MC884/MO444 - Aprendizado de Máquina

k-means

Erik de Godoy Perillo - RA135582

Universidade Estadual de Campinas

30 de outubro de 2016

1 Introdução

O objetivo do trabalho era explorar o algoritmo de *k-means* e suas métricas de desempenho.

1.1 Implementação

A linguagem de implementação escolhida foi o R. Todo o código utilizado no relatório encontra-se na seção 4. Ao longo do documento, linhas do código serão citadas para referência no mesmo. A função `main` (linha 18 da seção 4) executa tudo que é requisitado no enunciado, mostrando os resultados.

2 Metodologia

Os parâmetros para o *k-means* encontram-se na linha 11 da seção 4. Para métrica interna, foi selecionado o `dunn2`, que mede a razão entre a mínima dissimilaridade média entre clusters e a máxima dissimilaridade média intra-clusters. Para métrica externa, foi selecionado o *corrected rand*. A seleção dos melhores `k` para as duas métricas é feita no *loop* da linha 39 e as métricas são plotadas na linha 76.

3 Resultados

Para a métrica interna, o melhor `k` foi 2, com $dunn2 = 2.739$. Para a externa, o melhor `k` foi 4, com $rand = 0.258$. A saída da execução do código encontra-se na seção 5. Os *plots* encontram-se abaixo.

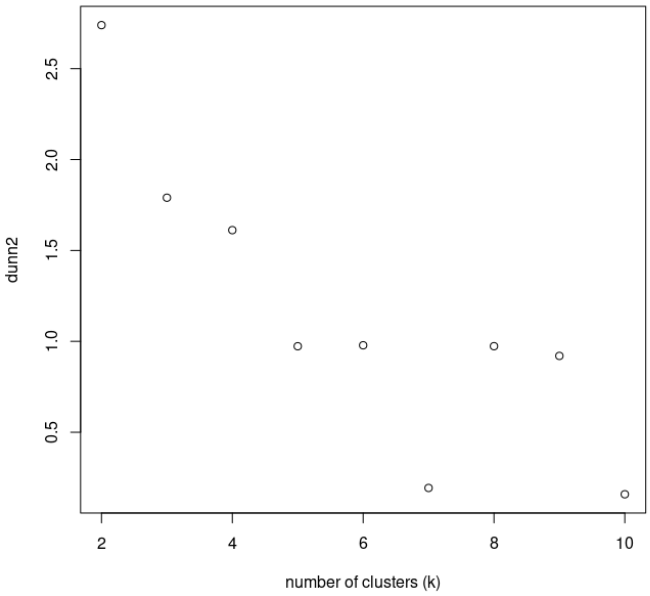

Figura 1: Métricas internas para cada k

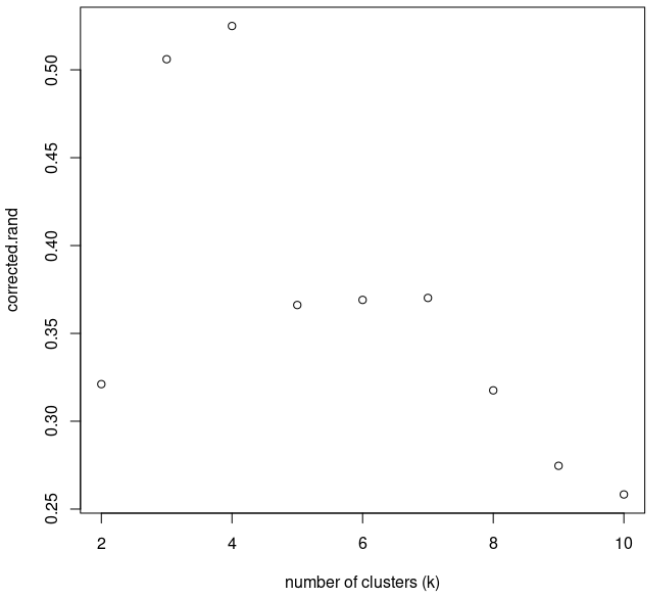

Figura 2: Métricas externas para cada k

4 Código-fonte

```r
#packages
library(caret)
library(stats)
library(fpc)

#file path of data
data_filepath <- "./cluster-data.csv"
labels_filepath <- "./cluster-data-class.csv"

#k-means parameters
ks <- seq(2, 10)
n_start <- 5

#wrapper for sprintf
printf <- function(...) cat(sprintf(...))

#main method for whole challenge
main <- function()
{
    #reading data
    x <- read.csv(data_filepath, header=TRUE, sep=",")
    x <- as.matrix(x)
    y <- read.csv(labels_filepath, header=TRUE, sep=",")
    y <- as.matrix(y[, 1])

    #best index of internal metrics and external metrics
    int_best_id <- 1
    ext_best_id <- 1

    #internal and external metrics values
    int_metrics <- c()
    ext_metrics <- c()

    #distances between points in x
    printf("getting distances object for points... ")
    dst <- dist(x)
    printf("done.\n")

    for(i in seq(length(ks)))
    {
        printf("k = %d:\n", ks[i])

        #getting k-means
        printf("\tcomputing k-means... ")
        means <- kmeans(x, ks[i], nstart=n_start)
        printf("done.\n")

        #getting stats
        printf("\tcomputing clustering stats... ")
        stats <- cluster.stats(dst, means$cluster, y)
        printf("done.\n")

        #printing metrics
        printf("\tdunn2: %.6f | corrected.rand: %.6f\n",
            stats$dunn2, stats$corrected.rand)
        printf("\n")

        #appending metric for later plotting
        int_metrics <- c(int_metrics, stats$dunn2)
        ext_metrics <- c(ext_metrics, stats$corrected.rand)

        #checking for best metric
        if(stats$dunn2 > int_metrics[int_best_id])
            int_best_id <- i
        if(stats$corrected.rand > ext_metrics[ext_best_id])
            ext_best_id <- i
```

```
    }

    #printing best scores
    printf("best internal metric score: %.6f (k=%d)\n",
        int_metrics[int_best_id], ks[int_best_id])
    printf("best external metric score: %.6f (k=%d)\n",
        ext_metrics[ext_best_id], ks[ext_best_id])

    #plotting metrics
    printf("plotting metrics...\n")
    dev.new()
    plot(int_metrics ~ ks,
         xlab="number of clusters (k)", ylab="dunn2")
    dev.new()
    plot(ext_metrics ~ ks,
         xlab="number of clusters (k)", ylab="corrected.rand")

}
```

5 Saída do código

```
getting distances object for points... done.
k = 2:
        computing k-means... done.
        computing clustering stats... done.
        dunn2: 2.739483 | corrected.rand: 0.321113

k = 3:
        computing k-means... done.
        computing clustering stats... done.
        dunn2: 1.790139 | corrected.rand: 0.506041

k = 4:
        computing k-means... done.
        computing clustering stats... done.
        dunn2: 1.611723 | corrected.rand: 0.524948

k = 5:
        computing k-means... done.
        computing clustering stats... done.
        dunn2: 0.972936 | corrected.rand: 0.366150

k = 6:
        computing k-means... done.
        computing clustering stats... done.
        dunn2: 0.972602 | corrected.rand: 0.369398

k = 7:
        computing k-means... done.
        computing clustering stats... done.
        dunn2: 0.971041 | corrected.rand: 0.348167

k = 8:
        computing k-means... done.
        computing clustering stats... done.
        dunn2: 0.820141 | corrected.rand: 0.290464

k = 9:
        computing k-means... done.
        computing clustering stats... done.
        dunn2: 0.823787 | corrected.rand: 0.256664

k = 10:
        computing k-means... done.
        computing clustering stats... done.
        dunn2: 0.155116 | corrected.rand: 0.247585

best internal metric score: 2.739483 (k=2)
best external metric score: 0.524948 (k=4)
plotting metrics...
```