Fuzzy C-means Clustering of Iris Dataset

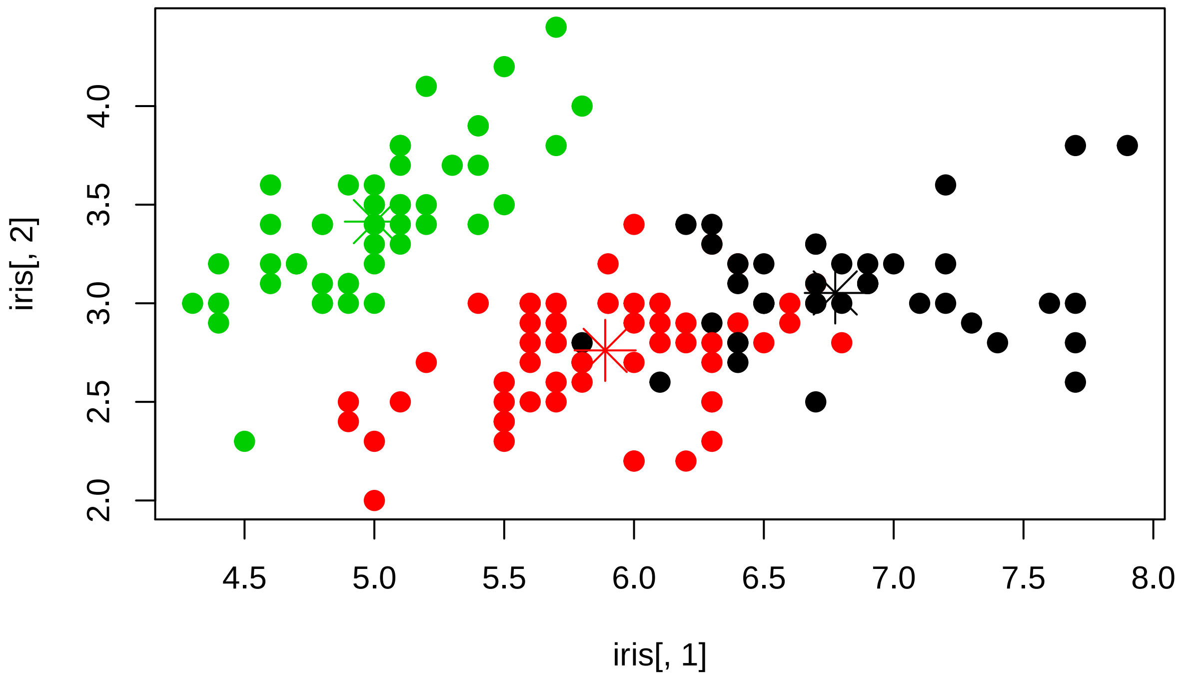
Jon Kinsey

Mon Dec 1 13:02:06 2014

```
# Unlike K-Means where each data point belongs to only one cluster,
# in fuzzy cmeans, each data point has a fraction of membership to each cluster.
# The goal is to figure out the membership fraction that minimize the
# expected distance to each centroid. This method utilizes an overlapping
# clustering algorithm that was developed by Dunn in 1973 and improved
# by Bezdek in 1981. It is frequently used in pattern recognition.
#
# Advantages:
# 1. Gives best result for overlapped data set and comparatively better than
#    k-means algorithm.
# 2. Unlike k-means where data point must exclusively belong to one cluster
#    center here data point is assigned membership to each cluster center as
#    a result of which data point may belong to more than one cluster center.
# Disadvantages:
# 1. Apriori specification of the number of clusters.
# 2. With lower value of beta we get the better result but at the expense of more
#    number of iteration.
# 3. Euclidean distance measures can unequally weight underlying factors.
#
# J. C. Dunn (1973): "A Fuzzy Relative of the ISODATA Process and Its Use in
#   Detecting Compact Well-Separated Clusters", Journal of Cybernetics 3: 32-57
# J. C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algoritms",
#   Plenum Press, New York

library(e1071)

result <- cmeans(iris[,-5], 3, 100, m=2, method="cmeans")
plot(iris[,1], iris[,2], col=result$cluster,pch=20,cex=2)
points(result$centers[,c(1,2)], col=1:3, pch=8, cex=3)
```




```r
# the visual output is very similar to K-Means
result$membership[1:3,]
```

```
##             1        2      3
## [1,] 0.001072 0.002304 0.9966
## [2,] 0.007498 0.016651 0.9759
## [3,] 0.006415 0.013760 0.9798
```

```r
result$centers
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1        6.775       3.052        5.647      2.0535
## 2        5.889       2.761        4.364      1.3973
## 3        5.004       3.414        1.483      0.2535
```