A Survey on Mining Large Graph Databases

Data Mining*(CS 568), Autumn 2015*

Vinay Chandragiri
Computer Science and Engineering
Indian Institute of Technology
Guwahati
chandragiri@iitg.ac.in

Revanth Gopi
Computer Science and Engineering
Indian Institute of Technology
Guwahati
konakanchi@iitg.ac.in

Amit Awekar
Asst. Professor, Dept. of CSE
Indian Institute of Technology
Guwahati
awekar@iitg.ac.in

ABSTRACT

Graph database models are used in areas where information about interconnectivity or topology has equivalent importance as data. As of today, Mining recurring patterns from large graph databases is essential in areas like Bioinformatics, Web, Social Networks etc. Keeping in mind the computational power required to process queries in real time, this turns out to be a challenging problem.

The main objective of this survey is to give a gist of problems that can be addressed by mining of frequent subgraphs or patterns from large databases and present few efficient algorithms which are developed for the same.

This paper discusses about few technologies/frameworks that are being used in order to achieve the required. Comparison of the efficiencies of the above algorithms is also presented

Keywords

Subgraphs, Frequent Patterns, Indexing, Query Processing

1. INTRODUCTION

In this era where every simple minute information is considered as a data point there is huge amount of data that is being generated. It has been said that within the next 5 years some 75 billion connected devices will be in use worldwide. So imagine the data generated by these devices. There are many models in which this data can be stored as like the Network models, Relational models, Graph models & then Object Oriented models.

Graph database models are prevalent in areas where information about the connections between the data points is more or equally important than the data points itself in isolation. Graph databases are highly relevant in the area of Social Networks, Web, Bioinformatics etc. In these areas the amount of data being generated is very huge so in general Graph databases are very large. Mining Graph Patterns from these large databases is useful for understanding the underlying characteristics of data and also to identify conceptually interesting patterns. Mining frequent subgraphs in traditional way on large graph databases is not efficient.

Traditional mining techniques take hours to days sometimes to when scaled up. This problem has attracted much research interest because of its varied applications and importance in current world. In this survey three major subgraph Mining techniques namely GSPAN, Fast Frequent Subgraph Mining(FFSM) and SPIN. In large graph databases where frequent subgraphs are also large in number SPIN only mines maximal frequent subgraphs thereby reducing the computational costs. Experimental studies show that FFSM always gives on par or better performance results compared to GSPAN whereas SPIN out performs gSPAN by order of magnitude. This survey also gives a flavour of two graph database technologies Neo4J & Giraph.

These are the most widely used databases with many ready to use application programming interfaces(API?s). Rest of the survey is organised as follows. In section 2, we address why graph databases are essential and kind of problems that need to be addressed. Section 3 presents required background of pattern mining. In section 4, we present gSPAN and FFSM algorithms for mining patterns on graphs in general and how SPIN scales with large graph databases. Section 5 presents the comparison of performance between the three algorithms. Section 6 presents application level information of two latest graph database technologies Neo4J and Giraph. In Section 7 all the references are listed.

2. WHY IS IT ESSENTIAL ?

Graphical representation of data allows natural modelling of data where queries can directly mapped to the graphical structure such as finding shortest paths etc.

Interconnectivity of components is the key feature of a Graph database model as it helps to identify complex relationships and interdependencies among data points. These representations of data make it easy without having complex schemas.

Important feature of Graph Databases is that it is easy to unlock hidden meaning in the data. Inference and Semantic Data Integration are the most important attributes of a graph database.

Inference allows us to create new facts from the existing ones. Semantic Data Integration allows us to integrate many forms of data while maintaining connections with original sources.

2.1 What kind of problems that need to be addressed ?

MANAGING TRANSPORT NETWORKS

Roadways/Railways
In large metropolitan cities where people commute with in the city very often, properly planned public road transportation is the only way to avoid too many personal vehicles. Deciding on number of buses to be commuted in each route, deciding hassle free routes for the passengers has to be done in an efficient way.

Mining frequent subgraphs from a graph with different destinations as nodes and number of passengers travelling between two destinations as labels to the edges will provide an optimistic transport plan.

Air Routes
Insights on customers travel information between different cities/countries along with their dates can be mined using frequent subgraphs to minimise the number of airbuses and thereby saving revenue loss for the airlines.

SOCIAL NETWORKS

In social networks where graph databases are prevalent, mining frequent subgraphs in traditional way takes large amount of time.

Later on, in this paper we will discuss SPIN algorithm that performs especially well for mining frequent subgraphs on large graph databases.

INFORMATION NETWORKS

Many web applications use subgraph mining to identify which pages its users navigate frequently, which sequence it?s products are bought and much more.

BIOLOGICAL NETWORKS

Drug Development
Molecular structures for existing drug compounds are modelled into graphs to which subgraph mining is applied to develop new drugs.

Protein Reconstruction
Proteins interact with each other in a certain way to enable a biological process inside a cell. Modelling these interactions into labels of the graph and there by deriving conclusions with mining rules is prominent these days.

INTERNET OF THINGS

Its the connections at different points between people, places, machines that makes IoT work. All these connections are modelled into graphs. Graph databases are used to store this massive information. frequent subgraphs from these can give useful insights on improving performance, business models.

Graph Mining is also essential in areas in Financial Services, Recommendation Systems, Telephone Networks etc.

3. BACKGROUND

3.1 Road Map to Pattern Mining

KINDS OF PATTERNS AND RULES

Basic Patterns
This mainly involves frequent patterns, closed patterns and association rules.

Multilevel & Multidimensional Patterns
Multi level mining have flexible support thresholds depending on the levels. In multi dimensional mining two are more predicates are used.

Extended Patterns
These include approximate patterns, uncertain patterns and rare/compressed patterns.

MINING METHODS

Basic Mining Methods
Candidate generation is the most basic and frequent mining method used. A-priori, Partitioning, Sampling are some common algorithms in candidate generation. Pattern Growth, Vertical Format are other basic mining methods.

Mining Interesting Patterns
Some of the types of these patterns are Interestingness (Subjective Vs Objective), Constraint-based mining, Correlation rules and exception rules.

Distributed, parallel, incremental
Here we include distributed/parallel mining, Incremental mining, and Stream pattern.

EXTENSIONS AND APPLICATIONS

Extended Data Types
These include sequential ad time series patterns, structural patterns, spatial patterns, temporal (evolutionary, periodic), image, video and multimedia patterns, network patterns.

Applications
These include pattern-based classification, pattern-based clustering, pattern-based semantic annotation, collaborative filtering, and privacy-preserving.

4. EXAMPLE ALGORITHMS FOR MINING

Here we also need to address different types of Problems:

- Frequent Pattern Mining
- Frequent Subgraph Mining

The following are the major properties of Graph Mining Algorithms.

- Order of Search (Breadth Vs Depth)
- Generation of Candidate Sub-Graphs
- Elimination of Duplicate Sub-Graphs
- Support Calculation
- Discover Order of Patterns

4.1 gSpan

gSpan is graph based substructure pattern mining. It discovers frequent substructures without candidate generation by building new lexicographical order among graphs and there after mapping each graph to a unique minimum DFS code.

Pseudo Code

```
gSpan(D, F, g)
1: if g ≠ min(g)
     return;
2: F ← F ∪ { g }
3: children(g) ← [generate all g' potential children with one edge growth]
4: Enumerate(D, g, children(g))
5: for each c ∈ children(g)
     if support(c) ≥ #minSup
        SubgraphMining (D, F, c)
```

4.2 FFSM

Fast Frequent Subgraph Mining addresses two main issues of any apriori based algorithms (i) Cost Efficient subgraph testing (ii) Better candidate enumeration scheme. FFSM completely avoids subgraph isomorphism testing by maintaining an embedding set for each subgraph.

FFSM

1: $S \leftarrow$ { the CAMs of the frequent nodes }
2: $P \leftarrow$ { the CAMs of the frequent edges }
3: FFSM-Explore(P, S);

FFSM-Explore (P, S)

1: **for** $X \in P$ **do**
2: **if** $(X.isCAM)$ **then**
3: $S \leftarrow S \cup \{X\}, C \leftarrow \Phi$
4: **for** $Y \in P$ **do**
5: $C \leftarrow C \cup$ FFSM-Join(X, Y)
6: **end for**
7: $C \leftarrow C \cup$ FFSM-Extension(X)
8: remove CAM(s) from C that is either infrequent or not suboptimial
9: FFSM-Explore(C, S)
10: **end if**
11: **end for**

4.3 SPIN

SPIN is Spanning Tree based Maximal Subgraph Mining.

It is used to mine maximal frequent subgraphs from large graph database. This is a new framework, which partitions frequent subgraphs into equivalence classes is proposed together with a group of optimization techniques.

SPIN offers good efficiency over large graph databases in order of magnitude performance. It also give high performance in terms of scalability i.e over different data sizes.

5. TECHNOLOGIES FOR LARGE SCALE GRAPH PROCESSING

5.1 Neo4J

Neo4J is a NoSQL(Non Relational) database. These kinds of Databases are used in Real Time Web applications and deal with Large amounts of Data. Other examples for NoSQL databases are MongoDB, Cassandra etc.

Neo4J is a Desktop Platform. It Provides fast read and write performance protecting the data integrity. It combines native-graph storage, along with Scalable architecture for Speed. It Supports operations as ACID transactions.The speed is optimised and will not be affected while importing or exporting data at a sarge scale.

It uses a two level main memory caching technique to ensure its speed. It has inbuilt plugins to most of the graph algorithms for example Shortest Path, Flow, Pairing etc. Its traverses all nodes using BFS or DFS in a graph and uses reference vertices for a particular traversal.

Its is easy to use and can be done by Cypher, a productive graph query language similar to SQL. Gremlin is also a generic graph Query Language which is also used. Its also helpful in writing custom extensions. its?s user interface is easy to handle and provides better enhancements. Its also provided direct access to JVM based applications.

These technologies can be used to address problems like Movie Recommendation, Document Similarity etc The performance can be worst when its out of memory and it is being executed on a single machine.

5.2 Giraph

Giraph is a cluster based platform which is used for large scale graph processing distributed system similar to Hadoop-Map Reduce. It is built based on Preleg Programming model that is adapts to Bulk Synchronous Parallel (BSP) Model.

Giraph processes graph in memory and uses dynamic computation where in which only few of the vertices will be processed in all iterations which is actually a disadvantage when large amount of memory is needed and Hence may crash.

Giraph turned out to be the fastest platform in test experiments made by Mark. There are many other platform for large scale graph processing for example GraphX etc.

The Programming model and Platform design are the two important and significant factors to be considered for large scale data processing platforms.

6. REFERENCES

- Graphs-at-a-time: Query Language and Access Methods for Graph Databases
- Algorithmics and Applications of Tree and Graph Searching
- Are graph databases ready for bioinformatics?
- Graph Databases- An Overview
- Kineograph: Taking the Pulse of a Fast-Changing and Connected World
- Query Languages for Graph Databases
- Survey of Graph Database Models
- Efficient Mining of Frequent Subgraph in the Presence of Isomorphism