

Nama : Fuad Zauqi Nur

Kelas : IF-40-03

NIM : 1301164392

Tugas 2

CSH3L3 Pembelajaran Mesin Genap 2018/2019

Jelaskan kelebihan dan kekurangan *k-Means Clustering* dengan contoh kasus dan ilustrasi yang representatif. Gunakan bahasa Anda sendiri. (CLO1: 5%).

Jawab :

Kelebihan dan Kekurangan k-Means Clustering menurut Fuad Zauqi Nur

Kelebihan pada k-Means Clustering diantaranya mudah untuk diimplementasi, waktu untuk pembelajarannya cepat, prinsip yang sederhana mudah dijelaskan dalam non-statistik, dikarenakan prinsip yang sederhana, memungkinkan pengguna k-Means Clustering lebih umum dan yang terakhir adalah algoritma yang sangat fleksibel sehingga adaptasi yang mudah untuk dilakukan.

Kekurangan k-Means Clustering diantaranya adalah K diinisialisasikan secara random yang mengakibatkan pengelompokan data yang dijalankan berbeda-beda, apabila nilai yang diperoleh kurang baik bisa didapatkan pengelompokan yang kurang optimal. Kekurangan selanjutnya adalah ketika k-Means Clustering digunakan untuk pembelajaran pada dimensi yang sangat banyak akan kesulitan dan memakan banyak waktu untuk memperoleh jarak k titik dan titik lainnya. Kekurangan selanjutnya adalah jika terdapat banyak titik data sampel misalkan dua juta data, maka perhitungan dan pencarian titik terdekat akan membutuhkan waktu yang sangat lama, proses ini dapat dipercepat tapi membutuhkan struktur data yang lebih kompleks seperti kD-tree atau hashing untuk melakukan proses tersebut. Dan yang terakhir adalah penggunaan k buah random yang mengakibatkan tidak adanya jaminan untuk menemukan kumpulan kluster yang optimal.

Nama : Fuad Zauqi Nur

Kelas : IF-40-03

NIM : 1301164392

Jelaskan konsep dasar *Agglomerative Hierarchical Clustering* dengan contoh kasus dan ilustrasi yang representatif. Gunakan bahasa Anda sendiri. (CLO1: 5%).

Jawab :

Konsep dasar Agglomerative Hierarchical Clustering

Hierarchical Clustering adalah metode analisis kelompok yang pembelajarannya untuk membangun sebuah hirarki kelompok data. Pengelompokannya terdapat 2 yaitu Agglomerative (Bottom-Up) dan Devisive (Top-Down), pada jawaban ini akan membahas mengenai Agglomerative (Bottom-Up).

Langkah Algoritma Agglomerative Hierarchical Clustering sebagai berikut :

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

Catatan :

1. Membentuk Matrik Jarak bisa menggunakan Manhattan Distance atau Euclidian Distance
2. Metode pengelompokan AHC adalah :
 - a. Jarak Terdekat
$$d_{uv} = \min\{d_{uv}\}, d_{uv} \in D$$
 - b. Jarak Terjauh
$$d_{uv} = \max\{d_{uv}\}, d_{uv} \in D$$
 - c. Jarak Rata-Rata
$$d_{uv} = \text{average}\{d_{uv}\}, d_{uv} \in D$$

Nama : Fuad Zauqi Nur

Kelas : IF-40-03

NIM : 1301164392

I. Deskripsi Masalah

Diberikan sebuah dataset berisi 600 objek data yang memiliki dua atribut tanpa label kelas. Bangunlah sebuah model klasterisasi (*clustering*) menggunakan metode *Self Organizing Map* (SOM) untuk menghasilkan sejumlah klaster yang paling optimum.

II. Strategi Penyelesaian

IDE yang digunakan adalah Jupyter python 3, dan berikut adalah source code sekaligus penjelasan mengenai kodingannya.

Self Organized Maps (Tugas 2 Machine Learning) Fuad Zauqi Nur (1301164392)

Import beberapa library yang diperlukan seperti csv, random, math, matplotlib, dll untuk mendukung algoritma yang berjalan.

```
In [979]: import csv
import random
import math
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

Inisialisasi beberapa variabel seperti Neighborhood dan Learning Rate. Keterangan : Learning Rate Awal = 0.6 Neighborhood Awal = 3 Jumlah Neuron = 12

```
In [980]: nh = 3
lr = 0.6
jneuron = 12
dneuron = []
x1 = []
x2 = []
list_bobot=[]
neuron = 0
bobotbaru1=0
bobotbaru2=0
train1=[]
train2=[]
```

Mengimport DatasetTanpaLabel.csv dan disimpan pada variabel datas, lalu di jadikan dua list yaitu kolom 1 dan kolom 2 dan diubah tipe data setiap valuenya menjadi float agar dapat dioperasikan.

```
In [981]: datas = csv.reader(open("DatasetTanpaLabel.csv"))
for row in datas:
    x1.append(float(row[0]))
    x2.append(float(row[1]))
```

Membuat Neuron 2 dimensi sebanyak 12 secara acak lalu disimpan pada list dneuron

```
In [982]: for h in range(jneuron):
dneuron.append([np.random.uniform(4,16),np.random.uniform(4,16)])
```

Rumus Euclidean Distance untuk menghitung jarak antar neuron dengan objeknya

```
In [983]: def eucl_dist(o1,o2,n):
return math.sqrt((n[0]-o1)**2+(n[1]-o2)**2)
```

Nama : Fuad Zauqi Nur

Kelas : IF-40-03

NIM : 1301164392

Fungsi untuk mencari Neuron pemenang setelah dihitung jarak menggunakan Euclidean Distance

```
In [984]: def neuronwin(d1,d2,d3,d4,d5,d6,d7,d8,d9,d10,d11,d12):
    if d1<d2 and d1<d3 and d1<d4 and d1<d5 and d1<d6 and d1<d7 and d1<d8 and d1<d9 and d1<d10 and d1<d11 and d1<d12 :
        return 0
    elif d2<d1 and d2<d3 and d2<d4 and d2<d5 and d2<d6 and d2<d7 and d2<d8 and d2<d9 and d2<d10 and d2<d11 and d2<d12 :
        return 1
    elif d3<d1 and d3<d2 and d3<d4 and d3<d5 and d3<d6 and d3<d7 and d3<d8 and d3<d9 and d3<d10 and d3<d11 and d3<d12 :
        return 2
    elif d4<d1 and d4<d2 and d4<d3 and d4<d5 and d4<d6 and d4<d7 and d4<d8 and d4<d9 and d4<d10 and d4<d11 and d4<d12 :
        return 3
    elif d5<d1 and d5<d2 and d5<d3 and d5<d4 and d5<d6 and d5<d7 and d5<d8 and d5<d9 and d5<d10 and d5<d11 and d5<d12 :
        return 4
    elif d6<d1 and d6<d2 and d6<d3 and d6<d4 and d6<d5 and d6<d7 and d6<d8 and d6<d9 and d6<d10 and d6<d11 and d6<d12 :
        return 5
    elif d7<d1 and d7<d2 and d7<d3 and d7<d4 and d7<d5 and d7<d6 and d7<d8 and d7<d9 and d7<d10 and d7<d11 and d7<d12 :
        return 6
    elif d8<d1 and d8<d2 and d8<d3 and d8<d4 and d8<d5 and d8<d6 and d8<d7 and d8<d9 and d8<d10 and d8<d11 and d8<d12 :
        return 7
    elif d9<d1 and d9<d2 and d9<d3 and d9<d4 and d9<d5 and d9<d6 and d9<d7 and d9<d8 and d9<d10 and d9<d11 and d9<d12 :
        return 8
    elif d10<d1 and d10<d2 and d10<d3 and d10<d4 and d10<d5 and d10<d6 and d10<d8 and d10<d9 and d10<d7 and d10<d11 and d10<d12 :
        return 9
    elif d11<d1 and d11<d2 and d11<d3 and d11<d4 and d11<d5 and d11<d6 and d11<d8 and d11<d9 and d11<d10 and d11<d7 and d11<d12 :
        return 10
    else :
        return 11
```

Fungsi S adalah mencari jarak neuron indeks j ke indeks neuron pemenang Fungsi T adalah rumus T untuk mencari bobot baru Fungsi bobotbaru1 adalah fungsi untuk mencari bobot baru pada kolom 1 Fungsi bobotbaru2 adalah fungsi untuk mencari bobot baru pada kolom 2

```
In [999]: def S(n,win):
    return math.sqrt((n[0]-win[0])**2+(n[1]-win[1])**2)
def T(s,lr):
    return math.exp((-s**2)/2*nh**2)
def bobotbaru1(n,T,x1,w,i):
    return n*T*(x1[i]-w)
def bobotbaru2(n,T,x2,w,i):
    return n*T*(x2[i]-w)
```

Program utama pada program ini. variabel c adalah jumlah iterasi pada looping. nilai i didapat secara random dari 0 hingga 599, berfungsi untuk memanggil index i secara acak. D1-D12 mencari jarak dengan fungsi euc_dist, lalu mencari jarak minimum dengan fungsi neuronwin(). variabel j adalah iterasi untuk sejumlah neuron. x adalah variabel penampung untuk mencari S. y adalah variabel penampung untuk mencari t. Melakukan penghitungan bobot baru menggunakan fungsi bobotbaru1() dan bobotbaru2() Kemudian memperbarui nilai learning rate dan neighborhood. Nilai pada dneuron kolom 1 dan kolom 2 dimasukan kedalam 2 list berbeda yang berfungsi untuk agar mudah dipanggil untuk memvisualisasikan hasil akhir menggunakan matplotlib. Visualisasi dengan warna biru sebagai objeknya dan kuning sebagai neuronnya.

Nama : Fuad Zauqi Nur

Kelas : IF-40-03

NIM : 1301164392

```
In [1000]: c=0
while c < 102000:
    i=np.random.randint(0,599)
    D1=eucl_dist(x1[i],x2[i],dneuron[0])
    D2=eucl_dist(x1[i],x2[i],dneuron[1])
    D3=eucl_dist(x1[i],x2[i],dneuron[2])
    D4=eucl_dist(x1[i],x2[i],dneuron[3])
    D5=eucl_dist(x1[i],x2[i],dneuron[4])
    D6=eucl_dist(x1[i],x2[i],dneuron[5])
    D7=eucl_dist(x1[i],x2[i],dneuron[6])
    D8=eucl_dist(x1[i],x2[i],dneuron[7])
    D9=eucl_dist(x1[i],x2[i],dneuron[8])
    D10=eucl_dist(x1[i],x2[i],dneuron[9])
    D11=eucl_dist(x1[i],x2[i],dneuron[10])
    D12=eucl_dist(x1[i],x2[i],dneuron[11])
    neuron=neuronwin(D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12)
    j=0
    while j <=11 :
        x=S(dneuron[j],dneuron[neuron])
        y=T(x,lr)
        dneuron[j][0]= bobotbaru1(lr,y,x1,dneuron[neuron][0],i) + dneuron[j][0]
        dneuron[j][1]= bobotbaru2(lr,y,x2,dneuron[neuron][1],i) + dneuron[j][1]
        j=j+1
    lr=lr*(math.exp(-(i+1)/102001))
    nh=nh*(math.exp(-(i+1)/102001))
    c=c+1
for row in dneuron:
    train1.append(row[0])
    train2.append(row[1])
print(dneuron)
plt.scatter(x1, x2, c='blue', marker='o')
plt.scatter(train1,train2, c='yellow', marker='o')
plt.show()
```

```
471], [12.809658073940918, 4.512516300030707], [10.80253406720234, 11.00662404492172], [13.688364287607232, 14.803238511
524594], [4.324276310818119, 7.187573493829166], [8.72874080829497, 3.8181783085850354], [8.578048787571944, 16.37160513
8661963], [10.27584453345188, 8.255427855230609], [14.338776998701087, 4.998009218087812], [8.113527058148442, 9.73175407
8455255]]
```

