(/) MLlib

Latest News

Spark 2.1.1 released (/news/spark-2-1-1-released.html) (May 02, 2017)

Spark Summit (June 5-7th, 2017, San Francisco) agenda posted (/news/spark-summit-june-2017-agenda-posted.html) (Mar 31, 2017)

Spark Summit East (Feb 7-9th, 2017, Boston) agenda posted (/news/spark-summit-east-2017-agenda-posted.html) (Jan 04, 2017)

Spark 2.1.0 released (/news/spark-2-1-0-released.html) (Dec 28, 2016)

Archive (/news/index.html)

MLlib is Apache Spark's scalable machine learning library.

Ease of Use

Usable in Java, Scala, Python, and R.

MLlib fits into Spark (/)'s APIs and interoperates with NumPy (http://www.numpy.org) in Python (as of Spark 0.9) and R libraries (as of Spark 1.5). You can use any Hadoop data source (e.g. HDFS, HBase, or local files), making it easy to plug into Hadoop workflows.

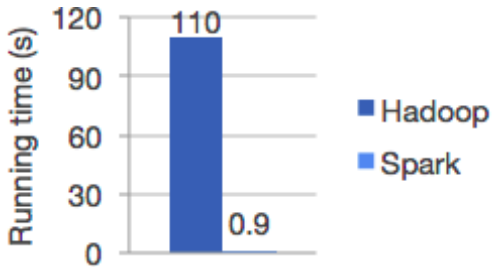
```
data = spark.read.format("libsvm")\
  .load("hdfs://...")

model = KMeans(k=10).fit(data)
```

Calling MLlib in Python

Performance

High-quality algorithms, 100x faster than MapReduce.

Spark excels at iterative computation, enabling MLlib to run fast. At the same time, we care about algorithmic performance: MLlib contains high-quality algorithms that leverage iteration, and can yield better results than the one-pass approximations sometimes used on MapReduce.



Logistic regression in Hadoop and Spark

Easy to Deploy

Runs on existing Hadoop clusters and data.

If you have a Hadoop 2 cluster, you can run Spark and MLlib without any pre-installation. Otherwise, Spark is easy to run standalone (/docs/latest/spark-standalone.html) or on EC2 (/docs/latest/ec2-scripts.html) or Mesos (https://mesos.apache.org). You can read from HDFS (https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html), HBase (https://hbase.apache.org), or any Hadoop data source.

Algorithms

MLlib contains many algorithms and utilities.

ML algorithms include:

- Classification: logistic regression, naive Bayes,...
- Regression: generalized linear regression, survival regression,...
- Decision trees, random forests, and gradient-boosted trees
- Recommendation: alternating least squares (ALS)
- Clustering: K-means, Gaussian mixtures (GMMs),...
- Topic modeling: latent Dirichlet allocation (LDA)
- Frequent itemsets, association rules, and sequential pattern mining

ML workflow utilities include:

- Feature transformations: standardization, normalization, hashing,...
- ML Pipeline construction
- Model evaluation and hyper-parameter tuning
- ML persistence: saving and loading models and Pipelines

Other utilities include:

- Distributed linear algebra: SVD, PCA,...
- Statistics: summary statistics, hypothesis testing,...

Refer to the MLlib guide (/docs/latest/ml-guide.html) for usage examples.

Community

MLlib is developed as part of the Apache Spark project. It thus gets tested and updated with each Spark release.

If you have questions about the library, ask on the Spark mailing lists (/community.html#mailing-lists).

MLlib is still a rapidly growing project and welcomes contributions. If you'd like to submit an algorithm to MLlib, read how to contribute to Spark (/contributing.html) and send us a patch!

Getting Started

To get started with MLlib:

- Download Spark (/downloads.html). MLlib is included as a module.