Chapter 2
Big Data Analysis and Mining

Carson K. Leung
University of Manitoba, Canada

ABSTRACT

Big data analysis and mining aims to discover implicit, previously unknown, and potentially useful information and knowledge from big databases that contain high volumes of valuable veracious data collected or generated at a high velocity from a wide variety of data sources. Among different big data mining tasks, this chapter focuses on big data analysis and mining for frequent patterns. By relying on the MapReduce programming model, researchers only need to specify the "map" and "reduce" functions to discover frequent patterns from (1) big databases of precise data in a breadth-first manner or in a depth-first manner and/or from (2) big databases of uncertain data. Such a big data analysis and mining process can be sped up. The resulting (constrained or unconstrained) frequent patterns mined from big databases provide users with new insights and a sound understanding of users' patterns. Such knowledge is useful is many real-life information science and technology applications.

INTRODUCTION

Advancements in the field of information science and technology enables users to collect or generate high volumes of valuable data of different levels of veracities —such as streams of banking, financial, and shopper market basket data—at high velocities from wide varieties of data source in various real-life business, engineering, scientific applications in modern organizations and society. Embedded in these *big data* (Madden, 2012; Leung, 2015) is implicit, previously unknown, and potentially useful information and knowledge. However, these big data come with volumes beyond the ability of commonly-used software to capture, manage, and process within a tolerable elapsed time. Hence, new forms of information science and technology—such as *big data analysis and mining*—are needed to process and analyze these big data so to as enable enhanced decision making, insight, knowledge discovery, and process optimization. Over the past few years, algorithms have been proposed for various big data analysis and mining tasks, including clustering (which groups similar data together), classification (which categorizes groups of similar data), outlier detection (which identifies anomalies), and frequent pattern mining (which discovers interesting knowledge in the forms of frequently occurring sets of merchandise items

DOI: 10.4018/978-1-5225-7598-6.ch002

or events). Most of these algorithms use the *MapReduce* model—which mines the search space with distributed or parallel computing (Shim, 2012). Among different big data analysis and mining tasks, this chapter focuses on applying the MapReduce model to big data for the discovery of frequent patterns.

BACKGROUND

Since the introduction of the research problem of *frequent pattern mining* (Agrawal, Imieliński, & Swami, 1993), numerous algorithms have been proposed (Hipp, Güntzer, & Nakhaeizadeh, 2000; Ullman, 2000; Ceglar & Roddick, 2006). Notable ones include the classical Apriori algorithm (Agrawal & Srikant, 1994) and its variants such as the Partition algorithm (Savasere, Omiecinski, & Navathe, 1995). The Apriori algorithm uses a level-wise breadth-first bottom-up approach with a candidate generate-and-test paradigm to mine frequent patterns from transactional databases of precise data. The Partition algorithm divides the databases into several partitions and applies the Apriori algorithm to each partition to obtain patterns that are locally frequent in the partition. As being locally frequent is a necessary condition for a pattern to be globally frequent, these locally frequent patterns are tested to see if they are globally frequent in the databases. To avoid the candidate generate-and-test paradigm, the tree-based FP-growth algorithm (Han, Pei, & Yin, 2000) was proposed. It uses a depth-first pattern-growth (i.e., divide-and-conquer) approach to mine frequent patterns using a tree structure that captures the contents of the databases. Specifically, the algorithm recursively extracts appropriate tree paths to form projected databases containing relevant transactions and to discover frequent patterns from these projected databases.

In various real-life business, engineering, scientific applications in modern organizations and society, the available data are not *precise* data but *uncertain* data (Tong et al., 2012; Leung, Cuzzocrea, & Jiang, 2013; Leung, MacKinnon & Tanbeer, 2014; Jiang & Leung, 2015; Ahmed et al., 2016). Examples include sensor data and privacy-preserving data. Over the past few years, several algorithms have been proposed to mine and analyze these uncertain data. The tree-based UF-growth algorithm (Leung, Mateo, & Brajczuk, 2008) is an example.

With high volumes of big data, it is not unusual for users to have some phenomenon in mind. For example, a store manager is interested in some promotional items. Hence, it would be more desirable if data mining algorithms return only those patterns containing the promotional items rather than returning all frequent patterns, out of which many may be uninteresting to the store manager. It leads to *constrained mining*, in which users can express their interests by specifying constraints and the mining algorithm can reduce the computational effort by focusing on mining those patterns that are interesting to the users.

Besides the aforementioned algorithms discover frequent patterns *in serial*, there are also *parallel and distributed* frequent pattern mining algorithms (Zaki, 1999). For example, the Count Distribution algorithm (Agrawal & Shafer, 1996) is a parallelization of the Apriori algorithm. It divides transactional databases of precise data and assigns them to parallel processors. Each processor counts the frequency of patterns assigned to it and exchanges this frequency information with other processors. This counting and information exchange process is repeated for each pass/database scan.

As we are moving into the new era of big data, more efficient mining algorithms are needed because these data are wide varieties of valuable data of different veracities with volumes beyond the ability of commonly-used algorithms for mining and analyzing within a tolerable elapsed time. To handle big data, researchers proposed the use of the *MapReduce programming model*.