

# Duplication Reduction Using Map Reduce Technique

Mayuri Jadhav, Komal Bidkar, Punam Jiwane, Prachi Ghorpade Prof. K.S.Warke

**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. Our primary aim is to design techniques to reduce generating duplicate substructures using graph mining concept. We set up theoretical correctness of each method as well as its analysis with respect to graph characteristics such as degree, number of unique labels, and label distribution. We also examine the applicability of their combination for improvements in duplicate reduction. Finally, we discuss the effects of the limitations with respect to the partitioning schemes used in graph mining. Our experiments demonstrate 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.

**Introduction:** The system can show the graphical view of the data to user for easy understanding. We 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. We are going to 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 (for correctness) not only incurs generation and storage cost but also additional computation for its elimination. Our primary aim is to design techniques to reduce generating duplicate substructures as we show that they cannot be eliminated. 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. We establish 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. We also investigate the applicability of their combination for improvements in duplicate reduction.

**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 (or in a database of graphs) and count them. This can be used to find identical or even similar patterns in a large graph to a known pattern. The second class of problem is to find the best substructure that transforms a given graph (or a forest) to satisfy a metric. 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. Different approaches have been proposed for these applications.

### Motivation:

Our Motivation is this System we used different E-commerce application and product. We used Stock market domain and Bitcoin related Concept. The proposed method uses MapReduce model in cloud system to parallel and builds balanced partitions of a graph database over a set of machines to save time and memory.

### System Architecture:

In our system, the user will first register and login into the system. After login user will upload the data on the system. There are two types of data in the first structured and another one is unstructured data.

### Fig: Proposed System

Our system will understand both formats of the data. Then the data will be converted into .arff (Attribute-Relation File Format) file if needed. We are going to draw multiple graphs of data given by the user. Now we are removing the duplicate data by using the Mapper-Reducer concept in Hadoop. On this reduced data, we are creating a final graph without any duplication in data. Information storage is the part of the accounting system that keeps data accessible to the information processors. In other words, an accounting system's information storage unit is either a hard drive or server that usually contains a database. After the input devices enter data into an accounting system, the information processors take the raw data and convert it into a usable form. This information is then stored, often in the form of a database, on the information storage component of the accounting system. A database is just a system that organizes data. You have a set of data, perhaps some order transactions, and the database organizes those transactions based on settings you define. In that system architecture we are taking user information like Name, email ID, etc. Which store the information about user in the database when user registered the account. This information is also accessible to vendor also, this information is securely stored in database. Our primary aim was to predict the sales of an item given the Best Seller Rank on E-commerce site. Predicting the sales helps me in other use cases like suggesting sellers the best products to sell. Our final aim is to provide data insights about any product: How much it will sell as well as when, where and how. While e-commerce has grown quickly in recent years, more and more people are used to utilize this popular channel to purchase products and services on the Internet. Therefore, it becomes very important for shopping sites to predict precisely which items their customers would buy so as to increase sales or improve customer satisfaction. Traditional algorithms such as Collaborative Filtering, has been very popular in predicting users' preferences in shopping, or music recommendation areas, but they face the problem that rating data is very sparse or even not available in shopping domain. Compared to the small amount of ratings in e-commerce shopping sites, the quantity of user clicking data is abundant and also contains sufficient information about users' purchase preferences. Therefore, in this project we propose a prediction method based on probability statistics making use of user clicking behavior data. Suppose you have a data-files which are having duplicate records i.e. a line of a file is occurring more than one times. You want those records which are present in file multiple times. So the requirement is how to find the duplicate record using Map Reduce. HDFS: Here it is used to store the input data which will get pass to the Map Reduce as an input and also store the MapReduce output. Map Reduce: Here it is used to process the data/file available in HDFS and store the output to the HDFS. We are taken a sample of records & copied and pasted few records in the same sample file for the duplicate. In the first step, we will set up the data for the map reduce job. The data file should available at HDFS path, In order to write map reduce program, create a maven project. We will add the dependency for the required package. Now, we have to write a reducer class which will take input from mapper class. We will filter the duplicate record in this class. After

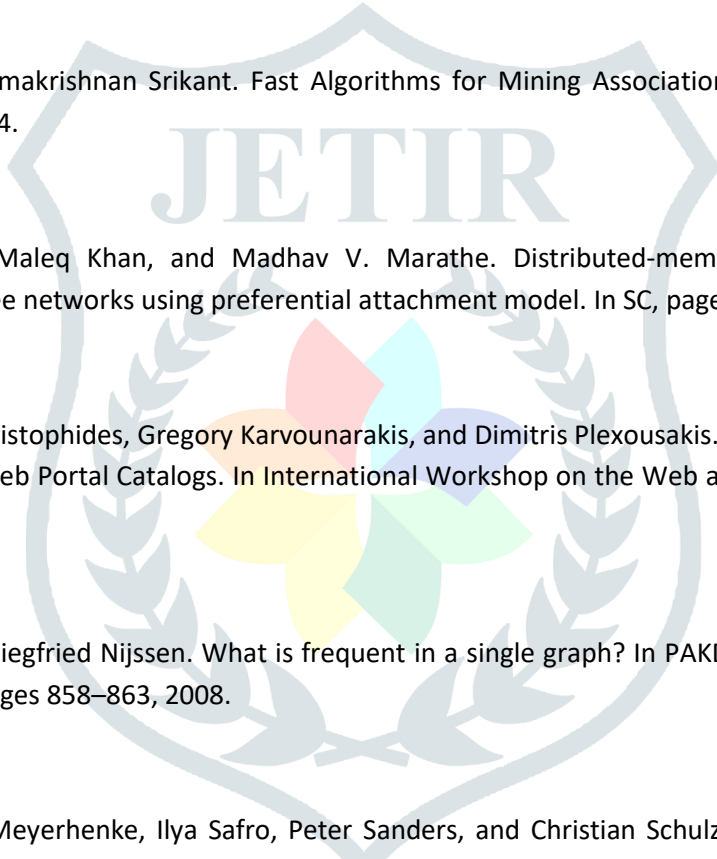
that we will write a driver class in order to execute the mapper and reducer. At last, we need to validate the output of our map reduce job. We have seen how to find the duplicate records present in data files. An input to a MapReduce job is divided into fixed-size pieces called input splits. Input split is a chunk of the input that is consumed by a single map. Mapping is the very first phase in the execution of map-reduce program. In this phase data in each split is passed to a mapping function to produce output values. In our example, a job of mapping phase is to count a number of occurrences of each word from input splits (more details about input-split is given below) and prepare a list in the form of <word, frequency>. In shuffling it consumes the output of Mapping phase. Its task is to consolidate the relevant records from Mapping phase output. In our example, the same words are clubbed together along with their respective frequency. In reducing, output values from the Shuffling phase are aggregated. This phase combines values from Shuffling phase and returns a single output value. In short, this phase summarizes the complete dataset. Graphic Representation is a method to show and represent values, increases, decreases, comparisons to either make predictions or show a report of how certain situation was yesterday and how it is today. It is the visual display of data through charts and graphs. These types of graphic representations are used in Algebraic Equations, Stocks values in the market, Whether tracking, financial analysis, companies projects. The famous kaizen system. These graphics representations are used by business analyst, traders and investors, mathematics professors, scientists, business and financial advisors, marketers, informers, architects, army or navy officers. There are four primary types of charts used by investors and traders depending on the type of information they're seeking and their desired goals. These chart types include line charts, bar charts, candlestick charts, and point and figure charts.

There are four primary types of charts used by investors and traders depending on the type of information they're seeking and their desired goals. These chart types include line charts, bar charts, candlestick charts, and point and figure charts. In the following sections, we will focus on the S&P 500 over the same period to illustrate the differences between the charts when the underlying data set is the same. Line Charts Line charts are the most basic type of chart because it represents only the closing prices over a set period. The line is formed by connecting the closing prices for each period over the timeframe. While this type of chart doesn't provide much insight into intraday price movements, many investors consider the closing price to be more important than the open, high, or low price within a given period. These charts also make it easier to spot trends since there's less „noise“ happening compared to other chart types. Bar Charts Bar charts expand upon the line chart by adding the open, high, low, and close – or the daily price range, in other words – to the mix. The chart is made up of a series of vertical lines that represent the price range for a given period with a horizontal dash on each side that represents the open and closing prices. Pie charts A Pie Chart is a type of graph that displays data in a circular graph. The pieces of the graph are proportional to the fraction of the whole in each category. In other words, each slice of the pie is relative to the size of that category in the group as a whole. The entire “pie” represents 100 percent of a whole, while the pie “slices” represent portions of the whole. This graphs is also viewable to vendor also for analysis of graph from previous dataset. A user review is a review conducted by a computer user and published to a review site following product testing or the evaluation of a service. User reviews are commonly provided by consumers who volunteer to write the review, rather than professionals who are paid to evaluate the product or service. User reviews might be compared to professional nonprofit reviews from a consumer organization, or to promotional reviews from an advertiser or company marketing a product. Various systems have been proposed to evaluate the quality of user reviews so that consumers can access the best ones, avoid lower quality ones, and prevent mixing of honestly provided reviews with less honest reviews from advertisers or people with an agenda other than non-partial evaluation. Consumers perceive user reviews using good grammar and persuasive writing style to be of higher quality than those written in other ways Here is a nice bar chart that clearly shows how multiple units does compared to each other based on same criteria. Since this chart can display positive and negative development very good, I will call it positive negative bar chart using different colors in our charts to call out facts about our data is a very good way to instantly tell a story. Your audiences' eyes can instantaneously split the data can focus on the results you want to direct them to first. All this can be done without saying a single word! .Our goal as analysts is to

tell a story with our data and it is a well-known fact that data can be consumed much faster through visualization than through text. That is why it is vital that we put serious thought into how we present our data to the leaders we support.

Conclusion: In our system we are identifying the effect of duplicates on the performance of graph mining algorithms. Based on that observation, it proposes Mapper-Reducer to reduce the number of duplicates generated to significantly improve the performance of these algorithms. Further, we establish their accuracy as well as their performance analysis for a number of graph characteristics. Based on these Graphs, we show that it is possible to choose the best heuristic whether we have additional information about the graphs or not.

#### References:

- 
- [1] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules. In Very Large Data Bases, pages 487–499, 1994.
- [2] Md. Maksudul Alam, Maleq Khan, and Madhav V. Marathe. Distributed-memory parallel algorithms for generating massive scale-free networks using preferential attachment model. In SC, page 91, 2013.
- [3] Sofia Alexaki, Vassilis Christophides, Gregory Karvounarakis, and Dimitris Plexousakis. On Storing Voluminous RDF Descriptions: The Case of Web Portal Catalogs. In International Workshop on the Web and Databases, pages 43–48, 2001.
- [4] Björn Bringmann and Siegfried Nijssen. What is frequent in a single graph? In PAKDD 2008, Osaka, Japan, May 20-23, 2008 Proceedings, pages 858–863, 2008.
- [5] Aydın Buluc., Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. Recent advances in graph partitioning. In Algorithm Engineering, pages 117–158. Springer, 2016.
- [6] Horst Bunke and Kim Shearer. A graph distance metric based on the maximal common subgraph. Pattern Recognition Letters, 19:255–259, 1998.
- [7] Deepayan Chakrabarti, Yiping Zhan, and Christos Faloutsos. R-MAT: A recursive model for graph mining. In SIAM, Florida, USA, April 22-24, 2004, pages 442–446, 2004.
- [8] Sharma Chakravarthy and Subhesh Pradhan. DB FSG: An SQL-Based Approach for Frequent Subgraph Mining. In

- [9] Soumyava Das. Divide and Conquer Approach to Scalable Substructure Discovery: Partitioning Schemes, Algorithms, Optimization and Performance Analysis using Map/Reduce Paradigm. PhD thesis, The University of Texas at Arlington, May 2017.
- [10] Soumyava Das and Sharma Chakravarthy. Challenges and approaches for large graph analysis using map/reduce paradigm. In BDA, pages 116–132, 2013 [11] Soumyava Das and Sharma Chakravarthy. Partition and conquer: Map/reduce way of substructure discovery. In DaWaK 2015, Valencia, Spain, September 1-4, 2015, pages 365–378, 2015.
- [12] Soumyava Das, Ankur Goyal, and Sharma Chakravarthy. Plan before you execute: A cost-based query optimizer for attributed graph databases. In DaWaK 2016, Porto, Portugal, September 6-8, 2016, pages 314–328, 2016.
- [13] Mukund Deshpande, Michihiro Kuramochi, and George Karypis. Frequent Sub-Structure-Based Approaches for Classifying Chemical Compounds. In IEEE International Conference on Data Mining, pages 35–42, 2003.
- [14] M. Elseidy, E. Abdelhamid, S. Skiadopoulos, and P. Kalnis. GRAMI: frequent subgraph and pattern mining in a single large graph. PVLDB, 7(7):517–528, 2014.
- [15] Giuseppe Di Fatta and Michael R. Berthold. Dynamic load balancing for the distributed mining of molecular structures. volume 17, pages 773–785, 2006.
- [16] Steven Hill, Bismita Srichandan, and Rajshekhar Sunderraman. An iterative mapreduce approach to frequent subgraph mining in biological datasets. In Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine, BCB '12, pages 661–666, 2012.
- [17] Lawrence B. Holder, Diane J. Cook, and Surnjani Djoko. Substructure Discovery in the SUBDUE System. In Knowledge Discovery and Data Mining, pages 169–180, 1994.