Introduction to Machine Learning

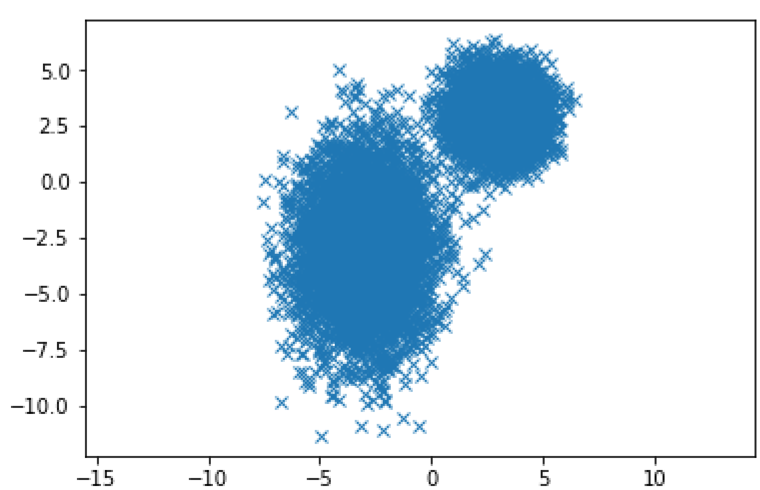
Unsupervised Learning Exercise

These exercises can be solved in any programming language of your choice. It is assumed you are proficient with programming. It will be helpful if the language has a library to plot graphics.

This exercise will demonstrate how a learning algorithm can distinguish between two distributions of points generated with different parameters.

1. Generate 2D points with a multivariate Gaussian distribution. Generate two sets with different centers and write them to one file shuffled. To avoid being blocked by this question if you fail to generate the points you may use this text file you can find in the same folder, with 2D points (the first 10000 numbers are the x values the last 10000 numbers are the y values, use the numpy loadtxt function) with the distribution seen in the image below, generated with the following parameters:

mean1 = [3, 3]
cov2 = [[1, 0], [0, 1]]
mean2 = [-3, -3]
cov2 = [[2, 0], [0, 5]]



(check if the reading was correct by making this graph)

2. Start by choosing two random points in the dataset `r1` and `r2` and apply the following adaptation rule:

```
For each point x in the data-set

      If r1 is closest to x than r2

            r1 = (1 - alfa) r1 + alfa * x

      If r2 is closest to x than r1

            r2 = (1 - alfa) r2 + alfa * x
```

Repeat for 10 times a passage through all the elements of the data-set with alfa = 10E-7 and save the consecutive values of `r1` and `r2` for the first passage. Plot the consecutive positions of `r1` and `r2`. Repeat the process 30 times without saving plots (verify if standard-deviation of the values of `r1` and `r2` is high or low). Control the seeds of the 30 tests to enable repetition.

Is there any relation between the (most common) final values of `r1` and `r2` and the parameters used to generate the points? How can you automatically assign r1 and r2 to the "right" group for better comparison?

Instead of changing the value for each example, accumulate the values of the difference (x - r) and change the value only when all examples have been observed.

```
            for all x
                  d += (x - r)

            r += (alfa/n_examples) * d
```

Plot the consecutive positions of r1 and r2 and compare with the above plot. What do you observe?

3. Plot with one colour the points closest to `r1` and with another the points closest to `r2`. What do you observe?

4. Implement a simplified version of agglomerative hierarchical clustering:

```
While there are more than two points
      Find the closest two points
            Replace both points by their average
```

Are the last two points good representatives of each cluster? If not, what simplifications of the hierarchical clustering procedure may have caused this?

5. Create a matrix of representatives (between 4x4 and 10x10, randomly initialized from dataset points) and repeat the process described in 2 with a variation. After updating representative in position `(i,j)` using equation in exercise 2, representatives in positions `(i+1,j), (i-1,j), (i,j+1)` and `(i,j-1)` also update their values according to:

```
r = (1 - alfa/2) r + (alfa/2) * x
```

Print snapshots of the representatives' positions at epochs 0, 100, 200, 300, … until the representatives stabilize (define a suitable criteria). Repeat the process 30 times and show one typical evolution of the representative matrix.

Notice that points that are neighbors keep close together, and classify parts of the data-set that are close to one another.

6. Implement DBScan algorithm as described in https://www.youtube.com/watch?v=_A9Tq6mGtLI and demonstrate it