基于 R 软件的大数据处理与分析建模平台

一、环境要求

1、CentOS 系统环境要求：

Java (>= 1.7)　spark (2.1.1)

2、通过 YUM 源安装 R，和相应的库

R (>=3.3.1) 版本，已通过集群测试。建议同步最新的 yum 源。

```
[redacted]Rstudio]# yum list -q R-\*
Repodata is over 2 weeks old. Install yum-cron? Or run: yum makecache fast
已安装的软件包
R.x86_64                                                    3.3.3-1.el7
R-Rcpp.x86_64                                               0.12.10-1.el7
R-Rcpp-devel.x86_64                                         0.12.10-1.el7
R-core.x86_64                                               3.3.3-1.el7
R-core-devel.x86_64                                         3.3.3-1.el7
R-devel.x86_64                                              3.3.3-1.el7
R-java.x86_64                                               3.3.3-1.el7
R-java-devel.x86_64                                         3.3.3-1.el7
```

```
# 已安装里面需要包括 Rcpp, Rcpp-devel, java, java-devel,
sudo yum install R-Rcpp.x86_64
sudo yum install R-Rcpp-devel.x86_64

# 继续安装依赖的 rpm
sudo yum install libcurl-devel
sudo yum install openssl-devel
sudo yum install libxml2-devel

# 如有必要需要重新配置 R 里面的 java 环境变量
R CMD javareconf
```

3、安装 Rstudio Server

下载地址说明

https://www.rstudio.com/products/rstudio/download-server/

```
$ wget https://download2.rstudio.org/rstudio-server-rhel-1.0.153-x86_64.rpm
$ sudo yum install --nogpgcheck rstudio-server-rhel-1.0.153-x86_64.rpm
```

RedHat/CentOS 6 and 7

To download and install RStudio Server open a terminal window and execute the commands corresponding to the 32 or 64-bit version as appropriate.

64bit

Size: 39.5 MB MD5: 5c72047380e2944926acb77b69e0c4a0 Version: 1.0.153 Released: 2017-07-20

```
$ wget https://download2.rstudio.org/rstudio-server-rhel-1.0.153-x86_64.rpm
$ sudo yum install --nogpgcheck rstudio-server-rhel-1.0.153-x86_64.rpm
```

32bit

Size: 38.4 MB MD5: f69f5ba72295a1b97ed3b4d84c6bb01e Version: 1.0.153 Released: 2017-07-20

```
$ wget https://download2.rstudio.org/rstudio-server-rhel-1.0.153-i686.rpm
$ sudo yum install --nogpgcheck rstudio-server-rhel-1.0.153-i686.rpm
```

并配置相关用户的用户名和密码，开放服务器端口 8787。参考文档:

https://github.com/BruceZhaoR/myconfig/blob/master/rstudio-server/rstudio-server-notes.md

4、相关 R 包和集成软件

4.1 通过 CRAN 下载下列包

```
install.packages(c("dplyr", "sparklyr", "rsparkling", “ggplot2”), repos =
"https://mirrors.tuna.tsinghua.edu.cn/CRAN")
```

通过镜像下载更快

4.2 编译安装相应版本的 h2o 包

安装特定 h2o 版本

http://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/R/src/contrib/h2o_3.14.0.2.tar.gz

```
R CMD INSTALL h2o_3.14.0.2.tar.gz
```

4.3 下载 sparkling water，放在用户可读写访问的文件夹下

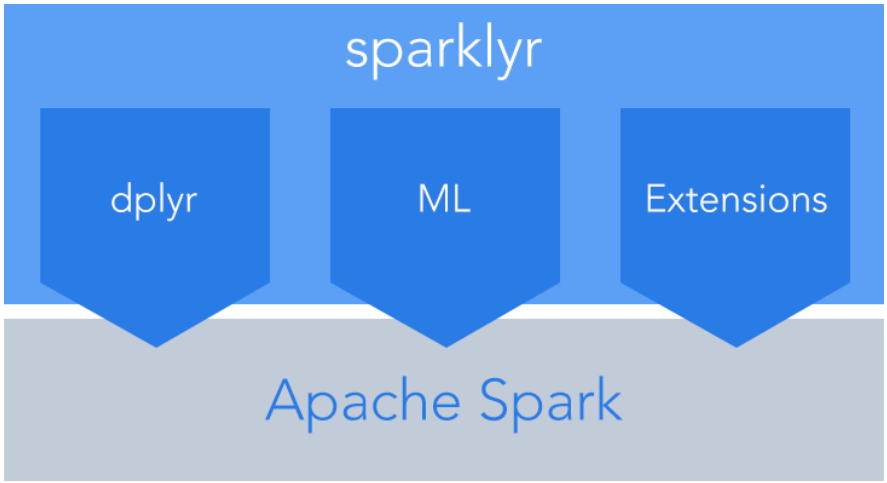
Sparkling water (2.1.14)

http://h2o-release.s3.amazonaws.com/sparkling-water/rel-2.1/14/sparkling-water-2.1.14.zip

二、分析流程

1、数据处理

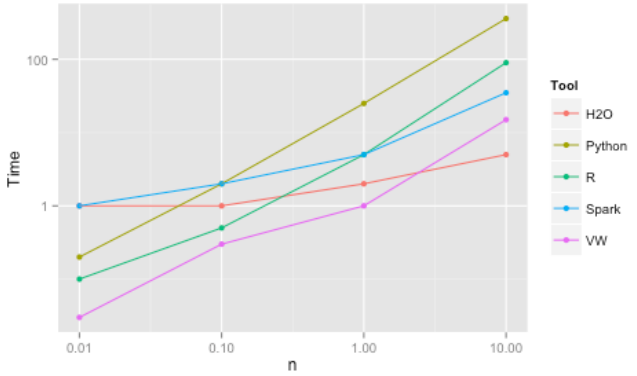
采用 spark (2.1.1) 来处理数据。通过 dplyr、sparklyr 包将 R 代码转为 spark SQL 来操作数据。dplyr 是一个高效、简洁的数据处理包，通过 sparklyr 连接 spark，仿佛就是在本地处理数据，能快速、高效地完成数据清洗任务。

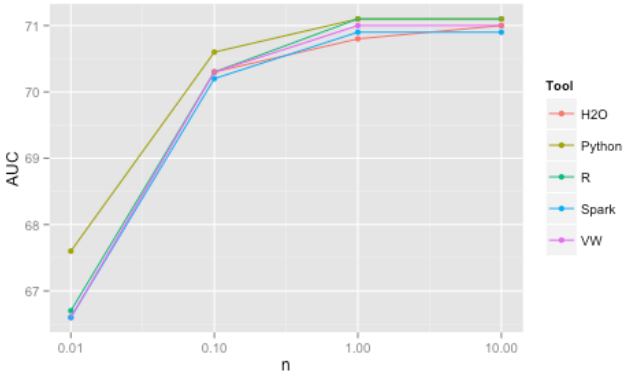


2、建模

由于数据是分布式，spark MLlib 在计算广义线性模型时速度较慢、方法较为单一，故采用 h2o 的 spark 扩展 sparkling water。通过 R 的 rsparkling 包进行连接操作。具体说明请见下图：

benchmark： https://github.com/szilard/benchm-ml#linear-models





H2O 算法优势:

H2O's GLM Overview

- Fully Distributed and Parallel
  - handles datasets with up to 100s of thousand of predictors
  - scales linearly with number of rows
  - processes datasets with 100s of millions of rows in seconds
- All standard GLM features
  - standard families
  - support for observation weights and offset
- Elastic Net Regularization
  - lambda-search - efficient computation of optimal regularization strength
  - applies strong rules to filter out in-active coefficients
- Several solvers for different problems
  - Iterative re-weighted least squares with ADMM solver
  - L-BFGS for wide problems
  - Coordinate Descent (Experimental)

H2O's GLM Overview (2)

- Automatic handling of categorical variables
  - automatically expands categoricals into 1-hot encoded binary vectors
  - Efficient handling (sparse acces, sparse covariance matrix)
  - (Unlike R) uses all levels by default if running with regularization

- Missing value handling
  - missing values are not handled and rows with any missing value will be omitted from the training dataset
  - need to impute missing values up front if there are many

H2O GLM Wit Elastic Net Regularization

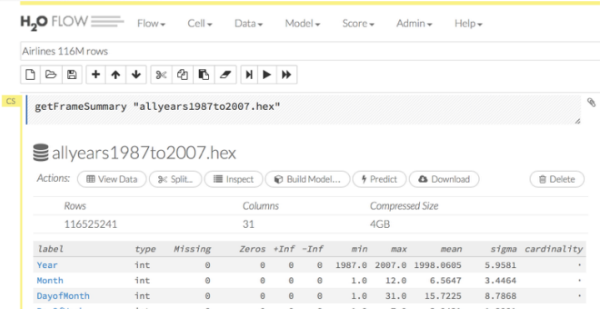
- Standard GLM: Beta = min -loglikelihood(data)
- Penalized GLM: Beta = min -loglikelihood(data) + penalty(Beta)
- Elastic Net Penalty
  - p(Beta) = lambda * (alpha*l1norm(Beta) + (1-alpha)*l2norm2(Beta))
  - lambda is regularization strength
  - alpha controls distribution of penalty between l1 and l2
    - alpha = 0 -> Ridge regression
    - alpha = 1 -> Lasso
    - 0 < alpha < 1 -> Elastic Net
- Lambda Search:
  - Efficient search of whole regularization path
  - Build the solution from the ground up
  - Allows computation of a sparse solution of very wide datasets

3、结果展示

h2o flow 界面自带一些结果图，其他的结果可以通过 ggplot2 画出来。

Scales to “Big Data”

Airline Data: Predict Delayed Departure



20 years of domestic airline flight data

116M rows
31 colums
12 GB CSV
4 GB compressed

"Big Data"

EC2 Demo Cluster: 8 nodes, 64 cores

H2O FLOW — Flow · Cell · Data · Model · Score · Admin · Help

Higgs 10M rows DL

Jobs
Cluster Status
Water Meter (CPU meter)

```
getCloud
```

H2ODemo

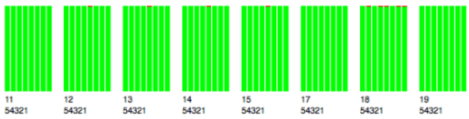
CLOUD STATUS

HEALTHY CONSENSUS LOCKED

Version	Started	Nodes (Used / All)
3.1.0.3013	10 hours ago	8 / 8

NODES

Name	Ping	Cores	Load	My CPU %	Sys CPU %
10.10.0.11:54321	a few seconds ago	8	7.090	100	100
10.10.0.12:54321	a few seconds ago	8	6.860	100	100
10.10.0.13:54321	a few seconds ago	8	6.820	99	100
10.10.0.14:54321	a few seconds ago	8	7.680	99	100
10.10.0.15:54321	a few seconds ago	8	6.450	99	100
10.10.0.17:54321	a few seconds ago	8	7.540	99	100
10.10.0.18:54321	a few seconds ago	8	7.680	100	100
10.10.0.19:54321	a few seconds ago	8	6.720	100	100
TOTAL	-	64	56.840	-	-



Legend

Each bar represents one CPU.

Blue: idle time
Green: user time
Red: system time
White: other time (e.g. i/o)

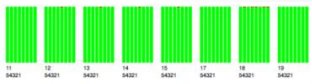
Node utilization during computation

"Big Data"(2)

allyears1987to2007.hex

Actions: View Data · Split... · Inspect · Build Model... · Predict · Download

Rows	Columns	Compressed Size
116525241	31	4GB



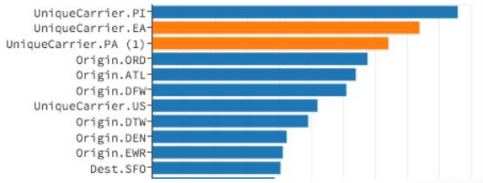
All cores maxed out

Logistic Regression: ~20s

elastic net, alpha=0.5, lambda=1.379e-4 (auto)

MSE	0.231287
r2	0.053357
logloss	0.654437
AUC	0.634441

STANDARDIZED COEFFICIENT MAGNITUDES



Chicago, Atlanta, Dallas:
often delayed