Duplicate Detection with Map Reduce and Deletion Procedure

Ms.Tanvee Meshram
Computer Engineering department
G.H.Raisoni college of engineering and Management
Wagholi , Pune,India

Prof.Nivedita Kadam
Computer Engineering department
G.H.Raisoni college of engineering and Management
Wagholi,Pune,India

Abstract—*In the real world entities have two or more repetition in database. Duplicate detection is method of detecting all cases of multiple illustration of some real world objects, example customer relationship management or data mining. A representative example customer relationship management, where a company loses money by sending multiple catalogs to the same person that would be lowering customer satisfaction. Another application is Data Mining i.e to correct input data is necessary to construct useful reports that from the basis of mechanisms. In this paper to study about the progressive duplication algorithm with the help of map reduce to detect the duplicates data and delete those duplicate records.*

Keywords*— Duplicate Detection,Data Cleaning, PSNM,Map Reduce.*

1. Introduction

Data mining depends on effective data collection and warehousing as well as computer processing. Most important property of a company is „Data" but when data change or poor data entry, data errors such as duplicate detection occurs we want to make data cleaning for duplicate detection. However, duplicate detection processes expensive due to pure size of dataset. Duplicate detection is the procedure of identifying various representations same real-world objective in a information source. The quality of duplicate detection, i.e., its effectiveness, scalability cannot be ignored because of the significant size of the database. The duplicate detection problem has two aspects: First, the multiple representations are usually not the same but contain differences, such as misspellings, changed addresses, or missing values. This makes it difficult to detect these duplicates. Second, duplicate detection is a very expensive operation, as it requires the comparison of every possible pair of duplicates using the typically complex similarity calculate. The paper proposes the Parallel Duplicate Detection with Map reduce concept. Duplicate detection is the process of identifying multiple representations of same real world entities. Today, duplicate detection methods need to process ever larger datasets in ever shorter time: maintaining the quality of a dataset becomes increasingly difficult. We present 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 our progressive algorithms can double the efficiency over time of traditional duplicate detection and significantly improve upon related work.

2 LITERATURE SURVEY

2.1 Pay-as-you-go Entity resolutions

In this paper they introduced pay-as-you-go ER algorithm to sorted the list of records in the dataset and then find the duplicate records.

Advantage: Entity resolution algorithm maintains quality of records. Disadvantage: Execution Process is slow.

2.2 Duplicate record detection: A survey

In this paper duplicate record detection using two technique first Distance based approach and rule based approaches for duplicate records detection.

Advantage: Data with good indexing

Disadvantage: Duplicate records still present in the dataset

2.3 On threshold for duplicate detection

In this paper specify about the similarity measure and some threshold that declares duplicity for record pairs. The similarity measure largely depends on the nature of the data and its contained errors that cause the duplicates.

Advantage: Threshold give the perfect ratio to find duplicate records.

Disadvantage: Not used the large dataset for the perfect ratio.

2.4 Different Duplicate detection method

In this paper different duplicate detection methods are used ISNM, DCS++ and PSNM method.

Advantage: PSNM this progressive method is more efficient to find duplicate records

Disadvantage: In ISNM and DCS++ in this method no windowing concept and less efficient.

3. SYSTEM DESIGN

3.1 Existing System

In the existing system they used two novel of Progressive Duplicate detection i.e PSNM& PB algorithm.PSNM find duplicate data and PB delete that duplicate data, this will be done in one by one, but they have some limitations.

Limitation on existing system:

- Slow in Processing
- Lengthy Process
- No parallelism
- Duplicates records are still present in dataset

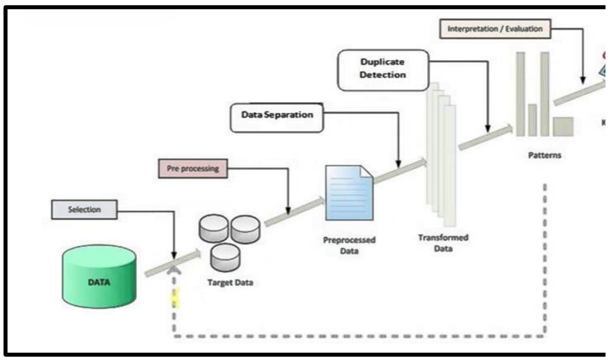


Fig No 1 Duplicate Detection

3.2 Problem statement

To Improve the performance of Parallel Progressive Duplicate detection method using Map Reduce Algorithm on Large Data Sets which are used across industries and hence increasing the efficiency and effectiveness of the process

3.3 Proposed Idea

In propose system there is two progressive duplicate detection algorithms namely progressive sorted neighborhood method (PSNM) and progressive blocking (PB) with Map reduce.

Map reduce apply on the selective input dataset (Cluster) that used for data cleaning, data pre-processing, and data separation.

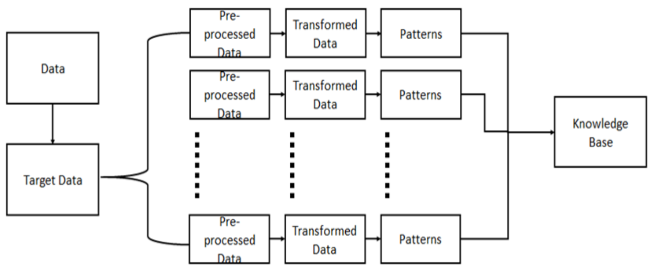


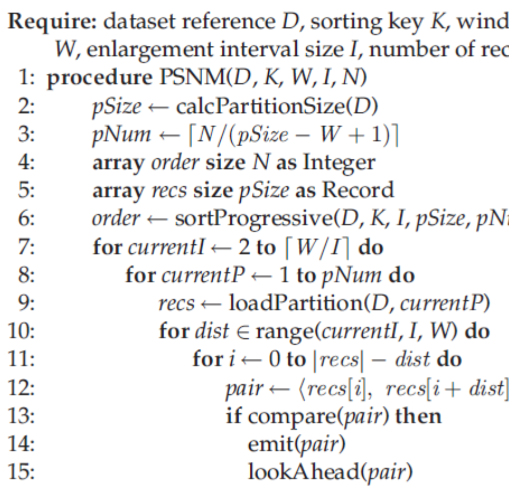
Fig.No.2 Proposed System Architecture

- It will be prepared three datasets that have been used frequently for duplicate detection:
- CORA (Scientific research paper data)
- Restaurant data from Riddle repository
- CD Dataset (audio, video, images)

4. Algorithm

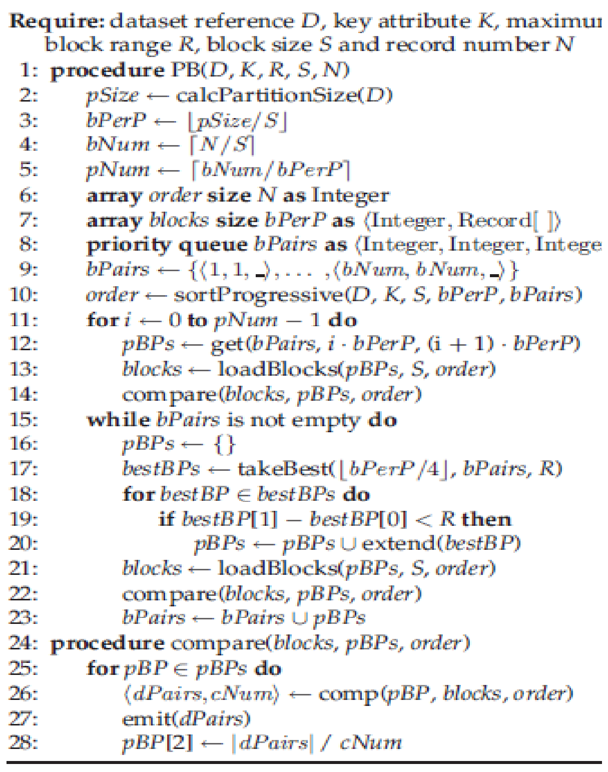
- PSNM (Progressive sorted neighborhood method)
- PB (Progressive Blocking)

1) PSNM Algorithm



```
Require: dataset reference $D$, sorting key $K$, wind
    $W$, enlargement interval size $I$, number of rec
procedure PSNM($D, K, W, I, N$)
    $pSize \leftarrow$ calcPartitionSize($D$)
    $pNum \leftarrow \lceil N/(pSize - W + 1) \rceil$
    array $order$ size $N$ as Integer
    array $recs$ size $pSize$ as Record
    $order \leftarrow$ sortProgressive($D, K, I, pSize, pN$
    for $currentI \leftarrow 2$ to $\lceil W/I \rceil$ do
        for $currentP \leftarrow 1$ to $pNum$ do
            $recs \leftarrow$ loadPartition($D, currentP$)
            for $dist \in$ range($currentI, I, W$) do
                for $i \leftarrow 0$ to $|recs| - dist$ do
                    $pair \leftarrow \langle recs[i],\ recs[i + dist]$
                    if compare($pair$) then
                        emit($pair$)
                        lookAhead($pair$)
```

2) PB Algorithm



```
Require: dataset reference $D$, key attribute $K$, maximu
    block range $R$, block size $S$ and record number $N$
procedure PB($D, K, R, S, N$)
    $pSize \leftarrow$ calcPartitionSize($D$)
    $bPerP \leftarrow \lfloor pSize/S \rfloor$
    $bNum \leftarrow \lceil N/S \rceil$
    $pNum \leftarrow \lceil bNum/bPerP \rceil$
    array $order$ size $N$ as Integer
    array $blocks$ size $bPerP$ as $\langle$Integer, Record[ ]$\rangle$
    priority queue $bPairs$ as $\langle$Integer, Integer, Intege
    $bPairs \leftarrow \{\langle 1, 1, \_\rangle, \ldots, \langle bNum, bNum, \_\rangle\}$
    $order \leftarrow$ sortProgressive($D, K, S, bPerP, bPairs$)
    for $i \leftarrow 0$ to $pNum - 1$ do
        $pBPs \leftarrow$ get($bPairs, i \cdot bPerP, (i + 1) \cdot bPerP$)
        $blocks \leftarrow$ loadBlocks($pBPs, S, order$)
        compare($blocks, pBPs, order$)
    while $bPairs$ is not empty do
        $pBPs \leftarrow \{\}$
        $bestBPs \leftarrow$ takeBest($\lfloor bPerP/4 \rfloor, bPairs, R$)
        for $bestBP \in bestBPs$ do
            if $bestBP[1] - bestBP[0] < R$ then
                $pBPs \leftarrow pBPs \cup$ extend($bestBP$)
        $blocks \leftarrow$ loadBlocks($pBPs, S, order$)
        compare($blocks, pBPs, order$)
        $bPairs \leftarrow bPairs \cup pBPs$
procedure compare($blocks, pBPs, order$)
    for $pBP \in pBPs$ do
        $\langle dPairs, cNum \rangle \leftarrow$ comp($pBP, blocks, order$)
        emit($dPairs$)
        $pBP[2] \leftarrow |dPairs|\ /\ cNum$
```

5. Advantages

- Faster Processing
- Optimized Process
- Duplicate records are removed

6. Disadvantages

- Map reduce have high latency.
- It would affect processing small datasets. However, for large datasets it is negligible

7. Result

PSNM and PB its utilization for duplicate record detection, and duplicate record deletion. On the other hand, the improvement in detection effectiveness is consistently observed in two applications. This is achieved by indexing the PSNM & PB with Map Reduce.

8. Acknowledgment

I would wish to thank to all the people who showered their kind support required for the whole analysis. The authors also want to thank the anonymous reviewers for their useful and constructive comments.

REFERENCES

[1] Thorsten Papenbrock, ArvidHeise, and Felix Naumann,‟ Progressive Duplicate Detection‟ IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 25, no. 5, 2014.

[2] S. Yan, D. Lee, M. yen Kan, and C. L. Giles, “Adaptive sorted neighborhood methods for efficient record linkage,” in International Conference on Digital Libraries (ICDL), 2007.

[3] M. A. Hernández and S. J. Stolfo, “Real-world data is dirty: Data cleansing and the merge/purge problem,” Data Mining and Knowledge Discovery, vol. 2, no. 1, 1998.

[4] X.Dong, A.Halevy, and J.Madhavan, “Reference reconciliation in complexinformation spaces,” in Proceedings of the International Conference on Management of Data (SIGMOD), 2005.

[5] S.E.Whang, D.Marmaros, and H.Garcia-Molina, “Pay-as-you-go entity resolution” IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 25, no. 5, 2012.

[6] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, “Duplicat record detection: A survey,” IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 19, no. 1, 2007.

[7] U.Draisbach, F.Naumann, S.Szott, and O. Wonneberg, “Adaptive windows for duplicate detection,” in Proceedings of the International Conference on Data Engineering (ICDE), 2012.

[8] U.Draisbach and F. Naumann, “A generalization of blocking and windowing algorithms for duplicate detection.” in International Conference on Data and Knowledge Engineering (ICDKE), 2011.

[9] L. Kolb, A. Thor, and E. Rahm, “Parallel sorted neighbourhoodblockingwithmapreduce,” in Proceedings of the Conference Datenbank system in Büro, Technik und Wissenschaft(BTW