Lesson 17

Mahout

Mahout

- Fast and Efficient Processing of Big Data
- Processes very large datasets at the cluster of machines with high efficiency for above 10 M data points in shared-nothing environment.
- Big Data analysis: datasets with over 1 million data points

Mahout

- Can run on multiple machines as well as multi-core processing units
- Compute tasks fast when distributed over the cloud, tasks runs parallel, and run in shared nothing-computational environment of multiple machines and multi-core units

Mahout in sequential shared environment

- Higher time efficiency for less than 1 M data points.

Mahout Eclipse and Maven Installing

- Download Maven and Eclipse

https://www.eclipse.org/downloads/

Add -Name: m2eclipse-Location: http://download.eclipse.org/technology/m2e/releases (Google “install m2eclipse”)

- Ubuntu: sudo apt-get install git

sudo apt-get install maven

Mahout Installing

- http://mahout.apache.org/general/downloads.html

Latest Release: 0.13.0 - mahout-distribution-0.13.tar.gz

- Apache Preparing for Mahout version 0.14.0 (June 2018)

Git Hub

- GitHub: Mahout Mirror https://github.com/apache/mahout
- git://git.apache.org/mahout.git mahout-trunk

Git Hub Maven under mahout-trunk folder:

- <dependency>
- <groupId>org.apache.mahout</groupId>
- <artifactId>mahout-core</artifactId>
- <version>${mahout.version}</version>
- </dependency>

Mahout Environment Setting

- export MAHOUT_HOME=/path/to/mahout
- export MAHOUT_LOCAL=true # for running standalone on your dev machine,
- # unset MAHOUT_LOCAL for running on a cluster

Feature in Mahout 0.13.0 (2017)

- Enables easier implementations of most modern machine learning and deep learning algorithms
- Open-source distributed scalable linear algebra library ViennaCL, the Java wrapper library interface JavaCPP

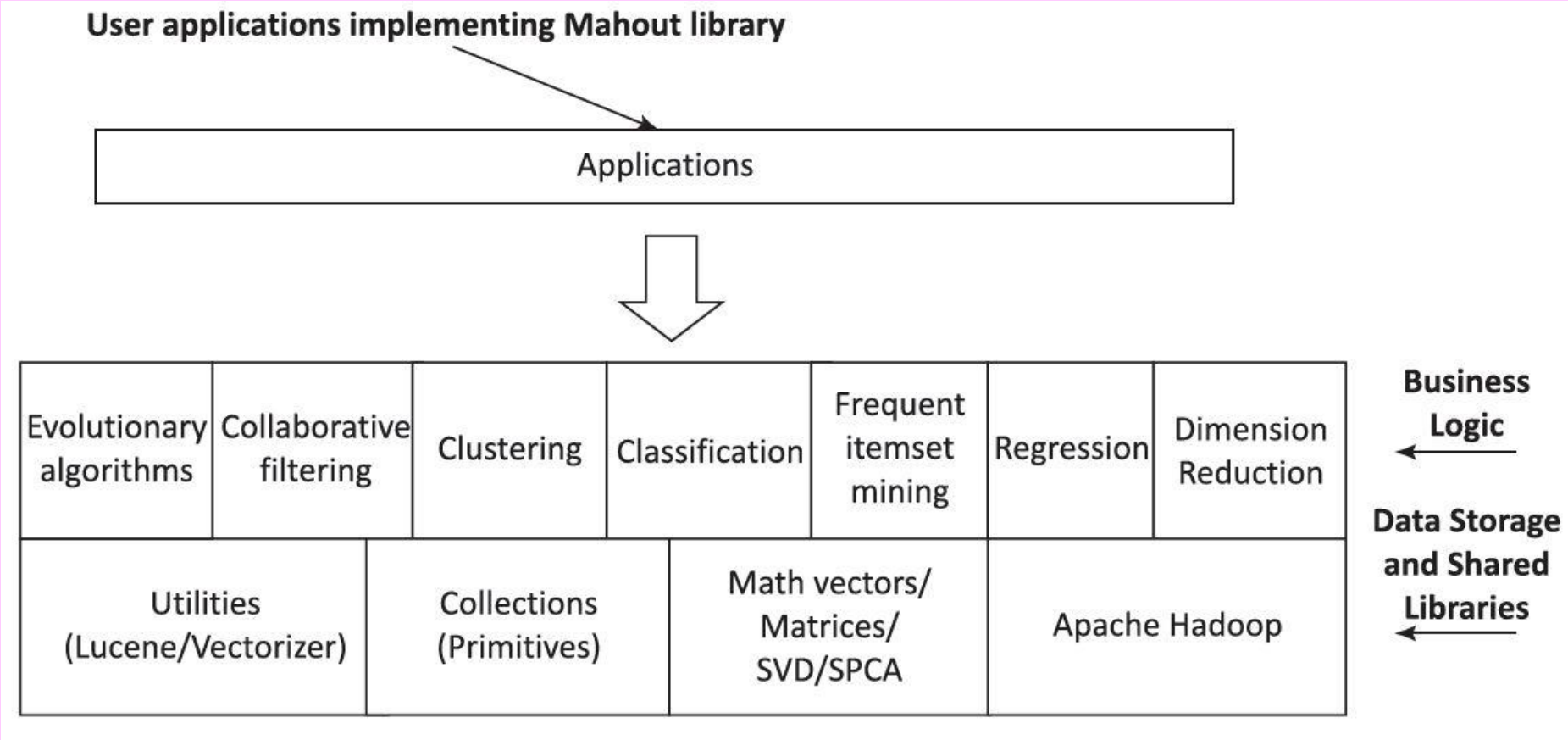
Feature in Mahout 0.13.0 (2017)

- The graphics processor manufacturer, NVIDIA CUDA bindings directly into Mahout
- Easier to run matrix mathematics on graphics cards (used in computers for fast graphic computations)

Java Implementations

- Java libraries for common mathematics and statistical operations (focused on linear algebra) and primitive Java Collection Interfaces

Figure 6.21 Mahout architecture



Features of Mahout

- Designed on top of Apache Hadoop
- Supports algebraic platforms like fast computing Apache Spark paradigm and MapReduce

Features of Mahout

- A scalable generalized tensor and linear algebra solving engine, designed on top of Apache Hadoop.
- Supports algebraic platforms like fast computing Apache Spark paradigm and MapReduce
- Distributed row matrix (DRM), scalable libraries for matrix and vectors

Random Access Feature

- Random access means accessing vectors using key, index or hash followed by values, in any order

Clustering Implementations

- Contains several Spark and MapReduce enabled, such as k-means, fuzzy k-means, Canopy, Dirichlet and mean shift, Latent Dirichlet Allocation,
- Spectral Clustering, MultiHash Clustering
- Hierarchical clustering

Classification Implementations

- Single Machine Sequential: Stochastic Gradient Descent (SGD), Logistic Regression trained via SGD, Hidden Markov, Multi-layer Perceptron
- Parallel Distributed (Map Reduce):
- Naïve Bayes, complementary Naïve Bayes and Random Forest

Collaborative Filtering

- Enables making automatic predictions about items and item sets of interests
- Similar item-sets mining

SVD++ and SGD

- Weighted matrix factorization SVD++, and parallel SGD (in sequentially in shared-data environment)
- SGD [an iterative learning algorithm in which each training example is used to pull the model M program slightly to reach more closer to correct answer]

Recommender Implementation

- Collaborative filtering based recommender,
- SVD recommender
- KNN-item-based recommender (linear interpolation item based recommender)
- Cluster-based recommender

Implementation APIs

- APIs for distributed and in-core first and second moment routines
- Distributed and local principal component analysis (DSPCA and SPCA) and stochastic singular value decomposition (DSSVD and SSVD), singular value decomposition (SVD),

Implementation APIs

- Distributed Cholesky QR (thinQR)
- Distributed regularized Alternating Least Squares (DALS)
- Matrix factorization with alternating least squares

Principal Component Analysis (PCA)

- Means a linear transformation method
- Finds the directions of maximum variance in high-dimensional data
- Projects those data for transformation onto a smaller dimensional subspace while retaining most of the information

Principal Component Analysis (PCA)

- Identifies patterns in data sets
- Detect the correlation among the variables
- Strong correlation: Try for reducing the dimensionality

PCA Applications

- PCA applies in a number of use cases, such as stock market predictions, and the analysis of gene expression data.

Summary

We learnt:

- Mahout for Fast and Efficient Processing of Big Data
- A scalable generalized tensor and linear algebra solving engine, designed on top of Apache Hadoop.
- Mahout implementations of Machine Learning Algorithms
-

Summary

We learnt:

- Mahout for Similar item sets, Clustering, Classification, Recommender
- PCA, SVD++, SGD
- Collaborative Filtering

End of Lesson 17 on

Mahout