rcppeasy.md

Aditya Samantaray

Easy Test

For an example for the kMeans function implemented in RcppMLPACK2, we'll use the trees and the wine datasets ## Including the libraries and dataset for this test

```
library("data.table")
library("RcppMLPACK")
library("microbenchmark")
library("directlabels")
library("ggplot2")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

```
data(wine, package = "rattle")
data(trees, package = "datasets")
```

Fitting the kMeans model with the RcppMLPACK package

```
kMeans(t(trees), 3)
```

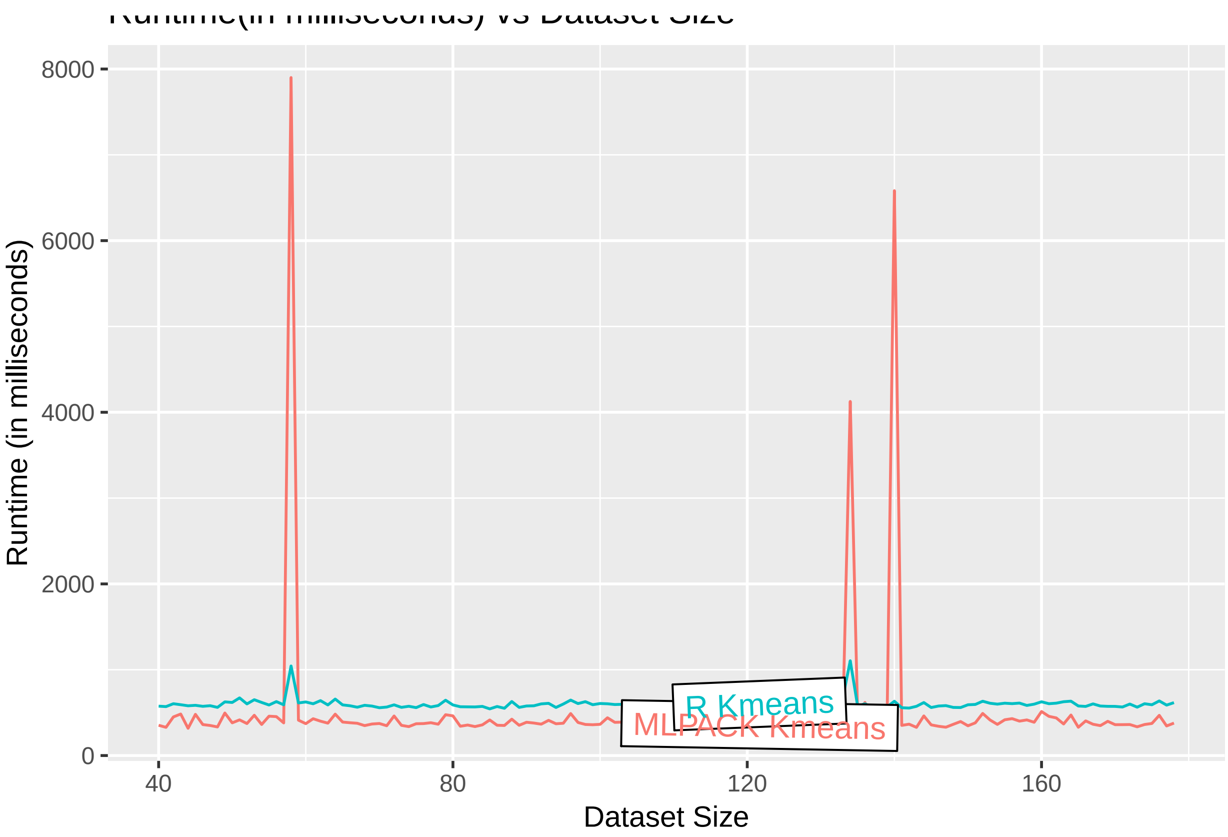
```
## $clusters
## [1] 3
##
## $result
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 2
```

```
wine$Type <- NULL
kMeans(t(wine), 4)
```

```
## $clusters
## [1] 4
##
## $result
##   [1] 1 1 1 3 0 3 3 3 1 1 3 3 3 1 3 3 3 1 3 0 0 0 1 1 0 0 1 3 1 1 3 3 1 3 1
##  [36] 1 1 1 1 0 0 1 1 0 1 1 1 1 1 3 1 3 1 3 1 1 1 3 3 2 0 2 0 2 2 0 2 2 0 0
##  [71] 0 2 2 1 1 2 2 2 0 2 2 0 0 2 2 2 2 2 0 0 2 2 2 2 2 1 0 2 0 2 0 2 2 2 0
## [106] 2 2 2 2 0 2 2 0 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 0 2 2 0 0 0 0 2 2 0
## [141] 0 0 2 2 0 0 2 0 0 2 2 2 2 0 0 0 2 1 0 0 2 0 2 0 0 2 0 0 0 0 2 2 0 0 0
## [176] 0 0 2
```

Now, we'll try to benchmark the RcppMLPACK kMeans implementation with the default kmeans implementation in R
Note: kMeans() is the RcppMLPACK implementation whereas kmeans() is the default implementation in R

```
X <- wine
res <- data.frame()
for(i in 40:nrow(X))
{
  evaltime <- microbenchmark(kMeans(t(X), 3), kmeans(X, 3), times = 100L)
  res <- bind_rows(res, data.frame(i, list(summary(evaltime)[,c('min','mean','max')])))
}
res <- cbind.data.frame(c("MLPACK Kmeans", "R Kmeans"), res)
names(res) <- c("expr", names(res)[2:5])
p <- ggplot(res, aes(x = i))+
  #geom_ribbon(aes(ymin = min, ymax = max, fill = expr, group = expr), alpha = 1/2)+
  geom_line(aes(y = mean, group = expr, colour = expr))+
  ggtitle('Runtime(in milliseconds) vs Dataset Size') +
  xlab('Dataset Size') +
  ylab('Runtime (in milliseconds)')
direct.label(p,"angled.boxes")
```


Runtime(in milliseconds) vs Dataset Size
Runtime (in milliseconds)
Dataset Size
R Kmeans
MLPACK Kmeans