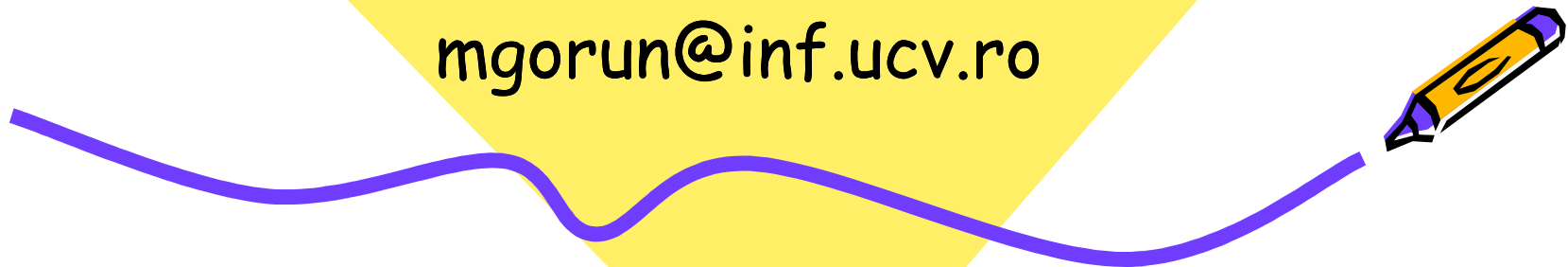




Clustering

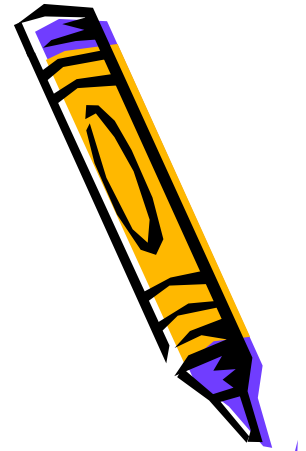
Marina Gorunescu
mgorun@inf.ucv.ro

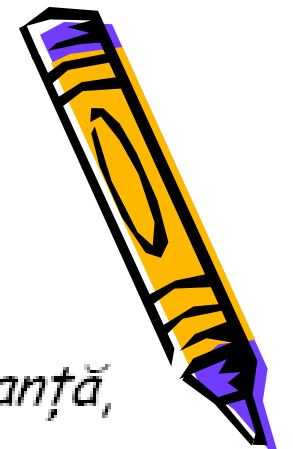


cluster

Clustering - ul este o tehnică de învățare nesupervizată, care determină o structură intrinsecă într-o mulțime de date. Această metodă este un proces de organizarea a obiectelor, care sunt asemănătoare (similare) dintr-un anumit punct de vedere.

Cluster-ul este o mulțime de obiecte asemănătoare între ele și diferite de obiectele aparținând altui cluster.





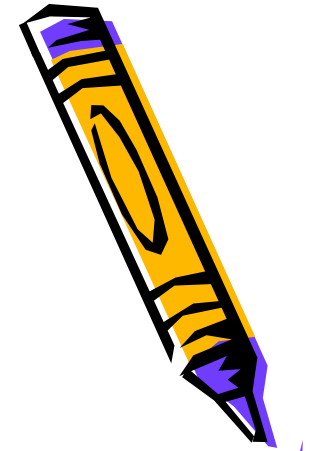
Putem vorbi despre clustering, bazat pe o anumită *distanță*, în sensul că avem următorul criteriu de similaritate:

Două sau mai multe obiecte aparțin aceluiași cluster dacă sunt foarte apropiate, relativ la distanța considerată.

În *clustering- ul conceptual*, obiectele sunt grupate în același cluster dacă au aceeași proprietate, un cluster fiind definit de acea proprietate (concept).



probleme rezolvabile utilizand clustering-ul

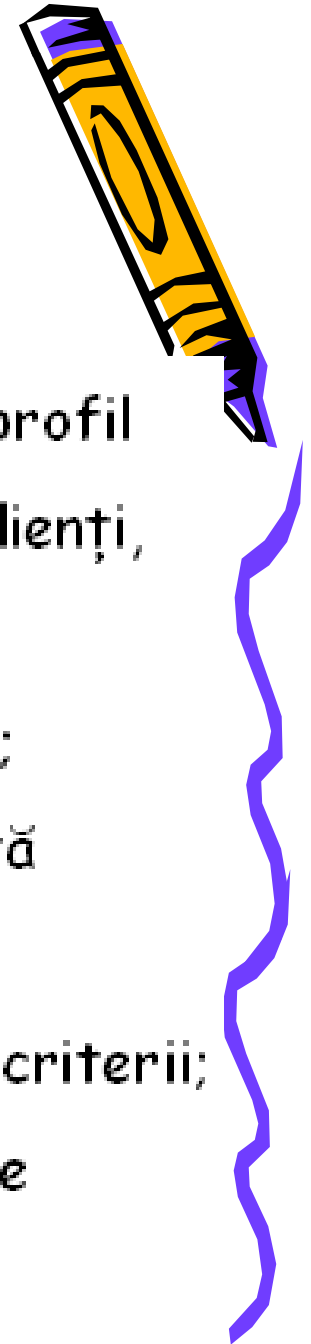


- reducerea datelor (*data reduction*), găsind reprezentanții unui grup omogen;
- determinarea unor clase convenabile („*useful data classes*”);
- determinarea unor clustere naturale și descrierea proprietăților lor necunoscute („*natural data types*”);
- găsirea unor obiecte mai rar întâlnite (*outlier detection*)

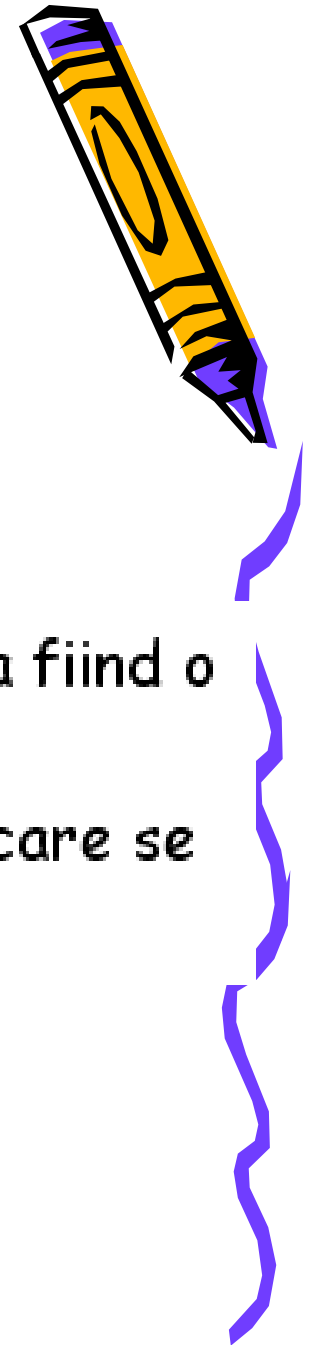


aplicatii

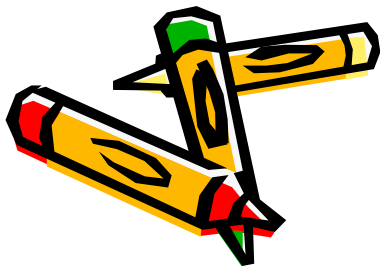
- *marketing*, pentru găsirea grupurilor de clienți cu profil asemănător, pe baza unei baze de date a acestor clienți, cu atributele lor și cumpărăturile lor anterioare;
- *biologie*, pentru clasificarea plantelor și animalelor;
- *asigurări*, pentru identificarea clienților ce prezintă riscuri mari și pentru identificarea fraudelor;
- *biblioteci*, pentru ordonarea cărților după anumite criterii;
- *internet*, pentru împărțirea documentelor în funcție de tematică.



masura de similaritate



Măsura de similaritate poate fi considerată ca fiind o metrică definită pe o anumită mulțime M .
Este aleasă în concordanță cu tipul de date cu care se lucrează și cu scopul propus.





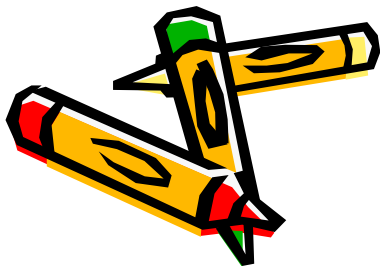
O bază de date formată din m vectori linie, n -dimensionali

x_1, \dots, x_m , unde $x_k = (x_1^k, \dots, x_n^k)$ poate fi reprezentată ca fiind

o matrice cu m linii și n coloane

Vom defini mai multe metrici (distanțe) între vectorii x_k

și x_l :

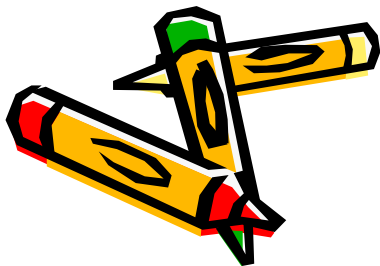
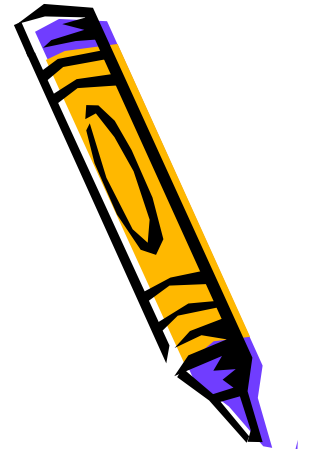


distanta euclidiană

- distanta euclidiană



$$d(\mathbf{x}_k, \mathbf{x}_l) = \sqrt{(\mathbf{x}_k - \mathbf{x}_l) \cdot (\mathbf{x}_k - \mathbf{x}_l)'} = \sqrt{\sum_{i=1}^n (x_i^k - x_i^l)^2}$$



distanta euclidiană standardizată

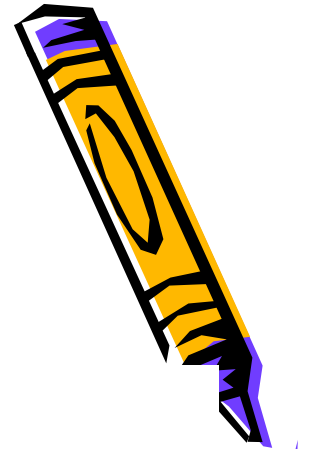
- distanța euclidiană standardizată

$$d_s(x_k, x_l) = \sqrt{(x_k - x_l) \cdot D^{-1} \cdot (x_k - x_l)'}$$

unde D este matricea diagonală ce are ca elemente

dispersia vectorului $X_i = (x_i^1, \dots, x_i^m)$, notată d_i^2 , adică:

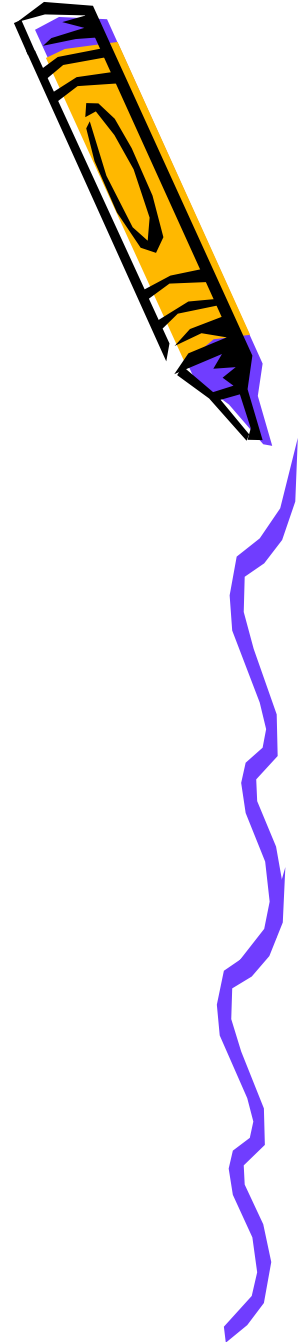
$$d_s(x_k, x_l) = \sqrt{\sum_{i=1}^n \left(\frac{x_i^k - x_i^l}{d_i} \right)^2}$$



distanta city block

- distanta city block (Manhattan)

$$d_c(x_k, x_l) = \sum_{i=1}^n |x_i^k - x_i^l|$$



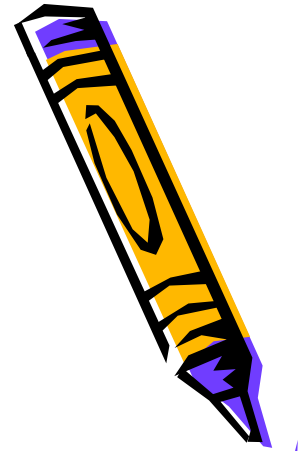
distanța Minkowski

- distanța Minkowski

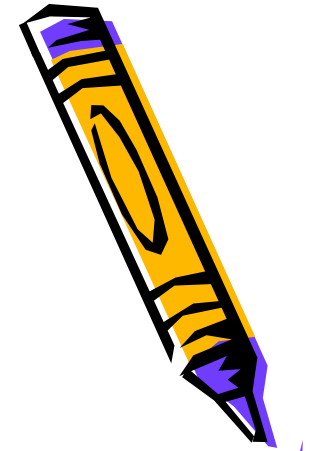
$$d_M(x_k, x_l) = \left(\sum_{i=1}^n |x_i^k - x_i^l|^p \right)^{\frac{1}{p}}$$

Se observă că:

- dacă $p = 1$ obținem distanța *Manhattan*;
- dacă $p = 2$ avem distanța *euclidiană*.



distanta Minkowski ponderata



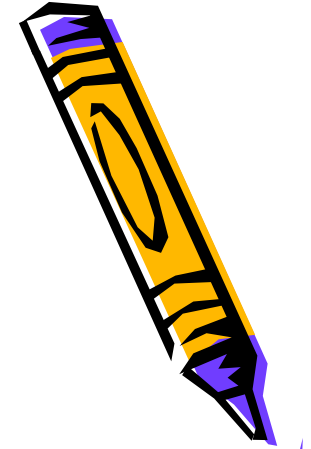
- Pentru ierarhizarea fiecărui atribut în funcție de scopul propus, pot fi incorporate ponderi și astfel distanța *Minkowski ponderată* este de forma:

$$d_{\alpha}(x_k, x_l) = \left(\sum_{i=1}^n \alpha_i \cdot |x_i^k - x_i^l|^p \right)^{\frac{1}{p}}, \quad \alpha_i > 0, 1 \leq i \leq n,$$

$$\sum_{i=1}^n \alpha_i = 1, \text{ unde } \alpha_i \text{ reprezintă ponderea atributului } i.$$



distanța coeficientului de corelație



- distanța *Pearson's r* (distanța coeficientului de corelație):

$$r(X_k, X_1) = 1 - \frac{(X_k - \bar{X}_k) \cdot (X_1 - \bar{X}_1)'}{\sqrt{(X_k - \bar{X}_k) \cdot (X_k - \bar{X}_k)'} \cdot \sqrt{(X_1 - \bar{X}_1) \cdot (X_1 - \bar{X}_1)'}}$$

unde $\bar{X}_k = \frac{1}{n} \sum_{i=1}^n x_i^k$

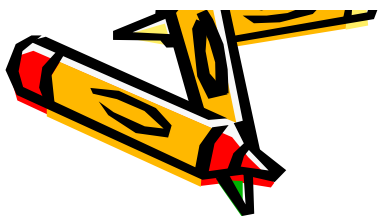
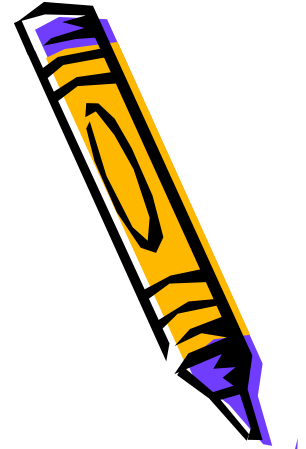


distanta Mahalanobis

- Considerăm o mulțime de n vectori priviți ca un eșantion de dimensiune n a n variabile aleatoare independente.

Pentru fiecare pereche de vectori \mathbf{x}_k și \mathbf{x}_l , covarianța lor este definită de formula:

$$\text{cov}(\mathbf{x}_k, \mathbf{x}_l) = \frac{1}{n} \cdot \sum_{i=1}^n x_i^k \cdot x_i^l - \bar{\mathbf{x}}_k \cdot \bar{\mathbf{x}}_l$$





Notând cu $\text{cov}(D)$ matricea de covarianță corespunzătoare eșantionului, măsura *Mahalanobis* este dată de formula:

$$D_M(\mathbf{x}_k, \mathbf{x}_1) = \sqrt{(\mathbf{x}_k - \mathbf{x}_1) \cdot \text{cov}(D)^{-1} \cdot (\mathbf{x}_k - \mathbf{x}_1)'}$$

Dacă variabilele eșantionului nu sunt corelate, distanța Mahalanobis se rezumă la distanța euclidiană.



masuri fuzzy



- Măsurile *fuzzy* sunt utilizate pentru compararea de vectori ale căror componente iau valori în $[0,1]$.

Pentru vectorul $\mathbf{x} = (x_1, x_2, \dots, x_n)$, $x_i \in [0,1]$, valoarea lui x_j reprezintă gradul în care vectorul \mathbf{x} posedă al j -lea atribut (caracteristică).

În scopul definirii măsurilor de similaritate *fuzzy* definim:

$$s(x_i, y_i) = \max\{\min\{x_i, y_i\}, \min\{1 - x_i, 1 - y_i\}\}$$

plecând de la cazul clasic, avem măsura *fuzzy Minkowski*:

$$d_F(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n s(x_i, y_i)^p \right)^{\frac{1}{p}}.$$

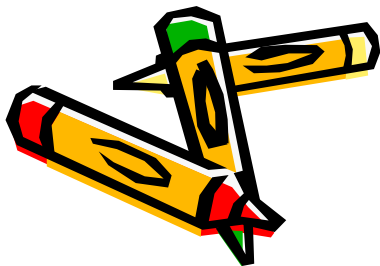


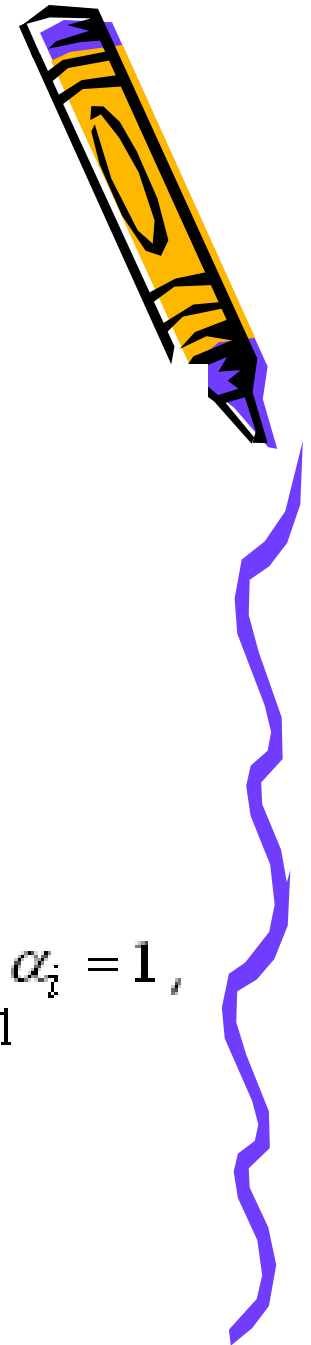
masura de similaritate mixta si ponderata



Obiectele au ca reprezentare un vector în care componentele au semnificații diferite, având naturi diferite: numerice, nominale, fuzzy etc.

Vom defini o măsură de similaritate *mixtă și ponderată*, pentru cuantificarea tipului de date, respectiv semnificația atributelor.





Avem de comparat două obiecte:

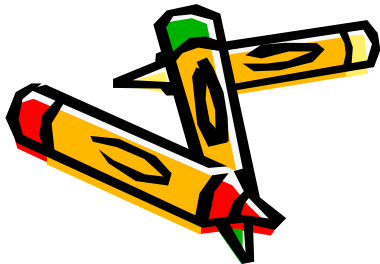
$$\mathbf{x} = ((x_1^1, \dots, x_{k_1}^1), \dots, (x_1^p, \dots, x_{k_p}^p))$$

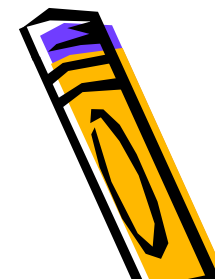
$$\mathbf{y} = ((y_1^1, \dots, y_{k_1}^1), \dots, (y_1^p, \dots, y_{k_p}^p))$$

care au p tipuri de date de dimensiuni k_1, \dots, k_p .

-vom pondera cele p secvențe cu valorile $\alpha_i > 0, \sum_{i=1}^p \alpha_i = 1,$

în funcție de importanța lor în context;





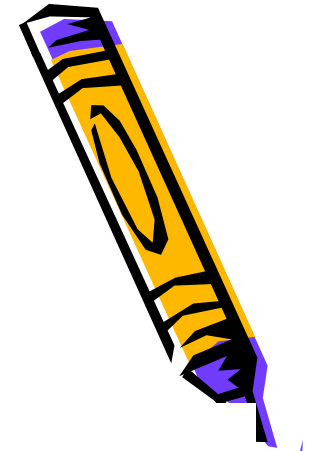
- vom aplica fiecărui cuplu de secvențe $x_j = (x_1^j, \dots, x_{k_j}^j)$ și $y_j = (y_1^j, \dots, y_{k_j}^j)$ măsura de similaritate specifică $d_j, 1 \leq j \leq p$.

Astfel:

$$d(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^p \alpha_j \cdot d_j(x_j, y_j)$$



standardizarea caracteristicilor



Standardizarea caracteristicilor este o problemă ce o avem de rezolvat înainte de a calcula măsura de similaritate între obiecte.

Considerăm obiectele x_1, \dots, x_n , unde $x_k = (x_1^k, \dots, x_p^k)$, care pot fi reprezentate sub forma unei matrice X :

$$X = \begin{pmatrix} x_1^1 & \dots & x_i^1 & \dots & x_p^1 \\ \dots & \dots & \dots & \dots & \dots \\ x_1^k & \dots & x_i^k & \dots & x_p^k \\ \dots & \dots & \dots & \dots & \dots \\ x_1^n & \dots & x_i^n & \dots & x_p^n \end{pmatrix}$$





Pentru standardizare putem folosi transformările:

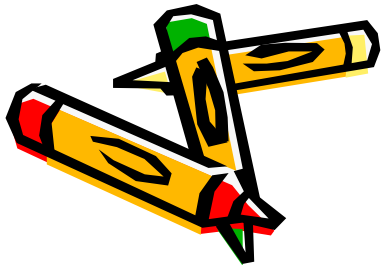
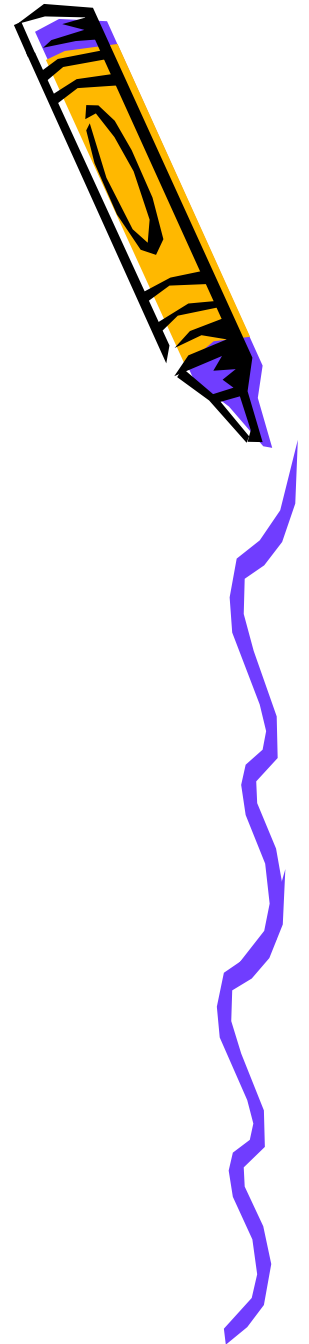
$$- x_i^k \rightarrow \frac{x_i^k - \bar{x}^i}{SD(x^i)}, \text{ unde } x^i = \begin{pmatrix} x_i^1 \\ \dots \\ x_i^n \end{pmatrix}, 1 \leq k \leq n, 1 \leq i \leq p$$

$$- x_i^k \rightarrow \frac{x_i^k - \bar{x}_k}{SD(x_k)}, \text{ unde } x_k = (x_1^k, \dots, x_p^k), 1 \leq k \leq n, 1 \leq i \leq p.$$

Pentru fiecare vector, media este zero și dispersia egală cu unitatea.



Metode de clustering neierarhic

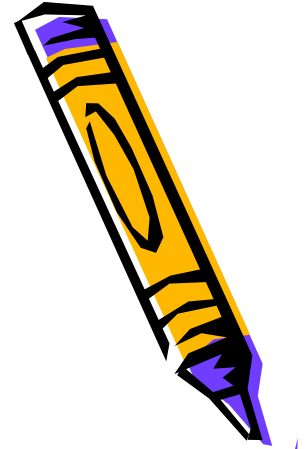


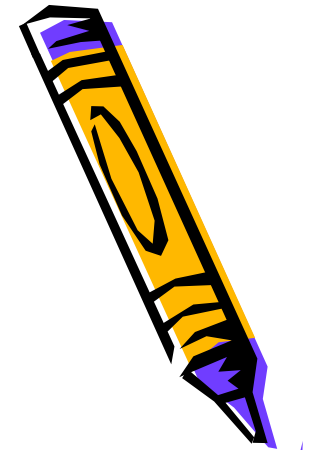
K-means

Algoritmul *k-means* (Mac Queen, 1967) clasifică o mulțime de n obiecte într-un număr k dat *a priori* de clusterare.

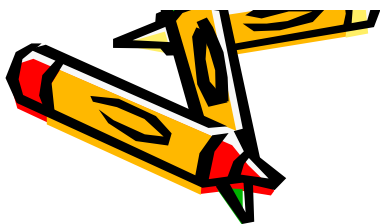
- alegem aleator k puncte în spațiul definit de obiectele din baza de date, ce urmează a fi clusterizate. Acestea sunt *centroizii* inițiali.


De obicei încercăm să îi plasăm cât se poate de departe unul de altul.





- asociem fiecare obiect din mulțimea de date celui mai apropiat centroid (în sensul distanței considerate), obținând astfel un prim grupaj.
- calculăm centrele de greutate ale clusterelor obținute, definind astfel noii centroizi.
- reluăm procedeul și astfel cei k centroizi își vor schimba locul pas cu pas, până ajung în poziția dorită.





În final, algoritmul urmărește minimizarea *funcției*

obiectiv $J = \sum_{j=1}^k \sum_{i=1}^n \left\| x_i^{(j)} - c_j \right\|^2$ unde $\left\| x_i^{(j)} - c_j \right\|^2$ este distanța

aleasă între punctul $x_i^{(j)}$ din baza de date și centroidul c_j al clusterului C_j .

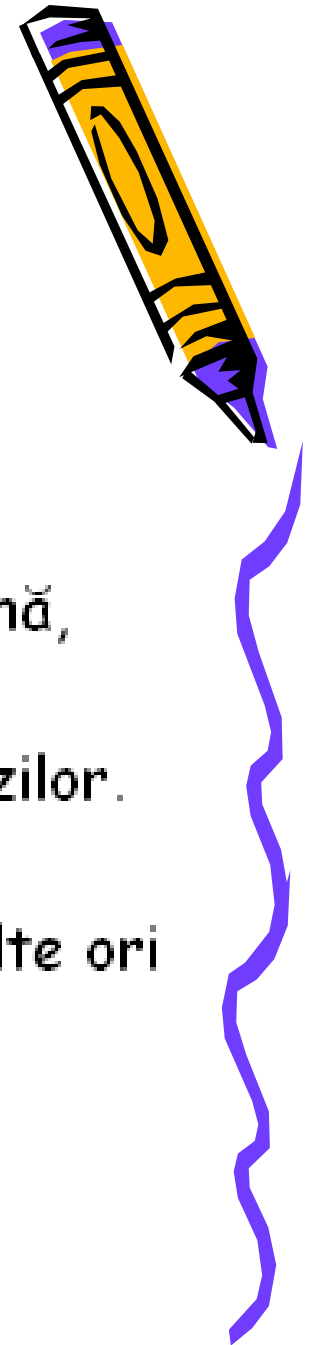
Funcția obiectiv, care este în acest caz funcția erorii pătratice, este un indicator al distanței dintre cele n puncte din baza de date și centrele clusterelor corespunzătoare.



probleme

- algoritmul nu găsește întotdeauna configurația optimă, corespunzătoare minimului global al funcției obiectiv.
- algoritmul este sensibil la alegerea inițială a centroizilor.

Pentru reducerea acestor efecte se rulează de mai multe ori algoritmul.



exemplu

aplicație referitoare la cancerul hepatic:

considerăm un lot de 17 pacienți dintre care 6 au cancer hepatic (HCC).

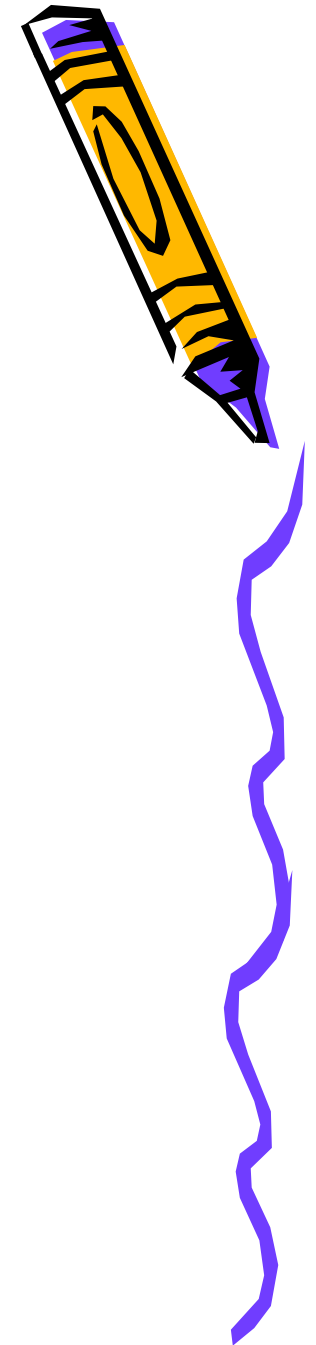
Pentru a decide dacă un pacient are sau nu cancer hepatic, studiile clinice arată că se iau în considerare anumite enzime serice (de obicei un număr de 15).

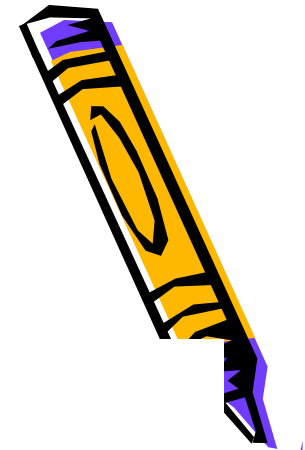
Pentru simplificarea calculelor, le vom lua în considerare pe cele mai importante din punct de vedere clinic:

alkaline phosphatase (FA), g-glutamyl transferase (gGT), leucine amino peptidase (LAP) și colesterol (C)



	FA	gCT	LAP	C	diagnostic
P1	162	255	74	258	non HCC
P2	210	324	93	220	non HCC
P3	259	208	115	266	non HCC
P4	120	114	89	171	non HCC
P5	246	173	98	210	non HCC
P6	138	189	76	165	non HCC
P7	132	177	48	138	non HCC
P8	152	183	105	178	non HCC
P9	186	220	140	148	non HCC
P10	180	119	114	171	non HCC
P11	236	270	88	150	non HCC
P12	422	488	183	292	HCC
P13	607	259	65	275	HCC
P14	446	283	176	309	HCC
P15	460	1053	145	221	HCC
P16	680	381	280	275	HCC
P17	561	450	180	164	HCC





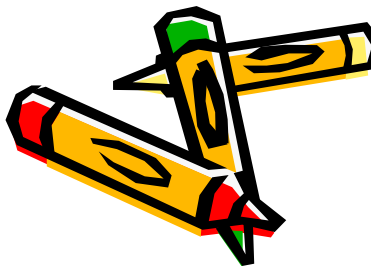
Alegem aleator doi centroizi:

$C1(150,100,100,100)$ și $C2(400,400,200,200)$;

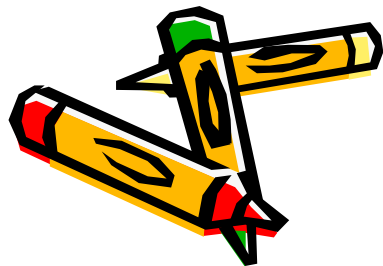
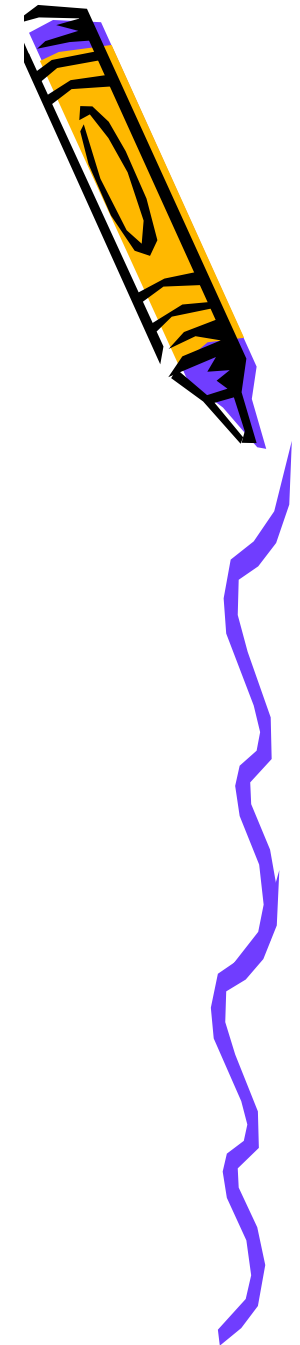
calculăm distanțele dintre aceștia și punctele P_i
(pacienții din baza de date), împărțindu-le în
două submulțimi după regula:

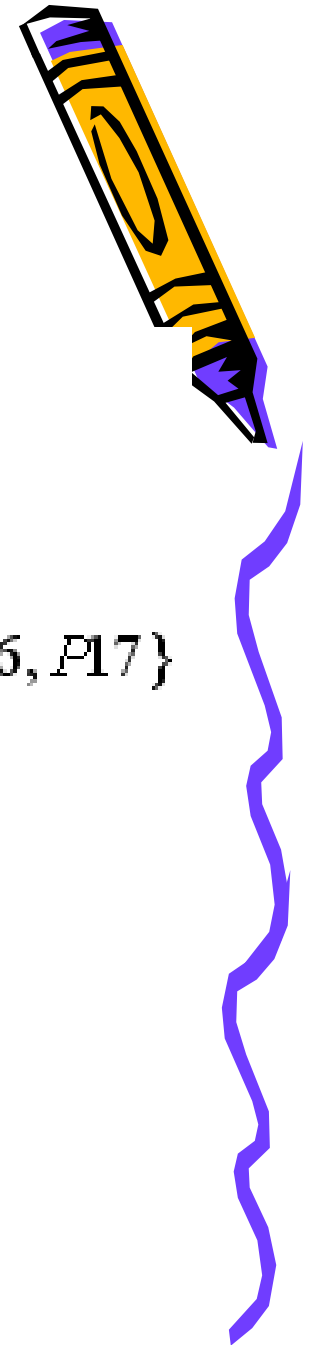
„dacă $d(C1, P_i) < d(C2, P_i)$ pacientul P_i aparține
mulțimii $M1$;

dacă $d(C2, P_i) < d(C1, P_i)$ pacientul P_i aparține
mulțimii $M2$.”



	$d(C1, P_i)$	$d(C2, P_i)$	mulțimea
P1	231.3	153.9	M2
P2	276.9	165.3	M2
P3	254.4	122.9	M2
P4	75.8	164.1	M1
P5	196.8	115.5	M2
P6	119	143.4	M1
P7	105.3	179.1	M1
P8	125.3	110	M2
P9	160.3	83	M2
P10	109.5	123.2	M1
P11	223.6	145.7	M2
P12	545.8	375.4	M2
P13	560.5	439.2	M2
P14	450.1	282.5	M2
P15	1026.9	893.6	M2
P16	691.6	524.5	M2
P17	587.8	441	M2





Am obținut mulțimile

$$M1 = \{P4, P6, P7, P10\}$$

și

$$M2 = \{P1, P2, P3, P5, P8, P9, P11, P12, P13, P14, P15, P16, P17\}$$

Calculăm centroizii acestor mulțimi:

$$C_1 (142.5, 149.7, 81.7, 161.2)$$

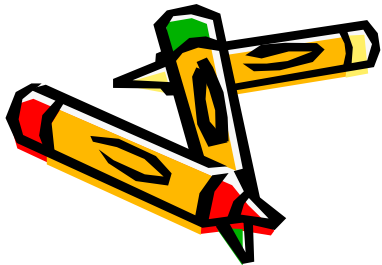
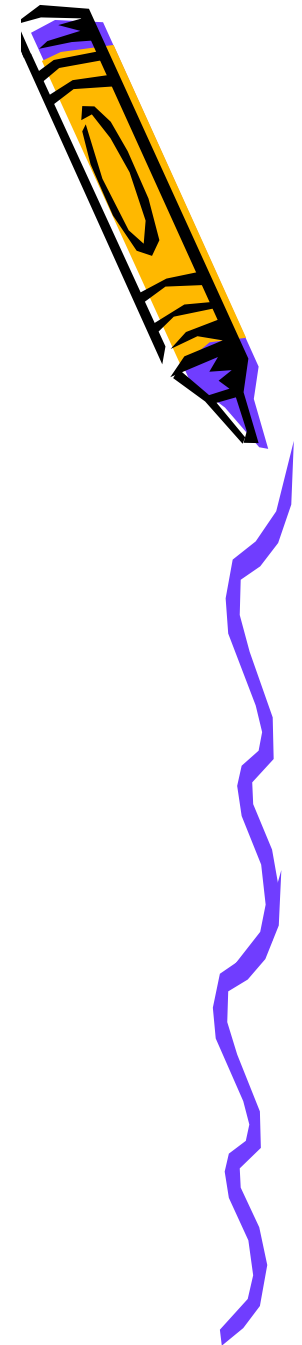
$$C_2 (355.9, 349.7, 134, 228.1)$$

și distanțele dintre vectori (pacienți) și aceștia.



#

	$d(C_1, P_i)$	$d(C_2, P_i)$	multimea
P1	144.5	225.9	M'_1
P2	196.2	153.9	M'_2
P3	170.4	176.8	M'_1
P4	43.9	341.3	M'_1
P5	117.9	211.9	M'_1
P6	40.1	283.9	M'_1
P7	50.2	308.9	M'_1
P8	44.9	296.6	M'_1
P9	102	228	M'_1
P10	59	296.3	M'_1
P11	152.9	170.1	M'_1
P12	468.9	173.1	M'_2
P13	490.8	279.7	M'_2
P14	374.9	144.4	M'_2
P15	961.4	711.1	M'_2
P16	628.2	359.9	M'_2
P17	524.3	241.1	M'_2





Calculăm centroizii mulțimilor M_1 și M_2

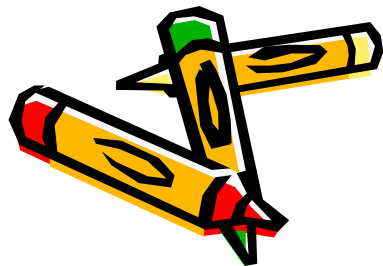
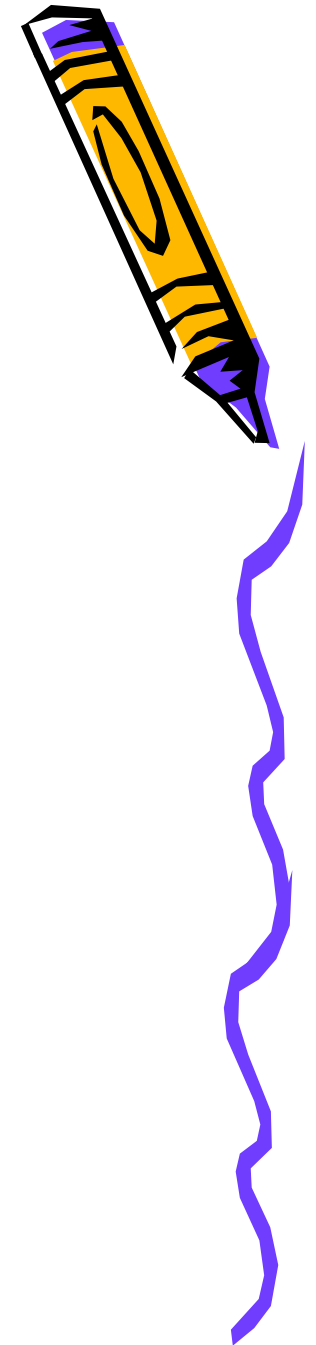
$C_1' (181.1, 190.8, 94.7, 185.5)$

$C_2' (485, 462.5, 160.2, 250.8)$



¶

	$d(C'_1, P_i)$	$d(C'_2, P_i)$	mulțimea
P1	100.8	393.5	$M_1^{\#}$
P2	140.6	316.6	$M_1^{\#}$
P3	115.1	343.6	$M_1^{\#}$
P4	99.3	515.8	$M_1^{\#}$
P5	71.6	382.7	$M_1^{\#}$
P6	51.2	457.8	$M_1^{\#}$
P7	83.8	481	$M_1^{\#}$
P8	32.7	444.2	$M_1^{\#}$
P9	65.8	398.9	$M_1^{\#}$
P10	75.7	468.5	$M_1^{\#}$
P11	102.9	338.2	$M_1^{\#}$
P12	406.8	82.6	$M_2^{\#}$
P13	441.5	286.7	$M_2^{\#}$
P14	317	193.3	$M_2^{\#}$
P15	908.2	591.9	$M_2^{\#}$
P16	572.2	244.1	$M_2^{\#}$
P17	468.2	117.7	$M_2^{\#}$





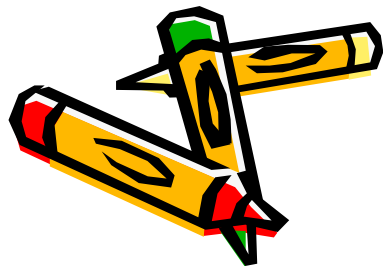
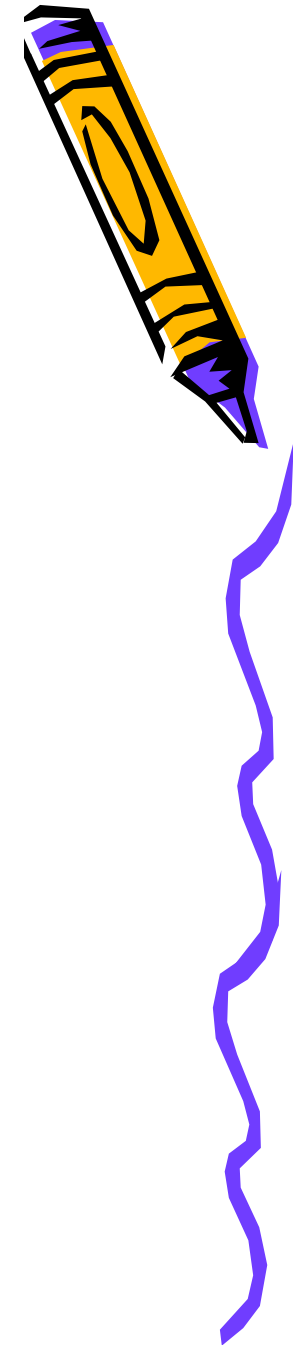
Continuând procedeul, centroizii mulțimilor obținute sunt:

$$C_1^{\#} (183.7, 202.9, 94.5, 188.6)$$

$$C_2^{\#} (355.9, 349.7, 134, 228.1).$$



	$d(C_1^#, P_i)$	$d(C_2^#, P_i)$	mulțimea
P1	91.7	445.7	$M_1^{'''}$
P2	127.8	369.4	$M_1^{'''}$
P3	110	392.7	$M_1^{'''}$
P4	110.9	566.4	$M_1^{'''}$
P5	72.4	431.6	$M_1^{'''}$
P6	56.3	532.4	$M_1^{'''}$
P7	89.9	532.4	$M_1^{'''}$
P8	40.2	495.5	$M_1^{'''}$
P9	63.3	449.5	$M_1^{'''}$
P10	87.9	517.6	$M_1^{'''}$
P11	93.6	389.3	$M_1^{'''}$
P12	395.7	115.2	$M_2^{'''}$
P13	436.6	282.4	$M_2^{'''}$
P14	310.4	225.9	$M_2^{'''}$
P15	895.8	573.4	$M_2^{'''}$
P16	565.6	212.9	$M_2^{'''}$
P17	459.7	103.5	$M_2^{'''}$

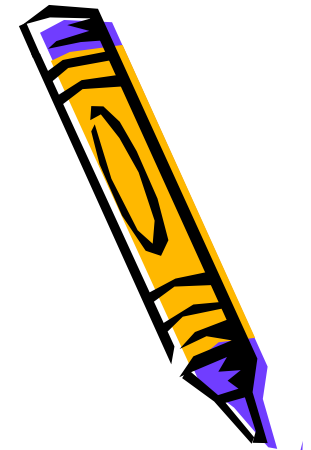




Am obținut aceleași mulțimi, ca în tabelul precedent, ceea ce ne indică faptul că procesul s-a încheiat. Acest fapt constituie condiția de stop a algoritmului.

Am împărțit mulțimea de obiecte în două cluster, și testând cu mulțimea noastră de date, rezultatul este optim.





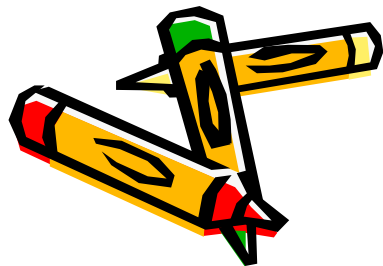
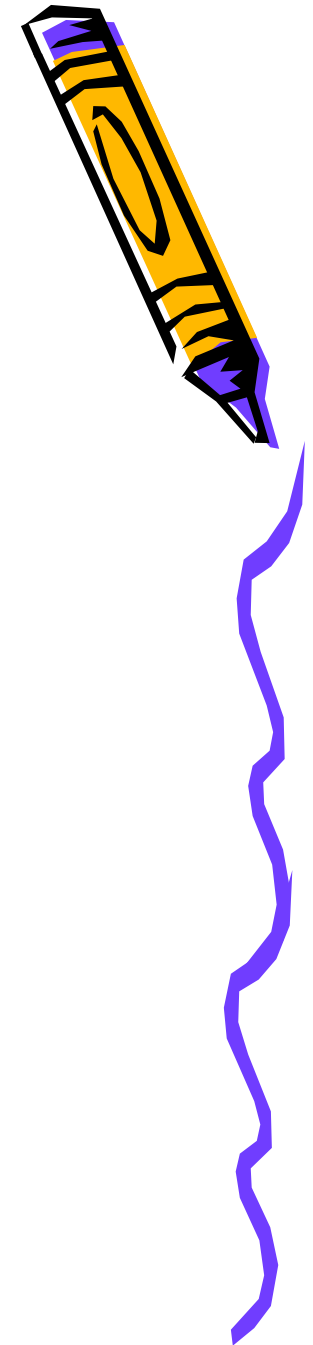
Algoritmul este sensibil la alegerea centroizilor inițiali.
în funcție de date
alegem următorii centroizi:

$C1(150,100,100,100)$ și $C2(400,400,200,200)$

și obținem un rezultat asemănător cu rezultatul
din cel de-al doilea tabel.



	$d(C1, P_i)$	$d(C2, P_i)$	mulțimea
P1	231.3	311.3	M1
P2	276.9	231.7	M2
P3	254.4	261.3	M1
P4	75.8	416.3	M1
P5	196.8	292.8	M1
P6	119	360.2	M1
P7	105.3	385.3	M1
P8	125.3	343.6	M1
P9	160.3	290.6	M1
P10	109.5	368.2	M1
P11	223.6	242.5	M1
P12	545.8	130.3	M2
P13	560.5	294.2	M2
P14	450.1	168.3	M2
P15	1026.9	658.3	M2
P16	691.6	301.3	M2
P17	587.8	173.5	M2

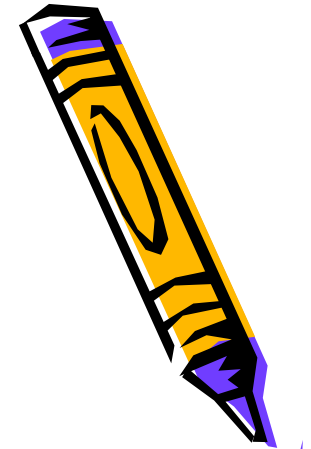




Metoda *k-means* nu este aplicabilă la date ne-numerice, deoarece definirea mediei devine o problemă.



metoda euristica de clusterizare rapida



Prezentăm o metodă euristică de clusterizare rapidă care dă rezultate bune, dar nu optime.

Avem o mulțime de obiecte, care urmează a fi împărțită în k cluster:

- se începe cu un singur cluster al cărui centroid este media obiectelor din baza de date.
- se calculează distanțele dintre obiectele din baza de date și centroid și stabilind euristic o valoare -treshhold, se împarte clusterul în două submulțimi.





- calculăm mediile obiectelor din cele două submulțimi și obținem doi centroizi.
- în funcție de distanțele dintre obiectele considerate și cei doi centroizi, obținem două cluster.
- calculăm centroizii celor două cluster și obținem eventual o altă grupare a obiectelor, prin determinarea distanțelor dintre acestea și acești centroizi.
- această parte a procesului se oprește când nu mai apar modificări în componența clusterelor.





- se continuă procedeul împărțind fiecare cluster în două, până obținem numărul de clustere cerut.

Dacă numărul k nu este o putere a lui 2, ne oprim la cea mai mică putere a lui 2 mai mare decât k și renunțăm la clusterelor ce sunt mai puțin importante.

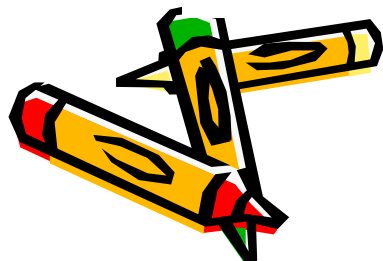


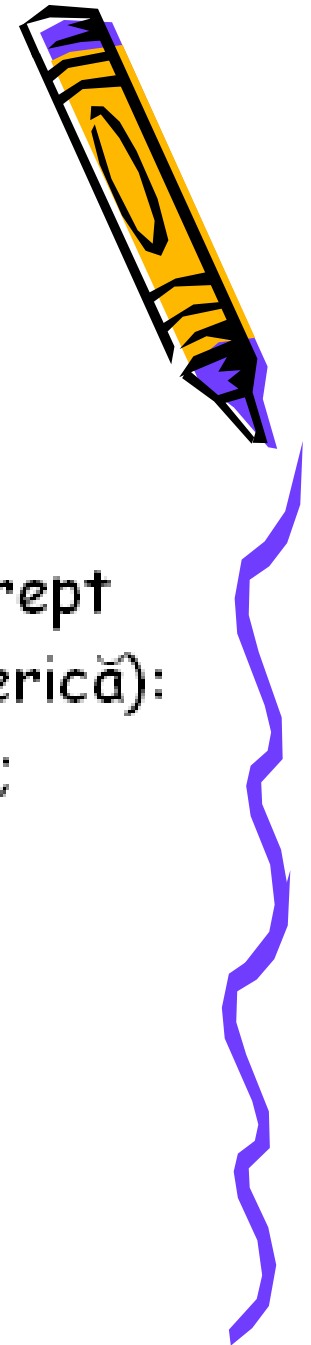


Reluăm exemplul prezentat anterior: aplicația cu determinarea bolnavilor cu cancer hepatic, folosind cele 4 enzime serice.

Avem două clase:

- HCC = bolnavi cu cancer hepatic
- Non-HCC = bolnavi cu alte afecțiuni hepatice.





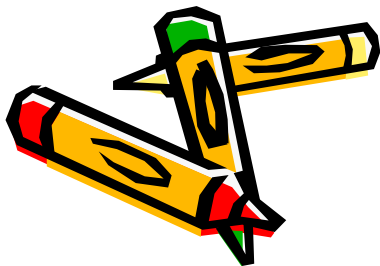
Pentru început vom calcula centroidul ce are drept componente mediile fiecărui atribut (enzimă serică):

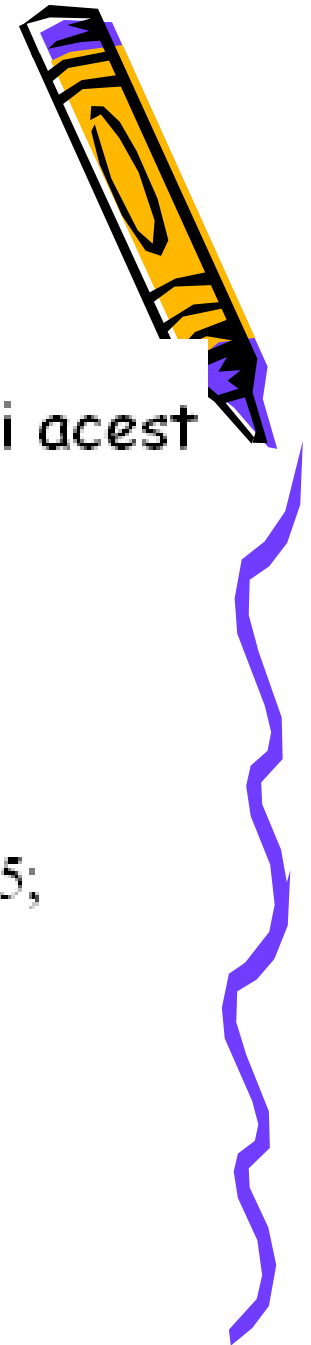
$$c1 = \text{mean}(FA) = 305.7; \quad c2 = \text{mean}(gGT) = 214.3;$$

$$c3 = \text{mean}(LAP) = 97.2; \quad c4 = \text{mean}(C) = 193.2.$$

centroidul inițial este punctul de coordonate:

$$C(195.8, 214.3, 97.2, 193.2).$$





Calculăm distanța dintre fiecare vector (pacient) și acest centroid, notând $d_i = d(C, P_i)$:

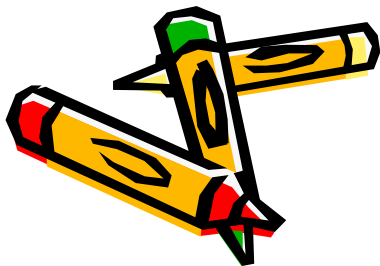
$$d_1 = 165.1; \quad d_2 = 102.4; \quad d_3 = 118.6; \quad d_4 = 269.9;$$

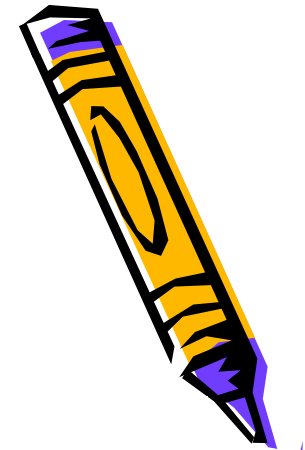
$$d_5 = 144.7; \quad d_6 = 213; \quad d_7 = 238.6; \quad d_8 = 198.5;$$

$$d_9 = 160.1; \quad d_{10} = 226.5; \quad d_{11} = 104.6; \quad d_{12} = 240.7;$$

$$d_{13} = 315.8; \quad d_{14} = 763.7; \quad d_{15} = 766.4; \quad d_{16} = 418.5;$$

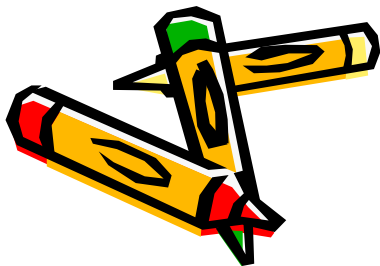
$$d_{17} = 304.3$$





Vom împărți lotul de pacienți în două submulțimi:

- $M1$: pacienții pentru care distanța d_i este mai mică decât 200 (aleasă euristic) $\{P1, P2, P3, P5, P8, P9, P11\}$
- $M2$: restul pacienților.





Calculăm centrele de greutate a celor două mulțimi:

$C1(257.9, 244.9, 104.8, 200.4)$

$C2(528.6, 572.3, 200.3, 268.3)$.

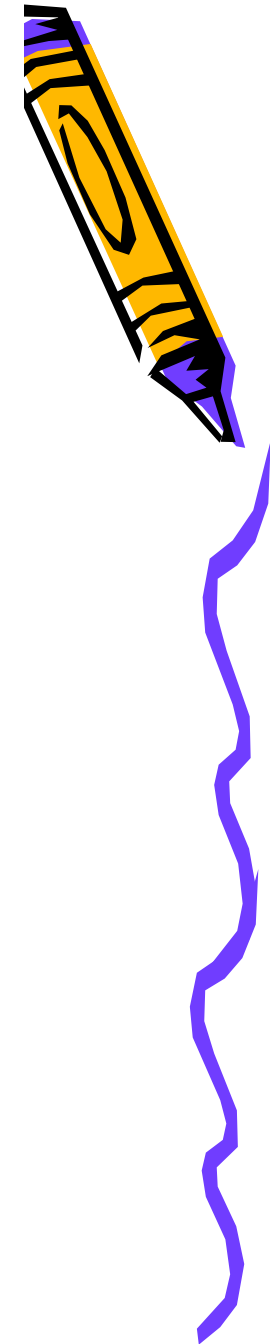
Evaluăm distanțele dintre P_i și centroizii găsiți:

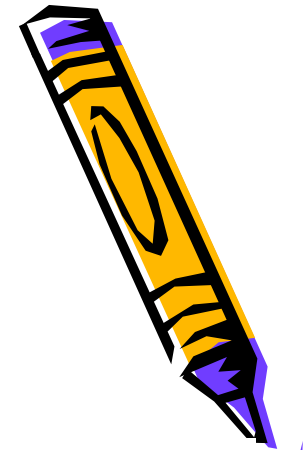
dacă $d(C1, P_i) < d(C2, P_i)$ pacientul P_i va aparține clusterului de centru $C1$.



⊕

	$d(C1, P_i)$	$d(C2, P_i)$	cluster
P1	116.4	501.1	M'_1
P2	95.2	420.7	M'_1
P3	75.9	461.1	M'_1
P4	193.04	631.5	M'_1
P5	73.8	503.1	M'_1
P6	139.9	570.6	M'_1
P7	166.1	594.7	M'_1
P8	124.7	557.3	M'_1
P9	98.8	509.5	M'_1
P10	151.2	586.4	M'_1
P11	62.7	451.2	M'_1
P12	317.1	139	M'_2
P13	359.4	350.2	M'_2
P14	231.7	304.5	M'_1
P15	834.2	490.9	M'_2
P16	482.6	256.7	M'_2
P17	375.3	165.2	M'_2





Am obținut mulțimile:

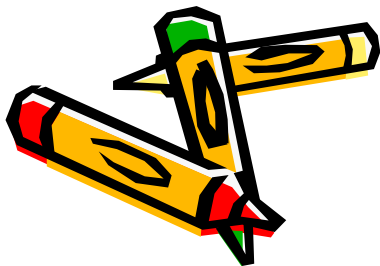
$$M'_1 = \{P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P14\}$$

$$M'_2 = \{P12, P13, P15, P16, P17\}$$

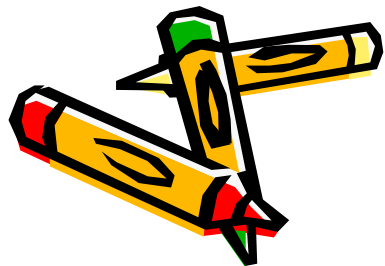
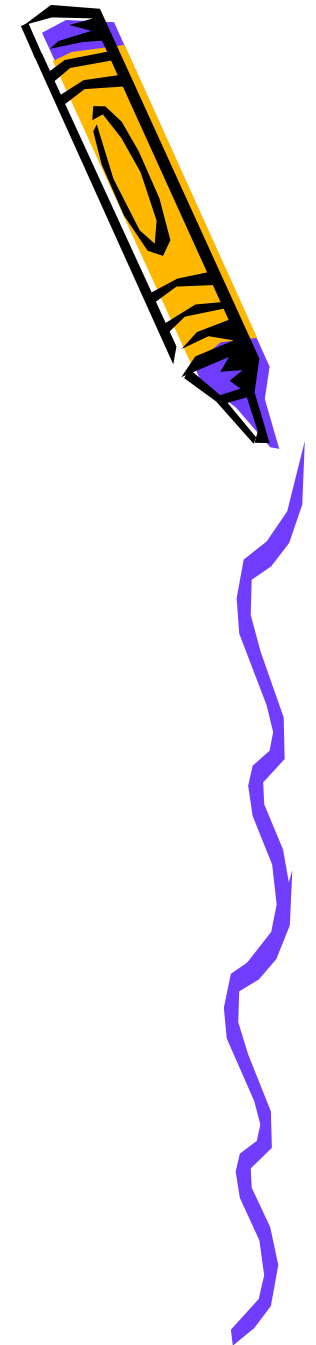
Centroizii acestor mulțimi sunt:

$$C'1 = (205.5, 209.5, 101.3, 198.6)$$

$$C'2 = (546, 526.2, 170.6, 245.4)$$



	$d(C'1, P_i)$	$d(C'2, P_i)$	cluster
P1	90.9	480	M'_1
P2	115.7	400.5	M'_1
P3	87	432.5	M'_1
P4	131.8	602.9	M'_1
P5	55.7	470.3	M'_1
P6	82.2	543.6	M'_1
P7	112	565.5	M'_1
P8	63.3	530.9	M'_1
P9	67.4	483.5	M'_1
P10	98.9	555.4	M'_1
P11	84.3	421.5	M'_1
P12	373.8	138.4	M'_2
P13	413.1	295.2	M'_2
P14	284.4	270.5	M'_2
P15	882.3	534.9	M'_2
P16	540.5	237.7	M'_2
P17	437.6	112.8	M'_2





Avem mulțimile:

$$M_1^n = \{P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11\}$$

$$M_2^n = \{P12, P13, P15, P16, P17\},$$

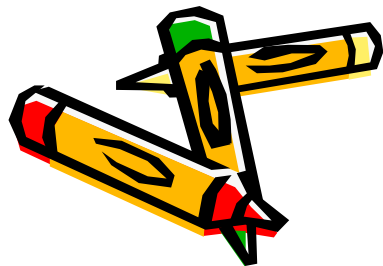
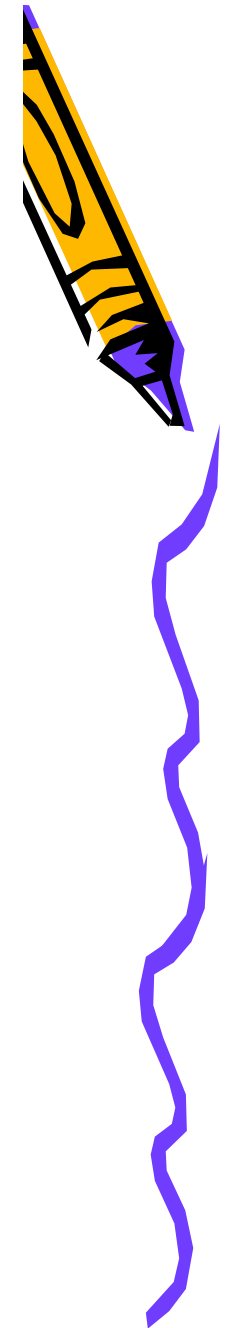
cu centroizii:

$$C^1(183.7, 202.9, 94.5, 188.6)$$

$$C^2(529.3, 485.6, 171.5, 256)$$



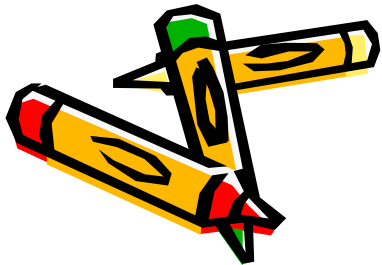
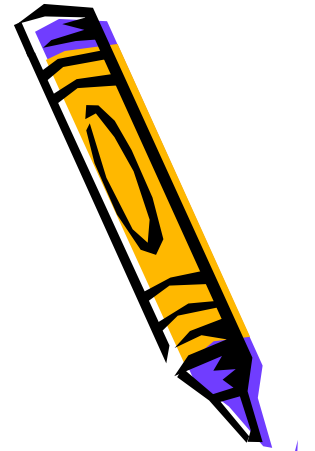
	$d(C^*1, P_i)$	$d(C^*2, P_i)$	cluster
P1	91.7	444.5	M'_1
P2	127.8	368.1	M'_1
P3	109.9	391.7	M'_1
P4	110.9	565.4	M'_1
P5	72.3	430.7	M'_1
P6	56.4	508.4	M'_1
P7	89.8	531.3	M'_1
P8	40.3	494.4	M'_1
P9	63.3	448.4	M'_1
P10	88	516.7	M'_1
P11	93.6	388.2	M'_1
P12	395.6	113.8	M'_2
P13	436.6	262.8	M'_2
P14	310.3	225.4	M'_2
P15	895.8	573.2	M'_2
P16	565.5	213.9	M'_2
P17	459.6	103.9	M'_2



Am obținut aceleași cluster, ceea ce înseamnă încheierea procesului.



Clustering ierarhic

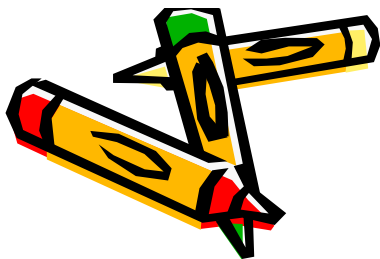




Metoda de *clustering ierarhic* constă în gruparea obiectelor în mod iterativ, utilizând o anumită legătură - *linkage*.

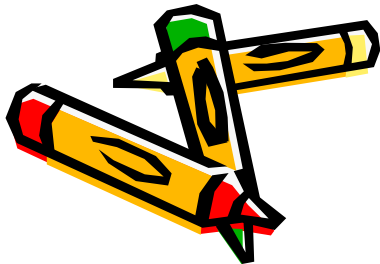
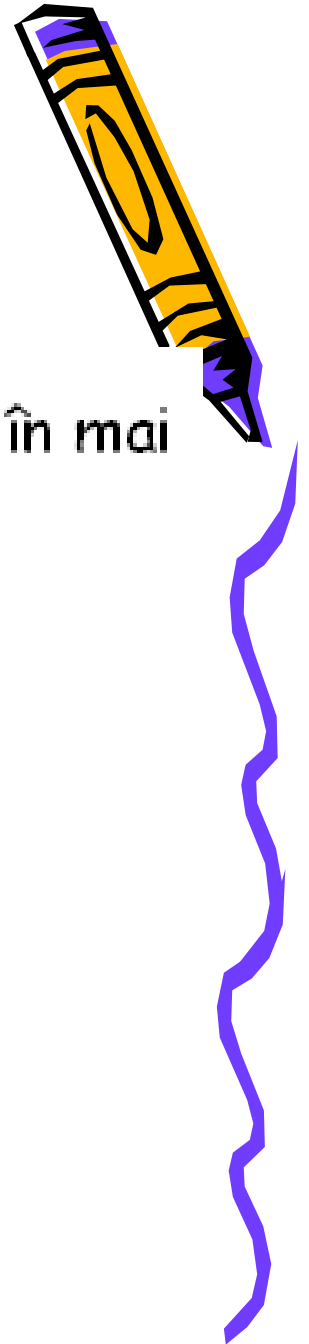
Există două tipuri de metode:

- *aglomerativă*, care constă într-o serie de fuziuni a celor n obiecte în grupuri (cea mai utilizată)
- *de divizare*, care separă succesiv cele n obiecte în grupuri mai fine.



Putem defini distanța dintre două clustere în mai multe moduri:

- *Single linkage*
- *Complete linkage*
- *Average linkage*
- *Centroid linkage*
- *Ward linkage*

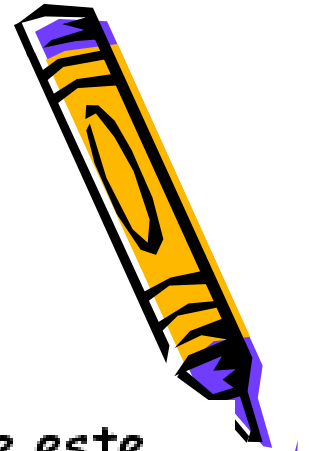
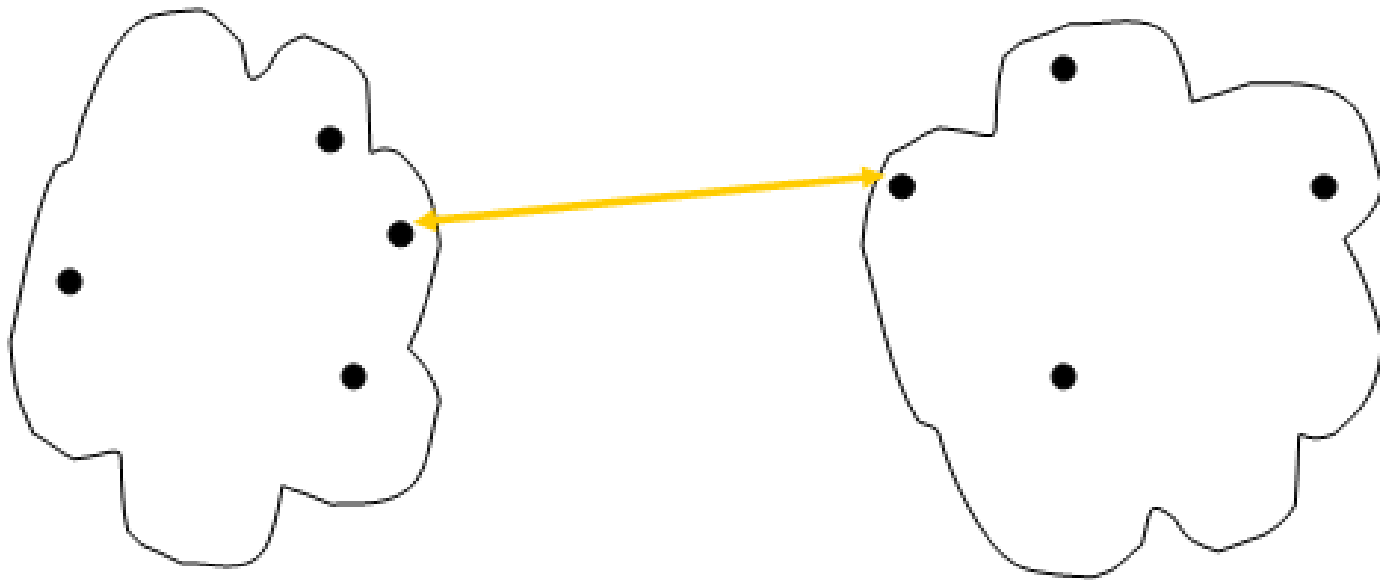


single linkage

- *Single linkage*, caz în care distanța dintre clusterere este determinată de distanța între cele mai apropiate obiecte:

$$d(A, B) = \min\{d(x_k, y_j), x_k \in A, y_j \in B\},$$

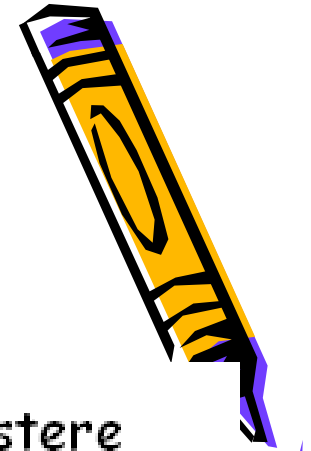
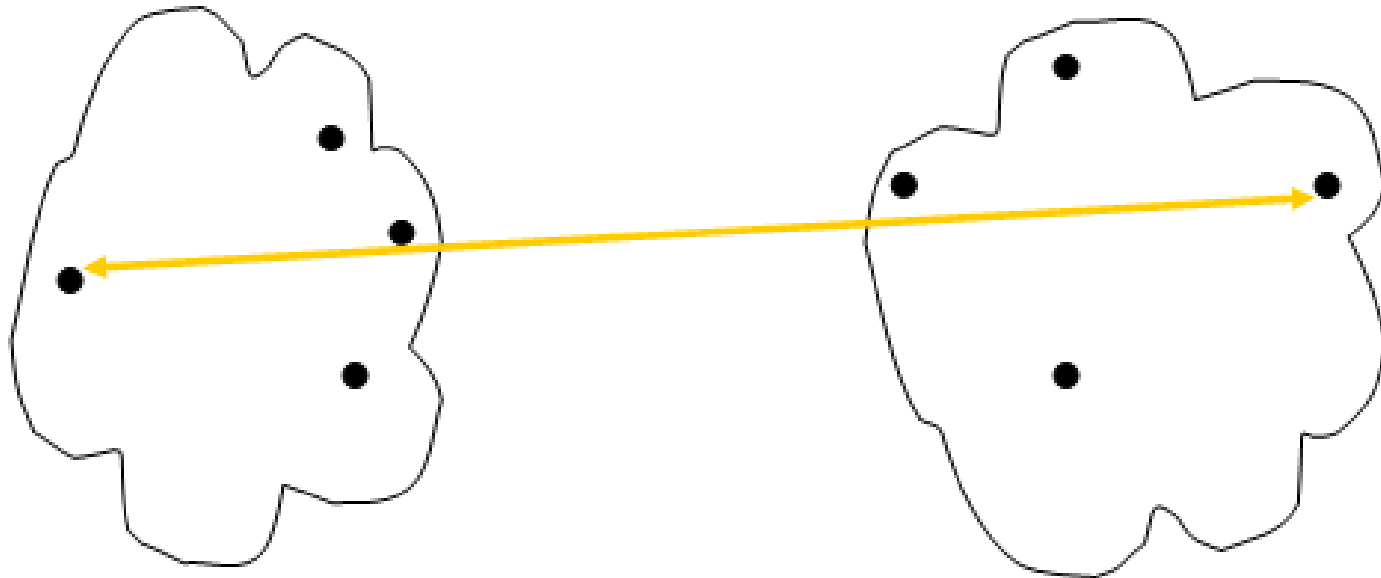
unde am notat cu A și B cele două clusterere



complete linkage

- *Complete linkage*, caz în care distanța dintre clustere este determinată de distanța între cele mai îndepărtate obiecte:

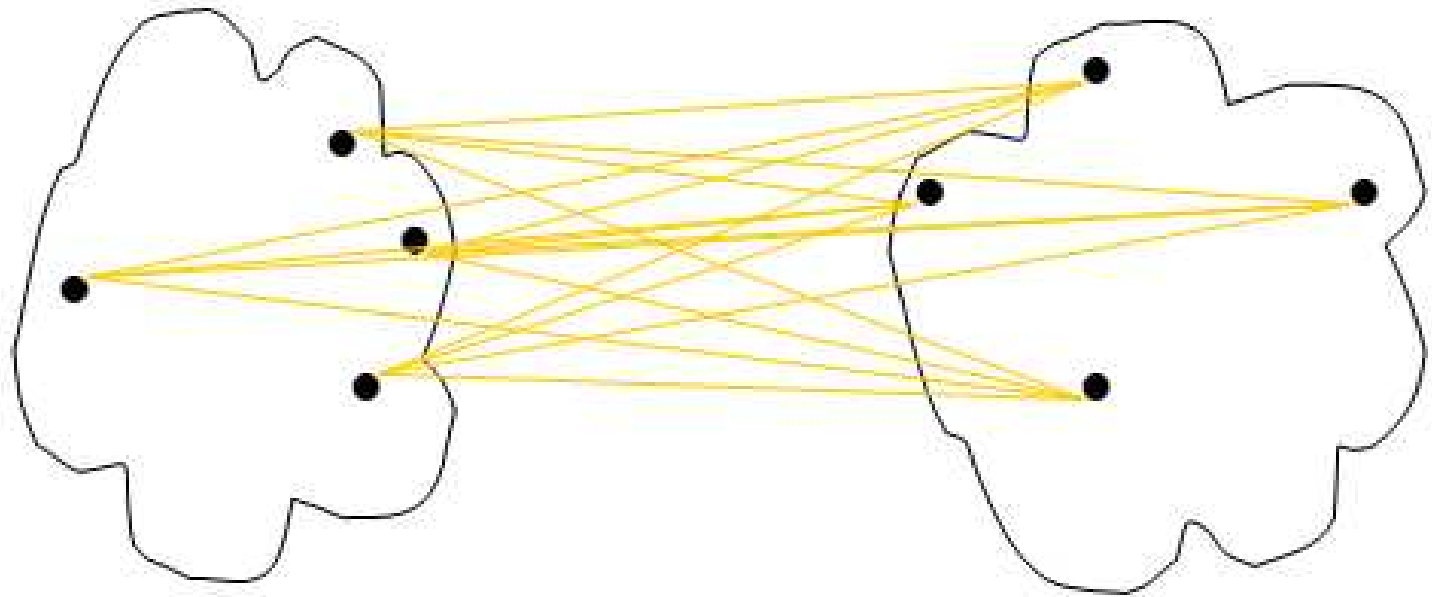
$$d(A, B) = \max\{d(x_k, y_j), x_k \in A, y_j \in B\}$$



average linkage

- *average linkage*, caz în care distanța este calculată

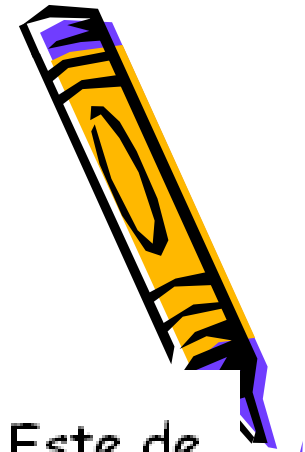
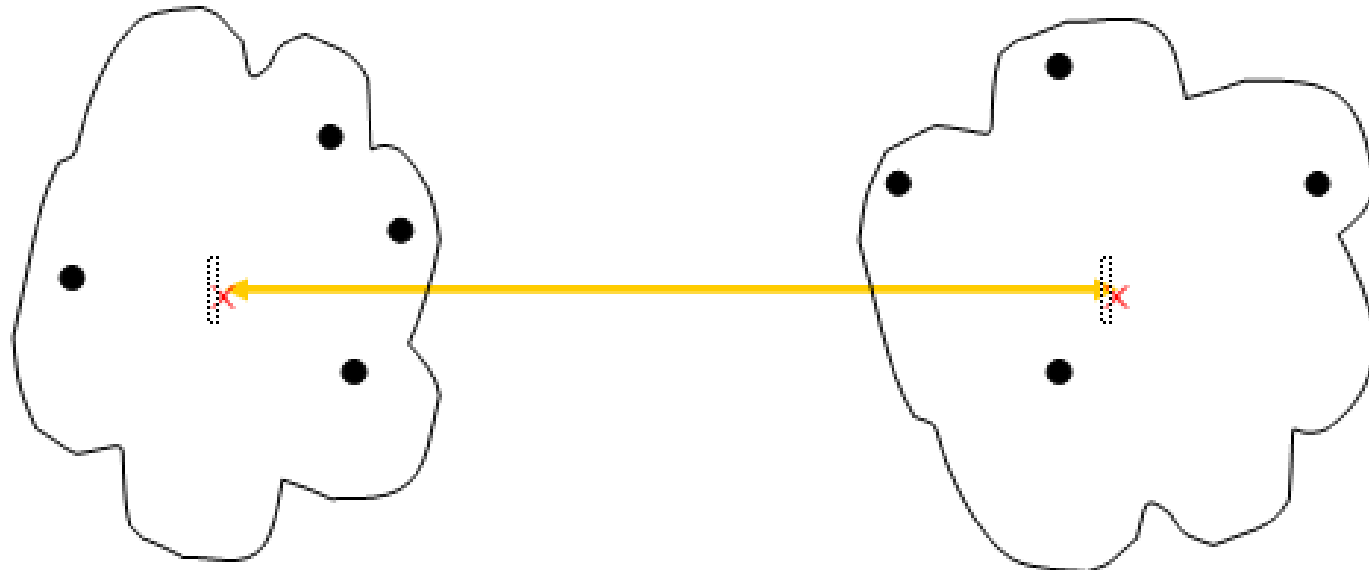
$$\text{conform formulei: } d(A, B) = \frac{\sum_{i=1}^n \sum_{j=1}^m d(x_i, y_j)}{m \cdot n},$$



centroid linkage

- *centroid linkage* este distanța dintre centroizi. Este de menționat faptul că aceasta poate fi folosită doar dacă se utilizează distanța euclidiană.

$$d_{cen}(A, B) = d(\bar{x}_A, \bar{x}_B) \text{ unde } \bar{x}_A = \frac{1}{n} \sum_{i=1}^n x_A^i .$$



ward linkage

Suma pătratelor distanțelor din interiorul unui cluster este definită ca fiind suma pătratelor distanțelor dintre obiectele existente în cluster și centroidul acestuia.

Ward linkage se definește prin formula:

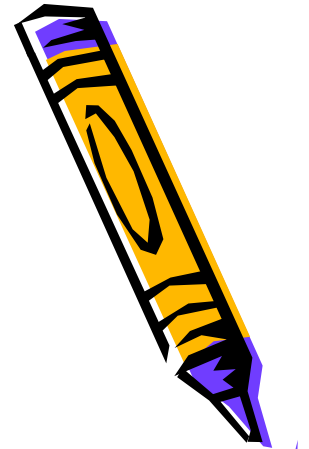
$$d_W(A, B) = \frac{n \cdot m \cdot d_{cen}(A, B)^2}{n + m}$$

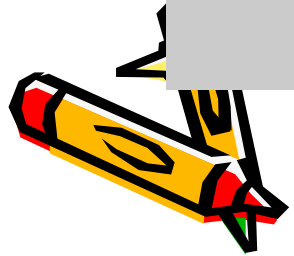
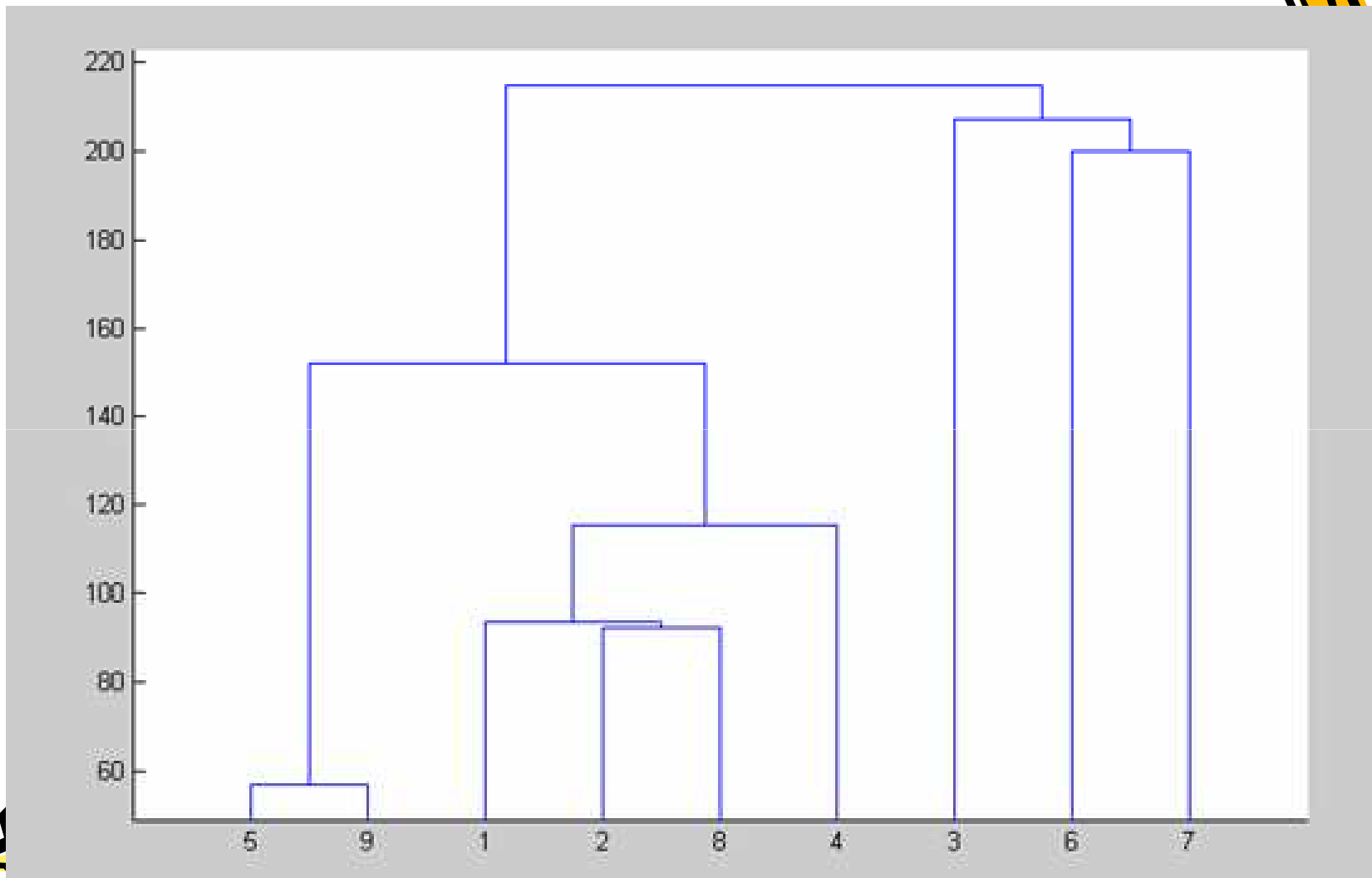
unde clusterul A conține n obiecte și clusterul B conține m obiecte



dendrograma

Clustering-ul ierarhic permite o reprezentare printr-o diagramă bidimensională, numită *dendrogramă*, care ilustrează fuziunile sau divizările din fiecare pas al procesului efectuat.

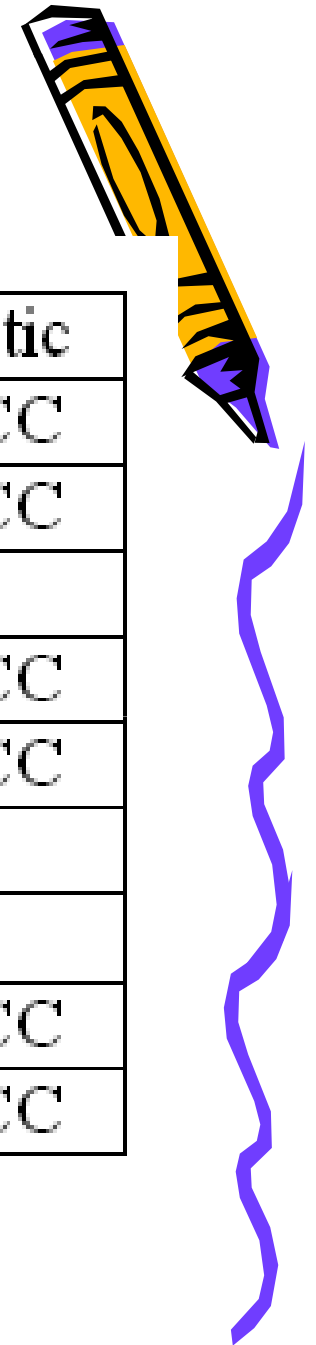




exemplu

Vom prezenta o aplicație referitoare la cancerul hepatic.:
Considerăm un lot de pacienți dintre care 3 au cancer hepatic (HCC). Pentru a decide dacă un pacient are sau nu cancer hepatic, studiile clinice arată că se iau în considerare anumite enzime serice (de obicei un număr de 15).
Pentru simplificarea calculelor, le vom lua în considerare pe cele mai importante din punct de vedere clinic:
alkaline phosphatase (FA), g-glutamyl transferase (gGT),
leucine amino peptidase (LAP) și colesterol (C)





	FA	gCT	LAP	C	diagnostic
P1	162	255	74	258	non HCC
P2	210	324	93	220	non HCC
P3	422	488	183	292	HCC
P4	259	208	115	266	non HCC
P5	129	114	89	171	non HCC
P6	607	259	65	275	HCC
P7	446	283	176	309	HCC
P8	236	270	88	150	non HCC
P9	180	119	114	171	non HCC



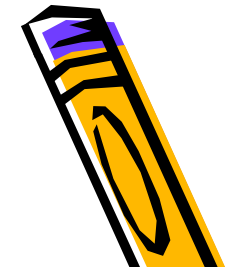


Avem 9 clustere pentru început.

Calculăm distanțele între vectorii standardizați și vom alcătui o matrice, matricea distanțelor, unde elementul $a_{ij} = d(P_i, P_j)$.

Vom determina astfel vectorii cei mai apropiați, în sensul distanței, pe care îi vom grupa în clustere:





0	94.2	367.3	115.6	169.6	445,4	307.3	132.5	162.3
94.2	0	291.7	135.8	230.4	407	268.7	92.3 *	211.9
367.3	291.7	0	322	499.2	317.6	207,2	333.6	460.2
115.6	135.8	322	0	188.2	355.3	214.8	136.2	146.2
169.6	230.4	499.2	188.2	0	510.8	394.5	190.3	57.3 *
445,4	407	317.6	355.3	510.8	0	199.9 *	392.3	461.7
307.3	268.7	207,2	214.8	394.5	199.9 *	0	278	343.7
132.5	92.3 *	333.6	136.2	190.3	392.3	278	0	166
162.3	211.9	460.2	146.2	57.3 *	461.7	343.7	166	0





Alăturând pacienții 2&8, 5&9, 6&7, obținem 6 cluster
(deocamdată pacienții 1,3 și 4 formează cluster
separate).



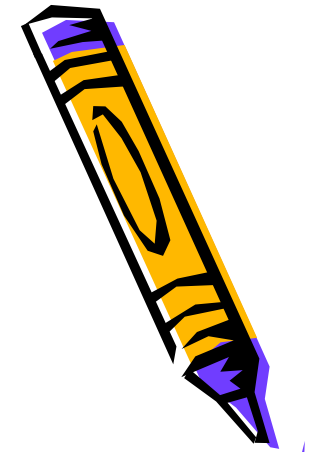


Avem următorii vectori:

- $P1(162, 255, 74, 258)$;
- $P3(422, 488, 183, 292)$;
- $P4(259, 208, 115, 266)$;
- $C1(223, 297, 90.5, 185)$ corespunzător clusterului 2&8;
- $C2(154.5, 116.5, 101.5, 176)$ corespunzător clusterului 5&9;
- $C3(526.5, 271, 120.5, 292)$, corespunzător clusterului 6&7

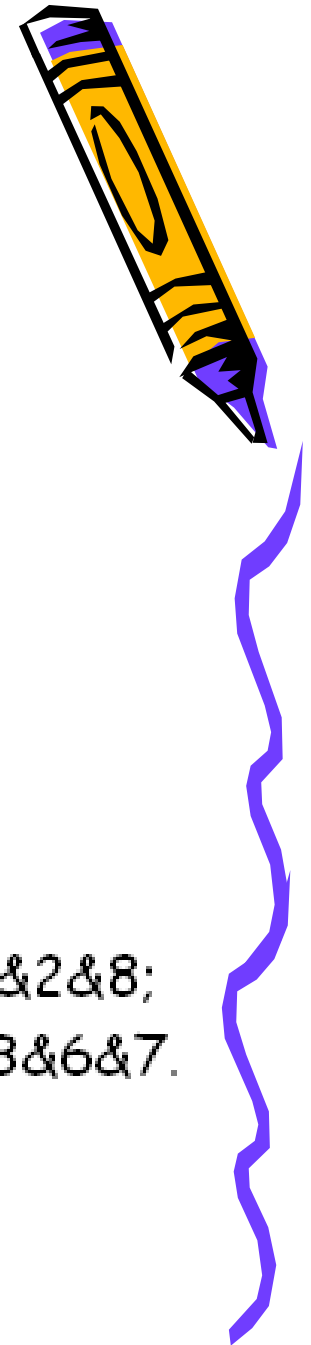
Prezentăm matricea distanțelor dintre acești vectori:





0	367.3	115.6	105.2 *	163.5	369.4
367.3	0	322	310	479.2	248.8 *
115.6	322	0	128	166	276.1
105.2 *	310	128	0	193.6	324.2
163.5	479.2	166	193.6	0	419.6
369.4	248.8 *	276.1	324.2	419.6	0

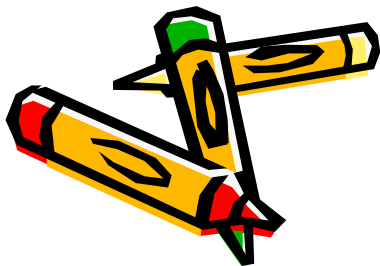




Formăm astfel noi clustere: 1&2&8, 3&6&7, 5&9 și
pacientul 4

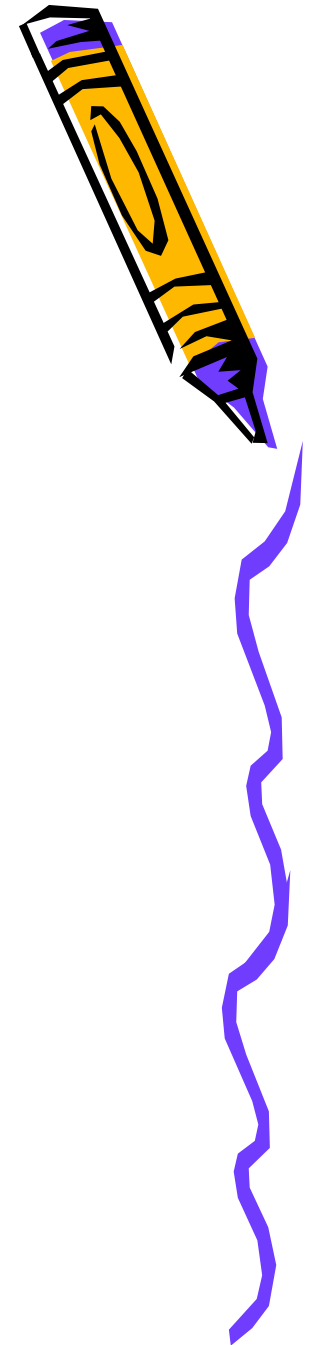
Continuăm procedeul cu următorii vectori:

- P4(259, 208, 115, 266);
- C2(154.5, 116.5, 101.5, 176) ;
- C4(192.5 276. 82.2 221.5) corespunzător clusterului 1&2&8;
- C5 (474.2 379.5 151.7 292) corespunzător clusterului 3&6&7.



Calculăm matricea distanțelor:

$$\begin{pmatrix} 0 & 219.4 & 52.2^* & 285.5 \\ 219.4 & 0 & 171.2 & 432.9 \\ 52.2^* & 171.2 & 0 & 316 \\ 285.5 & 432.9 & 316 & 0 \end{pmatrix}$$





$$\begin{pmatrix} 0 & 432.8 & 159.4^* \\ 432.8 & 0 & 292.9 \\ 159.4^* & 292.9 & 0 \end{pmatrix}$$

și astfel obținem cele două cluster, corespunzătoare celor două boli hepatice

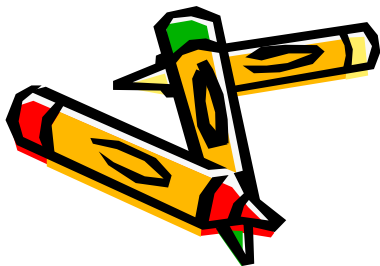
(1&2&4&5&8&9) și (3&6&7)





Testând cu mulțimea de date, observăm că rezultatul este validat 100%.

În final, este necesară aprecierea validității modelului, lucru realizabil prin repetarea procesului folosind alte măsuri de similaritate și eventual alte tehnici de clustering.



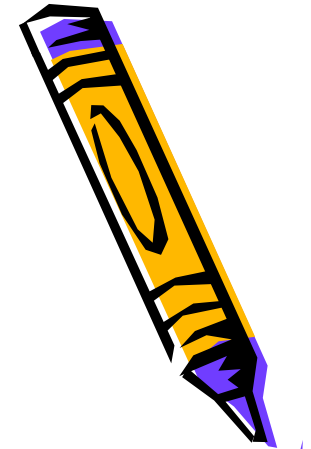


Clusteringul ierarhic este o metodă care studiază modul de grupare a obiectelor dintr-o mulțime, prin crearea unui cluster tree (arbore de cluster).

Acest arbore nu este o mulțime de clustere, ci o ierarhie pe mai multe nivele în care clusterelor de pe un anumit nivel se unesc cu alte clustere la nivelul superior următor, fapt ce ne permite să decidem care nivel de clusterizare este corespunzător aplicației noastre.



clustering ierarhic in MATLAB



Prezentăm rezolvarea acestei probleme în MATLAB:

Pentru a face analiza clusterului ierarhic pe o baza de date, vom folosi următoarea procedură:

1) găsim similaritatea și disimilaritatea între fiecare pereche de obiecte din baza de date.

Calculăm distanța între obiecte folosind funcția `pdist`.

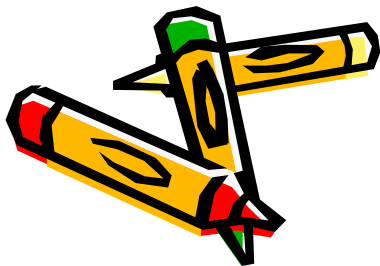


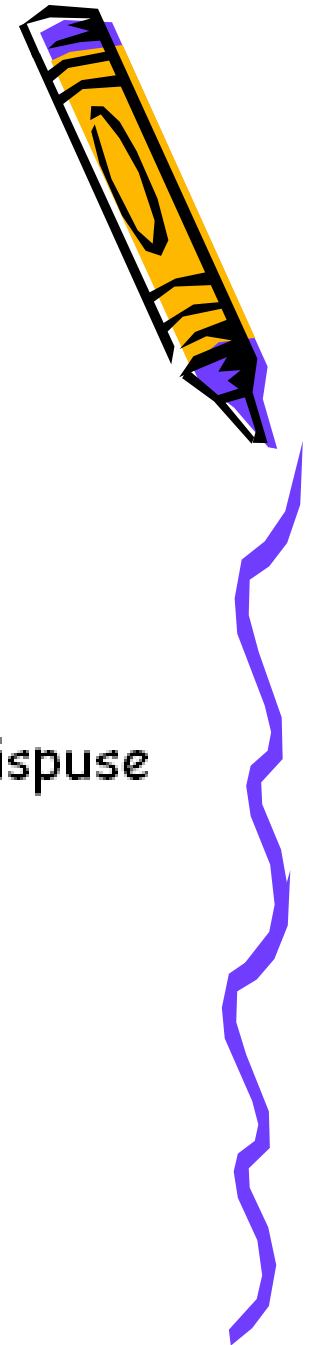
$$Y = \text{pdist}(X)$$

Folosind aceasta funcție, calculăm distanța dintre oricare două puncte din baza de date.

Intr-o mulțime cu m obiecte avem $m(m-1)/2$ de perechi de obiecte.

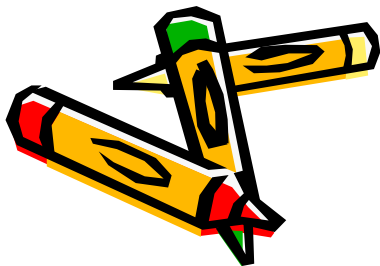
Rezultatul acestor calcule este matricea disimilarităților





$Y = \text{pdist}(X)$ calculează distanța euclidiană.

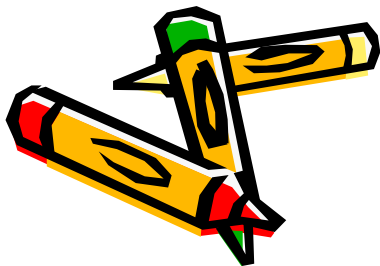
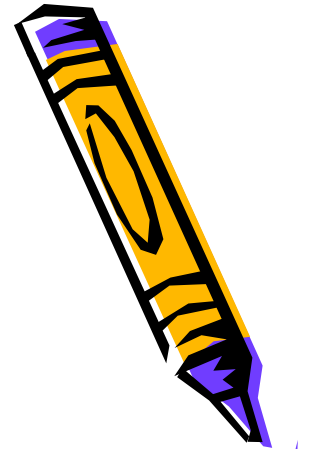
Output-ul Y este un vector cu $m(m-1)/2$ componente, dispuse în ordine $(1,2), (1,3), \dots, (1,m), (2,3), \dots, (2,m), \dots, (m-1,m)$.



exemplu

Reluăm exemplul referitor la bolile hepatice

```
»X=[162 255 74 257;210 324 93 220;422 488 183 292;  
259 208 115 266; 129 114 89 171; 607 259 65 275;  
446 283 176 309;236 270 88 150;180 119 114 171];  
» Y=pdist(X)
```





Y=

Columns 1 through 6

93.7817 367.4167 115.6719 169.0887 445.4728 307.4866

Columns 7 through 12

131.7042 166.7813 291.7602 135.8565 230.3866 406.9926

Columns 13 through 18

268.6764 92.2876 213.9322 332.0678 499.2014 317.6145

Columns 19 through 24

207.2173 333.6300 462.7386 188.2472 355.3674 214.8581

Columns 25 through 30

136.2278 152.2761 510.7847 394.5415 190.3339 57.0175

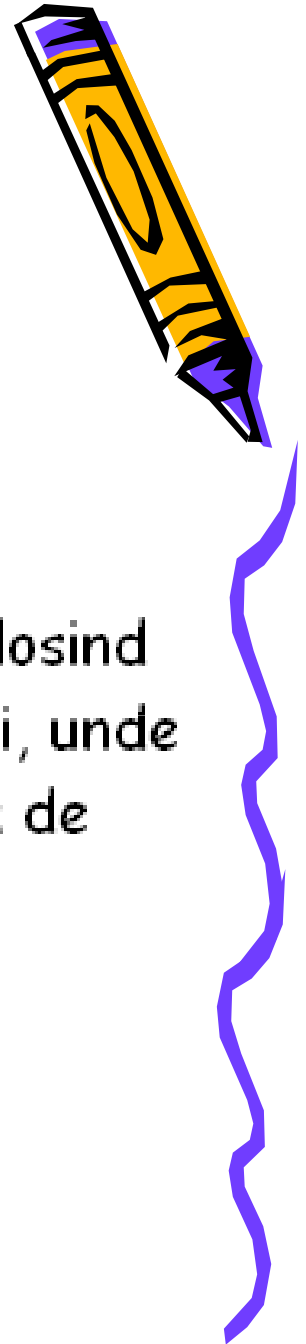
Columns 31 through 36

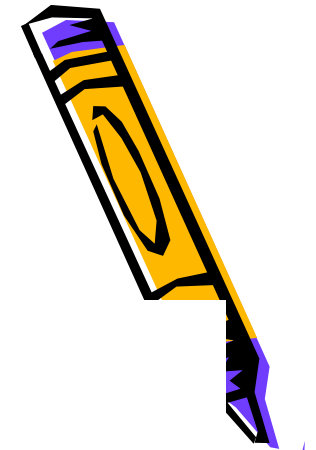
199.9350 392.3213 463.8383 278.0180 347.1887 164.4810



squareform(Y)

Putem converti acest vector într-o matrice pătratică folosind funcția `squareform`, astfel încât elementul i,j al matricei, unde $i < j$ corespunde distanței dintre obiectele i și j din baza de date.



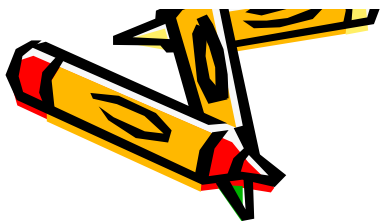


```
» squareform(Y)
```

```
ans =
```

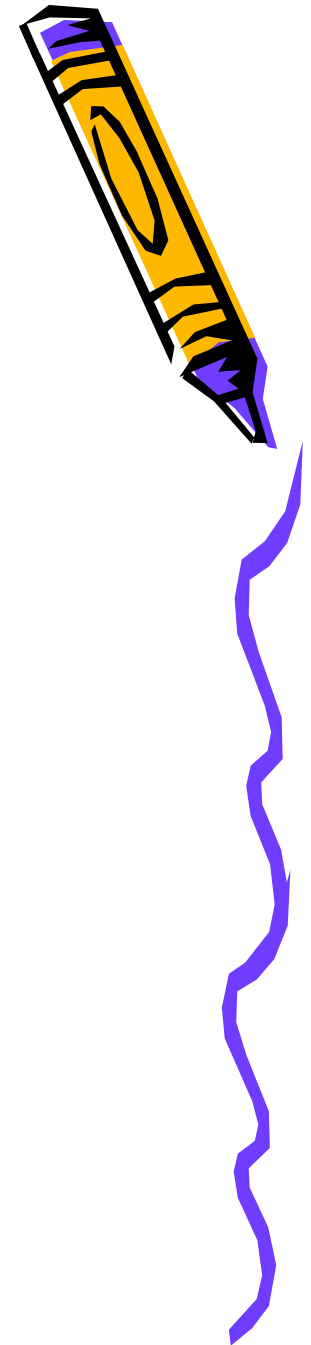
```
Columns 1 through 6
```

```
    0    93.7817  367.4167  115.6719  169.0887  445.4728
  93.7817     0   291.7602  135.8565  230.3866  406.9926
 367.4167 291.7602     0   332.0678  499.2014  317.6145
 115.6719 135.8565  332.0678     0   188.2472  355.3674
 169.0887 230.3866  499.2014  188.2472     0   510.7847
 445.4728 406.9926  317.6145  355.3674  510.7847     0
 307.4866 268.6764  207.2173  214.8581  394.5415  199.9350
 131.7042  92.2876  333.6300  136.2278  190.3339  392.3213
 166.7813 213.9322  462.7386  152.2761  57.0175  463.8383
```



Columns 7 through 9

307.4866	131.7042	166.7813
268.6764	92.2876	213.9322
207.2173	333.6300	462.7386
214.8581	136.2278	152.2761
394.5415	190.3339	57.0175
199.9350	392.3213	463.8383
0	278.0180	347.1887
278.0180	0	164.4810
347.1887	164.4810	0



'metric'

$Y = \text{pdist}(X, \text{'metric'})$ calculează distanțele dintre obiecte utilizând distanța specificată în 'metric'

- 'euclidean' - distanța euclidiană
- 'seuclidean' - distanța euclidiană standardizată
- 'mahalanobis' – distanța Mahalanobis
- 'cityblock' - distanța City Block
- 'minkowski' – distanța Minkowski
- 'correlation' - distanța coeficientului de corelație
- 'chebychev' - distanța Cebâșev



linkage



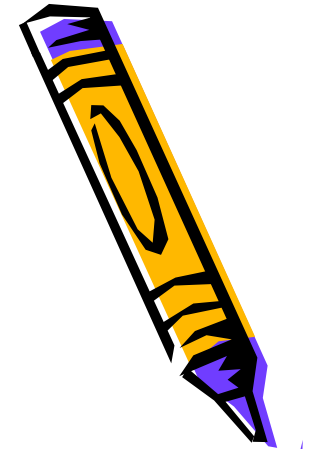
2) Grupăm obiectele într-un cluster tree binar (arbore binar de cluster ierarhic), legând perechi de obiecte, care sunt apropiate, folosind funcția linkage. Funcția linkage folosește informațiile asupra distanței pentru a determina apropierea obiectelor, unele de altele. Cum obiectele sunt cuplate în cluster binare, noile cluster formate vor fi grupate în cluster mai mari, și astfel vom construi arborele ierarhic.



Z=linkage(Y,'method')

Z = linkage(Y,'method') calculează clusterul ierarhic folosind algoritmul cerut prin „method”

- 'single' = *Single linkage*
- 'complete' = *Complete linkage*
- 'average' = *Average linkage*
- 'ward' = *Ward linkage.*
- 'centroid' = *Centroid linkage*

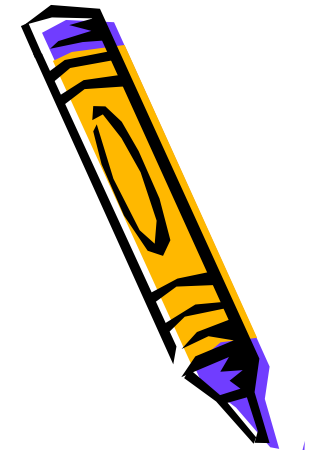


centroid linkage



Această metodă poate da naștere unui cluster tree care nu este monoton și anume în cazul în care distanța de la centroidul reuniunii clusterelor $A \cup B$ la centroidul altui cluster C este mai mică decât distanța dintre centroizii clusterelor A și C sau dintre centroizii clusterelor B și C . În acest caz este indicată utilizarea altei metode (linkage).





Outputul Z este o matrice de tip $(m - 1) \times 3$ ce conține informație despre cluster-tree.

Nodurile frunze in ierarhia clusterului sunt obiectele din baza de date numerotate de la 1 la m . Ele sunt clusterere cu un singur element, din care vor fi construite următoarele clusterere.





Fiecărui cluster nou format corespunzător liniei i în Z îi asociem indicele $m+i$, unde m este numărul frunzelor inițiale. Coloanele 1 și 2 din Z conțin indicele obiectelor ce sunt grupate pentru a forma un nou cluster. În coloana 3, pe linia i din Z avem afișată distanța dintre obiectele ce apar pe această linie.



exemplu

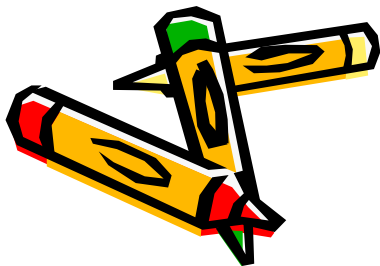


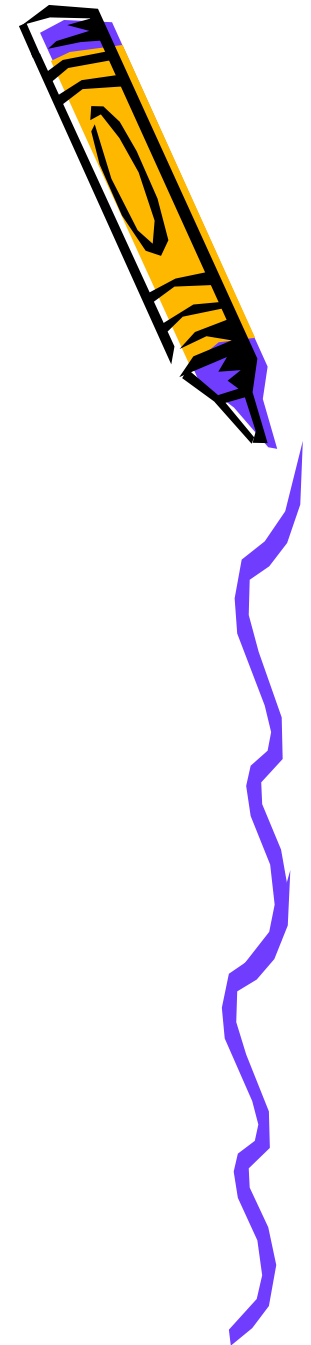
Considerăm $m=30$; dacă al 10-lea cluster combină prin funcția linkage obiectul 5 cu obiectul 7 și distanța dintre ele este 1.5, atunci cea de-a 10-a linie a lui Z va fi

5 7 1.5

și noul cluster format va avea indicele $30+10=40$

Dacă pe altă linie apare clusterul 40, înseamnă ca acest cluster a fost combinat cu altul, într-un cluster mai mare.





Revenind la *exemplul anterior* vom avea:

» $Z = \text{linkage}(Y)$

$Z =$

5.0000	9.0000	57.0175
2.0000	8.0000	92.2876
1.0000	11.0000	93.7817
12.0000	4.0000	115.6719
13.0000	10.0000	152.2761
6.0000	7.0000	199.9350
3.0000	15.0000	207.2173
14.0000	16.0000	214.8581





Pentru început se formează un cluster între pacientul 5 și pacientul 9, între care distanța euclidiană este de 57.0175, cluster ce va fi indexat cu numărul 10.

Obținem apoi clusterul 11 format prin alăturarea pacientului 2 cu pacientul 8.

Clusterul 11 este unit cu pacientul 1 obținând clusterul 12.

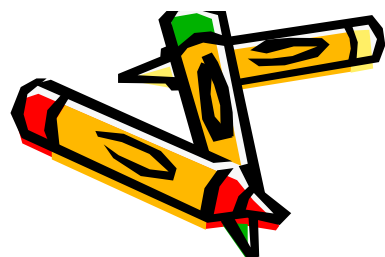
Din clusterul 12 unit cu pacientul 4 obținem clusterul 13.

Clusterul 13 se unește cu clusterul 10, rezultând clusterul 14.

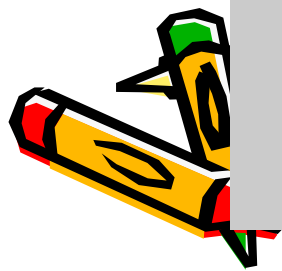
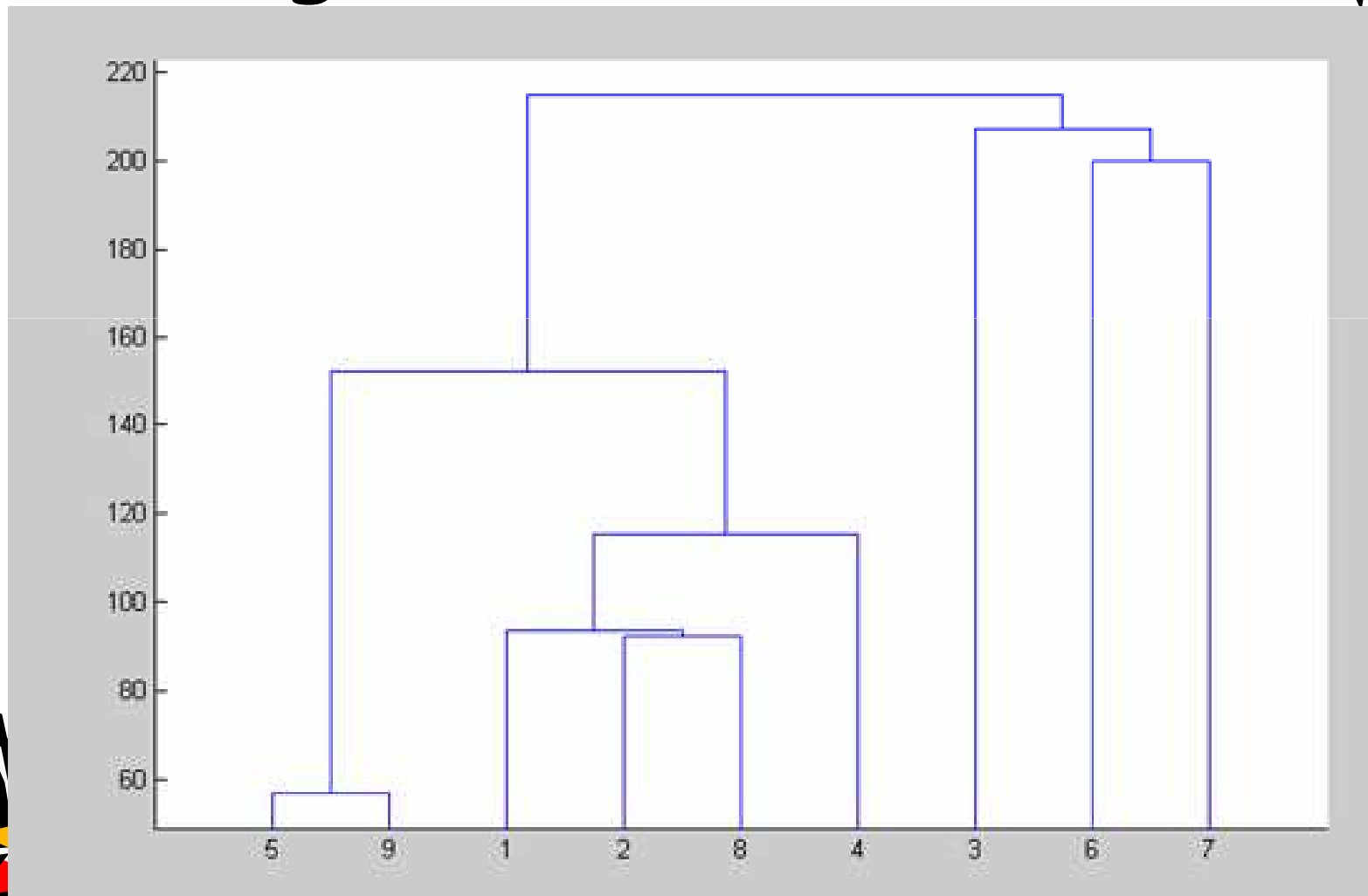
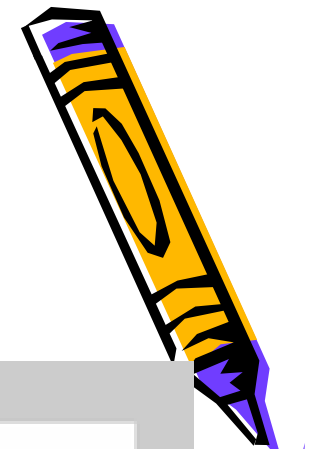
Pacientul 6 unit cu pacientul 7 formează clusterul 15.

Pacientul 3 împreună cu clusterul 15 formează clusterul 16.

În final se calculează distanța între clusterul 14 și clusterul 16.



dendrogram(Z)





3) După ce am grupat obiectele în clustere avem de verificat dacă clusterul ierarhic reprezintă grupări de obiecte similare. Aceasta verificare se face măsurând valabilitatea informației generată de funcția linkage și anume prin compararea cu datele generate de funcția pdist.

Dacă clusterizarea este validă legăturile obiectelor în cluster trebuie să aibă o puternică corelație cu distanțele dintre obiecte, din vectorul distanță.

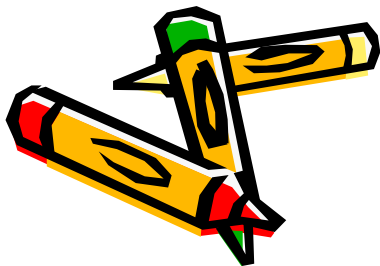


cophenet

Functia cophenet compara aceste două mulțimi de date și calculează corelata lor, returnând o valoare numită coeficient cophenetic de corelație.

`c = cophenet(Z, Y)`

Cu cât acest coeficient este mai apropiat de 1, cu atât mai bună este soluția de clusterizare.





Putem folosi coeficientul cophenetic pentru a compara rezultatele clusterizării aceluiași date, folosind distanțe diferite sau algoritmi diferiți de clusterizare.





În exemplul prezentat, folosind distanța euclidiană și *Single linkage* avem:

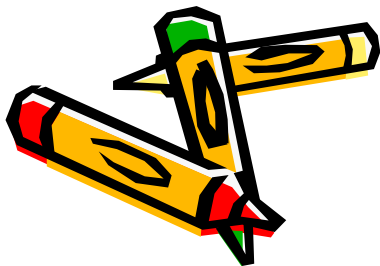
» $Y = \text{pdist}(X)$

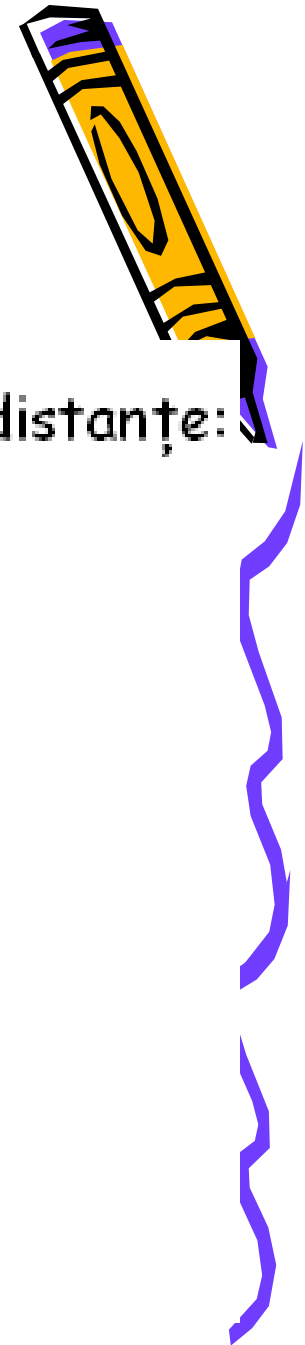
» $Z = \text{linkage}(Y)$

» $c = \text{cophenet}(z,y)$

$c =$

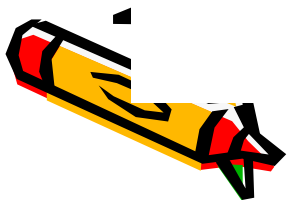
0.8321

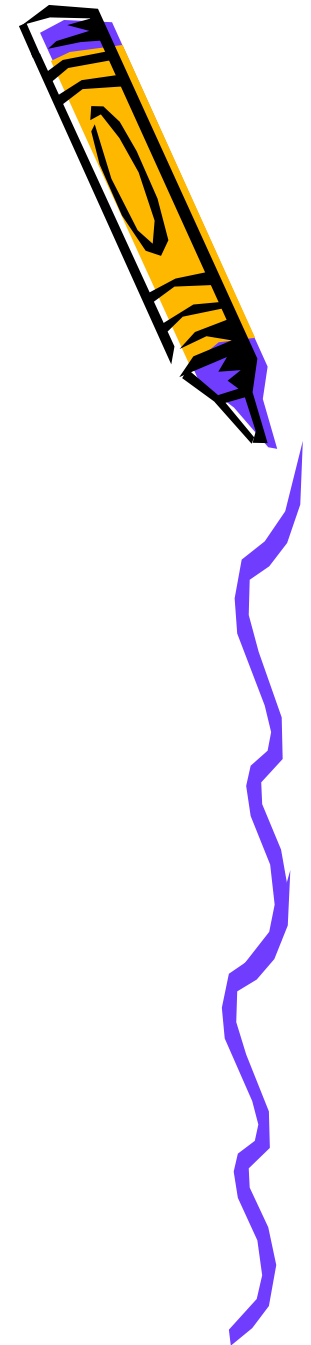




Pentru început folosim *Single linkage* și diferite distanțe:

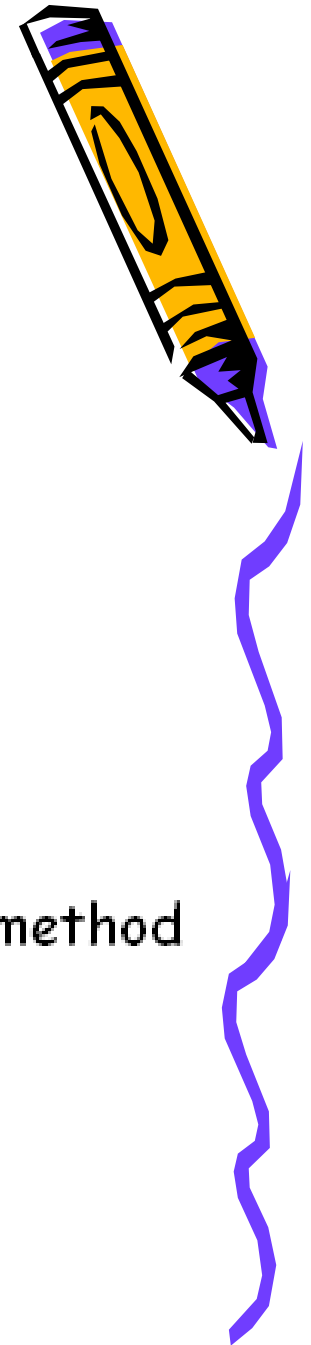
- » `Y1 = pdist(X, 'seuclidean');`
» `Z1 = linkage(Y1);`
» `c = cophenet(Z1, Y1)`
`c =`
`0.7672`
- » `Y2 = pdist(X, 'mahalanobis');`
» `zZ = linkage(Y2);`
» `c2 = cophenet(Z, Y2)`
`c2 =`
`0.6859`





- »Y3=pdist(X,'cityblock');
»Z = linkage(Y3);
»c3= cophenet(Z, Y3)
c3 =
0.8446
- »Y4=pdist(X,'minkowski');
»Z = linkage(Y4);
» c4= cophenet(Z, Y4)
c4 =
0.8321

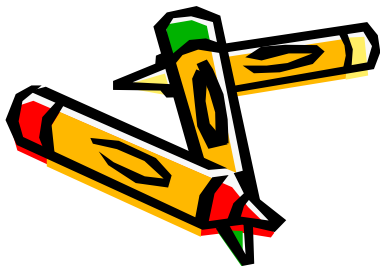


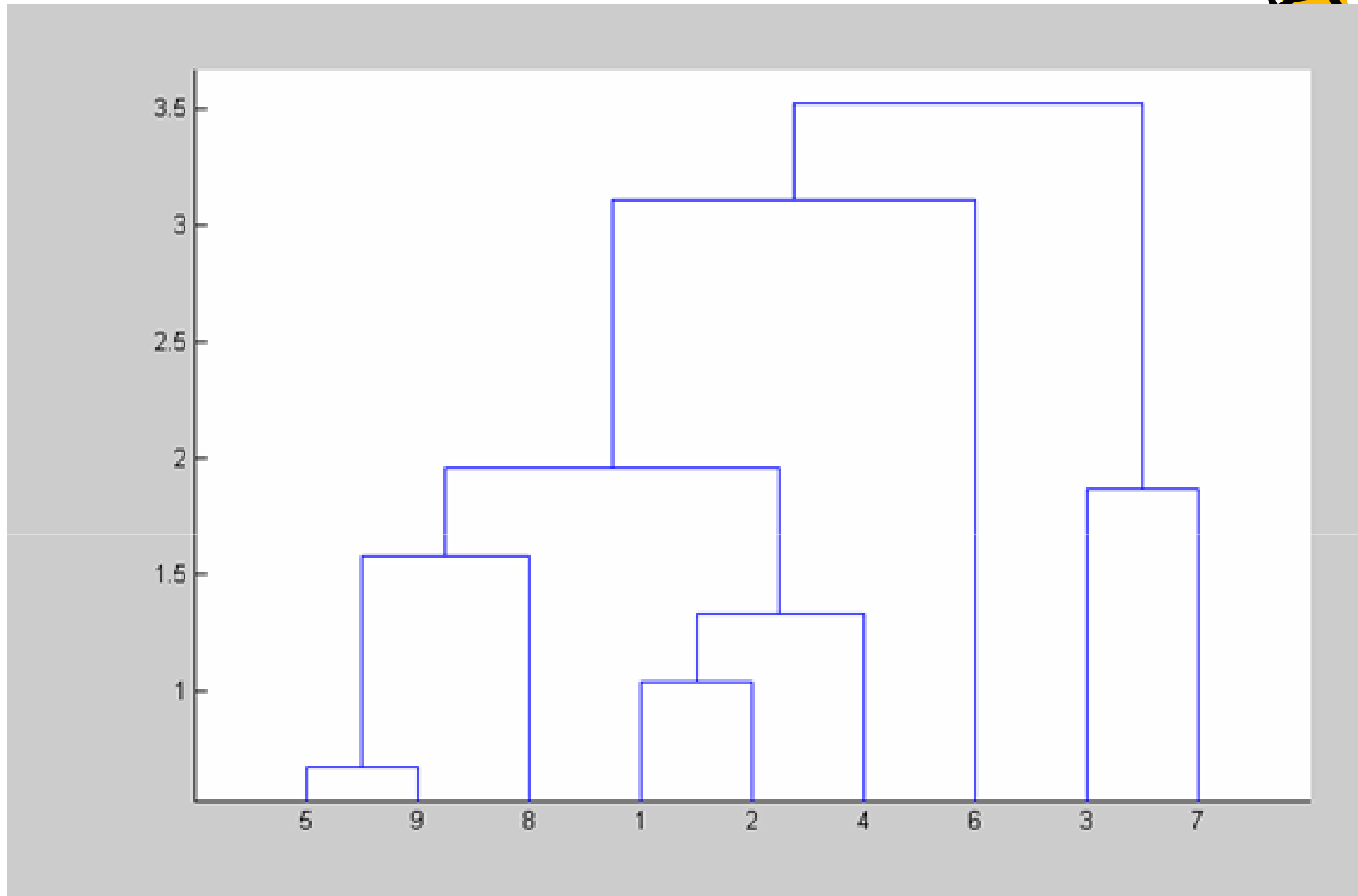


Continuăm cu alte variante:

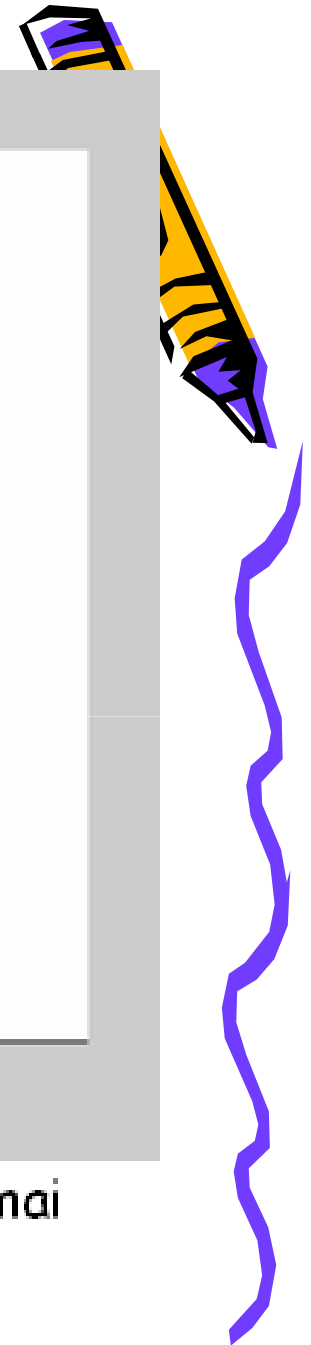
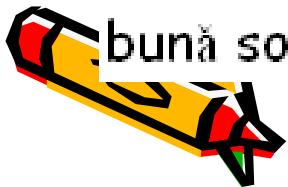
- » `y1=pdist(X,'mahalanobis');`
 » `z1=linkage(y1,'centroid');`

Warning: Non-monotonic cluster tree -- the centroid method is probably not appropriate.





Dendrograma pentru cel mai bun coeficient, nu este și cea mai bună soluție.





O posibilitate de a determina divizările naturale ale clusterului într-o mulțime de date constă în a compara înălțimea fiecărui link din cluster tree cu înălțimile linkurilor vecine, care se află sub acesta în arbore.





Faptul că un link este aproximativ la aceeași înălțime cu linkurile vecine, indică existența similarităților între obiectele unite la același nivel ierarhic.

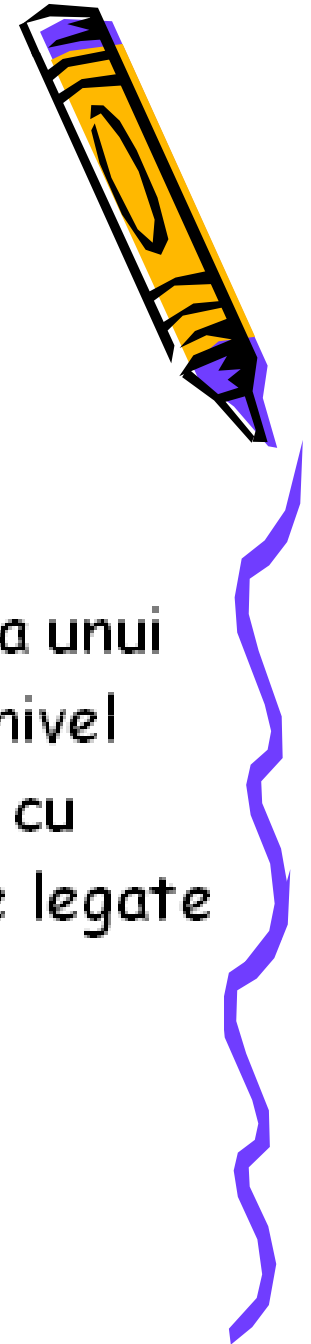
Se spune că aceste linkuri prezintă un înalt nivel de consistență.



Faptul că înălțimea unui link diferă de cele ale linkurilor din vecinătate indică existența disimilarităților între obiecte la acest nivel al cluster arborelui și se spune că acest link este inconsistent cu linkurile din jurul lui.



coeficientul de inconsistenta



Coeficientul de inconsistență, compară lungimea unui link cu media lungimilor linkurilor de la același nivel ierarhic. Cu cât aceasta valoare este mai mare cu atât mai puține asemănări există între obiectele legate de acest link.



U=inconsistent(Z)

»U=inconsistent (Z) calculeaza coeficientul de inconsistență,

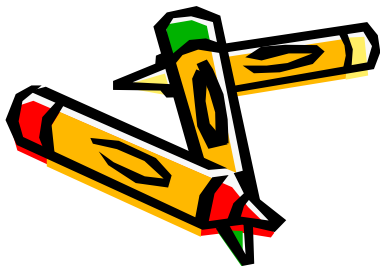
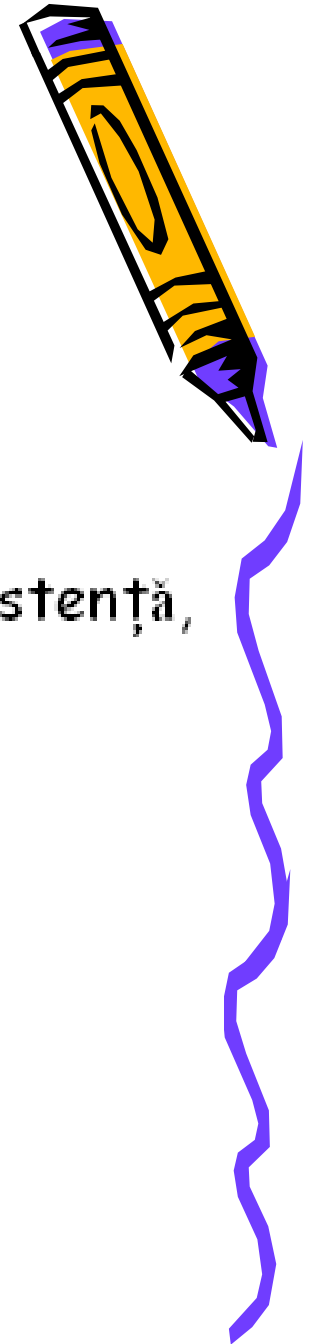
U este o matrice cu m-1 linii si 4 coloane:

Coloana 1 media lungimii linkurilor luate în calcul

Coloana 2 deviația standard a linkurilor luate în calcul

Coloana 3 numărul linkurilor luate în calcul

Coloana 4 coeficientul de inconsistență



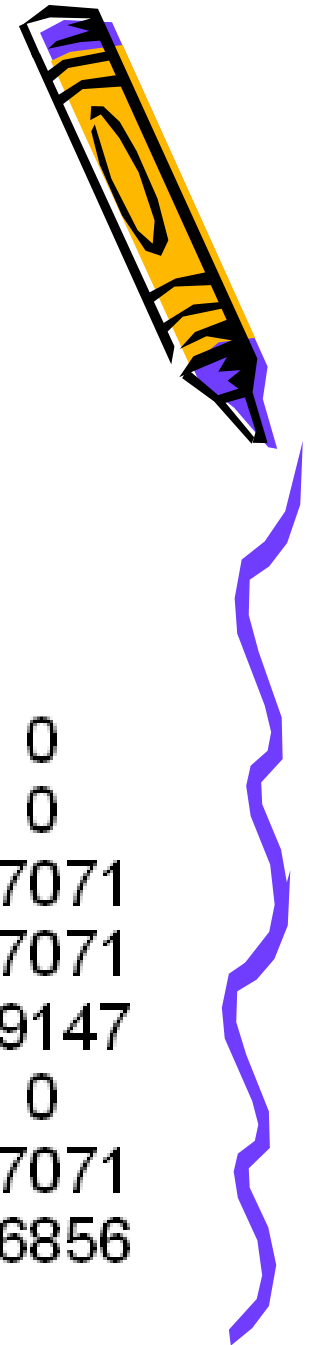


Tinând seama de descrierea matricei $Z = \text{linkage}(Y)$, vom avea următoarea formula de calcul pentru coeficientul de inconsistență:

$$U(k,4) = \frac{Z(k,3) - U(k,1)}{U(k,2)}$$



exemplu

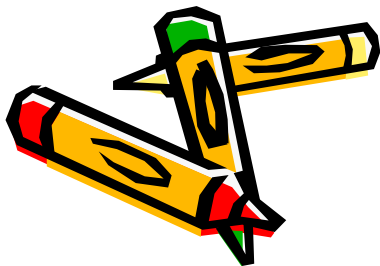


În exemplul prezentat avem:

» $U = \text{inconsistent}(Z)$

$U =$

57.0175	0	1.0000	0
92.2876	0	1.0000	0
93.0346	1.0565	2.0000	0.7071
104.7268	15.4788	2.0000	0.7071
108.3219	48.0527	3.0000	0.9147
199.9350	0	1.0000	0
203.5761	5.1494	2.0000	0.7071
191.4505	34.1405	3.0000	0.6856





În analiza clusterelor linkurile inconsistente indică frontiera unei împărțiri naturale în mulțimea datelor.

Functia cluster utilizeaza o măsura a inconsistenței pentru a determina unde să împărțim mulțimea de obiecte în alte cluster.





Intr-un cluster tree ierarhic, mulțimea de date poate fi natural împărțită în clustere.

Aceasta poate fi evident fi într-o dendogramă în care grupuri de obiecte se află într-o anumită arie.

Coeficientul de inconsistența a linkurilor poate identifica acele puncte in care se schimbă similaritatea obiectelor.



cluster

Dacă folosim funcția cluster pentru a grupa obiectele în clustere specificăm treshhold-ul pentru coeficientul de inconsistență

» `T=cluster(Z,0.9)`

T =

1

1

3

1

2

3

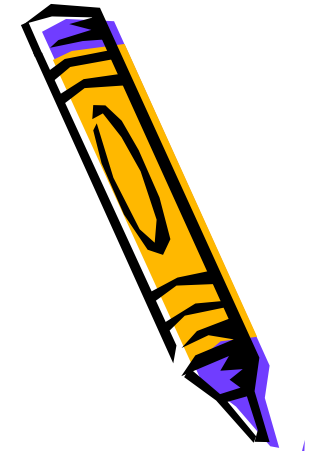
3

1

2



find



Ceea ce înseamnă ca pacienții 1,2, 4 și 8 aparțin primului cluster, 5 și 9 celui de al doilea și pacienții 3,6 și 7 aparțin celui de al treilea cluster; aceste explicații detaliate le putem obține și folosind funcția find

```
»find(T==1)
```

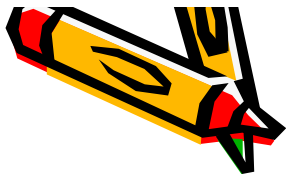
```
ans =
```

```
1
```

```
2
```

```
4
```

```
8
```



» find(T==2)

ans =

5

9

» find(T==3)

ans =

3

6

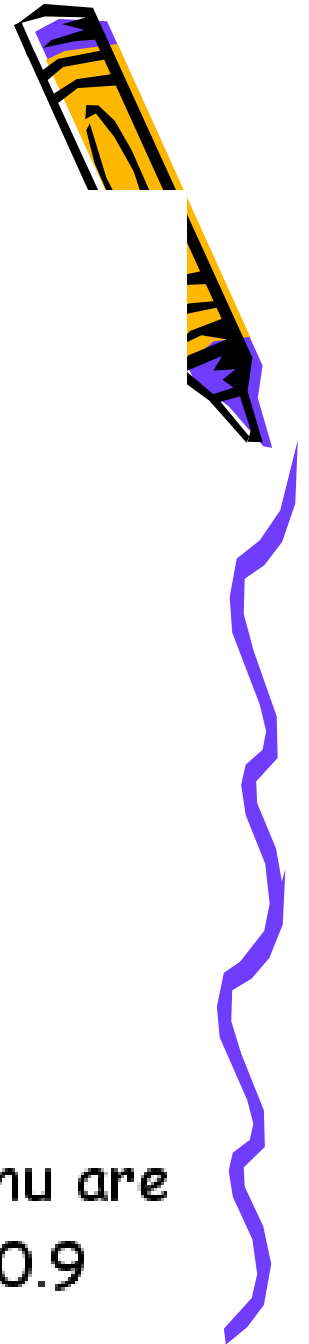
7

» find(T==4)

ans =

Empty matrix: 0-by-1

În acest caz niciun link din clusterul ierarhic nu are coeficientul de inconsistență mai mare decât 0.9





Să considerăm treshhold-ul pentru coeficientul de
inconsistență 0.7

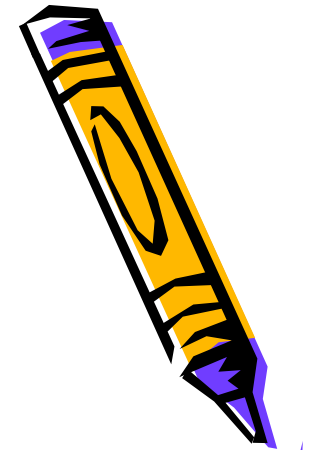
» $T = \text{cluster}(Z, 0.7)$

T =

1
3
2
4
5
6
6
3
5

caz in care avem 6 cluster





sau când treshholdul este 0.915

» $T = \text{cluster}(Z, 0.915)$

$T =$

1
1
1
1
1
1
1
1
1
1

caz în care avem un singur mare cluster.



T=cluster(Z, 'maxclust',n)

Funcția

T = cluster(Z,'maxclust',n)

specifică numărul maxim de clustere, n ce dorim să obținem în cluster tree-ul ierarhic.



Revenind la exemplul comentat

```
T = cluster(Z, 'maxclust', 2)
```

```
» T =
```

```
1
```

```
1
```

```
2
```

```
1
```

```
1
```

```
2
```

```
2
```

```
1
```

```
1
```





Folosind aceste tehnici de clusterizare descoperim clase „naturale” în care se plasează obiectele din baza de date și nu realizăm o nouă ordine în structura datelor.

