# **Algorithm Illustration Hierachical Clustering Accuracy**

Step 1: We prepare the data:

* Cluster data according to different standard
* Divide the processed data into 2 group (set label)

|  |
| --- |
| res.agnes <- agnes(x = USArrests, # data matrix  stand = TRUE, # Standardize the data  metric = "euclidean", # metric for distance matrix  method = "average" # Linkage method  )  # Cut tree into 4 groups  grp <- cutree(res.agnes, k = 4)  head(grp, n = 4) |

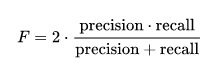
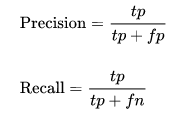


Step 2: Change the value of label

|  |
| --- |
| for (i in 1:50)  {  if (grp[i]!=1) grp[i]<-0  } |



Step 3: Create a function precision, recall and f-measure. Here are some formulas regarding to calculating:



I suppose those 1's in the predicted values (grp) are the retrieved. The total number of retrieved is **sum(predict)**

Precision which is the fraction of retrieved instances that are relevant, is

**precision <- sum(predict & actual\_labels) / sum(predict)**

Recall which is the fraction of relevant instances that are retrieved, is

**recall <- sum(predict & actual\_labels) / sum(actual\_labels)**

F-measure is 2 \* precision \* recall / (precision + recall) is

**fmeasure <- 2 \* precision \* recall / (precision + recall)**

|  |
| --- |
| measurePrecisionRecall <- function(predict, actual\_labels){  precision <- sum(predict & actual\_labels) / sum(predict)  recall <- sum(predict & actual\_labels) / sum(actual\_labels)  fmeasure <- 2 \* precision \* recall / (precision + recall)    cat('precision: ')  cat(precision \* 100)  cat('%')  cat('\n')    cat('recall: ')  cat(recall \* 100)  cat('%')  cat('\n')    cat('f-measure: ')  cat(fmeasure \* 100)  cat('%')  cat('\n')  } |

Step 3: Create a random list which is real label

|  |
| --- |
| true <- sample(c(0, 1), 50, replace=T) |

Step 4: Use the result in step 1 as predict and real label, then pass two parameters into the function.

|  |
| --- |
| measurePrecisionRecall(grp,true) |

