Exp 6 - Expectation–maximization (EM) algorithm

September 06, 2022

The **expectation–maximization (EM) algorithm** is an iterative method in statistics to find (local) maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables.

The **k-means clustering** on the other hand, is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

1 Experimental Description

1.1 Objective

To apply EM algorithm to cluster a set of data and use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering.

1.2 Algorithm

- EM Algorithm
    1. The EM algorithm alternates between two steps (E-step and M-step).
    2. In the E-step the algorithm tries to find a lower bound function on the original likelihood using the current estimate of the statistical parameters.
    3. In the M-step the algorithm finds new estimates of those statistical parameters by maximizing the lower bound function (i.e. determine the MLE of the statistical parameters).
    4. The process is continued where at each step, the lower bound function is maximized and the algorithm always produces estimates with higher likelihood than the previous iteration and ultimately converges to a maxima.
- K-Means Algorithm
    1. The K-means algorithm attempts to detect clusters within the dataset.
    2. The optimization criteria is that the sum of the inter-cluster variances is minimized.
    3. Hence K-Means clustering algorithm produces a Minimum Variance Estimate (MVE) of the state of the identified clusters in the data.

1.3 Procedure

We compare the results of both the algorithms by the following procedure:

- In K-Means algorithm, we perform the clustering by:

1. Choose the number of clusters k.
2. Select k random points from the data as centroids.
3. Assign all the points to the closest cluster centroid.
4. Recompute the centroids of newly formed clusters.

- In EM algorithm, the clustering is performed by:
  1. Given a set of incomplete data, consider a set of starting parameters.
  2. Expectation step (E – step): Using the observed available data of the dataset, estimate (guess) the values of the missing data.
  3. Maximization step (M – step): Complete data generated after the expectation (E) step is used in order to update the parameters.
  4. Repeat the steps until convergence.

1.4 System Requirements

Windows/Linux OS/Mac OS with R. We require the **DT**, **EMCluster**, **FactoMineR**, **reshape2** and **summarytools** packages to be installed.

1.5 Dataset Summary

For the project, we use the well-known Iris dataset. Containing 150 plants from 3 types of iris plants, 4 attributes were measured such as petal length and width. The goal is to properly classify every plant into its true type.

2 Code and Output

```
# Loading required libraries

library(ggplot2)
library(DT)
library(EMCluster)
library(FactoMineR)
library(reshape2)
library(summarytools)
library(dplyr)
```

```
version$version.string
```

```
## [1] "R version 4.2.1 (2022-06-23 ucrt)"
```

```
# Scale the data

set.seed(123)
iris_sc=scale(iris[,-5],center = T,scale = T)
```

```
# Perform k-means clustering on scaled data

km_raw=kmeans(iris_sc, centers=3)
```

```
# Perform EM clustering on scaled data

emobj <- exhaust.EM(iris_sc,nclass = 3)
```

```
emobj <- shortemcluster(iris_sc, emobj,maxiter = 1000)
em_raw <- emcluster(iris_sc, emobj, assign.class = TRUE)
```

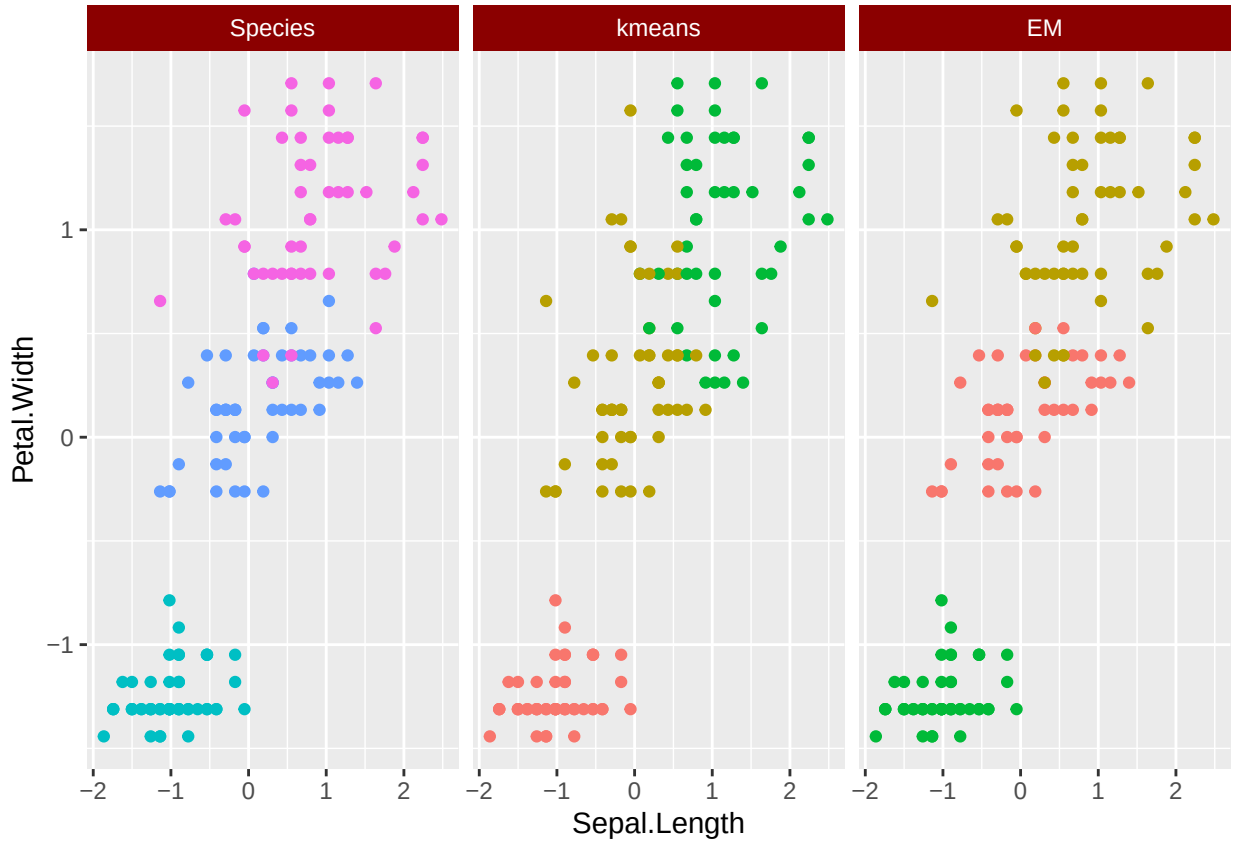
```
# Construct a data frame of the clustered data for both the algorithms

dat_clust_raw=data.frame(iris_sc,"Species"=as.factor(iris$Species),
kmeans=as.factor(km_raw$cluster), EM=as.factor(em_raw$class))

dat_raw=dat_clust_raw %>% melt(id.var=colnames(iris[,-5]))
```

```
# Plotting the clusters

dat_raw %>% ggplot(aes(x=Sepal.Length,y=Petal.Width, color=value))+geom_point()+
facet_wrap(~variable)+theme_grey()+
theme(strip.background =element_rect(fill="darkred"))+
theme(strip.text = element_text(colour = 'white'))+guides(color=FALSE)
```



From the output, we can interpret that:

- The results from EM algorithm were closest to the actual species. The diagonal variant of the EM allowed each cluster to have different sizes whereas the generic covariance matrix additionally gave each cluster a different orientation in the feature space.
- Unlike K-means, in EM, the clusters are not limited to spherical shapes. In EM we can constrain the algorithm to provide different covariance matrices (spherical, diagonal and generic). These different covariance matrices in return allow us to control the shape of our clusters and hence we can detect sub-populations in our data with different characteristics.

3 Conclusion

From the experiment conducted, it can be concluded that:

- EM Algorithm is a solid alternative to traditional k-means clustering on semi-supervised learning.
- It produces stable solutions by finding multivariate Gaussian distributions for each cluster.
- The EM algorithm offers a powerful alternative to the popular k-means with greater control over the characteristics of the cluster.
- However the EM algorithm, like the k-means, also yields a sub-optimal solution.