Regular paper | Journal of the KIECS. pp. 85-90, vol. 18, no. 1, Feb. 28. 2023, t. 117, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2023.18.1.85

스파크에서 스칼라와 R을 이용한 머신러닝의 비교

류우석*

Comparison of Scala and R for Machine Learning in Spark

Woo-Seok Ryu*

요 약

보건의료분야 데이터 분석 방법론이 기존의 통계 중심의 연구방법에서 머신러닝을 이용한 예측 연구로 전환되고 있다. 본 연구에서는 다양한 머신러닝 도구들을 살펴보고, 보건의료분야에서 많이 사용하고 있는 통계 도구인 R을 빅데이터 머신러닝에 적용하기 위해 R과 스파크를 연계한 프로그래밍 모델들을 비교한다. 그리고, R을 스파크 환경에서 수행하는 SparkR을 이용한 선형회귀모델 학습의 성능을 스파크의 기본 언어인 스칼라를 이용한 모델과 비교한다. 실험 결과 SparkR을 이용할 때의 학습 수행 시간이 스칼라와 비교하여 10~20% 정도 증가하였다. 결과로 제시된 성능 저하를 감안한다면 기존의 통계분석 도구인 R을 그대로 활용 가능하다는 측면에서 SparkR의 분산 처리의 유용성을 확인하였다.

ABSTRACT

Data analysis methodology in the healthcare field is shifting from traditional statistics-oriented research methods to predictive research using machine learning. In this study, we survey various machine learning tools, and compare several programming models, which utilize R and Spark, for applying R, a statistical tool widely used in the health care field, to machine learning. In addition, we compare the performance of linear regression model using scala, which is the basic languages of Spark and R. As a result of the experiment, the learning execution time when using SparkR increased by 10 to 20% compared to Scala. Considering the presented performance degradation, SparkR's distributed processing was confirmed as useful in R as the traditional statistical analysis tool that could be used as it is.

키워드
Machine Learning, Healthcare, Scala, SparkR
머신 러닝, 보건 의료, 스칼라, 스파크R

Ⅰ. 서 론

머신러닝(Machine Learning)은 구글에서 발표한 알파고와 그 기반기술인 딥러닝(Deep Learning)을 통해 대외적으로 널리 알려진 인공지능의 한 분야로서, 빅데이터와 함께 4차 산업혁명의 기본이 되는 기술로

* 교신저자 : 부산가톨릭대학교 병원경영학과
• 접 수 일 : 2022. 12. 11
• 수정완료일 : 2023. 01. 11
• 게재확정일 : 2023. 02. 17
• Received : Dec. 11, 2022, Revised : Jan. 11, 2023, Accepted : Feb. 17, 2023
• Corresponding Author : Woo-Seok Ryu
Dept. of Health Care Management, Catholic University of Pusan,
Email : wsryu@cup.ac.kr

각광받고 있다. 2010년 이후 산업의 전 분야에서 머신러닝이 적용되어 이미지 및 음성인식 분야는 물론 자연어 처리 및 음성인식, 텍스트 마이닝 등을 통해 산업의 전 분야에서 활발하게 응용되고 있다.

보건의료분야에서는 IBM의 왓슨을 시작으로 방대한 의학 문헌의 러닝을 통한 암 진단 및 조기진단, 방사선 영상의 러닝을 통한 질병 진단, 환자 데이터 기반 위험 분석, 정밀의료 등의 다양한 분야에서 그 적용범위를 넓혀가고 있다. 의료기관 간 경쟁이 치열해지고 있는 현실에서 최선의 기술을 활용하여 경영환경의 변화를 모색하고 환자 서비스 강화 및 경영환경 개선을 위해서는 보건의료정보의 머신러닝을 통한 새로운 통찰을 끌어내고 이에 기반 한 다양한 서비스의 개발이 필요하다.

머신러닝을 위한 도구는 파이썬, R, 스파크, 머하웃, 텐서플로 등 매우 다양하게 제시되어 있다. 그 중, R은 통계 분석을 목적으로 개발된 인터프리터 언어로 프로그래밍 언어를 전문적으로 배우지 않은 사용자도 명령어 입력을 통해 통계 분석 및 머신러닝을 수행할 수 있는 장점이 있음에 따라 보건의료 분야에서 통계 및 가시화를 목적으로 광범위하게 사용되고 있다[1].

하지만, R은 높은 사용성에도 불구하고 메모리 제한 문제 등으로 인해 빅데이터를 분석하지 못하는 한계가 있다[2]. R이 가지는 처리용량 한계점은 병렬분산 플랫폼인 아파치 스파크(Apache Spark)와 결합을 통해 해결이 가능하다. 스파크는 ML 라이브러리를 이용하여 대규모 데이터에 대한 머신러닝을 분산 환경에서 수행할 수 있는 장점이 있다[3]. 스파크는 내장 프로그래밍 인어인 스칼라 뿐만 아니라 파이썬으로 작성된 코드를 PySpark를 통해 분산 처리할 수 있으며[4], R로 작성된 코드를 수행할 수 있는 sparkR 인터페이스를 제공함으로써 R 사용자가 새로운 언어를 배우지 않아도 R을 이용하여 병렬분산 처리를 수행할 수 있다. 이에 본 연구에서는 R로 작성하여 SparkR을 통해 머신러닝을 수행하는 환경과 내장 언어인 스칼라를 이용하여 머신러닝을 수행하는 환경을 비교 분석함으로써 스파크와 결합하여 머신러닝을 수행하기 위한 도구로서의 R의 효용성을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 머신러닝을 위한 다양한 도구들을 살펴하고, 3장에서는 R과 스칼라 언어를 이용한 선형회귀모델을 분산환경에서 스파크에서 실행할 때의 학습 시간을 비교하여 두 언어의 수행 성능을 분석한다. 그리고, 4장에서는 결론과 향후 연구를 기술한다.

II. 머신러닝 도구 분석

이 장에서는 여러 연구에서 활용되고 있는 다양한 머신러닝 도구들을 비교 분석하고자 한다. 통계분석 관점에서의 기존의 도구인 R과 빅데이터 머신러닝을 위한 도구, 그리고 딥러닝 도구들을 라이브러리, 기능 및 성능 특성을 기준으로 비교하고자 한다.

머신러닝을 위해 보건의료 분야에서 가장 많이 활용하고 있는 도구는 R이다. R은 통계분야에서 오랜 기간 동안 활용되고 있는 데이터 분석 도구로서 통계 분석 및 가시화에서 큰 장점을 가지고 있다. 또한 R은 선형회귀모델, 랜덤포레스트, 신경망 등 다양한 머신러닝 알고리즘을 지원하고 있다. R은 데이터 분석 시 데이터를 메인 메모리로 모두 로딩하여 처리하는 특징이 있는데 이로 인해 시스템의 메모리 용량을 넘어서는 빅데이터는 처리가 어려운 문제가 있다[2].

빅데이터의 머신러닝을 위한 도구로서 빅데이터 처리 플랫폼인 하둡 에코 시스템에서는 두 가지 프레임워크를 제시하고 있다. 첫 번째는 인메모리 기반 빅데이터 통합 처리 플랫폼인 아파치 스파크에서 제공하는 ML 라이브러리(MLlib)를 이용하는 방법이다[3]. 자체 언어인 스칼라 뿐만 아니라 파이썬, R 등의 데이터 분석 언어를 사용할 수 있으며, ML 라이브러리를 통해 분류, 회귀, 의사결정나무, 추천, 군집분석 등 다양한 머신러닝 알고리즘을 지원한다. 그리고, HDFS, HBase 등 다양한 분산 파일시스템에 저장된 대량의 데이터에 접근이 가능한 특징이 있다[5].

두 번째는 머신러닝 라이브러리인 머하웃(Mahout)이다[6]. 이는 맵리듀스를 이용하는 아파치 하둡 위에 위치하고 있으며 유사 데이터들의 분류 및 필터링 분야에 강점을 가지고 있다. 프로그래밍 언어로는 자바와 스칼라를 지원하며 맵리듀스 프레임워크를 이용한 분산 수행이 가능함에 따라 하둡 기반의 머신러닝에 최적화되어 있으나, 인메모리 기반의 연산을 수행하는 스파크에 비해 상대적으로 속도가 느린 단점이 있다.

데이터 과학 분야에서 많이 사용하고 있는 프로그램 도구인 파이썬(Python)은 사이킷런(scikit-learn) 라이브러리를 통해 머신러닝 알고리즘을 활용할 수 있다[7]. 사이킷런은 머신러닝에 필요한 API를 단순화하여 제공하고 있음에 따라 쉽고 효율적인 개발이 가능한 특징이 있다. 변환, 정규화, 스케일링 등의 데이터 전처리 기능 및 교차검증, 분류, 회귀, 클러스터링, SVM 등의 다양한 머신러닝 알고리즘을 지원하는 특징이 있다. 머신러닝 교육에 많이 활용되고 있는 오렌지3는 오픈소스 머신러닝 도구로서 워크플로 생성과 데이터 가시화를 별도의 프로그램 도구 없이 직관적으로 수행할 수 있는 장점이 있다[8]. 이에, 최근 데이터 분석 비전문가 또는 학생들을 중심으로 머신러닝을 학습하는 목적으로 많이 사용되고 있다.

머신러닝 알고리즘 중 딥러닝을 지원하는 프레임워크는 텐서플로, 케라스, 파이토치가 있다. 텐서플로는 텐서(Tensor)와 데이터플로우 그래프 구조를 사용하여 이미지 인식, 반복 신경망 구성 등의 각종 신경망 학습에 사용되고 있다[9]. 대규모 예측 모델 구성에 뛰어난 장점이 있으나 메모리 사용의 효율성은 다소 부족한 단점이 있다. 텐서플로는 자체적인 병렬분산처리 기능을 제공하고 있으며, TensorflowOnSpark를 통해 텐서플로와 스파크를 연동할 수 있는 방법을 제시하고 있다. 케라스(Keras)는 딥러닝 모델 개발을 위한 인터페이스를 단순화하여 제공하는 라이브러리이다[10]. 텐서플로와 비교해서 볼 때 모델 개발에 소요되는 시간을 단축시킬 수 있는 장점이 있으나, 상대적으로 복잡한 모델을 생성하기 어려운 특징이 있다. 텐서플로와 케라스는 연계 지원하고 있으며, 케라스에서 작성한 딥러닝 모델을 텐서플로에서 학습할 수 있도록 지원하고 있다[7].

파이토치(PyTorch)는 페이스북에서 개발한 파이썬 기반 오픈소스 머신러닝 라이브러리이다[11]. 텐서플로와 비교해서 모델을 생성하는 절차가 간단하고 그래프를 동적으로 변화할 수 있으며, 학습 속도도 텐서플로보다 우수한 장점이 있다. 또한, 텐서플로는 유기적인 신경망을 만들기 어려우나 파이토치는 메모리상에서 연산을 수행하면서도 신경망의 규모를 최적으로 바꾸면서 동작할 수 있음에 따라 최근에 많이 활용되고 있는 추세이다.

III. SparkR과 스파크의 성능 비교

3.1 성능 비교 목표 및 대상 설정

이 장에서는 보건의료 빅데이터를 이용한 머신러닝을 수행하는 프레임워크에 대한 수행 성능을 비교하고자 한다. 성능 비교 대상이 되는 프레임워크는 보건의료분야 연구에서 많이 다루는 통계 및 머신러닝 패키지인 R과 빅데이터의 분산 분석에 유용한 스파크 프레임워크이다. 선행연구를 통해 로컬 환경에서 동작하는 R과 비교하여 sparkR이 분산 처리에 장점이 있음을 제시하였으나 sparkR과 스파크 프레임워크에 내장된 언어인 스칼라(scala)와의 비교는 제시되어 있지 않다[2]. 이 장에서는 스파크의 내장 언어인 스칼라를 이용하여 머신러닝을 수행할 때와 비교하여 SparkR의 수행 성능이 어느 정도 차이가 발생하는지를 비교하고자 한다.

3.2 실험 환경 설정

R과 스파크의 머신러닝 성능을 비교하기 위한 실험 환경으로 5대의 노드로 구성된 분산 클러스터를 구성하였다. 각 노드는 2-코어 인텔 펜티엄 프로세서와 4GB 메모리, 500GB의 저장장치로 구성되어 있으며 우분투 16.04 운영체제를 탑재하고 1기가비트 이더넷으로 네트워크를 연결하였다. 소프트웨어 플랫폼은 R의 경우 4.1.2 버전을 사용하였으며 스파크는 3.2.0을 이용하였다. 분산 데이터 저장 및 잡 스케쥴링을 위한 하둡은 2.7.4를 이용하였으며 스파크의 분산 클러스터 매니저는 하둡 YARN을 사용하였다.

본 연구에서 비교할 머신러닝 알고리즘은 보건의료 연구에서 많이 활용하고 있는 선형회귀모델(Linear Regression)이다. 선형회귀모델의 입력 데이터셋은 국민건강보험공단 홈페이지에서 제공하는 공개 데이터셋인 진료내역정보 2018년 자료를 이용하였다. 데이터에 포함된 국민건강보험 가입자 100만명의 2018년도 진료기록 약 1297만건에 대해 본 연구에서는 데이터셋 크기별 성능 비교를 위해 20만명(200k), 40만명(400k), 60만명(600k), 80만명(800k), 그리고 100만명(1000k)의 5개 데이터셋으로 구분하였다. 이때 저장된 데이터의 용량은 각각 362MB, 729MB, 1,060MB, 1,420MB, 1,770MB이다.

학습을 위해 사전에 데이터셋을 libsvm 포맷으로

변환하여 HDFS(Hadoop Distributed File System)에 저장해두었으며, 데이터를 로딩하고 MLlib에 내장된 LinearRegression() 함수를 호출하여 학습을 수행하는 코드를 그림 1과 같이 R과 스칼라를 이용하여 각각 작성하였다. 이때, 작성한 코드는 데이터를 로딩하는 코드와 학습을 수행하는 코드로 최대한 단순화하여 작성하였다. 이때, 선형회귀모델은 총 19개의 변수 중 내원일수를 종속변수로 지정하고 나머지 변수 중 상병코드를 제외한 16개 변수 모두를 독립변수로 지정하였다.

```r
// R code for SparkR linear regression
df <- read.df("/nhis/nhis_libsvm_200k", source="libsvm")
model = spark.lm(df, label ~ features,
                 regParam=0.3, elasticNetParam=0.8)
```

```scala
// scala code for Spark linear regression
import org.apache.spark.ml.regression.LinearRegression

val data = spark.read.format("libsvm") .load(filename)
val lr = new LinearRegression()
        .setMaxIter(10)
        .setRegParam(0.3)
        .setElasticNetParam(0.8)
val lrModel = lr.fit(data)
```

그림 1 선형회귀모델의 R 및 스칼라 학습 코드
Fig. 1 R and scala codes for linear regression

수행 시간을 비교하기 위한 머신러닝 수행 환경으로 스파크에서는 분산 수행 성능을 확인하기 위해 하나의 노드를 마스터 노드로 설정하고 워커 노드는 1개부터 4개까지 클러스터의 규모를 달리하여 스칼라로 작성된 코드와 SparkR로 작성된 코드를 각각 실행하여 그 수행시간을 측정하였다. 두 코드 모두 동일한 조건에서의 성능 비교를 위하여 R 코드는 sparkR을 실행한 후 소스코드를 실행하였으며 스칼라 코드는 spark-shell를 실행한 후 프로그램을 실행하였다.

3.3 성능 비교 결과

그림 2는 클러스터의 워커 노드 수를 1에서 4까지 달리하면서 스칼라와 sparkR의 수행 시간을 측정한 결과이다. 400k와 1000k 두 데이터셋에 대해 sparkR의 수행 시간이 스칼라의 수행 시간보다 약 10~20% 증가하였으며, 워커 노드의 수가 증가하더라도 sparkR과 스칼라가 동일하게 분산 처리가 됨을 확인할 수 있다. sparkR의 수행 시간 증가분을 오버헤드라고 할 때 sparkR이 스칼라와 비교하여 가지는 오버헤드가 데이터셋, 워커 노드의 수와 상관없이 어느 정도 일정하게 유지되는 것을 확인할 수 있다. 그림에는 표시되어 있지 않지만 나머지 데이터셋에서도 동일한 결과를 확인할 수 있었다.

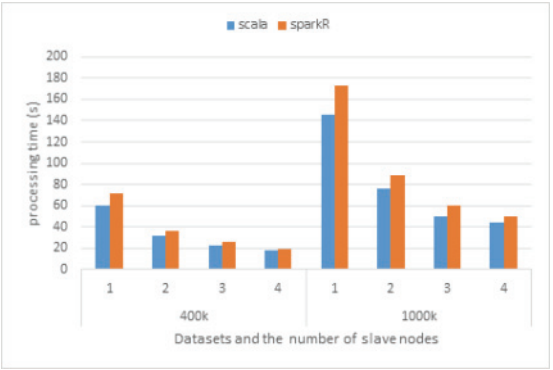

그림 2. 스칼라와 sparkR의 처리시간 비교
Fig. 2 Comparison of processing time between scala and sparkR

그림 3은 sparkR의 수행과정에서 구체적으로 오버헤드가 발생하는 부분을 확인하기 위해 4개의 워커 노드로 구성된 분산 클러스터 환경에서 데이터 로딩 시간과 학습 시간을 구분하여 스칼라와 sparkR의 처리 시간을 비교한 그림이다. 실험 환경에서 데이터셋의 로딩시간은 두 개발환경이 거의 동일하게 측정되었으며, 학습 단계에서 약 20% 정도의 수행 시간 차이가 발생하였다.

학습 시간의 성능 차이는 스파크의 구동방식의 차이로 인한 것[12]으로 스칼라는 spark-shell을 통한 클라이언트 모드로 분산 구동되며, sparkR은 내부적으로 spark-submit이 제출됨에 따라 클러스터 모드로 수행되므로 워커 노드의 프로세싱 코어를 하나 더 소모하므로 그에 따라 학습 시간의 차이가 발생한 것으로 추정된다. 하지만, R도 스파크의 MLlib을 내부적으로 그대로 사용함에 따라 비교적 적은 오버헤드 수준에서 분산처리가 될 수 있으며, 클러스터 규모가 충분히 큰 상황에서는 R이 스칼라와 동일한 학습 성능을 보일 수 있음을 예상할 수 있다.

그림 3 스칼라와 sparkR의 로딩 및 학습시간 비교
Fig. 3 Comparison of loading and training time between scala and sparkR

Ⅳ. 결 론

본 연구에서는 보건의료정보를 이용하여 머신러닝을 수행하기 위한 다양한 머신러닝 및 딥러닝 도구를 살펴보고, 보건의료분야 통계분석에 많이 활용되고 있는 도구인 R이 가지고 있는 한계점인 대용량 데이터 처리 문제 문제의 해결을 위해 분산처리플랫폼인 스파크와 결합하여 머신러닝을 수행하였을 때의 유효성을 검증하였다. 실험으로 통해 스파크 프레임워크에서 R과 스칼라로 작성된 코드의 학습 성능을 비교한 결과 분산 처리 성능의 차이가 20% 이내임을 확인하였다. 기존의 통계분석 도구인 R에 익숙한 사용자는 20% 정도의 성능 오버헤드만 감안한다면 새로운 언어인 스칼라를 익히지 않더라도 SparkR과 스파크 프레임워크를 통해 대용량 대이터의 머신러닝을 수행할 수 있음을 입증하였다. 향후 연구로 데이터 분석에 필수적인 데이터 전처리 및 가공 과정에서의 성능을 추가로 비교 분석하는 것이 필요하다.

감사의 글

이 논문은 2020년도 부산가톨릭대학교 교내연구비에 의하여 연구되었음

References

[1] 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.

[2] W. Ryu, “Distributed Processing of Big Data Analysis based on R using SparkR,” *J. of the Korea Institute of Electronic Communication Sciences*, vol. 17, no. 1, Feb. 2022, pp. 161-166.

[3] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, and D. Xin, “Mllib: Machine learning in apache spark,“ *The J. of Machine Learning Research,* vol. 17, no. 1, 2016, pp. 1235-1241.

[4] K. Ji and Y. Kwon, “Performance Comparison of Python and Scala APIs in Spark Distributed Cluster Computing System,” *J. of Korea Multimedia Society*, vol. 28, no. 2, Feb. 2020, pp. 241-248.

[5] 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, 2016, pp. 56-65.

[6] R. Anil, G. Çapan, I. Drost-Fromm, T. Dunning, E. Friedman, T. Grant, S. Quinn, P. Ranjan, S. Schelter, and O. Yilmazel, “Apache Mahout: Machine Learning on Distributed Dataflow Systems,” *J. Machine Learning Research*, vol. 21, no. 127, 2020, pp. 1-6.

[7] J. Jo, “Performance Comparison Analysis of AI Supervised Learning Methods of Tensorflow and Scikit-Learn in the Writing Digit Data,” *J. of the Korea Institute of Electronic Communication Sciences*, vol. 14, no. 4, Aug. 2019, pp. 701-705.

[8] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, and M. Štajdohar, “Orange: Data Mining Toolbox in Python,” *J. of Machine Learning Research*, vol. 14,

Aug. 2013, pp. 2349-2353.

[9] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur, "{TensorFlow}: A System for {Large-Scale} Machine Learning," In *12th USENIX Symp. on Operating Systems Design and Implementation (OSDI 16)*, Savannah, GA, USA, Nov. 2016, pp. 265-283.

[10] J. Jo, "Time Series Data Processing Deep Learning system for Prediction of Hospital Outpatient Number," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 16, no. 2, Apr. 2021, pp. 313-318.

[11] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," In *Advances in neural information processing systems*, Vancouver, Canada, Dec. 2019, pp. 8024-8035.

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

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