Big Data Mining and Analytics With MapReduce

Carson K. Leung
https://orcid.org/0000-0002-7541-9127
University of Manitoba, Canada

INTRODUCTION

Big Data and *machine learning* are driving Industry 4.0, which is also known as the Fourth Industrial Revolution. Note that the (First) Industrial Revolution transformed manual production to machine production from the late 18th to mid-19th century. The Technological Revolution, which was also known as the Second Industrial Revolution, further industrialized and modernized the industry from the late 19th century to early 20th century through technological advancements and standardization, installations of extensive railroad and telegraph networks, as well as electrification. The Digital Revolution, which was also known as the Third Industrial Revolution, shifted from mechanical and analogue electronic technology to digital electronics, computing and communication technologies in the late 20th century. Now, Big Data have become one of the greatest sources of power in the 21st century, and they have become a critical part of the business world and daily life. In the current era of Big Data, numerous rich data sources are generating huge volumes of a wide variety of valuable data at a high velocity. These Big Data can be of different levels of veracity: They are precise, whereas some others are imprecise and uncertain. Embedded in these Big Data are implicit, previously unknown and potentially useful information and knowledge. This calls for data science, which makes good use of *Big Data mining* and analytics, machine learning, mathematics, statistics and related techniques to manage, mine, analyze and learn from these Big Data to discover hidden gems. This, in turn, may maximize the citizens' wealth and/or promote all society's health. As one of the important Big Data mining and analytics tasks, frequent pattern mining aims to discover interesting knowledge in the forms of frequently occurring sets of merchandise items or events. For example, patterns discovered from business transactions may help reveal shopper trends, which in turn enhances inventory, minimizes customers' cost, and maximizes citizens' wealth. As another example, patterns discovered from health records may help reveal important relationships associated with certain diseases, which in turn leads to improve and promote all society's health. To mine and analyze huge volumes of Big Data in a scalable manner, several algorithms have been proposed that use the MapReduce model—which mines the search space with distributed or parallel computing—for different Big Data mining and analytics tasks. This encyclopedia article covers *Big Data mining and analytics with high performance computing* (*HPC*) and focuses on *frequent pattern mining from Big Data with MapReduce*.

DOI: 10.4018/978-1-7998-9220-5.ch010

BACKGROUND

Since the introduction of the research problem of *frequent pattern mining* (Agrawal et al., 1993), numerous algorithms have been proposed (Hipp et al., 2000; Ullman, 2000; Ceglar & Roddick, 2006; Aggarwal et al., 2014; Alam et al., 2021, 2022; Chowdhury et al., 2022). Notable ones include the classical Apriori algorithm (Agrawal & Srikant, 1994) and its variants such as the Partition algorithm (Savasere et al., 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 Frequent Pattern-growth (FP-growth) algorithm (Han et al., 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.

For different real-life business, engineering, healthcare, scientific, and social applications and services in modern organizations and society, the available data are not necessarily *precise* but *imprecise or uncertain* (Leung et al., 2014; Cheng et al., 2019; Rahman et al., 2019; Davashi, 2021; Li et al., 2021). Examples include sensor data and privacy-preserving data (Chen et al., 2019; Li & Xu, 2019; Eom et al., 2020; Olawoyin et al., 2021; Jangra & Toshniwal, 2022). Over the past decade, several algorithms have been proposed to mine and analyze these uncertain data. The tree-based UF-growth algorithm (Leung et al., 2008) is an example.

When handling huge volumes of Big Data, it is not unusual for users to have some phenomenon in mind. For example, a manager in an organization 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 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.

In addition to the aforementioned algorithms that 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 moves 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*.