Wk 8 Mini Project

Aaron Palumbo

11/1/2015

```
library(rmr2)
```

```
## Warning: S3 methods 'gorder.default', 'gorder.factor', 'gorder.data.frame',
## 'gorder.matrix', 'gorder.raw' were declared in NAMESPACE but not found
```

```
## Please review your hadoop settings. See help(hadoop.settings)
```

```
##########################################################
# We'll keep the kmeans map reduce function in tact. #
##########################################################

## @knitr kmeans-signature
kmeans.mr =
  function(
    P,
    num.clusters,
    num.iter,
    combine,
    in.memory.combine) {
## @knitr kmeans-dist.fun
    dist.fun =
      function(C, P) {
        apply(
          C,
          1,
          function(x)
            colSums((t(P) - x)^2))}
## @knitr kmeans.map
    kmeans.map =
      function(., P) {
        nearest = {
          if(is.null(C))
            sample(
              1:num.clusters,
              nrow(P),
              replace = TRUE)
          else {
            D = dist.fun(C, P)
            nearest = max.col(-D)}}
        if(!(combine || in.memory.combine))
          keyval(nearest, P)
        else
          keyval(nearest, cbind(1, P))}
## @knitr kmeans.reduce
    kmeans.reduce = {
      if (!(combine || in.memory.combine) )
```

```
        function(., P)
          t(as.matrix(apply(P, 2, mean)))
      else
        function(k, P)
          keyval(
            k,
            t(as.matrix(apply(P, 2, sum))))}
## @knitr kmeans-main-1
    C = NULL
    for(i in 1:num.iter ) {
      C =
        values(
          from.dfs(
            mapreduce(
              P,
              map = kmeans.map,
              reduce = kmeans.reduce)))
      if(combine || in.memory.combine)
        C = C[, -1]/C[, 1]
## @knitr end
#      points(C, col = i + 1, pch = 19)
## @knitr kmeans-main-2
      if(nrow(C) < num.clusters) {
        C =
          rbind(
            C,
            matrix(
              rnorm(
                (num.clusters -
                   nrow(C)) * nrow(C)),
              ncol = nrow(C)) %*% C) }}
        C}
```

Now we'll load in out data:

```
birch1 <- read.table("birch1.txt")
```

And run the map reduce job

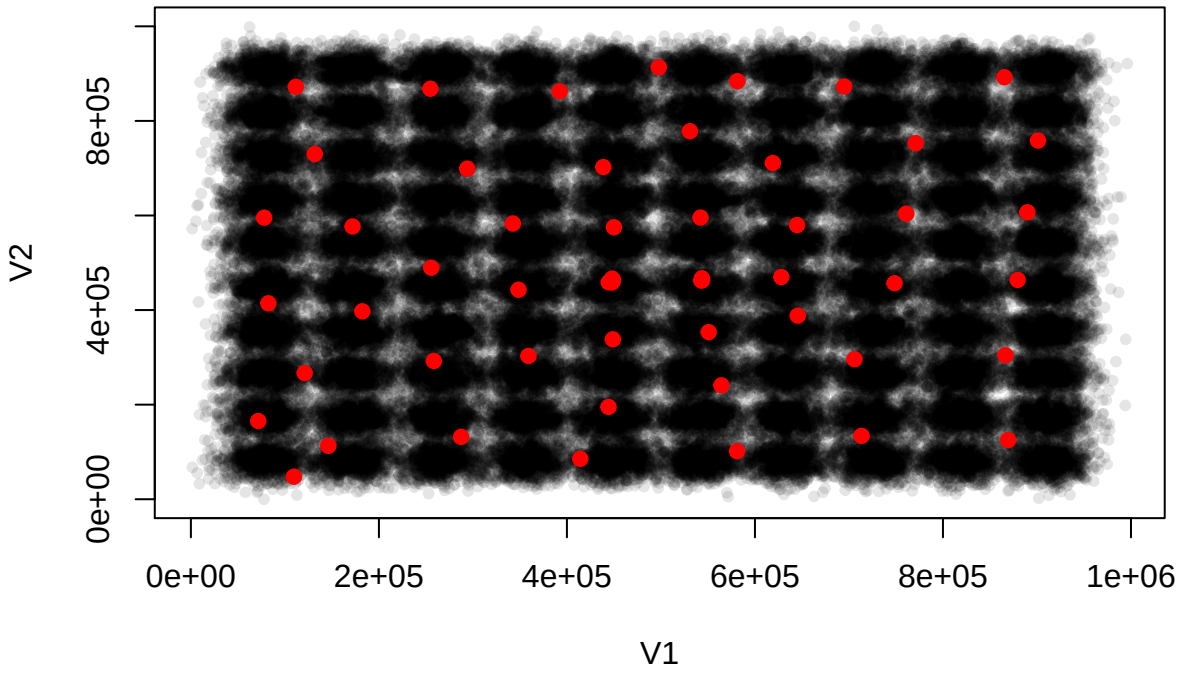
```
## Points
birch1 <- as.matrix(birch1)

## Now let's see what happens

out <- kmeans.mr(
          to.dfs(birch1),
          num.clusters = 100,
          num.iter = 10,
          combine = FALSE,
          in.memory.combine = FALSE)
```

Let's see how we did:

```
plot(birch1, pch=20, col=rgb(0, 0, 0, alpha=0.1))
points(out, pch=19, col="red")
```



The performance is not that great. It seems to be doing okay with a few toward the center, but it misses a lot of the outer centers. This is currently run with only 10 iterations. Perhaps the accuracy would improve with more iterations.