L3: Statistical Modeling with Hadoop
(Hadoop 和统计建模)

Feng Li

feng.li@cufe.edu.cn

School of Statistics and Mathematics

Central University of Finance and Economics

Revised on November 30, 2016

Today we are going to learn...
(本节知识要点)

1. **Linear regression with Hadoop（利用 Hadoop 计算一个线性回归模型)**

2. **Logistic regression with Hadoop（利用 Hadoop 计算逻辑斯蒂回归模型)**

Linear Regression (线性回归模型) I

- Assume we have a large dataset. How will we perform regression data analysis now?
- In such cases, we can use R and Hadoop integration to perform parallel linear regression by implementing Mapper and Reducer.
- It will divide the dataset into chunks among the available nodes and then they will process the distributed data in parallel.
- It will not fire memory issues when we run with an R and Hadoop cluster because the large dataset is going to be distributed and processed with R among Hadoop computation nodes.
- Also, keep in mind that this implemented method does not provide higher prediction accuracy than the lm() model.

Linear Regression
(线性回归模型) II

- Assume we have data set contains both $y_{n\times 1}$ and $X_{n\times p}$. The linear model

$$y = X\beta + \epsilon$$

yields the following solution to $\hat{\beta}$

$$\hat{\beta} = (X'X)^{-1}X'y$$

- In OLS, to find coefficients is equal to find the solution of the linear system

$$(X'X)\beta = X'y$$

- The Big Data problem: $n >> p$
  - The calculations of $X'X$ and $X'y$ is very computational demanding.
  - But notice that the final output of $(X'X)_{p\times p}$ and $(X'y)_{p\times 1}$ are fairly small.

Linear Regression
(线性回归模型) III

- The solution:
  - Let Hadoop calculate $X'X$ and $X'y$.
  - The final results can be obtained afterwards.
- The outline of the linear regression algorithm is as follows:
  1. Calculating the $X'X$ value with MapReduce job1.
  2. Calculating the $X'y$ value with MapReduce job2.
  3. Deriving the coefficient values with $solve(X'X, X'y)$.

Logistic regression
(逻辑斯蒂回归模型) I

- In statistics, logistic regression or logit regression is a type of probabilistic classification model.
- Logistic regression is used extensively in numerous disciplines, including the medical and social science fields. It can be binomial or multinomial.
- Binary logistic regression deals with situations in which the outcome for a dependent variable can have two possible types.
- Multinomial logistic regression deals with situations where the outcome can have three or more possible types.
- Logistic regression can be implemented using logistic functions.
- The **logit model** connects the explanatory variables in this way

$$P_i = \frac{1}{1 + \exp(-(\beta_1 + \beta_2 X_i))}$$

Logistic regression
(逻辑斯蒂回归模型) II

- Alternatively we can write the model in this way

$$\log \frac{P_i}{1 - P_i} = \beta_1 + \beta_2 X_i$$

where $P_i/(1 - P_i)$ is called **odds ratio**: the ratio of probability of a family will own a house to the probability of not owing a house.

- This model can be easily estimated with R

```
> glm(formula, data=inputdata, family="binomial")
```

- **Bad news**: The above estimation requires sequential iterative method.
- Will the following hypothetical Hadoop workflow work?
  - Defining the Mapper function
  - Defining the Reducer function
  - Defining the Logistic Regression MapReduce function

Logistic regression (逻辑斯蒂回归模型) III

- Logistic regression is the standard industry workhorse that underlies many production fraud detection and advertising quality and targeting products. The most common implementations use Stochastic Gradient Descent (SGD) to all large training sets to be used. The good news is that it is blazingly fast and thus it is not a problem for Hadoop implementation to handle training sets of tens of millions of examples. With the down-sampling typical in many data-sets, this is equivalent to a dataset with billions of raw training examples. The ready to use solutions:
  - **Apache Mahout** (see next lecture)

    `https://mahout.apache.org/users/classification/logistic-regression.html`
  - **RHadoop**: The RHadoop package `rmr2` allow you to code your own gradient descent method with Hadoop.

    `https://github.com/RevolutionAnalytics/rmr2`

Logistic Regression: The divide and conquer approach (逻辑斯蒂回归模型：各个击破方法） I

- Consider the following logistic regression model

$$p(y_i) = \frac{\exp\{x_i'\beta\}}{1+\exp\{x_i'\beta\}}$$

where $x_i = (x_{i1}, ..., x_{ip})'$, $i = 1, ..., n$. The $\widehat{\beta}$ for β coefficients can be estimated via maximum likelihood method.

- The Logistic Regression with subsets
  - Divide n sample into k blocks that each block consists of m observations。
  - Do logistic regression with each block on a single node.

$$\widehat{\beta}_l = \arg \max \sum_{i=1}^{m} \{y_{li} x_{li}'\beta - \log(1+\exp\{x_i'\beta\})\}.$$

Logistic Regression: The divide and conquer approach (逻辑斯蒂回归模型：各个击破方法) II

- The Full Logistic Regression model with coefficients $\widehat{\beta}$ can be approximated by weighted average of $\widehat{\beta}_l$

$$\widehat{\beta} = \frac{1}{k}\sum_{l=1}^{k}\widehat{\beta}_l.$$

- Some properties:

  When $m \to \infty$, $n \to \infty$, it can be shown that

  - Consistency

$$\widehat{\beta} \xrightarrow{p} \beta.$$

  - Asymptotic Normal

$$\sqrt{n}(\widehat{\beta} - \beta) \sim N(0, I(\beta))$$

  where

$$I(\beta) = \lim_{m\to\infty}\frac{1}{k}\sum_{l=1}^{k}\{\frac{1}{m}\sum_{i=1}^{m}var(s(\beta_l))\}^{-1}.$$

Assignment (III)

- Use the dataset given on Hadoop server to implement at least one of statistical models you have learned before.