IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.7, Special Issue.2, Jan 2019 E-ISSN: 2347-2693

Frequent Mining Techniques In Bigdata : Study

Muthamiz Selvi[1*], P. Srivaramangai[2]

[1,2]Department of Computer Science, Marudupandiyar College(Affiliated to Bharathidasan University), Thanjavur - 613 403, Tamilnadu, India

Available online at: www.ijcseonline.org

***Abstract**:* Big data is a collection of large amount of data with various types of data and usable to be processed at much higher frequency. Frequent Itemset Mining is one of the classical data mining problems in most of the data mining applications in big data era. In data mining, association rule mining is key technique for discovering useful patterns from large collection of data. Frequent itemset mining is a famous step of association rule mining. Many efficient pattern mining algorithms have been discovered in the last two decades, yet most do not hold good for Big Dataset. In association rule mining (ARM) a Frequent Itemset Mining (FIM) is a well-known step. In last two decades, many efficient pattern mining algorithms have been discovered, up till now most do not hold good for Big Dataset. The Apriori, FP-growth and Eclat algorithms are the most famous algorithms which can be used for Frequent Pattern mining. However, these **parallel** mining algorithms lack features like automated parallelization, fine load balancing, and distribution of data on large clusters. To overcome these problems various parallelized approaches using Hadoop MapReduce model are developed to perform frequent itemsets mining from big data. This paper gives overall study about frequent pattern mining in big data.

Keywords: Big data, Pattern Mining, Frequent Itemset Mining**,** Data Mining, ItemSets

I. INTRODUCTION

Data mining allows users to understand and discover knowledge in large amounts of data by mining data patterns (or simply called *patterns*) [1] [2] [3] [4]. A *pattern* can be any type of regularity that appears in data collections, which are considered a kind of summary of the input data [5]. For example, a set of frequent bought together products from a shopping basket analysis, a piece of abnormal gene sequence carried by patients for drug research, a historical record of a visitor's past travelling experiences for planning the next trip, or the reaction of a particular enzyme to the external stimulus for the study of disease treatment. All of these patterns carry useful insights from the collected data and have the potential to solve the problems that occur in practical applications. *Pattern mining* is a mining process for extracting these valuable data patterns from large amounts of data [3]. With the fast development of computing technology, purely manual data analysis has been replaced by an automatic or semi-automatic data mining process [6]. Various state-of-the-art *pattern mining* algorithms have been reported in the literature, however *why do we still need to explore this topic at a deeper level?* To discover patterns is not a difficult task, but to discover *interesting* patterns from large-scale databases is not easy. A pattern is interesting if it can provide useful and beneficial knowledge to end users for solving their practical application problems. This is where the complexity of the work comes from. In addition, the large size of the pattern set often causes confusion to the end users so that insignificant knowledge is finally returned to target the expected findings of the data analysis. Thus, the new challenge in the research field of *pattern mining* is ***to find the most targeted data patterns that are highly interesting and useful to the end users to meet the requirements of their specific applications***.

Due to the variety of the forms representing numerous data in the existing research domains, it is infeasible to develop one single *pattern mining* algorithm that can meet all requirements. The work reported in this thesis focuses on mining data patterns from two forms of complex data: *tree structures* and *relational data*, which are popular in many emerging research domains, such as Web mining, Chemistry, Biology, social networks, business and marketing analysis [7] [8] [9] [10] [11].

II. ASSOCIATION RULE MINING

One of the fundamental methods from the prospering field of Data Mining is the generation of association rules that describe relationships between items in data sets. Association rule mining is primarily focused on finding frequent co-occurring associations among a collection of items. It is sometimes referred to as "Market Basket Analysis", since that was the original application area of association mining. The goal is to find associations of items that occur together more often than you would expect from a random sampling of all possibilities. Generally speaking an Association Rule is an implication of the form:

$$X \rightarrow Y$$

Where X and Y are distinct sets of items. The meaning of such rule is quite intuitive: Let *DB* be a transaction database, where each transaction $T \in D$ is a set of items. An association rule $X \rightarrow Y$ expresses that "Whenever a transaction T contains X then this transaction T also contains Y with probability *conf*". The probability *conf* is called the rule confidence and is supplemented by further quality measures like rule support and interest. The support is an indication of how the itemset appears frequently in the database. It is sometimes expressed as a percentage of the total number of records in the database. The confidence is an indication of how often the rule has been found to be true. An Example for Association Rule Mining is identifying the items that occur frequently from a large transactional database. For this, association rule mining can be used, even if the customers who bought the items are unknown. *An A*ssociation Rule Mining searches for interesting relationship among those items and displays it in a rule form. An association rule "{*bread, jam*} (*sup* = 2%*; conf* =80%)" states that 2% of all the transactions under analysis show that bread and jam are purchased together and 80% of the customers who bought bread also bought jam. Such rules can be useful for decisions concerning product pricing, promotions, and many things. Association rules are also widely used in various areas such as telecommunication networks, market and risk management, inventory control, etc.

Phases of Association Rule Mining:

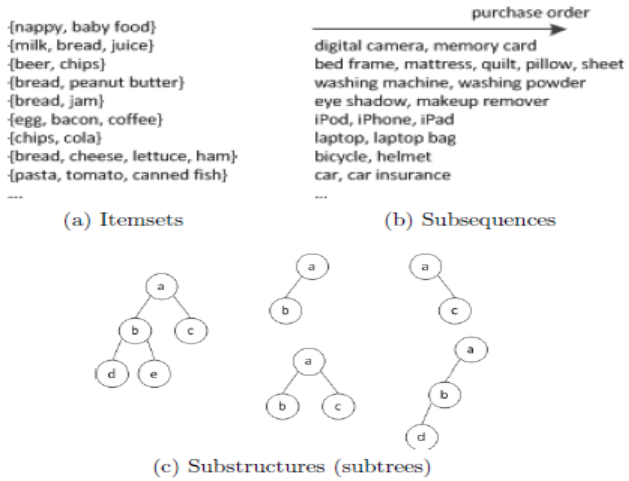
It consists of two phases:

- Finding all frequent patterns. By definition, each of these patterns will occur at least as frequently as a pre-defined minimum support threshold. Minimum Support threshold is the minimum support for an itemset to be identified as frequent.
- Generating association rules from frequent patterns. Association rules can be formed only by satisfying the pre-defined minimum support threshold and minimum confidence threshold.

The second phase is straightforward and less expensive. Therefore the first phase of FPM is a crucial step of the two and determines the overall performance of mining association rules. In addition to this, frequent pattern plays an essential role in many Data Mining tasks that try to find interesting patterns from databases, such as association rules.

III. PATTERN MINING

Pattern mining is quite similar to ore mining [12] which removes soil (noisy data) and extracts valuable minerals (useful patterns) from underground ore bodies. No matter what advanced techniques miners use or how well they have been trained, the most important precondition of any successful mining activity is that miners known what minerals they are looking for. For example, metallic *aluminium* [13] is mainly produced from a special type of ore, called *Bauxite* [14]. If miners have no idea that bauxite is a red-brown rock, they might waste time and energy on finding the shinning silvery gray blocks that rarely exist in the earth's solid surface. Therefor, before any *pattern mining* task begins it is important to ask: ***What is an interesting pattern?*** In *data mining* tasks, a pattern can be an *itemset*, a *subsequence* or *substructures* appearing with a certain frequency in a database [5]. This pattern is believed to carry some kind of useful information, which can be used to represent particular characteristics of data instances within the database.



(a) Itemsets (b) Subsequences

(c) Substructures (subtrees)

Figure 1: Patterns in Various Data Forms

A pattern in the form of an itemset was originally discovered from market-basket data analysis using the *association rule mining* algorithm [10]. Each data instance consists of a number of product items that have been purchased by a customer. The purpose of the analysis task is to find the itemsets that frequently appear in the database. Each discovered itemset shows a set of product items that have been frequently purchased together by the customers in a certain store or supermarket (as shown in Figure 1 a). A sequence can be considered as a set of ordered events, elements or items with or without a concrete notion of time [15]. For example, a new car is purchased before buying car insurance as shown in Figure 1 b. Many *sequential pattern mining* algorithms have been developed to extract a set of subsequences that frequently appear in a special order. In addition, if the database records the data elements of objects together with their structural information, the potential patterns to be discovered are often in the form of substructures, such as sub graphs in a graph database [16] or sub trees in a tree database [4] (Figure 1 c).

The complexity of mining algorithms increases from mining item sets, and subsequences to substructures. When the order of items is considered, the traditional *itemset pattern mining* becomes *subsequential pattern mining* [17]. *Substructure pattern mining* [18] can also be regarded as mining patterns from a database representing structural information, where each data structure is a representation of two or more sequences that merge together at the common items.

The rest of this chapter reviews the most influential data mining techniques that have been developed to help discover interesting patterns for various data analysis and mining applications. This chapter begins with an introduction of data patterns that attract the attention of researchers. The relevant *pattern mining* algorithms and applications are surveyed separately in individual sections with respect to the certain types of data patterns they are interested in. Finally, we discuss current difficulties and problems that still exist in *pattern mining*, with focus on the challenges of such research.

3.1 Pattern Types

The word *pattern*, as defined in *The Oxford Dictionary of English* [19], is originally from Old French patron, which refers to "*a regular and intelligible form or sequence discernible in the way in which something happens or is done*". Two key points need to be taken from this general definition. Firstly, a pattern *appears regularly* in the observation data, while secondly, a pattern has a specific *role* for the occurrence or the implementation process of something. Hence, in this thesis, there are two factors considered for the mining process of interested patterns:

- *Occurrence Frequency*: the regularity of a pattern is usually determined by counting the frequency of its occurrence in data;
- *Application Purpose*: the patterns are mined with a particular purpose that is highly relevant to the target of the application task.

In the consideration of an occurrence frequency, a pattern can be classified as either *frequent* or *infrequent*, which is determined by the standard constraints [20]. If the application aims to find out the most common or the similar characteristics among data instances in the database, the *common* patterns will be returned by the mining process. On the other hand, if the target is to discover the difference, then output patterns are *contrast* ones that can be used to distinguish or classify data instances [21]. To propose efficient *pattern mining* algorithms that can effectively discover useful patterns from large-scale databases, it is necessary to have a good understanding of different pattern types and their characteristics. In the rest of section, we will provide detailed definitions of these pattern types and discuss the specific methods used to distinguish them.

3.1.1 Frequent Pattern and Infrequent Pattern

To determine whether a pattern is frequent or not, a well-known measurement is the minimum *support* threshold. The concept of *support* was successfully employed in *Association Rule Mining* by Agrawal et al. [10], frequent patterns in the form of itemsets could be discovered from the market-basket transaction databases.

Let I be the set of all items in the database D_{db}, A and B be two items from I,
$A \in I$, $B \in I$, and $A \cap B = \emptyset$. The *support* of the itemset *{AB}* is calculated by

$$supp(A \cup B) = \frac{\text{number of transactions containing } A \cup B \text{ in } D_{db}}{\text{total number of transactions in } D_{db}} \quad (3.1.1)$$

{AB} is a *frequent itemset* if and only if $supp\ (A \cup B) \geq \delta$, where _ is a user-specified minimum support threshold; in contrast, it is an *infrequent itemset* due to $supp(A \cup B) < \delta$. The *support* is a common and traditional measurement in mining frequent patterns in other data forms such as sequences, graphs and trees [22]. The discovered frequent pattern summarizes the correlations among a set of items in the database. The mining task that focuses on discovering frequent patterns from the databases is called *frequent pattern mining*. In *frequent pattern mining*, only frequent patterns are returned while infrequent patterns are simply discarded without further consideration. This is because the most valuable information is carried by the frequent patterns and the infrequent patterns cannot adequately reflect the typical characteristics from the data because of their rare occurrence. However, since the late 1990s, more and more researchers have realized the importance of infrequent patterns with the increasing demands from applications of anomaly detection, especially in medicine, genetics, molecular biology and network security. In these areas, infrequent patterns are considered significant due to the huge influence they may have. In the study of finding a better treatment approach for a special disease, researchers would like spend more time on studying an abnormal case rather than reading the millions of records of healthy people [23]. In this scenario, more effort has been put into the development of *infrequent pattern mining*. More detail of related work in the development of *frequent and infrequent pattern mining* algorithms is reviewed in below Section.

IV. FREQUENT PATTERN MINING ALGORITHMS

As one of the most important and well-explored topics in *data mining*, *Frequent Pattern Mining* (FPM) has been studied for over two decades [24]. The purpose of *frequent pattern mining* is to discover all frequent itemsets, subsequences or substructures that appear in a large-scale database. The frequency of these patterns must be equal or over a pre-defined *minimum support threshold* provided by a user. The discovered frequent patterns can be used to make further contributions to other *data mining* topics, such as association mining and correlation mining.

In 1993, Agrawal et al. [10] proposed the first *frequent pattern mining* algorithm, the *AIS* algorithm, which employed a *multi-pass technique* to generate candidate itemsets from a transaction database. A pass means a movement from one transaction to the next transaction. In each pass, a set of known-frequent itemsets are determined by repeatedly scanning a database to measure their *supports*, with new candidate itemsets generated by extending the frequent itemset with the items in each transaction. However, the *AIS* algorithm suffers from generating too many candidates that are identified as infrequent and discarded in a later process. Hence, in order to reduce the computational cost and complexity, a downward closure property, called *Apriori*, was proposed by Agrawal and Srikant. In the 1990*s*, *Apriori* was popular in many *frequent pattern mining* algorithms. Many algorithms used *Apriori* or its alternatives, called the *Aprrori-based Algorithms*.

4.1 *Apriori-Based* Algorithms

In an *Apriori-based* algorithm, a candidate itemset (subsequence or substructure) is identified as frequent, if and only all of its subsets are frequent [25]. That is, it is impossible to have a candidate who has an infrequent subset. Based on this, the *Apriori-based* algorithm has significantly optimized the pruning process. Unlike the *AIS* algorithm, the *Apriori-based* algorithm generates new candidate *k*-itemsets based on the known-frequent ($k - 1$)-itemsets, but not the items from the current scanned transaction. Besides, in an *Apriori-based* algorithm,a *hashtree* data structure is utilized to store the frequency counters of each candidate.

Let us take Agrawal and Srikant's *Apriori-based frequent itemset mining algorithm* as an example of a traditional *Apriori-based* algorithm. Given a transaction database D_{db}, $I = \{A1; A2; : : : ;AL\}$ is a set of L items that appear in D_{db}. Each transaction t_i in D_{db} is represented by a set of items from I, i.e. $t1 = \{A1; A2; A4; A7; A10\}$. A *minimum support threshold* δ is also determined for pruning infrequent candidate itemsets. Database D_{db} is scanned for the first time to extract all of the frequent 1-itemsets that contains only one item, $F^{(1)} = \{A^{(1)}\ 1\ ;A^{(1)}\ 2\ ; : : : \}$ and $A^{(1)}{}_p = \{A_i\}$. By extending any frequent 1-itemset with one item from $F^{(1)}$,a set of candidate 2- itemsets are generated, in which each 2-itemset contains two items $A^{(2)}\ p = \{A_i;A_j\}$. The *support* of each candidate 2-itemset is calculated and compared with the *minimum support threshold* δ. If and only if $supp(A^{(2)}\ p\ ) \geq \delta$, then such $A^{(2)}{}_p$ is identified as a frequent 2-itemset $F^{(2)} = \{A^{(2)}{}_p\ \}$; while any infrequent candidate with *support* less than δ is discarded. Once all of the frequent 2-itemsets are discovered, they will be extended to form a candidate 3-itemset. This process iterates to generate frequent *k*-itemsets $F^{(k)}$ and stops when there is no frequent ($k + 1$)-itemset that can be generated.

Compared with the original *AIS* algorithm, the traditional *Apriori-based* algorithm adopts a more efficient mining method by cutting a large number of candidates; however, it suffers from the wastage of generating too many infrequent candidates. Another main drawback is that the traditional *Apriori-based* algorithm also needs repetitive scanning of the database. The number of scans is determined by the length of the longest frequent candidate. If there exists frequent *k*-itemsets, then the database needs to be scanned *k* times. In order to further improve the efficiency of *Apriori-based* algorithms, a number of extended studies [26] have been carried out to accomplish the mining process with*two* database scans at most. Savasere et al. proposed a *partitioning technique* that logically divides the entire database into a number of disjointed partitions. A *tid-list* (a list of transactions that contain a certain candidate) is utilized to record the frequency counting for each candidate in each partition. A candidate is considered as frequent in the entire database only if it is frequent in at least one partition. A *sampling approach* was proposed by Toivonen in 1996 to provide approximate mining results, where a set of frequent candidates was discovered from a randomly selected sample transaction in the database [27]. The *Dynamic Itemset Counting* algorithm and *Continuous Association Rule Mining* algorithm allow a dynamical generation and removal of candidates after scanning of a fixed number of transactions. These extended algorithms of *Apriori* successfully simplified the mining process by adopting less database scans but they still suffered from the redundant work of traversing the data structures that store the frequency counters. This is called the *tuple-by-tuple* problem due to the frequency counter of a candidate updating only after a complete reading of each transaction.

4.2 Frequent Pattern-growth Algorithms

Although *Apriori-based* algorithms and their extensions made significant contributions in the early days of *frequent pattern mining*, they still have high computational and storage costs in generating a large number of candidates. Therefore, in 2000, a new *frequent pattern mining* approach was proposed by Han et al. In this approach, the candidate generation process was no longer required and two database scans were requested to construct an enumeration tree structure that can represent all frequent candidates in the database. Such a tree structure is named *Frequent Pattern tree* (FP-tree) due to the enumeration technique utilized, which is called *Frequent Pattern-growth* (FP-growth) [28].

After the first scanning of the database, a list of frequent items, together with their associations are obtained. These items are ordered based on their frequency in a descending order. The association information helps to construct a *FP-tree* that holds all these frequent items. At this stage, the database is no long required in the mining process, with all frequent patterns extracted from the built *FP-tree* by a bottom-up approach based on the *divide-and-conquer* principle. The mining process is started by enumerating each frequent 1-item (the leaf node in *FP-tree*) and corresponding *sub- FP-tree* containing all the prefix paths leading to the frequent 1-item. This frequent 1-item is a suffix pattern and is removed from the *sub-FP-tree*. The remaining part of the *sub-FP-tree* becomes the *conditional FP-tree* of that certain suffix pattern. Inside the *conditional FP-tree*, this process is recursively performed to identify a new suffix pattern and its *conditional FP-tree*. The frequency of the suffix patterns are counted based on their prefix paths. By concatenating the suffix pattern with the frequent patterns from its *conditional FP-tree*, the pattern grows. The mining algorithm that adopts the *FP-growth* technique to discover frequent patterns is called the *FP-growth* approach.

The *FP-growth* approach has been extended by many researchers in *frequent pattern mining* studies. Agarwal et al. proposed a depth-first generation of frequent itemsets in 2001 and a pattern mining algorithm based on *hyper-structure* was presented by Pei et al. Liu et al. introduced a mining algorithm utilizing both top-down and bottom-up traversals of their proposed *Condensed FP-tree*. In addition to these, another array-based prefix-tree structure was proposed by Grahne and Zhu to improve the efficiency of *FP-growth* algorithms. This *FP-growth* approach avoids the drawbacks of *Apriori-based* algorithms and mines frequent patterns without candidate generation, however, it is a time consuming process to construct a *FP-tree*. In addition, a *FP-tree* may have a complex structure and be large in size due to a number of the items involved in the database and the complexity of their associations. Once a *FP-tree* is set up, it is not easy to make changes, which makes the computation of *FP-growth* mining algorithms neither flexible nor reusable. Hence, if the frequent patterns to be discovered are in complex data forms like subsequences or substructures, many algorithms will still follow the *Apriori-based* approach rather than *FP*-growth [29].

4.3 Infrequent Pattern Mining Algorithms

The motivation of *Infrequent Pattern Mining* (IPM) comes from the argument that infrequent patterns are also interesting in many real-life cases. In *frequent pattern mining*, if a candidate pattern has a lower *support* than a pre-defined *minimum support threshold*, it will be discarded and no longer considered in the later process.

A common way to keep more potentially interesting patterns is to set a low *minimum support threshold*. However, a low threshold may cause many problems and difficulties in the mining process, such as a large number of candidates identified as frequent and kept for further processing so the computational cost is increased. In addition, no matter how low the value is assigned, it is always possible that some interesting patterns will be filtered out. Hence, some of the researchers in the area of *pattern mining* started to work out a solution to address this issue.

Wu et al. proposed a mining algorithm to extract both frequent and infrequent patterns from transaction databases. They extended the candidate generation process in existing *frequent pattern mining* by keeping the candidates that were identified as infrequent rather than discarding them. When the *support* of a candidate is less than the pre-defined *minimum support threshold*, it is identified as *infrequent* and is added into an *infrequent candidate pattern list*. As the number of frequent candidates is large and once the infrequent ones are counted into consideration, the number increases in an exponential manner. In order to control the number of generated candidates in a reasonable range, as well as satisfy the end user’s interests, Wu et al. introduced a method where only frequent *k*-candidates are allowed to join the next iteration of generating (k + 1)-candidates.

By taking advantage of mining infrequent patterns, many successful mining algorithms have been developed for discovering interesting patterns in different research domains. Wan and An proposed a *HI-mine* algorithm to discover indirect associations hidden in databases. Yan et al. extracted *surprising periodic patterns* that occur infrequently in biological data. Dong mined both frequent and infrequent itemsets using a minimum correlation strength as measurement to improve the performance of the mining model based on multiple level minimum supports. Even though *infrequent pattern mining* is still an emerging research field and has been studied for a decade, there are still many unsolved topics that can be explored, such as how to further control the number of generated candidate and how to improve the efficiency of the mining process by providing more targeted candidates, etc [30].

V. CONCLUSION

In recent years the size of database has increased rapidly. Therefore require a system to handle such huge amount of data. There are various parallel mining algorithms available for frequent itemsets mining, such as Apriori, Fp-Growth algorithms. But it becomes a very difficult task when they are applied to Big Data. Recent improvements in the field of parallel programming already provide good tools to tackle this problem. Hadoop is one such tool which is used to process the big data in parallel using MapReduce. In this paper, present the deep review on frequent itemsets mining (FIM) techniques. To solve the scalability and load balancing challenges in the existing parallel mining algorithms for frequent itemsets, we have to develop a parallel frequent itemsets mining algorithm.

REFERENCES

[1]. K. Mehmed. *Data Mining: Concepts, Models, Methods, and Algorithms*. John Wiley & Sons., 2003.

[2]. J. Han and M. Kamber. *Data Mining: Concpets and Techniques*. Academic Press, 2nd edition edition, 2006.

[3]. J. Vreeken. *Making Pattern Mining Useful*. PhD thesis, the Dutch Research School for Information and Knowledge Systems, Netherlands, 2009.

[4]. F. Hadzic, H. Tan, and T.S. Dillon. *Mining of Data with Complex Structures*, volume 333. Springer, 2010.

[5]. J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: Current status and future direction. *Data Mining and Knowledge Discovery*, 15:55–86, 2007.

[6]. I.H. Witten, E. Frank, and A.M. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 3rd edition edition, 2011. [7] R. Kosala and H. Blockeel. Web mining research: A survey. *SIGKDD Explorations*, 2(1):1–15, 2000.

[7]. F. Medici, M.I. Hawa, A. Giorgini, A. Panelo, C.M. Solfelix, R.D.G. Leslie, and P. Pozzilli. Antibodies to GAD65 and a tyrosine phosphatase-like molecule IA-2ic in Filipino Type I diabetic patients. *Diabetes Care*, 22(9):1458–1461, 1999.

[8]. W. Shi, F.K. Ngok, and D.R. Zusman. Cell density regulates cellular reversal frequency in Myxococcus xanthus. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 93(9), pages 4142–4146, 1996.

[9]. R. Agrawal, T. Imieinski, and A. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pages 207–216, Washington DC, 1993. ACM Press.

[10]. A. Jim´enez, F. Berzal, and J. Cubero. Frequent tree pattern mining: A survey. *Intelligent Data Analysis*, 14:603–622, 2002.

[11]. M.J. Zaki. Efficiently mining frequent trees in a forest: Algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1021– 1035, 2005.

[12]. X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In *Proceedings of 2002 International Conference on Data Mining (ICDM'02)*, pages 721–724, Maebashi, Japan, December 2002.

[13]. Y. Chi, R.R. Muntz, S. Nijssen, and J.N. Kok. Frequent subtree mining – an overview. *Special Issue on Graph and Tree Mining*, 66(1-2):161–198, 2005.

[14]. P. Cserk´uti, T. Levendovszky, and H. Charaf. Survey on subtree matching. In *Proceedings of the International Conference on Intelligent Engineering Systems (INES 200)*, pages 216–221, London, September 2006.

[15]. R. Shamir and D. Tsur. Faster subtree isomorphism. *Journal of Algorithms*, 33:267–280, 1999.

[16]. A. Gupta and N. Nishimura. The complexity of subgraph isomorphism: Duality results for graphs of bournded path- and tree-width. Technical report, University of Waterloo, April 1995.

[17]. M.B. Miles and A.M. Huberman. *Qualitative Data Analysis: An Expanded Sourcebook*. SAGE Publications, Inc., 2nd edition edition, 1994.

[18]. A.N. Oppenheim. *Questionnaire Design and Attitude Measurement*. Pinter Publications, 1992.

[19]. B. Boukhatem, S. Kenai, A. Tagnit-Hamou, and M. Ghrici. Application of new information technology on concrete: An overivew. *Journal of Civil Engineering and Management*, 17(2):248–258, 2011.

[20]. D.L. Olson. Data mining in business services. *Service Business*, 1(3):181–193, 2007.

[21]. N.R.S. Raghavan. Data mining in e-commerce: A survey. *Sadhana – Academy Proceedings in Engineering Sciences*, 30(2-3):275–289, 2005.

[22]. C. Romero and S. Ventura. Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 40(6):601–618, 2010.

[23]. R. Law, R. Leung, and D. Buhalis. Information technology applicaions in hospitality and tourism: A review of publications from 2005 to 2007. *Journal of Travel and Tourism Marketing*, 26:599–23, 2009.

[24]. C.J. Date. *An Introduction to Database Systems*. Addison-Wesley, 6th edition edition, 1994.

[25]. R.B. Sher and R.J. Daverman. *Handbook of Geometric Topology*. Elsevier, North-Holland, 2002.

[26]. A. Ceglar and J.F. Roddick. Association mining. *ACM Computing Survey*, 38(2):5, 2006.

[27]. V.P. Magnini, E.D.Jr. Honeycutt, and S.K. Hodge. Data mining for hotel firms: Use and limitations. *Cornell Hotel and Restaurant Administration Quarterly*, 44(2):94–105, 2003.

[28]. G.G. Emel, C, . Takin, and ¨ O. Akat. Profiling a domestic tourism market by means of association rule mining. *Anatolia*, 18(2):334–342, 2007.

[29]. X. Wu, C. Zhang, and S. Zhang. Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems*, 22(3):381–405, 2004.

[30]. Y.-J. Tsay, T.-J. Hsu, and J.-R. Yu, "FIUT: A new method for mining frequent itemsets," Inf. Sci., vol. 179, no. 11, pp. [20] 1724–1737, 2009.