ISSN: 2456-8686, Volume 3, Issue 2, 2019:41-50
DOI : http://doi.org/10.26524/cm53

Journal of Computational Mathematica

Open Access

Analysis of Progressive Duplicate Data Detection

P.Veeramuthu[1*]
[1]Department of Computer Science, Besant Theosophical College, Madanapalle, Andhra Pradesh, India.

Abstract

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.

Key words: Sorted Neighborhood Method (SNM), progressive blocking (PB), progressive sorted neighborhood method (PSNM) and Data Mining.

1. Introduction

Generally, data mining is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

2. Data Mining

Data mining is a powerful new technology with great potential to help companies focus on the most important information in the data they have collected about the behavior of their customers and potential customers [3]. Data mining involves the use of sophisticated data analysis tools to discover previously unknown,

[1*]er.veera86@gmail.com

ISSN: 2456-8686, Volume 3, Issue 2, 2019:41-50
DOI : http://doi.org/10.26524/cm53

valid patterns and relationships in large data set. These tools can include statistical models, mathematical algorithm and machine learning methods. It discovers information within the data that queries and reports can't effectively reveal. Decision tree is a decision support tool in the field of data mining that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility [4]. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal. Another use of decision trees is as a descriptive means for calculating conditional probabilities. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume. Instances are classified starting at the root node and sorted based on their feature values.

2.1 Different Levels of Analysis

- **Artificial neural networks:** Non-linear predictive models that learn through training and resemble biological neural networks in structure.
- **Genetic algorithms:** Optimization techniques that use process such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.
- **Decision trees:** Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome. CART segments a dataset by creating 2-way splits while CHAID segments using chi square tests to create multi-way splits. CART typically requires less data preparation than CHAID. "
- **Nearest neighbor method:** A technique that classifies each record in a data set based on a combination of the classes of the k record(s) most similar to it in a historical dataset (where k=1). Sometimes called the k-nearest neighbor technique. " Rule induction: The extraction of useful if-then rules from data based on statistical significance.
- **Data visualization:** The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.

[1*]er.veera86@gmail.com

ISSN: 2456-8686, Volume 3, Issue 2, 2019:41-50
DOI : http://doi.org/10.26524/cm53

3. Existing System

Much research on duplicate detection, also known as entity resolution and by many other names focuses on pair selection algorithms that try to maximize recall on the one hand and efficiency on the other hand. The most prominent algorithms in this area are Blocking and the sorted neighborhood method (SNM).

U.Draisbach and F.Naumann in [9, 15] proposed two major methods called blocking and windowing used to reduce the comparisons are studied in this paper. Sorted Blocks that denotes a generalization of these two methods are also analyzed here. Blocking divides the records to disjoint subsets and windowing slides a window on the sorted records and then comparison is made between records within the window. The sorted Blocks have advantages like the variable size of partition size instead of the size of the window.

M. A. Hernandez and S. J. Stolfo, [5, 11] the problem of merging multiple databases of information about common entities is frequently encountered in KDD and decision support applications in large commercial and government organizations. The problem we study is often called the Merge/Purge problem and is difficult to solve both in scale and accuracy. Large repositories of data typically have numerous duplicate information entries about the same entities that are difficult to cull together without an intelligent "equational theory" that identifies equivalent items by a complex, domain-dependent matching process[6]. We have developed a system for accomplishing this Data Cleansing task and demonstrate its use for cleansing lists of names of potential customers in a direct marketing-type application. Our results for statistically generated data are shown to be accurate and effective when processing the data multiple times using different keys for sorting on each successive pass. Combing results of individual passes using transitive closure over the independent results, produces far more accurate results at lower cost. The system provides a rule programming module that is easy to program and quite good at finding duplicates especially in an environment with massive amounts of data[16][17]. This paper details improvements in our system, and reports on the successful implementation for a real-world database that conclusively validates our results previously achieved for statistically generated data.

Xiao et al. [13] proposed a top-k similarity join that uses a special index structure to estimate promising comparison candidates. This approach progressively resolves duplicates and also eases the parameterization problem.

S. E. Whang et al. [1, 2] stated a survey on the active methods and non identical duplicate entries present in the records of the database records are

[1]*er.veera86@gmail.com

ISSN: 2456-8686, Volume 3, Issue 2, 2019:41-50
DOI : http://doi.org/10.26524/cm53

all investigated in this paper. It works for both the duplicate record detection approaches. 1) Distance Based technique that measures the distance among the individual fields, by using distance metrics of all the fields and later computing the distance among the records. 2) Rule based technique that uses rules for defining that if two records are same or different. Rule based technique is measured using distance based methods in which the distances are 0 or 1. The techniques for duplicate record detection are very essential to improve the extracted data quality."

A.Thor et al. [21] proposed a theory of deduplication which is also known as Entity Resolution which is used for determining entities associated to similar object of the real world. It is very important for data integration and data quality. Map Reduce is used for SN blocking execution. Both blocking methods and methods of parallel processing are used in the implementation of entity resolution of huge datasets.

P. Christen [19, 18], Record linkage is the process of matching records from several databases that refer to the same entities. When applied on a single database, this process is known as deduplication. Increasingly, matched data are becoming important in many application areas, because they can contain information that is not available otherwise, or that is too costly to acquire. Removing duplicate records in a single database is a crucial step in the data cleaning process [20], because duplicates can severely influence the outcomes of any subsequent data processing or data mining. With the increasing size of today's databases, the complexity of the matching process becomes one of the major challenges for record linkage and deduplication. In recent years, various indexing techniques have been developed for record linkage and deduplication. They are aimed at reducing the number of record pairs to be compared in the matching process by removing obvious non-matching pairs, while at the same time maintaining high matching quality. This paper presents a survey of 12 variations of 6 indexing techniques. Their complexity is analyzed, and their performance and scalability is evaluated within an experimental framework using both synthetic and real data sets. No such detailed survey has so far been published.

O. Hassanzadeh, F. Chiang, H. C. Lee, and R. J. Miller [8], The Author presence of duplicate records is a major data quality concern in large databases. To detect duplicates, entity resolution also known as duplication detection or record linkage is used as a part of the data cleaning process to identify records that potentially refer to the same real-world entity. We present the Stringer system that provides an evaluation framework for understanding what barriers remain towards the goal of truly scalable and general purpose duplication detection algorithms [19]. In

[1]*er.veera86@gmail.com

ISSN: 2456-8686, Volume 3, Issue 2, 2019:41-50
DOI : http://doi.org/10.26524/cm53

this paper, we use Stringer to evaluate the quality of the clusters (groups of potential duplicates) obtained from several unconstrained clustering algorithms used in concert with approximate join techniques [7]. Our work is motivated by the recent significant advancements that have made approximate join algorithms highly scalable. Our extensive evaluation reveals that some clustering algorithms that have never been considered for duplicate detection, perform extremely well in terms of both accuracy and scalability.

3.1 Disadvantages of Existing System

1. A user has only limited, maybe unknown time for data cleansing and wants to make best possible use of it. Then, simply start the algorithm and terminate it when needed. The result size will be maximized.
2. A user has little knowledge about the given data but still needs to configure the cleansing process.
3. A user needs to do the cleaning interactively to, for instance, find good sorting keys by trial and error. Then, run the progressive algorithm repeatedly; each run quickly reports possibly large results.
4. All presented hints produce static orders for the comparisons and miss the opportunity to dynamically adjust the comparison order at runtime based on intermediate results.

4. Proposed System

- In this work, however, we focus on progressive algorithms, which try to report most matches early on, while possibly slightly increasing their overall runtime. To achieve this, they need to estimate the similarity of all comparison candidates in order to compare most promising record pairs first.
- We propose two novel, progressive duplicate detection algorithms namely progressive sorted neighborhood method (PSNM), which performs best on small and almost clean datasets, and progressive blocking (PB), which performs best on large and very dirty datasets. Both enhance the efficiency of duplicate detection even on very large datasets.
- We propose two dynamic progressive duplicate detection algorithms, PSNM and PB, which expose different strengths and outperform current approaches.
- We introduce a concurrent progressive approach for the multi-pass method and adapt an incremental transitive closure algorithm that together forms the first complete progressive duplicate detection workflow.
- We define a novel quality measure for progressive duplicate detection to objectively rank the performance of different approaches.

[1]*er.veera86@gmail.com

ISSN: 2456-8686, Volume 3, Issue 2, 2019:41-50
DOI : http://doi.org/10.26524/cm53

- We exhaustively evaluate on several real-world datasets testing our own and previous algorithms

4.1 Algorithm of Duplicate Detection

Step:1 Collect the Raw data

Step:2 Pre-processing the Data

Step:3 Separation of data

Step:4 Cluster Block Size

Step:5 Duplicate Detection

Step:6 Evaluation Result

4.2 Algorithm Description

- **Data set Collection:** To collect and/or retrieve data about activities, results, context and other factors. It is important to consider the type of information it want to gather from your participants and the ways you will analyze that information. The data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable.
- **Preprocessing Method:** Data Pre processing or Data cleaning, Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data. And also used to removing the unwanted data. Commonly used as a preliminary data mining practice, data pre processing transforms the data into a format that will be more easily and effectively processed for the purpose of the user.
- **Data Separation:** After completed the pre processing, the data separation to be performed. The blocking algorithms assign each record to a fixed group of similar records and then compare all pairs of records within these groups. Each block within the block comparison matrix represents the comparisons of all records in one block with all records in another block, the equidistant blocking, all blocks have the same size.
- **Duplicate Detection:** The duplicate detection rules set by the administrator, the system alerts the user about potential duplicates when the user tries to create new records or update existing records. 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.

[1*]er.veera86@gmail.com

ISSN: 2456-8686, Volume 3, Issue 2, 2019:41-50
DOI : http://doi.org/10.26524/cm53

- **Quality Measures:** The quality of these systems is, hence, measured using a cost-benefit calculation. Especially for traditional duplicate detection processes, it is difficult to meet a budget limitation, because their runtime is hard to predict. By delivering as many duplicates as possible in a given amount of time, progressive processes optimize the cost-benefit ratio. It is brought about by strict and consistent commitment to certain standards that achieve uniformity of a product in order to satisfy specific customer or user requirements.

4.3 Advantages of Proposed System

1. Improved early quality
2. Same eventual quality
3. Our algorithms PSNM and PB dynamically adjust their behavior by automatically choosing optimal parameters, e.g., window sizes, block sizes, and sorting keys, rendering their manual specification superfluous. In this way, we significantly ease the parameterization complexity for duplicate detection in general and contribute to the development of more user interactive applications

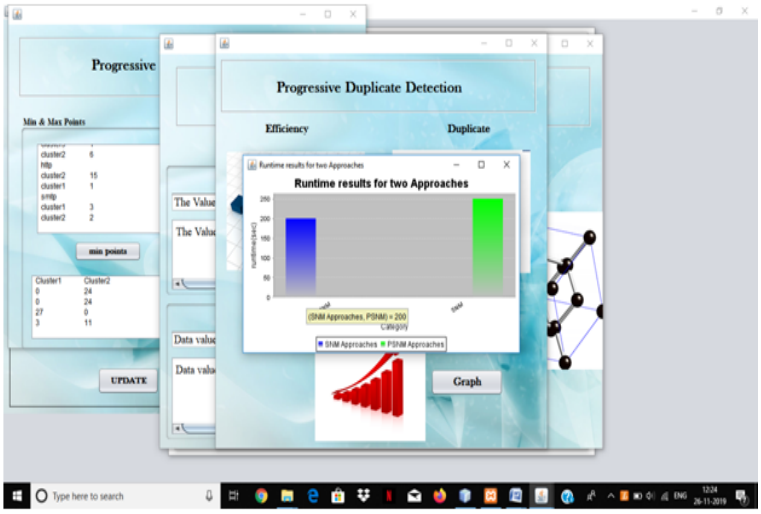

Figure 1: Runtime Results for Two Approaches

5. Result and Discussion

The work 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. To determine the performance

[1]*er.veera86@gmail.com

ISSN: 2456-8686, Volume 3, Issue 2, 2019:41-50
DOI : http://doi.org/10.26524/cm53

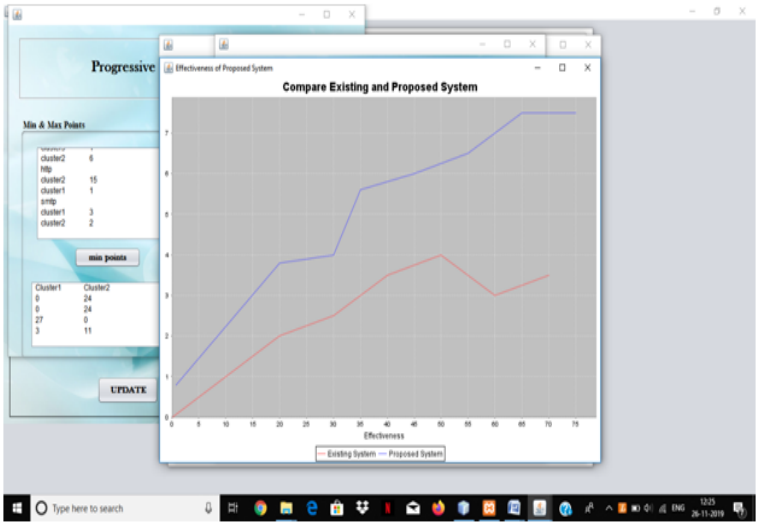

Figure 2: Comparison of Existing and Proposed System

gain of our algorithms, we proposed a novel quality measure for progressiveness that integrates seamlessly with existing measures. Using this measure, experiments showed that our approaches outperform the traditional SNM by up to 100 percent and related work by up to 30 percent for the construction of a fully progressive duplicate detection workflow, we adapted a progressive sorting method, Magpie, a progressive multi-pass execution model, Attribute Concurrency, and an incremental transitive closure algorithm.

6. Conclusion

The adaptations AC-PSNM and AC-PB use 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. In future work, we want to combine our progressive approaches with scalable approaches for duplicate detection to deliver results even faster. In particular, Kolb et al. introduced a two phase parallel SNM , which executes traditional SNM on balanced, overlapping partitions. Here, we can instead use our PSNM to progressively find duplicates in parallel.

References

[1] Whang E, Marmaros D, Garcia-Molina H, Pay-as-you-go entity resolution, IEEE Trans. Knowl. Data Eng., 25(5), 2012, 1111-1124.

[1]*er.veera86@gmail.com

ISSN: 2456-8686, Volume 3, Issue 2, 2019:41-50
DOI : http://doi.org/10.26524/cm53

[2] Elmagarmid AK, Ipeirotis PG, Verykios VS, Duplicate record detection: A survey, IEEE Trans. Knowl. Data Eng., 19(1), 2007, 01-16.

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

[4] Newcombe HB, Kennedy JM, Record linkage: Making maximum use of the discriminating power of identifying information, Commun. ACM, 5(11), 1962, 563-566.

[5] Hernandez MA, Stolfo SJ, Real-world data is dirty: Data cleansing and the merge/purge problem, Data Mining Knowl. Discovery, 2(1), 1998, 09-37.

[6] Dong X, Halevy A, Madhavan J, Reference reconciliation in complex information spaces, in Proc. Int. Conf. Manage. Data, 85-96, (2005).

[7] Hassanzadeh O, Chiang F, Lee HC, Miller RJ, Framework for evaluating clustering algorithms in duplicate detection, Proc. Very Large Databases Endowment, 2, 2009, 1282- 1293.

[8] Hassanzadeh O, Miller RJ, Creating probabilistic databases from duplicated data, VLDB J., 18(5), 2009, 1141-1166.

[9] Draisbach U, Naumann F, Szott S, Wonneberg O, Adaptive windows for duplicate detection, in Proc. IEEE 28th Int. Conf. Data Eng., 1073-1083, (2012).

[10] Yan S, Lee D, Kan MY, Giles LC, Adaptive sorted neighborhood methods for efficient record linkage, in Proc. 7th ACM/IEEE Joint Int. Conf. Digit. Libraries, 185-194, (2007).

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

[12] Jeffery SR, Franklin MJ, Halevy AY, Pay-as-you-go user feedback for dataspace systems, in Proc. Int. Conf. Manage. Data, 847-860, (2008).

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

[14] Indyk P, A small approximately min-wise independent family of hash functions, in Proc. 10th Annu. ACM-SIAM Symp. Discrete Algorithms, 454-456, (1999). Fig. 10. Duplicates found in the plista-dataset.

[1*]er.veera86@gmail.com

ISSN: 2456-8686, Volume 3, Issue 2, 2019:41-50
DOI : http://doi.org/10.26524/cm53

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

[16] Warren HS, Jr., A modification of Warshall's algorithm for the transitive closure of binary relations, Commun. ACM, 18(4), 1975, 218-220.

[17] Wallace M, Kollias S, Computationally efficient incremental transitive closure of sparse fuzzy binary relations, in Proc. IEEE Int. Conf. Fuzzy Syst., 1561-1565, (2004).

[18] Damerau FJ, A technique for computer detection and correction of spelling errors, Commun. ACM, 7(3), 1964, 171-176.

[19] Christen P, A survey of indexing techniques for scalable record linkage and deduplication, IEEE Trans. Knowl. Data Eng., 24(9), 2012, 1537-1555.

[20] Kille B, Hopfgartner F, Brodt T, Heintz T, The Plista dataset, in Proc. Int. Workshop Challenge News Recommender Syst., 16-23, (2013).

[21] Kolb L, Thor A, Rahm E, Parallel sorted neighborhood blocking with MapReduce, in Proc. Conf. Datenbanksysteme in B'uro, Technik und Wissenschaft, (2011).

[1*]er.veera86@gmail.com