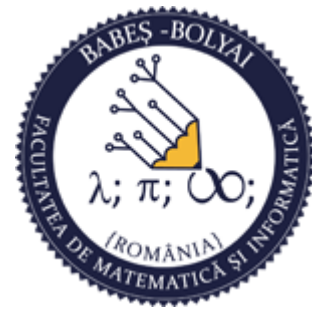




UNIVERSITATEA BABEŞ-BOLYAI
Facultatea de Matematică și Informatică



INTELIGENȚĂ , ARTIFICIALĂ

Sisteme inteligente

Sisteme care învață singure

– arbori de decizie –

Laura Dioșan

Sumar

A. Scurtă introducere în Inteligența Artificială (IA)

B. Rezolvarea problemelor prin căutare

- Definirea problemelor de căutare
- Strategii de căutare
 - Strategii de căutare neinformate
 - Strategii de căutare informate
 - Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
 - Strategii de căutare adversarială

C. Sisteme inteligente

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

Materiale de citit și legături utile

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

Conținut

□ Sisteme inteligente

■ Sisteme care învață singure (SIS)

□ Sisteme

- Logistic regression
- kNN
- Arbori de decizie
- SVM

Sisteme inteligente – SIS

□ Logistic regression

- *Remember gradient descent*
- *= gradient descent + sigmoid activation*
- *New update rules*

Sisteme inteligente – SIS

□ kNN (cei mai apropiași k vecini)

- 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 apropiași k vecini și se preia clasa majoritară a acestor k vecini
- Căutarea vecinilor se bazează pe:
 - distanța Minkowski (Manhattan, Euclidiană) – atribute continue
 - distanța Hamming, Levensthein – analiza textelor
 - alte distanțe (funcții kernel)

Sisteme inteligente – SIS – Arbori de decizie

□ Arbori de decizie

- Scop
- Definiere
- Tipuri de probleme rezolvabile
- Exemplu
- Proces
- Tool-uri
- Avantaje și limite

Sisteme inteligente – SIS – 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-metriche

□ 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

Sisteme inteligente – SIS – Arbori de decizie

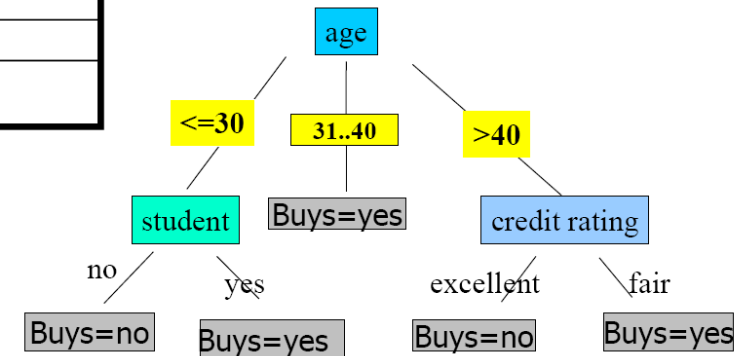
□ Tipuri de probleme

- Exemplele (instanțele) sunt reprezentate printr-un număr fix de atribute, fiecare atribut putând avea un număr limitat de valori
- Funcția obiectiv ia valori de tip discret
- AD reprezintă o disjuncție de mai multe conjuncții, fiecare conjuncție fiind de forma atributul a_i are valoarea v_j
- Datele de antrenament pot conține erori
- Datele de antrenament pot fi incomplete
 - Anumitor exemple le pot lipsi valorile pentru unele atribute
- Probleme de clasificare
 - Binară
 - exemple date sub forma [(atribut $_{ij}$, valoare $_{ij}$), clasă $_i$, $i=1,2,\dots,n$, $j=1,2,\dots,m$, clasă $_i$ putând lua doar 2 valori]
 - Multi-clasă
 - exemple date sub forma [(atribut $_{ij}$, valoare $_{ij}$), clasă $_i$, $i=1,2,\dots,n$, $j=1,2,\dots,m$, clasă $_i$ putând lua k valori]
- Probleme de regresie
 - AD se construiesc similar cazului problemei de clasificare, dar în locul etichetării fiecărui nod cu eticheta unei clase se asociază nodului o valoare reală sau o funcție dependentă de intrările nodului respectiv
 - Spațiul de intrare se împarte în regiuni de decizie prin tăieturi paralele cu axele Ox și Oy
 - Are loc o transformare a ieșirilor discrete în funcții continue
 - Calitatea rezolvării problemei
 - Eroare (pătratică sau absolută) de predicție

Sisteme inteligente – SIS – Arbori de decizie

□ Exemplu

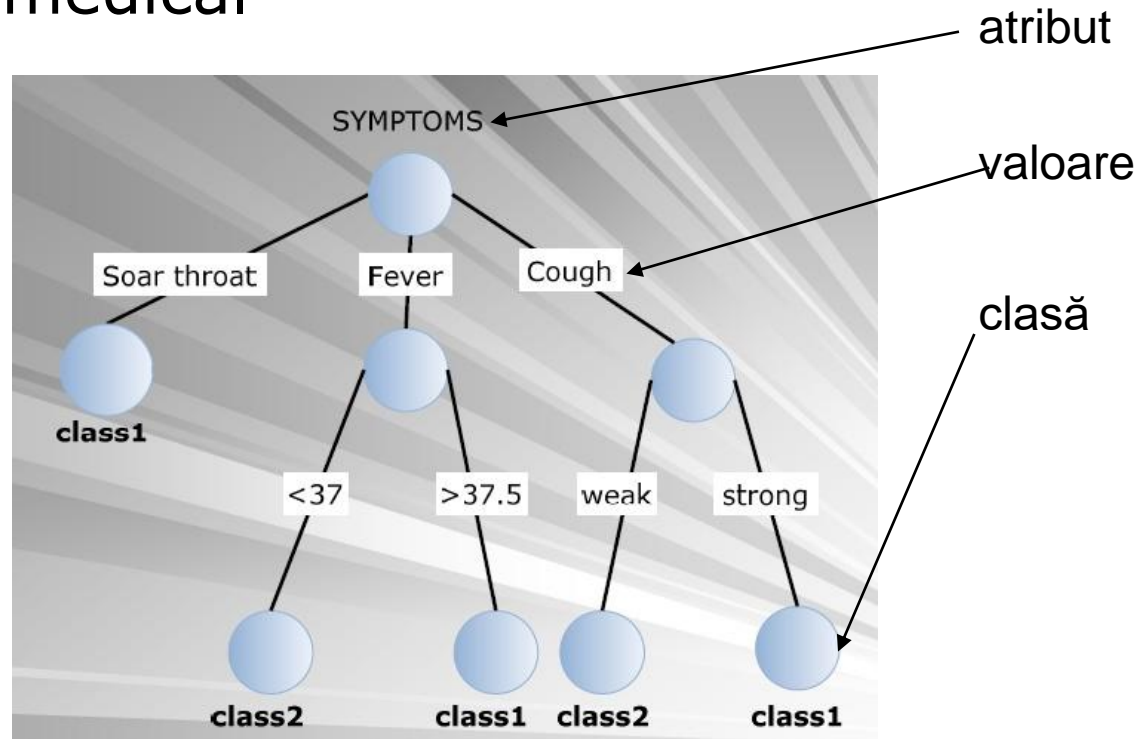
rec	Age	Income	Student	Credit_rating	Buys_computer(CLASS)
r1	<=30	High	No	Fair	No
r2	<=30	High	No	Excellent	No
r3	31...40	High	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Low	Yes	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31...40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes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



Sisteme inteligente – SIS – Arbori de decizie

□ Exemplu

■ Sistem medical



Sisteme inteligente – SIS – Arbori de decizie

□ Exemplu

■ Acordarea de credite

aprobat sau nu

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

Sisteme inteligente – SIS – Arbori de decizie

□ Proces

- Construirea (creșterea, inducția) arborelui
 - Se bazează pe un set de date de antrenament
 - Lucrează de jos în sus sau de sus în jos (prin divizare – *splitting*)
- Utilizarea arborelui ca model de rezolvare a problemelor
 - Ansamblul deciziilor efectuate de-a lungul unui drum de la rădăcină la o frunză formează o regulă
 - Regulile formate în AD sunt folosite pentru etichetarea unor noi date
- Tăierea (curățirea) arborelui (pruning)
 - Se identifică și se mută/elimină ramurile care reflectă zgomote sau excepții

Sisteme inteligente – SIS – Arbori de decizie

□ Proces → Construirea AD

■ Divizarea datelor de antrenament în subseturi pe baza caracteristicilor datelor

- Un nod → întrebare legată de o anumită proprietate a unui obiect dat
- Ramurile ce pleacă din nod → etichetate cu posibilele răspunsuri la întrebarea din nodul curent
- La început toate exemplele sunt plasate în rădăcină
 - La pornire, un atribut va fi rădăcina arborelui, iar valorile atributului vor deveni ramuri ale rădăcinii
- Pe următoarele nivele exemplele sunt partiționate în funcție de atribute → ordinea considerării atributelor
 - Pentru fiecare nod se alege în mod recursiv câte un atribut (cu valorile lui pe ramurile descendente din nodul curent)
- Divizarea → greedy în luarea decizilor

■ Proces iterativ

- Reguli de oprire
 - toate exemplele aferente unui nod fac parte din aceeași clasă → nodul devine frunză și este etichetat cu Ci
 - Nu mai sunt exemple → nodul devine frunză și este etichetat cu clasa majoritară în setul de date de antrenament
 - nu mai pot fi considerate noi atribute

Sisteme inteligente – SIS – Arbori de decizie

- Proces → Construirea AD
 - Exemplu

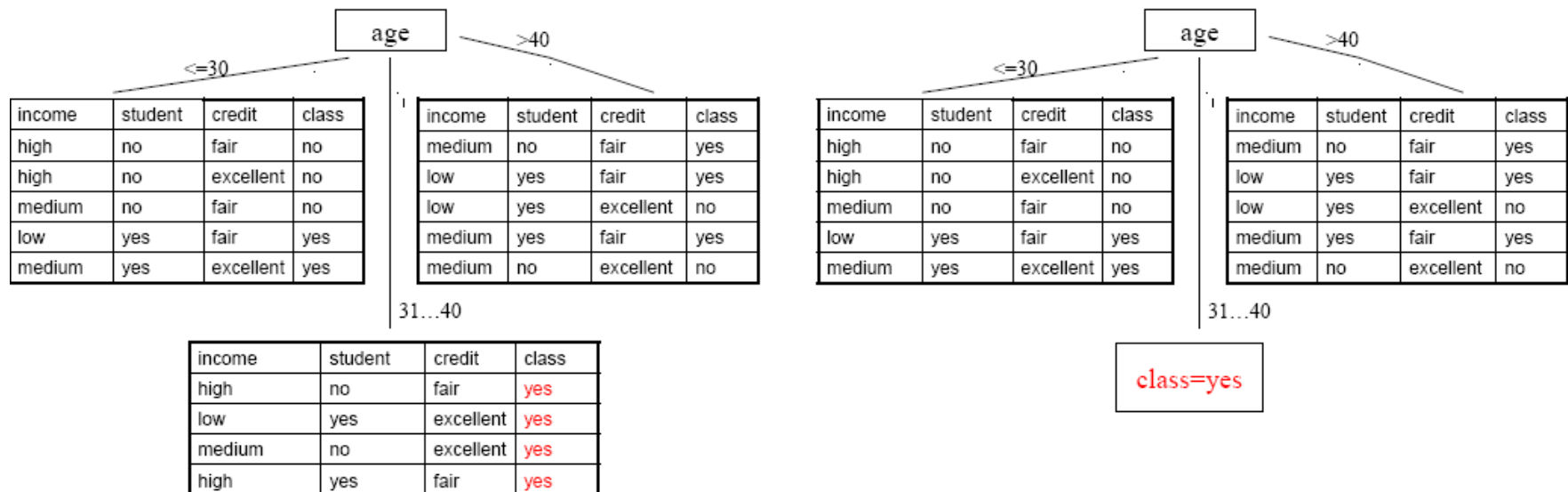
rec	Age	Income	Student	Credit_rating	Buys_computer(CLASS)
r1	<=30	High	No	Fair	No
r2	<=30	High	No	Excellent	No
r3	31...40	High	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Low	Yes	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31...40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes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

Sisteme inteligente – SIS – Arbori de decizie

□ Proces → Construirea AD

■ Exemplu

- Pentru rădăcină se alege atributul *age*

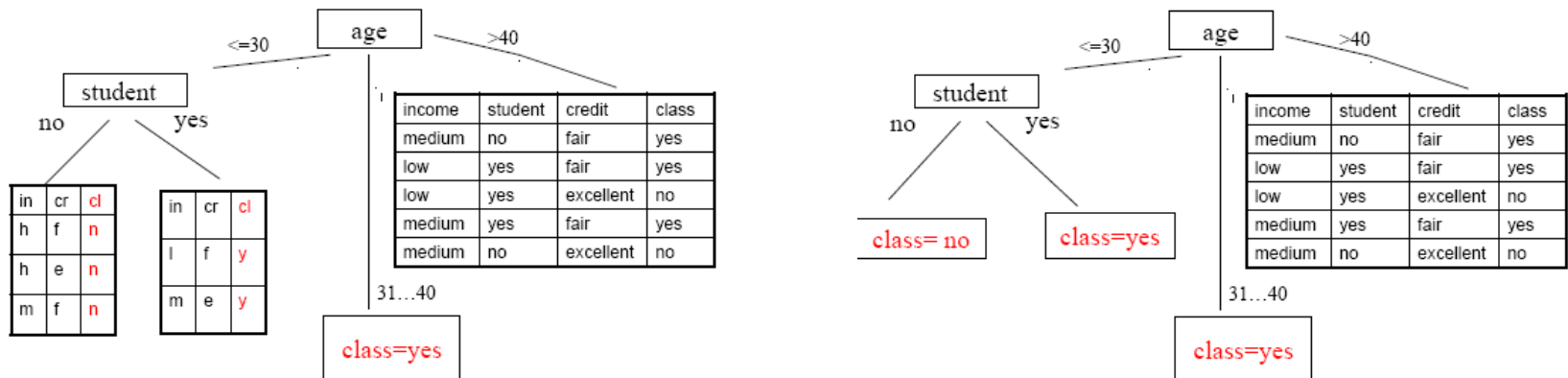


Sisteme inteligente – SIS – Arbori de decizie

□ Proces → Construirea AD

■ Exemplu

- Pentru rădăcină se alege atributul *age*
- Pe ramura ≤ 30 se alege atributul *student*

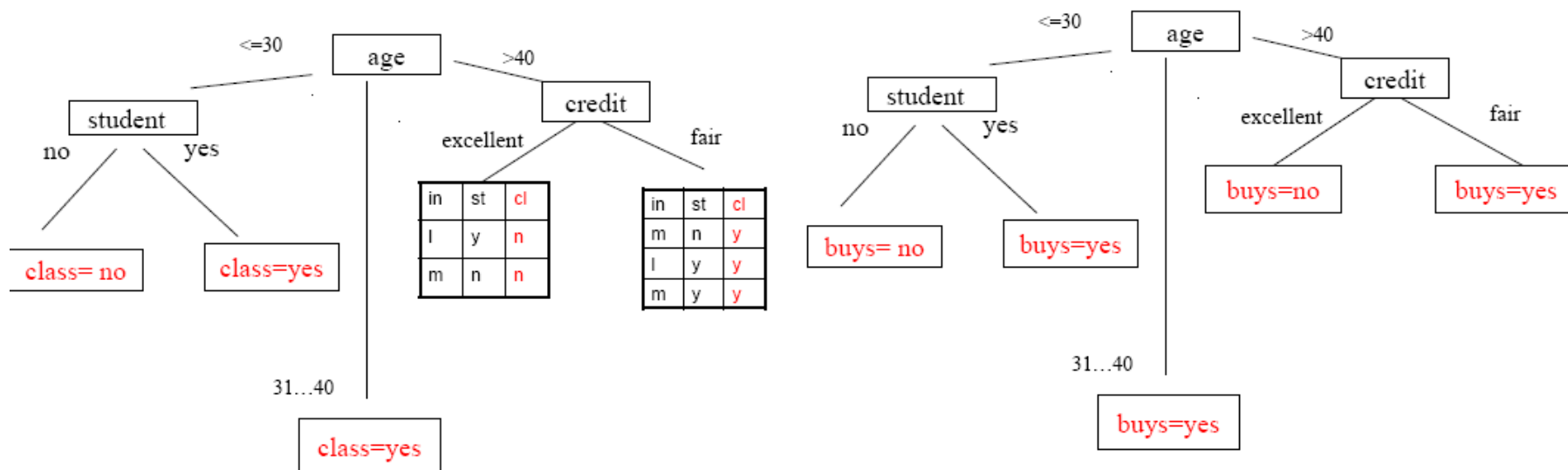


Sisteme inteligente – SIS – Arbori de decizie

□ Proces → Construirea AD

■ Exemplu

- Pentru rădăcină se alege atributul *age*
- Pe ramura ≤ 30 se alege atributul *student*
- Pe ramura > 40 se alege atributul *credit*



Sisteme inteligente – SIS – Arbori de decizie

- Proces → Construirea AD → Algoritmul ID3/C4.5
 - Greedy, recursiv, top-down, divide-and-conquer

```
generare(D, A){ //D – partiționare a exemplilor de antrenament, A – lista de atribute
  Crearea unui nod nou N
  Dacă exemplele din D fac parte dintr-o singură clasă C atunci
    nodul N devine frunză și este etichetat cu C
    returnează nodul N

  Altfel
    Dacă A=∅ atunci
      nodul N devine frunză și este etichetat cu clasa majoritară în D
      returnează nodul N

    Altfel
      atribut_separare = Selectează_atribut(D, A)
      Etichetează nodul N cu atribut_separare
      Pentru fiecare valoare posibilă vj a lui atribut_separare
        Fie Dj mulțimea exemplilor din D pentru care atribut_separare = vj
        Dacă Dj = ∅ atunci
          Atașează nodului N o frunză etichetată cu clasa majoritară în D
        Altfel
          Atașează nodului N un nod returnat de generare(Dj, A –atribut_separare)
      Returnează nodul N
}
```

Sisteme inteligente – SIS – Arbori de decizie

- Proces → Construirea AD → Algoritmul ID3/C4.5
 - Selectează_atribut(D, A) → Alegerea atributului aferent unui nod (rădăcină sau intern)
 - Aleatoare
 - Atributul cu cele mai puține/multe valori
 - Pe baza unei ordini prestabilite a atributelor
 - Câștigul de informație
 - Rata câștigului
 - Indicele Gini
 - Distanța între partițiile create de un atribut

Sisteme inteligente – SIS – Arbori de decizie

- Proces → Construirea AD → Algoritmul ID3/C4.5 → Selectare atribut
 - Câștigul de informație
 - O măsură de impuritate
 - 0 (minimă) dacă toate exemplele fac parte din aceeași clasă
 - 1 (maximă) dacă avem număr egal de exemple din fiecare clasă
 - Se bazează pe entropia datelor
 - măsoară impuritatea datelor
 - numărul sperat (așteptat) de biți necesari pentru a coda clasa unui element oarecare din setul de date
 - clasificare binară (cu 2 clase): $E(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$, unde
 - p_+ - proporția exemplelor pozitive în setul de date S
 - p_- - proporția exemplelor negative în setul de date S
 - clasificare cu mai multe clase: $E(S) = \sum_{i=1, 2, \dots, k} -p_i \log_2 p_i$ - entropia datelor relativ la atributul țintă (atributul de ieșire), unde
 - p_i - proporția exemplelor din clasa i în setul de date S
 - câștigul de informație (*information gain*) al unei caracteristici a (al unui atribut a) datelor
 - Reducerea entropiei setului de date ca urmare a eliminării atributului a
 - $Gain(S, a) = E(S) - \sum_{v \in \text{valori}(a)} |S_v| / |S| E(S_v)$
 - $\sum_{v \in \text{valori}(a)} |S_v| / |S| E(S_v)$ - informația scontată

Sisteme inteligente – SIS – Arbori de decizie

- Proces → Construirea AD → Algoritmul ID3/C4.5 → Selectare atribut
 - Câștigul de informație
 - exemplu

	a1	a2	a3	Clasa
d1	mare	roșu	cerc	clasa 1
d2	mic	roșu	pătrat	clasa 2
d3	mic	roșu	cerc	clasa 1
d4	mare	albastru	cerc	clasa 2

$$S = \{d1, d2, d3, d4\} \rightarrow p_+ = 2/4, p_- = 2/4 \rightarrow E(S) = -p_+ \log_2 p_+ - p_- \log_2 p_- = 1$$

$$S_{v=mare} = \{d1, d4\} \rightarrow p_+ = 1/2, p_- = 1/2 \rightarrow E(S_{v=mare}) = 1$$

$$S_{v=mic} = \{d2, d3\} \rightarrow p_+ = 1/2, p_- = 1/2 \rightarrow E(S_{v=mic}) = 1$$

$$S_{v=roșu} = \{d1, d2, d3\} \rightarrow p_+ = 2/3, p_- = 1/3 \rightarrow E(S_{v=roșu}) = 0.923$$

$$S_{v=albastru} = \{d4\} \rightarrow p_+ = 0, p_- = 1 \rightarrow E(S_{v=albastru}) = 0$$

$$S_{v=cerc} = \{d1, d3, d4\} \rightarrow p_+ = 2/3, p_- = 1/3 \rightarrow E(S_{v=cerc}) = 0.923$$

$$S_{v=patrat} = \{d2\} \rightarrow p_+ = 0, p_- = 1 \rightarrow E(S_{v=patrat}) = 0$$

$$\text{Gain}(S, a) = E(S) - \sum_{v \in \text{valori}(a)} |S_v| / |S| E(S_v)$$

$$\text{Gain}(S, a_1) = 1 - (|S_{v=mare}| / |S| E(S_{v=mare}) + |S_{v=mic}| / |S| E(S_{v=mic})) = 1 - (2/4 * 1 + 2/4 * 1) = 0$$

$$\text{Gain}(S, a_2) = 1 - (|S_{v=roșu}| / |S| E(S_{v=roșu}) + |S_{v=albastru}| / |S| E(S_{v=albastru})) = 1 - (3/4 * 0.923 + 1/4 * 0) = 0.307$$

$$\text{Gain}(S, a_3) = 1 - (|S_{v=cerc}| / |S| E(S_{v=cerc}) + |S_{v=patrat}| / |S| E(S_{v=patrat})) = 1 - (3/4 * 0.923 + 1/4 * 0) = 0.307$$

Sisteme inteligente – SIS – Arbori de decizie

- Proces → Construirea AD → Algoritmul ID3/C4.5 → Selectare atribut
 - Rata câștigului
 - Penalizează un atribut prin încorporarea unui termen – *split information* – sensibil la gradul de împrăștiere și uniformitate în care atributul separă datele
 - *Split information* – entropia relativ la valorile posibile ale atributului a
 - S_v – proporția exemplelor din setul de date S care au atributul a evaluat cu valoarea v
 - $splitInformation(S,a)=$
 - $\sum_{v=value(a)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$

□ Problema vampirilor

Items	Shadow	Garlic	Complexion	Accent	Vampire
i1	?	Yes	Pale	None	No
i2	Yes	Yes	Ruddy	None	No
i3	?	No	Ruddy	None	Yes
I4	No	No	Average	Heavy	Yes
5	?	No	Average	Odd	Yes
i6	Yes	No	Pale	Heavy	No
i7	Yes	No	Average	Heavy	No
i8	?	Yes	Ruddy	Odd	No

Sisteme inteligente – SIS – Arbori de decizie

□ Proces

- Construirea arborelui
- Utilizarea arborelui ca model de rezolvare a problemelor
 - Ideea de bază
 - Se extrag regulile formate în arborele anterior construit → Reguli extrase din arborele dat în exemplul anterior:
 - IF *age* = " ≤ 30 " AND *student* = "no" THEN *buys_computer* = "no"
 - IF *age* = " ≤ 30 " AND *student* = "yes" THEN *buys_computer* = "yes"
 - IF *age* = "31...40" THEN *buys_computer* = "yes"
 - IF *age* = " > 40 " AND *credit_rating* = "excellent" THEN *buys_computer* = "no"
 - IF *age* = " > 40 " AND *credit_rating* = "fair" THEN *buys_computer* = "yes"
 - Regulile sunt folosite pentru a clasifica datele de test (date noi). Fie *x* o dată pentru care nu se știe clasa de apartenență → Regulile se pot scrie sub forma unor predicate astfel:
 - IF *age*(*x*, ≤ 30) AND *student*(*x*, no) THEN *buys_computer* (*x*, no)
 - IF *age*(*x*, ≤ 30) AND *student* (*x*, yes) THEN *buys_computer* (*x*, yes)

Sisteme inteligente – SIS – Arbori de decizie

□ Proces

- Construirea arborelui
- Utilizarea arborelui ca model de rezolvare a problemelor
 - Dificultăți
 - *Underfitting* (sub-potrivire) → AD indus pe baza datelor de antrenament este prea simplu → eroare de clasificare mare atât în etapa de antrenare, cât și în cea de testare
 - *Overfitting* (supra-potrivire, învățare pe derost) → AD indus pe baza datelor de antrenament se potrivește prea accentuat cu datele de antrenament, nefiind capabil să generalizeze pentru date noi
 - Soluții:
 - fasonarea arborelui (pruning) → Îndepărtarea ramurilor nesemnificative, redundante → arbore mai puțin stufos
 - validare cu încrucișare

Sisteme inteligente – SIS – Arbori de decizie

□ Proces

- Construirea arborelui
- Utilizarea arborelui ca model de rezolvare a problemelor
- Tăierea (fasonarea) arborelui

□ Necesitate

- Odată construit AD, se pot extrage reguli (de clasificare) din AD pentru a putea reprezenta cunoștințele sub forma regulilor *if-then* atât de ușor de înțeles de către oameni
- O regulă este creată (extrasă) prin parcurgerea AD de la rădăcină până la o frunză
- Fiecare pereche (atribut, valoare), adică (nod, muchie), formează o conjuncție în ipoteza regulii (partea dacă), mai puțin ultimul nod din drumul parcurs care este o frunză și reprezintă consecința (ieșirea, partea atunci) regulii

□ Tipologie

- Prealabilă (*pre-pruning*)
 - Se oprește creșterea arborelui în timpul inducției prin sistarea divizării unor noduri care devin astfel frunze etichetate cu clasa majoritară a exemplurilor aferente nodului respectiv
- Ulterioară (*post-pruning*)
 - După ce AD a fost creat (a crescut) se elimină ramurile unor noduri care devin astfel frunze → se reduce eroarea de clasificare (pe datele de test)

Sisteme inteligente – SIS – 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>

□ Biblio

- <http://www.public.asu.edu/~kirkwood/DASstuff/decisiontrees/index.html>

Sisteme inteligente – SIS – Arbori de decizie

□ Avantaje

- Ușor de înțeles și interpretat
- Permit utilizarea datelor nominale și categoriale
- Logica deciziei poate fi urmărită ușor, regulile fiind vizibile
- Lucrează bine cu seturi mari de date

□ Dezavantaje

- Instabilitate → modificarea datelor de antrenament
- Complexitate → reprezentare
- Greu de manevrat
- Costuri mari pt inducerea AD
- Inducerea AD necesită multă informație

Sisteme inteligente – SIS – Arbori de decizie

□ Dificultăți

■ Existența mai multor arbori

- Cât mai mici
- Cu o acuratețe cât mai mare (ușor de "citit" și cu performanțe bune)
- Găsirea celui mai bun arbore → problemă NP-dificilă

■ Alegerea celui mai bun arbore

- Algoritmi euristici
- ID3 → cel mai mic arbore acceptabil
 - → teorema lui Occam: "always choose the simplest explanation"

■ Attribute continue

- Separarea în intervale
 - Câte intervale?
 - Cât de mari sunt intervalele?

■ Arbori prea adânci sau prea stufoși

- Fasonarea prealabilă (pre-pruning) → oprirea construirii arborelui mai devreme
- Fasonarea ulterioară (post-pruning) → înlăturarea anumitor ramuri

Sisteme Inteligente – SI – Învățare automată

□ Tipologie

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

- În funcție de modelul învățat (algoritmul de învățare):
 - Arbori de decizie
 - Rețele neuronale artificiale
 - **Mașini cu suport vectorial (MSV)**
 - Algoritmi evolutivi
 - Modele Markov ascunse

Sisteme inteligente – SIS – MSV

- Mașini cu suport vectorial (MSV)
 - Definire
 - Tipuri de probleme rezolvabile
 - Avantaje
 - Dificultăți
 - Tool-uri

Sisteme inteligente – SIS – MSV

□ Definiere

- Dezvoltate de Vapnik în 1970
- Popularizate după 1992
- Clasificatori liniari care identifică un hiperplan de separare a clasei pozitive de clasa negativă
- Au o fundamentare teoretică foarte riguroasă
- Funcționează foarte bine pentru date de volum mare (analiza textelor, analiza imaginilor)

■ Reamintim

- Problemă de învățare supervizată în care avem un set de date de forma:
 - (x^d, t^d) , cu:
 - $x^d \in \mathbf{R}^m \rightarrow x^d = (x^d_1, x^d_2, \dots, x^d_m)$
 - $t^d \in \mathbf{R} \rightarrow t^d \in \{1, -1\}$, $1 \rightarrow$ clasă pozitivă, $-1 \rightarrow$ clasă negativă
 - cu $d = 1, 2, \dots, n, n+1, n+2, \dots, N$
- Primele n date (se cunosc x^d și t^d) vor fi folosite drept bază de antrenament a MSV
- Ultimele $N-n$ date (se cunosc doar x^d , fără t^d) vor fi folosite drept bază de testare a MSV

Sisteme inteligente – SIS – MSV

□ Definire

- MSV găsește o funcție liniară de forma $f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b$, (\mathbf{w} -vector pondere) a.î.

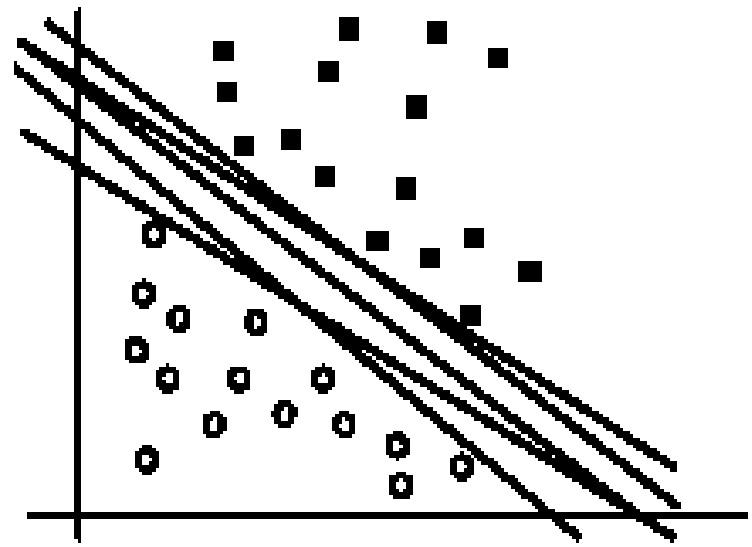
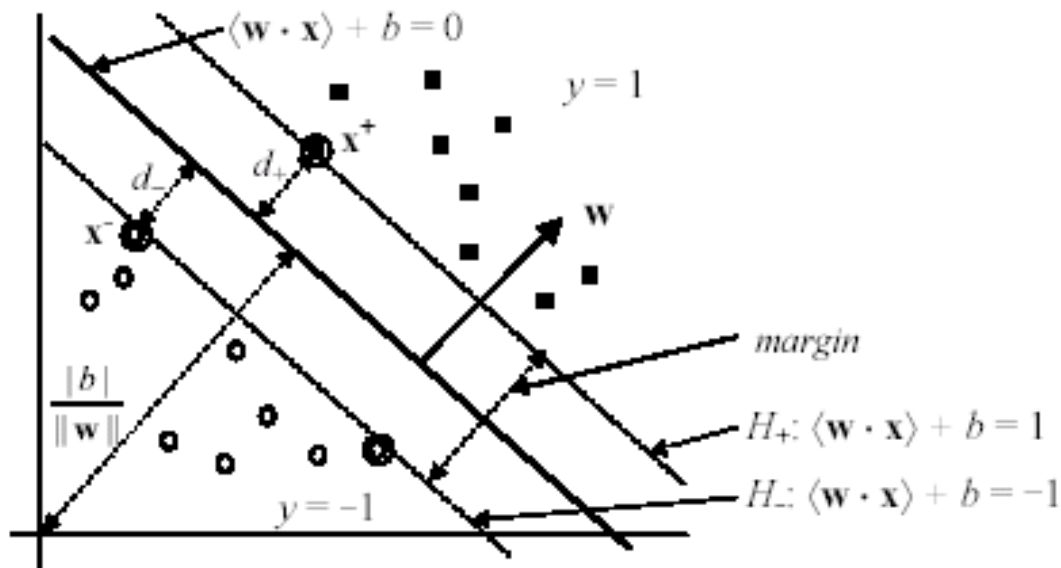
$$y_i = \begin{cases} 1 & \text{if } \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 0 \\ -1 & \text{if } \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b < 0 \end{cases}$$

- $\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0 \rightarrow$ hiperplanul de decizie care separă cele 2 clase

Sisteme inteligente – SIS – MSV

□ Definiere

- Pot exista mai multe hiperplane
 - Care este cel mai bun hiperplan?
- MSV caută hiperplanul cu cea mai largă margine (cel care micșorează eroarea de generalizare)
 - Algoritmul SMO (*Sequential minimal optimization*)



Sisteme inteligente – SIS – MSV

□ Tipuri de probleme rezolvabile

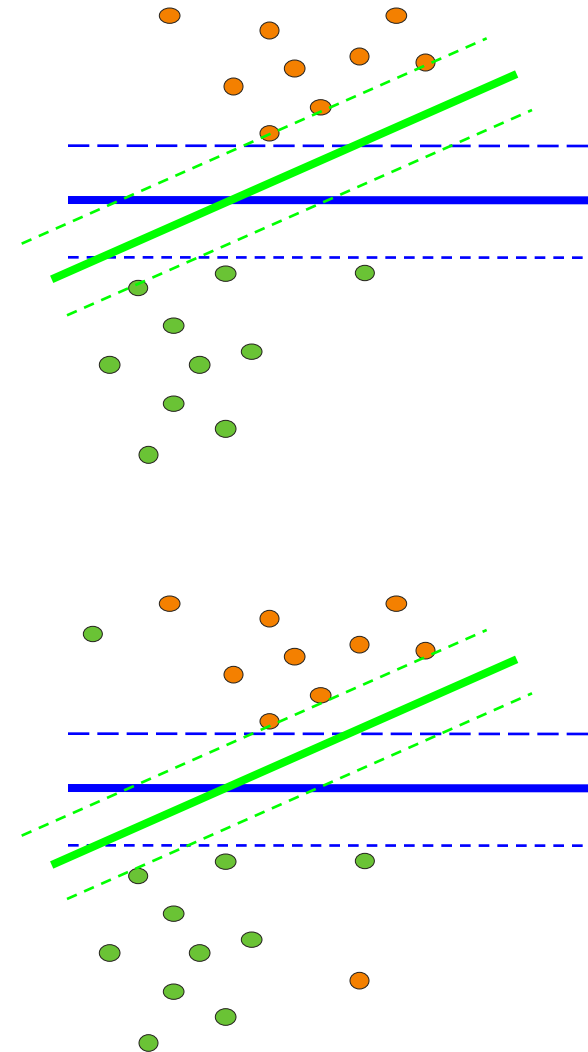
■ Probleme de clasificare → Cazuri de date

□ Liniar separabile

- Separabile
 - Eroarea = 0

■ Ne-separabile

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

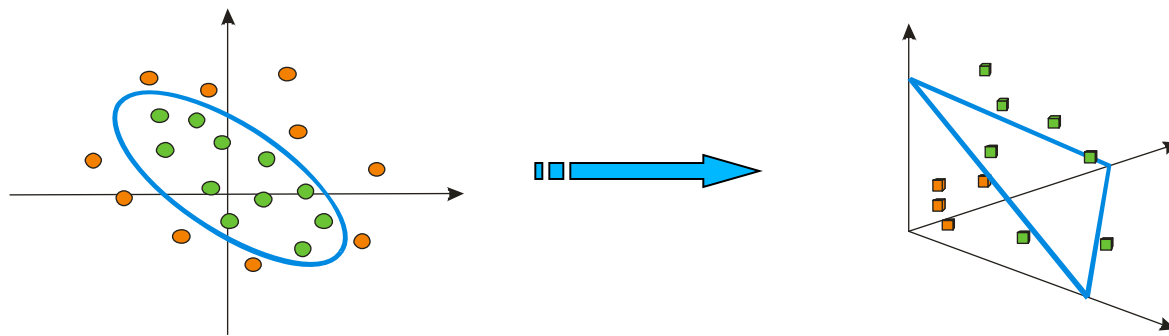


Sisteme inteligente – SIS – MSV

□ 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 liniar separabile
- În MSV, funcțiile kernel calculează distanța între 2 puncte
 - → kernelul \sim funcție de similaritate



Sisteme inteligente – SIS – MSV

□ Cazuri de date

- Non-liniar separabile → Kernele posibile

□ Clasice

- Polynomial kernel: $K(\mathbf{x}^{d1}, \mathbf{x}^{d2}) = (\mathbf{x}^{d1}, \mathbf{x}^{d2} + 1)^p$
- RBF kernel: $K(\mathbf{x}^{d1}, \mathbf{x}^{d2}) = \exp(- \|\mathbf{x}^{d1} - \mathbf{x}^{d2}\|^2 / 2\sigma^2)$

□ Kernele multiple

- Liniare: $K(\mathbf{x}^{d1}, \mathbf{x}^{d2}) = \sum w_i K_i(\mathbf{x}^{d1}, \mathbf{x}^{d2})$
- Ne-liniare
 - Fără coeficienți: $K(\mathbf{x}^{d1}, \mathbf{x}^{d2}) = K_1(\mathbf{x}^{d1}, \mathbf{x}^{d2}) + K_2(\mathbf{x}^{d1}, \mathbf{x}^{d2}) * \exp(K_3(\mathbf{x}^{d1}, \mathbf{x}^{d2}))$
 - Cu coeficienți: $K(\mathbf{x}^{d1}, \mathbf{x}^{d2}) = K_1(\mathbf{x}^{d1}, \mathbf{x}^{d2}) + c_1 * K_2 * (\mathbf{x}^{d1}, \mathbf{x}^{d2}) \exp(c_2 + K_3(\mathbf{x}^{d1}, \mathbf{x}^{d2}))$

□ Kernele pentru stringuri

□ Kernele pentru imagini

□ Kernele pentru grafe

Sisteme inteligente – SIS – MSV

□ Configurarea MSV

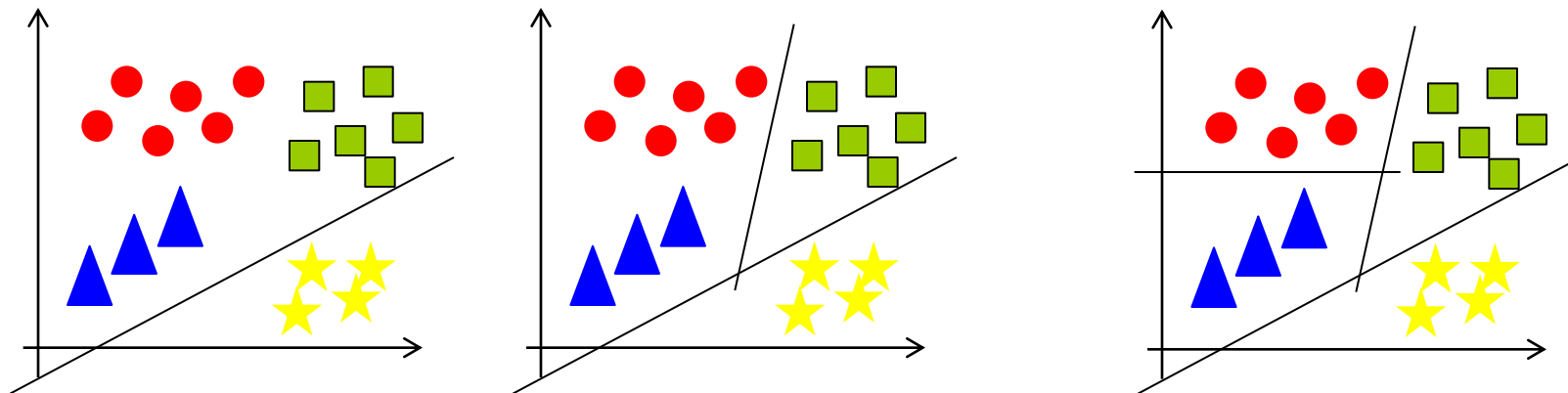
■ Parametrii unei MSV

- Coeficientul de penalizare C
 - C – mic → convergență lentă
 - C – mare → convergență rapidă

- Parametrii funcției kernel (care kernel și cu ce parametri)
 - Dacă m (nr de attribute) este mult mai mare decât n (nr de instanțe)
 - MSV cu kernel liniar (MSV fără kernel) → $K(\mathbf{x}^{d1}, \mathbf{x}^{d2}) = \mathbf{x}^{d1} \cdot \mathbf{x}^{d2}$
 - Dacă m (nr de attribute) este mare, iar n (nr de instanțe) este mediu
 - MSV cu kernel Gaussian $K(\mathbf{x}^{d1}, \mathbf{x}^{d2}) = \exp(-\|\mathbf{x}^{d1} - \mathbf{x}^{d2}\|^2 / 2\sigma^2)$
 - σ – dispersia datelor de antrenament
 - Attributele instanțelor trebuie normalizate (scalate la (0,1))
 - m (nr de attribute) este mic, iar n (nr de instanțe) este mare
 - Se adaugă noi attribute, iar apoi
 - MSV cu kernel liniar

Sisteme inteligente – SIS – MSV

- MSV pentru probleme de clasificare supervizate cu mai mult de 2 clase
 - Una vs. restul (one vs. all)



Sisteme inteligente – SIS – MSV

□ MSV structurate

■ Învățare automată

□ Normală $f: \mathcal{X} \rightarrow \mathbb{R}$

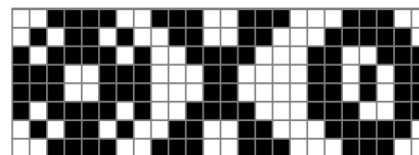
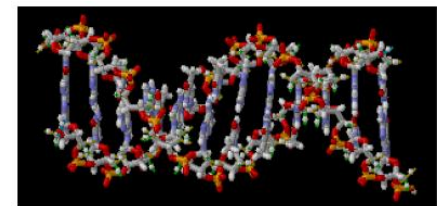
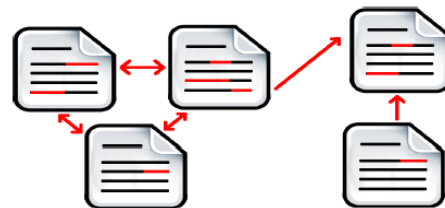
- Intrări de orice fel
- Ieșiri numerice (naturale, întregi, reale)

□ Structurată: $\mathcal{X} \rightarrow \mathcal{Y}$

- Intrări de orice fel
- Ieșiri de orice fel (simple sau structurate)

■ Informații structurate

- Texte și hiper-texte
- Molecule și structuri moleculare
- Imagini



Sisteme inteligente – SIS – MSV

□ MSV structurate

■ Aplicații

- Procesarea limbajului natural
 - Traduceri automate (ieșiri → propoziții)
 - Analiza sintactică și/sau morfologică a propozițiilor (ieșiri → arborele sintactic și/sau morfologic)

- Bioinformatică
 - Predicția unor structuri secundare (ieșirile → grafe bi-partite)
 - Predicția funcționării unor enzime (ieșirile → *path*-uri în arbori)

- Procesarea vorbirii
 - Transcrieri automate (ieșiri → propoziții)
 - Transformarea textelor în voce (ieșiri → semnale audio)

- Robotică
 - Planificare (ieșirile → secvențe de acțiuni)

Sisteme inteligente – SIS – MSV

□ Avantaje

- Pot lucra cu orice fel de date (liniar separabile sau nu, distribuit uniform sau nu, cu distribuție cunoscută sau nu)
 - Funcțiile kernel care crează noi atribute (features) → straturile ascunse dintr-o RNA
- Dacă problema e convexă oferă o soluție unică → optimul global
 - RNA pot asocia mai multe soluții → optime locale
- Selectează automat mărimea modelului învățat (prin vectorii suport)
 - În RNA straturile ascunse trebuie configurate de către utilizator *a priori*
- Nu învață pe derost datele (overfitting)
 - RNA se confruntă cu problema overfitting-ului chiar și cand modelul se învață prin validare încrucișată

□ Dificultăți

- 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/>
- SVM Torch → <http://www.torch.ch/>
- <http://www.support-vector-machines.org/>

Recapitulare



- Sisteme care învață singure (SIS)
 - Regresie logistică

 - kNN

 - Arbori de decizie
 - Fiecare nod intern corespunde unui atribut
 - Fiecare ramură de sub un nod (atribut) corespunde unei valori a atributului
 - Fiecare frunză corespunde unei clase (etichete) – conține toate datele din acea clasă

 - Mașini cu suport vectorial
 - Hiperplan de separare
 - Funcții kernel

Cursul următor

A. Scurtă introducere în Inteligența Artificială (IA)

B. Rezolvarea problemelor prin căutare

- Definirea problemelor de căutare
- Strategii de căutare
 - Strategii de căutare neinformate
 - Strategii de căutare informate
 - Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
 - Strategii de căutare adversarială

C. Sisteme inteligente

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

Cursul următor –

Materiale de citit și legături utile

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

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

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

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

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