

Carbon

Jiayu Wang

2.Clustering

1. Load data

```
#load module data
import data
#create a dataset object and read data from file sample.txt
sample = data.DataSet()
#To read nominal data, you have to add argument 'nominal',
default is 'numeric'
#Read data from 'sample.txt'
sample.read('sample.txt', 'numeric')
#create train dataset and test dataset using 1:10 hold out
train,test = data.holdOut(sample,0.1)
```

2. kMeans

A kMeans algorithm which also support bi-kMeans.

The algorithm only works for: numerical data

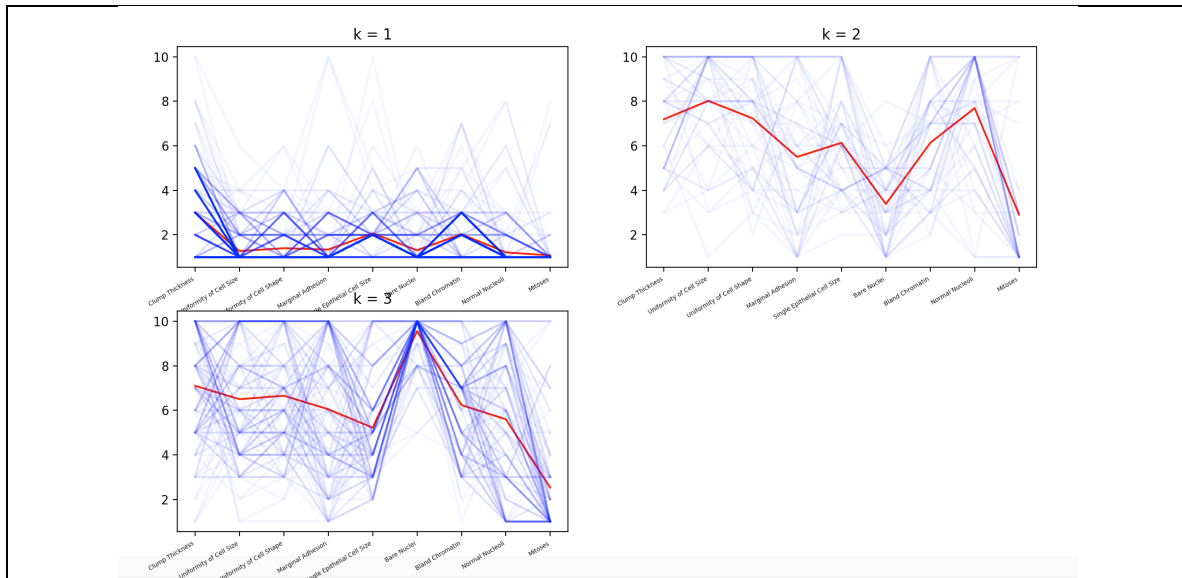
Parameters:

k: int from 1 to inf

dist: 'euclidean'

method: 'KMeans', 'biKMeans'

```
#load kMeans module
from clustering import kMeans
#create an instance with k=3, euclidean distance, KMeans
instance = kMeans.build(k=3, dist='euclidean', method='KMeans')
#train the classifier with train data and k=4
centroids,clusters = instance.cluster(train.x)
#centroids contains the centers of each cluster
centroids
matrix([[ 7.25555556,  4.87777778,  5.12222222,  4.9          ,
  4.02222222,  9.04444444,  5.27777778,  3.75555556,  1.7          ],
 [ 2.96933962,  1.26650943,  1.4009434 ,  1.31603774,
  2.0754717 ,  1.29009434,  2.02830189,  1.22169811,  1.07075472],
 [ 7.04310345,  8.40517241,  8.05172414,  6.72413793,  6.5
 ,  7.17241379,  6.89655172,  7.88793103,  3.3362069 ]])
#clusters contains the cluster labels and distances from the
centers
clusters
matrix([[ 1.          ,  5.85476927],
 [ 1.          ,  6.9962787 ],
 [ 2.          , 125.31599287],
 ...,
 [ 2.          ,  49.50564804],
 [ 2.          ,  62.02288942],
 [ 2.          ,  53.57461356]])
#to view the clusters
kMeans.view(train, centroids, clusters)
```



3. hierarchical

A hierarchical algorithm supports ward method.

The algorithm only works for: numerical data

Parameters:

k: +int

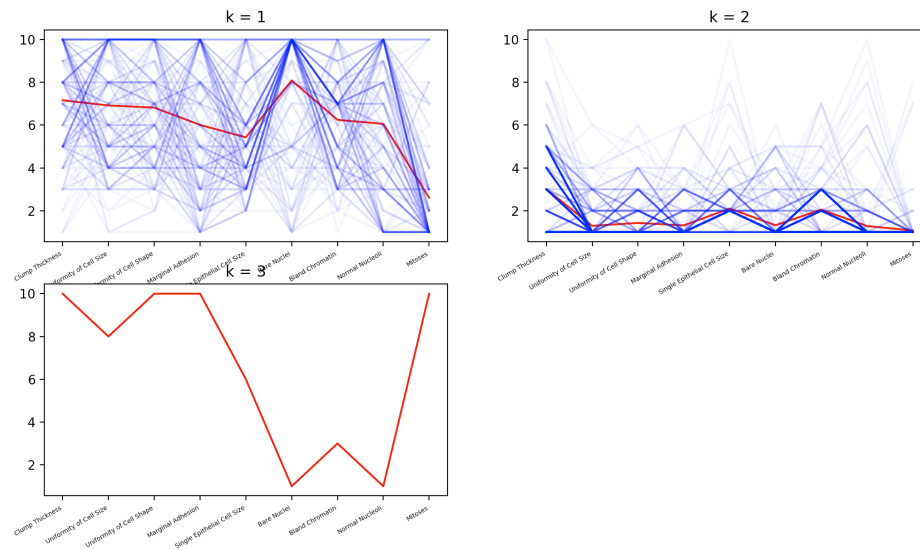
dist: 'euclidean'

method: 'ward'

```
#load hierarchical module
from clustering import hierarchical
#create an instance with k=3, euclidean distance, hierarchical
instance = hierarchical.build(k=3, dist='euclidean',
method='ward')
#train the classifier with train data and k=4
centroids,clusters = instance.cluster(train.x)
#centroids contains the centers of each cluster
centroids
matrix([[ 7.16      ,  6.92      ,  6.8175     ,  6.00875    ,
 5.42875     ,  8.08875     ,  6.245      ,  6.0575     ,
 2.605      ],
 [ 3.00225816,  1.29173951,  1.42693765,  1.31155303,
 2.10278263,  1.32000291,  2.05150058,  1.28256119,
 1.07473776],
 [ 10.        ,  8.         ,  10.         ,  10.         ,
 6.          ,  1.         ,  3.          ,  1.         ,
 10.         ]])
#clusters contains the cluster labels
clusters
array([ 1.,  1.,  0.,  1.,  0.,  1.,  1.,  1.,  1.,  1.,  1.,
 1.,  1.,
 ...,
 0.,  0.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  0.,  1.,
 1.,  1.,  1.,  0.,  0.,  0.]])
```

```
#to view the clusters
```

```
hierarchical.view(train, centroids, clusters)
```



```
#fast start
```

```
import data
```

```
sample = data.DataSet()
```

```
sample.read('sample.txt')
```

```
train,test = data.holdOut(sample,0.1)
```

```
from imp import reload
```

```
reload(hierarchical)
```

```
instance = hierarchical.build(k=1,dist='euclidean',method='ward')
```

```
old,clusters = instance.cluster(train.x)
```

```
new = mean(train.x,0)
```