Analysis and Evaluation of Data - Duplication Reduction Using Graph Mining

Prof. R.V. Tatiya, Pushkar Mundlik, Shivam Prakash, Shubhansh Singh, Gyan Prakash

Department of Computer Engineering, Sinhgad Institute of Technology and Science, Narhe, Pune, Maharashtra, India.

Abstract: To carry out operations many of the industries depend on the accuracy of databases. Independent expansion of substructure leads to the generation of duplicates. Eliminating these duplicates not only incurs generation and storage cost but also additional computation for its elimination. The primary aim of this project is to design techniques to reduce generating duplicate substructures using graph mining concept. Theoretical correctness of each method as well as its examination about the applicability of their combination for improvements in duplicate reduction is set up. Finally, the paper discusses the effects of the limitations with respect to the partitioning schemes used in graph mining. This survey demonstrates significant benefits of these constraints in terms of storage, computation, and communication cost across graphs with varied characteristics.

Keywords: Constraint-Based Heuristics, Duplicate Reduction, Partitioning of Graphs, Data Mining.

2. INTRODUCTION

The system can show the graphical view of the data to user for easy understanding. This system can easily analyze and study the existing data via graph for business purpose. The duplicates once identified need to be removed to ensure correctness, incurring additional computation cost. The system will generate a graph from the existing data and that graph will be according to day, month and year wise. That graph will be of different types like Bar graph, Line graph, Pie chart, etc. At the core of graph mining lies independent expansion of substructures where a substructure independently grows into a number of larger substructures in each iteration. Such an independent expansion, invariably, leads to the generation of duplicates. In the presence of graph partitions, duplicates are generated both within and across partitions. Eliminating these duplicates not only incurs generation and storage cost but also additional computation for its elimination.

This system will use different E-commerce application and product. The proposed method uses Map-Reduce model in cloud system to parallel and builds balanced partitions of a graph database over a set of machines to save time and memory.

This paper introduces three constraint-based optimization techniques, each significantly improving the overall mining cost by reducing the number of duplicates generated. These alternatives provide flexibility to choose the right technique based on graph properties. The paper establishes theoretical correctness of each technique as well as its analysis with respect to graph characteristics such as degree, number of unique labels, and label distribution. The system also investigates the applicability of their combination for improvements in duplicate reduction.

3. RELATED WORK

Most of the work on graphs can be categorized as addressing two types of problems. The first one is to find all occurrences of a given substructure in a large graph and count them[1]. This can be used to find identical or even similar patterns in a large graph to a known pattern[2]. The second class of problem is to find the best substructure that transforms a given graph to satisfy a metric[4]. Finding a substructure that minimizes the minimum description length (MDL) or occurs above a certain frequency is important as that substructure demonstrates some interesting property of that graph (e.g., maximal concept in a graph). For both of these problems, it is important to generate substructures of increasing sizes and analyze them in various ways[3]. Different approaches have been proposed for these applications.

Main memory approaches required loading the entire graph into main memory. Main memory graph mining algorithms have used either apriori-based approach like AGM [6], FSG [12] or pattern growth as in gSpan [5], FFSM [7] and GASTON [9]. Disk-based graph mining techniques followed suit by storing the graph partly in memory and the rest on the disk. An alternative approach was to use database-oriented techniques by mapping these graph mining algorithms to SQL [13] and storing data in a DBMS. Although scalability was achieved for graphs with over a million nodes and edges, substructure expansion generated duplicate substructures, the removal[13] of which required sorting columns (in row-based DBMSs) making it computationally expensive.

Algorithm: -

1. Map Reduce Algorithm: -

Map Reduce is a Distributed Data Processing Algorithm. Algorithm is mainly inspired by Functional Programming model. Map Reduce algorithm is mainly useful to process huge amount of data in parallel, reliable and efficient way in cluster environments’ hope, everyone is familiar with “Divide and Conquer” algorithm. It uses Divide and Conquer technique to process large amount of data. It divides input task into smaller and manageable sub-tasks to execute them in-parallel.
**Divide and Conquer Algorithm:**
Divide and Conquer Algorithm is an algorithm based on QR algorithms, each eigenvalue of its sub-block matrix is calculated by QR method, and when the eigenvalue is glued, we use parabolic interpolation method to get the mother matrix eigenvalues, so it is stable, and there is good convergence, in particular the flexibility of this algorithm is large, and particularly suitable for parallel computing. But it cannot be guaranteed that we can get all the matrix eigenvalue. In addition, when given the eigenvalue approximation, it cannot also give an approximation of the error limit, which is the bureau of the method; and it is the problem to be solved when we later find a better way.

**Advantages:**
- MapReduce programming allows execution in parallel. Parallel processing results in running entire programs in less time.
- The concept of this paper is not restricted to a single domain, it can be used in any field such as Business, Sales, etc.

**Disadvantages:**
- Problem decomposition may be very complex and thus not really suitable to divide and conquer.
- Recursive nature of the solution may end up duplicating sub-problems, dynamic/memorized solutions may be better in some of these cases, like Fibonacci.
- Recursion into small/tiny base cases may lead to huge recursive stacks, and efficiency can be lost by not applying solutions earlier for larger base cases.

**4. CONCLUSION**
We are going to generate a graph from the existing data and that graph will be according to attribute wise. That graph will be of different types like Bar graph, Line graph, Pie chart, etc.

The graph will be of different domains like the Banking sector, Medical field, E-commerce Product. In our system, we are reducing the duplication in the existing graph by Graph mining technique and finally generate a new graph by using Map-reduce Algorithm.

**REFERENCES**


