June 8, 2022

The results below are generated from an R script.

```
## Install a package manager and packages
if (!require("pacman")) {
  install.packages("pacman")
}
pacman::p_load(Rfast, foreach, doParallel, mvnfast, rstudioapi)
current_path = rstudioapi::getActiveDocumentContext()$path
setwd(dirname(current_path))

pacman::p_load_gh("pkimes/sigclust2")
shc = get("shc", env = environment(shc))

source("sequential_function.R")

# k = 3 # number of clusters (3 or 10)
# uneven = FALSE #whether or not to have uneven weights
# distribution = 't' # t distribution or normal distribution?
# iterations = 50 # number of iterations


n = 500 # total number of samples
alpha = 0.05


if (distribution=='t'){
  distribution_name = 'True distribution components: t-distrbution (df=3) mixture distribution'
  samplefunc <- function(n, mu, sigma, w){
    rmixt(n = n,mu = mu,sigma = sigma,w = w,df = 3)
  }
}else{
  distribution_name = 'True distribution: Normal mixture distribution'
  samplefunc <- function(n, mu, sigma, w){
  rmixn(n=n, mu=mu, sigma=sigma, w=w)
}
}


# formulating d, delta (dimension and distance between clusters)
if (k == 10){
  a = c(2, 20, 2, 40, 2, 60, 2, 80, 2, 100, 2, 150, 2, 200) # dim2
  b =c(8, 20, 8, 40, 8, 60, 8, 80, 8, 100, 8, 150, 8, 200) # dim8
  d_delta = matrix(c(a, b) , ncol = 2, byrow = T)
} else if (k == 3){
  a = c(2, 1, 2, 2, 2, 3, 2, 4, 2, 5, 2, 6, 2, 7, 2, 8, 2, 9) # dim2
```

```
  b = c(8, 1, 8, 2, 8, 3, 8, 4, 8, 5, 8, 6, 8, 7, 8, 8, 8, 9) # dim8
  d_delta = matrix(c(a, b) , ncol = 2, byrow = T)
} else {
  stop("k != 3 or 10")
}


#weights
w = rep.int(1, k)
if (uneven){
  w[1] = 1 / 4
  w[2] = 1 / 2
}
w = w / sum(w)


K = floor(sqrt(n / 2)) #num clusters to test
K = min(K, 14L) # to ensure not estimating too many clusters

coresToUse = floor(detectCores() / 2) # cores to use

# function which creates data and performs one iteration
simulation <- function(iteration) {

  mu = matrix(runif(k*d, min = 0, max = delta), nrow = k)

  # simulate data
  set.seed(18 + iteration)
  data = samplefunc(n=n, mu=mu, sigma=sigma, w=w)


  D1 = data[1:floor(n / 2), ]
  D2 = data[(floor(n / 2) + 1):n, ]

  # Estimate no.clusters
  Cluster_numbers = estimate.cluster.all(D1, D2, alpha, K)
  sigclust_splits = sum(shc(data, alpha = alpha)$nd_type == "sig")
  return(c(unlist(Cluster_numbers, use.names = F), sigclust_splits + 1L))
}


meanEstimate = matrix(nrow = nrow(d_delta), ncol = iterations)
medianEstimate = meanEstimate
meanEstimatel2 = meanEstimate
medianEstimatel2 = meanEstimate
AICEstimate = meanEstimate
BICEstimate = meanEstimate
sigclustEstimate = meanEstimate
RIFThierEstimate = meanEstimate

# For parallel computing
cl <- makeCluster(coresToUse) #not to overload computer
registerDoParallel(cl)
```

```
for (j in 1:nrow(d_delta)) {
  d = d_delta[j, 1]
  delta = d_delta[j, 2]



  #sigma = lapply(c(3,1,1), function(x) diag(x, nrow=d))
  sigma = lapply(rep.int(1, k), function(x)
    diag(x, nrow = d))




  estimates <-
    foreach(
      i = 1:iterations,
      .combine = cbind,
      .inorder = F,
      .packages = c("mclust", "Rfast", "mvnfast", "MASS"),
      .verbose = F
    ) %dopar% {
      simulation(i)
    }

  # format data into table
  meanEstimate[j, ] = estimates[1, ]
  medianEstimate[j, ] = estimates[2, ]
  meanEstimatel2[j, ] = estimates[3, ]
  medianEstimatel2[j, ] = estimates[4, ]
  BICEstimate[j, ] = estimates[5, ]
  AICEstimate[j, ] = estimates[6, ]
  RIFThierEstimate[j, ] = estimates[7, ]
  sigclustEstimate[j, ] = estimates[8, ]
  df = stack(data.frame(
    cbind(
      "Mean" = meanEstimate[j, ],
      "Meanl2" = meanEstimatel2[j, ],
      "Median" = medianEstimate[j, ],
      "Medianl2" = medianEstimatel2[j, ],
      "AIC" = AICEstimate[j, ],
      "BIC" = BICEstimate[j, ],
      "RIFT.hc" = RIFThierEstimate[j, ],
      "shc" = sigclustEstimate[j, ]
    )
  ))
  print(paste0("(dimension, delta) = (", d, ",", delta, ")"))
  colnames(df) = c("ESTIMATE" , "METHOD")
  tableEstimates = with(df, table(METHOD, ESTIMATE))
  print(tableEstimates)
}
```

```
## [1] "(dimension, delta) = (2,20)"
##           ESTIMATE
```

```
## METHOD      1  3  4  5  6  7  8  9 10 11 12 13 14 16 18
##   Mean      0  0  6 10 17 18 25 22  2  0  0  0  0  0  0
##   Meanl2    0  0  6 10 17 18 25 22  2  0  0  0  0  0  0
##   Median    0  0  1  7 17 11 18 21 12  7  1  0  5  0  0
##   Medianl2  0  0  1  7 17 11 18 21 12  7  1  0  5  0  0
##   AIC       0  0  0  1  3 11 21 39 23  2  0  0  0  0  0
##   BIC       0  0  3  2  8 16 27 33 11  0  0  0  0  0  0
##   RIFT.hc  11  3 12 18 26 21  9  0  0  0  0  0  0  0  0
##   shc       1  0  1  0  1  7 11 27 17 16  9  2  5  2  1
## [1] "(dimension, delta) = (2,40)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14
##   Mean      0  0  0  0  2  2 10 21 35 28  2  0  0  0
##   Meanl2    0  0  0  0  2  2 10 21 35 28  2  0  0  0
##   Median    0  0  0  0  1  6 17 21 12 20  5  4  4 10
##   Medianl2  0  0  0  0  2  7 16 20 12 19  8  4  5  7
##   AIC       0  0  0  0  0  1  2 18 34 41  4  0  0  0
##   BIC       0  0  0  0  0  1  2 15 38 41  3  0  0  0
##   RIFT.hc   8  2  7  2  4  6 11 25 23 11  1  0  0  0
##   shc       1  1  0  0  0  0  0  2 22 51 16  4  3  0
## [1] "(dimension, delta) = (2,60)"
##           ESTIMATE
## METHOD      1  3  4  5  6  7  8  9 10 11 12 13 14
##   Mean      0  0  0  0  3 17 30 29 18  1  2  0  0
##   Meanl2    0  0  0  0  3 17 30 29 18  1  2  0  0
##   Median    0  0  0  4 17 22 23 17  5  3  2  2  5
##   Medianl2  0  0  0  5 18 22 24 13  5  3  3  3  4
##   AIC       0  0  0  0  2 13 24 32 26  1  2  0  0
##   BIC       0  0  0  0  1 15 23 32 26  1  2  0  0
##   RIFT.hc  11  4  3  2  2  4 15 27 30  2  0  0  0
##   shc       0  0  0  0  0  0  2 11 71 12  2  2  0
## [1] "(dimension, delta) = (2,80)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14
##   Mean      0  0  0  0  0  5 10 25 26 33  0  0  1  0
##   Meanl2    0  0  0  0  0  5 10 25 26 33  0  0  1  0
##   Median    0  0  0  1  3 10 19 24 16  6  1  4  7  9
##   Medianl2  0  0  0  1  4  9 25 21 13  7  1  4  8  7
##   AIC       0  0  0  0  0  3 10 18 33 34  1  0  1  0
##   BIC       0  0  0  0  0  3  9 20 32 35  0  0  1  0
##   RIFT.hc  15  4  6  6  1  2  7 13 16 30  0  0  0  0
##   shc       2  0  0  0  0  0  0  2 12 71 12  1  0  0
## [1] "(dimension, delta) = (2,100)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14
##   Mean      0  0  0  0  0  3  6 23 43 25  0  0  0  0
##   Meanl2    0  0  0  0  0  3  6 23 43 25  0  0  0  0
##   Median    0  0  0  3  0 12 14 30 27  5  3  1  1  4
##   Medianl2  0  0  0  3  1 14 17 30 21  5  3  0  1  5
##   AIC       0  0  0  0  0  3  4 22 45 26  0  0  0  0
##   BIC       0  0  0  0  0  3  4 21 41 31  0  0  0  0
##   RIFT.hc  12  2  3  4  2  7  3  6 18 43  0  0  0  0
##   shc       0  0  0  0  0  0  0  0  2 86 11  1  0  0
## [1] "(dimension, delta) = (2,150)"
```

```
##           ESTIMATE
## METHOD       1  3  4  5  6  7  8  9 10 11 12 13 14
##   Mean       0  0  0  0  2  1 25 42 29  0  0  0  1
##   Meanl2     0  0  0  0  2  1 25 42 29  0  0  0  1
##   Median     0  0  1  3  7 15 23 26  6  1  3  7  8
##   Medianl2   0  0  1  5  9 20 22 18  7  2  2  5  9
##   AIC        0  0  0  0  2  0 24 39 34  1  0  0  0
##   BIC        0  0  0  0  2  0 24 39 34  1  0  0  0
##   RIFT.hc   17  9  8  3  3  1  2 11 46  0  0  0  0
##   shc        0  0  0  0  0  0  0  1 84 14  1  0  0
## [1] "(dimension, delta) = (2,200)"
##           ESTIMATE
## METHOD       1  2  3  4  5  6  7  8  9 10 11 12 13 14
##   Mean       0  0  0  0  0  1  6 20 43 30  0  0  0  0
##   Meanl2     0  0  0  0  0  1  6 20 43 30  0  0  0  0
##   Median     0  0  0  0  2 16  6 27 20 13  2  1  6  7
##   Medianl2   0  0  0  0  3 20 15 23 11 12  3  0  7  6
##   AIC        0  0  0  0  0  1  6 18 44 30  1  0  0  0
##   BIC        0  0  0  0  0  1  5 19 44 31  0  0  0  0
##   RIFT.hc   18  4  6  6  4  7  0  3 11 39  2  0  0  0
##   shc        0  0  0  0  0  0  0  0  2 88 10  0  0  0
## [1] "(dimension, delta) = (8,20)"
##           ESTIMATE
## METHOD        8   9  10  11  12  13
##   Mean        0   0 100   0   0   0
##   Meanl2      0   0 100   0   0   0
##   Median      0  19  77   2   1   1
##   Medianl2    0  19  77   2   1   1
##   AIC         0   0 100   0   0   0
##   BIC         0   0 100   0   0   0
##   RIFT.hc    20  10  70   0   0   0
##   shc         0   0  84  13   2   1
## [1] "(dimension, delta) = (8,40)"
##           ESTIMATE
## METHOD        8   9  10  11  12  13
##   Mean        0   0 100   0   0   0
##   Meanl2      0   0 100   0   0   0
##   Median      0   3  94   1   0   2
##   Medianl2    0   3  94   1   0   2
##   AIC         0   0 100   0   0   0
##   BIC         0   0 100   0   0   0
##   RIFT.hc     9  11  80   0   0   0
##   shc         0   0  81  17   2   0
## [1] "(dimension, delta) = (8,60)"
##           ESTIMATE
## METHOD        8   9  10  11  12  13
##   Mean        0   0 100   0   0   0
##   Meanl2      0   0 100   0   0   0
##   Median      0   1  96   2   0   1
##   Medianl2    0   1  96   2   0   1
##   AIC         0   0 100   0   0   0
##   BIC         0   0 100   0   0   0
##   RIFT.hc     5   9  86   0   0   0
##   shc         0   0  84  14   1   1
```

```
## [1] "(dimension, delta) = (8,80)"
##           ESTIMATE
## METHOD       8   9  10  11  12  14
##   Mean       0   0 100   0   0   0
##   Meanl2     0   0 100   0   0   0
##   Median     1   0  96   1   0   2
##   Medianl2   1   0  96   1   0   2
##   AIC        0   0 100   0   0   0
##   BIC        0   0 100   0   0   0
##   RIFT.hc    5  10  85   0   0   0
##   shc        0   0  89  10   1   0
## [1] "(dimension, delta) = (8,100)"
##           ESTIMATE
## METHOD       8   9  10  11  12  14
##   Mean       0   0 100   0   0   0
##   Meanl2     0   0 100   0   0   0
##   Median     0   3  95   1   0   1
##   Medianl2   0   3  95   1   0   1
##   AIC        0   0 100   0   0   0
##   BIC        0   0 100   0   0   0
##   RIFT.hc    7   9  84   0   0   0
##   shc        0   0  88  10   2   0
## [1] "(dimension, delta) = (8,150)"
##           ESTIMATE
## METHOD       8   9  10  11  12  14
##   Mean       0   0 100   0   0   0
##   Meanl2     0   0 100   0   0   0
##   Median     0   3  95   0   0   2
##   Medianl2   0   3  95   0   0   2
##   AIC        0   0 100   0   0   0
##   BIC        0   0 100   0   0   0
##   RIFT.hc    3  10  86   1   0   0
##   shc        0   0  83  14   3   0
## [1] "(dimension, delta) = (8,200)"
##           ESTIMATE
## METHOD       8   9  10  11  12  14
##   Mean       0   0 100   0   0   0
##   Meanl2     0   0 100   0   0   0
##   Median     0   2  93   2   0   3
##   Medianl2   0   2  93   2   0   3
##   AIC        0   0 100   0   0   0
##   BIC        0   0 100   0   0   0
##   RIFT.hc    1   9  89   1   0   0
##   shc        0   0  85  12   3   0
```

```
#stop cluster (parallel computing)
stopCluster(cl)

print(distribution_name )
```

```
## [1] "True distribution: Normal mixture distribution"
```

```
print(paste(k, 'true clusters:'))
```

```
## [1] "10 true clusters:"
```

```
print('Cluster weights:')

## [1] "Cluster weights:"

print(w)

##  [1] 0.02857143 0.05714286 0.11428571 0.11428571 0.11428571 0.11428571 0.11428571
##  [8] 0.11428571 0.11428571 0.11428571
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Monterey 12.0.1
##
## Matrix products: default
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
##
## attached base packages:
## [1] grid      parallel  stats     graphics  grDevices utils     datasets  methods
## [9] base
##
## other attached packages:
##  [1] knitr_1.37         sigclust_1.1.0     mixtools_1.2.0     gridExtra_2.3
##  [5] ggplot2_3.3.5      MASS_7.3-54        pracma_2.3.6       mclust_5.4.9
##  [9] sigclust2_1.2.4    rstudioapi_0.13    mvnfast_0.2.7      doParallel_1.0.16
## [13] iterators_1.0.13   foreach_1.5.1      Rfast_2.0.6        RcppZiggurat_0.1.6
## [17] Rcpp_1.0.8         pacman_0.5.1
##
## loaded via a namespace (and not attached):
##   [1] colorspace_2.0-2       ellipsis_0.3.2         dynamicTreeCut_1.63-1
##   [4] rprojroot_2.0.3        htmlTable_2.4.0        XVector_0.34.0
##   [7] base64enc_0.1-3        ggdendro_0.1.23        fs_1.5.2
##  [10] remotes_2.4.2          bit64_4.0.5            AnnotationDbi_1.56.2
##  [13] fansi_0.5.0            codetools_0.2-18       splines_4.1.2
##  [16] cachem_1.0.6           impute_1.68.0          pkgload_1.2.4
##  [19] Formula_1.2-4          WGCNA_1.70-3           cluster_2.1.2
##  [22] kernlab_0.9-29         GO.db_3.14.0           png_0.1-7
##  [25] compiler_4.1.2         httr_1.4.2             backports_1.4.1
##  [28] Matrix_1.3-4           fastmap_1.1.0          cli_3.3.0
##  [31] htmltools_0.5.2        prettyunits_1.1.1      tools_4.1.2
##  [34] gtable_0.3.0           glue_1.6.1             GenomeInfoDbData_1.2.7
##  [37] dplyr_1.0.7            ggthemes_4.2.4         Biobase_2.54.0
##  [40] vctrs_0.4.1            Biostrings_2.62.0      preprocessCore_1.56.0
##  [43] xfun_0.30              fastcluster_1.2.3      stringr_1.4.0
##  [46] ps_1.7.0               brio_1.1.3             testthat_3.1.4
##  [49] lifecycle_1.0.1        devtools_2.4.3         zlibbioc_1.40.0
##  [52] scales_1.1.1           RColorBrewer_1.1-2     memoise_2.0.1
##  [55] rpart_4.1-15           segmented_1.3-4        latticeExtra_0.6-29
##  [58] stringi_1.7.6          RSQLite_2.2.10         highr_0.9
```

```
##  [61] S4Vectors_0.32.3       desc_1.4.1             checkmate_2.0.0
##  [64] BiocGenerics_0.40.0    pkgbuild_1.3.1         GenomeInfoDb_1.30.1
##  [67] rlang_1.0.2            pkgconfig_2.0.3        matrixStats_0.61.0
##  [70] bitops_1.0-7           evaluate_0.15          lattice_0.20-45
##  [73] purrr_0.3.4            htmlwidgets_1.5.4      bit_4.0.4
##  [76] tidyselect_1.1.1       processx_3.5.3         magrittr_2.0.2
##  [79] R6_2.5.1               IRanges_2.28.0         generics_0.1.1
##  [82] Hmisc_4.6-0            DBI_1.1.2              pillar_1.6.4
##  [85] foreign_0.8-81         withr_2.4.3            survival_3.2-13
##  [88] KEGGREST_1.34.0        RCurl_1.98-1.6         nnet_7.3-16
##  [91] tibble_3.1.6           crayon_1.4.2           utf8_1.2.2
##  [94] jpeg_0.1-9             usethis_2.1.6          data.table_1.14.2
##  [97] blob_1.2.2             callr_3.7.0            digest_0.6.29
## [100] stats4_4.1.2           munsell_0.5.0          sessioninfo_1.2.2
```

```
Sys.time()
```

```
## [1] "2022-06-08 14:25:06 BST"
```