Tugas 3 Pemebelajaran Mesin

Nama : Simiao Salvador da Gama

NIM : 1301163617

Kelas : IF-40-12

1. Kelebihan K-Means Clustering :

- Mudah dilakukan saat pengimpelementasian dan di jalankan.
- Waktu yang di butuhkan untuk melakukan pembelajaran relatif lebih cepat.
- Sangat fleksibel, adaptasi yang mudah untuk di lakukan
- Sangat umum penggunaannya.
- Menggunakan prinsip yang sederhana dapat di jelaskan dalam non-statistik.

Kekurangan K-Means Clustering :

- Sulit memprediksi Nilai-K.
- K-Means tidak bisa berfungsi dengan baik dalam global cluster.
- Partisi awal yang berbeda dapat menghasilkan cluster akhir yang berbeda.
- K-Means tidak bisa bekerja dengan baik dengan cluster (dalam data asli) dengan ukuran berbeda dalam densitas yang berbeda.

Contoh kasus : Lakukan clustering pada kumpulan data berikut

i	1	2
A	1	4
B	2	3
C	3	4
D	2	3
E	2	1

Dalam hal ini saya memilih k=2 dan memilih centroid secara acak yaitu A dan D. Lakukan perhitungan suatu data dengan centroid, yaitu dengan rumus sebagai berikut :

$$d(p,q) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + ...(q_n-p_n)^2}$$

Maka terdapat tabel sebagai berikut ini serta dengan clusternya(kelompoknya)

i	1	2	Cluster
A	0	1,414213562	1
B	1,414213562	0	2
C	2	2,828427125	1
D	1,414213562	0	2
E	3,16227766	2	2

Pilih kembali centroid untuk masing-masing cluster ,yaitu mean (rata-rata) nilai data dari setiap cluster yang sama . Lakukan penghitungan lagi dengan centroid yang baru.

Kesimpulan : Jadi seperti kita sudah melihat dalam permasalahan ini bahwa **kelebihan** K-Means clustering adalah mudah sekali bagi kita untuk melakukan pengimplementasian dan sangat sederhana; **kelemahannya** adalah sulit untuk memprediksi nilai ***k*** dan tidak akan bisa di implementasikan (dicluster) apabila ukuran data asli tidak sama dengan data densitas.

2. **Agglomerative hierarchical clustering** adalah sebuah metode clustering yang bertujuan untuk mengelompokkan objek objek sesuai dengan karakteristik yang dimilikinya, dimana dimulai dengan objek-objek individual sampai objek-objek tersebut bergabung menjadi suatu cluster yang tunggal.

Contoh Soal : Terdapat sebuah data dengan ukuran matriks 6x6 dimana kita diminta untuk mengelompokan sesuai dengan algoritma agglomerative hierarchial clustering :

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Ambil jarak terdekat (single linkage) dan digabungin. Terdapat D,F jadi kita gabungin.

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

Terdapat A,B dengan jarak terdekat. Ambil itu dan gabungin.

Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0

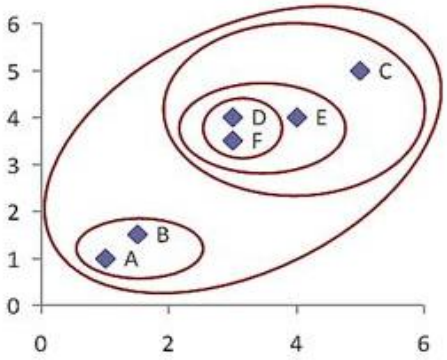
Terdapat jarak E dengan D,F yang terdekat jadi gabungkan.

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

Terdapat jarak E dengan D,F,E yang terdekat , jadi gabungkan

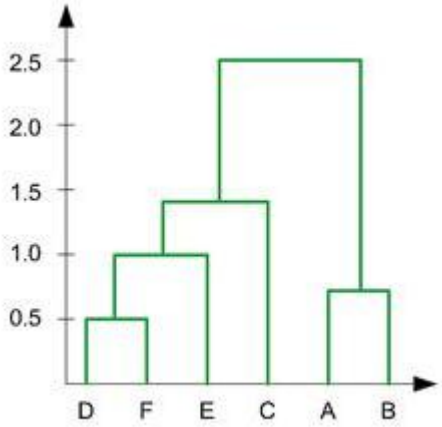
Dist	(A,B)	(D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	**2.50**	0.00

Gambar hierarki sesuai dengan pengelompokan



	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

Gambar dendogram sesuai dengan pengelompokan dan bobot yang dimiliki



3. **- Langkah 1** : Dalam permasalahan ini saya ambil label untuk dijadikan kelas , adalah : y = ***data train kolom1 / data train kolom2.*** Seteleh itu ambil ceil nya dimana terdapat empat label kelas (1,2,3,4)

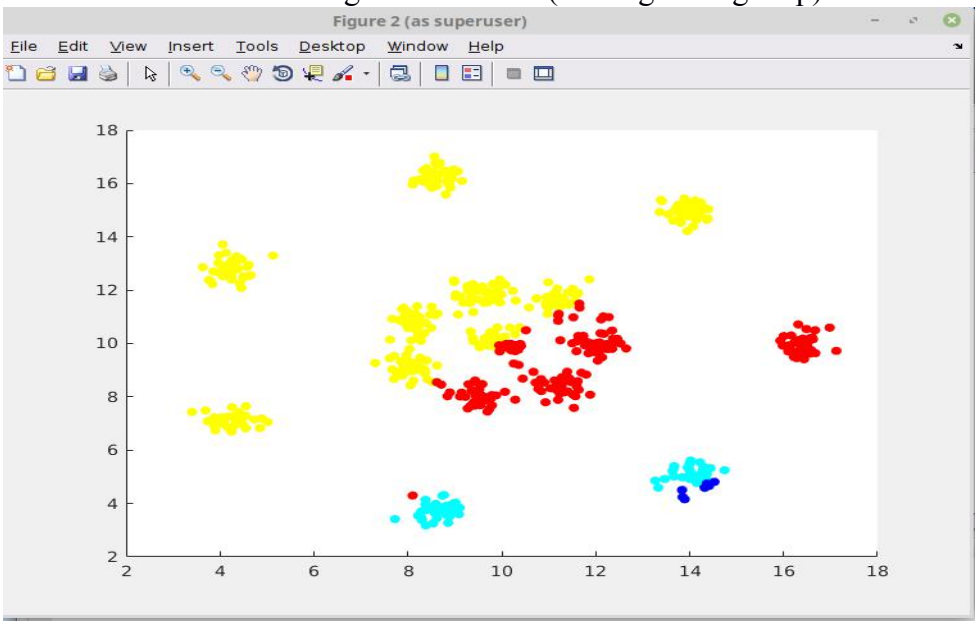
Untuk dimensi arkitektur SOM saya ambil 25x25.

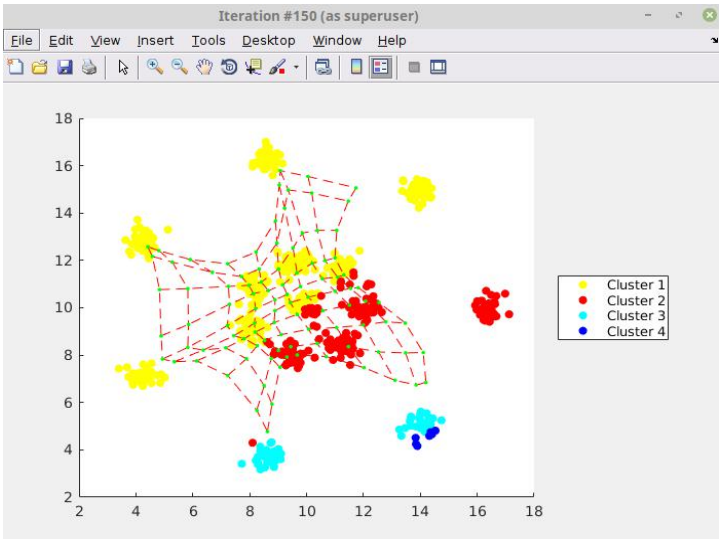
- Langkah 2 : Setting dan inisialisasi parameter atau variabel variabel yang dibutuhkan : - ukuran lebar untuk winning neuron = 15
- waktu konstan untuk topologi neighbourhood = iteration/log(5)
- waktu awal learning rate = 1
- Total iterasi = 150
- Waktu konstan untuk learning rate = Total iterasi

- Langkah 3 : Fungsi fungsi yang dibutuhkan dalam algoritma ini :
- ***findBestMatch*** : Fungsi ini akan mencari best matched vector(winning neuron) sesuai dengan input image.
- ***computeNeighbourhood*** : Fungsi ini akan menghitung jarak lateral(lateral distance) antara neurons i dan winning neurons.
- ***randInitializeWeights*** : Fungsi ini akan menginisialisai bobot(weight) vector pada tiap neuron secara acak antara 0 dan 1.
- ***updateWeight*** : Fungsi ini akan mengupdate semua neuron tergantung jarak antara winning neuron dan neuron lain
- ***plotData*** : fungsi untuk memplotting data.

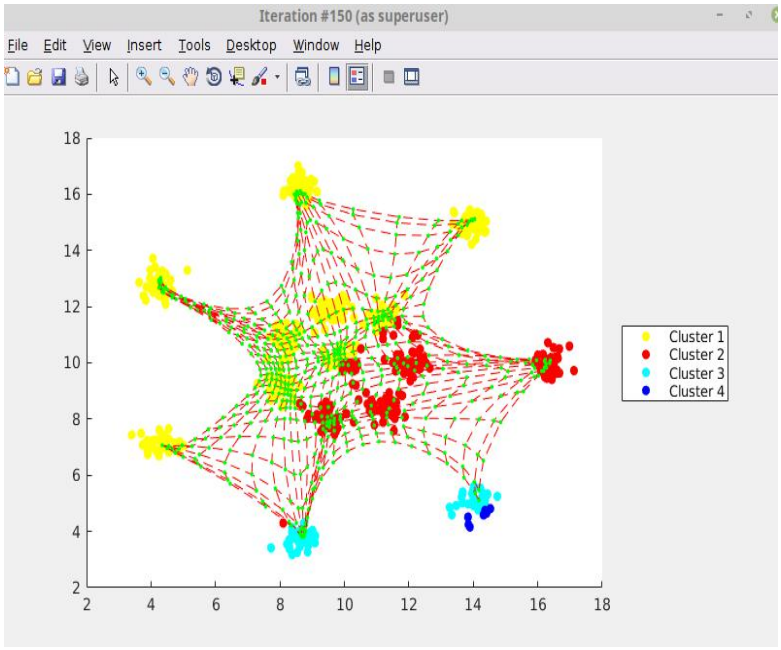
- Langkah 4 : - Inisialisasi bobot untuk tiap neuron secara acak (0 atau 1)
Dalam perulangan di main program :
- Mencari winning neuron yang paling cocok berdasarkan eucliden distance yang terkecil antara input dan masing masing neuron.
- Hitung lateral distance antara winning neuron dan setiap neurong.
- Update bobot neuron
- Plot data self organizing map denga kelas yang sudah ditentukan.

Hasil : - Hasil Plot sebelum mengeksekusi SOM(self organizing map)





Hasil 150 iterasi dengan dimensi 10 x 10



Hasil 150 iterasi dengan dimensi 25 x 25

Kesimpulan : Semakin besar dimensinya dan data trainnya maka pattern pengenalan neuron akan sangat bagus/baik. Untuk cluster yang paling optimum dalam hal ini ialah cluster 1 (dengan warna kuning) karena ukuran kumpulannya yang besar serta banyaknya kelompok yaitu 8 kelompok di antara 15 kelompok clustering.