METODE INTELIGENTE DE REZOLVARE A PROBLEMELOR REALE

Laura Dioşan

Tema 1
Algoritmi de învățare automată

UNIVERSITATEA BABEŞ-BOLYAI
Facultatea de Matematică şi Informatică

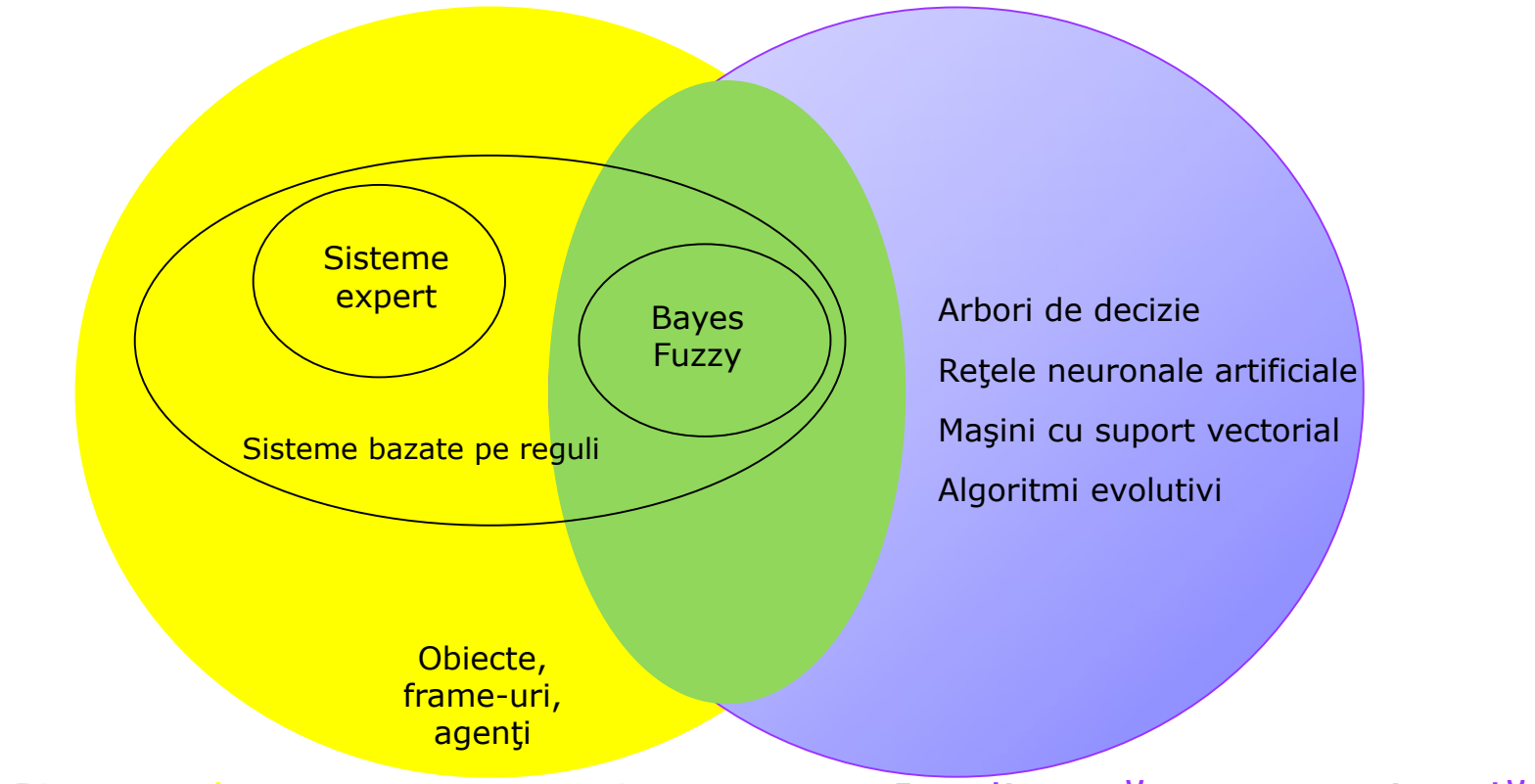
Materiale de citit şi legături utile

- capitolul VI (18) din *S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995*
- capitolul 10 şi 11 din *C. Groşan, 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*
- *capitolul 1 din C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006*
- *capitolul 1 din S. Guido, A. C. Müller, Introduction to Machine Learning with Python, O'Reilly Media, 2016*
- *capitolele 1 și 2 din A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 2019*

Conținut

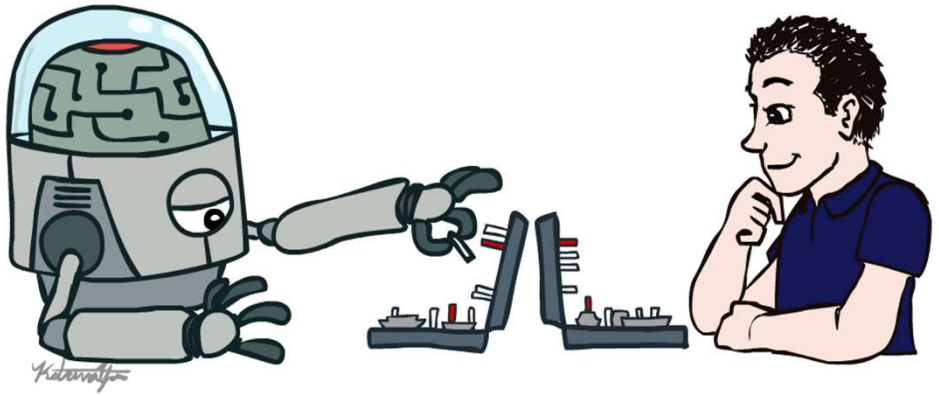
- Sisteme inteligente care învaţă singure (SIS)
  - Problematică
  - Proiectarea unui sistem de învăţare automată
  - Tipologie
    - **Învăţare supervizată**
    - Învăţare nesupervizată
    - Învăţare cu întărire
    - Teoria învăţării
  - Exemple de sisteme

Sisteme inteligente



Sisteme inteligente care învaţă singure – SIS

- Învățare automată = Machine Learning
- Problematica
  - *"How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?"*
- Aplicaţii
  - Recunoaştere de imagini şi semnal vocal
    - Recunoaşterea scrisului de mână
    - Detecţia feţelor
    - Înţelegerea limbajului vorbit
  - Computer vision
    - Detecţia obstacolelor
    - Recunoaşterea amprentelor
  - Supraveghere bio
  - Controlul roboţilor
  - Predicţia vremii
  - Diagnosticare medicală
  - Detecţia fraudelor

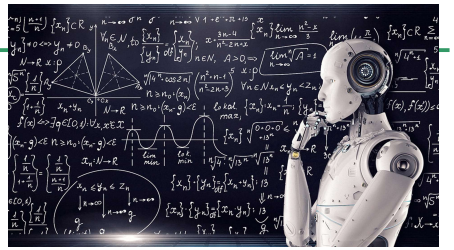


Sisteme inteligente care învaţă singure – SIS

- Definire
  - Arthur Samuel (1959)
    - "field of study that gives computers the ability to learn without being explicity programmed"
    - Înzestrarea computerelor cu abilitatea de a învăţa pe baza experienţei
  - 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.
  - John L. Hennessy, President of Stanford (2000–2016)
    - Machine learning is the hot new thing
  - Bill Gates (Microsoft co-founder)
    - A breakthrough in machine learning would be worth ten Microsofts

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

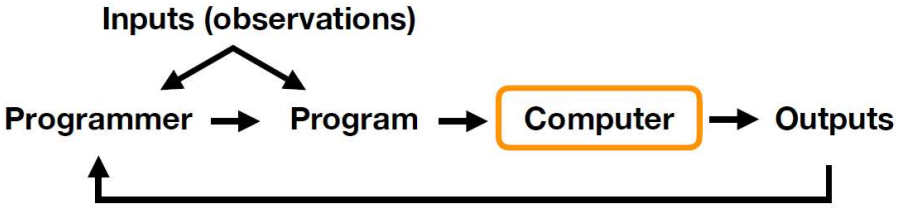
Sisteme inteligente care învață singure – SIS

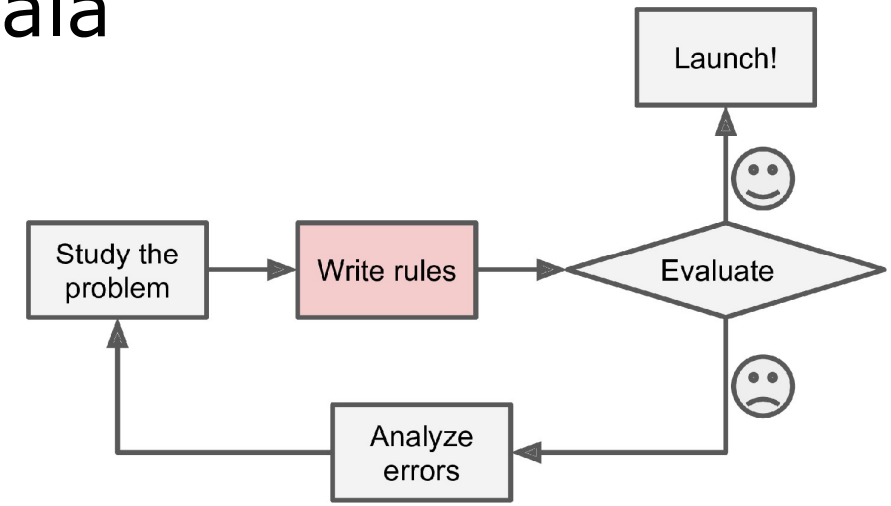


- Persoane importante și/sau interesante
  - Peter Norvig – Director of Research Google norvig.com/
  - Stuart Russell – Professor Berkeley people.eecs.berkeley.edu/~russell
  - Michael Jordan – Professor Berkley www2.eecs.berkeley.edu/Faculty/Homepages/jordan.html
    - Andrew Ng – www.DeepLearning.ai
  - Elon Musk SpaceX, Tesla
  - Fei Fei Li (AI for social good) – Professor Stanford https://profiles.stanford.edu/fei-fei-li
    - Andrej Karpathy – Tesla https://karpathy.ai
  - Richard Sutton – Professor Alberta, Deepmind http://incompleteideas.net
  - Geoffrey Hinton , Yann LeCun, and Yoshua Bengio (CNN and deep CNN)
  - John Koza (Genetic Programming)
  - Rana el Kaliouby (Affectiva)
  - **Horia F. Pop, Anca Andreica, Camelia Chira, Gabriela Czibula – UBB**
  - …alții…

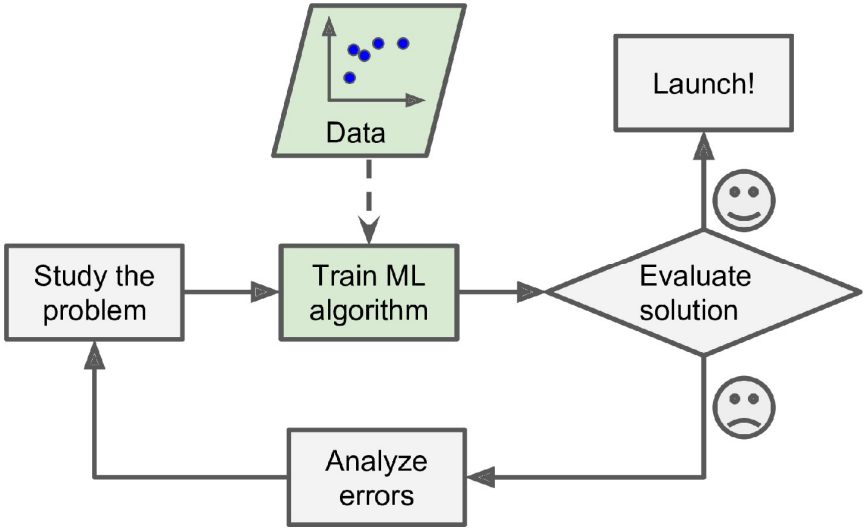
Sisteme inteligente care învață singure – SIS

- Programarea tradițională





- Machine Learning

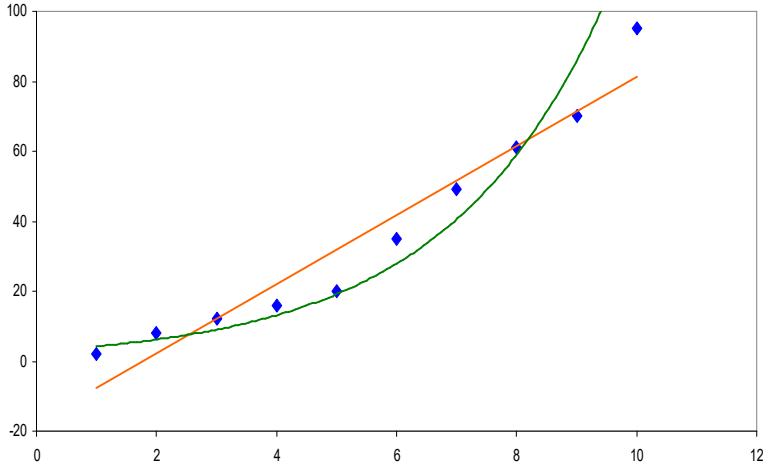


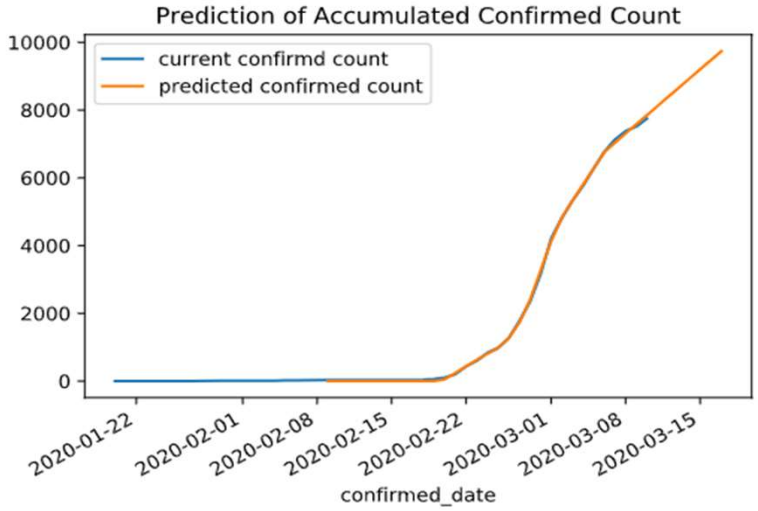
Sisteme inteligente care învaţă singure – SIS

- Proiectare
  - Îmbunătăţirea task-ului T
    - Stabilirea scopului (ceea ce trebuie învăţat) - funcţiei obiectiv – şi reprezentarea sa
    - Alegerea unui algoritm de învăţare care să realizeze inferenţa (previziunea) scopului pe baza experienţei
  - respectând o metrică de performanţă P
    - Evaluarea performanţelor algortimului ales
  - bazându-se pe experienţa E
    - Alegerea bazei de experienţă
  - Exemplu
    - T: jucarea jocului de dame
    - P: procentul de jocuri câştigate împotriva unui oponent oarecare
    - E: exersarea jocului împotriva lui însuşi
    - T: recunoaşterea 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 obişnuite
    - P: procentul de email-uri corect clasificate (spam sau normal)
    - E: baze de date cu email-uri adnotate

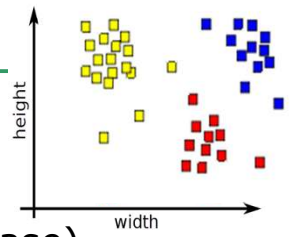
Sisteme inteligente care învaţă singure – SIS

- Proiectare → Stabilirea scopului (ceea ce trebuie învăţat)
  - SIS pentru predicţii / regresii
    - Scop: predicţia ieşirii pentru o intrare nouă folosind un model de predicție învăţat anterior
    - Exemplu: predicţia îmbolnăvirilor cu SARS-CoV-2
      - Se dau: numărul de persoane infectate cu SARS-CoV-2 pentru ultimele 3 luni, adică date (de intrare și ieșire) trecute
      - Se cer predicții viitoare pentru numărul de persoane care se vor infecta cu SARS-CoV-2 in urmatoarele 7 zile / 4 saptamani / 2 luni

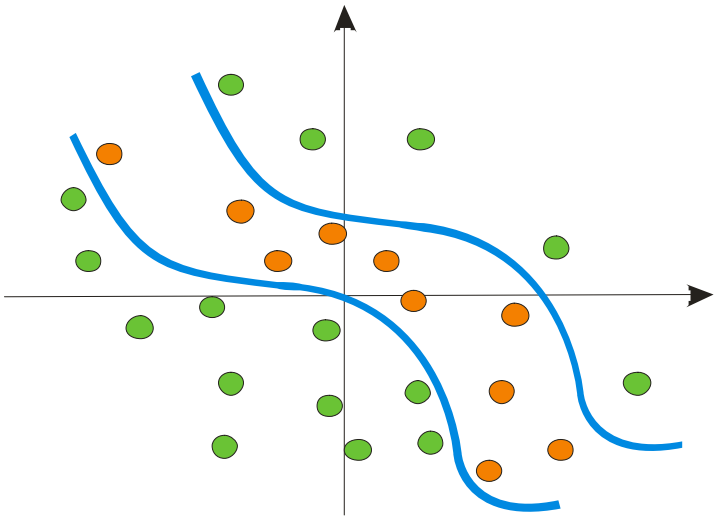


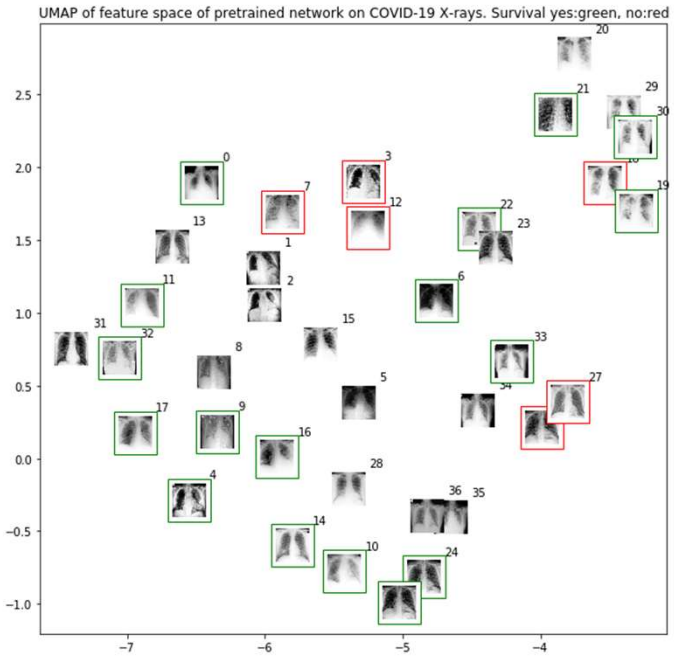


Sisteme inteligente care învață singure – SIS



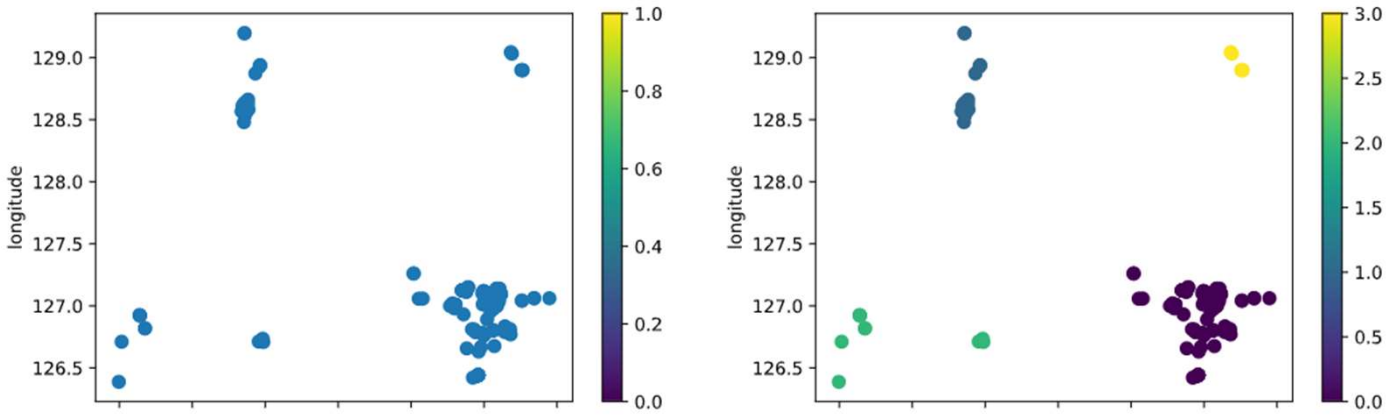
- Proiectare → Stabilirea scopului (ceea ce trebuie învăţat)
  - SI pentru clasificare
    - Scop: clasificarea unui obiect într-una sau mai multe categorii (clase) – cunoscute anterior sau nu - pe baza caracteristicilor (atributelor, proprietăţilor)
    - Exemplu: sistem de diagnoză pentru un pacient cu infecție SAR-CoV-2
      - Se dau imagini RMN de la pacienți infectați cu SARS-C0V-2 și de la martori (sănătoși), adică date (de intrare și ieșire) trecute
      - Se cere să se prezică, pe baza RMN-ului, dacă o persoană este infectată sau nu cu SARS-CoV-2





Sisteme inteligente care învaţă singure – SIS

- Proiectare → Stabilirea scopului (ceea ce trebuie învăţat)
  - SI pentru clusterizare
    - Scop: clasificarea unui obiect într-una sau mai multe categorii (clase) – necunoscute anterior - pe baza caracteristicilor (atributelor, proprietăţilor)
    - Ex.: sistem de control a modului de răspândire a unui virus
      - Se dau localizarea geografică a unor persoane infectate cu SARS-CoV-2, adică date (de intrare)
      - Se cere identificarea modului de grupare a celor infectați pe regiuni (Densitatea acestor regiuni), adică identificarea anumitor structuri în aceste date

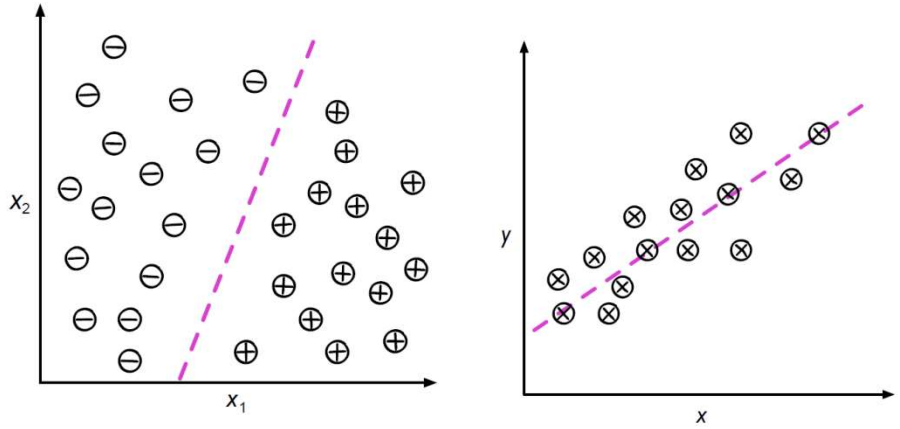


Sisteme inteligente care învaţă singure – SIS

- Proiectare → Alegerea funcţiei obiectiv
  - Care este funcţia care trebuie învăţată?
    - Ex.: pentru jocul de dame → funcţie care:
      - alege următoarea mutare
      - evaluează o mutare
    - obiectivul fiind alegerea celei mai bune mutări
- Proiectare → Alegerea unui algoritm de învăţare
  - Tipuri de algoritmi după metodologia de învățare automată
    - Învățare supervizată
      - Ex. regresie, clasificare
    - Învățare nesupervizată
      - Ex. clusterizare, reducerea numărului de dimensiuni
    - Învățare prin întărire
      - Ex. planning, gaming

Sisteme inteligente care învaţă singure – SIS

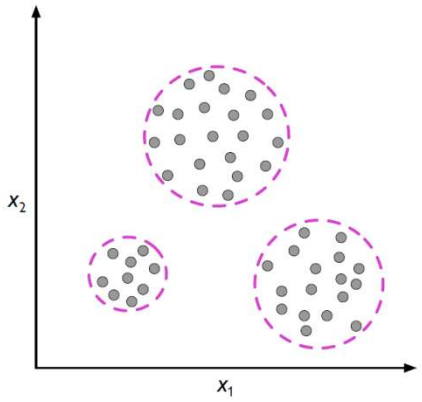
- Proiectare → Alegerea unui algoritm de învăţare → Tipuri de algoritmi după metodologia de învățare automată
  - Învățare supervizată
    - Ex. regresie, clasificare
    - Caracteristici
      - Date etichetate (se cunosc o parte din datele de intrare și ieșire *)
      - Feedback direct în timpul învățării – algoritmul se adaptează la datele de intrare și ieșire
      - Predicție a datelor de ieșire (fiind cunoscute niște date de intrare diferite de cele din *)

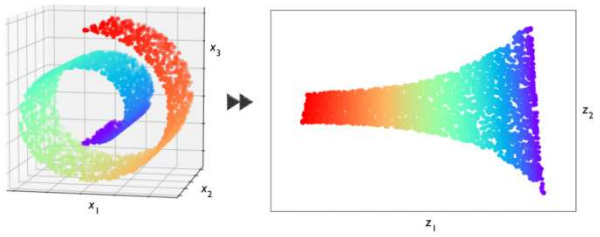


)

Sisteme inteligente care învaţă singure – SIS

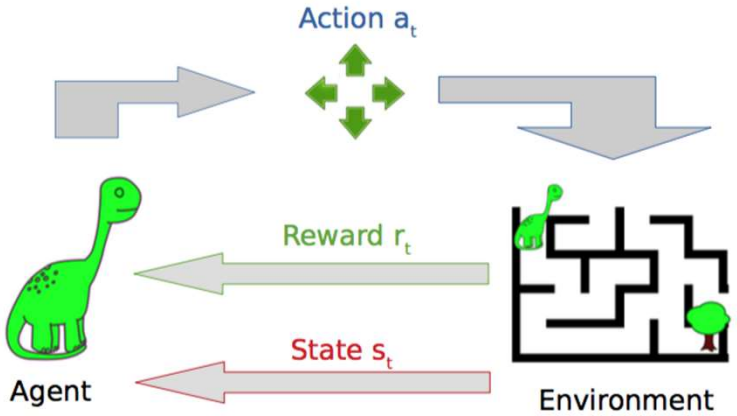
- Proiectare → Alegerea unui algoritm de învăţare → Tipuri de algoritmi după metodologia de învățare automată
  - Învățare supervizată
  - Învățare nesupervizată
    - Ex. clusterizare, reducerea numărului de dimensiuni
    - Carcteristici
      - Date neetichetate (se cunosc o parte din datele de intrare**)
      - Fără feedback direct în timpul învățării – pentru că nu se cunosc datele de ieșire
      - Identificarea unor structuri în date (generarea de date de ieșire pentru datele de intrare din **)





Sisteme inteligente care învaţă singure – SIS

- Proiectare → Alegerea unui algoritm de învăţare → Tipuri de algoritmi după metodologia de învățare automată
  - Învățare supervizată
  - Învățare nesupervizată
  - Învățare prin întărire
    - Ex.
    - Caracteristici
      - Predicția unor secvențe de decizii / de acțiuni
      - Sistem de recompense (pentru fiecare decizie / acțiune)
      - Se învață un model de acțiune (o serie de acțiuni care trebuie efectuate)



Sisteme inteligente care învaţă singure – SIS

- Proiectare → Evaluarea unui sistem de învăţare
  - Experimental
    - Compararea diferitelor metode pe diferite date (cross-validare)
    - Colectarea datelor pe baza performanţei
      - Acurateţe, timp antrenare, timp testare
    - Aprecierea diferenţelor dpdv statistic
  - Teoretic
    - Analiza matematică a algoritmilor şi demonstrarea de teoreme
      - Complexitatea computaţională
      - Abilitatea de a se potrivi cu datele de antrenament
      - Complexitatea eşantionului relevant pentru o învăţare corectă

Sisteme inteligente care învaţă singure – SIS

- Proiectare → Alegerea bazei de experienţă
  - Tipuri de atribute ale datelor
    - Cantitative → scară nominală sau raţională
      - 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 → maşină, autobus, tractor, camion)
  - Transformări asupra datelor
    - Standardizare → atribute numerice
      - Înlăturarea efectelor de scară (scări şi unităţi 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 instanţei i, μ_j (σ_j) este media (abaterea) atributelor j pt. toate instanţele
    - Selectarea anumitor atribute

Sisteme inteligente care învaţă singure – SIS

Învăţare supervizată

- Scop
  - Furnizarea unei ieşiri corecte pentru o nouă intrare

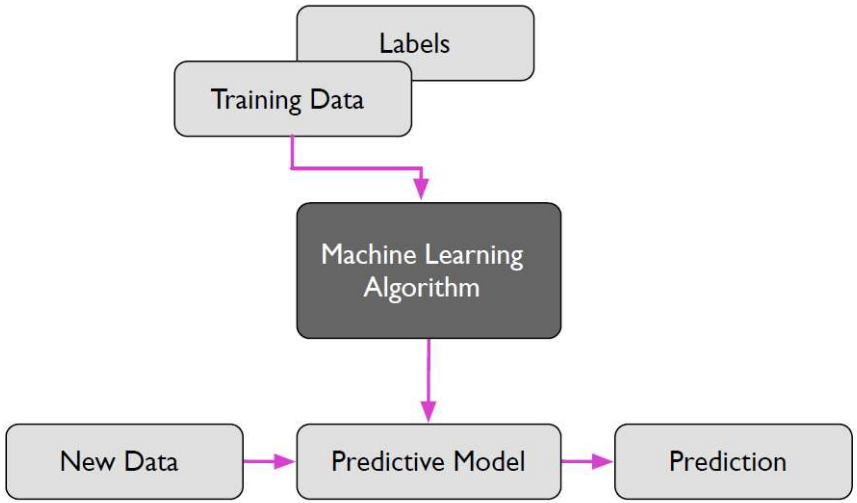
- Definire
  - Se dă un set de date (exemple, instanţe, cazuri)
    - date de antrenament – sub forma unor perechi (atribute_data$_i$, ieşire$_i$), unde
      - i =1,N (N = nr datelor de antrenament)
      - atribute_data$_i$= (atr$_{i1}$, atr$_{i2}$, ..., atr$_{im}$), m – nr atributelor (caracteristicilor, proprietăţilor) unei date
      - ieşire$_i$
        - o categorie dintr-o mulţime dată (predefinită) cu k elemente (k – nr de clase) ➔ problemă de clasificare
        - un număr real ➔ problemă de regresie
    - date de test - sub forma (atribute_datai), i =1,n (n = nr datelor de test).
  - Să se determine
    - o funcţie (necunoscută) care realizează corespondenţa atribute – ieşire pe datele de antrenament
    - ieşirea (clasa/valoarea) asociată unei date (noi) de test folosind funcţia învăţată pe datele de antrenament

- Alte denumiri
  - Clasificare (regresie), învăţare inductivă

Sisteme inteligente care învaţă singure – SIS

Învăţare supervizată

- Proces → 2 etape
  - Antrenarea
    - Învăţarea, cu ajutorul unui algoritm, a modelului de predicție
  - Testarea
    - Testarea modelului folosind date de test noi (*unseen data*)



- Caracteristic
  - BD experimentală adnotată (pt. învăţare)

Sisteme inteligente care învaţă singure – SIS

Învăţare supervizată

- Tip de probleme
  - regresie
    - Scop: predicţia output-ului pentru un input nou
    - Output continuu (nr real)
    - Ex.: predicţia ratei șomajului în funcție de produsul intern brut și rata inflației
  - clasificare
    - Scop: clasificarea (etichetarea) unui nou input
    - Output discret (etichetă dintr-o mulţime predefinită)
    - Ex.: detectarea tumorilor maligne în imagini RMN

- Exemple de probleme
  - Recunoaşterea scrisului de mână
  - Recunoaşterea pietonilor în imagini
  - Previziunea vremii
  - Detecţia spam-urilor

Sisteme inteligente care învață singure – SIS

- Învățare supervizată
  - Terminologie – e.g. Problema predicției consumului de înghețată pe baza temperaturii de afară și a sumei de bani avută la dispoziție
    - Exemplu (example, observation, instance, record)
      - o observație a datelor care trebuie procesate
      - dacă datele de intrare sunt tabelare, un exemplu este asociat cu o linie din tabel
      - format din proprietăți a datelor care trebuie procesate (de intrare și de ieșire)
    - Caracteristică (feature, property, attribute)
      - Proprietate cunoscută a unui exemplu, folosită drept dată de intrare pentru algoritmul de ML (variabilele independente din modelul de predicție)
      - dacă datele de intrare sunt tabelare, un o proprietate are asociate valorile dintr-o coloana a tabelului (pentru toate exemplele)
      - E.g. Temperatura, banii
    - Valoare țintă (target or real value/label, ground-truth)
      - Proprietate a unui exemplu folosită ca variabilă dependentă
      - Cunoscută pentru exemplele de antrenament
      - Ne-cunoscută pentru exemplele de testare
      - E.g. Nr de inghetate
    - Valoare calculată (computed value/label)
      - Proprietate a unui exemplu estimată cu ajutorul algoritmului de ML
      - Se dorește a fi cât mai aproape de valoarea target

Exemplu	Temperatura	Banii	Nr de inghetate
Ex1	30	25	2
Ex2	5	100	0
Ex3	19	55	2
Ex4	35	75	4

Sisteme inteligente care învaţă singure – SIS

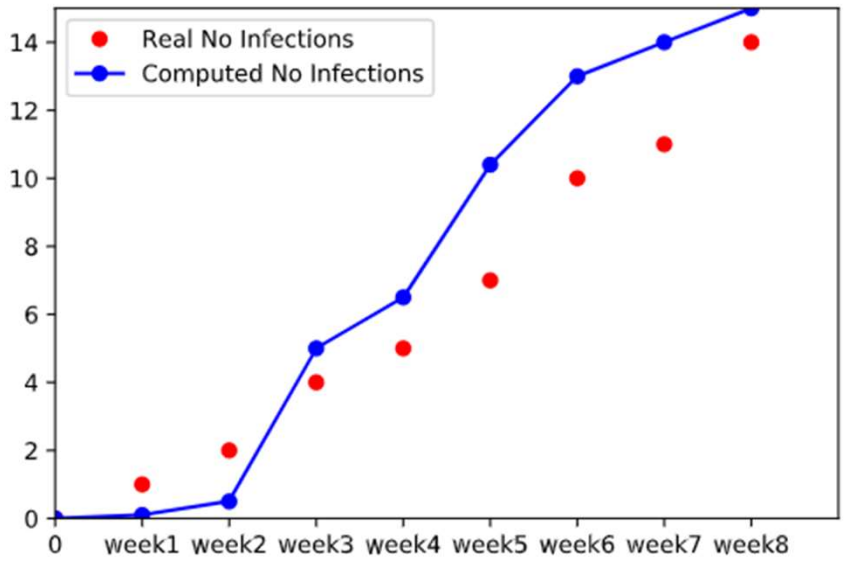
Învăţare supervizată

- Calitatea învăţării → Măsuri de performanţă → Măsuri statistice
  - Eroarea de predicție
    - Suma diferențelor absolute între valorile reale și cele calculate

$$Err = \frac{1}{noSamples} \sum_{i=1}^{noSamples} abs(real_i - computed_i)$$

    - Suma pătratelor diferențelor între valorile reale și cele calculate

$$Err = \sqrt{\frac{1}{noSamples} \sum_{i=1}^{noSamples} (real_i - computed_i)^2}$$



Sisteme inteligente care învaţă singure – SIS

Învăţare supervizată

- Calitatea învăţării → Măsuri de performanţă → Măsuri statistice
  - Eroarea de predicție
    - Acurateţea = Nr de exemple corect clasificate / nr total de exemple

Problemă

Se cunosc rezultatele analizei mai multor indicatori biochimici unui set de persoane in privinta infecției cu un virus; astfel, unele persoane au fost etichetate drept *infectate*, iar altele *sănătoase*.

Pe baza acestor date se antrenează un algoritm de ML cu scopul identificării automate a prezenței virusului. Antrenarea determină obținerea unei acurateți de 98%.

Este performant un astfel de algoritm?

Sisteme inteligente care învaţă singure – SIS

Învăţare supervizată

- Calitatea învăţării → Măsuri de performanţă → Măsuri statistice
  - Eroarea de predicție
    - Acurateţea = Nr de exemple corect clasificate / nr total de exemple
    - Precizia (P) = nr. de exemple pozitive corect clasificate / nr. total de exemple clasificate ca pozitive = *TP / (TP + FP)*
    - Rapelul (R) = nr. de exemple pozitive corect clasificate / nr. total de exemple pozitive = *TP/ (TP +FN)*

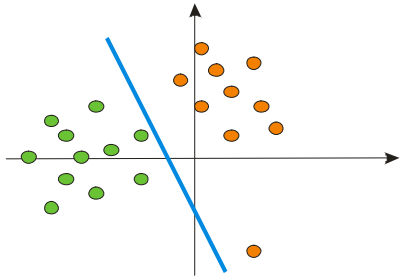
		Rezultate reale	
		Clasa pozitivă	Clasa(ele) negativă(e)
Rezultate calculate	Clasa pozitivă	*True positiv (TP)*	*False positiv (FP)*
	Clasa(ele) negativă(e)	*False negative (FN)*	*True negative (TN)*

Sisteme inteligente care învaţă singure – SIS

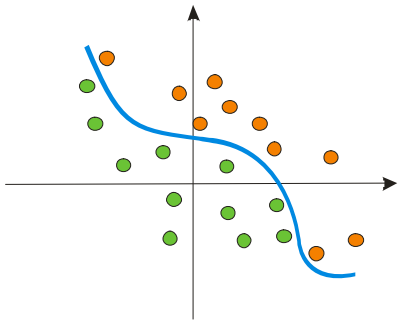
- După tipul de date de ieşire
  - Real → probleme de regresie
  - Etichete → probleme de clasificare (regresie logistică)
    - Clasificare binară
      - Ieşiri (output-uri) binare → nr binar de etichete posibile (*k = 2)*
        - Ex. diagnostic de cancer malign sau benign
        - Ex. email acceptat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

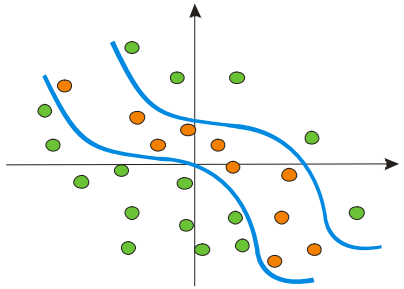
Sisteme inteligente care învaţă singure – SIS

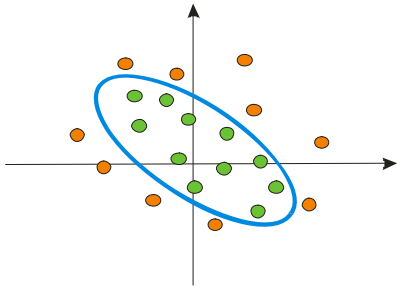
- După forma clasificatorului
  - Clasificare liniară



  - Clasificare ne-liniară
    - se crează o reţea de clasificatori liniari
    - se mapează datele într-un spaţiu nou (mai mare) unde ele devin separabile

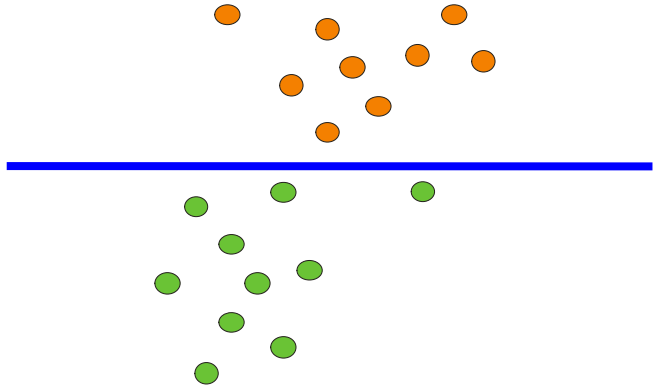




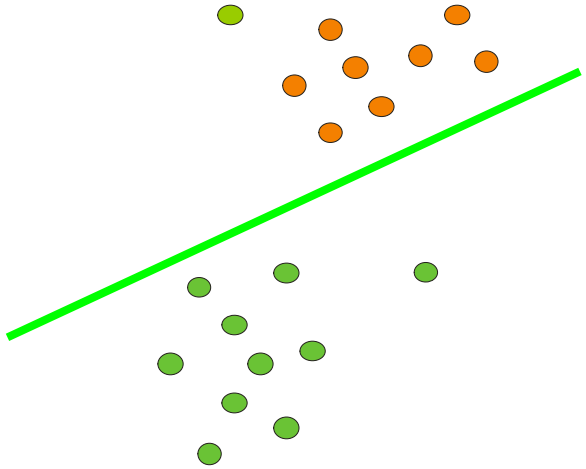


Sisteme inteligente care învaţă singure – SIS

- 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)



Sisteme inteligente care învaţă singure – SIS

- După algoritm
  - Bazate doar pe instanţe
    - Foloseşte 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, reţele neuronale artificiale, maşini cu suport vectorial, algoritmi evolutivi
  - Generative
    - Construieşte un model probabilistic
    - Ex. reţele Bayesiene

Sisteme inteligente care învaţă singure – SIS

Algoritmi

- Cel mai apropiat vecin
- Arbori de decizie
- Maşini cu suport vectorial
- Reţele neuronale artificiale
- Algoritmi evolutivi

Învățare supervizată – algoritmi

Problemă de clasificare

- Se dă un set de date (exemple, instanţe, cazuri)
  - date de antrenament – sub forma unor perechi (atribute_data$_i$, ieşire$_i$), unde
    - i =1,N (N = nr datelor de antrenament)
    - atribute_data$_i$= (atr$_{i1}$, atr$_{i2}$, ..., atr$_{im}$), m – nr atributelor (caracteristicilor, proprietăţilor) unei date
    - ieşire$_i$ = o categorie dintr-o mulţime dată (predefinită) cu *k* elemente (*k* – nr de clase)
  - date de test – sub forma (atribute_data$_i$), i =1,n (n = nr datelor de test)
- Să se determine
  - o funcţie (necunoscută) care realizează corespondenţa atribute – ieşire pe datele de antrenament
  - ieşirea (clasa) asociată unei date (noi) de test folosind funcţia învăţată pe datele de antrenament

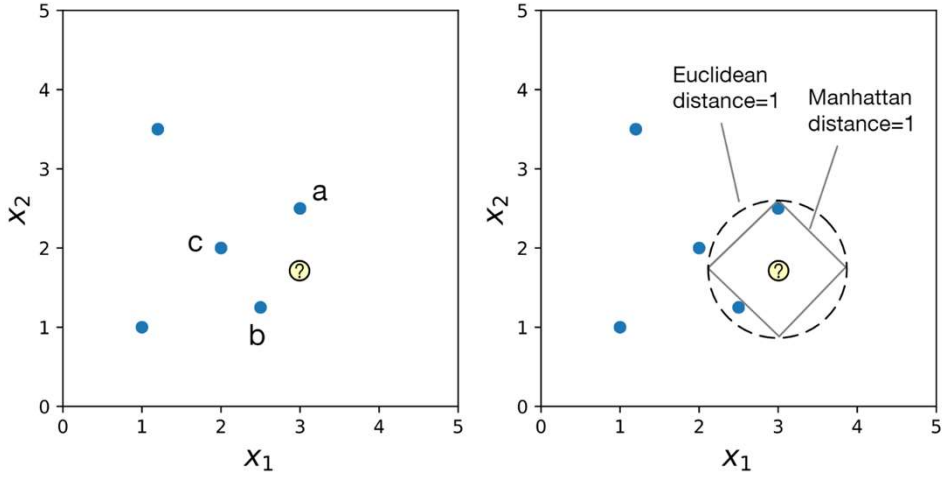
Problemă de regresie

- Se dă un set de date (exemple, instanţe, cazuri)
  - date de antrenament – sub forma unor perechi (atribute_data$_i$, ieşire$_i$), unde
    - i =1,N (N = nr datelor de antrenament)
    - atribute_data$_i$= (atr$_{i1}$, atr$_{i2}$, ..., atr$_{im}$), m – nr atributelor (caracteristicilor, proprietăţilor) unei date
    - ieşire$_i$ = un număr real
  - date de test – sub forma (atribute_data$_i$), i =1,n (n = nr datelor de test)
- Să se determine
  - o funcţie (necunoscută) care realizează corespondenţa atribute – ieşire pe datele de antrenament
  - Ieşirea (clasa/valoarea) asociată unei date (noi) de test folosind funcţia învăţată pe datele de antrenament

Învățare supervizată – algoritmi

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

- Unul dintre cei mai simpli algoritmi de clasificare
- În etapa de antrenament, algoritmul doar citeşte datele de intrare (atributele şi clasa fiecărei instanţe)
- În etapa de testare, pentru o nouă instanţă (fără clasă):
  - se caută (printre instanţele de antrenament) **cei mai apropiaţi *k* vecini**
    - distanţa Minkowski (Manhattan, Euclidiană) – atribute continue
    - distanţa Hamming, Levensthein – analiza textelor
    - alte distanţe (funcţii kernel)

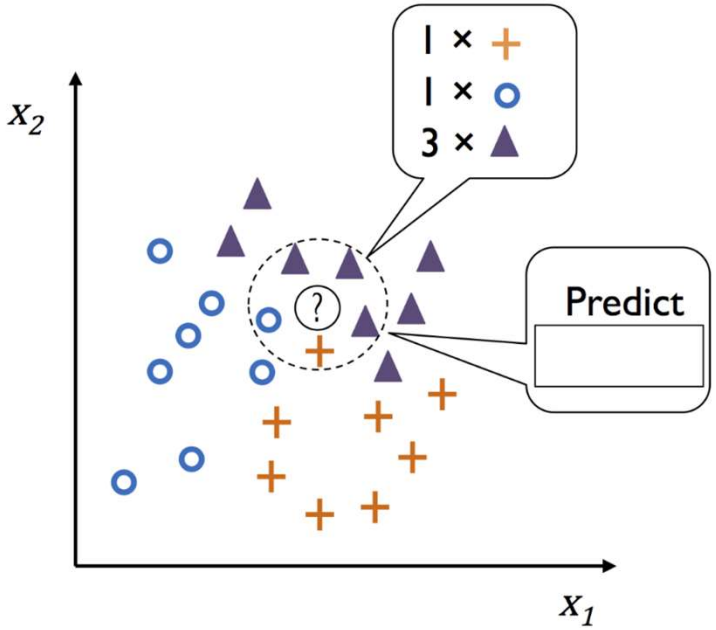


  - se preia clasa majoritară a acestor *k* vecini

Învățare supervizată – algoritmi

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

- Unul dintre cei mai simpli algoritmi de clasificare
- În etapa de antrenament, algoritmul doar citeşte datele de intrare (atributele şi clasa fiecărei instanţe)
- În etapa de testare, pentru o nouă instanţă (fără clasă)
  - se caută (printre instanţele de antrenament) cei mai apropiaţi *k* vecini şi
  - **se preia clasa majoritară a acestor *k* vecini**



Majority vote:

Plurality vote:

Majority vote: None

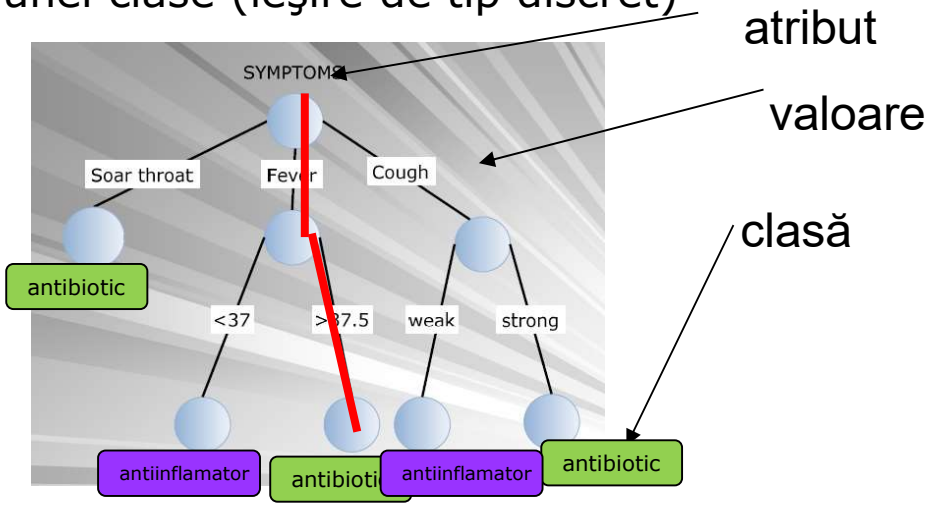
Plurality vote:

Învățare supervizată – algoritmi
kNN

- Tool-uri
  - Sklearn (python)
    - 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

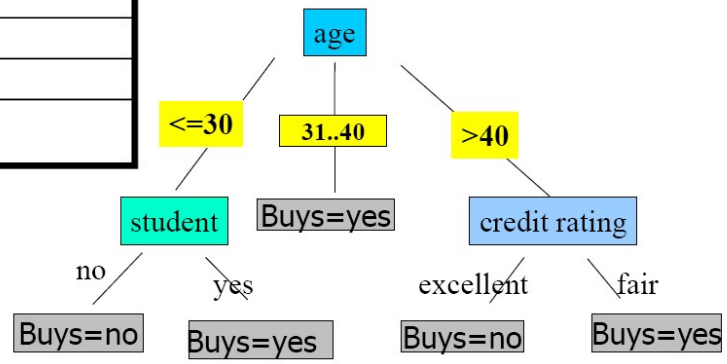
- Scop
  - Divizarea unei colecţii de articole în seturi mai mici prin aplicarea succesivă a unor reguli de decizie → adresarea mai multor întrebări
    - Fiecare întrebare este formulată în funcţie de răspunsul primit la întrebarea precedentă
  - Elementele se caracterizează prin informaţii non-metrice
- Definire
  - Fiecare nod intern corespunde unui atribut
  - Fiecare ramură de sub un nod (atribut) corespunde unei valori a atributului
  - Fiecare frunză corespunde unei clase (ieşire de tip discret)



Învățare supervizată – algoritmi
Arbori de decizie

- Exemplu

rec	Age	Income	Student	Credit_rating	Buys_computer(CLASS)
r1	<=30	High	No	Fair	No
r2	<=30	High	No	Excellent	No
r3	31…40	High	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Low	Yes	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31…40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes
r10	>40	Medium	Yes	Fair	Yes
r11	<=30	Medium	Yes	Excellent	Yes
r12	31…40	Medium	No	Excellent	Yes
r13	31…40	High	Yes	Fair	Yes
r14	>40	Medium	No	Excellent	No



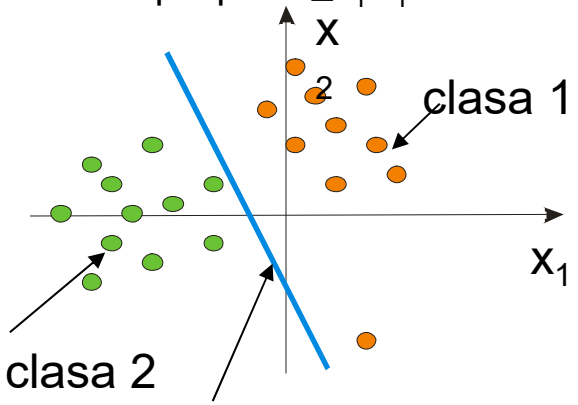
Învățare supervizată – algoritmi
Arbori de decizie

- 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

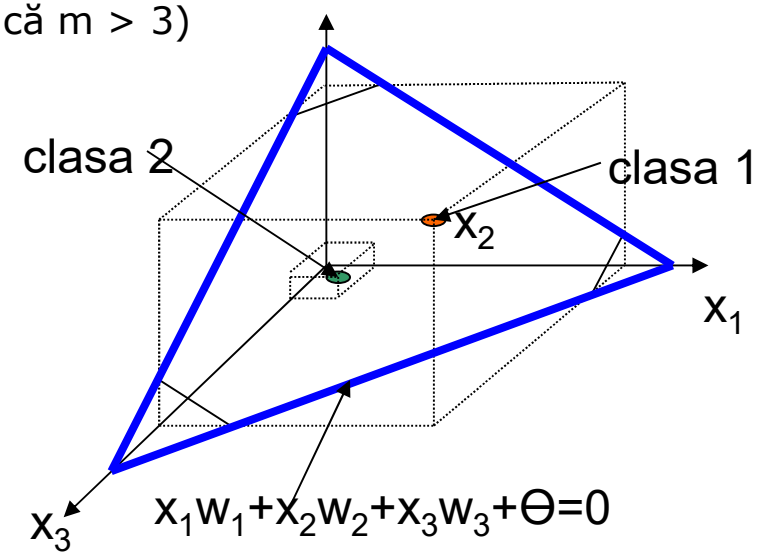
Învățare supervizată – algoritmi

- Clasificare binară pt orice fel de date de intrare (discrete / continue)
  - Datele pot fi separate de:
    - o dreaptă → ax + by + c = 0 (dacă m = 2)
    - un plan → ax + by + cz + d = 0 (dacă m = 3)
    - un hiperplan $\sum a_i x_i + b = 0$ (dacă m > 3)



$x_1w_1+x_2w_2+\Theta=0$

Clasificare binară cu m=2 intrări



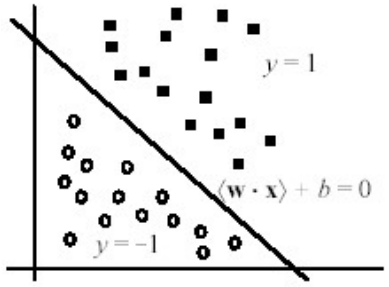
$x_1w_1+x_2w_2+x_3w_3+\Theta=0$

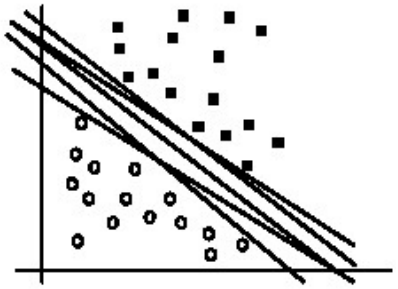
Clasificare binară cu m=3 intrări

  - Cum găsim modelul de separare (valorile optime pt. a, b, c, d, a_i şi forma modelului)?
    - Maşini cu suport vectorial
    - Reţele neuronale artificiale
    - Algoritmi evolutivi

Învățare supervizată – algoritmi
Maşini cu suport vectorial (MSV)

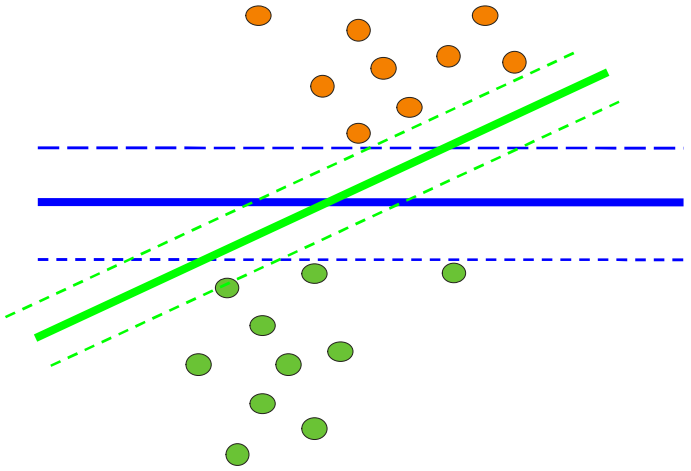
- Dezvoltate de Vapnik în 1970 și popularizate după 1992
  - Clasificatori liniari care identifică un hiperplan de separare între clasa pozitivă şi cea negativă; au o fundamentare teoretică foarte riguroasă
  - Funcţionează foarte bine pentru date de volum mare (ex. analiza textelor, analiza imaginilor)
- 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 $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$, unde
        - $\mathbf{x}_i = (x_1, x_2, \ldots, x_m)$ este un vector de intrare într-un spaţiu real $X \subseteq R^m$ şi
        - y_i este eticheta clasei (valoarea de ieşire), $y_i \in \{1, -1\}$
      - **w** – vector de ponderi / coeficienți de importanță
  - Pot exista mai multe hiperplanuri
    - Care este cel mai bun?
    - MSV caută hiperplanul cu cea mai largă margine (cel care micşorează eroarea de generalizare)
      - Algoritmul SMO (Sequential minimal optimization)



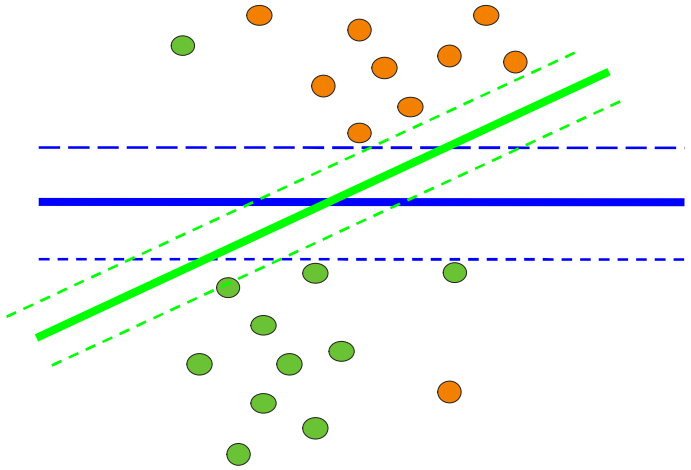


Învățare supervizată – algoritmi
Maşini cu suport vectorial

- Cazuri de date
  - Liniar separabile
    - Separabile
      - Eroarea = 0



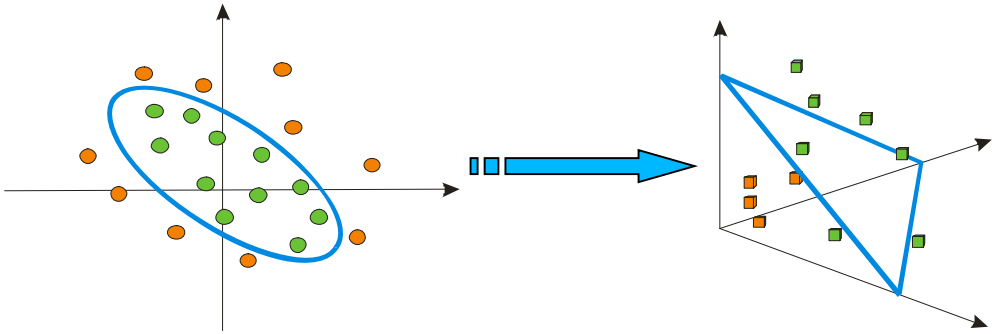
    - Ne-separabile
      - Se relaxează constrângerile → se permit unele erori
      - C – coeficient de penalizare



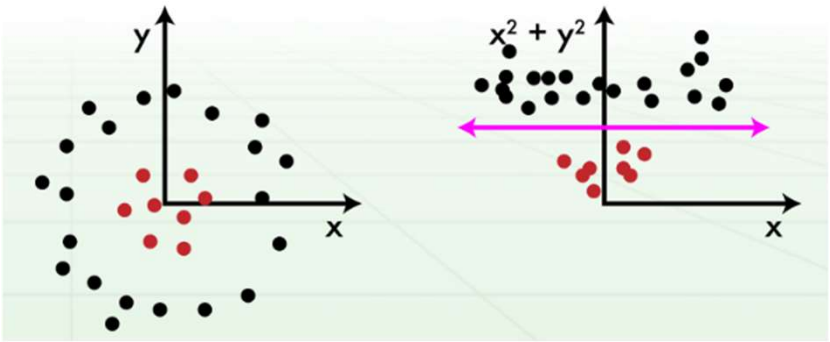
Învățare supervizată – algoritmi
Maşini cu suport vectorial

Cazuri de date

- Non-liniar separabile
  - Spaţiul de intrare se transformă într-un spaţiu cu mai multe dimensiuni (*feature space*), cu ajutorul unei funcţii kernel, unde datele devin liniar separabile



  - Kernele posibile
    - Clasice
      - Polynomial kernel: $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle + 1)^d$
      - RBF kernel: $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = exp(- \sigma |\boldsymbol{x}_1 - \boldsymbol{x}_2|^2)$
    - Kernele multiple
      - Liniare: $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \Sigma\, w_i\, K_i$
      - Ne-liniare
        - Fără coeficienţi: $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = K_1 + K_2 * exp(K_3)$
        - Cu coeficienţi: $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = K_1 + c_1 * K_2 * exp(c_2 + K_3)$



Învățare supervizată – algoritmi
Maşini cu suport vectorial

- Probleme
  - Doar atribute reale
  - Doar clasificare binară
  - Background matematic dificil

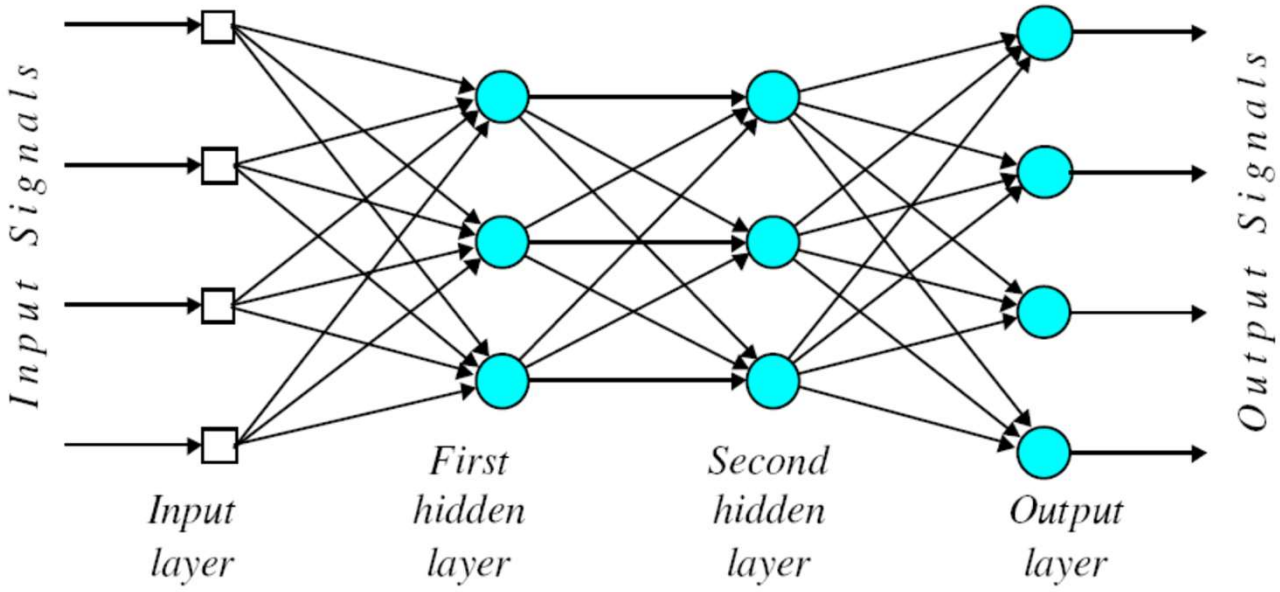
- Tool-uri
  - LibSVM → http://www.csie.ntu.edu.tw/~cjlin/libsvm/
  - Weka → SMO
  - SVMLight → http://svmlight.joachims.org/
  - SVMTorch → http://www.torch.ch/

- Biblio
  - http://www.support-vector-machines.org/

- Reviste științifice
  - https://www.jmlr.org/

Învățare supervizată – algoritmi
Rețele neuronale artificiale

- Similar unei reţele neuronale biologice
- O mulţime de neuroni dispuşi ca într-un graf (un nod → un neuron) pe mai multe straturi (*layere*)
  - Strat de intrare
    - Conţine *m* (nr de atribute al unei date) noduri
  - Strat de ieşire
    - Conţine *r* (nr de ieşiri) noduri
  - Straturi intermediare (ascunse) – rol în "complicarea" reţelei
    - 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/ and PyTorch https://pytorch.org/
  - scientific computing framework
  - Multi-dim array
  - NN
  - GPU
- Caffe
  - deep learning framework
  - Berkley

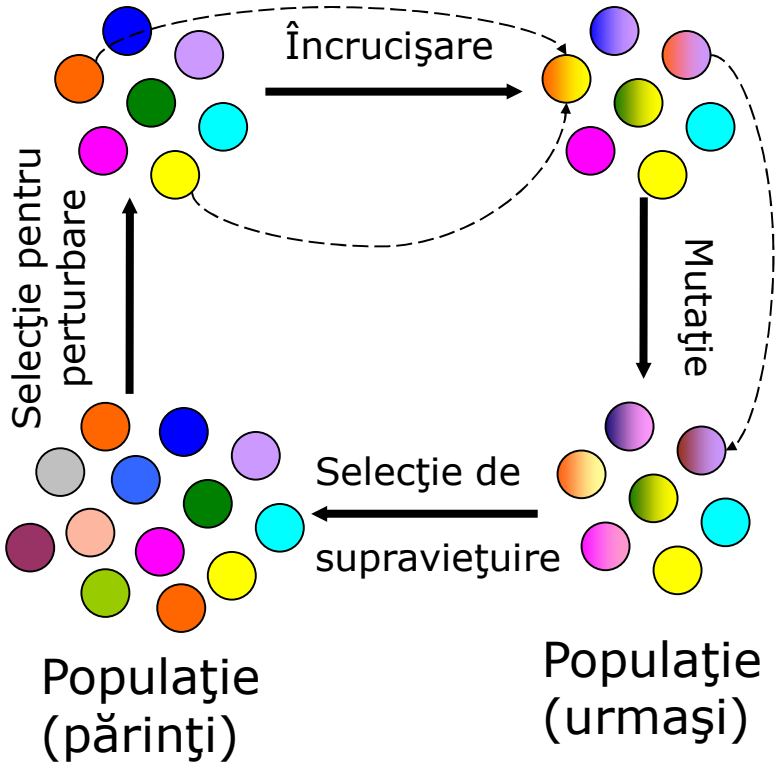
Învățare supervizată – algoritmi
Algoritmi evolutivi

- 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

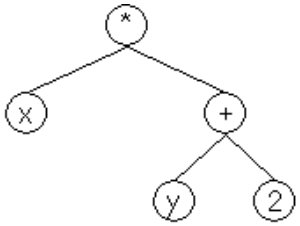
Învățare supervizată – algoritmi
Algoritmi evolutivi

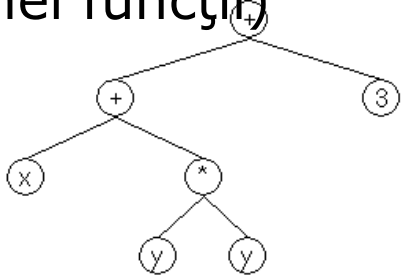
Initializare populaţie P(0)
Evaluare P(0)
g := 0; //generaţia
CâtTimp (not condiţie_stop) execută
    Repetă
      Selectează 2 părinţi p1 şi p2 din P(g)
      Încrucişare(p1,p2) =>o1 şi o2
      Mutaţie(o1) => o1*
      Mutaţie(o2) => o2*
      Evaluare(o1*)
      Evaluare(o2*)
      adăugare o1* şi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



- Definire
  - Propusă de john Koza în 1988
  - http://www.genetic-programming.org/
  - Un tip particular de algoritmi evolutivi
  - Cromozomi
    - sub formă de arbore care codează mici programe
  - Fitness-ul unui cromozom
    - Performanţa programului codat în el
  - Scopul PG
    - Evoluarea de programe de calculator
    - AG evoluează doar soluţii pentru probleme particulare

- Proiectare
  - Reprezentarea cromozomilor
    - Foarte importantă, dar este o sarcină dificilă
    - Cromozomul = un arbore cu noduri de tip
      - Funcţie → operatori matematici (+,-,*,/,*sin*,*log*, *if*,...)
      - Terminal → atribute ale datelor problemei sau constante (*x,y,z,a,b,c,...*)
    - care codează expresia matematică a unui program (problema regresiei → a unei funcţii)





$x(y+2)$ $x+y^2+3$

- Proiectare
  - Fitness
    - Eroarea de predicţie – diferenţa între ceea ce dorim să obţinem şi ceea ce obţinem de fapt
    - pp o problemă de regresie cu următoarele date de intrare (2 atribute şi o ieşire) şi 2 cromozomi:
      - $c_1 = 3x_1 - x_2 + 5$
      - $c_2 = 3x_1 + 2x_2 + 2$

$f^*(x1,x2) = 3x_1 + 2x_2 + 1$ – necunoscută

x_1	x_2	$f^*(x_1,x_2)$	$f_1(x_1,x_2)$	$f_2(x_1,x_2)$	$\lvert f^* - f_1 \rvert$	$\lvert f^* - f_2 \rvert$
1	1	6	7	7	1	1
0	1	3	4	4	1	1
1	0	4	8	5	4	1
-1	1	0	1	1	1	1
					$\Sigma=7$	$\Sigma= 4$

➔ c_2 e mai bun ca c_1

- Proiectare
  - Fitness
    - Eroarea de predicţie – diferenţa între ceea ce dorim să obţinem şi ceea ce obţinem de fapt
    - pp o problemă de clasificare cu următoarele date de intrare (2 atribute şi o ieşire) şi 2 cromozomi:
      - $c_1 = 3x_1-x_2+5$
      - $c_2 = 3x_1+2x_2+2$

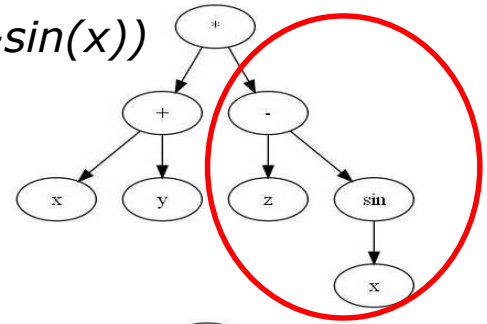
x_1	x_2	$f^*(x_1,x_2)$	$f_1(x_1,x_2)$	$f_2(x_1,x_2)$	$\|f^*-f_1\|$	$\|f^*-f_2\|$
1	1	Yes	Yes	Yes	0	0
0	1	No	Yes	No	1	0
1	0	Yes	No	No	1	1
-1	1	Yes	No	yes	1	0
					$\Sigma=3$	$\Sigma= 1$

➔ c_2 e mai bun ca c_1

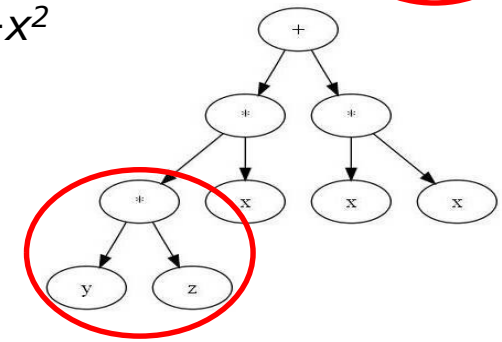
- Proiectare
  - Iniţializarea cromozomilor
    - Generare aleatoare de arbori corecţi → programe valide (expresii matematice valide)
    - Se stabileşte o adâncime maximă a arborilor D_{max}
  - Operatori genetici → Selecţia pentru recombinare
    - similar oricărui algoritm evolutiv
    - recomandare → selecţie proporţională

- Proiectare
  - Operatori genetici → Încrucişare
    - Cu punct de tăietură – se interchimbă doi sub-arbori
    - Punctul de tăietură se generează aleator
    - Notă: Dimensiunea descendenţilor diferă de dimensiune părinţilor

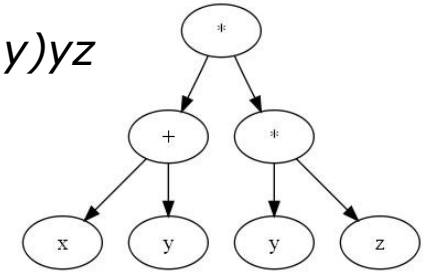
$p_1=(x+y)*(z-sin(x))$



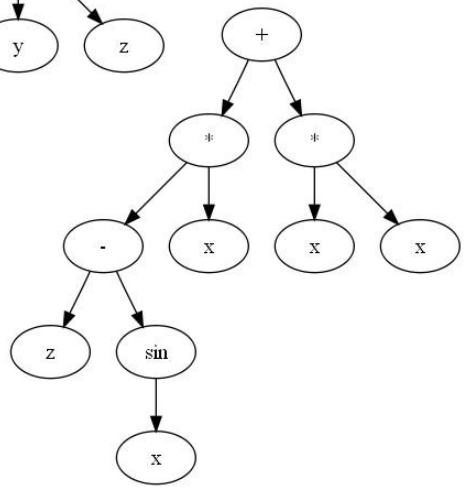
$p_2=xyz+x^2$



$f_1=(x+y)yz$

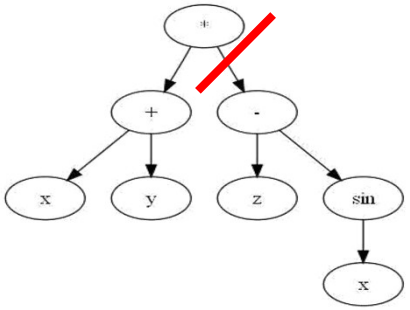


$f_2=(z-sin(x))x+x^2$

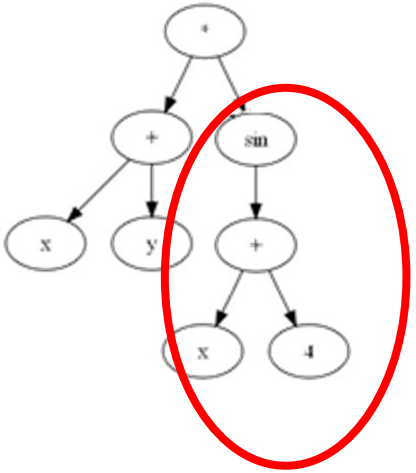


- Proiectare
  - Operatori genetici → Mutaţie
    - Mutaţie de tip *Koza* → Înlocuirea unui nod (intern sau frunză) cu un nou sub-arbore

p=(x+y)(z-sin(x))*



*f=(x+y)*sin(x+4)*



- Comparaţie AG şi PG
  - Forma cromozomilor
    - AG – cromozomi liniari
    - PG – cromozomi ne-liniari
  - Dimensiunea cromozomilor
    - AG – fixă
    - PG – variabilă (în adâncime sau lăţime)
  - Schema de creare a descendenţilor
    - AG – încrucişare şi mutaţie
    - PG – încrucişare sau mutaţie

- Avantaje
  - PG găseşte soluţii problemelor care nu au o soluţie optimă
    - Un program pentru conducerea maşinii → nu există o singură soluţie
    - Unele soluţii implică un condus sigur, dar lent
    - Alte soluţii implică o viteză mare, dar un risc ridicat de accidente
    - Coducerea maşinii ←→ compromis între viteză mare şi siguranţă
  - PG este utilă în problemele a căror variabile se modifică frecvent
    - Conducerea maşinii pe autostradă
    - Conducerea maşinii pe un drum forestier

- Limite
  - Timpul mare necesar evoluţiei pentru identificarea soluţiei

Învățare supervizată – algoritmi
Algoritmi evolutivi – programare genetică

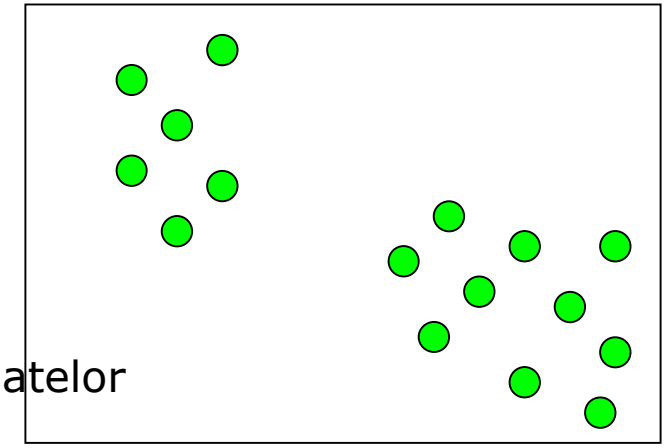
- Tool-uri
  - https://github.com/JesseBuesking/TinyGP-Java
  - https://github.com/lfarinha/TinyGP
  - http://geneticprogramming.com/software/
- Referințe
  - http://geneticprogramming.com/
  - http://www.genetic-programming.com/GPEM2010article.pdf
- Reviste științifice
  - https://www.springer.com/journal/10710

Învățare automată

- Învăţare supervizată
- **Învăţare ne-supervizată**
- Învăţare cu întărire
- Teoria învăţării

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 - clustering – 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

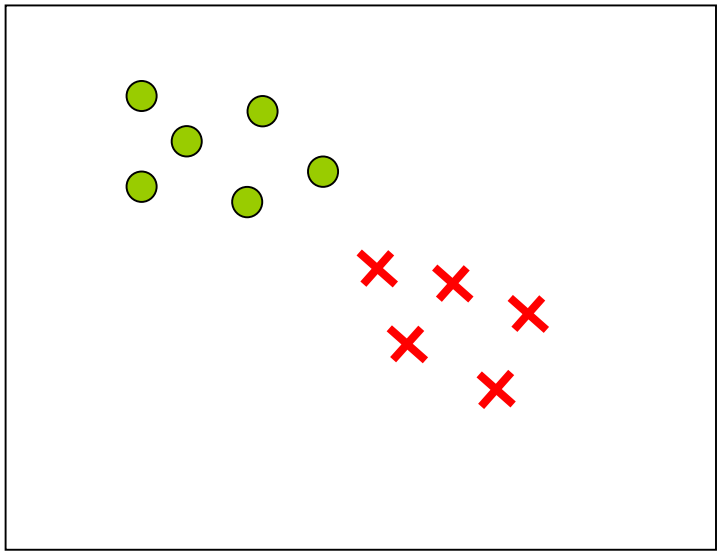
- Se dă
  - un set de date (exemple, instanţe, cazuri)
    - Date de antrenament
      - Sub forma ***atribute_data***$_i$, unde
        - $i = 1,N$ (N = nr datelor de antrenament)
        - ***atribute_data***$_i$= (atr_{i1}, atr_{i2}, ..., atr_{im}), m – nr atributelor (caracteristicilor, proprietăţilor) unei date
    - Date de test
      - Sub forma (***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

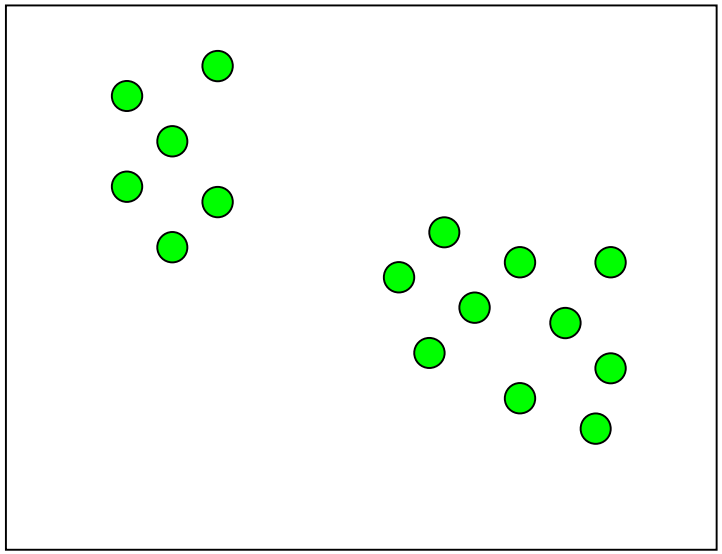
Alte denumiri

- Clustering

Învățare ne-supervizată - clustering – definire

- Supervizată *vs.* Ne-supervizată





Învățare ne-supervizată - clustering – 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 - clustering – definire

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

Învățare ne-supervizată - clustering – definire

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 - clustering – definire

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 - clustering – definire

- 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 - clustering – definire

- 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 - clustering – definire

- 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 - clustering – definire

- 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 - clustering – definire

- 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 - clustering – definire

- 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 - clustering – definire

- 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 - clustering – algoritmi

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

Învățare ne-supervizată

- Clustering
- Reducerea dimensiunii datelor
  - Liniară
  - Ne-liniară – manifold learning
- Detecția 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

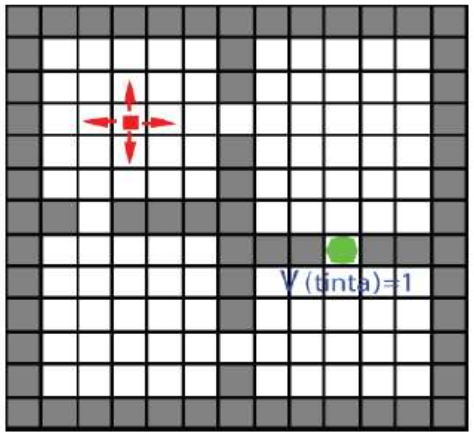
- Învăţare supervizată
- Învăţare ne-supervizată
- **Învăţare cu întărire**
- Teoria învăţării

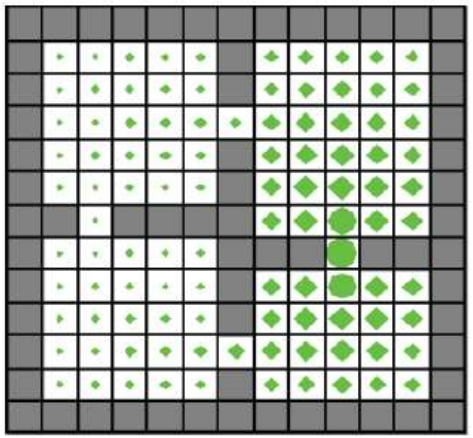
Învățare cu întărire

- Scop
  - Învăţarea, de-a lungul unei perioade, a unui mod de acţiune (comportament) care să maximizeze recompensele (câştigurile) pe termen lung
  - "make good sequences of decisions"
- Tip de probleme
  - Ex. Dresarea unui câine (good and bad dog)
- Caracteristic
  - Interacţiunea cu mediul (acţiuni → recompense)
  - Secvenţă de decizii

Învățare cu întărire – definire

- Exemplu: plecând din căsuţa roşie să se găsească un drum până la căsuţa verde





- Agentu[illegible] observarea rezultatelor obţinute din aceste interacţiuni
  - Este vorba de "cauză şi efect" -- modul în care oamenii îşi formează cunoaşterea aupra mediului pe parcursul vieţii
  - Acţiunile pot afecta şi recompensele ulterioare, nu numai pe cele imediate (efect întârziat)

Învățare cu întărire – definire

Învăţarea unui anumit comportament în vederea realizării unei sarcini ➔ execuţia unei acţiuni ➔ primeşte un feedback (cât de bine a acţionat pentru îndeplinirea sarcinii) ➔ execuţia unei noi acţiuni

Învăţare cu întărire

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

Definire

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

Alte denumiri

- Reinforcement learning
- Învăţare împrospătată

Învățare cu întărire – definire

	Planificare	Învățare supervizată	Învățare nesupervizată	Învățare cu întărire
Optimizare	X			X
Învățare din experiență (adnotată sau nu)		X	X	X
Generalizare	X	X	X	X
Consecințe ulterioare/întârziate	X			X
Explorare				X

Învățare cu întărire – exemple

- Robotică
  - Controlul membrelor
  - Controlul posturii
  - Preluarea mingii în fotbalul cu roboţii
- Cercetări operaţionale
  - Stabilirea preţurilor
  - 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


Questions
Answers