FMM

2023-05-29

The source code can be found in Github under the ‘debugging’ branch.

User-Defined functions

```
### Function: Summary Quantities "Mean (SD)".
bal_quan <- function(num_vec, rounding = 4){
  mean_val <- round(mean(num_vec), 4)
  sd_val <- round(sd(num_vec), 4)
  paste0(mean_val, " (", sd_val, ")")
}

### Function: Summary from the result
summary_para <- function(result_list){
  ### Collect the data
  n_cluster_vec <- rep(NA, n_para)
  time_vec <- rep(NA, n_para)
  clus_quality <- matrix(NA, ncol = 3, nrow = n_para)

  for(i in 1:n_para){
    n_cluster_vec[i] <- result_list[[i]]$n_cluster
    time_vec[i] <- result_list[[i]]$time
    clus_quality[i, ] <- result_model[[i]]$clus_measure[c(1, 5, 22), 2]
  }

  data.frame(n_cluster = bal_quan(n_cluster_vec), time = bal_quan(time_vec)) %>%
    data.frame(t(apply(clus_quality, 2, bal_quan))) %>%
    kbl(col.names = c("# cluster", "time", "Adjusted Rand", "Jaccard", "VI"))

}

### Function: Calculate mean and variance
mean_var <- function(num_vec){
  c(mean(num_vec), var(num_vec))
}
```

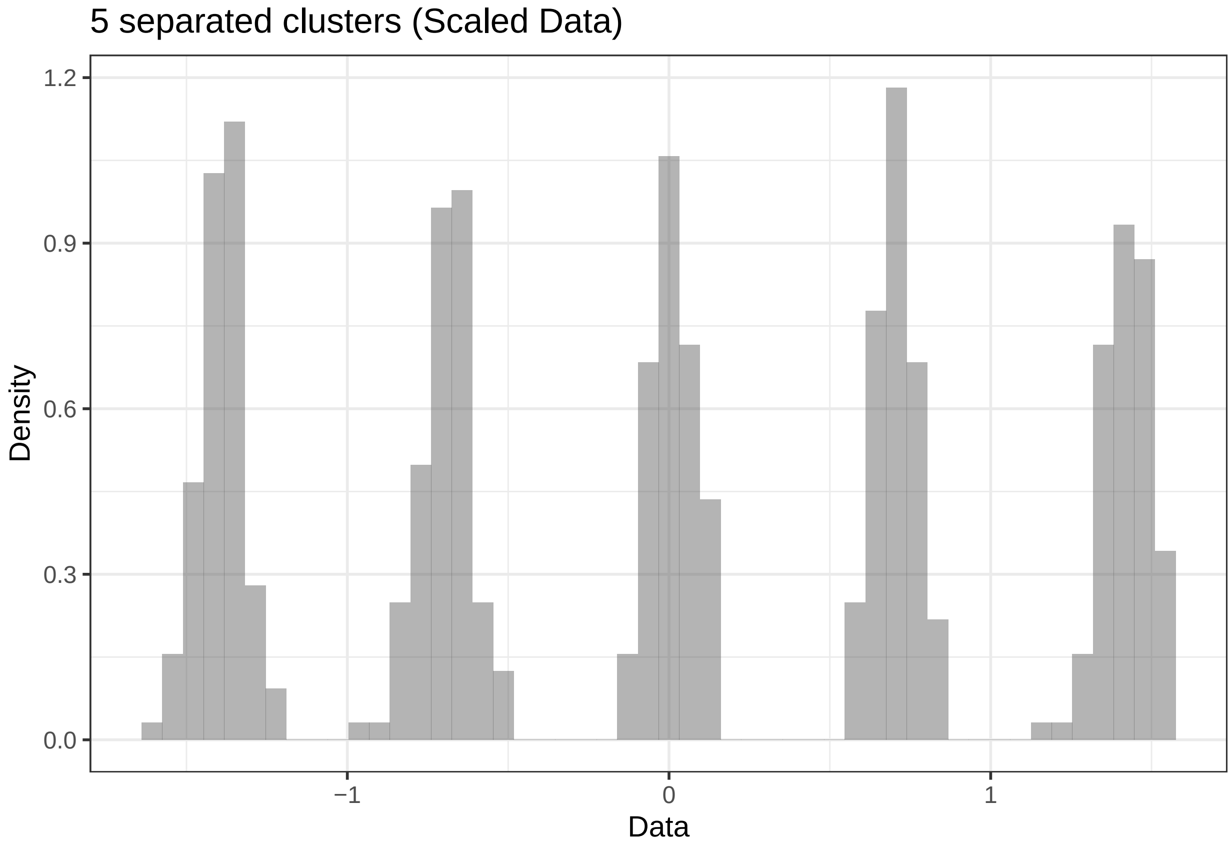
Overall Settings

I will run the model for 5,000 iterations for all cases while using the first 2,500 iterations as a burn-in. Also, I will run the model for 10 data sets parallel for each case.

```
iter <- 5000
burn_in <- 2500
overall_seed <- 31807
n_para <- 10
```

Part II: 5 Separated Clusters

Below is the plot for the standardized data for five separated clusters.



I will change the value for ξ ($\xi = 1, 0.1, 0.01, 0.001$) while keeping the other variables to be fixed. ($\mu = 0, a_\sigma = b_\sigma = \lambda = 1, K_{\max} = 10$)

# cluster	time	Adjusted Rand	Jaccard	VI
2.7 (0.6749)	25.2321 (3.7824)	0.535 (0.1636)	0.5055 (0.1277)	0.9763 (0.3579)

# cluster	time	Adjusted Rand	Jaccard	VI
3 (0.6667)	21.618 (3.0444)	0.6163 (0.1493)	0.5686 (0.1235)	0.8083 (0.3307)

# cluster	time	Adjusted Rand	Jaccard	VI
2.9 (0.3162)	20.3025 (2.8444)	0.6013 (0.0804)	0.5474 (0.0574)	0.8383 (0.1781)

# cluster	time	Adjusted Rand	Jaccard	VI
2.9 (0.3162)	19.4981 (2.8166)	0.6013 (0.0804)	0.5474 (0.0574)	0.8383 (0.1781)