    Aux ← 0
    for src : 1 . . . N do {Follow links in the graph}
        for all link from src to dest do
            Aux[dest] ← Aux[dest] + R[src]/outdegree(src)
        end for
    end for
    for i : 1 . . . N do {Apply damping factor α}
        R[i] ← Aux[i] ×α
        if distance > T then {Add to ranking value}
            Score[i] ← Score[i] + R[i]
        end if
    end for
    distance = distance +1
end while
return Score
```

Figure2: Hummingbird algorithm

The Magpie sort is one of the sorting technique used in progressive duplicate detection [1]. It serves its purpose similar to the selection sort. The selection sort is a sorting algorithm, specifically an in-placecomparison sort. It has O (n^2) time complexity, making it inefficient on large lists. The algorithm divides the input list into two parts: the sub list of items already sorted, which is built up from left to right at the front (left) of the list, and the sub list of items remaining to be sorted that occupy the rest of the list. Initially, the sorted sub list is empty and the unsorted sub list is the entire input list. The algorithm proceeds by finding the smallest (or largest, depending on sorting order) element in the unsorted sub list, exchanging (swapping) it with the leftmost unsorted element (putting it in sorted order), and moving the sub list boundaries one element to the right.

III.ALGORITHM

The Progressive duplicate detection algorithms are Progressive Sorted Neighborhood Method (PSNM) and Progressive blocking (PB) [1] .Progressive Sorted Neighborhood Method (PSNM) performs best on small and almost clean datasets. PSNM sorts the input data using a predefined sorting key and only compares records that are within a window of records in the sorted order. Progressive blocking (PB) performs best on large and very dirty datasets.PB sorts the input data and compares itself within the blocks.

The PSNM is expensive as it has to load all records in each iteration. To avoid this the window size was enlarged and divided into partitions so that the comparison to detect the duplicates can be found within the partition itself. The PSNM has two phases namely the load phase and compare phase. The records partitioned are read from disk into main memory in the first phase and the comparison is carried out in the second phase. The difference between the PSNM and PB is that PB sorts the record first and splits it in to blocks. It splits the similar records into blocks and then makes the comparison. It uses block comparison matrix [1].

The clustering algorithm is also one of the algorithms used for determining the duplication. Clustering is known as grouping of data based on their similarities [2]. This paper introduces an algorithm of k means for clustering of data streams and detection of outliers. The introduced technique for detection of outliers is based on distance as well as on time on which they arrive in the cluster. This paper also takes into account the selection of k canters and variable size of buckets with the help of which space can be effectively utilized during clustering. Most traditional algorithms make clustering a very difficult problem by reducing their quality for a better efficiency. This paper indicates that with a small increase in time you can efficiently cluster the data without much loss of quality of data [12].

Hummingbird is aimed at making interactions more human — capable of understanding the concepts and relationships between keywords. Hummingbird places greater emphasis on page content making search results more relevant and pertinent and ensuring that search engine delivers users to themost appropriate page of a website, rather than to a home page or top level page.

Search engine optimization changed little with the addition of Hummingbird, though more top ranking results are ones that provide natural content that reads conversationally. While keywords within the query still continue to be important, Hummingbird adds more strength to long-tailed keywords — effectively catering to the optimization of content rather than just keywords.

Webmasters will now have to cater towards queries that are asked naturally; with the growing number of conversational queries — namely those using voice search, targeting phrases that start with "Who, Why, Where, and How" will prove beneficial towards SEO. The use of keyword synonyms have also been optimized with Hummingbird; instead of listing results with exact phrases or keywords, database shows more theme-related results. So instead of using one word, you can use a completely different word. Admin understands that user mean the same thing but user said it without repeating the word over and over again.

In this paper, this algorithm is becoming constantly better at detecting natural quality content and duplicate spun content. The IP address, user id and user name are the parameters that are used for detecting the duplication. When the user seems to be using the above parameters repeatedly this algorithm identifies that the user has already rated the product. This algorithm however reduces the number of times a user can rate a product. Firstly the user will be provided with the login form that has to be filled by the user. Now is that the user can login with the id provided .The user when rates a product, his rating strategy will be stored onto the database. The admin

can upload the product specification and newly introduced products and the admin can view the ratings done by the user. When the same user rates the same product 'n' number of times, this algorithm automatically detects the forgery and it considers all the ratings as a single rating. The algorithm finds the rating to be forged when the user uses the same login id, does rating from the same IP address and when all certain criteria's are same as that of previously matched one.

IV.PROPOSED SYSTEM

Authentication is any process by which a system verifies the identity of a User who wishes to access it.In authentication the credentials provided are compared to those on file in a database of authorized users and thus a user name and a login is provided Thus firstly, the user has to fill-in the particulars so as to register to rate the product. The details of users have been stored in a separate database. So whenever the user wants to rate a product the user can login using the credentials got during the registration.

When the new products are into the market the admin has to update the details about the newly released product frequently.The admin has the right to authority as in other cases. The file includes the details of the products arrived newly in market. This is named as file uploading or updating. In the file the details like rate of the product, discount offered by the company as per norms, launch date etc. will be included. The database will be managed and updated by the admin.

The most important task is duplicate detection. The process of identifying multiple representation is generally named as duplicate detection. Generally, a particular user may rate the product two or more times. This multiple representation is named as duplication. The process of duplicate detection generally comprises three steps such as pair-selection, pair wise comparison and clustering. Here the efficiency of duplicate detection is improved by the concurrent approach. The detection and elimination of duplication is done concurrently.

A rating scale is a set of categories designed to elicit information about a quantitative or a qualitative attribute. A rating scale is a method that requires the user to assign a value, sometimes numeric, to the rated object, as a measure of some rated attribute. The rating scale can be of different categories. More than one rating scale is required to measure an attitude or perception due to the requirement for statistical comparisons. Rating scales are used widely online in an attempt to provide indications of consumer opinions of products .The ranking is done asper the ratings given by the user.

Firstly, the user would login using the id given that is done after filling in the form and registering .The registration form includes the user name, a password, mobile number and e-mail id and when submitted the details of the user will be stored to the database. Now is that, the user can get into the page that they want. The list of products are listed and the user when clicks a product can see its description along the cost. The user can give the review for the product. The users can also rate a product after they have seen its usage or by looking into the description provided to it. This is also stored along with users' credentials after the user submits it.Now the admin has the right to authority as in other scenarios. The admin is provided with a name, password and after logging in the admin can upload the files that includes the details regarding the products being newly launched in the market. These uploading of files takes place simultaneously as the user logins as and when product is newly launched. When the user who has rated a product tries to rate the same product using the above credentials then the data stored onto the database is retrieved and it tells that the person has already rated a product and how many ever times the user rates the product it is considered as a single rating only. The algorithm automatically detects the forgery when the user rates the product continuously by tracing the IP address and by ascertaining certain other conditions.

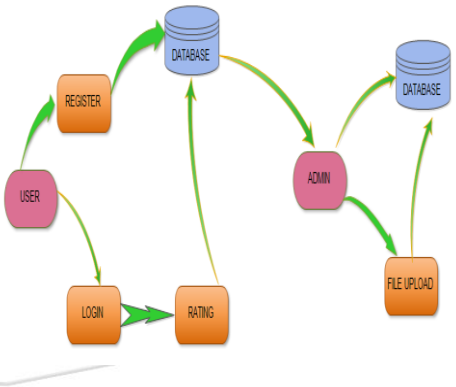

Figure3.Detetion of duplicate datasets using algorithm

V.RESULT EVALUATION

Detection of duplication was very arduous in the beginning. Later on many algorithms were proposed to detect the duplication, but each algorithm had its' own drawbacks. At each stage, certain technologies were used with advancement at each stages. In this paper, the advancement brought was to detect the duplication concurrently along with several other operations. The graph indicates the improvement of performance in detecting the duplication over a period of time drastically.

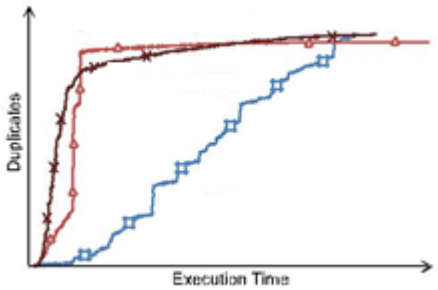

Figure4: Performance evaluation for detection of duplication

VI.CONCLUSION

This paper introduces an algorithm that is used for detecting duplication concurrently. This algorithm 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 parameterization complexity for duplicate detection is made comfortable in general and contribute to the development of more user interactive applications. The proposed system enhances the efficiency of duplicate detection even on very large datasets. The quality of datasets is enhanced by using the proposed algorithm. The duplicates can be detected within a very short span of time thus providing faster access.

REFERENCES

[1]. "Progressive Duplicate Detection", Thorsten Papenbrock, Arvid Heise, and Felix Naumann, IEEE Transactions on Knowledge and Data Engineering,May 2015.

[2]. Efficient and Effective Duplicate Detection Evaluating Multiple Data using Genetic Algorithm, Dr.M.Mayilvaganan, M.Saipriyanka, Sep 2015.

[3]. P. Christen, "A survey of indexing techniques for scalable record linkage and deduplication," IEEE Trans. Knowl. Data Eng., vol. 24, no. 9, Sep. 2012.

[4]. U. Draisbach and F. Naumann, "A generalization of blocking and windowing algorithms for duplicate detection," in Proc. Int. Conf. Data Knowl. Eng., 2011

[5]. S. E. Whang, D. Marmaros, and H. Garcia-Molina, "Pay-as-you-go entity resolution," IEEE Trans. Knowl. Data Eng., vol. 25, no. 5, May 2012.

[6]. Adaptive Windows for Duplicate Detection, Uwe Draisbach Felix Naumann , Sascha Szott , Oliver Wonneberg.

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

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

[9]. F. J. Damerau, "A technique for computer detection and correction of spelling errors," Commun. ACM, vol. 7, no. 3, pp. 171–176, 1964.

[10]. L. Kolb, A. Thor, and E. Rahm, "Parallel sorted neighborhood blocking with MapReduce," in Proc. Conf. Datenbanksysteme in B€uro, Technik und Wissenschaft, 2011.

[11]. B. Kille, F. Hopfgartner, T. Brodt, and T. Heintz, "The Plista dataset," in Proc. Int. Workshop Challenge News Recommender Syst., 2013, pp. 16–23.

[12]. 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.

[13]. 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.

[14]. 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.

[15]. 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.

[16]. L. Gu and R. Baxter, "Adaptive filtering for efficient record linkage," in Proceedings of the SIAM International Conference on Data Mining, 2004, pp. 477–481.

[17]. M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," in Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 2003, pp. 39–48.

[18]. X. Dong, A. Halevy, and J. Madhavan, "Reference reconciliation in complex information spaces," in Proceedings of the ACM International Conference on Management of Data (SIGMOD), 2005, pp. 85–96.

[19]. O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S. E. Whang, and J. Widom, "Swoosh: a generic approach to entity resolution," VLDB Journal.

[20]. R. Baxter, P. Christen, and T. Churches, "A comparison of fast blocking methods for record linkage," in Proceedings of the ACM SIGKDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation, 2003, pp. 25–27.