한국컴퓨터정보학회 동계학술대회 논문집 제25권 제1호 (2017. 1)

실시간 데이터 처리를 위한 아파치 스파크 기반 기계 학습 라이브러리 성능 비교

송준석[O], 김상영[*], 송병후[*], 김경태[*], 윤희용[**]
[O*]성균관대학교 정보통신대학 전자전기컴퓨터공학과
[**]성균관대학교 소프트웨어대학 소프트웨어학과
e-mail:alskpo@skku.edu[O*], impsoft@skku.edu[O*], by911129@skku.edu[O*], kyungtaekim76@gmail.com[*], youn7147@skku.edu[**]

A Performance Comparison of Machine Learning Library based on Apache Spark for Real-time Data Processing

Jun-Seok Song[O], Sang-Young Kim[*], Byung-Hoo Song[*], Kyung-Tae Kim[*], Hee-Yong Youn[**]
[O*]Dept. of Electrical and Computer Engineering, Sungkyunkwan University
[**]Dept. of Software, Sungkyunkwan University

● 요 약 ●

IoT 시대가 도래함에 따라 실시간으로 대규모 데이터가 발생하고 있으며 이를 효율적으로 처리하고 활용하기 위한 분산 처리 및 기계 학습에 대한 관심이 높아지고 있다. 아파치 스파크는 RDD 기반의 인 메모리 처리 방식을 지원하는 분산 처리 플랫폼으로 다양한 기계 학습 라이브러리와의 연동을 지원하여 최근 차세대 빅 데이터 분석 엔진으로 주목받고 있다. 본 논문에서는 아파치 스파크 기반 기계 학습 라이브러리 성능 비교를 통해 아파치 스파크와 연동 가능한 기계 학습라이브러리인 MLlib와 아파치 머하웃, SparkR의 데이터 처리 성능을 비교한다. 이를 위해, 대표적인 기계 학습 알고리즘인 나이브 베이즈 알고리즘을 사용했으며 학습 시간 및 예측 시간을 비교하여 아파치 스파크 기반에서 실시간 데이터 처리에 적합한 기계 학습 라이브러리를 확인한다.

키워드: 아파치 스파크(Apache Spark), 기계 학습(Machine Learning), MLlib, 아파치 머하웃, SparkR

I. Introduction

IoT(Internet of Things) 트랜드의 지속적인 발전으로 데이터는 점차 복잡한 데이터로 변화하고 있으며, 데이터 사이즈 역시 급격히 증가하고 있다[1]. 아파치 스파크(Apache Spark)는 인 메모리 처리 방식으로 대규모 데이터의 실시간 데이터 처리를 지원하고 다양한 기계 학습 라이브러리와 연동이 가능하다. 본 논문에서는 나이브 베이즈 알고리즘을 통해 아파치 스파크 기반 기계 학습 라이브러리인 MLlib, 아파치 머하웃, SparkR의 학습 시간 및 예측 시간을 비교하고, 아파치 스파크를 기반으로 실시간 데이터 처리에 적합한 기계 학습 라이브러리를 확인한다.

II. Preliminaries

1. Related works

1.1 아파치 스파크(Apache Spark)

아파치 소프트웨어 재단에서 제공하는 아파치 스파크는 RDD(Resilient Distributed Dataset) 기반의 인 메모리 처리 방식을 지원하는 분산 데이터 처리 플랫폼이며 다양한 프로그래밍 언어(Java, Scala, R)와 이를 위한 API를 제공한다[2]. 또한, 아파치 스파크는 효율적인 반복 처리를 통해 기계 학습 어플리케이션 개발에 적합한 환경을 지원하며, 기계 학습 라이브러리인 MLlib를 포함하여 자체적

으로 기계 학습 기능을 제공한다[3].

III. Performance Comparison

본 논문의 아파치 스파크 기반 기계 학습 라이브러리 성능 비교를 위한 클러스터 구성은 그림 1. 과 같다.

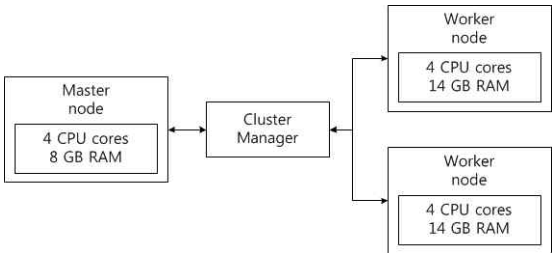

Fig. 1. Cluster Architecture

아파치 스파크 클러스터 환경 구축은 Ubuntu 환경에서 Java 1.7, Scala 2.10.4, Apache Spark 1.5.1을 이용했으며, 기계 학습 알고리즘인 나이브 베이즈 알고리즘을 사용했다. 또한, 나이브 베이즈 알고리즘의 학습 시간 및 예측 시간 비교를 위해 Sentiment140에서 제공하는 1,600,000개의 트위터 데이터를 사용했으며, 학습 데이터와 예측 데이터 비율은 각각 70%, 30%로 설정했다.

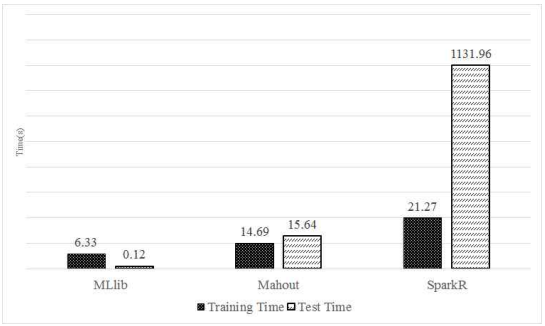

Fig. 2. Training and Predict Time(10,000 dataset)

데이터 10,000개로 성능을 측정한 결과, 아파치 스파크 기반 MLlib의 학습 시간 및 예측 시간이 다른 기계 학습 라이브러리보다 빠른 것을 확인했다.

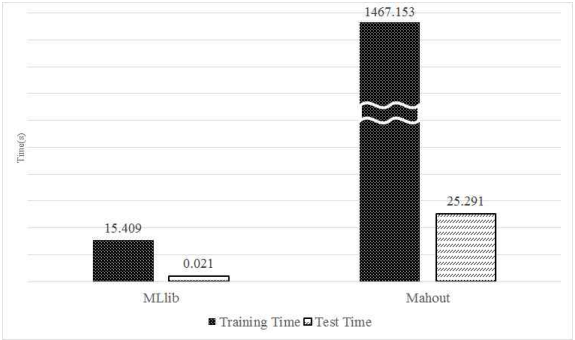

Fig. 3. Training and Predict Time(1,600,000 dataset)

속도가 가장 느렸던 SparkR을 제외하고 데이터 1,600,000개로 성능을 측정한 결과, 아파치 스파크 기반 MLlib의 학습 시간 및 예측 시간이 아파치 머하웃에 비해 약 100배, 약 120배 이상 빠른 것을 확인했다. 성능 비교 결과, MLlib가 아파치 스파크 기반 실시간 데이터 처리에 적합한 것을 확인했다.

IV. Conclusions

본 논문에서는 아파치 스파크에서 실시간 데이터 처리에 적합한 기계 학습 라이브러리를 확인하기 위해, 아파치 스파크 기반 기계 학습 라이브러리인 MLlib와 아파치 머하웃, SparkR의 학습 시간 및 예측 시간을 측정했다. 성능 측정 결과, 학습 시간과 예측 시간 모두 MLlib가 아파치 머하웃 및 SparkR보다 높은 성능을 보였다. 향후 연구방향으로는 아파치 스파크를 기반으로 다른 기계 학습 알고리즘을 이용하여 성능 평가를 진행할 예정이다.

Acknowledgment

본 연구는 Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.B0717-16-0070), Science and Technology (2016R1A6A3A11931385), the second Brain Korea 21 PLUS의 일환으로 수행되었음.

References

[1] M. Chen, S. Mao, and Y. Liu, "Big Data: A Survey," Mobile Networks and Applications, Vol. 19, No. 2, pp. 171-209, April, 2014.

[2] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing," In USENIX Symposium on Networked Systems Design and Implementation, 2012.

[3] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, DB Tasi, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar, "MLlib: Machine Learning in Apache Spark," Journal of Machine Learning Research, Vol. 17, No. 1, pp. 1235-1241, 2016.