



UNIVERSITATEA BABEȘ-BOLYAI  
Facultatea de Matematică și Informatică



# INTELIGENȚĂ ARTIFICIALĂ

**Sisteme inteligente**

Sisteme care învață singure

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 adversarială

## C. Sisteme inteligente

- Sisteme care învață singure
  - Arbori de decizie
  - Rețele neuronale artificiale
  - Mașini cu suport vectorial
  - Algoritmi evolutivi
- Sisteme bazate pe reguli
- Sisteme hibride

# Materiale de citit și legături utile

---

- ❑ capitolul VI (18) din *S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995*
- ❑ capitolul 10 și 11 din *C. Groșan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011*
- ❑ capitolul V din *D. J. C. MacKey, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003*
- ❑ capitolul 3 din *T. M. Mitchell, Machine Learning, McGraw-Hill Science, 1997*

# Conținut

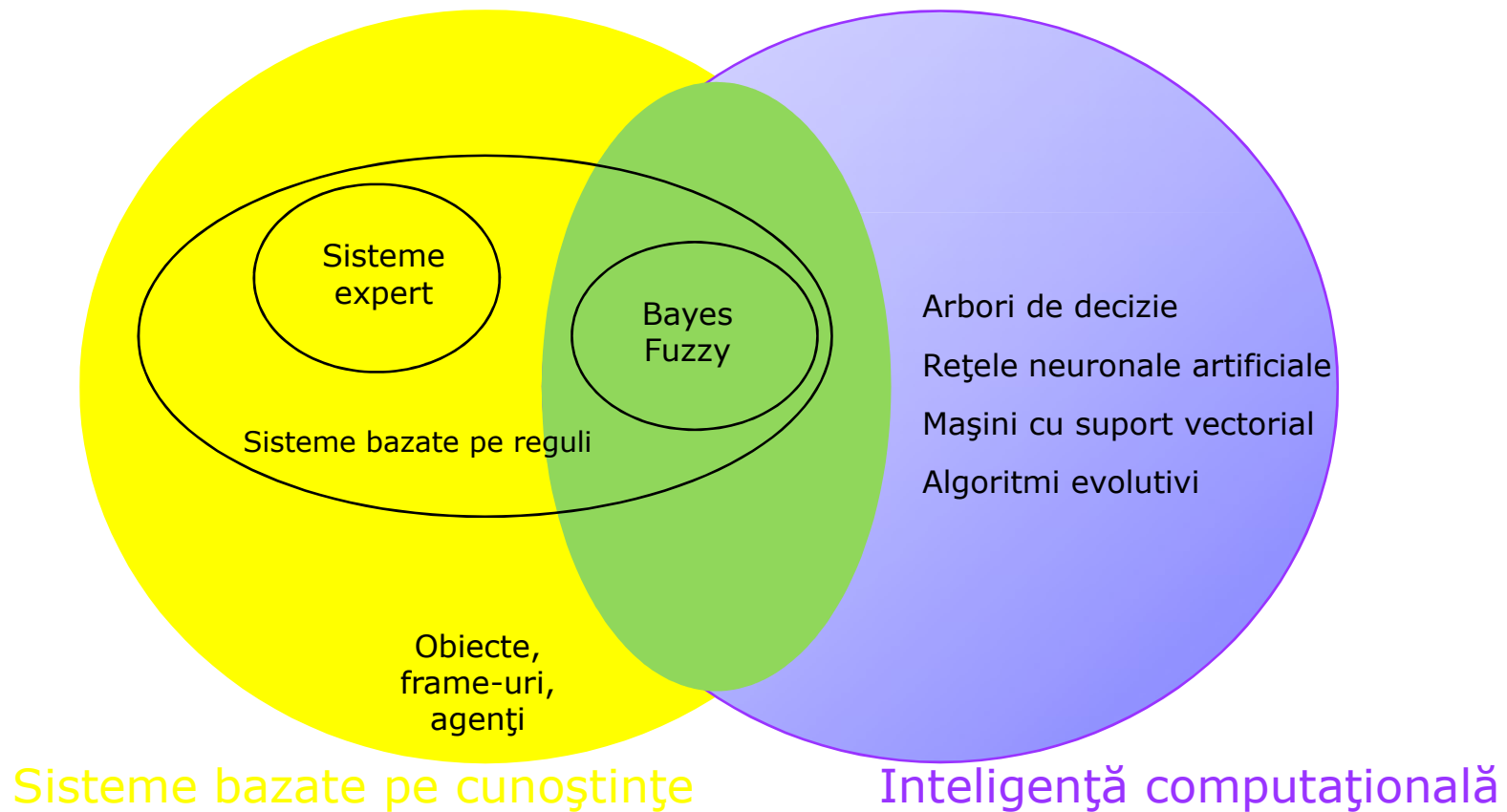
---

## □ Sisteme inteligente

### ■ Sisteme care învață singure (SIS)

- Instruire (învățare) automata (Machine Learning - ML)
  - Problematică
  - Proiectarea unui sistem de învățare automată
  - Tipologie
    - Învățare supervizată
    - Învățare nesupervizată
    - Învățare cu întărire
    - Teoria învățării
- Exemple de sisteme

# Sisteme inteligente



# Sisteme inteligente – SIS – Învățare automată

---

## □ Problematica

- “How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?”

## □ Aplicații

- Recunoaștere de imagini și semnal vocal
  - Recunoașterea scrisului de mână
  - Detecția fețelor
  - Înțelegerea limbajului vorbit
- Computer vision
  - Detecția obstacolelor
  - Recunoașterea amprentelor
- Supraveghere bio
- Controlul roboților
- Predicția vremii
- Diagnosticare medicală
- Detecția fraudelor

# Sisteme inteligente – SIS – Învățare automată

## □ Definire

- Arthur Samuel (1959)
  - "field of study that gives computers the ability to learn without being explicitly programmed"
  - Înzestrarea computerelor cu abilitatea de a învăța pe baza experienței
- Herbert Simon (1970)
  - "Learning is any process by which a system improves performance from experience."
- Tom Mitchell (1998)
  - "a well-posed learning problem is defined as follows: He says that a computer program is set to learn from an experience E with respect to some task T and some performance measure P if its performance on T as measured by P improves with experience E"
- Ethem Alpaydin (2010)
  - Programming computers to optimize a performance criterion using example data or past experience.

## □ Necesitate

- Sisteme computaționale mai bune
  - Sisteme dificil sau prea costisitor de construit manual
    - Sisteme care se adaptează automat
      - Filtre de spam
    - Sisteme care descoperă informații în baze de date mari → data mining
      - Analize financiare
      - Analize de text/imagini
- Înțelegerea organismelor biologice



# Sisteme inteligente – SIS – Învățare automată

---

## □ Proiectare

### ■ Îmbunătățirea task-ului T

- Stabilirea scopului (ceea ce trebuie învățat) - funcției obiectiv – și reprezentarea sa
- Alegerea unui algoritm de învățare care să realizeze inferența (previziunea) scopului pe baza experienței

### ■ respectând o metrică de performanță P

- Evaluarea performanțelor algoritmului ales

### ■ bazându-se pe experiența E

- Alegerea bazei de experiență

### ■ Exemplu

- T: jucarea jocului de dame
- P: procentul de jocuri câștigate împotriva unui oponent oarecare
- E: exersarea jocului împotriva lui însuși
  
- T: recunoașterea scrisului de mână
- P: procentul de cuvinte recunoscute corect
- E: baze de date cu imagini cu cuvinte corect adnotate
  
- T: separarea spam-urilor de mesajele obișnuite
- P: procentul de email-uri corect clasificate (spam sau normal)
- E: baze de date cu email-uri adnotate



# Sisteme inteligente – SIS – Învățare automată

---

## □ Proiectare → Alegerea funcției obiectiv

### ■ Care este funcția care trebuie învățată?

- Ex.: pentru jocul de dame → funcție care:
  - alege următoarea mutare
  - evaluează o mutare
- obiectivul fiind alegerea celei mai bune mutări

### ■ Reprezentarea funcției obiectiv

- Diferite reprezentări
  - Tablou (tabel)
  - Reguli simbolice
  - Funcție numerică
  - Funcții probabilistice
- Ex. Jocul de dame
  - Combinație liniară a nr. de piese albe, nr. de piese negre, nr. de piese albe compromise la următoarea mutare, r. de piese albe compromise la următoarea mutare
- Există un compromis între
  - expresivitatea reprezentării și
  - ușurința învățării
- Calculul funcției obiectiv
  - Timp polinomial
  - Timp non-polinomial

# Sisteme inteligente – SIS – Învățare automată

---

## □ Proiectare → Alegerea unui algoritm de învățare

### ■ Algoritmul

- folosind datele de antrenament
- induce definirea unor ipoteze care
  - să se potrivească cu acestea și
  - să generalizeze cât mai bine datele ne-văzute (datele de test)

### ■ Principiul de lucru de bază

- Minimizarea unei erori (funcție de cost – loss function)

## □ Proiectare → Evaluarea unui sistem de învățare

### ■ Experimental

- Compararea diferitelor metode pe diferite date (cross-validare)
- Colectarea datelor pe baza performanței
  - Acuratețe, timp antrenare, timp testare
- Aprecierea diferențelor dpdv statistic

### ■ Teoretic

- Analiza matematică a algoritmilor și demonstrarea de teoreme
  - Complexitatea computațională
  - Abilitatea de a se potrivi cu datele de antrenament
  - Complexitatea eșantionului relevant pentru o învățare corectă

# Sisteme inteligente – SIS – Învățare automată

## □ Proiectare → Evaluarea unui sistem de învățare

### ■ Compararea performanțelor a 2 algoritmi în rezolvarea unei probleme

#### □ Indicatori de performanță

- Parametrii ai unei serii statistice (ex. media)
- Proporție calculată pentru serie statistică (ex. acuratețea)

#### □ Comparare pe baza intervalelor de încredere

- Pp o problemă și 2 algoritmi care o rezolvă
- Performanțele algoritmilor:  $p_1$  și  $p_2$
- Intervalele de încredere corespunzătoare celor 2 performanțe  $I_1=[p_1-\Delta_1, p_1+\Delta_1]$  și  $I_2=[p_2-\Delta_2, p_2+\Delta_2]$
- Dacă  $I_1 \cap I_2 = \emptyset \rightarrow$  algoritmul 1 este mai bun decât algoritmul 2 (pt problema dată)
- Dacă  $I_1 \cap I_2 \neq \emptyset \rightarrow$  nu se poate spune care algoritm este mai bun

#### □ Interval de încredere pentru medie

- Pentru o serie statistică de volum  $n$ , cu media (calculată)  $m$  și dispersia  $\sigma$  să se determine intervalul de încredere al valorii medii  $\mu$
- $P(-z \leq (m-\mu)/(\sigma/\sqrt{n}) \leq z) = 1 - \alpha \rightarrow \mu \in [m - z\sigma/\sqrt{n}, m + z\sigma/\sqrt{n}]$
- $P = 95\% \rightarrow z = 1.96$
- Ex. Problema rucsacului rezolvată cu ajutorul algoritmilor evolutivi

#### □ Interval de încredere pentru acuratețe

- Pentru o performanță  $p$  (acuratețe) calculată pentru  $n$  date să se determine intervalul de încredere
- $P \in [p - z(p(1-p)/n)^{1/2}, p + z(p(1-p)/n)^{1/2}]$
- $P = 95\% \rightarrow z = 1.96$
- Ex. Problemă de clasificare rezolvată cu ajutorul Mașinilor cu suport vectorial

$P=1-\alpha$	$z$
99.9%	3.3
99.0%	2.577
98.5%	2.43
97.5%	2.243
95.0%	1.96
90.0%	1.645
85.0%	1.439
75.0%	1.151

# Sisteme inteligente – SIS – Învățare automată

## □ Proiectare → Alegerea bazei de experiență

### ■ Bazată pe

#### □ Experiență directă

- Perechi (intrare, ieșire) utile pt. funcția obiectiv
- Ex. Jocul de dame → table de joc etichetată cu mutare corectă sau incorectă

#### □ Experiență indirectă

- Feedback util (diferit de perechile I/O) pt funcția obiectiv
- Ex. Jocul de dame → secvențe de mutări și scorul final asociat jocului

### ■ Surse de date

#### □ Exemple generate aleator

- Exemple pozitive și negative

#### □ Exemple pozitive colectate de un “învățător” benevol

#### □ Exemple reale

### ■ Compoziție

#### □ Date de antrenament

#### □ Date de test

### ■ Caracteristici

#### □ Date independente

- Dacă nu → clasificare colectivă

#### □ Datele de antrenament și de test trebuie să urmeze aceeași lege de distribuție

- Dacă nu → învățare prin transfer (*transfer learning/inductive transfer*)
  - recunoașterea mașinilor → recunoașterea camioanelor
  - analiza textelor
  - filtre de spam

# Sisteme inteligente – SIS – Învățare automată

## □ Proiectare → Alegerea bazei de experiență

### ■ Tipuri de attribute ale datelor

#### □ Cantitative → scară nominală sau rațională

- Valori continue → greutatea
- Valori discrete → numărul de computere
- Valori de tip interval → durata unor evenimente

#### □ Calitative

- Nominale → culoarea
- Ordinale → intensitatea sunetului (joasă, medie, înaltă)

#### □ Structurate

- Arbori – rădăcina e o generalizare a copiilor (vehicol → mașină, autobus, tractor, camion)

### ■ Transformări asupra datelor

#### □ Standardizare → attribute numerice

- Înlăturarea efectelor de scară (scări și unități de măsură diferite)
- Valorile brute se transformă în scoruri  $z$ 
  - $Z_{ij} = (x_{ij} - \mu_j) / \sigma_j$ , unde  $x_{ij}$  – valoarea atributului al  $j$ -lea al instanței  $i$ ,  $\mu_j$  ( $\sigma_j$ ) este media (abaterea) atributelor  $j$  pt. toate instanțele

#### □ Selectarea anumitor attribute

# Sisteme inteligente – SIS – Învățare automată

## □ Tipologie

### ■ În funcție de scopul urmărit

#### □ SI pentru predicții

- Scop: predicția ieșirii pentru o intrare nouă folosind un model învățat anterior
- Ex.: predicția vânzărilor dintr-un produs pentru un moment de timp viitor în funcție de preț, lună calendaristică, regiune, venit mediu pe economie

#### □ SI pentru regresii

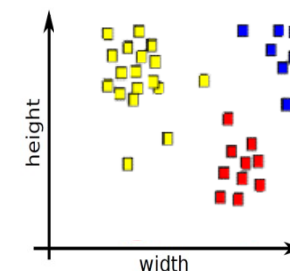
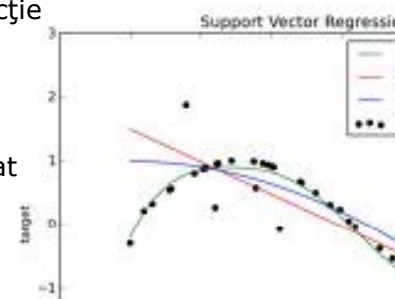
- Scop: estimarea formei unei funcții uni sau multivariată folosind un model învățat anterior
- Ex.: estimarea funcției care modelează conturul unei suprafețe

#### □ SI pentru clasificare

- Scop: clasificarea unui obiect într-una sau mai multe categorii (clase) – cunoscute anterior sau nu - pe baza caracteristicilor (atributelor, proprietăților) lui
- Ex.: sistem de diagnoză pentru un pacient cu tumoare: nevasculară, vasvulară, angiogenă

#### □ SI pentru planificare

- Scop: generarea unei succesiuni optime de acțiuni pentru efectuarea unei sarcini
- Ex.: planificarea deplasării unui robot de la o poziție dată până la o sursă de energie (pentru alimentare)

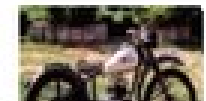
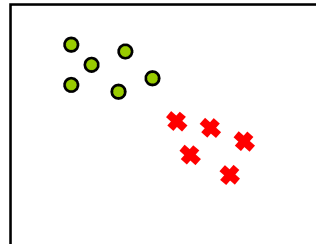


# Sisteme inteligente – SIS – Învățare automată

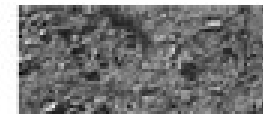
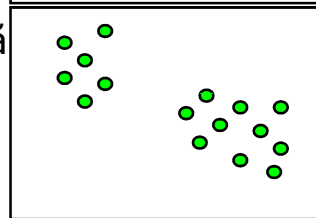
## □ Tipologie

### ■ În funcție de experiența acumulată în timpul învățării

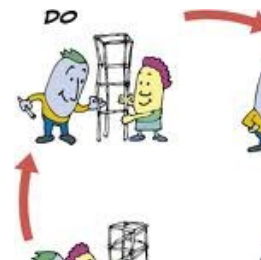
#### □ SI cu învățare supervizată



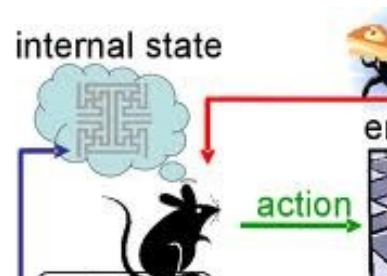
#### □ SI cu învățare nesupervizată



#### □ SI cu învățare activă



#### □ SI cu învățare cu întărire



# Sisteme inteligente – SIS – Învățare automată

---

## □ Învățare supervizată

- Definire
- Exemple
- Proces
- Calitatea învățării
  - Metode de evaluare
  - Măsuri de performanță
- Tipologie



# Sisteme inteligente – SIS – Învățare automată

## Învățare supervizată

- Scop
  - Furnizarea unei ieșiri corecte pentru o nouă intrare
- Definiere
  - Se dă un set de date (exemple, instanțe, cazuri)
    - date de antrenament – sub forma unor perechi ( $\text{attribute\_data}_i, \text{ieșire}_i$ ), unde
      - $i = 1, N$  ( $N = \text{nr datelor de antrenament}$ )
      - $\text{attribute\_data}_i = (\text{atr}_{i1}, \text{atr}_{i2}, \dots, \text{atr}_{im})$ ,  $m = \text{nr atributelor (caracteristicilor, proprietăților) unei date}$
      - $\text{ieșire}_i$ 
        - o categorie dintr-o mulțime dată (predefinită) cu  $k$  elemente ( $k = \text{nr de clase}$ ) → problemă de clasificare
        - un număr real → problemă de regresie
    - date de test - sub forma ( $\text{attribute\_data}_i$ ),  $i = 1, n$  ( $n = \text{nr datelor de test}$ ).
  - Să se determine
    - o funcție (necunoscută) care realizează corespondența atribut – ieșire pe datele de antrenament
    - ieșirea (clasa/valoarea) asociată unei date (noi) de test folosind funcția învățată pe datele de antrenament
- Alte denumiri
  - Clasificare (regresie), învățare inductivă
- Proces → 2 etape
  - Antrenarea
    - Învățarea, cu ajutorul unui algoritm, a modelului de clasificare
  - Testarea
    - Testarea modelului folosind date de test noi (unseen data)
- Caracteristic
  - BD experimentală adnotată (pt. învățare)

# Sisteme inteligente – SIS – Învățare automată

---

## Învățare supervizată

### □ Tip de probleme

- regresie
  - Scop: predicția output-ului pentru un input nou
  - Output continuu (nr real)
  - Ex.: predicția prețurilor
- clasificare
  - Scop: clasificarea (etichetarea) unui nou input
  - Output discret (etichetă dintr-o mulțime predefinită)
  - Ex.: detectarea tumorilor maligne

### □ Exemple de probleme

- Recunoașterea scrisului de mână
- Recunoașterea imaginilor
- Previziunea vremii
- Detectția spam-urilor

# Sisteme inteligente – SIS – Învățare automată

## Învățare supervizată

### □ Calitatea învățării

#### ■ Definire

- o măsură de performanță a algoritmului
  - ex. acuratețea ( $\text{Acc} = \text{nr de exemple corect clasificate} / \text{nr total de exemple}$ )
- calculată în
  - faza de antrenare
  - faza de testare

#### ■ Metode de evaluare

- Seturi disjuncte de antrenare și testare
  - setul de antrenare poate fi împărțit în date de învățare și date de validare
  - setul de antrenare este folosit pentru estimarea parametrilor modelului (cei mai buni parametri obținuți pe validare vor fi folosiți pentru construcția modelului final)
  - pentru date numeroase
- Validare încrucișată cu mai multe ( $h$ ) sub-seturi egale ale datelor (de antrenament)
  - separarea datelor de  $h$  ori în ( $h-1$  sub-seturi pentru învățare și 1 sub-set pt validare)
  - dimensiunea unui sub-set = dimensiunea setului /  $h$
  - performanța este dată de media pe cele  $h$  rulări (ex.  $h = 5$  sau  $h = 10$ )
  - pentru date puține
- Leave-one-out cross-validation
  - similar validării încrucișate, dar  $h = \text{nr de date}$  → un sub-set conține un singur exemplu
  - pentru date foarte puține

#### ■ Dificultăți

- Învățare pe derost (overfitting) → performanță bună pe datele de antrenament, dar foarte slabă pe datele de test

# Sisteme inteligente – SIS – Învățare automată

---

## Învățare supervizată

- Calitatea învățării
  - Măsuri de performanță
    - Măsuri statistice
      - acuratețea
      - Precizia
      - Rapelul
      - Scorul F1
    - Eficiența
      - În construirea modelului
      - În testarea modelului
    - Robustețea
      - Tratarea zgomotelor și a valorilor lipsă
    - Scalabilitatea
      - Eficiența gestionării seturilor mari de date
    - Interpretabilitatea
      - Modelului de clasificare
    - Proprietatea modelului de a fi compact
    - Scoruri

# Sisteme inteligente – SIS – Învățare automată

## Învățare supervizată

### □ Calitatea învățării → Măsurile de performanță → Măsurile statistice

#### ■ Acuratețea

- Nr de exemple corect clasificate / nr total de exemple
- Opusul erorii
- Calculată pe
  - Setul de validare
  - Setul de test
- Uneori
  - Analiză de text
  - Detectarea intrușilor într-o rețea
  - Analize financiare

este importantă doar o singură clasă (clasă pozitivă) → restul claselor sunt negative

#### ■ Precizia (P)

- nr. de exemple pozitive corect clasificate / nr. total de exemple clasificate ca pozitive
- probabilitatea ca un exemplu clasificat pozitiv să fie relevant
- $TP / (TP + FP)$

#### ■ Rapelul (R)

- nr. de exemple pozitive corect clasificate / nr. total de exemple pozitive
- Probabilitatea ca un exemplu pozitiv să fie identificat corect de către clasificator
- $TP / (TP + FN)$
- Matrice de confuzie → rezultate reale vs. rezultate calculate

#### ■ Scorul F1

- Combină precizia și rapelul, facilitând compararea a 2 algoritmi
- Media armonică a preciziei și rapelului
- $2PR / (P + R)$

		Rezultate reale	
		Clasa pozitivă	Clasa(e) negativă(e)
Rezultate calculate	Clasa pozitivă	<i>True positiv (TP)</i>	<i>False positiv (FP)</i>
	Clasa(e) negativă(e)	<i>False negative (FN)</i>	<i>True negative (TN)</i>

# Sisteme inteligente – SIS – Învățare automată

---

## □ Învățare ne-supervizată

- Definire
- Exemple
- Proces
- Metode de evaluare și măsuri de performanță
- Tipologie

# Sisteme inteligente – SIS – Învățare automată

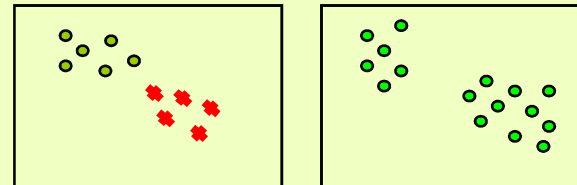
## Învățare ne-supervizată

### □ Scop

- Găsirea unui model sau a unei structuri utile a datelor
- Împărțirea unor exemple **neetichetate** în submulțimi disjuncte (clusteri) astfel încât:
  - exemplele din același cluster sunt foarte similare
  - exemplele din clusteri diferiți sunt foarte diferite

### □ Definiere

- Se dă un set de date (exemple, instanțe, cazuri)
  - Date de antrenament sub forma **attribute\_data<sub>i</sub>**, unde
    - $i = 1, N$  ( $N$  = nr datelor de antrenament)
    - **attribute\_data<sub>i</sub>** = ( $atr_{i1}, atr_{i2}, \dots, atr_{im}$ ),  $m$  – nr atributelor (caracteristicilor, proprietăților) unei date
  - Date de test sub forma (**attribute\_data<sub>i</sub>**),  $i = 1, n$  ( $n$  = 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
- Învățare supervizată vs. învățare ne-supervizată



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

# Sisteme inteligente – SIS – Învățare automată

---

## Învățare ne-supervizată

- Alte denumiri
  - Clustering
- 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
- Caracteristic
  - Datele nu sunt adnotate (etichetate)
- 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
- Exemple de probleme
  - Gruparea genelor
  - Studii de piață pentru gruparea clienților (segmentarea pieței)
  - news.google.com



# Sisteme inteligente – SIS – Învățare automată

---

## Învățare ne-supervizată

### □ Calitatea învățării (validarea clusterizării):

- Criterii interne → Similaritate ridicată în interiorul unui cluster și similaritate redusă între clusteri
  - Distanța în interiorul clusterului
  - Distanța între clusteri
  - Indexul Davies-Bouldin
  - Indexul Dunn
- Criterii externe → Folosirea unor benchmark-uri formate din date pre-grupate
  - Compararea cu date cunoscute – în practică este imposibil
  - Precizia
  - Rapelul
  - F-measure

# Sisteme inteligente – SIS – Învățare automată

## Învățare ne-supervizată

### Calitatea învățării → 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  $D_{nn}(c_j) = \sum_{x_{i1} \in c_j} \min_{x_{i2} \in c_j} ||x_{i1} - x_{i2}|| / n_j$
  - Distanța între centroizi  $D_c(c_j) = \sum_{x_i \in c_j} ||x_i - \mu_j|| / n_j$ , unde  $\mu_j = 1/n_j \sum_{x_i \in c_j} x_i$
- 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  $d_{ce}(c_{j1}, c_{j2}) = ||\mu_{j1} - \mu_{j2}||$
- Indexul Davies-Bouldin → min → clusteri compacți
  - $DB = 1/nc * \sum_{i=1, 2, \dots, nc} \max_{j=1, 2, \dots, nc, j \neq i} ((\sigma_i + \sigma_j)/d(\mu_i, \mu_j))$ , unde:
    - $nc$  – numărul de clusteri
    - $\mu_i$  – centroidul clusterului  $i$
    - $\sigma_i$  – media distanțelor între elementele din clusterul  $i$  și centroidul  $\mu_i$
    - $d(\mu_i, \mu_j)$  – distanța între centroidul  $\mu_i$  și centroidul  $\mu_j$
- 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

# Sisteme inteligente – SIS – Învățare automată

## 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

##### □ Ne-ierarhic

- Partițional → se determină o împărțire a datelor → toți clusterii deodată
- Optimizează o funcție obiectiv definită local (doar pe anumite atribute) sau global (pe toate atributele) care 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)
  - Pe modele probabilistice (ex. Identificarea distribuției datelor → Maximizarea așteptărilor)
  - Pe cel mai apropiat vecin
- Necesită fixarea apriori 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

##### □ 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

##### □ 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

# Sisteme inteligente – SIS – Învățare automată

## 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 centrul fiecărui cluster
3. Se determină o împărțire a datelor
4. Se recalculează centrul clusterelor
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

#### ■ După tipul de apartenență al datelor la clusteri

##### □ Clustering exact (*hard clustering*)

- Asociază fiecărei intrări  $x_i$  o etichetă (clasă)  $c_j$

##### □ Clustering fuzzy

- Asociază fiecărei intrări  $x_i$  un grad (probabilitate) de apartenență  $f_{ij}$  la o anumită clasă  $c_j \rightarrow$  o instanță  $x_i$  poate aparține mai multor clusteri

# 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
  - Algoritmi evolutivi
  - Mașini cu suport vectorial
  - Modele Markov ascunse

# Sisteme inteligente – SIS – Învățare automată

## □ Învățare activă

- Algoritmul de învățare poate primi informații suplimentare în timpul învățării pentru a-și îmbunătăți performanța
  - Ex. pe care din datele de antrenament este mai ușor să se învețe modelul de decizie

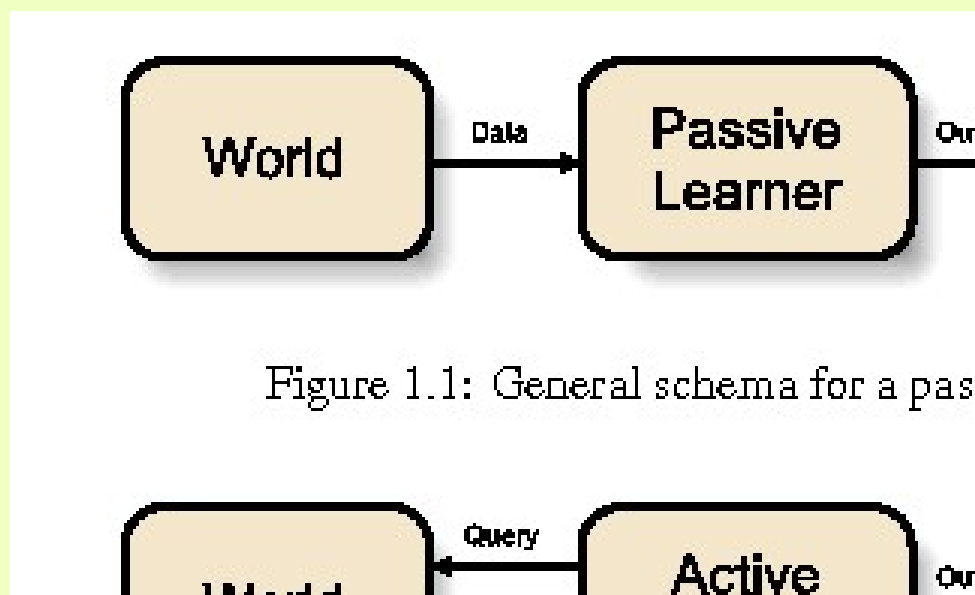


Figure 1.1: General schema for a pas:

# 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)
  - **Metoda celor mai mici pătrate**
  - Metoda gradient descent
  - Algoritmi evolutivi
  - Logistic regression
  - kNN
  - Arbori de decizie
  - Mașini cu suport vectorial
  - Rețele neuronale artificiale
  - Programare genetică
  - Modele Markov ascunse

# Recapitulare



- Sisteme care învață singure (SIS)
  - Instruire (învățare) automata (Machine Learning - ML)
    - Învățare supervizată → datele de antrenament sunt deja etichetate cu elemente din  $E$ , iar datele de test trebuie etichetate cu una dintre etichetele din  $E$  pe baza unui model (învățat pe datele de antrenament) care face corespondența date-etichete
    - Învățare nesupervizată → datele de antrenament NU sunt etichetate, trebuie învățat un model de etichetare, iar apoi datele de test trebuie etichetate cu una dintre etichetele identificate de model
  - Sisteme



# 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 care învață singure
  - Metoda celor mai mici patrate, Gradient Descent, Logistic regression
  - Arbori de decizie
  - Rețele neuronale artificiale
  - Mașini cu suport vectorial
  - Algoritmi evolutivi
- Sisteme bazate pe reguli
- Sisteme hibride

## Cursul următor – Materiale de citit și legături utile

---

- ❑ Capitolul VI (19) din *S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995*
- ❑ capitolul 8 din *Adrian A. Hopgood, Intelligent Systems for Engineers and Scientists, CRC Press, 2001*
- ❑ capitolul 12 și 13 din *C. Groșan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011*
- ❑ Capitolul V din *D. J. C. MacKey, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003*
- ❑ Capitolul 4 din *T. M. Mitchell, Machine Learning, McGraw-Hill Science, 1997*

---

□ 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](http://www.cs.ubbcluj.ro/~moltean)

■ Lect. Dr. Crina Groșan -  
[www.cs.ubbcluj.ro/~cgrosan](http://www.cs.ubbcluj.ro/~cgrosan)

■ Prof. Dr. Horia F. Pop -  
[www.cs.ubbcluj.ro/~hfpop](http://www.cs.ubbcluj.ro/~hfpop)