



Social Mining to Progress the Computational Efficiency using Mapreduce

By Sudhir Tirumalasetty, Dr. Sreenivasareddy Edara & Arunaj Jadda

Vasireddy Venkatadri Institute of Technology, India

Abstract- Graphs are widely used in large scale social network analysis. Graph mining increasingly important in modelling complicated structures such as circuits, images, web, biological networks and social networks. The major problems occur in this graph mining are computational efficiency (CE) and frequent subgraph mining (FSM). Computational Efficiency describes the extent to which the time, effort or efficiency which use computing technology in information processing. Frequent Sub graph Mining is the mechanism of candidate generation without duplicates. FSM faces the problem on counting the instances of the patterns in the dataset and counting of instances for graphs. The main objective of this project is to address CE and FSM problems. The paper cited in the reference proposes an algorithm called Mirage algorithm to solve queries using subgraph mining. The proposed work focuses on enhancing An Iterative Map Reduce based Frequent Subgraph Mining Algorithm (MIRAGE) to consider optimum computational efficiency. The test data to be considered for this mining algorithm can be from any domains such as medical, text and social data's (twitter). The major contributions are: an iterative MapReduce based frequent sub graph mining algorithm called MIRAGE used to address the frequent sub graph mining problem.

GJCST-C Classification : D.2.12 I.3.3 H.2.8



Strictly as per the compliance and regulations of:



Social Mining to Progress the Computational Efficiency using Mapreduce

Sudhir Tirumalasetty^α, Dr. Sreenivasareddy Edara^σ & Arunaj Jadda^ρ

Abstract- Graphs are widely used in large scale social network analysis. Graph mining increasingly important in modelling complicated structures such as circuits, images, web, biological networks and social networks. The major problems occur in this graph mining are computational efficiency (CE) and frequent subgraph mining (FSM). Computational Efficiency describes the extent to which the time, effort or efficiency which use computing technology in information processing. Frequent Sub graph Mining is the mechanism of candidate generation without duplicates. FSM faces the problem on counting the instances of the patterns in the dataset and counting of instances for graphs. The main objective of this project is to address CE and FSM problems. The paper cited in the reference proposes an algorithm called Mirage algorithm to solve queries using subgraph mining. The proposed work focuses on enhancing An Iterative Map Reduce based Frequent Subgraph Mining Algorithm (MIRAGE) to consider optimum computational efficiency. The test data to be considered for this mining algorithm can be from any domains such as medical, text and social data's (twitter). The major contributions are: an iterative MapReduce based frequent sub graph mining algorithm called MIRAGE used to address the frequent sub graph mining problem. Computational Efficiency will be increased through MIRAGE algorithm over Matrix Vector Multiplication. Performance of the MIRAGE will be demonstrated through different synthetic as well as real world datasets. The main aim is to improve the existing algorithm to enhance Computational Efficiency.

I. INTRODUCTION

a) Overview

Data mining is the computational process of discovering patterns in large datasets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. Data Mining is an analytic process designed to explore data (usually large amounts of data typically business or market related are known as "big data") in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. The process of data mining consists of three stages: the initial exploration, model building or pattern identification with validation/verification and deployment.

Author α : Associate Professor, Dept of Computers Science & Engineering, Vasireddy Venkatadri Institute of Technology, Guntur.
e-mail: mbstechinfo@gmail.com

Author σ : Acharya Nagarjuna University, Dept of Computers Science & Engineering, Guntur.

Author ρ : (M.Tech), Dept of Computers Science & Engineering, Vasireddy Venkatadri Institute of Technology, Guntur.

The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. It is used to extract patterns and knowledge from large amount of data. Aside from the raw analysis step, it involves database aspects, data pre-processing model and inference considerations, post-processing of discovered structures, visualization and online updating. The actual data mining task is the automatic or semi-automatic analysis of large quantities of data to extract previously unknown interesting patterns such as groups of data records (cluster analysis), unusual records (anomaly detection) and dependencies (association rule mining).

b) Objective

In recent years, online social network services such as Facebook and Twitter are becoming increasingly popular and have created huge amount of social network data. It is very difficult to store massive data in the computer memory. Many out-dated methods are not designed to handle massive amount of data. To address this problem all the outdate methods are re-designed under the computing framework that is well known of big data syndrome.

The main objective of the FSM is to extract the entire frequent sub graph in the given data set, whose occurrence counts will be specified above the specified threshold. This completely focus on effective mechanism for generating candidate subgraph (without duplicates) and to identify the frequent sub graph. Computational Efficiency (CE) emphasis on the efficiency extends to which time, effort or space is well used for the intended task. A scalable method for frequent sub graph mining and computation efficiency is of huge demand since many disciplines such as social networks, bioinformatics, cheminformatics and semantic web are much focused on this Frequent Sub Graph Mining and Computation Efficiency factors.

c) Scope Of The Project

MIRAGE is the Map Reduce algorithm for the frequent sub graph mining. This is mainly used for the creation of complete set of frequent subgraph for a given minimum support threshold. In map phase, it builds and recalls all patterns that have non-zero and in reducer phase it decides on which the pattern is frequent by aggregating their support through different computing nodes in order to ensure completeness. Mirage runs in an iterative manner such that output of

the reducers of iteration $i-1$ (where i denotes number of terms) is used as an input for the mappers in the iteration i where it will also compute the local support of candidate pattern. Reducers i then find the true frequent subgraph by aggregating their local supports. The proposed system will perform the data mining in an efficient way using the algorithm. An overview of the proposed work contains the below modules: Data Collection, Removal of duplicate sets, Establishing Iterative MapReduce framework and Comparison and analysis of results.

The major contributions are: an iterative MapReduce based frequent subgraph mining algorithm called MIRAGE used to address the frequent subgraph mining problem. Computational Efficiency will be increased through MIRAGE algorithm over Matrix Vector Multiplication. Performance of the MIRAGE will be demonstrated through different synthetic as well as real world datasets. The main aim is to improve the existing algorithm to enhance Computational Efficiency.

The Section 2 gives the related research work. Section 3 discusses the materials and methodologies and Section 4 presents our results and discussed them in section 5. Section 6 concludes the paper with future enhancements.

II. RELATED RESEARCH

Mansurulet al. [1] in their paper proposed the new algorithm for the frequent subgraph mining that address the key mechanism of candidate subgraph and is used to identify the subgraph. This paper clearly illustrates the Iterative MapReduce based algorithm to rectify the Frequent Subgraph Mining (FSM) problem in a very efficient manner. This algorithm gives the appropriate way to identify the frequent dataset and removes duplication. Social Graph Mining uses the same MIRAGE algorithm to address the Computational Efficiency (CE) problem.

Yi-Chen Lo et al. [2] in their paper represented the current need of the computational efficiency in mining large scaled social networks. This work presents the Computational Efficiency problem through the open source graph mining library called MapReduce Graph Mining Framework (MGMF). It deals with the large scaled social network mining tasks containing billions of entities where cloud computing is the solution. Author completely uses Matrix Vector Multiplication algorithms to resolve the Computational Efficiency problem.

SabaSehrish et al. [3] in their paper discussed about the high performance computing problems through Map Reduce with Access Patterns (MRAP) which will be a unique combination of the data access semantics and the programming framework used in implementing High Performance Computing (HPC) analytics application. This paper is referred to know the basic ideas of scheduling in MapReduce.

III. MATERIALS AND METHODS

a) Map Reduce Model

Map Reduce, proposed by Google, is a distributed model for processing large-scale data. Users specify a map function and a reduce function. MapReduce takes in a list of key value pairs, splits them among the possible map tasks and then each map function produces any number of intermediate key-value pairs. Pairs with similar keys are gathered together at the reduce tasks, and then each reduce function performs computations before outputting values, which are either the final results, or possibly input for the next iteration. Ideally, MapReduce frameworks consist of several computers, usually referred to nodes, on the scale of tens to thousands. Processing occurs on data stored in the file system. Computation should be parallelized across the cluster, fault tolerant, and scheduled efficiently.

b) Map Function

The mapper's job is to take in a key-value pair. This key-value pair often comes from a partition of data specified by the Map Reduce architecture. After processing, the map function will emit another key-value pair. An added bonus comes in the form of an in-mapper combiner, which can do local computations to lessen the burden on the file system by acting as a mini-reducer. After all mappers have finished, all of the results are shuffled, sorted, and sent to the reducers. Hadoop sends single lines from the input file to the mappers, to which each applies a map function to those lines. This initial map function will have the responsibility of sending the subgraph encoded in the input string to the correct reducer using the graph id. For the first iteration, the encoded input string will represent a single edge of the graph. For all other iterations, we have an encoded input string representing a subgraph of size $i - 1$.

c) Reduce Function

The reducer takes in a list of values corresponding to a specific key. Here, the reduce function can perform many operations, such as aggregations and summations. Since all the values we need have been grouped, bulk computations on those values become trivial.

d) Reducer For Constructing Subgraphs

Subgraphs of size $k - 1$ with the same graph id are gathered for the reducer function. Note all of the single edges in these subgraphs and use that information to generate the next generation of possible subgraphs of size k . Encodes this subgraph as a string just as was outputted from the previous map function. All labels are alphabetized and use special markers to designate differing nodes with the same labels. The

results of this step are written out to the Hadoop File System.

e) *Map Function For Gathering Subgraph Structures*

Similar to the process involving the first map function, Hadoop sends lines of input to the mappers. This second map function will have the responsibility of outputting the label-only subgraph encodings as a key and the node identification numbers and graph ids as values.

IV. ARCHITECTURE AND MODULE DESIGN

The basic architecture diagram of the entire system is given in figure 3.1. The above architecture illustrates the complete flow of social graph mining. Dataset are given as graph input data (social media, biological dataset). Graph data's are used to perform the data mining process. The process starts with frequent subgraph mining where all the duplicate sets are removed. MapReduce exactly perform mapping and reducing functions with the file system. Then FSM with mining process is carried out with the MIRAGE algorithm with three different phases such as partition phase, preparation phase and mining phase. FSM analysing is carried out for three different factors candidate generation, graph isomorphism and support counting. Final output is produced with the comparison result that increases computational efficiency.

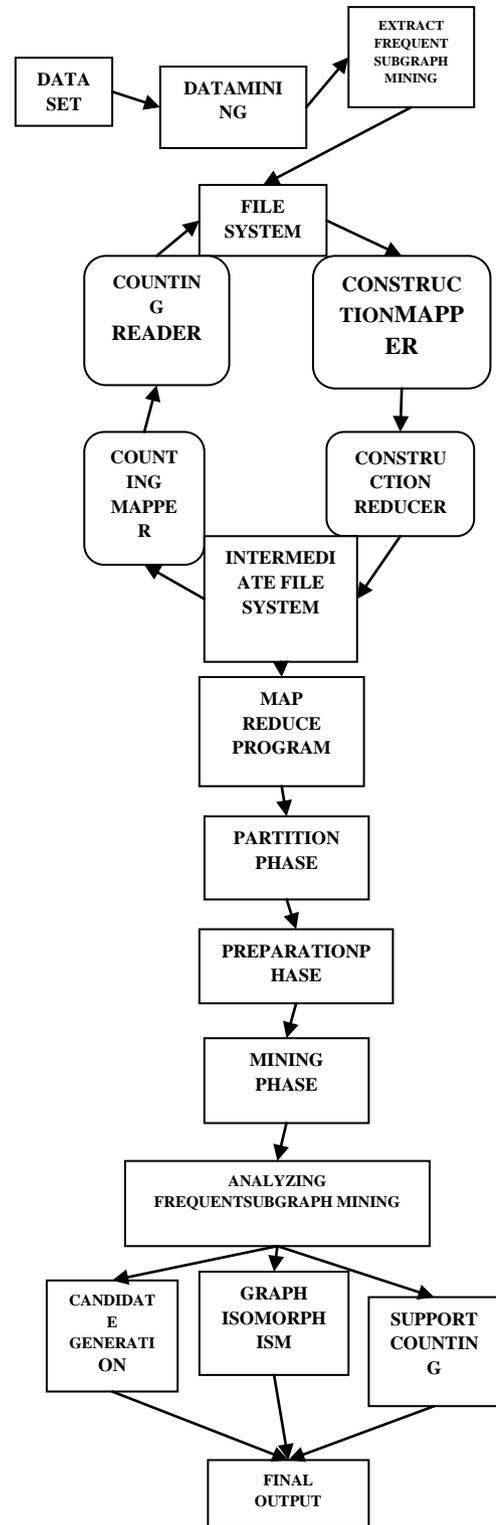


Figure 3.1 System Architecture

a) *Collect Graph Data*

This file contains the synthetic datasets and real world large graph dataset (social media). The graph



This page is intentionally left blank