CHAPTER 12.K-Means Clustering

Clustering 이란? K-Means 이란?

- Clustering 은 d-차원의 N 개의 객체를 유사한 것끼리 K 개의 그룹으로 묶는 process.
- 하나의 cluster 내에 있는 객체는 유사함.

Next, we'll formalize the K-Means clustering. Given n d-dimensional points:

$X_1 = (x_{11}, x_{12}, ..., x_{1d})$

$X_2 = (x_{21}, x_{22}, ..., x_{2d})$

...

$X_n = (x_{n1}, x_{n2}, ..., x_{nd})$

our goal is to partition these $\{X_1, X_2, ..., X_n\}$ into K clusters: $\{C_1, C_2, ..., C_k\}$. K-Means aims to find the positions μ_i $(i = 1, ..., K)$ of the clusters that minimize the distance from the data points to the cluster centroids. K-means clustering solves the following cost minimization algorithm:

$$\arg\min_{C} \sum_{i=1}^{k} \sum_{\mathbf{X}_j \in C_i} \| \mathbf{X}_j - \mu_i \|^2$$

Application Areas for Clustering

Marketing - 고객 구매이력 데이터셋에서 비슷한 구매패턴을 갖는 그룹을 발굴.

Document classification - 비슷한 접근 패턴을 찾아서 web log data 을 군집화

Insurance - 점재적인 사기보험과 같은 고비용 청부를 하는 자동차 보험 가입자를 그룹화

Run of Spark K-Means Implementation

- CentOS 6.7

- 사전 준비

```
sudo yum -y install libcurl-devel
```

- ubuntu 16.04
- 사전 준비

```
sudo apt-get install libcurl4-gnutls-dev
```

```
install.packages("sparklyr" , repos = 'http://cran.nexr.com')
install.packages("DBI" , repos = 'http://cran.nexr.com')

library(sparklyr)

## Warning: replacing previous import by 'magrittr::%>%' when loading
## 'sparklyr'

## Warning: replacing previous import by 'tibble::data_frame' when loading
## 'sparklyr'

#spark_available_versions()
#spark_install(version = "1.6.1")

if (nchar(Sys.getenv("SPARK_HOME")) < 1) {
  Sys.setenv(SPARK_HOME = "/home/biospin/.cache/spark/spark-1.6.1-bin-hadoop2.
6")
}
sc <- spark_connect(master = "local")

install.packages("dplyr", repos = "http://cran.nexr.com" )
install.packages('ggplot2', repos = 'http://cran.nexr.com')

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

iris_tbl <- copy_to(sc, iris)

## The following columns have been renamed:
## - 'Sepal.Length' => 'Sepal_Length' (#1)
## - 'Sepal.Width'  => 'Sepal_Width'  (#2)
## - 'Petal.Length' => 'Petal_Length' (#3)
## - 'Petal.Width'  => 'Petal_Width'  (#4)
```

```r
kmeans_model <- iris_tbl %>%
  select(Petal_Width, Petal_Length) %>%
  ml_kmeans(centers = 3)

# print our model fit
print(kmeans_model)
```

```
## K-means clustering with 3 clusters
##
## Cluster centers:
##   Petal_Width Petal_Length
## 1    0.246000     1.462000
## 2    2.037500     5.595833
## 3    1.342308     4.269231
##
## Within Set Sum of Squared Errors =  31.37136
```

```r
# predict the associated class
predicted <- sdf_predict(kmeans_model, iris_tbl) %>%
  collect
table(predicted$Species, predicted$prediction)
```

```
##
##               0  1  2
##   setosa     50  0  0
##   versicolor  0  2 48
##   virginica   0 46  4
```

```r
library(ggplot2)

# plot cluster membership
sdf_predict(kmeans_model) %>%
  collect() %>%
  ggplot(aes(Petal_Length, Petal_Width)) +
  geom_point(aes(Petal_Width, Petal_Length, col = factor(prediction + 1)),
             size = 2, alpha = 0.5) +
  geom_point(data = kmeans_model$centers, aes(Petal_Width, Petal_Length),
             col = scales::muted(c("red", "green", "blue")),
             pch = 'x', size = 12) +
  scale_color_discrete(name = "Predicted Cluster",
                       labels = paste("Cluster", 1:3)) +
  labs(
    x = "Petal Length",
    y = "Petal Width",
    title = "K-Means Clustering",
    subtitle = "Use Spark.ML to predict cluster membership with the iris data
set."
  )
```


K-Means Clustering
Use Spark.ML to predict cluster membership with the iris dataset.
Petal Width
Petal Length
0.0
0.5
1.0
1.5
2.0
2.5
2
4
6
Predicted Cluster
Cluster 1
Cluster 2
Cluster 3