Învățarea nesupervizată
Clustering

Învățarea automată
Direcții principale

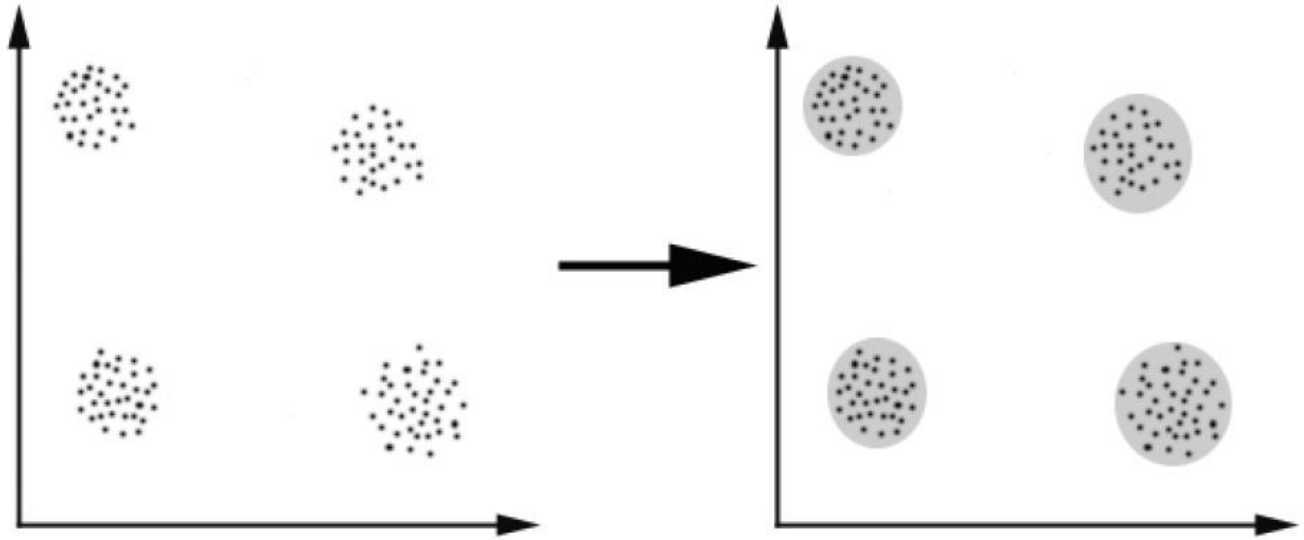
1. Clasificarea și regresia
2. Gruparea (clustering [clástering], clusterizare [clasterizáre])
3. Determinarea regulilor de asociere
4. Selecția trăsăturilor

Învățarea automată
Direcții principale

1. Clasificarea și regresia
2. Gruparea (clustering [clástering], clusterizare [clasterizáre])
3. Determinarea regulilor de asociere
4. Selecția trăsăturilor

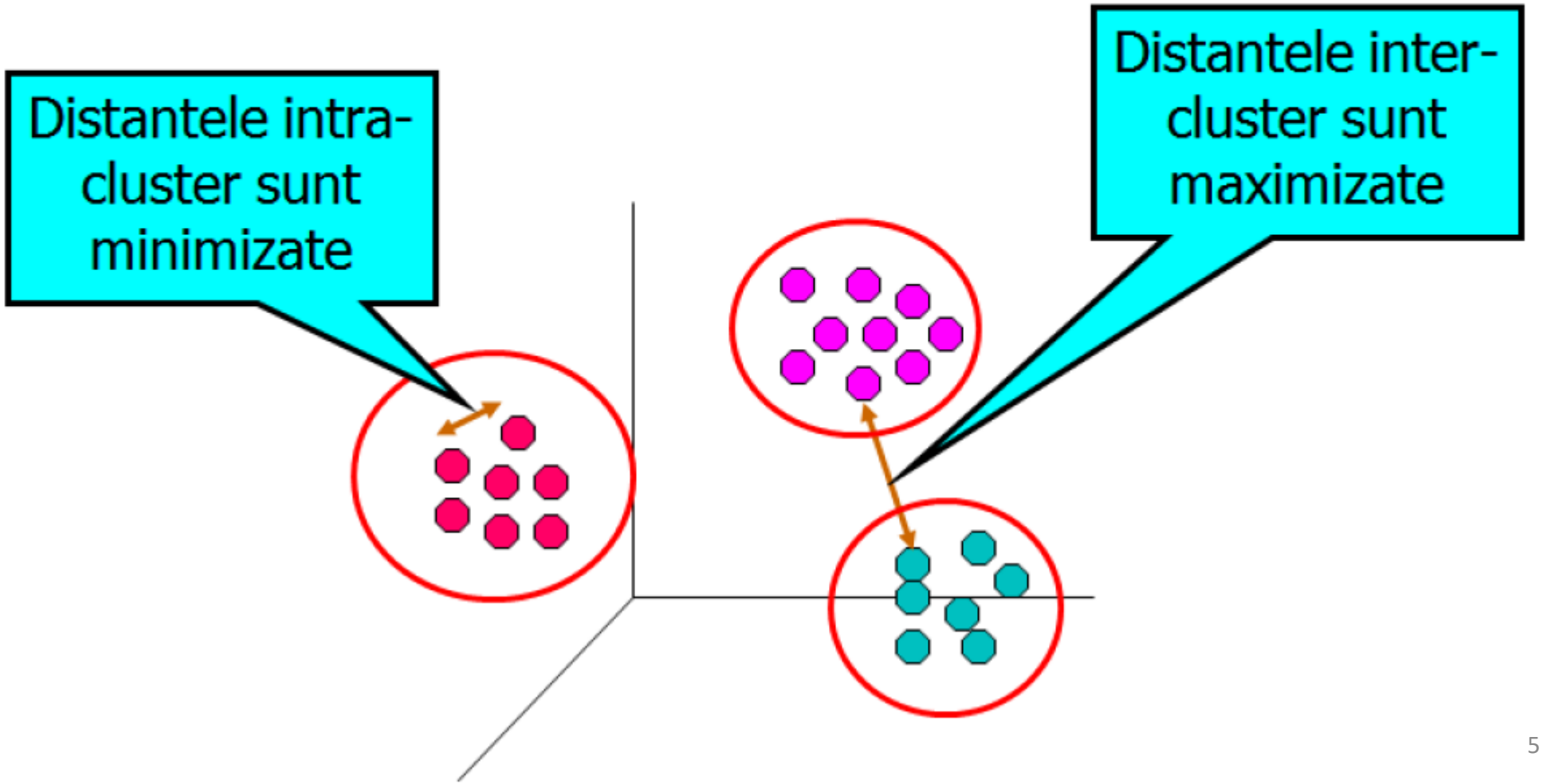
Gruparea (clusterizarea)

- Are ca scop găsirea unor grupuri astfel încât instanțele din același grup să fie mai asemănătoare între ele decât cu instanțele altor grupuri
- De obicei, este nesupervizată



Scop

- Analiza grupărilor (eng: cluster analysis) divide datele în grupuri, pe baza similarităților sau a relațiilor dintre ele
  - datele din același cluster sunt similare între ele
  - datele din alte clustere ar trebui să aibă similaritate mică cu cele din clusterul curent



Scopul analizei grupărilor

- Prin clustering se încearcă obținerea de grupări care sunt:
  - semnificative: clusterele trebuie să surprindă natura structurală a datelor
  - utile: sumarizarea unui volum mare de date, furnizare de explicații

  sau ambele

Scop

Domenii de utilizare:

- științe sociale
- biologie
- statistică
- recunoaștere de șabloane (pattern recognition)
- regăsirea informației
- analiza imaginilor
- bioinformatică

Clustering pentru facilizarea inţelegerii

- Scop: obținerea de clase de elemente, utile în analizarea și înțelegerea lumii înconjurătoare.
- Exemple:
  - **biologie:** obținerea de taxonomii: regn, încrengătură, clasă, ordin, familie, gen, specie;
  - **bioinformatică**: gruparea genelor și a proteinelor, presupuse a avea funcționalitate similară
  - **psihologie și medicină:** boli sau stări au diferențe minore; gruparea lor e folosită pentru descoperirea de subcategorii – *e.g.* diverse tipuri de depresii
  - **afaceri:** gruparea clienților în funcție de similaritățile de achiziție și realizarea de reclame particularizate pe grupuri de clienți

Clustering pentru scopuri utilitare

- Permite abstractizare față de datele care intră în cluster
- Alternativ: clusterele se pot caracteriza deseori prin intermediul prototurilor
- Din acest punct de vedere clustering-ul este modalitatea de determinare a celor mai reprezentativi centri de cluster
  - **sumarizare:** în loc de a considera datele de plecare, se poate lucra cu centroizii clusterelor; rezultatele pot avea acuratețe comparabilă
  - **compresie:** pentru fiecare cluster se consideră centroidul lui; se obține o aproximare bazată pe substituirea datei inițiale cu centroidul clusterului de care aparține; se mai numește și **vector quantization**
  - **găsirea eficientă a celor mai apropiați vecini:** în loc să se calculeze distanțe între toate perechile de obiecte, se poate restricționa calculul la doar obiectele din același cluster, sau clustere apropiate

Ce *nu este* analiza clusterelor?

- Clasificare supervizată: clustering–ul este de fapt învățare nesupervizată
- Segmentare simplă a datelor, precum gruparea studenților după prima literă din nume
- Partiționarea de grafuri: există câteva elemente comune, dar partiționarea de grafuri se face după specificații externe mai complexe decât pentru clustering, sau subgrafurile obținute nu au o separare considerabilă

Ambiguitatea noţiunii de cluster

- În multe domenii noțiunea de cluster nu e clar definită

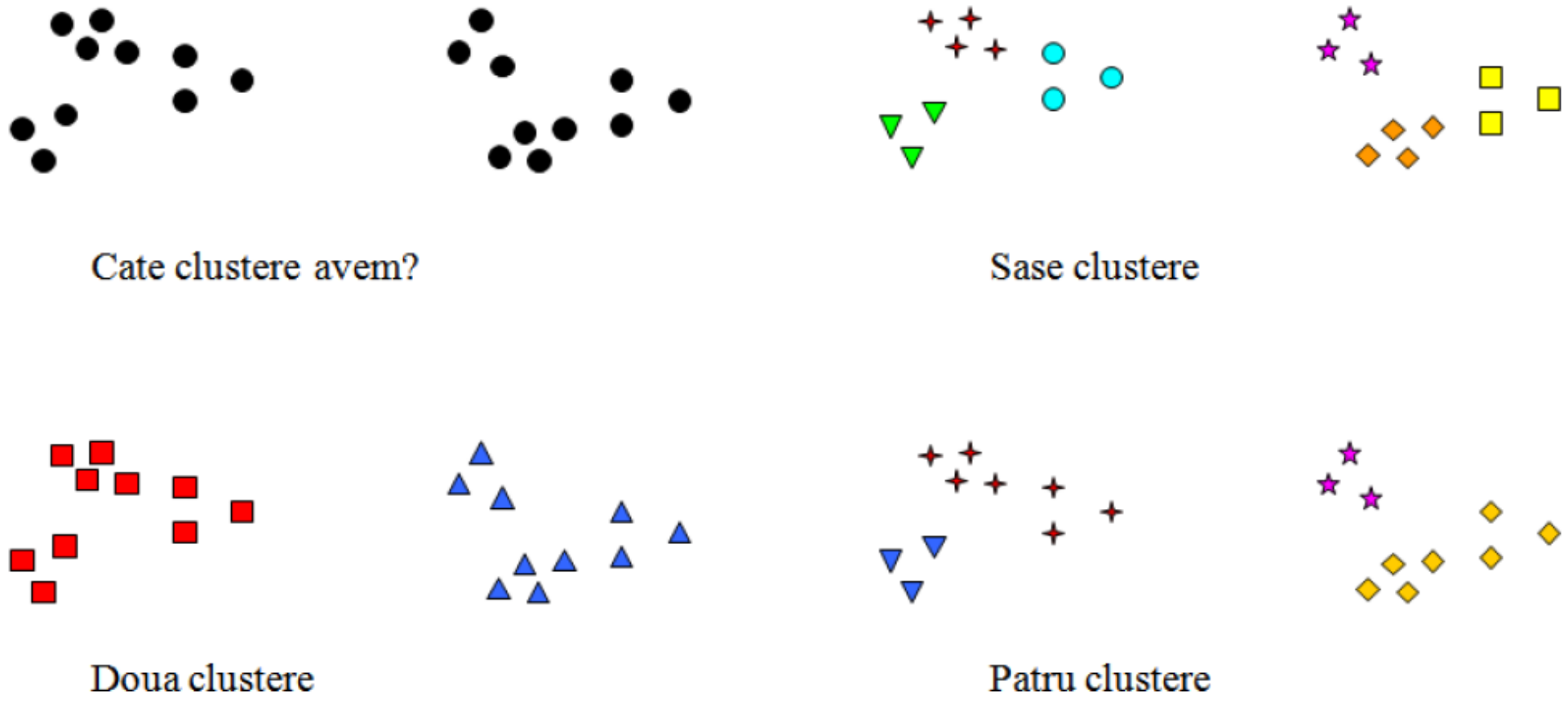


Figure: Moduri de grupare

Algoritmi de grupare

- Algoritmul k-medii (k-means)
- Aloritmul EM (Expectation-Maximization)
- Gruparea ierarhica

Algoritmi de grupare

- Algoritmul k-medii (k-means)
- Aloritmul EM (Expectation-Maximization)
- Gruparea ierarhica

Algoritmul k-medii (k-means)

- Algoritmul partiționează o mulțime de instanțe numerice în k grupuri (clustere)
- k reprezintă numărul de grupuri și este un parametru de intrare ales de utilizator

Algoritmul k-medii (k-means)
Modul de functionare

- Se inițializează aleatoriu cele k centre inițiale
- Se atribuie fiecărui centru instanțele cele mai apropiate de el

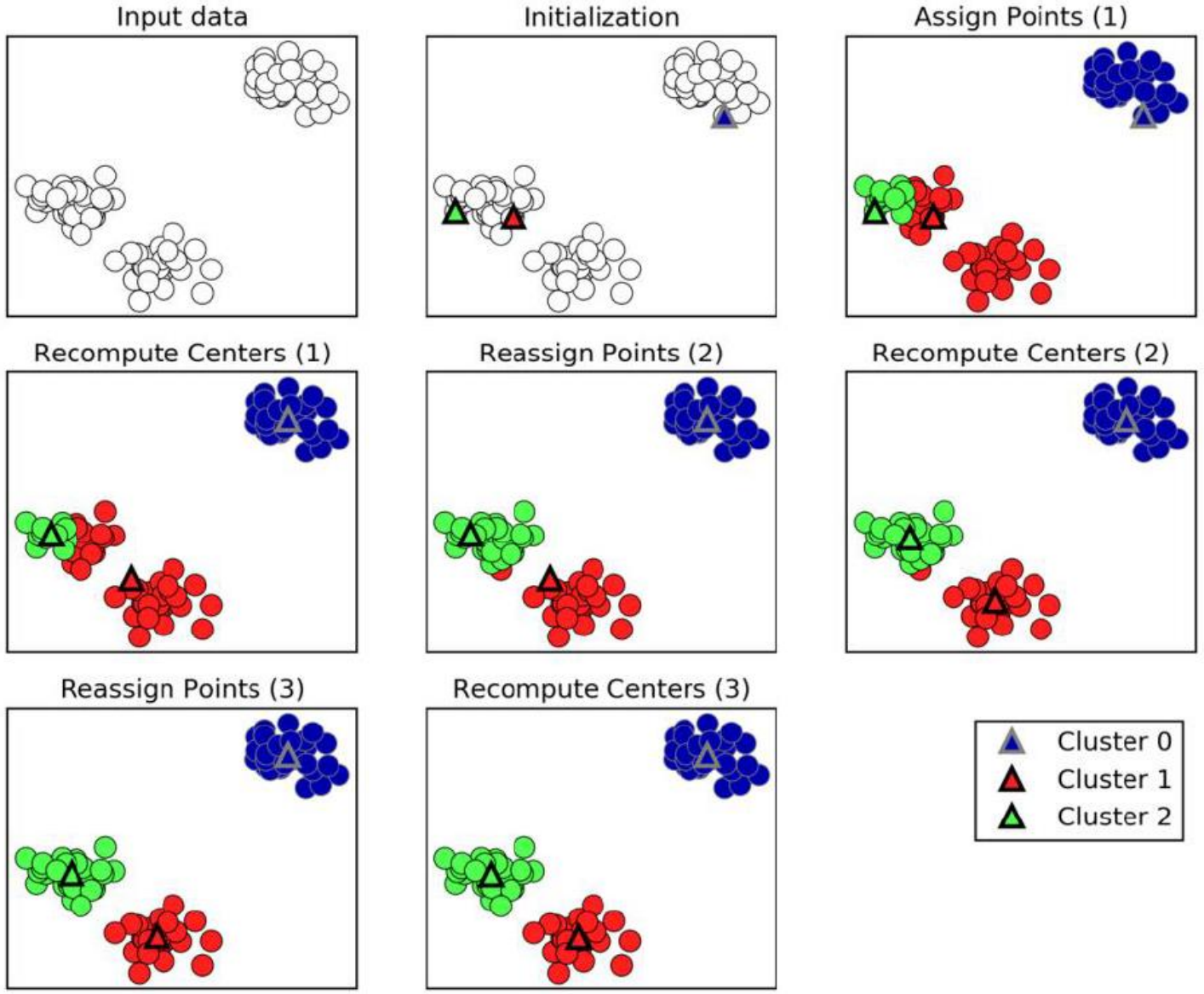
$$C(i) = \arg\min_{1 \le k \le K} \|x_i - m_k\|^2, \; i = 1, \ldots, N$$

- Se calculează centrul de greutate (media aritmetică) a tuturor instanțelor atribuite unui centru și se actualizează poziția centrului grupului respective

$$m_k = \frac{\sum_{i:C(i)=k} x_i}{N_k}, \; k = 1, \ldots, K.$$

- Se repetă cei doi pași până când nu se mai modifică poziția niciunui centru

Algoritmul k-medii (k-means) Exemplu



Algoritmul K-means

- Algoritm bazat pe prototipuri
- Prototipul este un centroid
- Tehnica se aplică pentru obiecte din spațiul continuu n–dimensional
- Cel mai popular algoritm de clustering
- Variantă înrudită: K-medoid — centroidul este unul din punctele din cluster

Paşii algoritmului K-means

1. Se aleg K centroizi inițiali; K e parametru de intrare
2. Fiecare punct este asignat celui mai apropiat centroid
3. Fiecare colecție de puncte asignată unui centroid este un cluster
4. Centroidul este modificat pe baza punctelor din clusterul pe care îl reprezintă
5. Se reia de la pasul 2 până când nu se mai schimbă poziția celor K centroizi

Algoritmul K-means: pseudocod

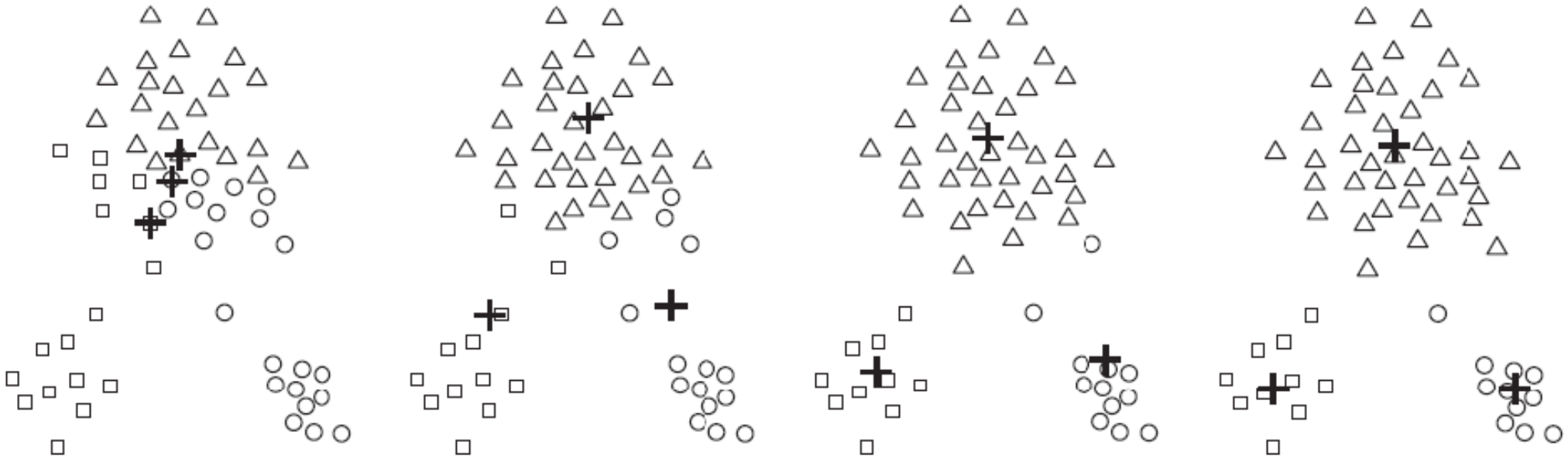
Algorithm 1 Basic K-means Algorithm.

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

Algoritmul K-means: observaţii

- Centroizii inițiali sunt cel mai des aleși aleator
- Centroidul este de regulă media punctelor din cluster
- Apropierea dintre punct și centroid se consideră ca distanță sau similaritate
  - distanță Euclidiană, similaritatea cosinus, coeficient de corelație etc.
- K-Means va converge pentru măsurile de similaritate de mai sus
- Convergența este rapidă, apare după relativ puține operații
- De multe ori condiția de oprire se schimbă în: "până când relativ puține puncte își schimbă clusterul"
- Complexitatea este $O(m \cdot K \cdot I \cdot d)$, unde m = numărul de puncte, K = numărul de clustere, I = numărul de iterații, d = dimensiunea spațiului de intrare

Algoritmul K-means: exemplificare



(a) Iteratia 1 (b) Iteratia 2 (c) Iteratia 3 (d) Iteratia 4

Figure: Exemplificare de iterații pentru K-means. Punctele marcate cu cruce sunt centroizii de la fiecare pas. La finalizare (pasul d) centroizii se stabilizează.

Algoritmul K-means: observaţii

- Pasul 4 din algoritm este:
  `Recompute the centroid of each cluster`
  deoarece centroidul se poate modifica
- Modificarea poziției centroidului duce la modificarea distanțelor de la punctele din cluster la centroid
- Problema de clustering poate fi privită ca una în care funcția obiectiv este de a minimiza suma pătratelor distanțelor de la puncte la centroizii clusterului din care fac parte
- Funcția obiectiv este utilizată pentru măsurarea calității clusteringului

Algoritmul K-means: clusteringul ca problemă de optimizare

- Funcția: **suma pătratelor erorilor**, orig: **sum of squared error (SSE)**, **scatter**
- Notații:

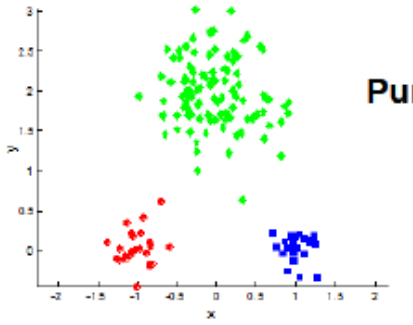
Simbol	Descriere
$\mathbf{x}$	Un obiect
C_i	Al i–lea cluster
$\mathbf{c}_i$	Centroidul clusterului i
$\mathbf{c}$	Centroidul tuturor punctelor
m_i	Numărul de obiecte din al i–lea cluster
m	numărul de obiecte din setul de date
K	numărul de clustere presupuse

- Formula:

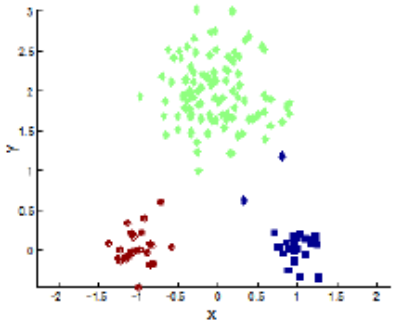
$$SSE = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} dist(\mathbf{c}_i, \mathbf{x})^2$$

Algoritmul K-means: clusteringul ca problemă de optimizare

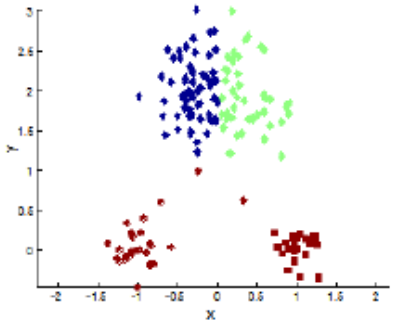
- Dacă se formează clustere diferite atunci valoarea SSE poate diferi de la caz la caz
- Putem avea deci clustering optimal și suboptimal



Punctele originale



Clustering optimal



Clustering suboptimal

Figure: Clustering optimal vs. suboptimal, relativ la funcția SSE.

Algoritmul K-means: clusteringul ca problemă de optimizare

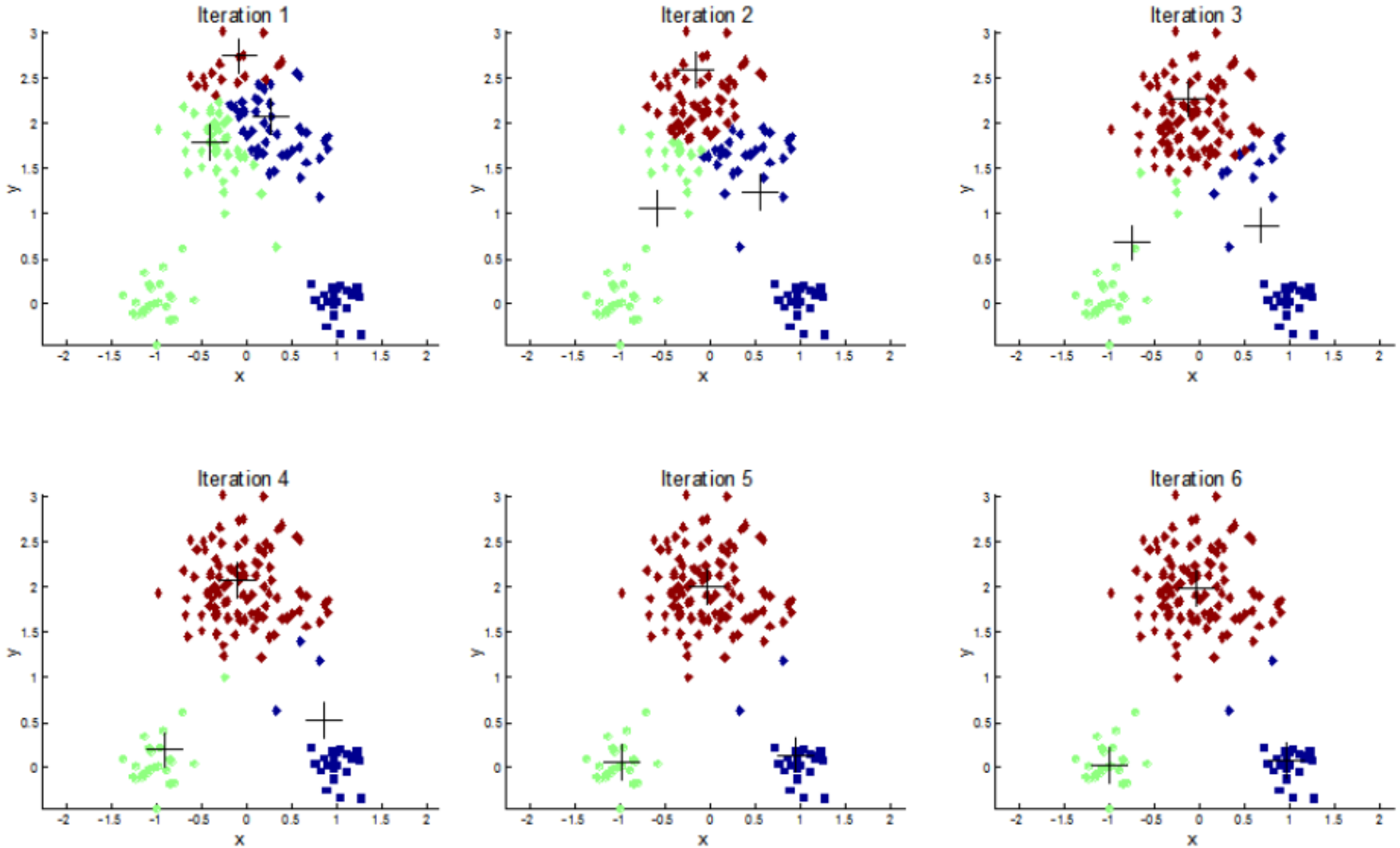
Algorithm 1 Basic K-means Algorithm.

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

- Pașii 3 și 4 din algoritm încearcă să minimizeze în mod direct SSE
- Optimul care se obține poate fi unul local: *se optimizează SSE pentru anumite alegeri ale locațiilor centroizilor, nu pentru toate variantele*

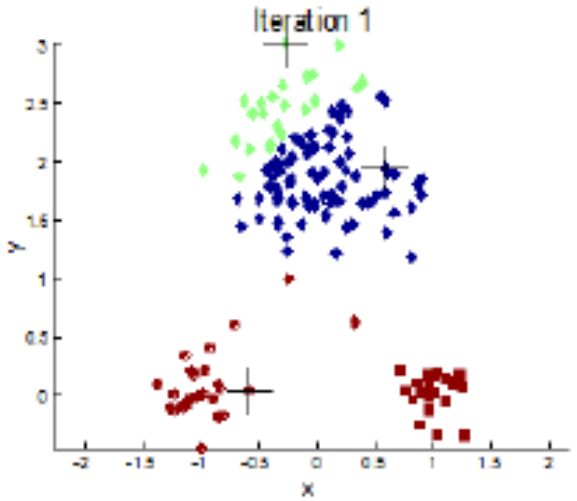
K-means: importanţa alegerii centroizilor iniţiali

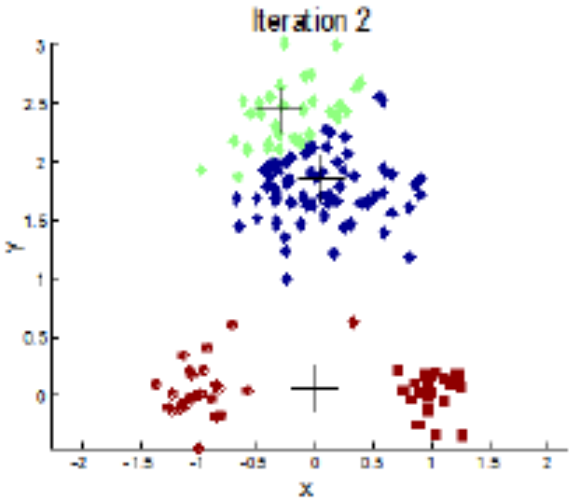
- Centroizi iniţiali aleşi "inspirat"

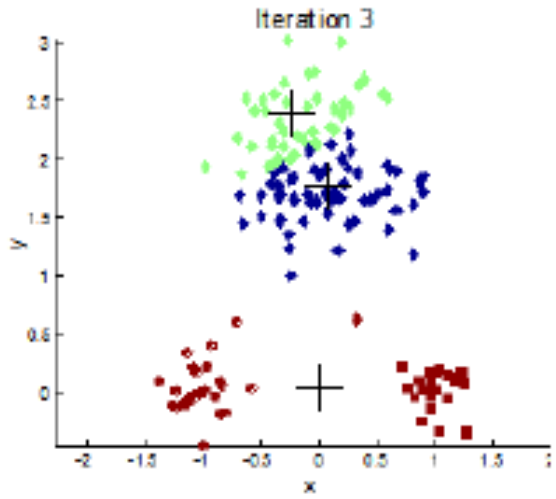


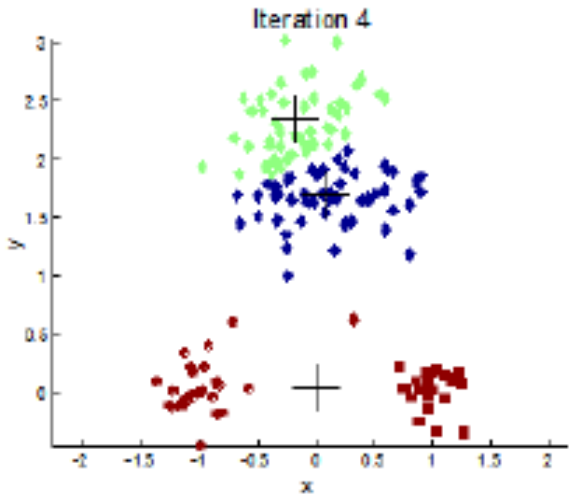
K-means: importanţa alegerii centroizilor iniţiali

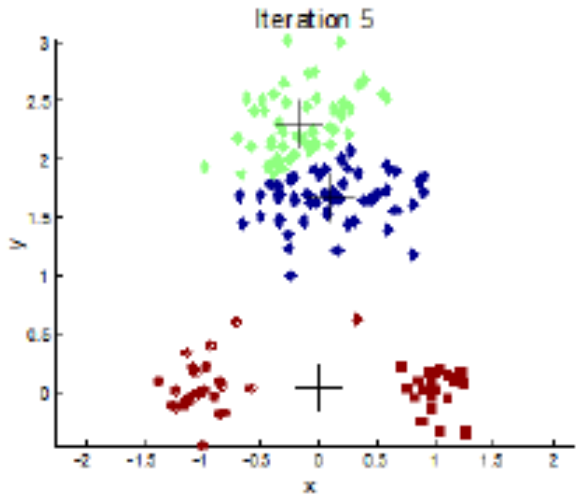
- Clustere inițiale alese nefavorabil











Algoritmul K-means: clusteringul ca problemă de optimizare

- K-means nu se reduce doar la date în spațiul Euclidian
- Pentru cazul în care datele se referă la documente: obiectivul este maximizarea similarității între documentele din același cluster = maximizarea **coeziunii** clusterului:

$$\text{Coeziunea totală} = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} \cos(\mathbf{x}, \mathbf{c}_i)$$

Algoritmul K-means: metode de reducere a impactului negativ de alegere nefavorabilă a centroizilor

- Se ia un eșantion de puncte și se aplică clustering ierarhic; centroizii rezultați sunt apoi folosiți ca și centroizi inițiali pentru K-means clustering
- K–means++ [1]
  1. Se alege uniform un centru din datele inițiale
  2. Pentru fiecare punct $\mathbf{x}$ se calculează distanța $D(\mathbf{x})$ de la x la cel mai apropiat centroid care a fost deja ales
  3. Alege un punct aleator, cu probabilitatea proporțională cu $D^2(\mathbf{x})$
  4. Repetă pașii 2 și 3 până când s-au ales K centroizi
- Bisecting K-means
- Postprocesare

Algoritmul K-means: problema clusterelor goale

- Algoritmul originar poate duce la crearea de clustere goale
- Cluster gol: niciun punct nu este alocat acelui cluster
- Dacă acest centroid este lăsat aşa, SSE creşte mai mult decat e cazul
- Soluţii:
  - se renunţă la acest centroid, se alege un altul (*e.g.* cel mai indepărtat de oricare din ceilalţi centroizi)
  - se alege un centroid din clusterul care contribuie cel mai mult la creşterea valorii SSE
- Dacă sunt mai multe clustere goale, se procedează in mod repetat ca mai sus

Algoritmul K-means: outliers

- Datele de tip outlier influenţează calculul centroizilor; aceştia pot să nu mai fie reprezentativi
- De regulă se preferă eliminarea lor inainte de clustering
  - totuşi: valorile outlier pot corespunde unor date de interes major
- Alternativ: se pot detecta şi elimina outliers in faza de postprocesare:
  - se determină acele date care contribuie cel mai mult la creşterea SSE
  - se formează clustere mici constand in grupări de outliers

Algoritmul K-means: reducerea SSE prin postprocesare

- Strategii:
  - clusterul care are valoarea SSE cea mai mare poate fi impărţit in mai multe subclustere
  - se introduce un nou centroid: cel mai depărtat punct faţă de orice cluster
  - dispersarea unui cluster: se ignoră centroidul unui cluster, punctele din el sunt reasignate altor clustere; se alege clusterul care produce creşterea maximă de SSE
  - unirea a două clustere: clusterele cu cel mai apropiat centroid se unesc şi rezultă unul singur

Algoritmul K-means: modificarea incrementală a clusterelor

- In varianta originară poziţia centroizilor este calculată doar după ce punctele sunt asignate celor mai aproppaiţi centri
- Variantă: poziţia centroidului să se calculeze după fiecare asignare a unui punct la acel centroid
- Avantaj: nu se ajunge la clustere vide
- Se poate decide asignarea unui punct la un alt centroid doar dacă manevra duce la imbunătăţirea funcţiei obiectiv
- Dezavantaj: algoritmul devine dependent de ordinea de prezentare a datelor

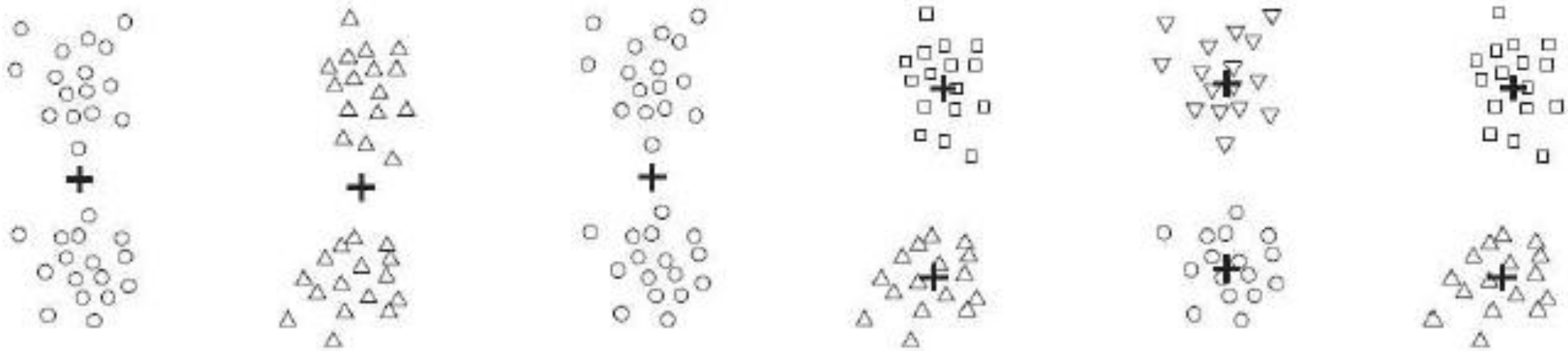
Bisecting K-means

- Se bazează pe împărțirea succesivă a unui cluster în două subclustere
- Alegerea clusterului care se împarte se poate face după mai multe criterii: cel mai larg clsuter, cel care contribuie cel mai mult la SSE, combinație de aceste două criterii
- Pas opțional: se aplică K-means cu centroizii obținuți prin bisecting K-means

Algorithm 8.2 Bisecting K-means algorithm.

1: Initialize the list of clusters to contain the cluster consisting of all points.
2: **repeat**
3: Remove a cluster from the list of clusters.
4: {Perform several "trial" bisections of the chosen cluster.}
5: **for** $i = 1$ to *number of trials* **do**
6: Bisect the selected cluster using basic K-means.
7: **end for**
8: Select the two clusters from the bisection with the lowest total SSE.
9: Add these two clusters to the list of clusters.
10: **until** Until the list of clusters contains K clusters.

Bisecting K-means: exemplu



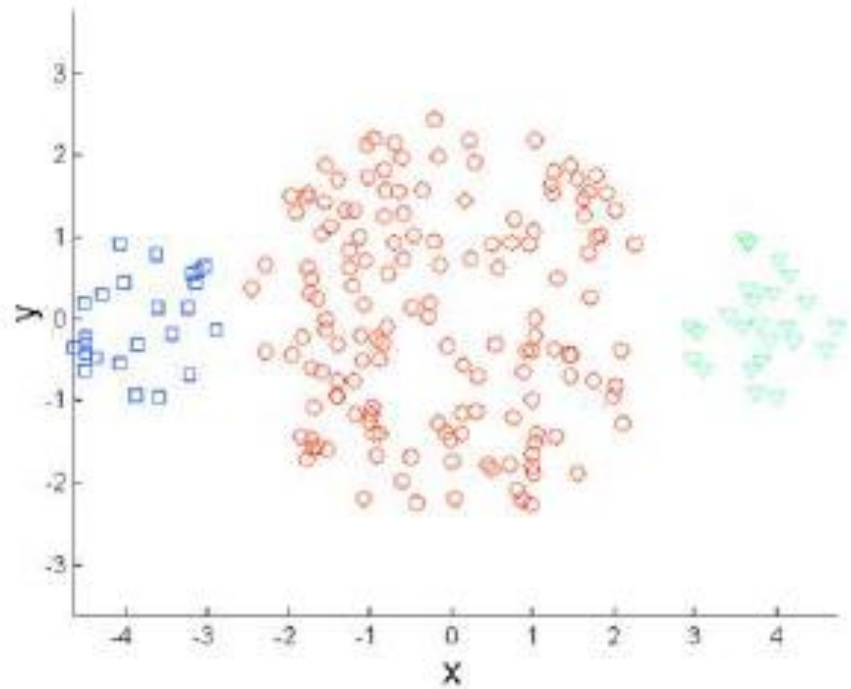
(a) Iteratia 1 (b) Iteratia 2 (c) Iteratia 3

Figure: Exemplu de aplicare a algoritmului bisecting K-means.

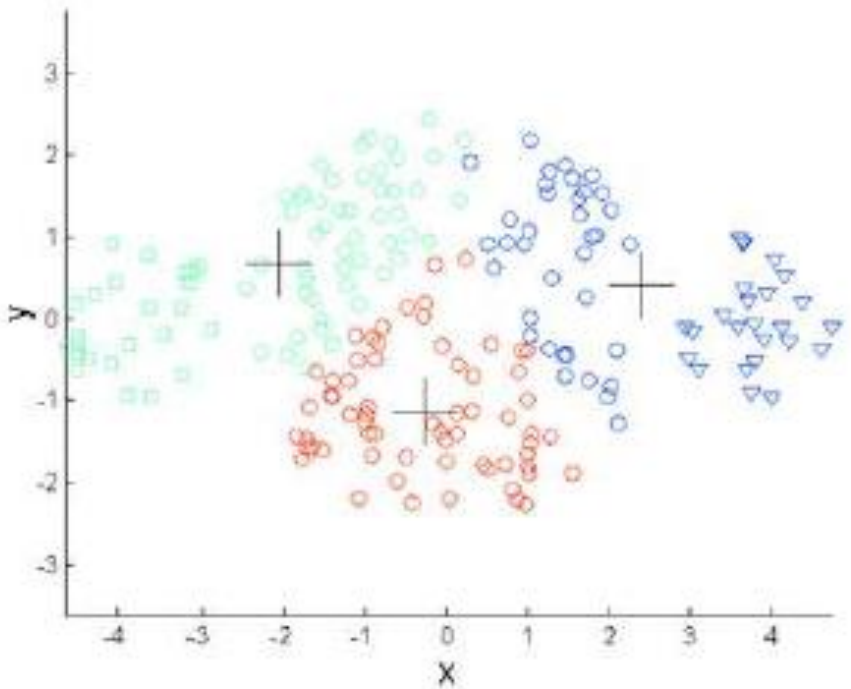
Limitări ale algoritmului K-means

- K-means intampină probleme cand clusterele sunt de diferite
  - dimensiuni
  - densităţi
  - forme, altceva decat cele circulare
- K-means are probleme cand datele prezintă outliers

Limitări ale K-means: clustere de dimensiuni diferite

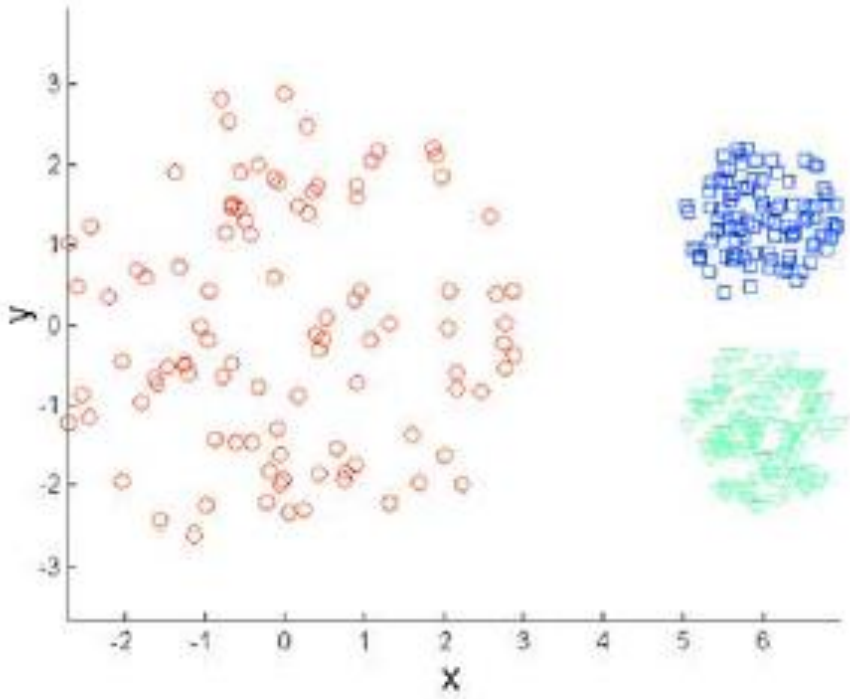


Punctele initiale

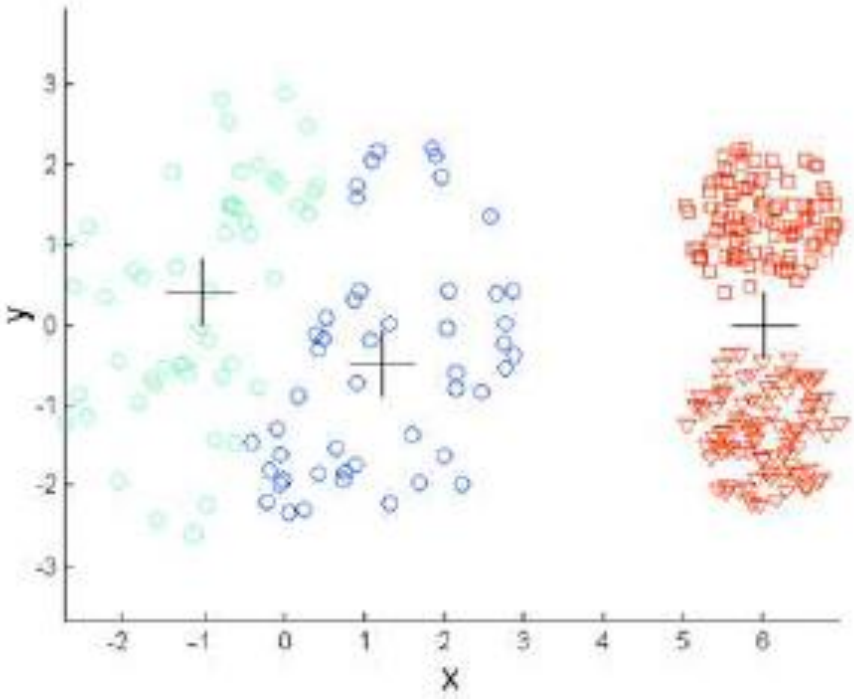


K-means (3 clustere)

Limitări ale K-means: clustere de densităţi diferite

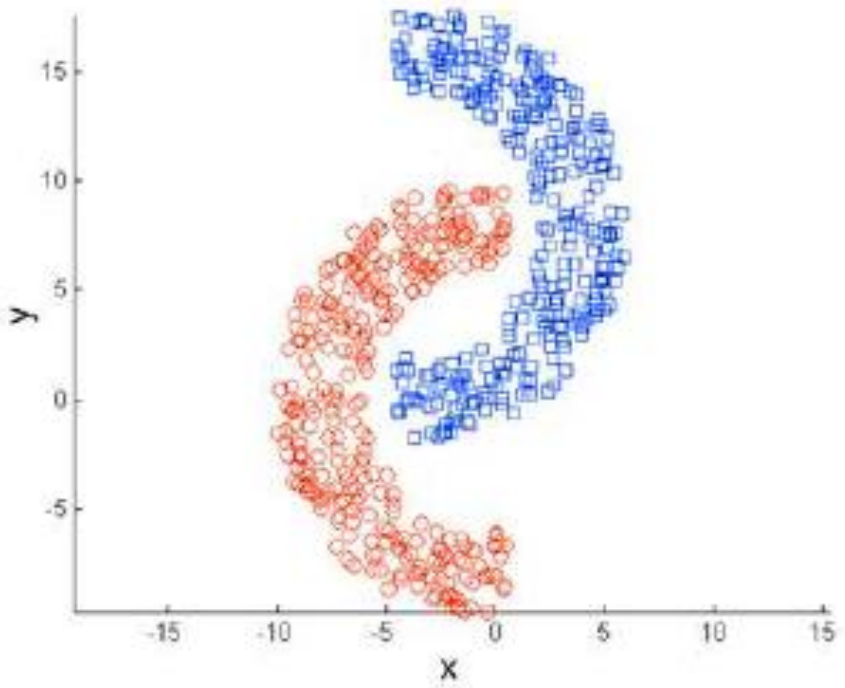


Punctele initiale

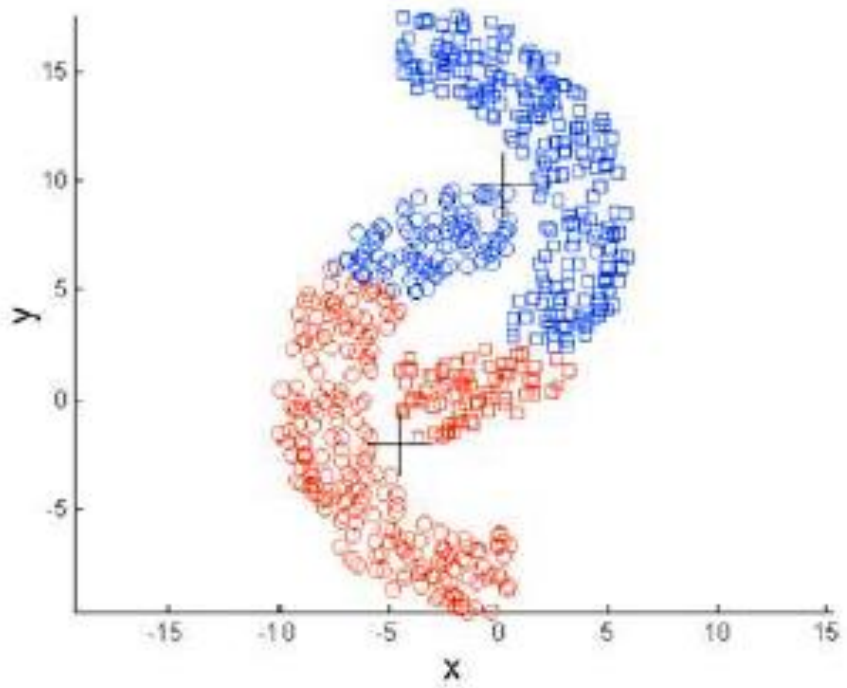


K-means (3 clustere)

Limitări ale K-means: clustere neglobulare



Punctele initiale



K-means (2 clustere)

K-means: puncte tari şi slabe

- Puncte tari:
  - simplu de implementat
  - eficient: se opreşte după puţine iteraţii
  - bisecting K-means, K-mean++: puţin sensibile la problema iniţializării clusterelor
- Puncte slabe:
  - nu poate manipula date ce prezintă grupuri non-globulare, de dimensiuni sau densităţi diferite
  - probleme la datele care conţin outliers
  - algoritmul e restricţionat la datele pentru care noţiunea de centroid are sens

Cuprins

- Generalități
- Algoritmul K-means
- Clustering ierarhic aglomerativ

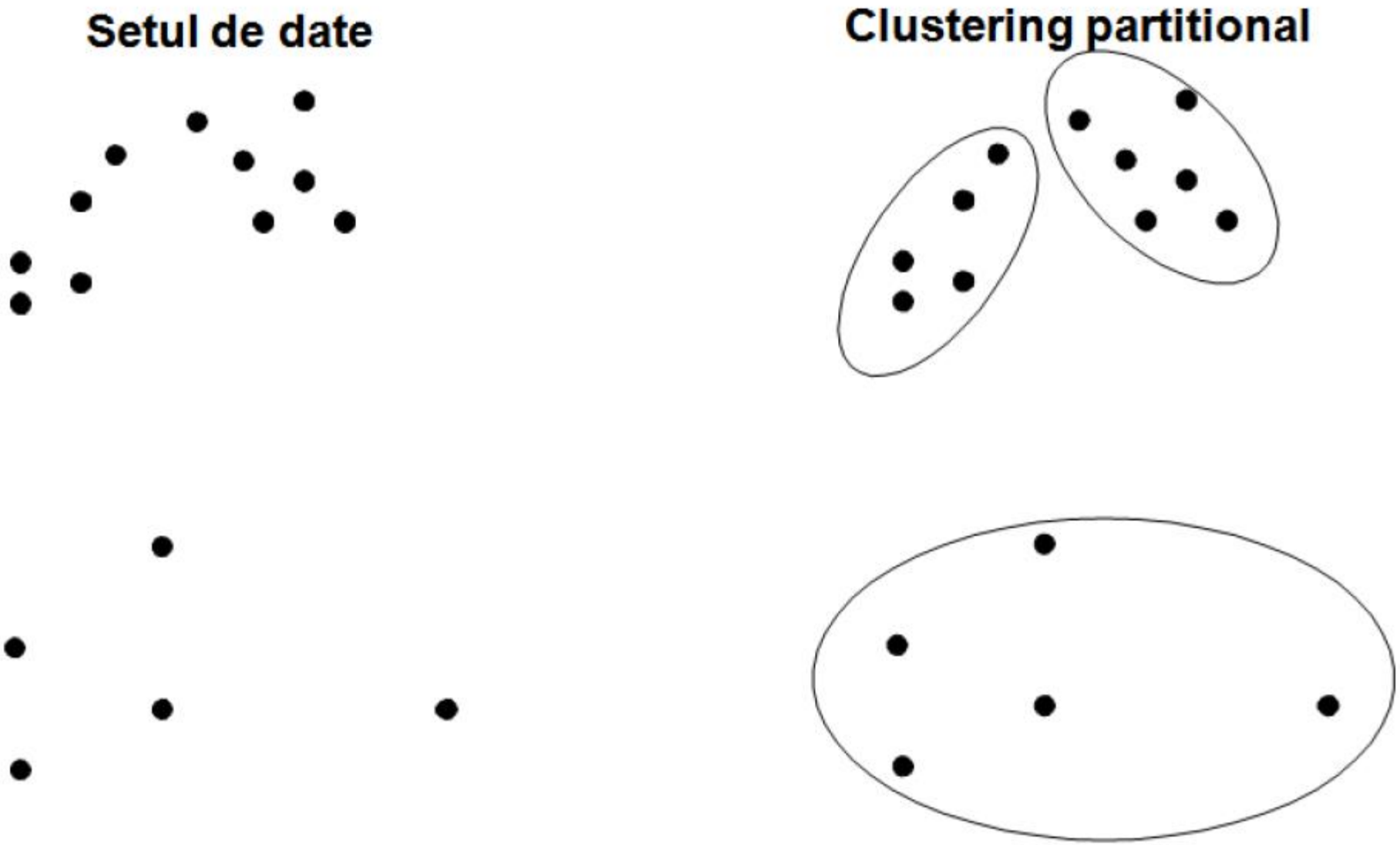
Tipuri de clustering

- Trei atribute ale unui proces de clustering:
  1. ierarhic vs. partițional
  2. exclusiv vs. cu suprapuneri vs. fuzzy
  3. complet vs. parțial

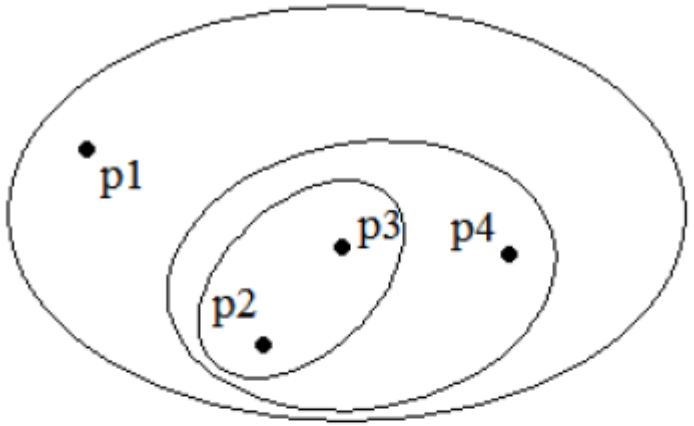
Clustering ierarhic vs. partiţional

- Gruparea poate fi cu imbricări (ierarhică) sau nu (partițională)
- Clustering partițional: datele sunt împărțite în subseturi fără suprapuneri; fiecare dată face parte din exact un cluster
- Dacă permitem ca un cluster să aibă subclustere $\Rightarrow$ clustering ierarhic
  - fiecare nod intern este reuniunea clusterelor reprezentate de nodurile copil
  - se poate ajunge ca nodurile frunză să conțină exact o înregistrare
  - o partiționare ierarhică poate fi văzută ca o secvență de partiționări aplicată pornind de la mulțimea originară a datelor și grupând succesiv fiecare partiție în parte; partiționarea poate fi făcută până când se ajunge la noduri cu un singur element sau se poate face retezare

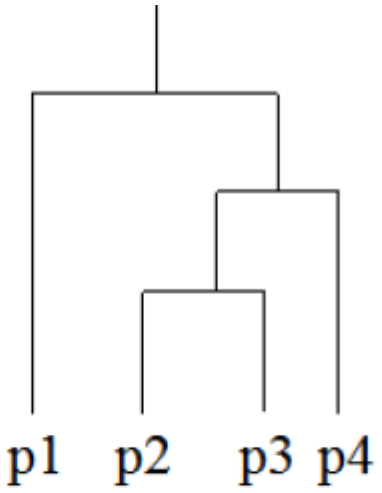
Clustering partiţional



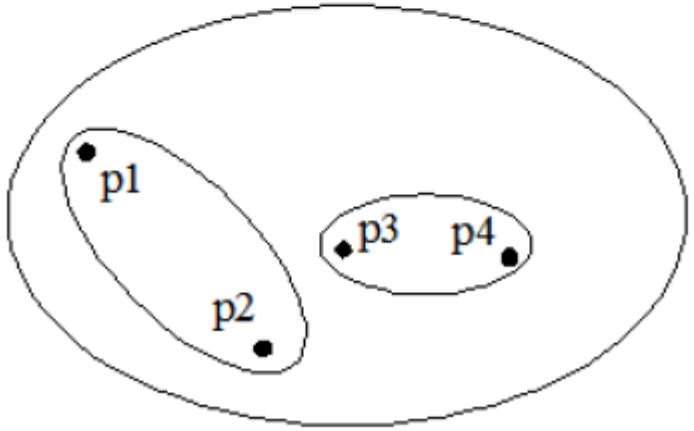
Clustering ierarhic



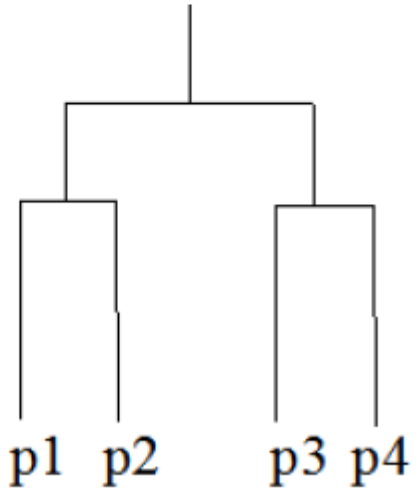
Clustering ierarhic traditional



Dendrograma traditionala



Clustering ierarhic netraditional



Dendrograma netraditionala

Clustering excluziv vs. cu suprapuneri vs. fuzzy

- Excluziv vs non-excluziv:
  - excluziv: un punct aparține unui singur cluster
  - non-exclusiv (eng: overlapping): un punct poate fi asociat mai multor clustere
  - utilitatea clusterelor ne–exclusive: pot reprezenta puncte ce aparțin simultan mai multor clase sau puncte apropiate de zona de separare
- Fuzzy clustering: fiecare obiect are o măsură fuzzy de apartenență la fiecare cluster
  - clusterele devin mulțimi fuzzy
  - clustere fuzzy pot fi convertite la unele de tip exclusiv prin alegerea pentru fiecare dată a acelui cluster pentru care măsura fuzzy este mai mare

Clustering parţial vs. complet

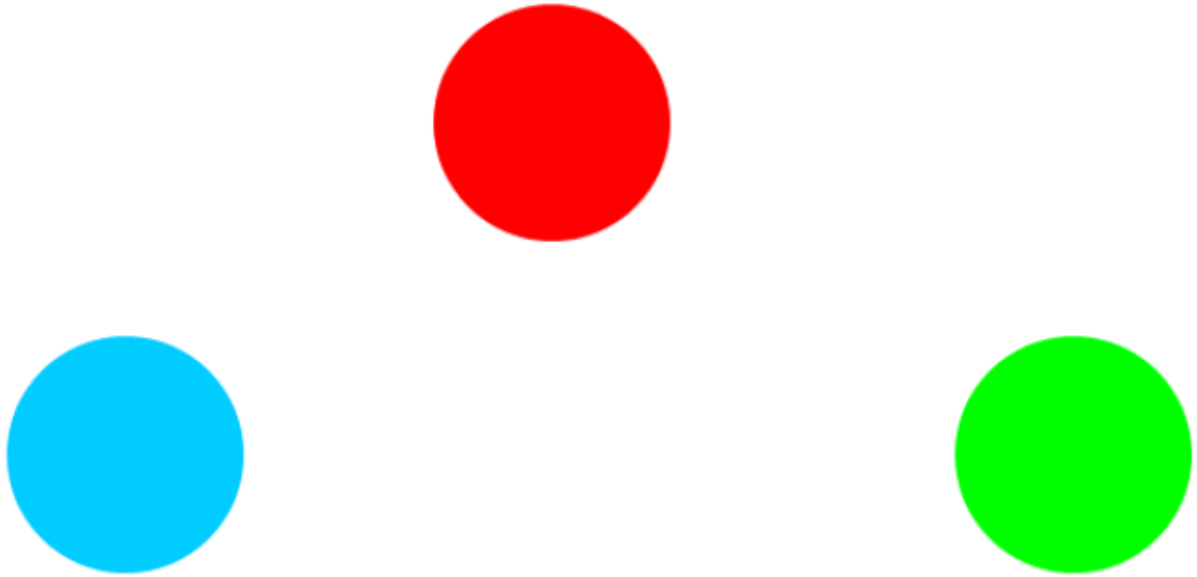
- Clustering complet: fiecare dată este asociată unui cluster
- Clustering parțial: pentru un obiect se poate să nu avem asignare la un cluster

Tipuri de clustere

- Bine separate
- Bazate pe prototipuri
- Bazate pe grafuri
- Bazate pe densitate
- Conceptuale

Clustere bine separate

- În această viziune: un cluster este un set de puncte astfel încât orice punct din cluster este mai aproape (sau mai similar) față de orice punct din acel cluster decât de puncte din afara lui
- Definiția e respectată doar când datele sunt grupate grupate natural în clustere care sunt depărtate unele de altele



3 clustere bine separate

Clustere bazate pe prototipuri

- În această viziune: un cluster e o mulțime cu proprietatea că un obiect din cluster este mai apropiat/similar de/cu prototipul clusterului decât cu alte prototipuri
- Ca prototip: centroid sau medoid
  - medoid — punct din setul de date inițial care e cel mai reprezentativ pentru cluster
  - centroid — *e.g.* centru de greutate; valoarea lui poate să nu coincidă cu niciunul din punctele din centroid

4 clustere bazate pe centri

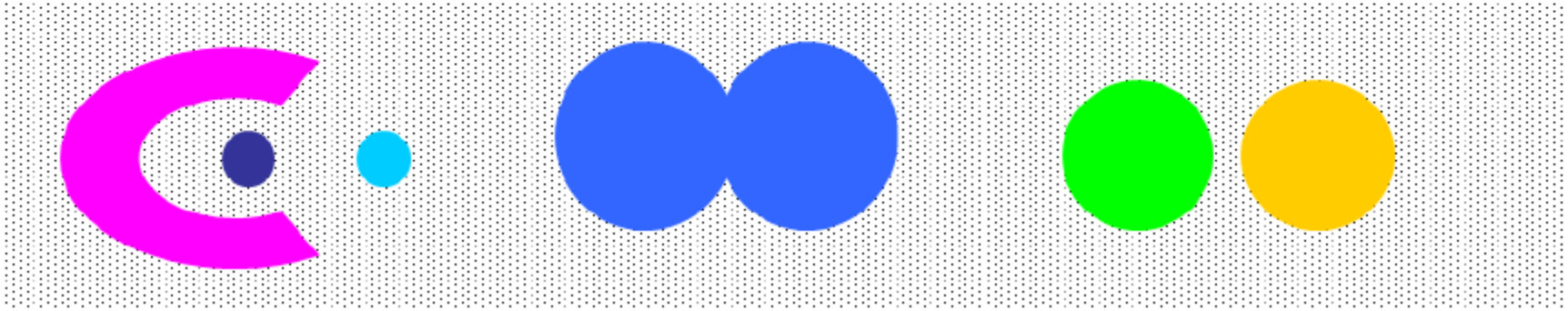
Clustere bazate pe grafuri

- Util pentru date reprezentate ca graf; un cluster este un grup de obiecte conectate
- Exemplu: clustere bazate pe contiguitate: două date sunt conectate doar dacă se află la o distanță mai mică decât un prag impus
- Posibile probleme: zgomotul poate crește clusterele în mod artificial - vezi clusterul cu albastru deschis

8 clustere contigue

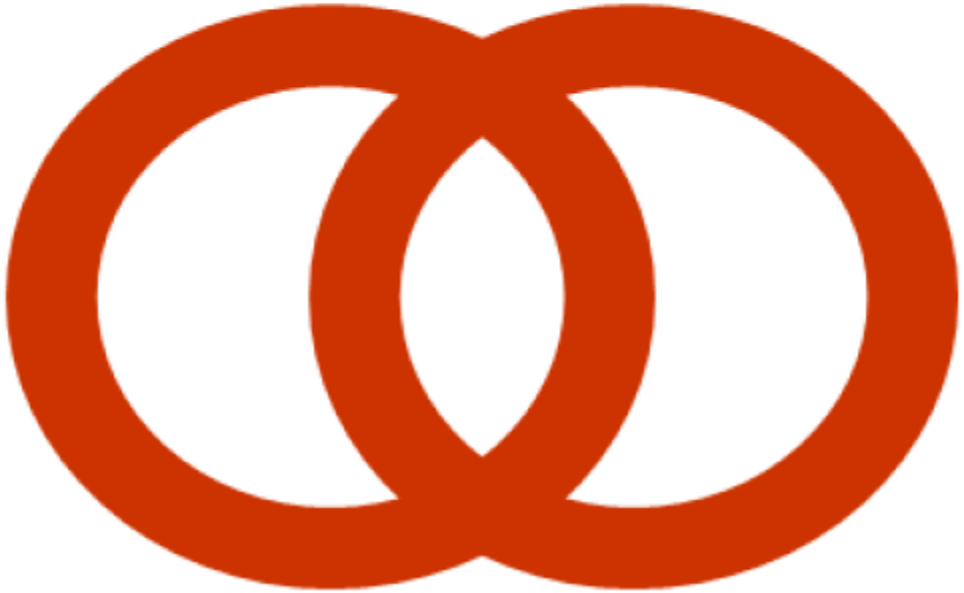
Clustere bazate pe densitate

- În această viziune: un cluster este o zonă densă înconjurată de zonă de densitate mică
- Zonele de densitate mică separă pe cele cu densitate mare
- Utilizate când clusterele sunt neregulate sau se înconjoară unele pe altele sau când zgomotul și datele outlier sunt prezente



Clustere conceptuale

- Clusterul este văzut ca un set de obiecte care partajează o proprietate ce reiese din mulțimea de puncte din cluster
- Un algoritm de clustering de acest tip ar trebui să detecteze concepte



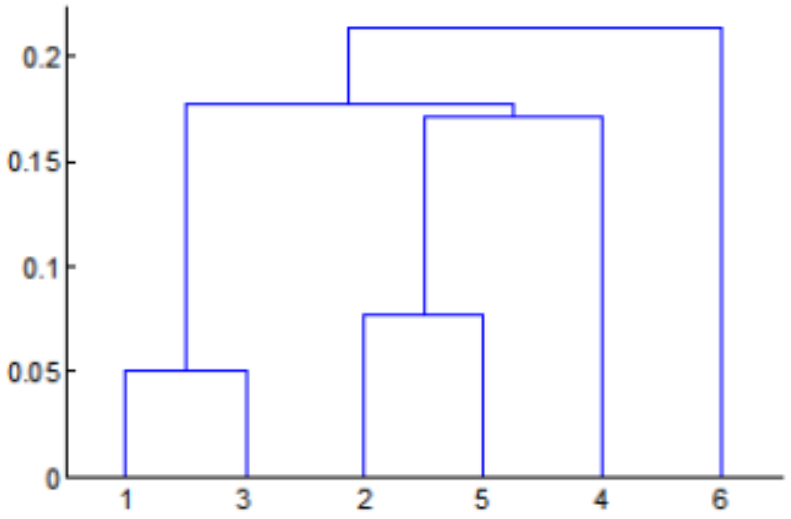
2 clustere sub forma de cercuri

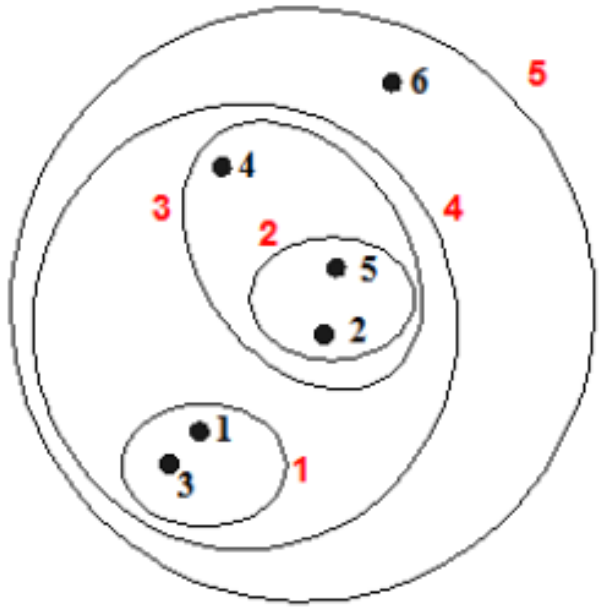
Cuprins

- Generalități
- Algoritmul K-means

Clustering ierarhic

- Produce un set de clustere imbricate organizate ca un arbore
- Clusterele pot fi vizualizate sub forma unei dendrograme





Clustering ierarhic

- Nu trebuie făcute presupuneri asupra numărului de clustere
  - se poate obţine orice număr de clustere prin retezarea dendrogramei la nivelul dorit
- Pot corespunde unei taxonomii
  - exemplu: taxonomia din ştiinţele biologice

Clustering ierarhic

- Două tipuri de clustering ierarhic
- **Aglomerativ:**
  - se porneşte cu fiecare punct ca şi cluster
  - la fiecare pas se uneşte cea mai apropiată pereche de clustere pană răman *k* clustere
- **Diviziv:**
  - se pleacă cu un cluster ce conţine toate punctele
  - la fiecare pas se divizează un cluster
  - se continuă procesul pană cand fiecare cluster conţine un punct, sau se face retezare când se ajunge la *k* clustere

Clustering ierarhic aglomerativ

- Tehnica este populară și cu vechime

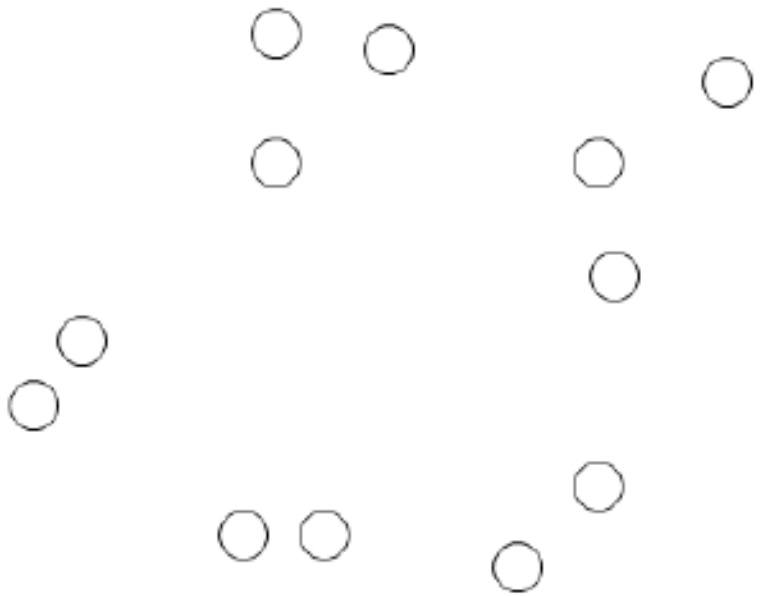
Algorithm 8.3 Basic agglomerative hierarchical clustering algorithm.

1: Compute the proximity matrix, if necessary.
2: **repeat**
3: Merge the closest two clusters.
4: Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
5: **until** Only one cluster remains.

- Operația esențială este calculul proximității a două clustere
- Diferențele între algoritmii ierahici aglomerativi sunt date de modul de calcul al proximității

Clustering ierarhic aglomerativ: pornirea procesului

- Se pornește cu clustere formate din câte un punct

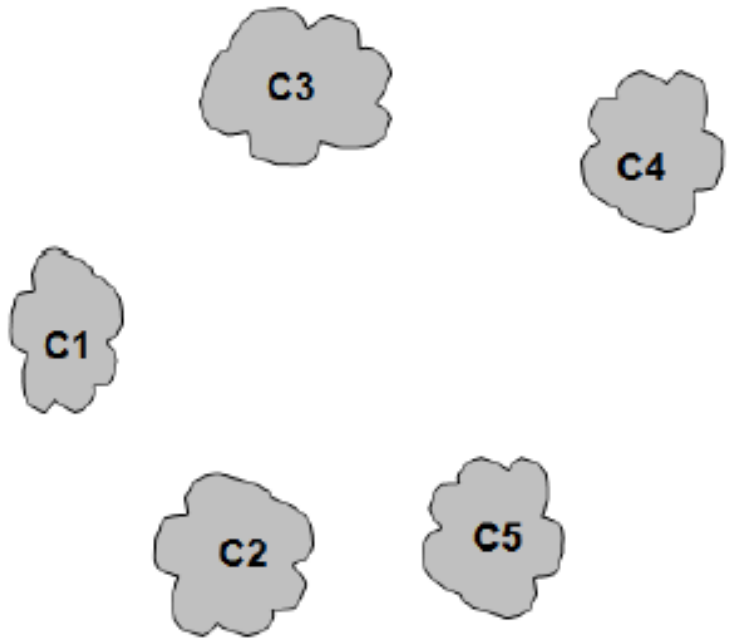


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						

Matrice de proximitate

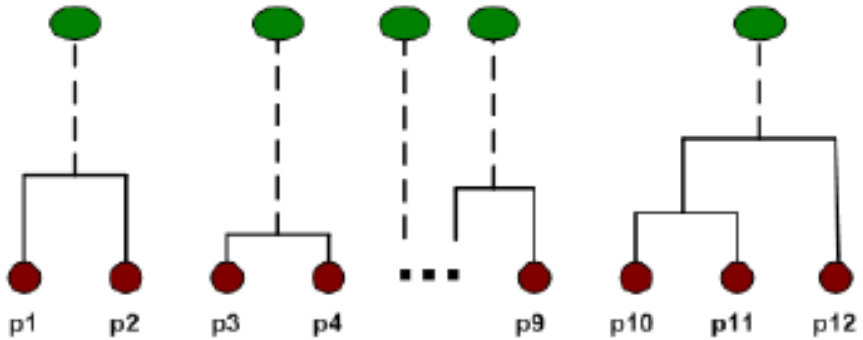
Clustering ierarhic aglomerativ: starea intermediară

- După câția pași avem câteva clustere



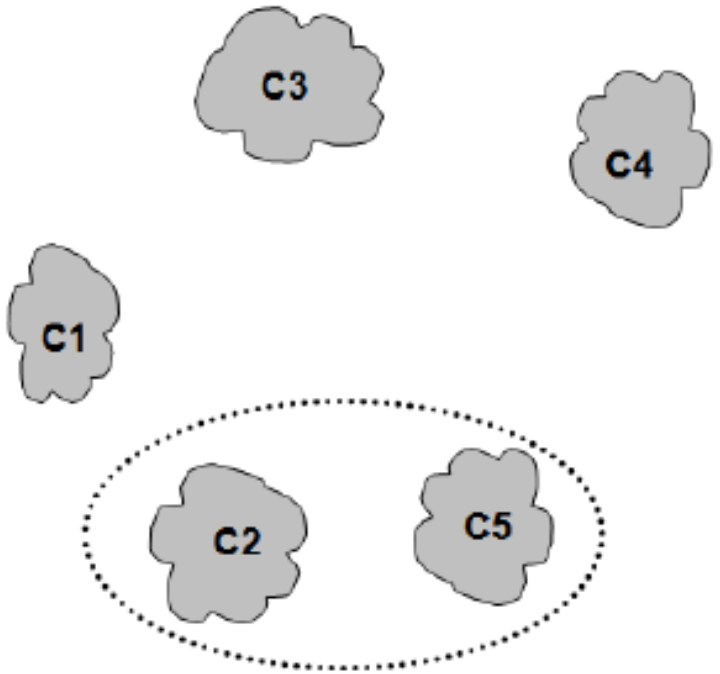
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

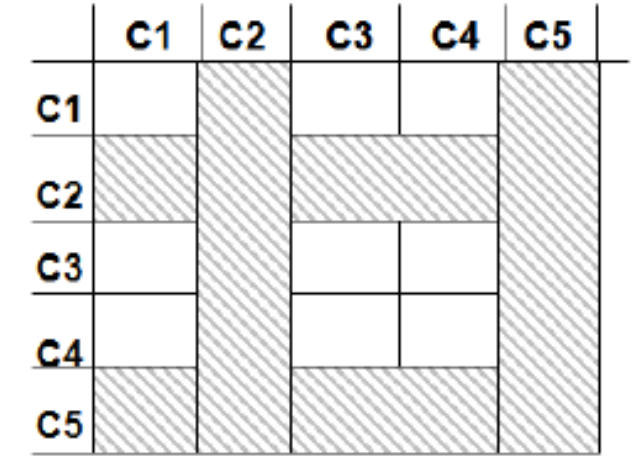
Matrice de proximitate



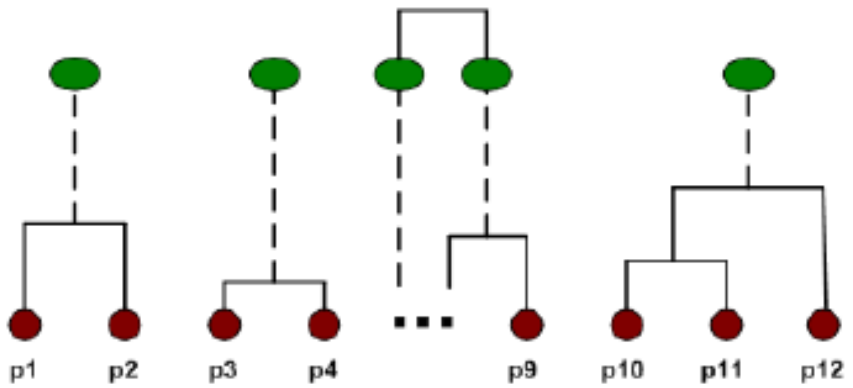
Clustering ierarhic aglomerativ: unirea de 2 clustere

- Unim cele mai apropiate două clustere $C2$ și $C5$ și modificăm matricea de proximitate





Matrice de proximitate



Materiale de citit

- **Capitolul 10** din cartea: **An Introduction to Statistical Learning with Applications in R**. Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani. Springer-Verlag, 2013. (carte disponibilă gratuit online aici: http://www-bcf.usc.edu/~gareth/ISL/)

- Informaţiile prezentate au fost colectate din diferite surse de pe internet, precum și din slide-urile de Data minining ale prof. Lucian Sasu, Univ. Transilvania din Brasov, disponibile pe internet.