Machine Learning

using H2O

What is H2O?

- machine learning and predictive analytics platform
- open source
- free to use
- distributed and parallel
- in-memory
- fast and scalable
- easy to productionize
- core written in Java
- interfaces from webUI, REST API, R, Python, Scala, Hadoop, Spark

Install and start H2O

SystemRequirements: Java (>= 1.7)

- OpenJDK: openjdk.java.net
- Oracle Java SE: www.oracle.com/technetwork/java/javase

```
# install h2o
install.packages("h2o")

# load h2o
library(h2o)

# initialize h2o instance
h2o.init()
```

seeds data set: wheat kernels properties

Measurements of geometrical properties of kernels belonging to three different varieties of wheat.

```
# import dataset using base R
df <- read.csv("seeds_dataset.txt", sep="\t", header=FALSE)

# import dataset using data.table
dt <- data.table::fread("seeds_dataset.txt")

# import dataset using H2O
hf <- h2o.uploadFile("seeds_dataset.txt")
```

base R k-means

```
head(df)
# exclude actual cluster info from dataset
dat <- df[, paste0("V", 1:7)]
# run k-means using 3 clusters
km1 <- kmeans(dat, centers = 3)
# preview model info
km1
# plot all variable pairs
plot(dat, col = km1$cluster + 1,
     main = "base R k-means",
     pch = 20, cex = 2)
```

H2O k-means

```
hf
# run k-means on C1-C7 variables, using 3 clusters
km2 <- h2o.kmeans(hf, x = paste0("C", 1:7), k = 3)
# preview model info
km2
# run prediction, exclude actual cluster info
p <- h2o.predict(km2, hf[, -8])
# plot all variable pairs
plot(as.data.frame(hf[, -8]), col = as.vector(p) + 2,
     main = "H2O k-means",
     pch = 20, cex = 2)
```

k-means: unify cluster labels

```
# inspect labels
df[, "V8"]   # actuals
km1$cluster  # base R
as.vector(p) # h2o
# decode base R cluster id
base_r_cluster_id <- km1$cluster
base_r_cluster <- ifelse(base_r_cluster_id==3, 1,
ifelse(base_r_cluster_id==1, 2, 3))
# decode H2O cluster id
h2o_cluster_id <- as.vector(p)
h2o_cluster <- ifelse(h2o_cluster_id==2, 1,
ifelse(h2o_cluster_id==0, 2, 3))
```

k-means metric: root mean square error

```
# root mean square error
rmse <- function(predicted, actuals) {
  sqrt( mean( (predicted-actuals)^2, na.rm=TRUE ) )
}

# base R k-means rmse
rmse(base_r_cluster, df[, "V8"])

# H2O k-means rmse
rmse(h2o_cluster, as.vector(hf[, "C8"]))
h2o.make_metrics(as.h2o(h2o_cluster), hf[, "C8"])
```

k-means metric: confusion matrix

```
# confusion matrix
table(base_r_cluster, df[, "V8"])
table(h2o_cluster, as.vector(hf[, "C8"]))

# correct classification ratio
base_r_correct <- sum(base_r_cluster == df[, "V8"])
h2o_correct <- sum(h2o_cluster == as.vector(hf[, "C8"]))
sprintf("base R k-means correct ratio: %.2f%%",
        base_r_correct * 100 / nrow(df))
sprintf("H2O k-means correct ratio: %.2f%%",
        h2o_correct * 100 / nrow(df))
```

cluster dendrogram

```
# calculate euclidean distance matrix
d <- dist(dat, method = "euclidean")

# hierarchical cluster analysis
hc <- hclust(d, method = "ward.D")

# plot hclust object
plot(hc)

# draw rectangles
rect.hclust(hc, k = 3, border = "red")
```

bivariate cluster plot

```
library(cluster)
# combine two plots horizontally
par(mfrow = c(1,2))
# plot base R k-means clusters
clusplot(dat, km1$cluster, color=TRUE, shade=TRUE, labels=2,
lines=0, main="base R k-means")
# plot H2O k-means clusters
clusplot(dat, as.vector(p)+1, color=TRUE, shade=TRUE,
labels=2, lines=0, main="H2O k-means")
# reset plot option to default
par(mfrow = c(1,1))
```

k-means summary

K-means is a clustering algorithm. It measure the distance between observations in dataset and cluster them into partitions. It is commonly applied in various industries:

- biology, computational biology and bioinformatics
- medicine
- business and marketing
- world wide web
- computer science
- social science
- others

more on wikipedia: Cluster analysis

H2O algorithms

Supervised Learning:

- Generalized Linear Modeling (GLM)
- Gradient Boosting Machine (GBM)
- Deep Learning
- Distributed Random Forest
- Naive Bayes
- Stacked Ensembles

Unsupervised Learning:

- Generalized Low Rank Models (GLRM)
- K-Means Clustering
- Principal Components Analysis (PCA)

more on docs.h2o.ai

H2O webUI: Flow

After starting H2O instance, for example using R command `h2o.init()`, user can open web browser and point it to localhost:54321 address to use H2O webUI.

H2O Flow is graphical user interface. User can import data and perform basic data science tasks from web browser. Flow is designed to be used as a notebook, like Jupyter Notebook (IPython) and similar. Therefore no programming skills are required to use H2O Flow. User can use multiple interfaces at the same time, for example load and pre-process data using R and h2o package, then perform statistical modeling in web browser with H2O Flow. Various interfaces will use the same H2O instance.

Homework:

- understand Euclidean distance and its application in K-means clustering
- preview H2O website www.h2o.ai
- play with H2O Flow
- provide feedback to Dr. Manoj Kumar Singh

Questions?

Jan Gorecki: github.com/jangorecki

Contact: **jan61ji@gmail.com**