Journal of Process Control 68 (2018) 1–13

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Big data quality prediction in the process industry: A distributed parallel modeling framework

Le Yao, Zhiqiang Ge*

State Key Laboratory of Industrial Control Technology, Institute of Industrial Process Control, College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, PR China

ARTICLE INFO

Article history:
Received 6 February 2017
Received in revised form
29 December 2017
Accepted 9 April 2018

Keywords:
Distributed modeling
MapReduce framework
Parallel computing
Quality prediction
Big data analytics

ABSTRACT

With the ever increasing data collected from the process, the era of big data has arrived in the process industry. Therefore, the computational effort for data modeling and analytics in standalone modes has become increasingly demanding, particularly for large-scale processes. In this paper, a distributed parallel process modeling approach is presented based on a MapReduce framework for big data quality prediction. Firstly, the architecture for distributed parallel data modeling is formulated under the MapReduce framework. Secondly, a big data quality prediction scheme is developed based on the distributed parallel data modeling approach. As an example, the basic Semi-Supervised Probabilistic Principal Component Regression (SSPPCR) model is deployed to concurrently train a set of local models with split datasets. Meanwhile, Bayesian rule is utilized in a MapReduce way to integrate local models based on their predictive abilities. Two case studies demonstrate the effectiveness of the proposed method for big data quality prediction.

1. Introduction

For the process industry, both production throughput and product quality are highly related to the profitability, as well as process safety and energy-saving [1,2]. Advanced process/quality monitoring and control techniques have been developed to meet those requirements of modern process industries [3]. One of the prerequisites to achieve these targets is to obtain the value of key process variables in real-time. However, the quality-related variables are typically difficult-to-measure variables like concentrations and melt indices, which are mostly determined through offline laboratory analyses or expensive analyzers [4,5]. Apparently, these methods are time-consuming, high-costly and may introduce a large measurement delay, which can hardly meet the demands of process control and monitoring.

In the last decades, due to the progress of Distributed Control Systems (DCS), large amounts of process data have been collected. Meanwhile, data-driven inferential models have been developed to provide real-time estimations for quality variables [6]. Instead of utilizing the complex process mechanism, most data-driven models concern how to deeply mine the implicit information of datasets through machine learning algorithms. Thus, it is more convenient for data-driven methods to establish the process model along with the adaptation measurements. With the rapid development of data-based techniques, data-driven methods are becoming the mainstream for process modeling. In particular, principal component regression (PCR) [7] and partial least squares (PLS) [8] are two of the most widely used linear data-driven modeling methods in the past years. In addition, data-based nonlinear modeling methods like Deep Learning methods (DBN, Auto-encoder) [9,10] and Kernel methods (KPLS, KPCA, SVM) [11–13] also have been used for nonlinear process quality prediction.

In recent years, newly developed technologies like Internet of things, wireless communications and data acquisition systems have been applied in the process industry, a huge amount of data have been collected and stored in the industrial database [14]. Accordingly, the whole industry has been overwhelmed with the increasing data size and has certainly entered the era of big data. However, all involved traditional methods of local data storage, centralized data pre-processing and modeling are encountering a predicament with low computation efficiency or out-of-memory [15,16]. To solve this problem and provide fast and cost-effective solutions, a parallel and distributed system is needed. Through splitting the huge dataset into several distributed blocks, the processing of data can be conducted concurrently. Under the distributed modeling approach, every block of data can be utilized to train a local model. Then local models can be integrated through a model fusion mechanism to get the global model of the whole dataset [17]. The parallel and distributed modeling strategy takes full advantage of the distributed computing resources and turns

* Corresponding author.
E-mail address: gezhiqiang@zju.edu.cn (Z. Ge).

https://doi.org/10.1016/j.jprocont.2018.04.004
0959-1524/

the heavy computation burden into parallelized small-scale processing, which shows a great potential in big data modeling and analytics [18,19]. As the most popular distributed and parallel computing system, Hadoop MapReduce has been widely utilized to deal with big data application fields that the standalone machine cannot accomplish, such as in biological information [20], text processing [21], etc. On one hand, Hadoop Distributed File System (HDFS) offers a high-reliability and fault-tolerant storage strategy for large-scale datasets. On the other hand, using a large amount of independent computing with resource scheduling, MapReduce is presented to provide a simplified distributed data processing platform for large-scale datasets on a cluster of computer nodes [22]. To deal with the current issue of quality prediction with large-scale datasets efficiently, the traditional modeling algorithm can be deployed on the MapReduce computing framework [23–25]. However, some issues need to be re-considered. First, the data partitioning and preprocessing should be conducted in a parallel manner. Second, the distributed model should be designed so that the entire modeling procedure can be assigned to the MapReduce compute cluster. Third, the local models on the MapReduce cluster should be properly fused to provide a whole predictive model for quality prediction.

In this paper, a distributed and parallel modeling framework based on MapReduce is presented to solve the quality prediction issue under very large scale of process data. Firstly, the huge training dataset is split into several distributed compute nodes and managed by the HDFS system in a reliable mode. Then the preprocessing of data is conducted in the cluster within the MapReduce tasks. According to the Map and Reduce programming functions, the quality prediction model is then deployed on the framework. In particular, a widely applied quality prediction model, Semi-supervised Principal Component Regression (SSPPCR) is implemented as an example. Meanwhile, the EM iteration based on MapReduce is proposed for parameter identification. Through concurrently computing on the cluster, a set of local models are obtained in a much more efficient way and the memory resident issue could be significantly alleviated. Subsequently, the Bayesian rule is introduced under the MapReduce framework to combine the local models into a global model of the entire dataset. Therefore, the big data quality prediction issue which should be originally resort to some high-level workstations can be effectively transformed into a series of simple and repetitive tasks which can also be easily realized on several connected personal computers.

The remainder of this paper is organized as follows. In Section 2, an architecture based on MapReduce for distributed modeling is presented. Then, distributed modeling and model fusion on MapReduce framework will be discussed in detail in Section 3. In Section 4, the proposed distributed modeling strategy is implemented on a numerical example and a real industrial process for quality prediction, respectively. Finally, conclusions are made.

2. Distributed parallel modeling architecture based on MapReduce

In recent years, more and more research efforts have been devoted toward developing system for large scale datasets [22]. The big data issues are commonly partitioned into several subtasks and processed on a cluster of computing nodes concurrently. To realize the distributed and parallel modeling for quality prediction with large scale dataset, a MapReduce framework is utilized for data storage and data processing. In this section, a brief introduction of the MapReduce framework is firstly given and then presented in detail for the architecture of a general MapReduce-based process quality prediction system.

2.1. Overview of MapReduce framework

The MapReduce framework, firstly proposed by Google, is a programming platform for modeling and analyzing massive amount of data in a distributed cluster of computing nodes. The computing nodes in MapReduce contain one Masternode (Namenode) and several Slavenodes (Datanode). The Masternode takes in charge of the file system metadata, and provides management and control services, and the Slavenodes provide block storage and computing services. Meanwhile, the Google File System (GFS) that underlies MapReduce provides efficient and reliable distributed data storage required for applications involving large datasets [26]. The Apache Hadoop project is the most popular and widely used open-source implementation of Google's MapReduce writing in java for reliable, scalable, distributed computing [27]. Hadoop can automatically deal with data splitting, parallel task scheduling/monitoring, parallel compute node communication management and also provide data redundancy and fault tolerance mechanisms, which means that the developers only need to focus on the explicit expressions of two functions: Map and Reduce. Map is a transformation step, in which individual input records are processed in parallel. Reduce is a summarization step, in which all associated records are processed together by a single entity.

In each Slavenode, the computation with respect to Map and Reduce phases both take data structure in the organized form of <key, value> pairs, following the steps as:

Map: $< key1, value1 > \rightarrow list < key2, value2 >$

Reduce: $< key2, list\ (value2) > \rightarrow list < key2, value3 >$

In Map function, each Slavenode takes the individual input < key1, value1 > pairs to produce a list of intermediate < key2, value2 > pairs. Then the MapReduce system shuffles and sorts those intermediate results by lists of those same-key pairs, and the shuffled lists of pairs are grouped by the specific keys denoted as < key2, list (value2) > and passed to the Reduce function. Finally, the Reduce function takes the pairs to compute the expected < key2, value3 > pair lists. For illustration, the MapReduce-based distributed computing can be seen in Fig. 1. Through comparing with the traditional standalone computing flowchart, the MapReduce can effectively conduct the computing in a parallel manner for big data.

2.2. Distributed parallel modeling framework for quality prediction

Based on the MapReduce framework, Fig. 2 shows an architecture for the process quality prediction system. In Step 0, according to the distributed quality prediction algorithms, a driver will be started to handle the task of modeling. Then the process data will be partitioned into several splits with reasonable size and then stored in the file system (see Step 1.1). Meanwhile, the driver will manage a mapping of files in Step 1.2. After that, processors will handle management of Mappers and Reducers. The Map task will be sent to the mappers (see Step 2.1). Corresponding algorithms and split data will also be received by Mappers (see Step 2.1.1 and 2.1.2). After the tasks are executed in the parallel Mappers, the intermediate results will be stored in a local file system for Reducers (see Step 2.1.3). The Reducers will be activated by the driver to process these local models for generating a global model (see Step 2.2). In this phase, Reducers use the model fusion algorithm and local models to produce a global model for quality prediction (see Step 2.2.1, 2.2.2 and 2.2.3). Finally, the global model is formed and returned for quality prediction (see Step 2.3 and 3). In order to clearly describe the sequence of operations, these steps are listed in Table 1 with explanations.

The whole process of modeling in the quality prediction system can be divided into two parts: the data storage part and the data processing part, as shown in Fig. 3. There are two kinds of