

INTELIGENȚĂ , ARTIFICIALĂ



Sisteme inteligente

Sisteme care învață singure

- K-means -

Laura Dioșan

Sumar

A. Scurtă introducere în Inteligența 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

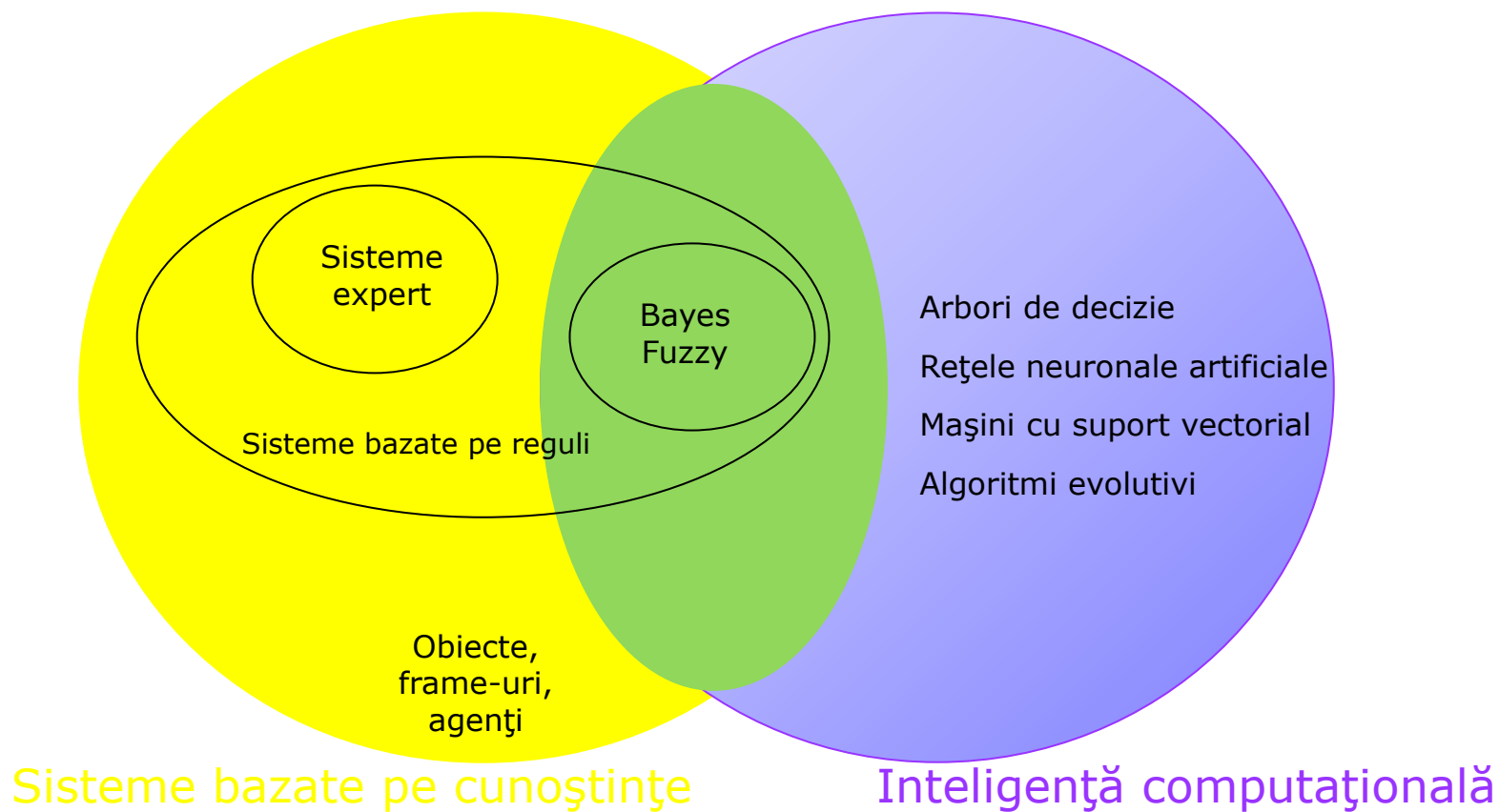
C. Sisteme inteligente

- Sisteme care învață singure
 - Arbori de decizie
 - Rețele neuronale artificiale
 - Algoritmi evolutivi
 - Algoritmi de clusterizare
- Sisteme bazate pe reguli
- Sisteme hibride

Materiale de citit și legături utile

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

Sisteme inteligente



Sisteme inteligente – SIS – Învățare automată

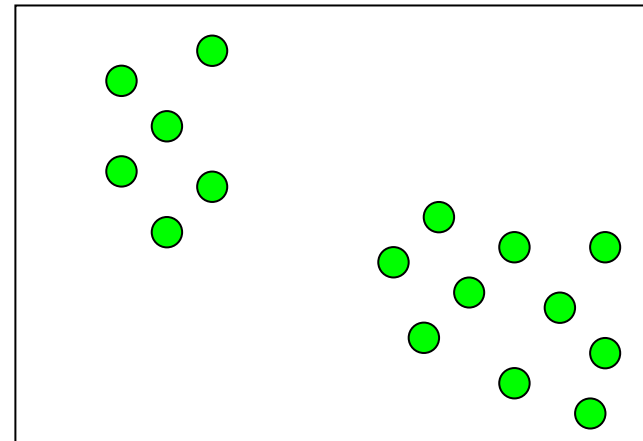
□ Tipologie

- În funcție de experiența acumulată în timpul învățării:
 - SI cu învățare supervizată
 - **SI cu învățare nesupervizată**
 - SI cu învățare activă
 - SI cu învățare cu întărire

- În funcție de modelul învățat (algoritmul de învățare):
 - Arbori de decizie
 - Rețele neuronale artificiale
 - Mașini 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
 - Identificarea unor grupuri (clusteri)
 - Analiza genelor
 - Procesarea imaginilor
 - Analiza rețelelor sociale
 - Segmentarea pieței
 - Analiza datelor astronomice
 - Clusteri de calculatoare
 - Reducerea dimensiunii
 - Identificarea unor cauze (explicații) ale datelor
 - Modelarea densității datelor
- Caracteristic
 - Datele nu sunt adnotate (etichetate)



Învățare ne-supervizată – definire

Împărțirea unor exemple **neetichetate** în submulțimi disjuncte (clusteri) astfel încât:

- exemplele din același cluster sunt foarte similare
- exemplele din clusteri diferiți sunt foarte diferite

Definire

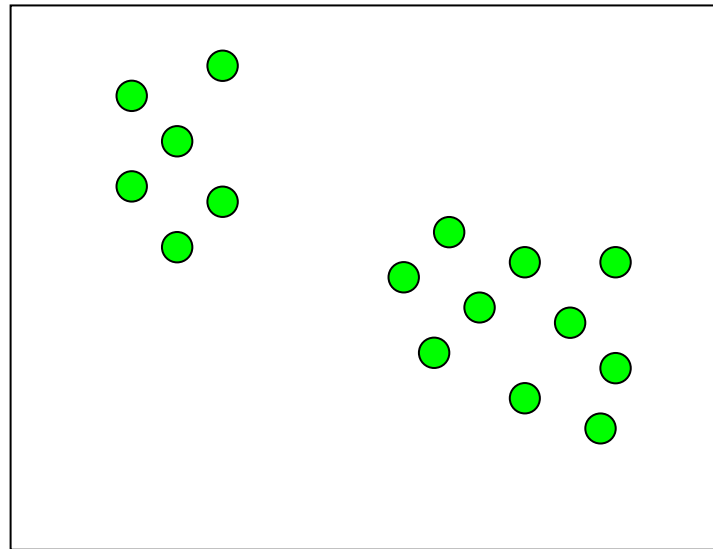
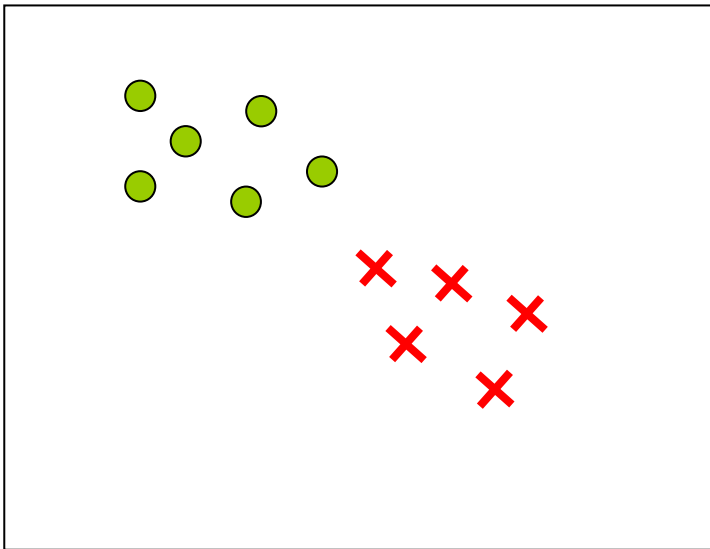
- Se dă
 - un set de date (exemple, instanțe, cazuri)
 - Date de antrenament
 - Sub forma **attribute_data_i**, unde
 - $i = 1, N$ (N = nr datelor de antrenament)
 - **attribute_data_i** = ($atr_{i1}, atr_{i2}, \dots, atr_{im}$), m – nr atributelor (caracteristicilor, proprietăților) unei date
 - Date de test
 - Sub forma (**attribute_data_i**), $i = 1, n$ (n = nr datelor de test)
- Se determină
 - o funcție (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 învățată pe datele de antrenament

Alte denumiri

- Clustering

Învățare ne-supervizată – definire

□ Supervizată vs. Ne-supervizată



Învățare ne-supervizată – definire

- Distanțe între 2 elemente \mathbf{p} și $\mathbf{q} \in R^m$
 - Euclideană
 - $d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{j=1,2,\dots,m} (p_j - q_j)^2}$
 - Manhattan
 - $d(\mathbf{p}, \mathbf{q}) = \sum_{j=1,2,\dots,m} |p_j - q_j|$
 - Mahalanobis
 - $d(\mathbf{p}, \mathbf{q}) = \sqrt{(\mathbf{p} - \mathbf{q})^T S^{-1} (\mathbf{p} - \mathbf{q})}$,
 - unde S este matricea de variație și covariație ($S = E[(\mathbf{p} - E[\mathbf{p}])(\mathbf{q} - E[\mathbf{q}])^T]$)
 - Produsul intern
 - $d(\mathbf{p}, \mathbf{q}) = \sum_{j=1,2,\dots,m} p_j q_j$
 - Cosine
 - $d(\mathbf{p}, \mathbf{q}) = \sum_{j=1,2,\dots,m} p_j q_j / (\sqrt{\sum_{j=1,2,\dots,m} p_j^2} * \sqrt{\sum_{j=1,2,\dots,m} q_j^2})$
 - Hamming
 - numărul de diferențe între \mathbf{p} și \mathbf{q}
 - Levenshtein
 - numărul minim de operații necesare pentru a-l transforma pe \mathbf{p} în \mathbf{q}
- Distanță vs. Similaritate
 - Distanța \rightarrow min
 - Similaritatea \rightarrow max

Învățare ne-supervizată – exemple

- Gruparea genelor
- Studii de piață pentru gruparea clienților (segmentarea pieței)
- news.google.com

Învățare ne-supervizată – proces

Procesul

- 2 pași:
 - Antrenarea
 - Învățarea (determinarea), cu ajutorul unui algoritm, a clusterilor existenți
 - Testarea
 - Plasarea unei noi date într-unul din clusterii identificați în etapa de antrenament

Calitatea învățării (validarea clusterizării):

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

Învățare ne-supervizată – evaluare

Măsuri de performanță

□ Criterii interne

- Distanța în interiorul clusterului
- Distanța între clusteri
- Indexul Davies-Bouldin
- Indexul Dunn

□ Criterii externe

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

Învățare ne-supervizată – evaluare

Măsuri de performanță

□ Criterii interne

- Distanța în interiorul clusterului c_j care conține n_j instanțe
 - Distanța medie între instanțe (average distance)
 - $D_a(c_j) = \sum_{x_{i1}, x_{i2} \in c_j} \|x_{i1} - x_{i2}\| / (n_j(n_j - 1))$
 - Distanța între cei mai apropiați vecini (nearest neighbour distance)
 - $D_{nn}(c_j) = \sum_{x_{i1} \in c_j} \min_{x_{i2} \in c_j} \|x_{i1} - x_{i2}\| / n_j$
 - Distanța între centroizi
 - $D_c(c_j) = \sum_{x_i \in c_j} \|x_i - \mu_j\| / n_j$, unde $\mu_j = 1/n_j \sum_{x_i \in c_j} x_i$

Învățare ne-supervizată – evaluare

Măsuri de performanță

□ Criterii interne

■ Distanța între 2 clusteri c_{j1} și c_{j2}

□ Legătură simplă

$$d_s(c_{j1}, c_{j2}) = \min_{x_{i1} \in c_{j1}, x_{i2} \in c_{j2}} \{ \|x_{i1} - x_{i2}\| \}$$

□ Legătură completă

$$d_{co}(c_{j1}, c_{j2}) = \max_{x_{i1} \in c_{j1}, x_{i2} \in c_{j2}} \{ \|x_{i1} - x_{i2}\| \}$$

□ Legătură medie

$$d_a(c_{j1}, c_{j2}) = \sum_{x_{i1} \in c_{j1}, x_{i2} \in c_{j2}} \{ \|x_{i1} - x_{i2}\| \} / (n_{j1} * n_{j2})$$

□ Legătură între centroizi

$$d_{ce}(c_{j1}, c_{j2}) = \| \mu_{j1} - \mu_{j2} \|$$

Învățare ne-supervizată – evaluare

Măsuri de performanță

□ Criterii interne

■ Indexul Davies-Bouldin → min → clusteri compacți

- $DB = 1/nc * \sum_{i=1,2,...,nc} \max_{j=1, 2, ..., nc, j \neq i} ((\sigma_i + \sigma_j)/d(\mu_i, \mu_j))$

- unde:

- nc – numărul de clusteri
- μ_i – centroidul clusterului i
- σ_i – media distanțelor între elementele din clusterul i și centroidul μ_i
- $d(\mu_i, \mu_j)$ – distanța între centroidul μ_i și centroidul μ_j

■ Indexul Dunn

- Identifică clusterii denși și bine separați

- $D = d_{min}/d_{max}$

- Unde:

- d_{min} – distanța minimă între 2 obiecte din clusteri diferiți – distanța intra-cluster
- d_{max} – distanța maximă între 2 obiecte din același cluster – distanța inter-cluster

Învățare ne-supervizată - tipologie

- După modul de formare al clusterilor
 - C. ierarhic
 - C. ne-ierarhic (partițional)
 - 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 cunoaște 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

- Partițional → se determină o împărțire a datelor → toți clusterii deodată
- Optimizează o funcție obiectiv definită
 - Local – doar pe anumite attribute
 - Global – pe toate attributelecare poate fi
 - Pătratul erorii – suma patratelor distanțelor între date și centroizii clusterilor → min
 - Ex. *K-means*
 - Bazată pe grafuri
 - Ex. Clusterizare bazată pe arborele minim de acoperire
 - Bazată pe modele probabilistice
 - Ex. Identificarea distribuției datelor → Maximizarea așteptărilor
 - Bazată pe cel mai apropiat vecin
- Necesită fixarea *a priori* a lui k → fixarea clusterilor inițiali
 - Algoritmii se rulează de mai multe ori cu diferiți parametri și 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 și conectivitatea datelor
 - Formarea clusterilor de bazează pe densitatea datelor într-o anumită regiune
 - Formarea clusterilor de bazează pe conectivitatea datelor dintr-o anumită regiune
 - Funcția de densitate a datelor
 - Se încearcă modelarea legii de distribuție 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, partițional sau bazat pe densitate
 - Pp. segmentarea spațiului 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 instanță formează inițial un cluster
 2. Se calculează distanțele între oricare 2 clusteri
 3. Se reunesc cei mai apropiați 2 clusteri
 4. Se repetă pașii 2 și 3 până se ajunge la un singur cluster sau la un alt criteriu de stop
 - Diviziv
 1. Se stabilește numărul de clusteri (k)
 2. Se inițializează centrii fiecărui cluster
 3. Se determină o împărțire a datelor
 4. Se recalculează centrii clusterilor
 5. Se repetă pasul 3 și 4 până partiționarea nu se mai schimbă (algoritmul a converș)
- După attributele considerate
 - Monotetic – attributele se consideră pe rând
 - Politetic – attributele se consideră simultan

Învățare ne-supervizată - tipologie

- După tipul de apartenență al datelor la clusteri
 - Clustering exact (*hard clustering*)
 - Asociază fiecărei intrări \mathbf{x}_i o etichetă (clasă) c_j
 - Clustering fuzzy
 - Asociază fiecărei intrări \mathbf{x}_i un grad (probabilitate) de apartenență f_{ij} la o anumită clasă $c_j \rightarrow$ o instanță \mathbf{x}_i poate aparține mai multor clusteri

Învățare ne-supervizată – algoritmi

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

Învățare ne-supervizată – algoritmi

Clustering ierarhic aglomerativ

1. Se consideră o distanță între 2 instanțe $d(x_{i1}, x_{i2})$
2. Se formează N clusteri, fiecare conținând câte o instanță
3. Se repetă
 - Determinarea celor mai apropiați 2 clusteri
 - Se reunesc cei 2 clusteri → un singur cluster
4. Până când se ajunge la un singur cluster (care conține toate instanțele)

Învățare ne-supervizată – algoritmi

Clustering ierarhic aglomerativ

□ Distanța între 2 clusteri c_i și c_j :

- Legătură simplă → minimul distanței între obiectele din cei 2 clusteri

- $d(c_i, c_j) = \max_{x_{i1} \in c_i, x_{i2} \in c_j} \text{sim}(\mathbf{x}_{i1}, \mathbf{x}_{i2})$

- Legătură completă → maximul distanței între obiectele din cei 2 clusteri

- $d(c_i, c_j) = \min_{x_{i1} \in c_i, x_{i2} \in c_j} \text{sim}(\mathbf{x}_{i1}, \mathbf{x}_{i2})$

- Legătură medie → media distanței între obiectele din cei 2 clusteri

- $d(c_i, c_j) = 1 / (n_i * n_j) \sum_{x_{i1} \in c_i} \sum_{x_{i2} \in c_j} d(\mathbf{x}_{i1}, \mathbf{x}_{i2})$

- Legătură medie peste grup → distanța între mediile (centroizii) celor 2 clusteri

- $d(c_i, c_j) = \rho(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$, ρ – distanță, $\boldsymbol{\mu}_j = 1/n_j \sum_{x_{i2} \in c_j} \mathbf{x}_{i2}$

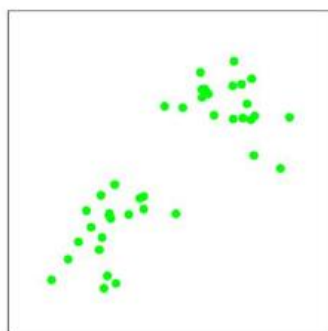
Învățare ne-supervizată – algoritmi

K-means (algoritmul Lloyd/iterația Voronoi)

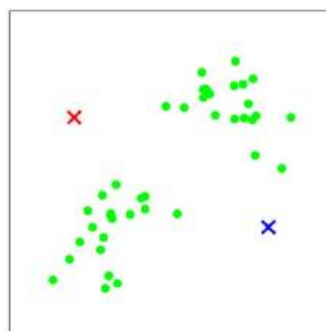
- Pp că se vor forma k clusteri
- Inițializează k centroizi $\mu_1, \mu_2, \dots, \mu_k$
 - Un centroid μ_j ($j=1, 2, \dots, k$) este un vector cu m valori (m – nr de attribute)
- Repetă până la convergență
 - Asociază fiecare instanță celui mai apropiat centroid → pentru fiecare instanță \mathbf{x}_i , $i = 1, 2, \dots, N$
 - $c_i = \arg \min_{j = 1, 2, \dots, k} ||\mathbf{x}_i - \mu_j||^2$
 - Recalculează centroizii prin mutarea lor în media instanțelor asociate fiecăruia → pentru fiecare cluster c_j , $j = 1, 2, \dots, k$
 - $\mu_j = \sum_{i=1, 2, \dots, N} 1_{c_i=j} \mathbf{x}_i / \sum_{i=1, 2, \dots, N} 1_{c_i=j}$

Învățare ne-supervizată – algoritmi

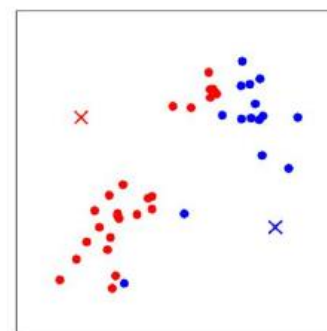
K-means



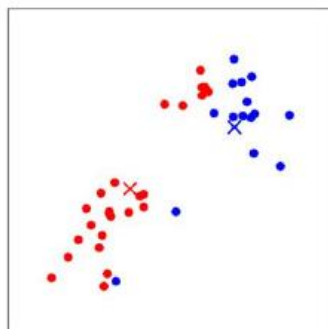
(a)



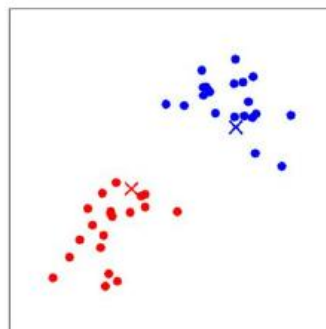
(b)



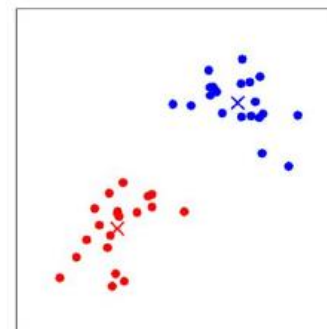
(c)



(d)



(e)



(f)

Învățare ne-supervizată – algoritmi

K-means

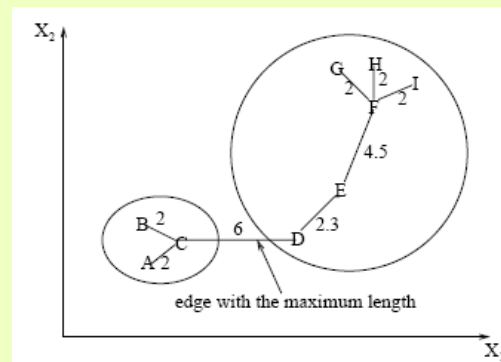
- Inițializarea a k centroizi $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_k$
 - Cu valori generate aleator (în domeniul de definiție al problemei)
 - Cu k dintre cele N instanțe (alese în mod aleator)

- Algoritmul converge întotdeauna?
 - Da, pt că avem funcția de distorsiune J
 - $J(c, \mu) = \sum_{i=1,2, \dots, N} ||\mathbf{x}_i - \boldsymbol{\mu}_{c_j}||^2$
care este descrescătoare
 - Converge într-un optim local
 - Găsirea optimului global \rightarrow NP-dificilă

Învățare ne-supervizată – algoritmi

Clusterizare bazată pe arborele minim de acoperire (AMA)

- ❑ Se construiește 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 instanțe
- Se repetă până la etichetarea tuturor instanțelor
 - O instanță ne-etichetată va fi inclusă în clusterul instanței cele mai apropiate
 - dacă distanța între instanța neetichetată și cea etichetată este mai mică decât un prag

Învățare ne-supervizată – algoritmi

Clusterizare fuzzy

- Se stabilește o partiționare fuzzy inițială
 - Se construiește matricea gradelor de apartenență U , unde u_{ij} – gradul de apartenență al instanței \mathbf{x}_i ($i=1,2, \dots, N$) la clusterul c_j ($j = 1, 2, \dots, k$) ($u_{ij} \in [0,1]$)
 - Cu cât u_{ij} e mai mare, cu atât e mai mare încrederea că instanța \mathbf{x}_i face parte din clusterul c_j
- Se stabilește o funcție obiectiv
 - $E^2(U) = \sum_{i=1,2, \dots, N} \sum_{j=1,2, \dots, k} u_{ij} ||\mathbf{x}_i - \boldsymbol{\mu}_j||^2$,
 - unde $\boldsymbol{\mu}_j = \sum_{i=1,2, \dots, N} u_{ij} \mathbf{x}_i$ – centrul celui de-al j -lea fuzzy cluster
 - care se optimizează (min) prin re-atribuirea instanțelor (în clusteri noi)
- Clusering fuzzy \rightarrow clusterizare *hard* (fixă)
 - impunerea unui prag funcției de apartenență u_{ij}

Invățare ne-supervizată – algoritmi

Algoritmi evolutivi

□ Algoritmi

- Inspirați din natură (biologie)

- Iterativi

- Bazați pe

 - populații de potențiale soluții

 - căutare aleatoare ghidată de

 - Operații de selecție naturală

 - Operații de încrucișare și mutație

- Care procesează în paralel mai multe soluții

□ Metafora evolutivă

Evoluție naturală	Rezolvarea problemelor
Individ	Soluție potențială (candidat)
Populație	Mulțime de soluții
Cromozom	Codarea (reprezentarea) unei soluții
Genă	Parte a reprezentării
Fitness (măsură de adaptare)	Calitate
Încrucișare și mutație	Operatori de căutare
Mediu	Spațiul de căutare al problemei

Învățare ne-supervizată – algoritmi

Algoritmi evolutivi

Initializare populație $P(0)$

Evaluare $P(0)$

$g := 0$; //generația

CâtTimp (not condiție_stop) execută

Repetă

 Selectează 2 părinți $p1$ și $p2$ din $P(g)$

 Încrucișare($p1, p2$) $\Rightarrow o1$ și $o2$

 Mutație($o1$) $\Rightarrow o1^*$

 Mutație($o2$) $\Rightarrow o2^*$

 Evaluare($o1^*$)

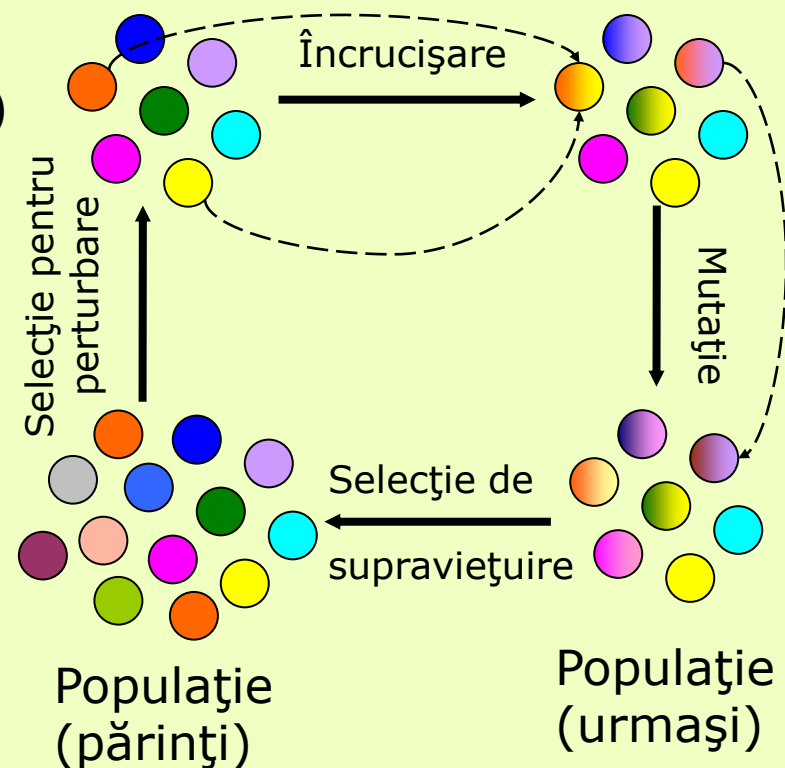
 Evaluare($o2^*$)

 adăugare $o1^*$ și $o2^*$ în $P(g+1)$

 Până când $P(g+1)$ este completă

$g := g + 1$

Sf CâtTimp



Învățare ne-supervizată – algoritmi

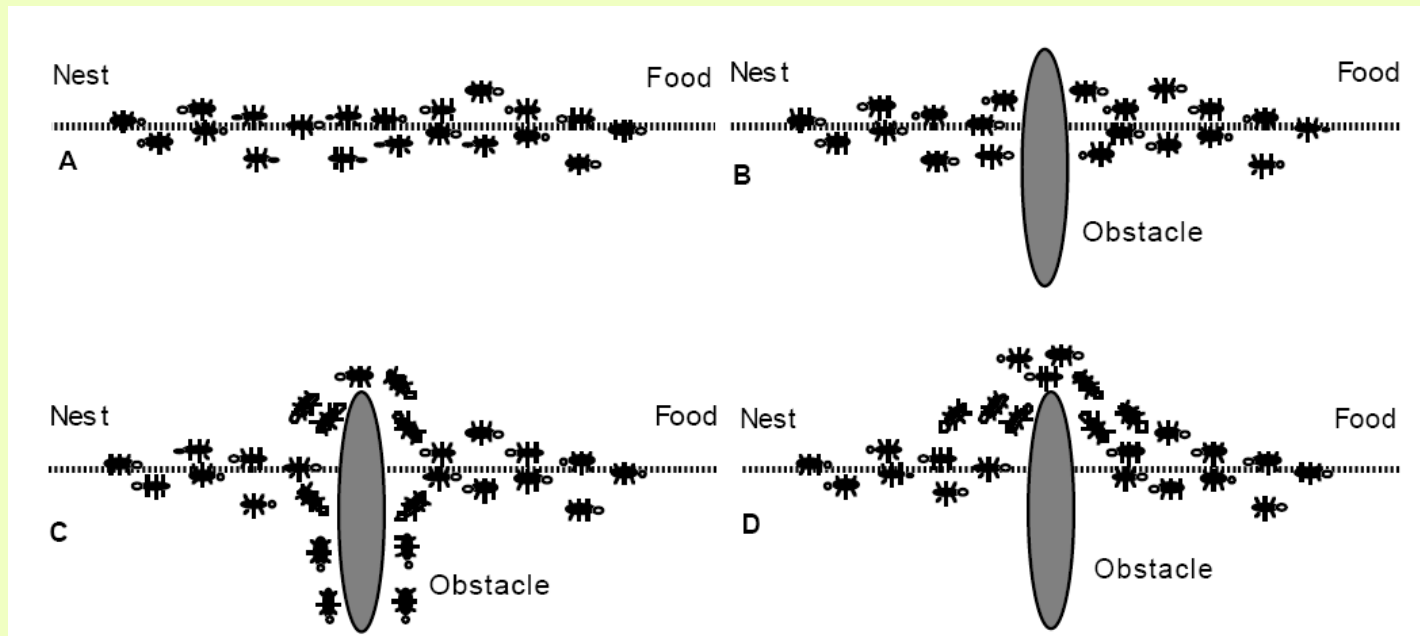
Algoritmi evolutivi

- Reprezentare
 - Cromozomul = o partiționare a datelor
 - Ex. 2 clusteri → cromozom = vector binar
 - Ex. K clusteri → cromozom = vector cu valori din $\{1, 2, \dots, k\}$
- Fitness
 - Calitatea partiționării
- Inițializare
 - Aleatoare
- Încrucișare
 - Punct de tăietură
- Mutație
 - Schimbarea unui element din cromozom

Învățare ne-supervizată – algoritmi

ACO

- ❑ Preferința pentru drumuri cu nivel ridicat de feromon
- ❑ Pe drumurile scurte feromonul se înmulțește
- ❑ 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 furnicuțele să le grupeze în funcție de asemănarea lor
- ❑ 2 reguli pentru furnicuțe
 - 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 locație 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 furnicuței
- ❑ Furnicuțele
 - au memorie
 - ❑ rețin obiectele din vecinătatea poziției curente
 - se mișcă ortogonal (N, S, E, V) pe grid pe căsuțele neocupate de alte furnici

Recapitulare



□ Sisteme care învață singure (SIS)

■ Invatare nesupervizata

■ K-means

□ Modele computaționale care

■ rezolvă probleme de clusterizare

■ → nu se cunosc etichetele claselor

■ prin

■ minimizarea diferențelor între elementele aceleași clase

■ maximizarea diferențelor între elementele claselor diferite

Cursul următor

A. Scurtă introducere în Inteligența 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 adversarială

C. Sisteme inteligente

- Sisteme bazate pe reguli în medii certe
- Sisteme bazate pe reguli în medii incerte (Bayes, factori de certitudine, Fuzzy)
- Sisteme care învață singure
 - Arbori de decizie
 - Rețele neuronale artificiale
 - Mașini cu suport vectorial
 - Algoritmi evolutivi
- Sisteme hibride

□ Informațiile prezentate au fost colectate din diferite surse de pe internet, precum și din cursurile de inteligență artificială ținute în anii anteriori de către:

■ Conf. Dr. Mihai Oltean –
www.cs.ubbcluj.ro/~moltean

■ Lect. Dr. Crina Groșan -
www.cs.ubbcluj.ro/~cgrosan

■ Prof. Dr. Horia F. Pop -
www.cs.ubbcluj.ro/~hfpop