Q47-WilliamKennedy-300015367

William Kennedy

2023-12-06

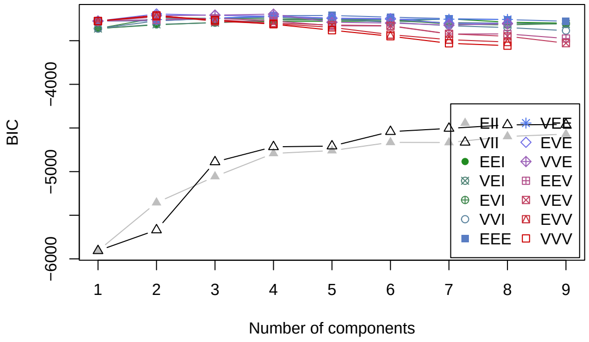
6.Wine datasets using expectation-maximization clustering, for various algorithm parameters. Validate your results.

```
wine = read.csv("wine.csv")
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 4.3.2
```

```
## Package 'mclust' version 6.0.1
## Type 'citation("mclust")' for citing this R package in publications.
```

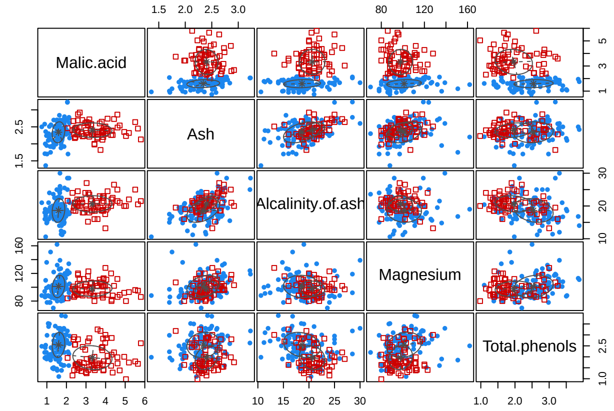
```
set.seed(0)
BIC <- mclustBIC(wine[,c(3:7)])
plot(BIC)
```



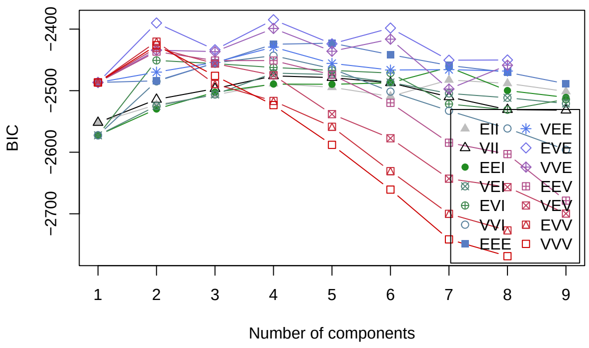
```
summary(BIC)
```

```
## Best BIC values:
##               EVE,2       VVE,4       VVE,3
## BIC      -3193.669 -3195.63921 -3206.51420
## BIC diff     0.000    -1.97045   -12.84543
```

```
mod1 <- Mclust(wine[,c(3:7)], x = BIC)
plot(mod1, what = "classification")
```



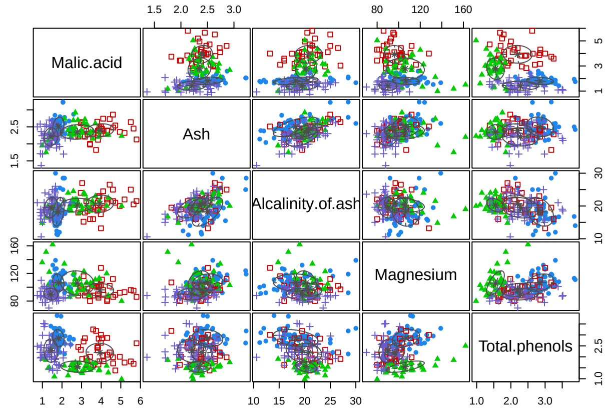
```
BIC.s <- mclustBIC(scale(wine[,c(3:7)]))
plot(BIC.s)
```



```
summary(BIC.s)
```

```
## Best BIC values:
##                 EVE,4        EVE,2        EVE,6
## BIC      -2384.787 -2390.062809 -2398.29990
## BIC diff     0.000    -5.276303   -13.51339
```

```
mod2 <- Mclust(wine[,c(3:7)], x = BIC.s)
plot(mod2, what = "classification")
```



The scaled output has much more overlap in the clustering than the unscaled version

7. Wine datasets using affinity propagation clustering, for various algorithm parameters. Validate your results.

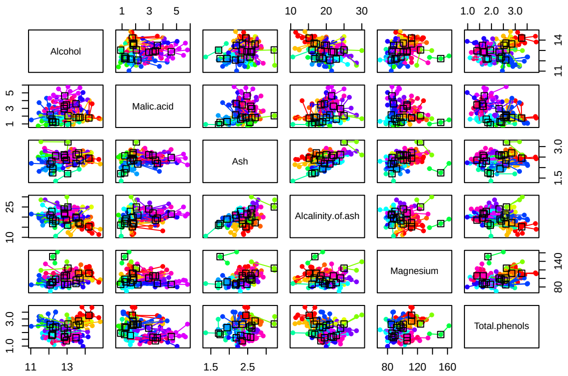
```
wine = read.csv("wine.csv")
wine.sc = data.frame(scale(wine))

#For simplicity sake, I took a different small subset features for each maximization cluster.
library(apcluster)
```

```
## Warning: package 'apcluster' was built under R version 4.3.2

##
## Attaching package: 'apcluster'

## The following object is masked from 'package:stats':
##
##     heatmap
```

```
ap.gap.1 = apcluster(negDistMat(r=2),
scale(wine[,c(2:7)]))
ap.gap.1
```

```
##
## APResult object
##
## Number of samples     =  178
## Number of iterations  =  165
## Input preference      =  -10.36567
## Sum of similarities   =  -335.4748
## Sum of preferences    =  -165.8507
## Net similarity        =  -501.3255
## Number of clusters    =  16
##
## Exemplars:
##    6 16 23 25 26 66 70 77 87 109 126 139 140 148 163 164
## Clusters:
##    Cluster 1, exemplar 6:
##       1 4 6 14 15 19 21 40 41 43 47 53 57 59
##    Cluster 2, exemplar 16:
##       5 8 16 17 18 20 29 34 35 37 54 56 58
##    Cluster 3, exemplar 23:
##       2 7 9 10 11 12 13 23 27 28 30 32 33 39 44 45 46 48 49 50 51 55 69
##    Cluster 4, exemplar 25:
##       3 25 31 36 38 52 72 159 160
##    Cluster 5, exemplar 26:
##       26 74 122
##    Cluster 6, exemplar 66:
##       24 61 65 66 85 86 110 114
##    Cluster 7, exemplar 70:
##       70 79 96
##    Cluster 8, exemplar 77:
##       60 63 67 77 102
##    Cluster 9, exemplar 87:
##       71 73 83 87 88 89 90 91 92 93 106 107 108 118 129 155
##    Cluster 10, exemplar 109:
##       62 68 76 81 98 101 104 105 109 117 135
##    Cluster 11, exemplar 126:
##       64 75 82 94 95 99 100 111 112 115 116 120 121 125 126 127
##    Cluster 12, exemplar 139:
##       42 119 139 142 147 149 157 165 166 168 172 173
##    Cluster 13, exemplar 140:
##       80 103 123 128 140 153 158 178
##    Cluster 14, exemplar 148:
##       84 124 130 137 138 144 148 156 161 174
##    Cluster 15, exemplar 163:
##       22 97 113 133 134 141 143 151 152 162 163 167 169 170 175
##    Cluster 16, exemplar 164:
##       78 131 132 136 145 146 150 154 164 171 176 177
```

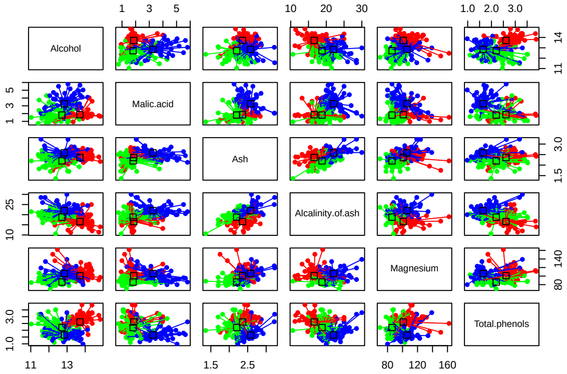
```
plot(ap.gap.1, wine[,c(2:7)])
```



```
ap.gap.2 = apcluster(negDistMat(r=2),
scale(wine[,c(2:7)]), q=0)
ap.gap.2
```

```
##
## APResult object
##
## Number of samples     =  178
## Number of iterations  =  148
## Input preference      =  -85.56096
## Sum of similarities   =  -714.9668
## Sum of preferences    =  -256.6829
## Net similarity        =  -971.6497
## Number of clusters    =  3
##
## Exemplars:
##    23 82 163
## Clusters:
##    Cluster 1, exemplar 23:
##       1 2 3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 27 29 30 31
##       32 33 34 35 36 37 38 39 40 41 43 44 45 46 47 48 49 50 51 52 53 54 55 56
##       57 58 59 69 70 72 79 96 159 160 173
##    Cluster 2, exemplar 82:
##       28 42 60 61 62 63 64 65 66 67 68 71 73 75 76 77 81 82 83 85 86 87 89 90
##       91 92 93 94 95 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112
##       114 115 116 117 118 119 120 121 125 126 127 129 135 136 139 142 155 172
##    Cluster 3, exemplar 163:
##       5 22 26 74 78 80 84 88 97 113 122 123 124 128 130 131 132 133 134 137 138
##       140 141 143 144 145 146 147 148 149 150 151 152 153 154 156 157 158 161
##       162 163 164 165 166 167 168 169 170 171 174 175 176 177 178
```

```
plot(ap.gap.2, wine[,c(2:7)])
```


Alcohol
Malic.acid
Ash
Alcalinity.of.ash
Magnesium
Total.phenols