Regular paper | Journal of the KIECS. pp. 161-166, vol. 17, no. 1, Feb. 28. 2022, t. 111, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2022.17.1.161

SparkR을 이용한 R 기반 빅데이터 분석의 분산 처리

류우석*

Distributed Processing of Big Data Analysis based on R using SparkR

Woo-Seok Ryu*

요 약

본 논문에서는 데이터 분석 도구인 R을 이용하여 빅데이터 분석을 수행할 때 발생하는 문제점을 분석하고, 빅데이터의 분산 처리를 효과적으로 지원하는 스파크와 R을 연계한 SparkR을 이용한 분석의 유용성을 제시하고자 한다. 먼저, 대량의 데이터를 로딩하고 연산을 수행할 때 발생하는 R의 메모리 할당 문제점과 R과 비교한 SparkR의 특징 및 프로그래밍 환경을 분석한다. 그리고, 선형 회귀 분석을 각각의 환경에서 수행할 때의 실행 성능을 비교 분석한다. 분석 결과 SparkR을 통해 추가적인 언어 학습 없이도 R을 그대로 이용하여 데이터 분석에 활용할 수 있음을 보였으며, SparkR을 이용하여 R로 작성된 코드를 클러스터 내 노드 수의 증가에 따라 효과적으로 분산 처리할 수 있었다.

ABSTRACT

In this paper, we analyze the problems that occur when performing the big data analysis using R as a data analysis tool, and present the usefulness of the data analysis with SparkR which connects R and Spark to support distributed processing of big data effectively. First, we study the memory allocation problem of R which occurs when loading large amounts of data and performing operations, and the characteristics and programming environment of SparkR. And then, we perform the comparison analysis of the execution performance when linear regression analysis is performed in each environment. As a result of the analysis, it was shown that R can be used for data analysis through SparkR without additional language learning, and the code written in R can be effectively processed distributedly according to the increase in the number of nodes in the cluster.

키워드

Big Data, Data Science, Distributed Processing, Programming Language, SparkR
빅데이터, 데이터 과학, 분산 처리, 프로그래밍 언어, SparkR

Ⅰ. 서 론

데이터 과학(Data Science)은 정형 및 비정형 데이터로부터 새로운 지식과 의사결정을 위한 통찰을 이끌어내기 위해 각종 통계학 및 정보학적 기법을 활용하는 분야로서, 최근 빅데이터와 인공지능으로 대변되는 머신러닝의 발전과 함께 다양한 산업 분야에서 적용되고 있다[1]. 데이터 과학 분야는 수학, 통계학, 컴

* 교신저자 : 부산가톨릭대학교 병원경영학과
• 접 수 일 : 2021. 11. 29
• 수정완료일 : 2022. 01. 08
• 게재확정일 : 2022. 02. 17
• Received : Nov. 29, 2021, Revised : Jan. 08, 2022, Accepted : Feb. 17, 2022
• Corresponding Author : Woo-Seok Ryu
Dept. of Health Care Management, Catholic University of Pusan,
Email : wsryu@cup.ac.kr

퓨터공학, 정보공학 등 다양한 분야를 망라하는 학제간 융합분야임에 따라 이를 다루기 위한 방법론도 매우 다양한 특성을 보인다. 데이터 과학을 위한 도구로서 C++, 자바와 같은 전통적인 프로그래밍 언어와 SQL과 같은 데이터베이스 언어, 그리고 통계 분야에서 주로 다루는 SAS, R과 같은 다양한 언어들이 활용되고 있다[2]. 또한, 텐서플로와 같은 머신러닝 도구에서는 파이썬(Python)이 사용되고 있음에 따라 다양한 데이터 분석 환경에 따라 적절한 언어의 선택이 요구되고 있다.

데이터 과학 분야에서 최근 가장 많이 사용되고 있는 언어는 파이썬과 R이다[1]. 파이썬은 범용 프로그래밍 언어로서 간결한 문법을 장점으로 머신러닝, 그래픽, 웹 등 다양한 분야에서 적용되고 있다. R은 데이터 과학에서 대두되고 있는 인터프리터 형식의 언어로서 데이터의 통계 분석 및 데이터의 가시화에 특화된 언어이다[3]. 이에, 프로그래밍을 전문적으로 학습하지 않은 사회과학, 의료보건 등의 분야에서 데이터 과학을 위해 보다 수월하게 이용할 수 있는 특성이 있다[4].

R은 앞에서 언급한 많은 장점에도 불구하고 연산을 수행하기 위해서는 사전에 메모리에 데이터를 적재해야 하므로 대규모 데이터셋을 다루기가 어려운 문제가 있다[5]. 이에, 하둡(Hadoop), 스파크(Spark)와 같은 빅데이터 분산처리 플랫폼이 각각 RHadoop, SparkR을 통해 R 언어로 작성된 코드를 분산처리할 수 있도록 지원하고 있다. 그중 스파크는 인메모리 기반 병렬분산 처리를 지원함으로써 하둡과 비교하여 월등히 우수한 성능을 보이고 있다[6]. 또한 MLlib을 통해 다양한 머신러닝 API를 지원한다[7][8]. 이에, 본 연구에서는 R의 편의성을 그대로 유지하면서 대규모 데이터셋을 다룰 수 있는 SparkR의 유용성을 분석하기 위해 R을 이용한 데이터 분석 환경과 스파크에서 제공하는 SparkR을 이용한 환경의 프로그래밍 모델과 성능을 비교하고자 한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 R과 SparkR의 특성을 비교하고, 3장에서는 두 개발 환경의 수행 성능을 선형회귀모델 학습 시간을 통해 비교한다. 그리고, 4장에서 결론 및 향후 연구를 기술한다.

1) Best 11 Data Science Programming Languages in 2021 https://medium.com/javarevisited/best-11-data-science-programming-languages-in-2020-122a7ea2bb63

II. R과 SparkR의 비교

R은 선형회귀 모델(lm), 일반화 선형 모델(glm), 랜덤포레스트(randomForest 패키지), 신경망(nnet 패키지) 등의 데이터 분석 및 머신러닝 기능들을 다양한 통계 패키지에서 제공하는 함수를 통해 지원한다. 이때, 공통으로 사용되는 데이터 타입은 테이블 데이터를 저장하는 data.frame 타입이다. R에서는 내부 메모리에 data.frame 타입의 데이터를 모두 로딩한 후에 학습을 수행한다. 그러므로, PC의 메모리 용량에 데이터셋 크기의 제한을 받게 되며 메모리 용량을 초과하는 데이터에 대한 처리가 어려운 문제가 있다[5].

아파치 스파크는 대규모 데이터의 고속 처리를 위한 오픈소스 기반의 통합 분석 엔진이다. 단일 노드 컴퓨터 및 클러스터를 지원하며 하둡의 맵리듀스 프레임워크와 달리 메모리 기반(In-Memory) 분산 처리를 지원하므로 맵리듀스(MapReduce)와 비교하여 수십 배 이상 빠른 성능을 보이고 있다. 스파크의 분산처리는 자체적인 분산 클러스터를 지원할 뿐만 아니라 기존의 분산 클러스터들, 즉 하둡의 얀(Yarn), 아파치 메소스(Apache Mesos) 등도 지원하므로 분산처리의 호환성이 높으며, 하둡 분산 파일시스템(HDFS), HBase 등의 다양한 분산 데이터 저장소를 지원한다[9][10].

스파크의 개발 환경으로는 자체적인 프로그래밍 언어로 자바 가상머신 환경에서 동작하는 스칼라(Scala)를 사용한다. 그 외에도 기존의 데이터 분석에서 많이 사용하는 파이썬과 R을 지원하고 있으며 표준 SQL을 이용한 데이터 분석이 가능하도록 Spark SQL을 지원한다. 그중, R을 지원하기 위한 SparkR(R on Spark)은 프로그래밍 언어인 R을 이용하여 스파크를 사용할 수 있도록 제공하는 프로그래밍 인터페이스이다. SparkR은 R로 작성된 사용자 코드를 스파크의 구조적 연산으로 변환하여 수행하므로 구조적 API를 이용하는 경우 스칼라로 작성된 코드와 동일한 성능을 보일 수 있다[5]. R과 SparkR의 가장 큰 차이점은 테이블 형태의 데이터를 저장하는 데이터프레임 타입의 처리 방법이다. SparkR에서 지원하는 SparkDataFrame은 R의 data.frame과 개념적으로 동일하며 R의 로컬 데이터 프레임은 물론 하이브의 테이블, 구조화된 데이터 파일과 같은 분산 데이터를 지

원하는 특징이 있다.

데이터 분석의 관점에서 볼 때 R에서 지원하는 다양한 머신러닝 관련 API는 SparkR의 MLlib에 대응된다[8]. SparkR에서는 스파크 세션을 이용하여 MLlib을 이용할 수 있으며, 일반화 선형 모델(glm), k-평균 모델, 로지스틱 회귀모델, 랜덤 포레스트모델 등의 다양한 알고리즘을 그대로 활용할 수 있다.

III. 실행 성능 비교

3.1 선형 회귀 모델 소스 코드 비교

이 장에서는 대규모 데이터 분석에서 SparkR의 유용성을 평가하기 위해 R과 SparkR의 프로그램 코드 및 처리 성능을 비교하고자 한다. R과 SparkR에서 선형 회귀 모델을 학습하기 위한 기본 코드는 그림 1과 같이 두 가지 모두 간단하게 구현할 수 있다. R에서는 데이터 파일을 읽는 read.csv() 함수와 선형회귀 모델을 학습하는 lm() 함수를 호출하는 코드로 구현이 가능하다. SparkR에서는 선형 회귀 모델 학습을 위해 csv 파일을 libsvm 포맷의 데이터형식으로 사전에 변환하여 HDFS에 저장하는 작업이 필요하였으나, 실제 변환 작업 후에는 libsvm 포맷의 데이터를 HDFS에서 읽는 read.df() 함수와 MLlib에 포함된 선형회귀 모델 학습 함수인 LinearRegression함수를 호출하는 코드로 구현이 가능하다. 이에 두 환경의 프로그래밍 코드는 매우 유사하다고 볼 수 있다.

```
1  # pseudo code of R
2  dataset <- read.csv(filename, header=F);
3  model <- lm(formula, dataset);
4
5  # pseudo code of SparkR
6  dataset <- read.df(filename, source="libsvm")
7  model = spark.lm(dataset, formula);
8
```

그림 1. 선형 회귀 모델의 학습 코드
Fig. 1 Program codes for the linear regression

3.2 성능 비교를 위한 실험 환경

R과 SparkR의 실행 성능을 비교하기 위한 환경으로 2-코어 인텔 펜티엄 프로세서, 4GB 메모리, 500GB 저장장치로 구성된 리눅스 PC 5대를 클러스터로 구성하였다. 운영체제는 우분투 16.04이며, R은 4.1.2버전을 사용하였다. SparkR의 구동을 위한 스파크 버전은 3.2.0을 이용하였으며 하둡 2.7.4에 내장된 분산 클러스터 매니저인 YARN을 스파크의 클러스터 매니저로 설정하였다. 이때 분산 파일시스템은 하둡에 포함된 HDFS를 이용하였다. 데이터 블록의 크기, 중복 계수 등의 설정은 스파크와 하둡의 기본 설정을 그대로 이용하였다.

실험 데이터셋은 국민건강보험공단 홈페이지에서 제공하고 있는 진료내역정보 데이터셋(2018년)을 이용하였다[2]. 이 데이터는 2018년 진료이력이 있는 국민건강보험 가입자 100만명의 약 1297만 건의 진료내역으로 구성되어 있다. 본 연구에서는 데이터셋의 규모별 R과 SparkR의 처리성능을 비교하기 위해 데이터셋을 인원수로 나누어서 표 1과 같이 5개의 데이터셋으로 생성하였다.

표 1. 실험 데이터셋
Table 1. Dataset for the experiment

Dataset	Number of Person (k)	CSV data size (MB)	libsvm data size (MB)
200k	200	223	362
400k	400	449	729
600k	600	674	1,060
800k	800	899	1,420
1000k	1,000	1090	1,770

본 데이터셋을 이용한 분석 프로그램은 그림 1을 그대로 이용하였다. 두 프로그램에 적용한 선형회귀 모델은 시스템의 부하를 주기 위한 목적으로 총 19개의 변수 중 내원일수를 종속변수로 설정하고 나머지 18개의 변수 중 명목변수로 측정된 주상병 코드, 부상병 코드를 제외한 16개의 변수를 독립변수로 설정하였다.

R은 5대의 노드 중 마스터 노드에서 실행시켜 프로그램의 수행 시간을 측정하였으며, SparkR은 분산처리의 성능을 확인하기 위하여 마스터노드를 제외한 4개의 워커 노드를 하나씩 추가하면서 시간을 측정하였다. 모든 프로그램은 동일한 환경에서 5회씩 실행 후 평균 실행 시간을 산출하였다.

2) 국민건강보험공단 공공데이터 개방서비스, 홈페이지: https://nhiss.nhis.or.kr/op/it/index.do

3.3 실험 결과

표 2는 R와 SparkR 각각에서 선형 회귀 모델의 학습 시간을 비교한 표이다. 이때 R은 단일 노드에서 수행하였으며, SparkR은 마스터 노드와 하나의 워커 노드에서 수행한 결과이다. R의 경우 모델 학습을 위해 모든 데이터셋을 메모리에 로딩해야 하므로 400k 데이터셋과 비교하여 볼 때 600k에서 3배 이상의 시간이 소요되는 것을 확인할 수 있으며, 또한 800k와 1000k 데이터셋은 lm()함수의 수행 도중에 메모리 할당 오류가 발생하여 정확한 시간을 측정할 수 없었다. 반면에 SparkR은 데이터량의 증가에 따른 처리 시간이 매우 선형적으로 증가하는 것을 확인할 수 있으며, R에서와 같은 메모리 오류 또한 발생하지 않음을 확인할 수 있다. 실험 결과 SparkR에서의 1000k 데이터셋 처리 성능이 R에서의 600k 데이터셋 처리 성능보다 높을 정도로 SparkR의 성능이 월등히 우수함을 확인할 수 있다. 다만, 두 경우 모두 200k 데이터셋의 처리 시간은 상대적으로 편차가 크게 발생하였는데 이는 반복 실험 횟수를 더욱 늘려서 정확한 시간을 산출하여 결과를 비교할 필요가 있다.

표 2. R과 SparkR의 성능 비교
Table 2. Performance comparison of R and SparkR

Dataset	R (second, M±SD)	SparkR (s, M±SD)
200k	39.3±7.0 s	43.5±10.2
400k	56.4±3.1 s	67.8±1.0
600k	175.3±8.4 s	96.7±2.3
800k	stopped with error	127.6±4.4
1000k	stopped with error	164.2±3.9

그림 2는 클러스터에서 워커 노드의 수를 늘려가면서 SparkR의 모델 학습 시간을 측정한 결과이다. 모든 클러스터 구성에 대해 데이터 크기에 따른 실행 시간의 증가가 선형적인 것을 확인할 수 있는데 이를 통해 SparkR은 클러스터의 규모에 따라 적절하게 병렬분산 처리가 수행됨을 확인할 수 있다. 워커 노드 수의 증가에 따른 처리 시간의 감소도 매우 선형적인 결과가 나왔으나, 워커 노드의 수가 3개일 때와 4개일 때의 수행 성능은 상대적으로 차이가 크지 않았다. 이는 800k 데이터셋과 1000k 데이터셋이 HDFS에 저장되는 블록의 수가 각각 8, 9개로 크게 차이가 발생하지 않음에 따라 데이터 블록의 수가 2배씩 증가하는 다른 데이터셋에 비해 성능 차이가 크지 않은 것으로 판단된다.

실험결과 SparkR에서 400k 데이터셋을 2개의 워커 노드에서 처리하는 시간은 600k 데이터셋을 3개의 워커 노드에서 처리하는 시간과 유사하였는데, 이는 800k 데이터셋을 4개의 워커 노드에서 처리하는 시간과도 거의 유사하였다. 표 2과 비교하여 볼 때 200k 데이터셋을 R에서 처리하는 시간과 거의 유사하였으며, 그 결과 R을 이용하여 대량의 데이터셋을 분석하고자 하는 경우 SparkR을 이용하면 워커 노드 수를 증가시키는 만큼 효과적으로 처리시간을 단축할 수 있음을 확인할 수 있다.

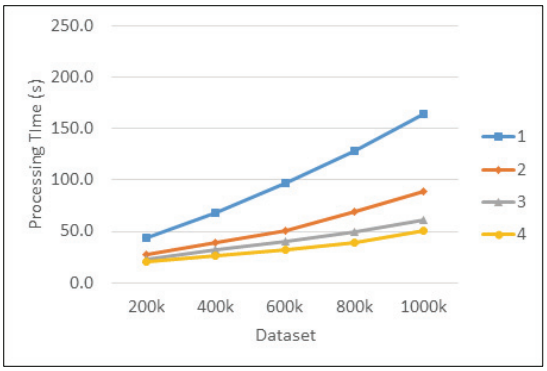

그림 2. 워커 노드의 수에 따른 SparkR의 분산 처리 성능 비교
Fig. 2 Performance comparison of distributed processing SparkR according to the number of worker nodes

Ⅳ. 결 론

본 논문에서는 R을 이용하여 대량의 데이터를 분석할 때 R이 가지는 문제점을 분석하고 R을 이용하여 빅데이터 분석을 분산 처리하기 위한 SparkR의 특징과 성능을 비교 분석하였다. 연구 결과 R은 데이터 분석의 성능이 PC의 메모리 사양에 제한되었지만, SparkR을 이용하면 단일 시스템에서도 대량의 데이터를 효과적으로 처리할 수 있었으며, 클러스터로 구성하는 경우 워커 노드의 수의 증가에 따라 실행 성

능도 선형적으로 증가시킴을 확인할 수 있었다. 또한, 개발 환경 또한 동일한 R 언어를 사용하므로 스파크 및 분산 클러스터의 설정만 추가로 수행하면 별도의 언어를 익힐 필요 없이 기존의 R을 이용해 병렬 분산 분석이 가능함을 확인할 수 있었다. 본 연구에서는 선형 회귀 모델의 학습을 주로 비교하였는데, 데이터 분석에 필요한 다양한 학습 모델들을 이용한 R과 SparkR의 비교가 필요하며, 데이터 분석에 필수적으로 요구되는 데이터 전처리 코드의 특성 등의 비교 분석도 추가적으로 수행할 필요가 있다.

감사의 글

이 논문은 2019년도 부산가톨릭대학교 교내연구비에 의하여 연구되었음

References

[1] A. Rabasa and C. Heavin, *An Introduction to Data Science and its Applications, Data Science and Productivity Analytics.* Berlin: Springer Cham, 2020, pp. 57-81.

[2] Y. Lim and K. Kim, "Methods to propel Tourism of Yeosu City Using Big Data," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 15, no. 4, Aug. 2020, pp. 739-746.

[3] K. Goztepe, "De Facto Language of Data Science: The R Project," *J. of Management and Information Science*, vol. 4, no. 4, Dec. 2016, pp. 104-107.

[4] M. Cho, "A Comparative Study on the Accuracy of Important Statistical Prediction Techniques for Marketing Data," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 14, no. 4, Aug. 2019, pp. 775-780.

[5] B. Chambers and M. Zaharia, *Spark: The definitive Guide: Big data processing made simple.* Newton, MA, USA: O'Reilly Media, Inc, Feb. 2018.

[6] M. Zaharia, R. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. Franklin, and A. Ghodsi, "Apache spark: a unified engine for big data processing," *Communications of the ACM,* vol. 59, no. 11, Oct. 2016, pp. 56-65.

[7] J. Jang, J. Park, H. Kim, and S. Yoon, "A Comparative Performance Analysis of Spark-Based Distributed Deep-Learning Frameworks," *KIISE(Korean Institute of Information Scientists and Engineers) Trans. Computing Practices*, vol. 23, no. 5, May, 2017, pp. 299-303.

[8] M. Assefi, E. Behravesh, G. Liu, and A. P. Tafti, "Big data machine learning using apache spark MLlib," In *2017 IEEE Int. Conf. on Big Data (Big Data)*, Boston, MA, U.S.A., 2017, pp. 3492-3498.

[9] W. Ryu, "Performance Factor of Distributed Processing of Machine Learning using Spark," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 16, no. 1, Feb. 2021, pp. 19-24.

[10] R. Myung, H. Yu, and S. Choi, "Performance Optimization Strategies for Fully Utilizing Apache Spark," *KIPS(Korea Information Processing Society) Trans. Computer and Communication Systems*, vol. 7, no. 1, Jan. 2018, pp. 9-18.

저자 소개

류우석(Woo-Seok Ryu)

1997년 부산대학교 컴퓨터공학과 졸업 (공학사)
1999년 부산대학교 대학원 컴퓨터공학과 졸업(공학석사)
2012년 부산대학교 대학원 컴퓨터공학과 졸업(공학박사)
2013년~현재 부산가톨릭대학교 병원경영학과 부교수
※ 관심분야 : 의료정보, 빅데이터, 병렬분산 처리, 머신러닝