Apache Mahout

NAVEENKUMAR RAMARAJU[1,*]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors:naveenkumar2703@gmail.com

Paper 2, April 12, 2017

Apache Mahout[1] is an extensible programming environment to build scalable machine learning algorithms. It has algorithms to work in tandem with frameworks like Hadoop, Spark, Flink and H2O which specializes on dealing with large scale data. Samsara is a vector math environment to do linear algebra operations using distributed computing.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Mahout, Samsara, Apache, Scalable, Machine learning

https://github.com/naveenkumar2703/sp17-i524/paper2/S17-IR-2029/report.pdf

1. INTRODUCTION

Mahout is an open source software to create scalable machine learning models. The initial version of Mahout targeted to implement all ten machine learning algorithms discussed in Andrew Ng's paper "Map-Reduce for Machine Learning on Multicore"[2] with scalability in mid. Further releases of Mahout added various implementations of clustering, classification, collaborative filtering and genetic algorithms in Java for usage in single machine as well as in clusters using map reduce.

As the popularity of in-memory softwares like Spark started gaining popularity over disk based softwares like Hadoop, new features rolled out by Mahout did not support MapReduce. "Samsara" is a module to do algebraic operations which was released in Mahout version 0.10 and is compatible only with frameworks like Spark[3], H2O[4] and Flink[5] but not MapReduce based frameworks. Samsara was mainly written in Scala and is optimized to operate well independent of the background. Another key feature of Samsara is that it supports R-like syntax for linear algebra operations.

Mahout has many algorithms for MapReduce based frameworks like Hadoop. Only some of them are implemented for in-memory based frameworks like Spark. A full list of algorithms available in Mahout and the frameworks supported by them is discussed in section 2.

2. FEATURES

Mahout supports wide range of machine learning algorithms like classification, collaborative filtering and clustering, dimensionality reduction techniques like SVD, PCA and QR decomposition[6]. Mahout Samsara environment provides linear algebra and statistics operations. Each of these are described in this section.

2.1. Samsara - Math Environment

Mahout Samsara[7] is a math environment to create and perform various math and linear algebra operations. Some of the key functionalities are BLAS (Basic Linear Algebra Subprogram), distributed row matrix, distributed ALS (Alternating Least Squares), PCA (principal component analysis), incore and distributed SPCA (Stochastic PCA), SVD (singular value decomposition), incore and distributed SSVD (Stochastic SVD), Eigen decomposition, Cholesky decomposition and similarity analysis.

One of the main advantage of Samsara is it supports R and Matlab like syntax using Scala's DSL (domain specific language) feature. DSL's are syntactic sugars for easy interpretation. An example is provided in section 5. Mahout Samsara is supported on Spark, H2O and Flink engines. Samsara is not available in Hadoop and MapReduce based engines but has a different implementation that supports all these math operations.

2.2. Classification

Mahout has logistic regression using stochastic gradient descent, naive Bayes, complementary naive Bayes, random forest and hidden markov algorithms for classification. Naive Bayes available in Spark and MapReduce is the only distributed classification algorithm. Others are supported only on single machines.

2.3. Clustering

Mahout supports clustering algorithms like K-means, fuzzy k-means, streaming k-means, spectral clustering and canopy clustering. These are available only for single machines and MapReduce based environments.

2.4. Collaborative Filtering

Mahout has implementation for user based and item based collaborative filtering algorithms for single machine, MapReduce and spark engines. It also has implementation of matrix factorization with ALS, weighted matrix factorization using SVD for single machine and MapReduce.

2.5. Other

Mahout supports several other features like Latent Dirichlet Allocation, row similarity job, collocations, sparse TF-IDF (Term frequency - inverse document frequency, a common feature used in information retrival and document search) vectors from text, XML Parsing, Email Archive Parsing for MapReduce engines. It also has Evolutionary Processes/Genetic algorithms implementation that runs on single machine.

3. LICENSING

Apache Mahout is an open source software available for free commercial usage and is licensed under Apache License, Version 2.0[8]. Associated frameworks like Spark, Hadoop, Flink and H2O are also open source products.

4. ECOSYSTEM

Mahout can be used with wide variety of distributed systems like Spark, H2O, Flink and Hadoop. It can also be used on single machine. A key benefit of Mahout is that the machine learning or math code can be used and written in same syntax independent of backends.

An illustration of how Mahout works by using Scala DSL for math operations with various engines is illustrated in figure 1.

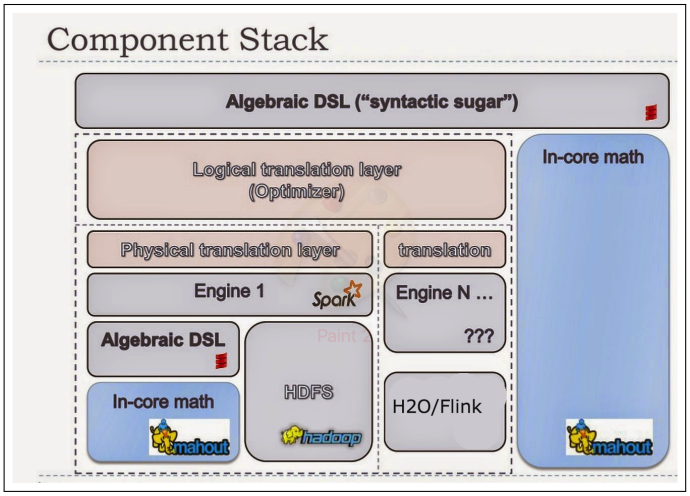

Fig. 1. Mahout Ecosystem.
Source: [9]

5. USE CASES

An use case for recommendation system and a simple linear algebra operation is provided in this section.

5.1. Recommendation system with Spark Engine

A recommendation system is used to recommend most relevant product that he/she might be interested and boost the sales in online platforms. Recommendation can be done based on user similarity or item similarity. In case of user similarity, recommendation is provided based on idea that users with similar interest would buy similar products. Similar users are identified, grouped and recommendations are made based on difference in set of products. Where as item similarity is based on identifying the products that were bought together and recommending the products that is not in customer's basket or purchase history. Recommendation based on item similarity is very popular in industry as number of product types is way less than number of customers with regular purchase patterns in large retail industry.

An illustration of how Mahout can be used with Spark for item based recommendation is illustrated in figure 2.

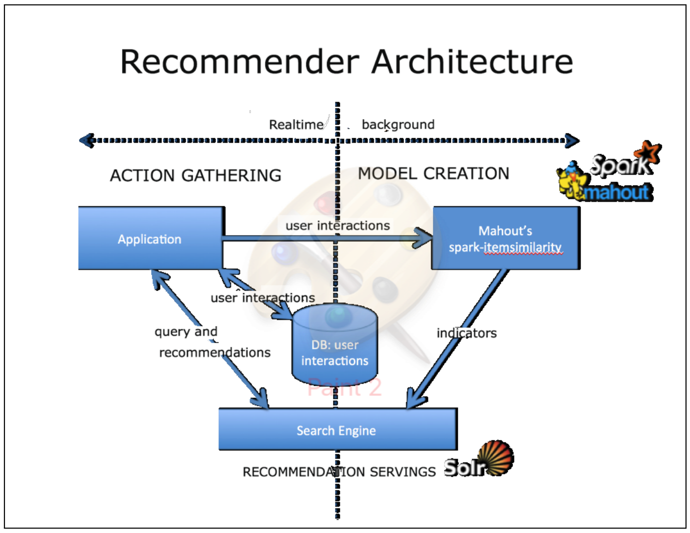

Fig. 2. Recommender Illustration.
Source: [10]

Mahout's MapReduce version of item similarity takes a text file that is expected to have user and item IDs and provide recommendations based on content.

5.2. SPCA

The equation to compute stochastic principal component analysis is given in equation 1

$$G = BB' - C - C' + s_q s_q{}' \xi' \xi \tag{1}$$

where G is the matrix representation of the result of SPCA, B is the input or feature matrix, C is correlation matrix of the inputs with B′ and C′ as their transpose matrix, s_q is standard deviation of each feature and ξ is covariance matrix and $s_q{}'$, ξ' are their respective transpose matrices.

One line Mahout code to compute this using Scala DSL is illustrated here,

val g = bt.t %*% bt - c - c.t + $(s_q crosss_q) * (xi\ dot\ xi)$

This could be used in single or distributed machine in Spark, H2O or Flink to perform SPCA.

6. USEFUL RESOURCES

Apache Mahout Cookbook[11] by Piero Giacomelli is a good introductory book on Mahout. Apache Mahout Beyond MapReduce[12] by Dmitriy Lyubimov provides an exhaustive coverage of Mahout Samsara and math operations.

Mahout website[13] has a compilation of useful resources like books, tutorials and talks about Mahout and machine learning.

7. CONCLUSION

Apache Mahout provides an open source environment with scalable algorithms for machine learning and math operations using DSL. It can be used with multiple backends like Spark, Scala, Flink and Hadoop. Same code can be used for single machine and distributed computing of algorithms in any supported backends.

ACKNOWLEDGEMENTS

This work was done as part of the course "I524: Big Data and Open Source Software Projects" at Indiana University during Spring 2017. Thanks to our Professor Gregor von Laszewski and associate instructors for their help and support during the course.

REFERENCES

[1] Apache Software Foundation, “Apache mahout,” Web Page, Jun. 2016, accessed: 2017-3-26. [Online]. Available: http://mahout.apache.org/

[2] C. tao Chu, S. K. Kim, Y. an Lin, Y. Yu, G. Bradski, K. Olukotun, and A. Y. Ng, “Map-reduce for machine learning on multicore,” in *Advances in Neural Information Processing Systems 19*, P. B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 281–288, accessed: 2017-3-26. [Online]. Available: http://papers.nips.cc/paper/3150-map-reduce-for-machine-learning-on-multicore.pdf

[3] Apache Spark Community, “Apache spark,” Web Page, Feb. 2017, accessed: 2017-4-4. [Online]. Available: https://spark.apache.org/

[4] H20.ai, “H20,” Web Page, Jan. 2016, accessed: 2017-4-4. [Online]. Available: https://www.h2o.ai/h2o/why-h2o/

[5] Apache Flink Community, “Introduction to Apache Flink,” Web Page, Mar. 2016, accessed: 2017-4-4. [Online]. Available: https://flink.apache.org/introduction.html

[6] Apache Software Foundation, “Mahout Features by Engine,” Web Page, Jan. 2016, accessed: 2017-4-4. [Online]. Available: https://mahout.apache.org/users/basics/algorithms.html

[7] Apache Software Foundation, “Mahout samsara incore references,” Web Page, May 2016, accessed: 2017-3-26. [Online]. Available: http://mahout.apache.org/users/environment/in-core-reference.html

[8] Apache Software Foundation, “Apache License 2.0,” Web Page, Jan. 2004, accessed: 2017-3-26. [Online]. Available: http://www.apache.org/licenses/LICENSE-2.0

[9] D. Lyubimov and A. Palumbo, “Mahout 0.10.x: first mahout release as a programming environment,” Web Page, Apr. 2015, accessed: 2017-3-26. [Online]. Available: http://www.weatheringthroughtechdays.com/2015/04/mahout-010x-first-mahout-release-as.html

[10] Apache Software Foundation, “Intro to cooccurrence recommenders with spark,” Web Page, May 2016, accessed: 2017-3-26. [Online]. Available: http://mahout.apache.org/users/algorithms/intro-cooccurrence-spark.html

[11] P. Giacomelli, *Apache Mahout Cookbook*. Packt Publishing, 2016, accessed: 2017-3-26.

[12] D. Lyubimov and A. Palumbo, *Apache Mahout: Beyond MapReduce*. Createspace Independent Publishing Platform, 2016, accessed: 2017-3-26.

[13] Apache Software Foundation, “Mahout book, tutorials, talks,” Web Page, May 2016, accessed: 2017-3-26. [Online]. Available: https://mahout.apache.org/general/books-tutorials-and-talks.html