BAB 7. CLUSTERING

Danny Kurnianto, 10/305827/PTK/06844
Jurusan Teknik Elektro dan Teknologi Informasi UGM
Yogyakarta

7.1 PENDAHULUAN

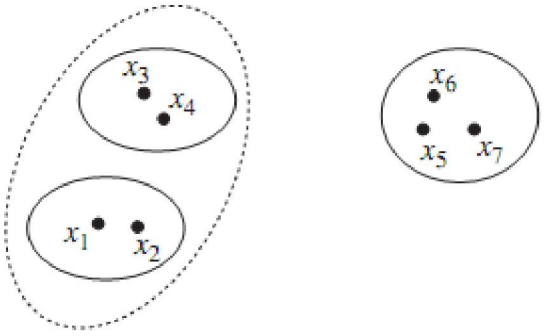
Pada bab sebelumnya, kita membahas mengenai pengenalan pola terbimbing (supervised)-yaitu dimana label kelas untuk setiap pola-pola pelatihan telah diketahui. Pada bab ke-7 ini, kita akan membahas kasus pengenalan pola tak terbimbing (unsupervised), dimana informasi mengenai label kelas tidak tersedia. Tujuannya sekarang adalah menentukan kelas-kelas yang mungkin dibentuk oleh pola-pola yang tersedia tersebut dalam rangka untuk mengekstrak informasi yang berguna yang berkaitan dengan kesamaan atau perbedaan diantara pola-pola tersebut.

7.2 KONSEP DASAR DAN DEFINISI

Seperti yang telah dibahas sejauh ini, kita mengasumsikan bahwa setiap pola pelatihan diwakili oleh seperangkat ciri l yang berbentuk sebuah vektor dimensi l, $x = [x(1),\ldots,x(l)]^T$. Jadi setiap pola pelatihan sesuai dengan titik (vektor) pada ruang dimensi l.

Definisi clustering: Diberikan seperangkat data vektor $X = \{x_1,\ldots..x_N\}$. Kelompokkan data vektor tersebut sedemikian hingga vektor yang lebih mirip akan berada dalam kelas yang sama dan vektor yang kurang mirip akan berada pada kelas yang berbeda. Seperangkat $\Re$ yang mengandung kelas-kelas ini disebut sebuah clustering dari X. (definisi yang lebih lengkap mengenai clustering dapat ditemukan di [Theo 09, Section 11.1.3]).

Contoh 7.2.1 Perhatikan vektor-vektor data seperti yang ditunjukkan pada Gambar 7.1. Dua clustering yang sejalan dengan definisi hanya diberikan oleh $\Re_1$={{x1,x2},{x3,x4},{x5,x6,x7}} dan $\Re_2$={{x1,x2,x3,x4},{x5,x6,x7}}.



Gambar 7.2.1 Contoh clustering pada contoh 7.2.1

Kedua kelas tersebut masuk akal dalam arti bahwa vektor-vektor yang dekat satu dengan yang lain termasuk kedalam kelas yang sama. Namun tidak ada informasi tambahan untuk menunjukkan kelas mana yang akhirnya harus dipilih. Secara umum, pendekatan terbaik saat berurusan dengan masalah clustering adalah berkonsultasi dengan para pakar dibidangnya.

Kemungkinan lain yang dapat di clustering-kan (kelompokan) yaitu $\Re_3$={{x1,x7},{x3,x4},{x5,x6,x2}}. Tapi tidak masuk akal kalau mengelompokkan x_1 dan x_7, dimana keduanya mempunyai jarak yang jauh satu sama lainnya sehingga tampak bertentangan dengan penalaran fisik. Hal yang sama berlaku untuk pengelompokan x_2 dengan x_5,x_6.

Catatan

- Istilah clustering tidak mempunyai definisi yang tegas. Ketidakmampuan untuk memberikan definisi yang tegas pada masalah clustering berasal dari fakta bahwa tidak adanya informasi luar (label kelas) yang tersedia, dan istilah kesamaan itu sendiri hilang. Sehingga konsekuensinya adalah subyektifitas merupakan ciri yang tak terhindarkan saat kita melakukan clustering (pengelompokkan).
- Beberapa pola mungkin termasuk ke dalam kelas tunggal (hard clustering) atau mungkin termasuk milik bersama lebih dari satu kelas sampai derajat tertentu (fuzzy clustering).
- Tingkat kedekatan dapat diukur melalui ukuran kedekatan. Ukuran kedekatan ini bisa menjadi ukuran kesamaan (biasanya dilambangkan dengan notasi s) dan ukuran perbedaan (biasanya dilambangkan dengan notasi d). Selain itu tergantung dari metode clustering yang digunakan, kedekatan dapat ditentukan (a) antara vektor dengan vektor (b) antara vektor dengan seperangkat vektor (c) antara seperangkat vektor dengan seperangkat vektor. [Theo 09, Section 11.2].

7.3 ALGORITMA CLUSTERING

Pendekatan naif untuk menentukan clustering yang paling sesuai dengan seperangkat data X adalah dengan mempertimbangkan semua clustering yang mungkin dan pilih salah satu yang paling masuk akal sesuai dengan kriteria dan rasionalitas. Sebagai contoh, seseorang dapat memilih clustering yang mengoptimasi kriteria yang terpilih, mengkuantisasi vektor-vektor yang lebih mirip kedalam satu kelas yang sama dan vektor-vektor yang kurang mirip kedalam kelas yang berbeda. Namun, jumlah semua clustering yang mungkin terjadi adalah besar, bahkan untuk sejumlah pola N yang tidak terlalu banyak. Cara untuk mengatasi masalah ini adalah dengan mengembangkan algoritma clustering, yang hanya mempertimbangkan sebagian kecil dari clustering yang mungkin terjadi. Pertimbangan clustering tergantung pada prosedur algoritma yang spesifik.

Beberapa algoritma clustering telah dikembangkan, beberapa diantaranya merupakan clustering tunggal, dan yang lainnya adalah clustering hierarki. Klasifikasi berikut berisi sebagian besar algoritma clustering yang terkenal.

Algoritma clustering tunggal meliputi :

- Sequential algorithms, dengan konsep sederhana, bekerja pada seperangkat data tunggal atau data yang sangat sedikit [Theo 09, Chapter 12].
- Cost function optimization algorithms, yang mengadopsi fungsi biaya J dengan mengkuantisasi istilah “masuk akal (sensible)” dan menghasilkan clustering dengan optimasi J. Yang termasuk dari kategori ini adalah hard clustering algoritms seperti k-means, fuzzy clustering algoritms seperti fuzzy c-means (FCM), probabilistic clustering algoritms seperti EM dan probabilistic algoritm [Theo 09, Chapter 14].
- Miscellaneous algorithms, yang tidak sesuai dengan kategori sebelumnya, sebagai contoh competitive learning algorithms, valley-seeking algorithms, density-based algorithms, and subspace-clustering algorithms [Theo 09, Chapter 15].

Algoritma clustering hierarki meliputi :

- Agglomerative algorithms, yang menghasilkan clustering sekuensial dari pengurangan sejumlah kelas, m. Pada setiap tahap, pasangan kelas terdekat pada clustering saat ini diidentifikasi dan digabung menjadi satu dalam rangka untuk membangkitkan clustering berikutnya.
- Divisive algorithms, yang berbeda dengan agglomerative algorithms, menghasilkan clustering sekuensial dari penambahan sejumlah kelas, m. Pada setiap tahap, sebuah kelas yang telah dipilih dibagi menjadi dua kelas yang lebih kecil [Theo 09, Chapter 13].

7.4 SEQUENTIAL ALGORITHM

Pada bagian ini, akan dibahas skema dasar algoritma sekuensial (BSAS) serta beberapa perbaikannya.

7.4.1 BSAS Algorithm

Algoritma BSAS bekerja single pass pada seperangkat data yang diberikan. Selain itu, setiap kelas diwakili oleh rerata vektor-vektor yang telah ditetapkan. BSAS bekerja sebagai berikut. Untuk setiap vektor x baru , jarak dari kelas yang telah terbentuk dihitung. Jika jaraknya lebih besar dari nilai ambang perbedaan Θ dan jika jumlah maksimum kelas q belum tercapai, sebuah kelas baru yang berisi x terbentuk. Jika tidak, x ditetapkan ke kelas terdekat dan wakil yang sesuai diperbaharui. Algoritma berakhir ketika semua vektor-vektor data telah dipertimbangkan sekali.

Untuk menerapkan BSAS pada seperangkat data X, maka tulis

[bel, repre] = BSAS(X,theta,q,order)

Dimana :

X adalah matrik l x N yang berisi vektor-vektor data pada kolom.

theta adalah nilai ambang perbedaan.

q adalah jumlah kelas maksimum.

order adalah sebuah vektor dimensi N yang berisi permutasi integer 1,2,….N, dimana elemen ke-i yang menentukan urutan penyajian dari vektor ke-i untuk algoritma,

bel adalah sebuah vektor dimensi N elemen ke-i yang menunjukkan kelas dimana vektor data ke-i ditetapkan.

repre adalah sebuah matrik yang berisi rerata dimensi l yang mewakili kelas pada kolom.

Catatan

- Aslinya, BSAS cocok untuk mengungkap kelas-kelas yang rapat (seperti kelas yang mempunyai titik-titik yang berkumpul disekitar titik tertentu pada ruang data).
- Algoritma sensitif terhadap urutan penyajian data dan pemilihan parameter Θ. Jika data yang disajikan dalam urutan yang berbeda dan/atau parameter Θ diberikan dengan nilai yang berbeda, maka clustering yang berbeda akan dihasilkan.
- BSAS bekerja cepat karena BSAS memerlukan single pass pada seperangkat data X. Jadi BSAS merupakan pilihan yang baik untuk memproses seperangkat data yang besar. Tetapi pada beberapa kasus mungkin akan menghasilkan clustering dengan kualitas yang rendah. Peningkatan dapat dicapai pada tahap perbaikan, yang akan didiskusikan pada sub bagian berikutnya.

- Terkadang, clustering yang dihasilkan oleh BSAS digunakan sebagai titik awal untuk algoritma clustering pintar yang lain, untuk memperoleh clustering dengan kualitas yang telah ditingkatkan.
- BSAS menghasilkan perkiraan kasar jumlah kelas yang mendasari seperangkat data.

Agar perkiraan jumlah kelas lebih akurat pada data X, BSAS mungkin bekerja untuk nilai parameter Θ yang berbeda pada jangkauan [Θmin, Θmax]. Untuk setiap nilai r, BSAS berjalan menggunakan urutan penyajian data yang berbeda; kemudian jumlah kelas m_Θ yang paling sering ditemukan diidentifikasi dan menggambarkan grafik m_Θ terhadap Θ. Daerah datar pada grafik merupakan indikasi adanya kelas-kelas; daerah paling datar kemungkinan besar sesuai untuk jumlah kelas yang mendasari X.

7.4.2 Perbaikan Clustering

Prosedur Reassignment

Prosedur ini diterapkan pada clustrering yang telah siap dihasilkan. Prosedur ini melakukan single pass atas seperangkat data. Kelas yang paling dekat dengan setiap vektor x ditentukan. Setelah semua vektor dipertimbangkan, setiap kelas ditetapkan kembali menggunakan vektor yang paling dekat dengan kelas tersebut. Jika kelas yang mewakilinya digunakan, kelas-kelas tersebut diperkirakan ulang (perwakilan yang biasa adalah rerata semua vektor dalam kelas).

Untuk menerapkan prosedur reassignment, tulis

[bel,new_repre] = reassign(X,repre,order)

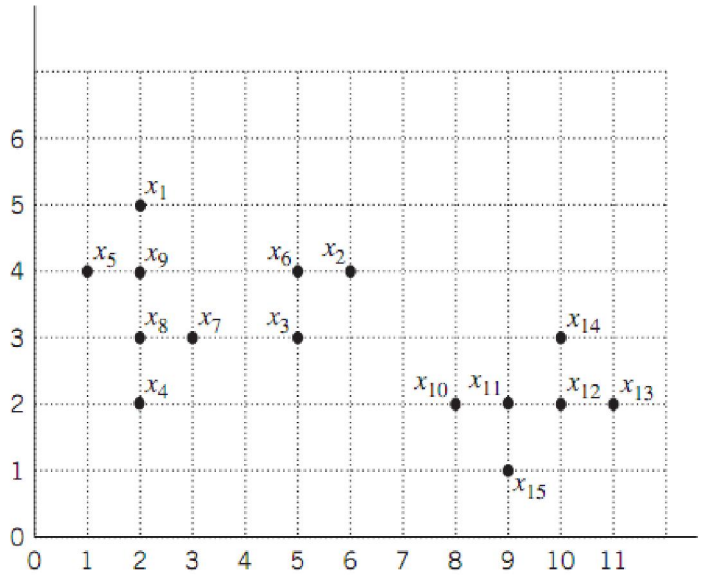
dimana new_repre berisi nilai perkiraan ulang dari rerata vektor kelas-kelas. Semua parameter yang lain ditetapkan sebagai fungsi BSAS.

Prosedur Merging

Prosedur ini juga diterapkan pada clustering $\Re$ dari seperangkat data yang diberikan. Dengan tujuan untuk menggabungkan kelas-kelas pada $\Re$ yang memperlihatkan kesamaan yang tinggi (perbedaannya rendah). Khususnya, pasangan kelas yang sesuai dengan ukuran perbedaan yang telah dipilih lebih dulu antara kelas-kelas, menunjukkan perbedaan yang paling besar ditentukan. Jika hal ini lebih besar daripada nilai ambang perbedaan kelas M_1, prosedur dihentikan. Pada sisi lain, dua kelas digabung dan prosedur diulangi pada clustering yang dihasilkan.

Contoh 7.4.1 Perhatikan seperangkat data 2 dimensi X yang berisi vektor-vektor sebagai berikut: $x_1=[2,5]^T$, $x_2=[6,4]^T$, $x_3=[5,3]^T$, $x_4=[2,2]^T$, $x_5=[1,4]^T$, $x_6=[5,4]^T$, $x_7=[3,3]^T$, $x_8=[2,3]^T$, $x_9=[2,4]^T$, $x_{10}=[8,2]^T$, $x_{11}=[9,2]^T$, $x_{12}=[10,2]^T$, $x_{13}=[11,2]^T$, $x_{14}=[10,3]^T$, $x_{15}=[9,1]^T$. Gambar 7.2 menunjukkan seperangkat data diatas.

1. Terapkan algoritma BSAS pada X, dengan urutan elemen sekarang adalah x_8, x_6, x_{11}, x_1, x_5, x_2, x_3, x_4, x_7, x_{10}, x_9, x_{12}, x_{13}, x_{14}, x_{15} untuk $\Theta = 2,5$ dan q = 15.
2. Ulangi langkah 1, sekarang dengan urutan penyajian data vektor sebagai berikut x_7, x_3 , x_1, x_5, x_9, x_6, x_8, x_4, x_2, x_{10}, x_{15}, x_{13}, x_{14}, x_{11}, x_{12}.
3. Ulangi langkah 1, sekarang dengan $\Theta = 1,4$.
4. Ulangi langkah 1, sekarang dengan jumlah maksimum yg diperblehkan untuk kelas, q sama dengan 2



Gambar 7.2 Seperangkat data pada contoh 7.4.1

Penyelesaian :

Untuk mendapatkan seperangkat data X (Gambar 7.2), tulis

```
% Untuk menghasilkan seperangkat data
x = [2 5; 6 4; 5 3; 2 2; 1 4; 5 4; 3 3; 2 3; 2 4; 8 2; 9 2; 10 2; 11 2; 10 3; 9 1]' ;
[1,N] = size(x) ;
% Gambarkan data
figure(1), plot(X(1,: ), X(2,: ), '.')
figure(1), axis equal
```

Pemeriksaan secara visual memperlihatkan ada tiga kelas yaitu C1={x_8, x_1, x_5, x_4, x_7, x_9}, C_2={x_6, x_2, x_3}, C_3={x_{11}, x_{10}, x_{12}, x_{13}, x_{14}, x_{15}}. Lanjutkan dengan langkah-langkah sebagai berikut:

Langkah 1. Terapkan algoritma BSAS pada X, tuliskan:

```
q = 15; % jumlah maksimum kelas
theta = 2.5; % nilai ambang perbedaan
order = [8 6 11 1 5 2 3 4 7 10 9 12 13 14 15];
[bel, repre] = BSAS(X, theta, q, order);
```

Clustering yang dihasilkan adalah sama dengan yang diperoleh dari pemeriksaan visual pada langkah sebelumnya.

Langkah 2. Ulangi skrip kode yang diberikan pada langkah 1 untuk urutan penyajian data yang baru, dan hasilnya diperoleh 4 kelas sebagai berikut : C_1={x_7, x_3, x_6, x_2}, C_2={x_1, x_5, x_9, x_8, x_4}, C_3={x_{10}, x_{15}, x_{11}}, C_4={x_{13}, x_{14}, x_{12}}. Hal ini menunjukkan ketergantungan BSAS pada urutan penyajian vektor data.

Langkah 3. Ulangi skrip kode pada langkah 1 untuk Θ = 1,4, hasilnya adalah 6 kelas sebagai berikut : C_1={x_4, x_7, x_8}, C_2={x_2, x_3, x_6}, C_3={x_{10}, x_{11}, x_{15}}, C_4={x_1, x_9}, C_5={x_5}, C_6={x_{12}, x_{13}, x_{14}}. Bandingkan dengan hasil pada langkah 2 untuk melihat pengaruh Θ pada hasil clustering.

Langkah 4. Ulangi langkah 1 dengan q=2 untuk menghasilkan kelas sebagai berikut: C_1={x_1, x_4, x_5, x_7, x_8, x_9}, C_2={x_2, x_3, x_6, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}}. Hasilnya menunjukkan pengaruh dari q pada pembentukan

kelas. Catatan bahwa satu hal yang harus diperhatikan dengan sangat hati-hati yaitu dalam menentukan batas jumlah maksimum kelas yang diperbolehkan, karena jika hal ini diremehkan akan dapat mencegah algoritma menemukan clustering dengan data yang paling cocok.

Pada contoh berikutnya, kami akan menunjukkan bagaimana algoritma BSAS digunakan untuk memperkirakan jumlah kelas (rapat) yang mendasari seperangkat data X.

Contoh 7.4.2 Hasilkan dan gambarkan seperangkat data X_1, yang terdiri dari N=400 vektor data 2-D. Vektor-vektor ini berbetuk empat grup yang berukuran sama, masing-masing grup berisi vektor yang berasal dari distribusi Gaussian dengan nilai rerata $m_1=[0,0]^{T,}$ $m_2=[4,0]$, $m_3=[0,4]$, $m_4=[5,4]^T$ dan matrik kovarian :

$$S_1 = 1, S_2 = \begin{bmatrix} 1 & 0.2 \\ 0.2 & 1.5 \end{bmatrix}, S_3 = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1.1 \end{bmatrix}, S_4 = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.5 \end{bmatrix}$$

Kemudian kerjakan sebagai berikut :

1. Tentukan jumlah kelas yang dibentuk dalam X_1 dengan melakukan langkah sebagai berikut:
   a. Tentukan jarak (Euclidean) maksimum d_{max} dan minimum d_{min} antara dua titik pada data.
   b. Tentukan nilai Θ yang akan menjadikan BSAS berjalan. Hal ini dapat ditetapkan sebagai Θmin, Θmin+s, Θmin+2s,…,Θmax, dimana Θmin = $0{,}25(d_{min}+d_{max})/2$ dan Θmin = $1{,}75(d_{min}+d_{max})/2$ dan s = (Θmin+Θmax) / nΘ-1, dimana nΘ adalah jumlah nilai Θ yang berbeda yang akan dipertimbangkan. Gunakan nΘ=50.
   c. Untuk setiap nilai Θ yang ditetapkan sebelumnya, algoritma BSAS berjalan dengan n_{times} = 10, sehingga vektor-vektor data yang diberikan dengan urutan yang berbeda untuk setiap kali algoritma BSAS berjalan. Dari nilai n_{times} yang memperkirakan jumlah kelas, pilihlah nilai mΘ yang paling sering muncul dan tetapkan sebagai yang paling akurat. m_{tot} menjadi vektor dimensi nΘ yang berisi nilai mΘ.
   d. Gambarkan grafik mΘ terhadap Θ.Tentukan daerah mendatar yang paling luas r dari Θ-nya (meniadakan nilai yang sesuai untuk kasus kelas tunggal) dan nr menjadi jumlah Θ-nya pada {Θmin, Θmin+s,…,Θmax}. Jika nilai nr signifikan (misalkan lebih besar dari 10% nΘ), jumlah kelas yang sesuai dipilih sebagai perkiraan yang terbaik (m_{best}) dan nilai rerata dari Θ dipilih sebagai nilai yang paling sesuai untuk Θ ($Θ_{best}$). Sebaliknya, clustering kelas tunggap dipakai.
2. Jalankan algoritma BSAS untuk $Θ=Θ_{best}$ dan gambarkan grafik data menggunakan warna dan symbol yang berbeda untuk titik-titik dari kelas yang berbeda.
3. Gunakan prosedur reassignment pada hasil clustering yang dihasilkan dalam langkah sebelumnya dan gambarkan clustering yang baru.

Penyelesaian :

Untuk menghasilkan seperangkat data yang dibutuhkan, tulislah :

```
randn('seed' , 0)
m = [0 0; 4 0; 0 4; 5 4];
S( :, :, 1 ) = eye(2);
S( :, :, 2) = [1.0  .2; .2  1.5];
S( :, :, 3) = [1.0  .4; .4  1.1];
```

```
S( :, :, 4) = [ .3   .2; .2   .5];
n_points = 100*ones(1,4);  % jumlah titik-titik per grup
X1 = [ ];
for i = 1:4
    X1 = [X1; mvnrnd(m(i, :) , S(:, :, i), n_points(i))];
end
X1 = X1';
```

Gambarkan data dengan menuliskan

```
figure(1), plot(X1(1, :), X1(2, :), '.b')
figure(1), axis equal
```

Seperti yang dapat diamati bahwa X1 terdiri dari empat kelas (yang tidak terlalu jelas dipisahkan).

Langkah 1. Untuk memperkirakan jumlah kelas, lakukan proses sebagai berikut:

a. Tentukan jarak maksimum dan minimum antara titik-titik X1 dengan menuliskan:

```
[1,N] = size(X1);
% menentukan matrik jarak
dista = zeros(N,N);
for  i=1:N
    for j=i+1:N
        dista(i, j)=sqrt(sum((X1(:, i) – X1(:, j)) .^2));
       dista(j, i)=dista(i, j);
   end
end
true_maxi = max(max(dista));
true_mini = min(dista( ~logical(eye(N))));
```

b. Tentukan nilai Θmin, Θmax, dan s dengan menuliskan

```
meani = (true_mini+true_maxi)/2;
theta_min = .25*meani;
theta_max =1.75*meani;
n_theta = 50;
s = (theta_max – theta_min) / (n_theta – 1);
```

c. Jalankan n_{times} BSAS untuk semua nilai Θ, setiap waktu dengan urutan data yang berbeda dengan menuliskan

```
q = N;
n_times = 10;
m_tot = [ ];
for theta = theta_min : s : theta_max
        list_m = zeros(1,q);
        for stat = 1:n_times % untuk setiap nilai theta BSAS berjalan n_times kali
                order = randperm(N);
                [bel, m] = BSAS(X1, theta, q, order);
               list_m(size(m,2)) = list_m(size(m,2))+1;
        end
```

```
    [q1, m_size] = max(list_m);
    m_tot = [m_tot   m_size];
end
```

d. Gambarkan grafik mΘ terhadap Θ (lihat Gambar 7.3(a)) dengan menuliskan

```
theta_tot = theta_min :s : theta_max;
figure(2), plot(theta_tot, m_tot)
```

Tentukan perkiraan akhir jumlah kelas dan nilai Θ yang sesuai, sebagaimana penjelasan yang telah lalu, dengan menuliskan

```
% Menentukan jumlah kelas
m_best = 0;
theta_best = 0;
siz = 0;
for i = 1: length(m_tot)
    if (m_tot (i)~ =1)  % mengeluarkan clustering kelas tunggal
        t = m_tot – m_tot(i);
        siz_temp = sum(t==0);
        if (siz < siz_temp)
            siz = siz_temp;
            theta_best = sum(theta_tot.*(t==0)) / sum(t==0);
            m_best = m_tot(i);
        end
    end
end
%  Cek clustering kelas tunggal
if (sum(m_tot == m_best)<.1*n_theta)
    m_best=1;
    theta_best = sum(theta_tot.*(m_tot ==1)) / sum(m_tot ==1);
end
```

Langkah 2. Jalankan algoritma BSAS untuk nilai $\Theta = \Theta_{best}$:

```
order = randperm(N);
[bel, repre] = BSAS(X1, theta_best, q, order);
```

Gambarkan hasilnya (lihat Gambar 7.3(b)), tulislah

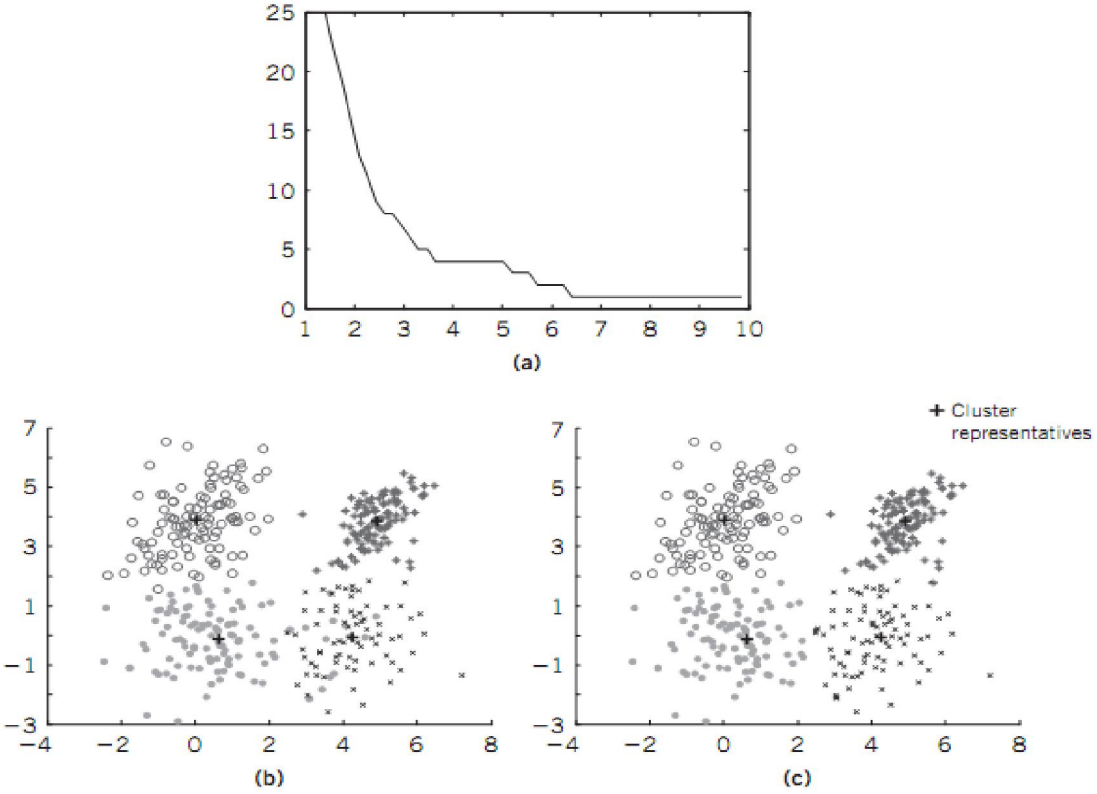
```
figure(11), hold on
figure(11), plot(X1(1,bel ==1), X1(2, bel ==1), ‘ r.’, X1(1,bel ==2), X1(2, bel ==2), ‘g*’ , X1(1, bel ==3), X1(2, bel ==3), ‘bo’, X1(1, bel ==4), X1(2, bel ==4), ‘cx’, X1(1,bel ==5), X1(2, bel ==5), ‘md’ , X1(1, bel ==6), X1(2, bel ==6), ‘yp’, X1(1, bel ==7), X1(2, bel ==7), ‘ks’)
figure(11), plot(repre(1, :) , repre(2, :), ‘k+’)
```

Langkah 3. Jalankan prosedur reassignment, tulislah

```
[bel, new_repre] = reassign(X1, repre, order);
```

Gambarkan hasil yang bekerja seperti pada langkah sebelumnya (lihat Gambar 7.3(c)). Bandingkan hasil yang diperoleh pada langkah-langkah sekarang dan sebelumnya, amati pengaruh reassignment pada hasil.

Latihan 7.4.1 Hasilkan dan gambarkan seperangkat data X_2 yang terdiri dari 300 titik-titik dimensi 2 yang berasal dari distribusi normal dengan nila rerata m1 = $[0,0]^T$ dan matrik kovarian sama dengan matrik identitas 2x2. Ulangi langkah satu pada contoh 7.4.2 dan simpulkan.



Gambar 7.3

Referensi :

T Sergio, dan K Konstantinos, “An Introduction to Pattern Recognition : A Matlab Approach”, Academic Press is an imprint of Elsevier.