INTELIGENȚĂ ARTIFICIALĂ

Sisteme inteligente

Sisteme care învaţă singure

- K-means -

Laura Dioşan

Sumar

A. Scurtă introducere în Inteligenţa Artificială (IA)

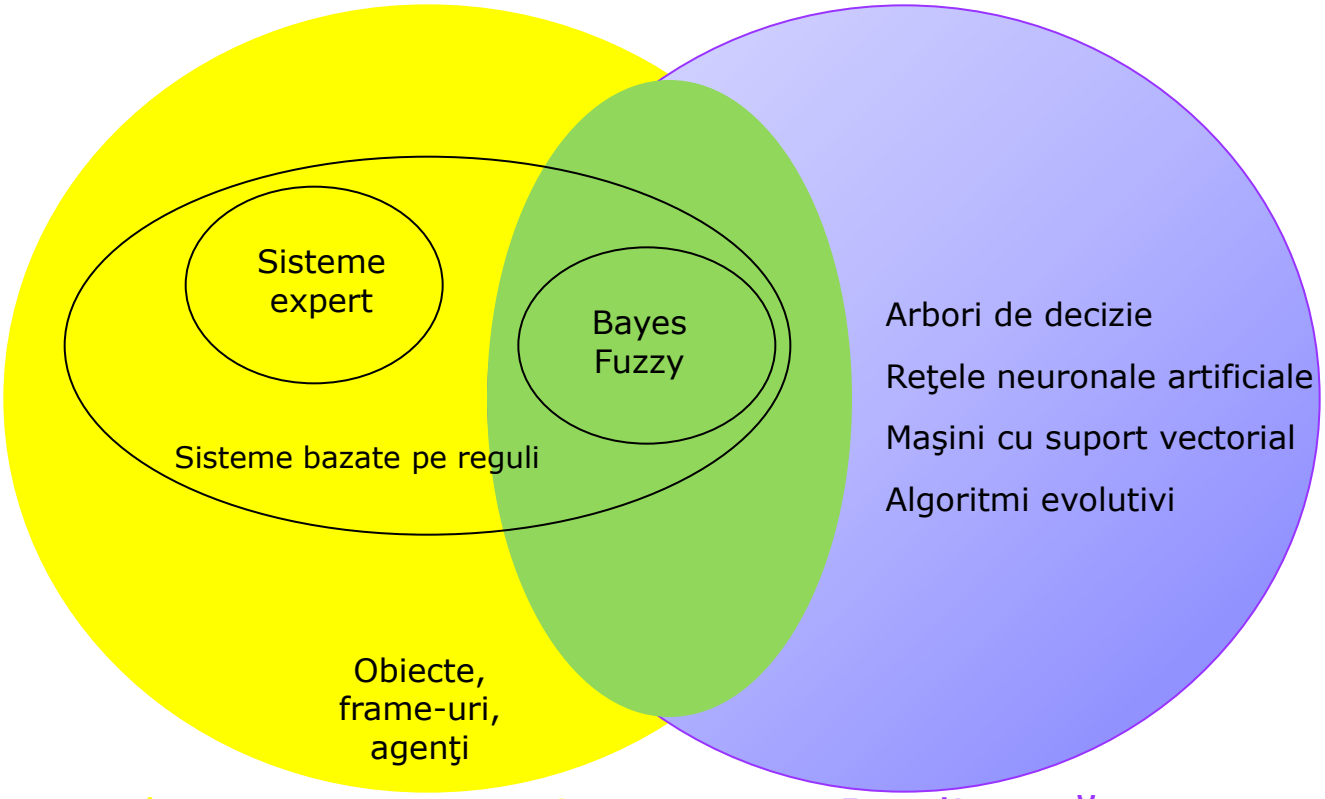
B. Rezolvarea problemelor prin căutare
- Definirea problemelor de căutare
- Strategii de căutare
  - Strategii de căutare neinformate
  - Strategii de căutare informate
  - Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
  - Strategii de căutare adversială

C. Sisteme inteligente
- Sisteme care învaţă singure
  - Arbori de decizie
  - Reţele neuronale artificiale
  - Algoritmi evolutivi
  - Algoritmi de clusterizare
- Sisteme bazate pe reguli
- Sisteme hibride

Materiale de citit şi legături utile

- capitolul 15 din *C. Groşan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011*
- Capitolul 9 din *T. M. Mitchell, Machine Learning, McGraw-Hill Science, 1997*
- Documentele din directorul *svm*

Sisteme inteligente



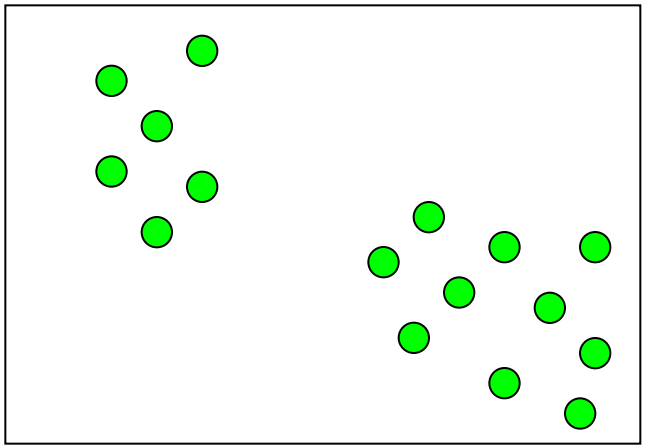
Sisteme inteligente – SIS – Învățare automată

- Tipologie
  - În funcţie de experienţa acumulată în timpul învăţării:
    - SI cu învăţare supervizată
    - **SI cu învăţare nesupervizată**
    - SI cu învăţare activă
    - SI cu învăţare cu întărire
  - În funcţie de modelul învăţat (algoritmul de învăţare):
    - Arbori de decizie
    - Reţele neuronale artificiale
    - Maşini cu suport vectorial (MSV)
    - Algoritmi evolutivi
    - Modele Markov ascunse
    - **K-means**

Învățare nesupervizată

- Scop
  - Găsirea unui model sau a unei structuri utile a datelor

- Tip de probleme
  - Identificara unor grupuri (clusteri)
    - Analiza genelor
    - Procesarea imaginilor
    - Analiza reţelelor sociale
    - Segmentarea pieţei
    - Analiza datelor astronomice
    - Clusteri de calculatoare
  - Reducerea dimensiunii
  - Identificarea unor cauze (explicaţii) ale datelor
  - Modelarea densităţii datelor



- Caracteristic
  - Datele nu sunt adnotate (etichetate)

Învățare ne-supervizată – definire

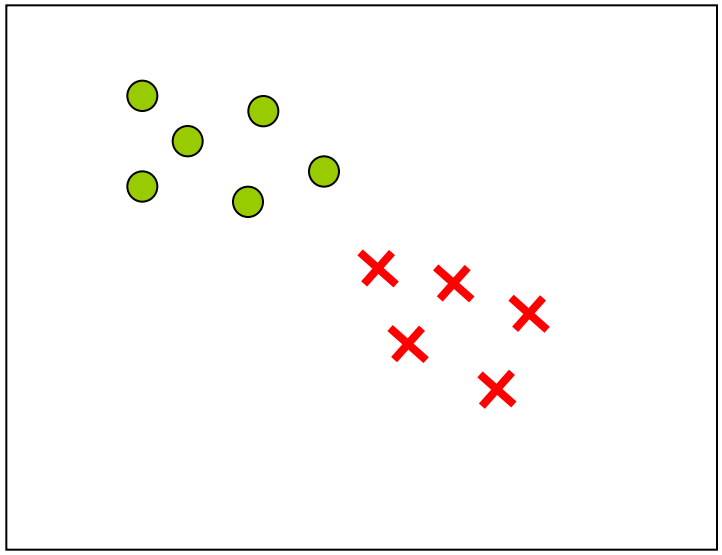
Împărţirea unor exemple neetichetate în submulţimi disjuncte (clusteri) astfel încât:
- exemplele din acelaşi cluster sunt foarte similare
- exemplele din clusteri diferiţi sunt foarte diferite

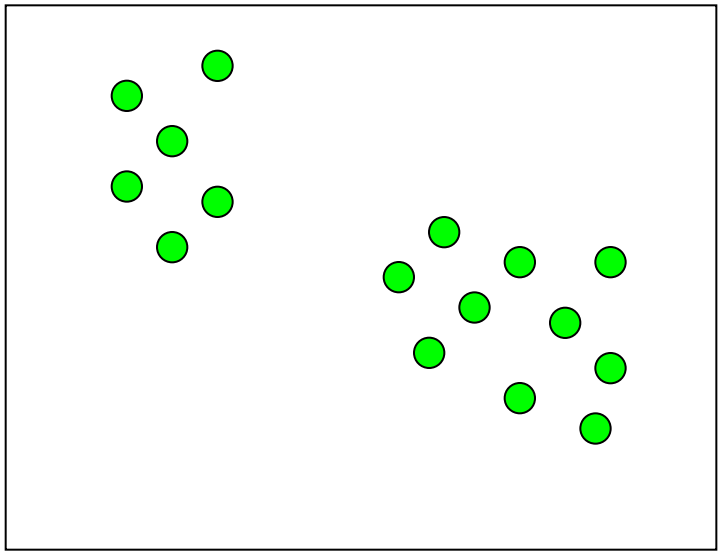
Definire
- Se dă
  - un set de date (exemple, instanţe, cazuri)
    - Date de antrenament
      - Sub forma $\boldsymbol{atribute\_data_i}$, unde
        - $i = 1,N$ (N = nr datelor de antrenament)
        - $\boldsymbol{atribute\_data_i} = (atr_{i1}, atr_{i2}, ..., atr_{im})$, m – nr atributelor (caracteristicilor, proprietăţilor) unei date
    - Date de test
      - Sub forma ($\boldsymbol{atribute\_data_i}$), $i = 1,n$ (n = nr datelor de test)
- Se determină
  - o funcţie (necunoscută) care realizează gruparea datelor de antrenament în mai multe clase
    - Nr de clase poate fi pre-definit (k) sau necunoscut
    - Datele dintr-o clasă sunt asemănătoare
  - clasa asociată unei date (noi) de test folosind gruparea învăţată pe datele de antrenament

Alte denumiri
- Clustering

Învățare ne-supervizată – definire

- Supervizată *vs.* Ne-supervizată





Învățare ne-supervizată – definire

- Distanţe între 2 elemente ***p*** şi ***q*** $\in R^m$
  - Euclideana
    - $d(\boldsymbol{p},\boldsymbol{q})=sqrt(\Sigma_{j=1,2,...,m}(p_j-q_j)^2)$
  - Manhattan
    - $d(\boldsymbol{p},\boldsymbol{q})=\Sigma_{j=1,2,...,m}|p_j-q_j|$
  - Mahalanobis
    - $d(\boldsymbol{p},\boldsymbol{q})=sqrt(\boldsymbol{p}-\boldsymbol{q})S^{-1}(\boldsymbol{p}-\boldsymbol{q})),$
      - unde *S* este matricea de variaţie şi covariaţie ($S= E[(\boldsymbol{p}-E[\boldsymbol{p}])(\boldsymbol{q}-E[\boldsymbol{q}])]$)
  - Produsul intern
    - $d(\boldsymbol{p},\boldsymbol{q})=\Sigma_{j=1,2,...,m}p_jq_j$
  - Cosine
    - $d(\boldsymbol{p},\boldsymbol{q})=\Sigma_{j=1,2,...,m}p_jq_j / (sqrt(\Sigma_{j=1,2,...,m}p_j^2) * sqrt(\Sigma_{j=1,2,...,m}q_j^2))$
  - Hamming
    - numărul de diferenţe între ***p*** şi ***q***
  - Levenshtein
    - numărul minim de operaţii necesare pentru a-l transforma pe ***p*** în ***q***

- Distanţă *vs.* Similaritate
  - Distanţa → min
  - Similaritatea → max

Învățare ne-supervizată – exemple

- Gruparea genelor
- Studii de piaţă pentru gruparea clienţilor (segmentarea pieţei)
- news.google.com

Învățare ne-supervizată – proces

Procesul

- 2 paşi:
  - Antrenarea
    - Învăţarea (determinarea), cu ajutorul unui algoritm, a clusterilor existenţi
  - Testarea
    - Plasarea unei noi date într-unul din clusterii identificaţi în etapa de antrenament

Calitatea învăţării (validarea clusterizării):

- Criterii interne
  - Similaritate ridicată în interiorul unui cluster şi similaritate redusă între clusteri
- Criteri externe
  - Folosirea unor benchmark-uri formate din date pre-grupate

Învățare ne-supervizată – evaluare

Măsuri de performanţă

- Criterii interne
  - Distanţa în interiorul clusterului
  - Distanţa între clusteri
  - Indexul Davies-Bouldin
  - Indexul Dunn

- Criteri externe
  - Compararea cu date cunoscute – în practică este imposibil
  - Precizia
  - Rapelul
  - F-measure

Învățare ne-supervizată – evaluare

Măsuri de performanţă

- Criterii interne
  - Distanţa în interiorul clusterului c_j care conţine n_j instanţe
    - Distanţa medie între instanţe (average distance)
      - $D_a(c_j) = \sum_{xi1,xi2 \epsilon cj} ||x_{i1} - x_{i2}|| / (n_j(n_j-1))$
    - Distanţa între cei mai apropiaţi vecini (nearest neighbour distance)
      - $D_{nn}(c_j) = \sum_{xi1 \epsilon cj} min_{xi2 \epsilon cj} ||x_{i1} - x_{i2}|| / n_j$
    - Distanţa între centroizi
      - $D_c(c_j) = \sum_{xi, \epsilon cj} ||x_i - \mu_j|| / n_j$, $unde\ \mu_j = 1/n_j \sum_{xi \epsilon cj} x_i$

Învățare ne-supervizată – evaluare

Măsuri de performanţă

- Criterii interne
  - Distanţa între 2 clusteri c_{j1} şi c_{j2}
    - Legătură simplă
      - $d_s(c_{j1}, c_{j2}) = min_{xi1 \epsilon cj1,\, xi2 \epsilon cj2} \{||x_{i1} - x_{i2}||\}$
    - Legătură completă
      - $d_{co}(c_{j1}, c_{j2}) = max_{xi1 \epsilon cj1,\, xi2 \epsilon cj2} \{||x_{i1} - x_{i2}||\}$
    - Legătură medie
      - $d_a(c_{j1}, c_{j2}) = \Sigma_{xi1 \epsilon cj1,\, xi2 \epsilon cj2} \{||x_{i1} - x_{i2}||\} / (n_{j1} * n_{j2})$
    - Legătură între centroizi
      - $d_{ce}(c_{j1}, c_{j2}) = ||\mu_{j1} - \mu_{j2}||$

Învățare ne-supervizată – evaluare

Măsuri de performanţă

- Criterii interne
  - Indexul Davies-Bouldin → min ➔ clusteri compacţi
    - $DB = 1/nc * \sum_{i=1,2,...,nc} max_{j=1, 2, ..., nc, j \neq i}((\sigma_i + \sigma_j)/d(\mu_i, \mu_j))$
    - unde:
      - *nc* – numărul de clusteri
      - μ_i – centroidul clusterului *i*
      - σ_i – media distanţelor între elementele din clusterul *i* şi centroidul μ_i
      - $d(\mu_i, \mu_j)$ – distanţa între centroidul μ_i şi centroidul μ_j
  - Indexul Dunn
    - Identifică clusterii denşi şi bine separaţi
    - $D=d_{min}/d_{max}$
    - Unde:
      - d_{min} – distanţa minimă între 2 obiecte din clusteri diferiţi – distanţa intra-cluster
      - d_{max} – distanţa maximă între 2 obiecte din acelaşi cluster – distanţa inter-cluster

Învățare ne-supervizată - tipologie

- După modul de formare al clusterilor
  - C. ierarhic
  - C. ne-ierarhic (partiţional)
  - C. bazat pe densitatea datelor
  - C. bazat pe un grid

Învățare ne-supervizată - tipologie

- După modul de formare al clusterilor
  - Ierarhic
    - se crează un arbore taxonomic (dendogramă)
      - crearea clusterilor (recursiv)
      - nu se cunoaşte *k* (nr de clusteri)
    - aglomerativ (de jos în sus) → clusteri mici spre clusteri mari
    - diviziv (de sus în jos) → clusteri mari spre clusteri mici
    - Ex. Clustering ierarhic aglomerativ

Învățare ne-supervizată - tipologie

- După modul de formare al clusterilor
  - Ne-ierarhic
    - Partiţional → se determină o împărţire a datelor → toţi clusterii deodată
    - Optimizează o funcţie obiectiv definită
      - Local – doar pe anumite atribute
      - Global – pe toate atributele

      care poate fi
      - Pătratul erorii – suma patratelor distanţelor între date şi centroizii clusterilor → min
        - Ex. *K-means*
      - Bazată pe grafuri
        - Ex. Clusterizare bazată pe arborele minim de acoperire
      - Bazată pe modele probabilistice
        - Ex. Identificarea distribuţiei datelor → Maximizarea aşteptărilor
      - Bazată pe cel mai apropiat vecin

    - Necesită fixarea *apriori* a lui k → fixarea clusterilor iniţiali
      - Algoritmii se rulează de mai multe ori cu diferiţi parametri şi se alege versiunea cea mai eficientă
    - Ex. *K-means, ACO*

Învățare ne-supervizată - tipologie

- După modul de formare al clusterilor
  - bazat pe densitatea datelor
    - Densitatea şi conectivitatea datelor
      - Formarea clusterilor de bazează pe densitatea datelor într-o anumită regiune
      - Formarea clusterilor de bazează pe conectivitatea datelor dintr-o anumită regiune
    - Funcţia de densitate a datelor
      - Se încearcă modelarea legii de distribuţie a datelor
    - Avantaj:
      - Modelarea unor clusteri de orice formă

Învățare ne-supervizată - tipologie

- După modul de formare al clusterilor
  - Bazat pe un grid
    - Nu e chiar o metodă nouă de lucru
      - Poate fi ierarhic, partiţional sau bazat pe densitate
    - Pp. segmentarea spaţiului de date în zone regulate
    - Obiectele se plasează pe un grid multi-dimensional
    - Ex. ACO

Învățare ne-supervizată - tipologie

- După modul de lucru al algoritmului
  - Aglomerativ
    1. Fiecare instanţă formează iniţial un cluster
    2. Se calculează distanţele între oricare 2 clusteri
    3. Se reunesc cei mai apropiaţi 2 clusteri
    4. Se repetă paşii 2 şi 3 până se ajunge la un singur cluster sau la un alt criteriu de stop
  - Diviziv
    1. Se stabileşte numărul de clusteri (*k*)
    2. Se iniţializează centrii fiecărui cluster
    3. Se determină o împărţire a datelor
    4. Se recalculează centrii clusterilor
    5. Se reptă pasul 3 şi 4 până partiţionarea nu se mai schimbă (algoritmul a convers)

- După atributele considerate
  - Monotetic – atributele se consideră pe rând
  - Politetic – atributele se consideră simultan

Învățare ne-supervizată - tipologie

- După tipul de apartenenţă al datelor la clusteri
  - Clustering exact (*hard clustering*)
    - Asociază fiecarei intrări $\boldsymbol{x_i}$ o etichetă (clasă) c_j
  - Clustering fuzzy
    - Asociază fiecarei intrări $\boldsymbol{x_i}$ un grad (probabilitate) de apartenenţă f_{ij} la o anumită clasă c_j ➔ o instanţă $\boldsymbol{x_i}$ poate aparţine mai multor clusteri

Învățare ne-supervizată – algoritmi

- Clustering ierarhic aglomerativ
- K-means
- AMA
- Modele probabilistice
- Cel mai apropiat vecin
- Fuzzy
- Reţele neuronale artificiale
- Algoritmi evolutivi
- ACO

Învățare ne-supervizată – algoritmi
Clustering ierarhic aglomerativ

- Se consideră o distanţă între 2 instanţe $d(x_{i1}, x_{i2})$
- Se formează *N* clusteri, fiecare conţinând câte o instanţă
- Se repetă
  - Determinarea celor mai apropiaţi 2 clusteri
  - Se reunesc cei 2 clusteri ➔ un singur cluster
- Până când se ajunge la un singur cluster (care conţine toate instanţele)

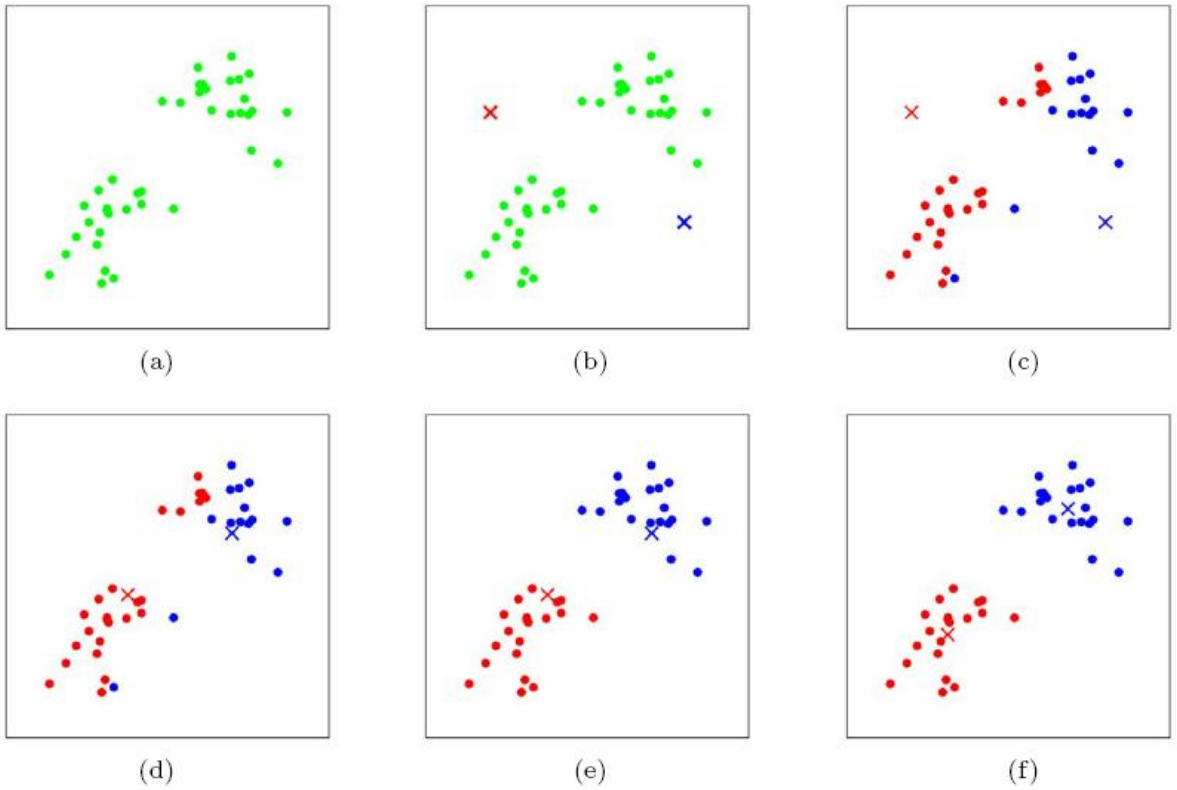
Învățare ne-supervizată – algoritmi
Clustering ierarhic aglomerativ

- Distanţa între 2 clusteri c_i şi c_j:
  - Legătură simplă → minimul distanţei între obiectele din cei 2 clusteri
    - $d(c_i, c_j) = max_{xi1 \epsilon ci,\ xi2 \epsilon cj}\ sim(\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2})$
  - Legătură completă → maximul distanţei între obiectele din cei 2 clusteri
    - $d(c_i, c_j) = min_{xi1 \epsilon ci,\ xi2 \epsilon cj}\ sim(\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2})$
  - Legătură medie → media distanţei între obiectele din cei 2 clusteri
    - $d(c_i, c_j) = 1 / (n_i * n_j) \sum_{xi1 \epsilon ci} \sum_{xi2 \epsilon cj} d(\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2})$
  - Legătură medie peste grup → distanţa între mediile (centroizii) celor 2 clusteri
    - $d(c_i, c_j) = \rho(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$, ρ – distanţă, $\boldsymbol{\mu}_j = 1/n_j \sum_{xi \epsilon cj} \boldsymbol{x}_i$

Învățare ne-supervizată – algoritmi
K-means (algoritmul Lloyd/iterația Voronoi)

- Pp că se vor forma k clusteri
- Iniţializează *k* centroizi $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$
  - Un centroid $\boldsymbol{\mu}_j$ ($i=1,2, \ldots, k$) este un vector cu *m* valori (*m* – nr de atribute)
- Repetă până la convergenţă
  - Asociază fiecare instanţă celui mai apropiat centroid → pentru fiecare instanţă $\boldsymbol{x}_i$, $i = 1, 2, \ldots, N$
    - $c_i = arg\ min_{j = 1, 2, \ldots, k} ||\boldsymbol{x}_i - \boldsymbol{\mu}_j||^2$
  - Recalculează centroizii prin mutarea lor în media instanţelor asociate fiecăruia → pentru fiecare cluster c_j, $j = 1, 2, \ldots, k$
    - $\boldsymbol{\mu}_j = \Sigma_{i=1,2, \ldots N} 1_{ci=j}\ \boldsymbol{x}_i \ / \ \Sigma_{i=1,2, \ldots N}\ 1_{ci=j}$

Învățare ne-supervizată – algoritmi K-means



Învățare ne-supervizată – algoritmi K-means

- Iniţializarea a *k* centroizi $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$
  - Cu valori generate aleator (în domeniul de definiţie al problemei)
  - Cu *k* dintre cele *N* instanţe (alese în mod aleator)

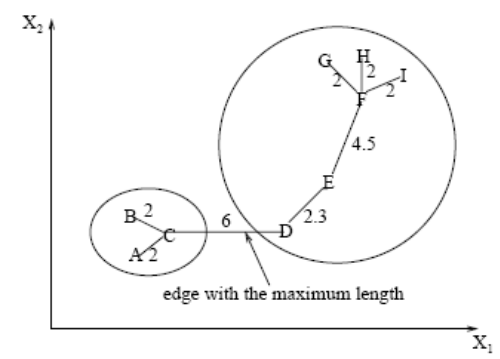
- Algoritmul converge întotdeauna?
  - Da, pt că avem funcţia de distorsiune J
    - $J(c, \mu) = \sum_{i=1,2,\ldots,N} ||\boldsymbol{x}_i - \boldsymbol{\mu}_{cj}||^2$

    care este descrescătoare
  - Converge într-un optim local
  - Găsirea optimului global → NP-dificilă

Învățare ne-supervizată – algoritmi

Clusterizare bazată pe arborele minim de acoperire (AMA)

- Se construieşte AMA al datelor
- Se elimină din arbore cele mai lungi muchii, formându-se clusteri



Învățare ne-supervizată – algoritmi
Modele probabilistice

- http://www.gatsby.ucl.ac.uk/~zoubin/course04/ul.pdf
- http://learning.eng.cam.ac.uk/zoubin/nipstut.pdf

Învățare ne-supervizată – algoritmi
Cel mai apropiat vecin

- Se etichetează câteva dintre instanţe
- Se repetă până la etichetarea tuturor instanţelor
  - O instanţă ne-etichetată va fi inclusă în clusterul instanţei cele mai apropiate
    - dacă distanţa între instanţa neetichetată şi cea etichetată este mai mică decât un prag

Învățare ne-supervizată – algoritmi
Clusterizare fuzzy

- Se stabileşte o partiţionare fuzzy iniţială
  - Se construieşte matricea gradelor de apartenenţă U, unde u_{ij} – gradul de apartenenţă al instanţei $\boldsymbol{x_i}$ ($i=1,2, \ldots, N$) la clusterul c_j ($j = 1, 2, \ldots, k$) ($u_{ij} \in [0,1]$)
    - Cu cât u_{ij} e mai mare, cu atât e mai mare încrederea că instanţa $\boldsymbol{x_i}$ face parte din clusterul c_j

- Se stabileşte o funcţie obiectiv
  - $E^2(U) = \sum_{i=1,2, \ldots, N} \sum_{j=1,2,\ldots,k} u_{ij} ||\boldsymbol{x}_i - \boldsymbol{\mu}_j||^2$,
    - unde $\boldsymbol{\mu}_j = \sum_{i=1,2, \ldots, N} u_{ij} \boldsymbol{x}_i$ – centrul celui de-al j-lea fuzzy cluster
  - care se optimizează (min) prin re-atribuirea instanţelor (în clusteri noi)

- Clusering fuzzy ➔ clusterizare *hard* (fixă)
  - impunerea unui prag funcţiei de apartenenţă u_{ij}

Invățare ne-supervizată – algoritmi
Algoritmi evolutivi

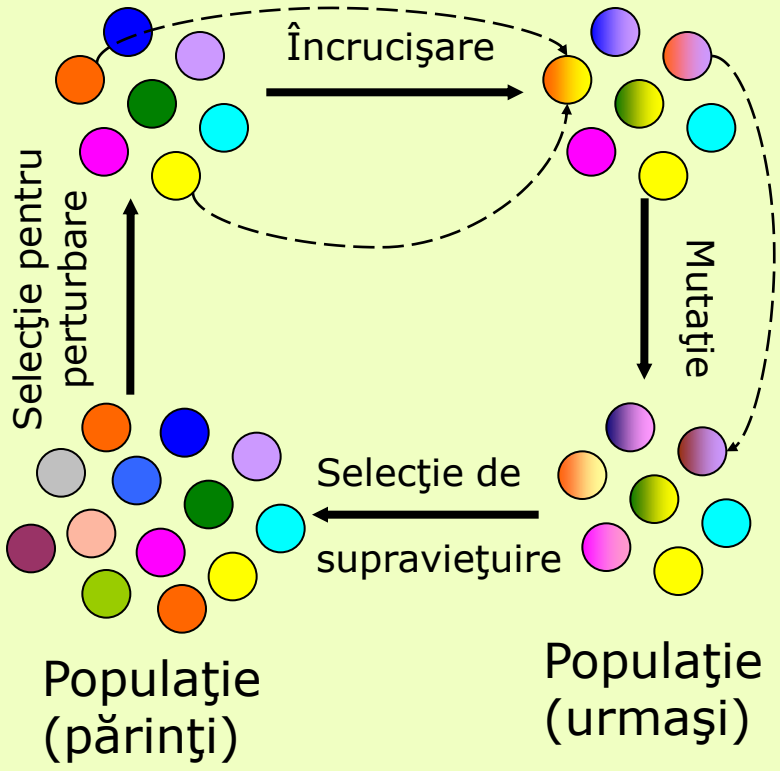
- Algoritmi
  - Inspiraţi din natură (biologie)
  - Iterativi
  - Bazaţi pe
    - populaţii de potenţiale soluţii
    - căutare aleatoare ghidată de
      - Operaţii de selecţie naturală
      - Operaţii de încrucişare şi mutaţie
  - Care procesează în paralel mai multe soluţii
- Metafora evolutivă

Evoluţie naturală	Rezolvarea problemelor
Individ	Soluţie potenţială (candidat)
Populaţie	Mulţime de soluţii
Cromozom	Codarea (reprezentarea) unei soluţii
Genă	Parte a reprezentării
Fitness (măsură de adaptare)	Calitate
Încrucişare şi mutaţie	Operatori de căutare
Mediu	Spaţiul de căutare al problemei

(k-means)

Învățare ne-supervizată – algoritmi
Algoritmi evolutivi

```
Initializare populaţie P(0)
Evaluare P(0)
g := 0; //generaţia
CâtTimp (not condiţie_stop) execută
    Repetă
      Selectează 2 părinţi p1 şi p2 din P(g)
      Încrucişare(p1,p2) =>o1 şi o2
      Mutaţie(o1) => o1*
      Mutaţie(o2) => o2*
      Evaluare(o1*)
      Evaluare(o2*)
      adăugare o1* şi o* în P(g+1)
    Până când P(g+1) este completă
    g := g + 1
Sf CâtTimp
```

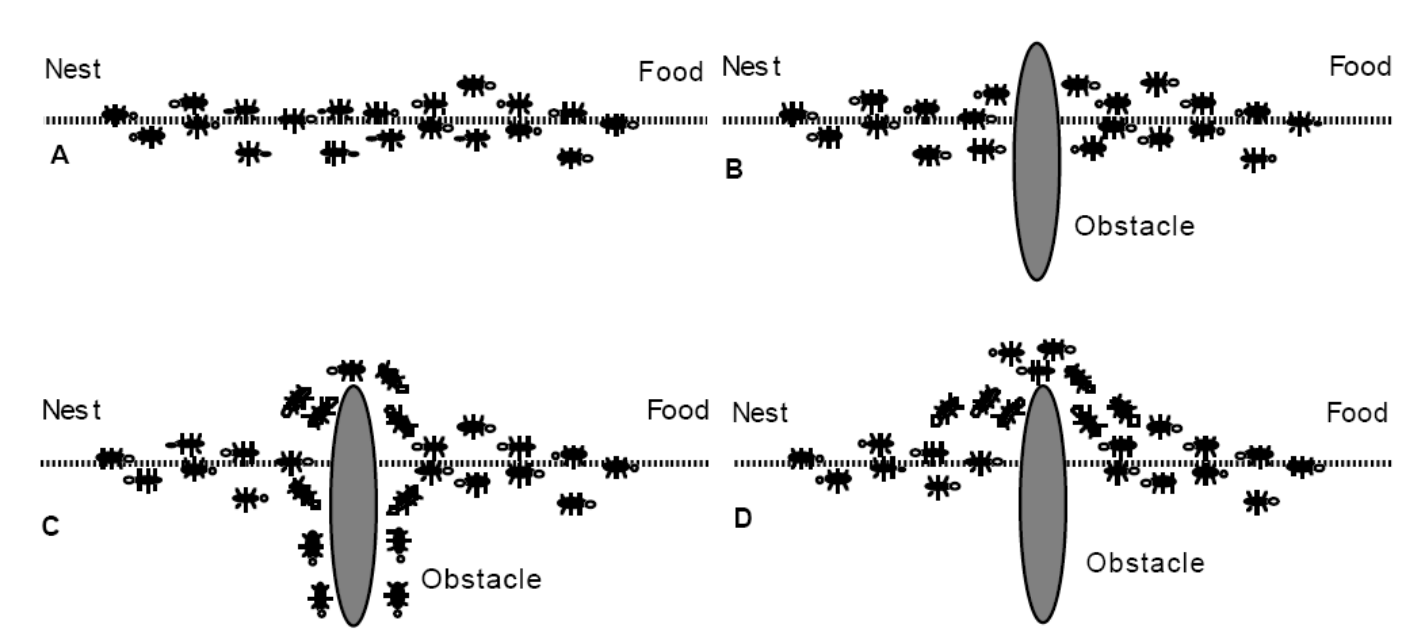


Învățare ne-supervizată – algoritmi
Algoritmi evolutivi

- Reprezentare
  - Cromozomul = o partiţionare a datelor
    - Ex. 2 clusteri ➔ cromozom = vector binar
    - Ex. *K* clusteri ➔ cromozom = vector cu valori din {*1,2,…,k*}
- Fitness
  - Calitatea partiţionării
- Iniţializare
  - Aleatoare
- Încrucişare
  - Punct de tăietură
- Mutaţie
  - Schimbarea unui element din cromozom

Învățare ne-supervizată – algoritmi ACO

- Preferinţa pentru drumuri cu nivel ridicat de feromon
- Pe drumurile scurte feromonul se înmulţeşte
- Furnicile comunică pe baza urmelor de feromon



Învățare ne-supervizată – algoritmi ACO

- Algoritm de clusterizare bazat pe un grid
- Obiectele se plasează aleator pe acest grid, urmând ca furnicuţele să le grupeze în funcţie de asemănarea lor
- 2 reguli pentru furnicuţe
  - Furnica "ridică" un obiect-obstacol
    - Probabilitatea de a-l ridica e cu atât mai mare cu cât obiectul este mai izolat (în apropierea lui nu se află obiecte similare)
    - $p(ridica)=(k^+/(k^++f))^2$
  - Furnica "depune" un obiect (anterior ridicat) într-o locaţie nouă
    - Probabilitatea de a-l depune e cu atât mai mare cu cât în vecinătatea locului de plasare se afla mai multe obiecte asemănătoare
    - $p(depune)=(f/(k^-+f))^2$
  - k^+, k^- - constante
  - f – procentul de obiecte similare cu obiectul curent din memoria furnicuţei
- Furnicuţele
  - au memorie
    - reţin obiectele din vecinătatea poziţiei curente
  - se mişcă ortogonal (N, S, E, V) pe grid pe căsuţele neocupate de alte furnici

Recapitulare

- Sisteme care învaţă singure (SIS)
  - Maşini cu suport vectorial (MSV)
    - Modele computaţionale care
      - rezolvă (în special) probleme de învăţare supervizată
      - prin identificarea celui mai bun hyper-plan de separare a datelor
  - K-means
    - Modele computaţionale care
      - rezolvă probleme de clusterizare
        - → nu se cunosc etichetele claselor
      - prin
        - minimizarea diferenţelor între elementele aceleaşi calse
        - maximizarea diferenţelor între elementele claselor diferite

Cursul următor

A. Scurtă introducere în Inteligenţa Artificială (IA)

B. Rezolvarea problemelor prin căutare
- Definirea problemelor de căutare
- Strategii de căutare
  - Strategii de căutare neinformate
  - Strategii de căutare informate
  - Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
  - Strategii de căutare adversială

C. Sisteme inteligente
- Sisteme bazate pe reguli în medii certe
- Sisteme bazate pe reguli în medii incerte (Bayes, factori de certitudine, Fuzzy)
- Sisteme care învaţă singure
  - Arbori de decizie
  - Reţele neuronale artificiale
  - Maşini cu suport vectorial
  - Algoritmi evolutivi
- Sisteme hibride

- Informaţiile prezentate au fost colectate din diferite surse de pe internet, precum şi din cursurile de inteligenţă artificială ţinute în anii anteriori de către:
  - Conf. Dr. Mihai Oltean – www.cs.ubbcluj.ro/~moltean
  - Lect. Dr. Crina Groşan - www.cs.ubbcluj.ro/~cgrosan
  - Prof. Dr. Horia F. Pop - www.cs.ubbcluj.ro/~hfpop