Practical 6: Data Stream Mining

2022-09-12

In this practical, we use the *stream* R package for analysing stream data. Please install the stream package to complete the practical.

I. Creating a data stream

1. We firstly create a generator to generate stream data points that will belong to one of three clusters (k=3). Each data point will have 2 dimensions (d=2). The data points will follow Gaussian distribution with 5% noise. When a new data point is requested from this data generator, a cluster will be chosen randomly using the probability weights in p.

```
library("stream")
stream <- DSD_Gaussians(k = 3, d = 2, noise = .05, p = c(.5, .3, .1))
stream
```

```
## Gaussian Mixture (d = 2, k = 3)
## Class: DSD_Gaussians, DSD_R, DSD
```

2. Generate 5 data points using the generator.

```
p <- get_points(stream, n = 5)
p
```

```
##           X1        X2 .class
## 1 0.23095777 0.2106454      1
## 2 0.17118235 0.2769560      1
## 3 0.20167268 0.2344885      1
## 4 0.03396614 0.8390771      2
## 5 0.20011582 0.2491737      1
```

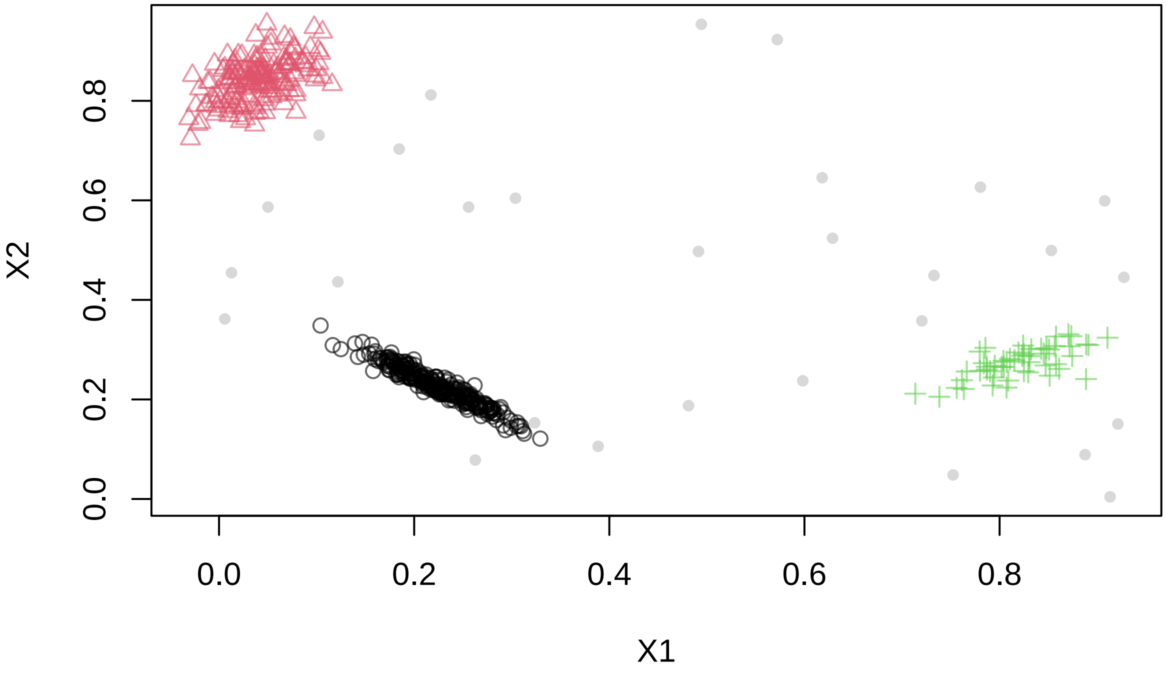
3. Use option class=TRUE to see which cluster a data point belongs to. Please note that noise data points (5%) do have the class labels (NA).

```
p <- get_points(stream, n = 10, class = TRUE)
p
```

```
##               X1        X2 .class
## 1   0.8145570610 0.2974015      3
## 2   0.0612473295 0.8316973      2
## 3  -0.0009832492 0.8362885      2
## 4   0.1905848241 0.2537432      1
## 5   0.1364594156 0.2962530      1
## 6   0.2691659649 0.1773113      1
## 7   0.1605845238 0.2879036      1
## 8   0.1772919051 0.2544033      1
## 9   0.2261587389 0.2146099      1
## 10  0.1656835415 0.2797510      1
```

4. Plot the 500 points from the data stream

```
plot(stream, n=500)
```



II. Reading and writing data streams

1. Write the created stream with 100 data points to a file called data.csv.

```
write_stream(stream, "data.csv", n = 100, sep = ",")
```

2. Read back the data.csv file to R.

```
stream_data = DSD_ReadStream("data.csv")
```

3. Note that the data has not been read to the stream_data until we use get_points

```
get_points(stream_data, n=5)
```

```
##           V1        V2
## 1 0.06657734 0.9038948
## 2 0.18231385 0.2757688
## 3 0.03083877 0.8503099
## 4 0.26080722 0.1935292
## 5 0.73737043 0.2391856
```

III. Reservoir Sampling

1. Create a stream with 3 clusters and 5% noise

```
stream <- DSD_Gaussians(k = 3, d = 2, noise = .05, p = c(.5, .3, .1))
```

2. Create a Reservoir sampling mechanism with 20 points will be sampled from the stream

```
sample <- DSAggregate_Sample(k = 20)
```

3. Update the data for sample using 500 data points from stream

```
update(sample, stream, 500)
```

```
## NULL
```

```
sample
```

```
## Reservoir samplingClass: DSAggregate_Sample, DSAggregate, DST
```

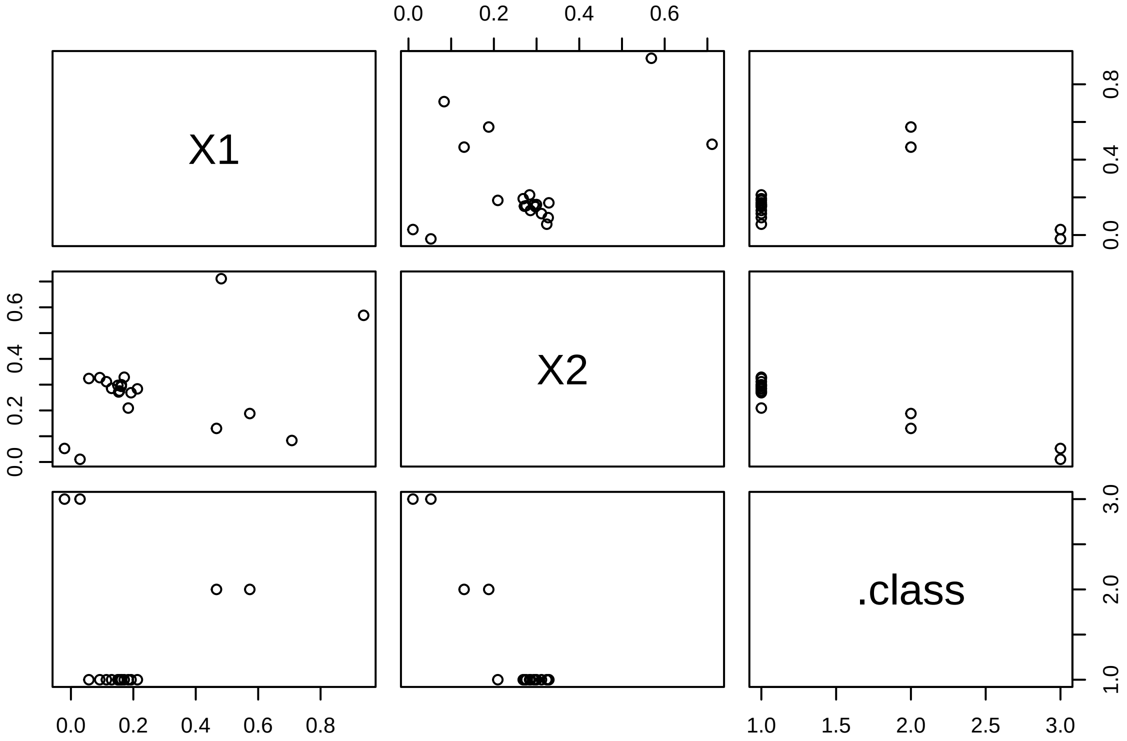
4. Get the data from sample

```
get_points(sample)
```

```
##             X1         X2 .class
## 206  0.70792925 0.08340397     NA
## 300  0.15297358 0.27156181      1
## 360  0.93798046 0.56886531     NA
## 143  0.21294737 0.28366773      1
## 279  0.15555464 0.27553274      1
## 334  0.18376398 0.20921992      1
## 49   0.48165434 0.71097681     NA
## 6    0.46648286 0.13027513      2
## 473  0.57318618 0.18796730      2
## 469  0.16166528 0.29979685      1
## 442  0.13031447 0.28566035      1
## 104  0.09244056 0.32723536      1
## 155  0.17083905 0.32878323      1
## 56   0.16138736 0.29341366      1
## 125  0.05751822 0.32383944      1
## 188  0.15052869 0.29728812      1
## 350 -0.02052869 0.05248499      3
## 67   0.02916187 0.01042358      3
## 471  0.19267060 0.26863388      1
## 22   0.11371753 0.31147971      1
```

5. Plot the data points in sample

```
plot(get_points(sample))
```



IV. Data Stream Clustering

1. We firstly prepare the clustering algorithm. We use DSC_DStream which implements the D-Stream algorithm (Tu and Chen 2009). D-Stream assigns points to cells in a grid. For the example we use a gridsize of 0.1.

```
dstream <- DSC_DStream(gridsize = .1, Cm = 1.2)
dstream
```

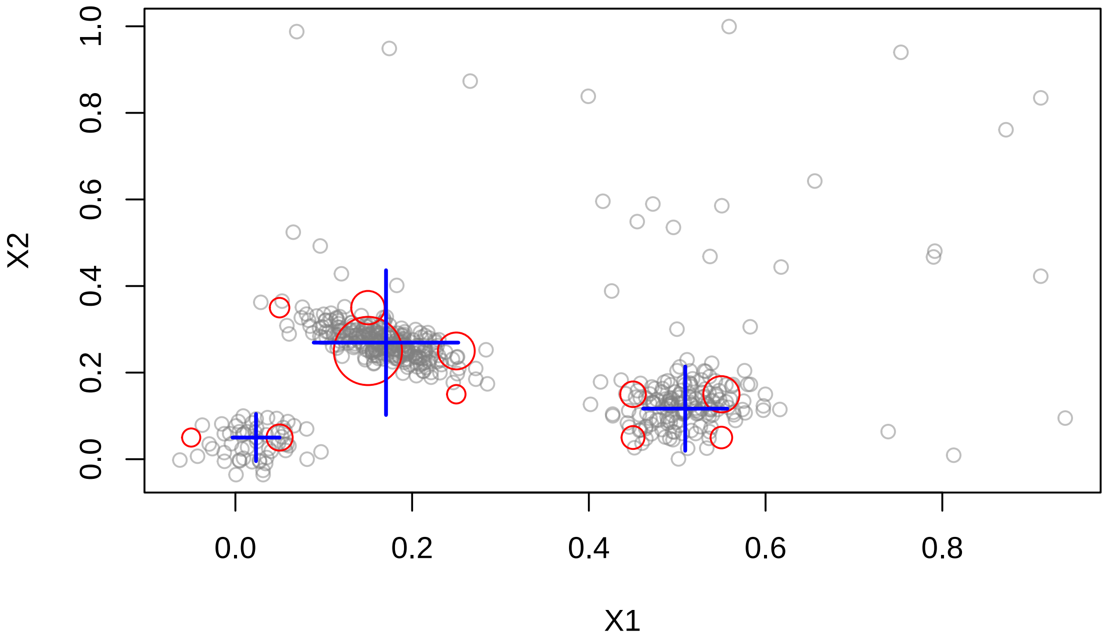
```
## D-Stream
## Class: DSC_DStream, DSC_Micro, DSC_R, DSC
## Number of micro-clusters: 0
## Number of macro-clusters: 0
```

2. The clusters are currently empty, but they are ready to get data points from the stream.

```
update(dstream, stream, n = 500)
dstream
```

```
## D-Stream
## Class: DSC_DStream, DSC_Micro, DSC_R, DSC
## Number of micro-clusters: 11
## Number of macro-clusters: 3
```

```
plot(dstream, stream)
```



3. There are a number of micro-clusters. We can get the centers of the micro-clusters using:

```
head(get_centers(dstream))
```

```
##      X1   X2
## 1 -0.05 0.05
## 2  0.05 0.05
## 3  0.05 0.35
## 4  0.15 0.25
## 5  0.15 0.35
## 6  0.25 0.15
```