L4: Statistical Learning with Mahout (机器学习工具)

Feng Li

feng.li@cufe.edu.cn

School of Statistics and Mathematics

Central University of Finance and Economics

Revised on December 9, 2016

Today we are going to learn...
(本章要点)

1. **Introduction to Mahout**（Mahout 介绍）
2. **Mahout with examples**（Mahout 示例）
3. **Classification with random forests(随机森林)**
4. Scala & Spark Bindings for Mahout（Scala & Spark 与 Mahout）

Introduction to Mathout (Mahout 介绍） I

- Mahout is a scalable machine learning library that implements many different approaches to machine learning.
- Mahout's machine learning algorithms are available in various areas like,
  - Collaborative Filtering
  - Classification
  - Clustering
  - Dimensionality Reduction
  - Topic Models

  for many modern distributed parallel computing system, such as **Hadoop**, **Spark**, or even on a single machine.
- To check out the complete list of supporting algorithms and platforms, visit `http://mahout.apache.org/users/basics/algorithms.html`

Install Mahout
(安装 Mahout)

- Download Mahout from
  `http://www.apache.org/dyn/closer.cgi/mahout/`
- Install Mahout, checkout the Mahout Guide at
  `http://feng.li/files/pc2014fall/Hadoop-Guide.html`
- Documentation and Examples are available from Mahout's homepage
  `https://mahout.apache.org/`

Mahout with examples

- See Hadoop Guide.
- And the Mahout homepage.

Classification with random forests (随机森林)

- Overview of Random Forest
  - The detailed discussion can be found in **Chapter 15** in *The Elements of Statistical Learning*.
- The KDD example
  - In this example we'll use the NSL-KDD dataset because its large enough to show the performances of the partial implementation.
  - You can download the dataset here http://nsl.cs.unb.ca/NSL-KDD/ we'll use the full `training dataset` "KDDTrain+.ARFF" and the `test set` "KDDTest+.ARFF".
  - Open the train and test files and remove all the lines that begin with '@'. All those lines are at the top of the files.
  - Put the data in HDFS: `testdata` directory
  - Then just need to train your model and test it with testing data. See my tutorial.

Scala & Spark Bindings for Mahout (Scala & Spark 与 Mahout) I

- **Apache Spark** is an open-source cluster computing framework originally developed in the AMPLab at UC Berkeley.
- In contrast to Hadoop's two-stage disk-based MapReduce paradigm, Spark's **in-memory primitives** provide performance up to 100 times faster for certain applications.
- By allowing user programs to load data into a cluster's memory and query it repeatedly, Spark is **well suited to machine learning algorithms**.

Scala & Spark Bindings for Mahout (Scala & Spark 与 Mahout) II

- **Mahout Scala and Spark Bindings** is a package aiming to provide a R-like look and feel to Mahout's in-core and out-of-core Spark-backed linear algebra.
- It is built in the image of R's base package. So if you are familiar with basic R matrix primitives, you should feel right at home.
- Take the following expression a look

$$\mathbf{G} = \mathbf{B}\mathbf{B}^\top - \mathbf{C} - \mathbf{C}^\top + \mathbf{s}_q\mathbf{s}_q^\top\xi^\top\xi$$

- Mahout Scala & Spark Bindings expression of the above:

```
val g = bt.t %*% bt - c - c.t + (s_q cross s_q) * (xi dot xi)
```

- The main idea is that a scientist writing algebraic expressions cannot care less of distributed operation plans and works entirely on the logical level just like he or she would do with R.

Scala & Spark Bindings for Mahout (Scala & Spark 与 Mahout) III

- Another idea is decoupling logical expression from distributed back-end. As more back-ends are added, this implies "**write once, run everywhere**".

External Readings
(参考文档)

- Apache Spark

  `http://spark.apache.org/`

- Scala and Spark bindings manual

  `http://mahout.apache.org/users/sparkbindings/ScalaSparkBindings.pdf`