Regression modelling for massive data sets

Ioannis Kosmidis
Department of Statistical Science
University College London
London, United Kingdom
i.kosmidis@ucl.ac.uk

12 November 2015
Department of Economics and Business
Universitat Pompeu Fabra
Barcelona, Spain

Outline

Resources

- Delivered resources include:
  - this set of slides
  - an R script with code chunks to reproduce the analyses herein
- You may install the R packages that are used or mentioned on these slides by typing:

```
# Update the installed packages first
update.packages(ask = FALSE, repos = 'http://cran.rstudio.com/')
# Install the packages for this tutorial
PFpackages <- c('biglm', 'ffbase', 'ggplot2', 'sgd')
install.packages(PFpackages, repos = 'http://cran.rstudio.com/')
```

- The slides contain some exercises as footnotes. For example, see below[1].

[1] exercise: install the **R** packages that are used in this tutorial.

Package dependencies

- You can ensure that all code chunks will work for you if, prior to copying an pasting the contents of the code chunks, you run the following code chunk.

```
# ffbase Provides support for data.frame like objects that connect to
# files stored out of memory
require(ffbase)
# biglm implements a bounded-memory fitter for glms
require(biglm)
# ggplot2 is for flexible plotting
require(ggplot2)
# sgd is for stochastic gradient descent methods
require(sgd)
```

- Below you can set the variable `mydir` to the directory where the "big data" sets we will be working with can be stored (ensure you have at least 50 GB of free space in the corresponding drive). The value of `mydir` for me is on an external hard drive (USB 3 is preferred if your machine and hard drive support it).

```
mydir <- "/Volumes/IK_PORTABLE/BigReg/"
```

Outline

Higgs data

- Relevant paper:

  Baldi, P., P. Sadowski, and D. Whiteson (2014). Searching for Exotic Particles in High-energy Physics with Deep Learning. *Nature Communications* **5**.
  `http://doi.org/10.1038/ncomms5308`

Higgs data

- Variables:
  - `signal` (0 if background, 1 if signal process for Higgs boson);
  - 21 basic features (`feature1`, ..., `feature21` with real values);
  - 7 high-level features (`HLfeature1`, ..., `HLfeature7` with real values) which are appropriate functions of the 21 basic features.
- Data[2]:
  - training set: 10.5×10^6 observations;
  - test set: 0.5×10^6 observations.
- Task: Distinguish between a signal process and a background process based on basic and/or high-level features.

[2]Source: `https://archive.ics.uci.edu/ml/datasets/HIGGS`; data was produced using Monte Carlo simulations.

Higgs data: download and prepare `ffdf` object

```
HiggsDir <- paste0(mydir, "/", "HIGGS"); cdir <- getwd()
HiggsURL <- "https://goo.gl/3j9Jpr"
## Download the data: this will download 2.6 GB in HiggsDir. If this does
## not work for you, then copy and paste the link to a browser.
setwd(HiggsDir); download.file(dataURL, "HIGGSdata.csv.gz")
## decompress: this takes a while and will create an 8.0 GB csv file. For
## linux/Mac OSX systems the following command should work out of the box.
## For windows install 7zip (http://www.7-zip.org) and extract manually.
system("gunzip -kdv HIGGSdata.csv.gz")
## Read the HIGGS data in a ffdf object: Had to wait ~ 15 minutes
## on my laptop for the following to complete
varnames <- c("signal", paste0("feature", 1:21), paste0("HLfeature",1:7))
Higgs_ffdf <- read.csv.ffdf(file = "HIGGSdata.csv",
                            header = FALSE,
                            VERBOSE = TRUE,
                            next.rows = 5e+05,
                            col.names = varnames,
                            colClasses = rep("double", length(varnames)))
## set test variable (0 if observation is for training, 1 if for test)
Higgs_ffdf$test <- c(ff(0, 10500000), ff(1, 500000))
## Write ffdf object to the disk and go back to the working directory
save.ffdf(Higgs_ffdf, dir = "./HIGGSffdf", overwrite = TRUE); setwd(cdir)
```

Airline performance data

- Data were made available as part of the "Data expo 2009" ASA competition[3].
- Data provides arrival and departure details for *all* commercial flights within the USA, form October 1987 to April 2008.
- There are nearly 120×10^6 records on 29 variables including:
  - departure delays (`DepDelay`) and arrival delays (`ArrDelay`) (in minutes; can be negative);
  - date/time information (e.g. `Year`, `DepTime`, `ArrTime`);
  - origin and destination airports (`Origin`, `Dest`), and so on.
- Overall aim: investigate relationships between delays and other variables.

[3] Source: `http://stat-computing.org/dataexpo/2009/`

Airline performance data: download ...

```
AirlinesDir <- paste0(mydir, "/", "airlines"); cdir <- getwd()
setwd(AirlinesDir)
years <- 1987:2008 ## Years to download
## Download and decompress the data for all years: this takes a while and
## will require 13.7 GB. If this does not work for you then download
## manually from http://stat-computing.org/dataexpo/2009/the-data.html.
## For linux/Mac OSX systems the following command should work out of
## the box. For windows install 7zip (http://www.7-zip.org) and extract
## manually.
for (year in years) {
    filename <- paste0(year, ".csv.bz2")
    dataurl <- paste0("http://stat-computing.org/dataexpo/2009/",
                      filename)
    download.file(dataurl, filename)
    system(paste("bzip2 -kdv", filename))
}
setwd(cdir)
```

... and prepare `ffdf` object

```
setwd(AirlinesDir)
varnames <- c("Year", "Month", "DayofMonth", "DayOfWeek", "DepTime",
              "CRSDepTime", "ArrTime", "CRSArrTime", "UniqueCarrier",
              "FlightNum", "TailNum", "ActualElapsedTime",
              "CRSElapsedTime", "AirTime", "ArrDelay", "DepDelay",
              "Origin", "Dest", "Distance", "TaxiIn", "TaxiOut",
              "Cancelled", "CancellationCode", "Diverted", "CarrierDelay",
              "WeatherDelay", "NASDelay", "SecurityDelay", "LateAircraftDelay")
colclasses <- c(rep("double", 8), "factor", "double", "factor",
                rep("double", 5), rep("factor", 2), rep("double", 11))
Airlines <- read.csv.ffdf(file = paste0(AirlinesDir, "/", years[1], ".csv"),
                          header = TRUE, next.rows = 5e+05,
                          col.names = varnames, colClasses = colclasses,
                          VERBOSE = TRUE)
for (year in years[-1]) {
    filepath <- paste0(AirlinesDir, "/", year, ".csv")
    yeardata <- read.csv.ffdf(file = filepath,
                              header = TRUE, next.rows = 5e+05,
                              col.names = varnames, VERBOSE = TRUE,
                              colClasses = if (year < 2003) colclasses else NA)
    Airlines <- ffdfappend(Airlines, yeardata); delete(yeardata)
}
save.ffdf(Airlines, dir = paste0(AirlinesDir, "/", "airlinesffdf"),
          overwrite = TRUE)
setwd(cdir)
```