June 2, 2022

The results below are generated from an R script.

```r
## 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 = mus,sigma = sigmas,w = w,df = 3)
  }
}else{
  distribution_name = 'True distribution: Normal mixture distribution'
  samplefunc <- function(n, mu, sigma, w){
  rmixn(n=n, mu=mus, sigma=sigmas, w=w)
}
}


# formulating d, delta (dimension and distance between clusters)
if (k == 10){
  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 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) {
  # simulate data
  set.seed(18 + iteration)
  data = samplefunc(n=n, mu=mus, sigma=sigmas, 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]



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


  #mus = zeros(k, d)
  #mus[1,1] = delta
  #mus[2,2] = -delta
  #mus[3,2] = delta
  mus = outer(rep.int(1L, k), seq.int(d)) + delta * seq.int(0, k - 1L)

  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) = (2,1)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 22
##   Mean     95  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Meanl2   95  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Median   86 14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Medianl2 86 14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   AIC      51 49  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   BIC      70 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   RIFT.hc  42 52  6  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   shc       2 25  1  6  6  9  6 10  6  8  5  3  4  6  1  1  1
## [1] "(dimension, delta) = (2,2)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 18
##   Mean     36 42 22  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Meanl2   36 42 22  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Median   16 30 50  2  1  0  1  0  0  0  0  0  0  0  0  0  0
##   Medianl2 16 30 50  2  1  0  1  0  0  0  0  0  0  0  0  0  0
##   AIC       0  7 93  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   BIC      14 12 74  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   RIFT.hc   1 69 26  4  0  0  0  0  0  0  0  0  0  0  0  0  0
##   shc       0  0 11 10  7  7 10 13 13  9  2  3  9  2  1  2  1
## [1] "(dimension, delta) = (2,3)"
##           ESTIMATE
## METHOD       1   2   3   4   5   6   7   8   9  11
##   Mean       0   8  92   0   0   0   0   0   0   0
##   Meanl2     0   8  92   0   0   0   0   0   0   0
##   Median     0   1  92   3   2   2   0   0   0   0
##   Medianl2   0   1  93   2   2   2   0   0   0   0
##   AIC        0   0  99   1   0   0   0   0   0   0
##   BIC        0   0 100   0   0   0   0   0   0   0
##   RIFT.hc    2   0  90   7   1   0   0   0   0   0
##   shc        0   0  65   5   9  13   1   5   1   1
## [1] "(dimension, delta) = (2,4)"
##           ESTIMATE
## METHOD       3   4   5   6   7   8
##   Mean     100   0   0   0   0   0
##   Meanl2   100   0   0   0   0   0
##   Median    91   1   5   2   0   1
##   Medianl2  93   1   4   2   0   0
##   AIC       98   2   0   0   0   0
##   BIC       99   1   0   0   0   0
##   RIFT.hc   94   5   1   0   0   0
##   shc       89   3   7   0   1   0
## [1] "(dimension, delta) = (2,5)"
##           ESTIMATE
## METHOD       3   4   5   6
##   Mean     100   0   0   0
##   Meanl2   100   0   0   0
##   Median    89   7   3   1
##   Medianl2  90   6   3   1
##   AIC       98   2   0   0
##   BIC       99   1   0   0
##   RIFT.hc   98   2   0   0
##   shc       91   2   7   0
## [1] "(dimension, delta) = (2,6)"
##           ESTIMATE
## METHOD       3   4   5
##   Mean     100   0   0
##   Meanl2   100   0   0
##   Median    91   6   3
##   Medianl2  91   6   3
##   AIC       98   2   0
##   BIC       99   1   0
##   RIFT.hc  100   0   0
##   shc       90   2   8
## [1] "(dimension, delta) = (2,7)"
##           ESTIMATE
## METHOD       3   4   5   6
##   Mean     100   0   0   0
##   Meanl2   100   0   0   0
##   Median    89   7   3   1
##   Medianl2  89   7   3   1
##   AIC       98   2   0   0
##   BIC       99   1   0   0
##   RIFT.hc   99   1   0   0
##   shc       91   2   7   0
## [1] "(dimension, delta) = (2,8)"
##           ESTIMATE
## METHOD       3   4   5
##   Mean      99   1   0
##   Meanl2    99   1   0
##   Median    90   7   3
##   Medianl2  90   7   3
##   AIC       98   2   0
##   BIC       99   1   0
##   RIFT.hc  100   0   0
##   shc       91   2   7
## [1] "(dimension, delta) = (2,9)"
##           ESTIMATE
## METHOD       3   4   5   6   8
##   Mean      99   1   0   0   0
##   Meanl2    99   1   0   0   0
##   Median    90   8   0   1   1
##   Medianl2  90   8   0   1   1
##   AIC       98   2   0   0   0
##   BIC       99   1   0   0   0
##   RIFT.hc  100   0   0   0   0
##   shc       91   2   7   0   0
## [1] "(dimension, delta) = (8,1)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 19 20 21
##   Mean     23 25 52  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Meanl2   23 25 52  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Median   40 58  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Medianl2 40 58  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   AIC       0  3 97  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   BIC       0 33 67  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   RIFT.hc   0 88 12  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   shc       0  0 23  5  4  4  8  4  4 10  8  4  5  5  5  5  1  2  2  1
## [1] "(dimension, delta) = (8,2)"
##           ESTIMATE
## METHOD       1   3   4   5   6
##   Mean       0 100   0   0   0
##   Meanl2     0 100   0   0   0
##   Median     0 100   0   0   0
##   Medianl2   0 100   0   0   0
##   AIC        0 100   0   0   0
##   BIC        0 100   0   0   0
##   RIFT.hc   27  70   3   0   0
##   shc        0  88   8   2   2
## [1] "(dimension, delta) = (8,3)"
##           ESTIMATE
## METHOD       1   3   4   5   6   7
##   Mean       0 100   0   0   0   0
##   Meanl2     0 100   0   0   0   0
##   Median     0 100   0   0   0   0
##   Medianl2   0 100   0   0   0   0
##   AIC        0  99   1   0   0   0
##   BIC        0 100   0   0   0   0
##   RIFT.hc    1  98   1   0   0   0
##   shc        0  86   7   4   1   2
## [1] "(dimension, delta) = (8,4)"
##           ESTIMATE
## METHOD       3   4   5   6   7  10
##   Mean     100   0   0   0   0   0
##   Meanl2   100   0   0   0   0   0
##   Median   100   0   0   0   0   0
##   Medianl2 100   0   0   0   0   0
##   AIC       99   1   0   0   0   0
##   BIC      100   0   0   0   0   0
##   RIFT.hc   99   1   0   0   0   0
##   shc       87   7   2   1   2   1
## [1] "(dimension, delta) = (8,5)"
##           ESTIMATE
## METHOD       3   4   5   6   7  10
##   Mean     100   0   0   0   0   0
##   Meanl2   100   0   0   0   0   0
##   Median   100   0   0   0   0   0
##   Medianl2 100   0   0   0   0   0
##   AIC       99   1   0   0   0   0
##   BIC      100   0   0   0   0   0
##   RIFT.hc  100   0   0   0   0   0
##   shc       86   7   3   1   2   1
## [1] "(dimension, delta) = (8,6)"
##           ESTIMATE
## METHOD       3   4   5   6   7  10
##   Mean     100   0   0   0   0   0
##   Meanl2   100   0   0   0   0   0
##   Median   100   0   0   0   0   0
##   Medianl2 100   0   0   0   0   0
##   AIC       99   1   0   0   0   0
##   BIC      100   0   0   0   0   0
##   RIFT.hc  100   0   0   0   0   0
##   shc       86   7   3   1   2   1
## [1] "(dimension, delta) = (8,7)"
##           ESTIMATE
## METHOD       3   4   5   6   7  10
##   Mean     100   0   0   0   0   0
##   Meanl2   100   0   0   0   0   0
##   Median   100   0   0   0   0   0
##   Medianl2 100   0   0   0   0   0
##   AIC       99   1   0   0   0   0
##   BIC      100   0   0   0   0   0
##   RIFT.hc  100   0   0   0   0   0
##   shc       86   7   3   1   2   1
## [1] "(dimension, delta) = (8,8)"
##           ESTIMATE
## METHOD       3   4   5   6   7  10
##   Mean     100   0   0   0   0   0
##   Meanl2   100   0   0   0   0   0
##   Median   100   0   0   0   0   0
##   Medianl2 100   0   0   0   0   0
##   AIC       99   1   0   0   0   0
##   BIC      100   0   0   0   0   0
##   RIFT.hc  100   0   0   0   0   0
##   shc       86   7   3   1   2   1
## [1] "(dimension, delta) = (8,9)"
##           ESTIMATE
## METHOD       3   4   5   6   7  10
##   Mean     100   0   0   0   0   0
##   Meanl2   100   0   0   0   0   0
##   Median   100   0   0   0   0   0
##   Medianl2 100   0   0   0   0   0
##   AIC       99   1   0   0   0   0
##   BIC      100   0   0   0   0   0
##   RIFT.hc  100   0   0   0   0   0
##   shc       86   7   3   1   2   1
```

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

print(distribution_name )
```

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

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

```
## [1] "3 true clusters:"
```

```r
print('Cluster weights:')
```

```
## [1] "Cluster weights:"
```

```r
print(w)
```

```
## [1] 0.3333333 0.3333333 0.3333333
```

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] arm_1.12-2         lme4_1.1-27.1      Matrix_1.3-4       knitr_1.37
##  [5] sigclust_1.1.0     mixtools_1.2.0     gridExtra_2.3      ggplot2_3.3.5
##  [9] MASS_7.3-54        pracma_2.3.6       mclust_5.4.9       sigclust2_1.2.4
## [13] rstudioapi_0.13    mvnfast_0.2.7      doParallel_1.0.16  iterators_1.0.13
## [17] foreach_1.5.1      Rfast_2.0.6        RcppZiggurat_0.1.6 Rcpp_1.0.8
## [21] pacman_0.5.1
##
## loaded via a namespace (and not attached):
##  [1] minqa_1.2.4            colorspace_2.0-2       ellipsis_0.3.2
##  [4] dynamicTreeCut_1.63-1  htmlTable_2.4.0        XVector_0.34.0
##  [7] base64enc_0.1-3        ggdendro_0.1.23        bit64_4.0.5
## [10] AnnotationDbi_1.56.2   fansi_0.5.0            codetools_0.2-18
## [13] splines_4.1.2          cachem_1.0.6           impute_1.68.0
## [16] Formula_1.2-4          nloptr_1.2.2.3         broom_0.7.12
## [19] WGCNA_1.70-3           cluster_2.1.2          kernlab_0.9-29
## [22] GO.db_3.14.0           png_0.1-7              compiler_4.1.2
## [25] httr_1.4.2             backports_1.4.1        fastmap_1.1.0
## [28] htmltools_0.5.2        tools_4.1.2            coda_0.19-4
## [31] gtable_0.3.0           glue_1.6.1             GenomeInfoDbData_1.2.7
## [34] dplyr_1.0.7            ggthemes_4.2.4         Biobase_2.54.0
## [37] vctrs_0.4.1            Biostrings_2.62.0      preprocessCore_1.56.0
## [40] nlme_3.1-153           xfun_0.30              fastcluster_1.2.3
## [43] stringr_1.4.0          lifecycle_1.0.1        zlibbioc_1.40.0
## [46] scales_1.1.1           RColorBrewer_1.1-2     yaml_2.3.4
## [49] memoise_2.0.1          rpart_4.1-15           segmented_1.3-4
## [52] latticeExtra_0.6-29    stringi_1.7.6          RSQLite_2.2.10
## [55] highr_0.9              S4Vectors_0.32.3       blme_1.0-5
## [58] checkmate_2.0.0        BiocGenerics_0.40.0    boot_1.3-28
## [61] GenomeInfoDb_1.30.1    rlang_1.0.2            pkgconfig_2.0.3
## [64] matrixStats_0.61.0     bitops_1.0-7           evaluate_0.15
## [67] lattice_0.20-45        purrr_0.3.4            htmlwidgets_1.5.4
## [70] bit_4.0.4              tidyselect_1.1.1       magrittr_2.0.2
## [73] R6_2.5.1               IRanges_2.28.0         generics_0.1.1
## [76] Hmisc_4.6-0            DBI_1.1.2              pillar_1.6.4
## [79] foreign_0.8-81         withr_2.4.3            survival_3.2-13
## [82] KEGGREST_1.34.0        abind_1.4-5            RCurl_1.98-1.6
## [85] nnet_7.3-16            tibble_3.1.6           crayon_1.4.2
## [88] utf8_1.2.2             rmarkdown_2.13         jpeg_0.1-9
## [91] data.table_1.14.2      blob_1.2.2             forcats_0.5.1
## [94] digest_0.6.29          tidyr_1.1.4            stats4_4.1.2
## [97] munsell_0.5.0
```

```
Sys.time()
```

```
## [1] "2022-06-02 17:47:15 BST"
```