



METODE INTELIGENTE DE REZOLVARE A PROBLEMELOR REALE

Laura Dioşan
Tema 1

Facultatea de Matematică şi Informatică
Universitatea Babeş-Bolyai



Conținut

- ❑ Instruire automata (Machine Learning - ML)
 - Problematică
 - Proiectarea unui sistem de învățare automată
 - Tipologie
 - ❑ Învățare supervizată
 - ❑ Învățare nesupervizată
 - ❑ Învățare cu întărire
 - ❑ Teoria învățării
- ❑ De citit:
 - S.J. Russell, P. Norvig – Artificial Intelligence - A Modern Approach → capitolul 18, 19, 20
 - www.google.com



Problematică

□ Date → Inteligență

■ Informal:

- Intrări $x \rightarrow$ Program \rightarrow Ieșiri Y
- Similar:
 - Reprezentarea relației dintre intrări și ieșiri
 - Inferența ieșirilor din intrări
 - Învățarea celui mai bun model care descrie datele

■ Formal:

- 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

Problematică

- Se cunosc sau nu ieșirile asociate unor date de antrenament?
 - Învățare supervizată – se cunosc
 - Învățare nesupervizată – nu se cunosc
 - Învățare semi-supervizată – se cunosc parțial

- Tipul de ieșiri
 - Valori discrete → Clasificare
 - Valori real → Regresie
 - Grupuri → Clusterizare

Proiectarea unui sistem de învățare automată



- 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



Proiectare – Alegerea funcției obiectiv

- Care este funcția care trebuie învățată?
 - Ex. pt jocul de dame
 - o funcție care
 - alege următoarea mutare
 - evaluează o mutare
 - obiectivul fiind alegerea celei mai bune mutări

Proiectare – 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, nr. de piese negre compromise la următoarea mutare
- Există un compromis între
 - expresivitatea reprezentării și
 - ușurința învățării
- Calculul funcției obiectiv
 - timp polinomial
 - timp non-polinomial





Proiectare – Alegerea unui model de învățare

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

- Principiul de lucru
 - minimizarea unei erori (funcție de cost – *loss function*)





Proiectare – Învățare automată – tipologie

- **Învățare supervizată**
- Învățare nesupervizată
- Învățare cu întărire



Învățare supervizată

Scop:

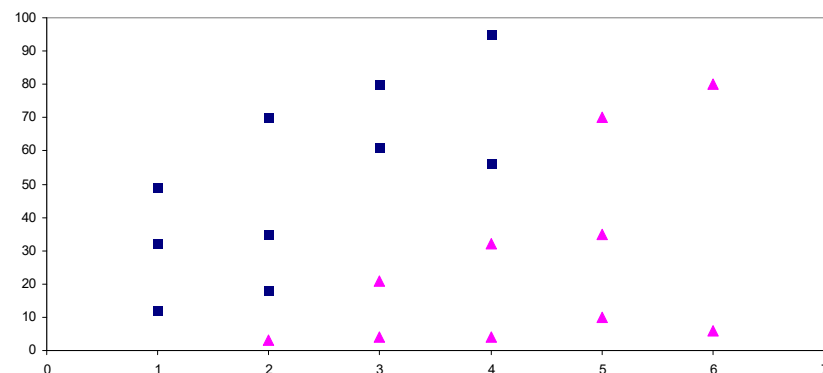
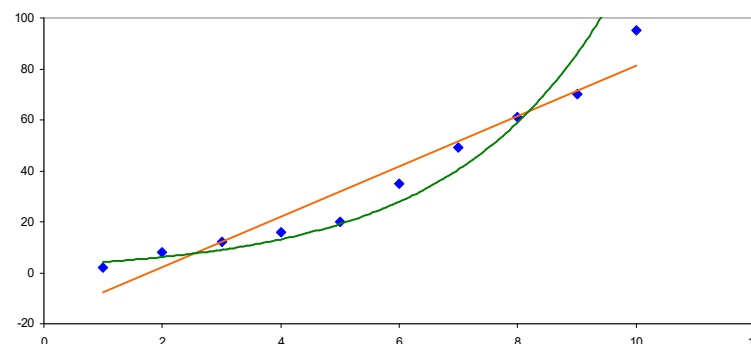
- Furnizarea unei ieșiri corecte pentru o nouă intrare

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

Caracteristic

- BD experimentală adnotată (pt. învățare)



Învățare supervizată – definire

Definire

□ Se dă

- un set de date (exemple, instanțe, cazuri)
 - date de antrenament – sub forma unor perechi (attribute_data_i , ieșire_i), unde
 - $i = 1, N$ (N = nr datelor de antrenament)
 - $\text{attribute_data}_i = (\text{atr}_{i1}, \text{atr}_{i2}, \dots, \text{atr}_{im})$, m – nr atributelor (caracteristicilor, proprietăților) unei date
 - ieșire_i
 - o categorie dintr-o mulțime dată (predefinită) cu k elemente (k – nr de clase) → problemă de clasificare
 - un număr real → problemă de regresie
 - date de test
 - sub forma (attribute_data_i), $i = 1, n$ (n = nr datelor de test).

□ Să se determine

- o funcție (necunoscută) – ipoteză – 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



Învățare supervizată – exemple

- ❑ Recunoașterea scrisului de mână
- ❑ Recunoașterea imaginilor
- ❑ Previziunea vremii
- ❑ Detecția spam-urilor
- ❑ Recomandări de produse
- ❑ Traduceri automate



Învățare supervizată – proces

Procesul

- 2 pași:
 - Antrenarea
 - Învățarea, cu ajutorul unui algoritm, a modelului de clasificare
 - Testarea
 - Testarea modelului folosind date de test noi (*unseen data*)

Calitatea învățării

- 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
 - Ansamblul de date antrenament se împarte în
 - Date de învățare
 - Date de validare
 - Performanța se apreciază pe sub-ansamblul de validare
 - O singură dată
 - De mai multe ori → validare încrucișată (cross-validation)
 - faza de testare
- probleme
 - Învățare pe derost (*overfitting*) → performanță bună pe datele de antrenament, dar foarte slabă pe datele de test

Învățare supervizată – evaluare

Metode de evaluare

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



Învățare supervizată – evaluare

Măsuri de performanță

- ❑ Măsuri statistice
 - Funcții de scor (acuratețe, precizie, etc.)
 - Funcții de loss (măsoară calitatea modelului învățat)
- ❑ 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





Învățare supervizată – evaluare

Măsuri de performanță

□ Măsuri 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



Învățare supervizată – evaluare

Măsuri de performanță

Măsuri statistice

■ Rezultate calculate

■ Precizia și Rapelul

□ 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
- $TP / (TP + FN)$

□ Matricea de confuzie

- Rezultate reale vs. rezultate calculate

■ Scorul F1

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

Măsuri de performanță	Rezultate reale	
	Clasa pozitivă	Clasa(e) negativă(e)
Clasa pozitivă	True positiv (TP)	False positiv (FP)
Clasa(e) negativă(e)	False negative (FN)	True negative (TN)



Învățare supervizată – evaluare

Măsuri de performanță

■ Măsuri statistice

■ Funcția de loss

- Diferența între ieșirea dorită (D) și cea calculată (C)
- L_2 norm - Quadratic cost (*mean squared error*) $\sum ||D - C||^2$
- L_1 norm $\sum |D - C|$
- SVM loss (hinge loss, max-margin loss) $\sum_i \sum_{j, j \neq y_i} \max(C_j - D_{y_i} + \Delta, 0)$
- Softmax loss $\sum [-\ln(\exp(D) / \sum_{j, j \neq y_i} \exp(C_j))]$
- Cross-entropy $-\sum [D \ln C + (1 - D) \ln(1 - C)] / n$





Învățare supervizată – evaluare

Condiții fundamentale

- Distribuția datelor de antrenament și test este aceeași
 - În practică, o astfel de condiție este adesea violată

- Exemplele de antrenament trebuie să fie reprezentative pentru datele de test



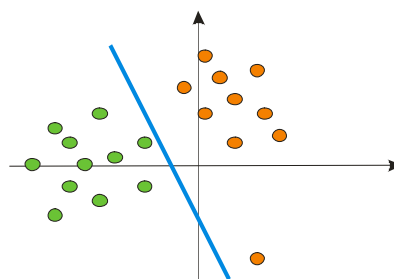
Învățare supervizată – tipologie

- După tipul de date de ieșire
 - Real → probleme de regresie
 - Etichete → probleme de clasificare (regresie logistică)
 - Clasificare binară
 - Ieșiri (output-uri) binare → nr binar de etichete posibile ($k = 2$)
 - Ex. diagnostic de cancer malign sau benign
 - Ex. email acceptat sau refuzat (spam)
 - Clasificare multi-clasă
 - Ieșiri multiple → nr > 2 de etichete posibile ($k > 2$)
 - Ex. recunoașterea cifrei 0, 1, 2,... sau 9
 - Ex. risc de creditare mic, mediu, mare și foarte mare
 - Clasificare multi-etichetă
 - Fiecărei ieșiri îi pot corespunde una sau mai multe etichete
 - Ex. frumos → adjectiv, adverb

Învățare supervizată – tipologie

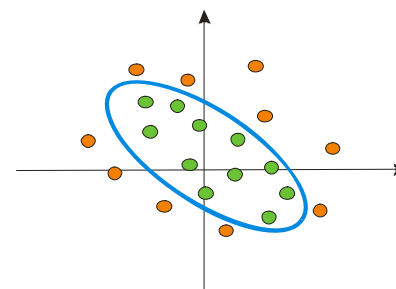
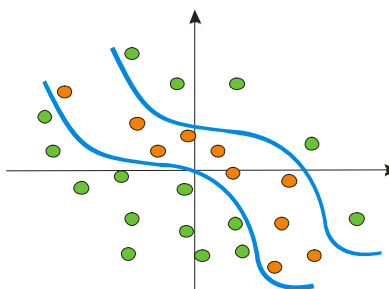
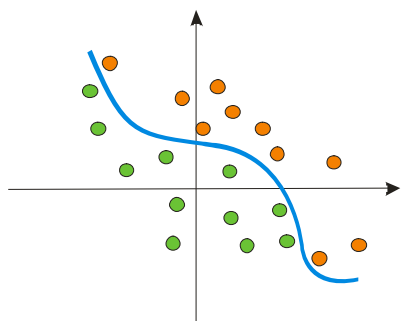
□ După forma clasificatorului

■ Clasificare liniară



■ Clasificare ne-liniară

- se crează o rețea de clasificatori liniari
- se mapează datele într-un spațiu nou (mai mare) unde ele devin separabile

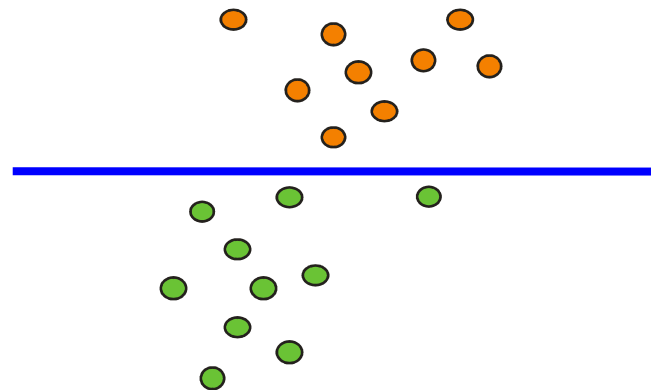


Învățare supervizată – tipologie

□ După caracteristicile datelor

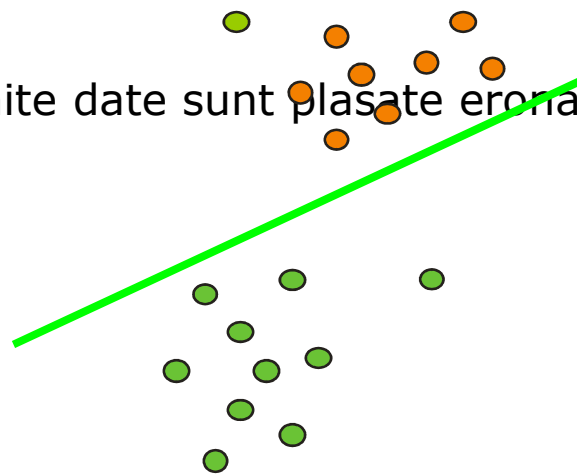
■ Clasificare pt date perfect separabile

□ Clasificare fără eroare



■ Clasificare pt date ne-separabile

□ Clasificare cu o anumită eroare (anumite date sunt plasate eronat în clase)





Învățare supervizată – tipologie

□ După algoritm

- Bazată doar pe instanțe
 - Folosește direct datele, fără a crea un model de separare
 - Ex. algoritmul cel mai apropiat vecin (*k-nearest neighbour*)
- Discriminative
 - Estimează o separare al datelor
 - Ex. arbori de decizie, rețele neuronale artificiale, mașini cu suport vectorial, algoritmi evolutivi
- Generative
 - Construiește un model probabilistic
 - Ex. rețele Bayesiene



Învățare supervizată – algoritmi

- ❑ Cel mai apropiat vecin
 - ❑ Arbori de decizie
 - ❑ Sisteme bazate pe reguli
 - ❑ Rețele neuronale artificiale
 - ❑ Mașini cu suport vectorial
 - ❑ Algoritmi evolutivi
- } clasificare
- regresie

Învățare supervizată – algoritmi

Problemă de clasificare

- 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 = nr datelor de antrenament)
 - $\text{attribute_data}_i = (\text{atr}_{i1}, \text{atr}_{i2}, \dots, \text{atr}_{im})$, m – nr atributelor (caracteristicilor, proprietăților) unei date
 - $\text{ieșire}_i =$
 - o categorie dintr-o mulțime dată (predefinită) cu k elemente (k – nr de clase)
 - date de test – sub forma $(\text{attribute_data}_i)$, $i = 1, n$ (n = nr datelor de test)
 - Să se determine
 - o funcție (necunoscută) care realizează corespondența atribut – ieșire pe datele de antrenament
 - ieșirea (clasa) asociată unei date (noi) de test folosind funcția învățată pe datele de antrenament

Învățare supervizată – algoritmi

Problemă de regresie

- 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 – nr atributelor (caracteristicilor, proprietăților) unei date
 - ieșire_i
 - un număr real
 - 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

Învățare supervizată – algoritmi

Cel mai apropiat vecin (*k*-nearest neighbour)



- ❑ Cel mai simplu algoritm de clasificare
- ❑ În etapa de antrenament, algoritmul doar citește datele de intrare (atributele și clasa fiecărei instanțe)
- ❑ În etapa de testare, pentru o nouă instanță (fără clasă) se caută (printre instanțele de antrenament) cei mai apropiați k vecini și se preia clasa majoritară a acestor k vecini
- ❑ Căutarea vecinilor se bazează pe:
 - distanța Minkowski (Manhattan, Euclidiană) – attribute continue
 - distanța Hamming, Levensthein – analiza textelor
 - alte distanțe (funcții kernel)



Învățare supervizată – algoritmi

kNN



□ Tool-uri

■ Sklearn (python)

- <https://scikit-learn.org/stable/modules/neighbors.html>
- <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

■ Weka (java)

- <https://weka.sourceforge.io/doc.dev/weka/classifiers/lazy/IBk.html>

□ Biblio

- https://github.com/rasbt/stat479-machine-learning-fs19/tree/master/02_knn





Învățare supervizată – algoritmi

Arbori de decizie

□ Scop

- Divizarea unei colecții de articole în seturi mai mici prin aplicarea succesivă a unor reguli de decizie → adresarea mai multor întrebări
 - Fiecare întrebare este formulată în funcție de răspunsul primit la întrebarea precedentă
- Elementele se caracterizează prin informații non-metrice

□ Definire

- Arborele de decizie
 - Un graf special → arbore orientat bicolor
 - Conține noduri de 3 tipuri:
 - Noduri de decizie → posibilitățile decidentului (ex. Diversele examinări sau tratamente la care este supus pacientul) și indică un test pe un atribut al articolului care trebuie clasificat
 - Noduri ale hazardului – evenimente aleatoare în afara controlului decidentului (rezultatul examinărilor, efectul terapiilor)
 - Noduri rezultat – situațiile finale cărora li se asociază o utilitate (apreciată aprioric de către un pacient generic) sau o etichetă
 - Nodurile de decizie și cele ale hazardului alternează pe nivelele arborelui
 - Nodurile rezultat – noduri terminale (frunze)
 - Muchiile arborelui (arce orientate) → consecințele în timp (rezultate) ale deciziilor, respectiv ale realizării evenimentelor aleatoare (pot fi însoțite de probabilități)
- Fiecare nod intern corespunde unui atribut
- Fiecare ramură de sub un nod (atribut) corespunde unei valori a atributului
- Fiecare frunză corespunde unei clase (ieșire de tip discret)





Învățare supervizată – algoritmi

Arbori de decizie

□ Tool-uri

- <http://webdocs.cs.ualberta.ca/~aixplore/learning/DecisionTrees/Applet/DecisionTreeApplet.html>
- WEKA → J48
- <http://id3alg.altervista.org/>
- <http://www.rulequest.com/Personal/c4.5r8.tar.gz>
- <https://scikit-learn.org/stable/modules/tree.html>

□ Biblio

- <http://www.public.asu.edu/~kirkwood/DASTuff/decisiontrees/index.html>
- https://github.com/rasbt/stat479-machine-learning-fs19/tree/master/06_trees



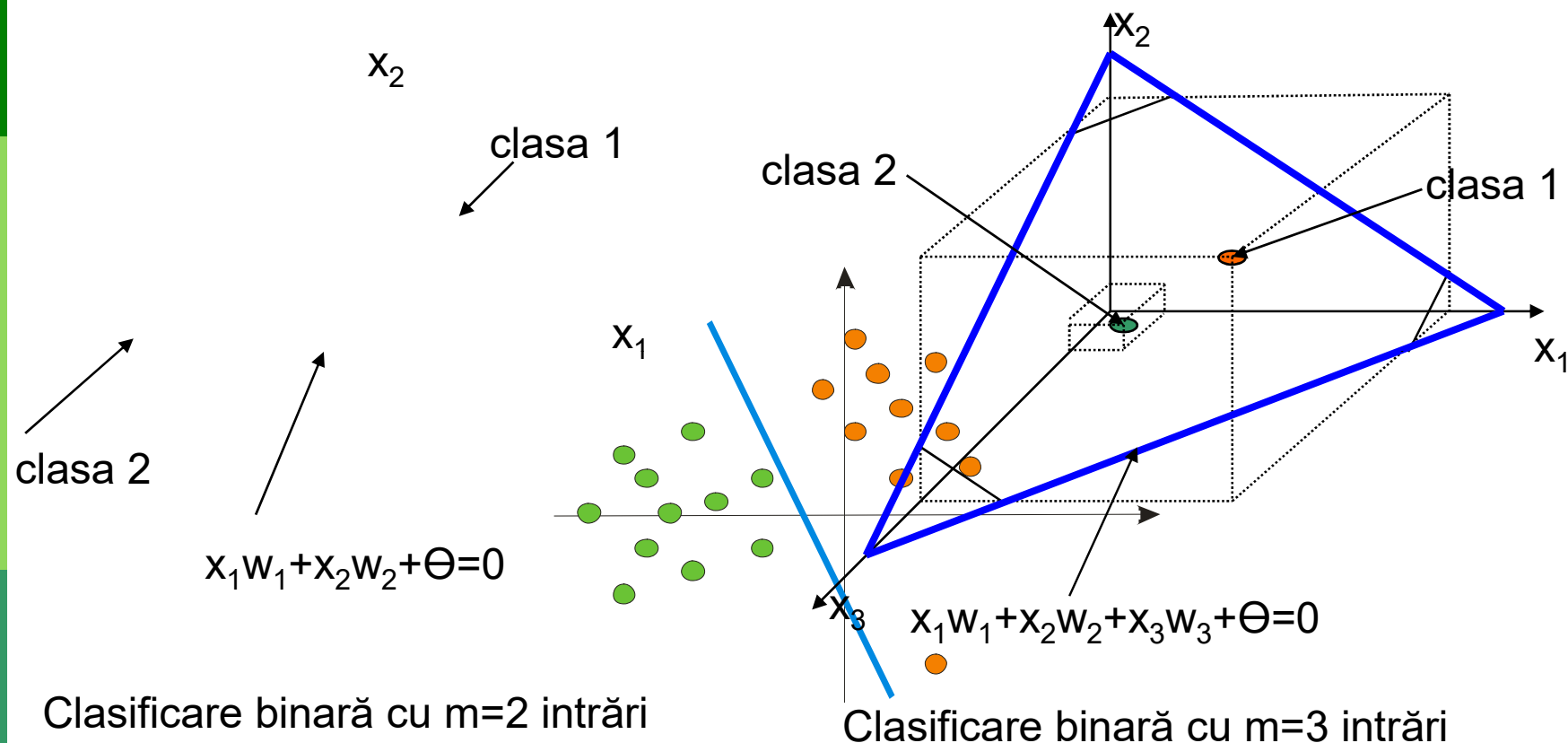
Învățare supervizată – algoritmi

Clasificare - reamintire

- Clasificare binară pt orice fel de date de intrare (discrete sau continue)
 - Datele pot fi separate de:
 - o dreaptă $\rightarrow ax + by + c = 0$ (dacă $m = 2$)
 - un plan $\rightarrow ax + by + cz + d = 0$ (dacă $m = 3$)
 - un hiperplan $\sum a_i x_i + b = 0$ (dacă $m > 3$)
 - Cum găsim modelul de separare (valorile optime pt. a, b, c, d, a_i și forma modelului)?
 - Rețele neuronale artificiale
 - Mașini cu suport vectorial
 - Algoritmi evolutivi

Învățare supervizată – algoritmi

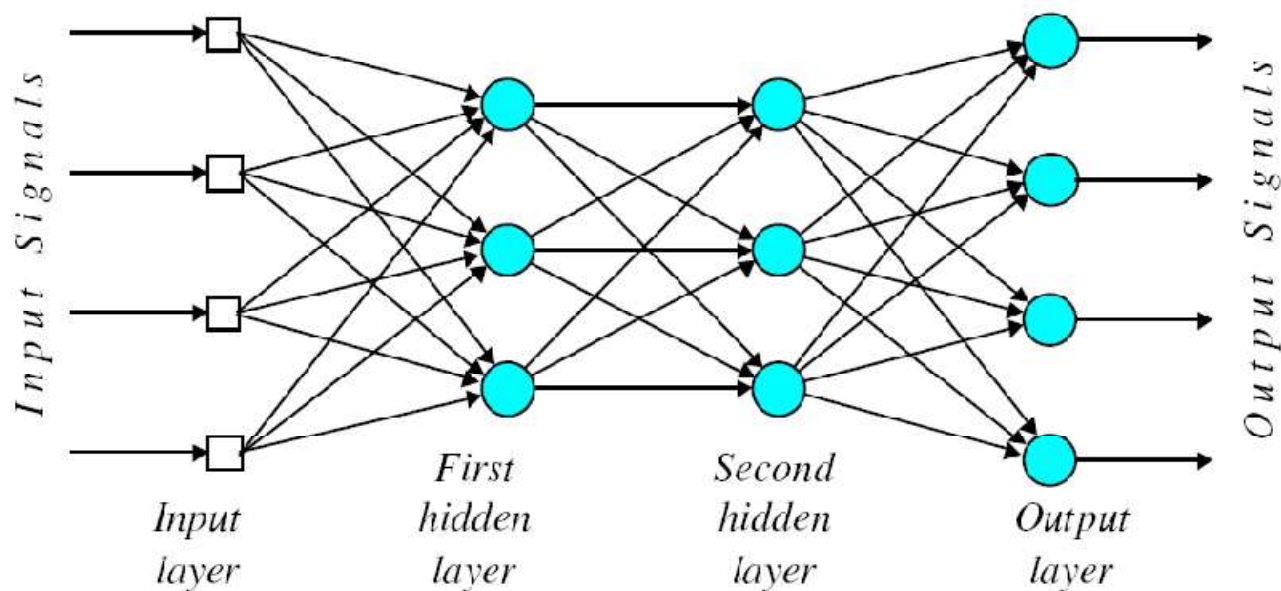
Rețele neuronale artificiale



Învățare supervizată – algoritmi

Rețele neuronale artificiale

- Similar unei rețele neuronale biologice
- O mulțime de neuroni dispuși ca într-un graf (un nod → un neuron) pe mai multe straturi (*layer*)
 - Strat de intrare
 - Conține m (nr de attribute al unei date) noduri
 - Strat de ieșire
 - Conține r (nr de ieșiri) noduri
 - Straturi intermediare (ascunse) – rol în “complicarea” rețelei
 - Diferite structuri
 - Diferite mărimi





Tools

- ❑ Sklearn https://scikit-learn.org/stable/modules/neural_networks_supervised.html
- ❑ WEKA <https://www.cs.waikato.ac.nz/ml/weka/>
- ❑ DL4J <https://deeplearning4j.org>
- ❑ openCV <https://opencv.org/>

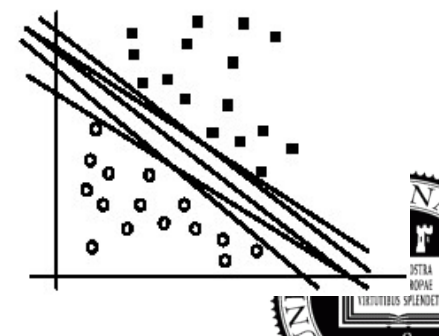
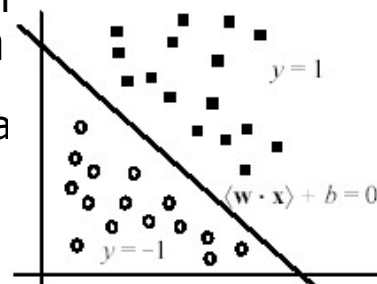
- ❑ Keras
 - NN API
 - <https://keras.io/>
 - + Theano (machine learning library; multi-dim arrays)
<http://www.deeplearning.net/software/theano/>
http://www.iro.umontreal.ca/~lisa/pointeurs/theano_scipy2010.pdf
 - + TensorFlow (numerical computation) <https://www.tensorflow.org/>
- ❑ Pylearn2 <http://deeplearning.net/software/pylearn2/>
 - ML library
 - + Theano
- ❑ Torch <http://torch.ch/>
 - scientific computing framework
 - Multi-dim array
 - NN
 - GPU
- ❑ Caffe
 - deep learning framework
 - Berkley



Învățare supervizată – algoritmi

Mașini cu suport vectorial (MSV)

- ❑ Dezvoltate de Vapnik în 1970 și popularizate după 1992
- ❑ Clasificatori liniari care identifică un hiperplan de separare între clasa pozitivă și cea negativă; au o fundamentare teoretică foarte riguroasă
- ❑ Funcționează foarte bine pentru date de volum mare (ex. analiza textelor, analiza imaginilor)
- ❑ Ideea de bază
 - Hiperplanul de decizie care separă cele 2 clase est
 - ❑ $\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$, unde
 - Date de antrenament de forma $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, unde
 - $\mathbf{x}_i = (x_1, x_2, \dots, x_m)$ este un vector de intrare într-un spațiu real $X \subseteq \mathbb{R}^m$ și
 - y_i este eticheta clasei (valoarea de ieșire), $y_i \in \{1, -1\}$
 - \mathbf{w} – vector de ponderi / coeficienți de importanță
 - Pot exista mai multe hiperplane
 - ❑ Care este cel mai bun?
 - ❑ MSV caută hiperplanul cu cea mai mică eroare de generalizare)
 - Algoritmul SMO (Sequential



Învățare supervizată – algoritmi

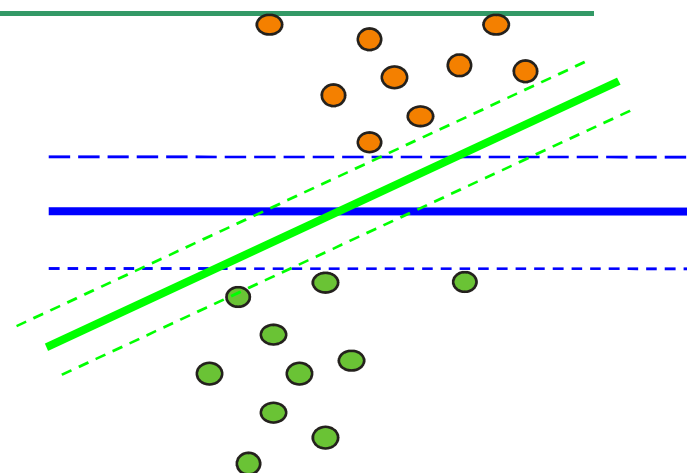
Mașini cu suport vectorial

□ Cazuri de date

■ Liniar separabile

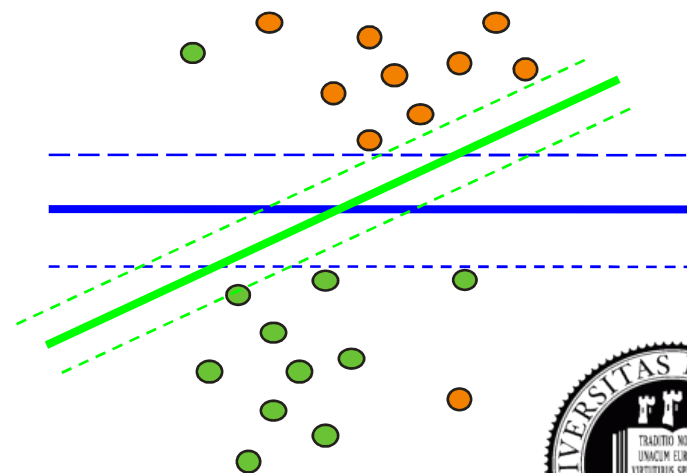
□ Separabile

- Eroarea = 0



□ Ne-separabile

- Se relaxează constrângerile → se permit unele erori
- C – coeficient de penalizare



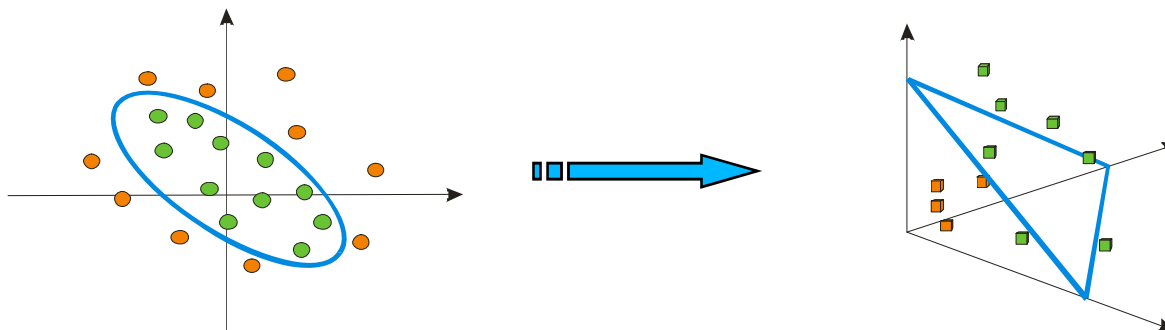
Învățare supervizată – algoritmi

Mașini cu suport vectorial

Cazuri de date

□ Non-linear separabile

- Spațiul de intrare se transformă într-un spațiu cu mai multe dimensiuni (*feature space*), cu ajutorul unei funcții kernel, unde datele devin linear separabile



■ Kernele posibile

□ Clasice

- Polynomial kernel: $K(\mathbf{x}_1, \mathbf{x}_2) = (\langle \mathbf{x}_1, \mathbf{x}_2 \rangle + 1)^d$
- RBF kernel: $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\sigma \|\mathbf{x}_1 - \mathbf{x}_2\|^2)$

□ Kernele multiple

- Liniare: $K(\mathbf{x}_1, \mathbf{x}_2) = \sum w_i K_i$
- Ne-liniare
 - Fără coeficienți: $K(\mathbf{x}_1, \mathbf{x}_2) = K_1 + K_2 * \exp(K_3)$
 - Cu coeficienți: $K(\mathbf{x}_1, \mathbf{x}_2) = K_1 + c_1 * K_2 * \exp(c_2 + K_3)$



Învățare supervizată – algoritmi

Mașini cu suport vectorial

❑ Probleme

- Doar attribute reale
- Doar clasificare binară
- Background matematic dificil

❑ Tool-uri

- LibSVM → <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- Weka → SMO
- SVMLight → <http://svmlight.joachims.org/>
- SVM Torch → <http://www.torch.ch/>
- <http://www.support-vector-machines.org/>



Învățare supervizată – algoritmi

Regresie - reamintim

- Studiul legăturii între variabile
- Se dă
 - un set de date (exemple, instanțe, cazuri)
 - date de antrenament – sub forma unor perechi ($attribute_data_i, ieșire_i$), 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
 - **$ieșire_i$ – un număr real**
 - date de test
 - sub forma (**$attribute_data_i$**), $i = 1, n$ (n = nr datelor de test)
- Să se determine
 - o funcție (necunoscută) care realizează corespondența atribut – ieșire pe datele de antrenament
 - Ieșirea (valoarea) asociată unei date (noi) de test folosind funcția învățată pe datele de antrenament
- Cum găsim forma (expresia) funcției?
 - Algoritmi evolutivi → Programare genetică

Învățare supervizată – algoritmi

Algoritmi evolutivi

Algoritmi	Rezolvarea problemelor
■ Evoluție naturală	Soluție potențială (candidat)
■ Individ	Mulțime de soluții
■ Inspirați din natură (biologie)	Codarea (reprezentarea) unei soluții
■ Populație	Parte a reprezentării
■ Iterativi	Calitate
■ Cromozom	Operatori de căutare
■ Bazați pe	Spațiul de căutare al problemei
■ Gene	
■ Fitness (măsură de calitate)	
■ Incrucișare aleatoare ghidată de	
■ Căutare	
■ Mediu	
■ Operații de selecție naturală	
■ Operații de încrucișare și mutație	

- Care procesează în paralel mai multe soluții

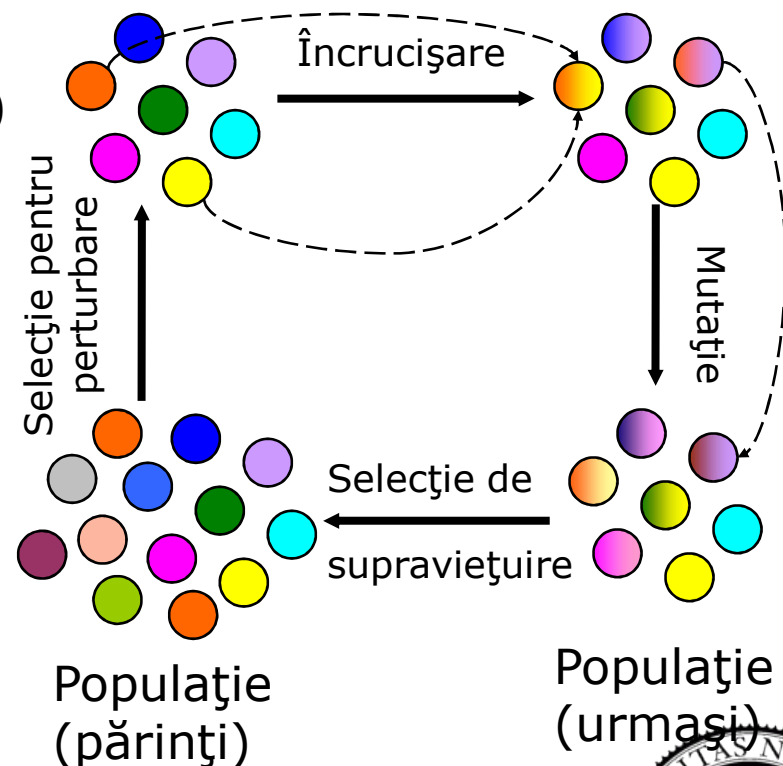
Metafora evolutivă

Învățare supervizată – algoritmi

Algoritmi evolutivi

```

Inicializare 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( $p1, p2$ ) =>  $o1$  și  $o2$ 
        Mutație( $o1$ ) =>  $o1^*$ 
        Mutație( $o2$ ) =>  $o2^*$ 
        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 supervizată – algoritmi

Algoritmi evolutivi – programare genetică

□ Tool-uri

- <https://github.com/JesseBuesking/TinyGP-Java>
- <https://github.com/lfarinha/TinyGP>
- <http://geneticprogramming.com/software/>

□ Referințe

- <http://geneticprogramming.com/>
- <http://www.genetic-programming.com/GPEM2010article.pdf>

□ Reviste științifice

- <https://www.springer.com/journal/10710>





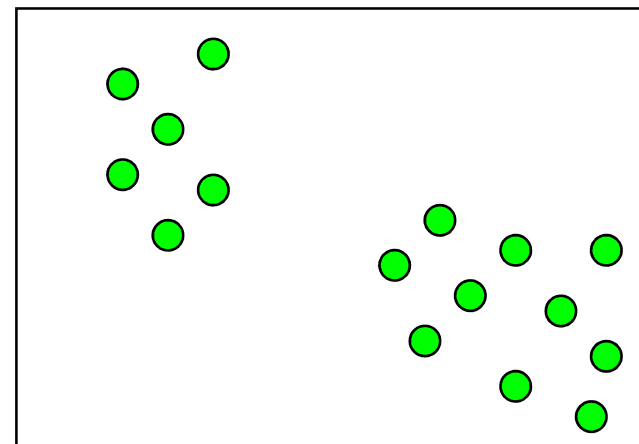
Învățare automată

- Învățare supervizată
- **Învățare ne-supervizată**
- Învățare cu întărire
- Teoria învățării



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 - clustering – 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

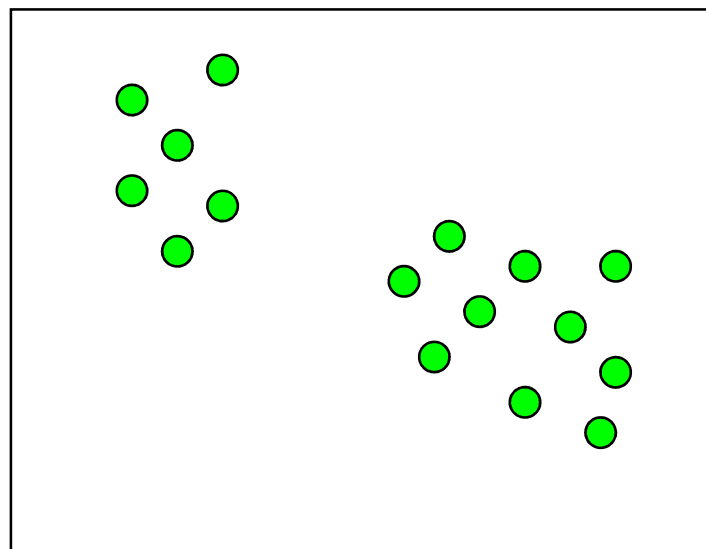
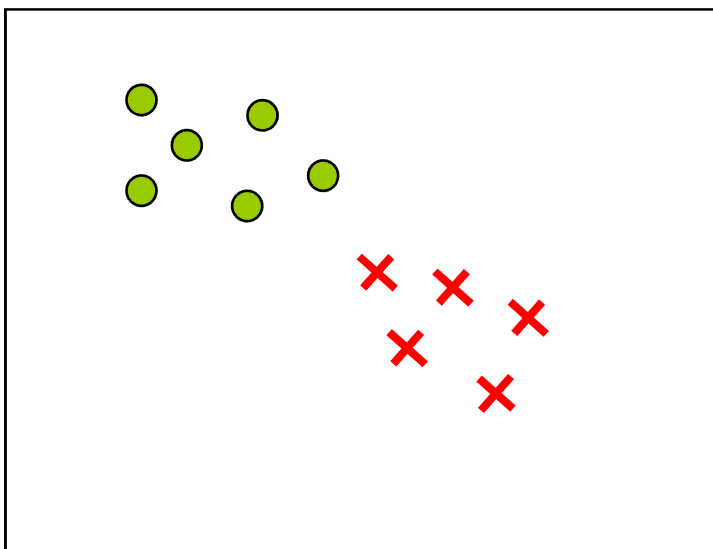
- Se dă
 - un set de date (exemple, instanțe, cazuri)
 - Date de antrenament
 - Sub forma **attribute_data_i**, unde
 - $i = 1, N$ (N = nr datelor de antrenament)
 - **attribute_data_i** = ($atr_{i1}, atr_{i2}, \dots, atr_{im}$), m – nr atributelor (caracteristicilor, proprietăților) unei date
 - Date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

□ Supervizată vs. Ne-supervizată



Învățare ne-supervizată - clustering – 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 - clustering – definire

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





Învățare ne-supervizată - clustering – definire

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 - clustering – definire

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 - clustering – definire

- 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 - clustering – definire

- 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 - clustering – definire

- 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 attributele
 - 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
 - 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 - clustering – definire

- 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 - clustering – definire

- 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 - clustering – definire

- 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 - clustering – definire

- 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 - clustering – algoritmi

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





Învățare ne-supervizată

- ❑ Clustering
- ❑ Reducerea dimensiunii datelor
 - Liniară
 - Ne-liniară – manifold learning
- ❑ Detectția anomaliilor





Învățare nesupervizată

□ Instrumente

- Python – https://scikit-learn.org/stable/unsupervised_learning.html
- Weka <https://www.cs.waikato.ac.nz/~ml/weka/book.html#Contents>
- Orange <https://orange.biolab.si/widget-catalog/>





Învățare automată

- Învățare supervizată
- Învățare ne-supervizată
- **Învățare cu întărire**
- Teoria învățării





Învățare cu întărire

□ Scop

- Învățarea, de-a lungul unei perioade, a unui mod de acțiune (comportament) care să maximizeze recompensele (câștigurile) pe termen lung
- "make good sequences of decisions"

□ Tip de probleme

- Ex. Dresarea unui câine (good and bad dog)

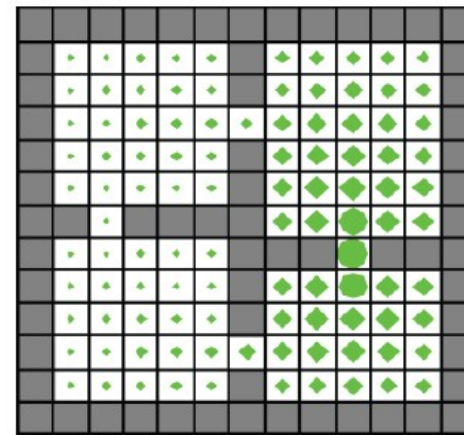
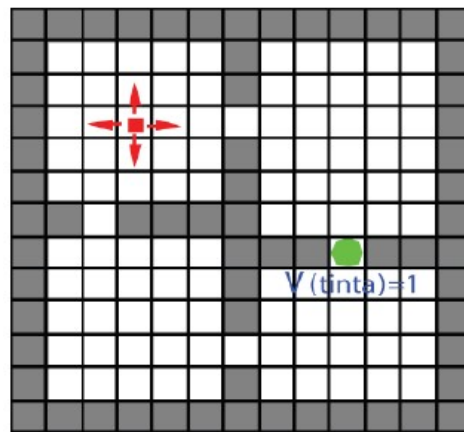
□ Caracteristic

- Interacțiunea cu mediul (acțiuni → recompense)
- Secvență de decizii



Învățare cu întărire – definire

- Exemplu: plecând din căsuța roșie să se găsească un drum până la căsuța verde



- Agentul învață prin interacțiunea cu mediul și prin observarea rezultatelor obținute din aceste interacțiuni
 - Este vorba de "cauză și efect" -- modul în care oamenii își formează cunoașterea asupra mediului pe parcursul vieții
 - Acțiunile pot afecta și recompensele ulterioare, nu numai pe cele imediate (efect întârziat)

Învățare cu întărire – definire

Învățarea unui anumit comportament în vederea realizării unei sarcini → execuția unei acțiuni → primește un feedback (cât de bine a acționat pentru îndeplinirea sarcinii) → execuția unei noi acțiuni

Învățare cu întărire

- Se primește o recompensă (întărire pozitivă) – dacă sarcina a fost bine îndeplinită
- Se primește o pedeapsă (întărire negativă) – dacă sarcina nu a fost bine îndeplinită

Definire

- Se dau
 - Stări ale mediului
 - Acțiuni posibile de executat
 - Semnale de întărire (scalare) – recompense sau pedepse
- Se determină
 - O succesiune de acțiuni care să maximizeze măsura de întărire (recompensa)

Alte denumiri

- Reinforcement learning
- Învățare împrăștiată

Învățare cu întărire – definire

	Plani- ficare	Învățar e super- vizată	Învățare nesuper- vizată	Învățare cu întărire
Optimizare	X			X
Învățare din experiență (adnotată sau nu)		X	X	X
Genralizare	X	X	X	X
Consecințe ulterioare/întârziat	X			X
Explorare				X

Învățare cu întărire – exemple

□ Robotică

- Controlul membrelor
- Controlul posturii
- Preluarea mingii în fotbalul cu roboții

□ Cercetări operaționale

- Stabilirea prețurilor
- Rutare
- Planificarea task-urilor



Învățare cu întărire

□ Instrumente

- Open AI GYM <https://gym.openai.com/>
- PyTorch
- DeepMind Lab

□ Referințe

- David Silver: <https://deepmind.com/learning-resources/-introduction-reinforcement-learning-david-silver>
- Sutton and Burton's book <http://www.incompleteideas.net/book/RLbook2020.pdf>





Învățare automată

□ Instrumente generale

- Weka <https://www.cs.waikato.ac.nz/ml/weka/>
- Scikit-Learn <http://scikit-learn.org/stable/>
- Pattern <https://www.clips.uantwerpen.be/pattern>
- Rapid Miner <https://rapidminer.com/>
- Orange <https://orange.biolab.si/>

□ Reviste

- Machine Learning <https://www.springer.com/journal/10994>
- IEEE Transactions on Neural Networks and Learning Systems
<https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=5962385>
- Pattern Recognition
<https://www.journals.elsevier.com/pattern-recognition>
- ...



Instrumente – analiză comparativă

	R-Programming	RapidMiner	Weka	Orange	Scikit	Shogun	Mlib
Open source	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Language based	R Language	Java	Java	Python, C++, Qt	Python library	Java/Python library	Java/Python library
Partitioning of dataset into training sets	Yes (limited partitioning methods)	Yes	Yes	Yes	Yes	Not mentioned	Not mentioned
Parameter optimization of machine learning methods	Not automatic	Yes	Not automatic	Not automatic	Not automatic	Not automatic	Not automatic
Model validation using cross-validation	Yes (but limited error measurement methods)	Yes	Yes (but cannot save model so you have to rebuild it for future experiments)	Yes (but cannot save model so you have to rebuild it for future experiments)	Yes	Yes	Not mentioned
Data visualization and analysis	Yes (also graphics visualization)	Yes	Yes	Yes	Data visualization for SOM, Cross-validated prediction	No	No
Intuitive GUI	Not very intuitive (a lot of graphics and statistics computations)	Not very intuitive (took a while to understand the flow)	Yes (easiest GUI to learn and use)	Yes	No GUI	No Gui	No Gui
Installation	Hard	Easy	Easy	Hard	Easy (command-line)	Easy (integration of jars)	Easy
Numerical Programming	Based on powerful array language			Needs external packages (e.g. numpy)	Yes, similar with numpy	Not mentioned	Not mentioned
Illegal Workflow	Not mentioned	Suggests quick fixes	Not mentioned	Does not compute	Not mentioned	Not mentioned	Not mentioned
Machine Learning Methods	Less specialized in data mining, focus on statistical calculations	Includes also algorithms from Weka	The most powerful and complete	Based mostly on data vizualization (clustering, SOM, DT);	Supervised, unsupervised methods	Clustering, regression, ANN	Regression, clustering, colaborative filtering
Input Files	Connectivity to DB, exports data to excel format	Handles DBs, csv	Worst connectivity to excel spreadsheet	Handles excel and cvs files	Python based for handling files	Java handling files	
Tutorials & Documentation	Light Documentation	Manual and tutorials	Help menu	Complete tutorials for different machine learning algorithms	Lot of tutorials	Poor documentation	Poor documentation