Tugas 2, *Machine Learning*
Laporan Mengenai Clustering pada Machine Learning

Oleh:
Enrico Farizky Rustam (1301164263)
IF 40-04 / S1 Informatika / Universitas Telkom

1. Kelebihan dan Kekurangan k – Means Clustering

K – means Clustering adalah suatu metode atau algoritma unsupervised learning yang digunakan pada data yang belum memiliki label. Tujuan dati metode atau algoritma k – Means clustering ini adalah membagi suatu data dalam beberapa cluster denganberdasarkan jumlah k yang sudah ditentukan dan diwakili oleh Mean (Rata - rata).

K – means clustering mempunyai adalah metode atau algoritma yang sangat umum, karena K – means clustering sangat fleksibel dan mudah beradaptasi. K – means clustering juga menggunakan prinsip yang sederhana dan waktu pengeksekusiannya relative cepat. Tetapi, dalam implementasinya K – means clustering mempunyai beberapa hambatan, yaitu K – means clustering tidak berjalan digunakan untuk data yang berjumlah sangat banyak. Karena k pada K – means clustering random, maka tidak menjamin bahwa metode ini selalu menghasilkan cluster yang optimal.

Contoh kasus :

Ditentukan banyaknya cluster yang dibentuk dua (k=2). Banyaknya cluster harus lebih kecil dari pada banyaknya data (k<n). Inisialisasi centroid dataset pada tabel dataset diatas adalah C1 = {1 , 1} dan C2 = {2 , 1}. Inisialiasasi centroid dapat ditentukan secara manual ataupun random.

n	a	b
1	1	1
2	2	1
3	4	3
4	5	4

Contoh Dataset K-means

Untuk pengulangan berikutnya (pengulangan ke-1 sampai selesai), centroid baru dihitung dengan menghitung nilai rata-rata data pada setiap cluster. Jika centroid baru berbeda dengan centroid sebelumnya, maka proses dilanjutkan ke langkah berikutnya. Namun jika centroid yang baru dihitung sama dengan centroid sebelumnya, maka proses clustering selesai. Rumus yang digunakan untuk menghitung jarak data dengan centroid menggunakan *Euclidiean Distance*.

$$[(x,y),(a,b)] = \sqrt{(x-a)^2 + (y-b)^2}$$

persamaan Euclidean Distance

Pengulangan 1

Jarak data dengan Centroid 1 (C1) :

$$d(x_1,c_1) = \sqrt{(a_1 - c_{1a})^2 + (b_1 - c_{1b})^2} = \sqrt{(1-1)^2 + (1-1)^2} = 0$$
$$d(x_2,c_1) = \sqrt{(a_2 - c_{1a})^2 + (b_2 - c_{1b})^2} = \sqrt{(2-1)^2 + (1-1)^2} = 1$$
$$d(x_2,c_1) = \sqrt{(a_3 - c_{1a})^2 + (b_3 - c_{1b})^2} = \sqrt{(4-1)^2 + (3-1)^2} = 3.605551$$
$$d(x_4,c_1) = \sqrt{(a_4 - c_{1a})^2 + (b_4 - c_{1b})^2} = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

Pengulangan ke-1 C1 K-means

Jarak data dengan Centroid 2 (C2) :

$$d(x_1,c_2) = \sqrt{(a_1 - c_{2a})^2 + (b_1 - c_{2b})^2} = \sqrt{(1-2)^2 + (1-1)^2} = 1$$
$$d(x_2,c_2) = \sqrt{(a_2 - c_{2a})^2 + (b_2 - c_{2b})^2} = \sqrt{(2-2)^2 + (1-1)^2} = 0$$
$$d(x_2,c_2) = \sqrt{(a_3 - c_{2a})^2 + (b_3 - c_{2b})^2} = \sqrt{(4-2)^2 + (3-1)^2} = 2.828427$$
$$d(x_4,c_2) = \sqrt{(a_4 - c_{2a})^2 + (b_4 - c_{2b})^2} = \sqrt{(5-2)^2 + (4-1)^2} = 4.242641$$

Pengulangan ke-1 C2 K-means

Hitung jarak pada setiap baris data,

n	a	b	dc1	dc2
1	1	1	0	1
2	2	1	1	0
3	4	3	3.605551	2.828427
4	5	4	5	4.242641

Kelompokan data sesuai dengan cluster-nya, yaitu data yang memiliki jarak terpendek.

Setelah mendapatkan label cluster, cari nilai rata-ratanya dengan menjumlahkan seluruh anggota masing-masing cluster dan dibagi jumlah anggotanya.

	a	b
c1	1	1
c2	3.666667	2.666667

Nilai Rata-Rata Centroid pada Pengulangan ke-1

Tahap ini dilakukan hingga pengulangan seluruhnya selesai. Seperti yang sudah dipaparkan diatas, centroid dihitung hingga pengulangan tertentu. Pada contoh soal ini, centroid dilakukan sebanyak pengulangan 3 kali, karena centroid baru tidak mengalami perubahan dengan centroid sebelumnya, maka proses clustering selesai.

	a	b
c1	1.5	1
c2	4.5	3.5

Nilai Rata-Rata Centroid pada Pengulangan ke-3

2. Konsep Dasar Agglomerative Hierarchical Clustering

Agglomerative Hierarchical Clustering adalah jenis pengelompokan hierarkis yang paling umum digunakan untuk mengelompokkan objek dalam kelompok berdasarkan kesamaan mereka. Agglomerative Hierarchical Clustering mengelompokkan data dari kelompok yang terkecil hingga kelompok yang terbesar.

Algoritma Agglomerative Hierarchical Clustering :

1. Hitung Matrik Jarak antar data.
2. Gabungkan dua kelompok terdekat berdasarkan parameter kedekatan yang ditentukan.
3. Ulangi langkah 2 hingga hanya satu kelompok yang tersisa.
4. Perbarui Matrik Jarak antar data untuk merepresentasikan kedekatan diantara kelompok baru dan kelompok yang masih tersisa.
5. Selesai.

Menentukan jarak menggunakan Manhattan Distance:

$$D_{man}(x,y)=\sum_{j=1}^{d}\left|x_j-y_j\right|$$

Atau menggunakan Euclidian Distance:

$$D(x_2,x_1)=\sqrt{\sum_{j=1}^{d}\left|x_{2j}-x_{1j}\right|^2}$$

Langkah Pengerjaan menggunakan metode Single Linkage::

1. Hitung jarak pada semua pasangan antar 2 data yang ada (x dan y)

Dman	1	2	3	4	5
1	0	3	1	5	7
2	3	0	4	4	4
3	1	4	0	4	6
4	5	4	4	0	2
5	7	4	6	2	0

2. Pilih jarak 2 kelompok terkecil

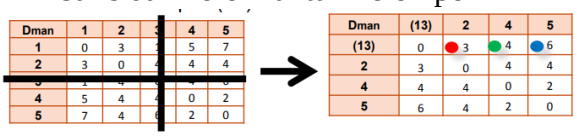
$$\min(D_{man})=\min(d_{13})=1$$

terpilih kelompok 1 dan 3, sehingga kedua kelompok ini digabungkan.

3. Hitung jarak 2 kelompok yang terpilih, dengan kelompok lain yang tersisa

$$d_{(13)2}=\min\{d_{12},d_{32}\}=\min\{3,4\}=3$$
$$d_{(13)4}=\min\{d_{14},d_{34}\}=\min\{5,4\}=4$$
$$d_{(13)5}=\min\{d_{15},d_{35}\}=\min\{7,6\}=6$$

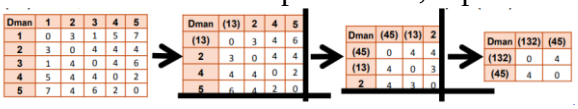
4. Hapus baris-baris dan kolom-kolom matrik jarak yang bersesuaian dengan kelompok 1 dan 3, serta menambahkan baris dan kolom untuk kelompok



5. Pilih jarak kelompok terkecil

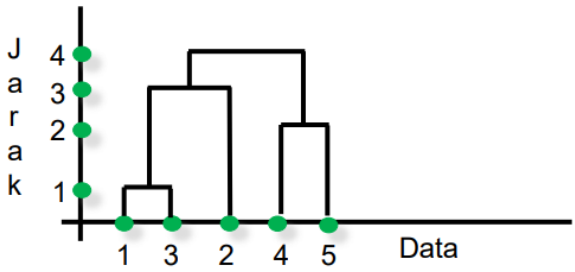
$$\min(D_{man})=\min(d_{45})=2$$

6. Lakukan langkah 2 – 5 hingga jarak anggota kelompok saling membentuk kelompok baru dan 1 kelompok tersisa, seperti ini :



Dman	(132)	(45)
(132)	0	4
(45)	4	0

7. Jadi kelompok (132) dan (45) digabung untuk menjadi kelompok tunggal dari lima data, yaitu kelompok (13245) dengan jarak terdekat 4. Hasil yang didapat::



3. Metode Self Organizing Map (SOM)

a. Pendahuluan

Metode Self Organizing Map (SOM) adalah jenis jaringan saraf tiruan (JST) yang dilatih menggunakan pembelajaran tanpa pengawasan untuk menghasilkan representasi dimensi rendah (biasanya dua dimensi), diskrit dari ruang input sampel pelatihan, yang disebut memetakan, dan karenanya merupakan metode untuk melakukan pengurangan dimensionalitas. Peta yang diatur sendiri berbeda dari jaringan saraf tiruan lainnya karena mereka menerapkan pembelajaran kompetitif sebagai lawan pembelajaran koreksi-kesalahan (seperti backpropagation dengan gradient descent), dan dalam arti bahwa mereka menggunakan fungsi lingkungan untuk melestarikan sifat topologi dari ruang input.

b. Deskripsi Soal Masalah

Masalah dari soal ini yaitu membangun sebuah model klasterisasi (clustering) menggunakan metode Self Organizing Map (SOM) untuk menghasilkan sejumlah klaster yang paling optimum dengan dataset yang di sajikan sebanyak 600 objek data yang memiliki dua atribut tanpa memiliki label data kelas.

c. Metode Penyelesaian

Pertama dataset yang berupa file .csv di read untuk di ambil datanya dengan code :

```
with open('dataset.csv', 'r') as f:
    reader = csv.reader(f)
    dataTrain = [float(r[0]), float(r[1]), -1] for r in reader]
```

Selanjutnya adalah inisialisasi , dengan code:

```
ns = 1200
lr = 0.1
thlr = 2
radius = 2
thradius = 2
convergence = 0.000000001
best = []
tempDW = 0
iterations = 1000
colours = ['black', 'grey', 'red', 'green', 'blue', 'yellow', 'magenta', 'tan', 'aqua', 'violet', 'crimson', 'pink']
```

ns merupakan ukuran dari neuron yang nantinya akan digunakan untuk menentukan jumlah hasil cluster. Nilai ns adalah 1200 untuk melihat banyak kemungkinan cluster yang terbentuk dengan optimal. Lr adalah learning rate dengan nilai 0,1 sebagai nilai yang ideal. Untuk tetha learning rate, radius atau sigma, dan tetha radius atau sigma, nilai yang di ambil disamakan dengan nilai pada slide. Variabel

convergence yang bernilai 0,0000001 berfungsi untuk memberhentikan perulangan. Dan iterasi di set 1000 dengan kemungkinan mendapatkan hasil cluster yang optimal.

Selanjutnya adalah meng generate nilai neuron secara random, dengan code :

```
neurons = [list(np.insert(np.random.rand(2), 2, x)) for x in range(ns)]
```

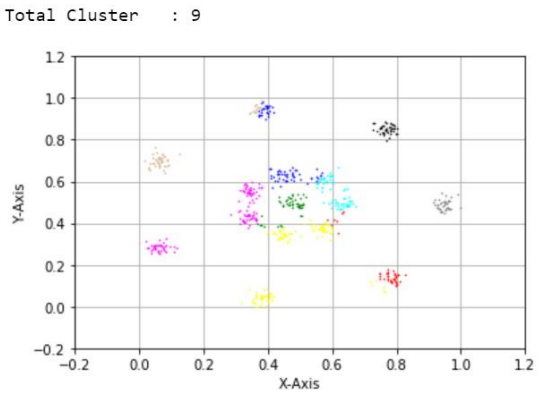
langkah selanjunya adalah perulangan iterasi, dengan code :

```
for iteration in range(iterations):
    data_rand = dataTrain1[np.random.randint(0,599)]
    #menghitung dataset secara acak ke semua neuron
    dj=[]
    for neuron in neurons :
        d= RumusEuc(data_rand,neuron)
        dj.append(d)
    #menentukan winner
    win = neurons[np.argmin(dj)]
    if np.argmin(dj) not in best:
best.append(np.argmin(dj))
    #menentukan tetanggaa bedasarkan radius
    tetangga = []
    for n in range(len(neurons)):
        if RumusEuc(win, neurons[n]) <= radius:

tetangga.append(n)
    #update neuron
    for p in tetangga :
        s = RumusEuc(win, neurons[p])
        tij= np.exp((-s**2)/(2 * radius ** 2))
        dW = lr * tij * (RumusEuc(data_rand, neurons[p]))
        neurons[p][0] += dW
        neurons[p][1] += dW
    dW = lr * tij* (RumusEuc(data_rand, win))
    a= np.absolute(dW - tempDW)
    if a < convergence:
        break
    lr *= np.exp(-iteration / thlr)
    radius *= np.exp(-iteration / thradius)
#train dataset
for g in dataTrain1:
    dj = []
    for u in best:

dj.append(RumusEuc(g, neurons[u]))
    win = neurons[np.argmin(dj)]
    g[2] = int(win[2])
```

d. Output Program



4. Sumber

1. Available at: https://informatikalogi.com/algoritma-k-means-clustering/ . Accessed : 21-04-2019.
2. Available at: https://brotodata.com/2018/06/28/penjelasan-k-means-clustering/ . Accessed : 21-04-2019.
3. Available at : https://towardsdatascience.com/self-organizing-maps-ff5853a118d4 . Accessed : 21-04-2019.
4. Available at : https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/ . Accessed : 21-04-2019.