

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/323388145>

Data Mining. Concepte, Modele si Tehnici. Ed. Albastra 2006

Book · February 2018

CITATIONS

0

READS

1,091

1 author:



Florin Gorunescu

University of Pitesti

107 PUBLICATIONS 1,971 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



New paper: F.Gorunescu, S. Belciug, Boosting backpropagation algorithm by stimulus-sampling: Application in computer-aided medical diagnosis, Journal of Biomedical Informatics, 2016 - <http://authors.elsevier.com/a/1TXHC5SMDQR4i> [View project](#)



Intelligent decision support systems in health care [View project](#)

CUPRINS

PREFATĂ.....

1. INTRODUCERE ÎN DATA MINING.....
1.1. Ce este și ce nu este Data Mining?.....
1.2. De ce Data Mining?.....
1.3. Cum se „minerește” în date?.....
1.4. Probleme rezolvabile cu Data Mining.....
1.4.1. Clasificarea.....
1.4.2. Clustering.....
1.4.3. Descoperirea regulilor de asociere.....
1.4.4. Descoperirea pattern-urilor secvențiale.....
1.4.5. Regresia.....
1.4.6. Detectarea deviațiilor/anomalyilor.....
1.5. Despre modelare și modele.....
1.6. Aplicații de Data Mining.....
1.7. Terminologie Data Mining.....
1.8. Confidentialitatea datelor.....
2. „MINA” DE DATE.....
2.1. Ce sunt datele?.....
2.2. Tipuri de mulțimi de date.....
2.3. Calitatea datelor.....
2.4. Tipuri de atribute.....
3. ANALIZA EXPLORATORIE A DATELOR.....
3.1. Ce este analiza exploratorie a datelor?.....
3.2. Statistica descriptivă.....
3.2.1. Parametrii descrierii statistice.....
3.2.2. Descrierea statistică a unei serii de cupluri.....
3.2.3. Reprezentarea grafică a unei mulțimi de date.....
3.3. Analiza matricei corelațiilor.....
3.4. Vizualizarea datelor.....
3.5. Examinarea repartițiilor variabilelor.....
3.6. Modele liniare și aditive avansate.....
3.6.1. Regresia liniară multiplă.....
3.6.2. Regresia logistică.....
3.6.3. Modelul de regresie Cox.....
3.6.4. Modele aditive.....
3.6.5. Serii temporale. Prognoză.....

3.7. Tehnici exploratorii multivariate.....
3.7.1. Analiza factorială.....
3.7.2. Analiza componentelor principale.....
3.7.3. Analiza canonică.....
3.7.4. Analiza discriminant.....
3.8. OLAP.....
3.9. Detectarea anomaliiilor.....
4. ARBORI DE CLASIFICARE ȘI DECIZIE.....
4.1. Ce este un arbore de clasificare și decizie?.....
4.2. Construirea unui arbore de clasificare și decizie.....
4.2.1. Indexul GINI.....
4.2.2. Câștigul de informație.....
4.2.3. Măsura de clasificare greșită.....
4.3. Metode computaționale.....
4.3.1. Acuratețea predictivă.....
4.3.2. Condiția de STOP pentru partiționare.....
4.3.3. Fasonarea arborilor de clasificare.....
4.3.4. Extragerea regulilor de clasificare din arborii de decizie.....
4.4. Avantajele arborilor de clasificare și decizie.....
5. TEHNICI ȘI MODELE DE DATA MINING.....
5.1. Metode în Data Mining.....
5.2. Clasificatori bayesieni.....
5.3. Rețele neuronale artificiale.....
5.3.1. Perceptronul.....
5.3.2. Tipuri de rețele neuronale artificiale.....
5.3.3. Rețele neuronale probabiliste.....
5.3.4. Mașini cu suport vectorial.....
5.4. Clasificare bazată pe reguli de asociere.....
5.5. k -nearest neighbor.....
5.6. Mulțimi rough.....
5.7. Clustering.....
5.7.1. Clustering ierarhic.....
5.7.2. Clustering neierarhic.....
5.8. Algoritmi genetici.....
5.8.1. Componentele unui algoritm genetic.....
5.8.2. Arhitectura unui algoritm genetic.....
6. PERFORMANȚA CLASIFICĂRII.....
6.1. Costul și acuratețea clasificării.....
6.2. Curbe de învățare.....
6.3. Curbele ROC.....
6.4. Comparația statistică a performanțelor clasificării.....

BIBLIOGRAFIE.....

INDEX.....

PREFĂTĂ

Interesul crescând în domeniul Data Mining poate fi motivat foarte simplu și convingător prin necesitatea stringentă, comună multor domenii de referință, de a stoca, accesă, descrie, modela și, mai ales, de a înțelege mulțimi foarte mari de date.

Procesul descoperirii de cunoștințe este la fel de vechi ca și omenirea. Începând cu descoperirea focului și ajungând la studiile actuale privind marketingul, omul a făcut „data mining” din totdeauna fără să-și dea seama, precum a făcut proză burghezul gentilom al lui Molière. Ajutat astăzi de puterea formidabilă de calcul a computerelor, se poate a冒tura acum în explorarea informației, utilizând mijloacele cele mai eficiente de lucru cu datele disponibile.

Ultimele decade au fost caracterizate de un adevărat „bombardament” de date în cele mai diverse domenii ca, de pildă: finanțele și marketingul, medicina, biologia, astronomia, meteorologia, Internetul, criptografia etc. Provocarea de a căuta înțelesuri în acest ocean informațional a dus la dezvoltarea de noi tehnici statistice pentru gestionarea datelor, dar acest lucru nu a fost nici pe deosebire suficient. În consecință, a fost nevoie de conceperea de noi arii de cercetare, care să poată gestiona eficient informația disponibilă. În acest context a apărut conceptul de „minerit” în date, cu toate că, din punct de vedere istoric, diferitele sale tehnici au fost stabilite în mod independent cu mult timp înainte. Din această perspectivă putem privi, sintetic, Data Mining ca o „intersecție” a Statisticii cu Inteligența Artificială și cu sistemele de baze de date. Data Mining reunește modele și tehnici bine stabilite în cadrul acestor domenii, creând o adevărată „orchestră” în care fiecare instrument își interpretează distinct partitura și toate împreună reușesc să creeze mediul ideal pentru deslușirea unor sensuri ascunse în date.

Această carte reprezintă o introducere în domeniul Data Mining, prezentând concepțele de bază, principalele modele și tehnici, precum și un număr însemnat de exemple și aplicații. Obiectivul cărții este deschiderea apetitului cititorului pentru acest domeniu fascinant, convingându-l să aprofundeze noțiunile prezentate și, de ce nu, să le aplice în domeniul său de interes. Cunoștințele cerute pentru înțelegerea celei mai mari părți a cărții sunt minimale, implicând noțiuni de bază din Statistică și Informatică. Pentru cititorul interesat de cunoștințe avansate există un număr suficient de referințe bibliografice. Deoarece există o foarte bogată literatură de specialitate și numeroase surse pe Internet, toate în limba Engleză, a fost adoptată pe cât a fost posibil prezentarea terminologiei în această limbă (alături, evident, de forma românească). În acest mod, se pot căuta cu mare ușurință noțiunile și referințele pe Internet sau în indexul alfabetic al cărților.

Cartea este structurată în şase capitulo, încercând să acopere cele mai importante topice ale domeniului Data Mining.

Capitolul 1 examinează chestiuni fundamentale privind Data Mining: ce este și ce nu este Data Mining, cum se „minerește” în date, principalele probleme rezolvabile cu Data Mining, modelare și modele, cele mai cunoscute aplicații, confidențialitatea datelor și terminologia specifică Data Mining.

Capitolul 2 explică ce reprezintă datele, materia primă a Data Mining, ocupându-se cu tipurile de date și atribute și calitatea acestora.

Capitolul 3 se concentrează asupra analizei exploratorii a datelor, explicând ce reprezintă aceasta și expunând principalele sale tehnici: statistica descriptivă, analiza matricei corelațiilor, vizualizarea datelor, examinarea repartițiilor variabilelor, modele liniare și aditive, tehnici exploratorii multivariate, OLAP și detectarea anomaliei din date.

Capitolul 4 se axează pe prezentarea unei tehnici populare de Data Mining, arborii de clasificare și decizie, plecând de la construirea lor și terminând cu extragerea regulilor de clasificare.

Capitolul 5, cel mai dens în informație, prezintă sintetic cele mai cunoscute modele și tehnici Data Mining: clasificatori bayesieni, rețelele neuronale, mașinile cu suport vectorial, regulile de asociere, *k*-nearest neighbor, multimile rough, metode de clustering și, în fine, algoritmii genetici, precum și câteva aplicații interesante ale tuturor acestora.

Capitolul 6, ultimul, explică principalele chestiuni privind măsurarea performanței clasificării, privită ca metodă fundamentală în Data Mining. Astfel, se trec în revistă noțiuni ca: acuratețea și costul clasificării, curbele de învățare, curbele ROC, compararea performanței modelelor.

Cartea este adresată tuturor celor care au tangență cu procesarea informației (informaticieni, matematicieni, economisti, biologi, meteorologi, medici, psihologii etc.), punându-le la dispoziție instrumente eficiente de lucru. Așa cum s-a mai spus, Data Mining nu este o „piatră filosofală” care transformă datele în „aurul” pur al cunoașterii, tehniciile sale permitând utilizatorului să extragă din date anumite înțelesuri, altfel nesenzabile. Putem privi în acest context Data Mining ca pe o „busolă” care ne ajută să găsim calea în vîrtejul informațional actual.

Scrierea acestei cărți a fost înlesnită de participarea la grantul „Rewarding and Developing Staff”, HR Strategy - International Exchange of Academic Staff and Research, inițiat și finanțat de University of Westminster, London, UK în perioada 1 August 2004-31 Iulie 2005.

1. INTRODUCERE ÎN DATA MINING

1.1. Ce este și ce nu este Data Mining?

Începând cu anii 1990 se tot vorbește de *Data Mining* (DM), sau mai pe românește „minerit” în date, în foarte multe medii, plecând de la cele academice și până la cele de afaceri sau medicale, îndeosebi. Fiind o arie de cercetare fără o istorie prea lungă, nedepășind faza „adolescentei”, este încă disputată de câteva domenii științifice care o revendică. Sunt notorii afirmațiile lui Pregibon -Research Scientist Google Inc.- că: „*Data Mining reprezintă un amestec de Statistică, IA (Inteligentă Artificială) și cercetare în baze de date*”, sau că sunt unii care consideră DM ca „*un cuvânt murdar în Statistică*” –cel mai probabil aceștia fiind statisticieni care nu considerau cu ceva timp în urmă DM ca pe ceva suficient de interesant pentru ei (D. Pregibon, *Data Mining, Statistical Computing and Graphics Newsletter*, 7, pp. 8, 1997).

În acest prim capitol vom trece în revistă chestiunile fundamentale legate de acest subiect, cum ar fi:



- Ce este (și ce nu este) Data Mining?
- De ce Data Mining?
- Cum se „sapă” („minerește”) în date?
- Probleme rezolvabile cu metode Data Mining
- Despre modelare și modele
- Aplicații Data Mining
- Terminologie Data Mining
- Confidențialitatea datelor

Înainte însă de a încerca o definiție pentru Data Mining, să punctăm câteva aspecte privind geneza sa. *Data Mining*, cunoscută și ca „*descoperirea de cunoștințe în baze de date*” (KDD = *knowledge-discovery in databases*), are trei rădăcini generice, de la care a împrumutat atât tehnici de lucru cât și terminologie:

- *Statistica* –cea mai longevivă rădăcină a DM și fără de care nu ar exista. Statistica clasică aduce în DM tehnici definitorii, pe care le putem rezuma în ceea ce este îndeobște cunoscut ca *Analiza exploratorie a datelor* (EDA = *Exploratory Data Analysis*), utilizată pentru a identifica relații sistematice între diferite variabile atunci când nu există informații

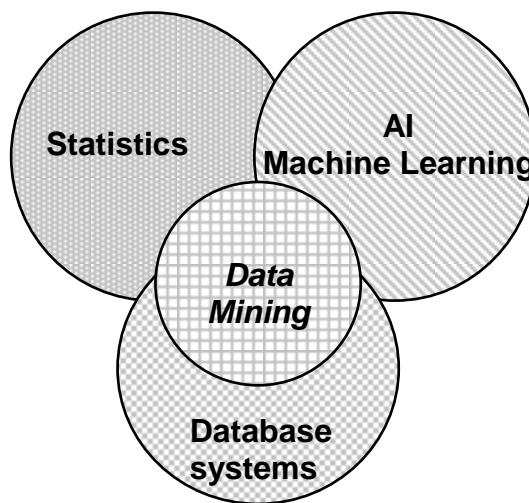
suficiente asupra naturii acestora. Dintre tehniciile EDA clasice, utilizate în DM, putem enumera:

(a) metodele computaționale: statistica descriptivă (repartiții, parametri statistici clasici (media, mediana, deviația standard etc.), corelații, tabele multiple de frecvență, metode statistice multivariate (clustering, analiza factorială, analiza funcție discriminant, arbori de clasificare și regresie, modele de regresie liniară și neliniară, analiza corespondențelor, serii temporale etc.);

(b) vizualizarea datelor, care poate fi considerată ca una dintre cele mai puternice și, în același timp, atractive metode de explorare a datelor prin reprezentarea informației în format vizual. Printre cele mai des întâlnite tehnici de vizualizare găsim: histograme, grafice rectangulare (box plots), grafice de împărțiere (scatter plots), grafice de contur, grafice matriciale etc.

Pentru cei interesați de aprofundarea tehniciilor EDA, facem referire la NIST/SEMATECH -*Engineering Statistics Handbook*, Cap. I „Exploratory data analysis”, <http://www.itl.nist.gov/div898/handbook/index.htm>.

- *Inteligenta artificială* (AI = Artificial Intelligence), construită pe heuristici spre deosebire de Statistică, contribuie la dezvoltarea DM prin tehnici de procesarea informației bazate pe modelul raționamentului uman. *Învățarea automată* (ML = machine learning) reprezintă o arie extrem de importantă a AI în raport cu dezvoltarea DM, prin utilizarea tehniciilor care permit computerului să „învețe” prin antrenament’.
- *Sisteme de baze de date* (DBS = database systems), reprezintă o a treia rădăcină a DM, procurând materialul care trebuie „minerit” utilizând metodele amintite mai sus.



Necesitatea procesului de „minerit” al datelor poate fi rezumată astfel, privită fiind prin prisma unor domenii care au nevoie vitală de astfel de tehnici de investigație:

- Punct de vedere *economic* (afaceri-finanțe) - există un imens volum de date deja colectate în diverse domenii ca: date Web, e-comerț, super/hyper market-uri, tranzacții financiar-bancare etc., gata pentru a fi analizate în vederea luării unor decizii optime;
- Punct de vedere *medical* – există actualmente multe și diverse baze de date din domeniul sănătății (medico-farmaceutic) care au fost doar parțial analizate, mai ales cu mijloacele specifice medicinii și care conțin foarte multă informație încă neexplorată suficient;
- Punct de vedere *științific-cercetare* – există imense baze de date din domeniul cercetării științifice din cele mai diverse domenii (astronomie, meteorologie, biologie, lingvistică etc.) care nu pot fi explorate cu mijloacele tradiționale.

Înțând cont de faptul că, pe de o parte există un imens volum de date încă neexplorate sistematic și, pe de altă parte domeniul computerelor și al Informaticii s-a dezvoltat exponential, a crescut presiunea utilizării de noi metode pentru descoperirea informației „ascunse” în date, informație care este aproape imposibil de detectat cu mijloacele tradiționale și utilizând doar capacitatea umană de analiză.

Datorită „tinereții” sale, acest nou domeniu științific nu are încă o terminologie (mai ales în românește) bine definită și acceptată de toți. Pe de altă parte, există oameni interesați de acest domeniu, care nu au pregătirea necesară, dar care vor să cunoască mai mult pentru a-și face o imagine cât de cât clară despre DM și, mai ales, despre posibilitățile de aplicare a sa. Din aceste cauze, peste tot unde a fost posibil, a fost utilizată și terminologia internațională (în limba Engleză, în original) îndeobște folosită, pentru a ușura atât utilizarea resurselor bibliografice internaționale dar, mai ales, căutarea pe Internet a termenilor de referință.

Să încercăm acum să definim ce este Data Mining. Este greu să optăm pentru o unică definiție, care să ne furnizeze o imagine cât de cât completă a fenomenului. De aceea vom prezenta câteva abordări mai mult sau mai puțin asemănătoare, care vor contura suficient de clar, sperăm, obiectul acestei discipline. Deci, prin Data Mining vom înțelege (abordări echivalente):

- Practica căutării automate de pattern-uri (modele, şabloane, tipare, forme etc.) în mulțimi mari de date, utilizând tehnici computaționale din Statistică, învățarea automată (*machine learning*) și recunoașterea formelor (*pattern recognition*);

- Extragerea netrivială din date a informației implicate, necunoscută încă, și potențial folositoare;
- Știința extragerii de informație folositoare din mulțimi mari de date sau baze de date;
- Explorarea și analiza unor mari cantități de date, prin mijloace automate sau semi-automate, în vederea descoperirii de pattern-uri folositoare;
- Procesul descoperirii automate a informației –identificarea pattern-urilor ascunse și relațiilor cu datele.

(vezi și http://en.wikipedia.org/wiki/Data_mining, [53], [88], [92], [162])

Metaforic vorbind, prin DM înțelegem procesul de căutare a unui ac într-un car cu fân, utilizând un senzor de metale, tocmai pentru a mări viteza de căutare, „automatizând” procesul respectiv.

Am văzut mai sus ce se poate înțelege prin DM. În acest context este interesant de văzut și „ce nu este DM”. Vom prezenta mai jos, tabelat, patru situații reale (adaptare [162]), ce ilustrează elocvent ce nu este DM prin comparație cu ce ar putea fi.

CE NU ESTE DATA MINING?	CE AR PUTEA FI DATA MINING?
Căutarea unui anumit număr de telefon într-o carte de telefoane	Căutarea unor nume de tipul „...escu” într-un stat din SUA
Căutarea unei anumite informații (e.g. despre bucătărie pe Google)	Gruparea laolaltă a informațiilor similare în funcție de context (e.g. despre bucătăria franceza, italiană etc., găsite pe Google)

Un exemplu sugestiv pentru a sublinia și mai mult diferența dintre ceea ce reprezintă ușual o căutare într-o bază de date și DM este următorul: „Cineva poate fi interesat de diferența între numărul de cumpărături de un anumit tip (e.g. electrocasnice) de la un supermarket în comparație cu un supermarket sau, eventual, de la supermarket-uri din două regiuni diferite”. În acest caz, respectivul dejă ia în considerație *a priori* ipoteza că există diferențe între un supermarket și un supermarket, sau între vânzări între cele două regiuni. În schimb în cazul DM în acest context, problema poate consta de exemplu în identificarea factorilor ce influențează volumul vânzărilor, fără a se baza pe nicio ipoteză considerată *a priori*. În concluzie, metodele DM încearcă identificarea de pattern-uri și relații ascunse, care nu sunt totdeauna evidente (și deci circumscrise unor anumite ipoteze ușor identificabile).

Așa după cum se observă din exemplele de mai sus, nu putem pune semnul egal între o căutare (cercetare) individuală a unui anumit obiect (indiferent de natura sa) și cercetarea de tip DM, care nu „caută” individualismul ci mulțimi de individualisme care, într-un fel sau altul, pot fi grupate după anumite criterii. Tot metaoric vorbind, diferența dintre o simplă căutare și un proces DM este aceea dintre căutarea unui copac anume și identificarea unei păduri (de unde și expresia „nu se vede pădurea de copaci” utilizată atunci când cercetarea nu este suficient de laxă în ceea ce privește constrângerile).

Să enumerăm mai jos două obiective ale DM [48], pentru a desluși mai clar aria sa de aplicabilitate.

- *Obiective predictive* (i.e. utilizarea unei părți din variabile pentru a prognoza una sau mai multe dintre celelalte variabile):
 - ▲ Clasificarea;
 - ▲ Regresia;
 - ▲ Detecția deviațiilor/anomaliiilor.
- *Obiective descriptive* (i.e. identificarea de pattern-uri ce descriu datele și care pot fi înțelese de utilizator):
 - ▲ Clustering (clusterizare);
 - ▲ Descoperirea regulilor de asociere;
 - ▲ Descoperirea de pattern-uri secvențiale.

1.2. De ce Data Mining?

Pare a fi ușor, la prima vedere, de a răspunde la o asemenea întrebare fără o prezentare prealabilă a tehnicielor DM și, mai ales, a aplicațiilor sale. Credem că mai sugestivă este prezentarea a două situații complet diferite în care DM a fost utilizată cu succes. Mai întâi este cazul unei situații, pe cât de dramatică pe atât de reală, privind rolul DM în rezolvarea unei probleme fundamentale, din nefericire, a zilelor noastre. Conform Wikipedia -*The Free Encyclopedia*- http://en.wikipedia.org/wiki/Data_mining, o utilizare recentă, mai mult decât notabilă a DM se referă la următorul fapt: „Serviciile de spionaj ale Armatei SUA ar fi descoperit cu un an înainte, utilizând tehnici DM, indicii privind atentatele de la 9.11.2001. Astfel, ar fi fost identificat atât liderul grupului de teroriști care au pus la cale atentatul sinucigaș, cât și alți trei membri ai unui grup Al Qaeda operând pe teritoriul SUA”. Din nefericire, se pare că informațiile nu au fost luate în seamă de autorități. Apoi este cazul unei întâmplări nostime, dar neplăcute pentru cel în cauză, relatată de Ramon C. Barquin –The Data Warehousing Institut– „Cuvânt înainte” [88]. Pe scurt, el

povestește că fiind plecat departe de casă, a fost contactat de compania telefonică la care era client pentru a i se aduce la cunoștință faptul că se presupunea că i-a fost furată cartela telefonică care a și fost folosită în mod fraudulos. Compania telefonică ajunsese la această concluzie pe baza analizei ultimelor con vorbiri telefonic efectuate utilizând cartela respectivă, con vorbiri către locații care nu se potriveau cu pattern-ul pe care compania îl construise pentru clientul său.

Iată două motive foarte solide pentru a considera cu toată seriozitatea acest domeniu, pe căt de fascinant, pe atât de complex, al descoperirii de informații acolo unde cunoașterea umană nu ne mai este de prea mare folos.

Există actualmente un mare număr de companii care au ca obiect de activitate DM [88]. Acest fapt se datorează mai ales cererii tot mai mari de servicii de DM de către piață finanțier-economică (a se vedea, de exemplu, domenii ca: *Business intelligence* (BI), *Business performance management* (BPM), *Customer relationship management* (CRM) etc.), și de medicină (Health Informatics, e-Health etc.), fără a neglija și alte domenii de interes precum telecomunicațiile, meteorologia, biologia etc.

Plecând de la prognoza în domeniul marketingului pentru marile companii transnaționale și trecând prin analiza tendințelor de tranzacționare a acțiunilor de la Bursă, construirea profilului clientului fidel, modelarea cererii de produse farmaceutice, automatizarea diagnosticării cancerului, detectarea fraudelor bancare, detectarea uraganelor, clasificarea stelelor și galaxiilor etc., observăm o paletă diversă de domenii în care tehniciile DM sunt folosite eficient, fapt ce dă un răspuns clar întrebării: *de ce Data Mining?*

Pe de altă parte nu trebuie considerat că DM poate rezolva orice problemă de descoperire de informație utilă în date. La fel ca și mineritul originar, și în cazul DM este posibil să se ,sape' în ,muntele' de date fără ca până la urmă să se dea peste filonul de aur al cunoașterii. Descoperirea de cunoștințe/informații utile depinde de foarte mulți factori, începând cu ,muntele' de date în care se ,minerește' și terminând cu ,uneltele' de DM utilizate și pricperea ,minerului'. Dacă nu există ,aur' în munte, degeaba se sapă. Pe de altă parte, filonul de aur, dacă există, trebuie identificat și evaluat corect și apoi, dacă este eficientă exploatarea sa, aceasta trebuie executată cu unelte de minerit corespunzătoare.

1.3. Cum se ,minerește' în date?

Să vedem acum în ce constă procesul de ,minerit' în date. Schematic, putem identifica trei pași definitorii ai procesului de DM:

- (a) *Explorarea datelor*, care se referă la ,curățarea' datelor, transformarea datelor, selectarea de submulțimi de date, selectarea caracteristicilor, acolo unde avem un mare număr de variabile etc.;

- (b) *Construirea modelului și validarea acestuia* se referă la considerarea a diferite modele și alegerea acelui care are cea mai bună performanță a progronei –evaluarea competitivă a modelelor (*competitive evaluation of models*);
- (c) *Aplicarea modelului* la date noi, în vederea producerii de progrone/estimații corecte pentru problemele cercetate.

Conform [88], putem identifica cinci principale etape ale „mineritului” în date:

- ◆ *Pregătirea datelor (data pre-processing)*. Înainte de a utiliza indiferent ce tehnică de DM, este absolut necesară pregătirea datelor „brute” (*raw data*) în vederea analizării eficiente. Există mai multe aspecte ale pregătirii inițiale a datelor înaintea procesării propriu-zise cu ajutorul tehnicilor DM [88], [162]. Mai întâi însă, se pune problema calității datelor (e.g. existența zgomotului (*noise*), a valorilor extreme/aberante (*outlier/anomaly*), a valorilor lipsă (*missing values*), a datelor dupicate, a datelor introduse incorect, a datelor expirate etc.). În funcție de problemele detectate privind calitatea datelor, se procedează la rezolvarea acestora cu metode specifice. De exemplu, în cazul existenței „zgomotului”, adică a unor distorsiuni ale valorilor (măsurătorilor) originale produse de perturbări aleatoare, se procedează la diferite tehnici de „filtrare”, care să îndepărteze/reducă efectul distorsiunilor. De exemplu, în cazul procesării semnalelor (*signal processing*), putem menționa, în afara filtrelor electronice (hard),filtrele „matematice” (soft), adică algoritmi matematici utilizați pentru modificarea componentei armonice a semnalului (e.g. moving average filter, filtrare Fourier etc.). În cazul existenței unor valori extreme, adică valori care se abat semnificativ de la media celorlalte valori ale datelor, se poate proceda fie la îndepărțarea lor, fie la utilizarea unor parametri (statistici) care să nu fie așa de sensibili la aceste valori extreme (e.g. utilizarea medianei în locul mediei, care este „sensibilă” la valorile extreme). Cazul valorilor lipsă este frecvent întâlnit în practica DM și are o multitudine de cauze. În această situație se folosesc diferite metode, ca: eliminarea obiectelor/instanțelor care au atribuite (valori) lipsă, estimarea valorilor lipsă, înlocuirea valorilor lipsă (e.g. cu media/mediana celorlalte, eventual ponderate), ignorarea lor, dacă este posibil, în cursul analizei etc. În cazul datelor dupicate (e.g. o persoană cu mai multe adrese e-mail) se procedează la „curățirea” datelor, adică ștergerea dupicatelor. Odată ce s-a rezolvat problema calității datelor, se trece la etapa propriu-zisă a pre-procesării lor, care constă, în principiu, în următoarele procedee:
 - ★ *Agregarea (aggregation)* care constă în combinarea mai multor obiecte/instanțe (attribute) existente într-un singur obiect/instanță (atribut) cu scopul reducerii volumului de date (a dimensionalității) și obținerea de date mai „stabile”, cu o variabilitate mai mică (e.g. „agregarea” satelor în comuna de care aparțin, a orașelor în județul de

care aparțin, a vânzărilor săptămânale în vânzări lunare sau anuale etc.).

- ★ *Eșantionarea (sampling)* este procesul de prelevare a unui eşantion reprezentativ pentru întreaga mulțime de date, reprezentând principala metodă de selectare a datelor. Metodele de creare a eşantioanelor formează un domeniu clasic al Statisticii și nu vom intra aici în detaliu tehnice (vezi [5], [28], [121], [164]). Menționăm totuși problema volumului eşantionului (*sample size*), care este importantă în echilibrul dintre eficientizarea procesului de DM (obținută prin diminuarea volumului de date procesate) și pierderea semnificativă de informație prin utilizarea unui volum redus de date - domeniu cunoscut în Statistică ca „Analiza puterii și calculul volumului eşantionului” (*power analysis and sample size calculation*) și utilizând diferite tehnici specifice (one mean *t*-test, two means *t*-test, one proportion Z-test etc.) care depind de problema respectivă.
- ★ *Reducerea dimensionalității (dimensionality reduction)*. Se știe că atunci când dimensiunea (numărul atributelor) crește, va crește implicit și „împrăștierea” datelor, ceea ce va duce la o procesare ulterioară dificilă prin creșterea memoriei necesare și scăderea vitezei de lucru. În DM acest fenomen este denumit, mai mult decât sugestiv, ca „blestemul dimensionalității” (*curse of dimensionality*). „Antidotul” pentru acest „blestem” este reducerea dimensionalității, prin care se obține reducerea memoriei alocate și a timpului de procesare, vizualizare superioară, eliminarea caracteristicilor irelevante și eventuala diminuare/eliminare a zgomotului. Ca tehnici de reducere a dimensionalității amintim: analiza factorială (*factor analysis*) și analiza componentelor principale (*principal components analysis*).
- ★ *Selectarea de submulțimi de caracteristici (feature subset selection)* se utilizează pentru eliminarea caracteristicilor redundante și a celor irelevante studiului, prin utilizarea de metode specifice (e.g. *brute-force approach*, *embedded approach*, *filter approach*, *wrapper approach* –vezi [162]).
- ★ *Crearea de caracteristici (feature creation)* se referă la procesul de creare de noi atribute (artificiale), care pot capta mai bine informațiile importante din date decât cele originale. Ca metode de creare de noi caracteristici amintim extragerea de caracteristici (*feature extraction*), aplicarea datelor într-un nou spațiu (*mapping data to new space*), construirea de caracteristici (*feature construction*) –idem.
- ★ *Discretizarea și binarizarea (discretization and binarization)* adică, pe scurt, trecerea de la date continue la date discrete (e.g. trecerea de la valori reale la valori întregi) și convertirea valorilor multiple în valori

binare (e.g. convertirea unei fotografii cu 256 de culori doar în alb/negru, trecerea de la mai multe categorii la doar două categorii etc.).

- ★ *Transformarea atributelor (attribute transformation)* reprezintă, în principiu, transformarea unor atribute vechi în atribute noi, utilizând o anumită transformare (e.g. transformare prin funcții matematice (e^x , $\log x$, $\sin x$, x^n etc.), normalizarea $x \rightarrow \frac{x}{\|x\|}$ etc.), transformare ce îmbunătățește procesul de DM.

- ◆ *Definirea studiului (cercetarea)* reprezintă al doilea pas în procesul de DM după etapa pre-procesării datelor. Să remarcăm că, în contextul întregului proces de DM, etapa de procesare a datelor se va repeta ori de câte ori va fi necesar, deci este într-adevăr corect să spunem etapa de pre-procesare. Întâi de toate, fiind un proces de analiză (minerit) de date, trebuie stabilită mulțimea datelor ce vor fi analizate, adică „mina” în care se va „săpa” pentru a descoperi informația ascunsă. Odată stabilit domeniul datelor în care se va „mineră”, trebuie stabilit modul de eșantionare a datelor, deoarece, de obicei, nu se lucrează cu întreaga bază de date. Să menționăm aici un aspect important al procesului de DM, și anume stabilirea modului în care se vor analiza datele alese. În acest context, ne vom opri, în trecere, asupra tipurilor de învățare automată -utilizată intensiv în DM, și anume învățarea supervizată/controlată și învățarea nesupervizată/necontrolată, deoarece definirea studiului DM depinde de metodologia aleasă. Pe scurt, prin termenul *învățarea supervizată (supervised learning)* în DM înțelegem acea tehnică a învățării automate utilizată pentru crearea unei funcții plecând de la mulțimea datelor de antrenament. Scopul învățării supervizate este acela de a prognoza valoarea (output-ul) funcției pentru orice obiect/eșantion/instanță de intrare (input) nou, după parcurgerea procesului de antrenament. Un astfel de exemplu de tehnică de învățare automată este tehnica clasificării (metodă predictivă). Spre deosebire de învățarea supervizată, în învățarea nesupervizată (*unsupervised learning*) în DM modelul construit este adaptat observațiilor, distingându-se deci prin faptul că nu există *a priori* nici un output. Un exemplu clasic de învățare nesupervizată este metoda de clustering (metodă descriptivă). Revenind, să observăm că în cazul utilizării metodelor de învățare supervizată definirea studiului se referă la identificarea unei variabile (atribut) dependente, care va fi considerată ca output pentru studiu și stabilirea celorlalte variabile care să „explice” variabila aleasă ca output (variabile predictoare). De exemplu, într-un studiu medical ne interesează să înțelegem cum este influențată declanșarea sau evoluția unei anumite boli (e.g. infarctul miocardic) de către anumiți factori „de risc” (e.g. greutatea, vârsta, fumatul, ereditatea etc.). În schimb, în cazul utilizării

metodelor de învățare nesupervizată, scopul general al unui model este gruparea obiectelor/instanțelor similare sau identificarea excepțiilor din setul de date. Un exemplu în acest sens este identificarea tipologiei clienților cu același comportament în ceea ce privește cumpărarea unor anumite tipuri de bunuri, iar procesul de identificarea excepțiilor poate avea ca scop descoperirea fraudelor (exemplul menționat mai sus în legătura cu utilizarea cartelei telefonice este sugestiv). Odată stabilită mulțimea datelor ce vor fi analizate, urmează să definim scopul procesului de DM. Vom prezenta, mai jos, câteva repere în acest sens [88]:

- * *Înțelegerea limitelor studiului* se referă la un set de probleme cu care se confruntă utilizatorul de tehnici de DM, plecând de la ideea de principiu că DM nu poate face minuni și există totuși limite privind așteptările legate de rezultatele aplicării sale. Prima problemă se referă la alegerea scopului studiului: „este sau nu necesar să se aibă în vedere *a priori* un anumit scop, sau se poate „mineră” ,orbește” în date în căutarea „aurului” ascuns acolo?” Un răspuns înțelept pentru această întrebare este că, totuși, trebuie stabilit un tel sau niște obiective generale ale studiului, tocmai pentru a opera corespunzător cu datele disponibile. Încă ne confruntăm cu veșnică controversă în această chestiune - fiind vorba de „săpat” în date - cât de importantă este definirea *a priori* a „țintelor” studiului? Așa cum am menționat mai sus, trebuie totdeauna să ne definim un scop, mai mult sau mai puțin precis, atunci când începem un studiu de DM („să căutăm acul în carul cu fân”, dar în mod intelligent). Acest lucru ne va scuti de mult efort și pierdere de timp, printr-o bună proiectare a studiului, începând chiar cu alegerea și pregătirea datelor și terminând cu identificarea potențialilor beneficiari. O a doua problemă se referă la modul în care se procedează în cazul datelor necorespunzătoare. În acest sens se poate aplica ideea că o bună înțelegere a datelor disponibile (chiar nu de prea bună calitate) poate duce la o mai bună utilizare a lor. Mai departe, odată cu proiectarea și apoi utilizarea unui model, întrebările nu încetează ci, mai degrabă, se multiplică (e.g. „se poate aplica modelul și în alt context?”, „mai există și alte metode de a obține rezultate asemănătoare?” etc.). În fine, este posibil ca după terminarea studiului să nu se obțină nimic nou, relevant sau util. Acest rezultat însă nu trebuie să ne opreasă de la a utiliza tehniciile de DM. Chiar dacă obținem un rezultat la care ne așteptam, mai ales în cazul unei probleme deja binecunoscute, avem totuși căștigul că acesta a fost încă odată confirmat de DM. Mai mult, utilizând repetat modelul care doar a confirmat informații cunoscute, dar pe date noi, este posibil ca la un moment dat să obținem rezultate diferite de ceea ce ne-am fi așteptat, ceea ce indică modificări de pattern-uri sau tendințe, necesitând investigații suplimentare ale datelor respective.

- * *Alegerea unui studiu corespunzător* pentru rezolvarea unei anumite probleme se referă la modul „natural” în care studiul ales este corelat cu soluția problemei respective. Un exemplu de studiu corect ales este identificarea profilului pacientului „standard” pentru o anumită boală în vederea îmbunătățirii tratamentului bolii respective. În schimb, un studiu necorespunzător este acela al cărui scop este înțelegerea cărui tip de telespectatori se circumscrizu cei cărora le place fotbalul, în vederea optimizării programării de filme romantice pe un canal TV(!).
- * *Tipurile de studii* se referă la scopurile propuse prin utilizarea tehniciilor de DM. Ca exemple de astfel de tipuri, menționăm: crearea pe baza datelor aflate la dispoziție a profilului fumătorului în raport cu un nefumător, descoperirea caracteristicilor diferitelor tipuri de corpori cerești pe baza datelor obținute de la telescoape (*Sky Survey Cataloging*) în vederea clasificării altora noi, segmentarea clientilor în categorii distincte care vor deveni „înțe” pentru vânzarea unor mix-uri de produse etc.
- * *Selectarea elementelor pentru analiză* este iarăși o problemă care nu este complet rezolvată și nici nu ar putea fi, deoarece de mulți factori. Astfel, una este să se procedeze pentru prima oară la analiza unui anumit set de date și alta dacă deja există o experiență anterioară în acest sens. Cel care debutează va alege tot setul de date disponibile, pe când cercetătorul experimentat se va concentra doar pe anumite aspecte relevante. În altă ordine de idei, este important tipul studiului pe care vrem să-l facem: clasificare (e.g. arbori de clasificare, rețele neuronale), clustering (e.g. k -means, two-way joining), regresie (liniară/neliniară multiplă, logistică) etc. De asemenea, este important scopul studiului în selectarea elementelor pentru analiză, mai ales dacă avem la dispoziție un set „heteroclit” de date. Astfel, dacă de exemplu ne interesează profilul cumpărătorului unui anumit „mix” de bunuri de consum în vederea optimizării aranjamentului mărfurilor într-un supermarket, trebuie selectate acele elemente (caracteristici) relevante (e.g. ocupația, venitul anual, sexul, vârstă, hobby-uri etc.) ale unui individ, ignorând alte elemente, precum status-ul marital sau starea de sănătate, de pildă, care nu par la prima vedere importante pentru scopul propus. Aceste tipuri de informații ce pot fi selectate dintr-o bază de date mai sunt cunoscute și ca *dimensiuni*, deoarece ele pot fi considerate ca dimensiuni ale profilului unui anumit individ, profil ce trebuie conturat cu ajutorul tehniciilor de DM, ținând cont de un anumit scop. În acest sens trebuie să subliniem faptul că unul dintre avantajele cele mai importante ale DM în raport cu alte metode este acela că, în principiu, nu trebuie să limităm arbitrar numărul de elemente pe care le analizăm, deoarece prin însăși natura sa DM are mijloace de a filtra informațiile. Evident că nu trebuie să utilizăm chiar toate informațiile

disponibile, atâtă timp cât o logică elementară ar putea exclude o parte din ele. Totuși, începătorii sau cei care abordează un domeniu complet necunoscut ar trebui să nu excludă nimic din ceea ce ar putea duce la descoperirea de cunoștințe utile.

- ★ *Problema eșantionării (sampling)* este cumva legată de cea anterioară și se referă la relevanța eșantionului ales, privită prin prisma atingerii scopului propus. Dacă ar fi vorba numai de latura statistică a procesului atunci lucrurile ar fi mult mai simple, aşa cum am arătat mai sus (vezi „Eșantionarea”), existând metode statistice clare de calcul al volumului eșantionului înănd cont de tipul de analiză ales. În cazul DM însă, prin natura particulară a procesului, regulile sunt mult mai laxe, deoarece scopul studiului este tocmai căutarea de informație utilă în mulțimi foarte mari de date, informație altfel greu dacă nu chiar imposibil de descoperit cu celelalte metode clasice. Totuși și în acest caz, pentru eficientizarea procesului (viteză mai mare/efort compuțațional mai mic), se poate construi modelul plecând de la un volum mai restrâns de date, obținute prin eșantionare, pentru ca apoi acesta să fie validat pe celelalte date.
- ◆ *Citirea datelor și construirea modelului.* După parcurgerea pașilor anterioari, am ajuns la momentul utilizării datelor disponibile pentru atingerea scopului propus. Primul lucru de făcut în acest moment este „citirea datelor” (*reading data*) din mulțimea de date existente. În esență, prin citirea datelor înțelegem accesarea datelor (e.g. extragerea (citirea) datelor dintr-un fișier text și introducerea lor într-o matrice unde liniile reprezintă cazurile iar coloanele reprezintă variabile, în scopul clusterizării lor (obținerea de cazuri similare –e.g. sinonime); citirea datelor dintr-un fișier Excel în vederea procesării lor cu un pachet de software statistic (e.g. SAS, Statistica, SPSS etc.) etc.). Este bine de știut că fiecare produs DM posedă un mecanism care poate „citi” datele. Odată datele citite, se trece la construirea modelului de DM. Orice model de DM va extrage din cantitatea de date disponibile diferenți indicatori utili în înțelegerea datelor respective (e.g. frecvențe ale anumitor valori, ponderi ale anumitor caracteristici, atribute corelate care explică (împreună și nu considerate separat) anumite comportamente etc.). Indiferent de modelul considerat, trebuie avute în vedere câteva caracteristici importante:
 - ★ *Acuratețea modelului* se referă la puterea modelului respectiv de a oferi date corecte, credibile, atunci când este utilizat în condiții reale. Vom discuta pe larg despre acest subiect pe parcursul cărții, aici doar subliniem faptul că acuratețea reală se măsoară pe date noi, necunoscute și nu pe datele de antrenament (*training data*), pe care modelul poate fi deosebit de „performant” (vezi cazul de *overfitting*).

- ★ *Inteligibilitatea modelului* se referă la caracteristica unui model de a fi ușor de înțeles de persoane diferite, cu diferite grade/tipuri de pregătire, începând cu modul cum se „leagă” input-urile modelului (datele introduse în „mașina” de minerit) cu output-urile sale (concluziile la care se ajunge) și terminând cu modul în care este prezentată acuratețea prognozei obținute. Cu toate că există modele „ermetice” (e.g. rețelele neuronale artificiale), care sunt similare unor „cutii negre” în care puțini știu ce se petrece, și terminând cu modelele „deschise” (e.g. modelele statistice regresive sau arborii de decizie), foarte „inteligibile”, este preferabil să construim și, mai ales să prezentăm un model DM de așa manieră încât să fie ușor de înțeles, chiar dacă nu în toate detaliile tehnice, de un eventual utilizator fără o pregătire de specialitate. Să nu uităm, în acest sens, că DM a fost creat și s-a dezvoltat aşa de puternic datorită cererii de pe piața business-ului, sănătății, comerțului etc., care nu cere o anumită pregătire specială a clientului potențial.
- ★ *Performanța* unui model DM este legată atât de viteza de construire a lui cât și de viteza de obținere a rezultatului prin aplicarea modelului. Referitor la acest ultim aspect, este importantă viteza de reacție a unui model (viteza de procesare) la aplicarea sa asupra unui volum mare de date (e.g. în cazul rețelelor neuronale probabiliste, viteza de procesare scade dramatic atunci când volumul datelor crește, deoarece ele utilizează în timpul predicției întreg „bagajul” de date de antrenament).
- ★ *Zgomotul* din date este un „dușman perfid” în construirea unui model eficient de DM, deoarece nu poate fi eliminat (filtrat) în întregime. Fiecare model are un anumit prag de toleranță la zgomot și tot în acest sens este deosebit de importantă și pre-procesarea inițială a datelor pentru obținerea unui model de DM care să nu fie „corupt”, în însăși esență sa, de zgomot.
- ◆ *Înțelegerea modelului* (vezi și „*inteligibilitatea modelului*”) se referă la momentul în care, după ce baza de date a fost „minerită” (studiată/analizată), un model DM a fost creat pe baza analizei acestor date, fiind gata să furnizeze informații utile despre ele. Pe scurt, elementele care trebuie avute în vedere în acest moment, indiferent de modelul ales, sunt următoarele:
 - ★ *Rezumarea modelului*, așa cum indică și numele, poate fi privită ca prezentarea concisă a sa sub forma unui raport succint și dens, accentuând cele mai importante informații (e.g. frecvențe, ponderi, corelații etc.) care explică rezultatele obținute din date (e.g. model pentru descrierea timpului de însănătoșire a pacienților [88], § 2.5).

- * *Informația specifică* pe care trebuie s-o producă un model se referă la acele elemente cauzale (input-uri) care sunt semnificative pentru un anume efect, spre deosebire de cele care nu sunt relevante. De exemplu, dacă avem ca scop identificarea tipului clienților unui supermarket care este probabil să viziteze des raionul de cosmetice, criteriul (input-ul) sex este deosebit de relevant (apare totdeauna în date (cu precădere –feminin) când este vorba de acest aspect), spre deosebire de ocupația profesională, care nu este relevantă (nu apare cu o anumită frecvență specifică în date pentru a fi luată în considerație). În concluzie, este foarte important să putem identifica acei factori care explică datele (prin prisma unui anumit scop) de cei care sunt irelevanți pentru analiza efectuată.
- * *Repartiția datelor*, la fel ca și în Statistică în procesul de eșantionare, este foarte importantă pentru soliditatea unei abordări DM. La fel ca acolo, trebuie să avem în primul rând un volum suficient de mare de date și, în al doilea rând, aceste date să fie reprezentative pentru analiză. Spre deosebire de Statistică, unde problema se pune să găsim o limită inferioară pentru volumul eșantionului prelevat în așa fel încât rezultatele să poată fi extrapolate cu o marjă suficientă de încredere la întreaga populație (*inferență statistică*), în acest caz se presupune că se „sapă” într-o cantitate apreciabilă de date. Cu toate acestea, trebuie să ne asigurăm că volumul de date este suficient de mare și de divers în structura sa pentru a putea fi relevant pentru o utilizare cât mai generală (e.g. profilul clientului de încredere pentru sistemul bancar ar trebui să fie destul de „acoperitor” pentru bănci în general, și nu pentru una în particular –dacă studiul nu a fost comandat de o anumită bancă, evident). În al doilea rând, așa cum am spus mai sus, datele trebuie să aibă o repartiție „corectă” pentru toate categoriile avute în vedere (e.g. dacă este interesant pentru analiză elementul „sex”, atunci cele două sexe trebuie reprezentate corect în date: un exemplu de repartiție corectă ar fi 51% vs. 49% (feminin vs. masculin) față de repartiția 98% vs. 2%, total neechilibrată).
- * *Diferențierea* se referă la proprietatea unei variabile predictive (input) de a produce diferențierea semnificativă între două rezultate (output-uri) ale modelului. De exemplu, dacă tinerilor le place să asculte atât muzică folk cât și rock, rezultă că această categorie de vârstă nu face diferențierea între cele două categorii de muzică. În schimb, dacă fetele ascultă cu placere folk (în raport de 20 la 1, să zicem), atunci sexul este important în diferențierea celor două genuri muzicale. După cum se vede, este foarte importantă identificarea acelor attribute ale datelor care creează diferențiere, mai ales în studiile dedicate construirii anumitor profiluri, e.g. studii de marketing.

- ★ *Validarea* reprezintă procesul de evaluare a acurateții predicției unui model. Validarea modelului se referă la producerea de predicții utilizând modelul existent și apoi compararea rezultatelor obținute cu rezultate deja cunoscute, reprezentând poate cel mai important pas în procesul de construire a unui model. Utilizarea unui model care nu se potrivește cu datele (*fitting data*) corect nu poate produce rezultate care să răspundă corespunzător scopului propus în studiu. În consecință, este de la sine înțeles că există o întreagă metodologie privind validarea unui model pe baza datelor existente (e.g. *holdout*, *random subsampling*, *cross-validation*, *stratified sampling*, *bootstrap* etc.). În fine, în procesul de înțelegere a modelului este important să identificăm factorii care conduc atât la obținerea ‚succeselor’ cât și a ‚eșecurilor’ în predicția produsă de model.
- ★ *Prognoza/predicția* (*prediction/forecast*) unui model se referă la capacitatea acestuia de a prognoza cel mai bun răspuns (output) - adică cel mai apropiat de realitate - pe baza datelor introduse (input). Cu cât diferența între ceea ce se așteaptă să se întâmple (*expected outcome*) și ceea ce se întâmplă de fapt (*observed outcome*) este mai mică, cu atât predicția este mai bună. Ca exemple clasice de predicții menționăm: prognoza vremii (e.g. pentru 24 sau 48 ore) obținută de un model DM meteorologic pe baza observațiilor complexe luate în considerație (satelit, aparatură specifică etc.) sau diagnosticul pentru o boală anume dat unui anumit pacient, pe baza datelor sale medicale. În ceea ce privește procesul de predicție, menționăm că unele modele furnizează, pe lângă prognoza respectivă, și modul cum aceasta a fost obținută, în timp ce altele furnizează doar rezultatul în sine, fără modul de obținere (fenomenul de ‚cutie neagră’ – *black box*). O altă chestiune legată de predicție se referă la predicțiile concurente celei mai bune. Deoarece nicio predicție nu este ‚infailibilă’, este necesar să cunoaștem, pe lângă cea mai probabilă, și competitoarele acesteia, în ordine ierarhică descrescătoare, tocmai pentru a avea un tablou complet al tuturor posibilităților. În acest context, dacă este posibil, este preferabil să putem să și care este diferența între prima predicție și a doua clasată. Este clar că, cu cât diferența între primele două clasate este mai mare, cu atât avem mai puține dubii cu privire la cea mai bună alegere. Vom încheia aceste scurte rânduri privind predicția unui model (DM), subliniind faptul că anumite domenii ca: fiabilitatea software (*software reliability*), dezastre naturale (e.g. cutremure, inundații, alunecări de teren etc.), pandemii, demografie - dinamica populației, meteorologia etc., sunt cunoscute ca având mari dificultăți în procesul de prognoză.

1.4. Probleme rezolvabile cu Data Mining

Nucleul procesului de Data Mining este constituit din *construirea* unui anumit model care să *reprezinte* setul de date care este „minerit” pentru rezolvarea anumitor probleme concrete din viața reală. Vom trece succint în revistă unele dintre cele mai importante probleme care necesită aplicarea de metode DM, metode care stau la baza construirii unui model DM.

În principiu, atunci când utilizăm metode DM pentru rezolvarea unor probleme concrete avem în vedere tipologia lor, care poate fi rezumată sintetic în două mari categorii, deja menționate la obiectivele DM:

- *Metode predictive*, care utilizează o parte dintre variabilele existente pentru a prognoza valorile ulterioare, necunoscute, ale altor variabile (e.g. clasificarea, regresia, detectarea deviațiilor etc.);
- *Metode descriptive*, care descoperă pattern-uri în date, ușor interpretabile de către utilizator (e.g. clustering, reguli de asociere, pattern-uri secvențiale etc.).

Vom prezenta pe scurt câteva probleme cu care se confruntă domeniul Data Mining și modul cum pot fi rezolvate acestea, pentru a ilustra într-o manieră cât mai sugestivă aria de aplicabilitate a acestiei.

1.4.1. Clasificarea

Există ideea larg răspândită că mintea omenească își organizează cunoștințele utilizând în mod natural procesul de clasificare. Dar atunci când vorbim de clasificare, vorbim de taxonomie. *Taxonomia* (gr. *tassein* = a clasifica + *nomos* = știință, lege) a apărut mai întâi ca știința clasificării organismelor vii (*taxonomia alfa*), dar apoi s-a dezvoltat ca știința clasificării în general, inclusiv aici și principiile clasificării (schemele taxonomice). Astfel, *clasificarea* (taxonomică) reprezintă procesul de a plasa un anumit obiect (concept) într-o mulțime de categorii, pe baza proprietăților obiectului (conceptului) respectiv. Menționăm în acest sens, cu titlu de pionierat, lucrarea de referință «Fisher R. A. (1936) - *The use of multiple measurements in taxonomic problems* - Annals of Eugenics, 7, Part II, pp.179-188», în care apare și celebra clasificare privind planta *Iris*, deja ajunsă clasică în domeniu. Clasificarea modernă își are originile în activitatea botanistului, zoologului și doctorului suedez Carl von Linné (Carolus Linnaeus) - sec. XVIII, care a clasificat speciile pe baza caracteristicilor lor fizice și care este considerat „tatăl taxonomiei moderne”.

Procesul de clasificare se bazează pe patru componente fundamentale:

- ▶ *Clasa* (variabila *dependentă* a modelului), care este o variabilă categorială, reprezentând „eticheta” sub care va fi cunoscută data clasificată. Exemple

de asemenea clase sunt: prezența infarctului miocardic, client fidel, clasa unei stele (galaxii), clasa unui cutremur (uragan) etc.

- ▶ *Predictori* (variabilele *independente* ale modelului), reprezentați de caracteristicile datelor ce urmează a fi clasificate, pe baza cărora se face clasificarea. Exemple de asemenea predictori sunt: fumatul, consumul de alcool, tensiunea arterială, frecvența cumpărăturilor, status-ul marital, caracteristicile imaginilor (prin satelit), înregistrări geologice specifice, viteza și direcția vântului, sezonul, locația producării fenomenului etc.
- ▶ *Mulțimea de antrenament (învățare)* – *training data set* – care este reprezentată de setul de date care conține valori pentru cele două componente anterioare, și care este utilizată pentru „antrenarea” modelului ca să recunoască clasa corespunzătoare pe baza predictorilor disponibili. Exemple de astfel de mulțimi sunt: loturi de pacienți testați referitor la infarctul miocardic, grupuri de clienți ai unui supermarket (investigați prin sondaje de opinie interne), baze de imagini de urmărire telescopică (e.g. Observatorul Astronomic Mountain Palomar (Caltech), California, SUA, <http://www.astro.caltech.edu/palomar/>), baze de date privind uraganele (e.g. centre de colectare de date și prognoză de tipul National Hurricane Center, SUA, <http://www.nhc.noaa.gov/>), baze de date privind cercetarea cutremurelor (e.g. centre de colectare de date și de prognoză a cutremurelor de tipul celui al Institutului Român de Seismologie Aplicată, <http://www.fotonsas.8m.com/index.html>).
- ▶ *Mulțimea de testare – testing data set* - conține date noi, care vor trebui clasificate de modelul (clasificatorul) construit anterior și astfel se poate evalua acuratețea clasificării – performanța modelului.

Terminologia unui proces de clasificare conține următorii termeni:

- Setul de *înregistrări/tuple/vectori/instanțe/obiecte/eșantioane* care formează mulțimea de antrenament;
- Fiecare înregistrare/tuplu/vector/instanță/obiect/eșantion conține un set de *attribute* (i.e. componente/caracteristici) dintre care una este *clasa*;
- *Modelul de clasificare (clasificatorul)* care, din punct de vedere matematic, reprezintă o funcție ale cărei variabile (argumente) sunt valorile atributelor (predictive/independente), iar valoarea sa este clasa corespunzătoare;
- Setul pentru *test*, conținând date de aceeași natură cu mulțimea de antrenament și pe care se testează acuratețea modelului.

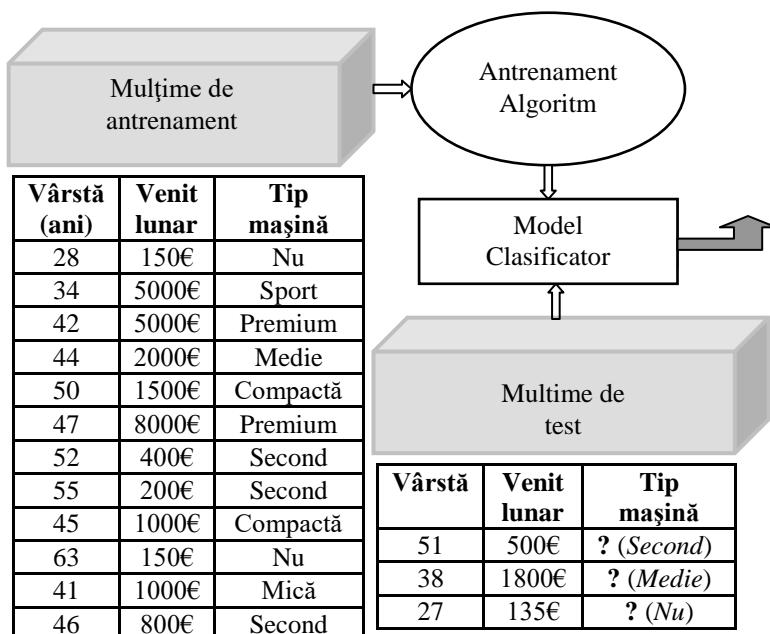
Reamintim că în domeniul învățării automate (*machine learning*), învățarea supervizată (*supervised learning*) reprezintă acea tehnică utilizată pentru crearea unei funcții plecând de la mulțimea datelor de antrenament. Scopul

învățării supervizate este acela de a prognoza valoarea (*output-ul*) funcției pentru orice obiect/eșantion/instanță de intrare (*input*) nou, după parcurgerea procesului de antrenament. Tehnica clasificării, ca metodă predictivă, este un astfel de exemplu de tehnică de învățare automată supervizată, presupunând existența câte unui grup de instanțe etichetate pentru fiecare categorie de obiecte.

Recapitulând, un proces de clasificare se caracterizează prin:

- ⇒ *Input*: un set de antrenament conținând instanțe cu atrbute, dintre care unul reprezintă eticheta clasei;
- ⇒ *Output*: un model (*clasificator*) care desemnează pentru fiecare instanță o anumită etichetă (clasifică instanța într-o anumită categorie), pe baza celorlalte atrbute;
- ⇒ Clasificatorul este utilizat pentru a prognoza clasa unor instanțe noi, necunoscute. O mulțime de testare este, de asemenea, utilizată pentru determinarea acurateții modelului.

Vom ilustra mai jos, grafic, etapele construirii unui model de clasificare a tipului de automobil pe care îl pot cumpăra diferite categorii de persoane. Este ceea ce s-ar numi construirea profilului cumpărătorului de automobile.



Recapitulând, observăm din desenul de mai sus că în prima fază construim modelul de clasificare (prin intermediul algoritmului respectiv), antrenând modelul pe mulțimea de antrenament. Practic, în această fază modelul ales își ajustează parametrii, plecând de la corespondența dintre inputurile date (vârstă și venit lunar) și output-urile corespunzătoare cunoscute (tip de mașină). Odată funcția de clasificare identificată, verificăm acuratețea clasificării utilizând mulțimea de testare, prin compararea output-ului așteptat (prognozat) cu cel observat, pentru a valida sau nu modelul (rata acurateții = % elementelor din mulțimea de testare clasificate corect).

Odată construit un model de clasificare, el va trebui comparat cu altele pentru a putea alege pe cel mai bun. În ceea ce privește compararea clasificatorilor (modelele de clasificare), vom enumera mai jos câteva elemente esențiale, de care trebuie să se țină cont.

- *Acuratețea prognozei (predictive accuracy)*, care se referă la abilitatea modelului de a clasifica corect o dată nouă, necunoscută;
- *Viteza (speed)*, care se referă la rapiditatea cu care poate modelul să proceseze datele;
- *Robustețea (robustness)*, care se referă la abilitatea modelului de a face predicții corecte chiar și în prezența „zgomotului” în datele procesate;
- *Scalabilitatea (scalability)*, care se referă la abilitatea modelului de a procesa atât cantități mari de date cât și de a procesa date din domenii suficient de diferite;
- *Interpretabilitatea (interpretability)*, care se referă la caracteristica modelului de a fi cu ușurință înțeles, interpretabil;
- *Simplicitatea (simplicity)*, care se referă abilitatea modelului de a nu fi prea complicat, în ciuda eficienței sale (e.g. mărimea unui arbore de clasificare/decizie; compactitatea regulilor etc.). În principiu, se alege cel mai simplu model care poate rezolva în mod eficient o anumită problemă - la fel ca și în matematică, unde demonstrația cea mai elegantă este cea mai simplă.

Printre cele mai cunoscute modele (metode) folosite pentru clasificare putem menționa următoarele, cu toate că acestea sunt utilizate, evident, și în alte scopuri:

- Arbori de clasificare/decizie (*decision/classification trees*);
- Clasificatori bayesieni (*Bayesian classifiers/Naïve Bayes*);
- Rețele neuronale (*neural networks*);
- Analiza statistică (*statistical analysis*);
- Algoritmi genetici (*genetic algorithms*);
- Mulțimi rough (*rough sets*);

- Clasificator de tip *k*-nearest neighbor (*k-nearest neighbor classifier*);
- Clasificatori bazați pe reguli (*rule-based methods*);
- Raționament pe bază de memorie (*memory based reasoning*);
- Mașini cu suport vectorial (*SVM - support vector machines*).

În ceea ce privește aria aplicabilității clasificării, credem că o scurtă trecere în revistă a celor mai cunoscute aplicații va fi mai mult decât sugestivă.

- ◆ *Identificarea profilului* cumpărătorului pentru un anumit produs (sau pentru un complex de bunuri). Scopul aplicării umui astfel de model de clasificare rezidă în optimizarea aprovizionării cu anumite produse și gestionarea mai bună a stocului. De exemplu, se dorește construirea profilului standard al unui cumpărător de mașini de spălat. Pentru aceasta se studiază datele disponibile referitoare la acest aspect (e.g. cele mai bine vândute tipuri de mașini, modul de achiziționare (cash/credit), venitul mediu lunar, durata de folosință a unui asemenea bun, tipul de locuință (bloc/casă) relativ la posibilitatea de uscare a rufelor), situația familială (căsătorit sau nu, total persoane în familie, copii mici etc.), ocupația, timp disponibil pentru treburile casnice etc.). La toate aceste informații = input, se adaugă și variabila categorială reprezentând categoria de cumpărător (i.e. va cumpăra/nu va cumpăra). După ce aceste date/informații au fost culese, ele se utilizează în faza de „învățare” (antrenare) a modelului ales, păstrând eventual o parte din ele ca set de testare pentru a fi utilizate în faza de validare a modelului (dacă nu dispunem de alte date noi pentru aceasta).



Remarcă: O extensie a modelelor de marketing privind clasificarea clienților este cea care are ca scop crearea unui profil al „coșului” de bunuri pe care le achiziționează un anumit tip de client, în vederea aranjării mărfurilor făcând parte din acest „coș” în rafturi vecine. În acest mod se ajunge la o optimizare a vânzării mărfurilor, clientul fiind „atras” ca pe lângă marfa pe care voia să cumpere de fapt să mai

cumpere și alte mărfuri ,adiacente'. De exemplu, se pot organiza mărfurile care au comun o anumită utilizare (e.g. raftul în care se vând ciocane este vecin cu cel în care se vând clești și cu cel în care se vând cuie; raftul cu deodorante este alăturat celui cu săpunuri și cu cel cu geluri de baie etc.). Aici însă intervin și alte modele DM încă din afară de cele de clasificare (e.g. clustering, descoperirea regulilor de asociere etc.).

- ◆ *Detectarea fraudelor* (e.g. în utilizarea cardurilor bancare) se folosește pentru a evita pe cât mai mult posibil utilizarea frauduloasă a cardurilor bancare în tranzacțiile comerciale. Pentru aceasta se colectează informațiile disponibile privind modul de utilizare a cardurilor de către diferiți clienți (e.g. ce cumpărături se fac uzuale cu cardul, cât de des, unde se folosește de obicei etc.). Se adaugă și ,etichetele' privind modul de utilizare (fraudulos, corect), pentru a întregii tabloul setului de date de antrenament. După antrenarea pentru a ,învăța' să deosebească cele două tipuri de utilizatori, se trece la validarea modelului și apoi aplicarea sa. Astfel, banca emitentă poate urmări corectitudinea tranzacțiilor cu ajutorul cardurilor prin urmărirea evoluției unui anumit cont. Ca un fapt divers, mulți comercianți, mai ales cei mici, evită vânzările prin card, preferând cash-ul, tocmai de spaimă de a fi escrocați. Mai nou, există încercări de a construi modele de clasificare (corect/fraudulos), plecând de la modul în care se tastează PIN-ul cardului respectiv, identificând după analizarea timpilor între două ,bătăi' consecutive a tastelor dacă posesorul cardului este legitim sau nu. S-a observat astfel că există în acest caz o similaritate cu mașina de ,detectat' minciuni – o persoană necinstită are o altă reacție la tastarea PIN-ului.



- ◆ *Clasificarea galaxiilor*. În anul 1926 celebrul astronom american Edwin Hubble (1889-1953) a început o dificilă lucrare de clasificare a galaxiilor. Inițial, a considerat ca principalele atribute ale acestora

culoarea și mărimea, dar ulterior a decis că cea mai importantă caracteristică este forma lor (*morfologia galaxiilor*). Astfel a început această disciplină astronomică de catalogarea a galaxiilor (e.g. galaxii eliptice, galaxii spiralate, galaxii spiralate barate –vezi fotografiile de mai jos).



1.4.2. Clustering

Prin *clustering*, termen ce vine de la cuvântul englezesc *cluster* = îngrămadire, fascicul, înțelegem metoda de a diviza (segmenta) un set de date (înregistrări/tuple/vectori/instanțe/obiecte/eșantioane) în mai multe grupuri (*clustere*), pe baza unei anumite similarități prestabilite. Să amintim că ideea de partiționare a diferitelor obiecte în grupuri, pe baza similarității lor, a apărut pentru prima dată la Aristotel și Teophrastos prin sec. IV î.H., dar metodologia științifică și termenul de „analiză cluster” (*cluster analysis*) a apărut prima dată, se pare, în «C. Tryon (1939), *Cluster Analysis*, Ann Arbor, MI: Edwards Brothers». Putem deci considera metoda de clustering ca o „clasificare” a unor obiecte similare în submulțimi ale căror elemente au anumite caracteristici comune (se mai spune că partiționăm/segmentăm o mulțime de obiecte în submulțimi de elemente similare în raport cu un anumit criteriu prestabilit). Să menționăm aici câțiva termeni sinonimi clustering-ului: *clasificare automată*, *taxonomie numerică*, *botriologie*, *analiză tipologică* etc. Nu trebuie confundat procesul de clustering cu procesul de clasificare, descris în paragraful precedent. Astfel, dacă la clasificare avem de a face cu o acțiune asupra unui obiect care primește astfel o „etichetă” de apartenență la o anumită clasă, în cazul clustering-ului acțiunea are loc asupra întregului set de obiecte care este partiționat în subgrupuri bine definite. Exemple de clustere sunt foarte ușor de observat în viața reală: într-un supermarket diferențele tipuri de produse asemănătoare sunt dispuse în raioane distințe (e.g. brânzeturi, produse din carne, electrocasnice etc.), persoanele care se strâng în grupuri (clustere) la o reuniune pe baza afinităților comune, împărțirea animalelor sau plantelor în specii bine definite etc.

În principiu, fiind dată o mulțime de obiecte/instanțe, fiecare dintre ele fiind caracterizat de un set de atrbute, și având la dispoziție o măsură de similaritate, se pune problema de a le împărți în grupuri (clustere) astfel încât:

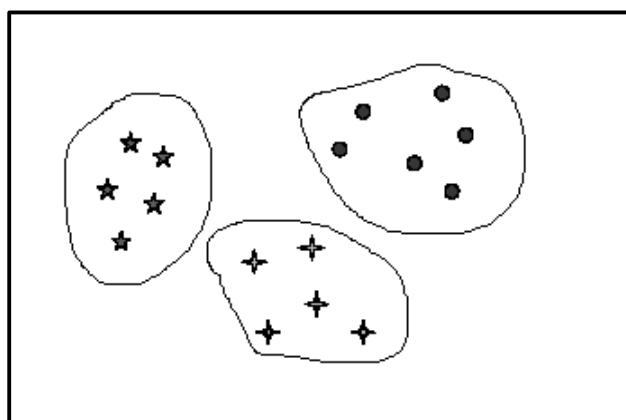
- Obiectele aparținând unui cluster sunt mai asemănătoare unul altuia;
- Obiectele aparținând la clustere diferite sunt mai puțin asemănătoare unul altuia.

Procesul de clustering va fi unul de succes dacă atât similaritatea intra-cluster cât și disimilaritatea inter-clustere sunt maxime.

Pentru a investiga similaritatea între două obiecte, se utilizează măsurile de similaritate, alese în funcție de natura datelor și scopul propus. Prezentăm mai jos, cu titlu informativ, câteva dintre cele mai cunoscute astfel de măsuri:

- * Distanța *Minkowski* (e.g. *Manhattan* (city block/taxicab), *Euclidiană*, *Chebychev*);
- * Măsura *Tanimoto*;
- * Măsura *Pearson's r*;
- * Măsura *Mahalanobis*;
- * Măsuri *fuzzy*.

Grafic, procesul de clustering este ilustrat în figura de mai jos.



În ceea ce privește aria aplicațiilor clustering-ului, vom face o scurtă trecere în revistă a unor exemple sugestive.

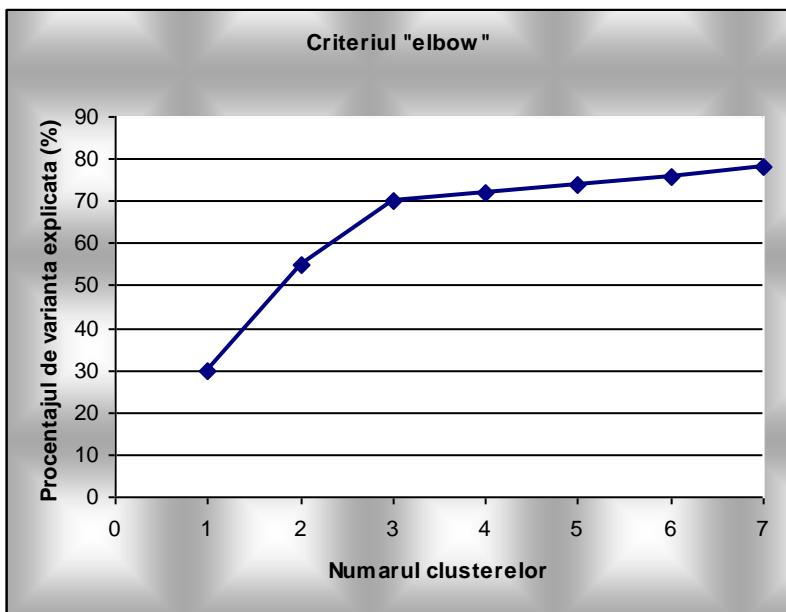
- ◆ *Segmentarea pieței (market segmentation)*, care are ca scop divizarea clienților în grupuri (clustere) distincte, pe baza similarității în ceea ce privește cumpărăturile făcute uzuale. Odată stabilite aceste grupuri, ele vor fi considerate ca grupuri „țintă” pentru oferte distincte de bunuri sau servicii.
- ◆ *Clasificarea documentelor (document clustering)*, care are ca scop împărțirea documentelor în grupuri bine definite, pe baza similarității lor, stabilită utilizând de obicei frecvența cu care apar anumiți termeni

de bază, definițorii (e.g. sport, financiar, politic etc.). Vom sublinia importanța unui asemenea demers pentru motoarele de căutare pe Internet a unor documente electronice pe o anumită tematică, sau organizarea de *e*-biblioteci sau *e*-lărbării.

- ◆ ,*Clasificarea* ' bolilor, tratamentelor sau simptomelor în medicină.
- ◆ ,*Clasificarea* ' diferitelor artefacte în arheologie în funcție de o anumită perioadă, cultură, tehnică de execuție etc.
- ◆ În Biologie, procesul de clustering are două aplicații importante (cf. Wikipedia) în domeniile biologiei computaționale și bioinformaticii:
 - În *transcriptomică (transcriptomics)* clusteringul este utilizat la construirea de grupuri de gene care au expresia pattern-urilor asemănătoare –vezi *genomica*.
 - În *analiza secvențelor (sequence analysis)* clusteringul este utilizat în procesul de grupare a secvențelor omoloage în familii de gene, aspect foarte important în biologia evolutivă.
- ◆ ,*Clasificarea* ' companiilor la Bursă pe baza analizei fluctuației acțiunilor lor (creștere/scădere –UP/DOWN). Exemplul ilustrat mai jos se referă la procesul de clustering pe baza datelor de piață Standard&Poor's (S&P500 Index). Ideea de bază este de a considera similare companiile care au evoluții asemănătoare de-a lungul unei anumite perioade de timp –în cazul acestui exemplu 1994-1996. Se verifică cu acest prilej un fapt deja bine-cunoscut pe piața financiară - comportamentul asemănător al companiilor dintr-un anumit domeniu (e.g. hi-tech, bancar etc.), care, în principiu, cresc sau scad în grup.

	<i>Discovered Clusters</i>	<i>Industry Group</i>
1	Applied-Matt-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconductor-DOWN,Oracle-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN,EMC-Corp-DOWN,Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal -UP, Schlumberger-UP	Oil-UP
5	Barrick-Gold-UP,Echo-Bay-Miner-UP, Homestake-Mining-UP,Newmont-Mining-UP, Placer-Dome-Inc-UP	Gold-UP
6	Alcan-Aluminum-DOWN,Asarco-Inc-DOWN, Cyrus-Amax-Min-DOWN,Inland-Steel-Inc-Down, Inco-LTD-DOWN,Nucor-Corp-DOWN,Praxair-Inc-DOWN, Reynolds-Metals-DOWN,Stone-Container-DOWN, USX-US-Steel-DOWN	Metal -DOWN

În încheiere, să amintim o chestiune deosebit de importantă în procesul de clustering - alegerea numărului optim de clustere (grupuri) de obiecte. Pentru rezolvarea acestei probleme se utilizează uzual *criteriul cotitură* (*elbow criterion*) care, în principiu, spune că vom alege acel număr de clustere astfel încât, dacă s-ar mai adăuga încă unul, nu s-ar mai obține suficientă informație în plus. Practic, se utilizează analiza varianței (dispersiei) pentru a „măsura” cât de bine a fost realizată segmentarea datelor, în sensul obținerii unei variabilități/dispersii mici intra-cluster și o variabilitate/dispersie mare inter-clustere în funcție de numărul de clustere ales. Figura de mai jos - graficul procentajului dispersiei explicată de clustere în funcție de numărul de clustere - ilustrează acest fapt. Se observă că, de la un număr de clustere mai mare ca trei, informația câștigată crește nesemnificativ, curba având un „cot” (elbow) în punctul 3, care va fi ales în acest caz ca număr optim de clustere.

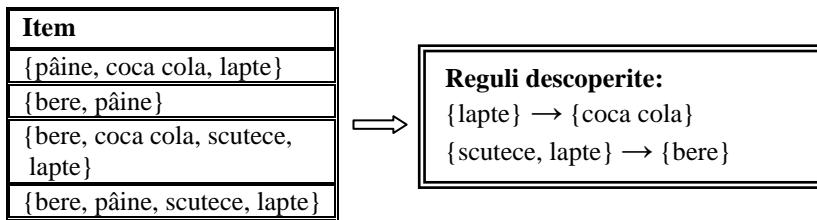


Printre alte criterii de alegere a numărului optim de clustere, menționăm BIC (*Schwarz Bayesian Criterion*) și AIC (*Akaike Information Criterion*).

1.4.3. Descoperirea regulilor de asociere

În principiu, prin *descoperirea regulilor de asociere* (*association rule discovery/association rule learner*) înțelegem procesul de identificare a regulilor de dependență dintre diferite grupuri de fenomene. Astfel, să presupunem că avem o colecție de mulțimi ce conțin fiecare un anumit număr

de obiecte/item-uri. Ne propunem să descoperim acele reguli ce asociază aceste obiecte, astfel încât pe baza acestor reguli să se poată prognoza apariția unui item/obiect, ținând cont de apariția altora. Pentru înțelegerea acestui proces, vom face apel la celebrul exemplu al asocierii <bere – scutec> (beer – diaper), bazat pe urmărirea comportamentului cumpărătorilor dintr-un supermarket. Vom prezenta mai jos, într-un mod sugestiv, acest exemplu din care reiese un fapt surprinzător – persoanele care cumpără bere sunt înclinate să cumpere și scutece și invers – de unde și numele exemplului [162].



Domeniul aplicațiilor metodei descoperirii regulilor de asociere este vast, aici vom face doar o scurtă trecere în revistă a unor exemple sugestive.

- ◆ Managementul raioanelor/rafturilor într-un supermarket, adică mai direct, modul de aranjare a raioanelor/rafturilor cu marfă astfel încât, pe baza procesării datelor privitoare la modul în care își fac cumpărăturile clienții, mărfuri care se cumpără de obicei împreună să fie așezate pe rafturi vecine (vândute în raioane vecine). Acest lucru se realizează pe baza datelor obținute de la *Cassa* de plată – utilizând codurile de bară citite de scannere pentru fiecare client în parte. Din baza de date astfel construită, în care apar mărfurile care au fost cumpărate cu aceeași ocazie, se pot descoperi regulile de asociere dintre ele. La fel ca mai sus, se poate obține o regulă care asociază, de exemplu, laptele cu coca cola, astfel încât raionul/raftul cu lactate să se găsească lângă cel cu sucuri.
- ◆ Web mining, care are ca punct de plecare modul de căutare pe Web a diferitelor produse, servicii, companii etc. Acest fapt ajută companiile care tranzacționează mărfuri online să-și gestioneze eficient pagina Web, pe baza URL-urilor accesate de clienți la o singură vizită pe server. Astfel, utilizând regulile de asociere, se poate ajunge la concluzia că, de exemplu, 35% dintre clienții care au accesat odată pagina Web cu URL-ul: /company name/products/product_1 /html au accesat și pagina Web cu URL-ul: /company name/products/product_3/html; 45% dintre clienții care au accesat pagina Web: /company name/announcements/special-offer.html au accesat în aceeași sesiune și pagina Web: /company name /products/product_3/html etc.

- ◆ Managementul echipamentelor și sculelor necesare intervențiilor de tip *service* la domiciliul clientului sau de urgență pentru o firmă de reparații și întreținere (e.g. auto-service în trafic pe autostradă, reparații sanitare la domiciliu etc.). Ideea este de a echipa mașinile de intervenție cu acele echipamente și dispozitive care sunt utilizate frecvent la diferite tipuri de intervenții, astfel încât, atunci când apare o cerere nouă pentru o anumită intervenție, autoutilitara de intervenție să fie corespunzător echipată, economisind timp și carburant necesare curselor inutile de reconfigurare. În acest caz, regulile de asociere se identifică prin procesarea datelor referitoare la tipul de dispozitive și piese de schimb utilizate în intervențiile anterioare pentru soluționarea diferitelor probleme apărute. Menționăm că o situație asemănătoare poate fi identificată și în cazul intervențiilor medicale de urgență, problema aici fiind echiparea ambulanțelor astfel încât să poată fi asigurat serviciul de prim-ajutor cu eficiență maximă și la timp.

1.4.4. Descoperirea pattern-urilor secvențiale

În multe aplicații ca, de exemplu: biologia computațională (e.g. secvențele de ADN sau de proteine), accesul Web (e.g. traseele de navigare a paginilor Web – secvențe de pagini Web accesate), analiza conectărilor (logărilor) pentru utilizarea unui sistem (e.g. logarea la diferite portaluri, webmail, etc.), datele sunt în mod natural sub forma unor secvențe. Sintetic vorbind, problema care se pune în acest context este ca, dată fiind o secvență de evenimente discrete (având constrângeri temporale) de forma: A B A C D A C E B A B C..., prin procesarea acestora să se descopere pattern-urile care se repetă frecvent. (e.g. A urmat de B, A urmat de C etc.). Fiind date secvențe de forma: „Timp#1 {Temp = 28°C} → Timp#2 {Hum = 67%; Pres = 756mm}”, formate din item-uri (atribut/valoare) și/sau seturi de item-uri, trebuie descoperite pattern-uri, apariția evenimentelor în aceste pattern-uri fiind guvernată de restricții temporale. Enumerăm câteva cazuri de utilizare a tehniciilor de descoperire a pattern-urilor secvențiale în viața reală:

- ◆ Un exemplu semnificativ pentru comerțul actual de tip <<direct cu clientul/comerț cu amănuntul>> (*retail*) îl reprezintă analiza bazelor mari de date în care sunt înregistrate date secvențiale privind diverse tranzacții comerciale (e.g. id-ul clientului - în cazul utilizării cardurilor pentru plată, data la care s-a efectuat tranzacția, mărfurile tranzacționate - utilizând tehnologia codurilor de bară etc.), cu scopul eficientizării vânzării.
- ◆ În medicină, în cazul diagnosticării unei boli, se analizează înregistrări de simptome în timp real, pentru a descoperi în ele pattern-uri secvențiale, semnificative pentru boala ca, de exemplu: „primele trei

zile de dureri de cap și tuse, urmate de alte două zile de febră cu valori ale temperaturii de 38-39 grade C etc.”.

- ◆ Meteorologie – descoperirea pattern-urilor în schimbarea globală a climei (vezi procesul de încălzire globală), în general sau, în particular, pentru descoperirea momentului producerii de uragane, tsunami etc.

1.4.5. Regresia

Analiza regresiei (regresia) ca, de altfel și corelația, își au originea în lucrările celebrului genetician Sir Francis Galton (1822-1911), care a lansat spre sfârșitul sec. al XIX-lea noțiunea de „regresie spre medie” (*regression towards the mean*) – principiul conform căruia, fiind date două măsurători dependente, valoarea estimată pentru a doua măsurătoare este mai apropiată de medie decât de valoarea observată a primei măsurătorii (e.g. tații mai înalți au copii mai scunzi, iar tații mai scunzi au copii mai înalți –înălțimea copiilor regrezează către medie).

În Statistică, prin *analiza regresiei* se înțelege modelul care stabilește (în principiu, prin ecuația de regresie) legătura între o mărime (variabilă) „răspuns” –variabila dependentă, și alte mărimi (variabile) „predictoare” – variabile independente. Cel mai cunoscut exemplu de regresie este poate cel al stabilirii relației între înălțimea și greutatea unei persoane pe baza unor rezultate din aplicarea ecuației de regresie, obținându-se astfel valoarea greutății ideale corespunzătoare unei anumite înălțimi date. Analiza regresiei se referă, în principiu, la:

- Stabilirea existenței unor relații de mărime/efect între mai multe variabile;
- Prognoza valorilor unei variabile în funcție de valorile altor variabile (determinarea efectului „predictorilor” asupra „răspunsului”).

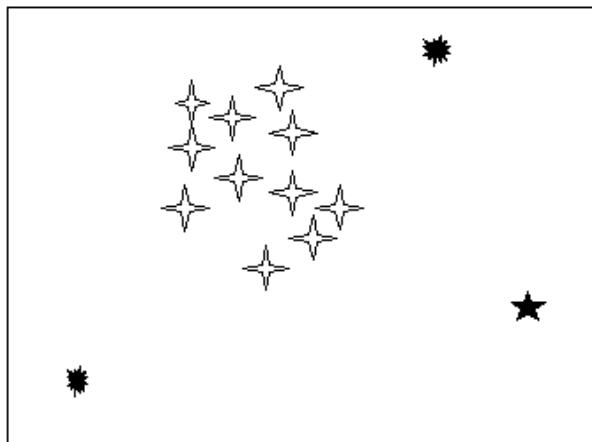
Aplicațiile acestei metode statistice în DM sunt multiple, aici limitându-ne a semnala următoarele:

- ◆ Comerț: prognoza vânzărilor unui nou produs pe baza cheltuielilor cu publicitatea;
- ◆ Meteorologie: prognoza vitezei și direcției vântului pe baza valorilor temperaturii, presiunii atmosferice, umidității etc.
- ◆ Marketing bursier: prognoze de tip serii temporale privind evoluția indicilor bursierii (parte a estimării tendințelor – *trend estimation*).

1.4.6. Detectarea deviațiilor/anomalialilor

Detectarea deviațiilor/anomalialilor (*outliers/anomalies*), așa cum arată și numele, se ocupă cu descoperirea deviațiilor semnificative de la

„comportamentul” normal (anomalii). Figura de mai jos ilustrează sugestiv existența anomalilor în date.



1.5. Despre modelare și modele

În cele două paragrafe precedente, ocupându-ne de modul în care se procesează datele în DM, am scos în relief câteva aspecte definitorii ale principalelor tehnici utilizate în modelele DM, precum și a problemelor uzuale rezolvate cu aceste metode. În acest paragraf vom face câteva considerații cu caracter mai general privind procesul de modelare și modelele, cu scopul declarat de a ilustra complexitatea unei asemenea demers, dar și fascinația exercitată asupra cercetătorului. În principiu, vom trece în revistă succint principalele aspecte privind procesul de construire a unui model, împreună cu problemele și soluțiile aferente acestui proces complex, aspecte ce se particularizează în mod specific în cazul Data Mining.

La orice vîrstă, începând cu anii senini ai copilăriei și terminând cu anii dificili ai senectății, și în orice circumstanțe ne-am afla, avem nevoie de modele. Suntem aproape în permanență nevoiți să modelăm anumite fenomene ca, de exemplu, diferite aspecte economice personale (e.g. construirea unui buget familial cât mai ajustat la realitatea înconjurătoare), lucrări la locul de muncă (e.g. programe economice, proiectări ale diferitelor modele: arhitectură, design industrial, automobile, „minerit” în date pentru descoperirea de patternuri utile, sisteme și rețele de calculatoare, cercetări în domeniul medico-farmaceutic, meteorologic etc.), astfel încât, pe de-o parte să le cunoaștem mai bine caracteristicile intime și, pe de altă parte să putem folosi aceste cunoștințe pentru a progresă în domeniul respectiv.

Este mai mult decât evident că aproape în toate cazurile, fenomenele reale abordate, care reprezintă prototipurile pentru modelele noastre, sunt fie

inabordabile direct (e.g. studiul mișcării uraganelor, modelarea evoluțiilor stelelor sau galaxiilor), fie prea complicate privite fiind în totalitatea lor (e.g. analiza mișcării unei insecte în vederea creării unor roboți industriali pe baze analogice), fie prea periculoase (e.g. modelarea unor procese ce țin de temperaturi înalte, medii toxice etc.). Este atunci preferabil și mult mai economic să le studiem caracteristicile folosind machete = modele și simulări ale “realului”, privite ca înlocuitori mai mult sau mai puțin fideli ai „prototipului” original.

Este de aceea natural ca, din cele arătate mai sus, Matematica și Informatica să aibă rolul hotărător în tehniciile modelării, indiferent din ce domeniu face parte prototipul care trebuie modelat (economie, industrie, medicină, sociologie, biologie, meteorologie etc.). În acest context se folosesc diferite noțiuni matematice pentru reprezentarea componentelor constitutive ale fenomenului ce trebuie modelat, ca apoi, cu ajutorul diferitelor ecuații, să poată fi reprezentate interconexiunile dintre aceste componente.

După ce partea “asamblării” prin ecuații a tuturor elementelor caracterizând componentele modelului a fost terminată, urmează ultima parte, aceea a implementării modelului matematic prin construirea și rularea programelor aferente lui. Se studiază apoi “output-urile”, modificându-se continuu parametrii modelului, până la obținerea acurateții dorite în imitarea realității de către modelul propus – simularea computerizată.

Din experiența acumulată în domeniul modelării, s-a ajuns la concluzia că orice întreprindere serioasă în acest domeniu trebuie obligatoriu să parcurgă următoarele etape:

- * *Identificarea.* Reprezintă primul pas în vederea găsirii modelului adecvat unei anumite situații concrete. În principiu nu există un drum bătătorit în acest sens, existând nenumărate posibilități în identificarea modelului celui mai bun. Totuși, putem indica două abordări extreme ale problemei, care pot fi cu ușurință mixate mai apoi. Este vorba, mai întâi, de *abordarea conceptuală*, care privește alegerea modelului din punct de vedere abstract, rational, pe baza unor cunoștințe și informații *apriorice* privind situația analizată și fără a ține seama de datele concrete ale prototipului. În identificarea conceptuală datele se ignoră, cel ce modelează se bazează pe idei, concepte, experiență sa în domeniu și multe, foarte multe referințe bibliografice. Procesul de modelare depinde de situația respectivă, variind de la o problemă la alta, de multe ori identificarea făcându-se natural, pe baza modelelor clasice din domeniul respectiv. Chiar dacă nu se găsește deja un model gata construit, care cu mici modificări să poată fi folosit, pe baza unor extrapolări și mixări multiple, este deseori probabil să se obțină o identificare corectă. În al doilea rând, este vorba de *identificarea empirică*, în care sunt considerate doar datele și legăturile dintre ele, fără a se face vreo referire la sensul lor sau modul cum au

rezultat. Astfel, ignorând voit orice model *aprioric*, ne întrebăm doar ce vor să ne “spună” datele. Se observă cu ușurință că aceasta este situația, în principiu, în ceea ce privește ,mineritul’ în date. Este într-adevăr foarte dificil să putem întrevedea vreo schemă doar din “citirea” datelor, fiind nevoie de multă experiență în prelucrarea lor, dar împreună cu cealaltă metodă, primele rudimente ale modelului căutat nu vor întârzia să apară. În final tragem concluzia evidentă că o identificare corectă a modelului are nevoie de o “fină” combinare a celor două metode.

- * *Estimarea și ajustarea.* După ce am depășit primul pas, acela al identificării modelului adekvat prototipului dat, urmează “personalizarea” acestuia cu datele numerice necesare obținerii unui model concret. Acum nu ne mai sunt de folos doar parametrii abstracți, desemnați doar prin litere, ci trebuie introduse datele concrete. Această etapă a trecerii de la forma generală a modelului ales la forma numerică concretă se numește *ajustarea modelului*. Procedeul prin care parametrilor modelului li se atribuie valori numerice se numește *estimare*.
- * *Testarea.* Această noțiune, despre care am vorbit în prealabil, termenul ales fiind sugestiv prin sine însuși, înseamnă de fapt considerarea valorii practice a modelului propus, eficiența sa probată pe alte date decât cele pe care a fost construit. Testarea reprezintă ultimul pas înaintea „lansării modelului pe piață” și reprezintă poate cea mai importantă etapă a procesului construirii unui model. De modul cum va ,răspunde’ modelul la aplicarea sa pe date noi, necunoscute, depinde dacă va primi Ok-ul de utilizare practică sau nu.
- * *Aplicarea practică.* Nu trebuie uitat că obiectivul oricărei modelări este reprezentat de găsirea modelului adekvat rezolvării unei anumite probleme reale. Deci nu găsirea de modele în sine este importantă aici, cu toate că și această întreprindere are importanță și farmecul ei, ci găsirea modelelor “naturale”, care să se “plieze” cât mai fidel pe un anumit prototip dat. Este o activitate într-adevăr fascinantă și extraordinară, care are propria sa istorie legată de diferitele ramuri ale științei, ca: matematică, fizica, biologia, știința materială, economia, psihologia, medicina etc., modelele găsindu-și cele mai variii aplicații.
- * *Iterația.* Atunci când construiește un anumit mecanism (fizic), constructorul are în vedere un anumit plan foarte riguros, care trebuie îndeplinit punct cu punct și în ordinea strictă a punctelor din program - evident aici nu e vorba de inventatorii care intră în aceeași sferă cu creatorii de modele și care toți au sansa unui

program mai “liberal” în concepție. Cu toate că în cele de mai sus am prezentat într-o ordine relativă etapele unei modelări generice, această ordine nu trebuie totuși considerată ca “literă de lege”. Modelarea presupune întoarceri frecvente la etapele anterioare, modificări ale modelului, descoperirea unor aspecte care inițial au fost ignorate, dar care la o gândire mai profundă sunt esențiale etc. Această repetare a etapelor, această regândire permanentă a modelului se numește iterație în cadrul procesului de modelare. În concluzie, primul model nu este MODELUL, unicul și cel de pe urmă, ci este doar începutul unui sir de iterări ale pașilor enunțați mai sus, având ca unic scop găsirea celui mai adecvat model pentru o anumită situație dată.

Atunci când ne propunem să modelăm un anumit fenomen, situație etc., este natural să ne interesăm asupra referințelor privind domeniul respectiv, în măsura posibilităților, pentru a obține informația necesară. Problema rezidă în modul cum putem procura informația necesară, precum și criteriile după care decidem ce e important și ce nu, ce se potrivește mai bine situației date. În cele ce urmează, vom trece în revistă câteva din cele mai importante forme de informații preliminare folosite în modelare.

- *Informații despre variabile.* În momentul conceperii unui model avem în minte foarte multe variabile care, într-un fel sau altul, ar putea intra în componența modelului. Alegerea variabilelor care începătorește sunt esențiale pentru model este cea mai importantă și în același timp delicată problemă, pentru că neglijarea unei variabile importante este mai “periculoasă” decât includerea uneia lipsită de importanță. În acest stadiu al modelării, cercetătorul trebuie să facă apel la specialistul în domeniul în care vrea să modeleze, ca să-i stabilească clar ierarhia variabilelor constitutive. Ideal ar fi să se lucreze, în echipă la modelare, pentru ca în orice moment să se aleagă varianta optimă. Imediat după alegerea variabilelor constitutive ale modelului, trebuie să se identifice domeniile în care aceste variabile iau valori (‘constrângerile’ impuse variabilelor), e.g. variabila X este întreagă și negativă, Y este continuă și pozitivă etc. Este de asemenea important să stabilim și relațiile între variabile, e.g. $X < Y$.
- *Informații despre date.* În primul rând este necesar ca modelul ales să fie potrivit volumului de date disponibile. Sunt modele mai mult sau mai puțin sofisticate, e.g. modele de prognoză meteo, în care este nevoie de un număr suficient de mare de date pentru ca modelul teoretic să poată fi ajustat datelor (*fitting model to data*). În al doilea rând, se pune problema analizei separate a datelor (*dezagregare*), sau analizarea împreună a lor (*agregare*), acest lucru depinzând de fiecare caz în parte, în principiu lucrându-se simultan cu cele două

tipuri de analiză. În al treilea rând, trebuie luate în considerație date de încredere (*reliable data*), altfel modelul propus nu are nicio valoare practică. În acest sens, există o întreagă literatură privind modul de colectare a datelor în funcție de domeniul considerat și de obiectivele propuse. În fine, baza de date trebuie să fie suficient de bogată pentru că pentru corectări ulterioare ale modelului este nevoie de noi date. Acest lucru este, prin natura faptelor, adevărat în Data Mining, deoarece aici se presupune *a priori* că se lucrează cu baze „urișe” de date. Trebuie amintit că există și metode de a depăși impedimentul unei baze de date relativ restrânse, atunci când nu există posibilitatea lărgirii sale, e.g. *randomizarea*, care presupune, în principiu, folosirea acelorași date reordonate în mod aleatoriu.

- *Informații despre erori și variabilitate.* În afară de informațiile de mai sus privind variabilele și datele legate de modelul în studiu, sunt necesare informații privind sursele de eroare ce pot intra în procesul de modelare, în achiziția datelor, precum și variabilitatea aleatoare ce trebuie modelată.
- *Informații despre modele.* În primul rând să observăm că atunci când suntem puși în fața problemei alegerii unui model adecvat pentru fenomenul analizat, ne întâlnim cu următoarea situație: pe de-o parte există o mulțime impresionantă de modele gata construite și disponibile prin intermediul diferitelor cărți sau reviste de specialitate, iar pe de altă parte, atunci când încercăm să aplicăm vreunul din ele datelor reale, ne lovim de faptul că nu se potrivește în mod satisfăcător, ducând la o aproximare insuficient de precisă. În al doilea rând, trebuie să alegem modele care să se potrivească datelor (discrete sau continue, categoriale sau numerice, acoperind uniform întregul domeniu de definiție al modelului sau acumulându-se numai în jurul unor anumite valori etc.). În acest sens, mai ales pentru cei cu o experiență limitată în domeniu sau chiar novici, trebuie o mare atenție privind natura datelor, pentru că există programe dedicate care nu au incluse avertismente privind natura și volumul datelor introduse, rezultatul obținut fiind astfel compromis.
- *Informații despre parametrii modelului.* Cunoscând natura intimă a fenomenului modelat, se pot face ipoteze și cu privire la parametrii modelului propus. Dacă, de exemplu, avem un proces ce leagă variabila dependentă Y de variabila explicativă X printr-o relație liniară de parametru b , e.g. $Y = bX + \varepsilon$ (i.e. regresie liniară simplă) și ele au aceeași tendință de evoluție (*trend*), atunci vom alege parametrul $b > 0$ de la început. În general, atunci când un model este construit pentru a fi folosit de nespecialiști, de exemplu un model privind un anumit tratament medical folosit de doctori

fără o pregătire în domeniul respectiv, trebuie pe cât e posibil să aibă o interpretare practică clar expusă pentru a fi înțeles bine de utilizator. Acest lucru este valabil atât pentru variabilele modelului cât și pentru parametrii săi.

- *Informații despre domeniul de aplicație a modelului - criterii de apreciere.* Este de domeniul evidenței faptul că un model nu se proiectează în sensul de a face ‘artă pentru artă’, ci pentru a fi folosit într-un context practic. De aceea, atunci când ne propunem construirea unui model, trebuie să avem informații clare privind domeniul de aplicabilitate. Aceste informații sunt folositoare atât în alegerea modelului, cât și în modul cum acesta se va ajusta datelor, precum și în ceea ce privește validarea sa. Ne interesează apoi criteriile după care va fi judecat modelul, dacă este considerat ca un tot unitar, aplicat unei singure situații, sau este parte a unui proces mai complet de modelare și atunci ne interesează și compatibilitatea cu celelalte componente ale modelului global. În fine, ne interesează dacă se aplică la o anumită situație într-un anumit context particular, sau este vorba de a aplicare regulată la o situație stabilă în timp. Trebuie menționată aici și caracteristica de scalabilitate a modelului, privită în sensul de arie de situații distințe la care se poate aplica.

O problemă deloc simplă în procesul de modelare este chiar maniera de alegere a modelului. Problema se complică atunci când, cel mai adesea, avem de ales între două sau mai multe modele. În această situație apar două posibilități:

1. *Alegerea modelului dintr-o clasă de modele.* Este situația în care, pe baza anumitor cunoștințe (experiențe) anterioare și a analizei datelor, alegem un anumit tip de model. În acest caz problema se reduce la a folosi tehniciile generale, corespunzătoare clasei de modele aleasă;
2. *Alegerea liberă a modelului.* Este cazul în care selecția tipurilor posibile se face din clase diferite de modele, la liberul arbitru al cercetătorului.

Indiferent de procedura de alegere a unui model dintr-o mulțime oarecare, va trebui ca modelele vizate să fie comparate pe baza unui set de criterii ca, de exemplu:

- Măsurarea erorilor în etapa estimării parametrilor modelului (*training*);
- Măsurarea erorilor în etapa testării/validării modelului (*testing/validation*);
- Diagnostice reziduale și teste de concordanță (goodness-of-fit tests);

- Considerații calitative.

În funcție de rezultatul comparației, vom decide care este modelul optim prototipului dat. În continuare, vom prezenta pe scurt câteva considerații generale privind maniera de alegere a unui model.

- *Modele ierarhice.* În acest caz este vorba de a alege modelul dintr-o clasă bine precizată de modele, astfel că fiecare model reprezintă un caz particular al unei clase generale. În principiu este vorba de două abordări extreme în alegerea unui model dintr-o clasă. Pe de-o parte se poate proceda ‘de sus în jos’, adică se va considera mai întâi modelul general, numit uneori și *modelul saturat* și, prin simplificare, ținând cont de contextul concret, se ajunge la modelul căutat. În ciuda multor dificultăți de calcul, legate de complexitatea modelului general, dacă acesta a fost bine ales, modelul propus va fi cu siguranță adecvat situației date. Pe de altă parte, se poate merge și invers în identificarea modelului în interiorul unei clase și anume ‘de jos în sus’. Ținând cont de principiul simplității, se pornește cu cea mai simplă variantă, cea care pune accentul pe caracteristica de bază a fenomenului analizat, și pe măsură ce necesitatea o cere, începe să se crească gradul de complexitate. Indiferent de metoda aleasă, procesul se oprește atunci când reducerea sau creșterea modelului nu mai influențează semnificativ concordanța sa cu datele concrete.
- *Modele libere.* Spre deosebire de cazul modelelor ierarhice, a căror structură bine definită permite compararea mai multor modele din aceeași clasă pentru alegerea finală a celui mai bun, în cazul modelelor libere, datorită lipsei de structură a mulțimii din care trebuie să alegem, alegerea se face, în principiu, între două variante. Folosind anumite tehnici formale, se poate măsura calitatea unui model privind adecvarea la date în vederea găsirii celui mai potrivit.

O procedură interesantă de alegere a modelelor este cea bayesiană. Metoda bayesiană de alegere a celui mai potrivit model se bazează pe probabilitatea condiționată de alegere a unui model în raport cu datele de care dispunem și se bazează pe celebra formulă a lui Thomas Bayes (1763):

$$P\{A|D\}P\{D\} = P\{D|A\}P\{A\},$$

care stabilește legătura între probabilitatea condiționată directă și cea inversă. Astfel, să presupunem că evenimentul A reprezintă alegerea fie a modelului M_1 , fie a modelului M_2 . Să notăm:

$$P\{A := M_1\} = P\{M_1\}, P\{A := M_2\} = P\{M_2\},$$

pe care le considerăm *probabilități anterioare* ale celor două modele și, fiind cunoscut fiecare model, putem calcula probabilitățile condiționate $P\{D|M_i\}$, $i = 1, 2$, i.e. probabilitățile ca datele, reprezentate aici prin D , să fie în concordanță cu opțiunea făcută în ceea ce privește modelul (*fitting data to model*). Ceea ce ne interesează însă este cum putem alege modelul pe baza datelor, deci probabilitatea condiționată inversă (*fitting model to data*). Folosind formula lui Bayes, avem:

$$P\{M_1|D\} = \frac{P\{D|M_1\}P\{M_1\}}{P\{D\}},$$

$$P\{M_2|D\} = \frac{P\{D|M_2\}P\{M_2\}}{P\{D\}},$$

numite și *probabilități posterioare*. Putem calcula aceste probabilități condiționate știind că (*formula probabilității totale*):

$$P\{D\} = P\{D|M_1\}P\{M_1\} + P\{D|M_2\}P\{M_2\}.$$

Vom alege astfel modelul pentru care obținem cea mai mare probabilitate posterioară.

Generalizând, să presupunem că avem la dispoziție un set complet de modele M_i , $i = 1, 2, \dots, k$, din care putem alege unul, cunoscând probabilitățile anterioare corespunzătoare:

$$P\{M_i\}, \quad \sum_{i=1}^k P\{M_i\} = 1.$$

Să presupunem că am simbolizat datele observate (disponibile) prin D . Atunci, folosind formula lui Bayes, obținem probabilitățile posterioare ale fiecărui model știind setul de date D , prin intermediul probabilităților:

$$P\{M_i|D\} = \frac{P\{D|M_i\}P\{M_i\}}{P\{D\}}, \quad i = 1, 2, \dots, k,$$

unde:

$$P\{D\} = \sum_{i=1}^k P\{D|M_i\}P\{M_i\}.$$

Astfel, vom alege modelul pentru care obținem cea mai mare probabilitate posterioară (vezi [57]).

În final, este bine ca atunci când alegem un model să nu uităm vorbele lui A. Einstein „*Toate modelele trebuie să fie cât mai simple posibil, dar nu mai simple decât este necesar*”, cu alte cuvinte conceptual K.I.S.S. (*Keep It Simple Series*): „dintre două modele care sunt comparabile ca performanțe, va fi ales cel mai simplu, care probabil este cel mai aproape de adevăr și este mai ușor acceptat de către ceilalți”.

Să presupunem că am ales deja un model care pare a fi corespunzător prototipului (i.e. problemei reale). Rămâne acum să ajustăm modelul la datele observate. Vom menționa trei criterii, cunoscute sub numele de *criterii de ajustare*, care stau la baza măsurării gradului de ajustare a modelului la date și pe baza cărora vom lua în considerație diferite metode de ajustare.

- *Stabilirea egalității între caracteristicile formei modelului și a caracteristicilor formei datelor;*
- *Măsurarea deviației dintre model și date;*
- *Stabilirea măsurii în care modelul este justificat de date.*

Odată modelul ajustat la date, mai rămâne ca acesta să fie și validat, înainte ca el să fie aplicat cu succes la rezolvarea unor probleme din viața reală. Notiunea de validare a unui anumit model acoperă o arie destul de largă de aspecte ce trebuie avute în vedere. Astfel, prin termenul de *validare* vom înțelege considerarea valorii practice a modelului propus în explicarea unui anumit fenomen dat, măsurând asemănarea sa cu prototipul modelat și eficiența sa în aplicarea directă la o situație dată. Vom vorbi de validarea modelului atunci când:

- (a) în dezvoltarea modelului, modelul ajustat relevă aspecte noi privind datele;
- (b) în etapa de testare, noi date sunt strânse și folosite pentru optimizarea modelului propus;
- (c) în faza de aplicare practică sunt introduse proceduri de “monitorizare” pentru a verifica dacă modelul propus inițial este eficient sau nu în condiții “live”.

Fără a insista asupra fiecărui aspect în parte, vom enumera elementele avute în vedere în decursul procesului de validare a unui model:

- *Analiza prototipului*, efectuată acum *post festum* pentru a releva aspecte noi privind compatibilitatea dintre prototip, modelul

propus și datele inițiale sau culese ulterior procesului de identificare;

- *Analiza aplicațiilor*, efectuată pentru a releva măsura în care modelul propus este eficient în rezolvarea problemelor practice pentru care a fost construit;
- *Analiza formei modelului*, privită ca o reconfirmare a alegerii inițiale din etapa de identificare;
- *Analiza comportării modelului* în momentul comparării cu prototipul, relativ la datele existente;
- *Analiza sensibilității modelului* relativ la modificările efectuate asupra datelor. În acest sens, modelul trebuie să ‘răspundă’ la anumite modificări ale parametrilor situației modelate;

În final, vom menționa faptul că și în cazul validării unui model procedurile tehnice din etapa de identificare se păstrează. Astfel, vom vorbi de validarea conceptuală a unui model, de validarea empirică a formei și a parametrilor modelului și, în final, de validarea eclectică a modelului. Orice proces de validare va fi încheiat în spiritul *“finis coronat opus”*, adică de validare a sa în ‘stare de funcționare reală’, în analizarea eficienței sale atunci când se aplică uneia dintre situațiile reale pentru care a fost proiectat sau, eventual, altora noi.

Ultima etapă și cea mai importantă de altfel în construirea unui model este cea a aplicării sale în practică. Este clar că atunci când se pornește la proiectarea și construirea unui model se are în vedere finalitatea unei asemenea lucrări, care rezidă în rezolvarea unei probleme practice cu ajutorul modelului respectiv. Problema ce trebuie rezolvată va conduce la alegerea mijloacelor potrivite pentru construirea modelului, astfel încât acesta să poată fi ulterior aplicat în rezolvarea problemei inițiale sau a altor probleme asemănătoare. În cele ce urmează vom prezenta o serie de aspecte legate de raportul aplicabilitate/modelare, pentru a pune în evidență legătura dintre ele.

Aplicații de tip descriptiv.

Întotdeauna când plecăm la construirea unui model, avem în vedere faptul că el reprezintă o descriere mai mult sau mai puțin fidelă a prototipului, deci a situației sau fenomenului real pe care dorim să-l modelăm. Înținând cont de acest fapt, mai întâi avem nevoie de o descriere acceptabilă a celor mai importante caracteristici ale prototipului pe care trebuie să le includă și ‘copia’ sa, adică modelul. În acest stadiu, modelul poate fi folosit din punct de vedere aplicativ doar ca o simplă descriere rezumativă, globală, a situației reale. În acest context, sunt o serie de modele care ‘rămân’ în acest stadiu primar de *modele descriptiv*e. În cazurile mai complexe (e.g. modelele dinamice)

descrierea este mult mai detaliată, cuprinzând descrierea în parte a diferitelor componente ale prototipului, precum și descrierea tuturor inter-relațiilor dintre componente. Este deci atât o descriere statică a componentelor, cât și una dinamică, a relațiilor dintre componente. Modelele mai complexe sunt deci folosite în aplicații nu doar pentru o simplă descriere ilustrativă a unui anumit fenomen, ci pentru verificarea și validarea sau invalidarea anumitor teorii propuse pentru explicarea sau rezolvarea unor chestiuni importante. Un model bine construit în raport cu datele unei anumite teorii poate duce la acceptarea sau respingerea ei, fiind, în ultima instanță, singurul mijloc de validare a teoriei (e.g. modele de verificare a teoriei relativității). și acest tip de utilizare a unui model intră tot în clasa aplicațiilor descriptive.

Aplicații de tip exploratoriu.

Atunci când studiem o anumită situație reală, așa-numitul prototip, una dintre cele mai importante întrebări care se poate pune este: „ce se întâmplă dacă se schimbă ceva în ceea ce privește datele prototipului, deci ce influență au anumite schimbări în funcționarea acestuia?” Problema care apare aici este aceea că nu putem verifica ce se întâmplă folosind chiar prototipul, deoarece ori este prea complicat, ori este imposibil, ori chiar îl putem deteriora. Pentru aceasta se poate folosi un model pe care putem exersa diferite modificări pentru a observa ce schimbări apar și astfel a extrapola cunoștințele obținute în cazul prototipului. Este ceea ce se numește o *aplicație exploratorie* (*What-if analysis*).

Aplicații de tip predictiv.

Prin aplicație de tip *predictiv* înțelegem acea aplicație a modelului în care se încearcă ‘prezicerea’ valorii pe care poate să o ia o anumită variabilă, ținând cont de ceea ce știm în prezent. Sunt multe exemple de astfel de aplicații predictive, una clasice fiind predicția apariției unui uragan de o anumită clasă în următorul interval de timp cu ajutorul unui model din meteorologie. Alte modele se aplică în cadrul teoriei așteptării prin predicția numărului de clienți la ‘rând’, a timpului mediu de așteptare, a încărcării unui server, a tendinței de creștere sau scădere a prețului acțiunilor la bursă, în medicină în predicția evoluției unei anumite boli sub un anumit tratament etc.

Aplicații decizionale.

În fine, un model construit pe baza unui anumit prototip poate fi folosit și la luarea unor anumite decizii. În acest context trebuie să facem următoarea precizare: având la dispoziție un anumit model, indiferent pentru ce scop a fost construit el, îl putem folosi pentru luarea anumitor decizii. De exemplu, în cazul unui model meteorologic este evident că vom folosi predicția apariției unui uragan pentru a lua anumite decizii importante la nivelul comunității afectate – să ne amintim numai consecințele uraganului Katrina

(gradul final 5) din anul 2005 de pe coastele Golfului Mexic, cu consecințele sale dramatice. Să punem însă în evidență și cazul în care un model se construiește exclusiv în scop decizional. De exemplu, atunci când construim un model de așteptare privind serviciile asigurate de o anumită companie, el va fi folosit cu precădere de staff-ul companiei respective pentru luarea deciziilor privind modul în care se execută service-ul (numărul de servere necesar și gradul lor de folosire, timpul mediu de așteptare etc.). În cazul aplicării modelelor decizionale în medicină, să amintim de programele dedicate diagnosticării, create exclusiv pentru asistarea medicului în alegerea celui mai bun diagnostic, implicit a celui mai bun tratament.

În final, să trecem în revistă câteva probleme privind riscul separării modelului de aplicație, care pot să apară atunci când se neglijeză relația dintre procesul de modelare, respectiv modelul propus, și aplicarea modelului la condițiile reale:

- ✓ *Forma modelului neadecvată aplicăției considerate.* Este vorba fie de gradul de detaliere a modelului în raport cu cerințele aplicăției, fie de incompatibilitatea dintre ipotezele cerute de forma modelului și condițiile reale, fie de prezentarea inadecvată a modelului în fața utilizatorului, ducând la aplicății improprii etc.
- ✓ *Estimarea parametrilor neadecvată aplicăției practice.* Este vorba de metodologia aleasă de stabilire a criteriilor privind estimarea parametrilor modelului și situația concretă, mai ales în ceea ce privește stabilirea erorilor estimărilor;
- ✓ *Supraevaluarea modelului* este un pericol de tip mai mult psihologic și constă în considerația că nu modelul trebuie să se supună realității ci invers. Nu trebuie uitat niciodată că un model este un model și nimic mai mult care, dacă este bine construit, ‘seamănă’ suficient de bine cu prototipul, ajutând la o mai bună înțelegere a acestuia, deci nu trebuie să facem din el un “tabu”. Nu trebuie uitată aserțiunea celebrului statistician G.E.P. Box cum că „*Statisticienii, ca și artiștii, au prostul obicei să se îndrăgostească de modelele lor*”, lucru valabil pentru orice cercetător, de altfel.

Revenind la domeniul Data Mining, procesul modelării poate fi rezumat foarte succint de următoarele trei puncte:

- ▶ Explorarea datelor (prepararea datelor, alegerea predictorilor, analize exploratorii, determinarea naturii și/sau complexității modelelor ce vor fi alese etc.);
- ▶ Construirea modelului și testarea/validarea sa (alegerea celui mai bun model pe baza performanței predictive – evaluarea competitivității modelelor);

► Aplicarea modelului în practică.

Referitor la implementarea diferitelor modele DM, prezentăm mai jos o listă cu diverse sisteme software, bazate pe acestea:

- Sisteme analitice orientate-subiect (*Subject-oriented analytical systems*):
 - MetaStock (Equis International)
 - SuperCharts (Omega Research)
 - Candlestick Forecaster (IPTC)
 - Wall Street Money (Market Arts)
- Pachete statistice (*Statistical packages*):
 - SAS (SAS Institute)
 - SPSS (SPSS)
 - Statgraphics (Statistical Graphics)
 - Statistica (Statsoft)
- Rețele neuronale (*Neural networks*):
 - Statistica Neural Network (Statsoft)
 - BrainMaker (CSS)
 - NeuroShell (Ward Systems Group)
 - OWL (Hyperlogic)
 - PolyAnalyst (Megaputer Intelligence)
- Programare evolutivă (*Evolutionary programming*):
 - PolyAnalyst (Megaputer Intelligence)
 - SWEEP-Re (Orbital research Inc.)
 - PEP -Plain Evolutionary Programming (Wayne State's ENCORE)
 - OPTIONS (CEDC-University of Southampton)
- Raționament bazat pe memorie (*Memory based reasoning*):
 - PolyAnalyst (Megaputer Intelligence)
 - KATE tools (Acknosoft)
 - Pattern Recognition Workbench (Unica)
- Arbori de decizie (*Decision trees*):
 - C5.0 (Rule Quest)
 - Clementine (Integral Solutions)
 - SIPINA (University of Lyon)
 - IDIS (Information Discovery)
 - C4.5 (CSE)

- Statistica (Statsoft)
- SPSS (SPSS)
- Algoritmi genetici (*Genetic algorithms*):
 - PolyAnalyst (Megaputer Intelligence)
 - GeneHunter (Ward Systems Group)
 - OPTIONS (CEDC-University of Southampton)
 - Genetics Studio (Silk Digital)
 - Jaga - Java Genetic Algorithm Package (GAUL)
- Metode regressive neliniare (*Nonlinear regression methods*):
 - PolyAnalyst (Megaputer Intelligence)
 - NeuroShell (Ward Systems Group)
 - Statistica (Statsoft)
 - SPSS (SPSS)

(mai multe amănunte în cadrul paragrafelor dedicate diferitelor modele și tehnici DM).

În paragraful anterior am amintit câteva dintre cele mai cunoscute modele de Data Mining precum și unele aplicații aferente acestora. Așa cum bine s-a observat, există un câmp foarte larg atât în ceea ce privește modelele/tehnicele considerate cât și problemele care pot fi rezolvate cu metodologia Data Mining. Ceea ce este incitant la Data Mining este tocmai această deschidere, destul de rar întâlnită la alte domenii de cercetare, în ceea ce privește atât aria de aplicabilitate cât și domeniul de tehnici utilizate.

1.6. Aplicații de Data Mining

În rândurile de mai sus au fost descrise, pentru fiecare metodă în parte, și diverse aplicații de succes pentru tehnicele de DM. În continuare, vom reaminti succint câteva domenii de mare interes pentru aplicarea tehnicilor DM.

- ▶ Domeniul bancar-financial este unul dintre primele și cele mai importante arii pentru aplicațiile DM. Astfel, în domeniul bancar, metodele DM au fost utilizate (și sunt utilizate cu succes) intensiv în:
 - modelarea și prognoza fraudelor în creditare;
 - evaluarea riscurilor;
 - realizarea analizelor de tendință (*trend analysis*);
 - analiza profitabilității;

- suport pentru campanii de marketing directe.

În domeniul financiar, aplicații DM găsim în:

- prognoza prețului acțiunilor (*stock price forecasting*);
- opțiune de comerț (*trading option*);
- evaluarea certificatelor de valoare (*bond rating*)
- managementul portofoliilor (*portfolio management*);
- prognoza prețului mărfurilor;
- fuzionarea și/sau achiziționarea firmelor;
- prognoza dezastrelor financiare etc.

- ▶ Domeniul de comerț cu amănuntul (*retail*) – cazul lanțurilor de magazine, evident – a profitat din plin de ajutorul tehniciilor DM:
 - data warehousing;
 - campanii de vânzare directă prin poștă (*direct mail campaign*);
 - segmentarea clienților;
 - evaluarea prețului unor produse specifice (antichități, mașini uzate, artă etc.).
- ▶ Domeniul sănătății este, de asemenea, una dintre primele arii de activitate foarte importante care au impulsionat dezvoltarea intensivă a DM, plecând de la tehnici de vizualizare, prognoza costurilor serviciilor medicale și terminând cu diagnosticul asistat de computer.
- ▶ Domeniul telecomunicațiilor, mai ales în ultimii ani, a profitat din plin de accesul la tehnologia DM. Din cauza concurenței acerbe pe care o cunoaște actualmente acest domeniu, probleme de identificare a profilului clienților, de crearea și menținerea fidelității lor, de vinderea de noi produse sunt vitale pentru companiile din domeniu. Câteva probleme care pot fi rezolvate prin tehnici DM în acest domeniu sunt:
 - prognoza fraudelor în telefonia mobilă;
 - crearea profilului clientului fidel/profitabil;
 - identificarea factorilor care influențează comportamentul clienților privind tipul de convorbiri telefonice;
 - identificarea riscurilor privind noi investiții în tehnologii de vârf (e.g. fibre optice, sateliți etc.)
 - identificarea diferențelor privind produsele și serviciile între firme concurente.

În ceea ce privește companiile care vând produse DM, lista lor este foarte mare, aici prezentând doar un mic „eșantion” din ea (companie/produs), aşa cum poate fi găsită la o simplă căutare pe Web.

*American Heuristics (Profiler) http://www.heuristics.com	*Integral Solutions (Clementine) http://www.isl.co.uk/index.html
*Angoss software (Knowledge Seeker) http://www.angoss.com	*IBM (Intelligent Data Miner) http://www.ibm.com/Stories/1997/04/data1.html
*Attar Software (XpertRule Profiler) http://www.attar.com	*Lucent Technologies (Interactive Data Visualization) http://www.lucent.com
*Business Objects (BusinessMiner) http://www.businessobjects.com	*NCR (Knowledge Discovery Benchmark) http://www.ncr.com
*DataMind (DataMind Professional) http://www.datamind.com	*NeoVista Sloutions (Decision Series) http://www.neovista.com
*HNC Software (DataMarksman, Falcon) http://www.hncs.com	*Nestor (Prism) http://www.nestor.com
*HyperParallel (Discovery) http://www.hyperparallel.com	*Pilot Software (Pilot Discovery Server) http://www.pilotsw.com
*Information Discovery Inc. (Information Discovery System) http://www.datamining.com	*Seagate Software Systems (Holos 5.0) http://www.holossys.com
	*SPSS (SPSS) http://www.spss.com
	*Thinking Machines (Darwin) http://www.think.com

Deoarece în ultimii ani am fost confruntați cu o creștere exponențială a produselor DM, a apărut necesitatea creării de standarde privind aceste produse. În acest sens putem menționa grupul DMG (Data Mining Group), ca grup independent de companii dezvoltatoare de standarde DM, grup cuprinzând companiile (<http://www.dmg.org/index.html>):

A. Membri DMG plini:

- IBM Corp., Somers, NY, USA
- KXEN, San Francisco, CA, USA
- Microsoft , Redmond, WA, USA
- MicroStrategy Inc., McLean, VA
- National Center for Data Mining, University of Illinois at Chicago, USA
- Open Data, River Forest, IL, USA
- Oracle Corporation, Redwood Shores, CA, USA
- prudsys AG, Chemnitz, Germania
- Salford Systems, San Diego, CA, USA

- SAS Inc., Cary, NC, USA
- SPSS Inc., Chicago, IL, USA
- StatSoft, Inc, Tulsa, OK, USA

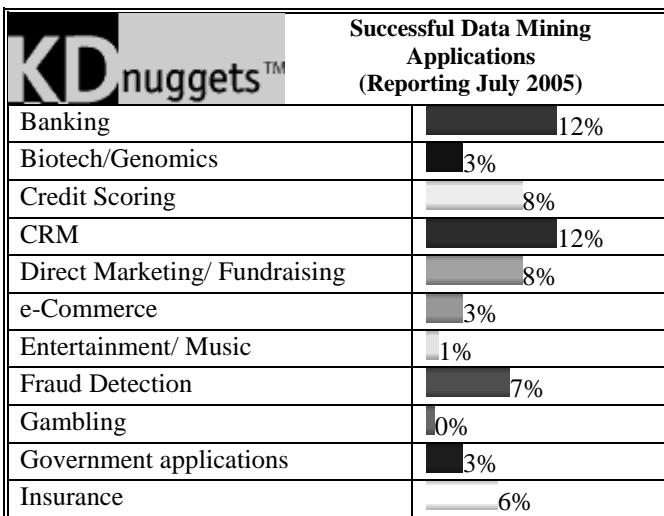
B. *Membri DMG asociați:*

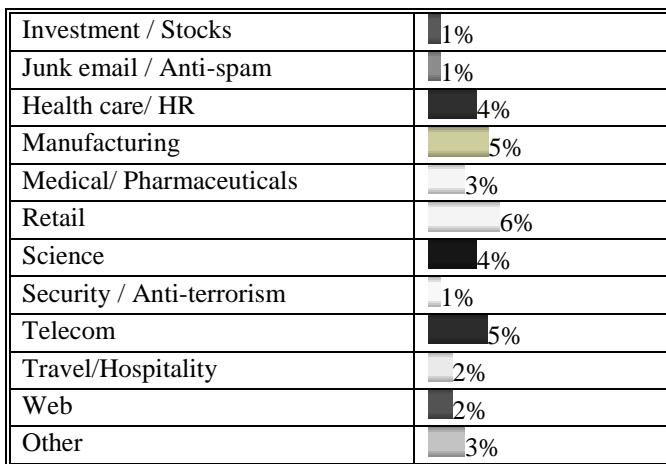
- Angoss Software Corp., Toronto, Canada
- Insightful Corp., Seattle, WA, USA
- Magnify Inc., Chicago, IL, USA
- NCR Corp., Dayton, OH, USA
- Quadstone, Boston, MA, USA
- Urban Science, Detroit, MI, USA
- SAP, Walldorf, Germania

Tot în acest sens vom menționa că în ultimii ani au avut loc conferințe internaționale având ca temă tocmai standardizarea procedurilor Data Mining. Ca exemple de asemenea manifestări recente, putem menționa:

- KDD-2004 ACM -Workshop on Data Mining Standards, Services and Platforms (DM-SSP 04), Seattle, WA, 2004;
- KDD-2005 ACM -Data Mining Standards, Services and Platforms (DM-SSP 05), Chicago, Illinois, 2005.

Tabelul de mai jos (http://www.kdnuggets.com/polls/2005/successful_data_mining_applications.htm) este sugestiv în ceea ce privește aria de răspândire și procentajul utilizării aplicațiilor DM în ultimii trei ani.





1.7. Terminologie Data Mining

În domeniul Data Mining există deja concepte și o terminologie fundamentale, chiar dacă domeniul este încă neajuns la maturitatea deplină. Așa cum se va vedea în continuare, foarte mulți termeni sunt cunoscuți anterior din alte domenii, dar deoarece tehniciile respective au fost „împrumutate” în DM, aceștia au fost adoptați fără complexe. Astfel, putem spune că este greu să se vorbească de o specificitate autentică a terminologiei DM. În concluzie, odată cu împrumutarea „unelțelor” s-a împrumutat și numele acestora. Așa cum am arătat încă de la început, DM utilizează orice metodă de analiză a datelor pentru descoperirea de informație ascunsă în acestea și, în mod natural, utilizează și terminologia corespunzătoare.

Este imposibil să trecem în revistă și numai o parte din termenii utilizați în DM. O succintă căutare pe Internet produce o listă enormă de termeni în conexiune cu DM. Prezentăm mai jos doar un eșantion de termeni – aleși aleator – cu referire la aplicațiile de tip DM. Termenii sunt prezenți în limba Engleză, pentru a fi mai ușor de descoperit la o căutare pe Net.

- Algorithms
- Artificial intelligence
- Artificial neural network
- Business intelligence
- Business performance management
- Database
- Data clustering
- Data stream mining
- Data warehouse
- Decision tree
- Discovery science
- Document warehouse
- Forecasting
- Java Data Mining
- Knowledge discovery
- Discriminant analysis

- | | |
|---|--|
| <ul style="list-style-type: none"> • Logit (logistic regression) • Machine learning • Nearest neighbor (pattern recognition) • Preprocessing data • Regression analysis • Statistics • Treatment learning • Vizualization | <ul style="list-style-type: none"> • Fraudling card • Modeling, models • Pattern recognition • Principal components analysis • Relational data mining • Text mining • Verification and Validation • Web mining |
|---|--|

O listă destul de bogată privind un posibil glosar cu terminologia DM explicată, poate fi găsită la adresa <http://www.twocrows.com/glossary.htm>.

1.8. Confidențialitatea datelor

Încă de la începuturile utilizării tehniciilor DM, s-a pus problema implicațiilor care pot apărea privind intimitatea persoanelor (*privacy issues*), relativ la sursa datelor procesate. De exemplu, informațiile financiare în domeniul bancar, pentru acordarea de credite persoanelor fizice sau juridice, pot fi utilizate într-un mod incorrect pentru individ/firma respectivă, relevând informații confidențiale periculoase privind evoluția sa viitoare. De asemenea, un alt domeniu „fierbinte” în acest context este acela al bazelor de date medicale. Utilizarea incorrectă a informației medicale a unui anumit individ îl poate prejudicia serios pe acesta (de exemplu, la încheierea unei asigurări pe viață, acordarea unui credit, angajare etc.).

Prin natura sa, Data Mining operează cu baze de date oferind informație care, de cele mai multe ori, nu poate fi obținută prin alte mijloace. În contextul în care aceste date se referă la persoane, trebuie rezolvate în mod clar problemele de intimitate, legalitate și etică. Deoarece se strâng anual milioane de date de toate felurile care sunt apoi procesate cu mijloace DM, este absolut necesară dezvoltarea unui cadru legal privind apărarea intimității personale, pentru a nu face loc apariției unui „e-Big Brother”. În acest sens este de remarcat, de pildă, că Uniunea Europeană interzice utilizarea de către bănci a informațiilor financiare în domeniul creditelor, cu toate că în S.U.A. acest lucru este permis cu anumite restricții [88].

2. „MINA” DE DATE

2.1. Ce sunt datele?

Cuvântul „date” reprezintă pluralul latinescului *datum*, provenind la rândul său din verbul *dare* (a da). Să menționăm, ca unul dintre primii care a utilizat acest termen, pe Euclid în lucrarea sa „Dedomena” -în latina „Data”- (aprox. 300 î.Hr.) (<http://en.wikipedia.org/wiki/Data>). Acest scurt recurs la istorie arată cât de veche este preocuparea omenirii pentru strângerea și apoi utilizarea informației ascunse în date.

Datele „brute”, așa cum au fost ele obținute direct prin diferite procese de achiziție, se referă la numere, figuri, imagini, sunete, programe de computer (privite ca și colecții de date interpretate ca instrucțiuni) etc. Aceste date, odată culese, sunt apoi procesate, obținându-se astfel informație care este stocată, utilizată sau transmisă mai departe într-un proces de tip „buclă”, adică cu posibilitatea ca o parte din datele procesate să reprezinte date brute pentru alte procesări ulterioare.

În cele ce urmează vom considera colecții de date, privite ca mulțimi de obiecte/eșantioane/vectori/instanțe/etc. - așezate pe liniile unui tablou - fiecare asemenea element al colecției fiind caracterizat de un set de caracteristici/attribute – așezate pe coloanele tabloului, ca în figura de mai jos.

The diagram shows a dataset represented as a table. On the left, a large curly brace labeled 'Obiecte' groups the entire table. Above the table, a horizontal curly brace labeled 'Atribute' groups the column headers. The table has 10 rows and 3 columns. The columns are labeled 'Vârstă (ani)', 'Venit lunar', and 'Tip mașină'. The data is as follows:

Vârstă (ani)	Venit lunar	Tip mașină
28	150€	Nu
34	5000€	Sport
42	5000€	Premium
44	2000€	Medie
50	1500€	Compactă
47	8000€	Premium
52	400€	Second
55	200€	Second
45	1000€	Compactă
63	150€	Nu

Atunci când vorbim de tipuri de date – în terminologie statistică – ne referim de fapt la atributele obiectelor din colecția de date. Am considerat necesar să facem această precizare deoarece, în general în literatura statistică, (e.g. [5], [51]), se vorbește de date (statistice) de tip cantitativ, calitativ, categorial etc., termenii referindu-se de fapt la atributele unor obiecte. Această precizare fiind făcută, de aici încolo vom considera ca dată un obiect/instanță-dată, acesta fiind caracterizat de atributele sale care pot fi cantitative, calitative, categoriale etc.

2.2. Tipuri de mulțimi de date

Să discutăm, pe scurt, despre câteva tipuri mai cunoscute de mulțimi/seturi de date. Astfel, putem menționa următoarele tipuri de seturi de date mai des întâlnite în probleme practice de Data Mining [162]:

- Înregistrări:
 - ▶ Date matriciale (*Data Matrix*);
 - ▶ Date document (*Document Data*);
 - ▶ Date tranzacționale (*Transaction Data*).
- Diagrame (*graph*):
 - ▶ World Wide Web;
 - ▶ Structuri moleculare.
- Mulțimi ordonate de date:
 - ▶ Date spațiale (*Spatial Data*);
 - ▶ Date temporale (*Temporal Data*);
 - ▶ Date secvențiale (*Sequential Data*);
 - ▶ Secvențe genetice (*Genetic Sequence Data*).

Datele de tip *înregistrare* constau în colecții de înregistrări, fiecare înregistrare (obiect/instanță) fiind caracterizată de un set de atribuție. În principiu, fiecare obiect-dată are un număr fixat de atribuție (i.e. lungime constantă), astfel încât el poate fi considerat ca un vector dintr-un spațiu vectorial multidimensional a cărui dimensiune este, evident, numărul de atribuție/componențe ale obiectului dat. O astfel de colecție de date poate fi deci reprezentată sub forma unei matrice de tip $m \times n$, unde fiecare din cele m linii corespunde unei instanțe, iar fiecare din cele n coloane corespunde unui atribut. Un exemplu clasic de astfel de colecție de date este reprezentat de registrele din spitale, conținând înregistrări medicale ale pacienților, unde fiecare rând din

registru reprezintă un pacient (obiect/instanță/vector), iar pe fiecare coloană a sa sunt înregistrate diferite valori (numerice sau nu) reprezentând înregistrări medicale specifice (e.g. vîrstă, sex, domiciliu, glicemie, colesterol, existența sau ne-existența unui anumit simptom, diagnostic etc.), aşa cum se vede mai jos (am ignorat voit prima coloană care se referă la identitatea pacienților).

	GGT	COLESTEROL	ALBUMINA	VÂRSTA	GLICEMIA	SEX
CIROZĂ	289	148	3.12	57	0.9	M
CIROZĂ	255	258	3.5	65	1.1	M
HEPATITĂ	32	240	4.83	60	1.14	F
HEPATITĂ	104	230	4.06	36	0.9	F
CIROZĂ	585	220	2.7	55	1.5	F
CIROZĂ	100	161	3.8	57	1.02	M
HEPATITĂ	85	188	3.1	48	1.09	M
CIROZĂ	220	138	3.84	58	0.9	M
CANCER	1117	200	2.3	57	2.2	F
CIROZĂ	421	309	3.9	44	1.1	M

În ceea ce privește datele document, fiecare document înregistrat în baza de date devine un vector „termen”, fiecare termen fiind un atribut al vectorului, astfel încât valoarea atribuită componentei respective să însemne numărul de apariții ale termenului în document, aşa cum se observă și în tabelul de mai jos.

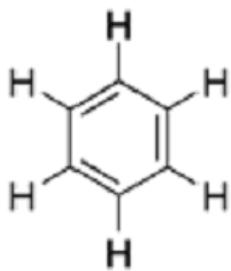
	culoare	valoare	scor	valută	partid	rețetă	echipă
Document A	30	25	17	15	0	2	43
Document B	4	13	2	0	14	2	2
Document C	6	42	0	0	0	123	0
Document D	0	104	0	23	0	0	12
Document E	3	585	0	0	0	60	0

Datele tranzacționale se referă îndeobște la tranzacții comerciale, fiecare înregistrare reprezentând o tranzacție a unui grup de mărfuri, deci matematic vorbind, un vector ale cărui componente (item-uri) înseamnă mărfurile tranzacționate.

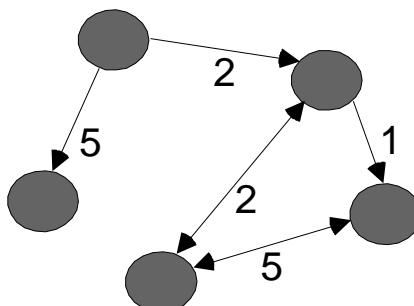
Tranzacție	Item
1.	{pâine, coca cola, lapte}
2.	{bere, pâine}
3.	{bere, coca cola, scutece, lapte}

4.	{bere, pâine, scutece, lapte}
5.	{coca cola, scutece, lapte}

Datele sub formă de diagrame (grafuri), aşa cum arată și numele, vor fi scheme care înglobează în ele informație de un anumit tip (e.g. WWW, structuri chimice moleculare etc.)



Molecula de benzen (C_6H_6)



Graf direcționat generic

```

<a href="papers/papers.html#bbbb">
Data Mining </a>
<li>
<a href="papers/papers.html#aaaa">
Graph Partitioning </a>
<li>
<a href="papers/papers.html#aaaa">
Parallel Solution of Sparse Linear System of Equations </a>
<li>
<a href="papers/papers.html#ffff">
N-Body Computation and Dense Linear System Solvers

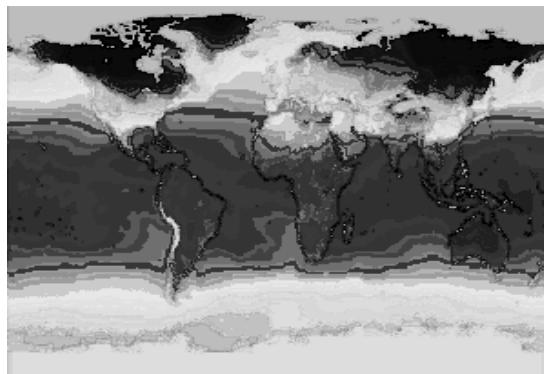
```

Link-uri HTML

Mulțimile ordonate de date se referă la acele seturi de date care, într-un fel sau altul, sunt guvernate de o anumită ordine în care apar obiectele. Mai jos sunt ilustrate date ordonate din domeniul geneticii – secvențe genomice (ADN) și meteorologiei – repartiția temperaturii medii pe glob pe o anumită perioadă determinată de timp. Nu mai este nevoie să subliniem faptul că problemele genomicii (i.e. studiul genomului unui organism = întreaga informație ereditară a unui organism codată în ADN) sau cele ale studiului evoluției meteorologice globale (e.g. efectul de încălzire globală) sunt printre

cele mai „calde” teme de cercetare actuală. De exemplu, în ceea ce privește problema genomicii, există baze de date publice dedicate (e.g. NCBI – <http://www.ncbi.nlm.nih.gov/mapview>; UCSC – <http://genome.ucsc.edu>; ENSEMBL –<http://www.ensembl.org>) care, în anumite condiții, pot fi accesate și utilizate. În ceea ce privește datele spațio-temporale de referință privind temperatura globală, pot fi accesate site-uri ca de exemplu cel al agenției NASA -http://www.nasa.gov/vision/earth/lookingearth/earth_warm.html.

T G A A A C T A A G
 C T T G G A C G T T
 C A G C G G T A C T
 A T G A G A T A C G
 A C A C A C A A A A
 T T T G C T G A T G
 A A C C A T G C T T
 T T A T C A A C C T
 G C T C A A C T C T
 T T T T T G C T G A
 T A G G A T A G G G
 C A A T T A T A T



Secvențe genomice (ADN)

Date meteorologice spațio-temporale

Alt exemplu de date ordonate se referă la secvențele obținute din date tranzacționale, adică secvențe de colecții de item-uri. Un exemplu tipic de astfel de date se referă la datele înregistrate în baza de date ale clienților (*customer database*), unde fiecare tranzacție reprezintă o colecție de mărfuri cumpărate de un client la o singură vizită în supermarket (*market basket data*). Alt tip de astfel de date de tip tranzacțional se referă la datele privind utilizarea Web-ului.

	Date tranzacționale comerciale	Date privind utilizarea Web
Date	Coșul de item-uri (<i>Basket of items</i>)	Vizitarea Iстории (<i>Visiting History</i>)
Pattern-uri	Secvențe de item-uri e.g. (<A,B>, <D>, <C,E,F>)	Secvență de acces e.g. <1, 3, 5>

2.3. Calitatea datelor

Calitatea datelor se referă la caracteristica lor de a se potrivi mai mult sau mai puțin cu utilizarea pentru care au fost culese sau, alternativ, dacă

reflectă corect contextul real din care provin. În ceea ce privește unghiul din care este privită calitatea datelor, vom menționa următoarele:

- ▶ Integrarea produsului (conformitatea cu specificațiile) cu service-ul (așteptarea consumatorului) [107];
- ▶ Calitatea formei, semnificației și utilizării –semiotica [130];
- ▶ Natura ontologică a sistemelor informaționale [172].

În esență, principalele caracteristici avute în vedere atunci când se discută calitatea datelor sunt: acuratețea, corectitudinea, valabilitatea/circulația, completitudinea și relevanța.

Principalele probleme cu care ne confruntăm în procesul de culegere a datelor sunt:

- ▶ Zgomotul și valorile extreme/excepționale (*noise, outliers*);
- ▶ Valori lipsă (*missing data*);
- ▶ Date dupicate.

2.4. Tipuri de attribute

Așa cum am specificat anterior, un obiect dintr-o mulțime de date este determinat de caracteristicile/attributele sale. Vom vorbi în continuare despre attribute, reamintind încă odată că, în special în studiile statistice, noțiunile de date și attribute se confundă.

Procesul de Data Mining este influențat de tipul atributelor obiectelor analizate. De exemplu, atunci când se procedează la o vizualizare a datelor, reprezentarea grafică a acestora depinde de natura observațiilor: există procedee diferite atunci când se analizează status-ul marital al unor persoane în comparație cu analizarea contului în bancă sau al investigării măsurii în care aceștia fumează sau consumă alcool etc. Din acest motiv este absolut necesar să detaliem problema naturii datelor cu care operează DM.

Datele pot fi împărțite, în principiu, în două tipuri importante: *date numerice (quantitative)* și *date categoriale (qualitative)*, cu toate că sunt folosite, mai rar, și alte tipuri de date. Să prezentăm, pe scurt, caracteristicile principale pentru fiecare tip de date.

- **Date numerice (numerical data).** Datele numerice, cantitative, sunt la rândul lor de două feluri: date discrete și date continue. Datele *discrete* apar atunci când este vorba de observații numerice întregi privitoare la un anumit proces de numărare ca, de exemplu, numărul de copii ai unei familii, pulsul, numărul de consultații pe an la care a fost supus un pacient, codul zip, codul numeric, codul PIN, numărul anumitor cuvinte apărute în documente, attribute binare etc. Menționăm în acest context și cele patru tipuri de stocare a datelor întregi (*integer*): (a) *byte* -8 biți- numere cuprinse între -128 și 127 (i.e. -2^7 și 2^7-1); (b) *short* -16 biți- numere cuprinse între

-32.768 și 32.767 (i.e. -2^{15} și $2^{15}-1$); (c) *int* -32 biți- numere cuprinse între -2.147.483.648 și 2.147.483.647 (i.e. -2^{31} și $2^{31}-1$); (d) *long* -64 biți- numere cuprinse între -9.223.372.036.854.775.808 și 9.223.372.036.854.775.807 (i.e. -2^{63} și $2^{63}-1$). Spre deosebire de datele numerice discrete, obținute de regulă în urma unui proces de numărare, datele numerice *continue* se obțin îndeobște în urma unor măsurători, de exemplu înălțimea, greutatea, tensiunea arterială, colesterolul unei anumite persoane, temperatura, presiunea atmosferică, viteza vântului, valoarea contului din bancă sau valoarea acțiunilor tranzacționate la Bursă etc. Aceste date sunt de regulă exprimate prin numere reale, spre deosebire de cele discrete care sunt restricționate la numerele întregi. Vom menționa aici că, în multe analize, datele discrete sunt tratate ca date continue, de exemplu numărul de bătăi pe minut al inimii. Pentru ca analiza unor asemenea date (discrete, dar considerate continue) să nu aibă de suferit, trebuie dispus de un număr suficient de mare de valori diferite posibile ale acestora care să creeze premisele ipoteticei naturi continue a lor. Menționăm în acest context datele continue (cu virgulă mobilă) care pot fi stocate: (a) *float* – numere cuprinse între 1.4E-45 și 3.4E+38; (b) *double* - numere cuprinse între 4.9E-324 și 1.7E+308.

- **Date categoriale (categorical data).** Spre deosebire de datele numerice, datele categoriale sau calitative sunt acele date care, aşa cum le spune şi numele, împart instanțele în diferite categorii, ca de exemplu:

1. bărbat/femeie
2. căsătorit/necăsătorit/văduv/divorțat
3. fumător/nefumător
4. hipertensiv/hipotensiv
5. stadii în anumite boli (e.g. cancer): I, II, III, IV
6. existență simptoame: DA, NU
7. tip diagnostic: A, B, C, D etc.

Să observăm că datele numerice discrete sunt câteodată tratate ca date categoriale, de exemplu numărul de copii născuți de o femeie, e.g. 0, 1, 2, 3, 4, împart mamele în categoriile corespunzătoare numărului de copii. Este important ca în această situație să se ignore noțiunile de ordine sau de parametri numerici ca, de exemplu, media. Invers, nu este corect să interpretăm datele categoriale ordonate ca date numerice, de exemplu, la stadiile în anumite boli, stadiul IV nu este de două ori mai rău decât stadiul II, și.a.m.d. Vom menționa că, în cadrul datelor categoriale, putem vorbi de date *nominale* ca, de exemplu: grupa sanguină (A/B/AB/O), culoarea ochilor, sexul etc. și de date *ordinale* ca, de exemplu: gradul fumatului (e.g. nefumător, fost fumător, fumător „amator”, fumător „înrăit”), ierarhizarea durerii (e.g. mică, medie, mare), ierarhizarea înălțimii (e.g. scund, mediu, înalt), duritatea mineralelor etc., în principiu, atunci când avem de a face cu o ierarhizare a „intensității” atributului respectiv.

- **Alte tipuri de date.** În afară de cele două mari tipuri de date menționate mai sus, în DM se mai operează câteodată și cu alte tipuri de date. Enumerăm mai jos tipurile cele mai cunoscute de astfel de date.
 - *Rangul* reprezintă locul pe care îl ocupă un obiect într-o ierarhie (e.g. competiție sportivă, examinare, preferința medicului pentru un anumit tratament, preferința clienților pentru un anumit produs etc.).
 - *Procentajul*, așa cum arată și numele, descrie o anumită proporție (raport) între două cantități (e.g. procentajul de bărbați dintr-o populație, greutatea corporală relativă (raportul dintre greutatea observată și greutatea ideală), procentajul de stângaci dintr-o populație, procentajul clienților fideli, procentajul obiectelor clasificate corect, procentajul datelor lipsă etc.).
 - *Rate și rapoarte* referitoare la frecvența observată a unui fenomen sau rapoartele dintre două mărimi, altele decât procentajele (e.g. mortalitatea raportată la mia de locuitori, rata de apariție a unei boli pe sexe sau arii geografice, rata de schimb monetară, raportul pret/calitate etc.)
 - *Scorul* este folosit atunci când nu este posibilă o măsurătoare directă și neambiguă și trebuie totuși cuantificată o anumită mărimă (e.g. scorul Apgar la nou-născuți, gravitatea unei boli cuantificată ca ușoară, moderată, severă, colorația pielii în anumite maladii, etc.)
 - *Scale vizuale analogice* folosite mai ales în studiile medicale atunci când subiectul este rugat să indice pe o scală punctul care este considerat a ilustra cel mai bine gradul de durere, de exemplu. Cu toate că este o reprezentare foarte subiectivă, aproape imposibil de cuantificat numeric, reprezintă totuși un mijloc de a „măsura” un anumit fenomen

O situație frecventă care apare în analiza exploratorie a datelor (abordarea DM statistică), mai ales a celor din domeniul medical, este aceea a datelor *cenzurate*. Sunt cazuri în care o anumită observație nu poate fi bine precizată. De exemplu, în analiza supraviețuirii (*survival analysis*) -o tehnică statistică clasica care studiază dinamica timpului de supraviețuire după o anumită operație sau tratament- o parte dintre subiecți inclusi în lotul de studiu decedează în perioada de observație, dar o altă parte dintre subiecți supraviețuiesc în această perioadă sau se retrag benevol și astfel momentul decesului nu mai poate fi înregistrat în mod clar, indubabil. Un alt exemplu este atunci când se efectuează anumite măsurători și aparatul respectiv nu poate înregistra valori mai mici sau mai mari decât scala sa, cu toate că acestea există în realitate -date nedetectabile de către aparat. Rezumând, în orice situație în care o anumită dată există dar, din diferite motive, nu poate fi precizată clar, spunem că avem de-a face cu date cenzurate.

Tot în contextul utilizării metodelor statisticice de DM este util să discutăm, pe scurt, atât despre variabilitatea datelor cât și despre modelul

probabilist al datelor, cu referire mai ales la termenul de *variabilă* care este de multe ori echivalent cu cel de *atribut* al unui obiect/instanțe.

Astfel, atunci când procesăm datele în cursul analizelor statistice incluse în procesul DM, este absolut necesar să existe aşa numita variabilitate a lor. Prin *variabilitate* înțelegem orice fel de modificare care are loc într-o mulțime de date, indiferent de tipul lor, cu alte cuvinte variabilitatea este opusul constanței datelor. Trebuie sătuit faptul că nu se poate face analiză statistică pe variabile care sunt constante. O bună parte a analizelor statistice clasice (e.g. regresia) face apel la legăturile care există între diferite date referitoare la aceeași subiecți, studiind modul cum variația unei unor influențează variația altora (e.g. legătura dintre înălțime și greutate, între factorii de risc și probabilitatea declanșării unei maladii etc.). Ori, dacă un factor din analiza statistică nu are variabilitate (i.e. este constant) atunci el este ca și inexistent în analiză. Cu cât variabilitatea datelor este mai mare cu atât analiza statistică este mai bogată în rezultate consistente.

În cele expuse în acest capitol privind datele din punct de vedere statistic, am accentuat doar partea descriptivă a lor, fără a încerca să le definim în context probabilist. Deoarece Statistica nu poate fi ruptă de Teoria Probabilităților, care îi oferă mijloacele teoretice de investigație, este absolut necesar să definim datele în acest context. Să presupunem că avem la dispoziție o anumită mulțime de obiecte/instanțe (o populație statistică, în terminologia statistică) și suntem interesați de analiza principalelor lor caracteristici care reprezintă, aşa cum se știe, atribute (*date, caractere statistice* în limbajul statistic). Să vedem cum este definită din punct de vedere probabilist noțiunea de dată statistică. Probabilist vorbind, prin *dată* (sau *caracter*) vom înțelege o aplicație definită pe mulțimea ce reprezintă populația (de obiecte) și cu valori într-o anumită mulțime ce depinde de natura datei respective. Mai mult, considerând un câmp de probabilitate (Ω, Σ, P) , unde Ω este chiar populația considerată, iar Σ este o σ -algebră de părți ale lui Ω (în cazul în care Ω este finită, Σ coincide cu mulțimea părților lui Ω), *data X* a populației statisticice Ω este o *variabilă aleatoare* pe câmpul de probabilitate (Ω, Σ, P) , atunci când data este numerică. O astfel de variabilă aleatoare, privită din punct de vedere probabilistic, mai este cunoscută, privită din punct de vedere statistic, ca *variabilă statistică*. În cazul în care X nu ia valori numerice, se poate căteodată, pe baza unor echivalări numerice a acestor valori, să privim pe X tot ca pe o variabilă aleatoare (e.g. echivalarea sexului M și F cu 0 și 1, echivalarea numerică a unor date ordonale etc.).

Cu aceste precizări făcute, se poate utiliza tot arsenalul probabilisto-statistic pentru procesarea datelor în cadrul tehniciilor DM.

3. ANALIZA EXPLORATORIE A DATELOR

3.1. Ce este analiza exploratorie a datelor?

Motivația care stă la baza primului pas într-o procesare de tip DM a unei baze de date – analiza exploratorie a datelor – este foarte simplă și, în același timp, foarte serioasă. Înainte de toate, o astfel de analiză face apel la abilitatea omenească de a recunoaște pattern-uri pe baza experienței proprii. Pe baza informațiilor și cunoștințelor acumulate în timp, oamenii pot recunoaște anumite forme, tendințe, şabloane etc., sistematice în date, care nu totdeauna pot fi puse în evidență de metodele clasice de investigare a lor. Pe de altă parte, tot experiența dobândită într-un anumit domeniu poate ajuta semnificativ la alegerea celor mai bune tehnici de pre-procesare sau analiză a datelor. Ori, pentru a putea utiliza eficient aceste oportunități, este nevoie de o analiză a datelor, o explorare a lor cu mijloacele statistiche binecunoscute, după care să se poată alege metodologia DM optimă pentru datele disponibile.

Analiza exploratorie a datelor (*Exploratory Data Analysis* -EDA) este partea Statisticii care se ocupă cu trecerea în revistă, comunicarea și utilizarea datelor în cazul unui nivel scăzut de informație asupra lor. Spre deosebire de cazul clasic al testării ipotezelor, utilizat în Statistică pentru a verifica anumite supozitii apriorice (e.g. anumite corelații între diferite mărimi/variabile despre care există informații că ar fi cumva dependente), în cazul EDA se utilizează diferite tehnici pentru a identifica relații sistematice între anumite mărimi/variabile despre care nu există nicio informație prealabilă. EDA a fost creată și numită astfel de statisticianul american John Tukey (Tukey J., *Exploratory Data Analysis*, Addison-Wesley, 1977). Tehnicile computaționale EDA includ atât metode statistice elementare cât și altele avansate (utilizând din plin oportunitățile deschise de procesarea computerizată a datelor) – *tehnici exploratorii multivariate* – create pentru a identifica anumite pattern-uri ascunse în mulțimi multivariate de date. EDA utilizează tehnici diverse – multe dintre ele de vizualizare – cu scopul de a:

- * maximiza cunoașterea intimă a datelor;
- * dezvăluie structura de bază;
- * extrage variabilele importante;
- * detectă valorile extreme/excepționale și anomaliiile;
- * identifică ipotezele fundamentale pentru a fi apoi testate;
- * dezvoltă modele suficient de simple;
- * determină setarea optimă a parametrilor;
- * sugera unele ipoteze privind cauzele fenomenelor observate;
- * sugera tehnici statistiche potrivite datelor disponibile;

- ★ furniza cunoștințe pentru colectarea ulterioară de date pentru cercetare sau experimentare.

Odată utilizate tehniciile EDA ca un preambul la procesul de DM, se pune problema verificării rezultatelor astfel obținute. Trebuie subliniat faptul că explorarea datelor nu poate fi privită decât ca un prim stadiu de analiză a datelor și rezultatele obținute vor fi considerate cu titlu experimental atâtă timp cât nu sunt validate alternativ. Dacă, de exemplu, rezultatul aplicării EDA sugerează un anumit model, atunci acesta trebuie validat aplicându-l la alt set de date și testându-i astfel calitatea predictivă.

În concluzie, EDA poate fi considerată – în principiu – o filosofie despre modul în care să fie ‘disecate’, să fie ‘privite’ și, în sfârșit, să fie interpretate datele. Pentru mai multe amănunte relativ la rolul și mijloacele EDA în DM, se pot utiliza, printre altele, și următoarele link-uri:

- ▶ Nist-Sematech: <http://www.itl.nist.gov/div898/handbook/eda/eda.htm>;
- ▶ Wikipedia: http://en.wikipedia.org/wiki/Exploratory_data_analysis;
- ▶ Statgraphics: <http://www.statgraphics.com/eda.htm>.

În prezentarea de față vom insista pe următoarele tehnici EDA:

- Statistica descriptivă –rezumare numerică/reprezentare grafică;
- Analiza matricei corelațiilor variabilelor;
- Vizualizarea datelor;
- Examinarea repartițiilor variabilelor (analiza simetriei, non-normalitate, multimodalitate);
- Modele liniare și aditive avansate;
- Tehnici exploratorii multivariate;
- OLAP -Online Analytical Processing.

3.2. Statistica descriptivă

Descrierea statistică grupează o suită de metode diverse, având scopul rezumării unui număr mare de observații privind datele, punând astfel în evidență principalele lor caracteristici. În acest sens, există două mari abordări a descrierii statistice a datelor:

- Determinarea unor parametri numerici, interesul fiind focalizat pe proprietățile lor matematice;

- Diverse reprezentări grafice simple ale datelor, a căror interpretare nu este dificilă, fiind de cele mai multe ori foarte sugestivă, în ciuda faptului că este totuși limitată din punct de vedere strict informațional.

3.2.1. Parametrii descrierii statistice

Statistica descriptivă, privită ca tehnică împrumutată de EDA de la Statistica aplicată, se referă în principiu la acei parametri numerici care dau o imagine sintetică asupra datelor (e.g. media, mediana, deviația standard, modul etc.), rezumând astfel principalele caracteristici statistice ale datelor.

Statistica descriptivă are scopul, utilizând diferite metode specifice, de a rezuma un mare număr de observații privind un set de obiecte/instanțe, punând astfel în evidență caracteristicile principale ale atributelor lor. Există două mari metode de a atinge acest scop: fie utilizând o reprezentare grafică simplă, de care vom vorbi mai târziu în cursul acestui capitol, fie utilizând reprezentări numerice care cuprind principalele caracteristici statistice ale datelor respective. Indiferent de modul de abordare, este vorba de reprezentarea variabilității unor date. Această variabilitate poate fi una cu cauze cunoscute, o variabilitate „deterministă” care este descrisă statistic pentru a o pune și mai bine în evidență și a o cuantifica precis, sau poate fi o variabilitate cu cauze doar bănuite sau chiar necunoscute – variabilitatea „aleatorie” – și care, folosind statistică, se speră a fi clarificată cauzal.

Așa după cum bine se știe, un obiect (o instanță) corespunde din punct de vedere probabilist unei variabile aleatoare multidimensionale (i.e. un vector aleator n -dimensional), subiect al analizei statistice multivariate. Fiecare componentă a instanței, reprezentând un anumit atribut, este privită la rândul său ca o variabilă aleatoare. Este normal să luăm în considerație în acest caz parametrii numerici (statistici) ce caracterizează o variabilă aleatoare, parametri deosebit de utili în descrierea dinamicii acestora.

În cele ce urmează, vom considera că fiecare atribut x al unei instanțe reprezintă o valoare pe care o poate lua variabilă aleatoare generatoare X , corespunzătoare aceluia atribut. Reamintim că, pentru variabilă aleatoare X , funcția $F_X: \mathbf{R} \rightarrow [0, 1]$, definită de:

$$F_X(x) = P\{X < x\} = P\{\omega \in \Omega; X(\omega) \in (-\infty, x)\},$$

se numește *funcția de repartiție* (*repartiția de probabilitate*), unde (Ω, Σ, P) este un spațiu (câmp de probabilitate). În cazul unei analize statistice, unei variabile aleatoare X îi corespunde variabilă statistică corespunzătoare, notată în mod firesc tot X . La fel ca în cazul unei variabile aleatoare, și în cazul unei variabile statistice putem defini, în context, noțiunea de funcție de repartiție. Astfel, prin *funcția de repartiție* sau *funcția cumulativă* (*freqvența cumulată*) a variabilei statistice X (corespunzătoare variabilei aleatoare X), definită de seria

statistică asociată (eșantionul corespondent) $\{x_i\}_{i=1,\dots,n}$, înțelegem aplicația $F: \mathbf{R} \rightarrow [0, 1]$, dată de:

$$F(x) = \frac{f_x}{n}, \quad x \in R,$$

unde f_x reprezintă numărul observațiilor x_i strict mai mici decât x .

Mai întâi, la fel ca și în cazul probabilist, putem defini *cuantila* (*quantile* - Kendall, 1940) de ordin α , $0 < \alpha < 1$, a variabilei statistice X , ca numărul q_α cu proprietatea $F(q_\alpha) = \alpha$. Din punctul nostru de vedere, al explorării datelor, cuantilele implică practic divizarea în $1/\alpha$ submulțimi de mărimi egale a unei mulțimi ordonate de date, ele reprezentând de fapt tocmai frontierele (pragurile) dintre submulțimile consecutive (cuantilă – lat. *quantillus* = câtime). În acest context, o *percentilă* (*percentile* - Galton, 1885) reprezintă oricare dintre cele 99 de valori care împart o mulțime ordonată de date în 100 de submulțimi de mărimi egale, consecutive. De exemplu a 50-a percentilă împarte setul de date în 50% date dedesubtul său și 50% date deasupra sa. Tot astfel, *decilele* (*deciles*) reprezintă cele 9 valori care împart o mulțime ordonată de date în 10 submulțimi consecutive de mărimi egale, iar *cuartilele* (*quartiles* - Galton, 1882) reprezintă cele trei valori notate Q_1 , Q_2 , Q_3 , care împart o mulțime ordonată de date în 4 submulțimi consecutive de mărimi egale (i.e. tip $q_{i/4}$). Cele mai utilizate cuantile sunt cuartilele Q_1 , Q_2 , și Q_3 . Astfel, prima cuartilă Q_1 (cuartila inferioară – a 25-a percentilă) are sub ea 25% din date și deasupra 75% din date; a doua cuartilă Q_2 are sub ea 50% din date și deasupra tot 50% din date; a treia cuartilă Q_3 (cuartila superioară – a 75-a percentilă) are sub ea 75% din date și deasupra 25% din date.

Ilustrăm mai jos rolul cuantilelor în cazul unui set de date corespunzător unui lot de 299 pacienți cu afecțiuni hepatice. Astfel, în *Tabelul 1* am reprezentat decilele iar în *Tabelul 2* cuartilele, referitoare la colesterol și glicemie.

Tabelul 1. Decilele corespunzătoare valorilor colesterolului și glicemiei

Colesterol									
$q_{1/10}$	$q_{2/10}$	$q_{3/10}$	$q_{4/10}$	$q_{5/10}$	$q_{6/10}$	$q_{7/10}$	$q_{8/10}$	$q_{9/10}$	
144	164	176	190	200	201	220	230	265	
Glicemie									
$q_{1/10}$	$q_{2/10}$	$q_{3/10}$	$q_{4/10}$	$q_{5/10}$	$q_{6/10}$	$q_{7/10}$	$q_{8/10}$	$q_{9/10}$	
0,78	0,81	0,87	0,90	0,96	1,00	1,06	1,12	1,29	

Tabelul 2. Cuartilele corespunzătoare valorilor colesterolului și glicemiei

Colesterol		
Q_1	Q_2	Q_3
170	200	227
Glicemie		
Q_1	Q_2	Q_3
0,85	0,96	1,10

În principiu, valorile tipice corespunzătoare unei analize a datelor sunt următoarele:

- măsuri tipice ale tendinței centrale (locației): *modul (mode)*, *mediană (median)*, *media (mean, arithmetic mean, average)*, *media geometrică (geometric mean)* și *media armonică (harmonic mean)*;
- măsuri tipice ale împrăștierii (deviației): *dispersia (variance)* și *deviația standard/abaterea medie pătratică (standard deviation)*;
- măsuri tipice ale formei repartiției: *asimetria (skewness)* și *excesul (kurtosis)*.

Cel mai comun parametru ce măsoară „tendința centrală” a unei serii statistice este *media*, care reprezintă practic media aritmetică a tuturor observațiilor, fiind dată de formula:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i .$$

Trebuie să amintim că, în ciuda faptului că media este cea mai uzitată măsură a tendinței centrale, fiind de altfel și o caracteristică foarte sugestivă a datelor pe care le reprezintă, ea este foarte „sensibilă” la existența valorilor extreme (*outliers*), care îi pot perturba serios capacitatea de ilustrare a datelor. Pentru a proba această observație, să considerăm, de exemplu, următoarea secvență de date ce pot reprezenta valorile înălțimii unor persoane:

$$\{1.70, 1.67, 1.87, 1.76, 1.79, 1.66, 1.69, 1.85, 1.58, 1.78, 1.80, 1.83, 2.20\}$$

Se observă că, în această succesiune de înălțimii, ultima valoare de 2,20m reprezintă o valoare „extremă” (o înălțime excepțională, de „baschetballist”). Calculând media valorilor de mai sus, cu și fără această valoare, obținem $m_1 = 1,78$ și $m_2 = 1,74$ m, valori suficient de distințe datorate influenței unei singure date. Pentru a evita asemenea situații, în astfel de cazuri se utilizează mediana în locul mediei. Astfel, *mediană* este definită ca numărul real care împarte în două efective egale seria statistică dată, observațiile fiind ordonate crescător, cu alte cuvinte mediana este chiar a doua quartilă Q_2 . Formal, mediana este dată de:

$$P\{X \leq Q_2\} = P\{X > Q_2\} = 1/2.$$

Dacă efectivul seriei statistice este un număr impar $n = 2k + 1$, atunci mediana este a $(k + 1)$ -a valoare a seriei, iar dacă efectivul este un număr par $n = 2k$, atunci mediana se înlocuiește cu *intervalul median* dat de valorile a k -a și a $(k + 1)$ -a (mediana se poate considera astfel ca mijlocul acestui interval – media aritmetică a capetelor sale). În continuarea la ceea ce am spus mai sus, putem aminti și faptul că mediana mai este folositoare atunci când există posibilitatea ca unele valori extreme ale seriei statistice să fie cenzurate. Atunci când există observații care se găsesc fie sub un anumit prag de jos, fie deasupra unui anumit prag de sus și, din diferite motive, nu sunt suficient de exact precizate, nu putem folosi media, înlocuind-o prin mediană dacă avem valori exacte pentru mai mult de jumătate din observații (cazul măsurătorilor fizico-chimice, când există valori în afara scalei normale a aparatului de măsură). Trebuie să înțelegem că ambele măsuri sunt la fel de eficiente și, cu toate că media este mai frecvent folosită decât mediana, aceasta din urmă poate fi mai valoroasă în anumite circumstanțe. În exemplul de mai sus, al setului de date privind înălțimea unor persoane, medianele corespunzătoare celor două cazuri (cu și fără valoarea extremă) sunt $med_1 = 1.78m$ și $med_2 = 1.77$, deci mediana nu este influențată semnificativ de valori extreme.

Altă măsură a tendinței centrale pe care o prezentăm aici este *modul* (sau *moda*), care reprezintă pur și simplu cea mai frecventă valoare a seriei, fiind rareori folosită în cazul datelor continue (unde reprezintă punctul de maxim al densității de repartiție corespunzătoare) și des utilizată în cazul datelor categoriale. Să menționăm că există și seturi de date plurimodale (multimodale), adică cu mai multe valori având aceeași frecvență maximă de apariție (mai multe mode). Dacă datele sunt grupate în clase (i.e. valorile aparțin unor intervale), vom numi *clasă modală* orice clasă corespunzând unui maxim al frecvenței.

Media geometrică este dată de formula:

$$\sqrt[n]{\prod_{1}^n x_1 \cdot x_2 \cdots x_n}$$

Media geometrică este utilizată cu precădere în cazul măsurătorilor cu scală neliniară (e.g. în psihometrie, unde rata intensității unui stimul este adesea o funcție logaritmică în raport cu intensitatea, caz în care se folosește media geometrică și nu media aritmetică).

Media armonică, dată de formula:

$$\frac{n}{\sum_{1}^n 1/x_i},$$

se utilizează câteodată în cazul determinării mediei frecvențelor.

Pentru cazul seriei statistice a înălțimilor, considerată mai înainte, avem următoarele valori pentru medie, modă, mediană, medie geometrică și medie aritmetică.

Tabelul 3. Principalii parametrii ai tendinței centrale

Medie	Modă	Mediană	Medie geometrică	Medie armonică
1.78	1.65/1.75	1.78	1.77	1.77

Din tabelul de mai sus se observă că:

- ⇒ media și mediana coincid în acest caz;
- ⇒ repartitia (continuă) este plurimodală (două clase modale (1.6, 1.7) și (1.7, 1.8) au frecvența 30.8%);
- ⇒ mediile geometrică și armonică coincid;
- ⇒ repartitia este suficient de „simetrică”, parametrii tendinței centrale fiind apropiati ca valoare.

Vom da acum un exemplu de calcul al medianei și modei pentru date discrete. Să considerăm următorul tabel de distribuire (repartiție) a datelor, în care valorile x_i au frecvențele de apariție n_i .

x_i	9	11	13	14	16
n_i	2	4	1	4	1

Se observă că repartitia datelor este bi-modală (valorile 11 și 14 au frecvența maximă de apariție egală cu 4) și există un interval median (11, 13) cu o „mediană” egală cu 12.

O altă abordare privind analiza datelor este reprezentată de măsurarea împărăstirii (dispersiei) față de medie, adică măsurarea distanței fiecărei valori a seriei statistice față de media eșantionului. Plecând de la cazul probabilist clasic al dispersiei, vom defini *dispersia* sau *varianța* (*variance*) - termen introdus de Fisher, 1918 - corespunzătoare unei serii statistice $\{x_i\}_{i=1, \dots, n}$, cu ajutorul formulei:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - m)^2,$$

unde m este media cunoscută a variabilei statistice (a populației originare, cu alte cuvinte). Deoarece, de obicei, considerăm că seria statistică cu care lucrăm nu reprezintă toată populația ci este doar un eșantion al ei și astfel media m nu este cunoscută ci putem calcula doar media eșantionului \bar{x} , vom folosi în locul formulei de mai sus o formulă de aproximare (o estimare) a dispersiei, înlocuind media m cu media seriei \bar{x} și împărțind prin $(n - 1)$ în loc de n , deci:

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2.$$

Remarcăm aici că, pentru serii statistice de dimensiuni mari, diferența dintre valoarea dată de formula de mai sus și formula clasică:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2,$$

este neglijabilă. De multe ori, este preferabil ca în loc de dispersie să folosim o mărime care este măsurată cu aceeași unitate ca și seria statistică, și anume *deviația standard* (sau *abaterea medie pătratică*), dată de formula:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Deviația standard este folosită în statistică descriptivă mai ales pentru definirea unor intervale în care se găsesc marea majoritate a observațiilor. Astfel, în cazul unor repartiții rezonabil de simetrice, marea majoritate a observațiilor ce compun seria statistică (aproximativ 95% din ele) se găsesc în intervalul definit de: *medie $\pm 2 \times$ deviația standard*, numit *interval de încredere (confidence interval)*. Subliniem încă odată că este absolut necesar să avem repartiții relativ simetrice, altfel cele spuse mai sus nu mai au relevanță. În cazul în care repartiția variabilei statisticice date este departe de o repartiție suficient de simetrică, există metode de descriere statistică a variabilității sale utilizând repartiții simetrice, de exemplu considerarea unei transformări matematice a seriei originale (e.g. logaritmând seria originală).

Vom ilustra măsurile tipice împărtășierii și formei repartiției în cazul seriei statisticice corespunzătoare înălțimii, valori prezentate în tabelul de mai jos.

Tabelul 4. Principalii parametrii ai împărtășierii (dispersia, deviația standard, intervalul de încredere pentru medie)

Dispersia	Deviația standard	Interval încredere 95%
0,023	0,151	(1.69, 1.87)

Remarcă: Înafara parametrilor statistici amintiți mai sus, se mai utilizează câteodată și:

- *Domeniul (range)* valorilor, reprezentat de diferența între maximul și minimul valorilor datelor;
- *Domeniul inter-cuartile (interquartiles range)*, definit de diferența $Q_3 - Q_1$;
- Media abaterilor absolute de la medie;

$$AAD(x) = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|$$

- Mediana abaterilor absolute de la medie:

$$MAD(x) = \text{mediana } \{ |x_1 - \bar{x}|, |x_2 - \bar{x}|, \dots, |x_n - \bar{x}| \}$$

3.2.2. Descrierea statistică a unei serii de cupluri

În paragraful precedent am prezentat diferite modalități pentru descrierea unei serii de observații (serie statistică/echantion), corespunzătoare unei singure variabile statistice, guvernând un anumit atribut. Atunci când considerăm două sau mai multe serii de observații, reprezentate de două sau mai multe serii statistice, corespunzând la două sau mai multe atribute diferite ale obiectelor analizate, în afara descrierii lor individuale, ca în subparagraful anterior, este indispensabilă analizarea legăturii ce poate exista între ele. Pentru aceasta sunt utilizate alte tehnici statistice (numerice sau grafice) ca, de exemplu, parametri condiționali și marginali, corelații, covarianță, regresie etc.

(a) *Parametri condiționali și marginali*

Să presupunem, simplificând, că dispunem de două variabile statistice X și Y , corespunzând la două atribute ale obiectelor bazei de date analizate (e.g. colesterolul și glicemia în cazul bazei de date corespunzătoare maladiilor hepatice). Teoretic vorbind, cele două variabile statistice X și Y corespund la două variabile aleatoare X și Y , de a căror repartitie de probabilitate comună suntem interesați. Să considerăm cuplul de variabile aleatoare (X, Y) pe același spațiu de probabilitate (Ω, Σ, P) . Funcția dată de:

$$F_{XY}(x, y) = P\{X < x, Y < y\},$$

reprezintă *funcția de repartitie* (bidimensională) comună asociată cuplului (X, Y) . Prin *repartitia marginală* a variabilei X înțelegem funcția:

$$F_X(x) = P\{X < x\} = F_{XY}(x, \infty).$$

Similar se definește și *repartitia marginală* a variabilei Y , adică:

$$F_Y(y) = F_{XY}(\infty, y).$$

În acest context, variabilele X și Y se numesc *independente* dacă:

$$F_{XY}(x, y) = F_X(x)F_Y(y), \forall x, y.$$

Mai departe,

$$F_{X|Y}(x|y) = \sum_{a \leq x} \frac{p_{XY}(a,y)}{P_Y(y)}, \quad p_Y(y) > 0 \text{ - cazul variabilelor (datelor) discrete}$$

$$F_{X|Y}(x|y) = \int_{-\infty}^x \frac{f_{XY}(t,y)}{f_Y(y)} dt, \quad \text{- cazul variabilelor (datelor) continue}$$

reprezintă *funcția de repartiție condiționată a lui X*, dacă $Y = y$.

Exemplu:

În spațiul de probabilitate (Ω, Σ, P) vom considera cuplul de variabile aleatoare discrete independente (X, Y) unde:

x_i	1	2	3	4	y_i	-1	0	1
p_i	0,1	0,2	0,3	0,4	p_i	0,25	0,5	0,25

În tabelul de mai jos este prezentată atât repartiția comună a cuplului cât și repartițiile marginale corespunzătoare.

$X \backslash Y$	-1	0	1	F_X
1	0,025	0,05	0,025	0,1
2	0,05	0,1	0,05	0,2
3	0,075	0,15	0,075	0,3
4	0,1	0,2	0,1	0,4
F_Y	0,25	0,5	0,25	1

Deoarece, în principiu, este posibil să existe o anumită legătură între atributele corespunzătoare variabilelor X și Y , vom considera ca parametri ai descrierii statistice mediile condiționate.

Astfel, teoretic vorbind, în cazul variabilelor (atributelor) discrete avem:

$$E[X | Y = y] = \sum_x \frac{x p_{XY}(x,y)}{p_Y(y)}, \quad p_Y(y) > 0,$$

care se numește *media condiționată a lui X* dat fiind $Y = y$, și atunci:

$$E[X] = \sum_x x p_X(x) = \sum_x \sum_y x p_{X|Y}(x|y) p_Y(y) = E[E[X | Y]].$$

În cazul variabilelor continue, *media condiționată a lui X* dat fiind $Y = y$ este dată de formula:

$$E[X | Y = y] = \int_{-\infty}^{\infty} x \frac{f_{XY}(x, y)}{f_Y(y)} dx,$$

și deci:

$$E[X] = \int_{-\infty}^{\infty} x f_X(x) dx = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x f_{X|Y}(x | y) f_Y(y) dx dy = E[E[X | Y]].$$

Tot în acest context vom aminti și următoarele formule de calcul. Astfel, mediile variabilelor X și Y aparținând cuplului (X, Y) sunt date de:

$$m_X = E[X] = \sum_i \sum_j x_i p_{ij}, \quad m_Y = E[Y] = \sum_i \sum_j y_j p_{ij},$$

unde $p_{ij} = P\{X = x_i, Y = y_j\}$ – cazul discret.

În cazul continuu, avem formulele:

$$m_X = E[X] = \iint x f_{XY}(x, y) dx dy, \quad m_Y = E[Y] = \iint y f_{XY}(x, y) dx dy.$$

În ceea ce privește dispersiile, acestea sunt date de formulele (cazul discret):

$$D^2(X) = \sum_i \sum_j (x_i - m_X)^2 p_{ij}, \quad D^2(Y) = \sum_i \sum_j (y_j - m_Y)^2 p_{ij},$$

respectiv (cazul continuu):

$$D^2(X) = \iint (x - m_X)^2 f_{XY}(xy) dx dy, \quad D^2(Y) = \iint (y - m_Y)^2 f_{XY}(xy) dx dy$$

Exemple:

1) Să considerăm un cuplu (X, Y) de variabile discrete, având tabloul repartiției comune dat de:

		1	2	3
		1	2	3
X	1	1/18	1/12	1/36
	2	1/9	1/6	1/18
		1/6	1/4	1/12

Rezultă că mediile celor două variabile aleatoare sunt:

$$m_x = 2 \frac{1}{3}, \quad m_y = 1 \frac{5}{6},$$

punctul $\left(2 \frac{1}{3}, 1 \frac{5}{6}\right)$ numindu-se centrul de dispersie al cuplului (X, Y) .

Dispersiile corespunzătoare sunt $D^2(X) = \frac{5}{9}$ și $D^2(Y) = \frac{17}{36}$.

2) Dacă un cuplu de variabile continue urmează o lege de probabilitate având densitatea comună dată de:

$$f(x, y) = \begin{cases} \frac{1}{2} \cdot \sin(x + y), & (x, y) \in D = \{0 \leq x \leq \frac{\pi}{2}, 0 \leq y \leq \frac{\pi}{2}\}, \\ 0, & (x, y) \notin D \end{cases}$$

atunci:

$$m_x = \frac{\pi}{4}, \quad m_y = \frac{\pi}{4}, \quad D^2(X) = \frac{\pi^2 + 8\pi - 32}{16}, \quad D^2(Y) = \frac{\pi^2 + 8\pi - 32}{16}$$

O mare parte a studiilor statistice uzuale se ocupă cu analiza relației între două variabile statistice (attribute) ce corespund aceluiași grup de obiecte/instanțe. Cel mai popular exemplu se referă la relația ce există între înălțimea și greutatea unui individ, ce aparține unor anumite grupuri populaționale. Pentru a o identifica, se studiază relația dintre cele două caracteristici/attribute măsurate pe obiectele dintr-un anumit set. Cu alte cuvinte, este vorba de două serii statistice în care cuplurile de valori (x_i, y_i) , corespunzând cuplului de variabile statistice (X, Y) , sunt măsurate pe același obiect.

Există două mari motive pentru care se efectuează un asemenea studiu:

- ⇒ Descrierea relației care ar putea exista între cele două variabile, analizând legătura între cele două serii de observații. Concret, se analizează dacă tendința ascendentă a uneia implică o tendință ascendentă, descendenta sau nici o tendință a celeilalte;
- ⇒ În ipoteza existenței unei legături reale între ele, identificată în primă instanță, să se poată prognostica valorile uneia în raport cu valorile celeilalte, pe baza ecuației ce stabilește legătura dintre ele.

Așa cum se observă din cele spuse mai sus, scopul final este *prognoza*, în condiția că este posibilă, cele două variabile fiind într-adevăr în conexiune. Metoda prin care analizăm posibilele asociații între valorile a două variabile statistice, prelevate de la același grup de obiecte, este cunoscută ca metoda

corelației și are ca indice coeficientul de corelație. Coeficientul de corelație poate fi calculat pentru orice set de date, dar, pentru ca el să aibă relevanță statistică, trebuie îndeplinite două condiții majore:

- (a) cele două variabile să fie definite de același lot de obiecte, cuplurile de date corespunzând aceluiași obiect;
- (b) cel puțin una din variabile să aibă o repartiție aproximativ normală, ideal fiind ca ambele să fie normal repartizate.

Dacă datele nu au o repartiție normală (cel puțin una din variabile) se procedează fie la transformarea lor pentru normalizare, fie la considerarea unor coeficienți de corelație neparametриci [5].

Așa cum am menționat la început, în cazul unui cuplu (X, Y) de variabile aleatoare, suntem interesați de modul cum putem identifica o eventuală legătură între componentele cuplului prin intermediul mediei și dispersiei. Astfel, teoretic, covarianța variabilelor X și Y este dată de formula:

$$\text{cov}(X, Y) = E[(X - E[X])(Y - E[Y])] = E[XY] - E[X]E[Y],$$

iar raportul:

$$r = r(X, Y) = \frac{\text{cov}(X, Y)}{\sqrt{D^2(X)D^2(Y)}},$$

se numește *coeficient de corelație* (Pearson product-moment correlation coefficient r – valoare introdusă de sir F. Galton). Așa cum arată și numele său, coeficientul de corelație ne dă o „măsură” a legăturii (corelației) dintre cele două variabile, el putând fi considerat ca o „intensitate” a relației (liniare) dintre ele.

Din punctul de vedere al Statisticii aplicate, formulele de mai sus capătă următoarele forme. Să considerăm deci două serii statistice $\{x_i\}_{i=1,\dots,n}$ și $\{y_i\}_{i=1,\dots,n}$, corespunzătoare cuplului de variabilele statistice X și Y . Atunci, formula covarianței celor două variabile este dată de:

$$\text{cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n x_i y_i - \bar{x}\bar{y}.$$

Coeficientul de corelație r al celor două variabile este un număr real cuprins între -1 și 1 , definit de formula:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}},$$

valoarea sa putând fi privită ca o măsură a elongației elipsei formată de norul de puncte din diagrama de împrăștiere (a se vedea subparagraful următor). Pentru calcule concrete se folosește formula de mai sus, scrisă sub forma:

$$r = \frac{\sum x_i y_i - (\sum x_i)(\sum y_i)/n}{\sqrt{\left[\sum x_i^2 - (\sum x_i)^2/n \right] \left[\sum y_i^2 - (\sum y_i)^2/n \right]}}.$$

Plecând de la faptul că variabila aleatoare:

$$z = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right),$$

este normal repartizată, rezultă că intervalul de încredere 95% pentru variabila z are forma (z_1, z_2) , unde:

$$z_1 = z - 1,96/\sqrt{n-3}, \quad z_2 = z + 1,96/\sqrt{n-3},$$

de unde rezultă, aplicând transformarea inversă, că intervalul de încredere 95% pentru r este dat de:

$$\left(\frac{e^{2z_1} - 1}{e^{2z_1} + 1}, \quad \frac{e^{2z_2} - 1}{e^{2z_2} + 1} \right).$$

Să interpretăm acum corelația dintre cele două variabile statistice. Așa cum am spus mai înainte, coeficientul de corelație r (Pearson) ia valori cuprinse între -1 și $+1$, trecând și prin 0 , care indică o asociație neliniară între cele două variabile (independență liniară). O valoare a lui r apropiată de -1 indică o corelație negativă puternică, adică tendința unei variabile de a scădea semnificativ când cealaltă variabilă crește, în timp ce o valoare a lui r apropiată de $+1$ indică o corelație pozitivă puternică, adică tendința de creștere semnificativă a unei variabile atunci când și cealaltă variabilă crește. Să notăm că există cazuri în care variabile dependente au coeficientul de corelație nul. Problema care se pune este stabilirea unui prag de la care să putem trage concluzia că cele două variabile sunt într-adevăr corelate. În acest sens se sugerează fie un prag definit de $|r|/\sqrt{n-1} \geq 3$, de la care se poate considera că legătura dintre cele două variabile este suficient de probabilă, fie utilizarea nivelului de semnificație statistică p , asociat calculării coeficientului r . Să notăm că, dacă în trecut, când nu existau computerele și nici programele statistice corespunzătoare, se folosea pragul de mai sus, astăzi este folosit exhaustiv doar nivelul de semnificație statistică p .

Remarcă: În profida celor expuse mai sus, nu trebuie pierdut din vedere faptul că un coeficient de corelație important nu implică totdeauna în mod necesar o legătură naturală, intrinsecă, între atributele ce definesc

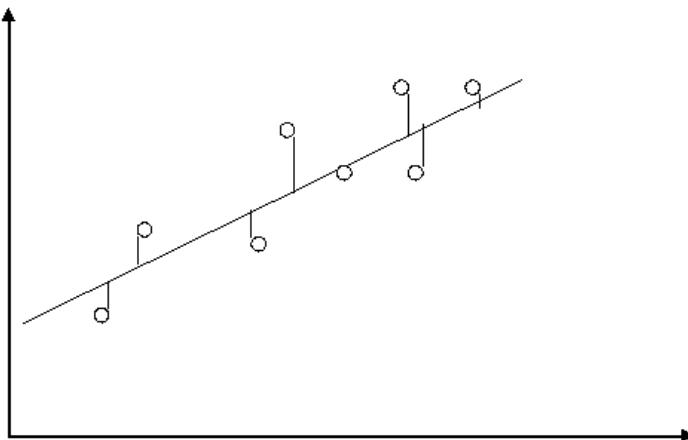
cele două variabile statistice analizate. Sunt cazuri în medicină, de exemplu, când valori mari ale coeficientului de corelație, indicând o corelație statistică semnificativă, nu au nici o relevanță medicală și invers. De exemplu, aceiași valoare redusă a coeficientului de corelație poate fi importantă în epidemiologie dar nesemnificativă din punct de vedere clinic [5]. În concluzie, coeficientul de corelație este o măsură a legăturii liniare „aritmetice” dintre cele două variabile, care poate fi câteodată și întâmplătoare, fără relevanță reală. Acest fapt trebuie avut în vedere mai ales în DM, unde nu există totdeauna cunoștințe anterioare bine structurate despre fenomenul analizat.

Presupunând că legătura dintre cele două variabile X și Y , reliefată de coeficientul de corelație r , nu este întâmplătoare, există trei explicații posibile:

- Variabila X influențează (cauzează) variabila Y ;
- Variabila Y influențează variabila X ;
- Ambele variabile X și Y sunt influențate de același fenomen din fundal.

Notă: Atunci când nu există informații suplimentare despre contextul în care acționează cele două variabile, mai ales cazul studiilor DM, este nerealist să folosim statistică pentru a valida una din cele trei ipoteze, fără o analiză alternativă.

Pasul următor în analiza legăturii dintre două variabile statistice, atunci când acestea sunt corelate, este să se stabilească concret natura legăturii liniare dintre ele, descriind-o printr-o ecuație matematică. Scopul final al acestei abordări este prognoza valorilor uneia dintre variabile pe baza valorilor celeilalte, prognoză efectuată pe baza ecuației ce descrie legătura dintre cele două seturi de date. Modul de prezentare a legăturii liniare dintre două variabile, atunci când aceasta există, se numește *metoda regresiei liniare* (*linear regression*). Pentru aceasta se consideră una dintre variabile ca *variabilă independentă* sau *variabilă predictor*, iar cealaltă variabilă ca *variabilă dependentă* sau *variabilă răspuns (outcome)*. Legătura liniară dintre cele două variabile este descrisă de o ecuație liniară, *ecuația de regresie* (*regression equation*), căreia îi corespunde geometric *dreapta de regresie* (*regression line*). Ca metodologie, variabila dependentă se distribuie pe axa ordonatelor, în timp ce variabila independentă se distribuie pe axa absciselor. Ecuația dreptei de regresie se stabilește pe baza metodei „celor mai mici pătrate” (*least squares method* -LSM) care, intuitiv, minimizează distanța între punctele reprezentate de perechile de date (*observed values*) și punctele corespunzătoare de pe dreaptă (*fitted values*), obținute pe verticalele corespunzătoare. Aceasta distanță se numește *reziduu* (*residual*) –vezi figura următoare.



Tehnic vorbind, se consideră aceste reziduuri la pătrat și se calculează suma lor. Dreapta pentru care se obține minimul sumei pătratelor este cea optimă, utilizând aceasta metodă. Metoda celor mai mici pătrate minimizează de asemenea și dispersia (varianța) reziduurilor. Această dispersie se numește dispersie reziduală (*residual variance*) și se folosește pentru a măsura ‘eficiența potrivirii’ (*goodness-of-fit*) dată de dreapta de regresie.

În final, obținem ecuația de regresie sub forma:

$$Y = a + b \cdot X,$$

unde a se numește *interceptor* iar b *coeficient de regresie*, cei doi parametri fiind obținuți cu ajutorul formulelor:

$$b = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad a = \bar{y} - b \bar{x}.$$

Din punct de vedere practic, se pot construi cu ușurință programe de calcul pentru dreapta de regresie, folosind formulele echivalente:

$$\sigma_{xx} = \sum x_i^2 - (\sum x_i)^2 / n,$$

$$\sigma_{yy} = \sum y_i^2 - (\sum y_i)^2 / n,$$

$$\sigma_{xy} = \sum x_i y_i - \sum x_i \sum y_i / n,$$

$$\text{de unde obținem } b = \frac{\sigma_{xy}}{\sigma_{xx}}.$$

Remarcă: În cazul în care legătura dintre cele două variabile nu este liniară (ca cea prezentată mai sus) și totuși bănuim că există un anumit tip de conexiune între ele, putem utiliza o *regresie neliniară* (e.g. regresia polinomială). Atunci, în loc de a găsi dreapta de regresie, se găsește curba respectivă de regresie.

Notă: Dacă se consideră perechi de date provenind de la două grupuri diferite de obiecte și având aceleași semnificații, putem folosi dreptele de regresie calculate pentru fiecare grup pentru a compara cele două grupuri. Dacă, de exemplu, cele două drepte de regresie au aproximativ aceeași pantă b (sunt paralele), atunci putem considera diferența pe axa verticală (y) ca fiind diferența între mediile variabilei Y între cele două grupuri, observație ce este apoi urmată de o testare a semnificației statistice a diferenței. O asemenea analiză statistică face parte dintr-un studiu statistic mai vast care se numește *analiza covariantei*.

Exemple:

1) Să considerăm datele culese de la un lot de 24 de pacienți având diabet de tip I, privind următoarele două variabile [5]:

- glucoza (G) în sânge pe stomacul gol (mmol/l);
- viteza medie de contractie Vcf (%/sec) a ventriculului stâng, obținută prin eco-cardiografie.

Pacient #	G	Vcf	Pacient #	G	Vcf
1	15,3	1,76	13	19,0	1,95
2	10,8	1,34	14	15,1	1,28
3	8,1	1,27	15	6,7	1,52
4	19,5	1,47	16*	8,6	*
5	7,2	1,27	17	4,2	1,12
6	5,3	1,49	18	10,3	1,37
7	9,3	1,31	19	12,5	1,19
8	11,1	1,09	20	16,1	1,05
9	7,5	1,18	21	13,3	1,32
10	12,2	1,22	22	4,9	1,03
11	6,7	1,25	23	8,8	1,12
12	5,2	1,19	24	9,5	1,70

Așa cum se observă din tabelul de mai sus, pacientul cu numărul 16 nu are specificată variabila Vcf. Putem, pe baza datelor de mai sus, utilizând funcția predictivă a regresiei liniare, să prognozăm valoarea Vcf corespunzătoare, mult mai dificil de obținut direct (cu aparatul medicală), pe baza valorii glucozei în sânge.

Tabelul de mai jos prezintă principalele caracteristici numerice ale regresiei liniare aplicate în acest caz.

Variabila	Media	Deviația standard	r	Nivel semnificație p
G	10,30	4,34		
Vcf	1,32	0,23	0,42	0,041

Reprezentarea grafică de mai jos ilustrează diagrama împrăștierii, dreapta de regresie și intervalul de încredere 95% pentru media Vcf.

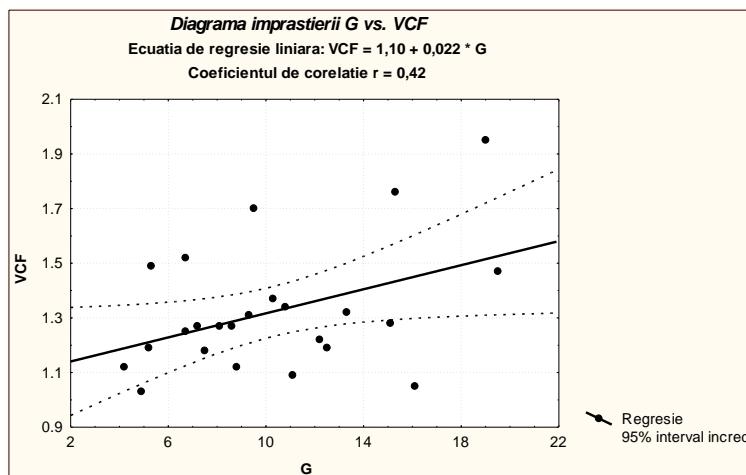
Așa după cum bine se observă, în ciuda faptului că valoarea coeficientului de corelație r nu pare prea importantă, totuși nivelul de semnificație $p = 0,041$ atestă o corelație statistic semnificativă ($< 0,05$). Ecuația de regresie liniară este dată de:

$$Vcf = 1,10 + 0,02 \cdot G,$$

de unde deducem că valoarea estimată (prognozată pe baza regresiei liniare) a variabilei Vcf pentru pacientul No. 16 este de 1,27%.

Să mai observăm că pe dreapta de regresie găsim valorile medii proгnozate ale variabilei Vcf în funcție de glucoză.

În figura de mai jos este reprezentat graficul dreptei de regresie.

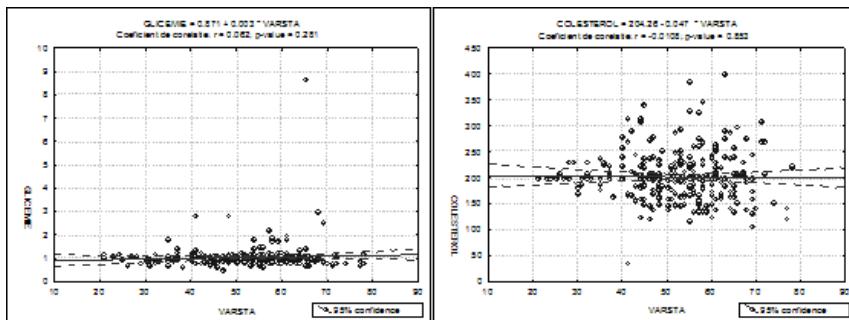


Remarcă: Să vedem cum se folosește intervalul de încredere 95% în acest caz.

Pentru aceasta să alegem o valoare (fixată) a variabilei G (glucoză) pe axa absciselor. Dacă apoi o perpendiculară pe această axă în punctul fixat și considerăm intervalul (măsurat pe axa ordonatelor) definit de intersecțiile perpendicularării cu cele două curbe punctate (una inferioară și alta superioară) ce definesc intervalul de încredere. Rezultă că orice valoare din acest interval este o posibilă estimare a

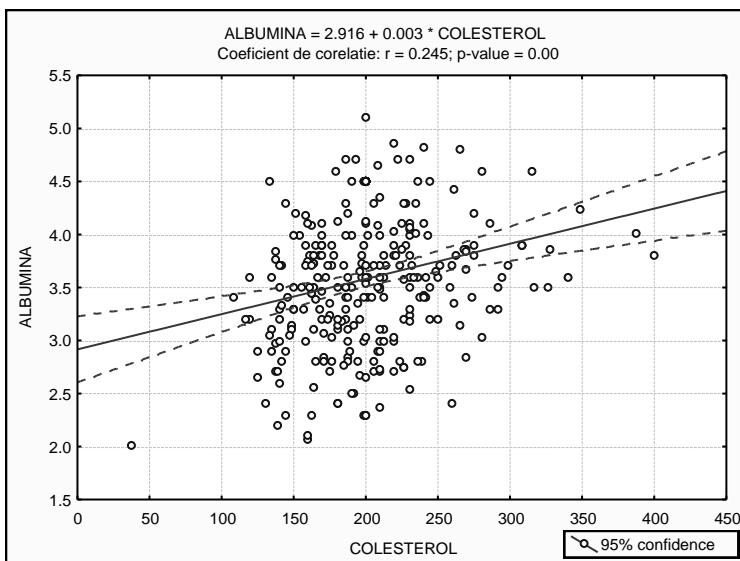
medie corespunzătoare a Vcf cu o probabilitate de 95%, punctul de mijloc al acestui interval fiind chiar pe dreapta de regresie și fiind ales (echiprobabil între celelalte valori ale intervalului) ca valoare standard pentru media căutată. De exemplu, pentru valoarea fixată a glucozei $G = 10$, obținem intervalul de încredere 95% pentru Vcf definit de $(1,22, 1,40)$ cu punctul mediu pe dreapta de regresie de 1,31. Rezultă deci că un pacient cu diabet de tip I, având valoarea G egală cu 10, ar trebui să aibă o valoare a Vcf (estimată) cuprinsă între 1,22 și 1,40 cu aceeași probabilitate de 95%. În acest caz, se alege ca valoare standard media celor două valori, care se găsește pe dreapta de regresie, adică 1,31.

2) Să considerăm cazul lotului de 299 subiecți analizați prin prisma bolilor hepatice și să considerăm ca variabile pentru studiul regresiei vârstă, pe de-o parte și valorile glicemiei și colesterolului, pe de altă parte. Mai întâi, vom investiga eventualele legături între vârstă pacienților și valorile corespunzătoare ale glicemiei și colesterolului, după care vom cerceta eventuala dependență între valorile glicemiei și ale colesterolului. Pentru a ușura interpretarea datelor, vom prezenta doar vizualizările rezultatelor obținute în cele trei analize ale regresiilor corespunzătoare, împreună cu datele numerice adiacente.

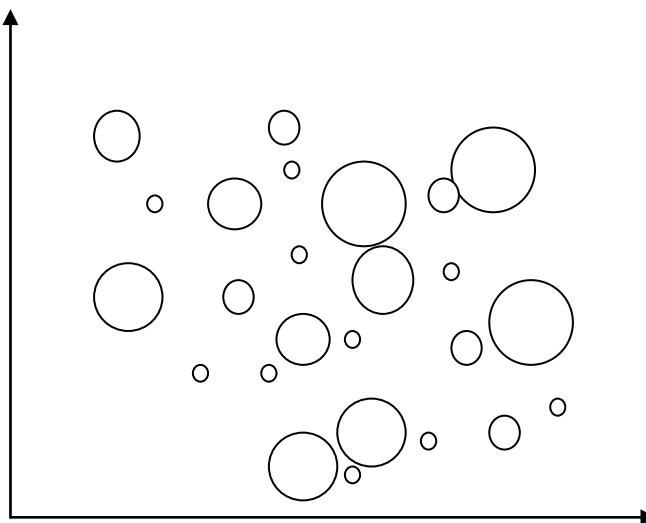


Așa cum se observă din cele două figuri de mai sus, nici glicemia nici colesterolul nu sunt corelate cu vârstă (drepte de regresie orizontale), cu menținerea că gruparea datelor este mai pronunțată pentru glicemie, ilustrată și de valoarea p mai redusă (0,28 față de 0,85).

În schimb, dacă considerăm valorile pentru colesterol și albumină, se observă în figura de mai jos o corelație înaltă (valoarea $p = 0,0$), în ciuda faptului că „norul” punctelor corespunzătoare datelor este destul de „împrăștiat”.



Remarcă: Observarea a două variabile (attribute) cantitative X și Y , corespunzătoare acelorași obiecte dintr-un set de date, conduce la reprezentarea grafică a unor cupluri de valori (x_i, y_i) relative la obiectul i . Geometric vorbind, cuplurile de valori (x_i, y_i) reprezintă un „nor” de puncte în sistemul de axe xOy . În această reprezentare grafică din exemplu, alegerea unităților de măsură pentru fiecare din cele trei variabile are importanță sa, deoarece de această alegere depinde cât de „alungit” este norul punctelor. Dificultatea apare atunci când multe dintre puncte se „confundă”, valorile fiind foarte apropiate. În acest caz se grupează datele, definind p clase pentru variabila X și q clase pentru variabila Y . Se construiește apoi un tablou de elemente $[a_{jk}]$, unde a_{jk} reprezintă numărul de cupluri astfel încât x_j și y_k aparțin claselor j și k , obținute din regruparea datelor inițiale ale variabilelor X și Y . Un astfel de tablou se numește tablou de *contingență*, fiind utilizat atât la definirea unor parametri cantitativi, cât și la reprezentarea grafică a datelor grupate, așa cum se vede în figura de mai jos. Se observă că suprafața fiecărui cerculeț este proporțională cu numărul cuplurilor corespunzătoare grupurilor.



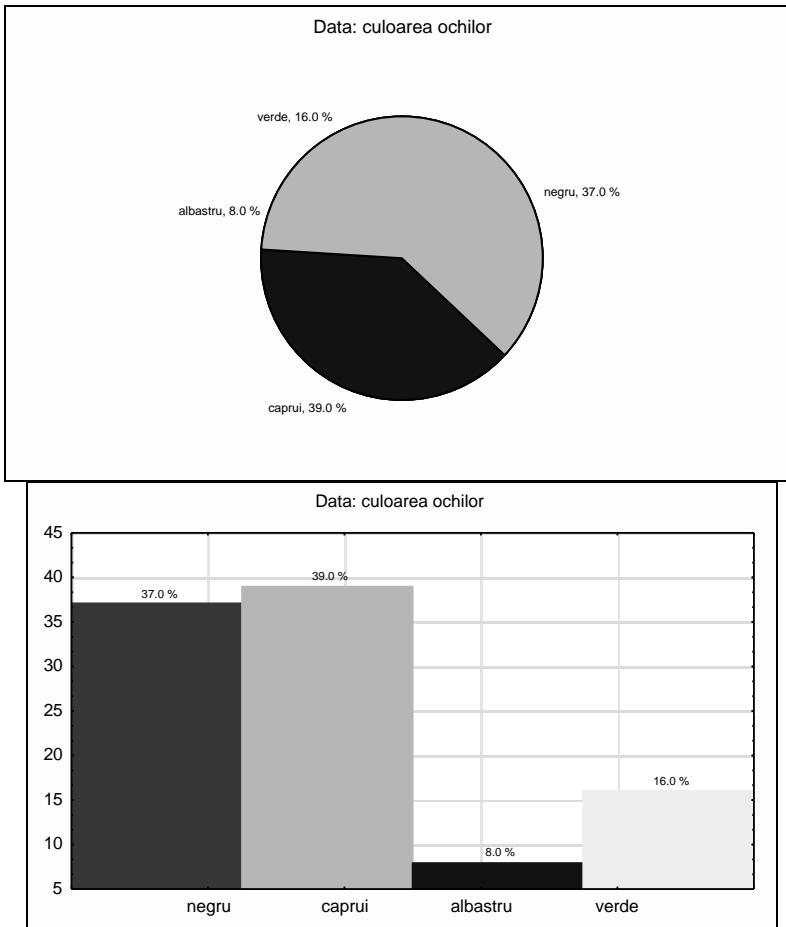
3.2.3. Reprezentarea grafică a unei mulțimi de date

Reprezentarea grafică elementară a datelor înseamnă conversia datelor în format vizual sau tabular simplu, astfel încât să se poată analiza și raporta rapid caracteristicile datelor, precum și relațiile existente între atrbute.

Reprezentările grafice aferente unui set de date (atribute ale unei instanțe în cazul de față) depind de natura atrbutorilor: quantitative sau qualitative.

Să considerăm mai întâi cazul datelor qualitative. Astfel de date pot fi reprezentate grafic cu ajutorul diferitelor diagrame formate din bastoane verticale sau orizontale, cercuri, elipse etc., bi- sau tri-dimensionale, plecând de la partitura setului de date pe care atrbutul o induce. Acest tip de vizualizare depinde de modalitatea aleasă. Astfel, în cazul utilizării cercurilor, elipselor etc. – reprezentarea circulară – întreaga mulțime de obiecte este reprezentată de cerc, elipsă etc.; fiecare atrbut al obiectelor este reprezentat printr-un sector circular a cărui suprafață este proporțională cu numărul obiectelor având atrbutul respectiv (sau cu procentajul corespunzător). Cealaltă modalitate de reprezentarea a unui atrbut qualitativ – cel al diagramelor cu bastoane – se raportează la o vizualizare într-un sistem de axe: pe axa absciselor apar atrbutele iar pe axa ordonatelor apare numărul obiectelor cu atrbutul respectiv (sau procentajul corespunzător). Prezentăm, mai jos, un astfel de mod de vizualizare. Să considerăm, de exemplu, că avem o anumită populație și suntem interesați de culoarea ochilor indivizilor – atrbut qualitativ. Presupunem că mulțimea colorilor ochilor este *Culoare_ochi* = {negru, albastru, verde, căprui} și că din studiul efectuat a rezultat că 37% din populație are ochii negri, 39% sunt ochii căprui, 8% sunt ochii albaștri și 16% sunt ochii verzi. Vom prezenta

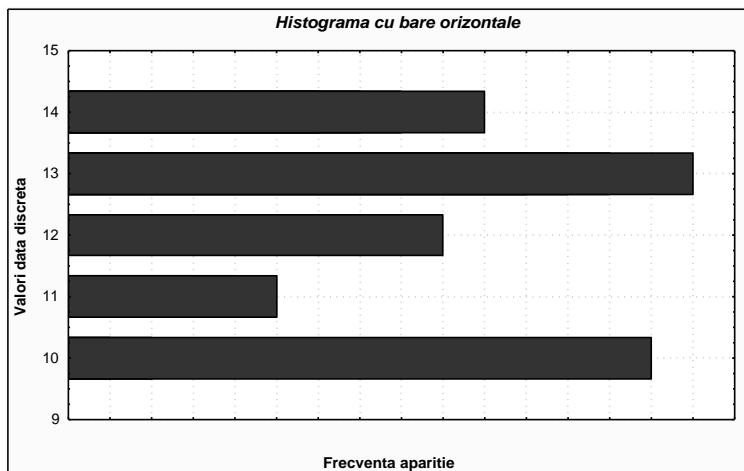
două tipuri de diagrame pentru acest caracter: reprezentarea circulară (numită și ‘pie’ ~ plăcintă în lb. Engleză) și reprezentarea prin bastoane dreptunghiulare. În fiecare caz în parte am reprezentat (fie prin sectoare circulare, fie prin bastoane/coloane) frecvența de apariție a diferitelor tipuri de culori ale ochilor în populația respectivă.



Remarcă: Vom remarca, în acest context, că se mai folosesc pentru aceste diagrame și termenul de histogramă, cu toate că, în principiu, reprezentarea prin histograme se referă la reprezentarea grafică a frecvențelor tabulate – cu referire directă la datele (attributele) cantitative. Matematic vorbind, *histograma* reprezintă aplicația ce produce numărul de observații (valori ale datelor) ce aparțin unor

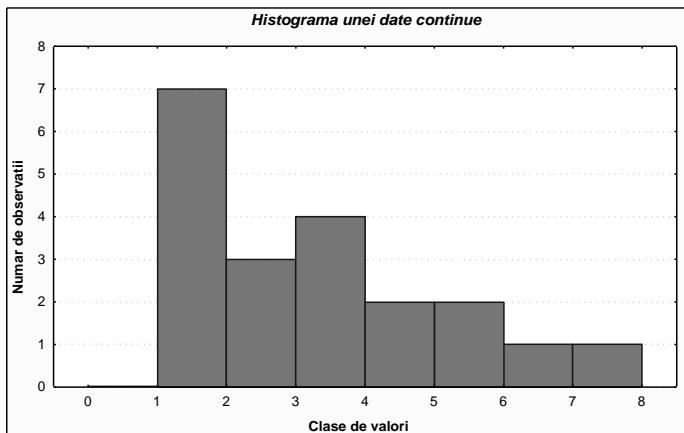
anumite intervale (echivalent, frecvența observațiilor aparținând intervalelor de valori). Vizual, prin histogramă (bidimensională) de exemplu, se înțelege o reprezentare grafică a acestei aplicații, adică a repartiției numărului de valori (frecvențelor) unui anumit atribut numeric în care fiecare baston (coloană) reprezintă un anumit interval de valori ale atributului iar înălțimea sa este proporțională cu numărul (frecvența) valorilor din intervalul respectiv. Termenul a fost introdus de Pearson -1895.

Să considerăm acum cazul atributelor numerice, cantitative. La reprezentarea grafică în cazul unor date numerice, întâlnim cele două moduri de reprezentare grafică prin histograme, corespunzătoare felului datelor: discrete sau continue. În cazul datelor discrete, reprezentarea grafică este asemănătoare cazului datelor calitative, cu toate că există diferență conceptuală subliniată în remarcă de mai sus. În acest caz, dacă considerăm diagramele cu bastoane, lungimea acestora are o semnificație numerică precisă. Concret, în histograma de mai jos, pe axa ordonatelor sunt reprezentate valorile variabilei (datei) discrete, în timp ce pe axa absciselor este reprezentată frecvența relativă a apariției fiecărei valori. Este ceea ce numim o *histogramă* a frecvenței relative cu bastoane orizontale.

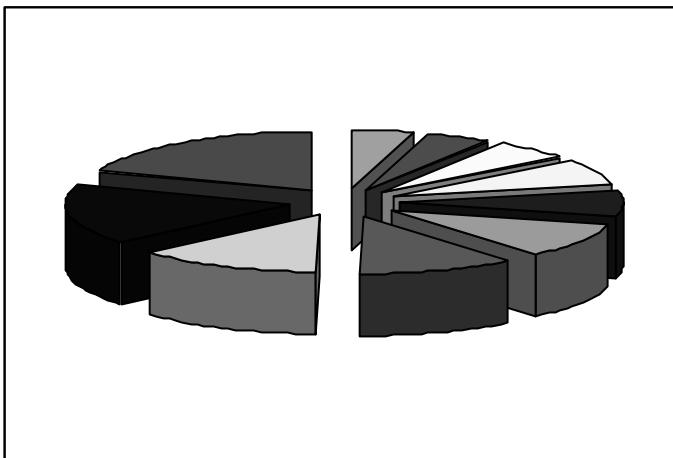


Problema se complică atunci când este vorba de date numerice continue. Aici, pentru trasarea unei histograme, este necesară împărțirea (gruparea) datelor numerice în anumite clase (de regulă intervale), cărora să le corespundă pe cealaltă axă frecvența relativă de apariție (sau numărul de observații), corespunzătoare fiecărei clase. În histograma de mai jos, pe axa absciselor sunt reprezentate clasele (i.e. intervale de valori) iar pe cea a ordonatelor este reprezentat numărul corespunzător de observații.

Să remarcăm că, înainte de construirea histogramelor, trebuie calculate toate înălțimile băstoanelor (coloanelor) reprezentând frecvențele, pentru a identifica scara optimă a valorilor acestora, astfel încât histograma respectivă să fie într-adevăr utilă din punct de vedere vizual.



Există și în acest caz posibilitatea reprezentării cu diagrame circulare (2D sau 3D) așa cum se poate observa în figura de mai jos.



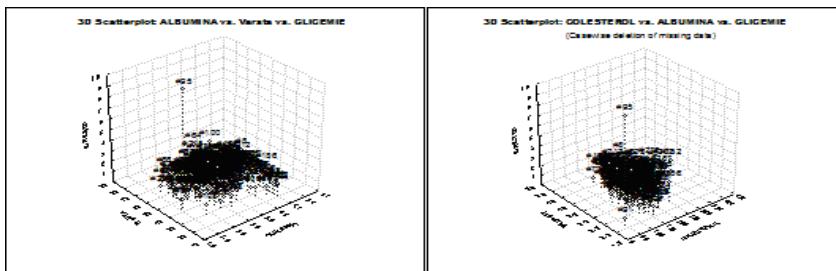
În ceea ce privește cuplurile de date, așa cum am specificat mai sus, datele sunt reprezentate fie individual prin „norul” de puncte corespunzătoare, fie prin datele grupate (tabloul de contingență).

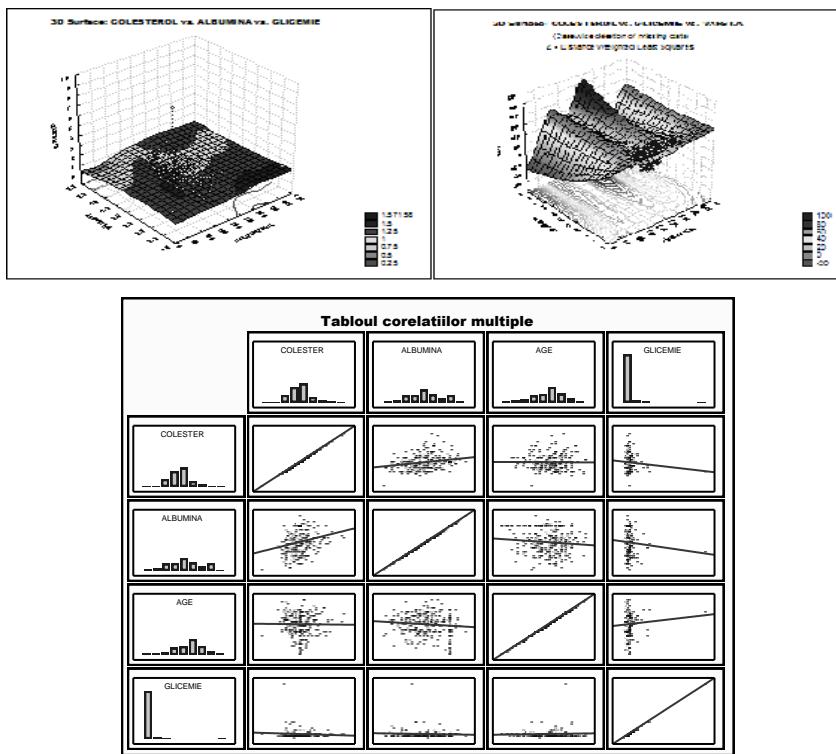
3.3. Analiza matricei corelațiilor

Dacă în cazul unei serii de cupluri au fost luate în considerație cupluri formate din două variabile (attribute), în cazul matricei corelațiilor putem lua în considerație *tuple* de variabile $X_1, X_2, \dots, X_k, k > 2$. La fel ca și în cazul cuplurilor de variabile, putem analiza corelațiile între fiecare două perechi de variabile, precum și „norul” împrăștierii lor și dreptele corespunzătoare de regresie liniară. Avantajul este că rezultatele numerice, precum și reprezentările grafice sunt prezentate grupat (matricea corelațiilor, respectiv diagramele de împrăștiere multiple).

Prezentăm mai jos, o asemenea analiză multiplă (multivariată) în cazul datelor privind bolile hepatice, pentru attributele vârstă, colesterol albumină și glicemie. Așa cum am menționat mai sus, avantajul prezentării legăturii între attributele obiectelor cu ajutorul matricei corelațiilor și nu cu corelațiile fiecărei perechi de attribute în parte constă în faptul că astfel avem o privire de ansamblu asupra tuturor conexiunilor între attributele analizate. De asemenea, „norul” punctelor reprezentate, ca și alte reprezentări grafice uzuale (e.g. suprafete, histograme tridimensionale etc.) sunt mult mai sugestive în cazul reprezentării colective a datelor decât luate separat (vezi figurile de mai jos).

	COLESTEROL	ALBUMINA	VÂRSTA	GLICEMIE
COLESTEROL	$r=1.00$	$r=0.245$	$r=-0.010$	$r=-0.091$
	$p=ns$	$p=0.00$	$p=0.853$	$p=0.115$
ALBUMINA	$r=0.2450$	$r=1.00$	$r=-0.098$	$r=-0.073$
	$p=0.00$	$p=ns$	$p=0.091$	$p=0.204$
VÂRSTA	$r=-0.010$	$r=-0.098$	$r=1.00$	$r=0.062$
	$p=0.853$	$p=0.091$	$p=ns$	$p=0.281$
GLICEMIE	$r=0.091$	$r=-0.073$	$r=0.062$	$r=1.00$
	$p=0.115$	$p=0.204$	$p=0.281$	$p=ns$

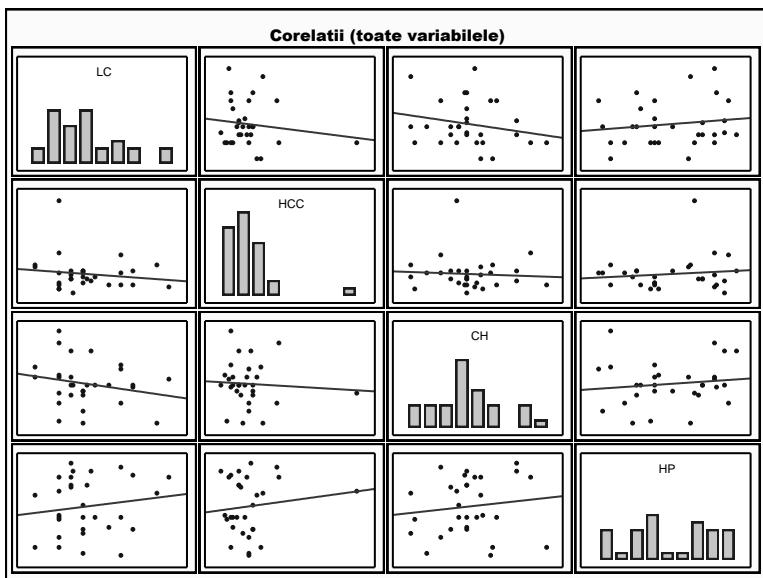




Matricea corelațiilor este de asemenea utilă și în cazul analizei unor caracteristici (attribute) ce corespund la obiecte din categorii diferite. De exemplu, în cazul datelor medicale privind bolile hepaticе, putem fi interesați în stabilirea eventualelor corelații privind valoarea unor parametri medicali importanți, ca glicemia și colesterolul, pentru diferitele tipuri de boli. O astfel de analiza dezvăluie legături ascunse ce pot exista între același tip de attribute, dar aparținând la obiecte diferite. Pentru cazul de mai sus, prezentăm o astfel de analiză referitoare la valorile colesterolului și glicemiei între cele patru clase de diagnostice: ciroză hepatică (LC), cancer hepatic (HCC), hepatită cronică de tip C (CH) și persoane sănătoase (HP). Tabelul de mai jos prezintă matricea corelațiilor multiple atât în cazul colesterolului (stânga) cât și al glicemiei (dreapta). De notat cazul colesterolului pentru persoanele sănătoase (HP) din lot (aceeași valoare indiferent de individ), ceea ce face ca în matricea corelațiilor corespunzătoare să apară notificarea „ns” (nesemnificativ).

	LC	HCC	CH	HP		LC	HCC	CH	HP
LC	<i>r</i> =1.00	<i>r</i> =0.019	<i>r</i> =-0.132	<i>r</i> =ns	LC	<i>r</i> =1.00	<i>r</i> =-0.143	<i>r</i> =0.218	<i>r</i> =0.145
	<i>p</i> =ns	<i>p</i> =0.918	<i>p</i> =0.485	<i>p</i> =ns		<i>p</i> =ns	<i>p</i> =0.450	<i>p</i> =0.246	<i>p</i> =0.442
HCC	<i>r</i> =0.019	<i>r</i> =1.00	<i>r</i> =-0.078	<i>r</i> =ns	HCC	<i>r</i> =-0.143	<i>r</i> =1.00	<i>r</i> =-0.067	<i>r</i> =0.122
	<i>p</i> =0.918	<i>p</i> =ns	<i>p</i> =0.679	<i>p</i> =ns		<i>p</i> =0.450	<i>p</i> =ns	<i>p</i> =0.724	<i>p</i> =0.520
CH	<i>r</i> =-0.132	<i>r</i> =-0.078	<i>r</i> =1.00	<i>r</i> =ns	CH	<i>r</i> =-0.218	<i>r</i> =-0.067	<i>r</i> =1.00	<i>r</i> =0.128
	<i>p</i> =0.485	<i>p</i> =0.679	<i>p</i> =ns	<i>p</i> =ns		<i>p</i> =0.246	<i>p</i> =0.724	<i>p</i> =ns	<i>p</i> =0.498
HP	<i>r</i> =ns	<i>r</i> =ns	<i>r</i> =ns	<i>r</i> =1.00	HP	<i>r</i> =0.145	<i>r</i> =0.122	<i>r</i> =0.128	<i>r</i> =1.00
	<i>p</i> =ns	<i>p</i> =ns	<i>p</i> =ns	<i>p</i> =ns		<i>p</i> =0.442	<i>p</i> =0.520	<i>p</i> =0.498	<i>p</i> =ns

În ceea ce privește ilustrarea grafică a corelațiilor de mai sus, vizualizarea următoare este foarte utilă, rezumând sugestiv toate informațiile numerice prezentate „arid” în tabel.

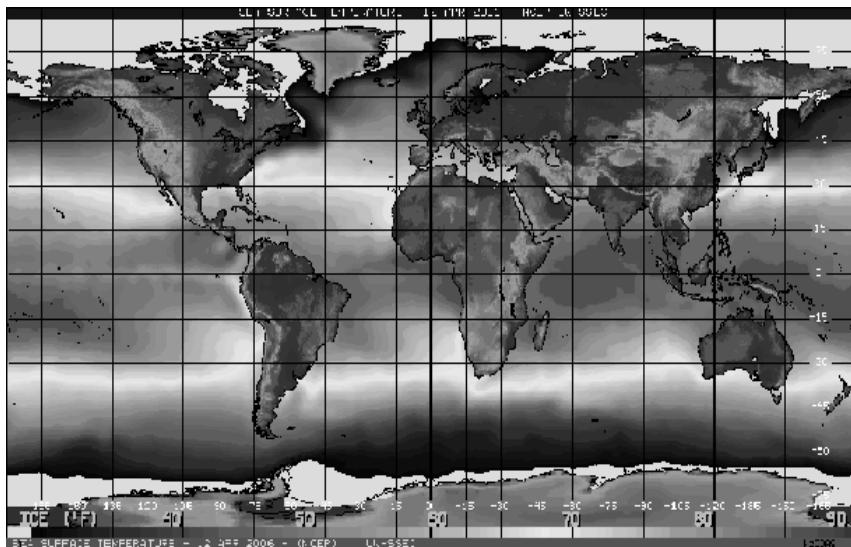


3.4. Vizualizarea datelor

Vizualizarea datelor reprezintă una dintre cele mai puternice și atrăgătoare tehnici de explorare a datelor, fiind unul dintre pilonii de baza ai EDA. Spre deosebire de reprezentarea grafică elementară, despre care am vorbit în paragraful anterior, atașată practic descrierii statistice, tehniciile de vizualizare utilizate în EDA fac apel la modalități mai sofisticate de prelucrare a datelor.

Vizualizarea datelor reprezintă prima luare de contact cu natura intimă a informației pe care încercăm să-o descifrăm în setul de date disponibile. Vizualizare face apel la puterea de pătrundere sintetică și capacitatea umană de a descifra și interpreta informații ascunse în imagini mai degrabă decât în

cifre seci. Mai jos este prezentată o imagine sugestivă privind repartiția căldurii la suprafața oceanelor (SST – *Sea Surfaces Temperatures*), imagine care „spune” dintr-o privire mult mai mult decât dacă ar fi în prealabil „digitalizată” și apoi tabulată.

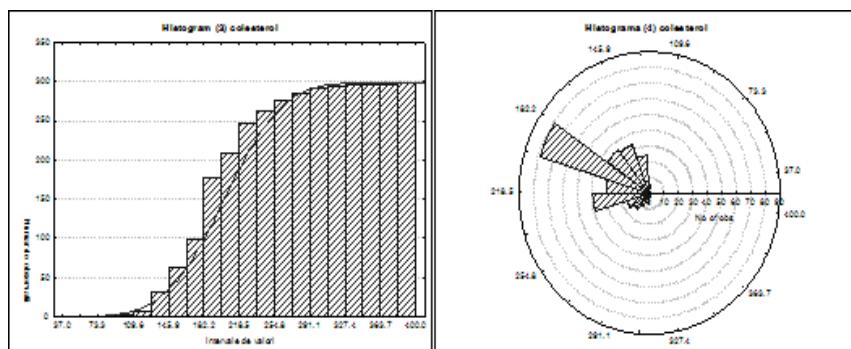
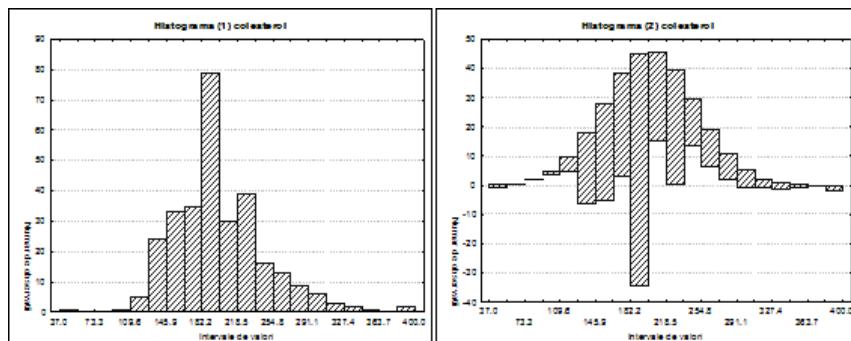


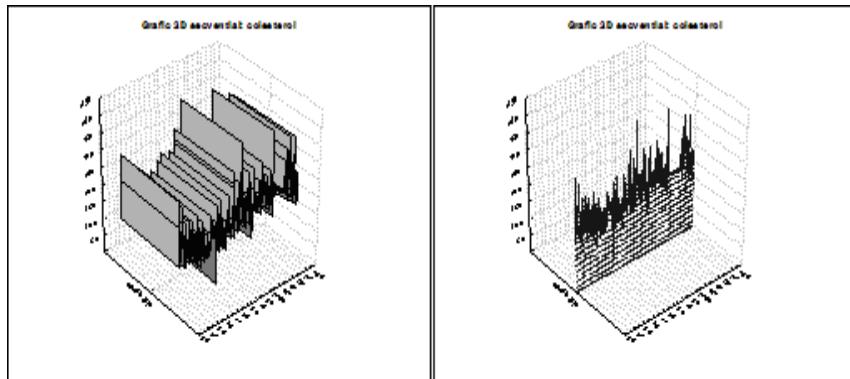
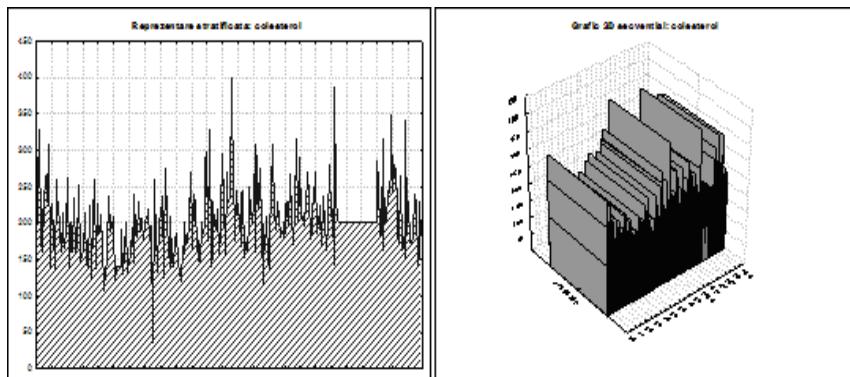
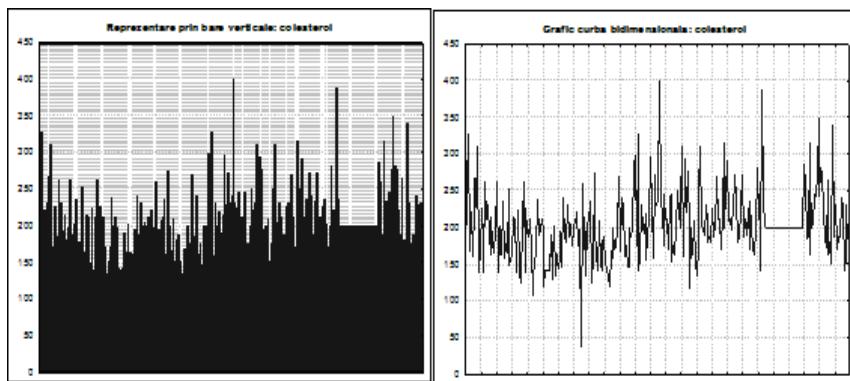
O altă tehnică simplă de vizualizare este reprezentată de „rearanjarea” într-o formă adecvată a datelor, rearanjare care relevă informații altfel ascunse privind legăturile între obiecte. De exemplu, să considerăm un set de obiecte care sunt analizate pe baza unui anumit criteriu ce furnizează valoarea 0 dacă obiectele respective nu verifică o anumită condiție, și valoare 1 dacă o verifică. Valorile pe care le furnizează criteriul pentru fiecare pereche de obiecte sunt reprezentate, așa cum se observă mai jos, sub formă numerică, în două tablouri 2×2 , obiectele fiind considerate în aranjamentul inițial – tabloul din stânga – și grupate în funcție de îndeplinirea condiției criteriului – tabloul din dreapta. Este ușor de observat ca rearanjarea din tabloul din dreapta furnizează informații valoroase privind clusterizarea obiectelor în subgrupuri distințe, informații altfel ascunse în tabloul inițial.

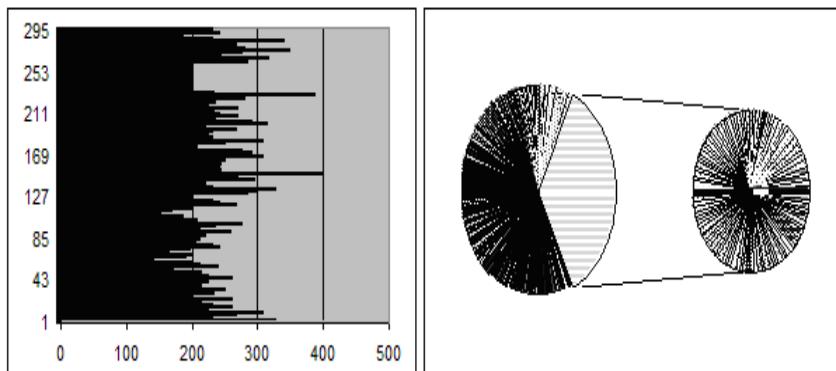
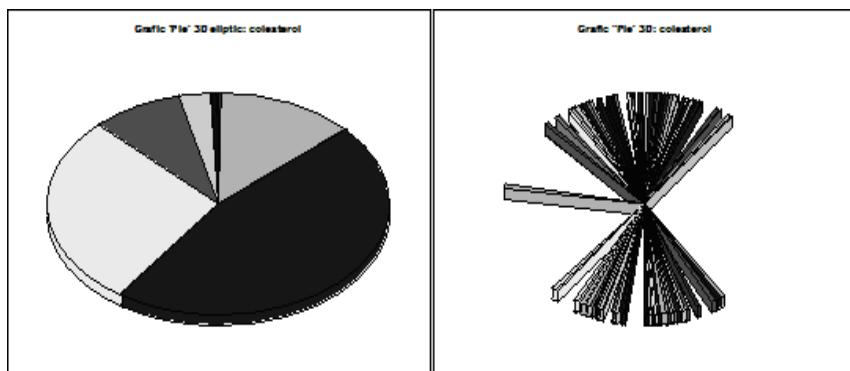
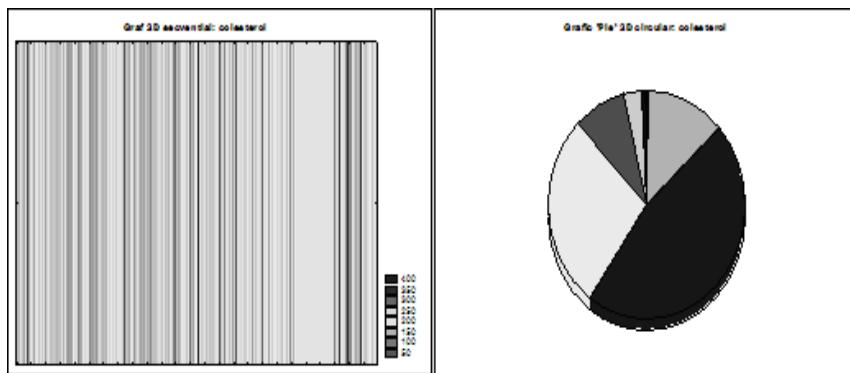
	1	2	3	4	5	6
1	0	1	0	1	1	0
2	1	0	1	0	0	1
3	0	1	0	1	1	0
4	1	0	1	0	0	1
5	0	1	0	1	1	0
6	1	0	1	0	0	1
7	0	1	0	1	1	0
8	1	0	1	0	0	1
9	0	1	0	1	1	0

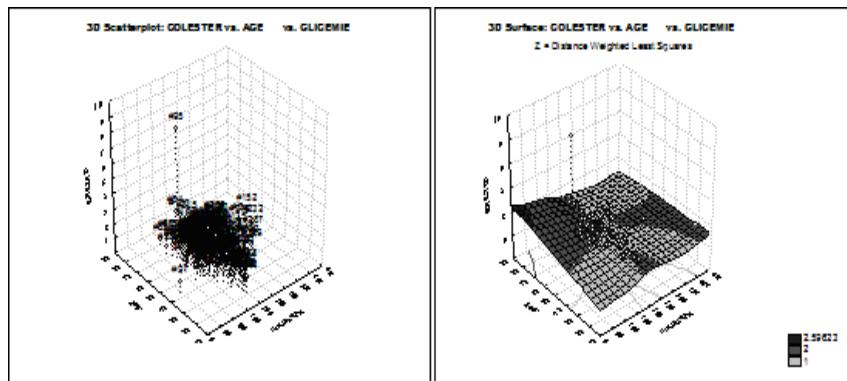
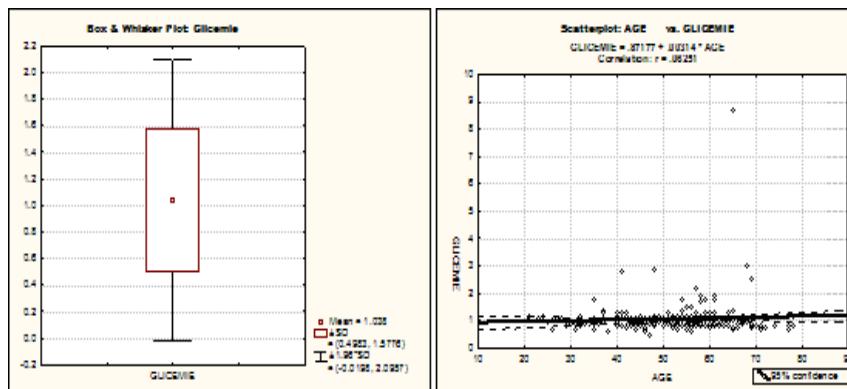
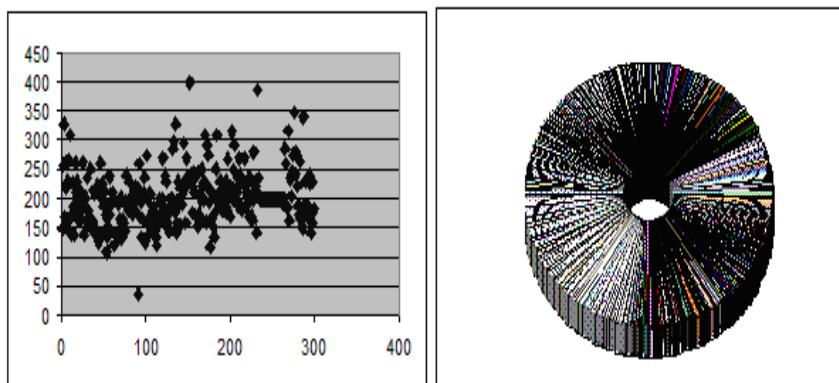
	6	1	3	2	5	4
4	1	1	1	0	0	0
2	1	1	1	0	0	0
6	1	1	1	0	0	0
8	1	1	1	0	0	0
5	0	0	0	1	1	1
3	0	0	0	1	1	1
9	0	0	0	1	1	1
1	0	0	0	1	1	1
7	0	0	0	1	1	1

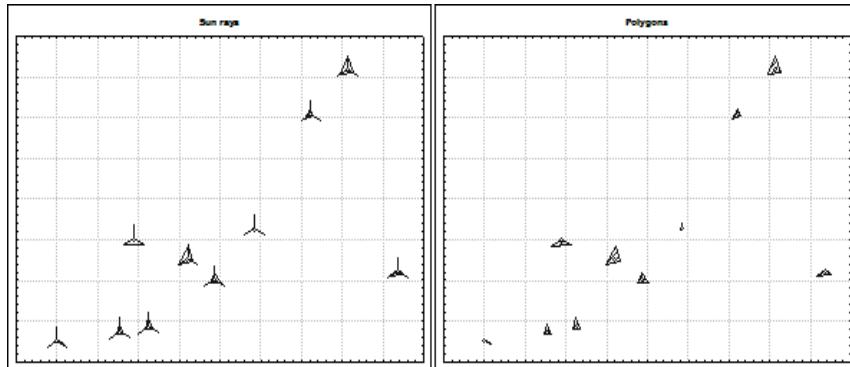
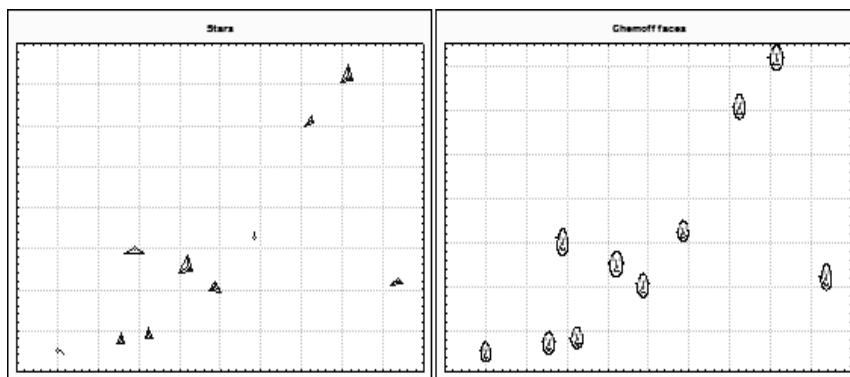
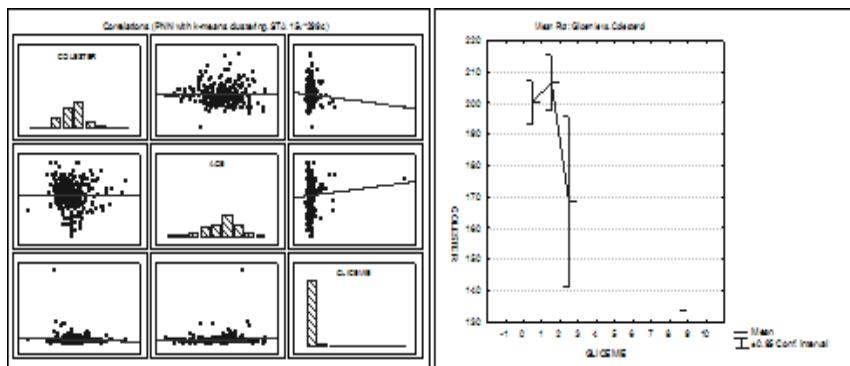
Plecând de la datele medicale referitoare la bolile hepatice, menționate anterior, și focalizându-ne asupra cazului concret al colesterolului, vom prezenta mai jos câteva variante de reprezentări grafice, utilizate mai des în vizualizarea datelor. Aceste reprezentări grafice sunt comune mai tuturor programelor de calculator în domeniul statistic, ele putând fi realizate într-o largă paletă de moduri diferite, chiar de către cei cu cunoștințe adecvate de grafică pe computer.

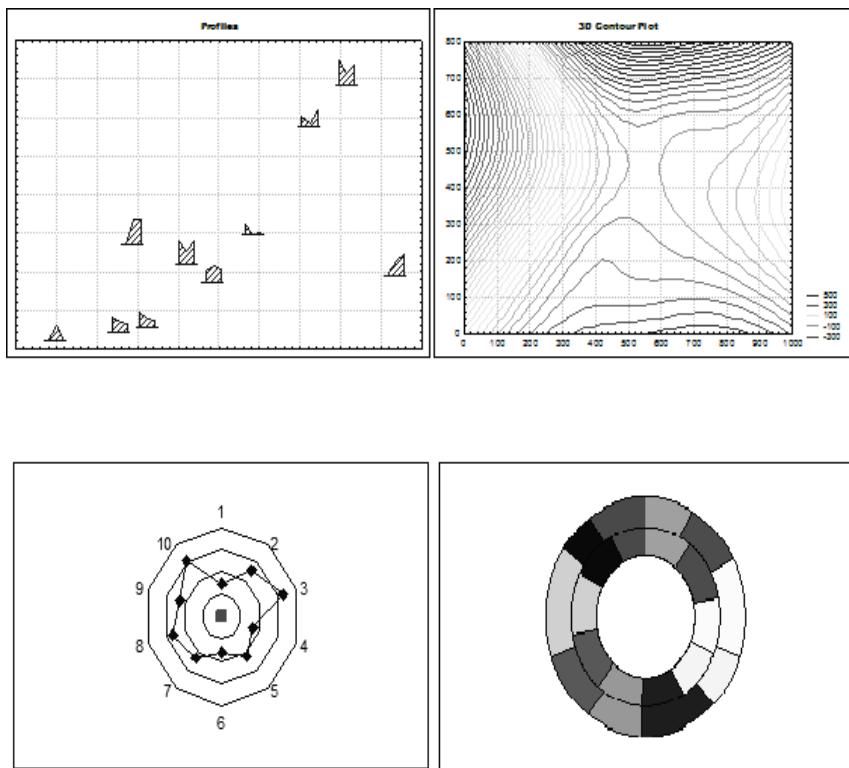












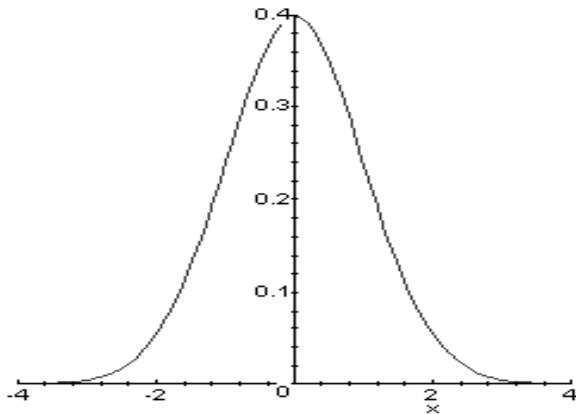
3.5. Examinarea repartițiilor variabilelor

O abordare distinctă în analiza datelor se referă la măsurile tipice ale formei repartiției - *asimetria (skewness)* și *excesul (kurtosis)* – la analiza non-normalității repartiției și la existența cazurilor de multimodalitate. Astfel, în cazul în care repartiția datelor nu este simetrică, se pune problema analizării abaterii de la simetrie. În acest sens, vom defini *asimetria (skewness* – termen utilizat prima dată de Pearson, 1895) ca fiind măsura deviației repartiției date de la simetrie. Formula după care se calculează asimetria este data de:

$$\text{Asimetria} = \frac{n \cdot \sum_{i=1}^n (x_i - \bar{x})^3}{(n-1)(n-2) \cdot \sigma^3}.$$

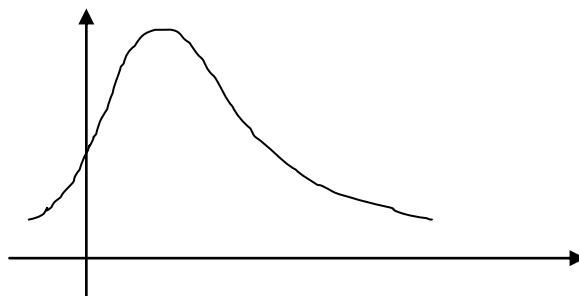
Dacă asimetria este diferită de zero, atunci repartiția este asimetrică. În schimb, repartiția normală (gaussiană) este perfect simetrică, reprezentând modelul

generic de simetrie. Figura de mai jos reprezintă „clopotul lui Gauss”, adică cea mai „simetrică” repartiție.

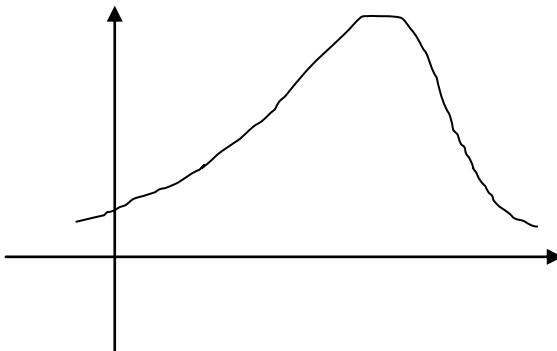


Clopotul lui Gauss - densitatea de repartiție normală standard $N(0, 1)$

Dacă, în schimb, repartitia este asimetrică, atunci ea va avea o “coadă” fie în stânga, fie în dreapta. Dacă această “coadă” este la dreapta, spunem că repartitia are o asimetrie *pozitivă*, iar în celălalt caz, al “cozii” la stânga, asimetria este *negativă*. În figurile de mai jos am desenat cele două tipuri de asimetrii nenele.



Repartiție cu asimetrie ‘pozitivă’.



Repartiție cu asimetrie ‘negativă’.

În ceea ce privește cealaltă măsură tipică formei repartiției, *excesul* (*kurtosis* – termen introdus de Pearson, 1905), acesta măsoară „ascuțimea” unei repartiții. Dacă excesul este diferit de zero, atunci repartiția este ori mai „plată” ori mai „ascuțită” decât repartiția normală, care este chiar zero. Excesul se calculează utilizând formula următoare:

$$Exces = \frac{n \cdot (n+1) \cdot \sum_{i=1}^n (x_i - \bar{x})^4 - 3 \cdot (n-1) \left[\sum_{i=1}^n (x_i - \bar{x})^2 \right]^2}{(n-1) \cdot (n-2) \cdot (n-3) \cdot \sigma^4}$$

Vom ilustra măsurile tipice formei repartiției, prezentate mai sus, în cazul seriei statistice corespunzătoare înălțimii, măsuri prezentate în tabelul de mai jos.

Tabelul 5. Asimetria și excesul

Asimetria	Exces
1,73	4,69

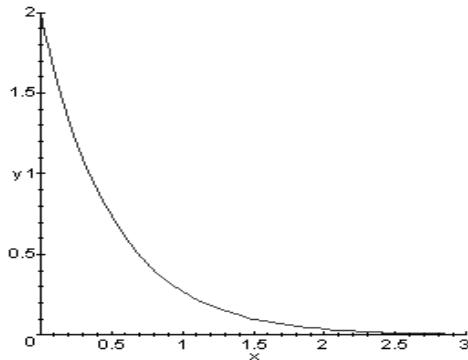
Înafara repartiției normale (gaussiene), există nenumărate alte repartiții continue, mai mult sau mai puțin „depărtate” de aceasta (vezi și caracteristica de asimetrie, prezentată mai sus), care corespund diferitelor tipuri de date. Toate aceste repartiții sunt repartiții non-normale (negaussiene), care caracterizează date întâlnite frecvent în realitate.

Vom prezenta, în continuare, patru asemenea repartiții, cu arie largă de utilizare în Statistică și Data Mining.

(a) Repartiția exponențială de parametru $\lambda > 0$, caracterizată de densitatea:

$$f_X(x) = \begin{cases} 0, & x < 0 \\ \lambda e^{-\lambda x}, & x \geq 0 \end{cases}$$

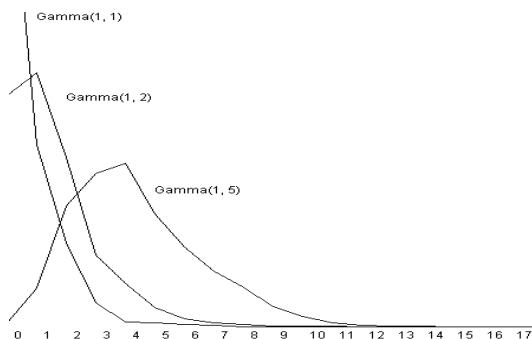
și având reprezentarea grafică de mai jos.



(b) Repartiția *gamma* de parametri $\lambda > 0$ și $k > 0$, având densitatea dată de:

$$f_X(x) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{\Gamma(k)}, \quad x \geq 0 \quad \text{și} \quad f_X(x) = 0, \quad x < 0,$$

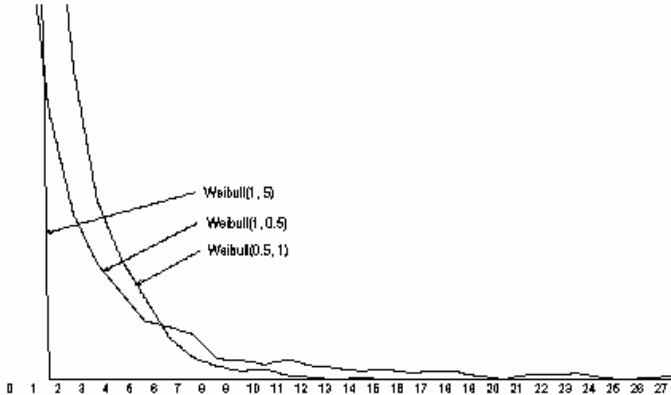
unde $\Gamma(k) = \int_0^\infty e^{-x} x^{k-1} dx$ este funcția *gamma*, și având reprezentarea grafică de mai jos.



(c) Repartiția Weibull de parametri $\alpha > 0$ și $\beta > 0$, având densitatea dată de:

$$f_X(x) = \alpha\beta(\alpha x)^{\beta-1} e^{-(\alpha x)^\beta}, \quad x \geq 0 \quad \text{și} \quad f_X(x) = 0, \quad x < 0,$$

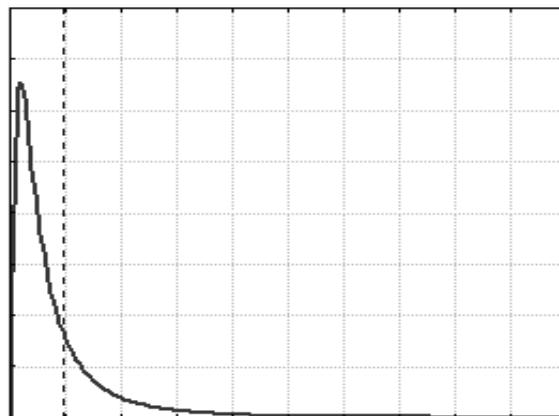
și având reprezentarea grafică de mai jos.



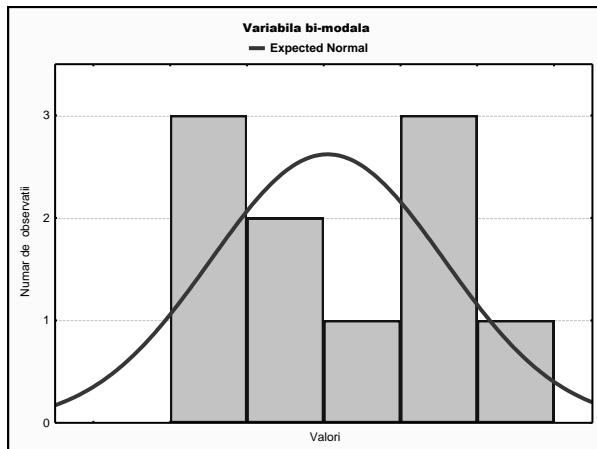
(d) Repartiția *log-normală*. Să considerăm o variabilă $X \geq 0$, astfel încât variabila $\ln(X)$ să fie normal repartizată. Vom spune atunci că variabila X este *log-normal repartizată*, având densitatea:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right], \quad x > 0,$$

și având reprezentarea grafică de mai jos.



În ceea ce privește caracteristica de multimodalitate a repartiției datelor, vizualizarea histogramei corespunzătoare pune în evidență existența unor valori cu frecvență de apariție similară (maximă), așa cum am văzut și atunci când am analizat caracteristicile numerice ale datelor. Ilustrăm în figura de mai jos o repartiție bi-modală.



3.6. Modele liniare și additive avansate

3.6.1. Regresia liniară multiplă

Spre deosebire de cazul regresiei liniare simple, în care am încercat să exprimăm o variabilă (dependentă) în funcție de o altă variabilă (independentă, explicativă, predictor), acum ne punem problema situației în care avem de-a face cu cel puțin trei variabile, dintre care una este dependentă iar celelalte sunt independente, predictoare. Vom prezenta astfel un model de *regresie liniară multiplă*, în care variabila dependentă este exprimată ca o combinație liniară de variabile independente sau variabile predictoare/covariate. Matematic vorbind, acest fapt se exprimă prin *ecuația de regresie multiplă*:

$$Y = a + b_1 X_1 + b_2 X_2 + \dots + b_k X_k$$

unde Y reprezintă variabila dependentă (*outcome*) iar variabilele X_1, \dots, X_k sunt variabilele explicative, predictoare. Constantele b_1, \dots, b_k reprezintă *coeficienții de regresie*, iar a este *constanta de regresie* sau *interceptorul*.

Prințe situațiile avute în vedere atunci când aplicăm regresia liniară multiplă enumerăm următoarele trei, considerate mai importante:

- ⇒ necesitatea de a înlătura posibilele efecte ale unor variabile neesențiale privind scopul propus, atunci când studiem relațiile dintr-un grup de variabile;
- ⇒ explorarea unor variabile prognostic, fără cunoașterea prealabilă a importanței fiecăreia dintre ele;
- ⇒ dezvoltarea unui index prognostic pornind de la mai multe variabile explicative, în scopul proponării variabilei dependente.

Regresia multiplă reprezintă o metodă directă atunci când avem o imagine clară *a priori* a variabilelor pe care le includem în model. În acest caz procedăm la metoda regresiei multiple standard care, practic, este o generalizare a celei simple. Cu totul altfel se petrec lucrurile atunci când nu avem cunoștințe prealabile despre situația reală pe care dorim să-o modelăm și ne confruntăm cu mai multe variabile virtual predictive, din care trebuie să le alegem doar pe cele realmente esențiale modelului. În acest caz este necesară o analiză exploratorie care să confirme modelul ales, pe baza testării semnificației statistice.

Remarcă: Dacă vom include în modelul regresiv variabile binare, luând valorile 0 și 1, atunci coeficienții de regresie respectivi vor indica diferența medie, relativ la variabila dependentă, între grupurile definite de variabilele binare, deoarece diferența între codurile grupurilor este 1. În cazul în care avem de-a face cu variabile categoriale cu mai mult de două valori, o cale de a reduce problema la cea precedentă este de a grupa valorile, creând astfel noi variabile binare. Probleme pot apărea în cazul în care procesăm variabile categoriale ordonate (e.g. stadiile unei maladii) și care pot fi tratate câteodată ca variabile discrete ordonate, indicând o anumită tendință (*trend*).

Așa cum am spus mai sus, atunci când știm dinainte care variabile vor fi incluse în analiza regresivă multiplă, modelul se poate construi fără dificultate, singura problemă rămânând estimarea concretă a ecuației de regresie. Dacă scopul propus este și stabilirea importanței predictorilor, atunci va trebui să alegem dintre toate variabilele modelului pe cele esențiale, pentru obținerea unui model clar și simplu. În acest caz va trebui să facem apel la nivelul p de semnificație statistică a fiecărei variabile pentru a decide ierarhia importanței lor.

Trecând acum la construirea efectivă a modelului regresiv liniar multiplu, în cazul în care nu cunoaștem dinainte care variabile predictive trebuie introduse în model, vom indica pe scurt cei doi algoritmi principali utilizați standard:

- (1) *regresia pas cu pas anterioară (forward stepwise regression);*
 (2) *regresia pas cu pas posterioară (backward stepwise regression).*

(1) *Algoritm pentru regresia pas cu pas anterioară.*

- a) se identifică variabila cu cel mai mare impact asupra variabilei dependente, i.e. variabila cea mai corelată cu variabila dependentă și se introduce în model (i.e. cea mai mică valoare a nivelului de semnificație p);
- b) se găsește variabila din cele rămase care are cea mai mare corelație (ignorând semnul) cu reziduurile modelului de mai sus (ea va „explica” cel mai bine variabilitatea modelului) și se introduce în model;
- c) se repetă pasul (b) până când se ajunge la nivelul de semnificație $p = 0.05$, corespunzător variabilei curente introdusă în model;
- d) când nivelul de semnificație p depășește valoarea de 0.05 se oprește procesul de introducere a predictorilor în model (condiția de *Stop*).

(2) *Algoritm pentru regresia pas cu pas posterioară.* În ceea ce privește algoritmul pentru cealaltă metodă (*regresia pas cu pas posterioară*), vom aborda problema din direcția opusă, adică luăm în considerație inițial toate variabilele și le excludem pas cu pas pe cele care au semnificația cea mai mică. Aici modelul inițial include toate variabilele, considerând că, cel puțin teoretic, toate variabilele pot fi importante. Se exclude apoi variabila cu cea mai mică influență asupra modelului, adică cu cel mai mare nivel de semnificație p privind corelația. Nivelul p de *Stop* este tot 0.05.

Exemplu:

Să considerăm analiza regresivă multiplă utilizată în exprimarea (predicția) *indexului de rezistență a mușchiului respirator PEmax* (exprimat în $\text{cm H}_2\text{O}$) în funcție de variabilele predictoare reprezentate de înălțime (H -cm), greutate (G-kg), vârstă (ani), sex, procentul masei corporale (%), BMP, volumul respirator forțat per secundă (FEV₁), volumul rezidual (RV), capacitatea funcțională reziduală (FRC) și capacitatea totală a plămânlui (TLC), pentru un lot de 25 bolnavii cu fibroză cistică [5]. Variabila dependentă a acestui model este reprezentată de indexul de rezistență a mușchiului respirator (*PEmax*). Tabelul de mai jos sintetizează toate aceste caracteistică.

VÂRSTĂ	SEX	H	G	BMP	FEV ₁	RV	FRC	TLC	PEMAX
--------	-----	---	---	-----	------------------	----	-----	-----	-------

7	0	109	13.1	68	32	258	183	137	95
7	1	112	12.9	65	19	449	245	134	85
8	0	124	14.1	64	22	441	268	147	100
8	1	125	16.2	67	41	234	146	124	85
8	0	127	21.5	93	52	202	131	104	95
9	0	130	17.5	68	44	308	155	118	80
11	1	139	30.7	89	28	305	179	119	65
12	1	150	28.4	69	18	369	198	103	110
12	0	146	25.1	67	24	312	194	128	70
13	1	155	31.5	68	23	413	225	136	95
13	0	156	39.9	89	39	206	142	95	110
14	1	153	42.1	90	26	253	191	121	90
14	0	160	45.6	93	45	174	139	108	100
15	1	158	51.2	93	45	158	124	90	80
16	1	160	35.9	66	31	302	133	101	134
17	1	153	34.8	70	39	204	118	120	134
17	0	174	44.7	70	49	187	104	103	165
17	1	176	60.1	92	29	188	129	130	120
17	0	171	42.6	69	38	172	130	103	130
19	1	156	37.2	72	21	216	119	81	85
19	0	174	54.6	86	37	184	118	101	85
20	0	178	64.0	86	34	225	148	135	160
23	0	180	73.8	97	57	171	108	98	165
23	0	175	51.5	71	33	224	131	113	95
23	0	179	71.5	95	52	225	127	101	195

În tabelul următor prezentăm matricea corelațiilor tuturor variabilelor modelului regresiv (i.e. matricea coeficienților de corelație), analiza corelației multiple fiind necesară pentru a stabili existența legăturii între variabilele analizate. Menționăm că în cazul de față variabila sex fiind o variabilă aleatoare calitativă (categorială) va fi codată binar: 1 = bărbați și 0 = femei. Practic, suntem interesați de examinarea relațiilor directe între fiecare variabilă explicativă a modelului și variabila dependentă, desemnată de *PEmax*.

	Vârstă	Sex	H	G	BMP	FEV ₁	RV	FRC	TLC	PEmax

Vârstă	1	-0.17	0.93	0.91	0.38	0.29	-0.55	-0.64	-0.47	0.61
Sex	-0.17	1	-0.17	-0.19	-0.14	-0.53	0.27	0.18	0.02	-0.29
H	0.93	-0.17	1	0.92	0.44	0.32	-0.57	-0.62	-0.46	0.6
G	0.91	-0.19	0.92	1	0.67	0.45	-0.62	-0.62	-0.42	0.64
BMP	0.38	-0.14	0.44	0.67	1	0.55	-0.58	-0.43	-0.36	0.23
FEV₁	0.29	-0.53	0.32	0.45	0.55	1	-0.67	-0.67	-0.44	0.45
RV	-0.55	0.27	-0.57	-0.62	-0.58	-0.67	1	0.91	0.59	-0.32
FRC	-0.64	0.18	-0.62	-0.62	-0.43	-0.67	0.91	1	0.7	-0.42
TLC	-0.47	0.02	-0.46	-0.42	-0.36	-0.44	0.59	0.7	1	-0.18
PEmax	0.61	-0.29	0.6	0.64	0.23	0.45	-0.32	-0.42	-0.18	1

Vom considera mai întâi prima variantă de regresie multiplă - *regresia pas cu pas anterioară*. În cazul de față, primele locuri eligibile pentru variabilele predictoare, în ordine descrescătoare (i.e. gradul de semnificație a corelațiilor variabilelor explicative în raport cu variabila dependentă, în cadrul modelului regresiv), sunt date în tabelul de mai jos. Se observă că ultima variabilă luată în considerație în regresie –variabila RV– nu poate face parte din model având nivelul de semnificație *p* superior lui 0.05.

	<i>b</i>	Eroarea standard (<i>b</i>)	<i>t</i> –test (20)	<i>p</i>
Interceptor	63.94669	53.27673	1.20027	0.2440
G	1.74891	0.38063	4.59475	0.0001
FEV ₁	1.54770	0.57761	2.67948	0.0051
BMP	-1.37724	0.56534	-2.43612	0.0210
RV	0.12572	0.08315	1.51199	0.0736

În ceea ce privește aplicarea celeilalte metode - „*regresia pas cu pas posterioară*”, datele utilizate la construcția modelului sunt prezentate în tabelul de mai jos.

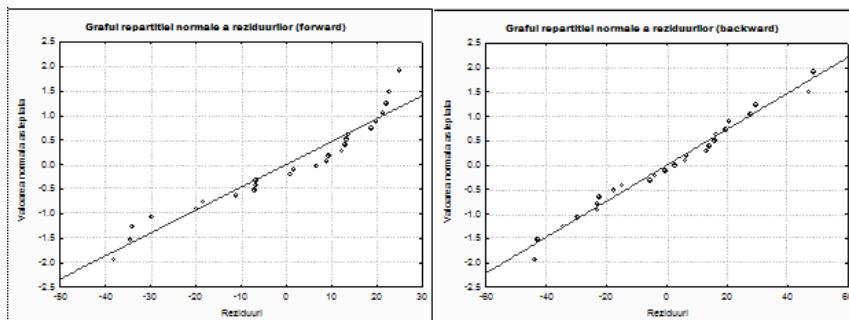
	<i>b</i>	Eroarea standard (<i>b</i>)	<i>t</i> –test (23)	<i>p</i>
Interceptor	63.54564	12.70163	5.002952	0.000046
G	1.18671	0.30086	3.944453	0.000646

Remarcă: În principiu, nici una dintre cele două variante de regresie multiplă de mai sus nu este cea ideală. Dacă vrem cel mai „larg” model, îl alegem pe cel „anterior”, iar dacă-l dorim pe cel mai „strict”, îl alegem pe cel „posterior”. Oricum, se observă de aici că folosirea doar a criteriului nivelului de semnificație *p* nu rezolvă problema complet.

Ecuațiile de regresie în cele două cazuri sunt:

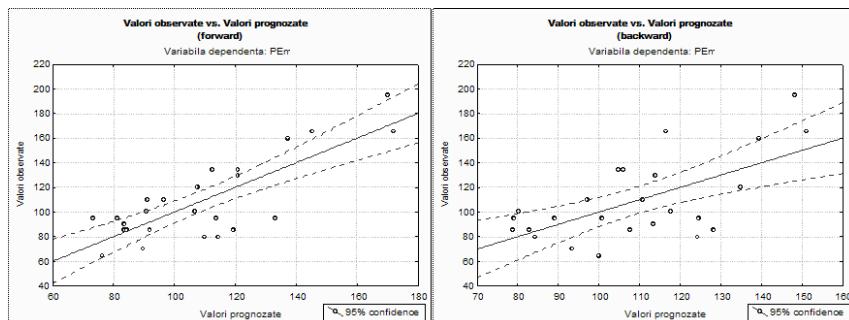
- (1) $PEmax = 63,95 + 1,75 \times G + 1,55 \times FEV_1 - 1,38 \times BMP$
- (2) $PEmax = 63,55 + 1,19 \times G$

În continuare, vom verifica gradul de precizie/acuratețe (*goodness-of-fit*) al predicției realizată cu ajutorul celor două metode de regresie multiplă. Pentru aceasta trebuie analizată repartiția valorilor reziduurilor (i.e. diferența între valorile observate și valorile proгnozate). O acuratețe bună a predicției implică o repartiție normală (gaussiană) a reziduurilor, adică o plasare a acestora de-a lungul unei linii drepte (*droite de Henry*). O abatere de la această aranjare a reziduurilor va indica, de asemenea, și existența unor valori extreme (*outliers*). Prezentăm mai jos repartiția reziduurilor pentru cele două variante de regresie multiplă.



Așa cum se observă din figura de mai sus, ambele grafice indică validarea satisfăcătoare a celor două modele regresive.

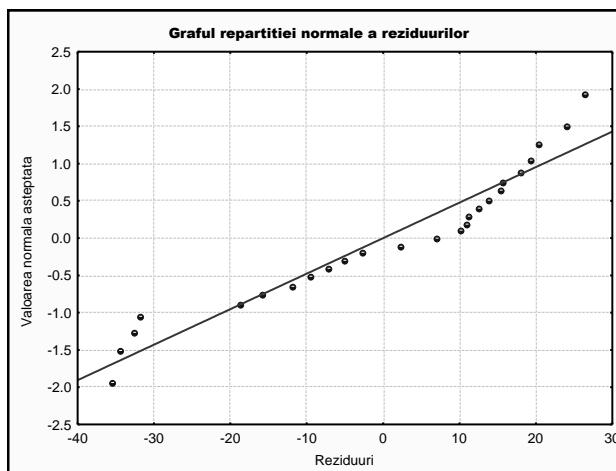
Mai departe, reprezentarea grafică a valorilor observate în raport cu valorile proгnozate este în mod special folosită pentru identificarea potențialelor clustere de cazuri care nu sunt bine proгnozate. Din reprezentarea grafică de mai jos deducem, de asemenea, o bună proгnoză a variabilei dependente.



În final, vom proceda la aplicarea metodei regresiei multiple standard, în care caz vom lua în considerație toate cele nouă variabile explicative. Rezultatul aplicării acestei metode este redat în tabelul de mai jos.

	b	Eroare standard (b)	t -test (15)	p
Interceptor	137.2298	207.9465	0.65993	0.519305
Vârstă	-2.4757	4.3674	-0.56685	0.579201
Sex	-1.3881	13.5972	-0.10209	0.920039
H	-0.3082	0.8641	-0.35668	0.726299
G	2.8789	1.8466	1.55903	0.139834
BMP	-1.7972	1.1157	-1.61078	0.128066
FEV1	1.4946	0.9708	1.53946	0.144519
RV	0.1781	0.1867	0.95381	0.355297
FRC	-0.1637	0.4753	-0.34432	0.735382
TLC	0.1143	0.4772	0.23944	0.814009

Se observă cu ușurință că în acest caz nicio variabilă predictivă nu trece pragul de 5% al nivelului de semnificație p . Totuși, ținând cont că acesta nu este un criteriu absolut de acceptare a modelului, vom lua în considerație și proghoza astfel obținută, mai ales că reprezentarea grafică a reziduurilor (ilustrată în figura de mai jos) indică un grad de acceptabil de acuratețe a modelului.



Ecuăția de regresie corespunzătoare este dată de formula:

$$PE_{\max} = 137.23 - 1.79BMP + 2.87G + 1.49FEV_1 + 0.17RV - 2.47Varsta - 0.3H - 0.16FRC + 0.11TLC - 1.38Sex$$

Remarcă: Utilizăm ecuația de regresie multiplă pentru a obține valorile variabilei dependente pentru orice valori individuale ale variabilelor explicative. În acest mod, pentru un anumit obiect cu attributele predictive cunoscute, se deduce valoarea atributului necunoscut, considerat ca atribut răspuns (*outcome*). În cazul de mai sus, pentru un anumit pacient căruia i se cunosc valorile celor nouă parametri medicali predictivi, i se poate prognoza, cu o acuratețe suficientă, valoarea PEmax, prin introducerea în ecuația de regresie a valorilor sale individuale. Spunem că, astfel, se obține o *valoare prognostic* (*index prognostic*), pe baza datelor cunoscute.

În finalul acestei scurte expuneri privind regresia multiplă, vom prezenta câteva considerente standard privind analiza regresivă:

- Dacă avem de ales între mai multe variabile predictive, trebuie să știm că nu există o certitudine clară în alegerea lor. În principiu, se folosește ca prag valoarea $p = 0.05$, dar sunt cazuri când, datorită rolului câștigat din punct de vedere al practicii, nu excludem variabile cu valoarea $p = 0.2$ sau chiar mai mare.
- Se va evita un model provenind dintr-un eșantion mic dar având multe variabile predictive. Ca regulă generală, numărul de predictori nu trebuie să depășească valoarea $n/10$, unde n este volumul eșantionului.
- Alegerea automată a modelului pe baza unui program statistic adevarat este normală, dar nu trebuie ignorat bunul simț al practicianului în evaluarea și validarea finală a modelului.
- Variabilele explicative, puternic corelate între ele, vor fi de așa natură selectate, încât să fie inclus în model doar un „reprezentant” al lor și nu toate (e.g. G sau H), pentru evitarea redundanței.
- Se va verifica *a priori* dacă există într-adevăr o legătură liniară între variabila dependență și fiecare predictor.
- Se presupune, din start, că efectul fiecărui predictor este independent de ceilalți. Dacă se bănuiește existența vreunei legături între doi predictori (aceasta nu se stabilește doar pe baza corelației, ci pe baze intrinseci, ce țin de natura intimă a fenomenului cercetat), trebuie adăugat eventual un termen de interacțiune a lor (i.e. o nouă variabilă rezultată ca funcție a lor, e.g. produsul lor) în model.
- Pentru mai multă siguranță, se verifică capacitatea modelului pe alt eșantion, dacă acest lucru este posibil.

3.6.2. Regresia logistică

În paragraful anterior am prezentat câteva noțiuni clasice privind regresia liniară multiplă, arătând cum se obține ecuația ce stabilește legătura liniară între mai multe variabile aleatoare, variabila dependentă fiind o variabilă continuă, extinzând astfel metoda regresiei liniare simple de la două la mai multe variabile. Sunt multe domenii de cercetare însă, din medicină, economie, fizică, meteorologie, astronomie, biologie etc., în care variabila dependentă nu mai este o variabilă continuă ci una binară, categorială. Putem cita ca exemple răspunsul unui pacient la un anumit tratament, sau categoriile generate de pacienții care prezintă sau nu un anumit simptom, fidelitatea unui client față de ofertă unui supermarket, clasificarea stelelor etc. În acest caz, când variabila dependentă se referă la două valori (categori), nu mai este de folos regresia multiplă, ci se utilizează o abordare oarecum similară ca formă, dar distinctă ca sens -*regresia logistică*. Astfel, în loc să se prognoseze valoarea variabilei dependente în raport cu valorile variabilelor explicative, se va prognoza o *transformare* a variabilei dependente. Această transformare se numește transformarea *logit*, desemnată ca *logit* (p), unde p este proporția de obiecte cu o anumită caracteristică (e.g. p reprezintă probabilitatea ca un individ să aibă infarct miocardic, sau p reprezintă probabilitatea ca un client să rămână fidel unui anumit supermarket sau produs etc.). Pentru a înțelege rațiunea acestei proceduri, să observăm că dacă am cuantifica variabila dependentă categorială utilizând valorile 1 și 0, valori corespunzătoare celor două situații posibile "A" sau "B", atunci media acestor valori, calculată pe un eșantion dat, reprezintă tocmai proporția obiectelor corespunzătoare uneia din cele două situații. Revenind la transformarea definită de *logit* (p), să menționăm formula după care se calculează:

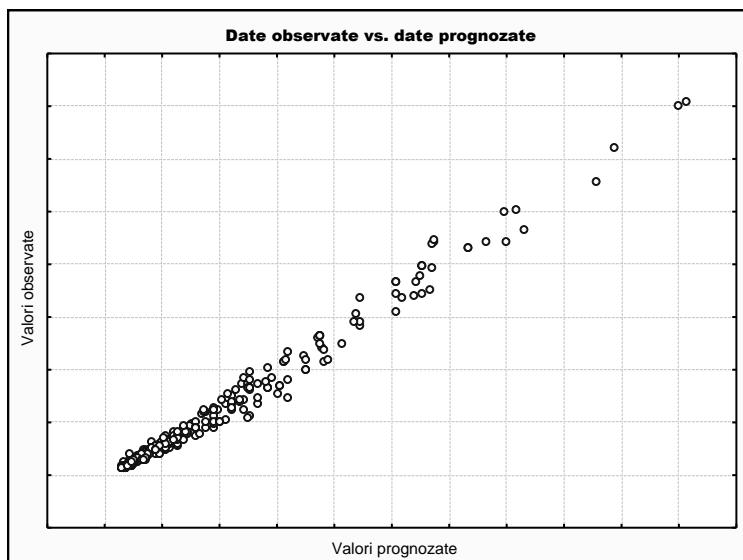
$$\text{logit} (p) = \ln \left(\frac{p}{1-p} \right).$$

Atunci când utilizăm metoda regresiei logistice, la sfârșitul calculelor vom obține valoarea $\text{logit} (p) = \alpha$ sub forma unei combinații liniare a variabilelor explicative. În aceste condiții, putem calcula valoarea efectivă a probabilității p , utilizând formula:

$$p = e^{\alpha} / (1 + e^{\alpha}).$$

Exemplu:

1) Vom considera baza de date privind cele 299 de persoane analizate prin prisma studierii caracteristicilor diferitelor boli hepatice. În acest caz, vom lua în considerare atât enzimele serice clasice: BRT, BRD, BRI, FA, GGT, LAP, TGP, TGO, LDH, IP, cât și măsurătorile medicale privind glicemia, colesterolul, albumina, gamma, precum și vârsta. Să considerăm diagnosticul de cancer hepatic (HCC) ca variabilă dependentă și celelalte atribute, în număr de 15, ca variabile explicative. În acest caz, parametrul p reprezintă probabilitatea ca o persoană să dezvolte cancer hepatic, pornind de la factorii de risc corespunzători celor 15 variabile predictive. Prezentăm mai jos, ilustrativ, doar graficul valorilor observate și cel al valorilor prognozate, din care reiese clar acuratețea acceptabilă a prognozei realizate cu această metodă.



2) Scopul studiului este reprezentat de stabilirea influenței fumatului, obezității și sforăitului asupra hipertensiunii arteriale, în sensul prognozei apariției acestieia pe baza variabilelor explicative mai sus amintite, privite ca factori de risc pentru această maladie [5]. Utilizând metoda regresiei logistice, obținem ecuația:

$$\text{logit}(p) = -2,378 - 0,068 \times \text{fumat} + 0,695 \times \text{obezitate} + 0,872 \times \text{sforăit},$$

ecuație din care putem obține probabilitatea ca un subiect să dezvolte hipertensiune arterială, pe baza valorilor individuale ale celor trei variabile explicative, privite ca factori de risc pentru hipertensiunea arterială și codate astfel: 0 = nefumător, 1 = fumător; 0 = ponderal, 1 = supraponderal; 0 = nu sforăie, 1 = sforăie.

Dacă dorim să efectuăm o comparație între fumători și nefumători, în raport cu riscul de a avea hipertensiune arterială, vom compara ecuațiile:

$$\text{logit } (p_{fum}) = -2,378 - 0,068 + 0,695 \times \text{obezitate} + 0,872 \times \text{sforăit},$$

$$\text{logit } (p_{nefum}) = -2,378 + 0,695 \times \text{obezitate} + 0,872 \times \text{sforăit},$$

După cum se observă fără dificultate, am considerat variabila „fumat” egală mai întâi cu 1, apoi cu 0. Rezultă că:

$$\text{logit } (p_{nefum}) - \text{logit } (p_{fum}) = 0.068,$$

de unde:

$$\ln \left[\frac{p_{nefum} (1 - p_{fum})}{p_{fum} (1 - p_{nefum})} \right] = 0.068,$$

sau:

$$\left[\frac{p_{nefum} (1 - p_{fum})}{p_{fum} (1 - p_{nefum})} \right] = 1.070,$$

valoare care poate fi interpretată ca o măsură a riscului de hipertensiune printre nefumători în raport cu fumătorii (pentru amănunte, a se vedea [5]).

Remarcă: 1. Modelul de regresie logistică permite prognoza probabilității unui anumit rezultat pe baza valorilor variabilelor predictive, dând astfel posibilitatea distingerea categoriilor de obiecte în raport cu acel rezultat. De exemplu, în cazul unui model medical, putem distinge (discrimina) persoanele ce pot dezvolta o anumită maladie în raport cu celelalte. La fel ca și în cazul regresiei multiple, putem utiliza modelul regresiei logistice ca un index (indice) prognostic pentru un anumit grup de obiecte. Practic, vom defini:

$$IP = \log \left(\frac{p}{1 - p} \right) = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_k \cdot x_k,$$

unde IP (index prognostic) este transformarea logit a probabilității p ca un obiect să aibă o anumită caracteristică, iar modelul conține k variabile explicative. Vom calcula valoarea IP pentru toate obiectele studiate și apoi vom compara repartițiile celor care au caracteristica de interes față de cele care nu o au. Astfel, putem descoperi cât de bună este separația (discriminarea) între cele două grupuri și putem identifica cel mai bun prag care maximizează discriminarea. Dacă toate variabilele explicative sunt binare, atunci IP poate avea puține valori distincte. Dacă, în schimb, una sau mai multe variabile

explicative este continuă, scorul *IP* va fi o variabilă continuă de asemenea. Mai multe mănuște în acest sens sunt de găsit în [5].

2. O altă abordare a problemei discriminării între grupuri de obiecte pe baza utilizării mai multor variabile este cunoscută în Statistică sub numele de *analiză discriminant* (*Discriminant analysis*). Analiza discriminant face parte din domeniul metodologic al *tehnicii exploratorii multivariate* (*Multivariate Exploratory Techniques*), de care ne vom ocupa mai pe larg în cursul acestui capitol.

3.6.3. Modelul de regresie Cox

Una dintre ramurile importante ale Statisticii, cu aplicații deosebit de interesante mai ales în domeniul medical și al sistemelor mecanice, este *analiza supraviețuirii* (*Survival Analysis*). În medicină este cunoscută chiar sub acest nume, în timp ce în științele inginerești se numește *teoria siguranței în funcționare* (*Reliability Theory*), iar în economie este cunoscută ca *analiza duratei* (*Duration Analysis*). Indiferent de context, elementul de bază al acestei teorii este termenul de „moarte”, „eșec”, „cădere”, „absență”, „ieșire din funcționare” etc., care este privit ca un *eveniment* în analiza supraviețuirii.

Pentru a înțelege mai bine despre ce este vorba vom considera cazul medical, de la care de fapt s-a și pornit în dezvoltarea acestei ramuri statistice. Astfel, problema analizei timpilor de supraviețuire, așa cum o indică și numele, se referă, în principiu, la supraviețuirea unui pacient în urma unei operații serioase, tratamentului unei anumite boli cu sfârșit letal, e.g. cancer, SIDA etc. Practic, se înregistrează timpul din momentul începutului procesului medical (operație, tratament etc.) până în momentul decesului, timpul respectiv fiind numit *temp de supraviețuire*, iar analiza să facă și obiectul analizei supraviețuirii. Să menționăm faptul că, în multe cercetări clinice, prin timp de supraviețuire se înțelege și timpul până la apariția unui simptom, până la revenirea bolii, până la remisiune etc. În acest context, prin *probabilitate de supraviețuire* se înțelege proporția indivizilor unui grup, supus unui anumit experiment medical comun, care ar putea supraviețui o anumită perioadă de timp (e.g. operație de transplant de inimă, chimioterapie etc.), în anumite circumstanțe date. Tehnica clasică de calculare a timpului de supraviețuire este următoarea. Se notează variabila aleatoare care reprezintă timpul de supraviețuire cu *X*. Probabilitatea de supraviețuire se calculează prin împărțirea timpului în mici intervale $(0, t_1), \dots, (t_{k-1}, t_k), \dots$, și estimarea apoi a probabilității:

$$P\{X \leq t_n\} = P\{X \leq t_1\}P\{X \leq t_2 | X = t_1\} \dots P\{X \leq t_n | X = t_{n-1}\}.$$

O problemă comună atât domeniului cercetării medicale (i.e. timpul de supraviețuire), cât și cercetării inginerești (i.e. timpul de funcționare sigură a

unui mecanism) este determinarea efectului unor variabile continue (independente) asupra timpului de supraviețuire, în particular identificarea existenței corelației între predictori și timpul de supraviețuire.

Metoda clasică de analiza supraviețuirii, bazată pe tehnici ca, de exemplu, *curba de supraviețuire Kaplan-Meier (Kaplan-Meier survival curve)*, *analiza tabelelor de viață (life table analysis)*, *testul logrank (logrank test)*, *raportul hazardului (hazard ratio)* etc., nu poate fi utilizată pentru a explora efectul simultan asupra supraviețuirii a mai multor variabile. Trebuie subliniat faptul că, în acest caz, nu se poate utiliza direct metoda regresiei multiple din cel puțin două motive: (a) variabila ce descrie timpul de supraviețuire nu este de cele mai multe ori normal repartizată (de obicei este exponențială sau Weibull) și (b) analiza supraviețuirii utilizează așa-numitele *date cenzurate*, adică unele observații sunt incomplete. Ca să se înțeleagă mai bine ce este cu aceste date în context, să ne imaginăm că un grup de pacienți cu cancer sunt urmăriți în cadrul unui experiment o anumită perioadă de timp (*follow-up*). După trecerea acestei perioade, pacienții care au supraviețuit nu mai sunt supravegheați și, atunci când se analizează timpul de supraviețuire, nu se mai știe cu exactitate dacă mai sunt încă în viață. Pe de altă parte, unii pacienți pot părăsi grupul în timpul perioadei de supraveghere, fără a se mai cunoaște situația lor ulterioară. Datele privind acest tip de pacienți sunt date cenzurate.

Analiza supraviețuirii utilizează mai multe metode regresive specializate, din care amintim:

- Analiza regresivă a hazardurilor proporționale –modelul Cox (*Cox proportional hazards regression analysis*), pe scurt -modelul Cox al hazardului proporțional (*Cox proportional hazard model*);
- Modelul Cox al hazardului proporțional cu covariante dependente de timp (*Cox proportional hazard model with time-dependent covariates*);
- Modelul regresiv exponențial (*Exponential regression model*);
- Modelul liniar regresiv log-normal (*Log-normal linear regression model*);
- Modelul liniar regresiv normal (*Normal linear regression model*).

În continuare, vom face o trecere succintă în revistă a principalelor elemente doar pentru modelul Cox al hazardului proporțional.

Mai întâi, *funcția de supraviețuire (survival function)* este definită prin probabilitatea:

$$S(t) = P\{T > t\},$$

unde t reprezintă timpul în general, iar T este timpul până la deces. Repartitia duratei de viață (*lifetime*) este dată de

$$F(t) = 1 - S(t),$$

unde $f(t) = \frac{d}{dt} F(t)$ reprezintă rata de deces pe unitatea de timp.

Mai departe, funcția hazard (*hazard function*) este definită de formula:

$$\lambda(t) = P(t < T < t + dt) = \frac{f(t)dt}{S(t)} = -\frac{S'(t)dt}{S(t)}$$

Funcția hazard reprezintă deci riscul de a deceda într-un interval foarte scurt de timp dt după un timp dat T , presupunând evident supraviețuirea până la acel moment. Modelul Cox al hazardului proporțional este foarte general între modelele regresive deoarece nu se bazează pe nicio ipoteză prealabilă privind repartiția supraviețuirii. El se bazează doar pe presupunerea că hazardul este o funcție doar de variabilele independente (predictive, covarianțe) Z_1, Z_2, \dots, Z_k , adică:

$$h(t; Z_1, Z_2, \dots, Z_k) = h_0(t) \cdot \exp(b_1 \cdot Z_1 + b_2 \cdot Z_2 + \dots + b_k \cdot Z_k),$$

sau, logaritmând,

$$\log \left[\frac{h(t; Z_1, \dots, Z_k)}{h_0(t)} \right] = b_1 \cdot Z_1 + b_2 \cdot Z_2 + \dots + b_k \cdot Z_k,$$

fiind deci un model semi-parametric. Termenul $h_0(t)$ se numește hazardul de bază (*baseline hazard, underlying hazard function*) reprezentând hazardul pentru un anumit individ atunci când toate variabilele independente sunt egale cu zero.

Să menționăm că, totuși, trebuie luate în considerație două condiții: (a) trebuie să existe o relație multiplicativă între $h_0(t)$ și funcția log-liniară a covarianțelor –ipoteza proporționalității, prin prisma hazardului și (b) trebuie să existe o relație log-liniară între hazard și variabilele independente.

Un exemplu interesant de utilizare a modelului Cox al hazardului proporțional în medicină se referă la analiza tratamentului cu azathioprină a pacienților cu ciroză biliară primară (pentru amănuite, a se vedea [5]).

Remarcă: 1. Selecția variabilelor explicative se face după aceeași regulă ca și în cazul regresiei liniare multiple.

2. Semnul coeficienților de regresie b_i trebuie interpretat după cum urmează. Un semn pozitiv indică un hazard ridicat deci, în consecință, un prognostic negativ pentru individul cu o valoare ridicată a acelei variabile. În schimb, un semn negativ pentru un anumit coeficient indică un hazard scăzut relativ la acea variabilă.

3. La fel ca și în cazul regresiei liniare multiple/logistice și în acest caz putem utiliza valoarea predictivă a modelului, folosind indicele prognostic definit de $IP = b_1 \cdot Z_1 + b_2 \cdot Z_2 + \dots + b_k \cdot Z_k$. Se poate calcula astfel funcția de supraviețuire $S(t) = \exp[-H_0(t)]^{\exp(IP)}$, unde $H_0(t)$, numit și *hazardul de bază cumulat*, este o funcție scără în variabila timp.

3.6.4. Modele aditive

Să presupunem, la fel ca și în cazul regresiei multiple, că avem o variabilă dependentă Y și k variabile explicative (predictoare) X_1, X_2, \dots, X_k . Spre deosebire de cazul modelelor liniare, cazul modelului aditiv implică posibila legătură între variabila dependentă și predictori sub forma:

$$Y = f_1(X_1) + f_2(X_2) + \dots + f_k(X_k) + \varepsilon,$$

unde $f_j, j = 1, 2, \dots, k$ sunt, în general, funcții *netede* (i.e. de clasa C^∞), în unele cazuri și funcții de clasă C^1), iar ε este o variabilă aleatoare repartizată normal standard $N(0, 1)$. Este ușor de observat că un model aditiv reprezintă generalizarea modelului regresiei liniare multiple (pentru $\varepsilon = 0$). Cu alte cuvinte, în loc de un singur coeficient *per* variabilă explicativă, la modelele aditive găsim o funcție nespecificată *per* fiecare predictor, care va trebui să fie estimată în vederea programei optime a valorilor variabilei dependente.

- Remarcă:** 1. Ipoteza aditivității $\sum f_i(X_i)$ este o restricție a cazului general al unui model predictiv de tipul $Y = f(X_1, X_2, \dots, X_n)$.
2. Funcțiile parametru ale modelului aditiv sunt estimate până la o constantă aditivă.
3. *Modele aditive generalizate* [94], [95]. Un *model liniar generalizat* este reprezentat prin ecuația:

$$Y = g(b_0 + b_1 \cdot X_1 + b_2 \cdot X_2 + \dots + b_k \cdot X_k),$$

unde g este o funcție netedă nedeterminată. Dacă notăm formal g^{-1} inversa funcție g , funcție numită *funcție legătură (link function)*, atunci putem scrie ecuația de mai sus sub forma ușor modificată:

$$g^{-1}(E[Y]) = b_0 + b_1 \cdot X_1 + b_2 \cdot X_2 + \dots + b_k \cdot X_k,$$

unde $E[Y]$ reprezintă media variabilei dependente Y . Să combinăm acum un model aditiv cu un model liniar generalizat. Vom obține atunci ecuația modelului, cunoscut sub denumirea de *model aditiv generalizat*, sub forma:

$$g^{-1}(E[Y]) = f_1(X_1) + f_2(X_2) + \dots + f_k(X_k).$$

Problema de bază la aceste modele este estimarea funcțiilor parametri f_i ale modelului. Cea mai cunoscută metodă de evaluare a funcțiilor f_i este reprezentată de o interpolare pe bază de diagrame de dispersie (*scatterplot smoother*), utilizând funcții spline cubice. De exemplu, în cazul unui model simplu cu doar două funcții f_1 și f_2 , având forma: $Y = f_1(X_1) + f_2(X_2) + \varepsilon$, folosind aproximarea spline, se obține forma celor două funcții:

$$f_1(X) = \delta_1 + X \cdot \delta_2 + \sum_{j=1}^{q_1-2} R(X, X_j^*) \cdot \delta_{j+2},$$

$$f_2(X) = \gamma_1 + X \cdot \gamma_2 + \sum_{j=1}^{q_2-2} R(X, X_j^*) \cdot \gamma_{j+2},$$

unde δ_j , γ_j sunt parametrii necunoscuți ai funcțiilor f_1 și f_2 , q_1 , q_2 reprezintă numărul parametrilor necunoscuți, iar X_j^* reprezintă nodurile de interpolare pentru cele două funcții. Un exemplu ilustrativ pentru utilizarea unui model aditiv generalizat în practică este dat de estimarea volumului masei lemnătoase a unui copac în funcție de circumferință și înălțimea sa. În acest caz se poate utiliza, în locul „clasicei” metode de calcul al volumului unui cilindru, modelul aditiv generalizat de ecuație:

$$\log(E[\text{Volum}]) = f_1(\text{Circumferință}) + f_2(\text{Înălțime}),$$

presupunând că volumul are o repartiție Gamma (detalii tehnice pentru topic [180]).

3.6.5. Serii temporale. Prognoză

În cursul observării celor mai multor fenomene sau situații din lumea reală, se constată că datele care ne interesescă și pe care le strângem au o cronologie care ne permite analizarea evoluției lor pe măsura scurgerii timpului. O asemenea secvență temporală de date (observații temporale), notată $(\xi_t, t \in T)$, unde $T \subseteq \mathbf{R}$ se referă la *temp*, se numește *serie temporală*, sau *serie dinamică*. Vom remarcă faptul că datele ξ_t se pot referi la observații în timp discret, i.e. zile, săptămâni, trimestre, ani etc., sau pot fi date obținute în timp continuu.

Menționăm două obiective principale ale analizei seriilor temporale:

- Identificarea naturii fenomenului reprezentat de secvența de observații;
- Prognoza valorilor viitoare posibile, plecând de la observațiile deja cunoscute.

Pentru îndeplinirea sarcinilor mai sus amintite este necesară identificarea prealabilă a pattern-ului seriei temporale observate. Odată patternul identificat și descris, îl putem integra în anumite clase bine definite de fenomene asemănătoare și interpreta în scopul predicției unor valori viitoare ale fenomenului studiat.

În ceea ce privește problema identificării pattern-ului unei serii temporale, se pleacă de la premiza că datele consistă într-un pattern sistematic (i.e. un set de componente identificabile) și zgromot aleator care îngreunează identificarea formei originare a datelor. Metodele utilizate vor face apel, în consecință, la diferite tehnici specifice de filtrare a datelor pentru îndepărtarea zgromotului.

Din nefericire, nu există o metodă standard pentru identificarea pattern-ului ascuns în date, după îndepărtarea zgromotului, procedându-se de la caz la caz. Astfel, dacă se dispune de un trend monoton (crescător sau descreșcător, înfără unor valori singulare), problema estimării concrete a acestuia și apoi etapa de prognoză nu sunt dificile. Dacă însă seria temporală are numeroase erori, atunci un prim pas este acela de „netezire” (*smoothing*) a datelor, adică o formă de aproximare locală a datelor pentru îndepărtarea componentelor nesistematice (e.g. *moving average* – tehnică de înlocuire a unei valori prin media simplă sau ponderată a *n* valori vecine acesteia).

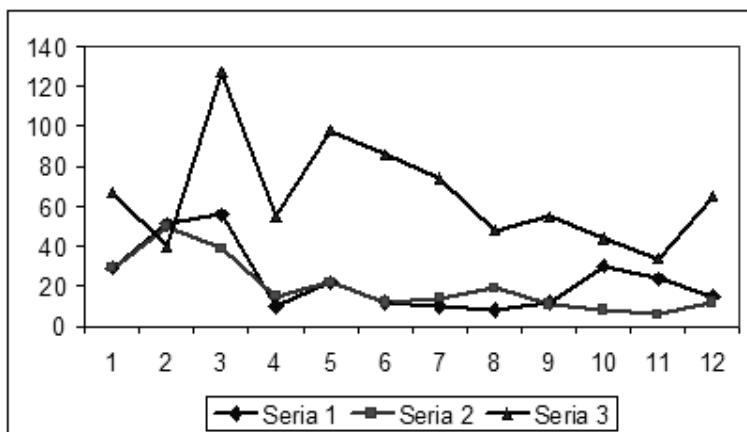
Din punctul de vedere al aplicațiilor seriilor temporale prin prisma Data Mining, vom menționa următorul fapt. Astfel, orice asociere secvențială de timpi și numere poate fi privită ca o serie temporală. De exemplu, fluctuațiile istorice ale prețului petrolier (e.g. Nymex Crude, IPE Crude, Brent etc.) pot fi considerate serii temporale. Analiștii economici vor studia aceste serii pentru a consilia companiile vânzătoare de petrol privind prognoza consumului de petrol ținând cont de diferite condiții sezoniere sau de lungă durată (e.g. sezonul cald sau rece, schimbări politice, tendințele de creștere economică regională etc.)

Din punct de vedere grafic, seriile temporale se vizualizează cel mai adesea prin tabele sau prin grafice, e.g. tabelul sau graficul fluxului de călători într-un aeroport într-un anumit interval de timp, graficul temperaturilor într-o anumită lună a anului, înregistrate în ultimul secol, graficul frecvenței apariției unei anumite boli pe o anumită perioadă de timp, graficul valorilor unor analize medicale ce se efectuează periodic, graficul evoluției ratei de schimb valutar etc. Prezentăm, pentru exemplificare, o asemenea reprezentare grafică referitoare la domeniul medical. În tabelul de mai jos sunt prezentate valorile pentru principalele enzime serice ce indică leziunile colestastice și hepatocelulare (în cazul afecțiunilor hepatici). Aceste observații medicale sunt în măsură să indice diagnosticul de hepatită cronică de tip C. Concret, este

vorba de analizele prelevate timp de 12 luni consecutive de la pacienți având hepatită cronică C, aceste analize lunare referindu-se la următoarele enzime serice: *alanină transaminasă (ALT)*, *aspartate transaminasă (AST)* și *gamma glutamyl transpeptidiasă (γ -GT)*.

Enzime serice	Ian	Feb	Mar	Apr	Mai	Iun	Iul	Aug	Sep	Oct	Nov	Dec
<i>ALT</i>	29	51	56	10	22	12	10	8	12	30	24	15
<i>AST</i>	29	50	39	15	22	12	14	19	11	8	6	12
γ - <i>GT</i>	67	40	127	55	98	86	74	48	55	44	34	65

În figura de mai jos am reprezentat graficele curbelor anuale, corespunzătoare celor trei enzime serice: (*ALT*) – Seria 1, (*AST*) – Seria 2 și (γ -*GT*) – Seria 3.



Revenind la problemele importante care privesc seriile temporale, vom enumera următoarele chestiuni mai cunoscute:

- *Prognoza (forecasting)*, care se referă la estimarea unor valori viitoare ξ_{T+h} , $h > 1$, pe baza datelor cunoscute $\xi_1, \xi_2, \dots, \xi_T$. De multe ori, în loc să se indice o anumită valoare prognozată, se indică un interval de prognoză;
- *Analiza tendinței (trend analysis)*, care se referă la faptul că, atunci când studiem mai multe serii temporale care vizează aceeași perioadă de timp, este uneori necesar să analizăm cauzele în care două sau mai multe asemenea serii au aceeași direcție (sunt puternic corelate), chiar dacă la prima vedere acest fapt pare inexplicabil;
- *Descompunerea sezonieră (seasonal decomposition)*, care se referă la descompunerea seriei temporale inițiale, privită ca un amalgam de tendințe diferite ce îngreunează identificarea pattern-ului de bază, în

componente bine structurate și analizarea acestora în inter-dependența lor globală. În general, o serie temporală poate fi descompusă în: (a) o componentă sezonieră S_t , (b) o componentă de trend T_t , (c) o componentă ciclică C_t și (d) o componentă neregulată –eroarea aleatoare ε_t . Menționăm aici două modele de descompunere sezonieră: (1) *modelul aditiv* de ecuație $\xi_t = T_t \cdot C_t + S_t + \varepsilon_t$ și (2) *modelul multiplicativ* de ecuație $\xi_t = T_t \cdot C_t + S_t \cdot \varepsilon_t$;

- *Distincția* între observațiile pe *tempor* scurt și cele pe *tempor* îndelungat, ce se referă la separarea dintre relațiile persistente în timp, observate la datele culese și relațiile conjuncturale.
- *Relația de cauzalitate*, care se poate observa între două sau mai multe serii temporale. Mai mult, în cazul determinării unei relații de cauzalitate, se studiază factorul de defazare care intervine între cauză și efect, privind seriile implicate.

Seriile temporale mai sunt cunoscute și sub numele de *serii dinamice*, datorită faptului că ilustrează cinetica (dinamica) unui fenomen evoluând în timp real. De aceea, în conexiune cu seriile temporale, putem vorbi și de *modelele dinamice*, adică modelele ce captează „mișcarea” unui anumit fenomen în raport cu timpul.

Din multitudinea de modele dinamice bazate pe seriile temporale, vom aminti aici doar trei tipuri: *modelele de ajustare*, *modelele autopredictive* și *modelele explicative*. În continuare, vom prezenta pe scurt principiile fiecăruia din cele trei tipuri de modelări.

A. Modele de ajustare. În acest caz, pe baza observațiilor obținute analizând datele reale, putem formula un model matematic ilustrat de o ecuație de forma:

$$\xi_t = f(t, u_t),$$

unde f este o funcție determinată de un număr finit de parametri necunoscuți, iar u_t reprezintă o variabilă aleatoare de medie zero, aleasă în funcție de situația reală modelată (factor de disturbare/zgomot/eroare). Vom menționa că ipotezele asupra variabilei u_t ca și estimarea funcției f se fac plecând de la așa-numitele *ajustări globale*, în care toate observațiile se bucură de aceeași considerație, având roluri egale în estimării, sau așa-numitele *ajustări locale*, în care fiecare observație își are rolul său în determinarea parametrilor modelului.

B. Modele autopredictive. În aceste modele se presupune că prezentul este influențat de trecut, deci matematic vorbind, un asemenea model este ilustrat de o ecuație de forma:

$$\xi_t = f(\xi_{t-1}, \xi_{t-2}, \dots, u_t),$$

unde u , reprezintă și aici factorul de disturbare, fiind reprezentat de o variabilă aleatoare.

C. Modele explicative. În cazul acestor modele, ecuația capătă forma:

$$\xi_t = f(x_t, u_t),$$

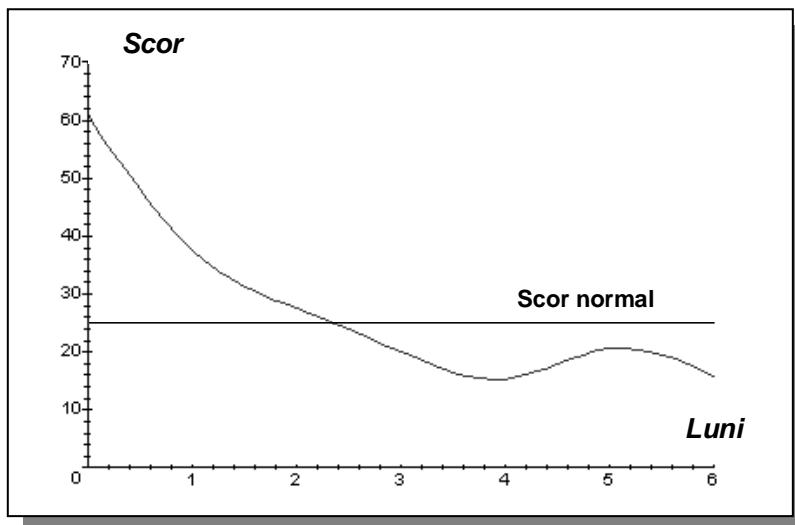
unde x_t este o variabilă observabilă, numită și *variabilă exogenă*, iar u_t reprezintă din nou factorul disturbant. În principal, aceste modele se împart în două cazuri: cele *static*, în care variabila exogenă x_t nu conține informații din trecutul lui ξ_t , iar u_t sunt mutual independente, și cele *dinamice* în care fie x_t conține informații privind trecutul lui ξ_t , fie u_t sunt autocorelate.

Vom menționa aici, pentru exemplificare, doar cazul unui model clasic autopredictiv, și anume *modelul ARIMA* (*Auto-Regressive Integrated Moving Average*), sau, cum mai este cunoscut, *modelul Box-Jenkins*. Acest model, ale cărui baze au fost puse de către Box și Jenkins în 1976 [18], reprezintă, cel puțin teoretic, cea mai largă clasă de modele de prognoză (*forecasting*) cunoscută, fiind în zilele noastre extrem de popular în diverse aplicații, datorită marii sale puteri și flexibilități. Astfel, din bogata clasă a modelelor de tip *ARIMA* (p, d, q), vom considera pentru o aplicație medicală simplă doar aşa-numitul model ‘mixt’ *ARIMA* (1, 1, 1), dat de ecuația:

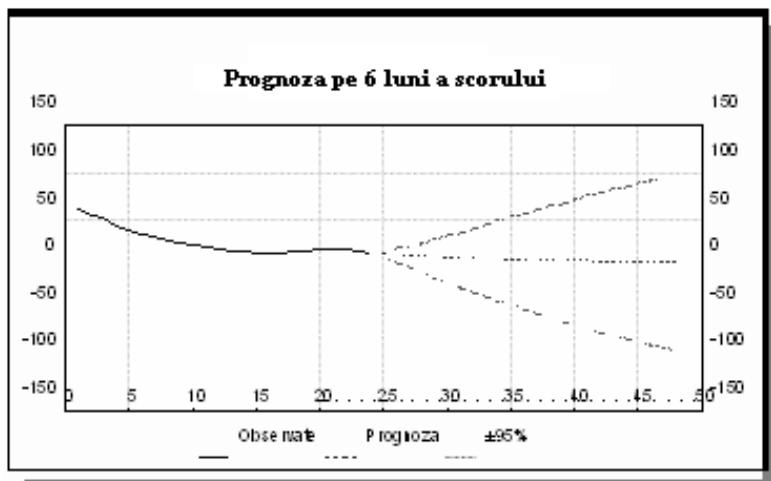
$$\hat{Y}(t) = \mu + Y(t-1) + \Phi \cdot (Y(t-1) - Y(t-2)) - \theta \cdot \varepsilon(t-1),$$

unde $\hat{Y}(t)$ reprezintă prognoza pentru seria temporală la momentul t , μ reprezintă o constantă, Φ denotă coeficientul de regresie, $\varepsilon(t-1)$ denotă eroarea la momentul $(t-1)$, iar θ reprezintă coeficientul erorii de decalaj al prognozei. Vom ilustra modelul de mai sus printr-o aplicație concretă privind un tratament clasic în bolile hepatice. Să considerăm, aşadar, studiul tratamentului cu interferon al hepatitei cronice *C* [138]. Astfel, vom fi interesați de prognoza evoluției stării unui pacient având hepatită cronică *C*, care a urmat un tratament standard de 6 luni cu interferon. Importanța prognozei rezidă în faptul că, pe de-o parte tratamentul cu interferon este foarte costisitor și, pe de altă parte, o fracțiune deloc neglijabilă dintre pacienți nu răspund pozitiv la acest tratament, continuarea acestuia nefiind eficientă din toate punctele de vedere. Din această cauză este importantă crearea unui model predictiv, suficient de sigur, care să prevadă comportamentul clinic al unui pacient în următoarele 3 – 6 luni, pentru a se decide astfel dacă este vorba de un aşa-numit *responder*, adică un pacient cu răspuns pozitiv la tratament, sau un *non-responder*, adică un pacient cu răspuns negativ la tratament, pentru care continuarea tratamentului nu-și mai

are rostul. Pentru modelarea comportamentului clinic al pacienților a fost ales clasicul model ‘mixt’ ARIMA de tipul (1, 1, 1). Deoarece procesarea computerizată a acestui model, ca de altfel și gradul de încredere în prognoză, cere un număr relativ ridicat de date temporale (de la 20 în sus), a fost folosit procedeul de *interpolare B-spline cubică*, care induce curbe suficient de netede la noduri, fiind astfel compatibile cu procesul clinic asociat tratamentului. Pornind de la datele înregistrate în primele 6 luni, prin divizarea perioadei de timp respective și interpolarea nodurilor obținute, s-a creat baza de date suficientă pentru aplicarea modelului ARIMA. Prezentăm mai jos graficul curbei de interpolare B-spline pentru scor, care reprezintă singura posibilitate de prognozare a evoluție stării de sănătate a unui pacient.



Plecând de la aceste date procesate prin interpolarea B-spline, prezentăm mai jos curba de prognoză ARIMA și intervalul de încredere (95%) corespunzător pentru următoarele 6 luni (săptămânilor 25-49), privind comportamentul clinic virtual al unui pacient de tip responder.



3.7. Tehnici exploratorii multivariate

Tehnicile exploratorii multivariate reprezintă acele metode special proiectate pentru a descoperi pattern-uri ascunse în datele multidimensionale, cuprinzând, printre altele: *analiza factorială* (*Factor Analysis*), *analiza componentelor principale* (*Principal Components Analysis*), *analiza canonică* (*Canonical Analysis*) și *analiza discriminant* (*Discriminant Analysis*). Alte tehnici aplicabile datelor multidimensionale, precum seriile temporale, modelele liniare, neliniare și aditive, au fost deja menționate anterior, iar altele, precum analiza cluster și arborii de clasificare, vor fi dezbatute ulterior în această carte. Să mai menționăm aici și alte tehnici, care nu vor fi prezentate în această carte, precum analiza corespondențelor, modele generale CHAID, analiza componentelor principale, scalarea multidimensională, teoria siguranței în funcționare și care, la rândul lor, sunt utilizate adesea în Data Mining.

3.7.1. Analiza factorială

Analiza factorială este privită din punctul de vedere DM, în principal pentru rezolvarea următoarelor două probleme:

- *Reducerea numărului de atribute ale obiectelor în vederea măririi vitezei de procesare a datelor;*
- *Detectarea structurilor ascunse în relațiile dintre date, în vederea clasificării atributelor obiectelor.*

Analiza factorială (termen introdus de către Thurstone, 1931) se referă la o varietate de tehnici statistice utilizate în scopul reprezentării unui set de variabile în funcție de un număr mai redus de variabile ipotetice, numite *factori*. Un exemplu simplu pentru ilustrarea rolului de reducere a datelor și identificarea structurii relațiilor, obținut prin utilizarea analizei factoriale, este cel privitor la procesul de construire a tipologiei consumatorului standard al unui supermarket. Astfel, în construirea bazei de date a consumatorilor pot apărea simultan atribute ca, de exemplu, venitul anual și impozitul anual. Deoarece cele două atribute sunt corelate prin formula de deducere a impozitului din venituri, este suficient numai unul, celălalt fiind redundant, deci poate fi îndepărtat fără pierdere de informație. Datorită faptului că cele două atribute (variabile) sunt corelate, relația dintre ele este foarte bine rezumată de dreapta de regresie ce trece prin „norul” punctelor generate de perechile de date, putând fi folosită aşadar pentru detectarea structurii (liniare) a relației dintre ele. În fapt, astfel reducem cele două variabile la un singur factor, acesta fiind o combinație liniară a celor două variabile inițiale.

Remarcă: Pentru mai mult de două variabile, filosofia reducerii variabilelor la un singur factor rămâne aceeași. Astfel, pentru cazul a trei variabile (corelate), de pildă, putem considera „dreapta” lor de regresie (dreaptă ce trece prin „norul” punctelor generate de tripletele de date, în „spațiul” tri-dimensional creat de ele) și astfel le reducem la un singur factor – o combinație liniară a lor.

În concluzie, aşa cum am mai spus, analiza factorială reprezintă acea metodologie statistică care rezumă variabilitatea dintre atributele date, privite ca variabile aleatoare, cu ajutorul unui număr restrâns de alte variabile – *factori*. Atributele (variabilele) luate în considerație sunt exprimate prin combinații liniare ale factorilor la care se adaugă și un termen desemnând *eroarea* modelului. Să amintim că analiza factorială a fost și este intens utilizată în diferite domenii ca, de exemplu, psihologie (e.g. psihometrie – C. Spearman), științele sociale, marketing, managementul producției, cercetări operaționale etc.

Pentru a înțelege mai bine modul de lucru în analiza factorială, să considerăm următorul exemplu. Să presupunem că staff-ul unui lanț de supermarket-uri dorește să măsoare gradul de satisfacție a cumpărătorilor relativ la serviciile oferite. Pentru aceasta se consideră doi „factori” de estimare ai satisfacției: (a) gradul de satisfacție privind modul de servire a unui client și (b) gradul de satisfacție privind calitatea produselor comercializate. Pentru aceasta se efectuează un sondaj printre, să zicem, $N = 1000$ de clienți fideli, care trebuie să dea răspuns unui chestionar cu, să zicem, $M = 10$ întrebări „cheie” care pot măsura gradul de satisfacție a clienților. Vom considera fiecare răspuns al unui client ca un „scor” referitor la chestiunea respectivă, acesta fiind considerat ca o variabilă *observată*. Deoarece clienții au fost selectați aleator dintr-o „populație” numeroasă, se poate presupune că cele 10 răspunsuri

(scoruri) reprezintă variabile aleatoare. Să presupunem, de asemenea, că scorul mediu *per client per întrebare* poate fi privit ca o combinație liniară a celor două tipuri de satisfacții (factori de satisfacție –variabile *neobserve*), de exemplu, pentru întrebarea $\#k$, $k = 1, 2, \dots, 10$, avem: $\{7*satisfacție\ serviciu + 5*satisfacție\ produs\}$, unde numerele 7 și 5 se numesc *încărcările factorilor (factor loadings)* și sunt identice pentru toți clienții. Să remarcăm că pot exista încărcări diferite pentru chestiuni diferite, 7 și 5 fiind încărcările factorilor relativ la chestiunea $\#k$. Doi clienți cu același grad de satisfacție în cele două direcții pot avea scoruri diferite la aceeași întrebare din chestionar, deoarece părerile individuale diferă față de medie, această diferență reprezentând *eroarea*.

Vom prezenta în continuare modelul matematic corespunzător analizei factoriale, adaptat exemplului de mai sus. Astfel, pentru fiecare client i din cei N clienți, cele M scoruri sunt date de ecuațiile:

$$x_{1,i} = b_1 + a_{1,1} \cdot s_{1i} + a_{1,2} \cdot s_{2i} + \varepsilon_{1,i}, \quad i = 1, 2, \dots, N$$

$$x_{2,i} = b_2 + a_{2,1} \cdot s_{2i} + a_{2,2} \cdot s_{2i} + \varepsilon_{2,i}, \quad i = 1, 2, \dots, N$$

.....
.....

$$x_{M,i} = b_M + a_{M,1} \cdot s_{Mi} + a_{M,2} \cdot s_{Mi} + \varepsilon_{M,i}, \quad i = 1, 2, \dots, N$$

unde:

- $x_{k,i}$ reprezintă scorul corespunzător întrebării $\#k$ pentru clientul $\#i$;
- s_{1i} reprezintă gradul de satisfacție privind modul de servire a clientului $\#i$;
- s_{2i} reprezintă gradul de satisfacție privind calitatea produselor comercializate a clientului $\#i$;
- b_{kj} reprezintă încărcările factorilor pentru întrebarea $\#k$ corespunzătoare factorului j , $j = 1, 2$;
- ε_{ki} reprezintă eroarea (i.e. diferența între scorul clientului $\#i$ pentru chestiunea $\#k$ și scorul mediu corespunzător chestiunii $\#k$ pentru toți clienții ale căror satisfacții referitoare la servicii și produse sunt aceleași ca la clientul $\#i$);
- b_k reprezintă niște constante aditive.

În limbaj matricial, ecuațiile de mai sus se transpun în următoarea ecuație:

$$X = b + AS + \varepsilon,$$

unde:

- X este matricea variabilelor aleatoare *observe*;

- b este vectorul constantelor *neobserve*;
- A este matricea încărcărilor factorilor (constante *neobserve*);
- S este o matrice de variabile aleatoare *neobserve*;
- ε este o matrice de variabile aleatoare *neobserve* (matricea *erorilor*).

Analiza factorială își propune estimarea matricei A a încărcărilor factorilor, a vectorului mediilor b și a dispersiei erorilor ε .

Remarcă: Trebuie subliniată distincția între filosofia din spatele tehnicielor de analiza factorială și aplicarea efectivă a acestor tehnici pe date concrete. Practic, analiza factorială se poate aplica numai utilizând programe specializate pe această topică. Pentru detalii tehnice privind principiile analizei factoriale, cititorul poate consulta, de pildă, [109], [110], în timp ce software specializat poate fi găsit în pachetele *Statistica*, SAS, SPSS etc.

3.7.2. Analiza componentelor principale

În conjuncție cu analiza factorială, vom prezenta acum câteva aspecte privind analiza componentelor principale (*Principal components analysis - PCA*), deoarece aceasta din urmă poate fi privită ca o tehnică de analiză factorială, atunci când dispersia totală a datelor este luată în considerație.

În principiu, analiza componentelor principale are ca scop reducerea numărului de variabile utilizate inițial, luând în considerație un număr mai mic de variabile „reprezentative” și necorelate. Ca o consecință a acestui demers, se obține o clasificare a variabilelor și cazurilor.

Pentru a înțelege mai bine despre ce este vorbă, să considerăm că dorim să cumpărăm un anumit produs dintr-un magazin și, pentru început, suntem interesați doar de două caracteristici ale sale, A și B. În acest caz putem considera „norul” de împărăștiere al punctelor generate de perechile de date corespunzătoare celor două atribute, după care vom considera dreapta care traversează centrul „norului” de puncte (în particular, centroidul „norului”), deci dreapta lor de regresie care este reprezentativă pentru cele două atribute. Să presupunem acum că luăm în considerație încă o caracteristică a produsului, notată C. Dacă în acest caz vom lua în considerație doar perechile de regresii între cele trei atribute, nu obținem o alegere satisfăcătoare, deoarece nu avem o imagine de ansamblu asupra tuturor celor trei atribute. Avem în schimb nevoie de ceva care să „însumeze” toate cele trei atribute deodată. Problema se complică și mai mult dacă luăm în considerație un număr și mai mare de atribute, în acest caz având nevoie, în loc de perechi de regresii, de un „scor” care să caracterizeze obiectul respectiv. Din punct de vedere geometric, acest scor poate fi generat de o dreaptă sau drepte (axe factor) care să treacă prin

centroidul „norului” de puncte generat de *tuplele* de date. Astfel, plecând de la spațiul inițial al datelor, se consideră un subspațiu generat de un set de axe noi, numite *axe factor (factor axes)*, subspațiu în care este proiectat spațiul inițial. În principiu, tehnica PCA caută dreapta care se „potrivește” cel mai bine „norului” de puncte din spațiul vectorial al instanțelor și atributelor. Matematic vorbind, considerând p atribute și q instanțe, tehnica PCA de identificarea a factorilor se referă la diagonalizarea matricei simetrice reprezentând matricea corelațiilor (matricea covarianțelor). Reamintim că, deoarece covarianța se calculează doar pentru perechi de variabile statistice, în cazul a trei variabile X , Y și Z , matricea covarianțelor este dată de:

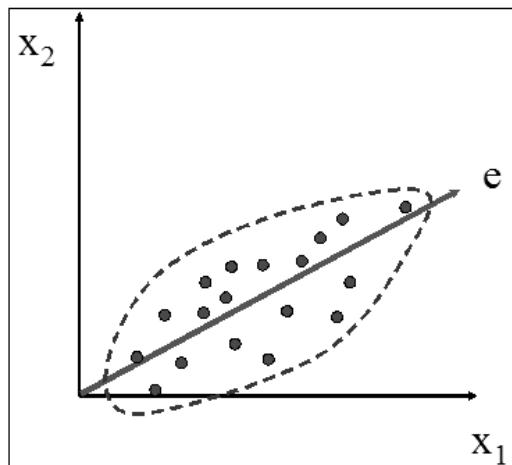
$$\text{Cov}(X, Y, Z) = \begin{pmatrix} \text{cov}(X, X) & \text{cov}(X, Y) & \text{cov}(X, Z) \\ \text{cov}(Y, X) & \text{cov}(Y, Y) & \text{cov}(Y, Z) \\ \text{cov}(Z, X) & \text{cov}(Z, Y) & \text{cov}(Z, Z) \end{pmatrix},$$

pentru cazul a n variabile se va proceda analog. În cazul de față, dacă matricea standardizată (adică centrată în raport cu mediile respective) X reprezintă datele corespunzătoare celor q instanțe și p atribute, atunci $X^t \cdot X$ reprezintă matricea covarianțelor și problema care se pune este diagonalizarea acesteia.

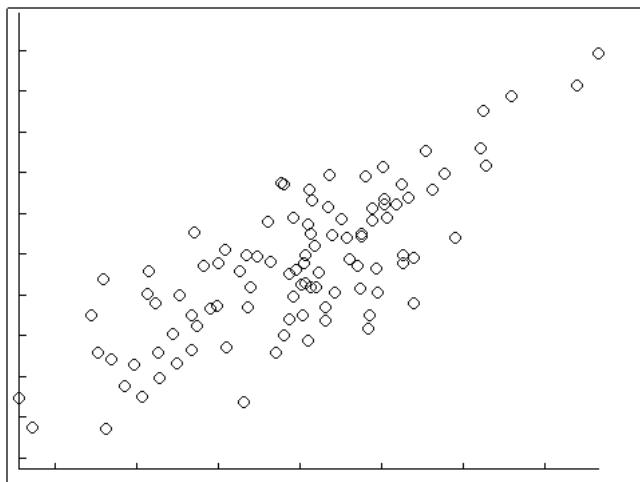
Rezultatul va fi un nou set de variabile – *componentele principale* – care sunt combinații liniare ale atributelor inițiale și sunt necorelate. Se obține astfel un spațiu de dimensiune mai mică, în care se proiectează instanțele și atributele și care păstrează maximum din variabilitatea datelor. Schematic, PCA poate fi rezumată în următorii pași:

- ⇒ identificarea vectorilor proprii ai matricei covarianțelor;
- ⇒ construirea noului spațiu generat de vectorii proprii.

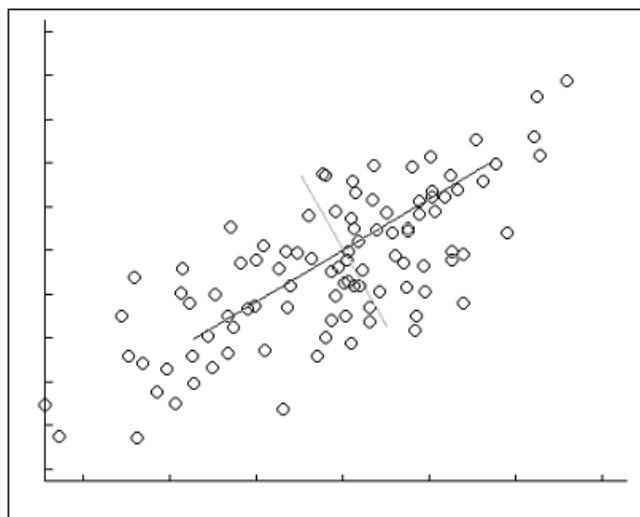
Figura de mai jos ilustrează în mod sintetic principiul metodologiei PCA



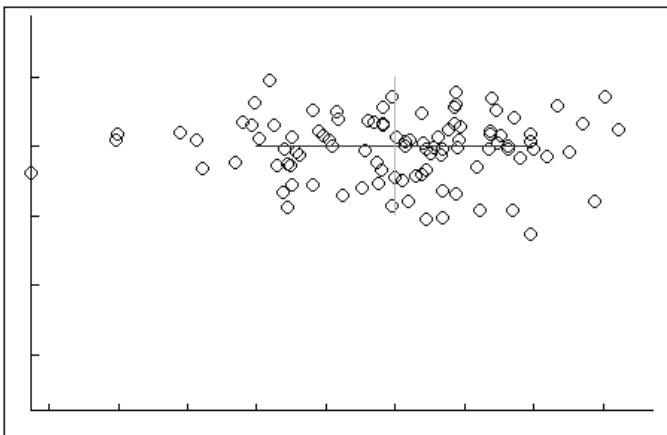
Vom ilustra schematic, în figurile următoare, pașii unei analize PCA. Tehnic vorbind, prima componentă principală reprezintă combinația variabilelor care „explică” cea mai mare dispersie a datelor. A doua componentă principală „explică” următoarea dispersie maximă a datelor, fiind independentă de prima și.a.m.d., putând considera, în principiu, atâțea componente principale câte variabile există. În prima figură de mai jos, prezentăm setul de date sub forma „norului” lor de împrăștiere.



După calcularea matricei covarianțelor și a celor două componente principale, figura următoare ilustrează noua configurație a spațiului datelor.



Dreapta oblică (cea pronunțată), care trece prin centrul „norului” punctelor, explicând cea mai mare dispersie, reprezintă prima componentă principală. A doua componentă principală este perpendiculară pe prima (independentă de aceasta) și explică restul de dispersie. În final, prin înmulțirea datelor inițiale cu componentele principale, datele vor fi rotite, astfel încât componentele principale formează axele noului spațiu, așa cum se observă în figura de mai jos.



În încheiere, să amintim că PCA mai este cunoscută și ca *transformarea Hotelling (Hotelling transform)* sau *transformarea Karhunen-Loève (Karhunen-Loève transform -KLT)*. Pentru amănunte privind calculele și principaliii algoritmii utilizăți (e.g. *metoda covarianței*, *metoda corelației*) cititorul este îndemnat să consulte [102] sau materialele „*A tutorial on Principal Components Analysis*”, L. Smith URL: (http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf), „*PCA - Principal Component Analysis: Introduction, Simple Example, MATLAB code*” URL: (<http://www.eng.man.ac.uk/mech/merg/Research/datafusion.org.uk/pca.html>).

3.7.3. Analiza canonica

Să presupunem că staff-ul unui supermarket este interesat să analizeze gradul de satisfacție a clienților săi relativ la modul de servire. Pentru aceasta, clienții sunt rugați să completeze un chestionar cu un anumit număr de întrebări relative la satisfacția privind modul de servire. În același timp, ei sunt de asemenea rugați să răspundă la alt set de întrebări, care se referă la măsurarea gradului de satisfacție în alte domenii decât al servirii (e.g. calitatea produselor, diversitatea produselor etc.). Problema care se pune este aceea de a identifica posibile conexiuni între satisfacția față de servire și satisfacția față de alte

aspecte ale activității supermarketului. Să ne reamintim acum de principiile regresiei (liniare) simple sau multiple. Acolo aveam de a face cu un set format dintr-una sau mai multe variabile predictoare, explicative, și de o variabilă dependentă (criteriu) care era determinată de celelalte. Așa după cum ușor se observă, este cazul în care variabila dependentă ar fi reprezentată doar de gradul de satisfacție față de modul de servire. În cazul de față există însă și alte variabile dependente (calitatea produselor, diversitatea produselor etc.), astfel încât suntem în situația să avem un set de variabile explicative și un set de variabile dependente în loc de o singură variabilă dependentă. În acest caz, generalizând tehnica regresiei liniare multiple, suntem interesați să corelăm un set de variabile dependente, fiecare din ele fiind ponderată, cu un set de variabile predictoare, de asemenea ponderate. Menționăm că ideea este oarecum asemănătoare cu cea introdusă de H. Hotelling (1936), sub numele de *analiza canonică a corelațiilor* (*analiza canonică variată*).

Formal, având un set de variabile explicative $\{X_1, X_2, \dots, X_q\}$ și un set de variabile dependente $\{Y_1, Y_2, \dots, Y_p\}$, trebuie identificată ecuația:

$$a_1.Y_1 + a_2.Y_2 + \dots + a_p.Y_p = b_1.X_1 + b_2.X_2 + \dots + b_q.X_q,$$

care stabilește legătura dintre cele două seturi de variabile.

Din punct de vedere computațional, există programe specializate pentru rezolvarea acestei probleme (e.g. *Statistica –Multivariate exploratory techniques, SPSS*).

3.7.4. Analiza discriminant

Să presupunem că pentru stabilirea clasei unui uragan avem la dispoziție mai multe măsurători relative la diferite caracteristici meteorologice premergătoare declanșării uraganului (variabilele predictive discriminatorii). Studiul pe care îl efectuăm își propune să stabilească care variabile sunt cele mai bune predictoare ale clasei uraganului, deci care variabile fac efectiv distincție (*discriminare*) între diferitele categorii de uragane. Analog, în comerț putem analiza ce caracteristici (variabilele discriminatorii) fac diferență în ceea ce privește rațiunea pentru care un cumpărător alege dintre mai multe categorii de produse unul anume. În domeniul medical, de asemenea, un doctor este interesat ce caracteristici pot determina modul în care un pacient se poate vindeca complet, parțial sau deloc.

După cum se observă din exemplele de mai sus, analiza discriminant reprezintă practic o metodă de clasificare a unor obiecte în anumite clase pe baza analizei unui set de variabile predictoare –input-uri. Modelul se bazează, în principiu, pe un set de observații pentru care se cunosc *a priori* clasele, formând setul de antrenament. Pe baza antrenamentului, se construiește un set de funcții discriminant, de forma:

$$L_i = b_1.X_1 + b_2.X_2 + \dots + b_n.X_n + c, \quad i = 1, 2, \dots, k,$$

unde X_1, X_2, \dots, X_n sunt variabilele predictoare (care discriminează între clase), b_1, b_2, \dots, b_n reprezintă *coeficienții discriminant*, iar c este o constantă. Fiecare funcție discriminant L_i corespunde unei clase Ω_i , $i = 1, 2, \dots, k$, în care trebuie să partaționăm observațiile. O nouă instanță va fi clasificată în acea categorie pentru care funcția discriminant corespunzătoare ia valoarea maximă.

În același context, vom menționa și topicele echivalente: *analiza discriminant liniară* (*Linear Discriminant Analysis -LDA*), precum și *discriminarea liniară Fisher* (*Fisher's linear discriminant*), care sunt utilizate cu aceste terminologii în învățarea automată. Ca domenii de aplicații concrete pentru analiza discriminant vom aminti recunoașterea fețelor (*face recognition*), marketing (distincția între clienți, managementul produselor etc.), medicina etc., menționând existența programelor specializate necesare (e.g. *Statistica*, *SPSS* etc.).

3.8. OLAP

Tehnica OLAP (*On-Line Analytical Processing*) sau FASMI (*Fast Analysis of Shared Multidimensional Information*) sau OLTP (*On Line Transaction Processing*) a fost inițiată de către E.F. Codd, considerat părintele bazelor de date relaționale, și se referă la metoda care permite generarea on-line de rezumări descriptive și comparative (inclusiv sub formă vizuală) a datelor sau a altor interogări analitice, în cazul bazelor de date multidimensionale. OLAP este inclusă în domeniul mai larg al *Business Intelligence* (BI), aplicațiile sale tipice fiind circumschise categoriilor raportărilor de afaceri în vânzări (*business reporting for sales*), marketingului, raportărilor de management (*management reporting*), managementului performanței în afaceri (*business performance management -BMP*), alocării bugetare și programei (*budgeting and forecasting*), raportărilor financiare etc. Să menționăm că, în ciuda numelui său, nu este vorba întotdeauna de procesarea on-line a datelor, exceptând cazurile de actualizare dinamică a bazei multidimensională de date sau de eventuale interogări. Spre deosebire de bazele de date relaționale, în cazul OLAP se utilizează secvențe multidimensionale (*multidimensional array*) pentru reprezentarea datelor. Plecând de la setul de date tabelate, acestea se reprezintă sub forma unei secvențe multidimensionale, luând în considerație următoarele două aspecte principale:

- ⇒ Identificarea atât a atributelor care vor genera dimensiunile reprezentării cât și a celor care vor reprezenta attributele „țintă” și ale căror valori vor fi întrări în secvența multidimensională.
 - Atributele reprezentând dimensiunile trebuie să fie variabile discrete;
 - Valoarea „țintă” este de obicei continuă sau valoare de contorizare;

- Nu este absolut necesară existența variabilelor „țintă”, cu excepția cazului contorizării instanțelor care au același set de valori ale atributelor.
- ⇒ Găsirea valorii fiecărei intrări în secvența multidimensională, obținută prin însumarea valorilor atributelor „țintă” sau contorizarea tuturor instanțelor care au valorile atributelor corespunzătoare acelei intrări.

Operațiunea de bază în cazul tehnologiei OLAP este construirea „cubului” de date. Prin „cub” de date se înțelege o reprezentare multidimensională a datelor sub formă de cub (evident că doar în spațiul tridimensional este vorba de un cub, dar ideea unui „cub” poate fi extrapolată pentru mai mult de trei dimensiuni –hipercub multidimensional), împreună cu toate totalurile posibile. Prin toate totalurile posibile se înțeleg *totalurile (aggregates)* rezultate prin însumarea în raport cu dimensiunile rămase după îndepărțarea unui anumit set de dimensiuni alese, ținând seama de un scop anume.

În principiu, există trei tipuri de tehnici OLAP:

- OLAP *multidimensional* (MOLAP), care reprezintă forma „clasică” a OLAP și utilizează structuri de baze de date ca, de exemplu, perioade temporale, locație, coduri de produse etc., reprezentând atrbute uzuale în domeniul respectiv; maniera în care fiecare dimensiune este totalizată este predefinită printr-o sau mai multe ierarhii.
- OLAP *relational* (ROLAP), care utilizează direct bazele de date relaționale, datele de bază și tabelele dimensiune fiind stocate ca tabele relaționale și tabele noi fiind create pentru a înregistra informațiile totalizate.
- OLAP *hbrid* (HOLAP), care presupune o bază de date, divizând datele într-o *stocare relatională* și o *stocare specializată* (e.g. tabele relaționale pentru anumite tipuri de date și, pe de altă parte, stocări speciale pentru alte tipuri de date).

Remarcă: 1. MOLAP este utilizat mai ales în cazul bazelor de date mai reduse (calcul rapid al totalurilor și răspunsuri eficiente, spațiu redus de stocare); ROLAP este considerat mai accesibil, cu toate că un volum mare de date nu poate fi procesat eficient, iar performanța în a răspunde la interogări nu este prea strălucită; HOLAP, ca tehnică hibridă, se situează între cel două de mai înainte, având o viteză de procesare rapidă și accesibilitate bună.

2. Dificultățile privind implementarea tehnologiei OLAP provin fie din construirea interogărilor, fie din alegerea datelor de bază, calitatea datelor și dezvoltarea schemei. În acest sens, menționăm

faptul că produsele OLAP actuale sunt livrate împreună cu librării uriașe de interogații pre-configure.

Prezentăm, în tabelul de mai jos, câteva dintre cele mai cunoscute produse comerciale OLAP.

<ul style="list-style-type: none">• Applix TM1• Beyond 20/20• Business Objects• Celequest• Cognos• DataWarehouse Explorer (CNS International)• Hyperion Solutions Corporation• IBM's DB2 Cube Views• TM1• Sagent• OLAP ModelKit™	<ul style="list-style-type: none">• Information Builders WebFOCUS• Microsoft Analysis Services (SQL Server)• MicroStrategy• Oracle Discoverer for OLAP/SiebelAnalytics• ProClarity• SAP BW• Essbase, Mondrian și produse de la SAS Institute• Systems Union• Holos
--	--

Ca să ne facem o idee despre dezvoltarea actuală a pieței produselor OLAP, prezentăm mai jos topul valorilor pieței bursiere pentru produsele comerciale OLAP (2005 – OLAP Report, URL:<http://www.olapreport.com/market.htm>):

1. Microsoft - 28.0%
2. Hyperion Solutions Corporation - 19.3%
3. Cognos - 14.0%
4. Business Objects - 7.4%
5. MicroStrategy - 7.3%
6. SAP AG - 5.9%
7. Cartesis SA - 3.8%
8. Systems Union/MIS AG - 3.4%
9. Oracle Corporation - 3.4%
10. Applix - 3.2%

Vom ilustra principiile exprimate mai sus, relative la tehnica OLAP, cu un exemplu clasic [162], referitor la baza de date pentru planta *Iris* (R.A. Fisher -1936; URL: <http://www.ics.uci.edu/~mlearn/MLRepository.html>).

În figura de mai jos este prezentată o imagine a acestor plante, aleasă pentru „ilustrarea artistică” a bazei de date.



Baza de date respectivă conține informații privind trei tipuri de flori de *iris* (150 flori):

- ★ *Setosa*;
- ★ *Virginica*;
- ★ *Versicolor*;

precum și patru attribute:

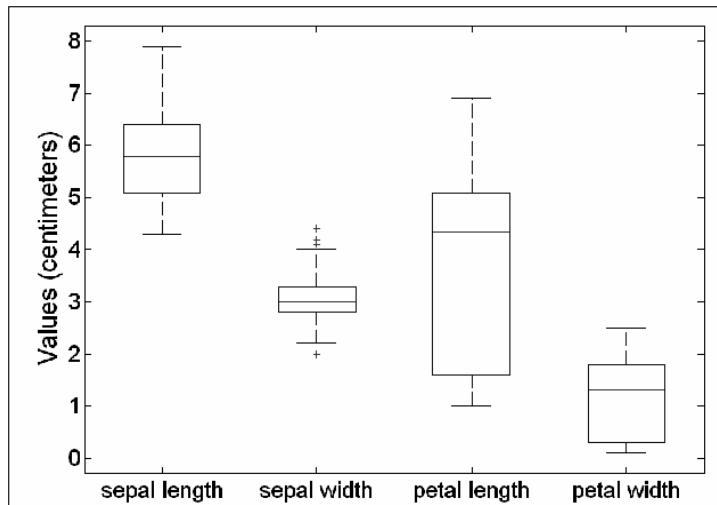
- Lungimea sepalei;
- Lățimea sepalei;
- Lungimea petalei;
- Lățimea petalei.

În figura de mai jos este reprezentat un „eșantion” din această bază de date, pregătit în vederea procesării DM.

■ Data: Irisdat.sta (5v by 150c)

	Fisher (1936) iris data: length & width of sepals and petals, 3 types of Iris				
	1 SEPALLEN	2 SEPALWID	3 PETALLEN	4 PETALWID	5 IRISTYPE
1	5	3.3	1.4	0.2	SETOSA
2	6.4	2.8	5.6	2.2	VIRGINIC
3	6.5	2.8	4.6	1.5	VERSICOL
4	6.7	3.1	5.6	2.4	VIRGINIC
5	6.3	2.8	5.1	1.5	VIRGINIC
6	4.6	3.4	1.4	0.3	SETOSA
7	6.9	3.1	5.1	2.3	VIRGINIC
8	6.2	2.2	4.5	1.5	VERSICOL
9	5.9	3.2	4.8	1.8	VERSICOL
10	4.6	3.6	1	0.2	SETOSA
11	6.1	3	4.6	1.4	VERSICOL
12	6	2.7	5.1	1.6	VERSICOL
13	6.5	3	5.2	2	VIRGINIC
14	5.6	2.5	3.9	1.1	VERSICOL
15	6.5	3	5.5	1.8	VIRGINIC
16	5.8	2.7	5.1	1.9	VIRGINIC
17	6.8	3.2	5.9	2.3	VIRGINIC
18	5.1	3.3	1.7	0.5	SETOSA

Figura de mai jos ilustrează, pentru comparație, o vizualizare a principalelor caracteristici statistice (percentilele 10%, 25%, 75%, 90% și mediana) a celor patru atrbute ale florilor de *iris*, utilizând metoda „box plot”.

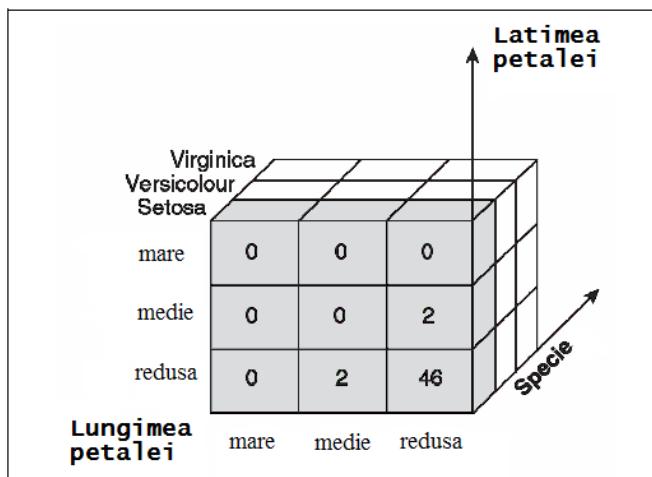


Luând în considerație doar două atribute: lungimea și lățimea petalelor, vom construi secvența multidimensională corespunzătoare, astfel:

- Vom discretiza atributele privind cele două dimensiuni – lungimea și lățimea petalelor (care sunt reprezentate de variabile continue) în trei categorii: *redusă*, *medie* și *mare*;
- Vom contoriza plantele în funcție de cele trei tipuri de dimensiune.

LUNGIMEA PETALEI	LĂȚIMEA PETALEI	TIPUL SPECIEI	CONTORIZARE
redusă	redusă	Setosa	46
redusă	medie	Setosa	2
medie	redusă	Setosa	2
medie	medie	Versicolor	43
medie	mare	Versicolor	3
medie	mare	Virginica	3
mare	medie	Versicolor	2
mare	medie	Virginica	3
mare	mare	Versicolor	2
mare	mare	Virginica	44

În acest mod, fiecare tripletă (*lungimea petalei*, *lățimea petalei*, *tipul speciei*) identifică un element din secvență, care apoi este contorizat pe ultima coloană. În figura următoare vizualizăm acest proces printr-un „cub” (tripletele nespecificate sunt cuantificate 0).



Principalele operațiuni OLAP sunt:

- *Felierea (slicing)*, adică selectarea unui grup de celule din „cub” prin indicarea unei valori specifice pentru una sau mai multe dimensiuni;
- *Tăierea (în cuburi) (dicing)*, adică selectarea unui grup de celule prin specificarea unui interval al valorilor unui atribut;
- *Agregarea (roll-up)*, adică însumarea valorile unor atribute în funcție de valorile altor atribute, obținând totaluri semnificative;
- *Divizarea (drill-down)*, adică, invers, împărțirea anumitor totaluri în subtotaluri semnificative.

De exemplu, în cazul „cubului” datelor referitoare la planta *iris*, prin operațiunea de „feliere” a cubului de mai sus, obținem următoarele tabele-felii, corespunzătoare celor trei specii, sugestive privind informațiile furnizate.

		Latime		
		redusa	medie	mare
Lungime	redusa	46	2	0
	medie	2	0	0
	mare	0	0	0

		Latime		
		redusa	medie	mare
Lungime	redusa	0	0	0
	medie	0	43	3
	mare	0	2	2

		Latime		
		redusa	medie	mare
Lungime	redusa	0	0	0
	medie	0	0	3
	mare	0	3	44

Remarcă: 1. Aşa cum am mai spus, deoarece în cazurile reale de procesare cu tehnologia OLAP există mult mai mult decât trei dimensiuni, termenul propriu utilizat este cel de „*hipercub*”.

2. Datele din hipercub pot fi actualizate oricând (eventual de diferite persoane), pot fi legate dinamic de alte hipercuburi și pot exista „alerte” în momentul în care anumite totaluri devin „expirate” datorită actualizărilor ulterioare.

În figura de mai jos prezentăm interfața produsului comercial OLAP ModelKit™.



The screenshot displays a complex OLAP cube interface with multiple dimensions and data levels. The visible dimensions are Year (1994, 1995, 1996, Total), Quarter (1994, 1995, 1996, Total), and CategoryName (Beverages, Condiments, Confections, Dairy Products, Grains/Cereals, Meat/Poultry, Seafood, Produce, Tot). The data is presented in a hierarchical grid format, with some cells containing percentages (e.g., 10.5% for Seafood in 1994). The bottom right corner of the interface shows a summary table with totals for each category across the years.

CategoryName	Year	Quarter	Total
Beverages	1994	1995	1996
Condiments	880.00	2720.00	1698.00
Confections	1117.00	3906.00	2883.00
Dairy Products	1433.00	4621.00	3095.00
Grains/Cereals	421.00	2562.00	1579.00
Meat/Poultry	641.00	2229.00	1210.00
Seafood	10.5%	7681.0	6290.8
Produce	7.8%	2990.0	4786.5
Meat/Poultry	13.2%	4199.0	7417.3
Grains/Cereals	7.4%	4562.0	4164.3
Seafood	Total	7681.0	7549.3
Seafood	Australia	886.0	4605.3
Seafood	Germany	640.0	11811.7
Seafood	Japan	701.0	56500.9
Seafood	USA	763.0	
Seafood	Total	29	
Meat/Poultry	Total	42	6290.8
Grains/Cereals	Total	45	4786.5
Dairy Products	Total	91	7417.3
Confections	Total	79	4164.3
Condiments	Total	52	9875.8
Beverages	Total	95	7549.3
Total	Total	51	4605.3
			11811.7
			56500.9

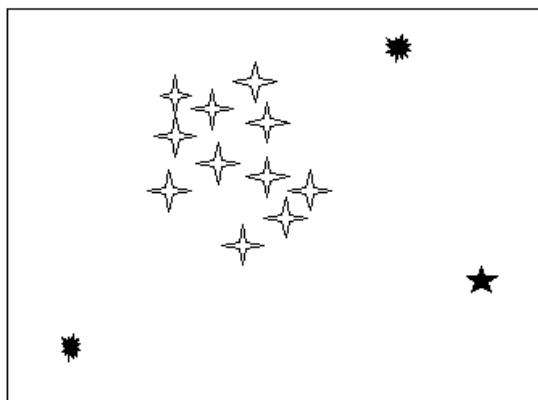
Listăm mai jos și câteva dintre cele mai cunoscute link-uri OLAP.

- *Pentaho BI and Reporting* (www.pentaho.org);

- *OLAP4All* (www.olap4all.com/);
- *Palo - Open-Source MOLAP* (www.palo.net/)
- *Mondrian-java-based, open source OLAP* (<http://mondrian.sourceforge.net/>)

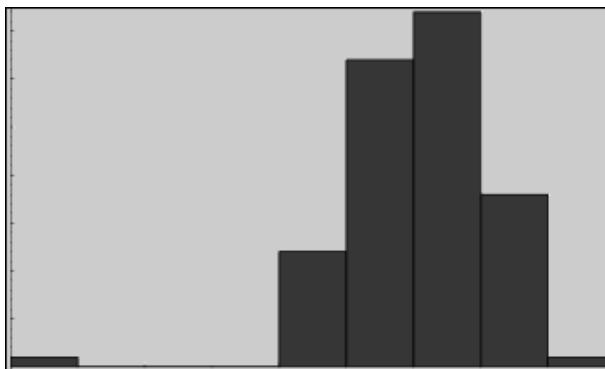
3.9. Detectarea anomaliiilor

Ultima chestiune pe care o abordăm în cadrul acestui capitol se referă la procesul de detectare a anomaliiilor sau valorilor extreme (*anomalies/outliers detection*). Cea mai simplă definiție pentru termenul de *anomalie/valoare extremă/valoare exceptională* este aceea dată în Statistică și care se referă la acea valoare care se găsește „foarte departe” de restul datelor, fiind, de fapt, o „singularitate” (un punct „izolat”) a setului de date. Apariția acestor valori în date este conceptual normală, chiar și repartitia gaussiană (normală) presupune existența lor (extremitățile „clopotului” lui Gauss). Aceste valori indică de cele mai multe ori fie date eronate în sine, fie date colectate eronat, fie chiar date corecte, generate în mod natural de fenomenul în cauză. Un exemplu simplu în acest sens poate fi cel referitor la datele reprezentând înălțimea unei populații. Cu toate că marea majoritate a indivizilor au înălțimi mai mult sau mai puțin apropriate de înălțimea medie, există și cazuri exceptionale de înălțimi foarte mici sau foarte mari. Un exemplu, de asemenea foarte cunoscut de apariție de valori anormale, este cel legat de efectuarea anumitor măsurători (e.g. viteza maxima înregistrată de vitezometrul unui automobil poate fi cu 10-15% mai mare decât cea reală). Figura de mai jos ilustrează aceste puncte izolate ale mulțimii datelor.

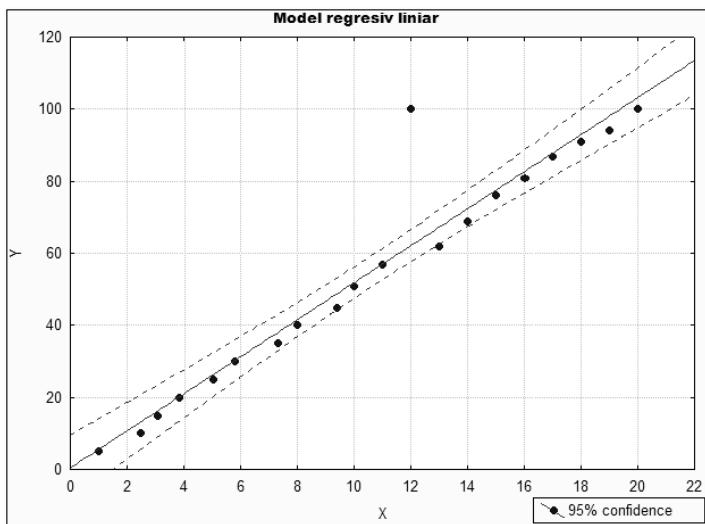


Trebuie remarcat faptul că repartitiile negaussiene, asimetrice (vezi paragraful 3.5.), pot prezenta astfel de valori (poziționate la „coada” curbei), examinarea

repartiției datelor și coroborarea acesteia cu natura problemei fiind necesare atunci când se hotărăște menținerea sau îndepărțarea lor.



Histograma de mai sus ilustrează foarte sugestiv modul cum putem utiliza reprezentarea grafică a datelor pentru identificarea valorilor excepționale. Figura de mai jos prezintă un model regresiv liniar cu o valoare extremă.



În încheierea discuției despre existența acestor valori nespecifice, este util să amintim că apariția lor în date poate influența semnificativ estimarea erorilor (e.g. eroarea Minkowski $-R$, metoda celor mai mici pătrate ($R = 2$) etc.).

Estimăriile și modelele care nu sunt influențate semnificativ de existența anomalieiilor se numesc *robuste* (e.g. *statistică robustă* –[103]).

Reamintim, în context că, de exemplu, mediana este mai robustă decât media la existența acestor valori (vezi exemplul privind datele referitoare la înălțime, subparagraful 3.2.1.); în cazul utilizării erorii Minkowski, $R = 1$ corespunde medianei, obținându-se o eroare mai mică decât pentru $R = 2$, corespunzând mediei [16]. Din această cauză, plecând de la valoarea medianei, se pot defini diferite tipuri de anomalii (e.g. dacă valoarea anomaliei este mai mare decât triplul medianei și mai mică decât de 5 ori mărimea acesteia, anomalia este de tipul I ș.a.m.d.). În cazul datelor multidimensionale, aceste comparații se pot face pe fiecare coordonată. Pentru mai multe detalii privind abordarea statistică a acestei teme, a se vedea și [123].

Printre aplicațiile pe care le are detectarea anomaliei, privite și prin prisma Data Mining, amintim:

- Detectarea fraudelor cu carduri bancare;
- Criptografie și detecția intruziunilor în diferite tipuri de rețele;
- Detecția defecțiunilor diferitelor sisteme;
- Procesarea sunetului și imaginilor;
- Detecția unor fenomene anormale, cu consecințe majore asupra echilibrului ecologic, local sau global (distrugerea stratului de ozon, poluarea etc.).

Atunci când inițiem o analiză pentru identificarea anomaliei în date (tehnica nesupervizată), se pleacă de la presupunția că vor exista mult mai multe valori „normale” în date (marea majoritate a lor) decât valori „anormale”. Plecând de la această ipoteză, există două etape ale procesului de detectare a anomaliei:

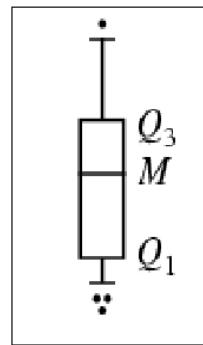
- ◆ Construcția profilului pattern-ului „normal” al datelor;
- ◆ Utilizarea acestui profil pentru identificarea anomaliei, pe baza măsurării diferenței față de „normalitate”.

Ca tehnici utilizate în acest proces, amintim:

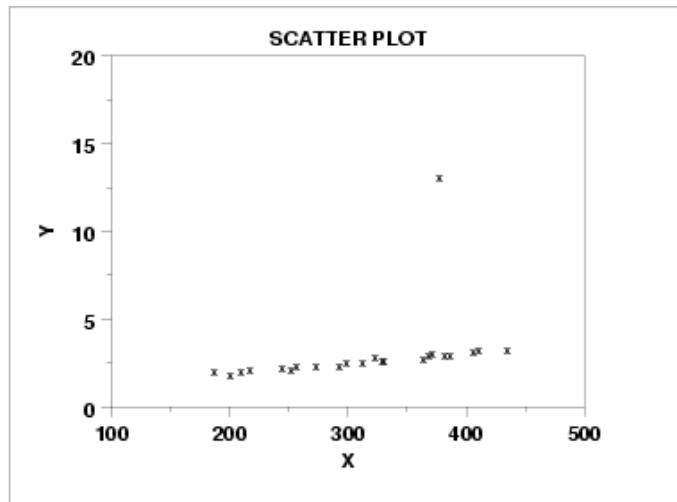
- Metode grafice;
- Metode statistice;
- Metode bazate pe măsurarea distanței;
- Metode bazate pe modele.

A. Dintre metodele grafice utilizate în detectarea anomaliei, cele mai utilizate sunt:

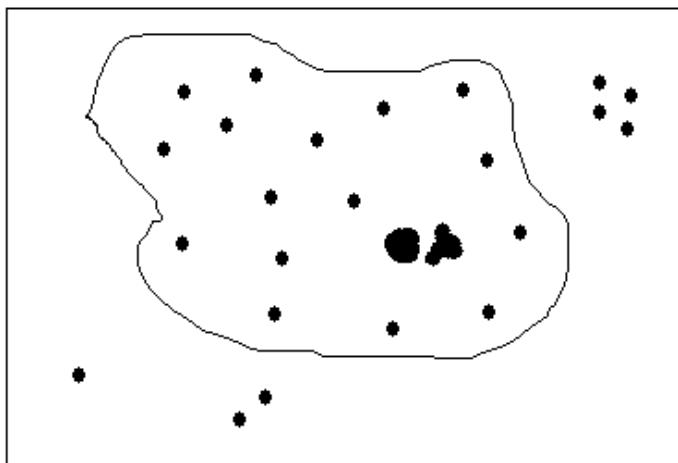
- ◆ Metoda *box-plot*, ilustrată mai jos.



- ◆ Metoda diagramelor de dispersie (*scatterplot*), ilustrată mai jos.



- ◆ Metoda acoperirii convexe (*convex hull*), ilustrată mai jos.



În acest caz, singura problemă notabilă este existența unor anomalii în interiorul acoperirii convexe (cele două „pete” negre din interior).

Metoda grafică are avantajul faptului că este sugestivă, dar, pe de altă parte este subiectivă, ceea ce poate duce la erori în detectarea valorilor anormale.

B. În ceea ce privește utilizarea metodelor statistice, în principiu se pleacă de la ipoteza privind un anumit model al repartiției datelor (pattern-ul datelor tipice), după care se utilizează diferite teste statistice pentru identificarea valorilor anormale în raport cu acest model, teste privind:

- Tipului repartiției;
- Parametrii repartiției;
- Intervalul de încredere.

În funcție de rezultatul analizării pattern-ului datelor referitoare la fenomenul studiat, se pot identifica acele eventuale valori atipice situației date (anomaliiile), pe baza testării statistice. În cele ce urmează vom prezenta două dintre cele mai cunoscute metode statistice de identificare a anomaliiilor.

◆ Testul Grubbs (*Grubbs' test-maximum normed residual test*) [89], [143] se aplică în cazul datelor univariate, bazându-se pe presupunerea că datele sunt normal repartizate, detectând o anomalie la fiecare pas. Concret, testul Grubbs compară ipotezele:

- H_0 : Nu există anomalii în date (*ipoteza nulă*)
- H_1 : Există cel puțin o anomalie în date (*ipoteza alternativă*),

utilizând statistica $G = \frac{\max |X - \bar{X}|}{SD}$, unde \bar{X} reprezintă media de sondaj, iar SD deviația standard (*two-sided test*). Astfel, se va respinge ipoteza nulă dacă $G > \frac{N-1}{N} \cdot \sqrt{\frac{t_{(\alpha/2N, N-2)}^2}{N-2+t_{(\alpha/2N, N-2)}^2}}$, unde t reprezintă repartiția *Student* iar $t_{(\alpha/2N, N-2)}^2$ valoarea critică a acesteia pentru cazul $(N-2)/2$ grade de libertate și un nivel de semnificație egal cu $\alpha/2N$ (pentru detalii, vezi [54]). La fiecare pas, utilizând testul de mai sus, se identifică câte o eventuală anomalie, se înlătură și se reiterează procedura (modificând parametrii) până nu mai există niciuna.

- ◆ Testul verosimilității (*likelihood test*) se aplică în cazul în care setul de date disponibile conține elemente care aparțin la două repartiții distincte, notate M –legea de repartiție a majorității datelor și A – legea de repartiție a anomalilor, care sunt presupuse minoritare. Deoarece setul de date conține elemente aparținând ambelor categorii, rezultă că repartiția globală a datelor, notată D , va fi una mixtă, ce poate avea forma unei combinații de tipul $D = (1-\alpha) \cdot M + \alpha \cdot A$. Concret, repartiția M se estimează pe baza datelor disponibile, utilizând metodele statistice clasice, în timp ce repartiția A este presupusă, de obicei, a fi cea uniformă (plecând de la ipoteza că anomalii sunt distribuite uniform, ceea ce nu este totdeauna conform cu realitatea; evident că în cazul unei cunoașteri mai profunde a fenomenului dat, se poate considera o anumită repartiție a anomalilor corespunzătoare). Pasul următor este reprezentat de calcularea verosimilității corespunzătoare repartiției D a datelor la momentul t . Reamintim că termenul de verosimilitate – *likelihood* – a apărut la Ronald A. Fisher (1922) [50], în contextul „metodei verosimilității maxime” (*maximum likelihood*), utilizată de Gauss în dezvoltarea procedurii celor mai mici pătrate, fiind una dintre cele mai vechi și mai eficiente metode de estimare. Pe scurt, în cazul repartițiilor discrete, dacă se consideră eșantionul de date $\{x_1, x_2, \dots, x_n\}$ asupra variabilei aleatoare discrete X , de repartiție $p(x, \theta)$, depinzând de parametrul θ , verosimilitatea este data de formula:

$$L(\theta) = L(x_1, x_2, \dots, x_n; \theta) =$$

$$P\{X = x_1, X = x_2, \dots, X = x_n\} = p(x_1, \theta)p(x_2, \theta) \dots p(x_n, \theta).$$

În cazul repartițiilor continue, având densitatea de repartiție $f(x, \theta)$ ce depinde de parametrul θ , verosimilitatea este dată de formula:

$$\begin{aligned}
L(\theta) &= L(x_1, x_2, \dots, x_n; \theta) = \\
P\{x_1 < X_1 < x_1 + h, \dots, x_n < X_n < x_n + h\} &= \\
&= h^n f(x_1, \theta) f(x_2, \theta) \dots f(x_n, \theta).
\end{aligned}$$

Revenind la testul verosimilității aplicat anomalilor, verosimilitatea corespunzătoare repartiției mixte D a datelor, la momentul t este dată de formula:

$$L_t(D) = \prod_{i=1}^N p_D(x_i) = \left[(1-\alpha)^{|M_t|} \cdot \prod_{x_i \in M_t} p_{M_t}(x_i) \right] \cdot \left[\alpha^{|A_t|} \cdot \prod_{x_i \in A_t} p_{A_t}(x_i) \right].$$

Vom remarcă faptul că, de obicei, pentru simplificarea expresiilor, nu se lucrează cu funcția de verosimilitate, ci cu logaritmul său (*log-likelihood*). Revenind la testul verosimilității, se presupune că, inițial, toate datele sunt tipice (supunându-se legii M), deci nu există anomalii și se calculează funcția de verosimilitate $L_t(D)$ la momentul t . Apoi, fiecare element x_t (corespunzând repartiției M) se transferă (temporar) repartiției A și se recalculează funcția de verosimilitate $L_{t+1}(D)$, corespunzătoare momentului $(t+1)$. Se calculează acum diferența $\Delta = L_t(D) - L_{t+1}(D)$ și se aplică regula următoare: „Dacă $\Delta > c$ (c fiind o constantă-prag), atunci x_t este considerat (definitiv) o anomalie, altfel fiind considerat valoare tipică” (pentru detalii, vezi [162]).

Remarcă: Slăbiciunile testării statistice se pot rezuma, în principiu, la următoarele două aspecte:

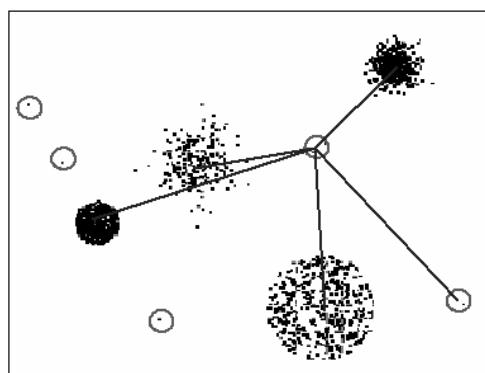
- Testele se aplică, de obicei, în cazul unui singur atribut și nu întregii secvențe;
- Repartiția datelor nu este, în general, cunoscută ci doar estimată. În cazul unor mulțimi de date de dimensiune mare, această estimare este deosebit de dificilă (blestemul dimensionalității – *the curse of dimensionality*).

C. În cazul metodelor bazate pe măsurarea distanțelor, datele sunt reprezentate ca vectori, aparținând unui anumit spațiu metric. Reamintim două metode clasice de detectare a anomalilor:

- ◆ Metoda *nearest-neighbor*. În domeniul recunoașterii formelor, metoda „ k -nearest neighbor” (k -NN) reprezintă metoda clasificării unui obiect pe baza celor mai apropiate (k) obiecte din vecinătate, accentul

punându-se deci pe gruparea obiectelor pe baza vecinătăților (cele mai apropiate) acestora –pentru amănuite tehnice asupra metodologiei, vezi capitolul 5, paragraful 5.5. În cazul detectării anomaliei, se pleacă de la pasul inițial al calculării distanțelor între fiecare două elemente din setul de date (în scopul identificării „vecinilor”), după care se trece la pasul al doilea, cel al modului de definire a ceea ce poate însemna o „anomale” în date. Astfel, o anomalie se poate defini ca, de exemplu, valoarea pentru care există un număr de „vecini” în setul de date mai mic decât un anumit prag predefinit. De la caz la caz există și alte metode de identificare a anomaliei. Este de remarcat faptul că, în cazul utilizării acestei metode, eficiența sa lasă de dorit atunci când se analizează spații ale căror dimensiuni sunt mari, situație în care noțiunea de proximitate (vecinătate) este dificil de definit (*the curse of dimensionality*). În acest caz, se poate apela la metoda reducerii dimensionalității prin proiectarea spațiului inițial pe un spațiu de dimensiune mai mică în care se poate opera mai eficient cu proiecțiile presupuselor anomalii – vezi [162].

- ◆ Metoda *clusteringului* se bazează pe divizarea setului de date în mănușchiuri (clustere) de date, pe baza similarității dintre ele, și identificarea anomaliei (valori atipice) prin evidențierea poziției lor singulare față de clusterele formate din valori tipice (vezi figura de mai jos). Schematic, se grupează datele disponibile în grupuri mai mult sau mai puțin omogene pe baza măsurării similarității dintre ele și a considerării unui „prag” pentru aceasta (pentru detalii tehnice, vezi capitolul 5, paragraful 5.7.), după care se consideră ca anomalii acele puncte aparținând unor clustere cu număr suficient de mici de membri (începând cu clusterele cu un singur element, dacă acestea există). Comparând distanța dintre aceste prezumtive anomalii și clusterele considerate tipice, se va decide dacă respectivele valori reprezintă sau nu adevărate anomalii.



D. Metodele bazate pe modele utilizează tehnici (modele) Data Mining pentru identificarea anomaliei în marea masă a datelor „normale” (tipice fenomenului considerat). Enumerăm câteva dintre aceste tehnici și modul lor de aplicare în această problemă.

- ◆ *Clasificarea.* Se utilizează un număr suficient de mare de date, atât „tipice” cât și „atipice”, pentru construirea unui model de clasificare (pentru amănunte, vezi capitolul 4 și capitolul 5, paragraful 5.2.) cu două categorii de decizie: (a) date „normale” și (b) „anomalii”, după care se aplică acesta la date noi, circumscrise acelaiași context, pentru detectarea eventualelor anomalii.
- ◆ *Mașini instruibile.* În acest caz se utilizează diferite forme de mașini instruibile (e.g. rețele neuronale artificiale, mașini cu suport vectorial etc.) pentru antrenarea în recunoașterea anomaliei și apoi detectarea acestora în seturi de date noi.
- ◆ Modele *autoregresive*, privind detectarea schimbărilor intervenite în datele corespunzătoare unor serii temporale utilizate de exemplu în probleme de prognoză (vezi și subparagraful 3.6.5.), în vederea detectării anomaliei.

La final, vom menționa câteva website-uri asociate cu detectarea anomaliei:

- *Anomaly Detection* -A Working Group in National Defense and Homeland Security – SAMSI (Statistical and Applied Mathematical Sciences Institute –parteneriat între Duke University, North Carolina State University (NCSU), University of North Carolina at Chapel Hill (UNC), National Institute of Statistical Sciences (NISS)} (<http://www.samsi.info/200506/ndhs/workinggroup/ad/>)
- CISCO Systems - Cisco Anomaly Detection and Mitigation (<http://www.cisco.com/warp/public/707/cisco-sa-20060215-guard.shtml>)
- Arbor Peakflow Network Security Management & Intrusion Detection (http://www.arbornetworks.com/products_platform.php)

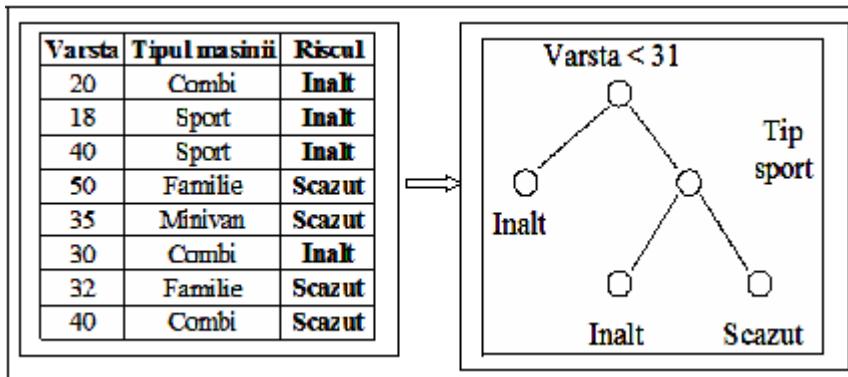
4. ARBORI DE CLASIFICARE ȘI DECIZIE

4.1. Ce este un arbore de clasificare și decizie?

Dintre metodele de clasificare pe care le-am enumerat în subparagraful 1.4.1 ne vom ocupa în acest capitol de arborii de clasificare. În principiu, *arborii de clasificare* sunt utilizati în prognоза apartenenței unor obiecte/instanțe la categorii distințe, plecând de la măsurile lor în raport cu una sau mai multe variabile predictoare. Așa după cum am mai subliniat, analiza arborilor de clasificare reprezintă una dintre principalele tehnici DM. Flexibilitatea acestei tehnici o face deosebit de atractivă, mai ales datorită faptului că prezintă și avantajul unei vizualizări sugestive (arborele ce sintetizează clasificarea obținută). Cu toate acestea, trebuie subliniat faptul că această tehnică trebuie neapărat coroborată cu alte tehnici tradiționale, mai ales în cazul în care ipotezele de lucru ale acestora (e.g. ipoteze privind repartiția datelor) sunt verificate. Totuși, ca tehnică exploratorie experimentală sau, mai ales atunci când tehniciile tradiționale nu pot fi utilizate, arborii de clasificare se pot folosi cu succes, fiind preferați altor tehnici de clasificare. Cu toate că arborii de clasificare nu sunt atât de răspândiți în domeniul recunoașterii formelor din punct de vedere probabilistico-statistic, sunt pe larg utilizati în alte ramuri ca, de exemplu, medicină (diagnostic), informatică (structura datelor), botanică (clasificare), psihologie (teoria decizilor) etc.

Vom ilustra modul de utilizare a arborilor de clasificare cu un exemplu clasic în domeniu –diagnosticul infarctului miocardic [21]. Astfel, când un pacient cu infarct miocardic este internat în spital, i se face un set de analize medicale (fiziologice) conținând, printre altele, pulsul, tensiunea arterială, electrocardiograma etc., și se consemnează istoricul său medical, vârstă și sexul. De asemenea, este supus unei monitorizări pe o perioadă de cel puțin 30 de zile pentru a se studia supraviețuirea în urma atacului de cord. Tot acest proces medical are ca scop identificarea atât a factorilor de risc cât și a profilului pacientului cu risc ridicat de infarct miocardic. Inițial (1984), s-a construit un arbore de clasificare (și decizie) relativ simplu, rezultând următoarea regulă: „**Dacă tensiunea arterială sistolică minimă a unui pacient este mai mare decât 91 (după primele 24 ore de la infarct), atunci dacă vârstă pacientului este peste 62,5 ani, atunci dacă pacientul prezintă tahicardie sinusală, atunci și numai atunci pacientul probabil că nu va supraviețui cel puțin 30 de zile**”.

Să vedem cum se formulează și cum se „crește” (construiește) un arbore de clasificare. În figura de mai jos este prezentată o mulțime de antrenament și arborele de clasificare construit plecând de la ea.



Să reamintim aici că procedura de construire a unui arbore de clasificare (și decizie) este un proces inductiv și de aceea termenul consacrat este de *inducție* a arborelui. După cum se observă din figura de mai sus, clasificarea obținută prin inducția arborelui de decizie (*decision tree induction*) se poate caracteriza prin următoarele aspecte:

- Fiecare nod (intern) al arborelui exprimă testarea după un anumit atribut;
- Fiecare ramură reprezintă rezultatul testului;
- Nodurile „frunze” reprezintă clasele.

Remarcă: 1. Să menționăm că pe parcursul acestei cărți vom utiliza desintagma „arbore de clasificare și decizie” deoarece, odată construit un arbore de clasificare, această structură este aproape întotdeauna utilizată pentru luarea unei decizii. Din această cauză, de multe ori se utilizează doar termenul de arbore de decizie.

2. Arborii de clasificare și decizie au trei denumiri clasice:

1. *Arbore de clasificare*, termen uzitat atunci când rezultatul predicției este clasa de apartenență a datelor;
2. *Arbore de regresie*, atunci când rezultatul prognosat poate fi considerat un număr real (e.g. prețul petrolului, valoarea unei case etc.);
3. *CART (C&RT)*, adică *Classification And Regression Tree* (Breiman, 1984), atunci când suntem în ambele situații de mai sus.

În ceea ce privește problema inducției unui arbore de clasificare, prezentăm mai jos unii dintre cei mai cunoscuți algoritmi (programe), utilizați de-a lungul timpului.

- * Algoritmul Hunt;
- * CART;
- * ID3;
- * C4.5 și C5.0;
- * SLIQ;
- * SPRINT;
- * QUEST;
- * FACT;
- * THAID;
- * CHAID.

4.2. Construirea unui arbore de clasificare și decizie

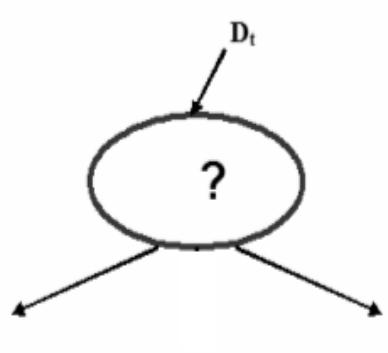
Deoarece actualmente există programe specializate de creare a arborilor de clasificare, nu vom intra în amănuntele tehnice necesare unui asemenea proces. Ceea ce este de amintit este modul general de inducție a unui arbore de clasificare și de aceea prezentăm, în continuare, algoritmul Hunt (*Hunt's Concept Learning System*, 1966 [104]) - unul dintre primii algoritmi de construire a unui arbore de clasificare, tocmai pentru a ilustra simplu și clar filosofia din spatele acestui proces.

Conceptual, algoritmul Hunt rezidă în următorii pași:

- Se notează cu D_t mulțimea de antrenament care se găsește la nodul t ;
- Dacă D_t este mulțimea vidă, atunci t este o „frunză” etichetată prestatibilită C_ϕ ;
- Dacă D_t conține instanțe ce aparțin aceleiași clase C_t , atunci t este o „frunză” etichetată C_t ;
- Dacă D_t conține instanțe ce aparțin la mai mult de o clasă, atunci se utilizează un atribut test pentru a divide nodul D_t în submulțimi (noduri) mai mici. Procedeul se aplică recursiv fiecărui nou nod.

Schema de mai jos ilustrează sintetic procedeul Hunt.

VARSTA	TIPUL MASINII	RISCUL
20	Combi	Inalt
18	Sport	Inalt
40	Sport	Inalt
50	Familie	Scazut
35	Minivan	Scazut
30	Combi	Inalt
32	Familie	Scazut
40	Combi	Scazut
25	Sport	Inalt
30	Sport	Inalt

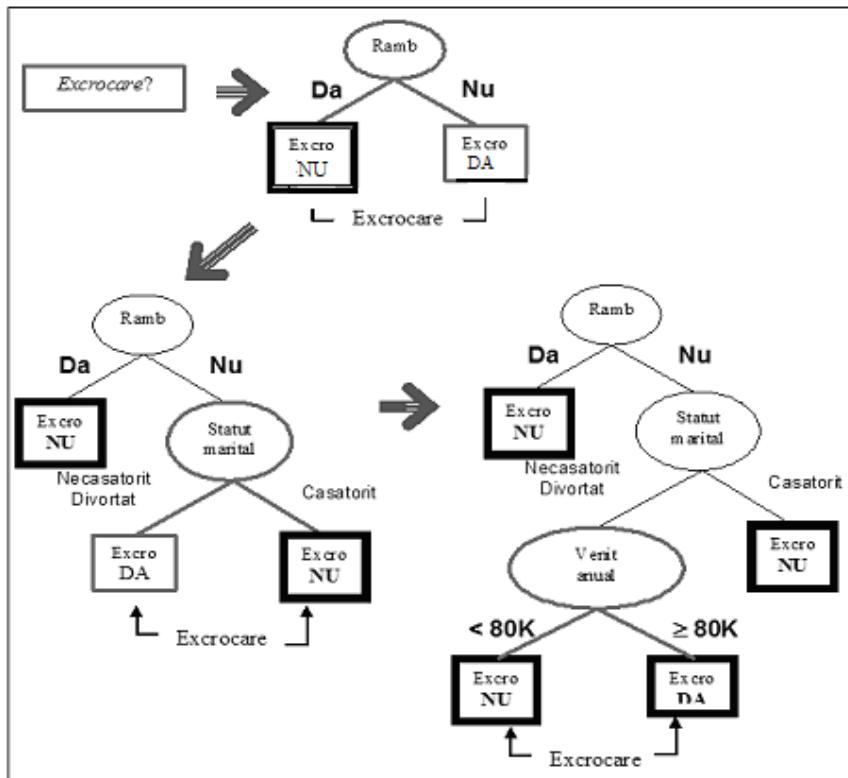


Exemplu:

Aplicarea algoritmului Hunt pentru detectarea fraudelor în cazul rambursărilor unor fonduri, de exemplu, este prezentat în tabelul de mai jos (adaptare [162]). Astfel, se pleacă de la o bază de date referitoare la înregistrări privind persoane care au beneficiat sau nu de rambursări (de exemplu, costul unei călătorii, transport, TVA etc.), în care, printre alte atribute, se specifică statutul marital și venitul anual impozabil, iar categoriile în care sunt împărțite aceste persoane sunt două, referitoare la posibilitatea de a escroca sau nu în ceea ce privește rambursările viitoare. Construind un arbore de clasificare, pe baza datelor deja înregistrate, a persoanelor ce pot sau nu frauda statul sau o anumită companie, se poate lua decizia dacă o persoană nouă, necunoscută, este sau nu probabil să comită fraudă în ceea ce privește rambursarea anumitor fonduri.

No.	Rambursare	Statut marital	Venit anual	Excrocere
1	Da	Necasatorit	125.000	NU
2	Nu	Casatorit	100.000	NU
3	Nu	Necasatorit	70.000	NU
4	Da	Casatorit	120.000	NU
5	Nu	Divorțat	95.000	DA
6	Nu	Casatorit	60.000	NU
7	Da	Divorțat	220.000	NU
8	Nu	Necasatorit	85.000	DA
9	Nu	Casatorit	75.000	NU
10	Nu	Necasatorit	90.000	DA

Schema grafică a inducției arborelui de clasificare, antrenat pe mulțimea de mai sus, este prezentată în figura următoare.



Plecând de la exemplul prezentat mai sus, să sintetizăm metodologia inducției unui arbore de clasificare. Așa după cum se observă cu ușurință, un arbore de clasificare și decizie este un discriminator de clasă (categorie) care partiziionează (divide) în mod recursiv mulțimea de antrenament până la obținerea „frunzelor”, adică a celor noduri finale care sunt constituite fie din aceeași categorie de obiecte fie dintr-o categorie dominantă. În acest sens, orice nod al arborelui care nu este „frunză” reprezintă un punct de partiziune pe baza unui atribut test care determină modul de divizare a nodului respectiv.

Strategia care stă la baza partiționării optime a unui nod este o metodă de tip *greedy algorithm*, adică o construcție recursivă „de sus în jos” de tip *divide and conquer*. Reamintim că algoritmii *greedy* sunt algoritmi care utilizează rezolvarea meta-heuristică a problemelor, prin identificarea optimelor locale și încercarea de găsire pe această bază a optimului global. Un exemplu clasic al unei asemenea abordări este cunoscuta „*travelling salesman problem*” (TSP), în care la fiecare pas se vizitează orașul cel mai apropiat, nevizitat încă. În ceea ce privește conceptul *divide and conquer* (D&C), el se bazează pe cunoscuta sintagmă latină *divide et impera*, și înseamnă împărțirea recursivă a unei probleme în două sau mai multe sub-probleme asemănătoare, până când se

ajunge la gradul de simplitate care permite rezolvarea acestora, ca apoi, plecând de la soluțiile acestor sub-probleme, să se încerce rezolvarea problemei inițiale.

Algoritmul de mai sus are mai multe variante în clasificare, dintre care amintim aici:

- ID3 și C4.5, C5.0 –Învățare automată;
- CART –Statistică;
- CHAID –Recunoașterea formelor.

În principiu, metodologia inducției arborelui de clasificare și decizie constă în două faze:

- Construirea arborelui inițial, utilizând mulțimea de antrenament disponibilă, până când fiecare „frunză” devine „pură” sau „aproape pură”;
- Fasonarea arborelui astfel „crescut” în vederea îmbunătățirii acurateței obținută pe mulțimea de testare.

Nu vom intra aici în amănunte tehnice privind aceste două aspecte fundamentale ale construirii unui arbore de clasificare și decizie, literatura de specialitate fiind suficient de bogată. Vom prezenta doar, sintetic, algoritmul care stă la baza acestui demers (în original în lb. Engleză).

Tree building algorithm

```
Make Tree (Training Data T)
{
    Partition(T)
}
Partition(Data S)
{
    if (all points in S are in the same class) then
        return
    for each attribute A do
        evaluate splits on attribute A;
    use the best split found to partition S into S1 and S2
    Partition(S1)
    Partition(S2)
}
```

Remarcă: 1. Pentru a măsura eficiența divizării nodurilor au fost propuși mai mulți indici (criterii) de partitioanare.
2. În timpul construirii arborelui, sarcina la fiecare nod este aceea de a determina punctul de partitie care divide cel mai bine obiectele din acel nod, în sensul obținerii unei purități optime a nodurilor subsecvențe.

3. Din punct de vedere tehnic, fiecare obiect (instanță/înregistrare) din mulțimea de antrenament este reprezentat printr-un vector de tip $(\mathbf{x}; \mathbf{y}) = (x_1, x_2, \dots, x_k; y_1, y_2, \dots, y_m)$, unde există m clase diferite de obiecte și k atribute pe baza cărora se clasifică acestea; să remarcăm că și variabilele y_j reprezintă tot atribute, dar ele sunt atribute „etichetă” de clasă.

În ceea ce privește indicii (criteriile) de partiționare, vom menționa aici pe cei mai cunoscuți:

- *Indexul GINI*, utilizat cu predilecție în CART și SPRINT, se bazează pe selectarea aceluia atribut de partaționare care minimizează impuritatea divizării;
- *Căștigul de informație (information gain)*, utilizat cu predilecție în ID3, C4.5 și C5.0, se bazează pe selectarea atributului de divizare care maximizează informația (prin reducerea maximă a entropiei);
- *Măsura de clasificare gresită (misclassification measure)* utilizată câteodată la măsurarea „impurității nodului”.

Vom prezenta în continuare, foarte succint, primele trei criterii de partaționare a nodurilor unui arbore de clasificare și decizie

4.2.1. Indexul GINI

Să notăm cu $f(i, j)$ frecvența de apariție a valorii j în nodul i sau, cu alte cuvinte, proporția de obiecte din clasa j care sunt atribuite nodului i (pentru m clase distincte de obiecte). Indexul GINI este dat de formula:

$$I_G(i) = 1 - \sum_{j=1}^m f^2(i, j)$$

Ca să înțelegem mai bine cum se lucrează cu indexul GINI, să exemplificăm modul de aplicare pe un caz simplu. Astfel, să considerăm că în nodul S sunt p obiecte din clasa **P** și n obiecte din clasa **N**. Aplicând formula de mai sus, rezultă că:

$$I_G(S) = 1 - f^2(S, p) - f^2(S, n) = 1 - \left(\frac{p}{p+n} \right)^2 - \left(\frac{n}{p+n} \right)^2.$$

Să presupunem că nodul S este partaționat în sub-nodurile S_1 și S_2 . Atunci indexul GINI de partaționare este definit de:

$$GINI_{split}(S) = \frac{p_1 + n_1}{p + n} \cdot I_G(S_1) + \frac{p_2 + n_2}{p + n} \cdot I_G(S_2),$$

unde p_1 și n_1 sunt obiectele din clasele **P** și **N** atribuite sub-nodului S_1 , iar p_2 și n_2 sunt obiectele din clasele **P** și **N** atribuite sub-nodului S_2 . Partiționarea optimă a nodului S este aceea care asigură cea mai mică valoare a indexului GINI de partiționare.

Exemplu:

Pentru o mai bună înțelegere a modului de calcul al indexului GINI și a utilizării sale în divizarea nodurilor arborilor de decizie, vom considera următorul exemplu [162], relativ la clasificarea riscului în cazul clienților unei societăți de asigurări –asigurare auto. Multimea de antrenament corespunzătoare este tabelată mai jos.

Varsta	Tipul masinii	Riscul
23	familie	inalt
17	sport	inalt
43	sport	inalt
68	famile	scazut
32	combi	scazut
20	familie	inalt

Din tabel se observă că, inițial, vom căuta punctele de partiționare printre valorile: 17, 20, 23, 32, 43 și 68. Care este valoarea optimă de partiționare însă? Pentru a răspunde la această întrebare, calculăm indexul GINI și cel de partiționare în raport cu fiecare punct în parte. Calculele sunt expuse mai jos. Pentru a face mai simplu de înțeles calculul, rezumăm datele corespunzătoare vârstei de 17 ani în următorul tabel.

Varsta	Risc inalt	Risc scazut
≤ 17	1	0
> 17	3	2

Aplicând formulele corespunzătoare, obținem:

$$I_{GINI}(Vârsta \leq 17) = 1 - (1^2 + 0^2) = 0$$

$$I_{GINI}(Vârsta \leq 17) = 1 - ((3/5)^2 - (2/5)^2) = 1 - (13/25)^2 = 12/25$$

$$GINI_{split}(Vârsta = 17) = (1/6) * 0 + (5/6) * (12/25) = 0.4$$

Analog se obțin următoarele rezultate:

$$GINI_{split}(Vârsta = 20) = 0.33$$

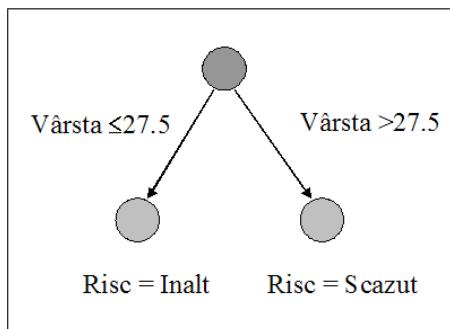
$$GINI_{split}(Vârsta = 23) = 0.22$$

$$GINI_{split} (Vârstă = 32) = 0.29$$

$$GINI_{split} (Vârstă = 43) = 0.26$$

$$GINI_{split} (Vârstă = 68) = 0.44$$

După cum se observă, cea mai mică valoare este obținută pentru vârstă de 23 ani. În acest caz, se ia media între această vârstă și vârstă cea mai apropiată, având valoarea cea mai mică (adică 0.29, obținută pentru vârstă de 32 ani). Rezultă deci că divizarea se va face la vârstă $(23 + 32)/2 = 27,5$ ani. Ilustrăm această divizare după atributul „vârstă” în figura de mai jos.

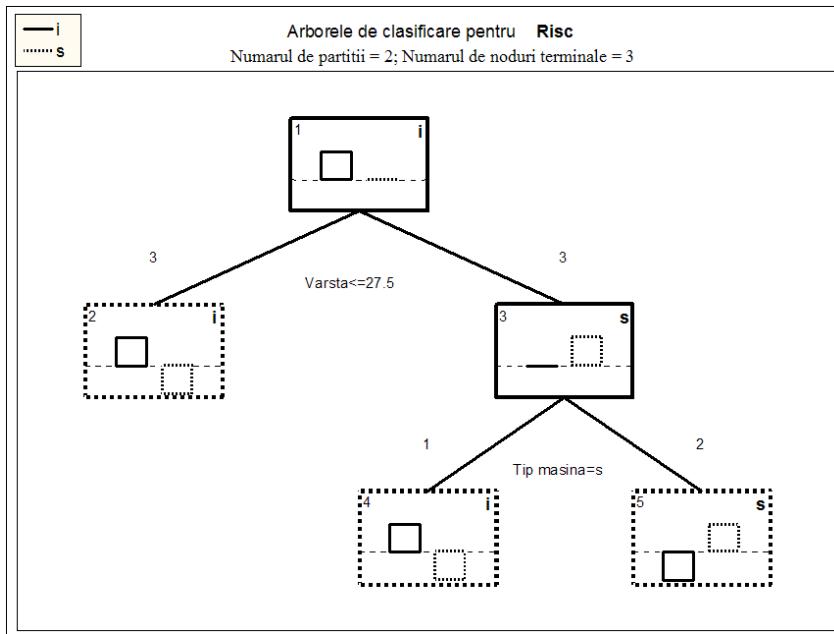


Se procedează apoi analog pentru divizarea recursivă a nodurilor arborelui, utilizând și celelalte attribute ale instanțelor din mulțimea de antrenament, în cazul de față singurul rămas, și anume „tipul mașinii”.

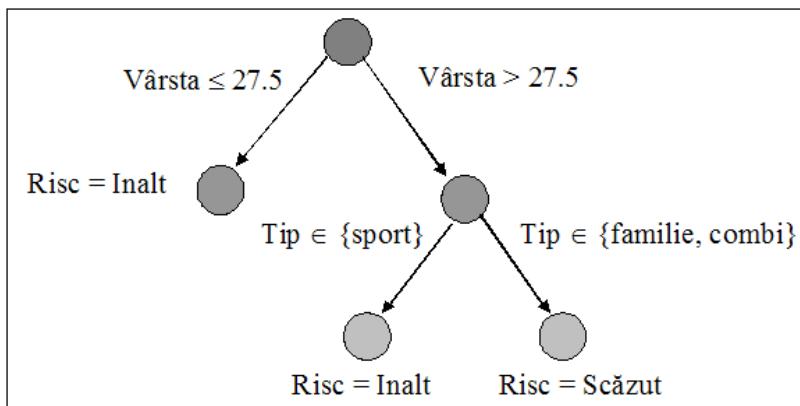
Mai jos prezentăm doi arbori de clasificare și decizie în care atributul „tipul mașinii” este utilizat diferit. În primul caz s-a folosit pentru partii doar tipul „sport”. Tabelul care sintetizează calculele în acest caz este prezentat în continuare.

Nodul	Stânga	Dreapta	n în clasa i	n în clasa s	Prognoza (clasa)	Prag de partii	Variabila de partii	Categorie de partii
1	2	3	4	2	i	27.5	Vârstă	
2			3	0	i			
3	4	5	1	2	s		Tip mașina	sport
4			1	0	i			
5			0	2	s			

Arborele corespunzător acestui caz este prezentat în figura de mai jos.



În al doilea caz, am considerat pentru 'tipul mașinii' ramurile {sport} și {familie, combi}. Arborele corespunzător este prezentat mai jos.



4.2.2. Câștigul de informație

Câștigul de informație (*information gain*) este utilizat pentru selectarea atributului de testare (divizarea nodului), pe baza maximizării informației sau, echivalent, reducerii maxime a entropiei la nodul respectiv. Astfel, atributul ales pe baza acestei metode va maximiza informația necesară clasificării obiectelor prin partitia rezultată. Să amintim aici că în Teoria informației și Învățarea automată „information gain” este deseori sinonim cu „divergența Kullback-Leibler”.

Formula de calcul a câștigului de informație este dată de:

$$I_E(i) = - \sum_{j=1}^m f(i, j) \cdot \log_2 [f(i, j)],$$

unde, la fel ca și în cazul indexului GINI, $f(i, j)$ reprezintă proporția de obiecte din clasa j care sunt atribuite nodului i . Ca să se înțeleagă mai clar cum funcționează această metodă, vom exemplifica calculele pe un caz simplu. Să presupunem că nodul S , conținând s obiecte din m clase, este partionat în subnodurile S_1 și S_2 și vrem să utilizăm atributul A pentru partionarea lui S . Să presupunem că valorile pe care le poate lua atributul A sunt $\{a_1, a_2\}$, astfel încât S_1 va conține acele obiecte din S care au valoarea a_1 a atributului A , iar S_2 va conține acele obiecte din S care au valoarea a_2 . Să notăm cu s_{i1} și s_{i2} numărul obiectelor din clasa i aparținând sub-nodurilor S_1 și S_2 . În acest context, putem defini *entropia* sau *informația așteptată* (scontată), obținută prin partionarea bazată pe atributul A , cu ajutorul formulei:

$$E(A) = \frac{\sum_{j=1}^2 (s_{1j} + s_{2j} + \dots s_{mj})}{s \cdot I_E(S)}.$$

În aceste condiții, informația câștigată prin partionarea nodului S pe baza atributului A este dată de formula:

$$Gain(A) = I_E(S) - E(A).$$

În consecință, se va alege pentru partionarea nodului curent acel atribut care furnizează cea mai mică entropie, implicând puritatea maximă a partitiilor, deci cel mai mare câștig de informație.

Exemplu:

Să considerăm mulțimea de antrenament tabulată mai jos, referitoare la profilul unui posibil cumpărător de computer.

Vârstă	Venit	Student	Cumpărător computer
≤ 30	Ridicat	Nu	Nu
≤ 30	Ridicat	Nu	Nu
31...40	Ridicat	Nu	Da
> 40	Mediu	Nu	Da
> 40	Scăzut	Da	Da
> 40	Scăzut	Da	Nu
31...40	Scăzut	Da	Da
≤ 30	Mediu	Nu	Nu
≤ 30	Scăzut	Da	Da
> 40	Mediu	Da	Da
≤ 30	Mediu	Da	Da
31...40	Mediu	Nu	Da
31...40	Ridicat	Da	Da
> 40	Mediu	Nu	Nu

Atributul de clasificare care conferă calitatea de cumpărător de computer unui client este reprezentat de ultima coloană, deci există două clase **{Da, Nu}**. Avem $s_1 = 9$ cazuri în categoria **Da** și $s_2 = 5$ cazuri în categoria **Nu**.

Câștigul de informație corespunzător este dat de:

$$I_E(S) = - (9/14) * \log_2(9/14) - (5/14) * \log_2(5/14) = 0.94.$$

Considerând atributul test **Vârstă**, obținem entropia respectivă:

$$E(\text{Vârstă}) = (5/14) * 0,971 + (4/14) * 0 + (5/14) * 0,971 = \mathbf{0.694}.$$

În aceste condiții, informația câștigată prin partiționarea pe baza vârstei este:

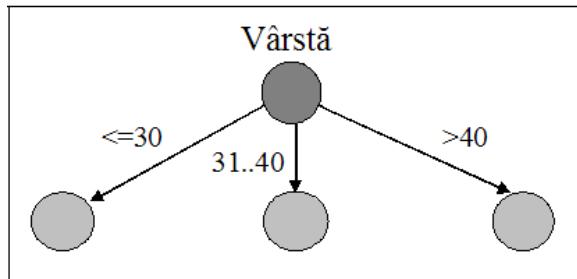
$$Gain(\text{Vârstă}) = 0,94 - 0,694 = 0,246.$$

Analog se calculează și valorile pentru informația câștigată dacă s-ar partiționa după celelalte atrbute:

$$Gain(\text{Venit}) = 0,029$$

$$Gain(\text{Student}) = 0.151$$

Deoarece informația ce se poate câștiga divizând nodul pe baza atributului *Vârstă* are valoarea maximă, acest atribut este ales pentru partitioare, un nod etichetat „*Vârstă*” fiind astfel creat, ramurile respective corespunzând valorilor atributului. Ilustrăm mai jos partitia respectivă.



4.2.3. Măsura de clasificare greșită

Un alt indice care se utilizează câteodată în partitioarea nodurilor este cel bazat pe măsura de clasificare greșită (*misclassification measure*). Acest indice „măsoară” eroarea în clasificare care se poate face la un nod utilizând o anumită partitioare și este dat de formula:

$$I_M(i) = 1 - \max_j f(i, j),$$

unde reamintim că $f(i, j)$ reprezintă proporția de obiecte din clasa j care sunt atribuite nodului i . Se observă că maximul erorii se obține atunci când obiectele de categorii diferite sunt distribuite în mod egal în nod, furnizând astfel cea mai săracă informație, în timp ce minimul erorii este obținută în cazul în care toate obiectele din nod aparțin aceleiași categorii, furnizând astfel cea mai bogată informație. În concluzie, vom alege partitioarea care minimizează eroarea.

Pentru exemplificarea modului de aplicare a acestei metode, să reconsiderăm cazul riscului relativ la tipul de mașină al clientului unei societăți de asigurări.

În cazul vîrstei de 17 ani, datele tabelate sunt următoarele:

Varsta	Risc inalt	Risc scazut
≤ 17	1	0
> 17	3	2

Aplicând formulele corespunzătoare, obținem:

$$I_M(Vârstă \leq 17) = 1 - \max\{1, 0\} = 1 - 1 = 0,$$

$$I_M(Vârstă > 17) = 1 - \max\{3/5, 2/5\} = 1 - 3/5 = 2/5,$$

$$I_{split}(Vârstă = 17) = (1/6)*0 + (5/6)*(2/5) = 1/3.$$

În cazul vîrstei de 20 ani, datele tabelate sunt următoarele:

Varsta	Risc înalt	Risc scăzut
≤ 20	2	0
> 20	2	2

Aplicând formulele corespunzătoare, obținem:

$$I_M(Vârstă \leq 20) = 1 - \max\{1, 0\} = 1 - 1 = 0,$$

$$I_M(Vârstă > 20) = 1 - \max\{1/2, 1/2\} = 1 - 1/2 = 1/2,$$

$$I_{split}(Vârstă = 20) = (2/6)*0 + (4/6)*(1/2) = 1/3.$$

În cazul vîrstei de 23 ani, datele tabelate sunt următoarele:

Varsta	Risc înalt	Risc scăzut
≤ 23	3	0
> 23	1	2

Aplicând formulele corespunzătoare, obținem:

$$I_M(Vârstă \leq 23) = 1 - \max\{1, 0\} = 1 - 1 = 0,$$

$$I_M(Vârstă > 23) = 1 - \max\{1/3, 2/3\} = 1 - 2/3 = 1/3,$$

$$I_{split}(Vârstă = 23) = (3/6)*0 + (3/6)*(1/3) = 1/6.$$

În cazul vîrstei de 32 ani, datele tabelate sunt următoarele:

Varsta	Risc înalt	Risc scăzut
≤ 32	3	1
> 32	1	1

Aplicând formulele corespunzătoare, obținem:

$$I_M(Vârstă \leq 32) = 1 - \max\{3/4, 1/4\} = 1 - 3/4 = 1/4,$$

$$I_M(Vârstă > 32) = 1 - \max\{1/2, 1/2\} = 1 - 1/2 = 1/2,$$

$$I_{split}(Vârstă = 32) = (4/6)*(1/4) + (2/6)*1/2 = 1/3.$$

În cazul vîrstei de 43 ani, datele tabelate sunt următoarele:

Varsta	Risc înalt	Risc scazut
≤ 43	4	1
> 43	0	1

Aplicând formulele corespunzătoare, obținem:

$$I_M(Vârstă \leq 43) = 1 - \max\{4/5, 1/5\} = 1 - 4/5 = 1/5,$$

$$I_M(Vârstă > 43) = 1 - \max\{0, 1\} = 1 - 1 = 0,$$

$$I_{split}(Vârstă = 43) = (5/6)*(1/5) + (1/6)*0 = \mathbf{1/6}.$$

În cazul vîrstei de 68 ani, datele tabelate sunt următoarele:

Varsta	Risc înalt	Risc scazut
≤ 68	4	2
> 68	0	0

Aplicând formulele corespunzătoare, obținem:

$$I_M(Vârstă \leq 68) = 1 - \max\{4/6, 2/6\} = 1 - 4/6 = 2/6,$$

$$I_M(Vârstă > 68) = 1 - \max\{0, 0\} = 1 - 0 = 1,$$

$$I_{split}(Vârstă = 68) = (6/6)*(2/6) + 0*1 = 1/3.$$

Se observă că există două valori ale vîrstei pentru care se obțin valori minime ale erorii, 23 ani, respectiv 43 ani.

Remarcă: Câteodată, ca metodă de selectare a partiționării, mai este utilizat și *tabelul de contingență χ^2* , în special în algoritmul CHAID, criteriu care selectează atributul de divizare care maximizează corelația între atribut și clase.

4.3. Metode computaționale

4.3.1. Acuratețea predictivă

Scopul construirii arborilor de clasificare este, evident, acela de a obține o predicție cât mai precisă. Cu toate că este deosebit de dificil, dacă nu chiar imposibil să se definească de o manieră indiscutabilă ce este aceea „acuratețea predicției”, totuși se pot considera, din punct de vedere practic, niște indicatori ai procesului, cunoscuți ca și „costuri” ale predicției. Astfel, o

predicție optimă va implica, în consecință, costuri minime de clasificare eronată. În esență, ideea costurilor predicției generalizează faptul că o prognoză bună conduce la o rată redusă de clasificare greșită. Din practică a rezultat faptul că, în cele mai multe cazuri, nu este importantă numai rata de clasificare ci și consecințele unei clasificări eronate. Să ne gândim numai la cazul în care medicul greșește diagnosticul în cazul unei maladii incurabile. Ce este mai grav: să spună că pacientul suferă de acea boală când de fapt nu este așa, sau că nu suferă și de fapt pacientul are boala respectivă? Aici nu este vorba doar de un caz clasificat eronat, ci de consecințele grave ale unui astfel de fapt.

Prezentăm, în continuare, principalele costuri legate de procesul de clasificare.

- **Probabilități prealabile** (*prior probabilities* sau *a priori probabilities*) sunt acei parametri care specifică probabilitatea ca un obiect să aparțină unei anumite clase. De obicei, atunci când nu dispunem de cunoștințe prealabile despre fenomenul studiat, care să ne permită să facem o alegere clară, se aleg acei parametri proporționali cu numărul de obiecte din fiecare clasă.
- **Costuri de clasificare greșită** (*misclassification costs*) se referă la faptul că, în procesul de clasificare, de obicei este nevoie de o clasificare mai precisă a unor categorii decât a altora. Reluând exemplul cu diagnosticul unei boli incurabile, este mult mai importantă acuratețea clasificării unui pacient ca având cancer hepatic decât ca având hepatită cronică. Costurile de clasificare greșită sunt alese astfel încât să reflecteze importanța fiecărei clase. Atunci când nu există preferințe pentru anumite clase, se aleg costuri egale.

4.3.2. Condiția de STOP pentru partaționare

După stabilirea criteriului de partaționare (divizare) a nodurilor, se pune problema alegerii criteriului de oprire a acestui proces. O caracteristică a „creșterii” arborilor de clasificare este aceea că procesul de partaționare se derulează până când toate nodurile terminale („frunzele”) sunt „pure” din punct de vedere al elementelor constitutive, atâtă timp cât nu există o condiție de stopare a creșterii arborelui. În acest caz, fiecare nod va conține elemente din aceeași categorie, ceea ce de cele mai multe ori este neproductiv, mai ales ținând cont de faptul că procesul creșterii arborelui este faza de antrenament a modelului de clasificare și nu cea de aplicare în practică. Cu alte cuvinte, nu este interesant ca pe multimea de antrenament clasificatorul să aibă acuratețea 100% („frunze” pure ca și componență), ci pe cea de testare/validare să aibă performanță maximă.

Uzual, sunt utilizate două reguli de *Stop*:

- **Minimul *n***, se referă la condiția de *Stop* care specifică un număr minim de obiecte care să fie conținute în nodurile terminale. În aceste condiții, divizarea unui nod ia sfârșit atunci când fie nodul este pur, fie nu conține mai mult decât numărul specificat de obiecte.
- **Proportia de obiecte**, se referă la condiția de *Stop* care impune ca divizarea unui nod să ia sfârșit atunci când fie nodul este pur, fie nu conține mai multe obiecte decât o proporție (procentaj) minimă din mărimea uneia sau mai multor clase.

4.3.3. Fasonarea arborilor de clasificare

Odată construit arboarele de clasificare și decizie, pe baza setului de obiecte de antrenament, este natural ca acesta să reflecteze, mai mult sau mai puțin, caracteristicile acestei mulțimi. Multe din ramurile sale vor fi influențate puternic de anomaliiile care se pot afla în mulțimea de antrenament, anomalii datorate „zgomotului” sau anumitor valori extreme scăpate de procesul de filtrare a datelor inițiale, dacă acesta a fost în prealabil efectuat. Menționăm că, în principiu, se poate construi un arbore plecând direct de la datele brute, fără o prealabilă procesare a acestora, astfel încât arboarele construit pe baza lor va reflecta fidel particularitățile mulțimii de antrenament. Deoarece un arbore se construiește pentru a putea fi aplicat la diverse alte seturi de date, este necesară evitarea acestei „potriviri” prea accentuate (*overfitting*) cu mulțimea pe care s-a făcut antrenamentul. Pe de altă parte, atunci când arboarele (clasificatorul) este prea simplu față de datele utilizate la antrenament și, în consecință, atât eroarea de antrenament cât și cea de testare sunt inadmisibil de mari, avem de-a face cu situația inversă de „sub-potrivire” (*underfitting*) a arborelui cu datele. Totuși, cel mai adesea ne întâlnim cu primul caz, *overfitting*-ul. În acest caz se utilizează binecunoscuta metodă de fasonare (*pruning*) a arborelui. În principiu, se utilizează metode statistice pentru îndepărțarea ramurilor nesemnificative, redundante, sau care nu urmează pattern-ul general al datelor, obținând astfel un arbore mai puțin „stufos”, cu o mai mare scalabilitate și viteză de clasificare.

Există două tipuri de fasonare a unui arbore de clasificare și decizie:

- *Fasonarea prealabilă (pre-pruning)*, care înseamnă că se oprește practic „creșterea” arborelui în timpul procesului de inducție, prin decizia de a se opri divizarea nodului, astfel încât acesta va deveni o „frunză”, etichetată cu numele clasei cu cele mai multe elemente. Principalele condiții de stopare sunt fie atunci când nodul e „pur”, fie când toate valorile atributelor sunt egale. Menționăm că există

și condiții mai „speciale” de stopare a creșterii, unele fiind amintite în subparagraf anterioar.

- *Fasonarea ulterioară (post-pruning)*, care are loc după terminarea „creșterii” arborelui, fiind un proces „de jos în sus”, bazat pe măsurarea erorii de clasificare a arborelui. Astfel, un nod va fi „fasonat” prin renunțarea la ramurile sale, el devenind o „frunză” etichetată în aceeași manieră ca mai sus, dacă eroarea de clasificare se diminuează prin această operație.

Remarcă: De obicei, se procedează la fasonarea ulterioară, obținută după inducția completă a arborelui. Fasonarea prealabilă ține de „filosofia” partiționării arborelui și poate fi controlată în timpul acestui proces, încă de la începutul inducției.

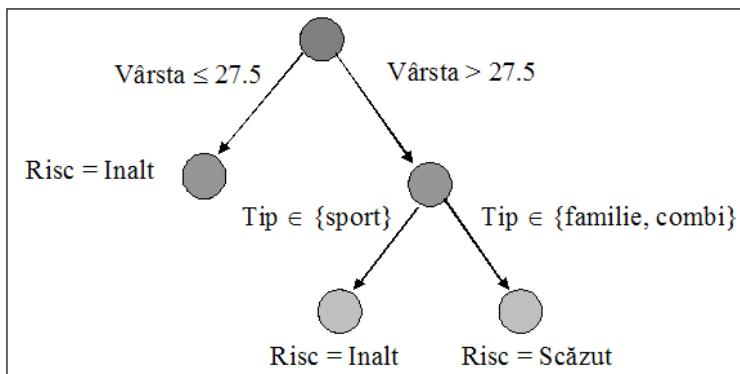
Așa după cum se observă cu ușurință, în cazul fasonării ulterioare este nevoie de cuantificarea erorii de clasificare la fiecare pas al „tăierii” unor ramuri, precum și de criterii de utilizare a acesteia, pentru a se stabili dacă astfel se obține sau nu mărirea performanței clasificatorului. În acest sens, menționăm următoarele aspecte:

- *Erorile de generalizare (generalization errors)*, se bazează pe măsurarea erorilor pe mulțimea de testare și compararea cu cele obținute la antrenament. În unele cazuri, se utilizează și eroarea de validare.
- *Occam's Razor*, având originea în lucrările logicianului englez, călugărul franciscan William of Ockham (sec. XIV), și multe aplicații în biologie, medicină, religie, filosofie, statistică etc., se bazează, în cazul arborilor de clasificare și decizie, pe următoarele principii:
 - Din doi arbori producând erori de generalizare similare, se alege cel mai simplu;
 - Un arbore mai complex are șansa mai mare să fie dependent de erorile din date;
 - Trebuie inclusă, deci, complexitatea clasificatorului în procesul de evaluare a arborilor.
- *Principiul lungimii minime de descriere (minimum description length –MDL, J. Rissanen, 1978)* este o formalizare a procedeului Occam's Razor, postulând că cea mai bună ipoteză rezultată pe un set de date este aceea care duce la cea mai mare compresie a datelor. În cazul procesului de estimare a clasificatorului optim, se utilizează formula: $\text{Cost}(\text{Model}, \text{Data}) = \text{Cost}(\text{Data}|\text{Model}) + \text{Cost}(\text{Model})$, unde costurile respective codează parametrul de clasificare greșită, numărul de ramuri și condiția de partiționare (mai multe amănunte în [162]).

4.3.4. Extragerea regulilor de clasificare din arborii de decizie

Odată construit arboarele de clasificare, modelul respectiv este utilizat în luarea unor decizii optime. Cunoștințele pe care le implică această construcție „arborică” poate fi „citită” lesne coborând pe „crengi” în josul său, extrăgând astfel reguli de clasificare de forma IF-THEN. O regulă este creată coborând din vârf (dacă este un arbore cu rădăcina -nodul de bază- sus) până la fiecare frunză. Orice pereche de valori ale unui atribut de-a lungul acestui traseu va forma o conjuncție în ipoteza regulii, iar frunza conținând clasa predictivă (cu care de altfel este și etichetată) va forma consecința regulii.

Vom exemplifica procesul de extragere a regulilor de clasificare, utilizând arboarele de decizie corespunzător riscului în asigurările auto, arbore foarte simplu, prezentat din nou mai jos.



- * IF $Vârstă = “\leq 27.5$ ani” THEN $Risc = “Inalt”$
- * IF $Vârstă = “> 27.5$ ani” AND $Tip = “Sport”$ THEN $Risc = “Inalt”$
- * IF $Vârstă = “> 27.5$ ani” AND $Tip = “familie/sport”$ THEN $Risc = Scăzut”$

4.4. Avantajele arborilor de clasificare și decizie

În capitolul următor vom prezenta și alte modele binecunoscute de clasificatori. Totuși, făcând o comparație între modelele de clasificare utilizate în Data Mining, putem remarca câteva avantaje ale arborilor de clasificare și decizie, pe care le enumerăm mai jos.

- ⇒ Sunt ușor de înțeles și interpretat, forma lor grafică reprezentând un atu puternic în acest sens;

- ⇒ Necesară un volum mic de pregătire a datelor în raport cu alte tehnici;
- ⇒ Permit utilizarea atât a datelor nominale cât și a celor categoriale, fără nicio restricție;
- ⇒ Reprezintă modele de tip „cutie albă” (*white-box*), în care logica deciziei poate fi urmărită ușor, regulile de clasificare fiind „la vedere”. Spre deosebire de arborii de decizie, alte tehnici utilizate și în clasificare, ca de exemplu rețelele neuronale artificiale, acționează ca niște „cutii negre” (*black-box*), nefurnizând direct utilizatorului regulile de clasificare;
- ⇒ Fac posibilă utilizarea unor tehnici statistice clasice pentru validarea modelului;
- ⇒ Sunt robuste, rapide și „lucrează” bine cu seturi mari de date.

5. TEHNICI ȘI MODELE DE DATA MINING

5.1. Metode în Data Mining

În capitolul precedent am prezentat metoda arborilor de clasificare și decizie, considerată ca o tehnică aparte de clasificare, fapt pentru care i-am dedicat în întregime acel capitolul. Așa cum am arătat în primul capitol, în domeniul Data Mining se utilizează și alte tehnici avansate, atât în clasificare, cât și în alte domenii de explorare automată a datelor, metode binecunoscute ca, de exemplu:

- Clasificatori bayesieni (*Bayesian classifiers/Naïve Bayes*);
- Rețele neuronale (*Neural networks*). Mașini cu suport vectorial (*SVM -Support vector machines*);
- Reguli de asociere (*Association rule*);
- *k*-nearest neighbor (*k-nearest neighbor classifier*);
- Multimi rough (*Rough sets*);
- Clustering;
- Algoritmi genetici (*Genetic algorithms*);

O bună parte a acestui capitol va fi dedicată prezentării conceptuale a acestor tehnici, pentru a familiariza cititorul cu principalele lor caracteristici, utilizând pentru ilustrarea metodologiei exemple simple, ușor de înțeles. Menționăm că și pentru aceste tehnici există o întreagă bibliotecă de software specializat, scopul prezentării fiind acela de familiarizare a cititorului cu principiile ce stau la baza programelor respective și nu cu proiectarea efectivă a algoritmilor.

5.2. Clasificatori bayesieni

Fie (Ω, Σ, P) un spațiu (câmp) de probabilitate și $\{A_1, A_2, \dots, A_n\}$ o partiție a spațiului Ω . Reamintim un rezultat deosebit de important din *Teoria Probabilităților*, pe baza căruia putem calcula probabilitatea oricărui eveniment aparținând σ - algebrei de părți Σ .

Teoremă (*formula probabilității totale*). Fie B un eveniment arbitrar și $\{A_1, A_2, \dots, A_n\}$ o partiție a spațiului Ω . Atunci:

$$P\{B\} = \sum_{i=1}^n P\{B | A_i\}P\{A_i\}, \quad P\{A_i\} > 0.$$

În anul 1763, reverendul Thomas Bayes a descoperit următorul rezultat celebru, deosebit de important prin aplicațiile sale, care, practic, „inversează” formula probabilității totale.

Teoremă (formula lui Bayes). Fie B un eveniment arbitrar din Σ și $\{A_1, A_2, \dots, A_n\}$ o partitură a spațiului Ω . Atunci:

$$P\{A_i | B\} = \frac{P\{B | A_i\}P\{A_i\}}{\sum_{i=1}^n P\{B | A_i\}P\{A_i\}}, \quad P\{B\} > 0, \quad P\{A_i\} > 0, \quad i = 1, 2, \dots, n.$$

Uzual, $P\{A_i | B\}$ este cunoscută ca probabilitate *posterioră* (*posterior*), $P\{A_i\}$ ca probabilitate *apriorică* (*prior probability*), $P\{B | A_i\}$ ca *verosimilitate* (*likelihood*), iar $P\{B\}$ ca *evidență/dovadă* (*evidence*). În acest context, formula lui Bayes capătă forma:

$$\text{posteriora} = \frac{\text{verosimilitatea} \times \text{probabilitatea apriorica}}{\text{evidența}}$$

Remarcă: Se pot extinde cele două rezultate anterioare și dacă, în loc de partitură a spațiului Ω , se consideră $\{A_1, A_2, \dots, A_n\}$ ca o familie de evenimente, astfel încât:

$$A_i \cap A_j = \emptyset, \quad \forall i \neq j \quad \text{și} \quad A_1 \cup A_2 \cup \dots \cup A_n \supseteq B$$

Exemple:

1) Să considerăm că într-o companie se produc același produse în trei unități B_1 , B_2 și B_3 , având capacitatele de producție de 60%, 30% și 10% (aceste procentaje reprezintă, practic, probabilitățile ca un produs oarecare să provină de la una din cele trei unități). Fiecare unitate are rata de a produce obiecte cu defecțiuni de 6%, 3% și 5%. Care este probabilitatea ca un produs defect, ales la întâmplare, să provină de la una din unitățile B_1 , B_2 sau B_3 ?

Să notăm cu A evenimentul ca un produs, ales la întâmplare, să fie defect. Conform formulei probabilității totale, rezultă că:

$$P\{A\} = \sum_{i=1}^3 P\{A | B_i\}P\{B_i\} = 0,06 \times 0,6 + 0,03 \times 0,3 + 0,05 \times 0,1 = 0,05 = 5\%.$$

Aplicând acum formula lui Bayes, obținem că:

$$P\{B_1 | A\} = \frac{P\{A | B_1\}P\{B_1\}}{P\{A\}} = \frac{0,06 \times 0,6}{0,05} = \frac{36}{50} = 72\%,$$

$$P\{B_2 | A\} = \frac{9}{50} = 18\%,$$

$$P\{B_3 | A\} = \frac{5}{50} = 10\%.$$

2) Personalul dintr-o companie este format din 45% bărbați și 55% femei. Dacă se consideră că, în general, 4% dintre bărbați și 6% dintre femei utilizează în mod frecvent telefoanele companiei în scopuri personale, atunci staff-ul poate evalua, pe baza metodei de mai sus, atât proporția persoanelor din companie care utilizează în mod frecvent telefoanele companiei în scopuri personale (5,1%), cât și proporția bărbaților din companie care fac acest lucru (35,3%).

Remarcă: De cele mai multe ori formula lui Bayes este prezentată într-o formă simplificată, în conjuncție cu formula probabilității condiționate. Astfel, plecând de la formula probabilității condiționate, care leagă două evenimente A și B , ($P\{B\} \neq 0$), prin relația:

$$P\{A | B\} = \frac{P\{A \cap B\}}{P\{B\}},$$

unde $P\{A|B\}$ reprezintă probabilitatea lui A condiționată de B , se definește formula lui Bayes prin:

$$P\{B | A\} = \frac{P\{A | B\} \cdot P\{B\}}{P\{A\}}.$$

Exemplu:

Să presupunem că, din statisticile cunoscute, se știe că meningita cauzează imobilitatea gâtului în 50% din cazuri, că proporția persoanelor care au făcut meningită este 1/50.000 și că proporția persoanelor cu imobilitatea gâtului este 1/20. Atunci procentul persoanelor care au avut meningită și acuză imobilitatea gâtului va fi de 0,02%. Calculul este simplu, bazându-se pe formula lui Bayes redusă. Astfel, dacă notăm:

- $P\{M|I\}$ = probabilitatea ca o persoană să fi avut meningită, condiționată de existența imobilității gâtului;
- $P\{I|M\}$ = probabilitatea ca o persoană să se plângă de imobilitatea gâtului, condiționată de existența meningitei;
- $P\{I\}$ = proporția persoanelor care se plâng de imobilitatea gâtului;
- $P\{M\}$ = proporția persoanelor care au avut meningită.

Atunci:

$$P\{M | I\} = \frac{P\{I | M\} \cdot P\{M\}}{P\{I\}} = \frac{1/2 \cdot 1/50000}{1/20} = 0.0002$$

Teoria bayesiană a deciziilor reprezintă o metodă statistică fundamentală în domeniul clasificării formelor (*pattern classification*). În teoria deciziilor, scopul tradițional este acela de a minimiza probabilitatea de a greși, sau *riscul așteptat*.

Regula de decizie bayesiană poate fi rezumată în următorul algoritm:

- Fie D_k regula de decizie referitoare la starea naturală A_k .
- Fiind dată o măsurătoare x , eroarea relativă la starea A_k este definită de $P\{\text{eroare}/x\} = 1 - P\{A_k|x\}$.
- Se *minimizează* probabilitatea de a greși.
- Regula bayesiană de decizie este dată de aserțiunea: „*Alege D_k dacă $P\{A_k|x\} > P\{A_j|x\}$, $\forall j \neq k$* ” sau, echivalent „*Alege D_k dacă $P\{x/A_k\} P\{A_k\} > P\{x/A_j\} P\{A_j\}$, $\forall j \neq k$* ”

Să considerăm un set de date care urmează a fi clasificate utilizând un clasificator bayesian și să presupunem pentru aceasta că fiecare atribut (inclusiv atributul corespunzător etichetei de clasă) este o variabilă aleatoare. Fiind dat un obiect cu attributele $\{A_1, A_2, \dots, A_n\}$, ne propunem clasificarea sa în clasa C . Clasificarea este corectă atunci când probabilitatea condiționată:

$$P\{C | A_1, A_2, \dots, A_n\},$$

este maximă. Problema concretă care se pune în procesul de clasificare este de a estima direct din date această probabilitate în vederea maximizării sale. Pentru aceasta, aplicăm formula lui Bayes astfel:

- Se calculează probabilitățile posterioare $P\{C_j | A_1, A_2, \dots, A_n\}$ pentru toate clasele C_j , utilizând formula:

$$P\{C_j | A_1 A_2 \dots A_n\} = \frac{P\{A_1 A_2 \dots A_n | C_j\} \cdot P\{C_j\}}{P\{A_1 A_2 \dots A_n\}}.$$

- Se alege apoi clasa C_k care maximizează $P\{C_j | A_1, A_2, \dots, A_n\}$. (echivalent, clasa C_k care maximizează $P\{A_1 A_2 \dots A_n | C_j\} \cdot P\{C_j\}$).

Din cele expuse mai sus rezultă deci că trebuie calculată probabilitatea $P\{A_1 A_2 \dots A_n | C_j\}$. O abordare în acest sens face apel la așa-numita *clasificare naivă Bayes* (*naive Bayes*, cunoscută și ca *Bayes idioată* - *Idiot's Bayes*), care presupune, de foarte multe ori fără niciun temei, independența evenimentelor,

de unde și numele de „naivă”. În cazul de față vom presupune independența reciprocă a atributelor (evident, ipoteză neadevărată de cele mai multe ori) pentru o anumită clasă C , adică:

$$P\{A_1 A_2 \dots A_n \mid C\} = P\{A_1 \mid C\} \cdot P\{A_2 \mid C\} \cdot P\{A_n \mid C\}.$$

Vom estima apoi probabilitățile $P\{A_i \mid C_j\}$ pentru toate attributele A_i și clasele C_j , astfel încât un obiect nou, necunoscut, va fi clasificat în clasa C_k dacă probabilitatea corespunzătoare acestei clase:

$$P\{C_k\} \cdot \prod P\{A_i \mid C_k\},$$

este maximă față de celelalte.

Exemplu:

Să reluăm exemplul prezentat în capitolul precedent, relativ la investigarea posibilității de fraudă la rambursarea unor sume, pentru care mulțimea de antrenament este tabelată mai jos.

No.	Rambursare	Statut marital	Venit anual	Excrocare
1	Da	Necasatorit	125.000	NU
2	Nu	Casatorit	100.000	NU
3	Nu	Necasatorit	70.000	NU
4	Da	Casatorit	120.000	NU
5	Nu	Divorțat	95.000	DA
6	Nu	Casatorit	60.000	NU
7	Da	Divorțat	220.000	NU
8	Nu	Necasatorit	85.000	DA
9	Nu	Casatorit	75.000	NU
10	Nu	Necasatorit	90.000	DA

După cum se observă din tabel, există două clase distințe: NU și DA din punctul de vedere al posibilității de fraudare. Probabilitățile celor două clase sunt $P\{\text{NU}\} = 7/10$ și $P\{\text{DA}\} = 3/10$.

În ceea ce privește probabilitățile condiționate de tipul $P\{A_i \mid C_j\}$, în cazul atributelor discrete, acestea se vor calcula în mod natural după formula:

$$P\{A_i \mid C_j\} = \frac{|A_{ij}|}{N_{C_j}},$$

unde $|A_{ij}|$ reprezintă numărul instanțelor având atributul A_i și care aparțin clasei C_j . Astfel, utilizând această formulă, obținem:

$$P\{\text{Statut marital} = \text{,,Căsătorit"}|\text{NU}\} = 4/7$$

$$P\{\text{Rambursare} = \text{,,Da"}|\text{DA}\} = 0.$$

În cazul atributelor de tip continuu, pentru a evalua probabilitățile condiționate $P\{A_i|C_j\}$, este nevoie de identificarea tipului de repartiție a atributului, privit ca variabilă aleatoare continuă. De obicei, în afară de cazurile în care dispunem de cunoștințe apriorice asupra acestora, se presupune că toate attributele continue urmează legea normală, urmând ca din date să se estimeze parametrii acesteia (i.e. media și dispersia). Odată densitatea de repartiție estimată, putem evalua probabilitatea condiționată $P\{A_i|C_j\}$ pentru fiecare clasă în parte. În cazul nostru, atributul *Venit anual* este considerat variabilă aleatoare continuă, de densitate:

$$P\{A_i | C_j\} = \frac{1}{\sqrt{2\pi} \cdot \sigma_{ij}} \exp\left(-\frac{(A_i - \mu_{ij})^2}{2\sigma_{ij}^2}\right).$$

Astfel, media variabilei *Venit anual*, condiționată de clasa NU, este egală cu 110.000, iar dispersia sa este egală cu 2.975. Putem deci calcula probabilitatea condiționată:

$$P\{\text{Venit anual} = 120.000 | \text{NU}\} = \frac{1}{\sqrt{2\pi} \cdot (51,54)} \exp\left(-\frac{(120.000 - 110.000)^2}{2 \cdot 2975}\right),$$

adică 0,0072. Analog se calculează și în cazul clasei DA.

Să analizăm acum modul de funcționare a clasificatorului astfel construit pe un caz nou. Să presupunem că avem de clasificat un individ care are următoarele attribute:

- ⇒ Rambursare = „Nu”;
- ⇒ Statut marital = „Căsătorit”;
- ⇒ Venit anual = 120.000.

Avem:

$$\begin{aligned} P\{\text{Nu, Căsătorit, 120.000}|\text{NU}\} &= P\{\text{Nu}|\text{NU}\} \times P\{\text{Căsătorit}|\text{NU}\} \times \\ &\quad \times P\{120.000|\text{NU}\} = 4/7 \times 4/7 \times 0,0072 = \\ &= 0,0024. \end{aligned}$$

$$\begin{aligned} P\{\text{Nu, Căsătorit, 120.000}|\text{DA}\} &= P\{\text{Nu}|\text{DA}\} \times P\{\text{Căsătorit}|\text{DA}\} \times \\ &\quad \times P\{120.000|\text{DA}\} = 1 \times 0 \times 1.2 \times 10^{-9} = 0,00. \end{aligned}$$

În concluzie, deoarece:

$P\{\text{Nu, Căsătorit, 120.000}|\text{NU}\}P\{\text{NU}\} > P\{\text{Nu, Căsătorit, 120.000}|\text{DA}\}P\{\text{DA}\}$
deci:

$$P\{\text{NU}|\text{Nu, Căsătorit, 120.000}\} > P\{\text{DA}|\text{Nu, Căsătorit, 120.000}\},$$

vom clasifica persoana necunoscută în categoria **NU**, adică este probabil să nu fraudeze la rambursare.

La final, să facem o scurtă trecere în revistă a principalelor avantaje ale clasificării bayesiene (naive).

- Este robustă în ceea ce privește izolarea zgomotului din date;
- În cazul valorilor lipsă, ignoră instanța în timpul estimării probabilităților;
- Este robustă la atributele irelevante;

5.3. Rețelele neuronale artificiale

Domeniul rețelelor neuronale artificiale (*Artificial Neural Networks – ANN*), sau mai pe scurt rețelelor neuronale (*Neural Networks -NN*), reprezintă încă, la peste 60 de ani de la prima ,atestare', un câmp relativ ,neclasicizat' de cercetare. În ultimele decade NN apar ca o tehnologie practică, proiectată pentru rezolvarea cu succes a multor probleme din varii domenii: științele neuronale, matematica, statistica, fizica, informatica, științele, ingineria, biologia etc. NN sunt aplicate în modelare, analiza seriilor temporale, recunoașterea formelor, procesarea semnalelor, teoria controlului etc., datorită caracteristicii lor fundamentale, abilitatea de a învăța din datele de antrenament „cu sau fără profesor”. Cu toate că acționează după principiul cutiei negre (*black-box*) și, spre deosebire de alte tehnici ,transparente' ca de exemplu arborii de decizie, nu ,etalează' direct modul de funcționare, eficiența lor în domeniile amintite este de necontestat.

Sintetic vorbind, NN reprezintă sisteme neprogramate (ne-algoritmice) de procesare adaptivă a informației. NN învăță din exemple și se comportă ca niște ,cutii negre', modul de procesare a informației fiind neexplicit.

Putem să considerăm NN ca pe un tip de arhitectură masivă de calcul paralel, o paradigmă a procesării informației, bazată pe modelul de procesare a cunoștințelor specific creierului biologic. În principiu, asemănarea cu modul de acțiune a creierului se poate condensa în următoarele două aspecte:

- Cunoștințele sunt achiziționate de către rețea prin intermediul procesului de învățare;

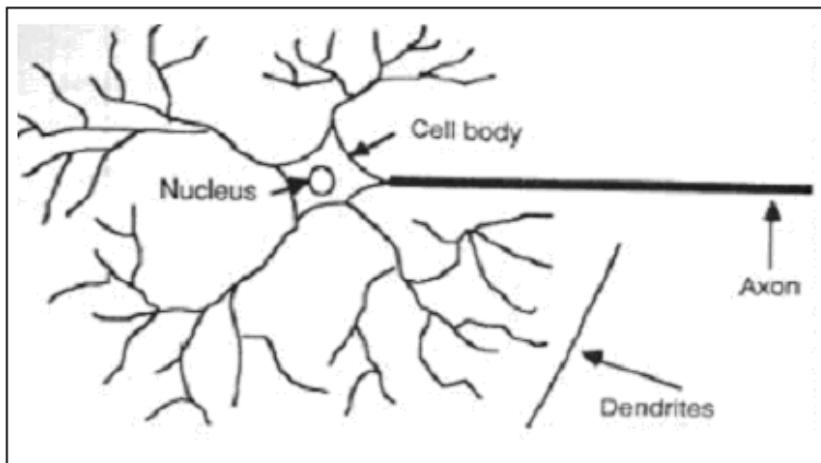
- Intensitățile conexiunilor inter-neuronale, cunoscute ca ponderi (sinaptice), sunt utilizate pentru stocarea cunoștințelor câștigate.

Paragraful acesta va fi dedicat prezentării sintetice a principiilor care stau la baza NN, precum și a câtorva aplicații ilustrând valența lor practică incontestabilă.

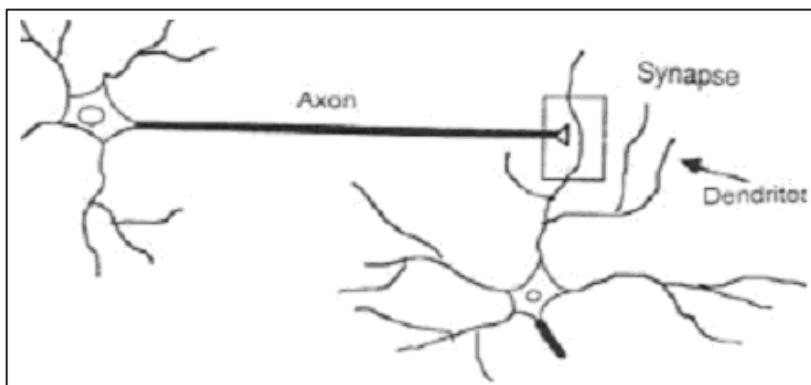
5.3.1. Perceptronul

Am putea intitula acest sub-paragraf, în mod sugestiv, astfel: „*De la neuronul artificial al lui McCulloch și Pitts la perceptronul lui Rosenblatt*”, deoarece ne propunem să prezentăm atât istoricul acestui domeniu de cercetare, cât și conceptul care stă la baza sa.

Să analizăm mai întâi conceptul biologic care stă la baza NN. Astfel, NN procesează informația în același fel ca și creierul uman, mod care a stat la baza construirii lor. Pentru a înțelege mai bine funcționarea unui model neuronal, să facem un mic recurs la neurologie. Astfel, fiecare *neuron* este o celulă specializată a creierului care poate propaga un semnal electrochimic. În creierul uman, un neuron tipic colectează semnale de la alții neuroni vecini printr-o structură de conexiuni, numite *dendrite*. Neuronul transmite apoi semnale, ca rezultat al activității electro-șimice interioare, prin intermediul unui „canal” de comunicație numit *axon*, care se divide în mii de ramuri. La capătul fiecărei ramuri există o structură numită *sinapsă*, care procesează semnalul transmis prin axon, transformându-l într-un „efect” electric care inhibă sau excită activitatea de la axon la neuronul la care este conectat. Atunci când un neuron primește un input suficient de puternic (în raport cu „pragul” de inhibiție), un așa numit input de excitație, el va trimite un semnal electrochimic („va deschide focul” (*fire*) -se mai zice) de-a lungul axonului. „Învățarea” apare prin schimbarea modului de acțiune a sinapselor, adică influența neuronului asupra celorlalți neuroni, conectați cu el. Rezumând cele de mai sus, neuronul primește informație de la ceilalți neuroni prin intermediul dendritelor, o procesează, trimite apoi semnale-răspuns prin intermediul axonului, moment în care sinapsele, prin modificarea unor „praguri” de inhibiție/excitare, controlează acțiunea asupra neuronului conectat. Prin „reglările” fine de la nivelul sinapselor, pe baza învățării din experiența acumulată, se obțin outputuri optime ca răspunsuri la input-urile primite. Un neuron este astfel fie inhibat, fie excitat, în funcție de semnalul primit de la alt neuron și, în raport de aceasta, va răspunde sau nu, influențând acțiunea neuronilor conectați în rețea. Figura de mai jos ilustrează sintetic arhitectura de bază a unui neuron natural.



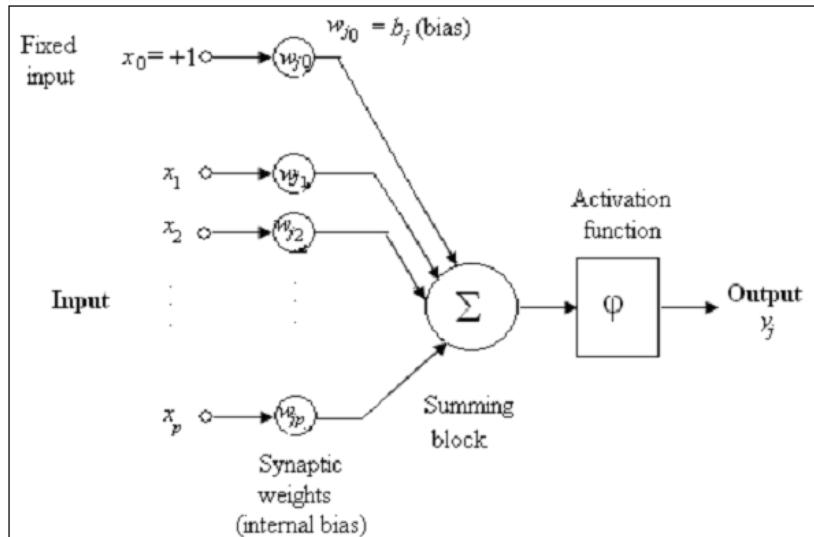
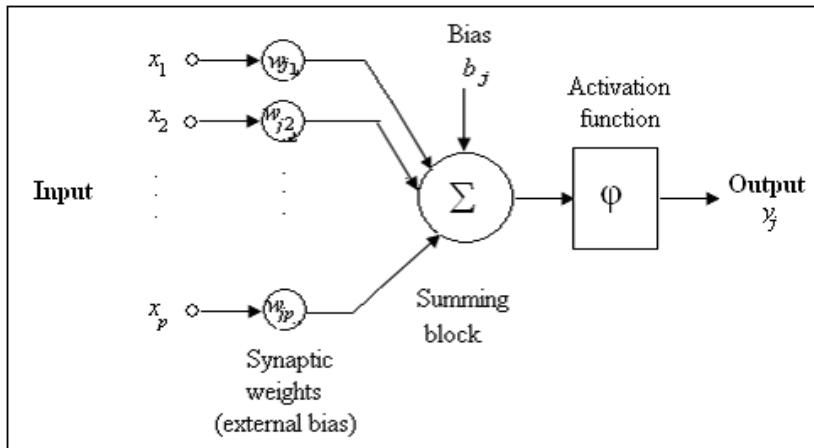
În figura următoare, ilustrăm conexiunea sinaptică care leagă neuronii într-o rețea neuronală.



Conform lui Kohonen [111], primul neuron artificial a fost „construit” în anul 1943 de către neurofiziologul Warren McCulloch și logicianul Walter Pitts. Cu toate că cei doi au dezvoltat ipoteze și teoreme privind modele de calcul natural, la acea vreme puține dintre ele au fost implementate, de vină fiind tehnologia computațională precară a perioadei. Putem deci privi acest demers al lor ca un simplu model matematic al comportamentului unui singur neuron natural, în cadrul unui sistem nervos.

Modelul unui neuron artificial, care caută să „imită” conceptul neuronului natural, este prezentat în următoarele două figuri (atât pentru cazul „deplasărilor” (*bias*) interne cât și externe). La fel ca și în cazul figurilor

anterioare, denumirile sunt considerate în lb. Engleză, în original, pentru a fi mai ușor corelate cu terminologia uzuală internațională.



Un neuron artificial, privit ca un singur mecanism, are un anumit număr p de intrări (*input-uri*), privite ca valori reale x_i , ponderate cu ponderile w_i , care sunt însumate și apoi introduse într-o funcție de activare φ , pentru a produce o anumită ieșire (*output*), depinzând de un anumit „prag” T .

La fel ca și în cazul unui sistem nervos natural, și aici NN este compusă dintr-un număr mare de procesare – neuronii artificiali – puternic

interconectați și lucrând în paralel pentru rezolvarea unei anumite probleme. În consecință, vom privi neuronul artificial ca pe componenta de bază a unei NN și nu ca pe un mecanism singular, lucrând independent. Astfel, într-o structură NN, un input real x_i care ajunge la intrarea în sinapsa i , conectată la neuronul j , va fi înmulțit cu ponderea sinaptică w_{ji} . În acest model neuronal matematic, input-urile (scalare) x_i reprezintă tocmai nivelele de activitate (output-uri) ale altor neuroni, care sunt conectați la neuronul în cauză, iar ponderile w_{ji} reprezintă constrângerile interconexiunilor (sinapselor) între neuroni. Să remarcăm că, spre deosebire de sinapsele din creier, ponderile sinaptice ale unui neuron artificial pot apartine unui interval care să conțină și valori negative. Mai departe, un „prag” prestabilit, notat T_j , este atribuit fiecărui neuron j . Neuronul artificial mai include și un input constant (fictiv), pe moment considerat extern, b_j , numit *deplasare (bias)*, având scopul de a crește sau descrește (a deplasa) input-ul de activare a rețelei, în funcție de semnul său (pozitiv sau negativ). Cu alte cuvinte, deplasarea b_j reprezintă „pragul” de declanșare (*firing*) a neuronului.

Dacă suma input-urilor mărită cu ponderile sinaptice (adică un produs scalar) depășește valoarea „prag” T_j , atunci ea va fi procesată de funcția de activare φ pentru a produce un anumit output nenul, altfel va fi considerat ca output valoarea zero. Din punct de vedere matematic, putem descrie „activitatea” neuronului j al rețelei neuronale prin următoarea ecuație:

$$u_j = \sum_{i=1}^p w_{ji} \cdot x_i = \mathbf{w}_j \cdot \mathbf{x}^T$$

unde $\mathbf{x} = (x_1, x_2, \dots, x_p)$ reprezintă vectorul input, $\mathbf{w}_j = (w_{j1}, w_{j2}, \dots, w_{jp})$ reprezintă vectorul ponderilor (sinaptice), iar u_j este output-ul corespunzător input-ului \mathbf{x} .

Activitatea de declanșare a neuronului este dată de ecuația:

$$y_j = \varphi(u_j + b_j) = \begin{cases} h_j, & u_j + b_j \geq T_j \\ 0, & u_j + b_j < T_j \end{cases}$$

unde b_j reprezintă *deplasarea*, φ este *funcția de activare* (în general, o funcție monotonă), iar y_j este *semnalul-iesire (output signal)* al neuronului. Să menționăm că uzual, pragul T_j este ales egal cu zero.

Inițial am considerat deplasarea b_j ca pe un parametru extern (vezi figura neuronului cu deplasare externă). O putem privi, alternativ, și ca parametru intern (vezi figura neuronului cu deplasare internă), considerând $b_j = w_0$ ca fiind o pondere de la un extra-input $x_0 = 1$, astfel încât:

$$u_j = \sum_{i=0}^p w_{ji} \cdot x_i ,$$

și

$$y_j = \varphi(u_j) .$$

Așa cum am arătat mai sus, funcția de activare φ definește ieșirea (output-ul) unui neuron artificial pe baza unei combinații liniare u , cunoscută sub numele de *output combinator liniar (linear combiner output)*. Menționăm, în continuare, șapte dintre cele mai cunoscute tipuri de funcții de activare.

1. Funcția de activare *Heaviside (funcția ,prag)*:

$$\varphi(u) = \begin{cases} 0, & u < 0 \\ 1, & u \geq 0 \end{cases} .$$

În acest caz, cunoscut și ca modelul *originar McCulloch-Pitts (MCP)*, există un output binar, descriind așa-numita proprietate ,totul-sau-nimic’ (*all-or-none property*) a unui MCP. O alternativă la funcția de activare Heaviside este cea dată de:

$$\varphi(u) = \begin{cases} -1, & u < 0 \\ +1, & u \geq 0 \end{cases} .$$

O generalizare simplă a funcției ,prag’ este reprezentată de funcția de activare ,treaptă’ (*step activation function (hard limiter)*), dată de:

$$\varphi(u) = \begin{cases} a, & u < h \\ b, & u \geq h \end{cases} .$$

2. Funcția de activare *liniară pe porțiuni (piecewise-linear activation function -ramp function)*, dată de (variantă):

$$\varphi(u) = \begin{cases} -a, & u \leq -c \\ u, & |u| < c \\ a, & u \geq c \end{cases} .$$

3. Funcția de activare *liniară (linear activation function)*, dată de:

$$\varphi(u) = a \cdot u .$$

4. Funcția de activare *gaussiană* (*gaussian activation function*), dată de:

$$\varphi(u) = \exp(-u^2 / a).$$

5. Funcția de activare *sigmoidă* (*sigmoid activation function*), cu o reprezentare grafică de tip „S” (de unde și numele), este pe de parte cea mai utilizată funcție de activare în NN. Un exemplu de astfel de funcție este reprezentat de *funcția logistică* (*logistic function*), care aplică intervalul $(-\infty, \infty)$ peste intervalul $(0, 1)$, fiind dată de formula:

$$\varphi(u) = \frac{1}{1 + \exp(-a \cdot u)},$$

unde a este parametrul *pantă* (*slope parameter*) al sigmoidei. Un alt exemplu de sigmoidă utilizată ca funcție de activare pentru NN este *funcția tangentă hiperbolică* (*hyperbolic-tangent function*), dată de:

$$\varphi(u) = \tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}},$$

care diferă de funcția logistică printr-o transformare liniară.

Remarcă: Modelul neuronal descris mai sus este unul determinist, cu alte cuvinte comportamentul intrare/ieșire (*input/output*) este determinat pentru toate input-urile. Există și o alternativă stochastică a modelului, în care decizia de declanșare este probabilistă. Astfel, fie X starea binară a unui neuron și fie u output-ul combinator liniar. Atunci activitatea de declanșare este dată de:

$$X : \begin{cases} \text{declanseaza,} & \text{cu probabilitatea } P(u) \\ \text{nu declanseaza,} & \text{cu probabilitatea } 1 - P(u) \end{cases}.$$

Modelul perceptronului Rosenblatt

Modelul de bază al unui neuron artificial, cunoscut și ca modelul McCulloch-Pitts ca o recunoaștere a muncii de pionierat a celor doi cercetători în „zorii epocii” NN, are aşa cum s-a observat ponderi fixe și output binar, din nefericire fără posibilitatea de învățare sau adaptare. Pasul următor a fost reprezentat de modelul perceptronului Rosenblatt (*Rosenblatt's perceptron model* -1957) [135], [136]). Astfel, plecând de la modelul inițial, perceptronul a fost dotat cu un mecanism simplu de învățare, bazat pe feedback-ul diferenței (erorii) între output-ul obținut și cel dorit. Numele de *perceptron* vine de la scopul originar al lui Rosenblatt: „să distingă (perceapă) imagini alb&negru ale

unor forme geometrice, utilizând input-uri binare primite de la foto-senzori, și funcția de activare „treaptă” (*hard limiter*)”.

Scopul utilizării perceptronului este acela de clasificare corectă a unui set de stimuli externi într-una din două clase (categorii decizionale) C_1 și C_2 . Regula decizională pentru problema de clasificare este aceea de a atribui (asigna) punctul \mathbf{x} , reprezentat de vectorul $\mathbf{x} = (x_1, x_2, \dots, x_p)$, clasei C_1 dacă output-ul perceptronului este +1 și clasei C_2 , dacă output-ul este -1 (sau echivalent, 0 și 1). În cele ce urmează vom utiliza doar modelul cu deplasare internă, adică atunci când vectorul input este $\mathbf{x} = (1, x_1, x_2, \dots, x_p)$ și vectorul ponderilor (sinaptice) este $\mathbf{w} = (w_0, w_1, w_2, \dots, w_p)$.

În forma cea mai simplă a unui perceptron există două regiuni de decizie în spațiul $(p + 1)$ -dimensional al input-urilor, regiuni separate de un hiperplan, dat de formula:

$$u = \sum_{i=0}^p w_i \cdot x_i = 0,$$

adică două regiuni liniar separabile. Concret, există un vector pondere \mathbf{w} , astfel încât:

- $\mathbf{w} \cdot \mathbf{x}^T > 0$ pentru orice vector input \mathbf{x} aparținând clasei C_1 ,
- $\mathbf{w} \cdot \mathbf{x}^T < 0$ pentru orice vector input \mathbf{x} aparținând clasei C_2 .

Ponderile (sinaptice) $w_0, w_1, w_2, \dots, w_p$ pot fi adaptate utilizând o procedură de tip *iteration-by-iteration*.

Prezentăm mai jos (în lb. Engleză, în original) algoritmul de învățare al perceptronului originar [98], cu două clase de decizie $C_1 \sim P^+$ și $C_2 \sim P^-$.

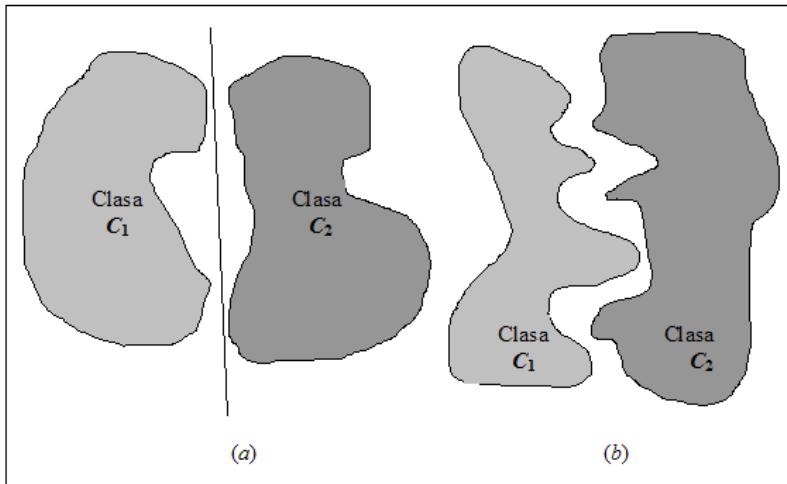
Input: A set of positive and negative vector samples, denoted by P^+ and P^- .

1. Initialise the weight vector \mathbf{w} to zero.
2. Choose a vector sample \mathbf{x} from the sample set $P^+ \cup P^-$.
3. IF $\{\mathbf{x} \in P^+ \text{ and } \mathbf{w} \cdot \mathbf{x}^T > 0\}$ or $\{\mathbf{x} \in P^- \text{ and } \mathbf{w} \cdot \mathbf{x}^T < 0\}$ THEN GOTO step 2.
4. IF $\{\mathbf{x} \in P^+ \text{ and } \mathbf{w} \cdot \mathbf{x}^T < 0\}$ THEN $\mathbf{w} = \mathbf{w} + \mathbf{x}$; GOTO step 2.
5. IF $\{\mathbf{x} \in P^- \text{ and } \mathbf{w} \cdot \mathbf{x}^T > 0\}$ THEN $\mathbf{w} = \mathbf{w} - \mathbf{x}$; GOTO step 2.

Output: A single weight vector \mathbf{w} for the linear threshold function that classify the input samples in P^+ and P^- , if such a vector exists.

Remarcă: Pentru ca perceptronul să „funcționeze” corespunzător, este necesar ca cele două clase C_1 și C_2 să fie suficient de „liniar separabile”, altfel decizia va depăși puterea de calcul a perceptronului. În figura

de mai jos (a și b) prezentăm ambele situații (liniar separabilitatea, respectiv nelinier separabilitatea claselor).



Pentru a demonstra valabilitatea utilizării algoritmului de învățare de tipul *corectarea erorii* (*error-correction*), vom prezenta un rezultat interesant care arată că, pentru orice set de date care este liniar separabil, regula de învățare va găsi în mod garantat o soluție într-un număr finit de pași.

Să presupunem că variabilele input ale perceptronului sunt extrase dintr-o mulțime de antrenament C , care este liniar separabilă, deci obiectele (pattern-urile) ce trebuie clasificate sunt suficient de separate unul față de altul, încât suprafața de decizie este un hiperplan. În consecință, va exista o partitie $\{C_1, C_2\}$ a lui C (i.e. $C = C_1 \cup C_2$, C_1 și C_2 disjuncte) și un hiperplan între aceste două componente, reprezentând frontieră de decizie (*decision boundary*) – cazul (a) din figura de mai sus. Din punct de vedere matematic, deoarece C_1 și C_2 sunt liniar separabile, va exista un vector pondere \mathbf{w} , astfel încât:

- $\mathbf{w} \cdot \mathbf{x}^T > 0$ pentru orice obiect de antrenament $\mathbf{x} \in C_1$,
- $\mathbf{w} \cdot \mathbf{x}^T < 0$ pentru orice obiect de antrenament $\mathbf{x} \in C_2$.

Regula de învățare din algoritmul de mai sus poate fi generalizată sub forma următoare:

- Dacă vectorul \mathbf{x}_n din C este clasificat corect cu ajutorul vectorului pondere $\mathbf{w}(k)$, calculat la a k -a iterație a algoritmului, atunci:

$$\begin{aligned} \Rightarrow \mathbf{w}(k+1) &= \mathbf{w}(k) & \text{if } \mathbf{w}(k) \cdot \mathbf{x}_n^T > 0 \text{ and } \mathbf{x}_n \in C_1; \\ \Rightarrow \mathbf{w}(k+1) &= \mathbf{w}(k) & \text{if } \mathbf{w}(k) \cdot \mathbf{x}_n^T < 0 \text{ and } \mathbf{x}_n \in C_2. \end{aligned}$$

- Altfel:

$$\Leftrightarrow \mathbf{w}(k+1) = \mathbf{w}(k) - \eta \cdot \mathbf{x}_n \quad \text{if } \mathbf{w}(k) \cdot \mathbf{x}_n^T > 0 \text{ and } \mathbf{x}_n \in C_2,$$

$$\Leftrightarrow \mathbf{w}(k+1) = \mathbf{w}(k) + \eta \cdot \mathbf{x}_n \quad \text{if } \mathbf{w}(k) \cdot \mathbf{x}_n^T < 0 \text{ and } \mathbf{x}_n \in C_1,$$

unde parametrul *rata-de-învățare* η este ales în mod ușual ca o constantă independentă de iterație.

Putem rezuma algoritmul de mai sus, astfel: „Se rulează toate obiectele din mulțimea de antrenament și se testează fiecare, utilizând valorile actuale ale ponderilor. Dacă obiectul este clasificat corect, se păstrează valorile curente ale ponderilor, altfel se adună vectorul-obiect multiplicat cu η la vectorul pondere, dacă obiectul aparține lui C_1 , sau se scade vectorul-obiect multiplicat cu η la vectorul pondere dacă obiectul aparține lui C_2 ”.

Deoarece $\|\mathbf{x}_n\|^2 > 0$, rezultă că:

$$\mathbf{w}(k+1) \cdot \mathbf{x}_n^T = \mathbf{w}(k) \cdot \mathbf{x}_n^T - \eta \cdot \mathbf{x}_n \cdot \mathbf{x}_n^T < \mathbf{w}(k) \cdot \mathbf{x}_n^T - \text{primul caz,}$$

$$\mathbf{w}(k+1) \cdot \mathbf{x}_n^T = \mathbf{w}(k) \cdot \mathbf{x}_n^T + \eta \cdot \mathbf{x}_n \cdot \mathbf{x}_n^T > \mathbf{w}(k) \cdot \mathbf{x}_n^T - \text{al doilea caz,}$$

și deci procedura de mai sus va produce reducerea erorii de clasificare.

Remarcă: Se observă că valoarea parametrului η nu este așa de importantă, atât timp cât este pozitivă, ea doar re-dimensionează ponderile, locația frontierei de decizie $\mathbf{w} \cdot \mathbf{x}^T = 0$ rămânând aceeași. Astfel, atunci când minimizăm criteriul perceptronului, putem alege $\eta = 1$.

Prezentăm acum *teorema de convergență* a perceptronului.

Teoremă (Rosenblatt, 1962). Să presupunem că submulțimile C_1 și C_2 ale mulțimii obiectelor de antrenament sunt liniar separabile. Atunci perceptronul va converge după un număr finit k_0 de iterări.

Demonstrație: Vom prezenta o demonstrație simplificată, urmând Hertz *et al.* (1991), [97].

Să demonstrăm mai întâi convergența unei reguli de adaptare cu incrementare fixată η , alegând $\eta = 1$.

Deoarece s-a presupus că cele două componente ale setului de antrenament sunt liniar separabile, atunci va exista cel puțin un vector pondere \mathbf{w}_0 , pentru care toate obiectele de antrenament sunt corect clasificate, deci $\mathbf{w}_0 \cdot \mathbf{x}_n^T > 0$, pentru toți vectorii \mathbf{x}_n aparținând clasei C_1 ; să punem $\alpha = \min_{\mathbf{x}_n \in C_1} \mathbf{w}_0 \cdot \mathbf{x}_n^T$. Procesul de învățare începe cu un vector pondere arbitrar, care poate fi considerat

$\mathbf{w}(0) = \mathbf{0}$. Să presupunem că perceptronul va clasifica incorect vectorii $\mathbf{x}_1, \mathbf{x}_2, \dots$, adică $\mathbf{w}(k) \cdot \mathbf{x}_n^T < 0$, pentru toți $n = 1, 2, \dots$, și că vectorul input \mathbf{x}_n aparține clasei C_1 . Atunci, la fiecare pas al algoritmului, vectorul pondere va fi actualizat după formula:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{x}_k, \quad (1)$$

pentru $\mathbf{x}_k \in C_1$, reprezentând un vector clasificat greșit. Deoarece după rularea algoritmului o anumită perioadă de timp, fiecare vector \mathbf{x}_k a fost luat în considerație și clasificat greșit, atunci, dată fiind condiția inițială $\mathbf{w}(0) = \mathbf{0}$, vectorul pondere curent va fi dat de formula:

$$\mathbf{w}(k+1) = \sum_j \mathbf{x}_j.$$

Înmulțind (scalar) ambii membri ecuației cu \mathbf{w}_0 , se obține:

$$\mathbf{w}_0 \cdot \mathbf{w}^T(k+1) = \sum_j \mathbf{w}_0 \cdot \mathbf{x}_j^T \geq k\alpha. \quad (2)$$

Utilizând acum inegalitatea Cauchy-Schwarz, obținem:

$$\|\mathbf{w}_0\|^2 \|\mathbf{w}(k+1)\|^2 \geq [\mathbf{w}_0 \cdot \mathbf{w}^T(k+1)]^2. \quad (3)$$

Din (2) și (3) avem:

$$\|\mathbf{w}_0\|^2 \|\mathbf{w}(k+1)\|^2 \geq k^2 \alpha^2,$$

sau, echivalent:

$$\|\mathbf{w}(k+1)\|^2 \geq \frac{k^2 \alpha^2}{\|\mathbf{w}_0\|^2}. \quad (4)$$

Rezultă că, deoarece \mathbf{w}_0 este fixat, valoarea lui $\|\mathbf{w}(k+1)\|^2$ este mărginită inferior de o funcție care crește liniar în raport cu pătratul numărului iterațiilor k^2 .

Să considerăm acum în alt mod mărimea vectorului pondere $\mathbf{w}(k+1)$. Astfel, din formula de actualizare (1), avem:

$$\|\mathbf{w}(j+1)\|^2 = \|\mathbf{w}(j)\|^2 + \|\mathbf{x}_j\|^2 + 2\mathbf{w}(j) \cdot \mathbf{x}_j^T \leq \|\mathbf{w}(j)\|^2 + \|\mathbf{x}_j\|^2, \quad j = 1, 2, \dots,$$

unde $\mathbf{w}(j) \cdot \mathbf{x}_j^T < 0$, deoarece obiectul $\mathbf{x}_j \in C_1$ a fost clasificat greșit de perceptron.

Astfel, schimbarea valorii normei vectorului pondere \mathbf{w} satisfacă inegalitatea:

$$\|\mathbf{w}(j+1)\|^2 - \|\mathbf{w}(j)\|^2 \leq \|\mathbf{x}_j\|^2, \quad j = 1, 2, \dots$$

Să notăm cu $\beta = \max_{\mathbf{x}_j \in C_1} \|\mathbf{x}_j\|^2$ - lungimea celui mai mare vector input din C_1 .

Atunci:

$$\|\mathbf{w}(j+1)\|^2 - \|\mathbf{w}(j)\|^2 \leq \beta. \quad (5)$$

Adunând inegalitățile (5) pentru $j = 1, 2, \dots, k$, atunci, după k actualizări ale vectorului pondere, avem:

$$\|\mathbf{w}(k+1)\|^2 \leq k\beta, \quad (6)$$

și astfel valoarea lui $\|\mathbf{w}(k+1)\|^2$ crește cel mult liniar în raport cu numărul de iterații k .

În final, din (4) și (6), obținem:

$$k\beta \geq \frac{k^2\alpha^2}{\|\mathbf{w}_0\|^2},$$

sau, echivalent,

$$k \leq \frac{\beta \|\mathbf{w}_0\|^2}{\alpha^2}. \quad (7)$$

Astfel, numărul k al iterațiilor nu poate crește la nesfârșit și deci algoritmul va converge după un număr finit de pași. Din (7) rezultă și că numărul k nu poate fi mai mare decât $k_{\max} = \frac{\beta \|\mathbf{w}_0\|^2}{\alpha^2}$.

Să considerăm acum cazul general al unui rate variabile $\eta(n)$, depinzând de numărul n de iterații. Pentru simplitate, să considerăm $\eta(n)$ ca cel mai mic întreg, astfel încât:

$$|\mathbf{w}(n) \cdot \mathbf{x}_n^T| < \eta(n) \cdot \mathbf{x}_n \cdot \mathbf{x}_n^T.$$

Să presupunem că la iterația n avem o clasificare greșită, adică:

$$\mathbf{w}(n) \cdot \mathbf{x}_n^T > 0 \text{ și } \mathbf{x}_n \in C_2,$$

sau:

$$\mathbf{w}(n) \cdot \mathbf{x}_n^T < 0 \text{ și } \mathbf{x}_n \in C_1.$$

Atunci, dacă vom modifica, fără a pierde din generalitate, secvența de antrenament la iterația $(n+1)$, punând $\mathbf{x}_{n+1} = \mathbf{x}_n$, obținem:

$$\mathbf{w}(n+1) \cdot \mathbf{x}_n^T = \mathbf{w}(n) \cdot \mathbf{x}_n^T - \eta(n) \cdot \mathbf{x}_n \cdot \mathbf{x}_n^T < 0 \text{ și } \mathbf{x}_n \in C_2,$$

sau:

$$\mathbf{w}(n+1) \cdot \mathbf{x}_n^T = \mathbf{w}(n) \cdot \mathbf{x}_n^T + \eta(n) \cdot \mathbf{x}_n \cdot \mathbf{x}_n^T > 0 \text{ și } \mathbf{x}_n \in C_1,$$

adică o clasificare corectă pentru obiectul \mathbf{x}_n .

Astfel, fiecare obiect de antrenament este luat în considerație în mod repetat de către perceptron, până când va fi clasificat corect.

q.e.d.

În concluzie, am demonstrat în teorema de mai sus că, dacă există un vector pondere \mathbf{w}_0 (nu neapărat unic) pentru care toate obiectele de antrenament sunt corect clasificate, atunci regula de adaptare a ponderilor sinaptice ale perceptronului trebuie să ia sfârșit după cel mult k_{\max} iterații.

Prezentăm mai jos, succint, algoritmul de convergență al perceptronului (Lippmann, 1987), [116], cu două clase de decizie C_1 și C_2 (în lb. Engleză, în original):

Input: $\mathbf{x}(n) = (1, x_1(n), x_2(n), \dots, x_p(n))$ - input training vector

$\mathbf{w}(n) = (b(n), w_1(n), w_2(n), \dots, w_p(n))$ - weight vector

$y(n)$ -actual response

$d(n)$ -desired response

η - constant learning-rate parameter (positive and less than unity)

1. *Initialisation.* Set $\mathbf{w}(0) = \mathbf{0}$.

Perform the following computations for time step $n = 1, 2, \dots$.

2. *Activation.* At time step n , activate the perceptron by applying continuous-valued input training vector $\mathbf{x}(n)$ and desired response $d(n)$.

3. *Computation of actual response.* Compute the actual response of the perceptron, given by:

$$y(n) = \text{sign} \left[\mathbf{w} \cdot \mathbf{x}(n)^T \right],$$

where *sign* represents the signum function.

4. *Adaptation of the weight vector.* Update the weight vector of the perceptron:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta [d(n) - y(n)] \cdot \mathbf{x}(n),$$

where:

$$d(n) = \begin{cases} +1, & \mathbf{x}(n) \in C_1 \\ -1, & \mathbf{x}(n) \in C_2 \end{cases}$$

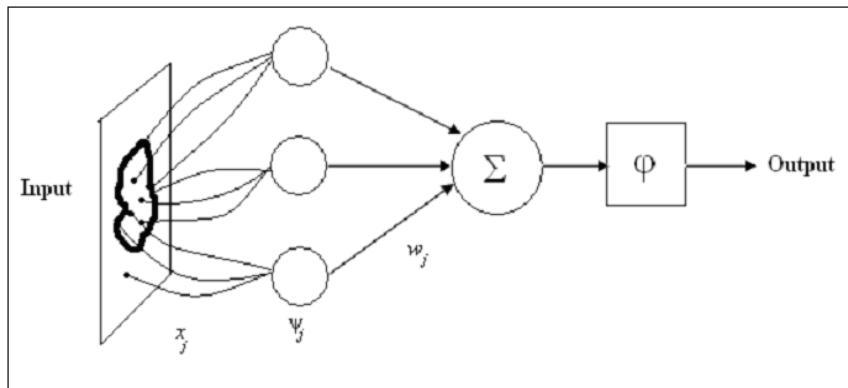
5. *Continuation.* Increment time step n by one and GOTO step 2.

Output: After a finite number of time steps n , the rule for adapting the synaptic weights of the perceptron must terminate.

(Pentru mai multe detalii, vezi și [96]).

Am văzut că perceptronul descris mai sus reprezintă o rețea cu un singur strat de ponderi sinaptice, care utilizează doar date input primare, neprelucrate, și care în consecință are capacitate foarte limitată.

Pentru a îmbunătăți performanța perceptronului originar, Rosenblatt a utilizat un strat adițional de elemente de procesare fixate $\Psi = (\psi_0, \psi_1, \dots, \psi_p)$, cu scopul de a transforma data input primară $\mathbf{x} = (x_0, x_1, x_2, \dots, x_p)$, aşa cum se observă în figura de mai jos (am utilizat din nou convenția $\psi_0 = 1$, cu deplasarea internă corespunzătoare w_0).



ACESTE ELEMENTE DE PROCESARE IAU DE OBICEI FORMA UNOR PONDERI FIXATE, CONECTATE LA O SUBMULȚIME ALEATOARE DE INPUT-URI ȘI AVÂND FUNCȚIE DE ACTIVARE BINARĂ. ASTFEL, OUTPUT-UL PERCEPTRONULUI VA FI DAT DE FORMULA:

$$y = \varphi \left(\sum_{i=0}^p w_i \cdot \psi_i(\mathbf{x}) \right) = \varphi(\mathbf{w} \cdot \Psi^T).$$

Deoarece scopul utilizării algoritmului de învățare al perceptronului este acela de a produce un sistem efectiv de clasificare, este natural să definim o *funcție eroare* în termeni de „*numărul total de clasificări eronate raportate la*

datele de antrenament'. În consecință, vom introduce o funcție eroare, care să fie continuă și liniară pe porțiuni, numită *criteriul perceptronului* (*perceptron criterion*). Vom asocia fiecărui vector de antrenament $\mathbf{x}_n \in C$ o valoare țintă t_n , definită de:

$$t_n = \begin{cases} +1, & \mathbf{x}_n \in C_1 \\ -1, & \mathbf{x}_n \in C_2 \end{cases}.$$

Deoarece din algoritmul de antrenare al perceptronului avem $\mathbf{w} \cdot \mathbf{x}_n^T > 0$ pentru vectori aparținând submulțimii C_1 și $\mathbf{w} \cdot \mathbf{x}_n^T < 0$ pentru vectori aparținând submulțimii C_2 , rezultă că, pentru toate obiectele de antrenament, $\mathbf{x}_n \in C$, avem $\mathbf{w} \cdot (t_n \cdot \mathbf{x}_n)^T > 0$. Dacă M reprezintă mulțimea vectorilor \mathbf{x}_n , incorect clasificări de către vectorul pondere curent \mathbf{w} , atunci putem defini funcția eroare (perceptron criterion) prin:

$$E(\mathbf{w}) = - \sum_{x_n \in M} \mathbf{w} \cdot (t_n \cdot \mathbf{x}_n)^T,$$

care este de obicei pozitivă (și egală cu zero dacă toate obiectele de antrenament sunt corect clasificate). În concluzie, în timpul procesului de învățare a perceptronului, încercăm să minimizăm funcția eroare $E(\mathbf{w})$.

Pentru mai multe detalii privind teorema de convergență a perceptronului, vezi [16] și [96].

Note și comentarii. Introducerea primului neuron artificial de către McCulloch și Pitts în anul 1943 poate fi considerată ca reprezentând „răsăritul” domeniului rețelelor neuronale. De notat că această noțiune a influențat pe J. von Neumann să utilizeze elemente switch-delay idealizate (*idealised switch-delay elements*) derivate din ea în construcția EDVAC (*Electronic Discrete Variable Automatic Computer*), care a dezvoltat celebrul ENIAC (*Electronic Numerical Integrator and Computer*) [7]. Să ne reamintim că ENIAC, cu cele 30 de tone greutate și 18.000 de tuburi electronice (construit la University of Pennsylvania) este considerat ca primul computer („creier” electronic), „adevărat”, cu alte cuvinte este „părintele” computerelor moderne cu chip-uri de silicon și zeci de mii de microelemente electronice pe milimetru pătrat. Ca fapt divers, putem remarcă, în acest context, frica de „bug”-uri care a rămas și astăzi vie în mintea computeriștilor, cu toate că acum se referă la cu totul altceva, ținând cont de tragedia reală implicată de pătrunderea unui gândac adevărat printre acele mii de tuburi electronice și scurtcircuitele generate de „prăjirea” sa mai mult decât probabilă.

Perceptronul reprezintă cea mai simplă formă de NN, utilizată pentru clasificarea pattern-urilor liniar separabile. În principiu, el constă dintr-un singur neuron artificial cu ponderi sinaptice ajustabile și deplasare.

Perceptronul construit pe baza unui singur neuron artificial este, prin natura faptelor, limitat în ceea ce privește capabilitatea clasificării de pattern-uri aparținând la mai mult de două clase. Astfel, prin extinderea stratului de ieșire (*output layer*), astfel încât să includă mai mult de un neuron, putem clasifica pattern-uri aparținând la mai mult de două categorii.

Aproape în același timp în care Rosenblatt dezvolta perceptronul, Widrow și colaboratorii săi lucrau în același sens, utilizând un sistem cunoscut sub numele de *ADALINE (ADaptive LINear Element)*, care avea, în principiu, aceeași formă ca și perceptronul, dar care utiliza un algoritm de învățare mult mai bun -*least mean square algorithm* (LMS), a cărui extensie este utilizată în cadrul perceptronului multi-strat (*multi-layer perceptron -MLP*) [177], [178].

Atunci când perceptronul a fost studiat experimental în perioada 1960, s-a observat că el ar putea rezolva eficient multe probleme. Așa cum se întâmplă adesea, după această perioadă inițială de entuziasm, domeniul NN a cunoscut o perioadă de frustrări și „proastă reputație”, totalmente nemeritate. S-a observat atunci că, pe de altă parte, există și multe alte probleme, părând a avea același grad de dificultate, care erau imposibil de rezolvat cu ajutorul său. În timpul acestei perioade „negre” din existența sa, au fost puține încercări de a duce mai departe cercetările. O reală „lovitură” primită de perceptron în acea perioadă a fost publicarea cărții *”Perceptrons”* de către Minsky și Papert (1969), [122]. În această carte, cei doi au rezumat toate frustrările epocii împotriva NN, care se regăseau printre cercetătorii care lucrau în domeniu, fiind acceptată fără un spirit critic adekvat. Utilizând un punct de vedere matematic formal, ei au arătat că există un număr mare de probleme pe care perceptronul nu le poate soluționa, din punct de vedere practic. Dificultatea reală în utilizarea perceptronului este, din nefericire, aceea că elementele de procesare ψ_j sunt fixate de la început și deci nu pot fi ajustate (adaptate) problemei particulare considerate. Minsky și Papert au discutat o paletă de forme diferite de perceptron (corespunzătoare formelor funcțiilor ψ_j) și, pentru fiecare în parte, au dat exemple de probleme ce nu pot fi rezolvate. Această situație a pus în umbră pentru o bună bucată de vreme dezvoltarea NN, dar poate a avut și partea ei pozitivă în încercare de depășire a handicapurilor inițiale ale NN. Soluția practică a problemei dificultăților întâlnite până atunci în utilizarea perceptronului este aceea de a permite elementelor de procesare să fie adaptive, deci să fie parte a procesului de învățare și astfel ajungem la considerarea rețelelor neuronale adaptive moderne cu straturi multiple (*multi-layer adaptive neural networks*).

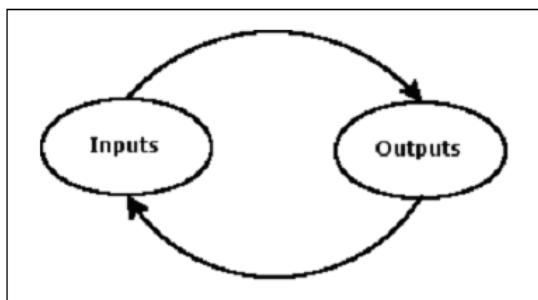
5.3.2. Tipuri de rețele neuronale artificiale

În subparagraful anterior am prezentat structura și modul de funcționare a unui singur neuron, „cărămidă” din care este construită o rețea neuronală. Pasul următor este reprezentat de modul în care mai mulți neuroni

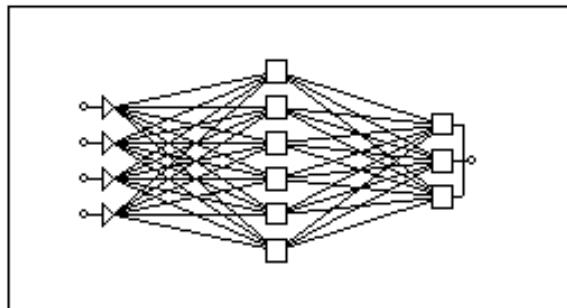
pot fi interconectați, pentru a crea o structură complexă și funcțională, o rețea neuronală adevărată. Elementele de bază ale acestei construcții sunt: unitățile de intrare (*input*), alimentate cu informații din lumea exterioară, unitățile din interiorul rețelei, neuroni ascunși (*hidden neurons*) controlând acțiunile din interiorul rețelei, precum și unitățile de ieșire (*output*), care sintetizează răspunsul rețelei. Toți acești neuroni trebuie interconectați pentru ca rețeaua să devină complet funcțională.

În demersul de a clasifica modelele NN, putem să ne bazăm pe arhitectura lor specifică, modul de operare și stilul de învățare [98]. Astfel, arhitectura NN se referă la organizarea topologică a neuronilor (numărul acestora, numărul straturilor de neuroni, structura straturilor, direcția semnalului și reciprocitatea). Modul de operare se referă la natura activităților în timpul procesării informației (dinamic sau static pentru fiecare nou input). În sfârșit, stilul de învățare se referă la modul în care NN achiziționează cunoștințe din datele de antrenament.

În acest context, o chestiune de bază este reprezentată de așa numitul *feedback* (reacție/conexiune inversă), prezent sau nu în astfel de sisteme. Pe scurt, spunem că într-un anumit sistem dinamic există feedback atunci când output-ul unui element din sistem are o anumită influență asupra input-ului aceluiași element, prin intermediul așa-numitului circuit de reacție (*feedback loop*) –vezi figura de mai jos.



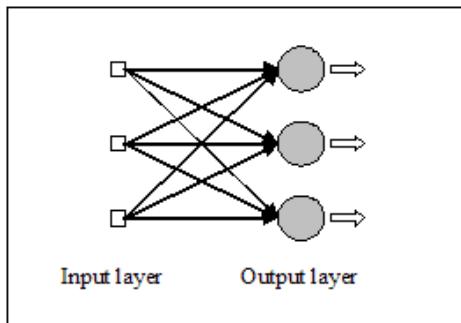
Spunem că o NN are o structură de tip *feedforward* dacă semnalul circulă de la intrare (input) spre ieșire (output), trecând prin toate unitățile ascunse ale rețelei, deci output-urile neuronilor trec spre straturile următoare și nu spre cele precedente. Aceste rețele au proprietatea că output-urile pot fi exprimate ca funcții deterministe de input-uri. Un set de valori aplicat la intrarea în rețea se transmite prin intermediul funcțiilor de activare de-a lungul rețelei spre ieșirea acestei, prin așa-numitul proces de propagare înainte (*forward propagation*). O astfel de rețea are un comportament stabil în funcționare. Figura de mai jos ilustrează schematic o astfel de structură.



Spre deosebire de NN de tip *feedforward*, există și rețele în care există circuite de reacție, rețele numite NN *recurrente*.

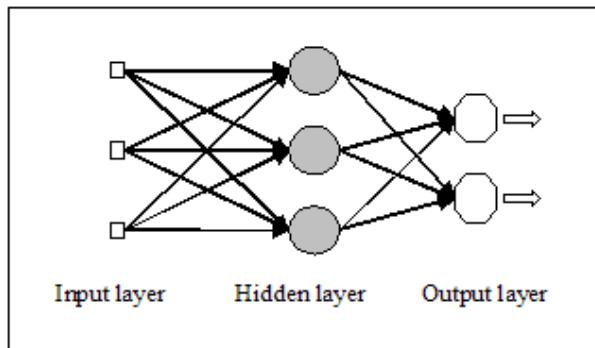
În ceea ce privește arhitectura NN, vom menționa trei categorii fundamentale de astfel de rețele. Mai întâi însă, vom reaminti că într-o rețea neuronală stratificată (*layered NN*) neuronii sunt organizați într-unul sau mai multe straturi (*layers*).

- ◆ NN feedforward cu un singur strat (*single-layer feedforward networks*). În acest caz, cel mai simplu, există un strat intrare (*input layer*) al nodurilor sursă, urmat de stratul ieșire (*output layer*) al nodurile de calcul. Menționăm că termenul de strat unic (*single layer*) se referă doar la stratul de ieșire, deoarece acesta este implicat în calcul. În figura de mai jos ilustrăm o astfel de rețea.

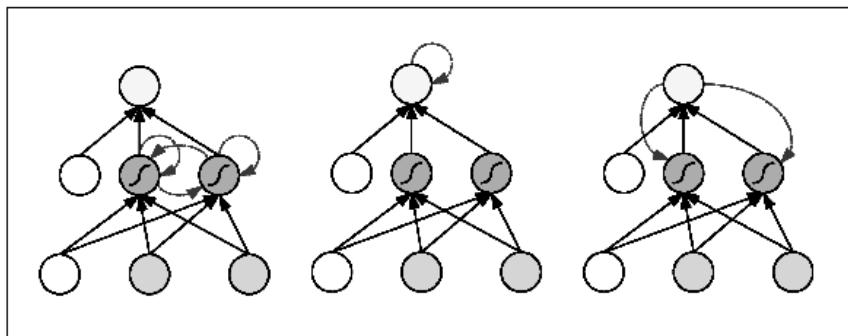


- ◆ NN feedforward multistrat (*multilayer feedforward networks*). Spre deosebire de rețeaua anterioară, în acest caz există unul sau mai multe straturi ascunse (*hidden layers*), ale căror elemente (de calcul) se numesc neuroni ascunși, rolul lor fiind acela de a acționa între stratul de intrare și cel de ieșire, în așa fel încât să îmbunătățească performanța rețelei. Schematic, prin stratul de intrare se introduce informația din exterior, care se constituie în input-urile aplicate neuronilor din stratul al doilea (i.e. primul strat ascuns), apoi fiind procesată de aceștia va deveni

output ce va constitui input-ul din stratul următor (i.e. al doilea strat ascuns) și.a.m.d. Figura de mai jos ilustrează o asemenea rețea cu un singur strat ascuns.



- ◆ NN recurente (*recurrent networks*). Așa cum am mai spus, acest tip de rețea se deosebește de cele de tip feedforward prin existența a cel puțin unui circuit de reacție. Prezența unui astfel de circuit are o importanță majoră atât în modul de învățare al rețelei cât și în performanța sa. În figura de mai jos ilustrăm trei cazuri de asemenea rețele.



Învățarea, privită în contextul NN, reprezintă procesul de adaptare la mediul exterior a rețelei (i.e. adaptarea/acordarea/reglarea parametrilor) printr-un proces de stimulare din partea mediului. Schematic, mediul stimulează rețea (NN primește input-uri din mediul exterior), parametrii sistemului capătă anumite valori ca reacție la acești stimuli, după care NN răspunde mediului exterior pe baza noii sale configurații. Deoarece există mai multe moduri de „reglare” a parametrilor rețelei, vor exista mai multe tipuri de învățare (reguli de învățare). Menționăm aici câteva dintre cele mai cunoscute astfel de reguli:

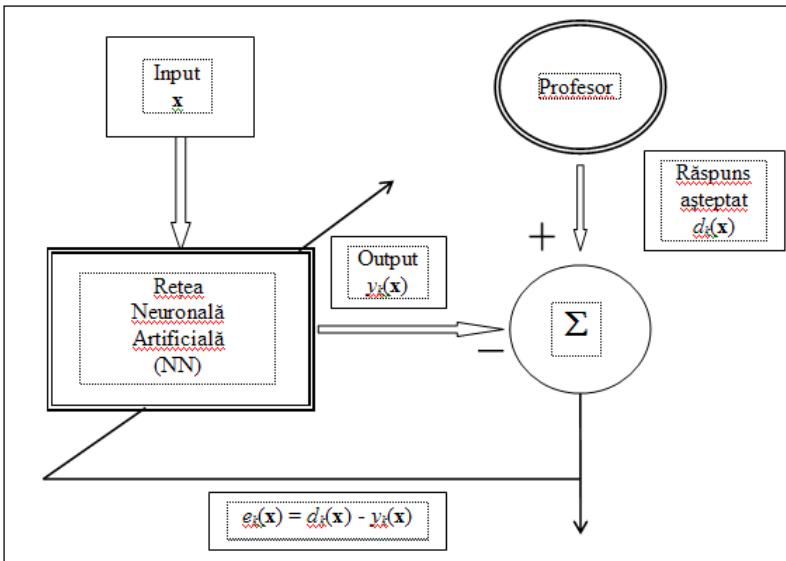
- Învățarea cu corectarea erorii (*error-correction learning*), bazată pe un mecanism de control al diferenței dintre răspunsul corect și cel real al rețelei (eroarea);
- Învățarea bazată pe memorie (*memory-based learning*), utilizând memorarea explicită a datelor de antrenament;
- Învățarea hebbiană (*hebbian learning*), bazată pe considerații neurobiologice;
- Învățarea competitivă (*competitive learning*), bazată la fel ca și cea hebbiană pe considerații neurobiologice;
- Învățarea Boltzmann (*Boltzmann learning*), bazată pe idei din mecanica statistică.

În continuare vom menționa două paradigmă fundamentale ale procesului de învățare:

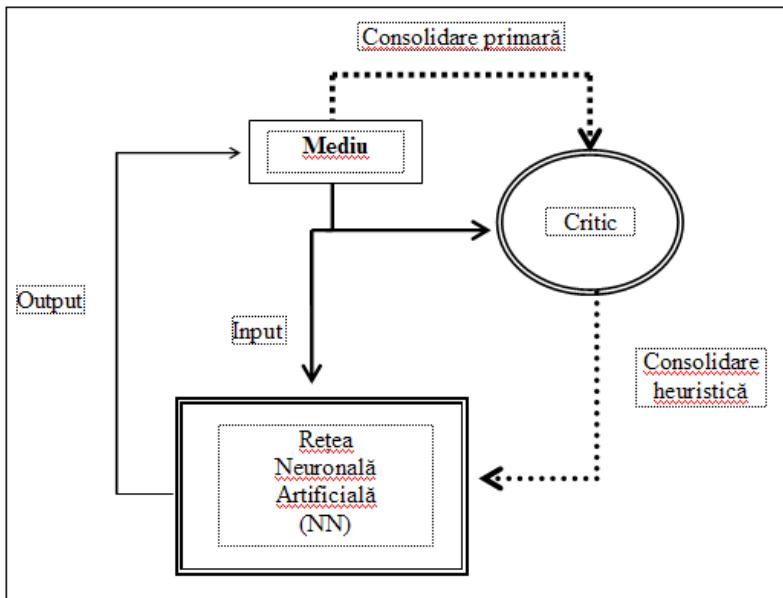
- (a) Învățarea ,cu profesor'. În acest caz, la fel ca și la școală, i se prezintă rețelei exemple complete (i.e. input-output) pentru învățare, optimizarea parametrilor efectuându-se pe baza măsurării erorii dată de diferența dintre output-ul (răspunsul) rețelei (,elevului') și răspunsul așteptat de ,profesor';
- (b) Învățarea ,fără profesor'. În acest caz se utilizează doar input-uri (fără output-urile corespunzătoare) și reglarea rețelei (i.e. optimizarea parametrilor) nu mai beneficiază de tutela unui ,profesor'. Există două categorii de astfel de învățare, depinzând de metoda după care se realizează adaptarea parametrilor.
- Învățare consolidată;
 - Învățare auto-organizată (nesupervizată).

Fără a intra în amănunte, vom prezenta pe scurt caracteristicile rețelelor neuronale, în funcție de modul de învățare.

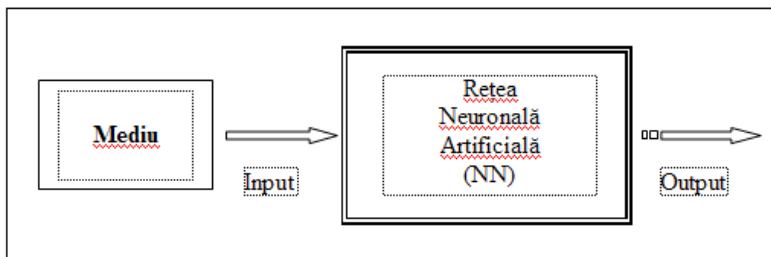
- ◆ NN cu învățare supervizată (*supervised learning*) –învățare cu profesor. În acest caz, NN este antrenat să execute o sarcină în mod repetat de către un ,profesor', prin prezentarea de exemple reprezentative de perechi input/output. În timpul fiecărei iterării de antrenament (învățare) se calculează eroarea de clasificare, privită ca diferența dintre răspunsul așteptat și cel real, furnizat de NN. Odată estimată această eroare, ea este utilizată pentru ajustarea ponderilor, în funcție de anumiți algoritmi de învățare. Pe măsură ce NN ,învață', rezultă o diminuare a erorii, până când se atinge un anumit ,prag' de acuratețe acceptabilă. Prezentăm în figura de mai jos, schematic, tipul de NN cu învățare supervizată [96], [184].



- ◆ NN cu învățare consolidată (*reinforcement learning*) - învățare fără profesor. În acest caz, în loc de calcularea exactă a erorii de clasificare (ca diferența dintre răspunsul așteptat și cel real) și utilizarea ei în optimizarea output-ului, rețelei i se indică cât de bine „lucrează” prin intermediul unui „critic” (agent) / semnal de tip *pass/fail* pentru fiecare pattern de antrenament [161]. Dacă se atribuie un eșec (*fail*), atunci rețeaua va continua să ajusteze parametrii săi până când i se atribuie o „undă verde” (*pass*), sau continuă un număr predeterminat de încercări până va accepta primul pattern de antrenament sosit. Privită câteodată ca un caz special de învățare supervizată (rolul profesorului fiind „interpretat”, în altă manieră, de către „critic”), această procedură s-a dovedit lentă și ineficientă în multe aplicații, datorită fie lipsei „profesorului” care să furnizeze răspunsul dorit, fie consolidării care apare cu o anumită întârziere (*delayed reinforcement*) [96]. În ciuda dificultăților inerente modului său de lucru, acest sistem de învățare este totuși eficient, deoarece se bazează pe interacțunea sa cu mediul, fiind utilizat în sisteme complexe de NN [160], printre aplicațiile sale putându-se cita controlul roboților, telecomunicațiile, jocul de șah. Prezentăm în figura de mai jos, schematic, un tip de NN cu învățare consolidată [10], [96].



- ◆ NN cu învățare auto-organizată (nesupervizată) (*self-organizing/unsupervised learning*) -învățare fără profesor. În acest caz, scopul învățării este fie modelarea repartițiilor datelor (input), fie descoperirea (automată) a anumitor clustere sau structuri în mulțimea input-urilor, pe baza anumitor similarități, și astfel este posibil ca acestor clustere/structuri să li se atribuie anumite categorii în funcție de natura lor și a problemei de rezolvat. Odată grupurile formate (i.e. rețeaua acordată cu modelul statistic al datelor input), NN poate fi utilizată la clasificarea pattern-urilor noi, la fel ca și în cazul învățării supervizate [16], [96], [184]. Prezentăm în figura de mai jos, schematic, un tip de NN cu învățare nesupervizată.



Vom prezenta acum, pe scurt, câteva dintre cele mai cunoscute tipuri de rețele neuronale.

- Perceptron cu un singur strat (*single-layer perceptron* -SLP). În subparagraful 5.3.1. am prezentat noțiunea de perceptron, acesta fiind cea mai simplă formă de rețea neuronală. Reamintim că, în principiu, el constă dintr-un singur neuron cu ponderi (sinaptice) ajustabile (adaptabile) și deplasare, fiind utilizat în probleme de clasificare a pattern-urilor liniar separabile. Tehnic vorbind, un SLP posedă un singur strat de ponderi adaptabile, input-urile sunt aplicate direct prin intermediul ponderilor, reprezentând, în consecință, cea mai simplă rețea de tip feedforward. Suma ponderată a input-urilor, calculată în fiecare nod, este comparată cu un anumit prag (de activare), în funcție de care se obține o anumită acțiune a rețelei. Regula de învățare în cazul SLP este așa-numita *regulă delta (delta rule)*, constând în calcularea diferenței (delta) dintre output-ul calculat de rețea și cel real și utilizarea acesteia, privită ca eroare, pentru ajustarea parametrilor (vezi și *error-correction learning*). Din cauza limitărilor semnificative ale unei astfel de arhitecturi, a fost necesară dezvoltarea de rețele cu mai multe straturi de ponderi ajustabile, creându-se astfel NN în adevăratul sens al cuvântului.
- Perceptron multi-strat (*multi-layer perceptron* –MLP). Acest tip de rețele constă în mai multe straturi de unități de calcul, conectate între ele în maniera ierarhică de tip feedforward. În principiu, caracteristicile de bază ale unui MLP pot fi sintetizate astfel.
 - Fiecare neuron al rețelei posedă o funcție de activare neliniară (*nonlinear activation function*) de clasă C^1 ;
 - MLP conține unul sau mai multe straturi ascunse;
 - Rețeaua presupune o conectivitate (sinaptică) înaltă.

MLP utilizează mai multe tehnici de învățare, cea mai populară fiind algoritmul *back-propagation*, abreviere pentru „propagarea înapoi a erorilor” (*backwards propagation of errors*). În principiu, valorile output sunt comparate cu valorile reale și se calculează eroarea cu ajutorul unei funcții-eroare predefinite, după care, în funcție de aceasta, se acționează înapoi în rețea pentru ajustarea ponderilor, în vederea minimizării erorii;

- ADALINE reprezintă o simplă rețea cu două straturi (i.e. o pondere, o deplasare și o funcție de însumare), funcția de activare la nodul output fiind neliniară de tip treaptă (cu prag binar). Există și o extensie a acestui tip de rețea, constând în conectarea în serie a două

sau mai multe astfel de rețele, cunoscută sub numele de **MADALINE** (Multiple-ADALINE).

- RBFNN (*Radial basis function neural network*). Spre deosebire de rețelele prezentate mai sus, în care unitățile de calcul utilizau o funcție de activare neliniară, bazată pe produsul scalar dintre vectorul input și vectorul pondere, în acest caz activarea unei unități (ascunse) se bazează pe distanța dintre vectorul input și un vector prototip (centru). Structura de bază a acestui tip de rețea implică trei straturi: (a) stratul input care este format din nodurile sursă ce conectează rețea la mediul exterior, (b) stratul ascuns (unicul) care aplică o transformare neliniară asupra input-ului și (c) stratul output, liniar, ce produce output-ul sistemului. Fără a intra în detalii, vom prezenta succint schema acestui tip de rețea. Astfel, dacă considerăm o NN de tip RBF, unde M reprezintă numărul de funcții bază (*basis function*), ușual mai mic decât numărul input-urilor, \mathbf{x} reprezintă un vector input, $y_k(\mathbf{x})$ este al k -lea output, w_{kj} reprezintă componenta pondere corespunzătoare și ϕ_j este o funcție bază, atunci legătura dintre input și output este dată de ecuația (vezi și subparagraful 5.3.1.):

$$y_k(\mathbf{x}) = \sum_{j=0}^M w_{kj} \cdot \phi_j(\mathbf{x}).$$

În particular, în cazul în care funcția bază este gaussiană, ecuația de mai sus devine:

$$y_k(\mathbf{x}) = \sum_{j=0}^M w_{kj} \cdot \exp\left(-\frac{\|\mathbf{x} - \mu_j\|^2}{2\sigma_j^2}\right),$$

unde μ_j reprezintă vectorul centru al funcției bază, iar σ_j este parametrul „lărgime” corespunzător. Ușual, în aplicații, se utilizează topologii cu mai mult de un singur strat ascuns. Astfel, un anumit număr de astfel de straturi, cu funcții de activare ca mai sus, sunt conectate într-o structură paralelă de tip feedforward (pentru mai multe amănunte, vezi [16], [96], [184]).

- *Kohonen self-organizing map* –SOM reprezintă o NN antrenată nesupervizată pentru a obține o reprezentare a unui spațiu input având o dimensiune arbitrară (mare) într-un spațiu output de dimensiune scăzută (o grilă regulată de noduri, 1 sau 2-dimensională), păstrând proprietățile topologice ale spațiului input. Această rețea are o structură de tip feedforward, având un singur strat dedicat calculului, format din neuroni aranjați pe linii și coloane (latici de neuroni).

Fiecare neuron este conectat la toate nodurile sursă din spațiul input, fiecărui nod output fiindu-i atașat un vector pondere (sinaptic) cu aceeași dimensiune ca și vectorii input. Aceste ponderi sunt inițializate cu valori aleatoare mici (obținute de la un generator de numere pseudo-aleatoare), înainte ca procesul de antrenament să înceapă. După inițializarea rețelei, formarea aplicației auto-organizate (*self-organizing map*) se bazează pe trei procese esențiale: (a) *competiția*, (b) *cooperarea* și (c) *adaptarea sinaptică* (pentru detalii, vezi [96]). Deoarece dimensiunea spațiului input este mult mai mare decât dimensiunea spațiului output, SOM sunt utilizate cu precădere în probleme de reducerea dimensionalității.

- NN *recurrente*, adică acele rețele care, așa cum am mai spus, prezintă circuite de reacție. Indiferent de arhitectura adoptată, aceste rețele au două trăsături comune:

- Încorporează un MLP static;
- Exploată capabilitatea MLP privind neliniaritatea aplicației.

Fără a intra în detalii (vezi [96]), menționăm patru tipuri de astfel de rețele:

- Modelul recurrent input-output (*input-output recurrent model*);
- Modelul spațiul-stărilor (*state-space model*);
- MLP recurrent (*recurrent multi-layer perceptron*);
- Rețea de ordinul II (*second-order network*).

- NN *Hopfield* (*Hopfield network/model*), inventată de fizicianul J. Hopfield (1982) reprezintă o rețea neuronală în care toate conexiunile sunt simetrice. Structura constă dintr-un set de neuroni și un sistem cu circuit multiplu de reacție (*multiple-loop feedback system*), numărul circuitelor de reacție egalând numărul neuronilor rețelei. În principiu, output-ul fiecărui neuron este conectat invers (feed-back) prin intermediul unui element unitate de întârziere (*unit delay element*) la ceilalți neuroni din rețea. O astfel de rețea este global asimptotic stabilă, având totuși limitări privind capacitatea de stocare a pattern-urilor în funcție de numărul neuronilor.

În final, vom trece în revistă câteva modele NN clasice, (împreună cu numele inventatorilor lor și anul descoperirii), făcând în acest fel și o mică incursiune în istoria domeniului:

- *Perceptronul* (Rosenblatt, 1957);
- *Adaline, Madaline*, (Widrow, Hoff, 1960-1962);

- *Avalanche* (Grossberg, 1967);
- *Cerebellation* (Marr, Albus, Pellionez, 1969);
- *Wisard* (Wilkie, Stonham, Aleksander, 1974-1980);
- *Backpropagation* (BPN), cunoscut ca *Multi-layer perceptron* (MLP) (Werbos, Parker, Rumelhart, 1974-1985);
- *Brain State in a Box* (Anderson, 1977);
- *Neocognitron* (Fukushima, 1978-1984);
- *Adaptive Resonance Theory* (ART) (Carpenter, Grossberg, 1976-1986);
- *Self-Organising Map* (SOM) (Kohonen, 1982);
- *Hopfield* (Hopfield, 1982);
- *Bi-directional Associative Memory* (Kosko, 1985);
- *Boltzmann/Cauchy machine* (Hinton, Sejnowsky, Szu, 1985-1986);
- *Counterpropagation* (Hecht-Nielsen, 1986);
- *Radial Basis Function Network* (RBFN) (Broomhead, Lowe, 1988);
- *Probabilistic Neural Network* (PNN) (Specht, 1988);
- *General Regression Neural Network* (GRNN) (Specht, 1991)/
Modified Probabilistic Neural Network (MPNN) (Zaknich *et al.* 1991);
- *Support Vector Machine* (SVM) (Vapnik, 1995).

Remarcă: 1. O clasă recentă de proceduri de învățare (statistică), inclusă tot în categoria rețelelor neuronale este reprezentată de *Mașinile cu Suport Vectorial* (SVM - *Support Vector Machine*), definite ca mașini liniare de tip feedforward, încorporând o procedură de învățare care este independentă de dimensiunea problemei. Vom vorbi mai pe larg despre aceste rețele universale de tip feedforward în sub-paragraful 5.3.4.

2. NN pot fi implementate atât ca simulări software obișnuite pe computere, ce țin de partea care ne interesează aici – domeniul DM, cât și pe hardware (partea de inginerie software), cunoscute ca neurocomputere, care sunt de două tipuri:

- Tipul cu *implementare completă*, în care există câte un procesor dedicat fiecărui neuron;
- Tipul *virtual*, care folosește un singur microcomputer de control, împreună cu o structură de neuroni virtuali (fictivi), implementați ca o serie de tabele de urmărire (*look-up tables*) a caracteristicilor rețelei (interconexiuni, ponderi etc.) -mai multe amănunte în [58].

5.3.3. Rețele neuronale probabiliste

În contextul utilizării intensive a NN ca și clasificatori, o interpretare interesantă a output-urilor rețelei este aceea de a estima probabilitatea de apartenență la o anumită clasă, caz în care NN învață, de fapt, să estimeze densități de probabilitate. Un astfel de caz special de NN, introdus de către Specht (1988, “*Probabilistic neural networks for classification mapping or associative memory*”, Proceedings IEEE International Conference on Neural Networks, 1, pp. 525-532) și cunoscut sub numele de *Rețea Neuronala Probabilistă* (*Probabilistic Neural Network* -PNN), înlocuiește funcția de activare (în general sigmoidă) cu o funcție exponențială.

Acest tip particular de NN furnizează o soluție generală pentru problemele de clasificare a formelor, utilizând o abordare probabilistă, bazată pe teoria bayesiană a deciziilor. Așa cum am arătat în paragraful 5.2., această teorie ia în considerație verosimilitatea relativă a evenimentelor și utilizează informația *a priori* pentru a îmbunătăți predicția. Astfel, paradigma rețelei utilizează estimatorii Parzen pentru obținerea densităților de probabilitate corespunzătoare claselor de decizie. În lucrarea sa clasică, Parzen (1962, “*On estimation of a probability density function and mode*”, Ann. Math. Stat., 33, pp. 1065-1076) a arătat că o clasă specială de estimatori converg asimptotic, în anumite condiții, către densitatea așteptată. Cacoulous (1966, “*Estimation of a multivariate density*”, Ann. Inst. Stat. Math (Tokyo), 18, pp. 179-189) a extins metoda lui Parzen în cazul multivariat.

PNN utilizează o mulțime supervizată de antrenament cu scopul de a estima densități de probabilitate, corespunzătoare claselor de decizie. Antrenarea PNN este mult mai simplă decât în cazul altor NN. Avantajul principal al PNN constă atât în faptul că antrenamentul necesită doar o singură procedură, cât și că hiper-suprafețele de decizie obținute tind garantat spre frontierele de decizie optimă Bayes, atunci când numărul pattern-urilor de antrenament crește. Nu trebuie însă să trecem sub tăcere principalul lor dezavantaj care se referă la faptul că PNN necesită stocarea mulțimii de antrenament pe parcursul etapei de clasificare a noi pattern-uri, ceea ce implică utilizarea intensă a memoriei și creșterea timpului de calcul, pe măsură ce dimensiunea vectorilor input crește, ca și volumul mulțimii de training. Cu alte

cuvinte, PNN necesită tot timpul bagajul de cunoștințe ,la purtător’, neînvățând odată pentru totdeauna, ca celelalte rețele.

Vom arăta acum maniera în care PNN utilizează regula de decizie Bayes. Pentru aceasta, să considerăm o problemă generală de clasificare în q categorii a unor obiecte/pattern-uri arbitrare, clasele de decizie fiind notate $\Omega_1, \Omega_2, \dots, \Omega_q$. Scopul utilizării PNN este acela de a determina apartenența unui obiect, reprezentat vectorial printr-un vector p -dimensional \mathbf{x} , la una din cele q clase $\Omega_1, \Omega_2, \dots, \Omega_q$, adică trebuie luată decizia $D(\mathbf{x}) := \Omega_i, i = 1, 2, \dots, q$. În acest caz, dacă se cunosc densitățile de probabilitate $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_q(\mathbf{x})$, corespunzătoare categoriilor $\Omega_1, \Omega_2, \dots, \Omega_q$, dacă se cunosc probabilitățile *a priori* $h_i = P(\Omega_i)$ de apartenență a unui pattern la categoria Ω_i și, în fine, dacă se cunosc parametrii *pierdere* (*loss*) sau *cost* (*cost*) l_i , asociați tuturor deciziilor incorecte, știind că $\Omega = \Omega_i$, atunci utilizând regula de decizie Bayes, vom clasifica pattern-ul \mathbf{x} în categoria Ω_i , dacă următoarea inegalitate are loc:

$$l_i h_i f_i(\mathbf{x}) > l_j h_j f_j(\mathbf{x}), i \neq j.$$

Astfel, frontierele între fiecare două clase de decizie Ω_i și Ω_j , $i \neq j$, sunt date de hiper-suprafețele:

$$l_i h_i f_i(\mathbf{x}) = l_j h_j f_j(\mathbf{x}), i \neq j,$$

iar acuratețea deciziei va depinde de acuratețea estimării densităților de probabilitate corespunzătoare claselor de decizie.

Așa cum am văzut mai sus, cheia utilizării regulii bayesiene de decizie în cadrul PNN este reprezentată de tehnica aleasă pentru estimarea densităților de probabilitate $f_i(\mathbf{x})$, corespunzătoare fiecărei categorii Ω_i , estimare bazată pe mulțimea de antrenament. Abordarea clasică în acest sens este aceea a utilizării unei sume de repartiții gaussiene multivariate, centrate în fiecare pattern de antrenament (vezi și RBF), adică:

$$f_{\Omega_i}(\mathbf{x}) = \frac{1}{(2\pi)^{p/2} \sigma^p} \cdot \frac{1}{m_i} \cdot \sum_{j=1}^{m_i} \exp\left(-\frac{d(\mathbf{x}, \mathbf{x}_j)^2}{2\sigma^2}\right), \quad i = 1, 2, \dots, q,$$

unde m_i este numărul de pattern-uri de antrenament din clasa Ω_i , \mathbf{x}_j reprezintă pattern-ul de antrenament cu numărul j din clasa Ω_i , p reprezintă, așa cum am mai spus, dimensiunea spațiului input-urilor, iar σ reprezintă singurul parametru ajustabil al rețelei, cunoscut sub numele de parametrul de *ajustare* sau *scalare* (*smoothing, scaling parameter*), obținut prin antrenarea rețelei. Parametrul σ (ca deviație standard) definește, așa după cum se observă cu ușurință, lărgimea ariei de influență a fiecărei decizii, și va descrește atunci când numărul pattern-urilor va crește. Putem privi, sugestiv, aceste densități ca

pe niște „clopote” – vezi *clopotul lui Gauss* - sub care se găsesc diferite tipuri de pattern-uri (fiecare cu clopotul său) și, cu cât acestea sunt mai multe, cu atât lărgimea clopotelor este mai mică, ca să poată avea loc toate. Așa după cum se observă, problema-cheie la PNN este modul de estimare a lui σ , deoarece de el depinde cât de mult se vor suprapune „clopotele” de clasificare, deci se vor suprapune două decizii distincte. Astfel, dacă σ este ales prea mare, atunci se vor pierde din vedere detaliile, iar dacă este ales prea mic, atunci „clopotele” vor fi prea „ascuțite” și astfel se va înrăutăți capacitatea de generalizare a rețelei. Uzual, parametrul σ se alege heuristic, domeniu de căutare fiind întreaga axă reală pozitivă \mathbf{R}_+ , frontiera de decizie variind continuu, de la o frontieră neliniară în cazul $\sigma \rightarrow 0$, la un hiperplan, în cazul $\sigma \rightarrow \infty$ (Specht 1990, "Probabilistic neural networks and the polynomial adaline as complementary techniques for classification," IEEE Trans. on Neural Networks, vol. 1, no. 1, pp. 111-121). Oricum, din fericire PNN nu sunt prea sensibile la variații rezonabile în ceea ce privește valoarea lui σ .

Remarcă: Înafara estimatorului de mai sus, în cadrul PNN se utilizează și estimatori alternativi pentru densitățile de probabilitate corespunzătoare claselor de decizii. Prezentăm, în continuare, câteva dintre cele mai cunoscute {Parzen, Cacoulous}, [69]:

- ♦ $f(x) = \frac{1}{n(2\lambda)^p} \sum_{j=1}^m 1$, atunci când $|x_i - x_{ij}| \leq \lambda$, $i = 1, 2, \dots, p$,
 $j = 1, 2, \dots, m$;
- ♦ $f(x) = \frac{1}{m\lambda^p} \sum_{j=1}^m \prod_{i=1}^p \left[1 - \frac{|x_i - x_{ij}|}{\lambda} \right]$, atunci când $|x_i - x_{ij}| \leq \lambda$,
 $i = 1, 2, \dots, p, j = 1, 2, \dots, m$;
- ♦ $f(x) = \frac{1}{n(2\pi)^{p/2} \lambda^p} \sum_{j=1}^m \prod_{i=1}^p e^{-\frac{1}{2} \frac{(x_i - x_{ij})^2}{\lambda^2}} =$
 $= \frac{1}{n(2\pi)^{p/2} \lambda^p} \sum_{j=1}^m \exp \left[\frac{-\sum_{i=1}^p (x_i - x_{ij})^2}{2\lambda^2} \right]$;
- ♦ $f(x) = \frac{1}{n(2\lambda)^p} \sum_{j=1}^m \prod_{i=1}^p e^{-|x_i - x_{ij}|/\lambda} =$

$$= \frac{1}{n(2\lambda)^p} \sum_{j=1}^m \exp \left[-\frac{1}{\lambda} \sum_{i=1}^p |x_i - x_{ij}| \right];$$

$$\blacklozenge \quad f(x) = \frac{1}{n(\pi\lambda)^p} \sum_{j=1}^m \prod_{i=1}^p \left[1 + \frac{(x_i - x_{ij})^2}{\lambda^2} \right]^{-1};$$

$$\blacklozenge \quad f(x) = \frac{1}{n(2\pi\lambda)^p} \sum_{j=1}^m \prod_{i=1}^p \left[\frac{\sin \frac{(x_i - x_{ij})}{2\lambda}}{\frac{|x_i - x_{ij}|}{2\lambda}} \right]^2;$$

$$\blacklozenge \quad f_{kn}(x) = \frac{1}{(2\pi)^{p/2} \sigma^p} \cdot \frac{1}{m} \cdot \sum_{j=1}^m \exp \left[k^n \cdot \left(-\frac{d(x, x_j)^2}{2\sigma^2} \right) \right],$$

$$k \geq 2, n \geq 1.$$

$$\blacklozenge \quad f_{Tr}(x) = \frac{1}{(2\pi)^{p/2} \sigma^p} \cdot \frac{1}{m} \cdot \sum_{j=1}^m \sum_{k=1}^r \frac{\left(-\frac{d(x, x_j)^2}{2\sigma^2} \right)^k}{k!}, \text{ for } r \geq 1.$$

PNN sunt private ca implementări ale unui algoritm statistic, cunoscut sub numele de *kernel discriminant analysis*, în care operațiile sunt organizate într-o rețea multi-strat de tip feedforward (*multilayered feedforward network*), conținând, per total, patru straturi:

- *Input layer*
- *Pattern layer*
- *Summation layer*
- *Output layer*

Arhitectura de bază a unei PNN, excluzând stratul inițial de input-uri, constă în noduri alocate în cele trei straturi de bază (Specht, 1988, 1990):

- ◆ *Pattern layer/unit*. Există un singur nod-pattern (*pattern node/pattern unit*) pentru fiecare obiect de antrenament. Fiecare nod-pattern formează un produs scalar $Z_i = \mathbf{x} \cdot \mathbf{W}_i^T$, între vectorul pattern input \mathbf{x} (pentru clasificare) și vectorul pondere \mathbf{W}_i , și apoi aplică o transformare neliniară asupra lui Z_i înainte de a trimite

nivelul său de activare nodului/unității de sumare. De remarcat că, spre deosebire de funcția de activare sigmoidă din cadrul MLP-ului (*back-propagation algorithm*), în acest caz transformarea neliniară utilizată este exponențială $\exp\left[(Z_i - 1)/\sigma^2\right]$. Presupunând că atât \mathbf{x} cât și \mathbf{W}_i sunt normalizați (i.e. norma unitate), acest lucru este echivalent cu transformarea $\exp\left[-(\mathbf{W}_i - \mathbf{x})(\mathbf{W}_i - \mathbf{x})^T/(2\sigma^2)\right]$.

- ◆ *Summation layer/unit.* Fiecare nod/unitate de însumare (*summation node/unit*) primește output-urile trimise de nodurile-pattern corespunzătoare unei anumite clase. În aceste noduri se însumează doar input-urile primite de la nodurile-pattern, ce corespund categoriei de care aparține pattern-ul de antrenament selectat, adică avem operația de sumare $\sum_i \exp\left[-(\mathbf{W}_i - \mathbf{x})(\mathbf{W}_i - \mathbf{x})^T/(2\sigma^2)\right]$.
- ◆ *Output (decision) layer/unit.* Nodurile/unitățile output ale rețelei sunt, de fapt, neuroni cu două input-uri, producând output-uri binare, relative la două categorii distincte Ω_r și Ω_s , $r \neq s$, $r, s = 1, 2, \dots, q$, prin utilizarea criteriului de clasificare:

$$\sum_i \exp\left[-(\mathbf{W}_i - \mathbf{x})(\mathbf{W}_i - \mathbf{x})^T/(2\sigma^2)\right] > \sum_j \exp\left[-(\mathbf{W}_j - \mathbf{x})(\mathbf{W}_j - \mathbf{x})^T/(2\sigma^2)\right]$$

Aceste noduri/unități au doar o singură pondere C , dată de parametrii de pierdere sau cost, probabilitățile apriorice de apartenență și numărul de pattern-uri de antrenament din fiecare

categorie. Concret, ponderea este dată de formula $C = -\frac{h_s l_s}{h_r l_r} \cdot \frac{n_r}{n_s}$.

Această pondere este determinată doar pe baza semnificației deciziei, astfel încât, în cazul în care nu sunt elemente care să dea o asemenea informație, se alege pur și simplu $C = -1$.

Tehnic vorbind, un algoritm (de antrenament) PNN poate avea următoarea formă [67].

PNN training algorithm

Input. Consider q classes of patterns/objects (p -dimensional vectors) $\Omega_1, \Omega_2, \dots, \Omega_q$. Each decision class Ω_i contains a number of m_i vectors (or training patterns), that is $\Omega_i = \{x_1, x_2, \dots, x_{m_i}\}$.

1) For each class Ω_i , $i = 1, 2, \dots, q$, compute the (Euclidian) distance between any pair of vectors and denote these distances by d_1, d_2, \dots, d_{r_i} , where

$$r_i = C_{m_i}^2 = \frac{m_i!}{2!(m_i-2)!}.$$

2) For each class Ω_i , $i = 1, 2, \dots, q$, compute the corresponding average

$$\text{distances and standard deviations } D_i = \frac{\sum_{j=1}^{r_i} d_j}{r_i}, \quad SD_i = \sqrt{\frac{\sum_{j=1}^{r_i} (d_j - D_i)^2}{r_i}}.$$

3) (*Searching process*) Compute the “smoothing” parameter searching domain D_σ , based on the 99,7% confidence interval, given by $D_\sigma = (0, 3 \times SD)$, where $SD = \max_i \{SD_i\}$.

4) For each decision class Ω_i , $i = 1, 2, \dots, q$, consider the decision functions

$$(\text{Parzen-Cacoulous classifier}) \quad f_i(x) = \frac{1}{(2\pi)^{p/2} \sigma^p} \cdot \frac{1}{m_i} \cdot \sum_{j=1}^{m_i} \exp\left(-\frac{d(\mathbf{x}, \mathbf{x}_j)^2}{2\sigma^2}\right).$$

5) (*Bayes decision rule*) In each decision class Ω_i (randomly) choose a certain vector \mathbf{x}_i^0 . Compare $f_i(\mathbf{x}_i^0)$ and $f_j(\mathbf{x}_i^0)$, for all $i \neq j$, following the algorithm:

“IF $l_i h_i f_i > l_j h_j f_j$ (for all $j \neq i$) THEN $\mathbf{x}_i^0 \in \Omega_i$ ELSE IF $l_i h_i f_i \leq l_j h_j f_j$ (for some $j \neq i$) THEN $\mathbf{x}_i^0 \notin \Omega_i$ ”

6) (*Measuring the classification accuracy*) For each (fixed) decision class Ω_i consider the 3-valued logic:

„TRUE –if $l_i h_i f_i > l_j h_j f_j$ (for all $j \neq i$), UNKNOWN –if $l_i h_i f_i = l_j h_j f_j$ (for some $j \neq i$) and FALSE –otherwise”.

Initially, each of the three variables is set to zero. Whenever a truth value is obtained, the corresponding variable is incremented with step size 1.

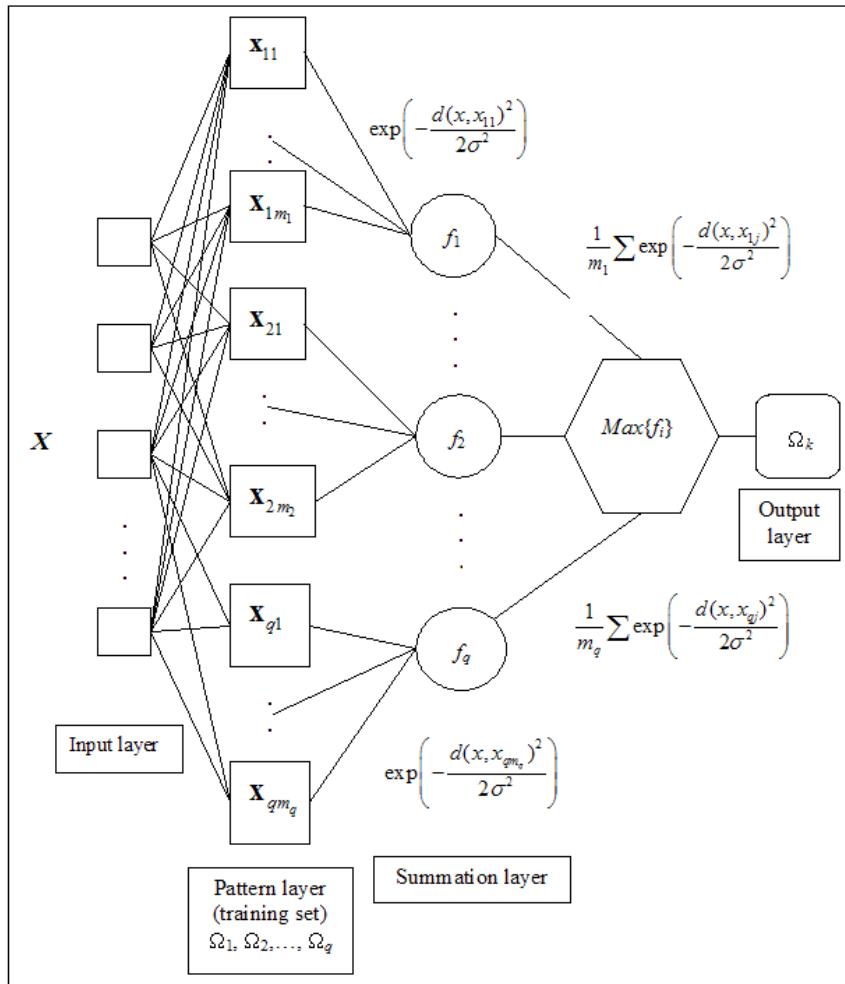
7) Repeat step 5 for another choice for \mathbf{x}_i^0 in Ω_i until all of them are chosen.
Increment counter.

8) Repeat step 5 for all vectors \mathbf{x}_j^0 in Ω_j , for all $j \neq i$.
Increment counter.

- 9) Obtain the classification accuracy in percentage (σ values are cached).
- 10) (*Estimating optimal smoothing parameter*) Choose a certain procedure to estimate the parameter σ .
- 11) If the current value of σ exceeds D_σ , then STOP.
Obtain the corresponding classification accuracy (σ values are cached).
- 12) Compute the maximum value MAX of the variable corresponding to the TRUE value.
- Output.** σ corresponding to MAX represents the optimal value of the „smoothing” parameter for each decision category $\Omega_i, i = 1, 2, \dots, q$.
- În ceea ce privește procedura de estimare a singurului parametru al rețelei și anume a lui σ , de la pasul 10 al algoritmului de antrenament, prezentăm în continuare trei variante, pe cât de simple, pe atât de eficiente [67], [68], [69], [76], [85], [133], împreună cu performanțele corespunzătoare, obținute în aplicații concrete.
- **Incremental approach.** Divide the searching domain D_σ by N dividing knots $\sigma_1, \sigma_2, \dots, \sigma_N$ into $(N + 1)$ equal sectors. Repeat step 5 by assigning σ the values $\sigma_k, k = 1, 2, \dots, N$.
 - **Genetic algorithm's approach.** Each chromosome is defined by the variable $X = (\sigma)$, the gene corresponding to the smoothing factor σ , taking its value from the value domain D_σ . A population of Y chromosomes is used. Selection is carried out by the Monte Carlo procedure. The average crossover $(X_1, X_2) \rightarrow \left(\frac{X_1 + X_2}{2} \right)$ is used to generate new chromosomes and for the mutation the following technique is applied: „assume we decide to mutate the gene σ of a chromosome. We will generate a random number, whose values are either 0 or 1. Then, the new value for the gene is determined by $\sigma \pm \delta$ (δ is a small enough value to fine tune the accuracy), “+” if 0 is generated, and “-” otherwise”.
Find the maximum of the cost function, counting the number of correct classifications.

- **Monte Carlo approach.** Generate in the searching domain D_σ a number of N random dividing points $\{P_1, \dots, P_N\}$, uniformly distributed in D_σ . Repeat step 5 by assigning $\sigma = P_k$, $k = 1, \dots, N$.

Arhitectura de bază a unei PNN, prezentată mai sus, este sugestiv ilustrată în figura următoare.



La finalul prezentării acestui foarte interesant domeniu privind rețelele neuronale artificiale, vom încerca să ilustrăm, prin câteva aplicații concrete, capabilitatea sa în ceea ce privește problema clasificării, problemă fundamentală de Data Mining.

Exemple:

1) Să considerăm setul de date corespunzător binecunoscutei plante *Iris*, set referitor la 50 plante din specia *Setosa*, 50 plante din specia *Versicolor* și 50 plante din specia *Virginica*. Am împărțit setul de date corespunzător celor 150 plante într-o mulțimea de antrenament de 80 instanțe, conținând 27 instanțe pentru *Setosa*, 26 instanțe pentru *Versicolor*, 27 instanțe pentru *Virginica*, și o mulțime de testare conținând 23 instanțe pentru *Setosa*, 24 instanțe pentru *Versicolor* și 23 instanțe pentru *Virginica*. Atributele luate în considerație în procesul de clasificare a florilor au fost dimensiunile petalelor și sepalelor (lungimea, respectiv lățimea).

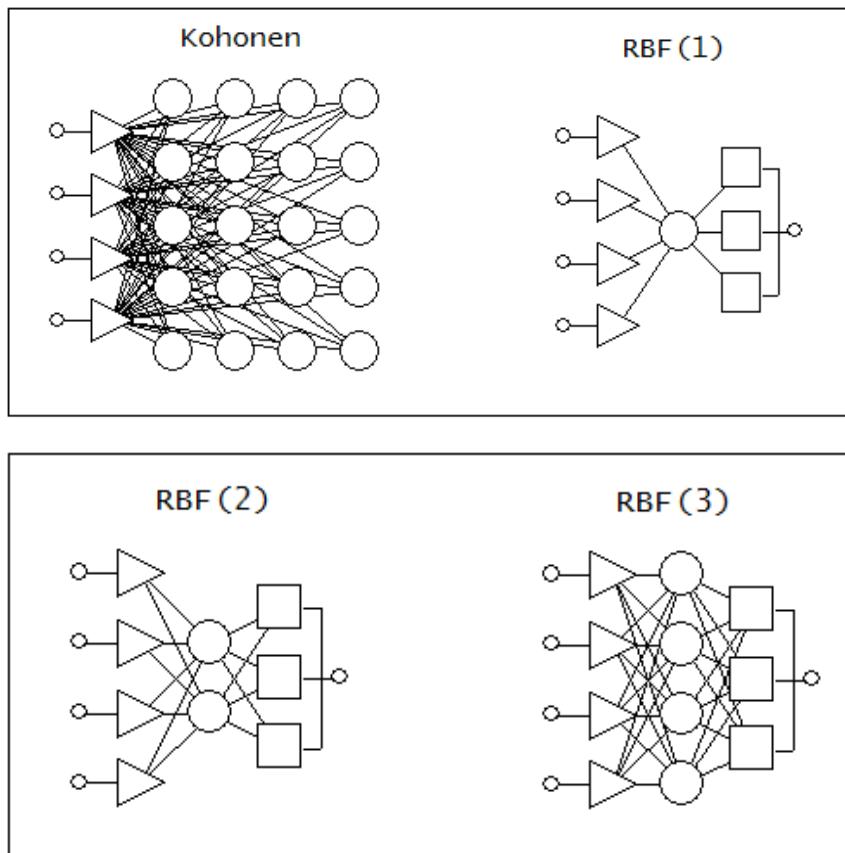
Prezentăm în tabelul de mai jos principalele rezultate obținute prin aplicarea diferitelor tipuri de rețele neuronale la clasificare florilor de *Iris*.

Tipul NN	Numărul straturilor/unităților ascunse	Acuratețea clasificării (testare)
Kohonen	5/-	91%
RBF (Radial Basis Function) -1	3/1	64%
RBF (Radial Basis Function) -2	3/2	91%
RBF (Radial Basis Function) -3	3/4	94%
RBF (Radial Basis Function) -4	3/10	95%
MLP (Multilayer Perceptron) -1	3/2	96%
MLP (Multilayer Perceptron) -2	3/4	96%
MLP (Multilayer Perceptron) -3	3/7	97%
MLP (Multilayer Perceptron) -4	3/8	97%
Model liniar	2/-	87%

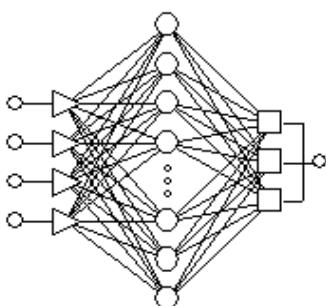
Tabelul de mai jos prezintă statistică de bază a clasificării plantei *Iris*.

	Setosa	Versicolor	Virginica	Setosa	Versicolor	Virginica
Total	27	26	27	23	24	23
Corect	27	26	27	23	23	22
Greșit	0	0	0	0	1	1
Necunoscut	0	0	0	0	0	0
Setosa	27	0	0	23	0	0
Versicolor	0	26	0	0	23	1
Virginica	0	0	27	0	1	22

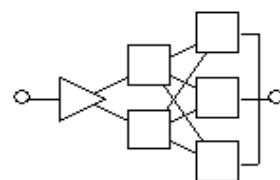
Pentru ilustrarea grafică a arhitecturii rețelelor folosite în aplicația de mai sus, prezentăm mai jos schema fiecăreia.



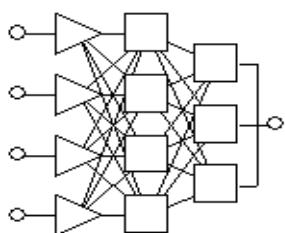
RBF (4)



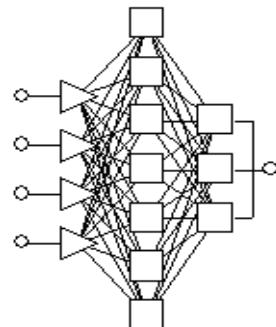
MLP (1)



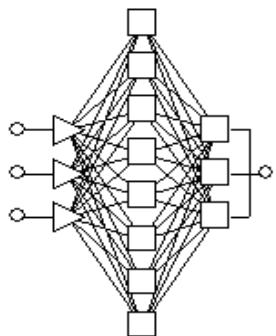
MLP (2)



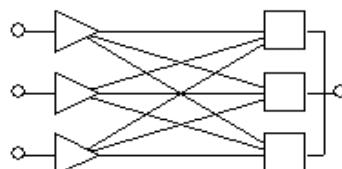
MLP (3)



MLP (4)



Liniar



2) Modelul PNN a fost aplicat pentru clasificarea/diagnosticul unui grup de 299 persoane în ceea ce privește patru categorii (clase de decizie) distincte, referitoare la diferite boli ale ficatului [68], [73], [76], [77], [85]:

- Hepatită cronică (CH);
- Ciroză hepatică (LC);
- Cancer hepatic (HCC)
- Lot control – indivizi sănătoși (HP)

Lotul de pacienții a provenit de la Spitalul Universitar de Urgență din Craiova și a constat din:

- 60 pacienți cu hepatită cronică;
- 179 pacienți cu ciroză hepatică;
- 30 pacienți cu cancer hepatic;
- 30 persoane sănătoase (lot control).

Fiecare individ din lot a fost reprezentat de un vector cu 15 componente, fiecare componentă reprezentând o anumită caracteristică semnificativă din perspectiva diagnosticului. Astfel, o anumită persoană este identificată cu o instanță cu 15 atribute: $x_1 = \text{TB}$ (total bilirubin), $x_2 = \text{DB}$ (direct bilirubin), $x_3 = \text{IB}$ (indirect bilirubin), $x_4 = \text{AP}$ (alkaline phosphatase), $x_5 = \text{GGT}$ (gamma glutamyl transpeptidase), $x_6 = \text{LAP}$ (leucine amino peptidase), $x_7 = \text{AST}$ (aspartate amino transferase), $x_8 = \text{ALT}$ (alanine amino transferase), $x_9 = \text{LDH}$ (lactic dehydrogenase), $x_{10} = \text{PI}$ (prothrombin index), $x_{11} = \text{GAMMA}$, $x_{12} = \text{ALBUMIN}$, $x_{13} = \text{GLYCEMIA}$, $x_{14} = \text{CHOLESTEROL}$, $x_{15} = \text{Vârsta}$.

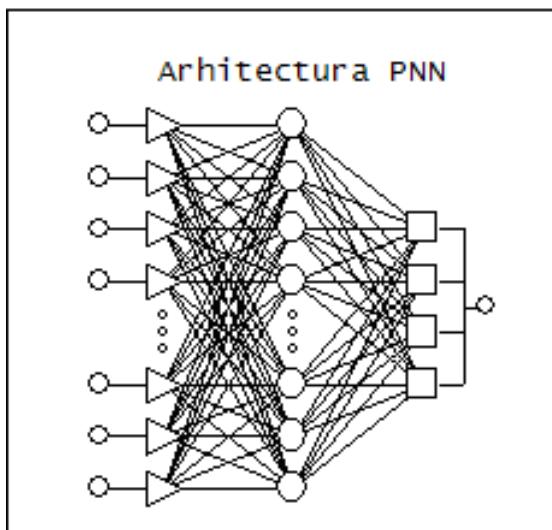
PNN a fost antrenată pe o mulțime de antrenament de 254 persoane (85% din total), iar pentru procesul de testare au fost păstrate restul de 45 persoane (15%).

Pentru antrenament, în vederea estimării parametrului σ , au fost utilizate cele trei metode indicate mai sus, și anume: căutarea incrementală, metoda Monte Carlo și căutarea cu ajutorul algoritmilor genetici. Rezultatele obținute au fost satisfăcătoare, acuratețea obținută în faza de antrenament, s-a situat în jurul valorii de 97%, în timp ce acuratețea la faza de testare a fost în jur de 95%. Evident că pot exista diferite valori pentru acuratețe (antrenament/validare), în funcție de metoda utilizată și parametrii aleși pentru fiecare metodă (e.g. numărul punctelor de căutare la metoda incrementală, numărul nodurilor la metoda Monte Carlo, volumul populației de cromozomi, numărul de generații și operatorii de variație aleși în cadrul utilizării algoritmilor genetici). În orice caz, s-a dovedit că PNN reprezintă clasificatorii eficienți în stabilirea diagnosticului în diferite ramuri medicale.

Algoritmul PNN (independent de alegerea metodologiei de căutare a parametrului σ) a fost codat în limbajul Java și a fost utilizată modalitatea JDBC (*Java Database Connectivity*) pentru procesarea datelor. Astfel,

programul este conectat la baza de date disponibilă, care poate fi oricând actualizată de către medici (în MS Access sau MS Excel), fără necesitatea modificării programului.

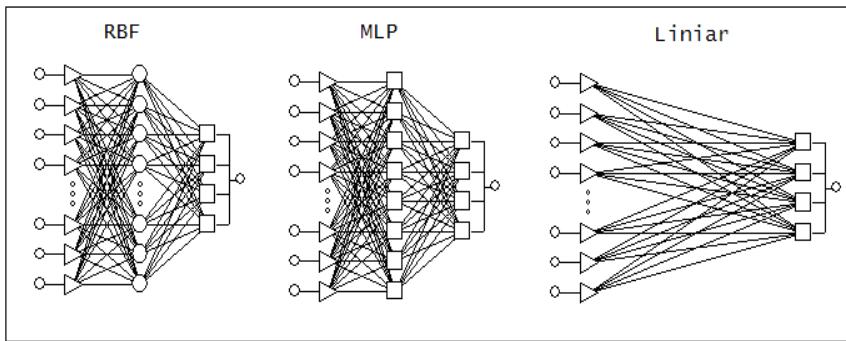
Arhitectura PNN utilizată la diagnostic este redată în figura de mai jos.



Pentru comparație, au fost utilizate pe aceeași bază de date și alte tipuri de NN. În tabelul de mai jos prezentăm performanțele astfel obținute (la testare), comparabile, cu excepția modelului liniar, cu cazul utilizării PNN.

Tipul NN	Numărul straturilor/unităților ascunse	Acuratețea
RBF (Radial Basis Function)	3/25	92%
MLP (Multilayer Perceptron)	3/8	94%
Model liniar	2/-	82%

În figurile de mai jos sunt ilustrate grafic arhitecturile NN utilizate alternativ la PNN.



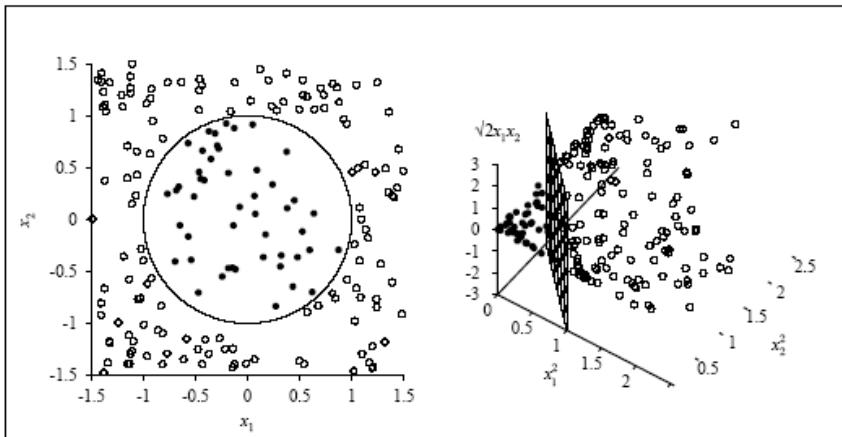
5.3.4. Mașini cu suport vectorial

O clasă specială de rețele universale de tip feedforward este reprezentată de așa-numitele *mașini cu suport vectorial* (*Support Vector Machines* -SVM), utilizate intens în probleme de clasificare formelor și de regresie neliniară. SVM-urile sunt bazate pe teoria învățării statistice (*statistical learning theory* -SLT), dintre promotorii căreia cităm: {Vapnik, Boser și Guyon -1992} [17], {Cortes și Vapnik -1995} [30], {Vapnik -1995, 1998} [169].

Conceptual, o SVM este o *mașină liniară* (*linear machine*), înzestrată cu caracteristici deosebite, având la bază principiile minimizării riscului structural (*Structural Risk Minimization* -SRM) și teoria învățării statistice. În consecință, o SVM poate produce o performanță bună a generalizării în cazul recunoașterii formelor, fără a încorpora cunoștințe despre domeniul problemei, ceea ce îi conferă o caracteristică unică printre modelele de învățare.

Din prezentarea anterioară a NN, reiese următoarea dilemă: „Perceptronul (*single-layer neural network*), în ciuda algoritmului simplu și eficient de învățare, are o putere limitată de clasificare, din cauza faptului că învăță doar frontiere de decizie liniare în spațiul input-urilor. Rețelele multi-strat (*multilayer networks*) au, în schimb, o putere mult mai mare de clasificare, dar, din nefericire, sunt dificil de antrenat din cauza multitudinii de minime locale și dimensiunii ridicate a spațiului ponderilor”. O soluție la această dilemă poate veni din partea SVM-urilor, sau, mai general, a *mașinilor cu nucleu* (*kernel machines*), care posedă algoritmi de antrenare eficienți și care pot reprezenta, în același timp, frontiere neliniare, complexe.

Fără a intra în detaliu, vom ilustra paradigma unei mașini cu nucleu prin următorul exemplu deosebit de simplu, prezentat în figura de mai jos.



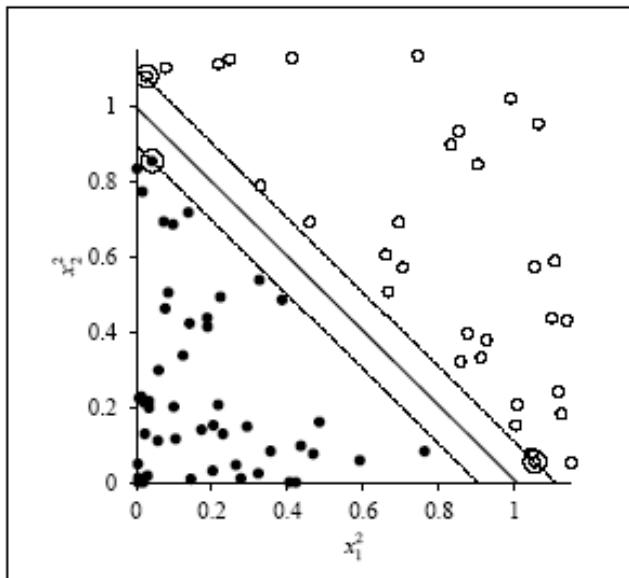
Astfel, în figura din stânga avem un spațiu input de dimensiune doi, în care un obiect este notat cu $\mathbf{x} = (x_1, x_2)$. Se observă că există două clase de obiecte:

- cele „*pozitive*” ($d = +1$), care se găsesc în interiorul cercului de ecuație $x_1^2 + x_2^2 = 1$;
- cele „*negative*” ($d = -1$), care se găsesc în afara cercului.

După cum se observă, nu există o separație liniară între cele două mulțimi de obiecte. Dacă vom aplica însă spațiul input (\mathbf{x}) de dimensiune doi în alt spațiu (\mathbf{z}), de dimensiune trei (superioară), printr-o aplicație $\mathbf{z} = g(\mathbf{x})$, dată de:

- ◆ $z_1 = g_1(x) = x_1^2$,
- ◆ $z_2 = g_2(x) = x_2^2$,
- ◆ $z_3 = g_3(x) = \sqrt{2}x_1x_2$,

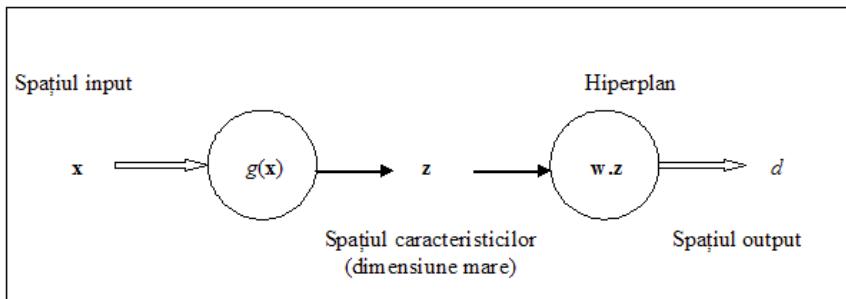
se observă, din figura din dreapta, că în noul spațiu tri-dimensional (\mathbf{z}), obiectele transformate sunt liniar separabile. Dacă proiectăm acest spațiu pe primele două axe, obținem imaginea de detaliu din figura de mai jos, în care separatorul este o dreaptă, și în care cele mai apropiate puncte – *vectorii suport* – sunt marcați cu cerculețe. *Marginea de separație* (*margin of separation*), care este lățimea benzii centrată pe separatorul liniar, introduce o „distanță” (separă) între obiectele „pozitive” și cele „negative”.



Generalizând, dacă spațiul inițial este aplicat într-un spațiu cu dimensiunea suficient de mare, atunci obiectele astfel transformate vor fi totdeauna liniar separabile.

Remarcă: În ceea ce privește ușurința cu care se poate găsi un separator liniar între puncte în spațiul (\mathbf{z}) , trebuie reamintit faptul că ecuația sa într-un spațiu n dimensional este definită de n parametri, deci va apărea fenomenul de *overfitting* în cazul în care n este aproximativ egal cu numărul N de puncte de clasificat. Din această cauză, mașinile cu nucleu caută, de obicei, separatorul liniar *optimal*, adică acela care are cea mai mare margine (distanță) între pattern-urile „pozitive” și cele „negative”. Se poate demonstra (*computational learning*) că acest separator are proprietăți convenabile în ceea ce privește generalizarea robustă la noi pattern-uri.

Schema generală a unei mașini cu nucleu este ilustrată în figura de mai jos.



Conceptual, funcționarea SVM-urilor (mașinilor cu nucleu, în general) se bazează pe următorii doi pași, ilustrați în figura de mai sus:

- ◆ Aplicarea (neliniară) a spațiului input într-un spațiu de dimensiune mare – *spațiu caracteristicilor (feature space)* – spațiu „ascuns” atât pentru input cât și pentru output;
- ◆ Construirea unui hiperplan de separație pentru caracteristicile obținute la pasul anterior.

Remarcă: Din punct de vedere teoretic, primul pas are la bază *teorema lui Cover privind separabilitatea pattern-urilor* [32], în timp ce al doilea se bazează pe principiile minimizării riscului structural.

Tehnic vorbind, construcția hiperplanului de separație se bazează pe evaluarea nucleului unui produs scalar. Astfel, fie \mathbf{x} un vector input arbitrar și $g(\mathbf{x}) = \{g_j(\mathbf{x}), j = 1, 2, \dots, m_1\}$, un set de transformări neliniare din spațiu input în spațiu caracteristicilor (m_1 = dimensiunea spațiului caracteristicilor). Datează fiind transformarea neliniară g , vom defini un hiperplan, privit în context ca o suprafață de decizie, prin ecuația:

$$\sum_{j=1}^{m_1} w_j \cdot g_j(\mathbf{x}) + b = 0,$$

unde $\mathbf{w} = \{w_j, j = 1, 2, \dots, m_1\}$ reprezintă un vector pondere, care leagă spațiul caracteristicilor de spațiu output, iar b este deplasarea. Notând $g_0(\mathbf{x}) = 1$ și $w_0 = b$, ecuația de mai sus devine:

$$\sum_{j=0}^{m_1} w_j \cdot g_j(\mathbf{x}) = 0.$$

Astfel, $g(\mathbf{x}) = (g_0(\mathbf{x}), g_1(\mathbf{x}), \dots, g_{m_1}(\mathbf{x}))$ reprezintă imaginea vectorului input în spațiu caracteristicilor, ponderată de $\mathbf{w} = (w_0, w_1, \dots, w_{m_1})$, iar ecuația:

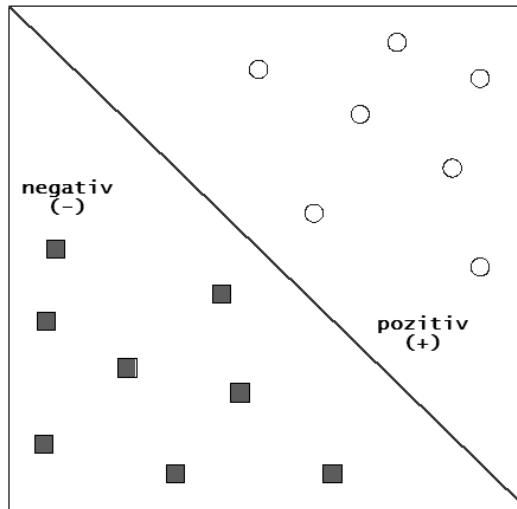
$$\mathbf{w} \cdot g^T(\mathbf{x}) = 0,$$

este asociată suprafeței de decizie din spațiul caracteristicilor. Notând un vector input de antrenament cu \mathbf{x}_i , $i = 1, 2, \dots, N$, se construiește nucleul produs-scalar (*inner-product kernel*):

$$K(\mathbf{x}, \mathbf{x}_i) = g(\mathbf{x}) \cdot g^T(\mathbf{x}) = \sum_{j=0}^{m_i} g_j(\mathbf{x}) \cdot g_j(\mathbf{x}_i), \quad i = 1, 2, \dots, N,$$

nucleu utilizat în găsirea hiperplanului optimal de separație, aplicând, de exemplu, optimizarea cuadratică (*quadratic optimization*) [96].

În continuare, vom prezenta succint principiile care stau la baza construcției și funcționării SVM-urilor. Astfel, să considerăm mulțimea de antrenament $T = \{\mathbf{x}_i, d_i; i = 1, 2, \dots, N\}$, în care \mathbf{x}_i reprezintă pattern-urile input, iar d_i output-urile corespunzătoare. Vom presupune, pentru început, că cele două clase $d_i = +1$ (pattern-urile „pozitive”) și $d_i = -1$ (pattern-urile „negative”) sunt liniar separabile.



Atunci ecuația hiperplanului de separație (decizie) va fi dată de:

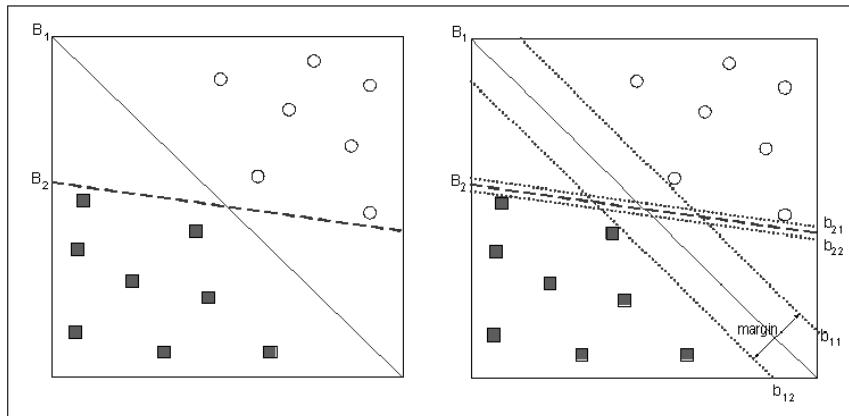
$$\mathbf{w} \cdot \mathbf{x}^T + b = 0,$$

unde \mathbf{x} este vectorul input, \mathbf{w} este vectorul pondere ajustabil, iar b deplasarea. Mai departe:

$$\mathbf{w} \cdot \mathbf{x}_i^T + b \geq 0, \quad \text{daca } d_i = +1,$$

$$\mathbf{w} \cdot \mathbf{x}_i^T + b < 0, \quad \text{daca } d_i = -1.$$

Pentru un vector pondere \mathbf{w} dat și deplasarea b cunoscută, separarea definită de hiperplan prin formulele de mai sus, între punctele cele mai apropiate din cele două regiuni, se numește *marginea* (separării). Scopul utilizării SVM-urilor este acela de a găsi acel hiperplan, pentru care marginea să fie maximă. În acest caz, suprafața de decizie se numește *hiperplan optimal*. În figura de mai jos, este ilustrată sugestiv această problemă.



Așa cum se observă din figură, hiperplanul optimal este B_1 .

Fie \mathbf{w}_0 și b_0 valorile corespunzătoare hiperplanului optimal, de ecuație:

$$\mathbf{w}_0 \cdot \mathbf{x}^T + b_0 = 0.$$

Atunci, *funcția discriminant* de ecuație:

$$g(\mathbf{x}) = \mathbf{w}_0 \cdot \mathbf{x}^T + b_0,$$

dă o măsură (analitică) a distanței de la punctul \mathbf{x} la hiperplanul optimal [37], [38]. Dacă notăm cu r distanța dorită, atunci:

$$g(\mathbf{x}) = \mathbf{w}_0 \cdot \mathbf{x}^T + b_0 = r \|\mathbf{w}_0\|,$$

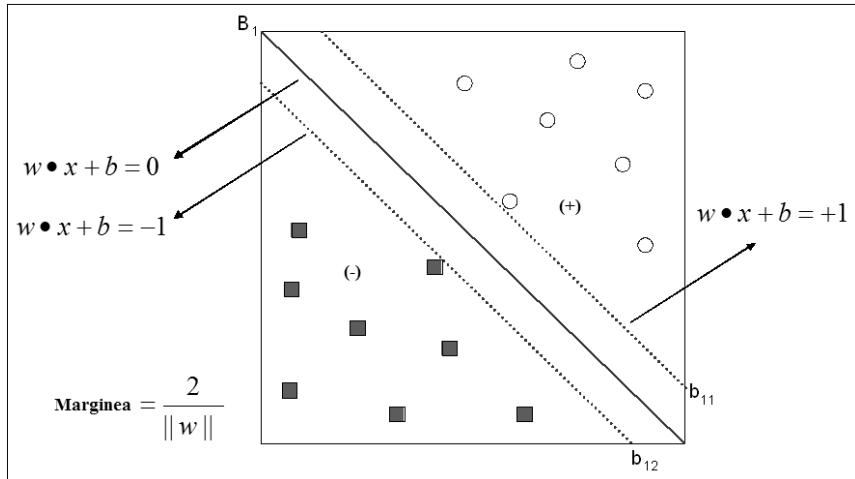
sau:

$$r = \frac{g(\mathbf{x})}{\|\mathbf{w}_0\|}.$$

Problema care trebuie rezolvată în acest context este aceea a estimării parametrilor \mathbf{w}_0 și b_0 , corespunzători hiperplanului optimal, pe baza mulțimii de antrenament T . Se observă că (vezi și figura de mai jos), perechea pondere/deplasare (\mathbf{w}_0, b_0) satisface condițiile:

$$\mathbf{w}_0 \cdot \mathbf{x}_i^\tau + b_0 \geq 1, \quad \text{daca } d_i = +1,$$

$$\mathbf{w}_0 \cdot \mathbf{x}_i^\tau + b_0 \leq -1, \quad \text{daca } d_i = -1.$$



Punctele particulare (\mathbf{x}_i, d_i) care se găsesc pe „dreptele” din figura de mai sus ce delimită marginea (i.e. satisfac cele două ecuații corespunzătoare), se numesc vectori suport (*support vectors*), de unde și numele modelului „mașini cu suport vectorial”. Acești vectori se găsesc, prin urmare, pe frontierele de decizie, separând cele două categorii, ei fiind cel mai dificil de clasificat. Cu alte cuvinte, pe ei se bazează modul concret cum separăm cele două categorii (suportul „vectorial” al actului de decizie).

Așa după cum ușor se poate calcula, valoarea marginii de separație este $\rho = \frac{2}{\|\mathbf{w}\|}$ și deci problema este de a găsi vectorul pondere \mathbf{w}_0 care să maximizeze sau, echivalent, care să minimizeze funcția cost, dată de:

$$L(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2},$$

având constrângerea:

$$f(\mathbf{x}_i) = \begin{cases} 1, & \text{daca } \mathbf{w} \cdot \mathbf{x}_i^\tau + b \geq 1 \\ -1, & \text{daca } \mathbf{w} \cdot \mathbf{x}_i^\tau + b \leq -1 \end{cases},$$

ceea ce se poate realiza utilizând, de exemplu, optimizarea cuadratică [96].

Remarcă: Am arătat la început că SVM-urile sunt privite ca modele provenind din principiile minimizării riscului structural și cele din teoria învățării statistice. Din teoria învățării statistice știm că *dimensiunea* VC (Vapnik & Chervonenkis) a unei mașini de învățat determină modul în care o structură în serie (*nested structure*) de funcții de aproximare poate fi utilizată. De asemenea, se mai știe că dimensiunea VC a unui set de hiperplane de separație într-un spațiu m dimensional este $m + 1$. Deci, pentru a aplica metoda minimizării riscului structural, este nevoie să se construiască un set de hiperplane de separație de dimensiune VC variabilă, astfel încât riscul empiric (i.e. eroarea de clasificare în faza de antrenament) și dimensiunea VC să fie minime simultan. La baza acestui proces din cadrul SVM-urilor stă următoarea teoremă.

Teoremă (Vapnik -1995, 1998) [169]. Fie D diametrul celei mai mici bile conținând toți vectorii input $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$. Mulțimea hiperplanelor descrise de ecuația:

$$\mathbf{w}_0 \cdot \mathbf{x}^\tau + b_0 = 0,$$

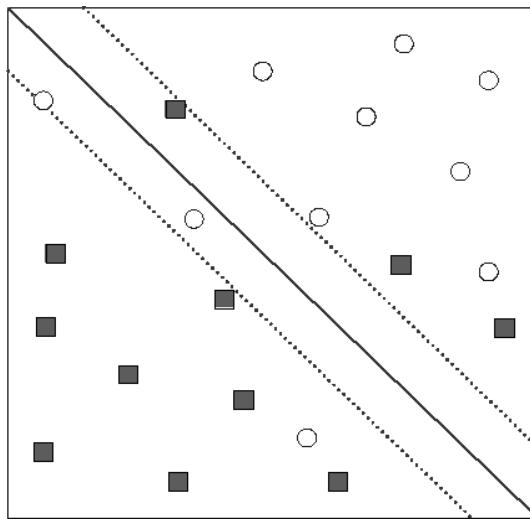
are dimensiunea VC, notată h , mărginită superior:

$$h \leq \min \left\{ \left[\frac{D^2}{\rho_0^2} \right], m_0 \right\} + 1,$$

unde $[.]$ reprezintă funcția „parte întreagă”, $\rho_0 = \frac{2}{\|\mathbf{w}_0\|}$, iar m_0 reprezintă dimensiunea spațiului input. Pentru detalii, a se vedea și [96], [184].

Remarcă: Din teorema de mai sus rezultă că este permis controlul asupra dimensiunii VC (i.e. complexității) a hiperplanului optimal, independent de dimensiunea m_0 a spațiului input, prin modul de alegere a marginii de separație.

Am vorbit până acum de cazul simplu al separabilității liniare a pattern-urile din clasele de decizie. Ce se întâmplă însă în cazul general, când această situație nu mai este posibilă, după cum se vede în figura de mai jos?



Fără a intra în detaliu tehnice, menționăm numai faptul că în acest caz este luat în considerare un set de variabile suplimentare $\{\xi_i, i = 1, 2, \dots, N\}$, numite *variabile de relaxare (slack variables)*, variabile ce măsoară deviația unui punct față de condiția ideală de separabilitate a pattern-urilor. În acest caz se pune problema minimizării funcției cost, dată de:

$$L(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^N \xi_i,$$

cu constrângerile:

$$f(\mathbf{x}_i) = \begin{cases} 1, & \text{daca } \mathbf{w} \cdot \mathbf{x}_i^\tau + b \geq 1 - \xi_i \\ -1, & \text{daca } \mathbf{w} \cdot \mathbf{x}_i^\tau + b \leq -1 + \xi_i \end{cases},$$

unde C se numește parametrul de *regularizare* și este selectat de către utilizator în următoarele două moduri:

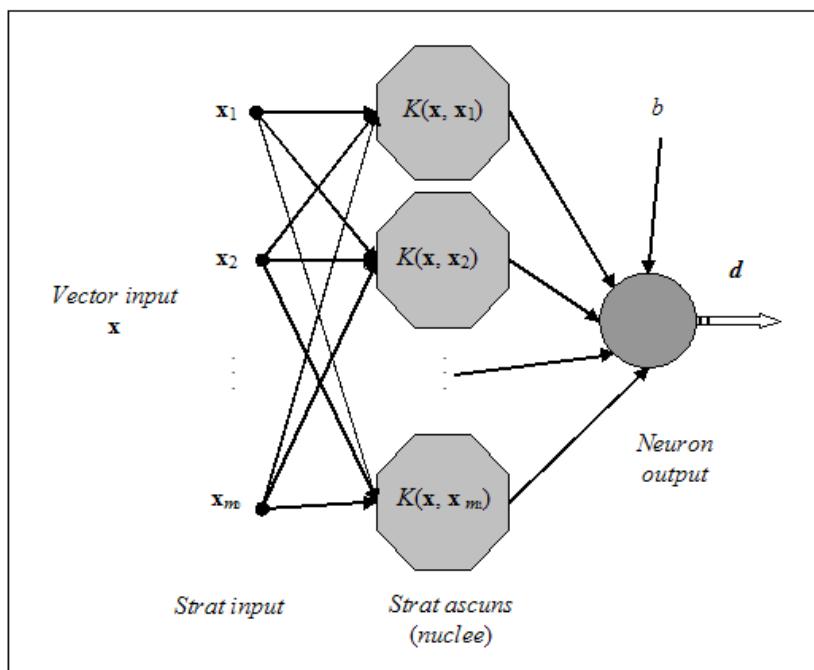
- Determinare experimentală, pe baza antrenamentului SVM;
- Determinare analitică, pe baza estimării dimensiunii VC.

(vezi [96], [162], [184]).

În final, vom prezenta schematic trei tipuri de SVM-uri, plecând de la tipurile de nuclee considerate, [96], [184]:

- *Mașină polinomială de învățare (polynomial learning machine)*, având nucleul de forma $K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x} \cdot \mathbf{x}_i^\tau + 1)^p$, unde parametrul p este specificat *a priori* de către utilizator;
- *Rețea cu funcție radial-basis –RBF (radial-basis function network)* având nucleul (gaussian) de forma $K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}_i\|^2\right)$, unde dispersia σ^2 , comună tuturor nucleelor, este specificată *a priori* de către utilizator (vezi și subparagraful 5.3.2.);
- *MLP cu un singur strat ascuns (one hidden-layer MLP)*, cunoscut și ca rețea de tip *perceptron cu două straturi*, având nucleul (sigmoid) de forma $K(\mathbf{x}, \mathbf{x}_i) = \tanh(\beta_0 \cdot \mathbf{x} \cdot \mathbf{x}_i^\tau + \beta_1)$, de parametri β_0 și β_1 , ce trebuie aleși cu grijă pentru a găsi soluția optimă (*idem*).

Cu toate că SVM-urile reprezintă un tip de mașini de învățare aparte, le-am considerat în cadrul domeniului NN, deoarece arhitectura lor urmează această „filosofie”, aşa după cum se poate vedea din schema de mai jos.



Exemplu:

Să considerăm o problemă de tip XOR (*exclusive OR problem*), care înseamnă în principiu situația în care o decizie este bazată pe una și numai una din două condiții care trebuie satisfăcute (*disjuncție exclusivă*). De exemplu, dacă nu ne place aglomerația, atunci vom decide să mergem la munte, dacă este soare, sau dacă este sărbătoare publică (zi liberă), dar nu simultan. Să reamintim aici că astfel de probleme nu pot fi rezolvate cu ajutorul unui singur perceptron. Problema XOR concretă, de care vorbim în acest exemplu, este redată în tabelul de mai jos [25], [96]:

Vector input \mathbf{x}	Output dorit d
(-1, -1)	-1
(-1, +1)	+1
(+1, -1)	+1
(+1, +1)	-1

Vom considera notația $\mathbf{x} = (x_1, x_2)$ și $\mathbf{x}_i = (x_{i_1}, x_{i_2})$, cu nucleul K având expresia:

$$K(\mathbf{x}, \mathbf{x}_i) = (1 + \mathbf{x} \cdot \mathbf{x}_i^\tau)^2.$$

Atunci:

$$K(\mathbf{x}, \mathbf{x}_i) = 1 + x_1^2 x_{i_1}^2 + 2x_1 x_2 x_{i_1} x_{i_2} + x_2^2 x_{i_2}^2 + 2x_1 x_{i_1} + 2x_2 x_{i_2}.$$

Vectorul imagine \mathbf{z} , aparținând spațiului caracteristicilor, obținut prin transformarea neliniară de mai sus, are forma:

$$\mathbf{z} = g(\mathbf{x}) = (1, x_1^2, \sqrt{2}x_1 x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2).$$

Similar, obținem că:

$$\mathbf{z}_i = g(\mathbf{x}_i) = (1, x_{i_1}^2, \sqrt{2}x_{i_1} x_{i_2}, x_{i_2}^2, \sqrt{2}x_{i_1}, \sqrt{2}x_{i_2}).$$

Dacă notăm cu K matricea $K(\mathbf{x}_i, \mathbf{x}_j), 1 \leq i, j \leq N$ de tip $N \times N$, atunci avem:

$$K = \begin{pmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{pmatrix}.$$

Utilizând optimizarea cuadratică – multiplicatorii Lagrange - (vezi [96]), se obține vectorul pondere optimal:

$$\mathbf{w}_0 = (0, 0, -1/\sqrt{2}, 0, 0, 0),$$

astfel încât hiperplanul optimal va fi definit de ecuația:

$$\mathbf{w}_0 \cdot g^T(\mathbf{x}) = (0, 0, -1/\sqrt{2}, 0, 0, 0) \cdot (1, x_{i_1}^2, \sqrt{2}x_{i_1}x_{i_2}, x_{i_2}^2, \sqrt{2}x_{i_1}, \sqrt{2}x_{i_2})^T = 0,$$

care se reduce la ecuația $x_1 \cdot x_2 = 0$.

Remarcă: 1. Așa cum am amintit anterior, SVM-urile sunt utilizate cu succes și în regresia neliniară, sub numele de SVR (*support vector regression*- Vapnik, Golowich și Smola, 1996) [168]. Prezentăm aici doar fundamentele acestei aplicații. Astfel, fie dată o problemă de regresie (neliniară), descrisă de ecuația de regresie:

$$d = f(\mathbf{x}) + \nu,$$

unde d este variabila dependentă, \mathbf{x} este vectorul predictor, f este funcția (neliniară) de legătură între variabila dependentă (output) și predictor (input), iar ν reprezintă „zgomotul”. Atunci, pe baza unui set de antrenament $T = \{\mathbf{x}_i, d_i; i = 1, 2, \dots, N\}$, trebuie estimată dependența între variabila dependentă și predictor (adică determinată funcția f și repartitia zgomotului ν). Notând cu y estimatorul lui d și cu $g(\mathbf{x}) = \{g_j(\mathbf{x}), j = 0, 1, 2, \dots, m_1\}$ setul de transformări neliniare de bază, să considerăm dezvoltarea:

$$y = \sum_{j=0}^{m_1} \mathbf{w}_j \cdot g_j(\mathbf{x}) = \mathbf{w} \cdot g^T(\mathbf{x}).$$

Considerând așa-numita funcție ε -nesensibilă de pierdere (ε -insensitive loss function) –Vapnik (1995, 1998), dată de:

$$L_\varepsilon(d, y) = \begin{cases} |d - y|, & |d - y| \geq \varepsilon \\ 0, & |d - y| < \varepsilon \end{cases},$$

rămâne de minimizat riscul (empiric):

$$R = \frac{1}{N} \sum_{i=1}^N L_\varepsilon(d_i, y_i),$$

cu constrângerea $\|\mathbf{w}\|^2 \leq c_0$, unde c_0 reprezintă o constantă. Considerând două seturi de variabile de relaxare $\{\xi_i, i = 1, 2, \dots, N\}$ și $\{\xi'_i, i = 1, 2, \dots, N\}$, rezolvarea problemei constă în minimizarea funcției cost:

$$L(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^N (\xi_i + \xi'_i), \text{ cu constrângerile:}$$

$$d_i - \mathbf{w} \cdot g^T(\mathbf{x}_i) \leq \varepsilon + \xi_i, \quad i = 1, 2, \dots, N,$$

$$\mathbf{w} \cdot g^T(\mathbf{x}_i) - d_i \leq \varepsilon + \xi'_i, \quad i = 1, 2, \dots, N,$$

$$\xi_i \geq 0, \quad i = 1, 2, \dots, N,$$

$$\xi'_i \geq 0, \quad i = 1, 2, \dots, N.$$

(pentru detalii tehnice, vezi [96]).

2. Componenta de optimizare standard din cadrul SVM, prin multiplicatorii Lagrange, poate părea destul de complicată în ceea ce privește înțelegerea deplină a calculelor și implementarea corectă a mecanismelor. În acest sens s-a concretizat o nouă abordare (vezi [152], [154]), numită *mașini evolutive cu suport vectorial* -ESVM, care oferă o alternativă simplă tehnicii standard de optimizare din cadrul SVM, adusă de algoritmii evolutivi (vezi paragraful 5.8). ESVM moștenesc punctul de vedere al tehnicii clasice în ceea ce privește învățarea, dar estimează parametrii funcției de decizie printr-un algoritm evolutiv canonic. Astfel, parametrii care trebuie estimați sunt fie ai hiperplanului de separație, în cazul clasificării, fie ai funcției de legătură între variabila dependentă și predictor, în cazul regresiei. ESVM pot determina întotdeauna parametrii funcției de învățare, ceea ce este adesea imposibil în cadrul tehnicii clasice. Mai mult, acestea obțin coeficienții direct din algoritmul evolutiv și pot face referință la ei în orice moment din cadrul unei rulări. ESVM au fost validate pe multe probleme de clasificare reale (e.g. detecția spam-ului [158], diagnoza diabetului mellitus [159], clasificarea plantelor iris [153], diagnoza bolilor la soia [157]), pe o problemă de regresie (e.g. problema imobiliarelor din Boston [155]), precum și pentru clasificarea unui volum mare de date [156]), rezultatele obținute demonstrând funcționalitatea acestei tehnici.

Încheiem scurta expunere privind SVM-urile, indicând câteva link-uri utile privind implementarea acestora (cf. Wikipedia):

- * [SVMlight](#) (implementarea algoritmului SVM de către Thorsten J. - Cornell University- pentru utilizarea în probleme de clasificare, regresie și rang);
- * [YALE](#) (machine learning toolbox, dezvoltată de *Artificial Intelligence Unit of the University of Dortmund* și conținând wrappers pentru SVMLight, LibSVM și MySVM);

- * LS-SVMLab (Matlab/C SVM toolbox).

5.4. Clasificare bazată pe reguli de asociere

O regulă de asociere (*association rule*) este o expresie de implicare de felul $X \rightarrow Y$, unde X și Y sunt articole (item-uri) distincte (i.e., $X \cap Y = \emptyset$), X fiind *antecedentul* regulii iar Y fiind *consecința* regulii. Ea este o tehnică nesupervizată DM, care caută legături între înregistrările dintr-un set de date (cunoscut îndeobște în domeniul business-ului ca *market basket transactions*). Metoda regulii de asociere (sau analiza asocierii) este câteodată definită ca *analiza coșului de consum*, care este de altfel și cea mai folosită aplicație a sa. Scopul este de a descoperi, de exemplu, ce articole sunt cel mai des cumpărate în același timp (coșul de consum), pentru a ajuta comercianții cu amănuntul să organizeze programe de stimulare a clienților și să-și aranjeze magazinele cât mai eficient. Ca exemplu clasic, reamintim regula denumită generic **<scutece – bere>**, în care o asemenea asociere este:

$$\{X \rightarrow Y\} \Leftrightarrow \{\text{scutece, lapte}\} \rightarrow \{\text{bere}\}.$$

Robustețea unei reguli de asociere se măsoară prin trei caracteristici principale: *suportul* (*support*), *încrederea* (*confidence*) și *diferența de nivel* (*lift*). Astfel, suportul măsoară (procentual) cât de des se poate aplica regula la o mulțime de date, adică cât de des apar anumite articole împreună în totalul tranzacțiilor. Încrederea măsoară cât de des apar articole Y în asociere (tranzacții) ce conțin pe X sau, mai simplu, cât de mult depinde un articol de altul. Diferența de nivel măsoară raportul dintre încrederea într-o regulă (e.g. privind cumpărăturile) și încrederea așteptată (ca al doilea produs să fie și el cumpărat), putând fi considerată ca o măsură a intensității unui efect.

Vom exemplifica cele de mai sus cu următorul caz din domeniul comercial. Să considerăm următorul exemplu privind comercializarea următoarelor două băuturi alcoolice: berea și băuturile cu grad înalt de alcoolizare („tăriile”) în 500.000 tranzacții:

- 20.000 tranzacții conțin tărie (4% din totalul tranzacțiilor);
- 30.000 tranzacții conțin bere (6% din totalul tranzacțiilor);
- 10.000 tranzacții conțin și bere și tărie (2% din totalul tranzacțiilor).

Atunci, avem:

- ♦ *Suportul* este 2% din total ($10.000/500.000$);
- ♦ Pentru *încredere* avem următoarele situații:
 - Regula „*Când oamenii cumpără tărie, cumpără de asemenea și bere*” are încrederea 50% ($10.000/20.000$);

- Regula „*Când oamenii cumpără bere, cumpără de asemenea și tărie*” are încrederea 33% (10.000/30.000).

Să notăm că cele două reguli au același suport (2%).

Dacă nu mai există informații suplimentare despre alte tranzacții, putem face următoarea afirmație plecând de la datele disponibile:

- *Clienții cumpără tărie 4% din timp.*
- *Clienții cumpără bere 6% din timp.*

Cele două procentaje, 4% și 6%, sunt numite *încrederea așteptată* de a cumpăra tărie sau bere, indiferent de celealte cumpărături.

Deoarece încrederea în regula de cumpărare *tărie-bere* este 50%, în timp ce încrederea așteptată pentru cumpărarea de *bere* este 6%, rezultă că *diferența de nivel* oferită de regula de cumpărare *tărie-bere* este 8,33 (50% / 6%).

Regula de cumpărare *tărie-bere* poate fi exprimată în termen de *diferență de nivel* prin aserțiunea: „*Clienții care cumpără tărie sunt de 8,33 ori mai tentați să cumpere și bere odată cu tăria*”. Interacțiunea dintre *„tărie”* și *bere* este deci foarte puternică.

Putem defini procesul de descoperire a regulilor de asociere în felul următor: „*Fiind dată o mulțime de tranzacții, să se descopere toate regulile posibile pentru care atât suportul cât și încrederea să fie mai mari sau egale decât anumite praguri prestabilite*”. Modul în care se găsesc regulile respective depinde de procedura aleasă. În acest sens, se poate arăta că numărul total R al regulilor care pot fi extrase dintr-un set de date, care conține un număr de d articole, este dat de formula:

$$R = 3^d - 2^{d+1} + 1,$$

deci există un volum suficient de mare de lucru ce trebuie efectuat în acest proces.

Vom prezenta, în continuare, un exemplu simplu de descoperire de reguli, plecând de la un set restrâns de articole (adaptare [162]), prezentate în tabelul de mai jos.

Denumire	Sânge	Naștere	Zboară	Mediu=apa	Clasa
om	cald	da	nu	nu	mamifer
cobra	rece	nu	nu	nu	reptilă
crap	rece	nu	nu	da	pește
balenă	cald	da	nu	da	mamifer
broască	rece	nu	nu	uneori	amfibian
iguana	rece	nu	nu	nu	reptilă
liliac	cald	da	da	nu	mamifer
porumbel	cald	nu	da	nu	pasăre
pisică	cald	da	nu	nu	mamifer
rechin	rece	da	nu	da	pește
țestoasă	rece	nu	nu	uneori	reptilă
pinguin	cald	nu	nu	uneori	pasăre
porc	cald	da	nu	nu	mamifer
țipar	rece	nu	nu	da	pește
salamandra	rece	nu	nu	uneori	amfibian
cameleon	rece	nu	nu	nu	reptilă
ornitorinc	cald	nu	nu	nu	mamifer
papagal	cald	nu	da	nu	pasăre
delfin	cald	da	nu	da	mamifer
vultur	cald	nu	da	nu	pasăre

Utilizând datele de mai sus, putem extrage următoarele reguli de asociere:

- ◆ R_1 : ($Naștere = nu$) \wedge ($Zboară = da$) \rightarrow Pasăre
- ◆ R_2 : ($Naștere = nu$) \wedge („ $Mediu = apă$ ” = da) \rightarrow Pește
- ◆ R_3 : ($Naștere = da$) \wedge ($Sânge = cald$) \rightarrow Mamifer
- ◆ R_4 : ($Naștere = nu$) \wedge ($Zboară = nu$) \rightarrow Reptilă
- ◆ R_5 : („ $Mediu = apă$ ” = uneori) \rightarrow Amfibian

Spunem că o regulă R acoperă un articol (obiect/instanță) x din setul de date, dacă toate atributele acestuia satisfac condiția regulii. Astfel, plecând de la datele tabelate mai jos, se observă că regula R_1 acoperă articolul „șoim”, iar regula R_3 acoperă articolul „urs brun”.

Denumire	Sânge	Naștere	Zboară	Mediu=apă	Clasa
șoim	cald	nu	da	nu	?
urs brun	cald	da	nu	nu	?

Vom mai aminti, în context, alte două caracteristici ale regulilor de asociere:

- Puterea de acoperire a regulii, definită ca procentajul de articole care satisfac antecedentul regulii;
- Accuratețea regulii, definită ca procentajul de articole care satisfac atât antecedentul cât și consecința regulii.

Pentru a vedea cum se utilizează cele două caracteristici definite mai sus, să reluăm exemplul relativ la investigarea posibilității de fraudă la rambursarea unor sume, având mulțimea de antrenament tabelată mai jos.

No.	Rambursare	Statut marital	Venit anual	Excrocare
1	Da	Necasatorit	125.000	NU
2	Nu	Casatorit	100.000	NU
3	Nu	Necasatorit	70.000	NU
4	Da	Casatorit	120.000	NU
5	Nu	Divorțat	95.000	DA
6	Nu	Casatorit	60.000	NU
7	Da	Divorțat	220.000	NU
8	Nu	Necasatorit	85.000	DA
9	Nu	Casatorit	75.000	NU
10	Nu	Necasatorit	90.000	DA

Astfel, în cazul regulii:

$$\{Statut\ marital = Necăsătorit\} \rightarrow NU,$$

acoperirea este 40% (4 din 10 cazuri), iar acuratețea este 50% (2 din 4).

Pentru a vedea cum se utilizează o regulă de asociere, să revenim la exemplul inițial, privind clasificarea diferitelor tipuri de viețuitoare. Astfel, dacă se prezintă noi articole, la fel ca în tabelul de mai jos,

Denumire	Sânge	Naștere	Zboară	Mediu=apă	Clasă
câine	cald	da	nu	nu	?
anaconda	rece	nu	nu	uneori	?
guppy	rece	da	nu	da	?

putem lua următoarele decizii pe baza regulilor de asociere:

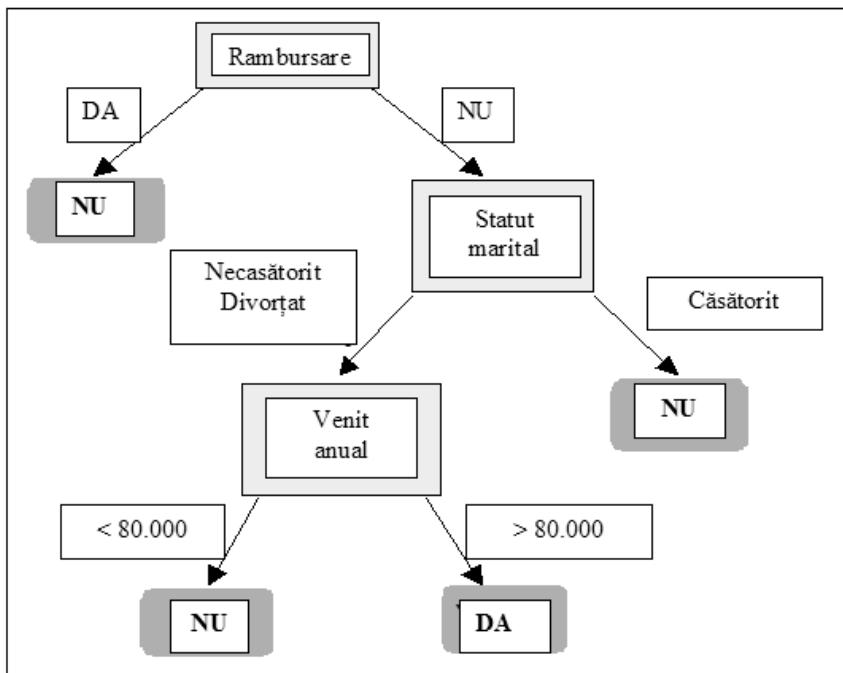
- ◆ *Câinele* se supune regulii R_3 : „(*Naștere* = da) \wedge (*Sânge* = cald) \rightarrow Mamifer”, ceea ce este absolut corect;
- ◆ *Anaconda* se supune regulilor R_4 : „(*Naștere* = nu) \wedge (*Zboară* = nu) \rightarrow Reptilă” și R_5 : „(*Mediu = apă*” = uneori) \rightarrow Amfibian”, ceea ce este parțial corect, ea fiind șarpe;

- ◆ *Guppy* (un simpatic și foarte popular pește de acvariu) nu se supune niciuneia dintre cele 5 reguli definite anterior, deci nu poate fi astfel clasificat.

Putem menționa două tipuri importante de reguli de asociere:

- *Reguli (reciproci) exclusive*, caz în care regulile sunt independente una de alta și fiecare articol este acoperit de cel mult o regulă;
- *Reguli exhaustive*, caz în care se consideră toate combinațiile posibile ale valorilor atributelor și fiecare articol este acoperit de cel puțin o regulă.

Pentru a exemplifica cele două tipuri de reguli menționate mai sus, considerăm metoda extragerii regulilor de asociere pe baza arborilor de decizie, prezentată în subparagraful 4.3.4. Astfel, fiind dat arborele de clasificare și decizie, referitor la investigarea posibilității de fraudă la rambursarea unor sume,



se observă că următoarele reguli de asociere:

- $R_1: (Rambursare = DA) \rightarrow \text{NU};$
- $R_2: (Rambursare = NU) \wedge (\text{Statut marital} = \text{Necăsătorit, Divorțat}) \wedge$

$\wedge (Venit\ anual < 80.000) \rightarrow \mathbf{NU};$

- $R_3: (Rambursare = \mathbf{NU}) \wedge (Statut\ marital = \text{Necăsătorit, Divorțat}) \wedge (Venit\ anual > 80.000) \rightarrow \mathbf{DA};$
- $R_4: (Rambursare = \mathbf{NU}) \wedge (Statut\ marital = \text{Căsătorit}) \rightarrow \mathbf{NU}.$

sunt reciproc exclusive și exhaustive.

Vom încheia prezentarea succintă a acestui domeniu, menționând cele mai importante metode de construire a regulilor de asociere:

- Metoda *directă*, care constă în extragerea regulilor direct din date. Algoritmi bazați pe această metodă sunt, de exemplu:
 - RIPPER ([29]), CN2 ([26]), Holte's 1R [100];
- Metoda *indirectă*, care constă în extragerea regulilor folosind alți clasificatori (e.g. arborii de clasificare și decizie, NN etc.). Un astfel de algoritm este C4.5 (sursă: <http://www2.cs.uregina.ca>)

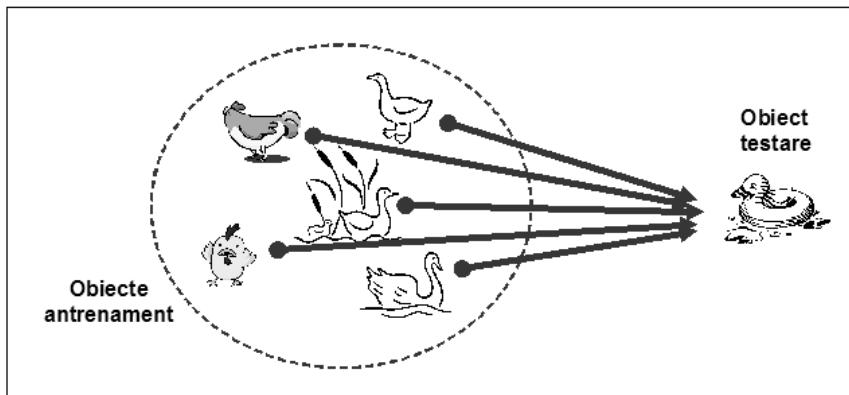
(mai multe amănunte privind aceste metode în [162]).

Din cele expuse mai sus, rezultă că principalele avantaje ale utilizării clasificării bazate pe reguli de asociere sunt următoarele:

- ◆ Expresivitate pronunțată (analogă cu cea a arborilor de clasificare și decizie);
- ◆ Ușurință în interpretare;
- ◆ Ușurință în generare;
- ◆ Viteză ridicată de clasificare a unor noi instanțe;
- ◆ Performanță generală comparabilă cu cea a arborilor de clasificare și decizie

5.5. *k*-nearest neighbor

În domeniul recunoașterii formelor, algoritmul „*k*-nearest neighbor” (*k*-NN) reprezintă metoda clasificării unui nou obiect pe baza celor mai apropiate (*k*) obiecte din vecinătate (*neighbor(hood)*). Pentru a înțelege mai bine despre ce este vorba, să privim cu atenție figura de mai jos (adaptare [162]).



În principiu, având mulțimea de antrenament dată (stânga) și un obiect nou ce trebuie clasificat (dreapta), se calculează „distanța” (similaritatea) față de obiectele de antrenament și se aleg primele k obiecte cele mai apropiate (asemănătoare/vecine). Pentru construirea algoritmului, avem nevoie de (input):

- O mulțime de înregistrări stocate (mulțimea de antrenament);
- O distanță (metrică) pentru a calcula similaritatea dintre obiecte;
- Valoarea lui k , adică a numărului (necesar) de înregistrări din mulțimea de antrenament, pe baza cărora vom face clasificarea unei noi instanțe.

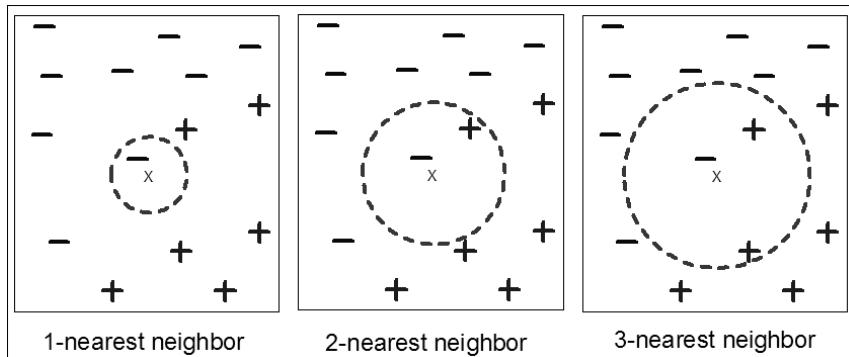
Pe baza acestor trei cerințe, o instanță nouă, necunoscută, va fi clasificată efectuând următorii pași:

- Se calculează distanța la obiectele (înregistrările) de antrenament;
- Se identifică cele mai apropiate k astfel de obiecte (k vecini);
- Se utilizează etichetele acestor vecini pentru etichetarea noului obiect.

Rezultatul (output-ul) clasificării noului obiect va fi eticheta cea mai semnificativă, obținută pe baza analizării celor mai apropiati k vecini (de exemplu, considerând majoritatea voturilor).

Remarcă: Să notăm că, pentru a lua în considerație importanța fiecărui vecin din cei luați în considerare pentru decizie, voturile sunt ponderate, adică fiecăruia dintre cei k cei mai apropiati vecini i se atribuie o anumită pondere a votului (e.g. $w = 1/d^2$, unde d reprezintă distanța dintre noua instanță și vecinul respectiv).

Figura de mai jos (*idem*) prezintă sintetic algoritmul pentru $k = 1, 2, 3$ vecini ai punctului \mathbf{x} .



Remarcă: 1) Cea mai bună alegere a parametrului k depinde de date. Astfel, cu cât k este mai mare, cu atât se reduce mai mult efectul zgomotului asupra clasificării, dar, pe de altă parte, frontierele între clase sunt mai puțin precise. O variantă bună de alegere a lui k se bazează pe metoda validării în cruce (*cross-validation*). În cazul în care $k = 1$, avem de a face cu algoritmul *nearest neighbour*.

2) Acuratețea algoritmului k -NN este foarte puternic influențată (în sens negativ) de prezența zgomotului în date, de caracteristici irelevante ale acestora, sau de scalare acestora necorelată cu importanța lor. În legătură cu problema scalării corespunzătoare a caracteristicilor, amintim utilizarea frecventă în vederea optimizării acestia fie a algoritmilor evolutivi, fie a metodei informației reciproce (*mutual information*).

3) Algoritmul k -NN este ușor de implementat, dar, pe de altă parte, este un clasificator „leneș”, mai ales în prezența unei mulțimi de antrenament mari. Mai precis, punctele sale negative pot fi rezumate la următoarele aspecte:

- Nu construiește modelul de clasificare în manieră explicită, la fel ca, de exemplu, arborii de clasificare și decizie sau regulile de asociere;
- Procesul de clasificare necesită timp îndelungat și resurse importante de calcul;
- Predicția se bazează pe informații locale, deci este susceptibilă de a fi influențată de valori extreme/anomalii (*outliers*).

Pentru contracarare „bilelor negre” de mai sus, de-a lungul timpului au fost luate în considerare diferite optimizări ale algoritmului, mai ales prin reducerea numărului de distanțe calculate.

În final, vom menționa o variantă binecunoscută a algoritmului *k*-NN, de tip *nearest neighbour* (*k* = 1), și anume PEBLS: (*Parallel Example-Based Learning System*) [31], având următoarele caracteristici:

- Utilizează atât variabile (attribute) continue cât și nominale;
- Fiecare înregistrare are ponderea proprie.

(pentru amănunte privind PEBLS: <http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/learning/systems/pebts/0.html>).

Exemplu:

Vom considera, din nou, exemplul privind riscul rambursării unor anumite sume, mulțimea de antrenament fiind dată în tabelul de mai jos.

No.	Rambursare	Statut marital	Venit anual	Excrocare
1	Da	Necasatorit	125.000	NU
2	Nu	Casatorit	100.000	NU
3	Nu	Necasatorit	70.000	NU
4	Da	Casatorit	120.000	NU
5	Nu	Divortat	95.000	DA
6	Nu	Casatorit	60.000	NU
7	Da	Divortat	220.000	NU
8	Nu	Necasatorit	85.000	DA
9	Nu	Casatorit	75.000	NU
10	Nu	Necasatorit	90.000	DA

Dacă în ceea ce privește distanța între attributele continue, metrica uzuală este cea Euclidiană, în ceea ce privește cazul atributelor nominale, se utilizează formula:

$$d(X_1, X_2) = \sum_i \left| \frac{n_{1i}}{n_1} - \frac{n_{2i}}{n_2} \right|,$$

unde X_l sunt atributele nominale în cauză, iar n_{lj} , n_l sunt frecvențele corespunzătoare. De exemplu, în ceea ce privește statutul marital, situație tabelată mai jos,

Clasa	Statut marital		
	Necăsătorit	Căsătorit	Divorțat
DA	2	0	1
NU	2	4	1

avem următoarele calcule:

$$d(\text{Necăsătorit}, \text{Căsătorit}) = |2/4 - 0/4| + |2/4 - 4/4| = 1;$$

$$d(\text{Necăsătorit}, \text{Divorțat}) = |2/4 - 1/2| + |2/4 - 1/2| = 0;$$

$$d(\text{Căsătorit}, \text{Divorțat}) = |0/4 - 1/2| + |4/4 - 1/2| = 1;$$

În ceea ce privește atributul *Rambursare*, situația este tabelată mai jos,

Clasa	Rambursare	
	Da	Nu
Da	0	3
Nu	3	4

iar distanța a fost calculată astfel:

$$d(\text{Rambursare} = \text{Da}, \text{Rambursare} = \text{Nu}) = |0/3 - 3/7| + |3/3 - 4/7| = 6/7$$

În ceea ce privește distanța între înregistrările X și Y , aceasta este dată de formula:

$$d(X, Y) = w_X \cdot w_Y \cdot \sum_i d^2(X_i, Y_i),$$

unde w_X și w_Y reprezintă ponderile respective, date de formula:

$$w_X = \frac{\text{Numarul de utilizari ale lui } X \text{ în predictie}}{\text{Numarul de predictii corecte ale lui } X}.$$

Pentru cei interesați de amănuțe privind algoritmul k -NN, menționăm [35], [140] și [162].

5.6. Mulțimi rough

Mulțimile rough (*rough sets* –RS) au fost propuse de către Zdzisław Pawlak [128], [129], în încercarea de a trata matematic concepțele *vagi*, principala lor menire fiind „transformarea automată a datelor în cunoștințe”. Pawlak a arătat că principiile învățării din exemple (utilizând mulțimea de antrenament, aşa cum am văzut din paragrafele precedente) pot fi formulate în contextul acestei abordări. În cele ce urmează vom prezenta, foarte succint, principiile care stau la baza acestui concept, precum și câteva aplicații în DM.

Ideea de bază în domeniul mulțimilor rough este aceea că, plecând de la o mulțime de obiecte, un set de atribute și valori de decizie, să se poată crea reguli pentru găsirea aproximării superioară și inferioare, precum și a regiunii de frontieră (de margine) a unei mulțimi de obiecte. Aceste reguli odată construite, un obiect nou poate fi cu ușurință clasificat într-o regiune. Prezentăm, mai jos, „algoritmul” corespunzător abordării cu mulțimi rough.

- ◆ Primul pas în procesul mulțimilor rough îl constituie utilizarea unei baze relațională de date (i.e. obiecte cu atribute și valori ale atributelor pentru fiecare obiect). La fel ca la orice problemă clasică de decizie, unul dintre atribute va fi ales ca atribut de decizie (atribut dependent), în timp ce celelalte atribute vor reprezenta atributele pe baza cărora se ia decizia (attribute independente, predictive, condiționale).
- ◆ Al doilea pas al procesului constă în formarea de clase de echivalență. Acestea sunt, de fapt, grupuri de obiecte pentru care valorile tuturor atributelor predictive sunt aceleași pentru fiecare obiect și, în consecință, acestea nu pot fi deosebite între ele.
- ◆ Al treilea pas al procesului constă în construirea matricei de discernabilitate (distincție, deosebire etc.) (*discernability matrix*), ale cărei elemente reprezintă atributele care deosebesc fiecare clasă de echivalență de celelalte.

Remarcă: 1) Nu toate atributele pot fi necesare pentru formarea claselor de echivalență. De aceea, se ia în considerație un redus relativ (*relative reduct*), format din informația suficientă, necesară pentru a discerne obiectele dintr-o clasă de cele din celelalte clase.

2) Acele clase pentru care există mai mult de o valoare a atributului de decizie (dependent), se numesc clase *vagi* (*vague classes*).

- ◆ Se construiesc aproximările: (a) inferioară (*lower approximation*) și (b) superioară (*upper approximation*) ale unei mulțimi de obiecte X . Astfel, *aproximarea inferioară* a lui X este o colecție de obiecte care pot fi clasificate sigur ca membre ale lui X (cu certitudine maximă), în timp ce *aproximarea superioară* a lui X este o colecție de obiecte care pot fi clasificate ca posibile elemente ale lui X (cu incertitudine).
- ◆ Se construiește apoi regiunea de frontieră/margine (*boundary region*) a lui X , cuprinzând obiectele ce nu pot fi clasificate cu certitudine nici în X , nici înapoia lui X .
- ◆ În final, plecând de la redusul relativ, se construiesc regulile. Considerând acum un nou obiect, necunoscut (i.e. fără valoare de decizie), utilizând regulile de decizie astfel construite, acesta va fi clasificat (i.e. i se va atribui o valoare de decizie).

În cele ce urmează, vom prezenta formalizat (matematic) câteva noțiuni din „lumea” mulțimilor rough.

Un *sistem de informații* (*information system* -IS) înseamnă un tabel în care fiecare linie reprezintă un eveniment, un caz, un obiect, o instanță, în timp ce fiecare coloană reprezintă un atribut (o observație, o proprietate etc.) care va avea o valoare atașată fiecărui obiect. Atributele sunt aceleși pentru fiecare obiect, doar valorile lor sunt diferite. Putem cumva privi un sistem de informații ca o bază relațională de date. Tabelul de mai jos reprezintă un astfel de sistem.

	Studii	Educație	Sex	Venit
1	da	bună	M	mediu
2	da	bună	M	scăzut
3	da	bună	F	mediu
4	nu	slabă	M	fără
5	nu	slabă	M	scăzut

În acest IS, atributul *Venit* reprezintă atributul de decizie (dependent), în timp ce attributele *Studii*, *Educație* și *Sex* reprezintă attributele predictive (independente). Scopul analizei este descoperirea regulilor, pe baza cărora să se poată prezice ce venit ar putea avea o nouă persoană (necunoscută), considerând valorile atributelor sale independente (*Studii*, *Educație*, *Sex*).

Formal, un *sistem de informații* este o pereche $I = (U, A)$, unde:

- U este o mulțime (finită și nevidă) de *obiecte*, numită *univers*,
 - A este o mulțime (finită și nevidă) de *attribute*,
- și o funcție $a: U \rightarrow V_a$, pentru orice $a \in A$, unde mulțimea V_a se numește *multimea valorilor* atributului a .

Remarcă: 1) Funcția a nu este injectivă, astfel încât pot exista obiecte diferite cu aceeași valoare a atributului a .

2) Deoarece am considerat că unul dintre attribute reprezintă atributul de decizie (în exemplul de mai sus -*Venitul*), sistemul de informații de mai sus poate fi privit ca *sistem de decizie*.

Următoarea definiție introduce conceptul de *relație de indiscernabilitate*, sau nedistincție (*indiscernability relation*). O astfel de relație există între două obiecte dacă toate valorile corespunzătoare unui anumit atribut sunt aceleași, deci ele nu pot fi deosebite luând în considerație acel atribut. Formal, dacă o_1 și o_2 sunt două obiecte diferite, pentru care:

$$a_i(o_1) = a_i(o_2),$$

spunem că o_1 și o_2 sunt *obiecte indiscernabile* (nedistincte) în raport cu atributul $a_i \in A$.

Pentru sistemul de decizie de mai înainte, se observă că obținem următoarele trei grupuri de persoane, cu valori ale atributelor predictive identice: $\{1, 2\}$, $\{3\}$, $\{4, 5\}$. Putem deci considera trei clase (de echivalență), plecând de la cele trei grupuri de persoane (nedistincte) de mai sus, prezentate sub forma următorului tabel.

	Studii	Educație	Sex
1	da	bună	M
2	da	bună	F
3	nu	slabă	M

Formal, dacă $I = (U, A)$ este un sistem de informații și $B \subseteq A$ o submulțime de attribute, atunci B induce o relație (de echivalență), notată $IND(B) \subseteq U \times U$, definită de:

$$IND(B) = \{(x, y) \in U \times U \mid \forall a \in B, a(x) = a(y)\}.$$

$IND(B)$ se numește *relație indusă de B pe obiectele din U* . Dacă $(x, y) \in IND(B)$, atunci obiectele x și y sunt indiscernabile (nedistincte) în funcție de attributele mulțimii B . Prin această relație, mulțimea de obiecte U se

partiționează în clase de echivalență, notate $[x]_B$. Formal, partiția lui U generată de $IND(B)$ este dată de:

$$U / IND(B) = \otimes \{a \in B \mid U / IND(\{a\})\},$$

unde:

$$A \otimes B = \{X \cap Y \mid \forall X \in A, \forall Y \in B, X \cap Y \neq \Phi\}.$$

Remarcă: Să împărțim mulțimea atributelor A în două submulțimi distincte: P și D , unde P reprezintă mulțimea atributelor predictive (condiționale), iar D este formată din atributul(ele) de decizie (să notăm, în context, că există cazuri în care mulțimea D conține mai multe attribute de decizie). Putem astfel să definim și partițiiile $U/IND(P)$, $U/IND(D)$, numite clasele de echivalență *de predicție* (condiție), respectiv de decizie.

Dacă $U = \{x_1, x_2, \dots, x_n\}$ reprezintă universul sistemului de informații (decizie), atunci *matricea de discernabilitate* este definită de formula:

$$m_{ij} = \{a \in P \mid a(o_i) \neq a(o_j) \wedge (d \in D, d(o_i) \neq d(o_j))\},$$

unde m_{ij} reprezintă mulțimea tuturor atributelor care clasifică obiectele o_i și o_j în clase diferite de decizie ale partiției $U/IND(D)$.

Exemplu:

Fie $U = \{x_1, x_2, \dots, x_7\}$ universul obiectelor, $P = \{a_1, a_2, a_3, a_4\}$ mulțimea atributelor predictive și $D = \{d\}$ mulțimea atributelor de decizie.

	a_1	a_2	a_3	a_4	d
x_1	1	0	2	1	1
x_2	1	0	2	0	1
x_3	1	2	0	0	2
x_4	1	2	2	1	0
x_5	2	1	0	0	2
x_6	2	1	1	0	2
x_7	2	1	2	1	1

Atunci, matricea de discernabilitate corespunzătoare este dată de:

	x_1	x_2	x_3	x_4	x_5	x_6
x_2	-					
x_3	$\{a_2, a_3, a_4\}$	$\{a_2, a_3\}$				
x_4	$\{a_2\}$	$\{a_2, a_4\}$	$\{a_3, a_4\}$			
x_5	$\{a_1, a_2, a_3, a_4\}$	$\{a_1, a_2, a_3\}$	-	$\{a_1, a_2, a_3, a_4\}$		
x_6	$\{a_1, a_2, a_3, a_4\}$	$\{a_1, a_2, a_3\}$	-	$\{a_1, a_2, a_3, a_4\}$	-	
x_7	-	-	$\{a_1, a_2, a_3, a_4\}$	$\{a_1, a_2\}$	$\{a_3, a_4\}$	$\{a_3, a_4\}$

De exemplu, partiția produsă de atributul de decizie d este dată de:

$$U/\{d\} = \{\{x_4\}, \{x_1, x_2, x_7\}, \{x_3, x_5, x_6\}\}$$

Să mai notăm că mulțimea:

$$CORE(P) = \{a \in P \mid \exists i, j \text{ astfel incat } m_{ij} = \{a\}\},$$

este numită *nucleul (core)* lui P . În cazul de față, $CORE(P) = \{a_2\}$.

Partiția produsă de nucleu este dată de:

$$U/\{a_2\} = \{\{x_1, x_2\}, \{x_3, x_6, x_7\}, \{x_5, x_4\}\}.$$

Fie $I = (U, A)$ un sistem de informații (decizie), mulțimile $B \subseteq A$ și $X \subseteq U$. Atunci, pe baza cunoștințelor extrase din B putem construi două mulțimi de aproximare a mulțimii X și, în raport cu aceste mulțimi, putem obține o nuanțare privind apartenența unor elemente ale lui U la mulțimea X . Cu alte cuvinte, X poate fi aproximată utilizând numai informația conținută în B , prin construcția acestor B -mulțimi de aproximare. Formal, avem:

$$\underline{B}X = \{x \in U \mid [x]_B \subseteq X\} - \text{aproximarea } B\text{-inferioară a lui } X,$$

care reprezintă, așa cum am mai spus, mulțimea de obiecte din U care pot fi clasificate *sigur* ca membre ale lui X (cu certitudine maximă) și:

$$\overline{B}X = \{x \in U \mid [x]_B \cap X \neq \emptyset\} - \text{aproximarea } B\text{-superioară a lui } X,$$

care reprezintă mulțimea de obiecte din U care pot fi clasificate ca *posibile* elemente ale lui X (cu incertitudine).

Remarcă: Uneori, cuplul $(\underline{B} X, \overline{B} X)$ mai este cunoscut ca *mulțime rough*.

Mulțimea $BN_B(X) = \overline{B} X - \underline{B} X$ se numește *regiunea de frontieră (de margine)* a lui X , cuprinzând obiectele ce nu pot fi clasificate cu certitudine nici în X , nici înapoia lui X .

În fine, o mulțime X se numește *mulțime rough* sau *B-rough* (pe baza cunoștințelor din B) dacă $BN_B(X) \neq \Phi$. Mulțimea X se numește *mulțime crisp* sau *B-definibilă* (pe baza cunoștințelor din B) dacă $BN_B(X) = \Phi$.

Exemplu:

Fie sistemul de decizie tabelat mai jos, în care atributele predictive sunt A_1 și A_2 , iar cel decizional este A_3 .

	A_1	A_2	A_3
x_1	16-30	50	Da
x_2	16-30	0	Nu
x_3	41-45	1-25	Nu
x_4	31-45	1-25	Da
x_5	46-60	26-49	Nu
x_6	16-30	26-49	Da
x_7	46-60	26-49	Nu

Dacă alegem $B = \{A_3\}$ și $X = \{x \mid A_3(x) = \text{Da}\}$, atunci avem:

$$\underline{B} X = \{x_1, x_6\},$$

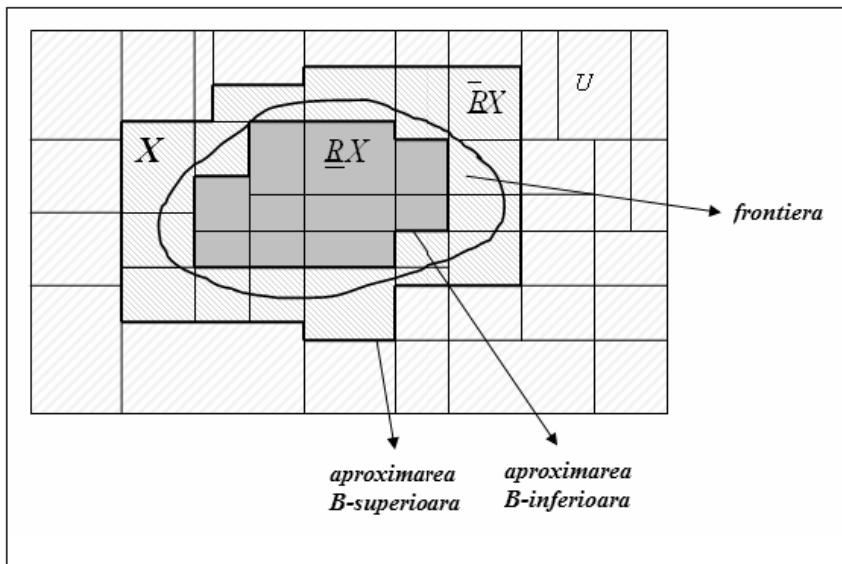
$$\overline{B} X = \{x_1, x_3, x_4, x_6\},$$

și deci, deoarece $BN_B(X) \neq \Phi$, rezultă că mulțimea X este o mulțime rough.

Remarcă: Putem defini acuratețea aproximării cu ajutorul formulei:

$$\text{Acuratețea} = \text{Card}(\underline{B} X) - \text{Card}(\overline{B} X).$$

Prezentăm în figura de mai jos o ilustrare sugestivă privind mulțimile de aproximare.



Am spus la început că nu toate atributele sunt necesare pentru formarea claselor de echivalență, și de aceea se ia în considerație un redus relativ, format din informația suficientă, necesară pentru a discerne obiectele dintr-o clasă de cele din celelalte clase. Formal, pentru un sistem de informații (decizie) $I = (U, A)$, *redusul relativ* al sistemului reprezintă o submulțime minimală de atrbute $B \subseteq A$, cu proprietatea ca $IND(B) = IND(A)$. Cu alte cuvinte, redusul este o mulțime minimală de atrbute din A , care păstrează relația de indiscernabilitate (nedistincție), deci și partițiile universului, și realizează aceleși clasificări ca și întreaga mulțime de atrbute A . Să notăm că se poate întâmpla să existe mai mulți reduși, iar:

$$CORE(P) = \cap RED(P),$$

unde $RED(P)$ reprezintă familia tuturor redușilor lui P .

Exemplu:

Fie următorul sistem de informații, tabelat mai jos.

	Durere de cap	Strănut	Temperatură	Viroză respiratorie
X_1	Da	Da	Normală	Nu
X_2	Da	Da	Mare	Da
X_3	Da	Da	Foarte mare	Da
X_4	Nu	Da	Normală	Nu
X_5	Nu	Nu	Mare	Nu

X_6	Nu	Da	Foarte mare	Da
-------	----	----	-------------	----

În acest caz, putem considera redușii:

- Redus 1 = $\{Strănut, Temperatură\}$, având tabelul corespunzător:

	Strănut	Temperatură	Viroză respiratorie
X_1, X_4	Da	Normală	Nu
X_2	Da	Mare	Da
X_3, X_6	Da	Foarte mare	Da
X_5	Nu	Mare	Nu

- Redus 2 = $\{Durere de cap, Temperatură\}$, având tabelul corespunzător:

	Durere de cap	Temperatură	Viroză respiratorie
X_1	Da	Normală	Nu
X_2	Da	Mare	Da
X_3	Da	Foarte mare	Da
X_4	Nu	Normală	Nu
X_5	Nu	Mare	Nu
X_6	Nu	Foarte mare	Da

În ceea ce privește relația dintre nucleu și reduși, dată de formula:

$$CORE(P) = \cap RED(P),$$

se observă că:

$$\begin{aligned} CORE &= \{Strănut, Temperatură\} \cap \{Durere de cap, Temperatură\} = \\ &= \{Temperatură\} \end{aligned}$$

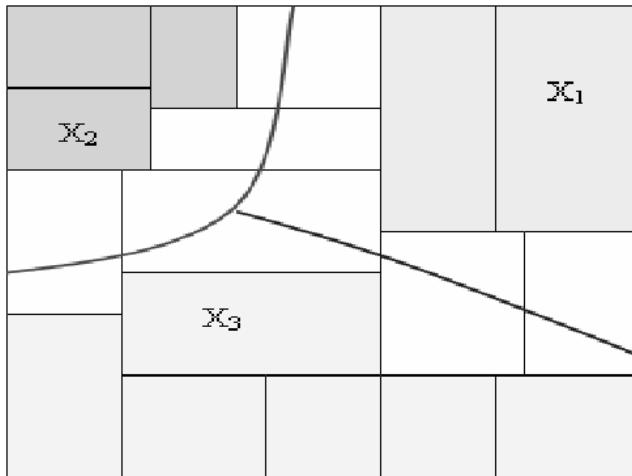
În final, vom menționa pe scurt aspectul *decizional* implicat de abordarea cu ajutorul mulțimilor rough. Astfel, să considerăm un sistem de decizie DS (i.e. un sistem de informații $I = (U, A)$ care conține și attribute de decizie, formând mulțimea D). Pentru simplitate, să presupunem că mulțimea atributelor de decizie D conține un singur atribut d . Atunci, decizia d determină partitia:

$$CLASS_I(d) = \{X_1, X_2, \dots, X_{r(d)}\}$$

a universului U , unde:

$$X_k = \{x \in U \mid d(x) = k\}, \quad 1 \leq k \leq r(d).$$

Mulțimea $CLASS_I(d)$, astfel construită, se numește *clasificarea* obiectelor din I determinată de decizia d , iar mulțimea X_k se numește *clasa a k-a de decizie* a lui I . Vom ilustra în figura de mai jos o asemenea clasificare, conținând trei clase de decizie. Partiția conține: *aproximarea superioară*, *aproximarea inferioară* și *regiunea de frontieră*.



Există un întreg „univers” privind mulțimile rough, pentru argumentare fiind suficient să amintim că există chiar și o *Societate Internațională pentru mulțimi rough (International Rough Set Society)* – <http://www.roughsets.org/>.

Aplicațiile mulțimilor rough sunt foarte diverse, aici vom menționa doar câteva:

- ◆ Mulțimile rough pot fi utilizate ca bază teoretică pentru rezolvarea unor probleme de învățare automată (în special în domeniul inducerii regulilor și selectarea caracteristicilor). Au stat de asemenea la baza unor cercetări în logică;
- ◆ Mulțimile rough au fost aplicate în cercetări avansate din domeniul medical (lavajul peritoneal în pancreatită, predicția toxicității, predicția decesului în pneumonie, dezvoltarea sistemelor expert în medicină, diagnosticului malformațiilor congenitale, predicția recidivelor în leucemia infantilă, diagnosticul diabetului infantil etc.). Menționăm, de asemenea, utilizarea lor cu succes, împreună cu rețelele neuronale probabiliste, în diagnosticul cirozei biliare primare [133], diagnosticul cancerului mamar [132], diagnosticul cancerului hepatic [77];

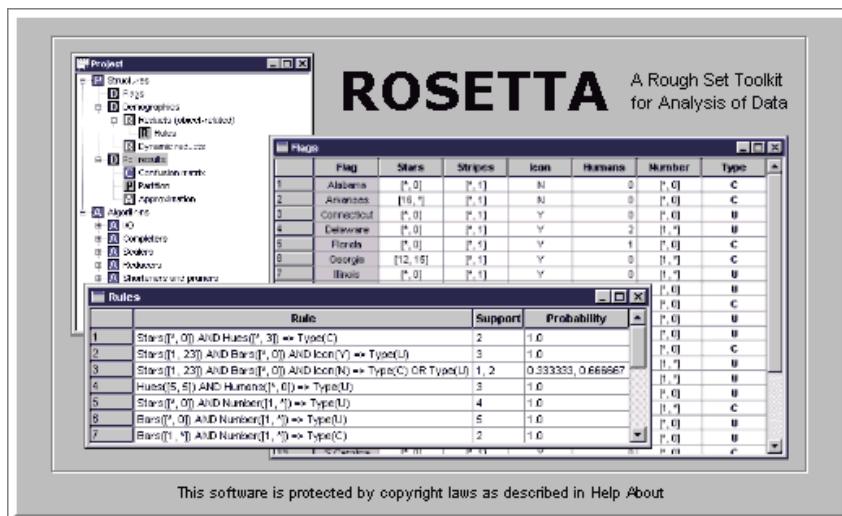
- ◆ Mulțimile rough au fost aplicate în cercetări din domeniul descoperirii de cunoștințe (*Knowledge Discovery*): construirea regulilor de decizie, procese de discretizare, imputări de date etc.

Să amintim că teoria mulțimilor rough a fost extinsă în domeniul fuzzy, prin considerarea claselor de echivalență fuzzy (*fuzzy-rough sets*).

Pentru cei ce doresc utilizarea mulțimilor rough în diferite domenii de cercetare, recomandăm utilizarea, printre altele, a unui sistem software specializat, numit ROSETTA (<http://rosetta.lcb.uu.se/>). Așa cum specifică autorii, ROSETTA este un *toolkit* pentru analiza datelor tabelate în cadrul teoriei mulțimilor rough. Acest sistem este proiectat pentru o întreagă gamă de probleme DM și de descoperire a cunoștințelor ca, de exemplu:

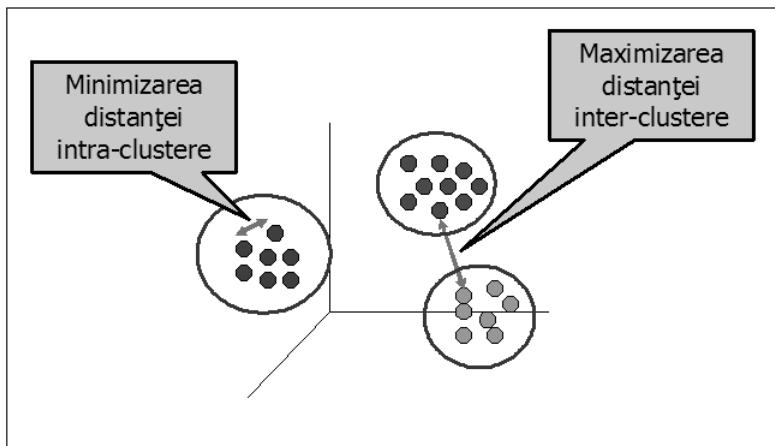
- Pre-procesarea inițială a datelor;
- Identificarea mulțimilor minimale de atribute;
- Generarea de reguli IF-THEN;
- Generarea de pattern-uri;
- Validarea și analiza regulilor induși sau a pattern-urilor.

Mai multe detalii pot fi găsite în [112], [124], [125]. Figura de mai jos ilustrează o „fereastră” de lucru a respectivului software.



5.7. Clustering

Așa cum am arătat în capitolul introductiv, această tehnică înseamnă găsirea grupurilor (clusterelor) de obiecte, pe baza similarității acestora, astfel încât, în interiorul fiecărui grup să existe o mare similaritate, în timp ce grupurile să fie cât mai diferite unul de celălalt. Ilustrăm în figura de mai jos această filosofie a procedurii de clustering.



Din punctul de vedere al învățării automate, procesul de clustering este o formă de învățare nesupervizată.

Dacă mai sus am explicat ce este un proces (analiză) clustering, să punctăm acum ce nu poate fi:

- Clasificare (supervizată) – bazată pe informații despre etichetele claselor;
- Segmentare simplă – bazată pe anumite reguli;
- Rezultatele unei chestionări – rezultate ale unei specificații externe;
- Partiționare grafică.

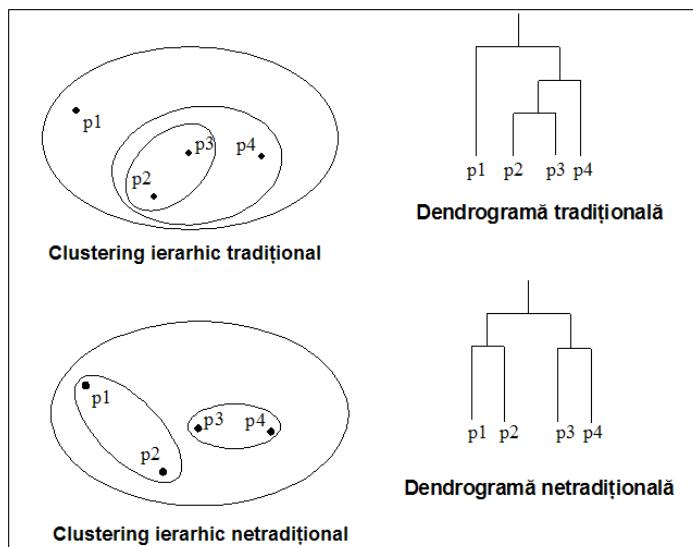
În principiu, metodologia clustering are două abordări distincte:

- ◆ Clustering ierarhic (*hierarchical clustering*);
- ◆ Clustering neierarhic/partițional (*non-hierarchical clustering/partitional/flat*).

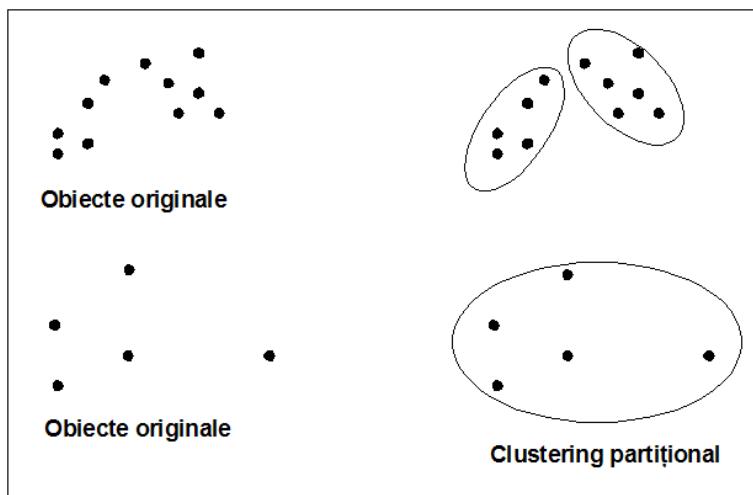
Clusteringul ierarhic descoperă clustere succesive, utilizând clusterele stabilite în prealabil, construind deci o ierarhie de clustere (producând o *dendrogramă* = diagramă arbore) și nu doar o simplă partiție a obiectelor. Numărul clusterelor nu este cerut ca o condiție input a algoritmului, în timp ce se poate utiliza o anumită condiție de terminare a sa (e.g. un număr prestabilit de clustere). Vom menționa trei tipuri de clustering ierarhic:

- *Aglomerativ* (*agglomerative* - bottom-up), în care perechi de obiecte/clustere sunt conectate succesiv producând clustere mai mari. Metoda constă în:
 - (a) Se plasează fiecare obiect în propriul cluster (i.e. obiect = cluster);
 - (b) Se unesc la fiecare pas cele mai asemănătoare două clustere, până când se obține doar unul singur sau, alternativ, condiția de stop este îndeplinită.
- *Diviziv* (*divisive/partitioning* - top-down), în care toate obiectele sunt plasate inițial într-un singur cluster și apoi, succesiv, împărțite în grupuri separate. Metoda constă în:
 - (a) Se începe cu un singur (mare) cluster, conținând toate obiectele;
 - (b) Se divide clusterul cel mai distinctiv în clustere mai mici, procedându-se astfel până când se obține un anumit număr de clustere, sau o altă condiție de stop este îndeplinită.
- *Conceptual*, care constă într-un nod-rădăcină gol, obiectele fiind adăugate una câte una (clustering incremental), utilizând clase deja existente, creând noi clase, combinând și divizând clase existente (e.g. *cobweb* [49]).

Ilustrăm mai jos, sintetic, diferența dintre procesul tradițional și netradițional de clustering ierarhic.



Clusteringul neierarhic, partițional, constă în partaționarea mulțimii inițiale a obiectelor în submulțimi (clustere) ce nu se suprapun, astfel încât fiecare obiect să aparțină exact unui cluster, așa cum se vede în figura de mai jos.



Procesul de clustering implică, în principiu, trei pași:

- Definirea unei măsuri de similaritate;

- Definirea unui criteriu pentru procesul de creare a clusterelor;
- Construirea unui algoritm care să construiască clustere pe baza criteriului ales.

Un algoritm de clustering are ca scop identificarea grupurilor naturale de obiecte/instanțe dintr-o mulțime dată și, ca atare, are nevoie de a măsura gradul lor de asemănare pe baza unui criteriu dat. Primul lucru care trebuie făcut este, deci, considerarea unei măsuri corespunzătoare naturii datelor și scopului propus, pentru a putea evalua „distanța” dintre obiecte.

Un alt aspect important în procesul de clustering este modul de apreciere a validității clusterelor construite pe baza unui anumit algoritm. Deși este dificilă considerarea unei funcții obiectiv, vom puncta câteva aspecte importante în această direcție:

- ◆ Validarea *externă*, care constă în compararea clusteringului obținut cu alte variante *a priori* de clustering;
- ◆ Validarea *internă*, care constă în verificarea dacă clusteringul este în mod intrinsec corespunzător datelor;
- ◆ Validarea *relativă*, care compară clusteringul obținut cu rezultatele partiționării datelor cu alte metode.

Principalele probleme ale unei analize cluster (*cluster analysis*) constau în următoarele:

- Pregătirea datelor – colectarea și aranjarea datelor în vederea procesului de clustering;
- Alegerea măsurii de similaritate – stabilirea modului cum se calculează „distanța” dintre obiecte;
- Utilizarea cunoștințelor disponibile, referitoare la domeniul dat, care pot ajuta la pregătirea datelor și alegerea măsurii de similaritate;
- Eficiența construcției clusterelor – calitatea construcției și timpul afectat ei.

Rezumând cele de mai sus, principalele puncte care trebuie avute în vedere în procesul de clustering sunt următoarele:

- *Formularea problemei* – selectarea obiectelor pentru clustering;
- *Alegerea măsurii de similaritate* – selectarea unei „distanțe” proprii obiectelor și scopului (criteriului) propus;
- *Selectarea procedurii* de clustering;
- *Selectarea numărului de clustere* (sau a condiției de terminare) – după cum e cazul;

- *Ilustrarea grafică și interpretarea clusterelor* (tragerea concluziilor);
- *Aprecierea validității și robusteții* modelului, utilizând diferite metode, ca de exemplu:
 - Repetarea procesului, utilizând și alte măsuri de similaritate corespunzătoare contextului;
 - Repetarea procesului, utilizând și alte tehnici de clustering;
 - Repetarea procesului de mai multe ori, dar ignorând unul sau mai multe obiecte, la fiecare iterație.

Ne-am axat până acum pe aspectul măsurării distanței (similarității) între obiectele/instanțele ce trebuie clusterizate, deci în contextul rezolvării problemei minimizării distanței intra-cluster. Pe de altă parte, trebuie rezolvată și problema maximizării distanței inter-clustere, cu alte cuvinte avem nevoie și de definirea unei distanțe (legături) între două clustere. Există mai multe metode de a rezolva această problemă, cele mai cunoscute fiind următoarele:

- Legătură/articulație unică (*single linkage*): distanța între două clustere reprezintă minimul distanței dintre oricare două obiecte din cele două clustere (i.e. distanța uzuală dintre două mulțimi);
- Legătură/articulație medie (*average linkage*): distanța dintre două clustere reprezintă media distanțelor dintre obiectele celor două clustere luate perechi (un element din primul și un element din al doilea);
- Legătură/articulație completă (*complete linkage*): distanța dintre două clustere reprezintă maximul distanței dintre oricare două obiecte din cele două clustere;
- *Distanța centroidă* (*centroid distance*): distanța dintre două clustere reprezintă distanța dintre centroizii acestora.

Remarcă: Să menționăm, în context, și alte trei abordării privind distanța dintre două clustere:

- Suma tuturor dispersiilor intra-cluster;
- Creșterea în dispersie (*criteriul Ward*);
- Probabilitatea ca cele două clustere să aibă aceeași repartiție (*Vaccaro-linkage*).

Vom prezenta, în continuare, câteva chestiuni de bază privind măsurarea similarității între obiecte, primul pas efectiv în procesul de clustering. În contextul metodologiei de clustering, măsura de similaritate indică cât de asemănătoare (similar) sunt două obiecte. De multe ori însă, în loc de similaritate se utilizează noțiunea de disimilaritate, care este mai

adecvată, în ideea măsurării unei distanțe. Indiferent de modul de a compara două obiecte, problema alegerii concrete a unei asemenea măsuri depinde esențial de problema în sine, de scopul partilionării și de așteptările avute.

De obicei, unei asemenea măsuri i se cere să aibă anumite proprietăți, aceasta depinzând și de problema concretă la care se aplică. În principiu, o măsură de *similaritate* este o funcție $d: D \times D \rightarrow \mathbb{R}$, aplicată pe o mulțime de obiecte D și având anumite proprietăți specifice. Conceptual, putem spune că *similaritatea = distanța⁻¹* și din această cauză terminologia proprie ar fi măsură de *disimilaritate*, privită ca distanță între două obiecte. Cu toate acestea, termenul consacrat este cel de măsură de *similaritate*. Privită ca distanță, o asemenea măsură ar trebui să posede proprietățile de bază ale unei metriki, și anume:

- ◆ Ne-negativitatea și identitatea: $d(A, B) \geq 0$, $d(A, B) = 0 \Leftrightarrow A = B$;
- ◆ Simetria: $d(A, B) = d(B, A)$;
- ◆ Inegalitatea triunghiului: $d(A, B) + d(B, C) \geq d(A, C)$.

Remarcă: Nu totdeauna unele dintre proprietățile de mai sus sunt necesare. De exemplu, proprietatea de simetrie (e.g. în procesarea imaginilor pot fi cazuri în care un imaginea unui copil este considerată mai similară unui părinte decât viceversa). La fel este cazul și pentru inegalitatea triunghiului. Să considerăm, spre exemplu, vectorii $\mathbf{x}_1 = (a, a, a, a)$, $\mathbf{x}_2 = (a, a, b, b)$ și $\mathbf{x}_3 = (b, b, b, b)$. Dacă putem spune că distanțele dintre \mathbf{x}_1 și \mathbf{x}_2 , la fel ca și dintre \mathbf{x}_2 și \mathbf{x}_3 , sunt suficient de mici, distanța între primul și ultimul poate fi considerată foarte mare. Ilustrarea foarte sugestivă a acestei situații este cunoscută comparație între om, centaur și cal.

Să menționăm, în context, și alte proprietăți pe care le poate avea o măsură de similaritate:

- ◆ Proprietăți de *contingență*, deseori întâlnite în domeniul recunoașterii formelor (pentru evitarea confuziilor de terminologie, termenii vor fi prezentați și în lb. Engleză în original): robustețe la perturbații (*perturbation robustness*), robustețe la curburi (*crack robustness*), robustețe la neclarități/pete (*blur robustness*), robustețe la zgomot și ocluzii (*noise and occlusion robustness*).
- ◆ Proprietăți de *invarianță*: o măsură de similaritate este invariantă la un grup de transformări G , dacă oricare ar fi $g \in G$, $d(g(A), g(B)) = d(A, B)$. De exemplu, în cazul recunoașterii formelor, este indicat ca măsura să fie invariantă la transformări afine.

Remarcă: Alegerea unei asemenea măsuri de similaritate este totdeauna în concordanță cu tipul de date disponibil (numerice, nominale, ordinale, fuzzy etc.).

Prezentăm în continuare unele dintre cele mai cunoscute măsuri de similaritate, aplicate în majoritatea cazurilor. Mai întâi însă, vom preciza că pentru a măsura similaritatea dintre două obiecte/instanțe, le vom considera ca vectori: $\mathbf{x} = (x_1, x_2, \dots, x_n)$, $\mathbf{y} = (y_1, y_2, \dots, y_n)$, pentru simplitate de aceeași dimensiune, cu toate că se pot lua în considerație (cu anumite modificări) și dimensiuni diferite.

1. Distanța *Minkowski*:

$$d_p(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}, p \in \mathbf{N}^*.$$

Notă: Pentru $p = 1$ se obține distanța *Manhattan* (*city block* sau *taxicab*):

$$d_{cb}(x, y) = \sum_{i=1}^n |x_i - y_i|,$$

pentru $p = 2$ se obține distanța *Euclidiană*:

$$d_E(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^2 \right)^{1/2},$$

și, în final, pentru $p \rightarrow \infty$, se obține distanța *Chebychev*:

$$d_C(x, y) = \max_i |x_i - y_i|$$

Remarcă: Distanța *Manhattan* pentru vectori binari devine distanța *Hamming*, cu alte cuvinte distanța Hamming între două coduri este numărul de simboluri diferite (sau numărul de biți diferenți).

2. Măsura *cosinus* (*Cosine*):

$$d_c(x, y) = \frac{x \cdot y^\top}{\|x\|_2 \cdot \|y\|_2},$$

adică cosinusul unghiului dintre cei doi vectori (vezi produsul scalar).

3. Măsura Tanimoto:

$$d_T(x, y) = \frac{x \cdot y^\tau}{x \cdot x^\tau + y \cdot y^\tau - x \cdot y^\tau}.$$

4. Măsura Jaccard:

$$d_J(x, y) = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2 + \sum_{i=1}^n y_i^2 - \sum_{i=1}^n x_i y_i}.$$

5. Măsura Pearson's r sau măsura *coeficientului de corelație r* :

$$r(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}.$$

6. Măsura Mahalanobis:

În general, măsura Mahalanobis este dată de formula:

$$d_M(x, y) = \sqrt{(x - y) \cdot B \cdot (x - y)^\tau},$$

unde B reprezintă orice matrice simetrică, pozitiv definită. În particular, să considerăm o mulțime de n vectori, priviți ca eșantion de dimensiune n a n variabile aleatoare independente, în cadrul unei analize a regresiei de pildă.

Dacă pentru fiecare pereche de vectori \mathbf{x} și \mathbf{y} , $\text{cov}(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^n x_i y_i - \bar{x} \bar{y}$

reprezintă covarianța lor, și $\text{cov}(D)$ reprezintă matricea de covarianță corespunzătoare eșantionului, atunci măsura Mahalanobis în acest caz este dată de formula:

$$d_M(x, y) = \sqrt{(x - y) \cdot \text{cov}(D)^{-1} \cdot (x - y)^\tau}.$$

Remarcă: O distanță Mahalanobis diferă de o distanță Euclidiană prin inserarea inversei covarianței în mijlocul formei pătratice. Dacă variabilele eșantionului nu sunt corelate, totul se rezumă la simpla distanță Euclidiană.

7. Măsuri fuzzy

Măsurile de similaritate fuzzy sunt utilizate ca instrumente gata construite pentru compararea de vectori sau matrice ale căror elemente iau valori în intervalul $[0, 1]$. Fie \mathbf{x} și \mathbf{y} doi vectori, astfel încât componentele x_i și y_i aparțin intervalului $[0, 1]$. Valorile lui x_i reprezintă măsura în care vectorul \mathbf{x} posedă a i -a caracteristică (atribut), deci x_i denotă gradul în care elementul i aparține lui \mathbf{x} . În acest context, vectorii *crisp* reprezintă cazuri speciale de mulțimi fuzzy, caz în care x_i ia valori în mulțimea $\{0, 1\}$.

Să considerăm cantitatea:

$$s(p_i, q_i) = \max \{ \min \{ p_i, q_i \}, \min \{ 1 - p_i, 1 - q_i \} \},$$

utilizată în definirea măsurilor de similaritate fuzzy, plecând de la cazul clasic corespunzător. De exemplu, măsurile *fuzzy Minkowski* și *fuzzy Hamming* sunt date de formulele:

$$d_F^p(x, y) = \left(\sum_{i=1}^n s(x_i, y_i)^p \right)^{1/p},$$

$$d_F(x, y) = \sum_{i=1}^n s(x_i - y_i).$$

Remarcă: 1) Pot fi încorporate ponderi în orice măsură de similaritate, cu scopul de a ierarhiza importanța fiecărui atribut, din punctul de vedere al scopului propus. De exemplu, o măsură *Minkowsky ponderată* este de forma:

$$d_{p,\alpha}(x, y) = \left(\sum_{i=1}^n \alpha_i |x_i - y_i|^p \right)^{1/p}, \alpha_i > 0, \sum_i \alpha_i = 1,$$

unde α_i reprezintă ponderea atributului x_i .

2) În multe aplicații reale, există situații în care obiectele/instanțele au ca reprezentare un vector mixt, în care componentele au semnificații diferite, având deci nării diferite (e.g. numerice, nominale, ordinale, fuzzy etc.). De exemplu, dacă vom considera un caz din domeniul medical, în care vectorul \mathbf{x} reprezintă un pacient, atunci unele atribute vor fi numerice (e.g. vârstă, greutatea, înălțimea), altele vor fi nominale (e.g. sexul, locul de rezidență), altele ordinale (e.g. stadiul bolii, mărimea tumorii), altele fuzzy (e.g. factorii de risc: fumat, consum băuturi alcoolice etc.) etc. În cazul studiilor meteorologice putem avea, la fel, atribute numerice (e.g. temperatura, presiunea), ordinale (e.g. umiditatea, tăria

vântului), nominale (e.g. locația) etc. Este evident ca în aceste cazuri particulare să se utilizeze măsuri de similaritate speciale, care să țină seama de specificitatea domeniului. O cale naturală de a rezolva asemenea probleme este aceea de a considera o măsură de similaritate *mixtă* (pentru a cuantifica tipul de date) și *ponderată* în același timp (pentru a cuantifica semnificația atributelor). Concret, să considerăm că cele două obiecte ce trebuie comparate au forma:

$$\mathbf{x} = ((x_1^1, x_2^1, \dots, x_{k_1}^1), (x_1^2, x_2^2, \dots, x_{k_2}^2), \dots, (x_1^s, x_2^s, \dots, x_{k_s}^s)),$$

$$\mathbf{y} = ((y_1^1, y_2^1, \dots, y_{k_1}^1), (y_1^2, y_2^2, \dots, y_{k_2}^2), \dots, (y_1^s, y_2^s, \dots, y_{k_s}^s)),$$

unde există s tipuri diferite de date, de dimensiuni k_1, k_2, \dots, k_s . Mai întâi vom ierarhiza cele s secvențe în ordinea importanței lor contextuale, ponderându-le astădat cu valorile $\alpha_j, \sum_j \alpha_j = 1$. Apoi,

vom aplica pentru fiecare secvență în parte un anumit tip de măsură de similaritate d_j , specifică cazului. Dacă vom nota secvențele componente cu:

$$x_j = (x_1^j, x_2^j, \dots, x_{k_j}^j),$$

$$y_j = (y_1^j, y_2^j, \dots, y_{k_j}^j),$$

atunci măsura de similaritate mixtă și ponderată are forma:

$$d(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^s \alpha_j \cdot d_j(x_j, y_j).$$

3) Standardizarea caracteristicilor (atributelor) reprezintă o problemă importantă atunci când măsurăm distanța între obiecte. În principiu, nu există o soluție general valabilă a acestei probleme, ci totul depinde de specificul situației. De exemplu, putem avea

transformarea $x_i \rightarrow \frac{x_i - \bar{x}_i}{SD(x_i)}$, astfel încât fiecare componentă are

media zero și dispersia egală cu unitatea, sau transformarea

$x_i \rightarrow \frac{x_i - \bar{x}}{SD(x)}$, astfel încât fiecare vector are media zero și dispersia

unu.

4) Am arătat mai sus că trebuie ținut seama de natura datelor în alegerea unei măsuri de similaritate corespunzătoare. Astfel, în cazul datelor ordonale, menționăm următoarele două abordări clasice:

- Convertirea în date numerice, considerând o scală normalizată $[0, 1]$, unde $min = 0$, $max = 1$, restul valorilor fiind interpolate (e.g. *foarte mic* = 0.0, *mic* = 0.2, *micuț* = 0.3, *mediu* = 0.5, *mare* = 0.7, *foarte mare* = 1);
- Utilizarea unei matrice (simetrice) de similaritate, unde pe diagonală avem 1 (similaritate) sau 0 (disimilaritate), ca de exemplu:

	<i>foarte mic</i>	<i>mic</i>	<i>micuț</i>	<i>mediu</i>	<i>mare</i>	<i>foarte mare</i>
<i>foarte mic</i>	1	0.8	0.7	0.5	0.2	0
<i>mic</i>	0.8	1	0.9	0.7	0.3	0.1
<i>micuț</i>	0.7	0.9	1	0.7	0.3	0.2
<i>mediu</i>	0.5	0.7	0.7	1	0.5	0.3
<i>mare</i>	0.2	0.3	0.3	0.5	1	0.8
<i>foarte mare</i>	0	0.1	0.2	0.3	0.8	1

În cazul datelor nominale se consideră de obicei reguli binare (e.g. IF $x_i = y_i$ for all i , THEN similarity = 1, ELSE 0), o proprietate semantică de bază (e.g. $d(\text{lemn}, \text{fier}) = \alpha \cdot |\text{densitate}(\text{lemn}) - \text{densitate}(\text{fier})|$, $d(\text{Ioana}, \text{Maria}) = \alpha \cdot |\text{vârstă}(\text{Ioana}) - \text{vârstă}(\text{Maria})|$, $d(\text{Craiova}, \text{Cluj}) = \alpha \cdot |\text{populație}(\text{Craiova}) - \text{populație}(\text{Cluj})|$ sau $d(\text{Craiova}, \text{Cluj}) = \alpha \cdot |\text{buget}(\text{Craiova}) - \text{buget}(\text{Cluj})|$ etc.), sau o anumită matrice de similaritate.

5.7.1. Clustering ierarhic

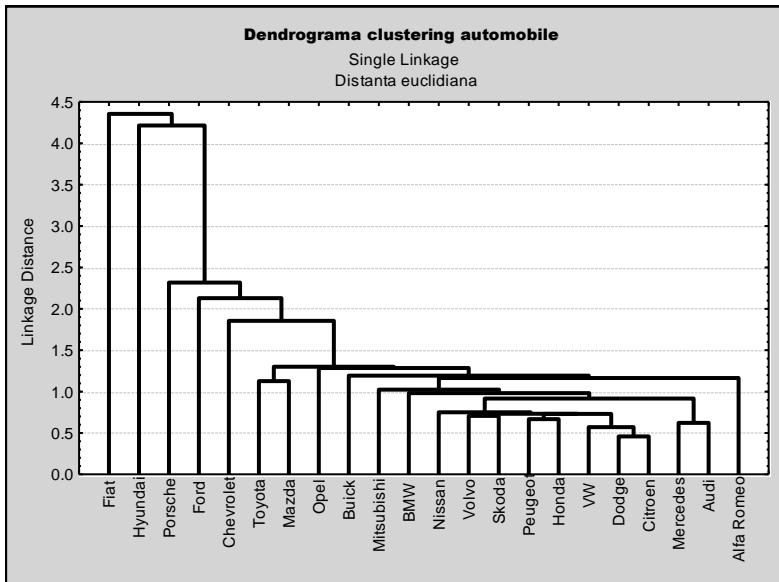
Odată aleasă o distanță de similaritate, pe baza căreia putem compara obiectele, mulțimea acestora poate fi partionată, utilizând o anumită metodologie. Așa cum am spus la început, rezultatul va putea fi reprezentat, în manieră clasică, sub forma unei dendrograme (structură arborescentă). Diferența între cele două metode principale de clustering ierarhic constă în direcția de construcție a arborelui: de jos în sus – clusteringul aglomerativ – și de sus în jos – clusteringul diviziv.

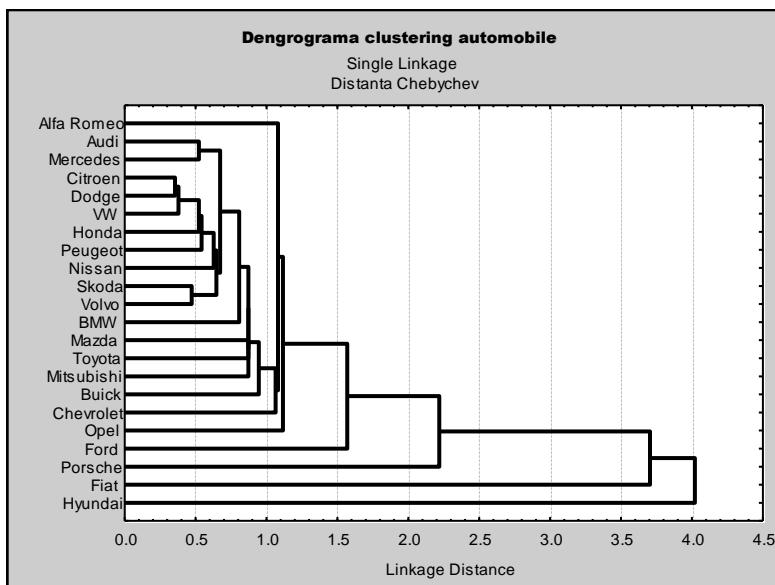
Modelul de clustering ierarhic rezidă în gruparea obiectelor în mod iterativ, utilizând o anumită metodă de „comasare/articulare” (*amalgamation/linkage* -vezi și modul de definire a distanței între clustere) ca, de exemplu:

- *single linkage* (nearest neighbor) – distanța dintre clustere este determinată de distanță între cele mai apropiate obiecte (*nearest neighbor*);

- *complete linkage* (furthest neighbor) – distanța dintre clustere este determinată de distanța între cele mai îndepărtate obiecte (*furthest neighbor*);
- *average linkage* (unweighted pair-group average) – distanța dintre clustere reprezintă media distanțelor (neponderată) între orice două perechi de obiecte din clustere diferite. Să notăm și varianta *weighted pair-group average* care, spre deosebire de cea neponderată, ia în considerație mărimea clusterelor (i.e. numărul obiectelor din fiecare cluster) ca ponderi în calcularea distanțelor;
- *centroid method* (unweighted pair-group centroid) – distanța între clustere este determinată de distanța între centroizii lor (centrele de greutate). Să notăm și varianta ponderată a acestei metode: *weighted centroid method* (weighted pair-group centroid (median));
- *Ward's method*, 1963 – această metodă, diferită de cele de mai sus, utilizează analiza varianțelor (dispersiilor) în evaluarea distanțelor între clustere;

Ilustrăm în figura de mai jos, schematic, metoda clusteringului ierarhic în cazul partiționării unor mărci de automobil, ținând cont de următoarele caracteristici (attribute): *Pret*, *Acceleratie*, *Frâne*, *Conducere* și *Kilometraj*.





Notă: Prima dendrogramă este sub forma unui arbore vertical, distanța de similaritate este Euclidiană, iar metoda este *single linkage*. În al doilea caz, distanța aleasă a fost Chebychev, iar metoda a rămas tot *single linkage*, cu arbore pe orizontală.

5.7.2. Clustering neierarhic

Clusteringul neierarhic sau partititional este mai cunoscut sub denumirea de clustering de tip *k-means*. Acest model este total diferit de cel ierarhic, metoda presupunând cunoașterea *a priori* a numărului de clustere. Problema care se pune în acest context este aceea de a crea un algoritm pe baza căruia să se poată forma exact atâtea clustere cât s-a decis inițial, cât mai distințe cu putință. Cu toate că, în principiu, metoda *k-means* clustering produce exact k clustere care divizează mulțimea inițială de obiecte cât mai distinct posibil, rămâne deschisă problema estimării numărului optim de clustere care să conducă la separarea cea mai bună a obiectelor. Vom menționa doar că metoda cel mai des utilizată pentru rezolvarea acestei chestiuni este algoritmul *v-fold cross-validation* pentru determinarea automată a numărului de clustere în mulțimea de date.

Din punct de vedere computațional, putem privi această metodă ca inversul metodei analizei varianțelor (dispersiilor) ANOVA din cadrul Statisticii. Astfel, modelul începe cu alegerea aleatoare a k clustere, după care aranjează obiectele în cele k clustere, având două obiective de îndeplinit:

- ◆ Minimizarea variabilității intra-cluster;
- ◆ Maximizarea variabilității inter-clustere.

În timp ce în cadrul metodei ANOVA, testul de semnificație evaluează variabilitatea dintre grupuri vs. variabilitatea din interiorul grupului atunci când se testează ipoteza că mediile din grup sunt diferite între ele, aici modelul „mută” obiectele în interiorul sau în exteriorul grupurilor (clusterelor) pentru a obține cea mai semnificativă partiție.

Prezentăm, în continuare, schema unui algoritm de tip *k-means* (în lb. Engleză in original):

Algorithm k-means

1. Select k points at random as cluster centers.
2. Assign instances to their closest cluster center according to some similarity distance function.
3. Calculate the centroid or mean of all instances in each cluster (this is the *mean* part).
4. Cluster the data into k groups where k is predefined.
5. GOTO the step 3. Continue until the same points are assigned to each cluster in consecutive rounds.

Tehnic vorbind, pașii algoritmului sunt următorii (în lb. Engleză in original):

- Suppose there are N data points $\mathbf{x}^l = (x_1, x_2, \dots, x_n)$ in total;
- Find a set of k representative vectors \mathbf{c}_j , where $j = 1, 2, \dots, k$;
- Partition the data points into k disjoint subsets S_j , containing N_j data points, in such a way as to minimize the sum-of-squares clustering function given by:

$$J = \sum_{j=1}^k \sum_{l \in S_j} \left\| \mathbf{x}^l - \mathbf{c}_j \right\|^2,$$

where \mathbf{c}_j is the mean of the data points in set S_j , given by:

$$\mathbf{c}_j = \frac{\sum_{l \in S_j} \mathbf{x}^l}{N_j}.$$

Notă. Există o problemă majoră în utilizarea metodei *k-means*, și anume necesitatea definirii mediei, ceea ce implică faptul că este neaplicabilă la date nenumerice (e.g. categoriale).

Remarcă: Menționăm, în context, încă trei modele de clustering:

- EM (*Expectation Maximization*) clustering;
- QT (*Quality Threshold*) clustering;
- Fuzzy *c*-means clustering.

În încheiere să punctăm „bilele albe” și „bilele negre” ale celor două mari tipuri de clustering.

- *Cluster ierarhic:*
 - (+) Intuitiv, ușor de înțeles;
 - (-) Persistența nedorită a clusterelor de la început;
 - (-) Sensibil la valori extreme (outliers) și la atribute irelevante;
 - (-) Neaplicabil la mulțimi mari de date.
- *Cluster partitional:*
 - (+) Mai puțin sensibil la valori extreme (outliers) și la atribute irelevante;
 - (+) Aplicabil la mulțimi mari și foarte mari de date;
 - (-) Are nevoie de alegerea prealabilă a numărului de clustere, ceea ce se dovedește a fi o sarcină nu tocmai ușoară (vezi și comentariile referitoare la estimarea numărului de clustere)

Judecând aspectele prezentate mai sus, ajungem la concluzia că este mai înțeleaptă utilizarea, dacă este posibil, a ambelor variante de clustering, simultan. De exemplu, putem utiliza clusteringul ierarhic pentru stabilirea unui număr convenabil de clustere, după care utilizarea celui neierarhic va îmbunătății rezultatele prin reconfigurarea clusterelor irelevante.

Pentru mai multe amănunte privind tehniciile de clustering, vezi și [134].

Exemple:

1) Prezentăm o aplicație a algoritmului *k-means* în creșterea performanței unui algoritm de tip PNN (rețea neuronală probabilistă), utilizat în diagnosticul cancerului hepatic [75]. Astfel, plecându-se de la un lot de 299 subiecți, împărțiți după cum urmează:

- ◆ 60 pacienți cu hepatită cronică;

- ◆ 179 pacienți cu ciroză hepatică;
- ◆ 30 pacienți cu cancer hepatic;
- ◆ 30 persoane sănătoase (grup martor),

a fost aplicat mai întâi algoritmul PNN pentru clasificarea persoanelor în cele patru categorii (de diagnostic), utilizându-se lotul inițial. Pe baza acestei clasificări se poate stabili profilul standard al bolnavului și, pe de altă parte, se poate crea un program de diagnoză care poate sprijini medicul în stabilirea diagnosticului optim. Menționăm că pentru clasificare s-a folosit un set de 15 date semnificative din punctul de vedere al diagnosticului (vezi și exemplul corespunzător din paragraful 5.3.3. despre aplicații PNN). Apoi a fost utilizat algoritmul *k-means* pentru reducerea numărului de pacienți necesari analizei, în vederea creșterii vitezei de procesare a rețelei neuronale. În acest sens, a fost ales un număr de: 250, 200, 150 și, în fine 100 de clustere. Astfel, în loc să fie clasificați toți cei 299 de pacienți, s-a ajuns la clasificarea unui număr de 250, 200, 150 sau 100 de pacienți. Practic, spre deosebire de cazul folosirii lotului inițial de 299 de pacienți (reali), fiecare din cele N clustere ($N = 250, 200, 150, 100$) au fost reprezentate de centroizii respectivi, priviți ca „pacienți virtuali”. Pentru căutarea parametrului σ al PNN s-au folosit atât algoritmi genetici (GA) cât și metoda Monte Carlo (MC). Rezultatele acestui experiment sunt prezentate în tabelul de mai jos.

Număr de pacienți (reali sau virtuali)	Acuratețe antrenament (%)		Acuratețe testare (%)	
	GA	MC	GA	MC
299	87	89	82	86
250	88	90	80	83
200	91	92	76	79
150	93	93	74	74
100	94	95	68	67

Din tabelul de mai sus observăm că reducerea efortului computațional (deci și creșterea corespunzătoare a vitezei de procesare) cu 16% (250 pacienți), nu a implicat o scădere semnificativă a acurateții clasificării. Pe de altă parte, atunci când avem mai puțini pacienți (virtuali) de clasificat, crește acuratețea (aparentă) la antrenament și scade acuratețea (reală) la testare. Spunem aparentă, deoarece nu se mai procesează persoane reale ci persoane „fictive” – centroizii clusterelor obținute. Se observă că o reducere cu până la 33% (200 pacienți) este încă rezonabilă din punct de vedere practic.

2) Dacă exemplul de mai sus ilustrează aplicabilitatea tehnicii clusteringului în domeniul diagnosticului medical asistat de computer, exemplul următor va ilustra aplicabilitatea sa în domeniul tratamentelor, mai concret pentru eficientizarea tratamentului pacienților în cazul oncologiei [80]. Astfel, datele

se referă la un număr de 75 de pacienți și patru tipologii clasice de tratament în domeniul oncologic:

- ◆ (C₁) *chimioterapie*(CT);
- ◆ (C₂) *chimioterapie*(CT)+*hormonoterapie*(HT),
- ◆ (C₃) *chimioterapie*(CT)+*radioterapie*(RT)+*curieterapie*(QT);
- ◆ (C₄) *chimioterapie*(CT)+*radioterapie*(RT)+*curieterapie*(QT)+
+*hormonoterapie* (HT)

În acest context, pentru estimarea tratamentului optim, fiecare pacient a fost echivalat cu un vector cu patru dimensiuni, primele trei reprezentând atrbute predictive specifice domeniului: diametrul mediu al tumorilor, vârsta și stadiul bolii, iar ultimul reprezentând tipul de tratament (decizia). În vederea aplicării metodei *k-means*, a fost ales un număr de 7 clustere. Acuratețea generală obținută a fost satisfăcătoare pentru această alegere a numărului clusterelor, optimizarea ei depinzând, evident, de numărul prestatibilității acestora. Prin această metodă de segmentare a pacienților după natura tratamentului, se obține o metodologie de a corela specificitatea pacientului cu tipul de tratament, ajutând medicul în procesul de stabilire a tratamentului adecvat fiecărui pacient.

5.8. Algoritmi genetici

Simulările evolutive cu ajutorul computerului se pare că au fost inițiate în anul 1954 de către Barricelli, odată cu simularea evoluției unui automat ce joacă un joc de cărți [9]. Mai departe, în anul 1957, Fraser se ocupă cu simularea cu ajutorul computerelor existente la acea vreme a unor sisteme genetice [52]. Mai putem aminti pentru această perioadă de pionierat pe Box, 1957, cu cercetări privind optimizarea productivității industriale [19] și pe Bremermann, 1962 [22], cu optimizări cu ajutorul evoluției și recombinării. Plecând de la acești primi pași inițiali, algoritmii genetici au fost recunoscuți ca un domeniu aparte în contextul metodelor de optimizare, odată cu cercetările din anii 60' întreprinse de către John Holland și colegii săi la Universitatea Michigan în domeniul automatelor celulare (*cellular automata*), fundamentarea venind odată cu publicarea cărții sale „*Adaptation in Natural and Artificial Systems*” în anul 1975 [99]. Menționăm, în context, și extinderea lor datorată lui Goldberg (1989), [59], [61]. Dezvoltarea, mai mult teoretică, a algoritmilor genetici a continuat după aceea, culminând cu „*The First International Conference on Genetic Algorithms*”, ținută în anul 1985 la Universitatea Illinois, după care, simultan cu creșterea exponențială a puterii de calcul, algoritmii genetici au cunoscut o reală dezvoltare practică, amintind aici primul pachet software comercial pentru computere personale -*Evolver* (Axcelis, Inc./ Palisade Corporation)-, apărut în anul 1989.

Algoritmii genetici se bazează pe teoria modernă a evoluției, având ca rădăcini atât principiul selecției naturale, statuat de către Darwin în a sa celebră carte „*Origin of species*” (1859), în care afirmă că la baza evoluției stă selecția naturală, cât și genetica lui Mendel (1865), care a relevat faptul că factorii ereditari care se transferă de la părinți la copii au caracter discret („*Versuche über Pflanzenhybride -Experimente în hibridizarea plantelor*”, lucrare prezentată la 8 februarie și 8 martie 1865 la Brunn Natural History Society).

Algoritmii genetici (GA) reprezintă o tehnică de identificare de soluții de aproximare pentru problemele de optimizare și căutare, fiind o clasă particulară de algoritmi evolutivi. Ei se regăsesc în domeniul mai larg al heuristicilor de căutare globală (*global search heuristics*) și pot fi priviți, în principiu, ca:

- ◆ rezolvitori de probleme;
- ◆ bază competentă pentru învățarea automată;
- ◆ modele computaționale de inovare și creativitate;
- ◆ filosofie computațională.

Un algoritm genetic, privit ca cel mai popular caz din categoria algoritmilor evolutivi (*evolutionary algorithms* -EA), reprezintă un algoritm de optimizare metaheuristică bazat pe o populație (de soluții potențiale) și care utilizează mecanisme specifice, inspirate din evoluția biologică ca, de exemplu: reproducere, mutație, recombinare, selecție, supraviețuire a celui mai bun exemplar. Reamintim aici câțiva termeni de bază din domeniul evoluției biologice, dintre care unii se regăsesc în vocabularul EA:

- *Cromozomii*, reprezentând purtătorii informației genetice. Cromozomii sunt structuri liniare având drept componente *gene*, care poartă caracteristicile ereditare ale părinților (unități funcționale ale eredității, codând caracteristicile fenotipice).
- *Fenotipul* unui individ, reprezentând fie imaginea/constituția sa fizică, fie manifestarea specifică a unei anumite caracteristici (culoarea ochilor, înălțimea etc.).
- *Genotipul*, reprezentând structura genetică specifică unui individ, sub forma ADN. Împreună cu variația externă, el codează fenotipul individual. Din punct de vedere genetic, fiecare individ este o entitate duală: genotipul său codează fenotipul, ecuația ce le leagă fiind simplă: *genotipul + mediu + variații aleatoare → fenotip*.
- *Selecția naturală*, reprezentând procesul prin care un individ cu caracteristici favorabile are mai multe șanse să supraviețuască și să se reproducă. Dat fiind un anumit mediu biologic, care poate susține

viața unui număr limitat de indivizi, și instinctul natural de reproducere, selecția naturală devine inevitabilă, favorizând indivizii cei mai adaptați mediului respectiv.

- *Evoluția*, reprezentând schimbarea în caracteristicile ereditare ale unei populații de-a lungul unor generații succesive. Este un proces operator asupra cromozomilor și se realizează prin intermediul reproducерii.

Ideea care stă la baza EA este simplă, indiferent de tipul lor: fiind dată o populație de indivizi, presiunea mediului înconjurător, care nu poate susține decât un număr limitat dintre ei, va implica o selecție a acestora, care va implica o creștere a adaptării lor la condițiile mediului. Fiind dată o funcție de performanță care trebuie maximizată, se creează în mod aleator un set de soluții candidat (i.e. elemente ale domeniului funcției), după care se aplică funcția de performanță, ca o măsură abstractă de potrivire (adecvare - *fitness*). Mai departe, pe baza acestei adecvări (cel mai mare = cel mai bun), sunt alesi unii dintre cei mai buni candidați pentru a se obține o nouă generație, aplicând recombinarea și/sau mutația asupra lor. Recombinarea este un operator care se aplică la doi sau mai mulți candidați selectați (,părinți'), rezultând unul sau mai mulți noi candidați (,copii'). Mutarea se aplică unui singur candidat, rezultând un nou candidat. Pe baza acestor mecanisme de sorginte biologică, se obține un set de noi candidați care vor intra în competiție cu cei vechi pentru a ocupa un loc în noua generație. Acest proces de inspirație biologică va fi iterat până când un candidat suficient de ,performant' va fi obținut sau o anumită condiție de STOP va fi îndeplinită.

EA sunt de mai multe tipuri, cu toate că toți au la bază ideea expusă mai sus, diferențele constând în anumite detalii tehnice de implementare, istoria apariției și tipul problemelor la care se aplică. Prezentăm, mai jos, tipurile principale de EA.

- *Algoritmii genetici*, reprezentând, aşa cum am spus mai înainte, tipul cel mai popular de EA. În cazul GA, se caută soluția candidat a problemei sub forma unei secvențe (*string*) finite de numere (i.e. secvență peste un alfabet finit). În mod tradițional, în cazul GA componentele sunt binare, cu toate că se utilizează acea reprezentare care reflectă cel mai bine natura problemei. Acest tip de EA se utilizează mai ales în probleme de optimizare (combinatorică), rolul recombinării fiind de operator primar de variație, rolul mutației fiind de operator secundar de variație, selecția ,părinților' fiind aleatorie, deplasată în funcție de adecvare, în timp ce selecția ,supraviețuitorilor' se face utilizând diferite metode (e.g. înlocuirea bazată pe vârstă, înlocuirea aleatoare, înlocuirea bazată pe adecvare, înlocuirea elitistă, înlocuirea celui mai puțin performant etc.).

- *Programarea evolutivă* (EP), pentru care soluția candidat se caută sub forma unor mașini descrise de automate finite (doar structura programului fiind fixată, parametrii evoluând). EP se aplică în probleme de optimizare, recombinarea nefiind utilizată niciodată, mutația reprezentând singurul operator de variație folosit, fiecare „părinte” creând un „copil”, iar selecția „supraviețuitorilor” se face aleator, deplasată în funcție de adecvare.
- *Programarea genetică* (GP), pentru care soluția candidat se caută sub forma unui arbore (program), adecvarea fiind determinată de abilitatea acestor programe de a rezolva problema în cauză. GP se utilizează în probleme de modelare, operatorii de variație utilizăți fiind specifici arborilor (e.g. mutație de micșorare/creștere, încruțișare de tip nod/sub-arbore/mixtă, clonare), selecția „părinților”/„supraviețuitorilor” fiind aleatoare, deplasată în funcție de adecvare.
- *Strategii evolutive* (ES), pentru care soluția candidat se caută sub forma unor vectori având componente numere reale. ES se aplică la probleme de optimizare continuă, rolul recombinării fiind important, dar secundar în același timp, rolul mutației fiind de asemenea important, uneori reprezentând unicul operator de variație, selecția „părinților” fiind aleatorie, uniformă, în timp ce selecția „supraviețuitorilor” este deterministă, deplasată în funcție de adecvare.
- *Sisteme (evolutive) de clasificare bazate pe învățare* (LCS - *learning classifier system*), care reprezintă un mecanism de învățare automată, strâns legat de învățarea consolidată și EA. În cazul LCS, avem o populație de reguli de tipul *if... then* în logica de ordinul I, dintre care se selectează cele mai bune reguli, pe baza metodelor de calcul evolutiv (GA acționează ca o componentă a descoperirii regului). LCS se utilizează cu predilecție în aplicațiile în care obiectivul este evoluarea unui sistem care „va răspunde” la starea curentă a mediului său prin sugerarea unui răspuns care maximizează cumva o recompensă (viitoare) provenită din mediu. Exemple de aplicații LCS: modelarea comportamentului de pe piețele financiare (sistemul XCS), rezolvarea de probleme de tip *multistep* (e.g. evoluarea unui sistem de control pentru un robot online (sistemul ZCS)) etc. Reprezentarea regulelor poate fi îmbunătățită prin utilizarea unei populații de rețele neuronale sau alte metode.

Așa după cum s-a observat din rândurile de mai sus, în acest domeniu există două componente fundamentale care stau la baza unui sistem evolutiv:

- ◆ Operatorii de variație (recombinare și mutație);
- ◆ Procesul de selecție

Schema generală a unui EA poate fi sintetizată în următorul algoritm pseudo-cod (în lb. Engleză, în original) [45]:

BEGIN

INITIALISE population with random candidate solutions;
EVALUATE each candidate;
REPEAT UNTIL (TERMINATION CONDITION is satisfied)

1. *SELECT* parents;
2. *RECOMBINE* pairs of parents;
3. *MUTATE* the resulting offspring;
4. *EVALUATE* new candidates;
5. *SELECT* individuals for the next generation;

END REPEAT

END

Încheiem această scurtă introducere în domeniul EA, citând următoarea aserțiune, deosebit de sugestivă: „*Computer programs that "evolve" in ways that resemble natural selection can solve complex problems even their creators do not fully understand*” (John.H. Holland -<http://www.econ.iastate.edu/tesfatsi/holland.GAIintro.htm>).

5.8.1. Componentele unui algoritm genetic

Vom prezenta, succint, principalele componente ale arhitecturii unui GA, care, evident, se regăsesc la EA în general, și despre care am amintit în introducerea de mai sus:

- Reprezentarea (definirea indivizilor);
- Funcția de performanță (evaluare/adecvare -*fitness function*);
- Populația:
 - Mecanismul de selecție a părinților;
 - Mecanismul de selecție a supraviețuitorilor.
- Operatorii de variație;
- Parametrii algoritmului;

A. Reprezentarea (definirea indivizilor)

Vom începe prin a defini ‘universul’ GA, plecând de la dualitatea entității *soluție/individ*. Astfel, în contextul problemei ce trebuie rezolvată – lumea reală – soluția candidat este echivalentă cu fenotipul, reprezentând un punct în spațiul soluțiilor posibile, adică spațiul fenotipului. Pe de altă parte, în universul GA, soluția posibilă/candidat, numită *cromozom*, este echivalentă cu genotipul, reprezentând un punct în spațiul de căutare evolutivă, adică spațiul genotipului. Termenul de *reprezentare* se referă, în principiu, fie la aplicația din spațiul fenotip în spațiul genotip –codare, fie la aplicația inversă –decodare, cu toate că se poate referi și la structura datelor din spațiul genotip.

Primul tip de reprezentare în cadrul GA este cel *binar* (genotipul constă într-o secvență binară – *bit string* – e.g. 10011; 11000; 10000 etc.), un exemplu de aplicație practică fiind cel al problemelor de decizie booleană. Cu toate că este cea mai simplă reprezentare posibilă, ea nu poate acoperi în mod eficient decât o arie restrânsă de probleme. Din această cauză, următorul pas în ierarhia tipurilor de reprezentare este cel al reprezentărilor *întregi*, care acoperă o arie mult mai mare de aplicații, de exemplu în găsirea unor valori optime în cazul unor variabile cu valori întregi (e.g. {1, 2, 3, 4} etc.). Totuși, cele mai multe probleme în care se pot aplica GA se referă la cazurile în care variabilele iau valori reale. În acest caz, genotipul unei soluții candidate va fi reprezentat printr-un vector de componente reale $\mathbf{x} = (x_1, x_2, \dots, x_p)$, fiecare genă x_k fiind un număr real – reprezentarea *reală*, sau virgulă mobilă (*floating point*) – e.g. (0.1, 2.3, 4.1, 7.5) etc.

În cazul în care trebuie decisă ordinea în care o secvență de evenimente poate apărea, reprezentarea aleasă va fi aceea a unei permutări de întregi, ca în care, spre deosebire de reprezentarea cu întregi, un număr nu poate apărea decât o singură dată (e.g. [1, 2, 3, 4], [2, 1, 4, 3] etc.). Trebuie menționat aici că alegerea permutărilor depinde de tipul problemei de rezolvat, în sensul că sunt probleme în care ordinea este importantă intrinsec, altele în care aceasta nu este. De exemplu, dacă considerăm problema optimizării unui proces de producție, și un cromozom conține 4 gene, care reprezintă componentele produsului respectiv, atunci este importantă ordinea producerii acestora în vederea asamblării, deci putem avea evaluări diferite pentru cromozomii [1, 2, 3, 4] sau [2, 1, 4, 3]. În alte cazuri, ordinea poate să nu fie la fel de importantă, ca în cazul bine-cunoscutei probleme a comis-voiajorului (*travelling salesman* -TSP), în care putem considera că punctul de plecare nu este important, deci secvențe de orașe ca [C, A, B, D] și [A, D, B, C] sunt echivalente.

Initializarea (initialisation) este un proces simplu în cadrul GA, practic creându-se o primă populație (populația *initială*) de cromozomi, fiecare dintre aceștia având o reprezentare binară, întreagă, reală etc. De obicei, populația inițială este generată aleatoriu, anumite heuristici putând fi alese la acest pas, pentru ca aceasta să corespundă cât mai bine problemei de rezolvat – soluții alese din regiuni în care este probabil ca soluțiile optimale să apară (vezi

și [45]). Mărimea populației depinde de natura problemei, mergând de la câteva sute până la câteva mii de cromozomi (soluții posibile).

Dacă inițializarea reprezintă începutul construcției unui algoritm genetic, finalul procesului este determinat de condiția de terminare (*termination condition*). În principiu, se pot considera două condiții de terminare:

- Cazul, foarte rar întâlnit, în care problema are un nivel optim de adecvare, cunoscut (provenind, de exemplu, de la cunoașterea optimului funcției obiectiv), caz în care atingerea acestui nivel implică oprirea procesului;
- Cazul uzual, bazat pe caracteristica stochastică a GA, ce implică negarantarea atingerii optimului, caz în care se aleg diferite condiții de terminare, dintre care menționăm:
 - Expirarea timpului alocat rulării algoritmului;
 - Atingerea unei anumite limite de către numărul total de evaluări;
 - Îmbunătățirea adecvării este plafonată o perioadă de timp sau un număr de generații;
 - Diversitatea populației este plafonată de un anumit nivel dat.
 - Inspecția manuală.
 - Combinarea celor de mai sus.

În concluzie, terminarea procesului se datorează fie atingerii optimului (dacă acesta este cunoscut, ceea ce nu este, din nefericire, cazul pentru marea majoritate a problemelor), fie se impune o anumită condiție de STOP.

B. Funcția de evaluare (potrivire, adecvare, performanță)

Odată reprezentarea soluțiilor aleasă și populația inițială definită, se pune problema evaluării calității indivizilor. În condițiile naturale, biologice, acest rol este asumat de către mediul înconjurător, care „selectează” cei mai buni indivizi pentru supraviețuire și reproducere. În universul GA, acest rol este asumat de către funcția de evaluare (*fitness function*), care reprezintă un tip particular de funcție obiectiv, având rolul de a cuantifica gradul de optimalitate a unei soluții (cromozom). Ea reprezintă baza selecției, generând, în consecință, „îmbunătățirile” ulterioare ale indivizilor. Din acest motiv, funcția de evaluare depinde direct de problema ce trebuie rezolvată. Deoarece în multe probleme este dificil de definit această funcție, s-a dezvoltat și domeniul conex al *algoritmilor genetici interactivi*, în care este utilizată și evaluarea umană (e.g. imagini, muzică etc.). Să notăm că funcția de evaluare este nenegativă și, în ipoteza că populația evaluată conține n indivizi $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, se definește

performanța totală (*total fitness*) prin formula $F = \sum_{i=1}^n f(x_i)$. Vom remarcă, în încheiere, că problema alegerii unei funcții de evaluare eficientă pentru problema dată devine dificilă în cazul problemelor de optimizare multicriterială (*multicriterion/multiobjective optimisation*), în care se consideră, de exemplu, optimalitatea Pareto, sau în cele cu constrângeri în care se consideră, de exemplu, funcțiile de penalitate.

Vom ilustra alegerea funcției de evaluare cu două exemple clasice [59], [120], deosebit de simple. În primul caz, trebuie maximizată funcția obiectiv $f : [0, 31] \rightarrow R_+$, dată de $f(x) = x^2$, utilizându-se o reprezentare binară cu 5 biți (e.g. pentru $x_1 = 13$, cromozomul corespunzător are forma 01101, pentru $x_2 = 8$, cromozomul corespunzător are forma 01000, iar pentru $x_3 = 19$, cromozomul corespunzător are forma 10011), alegându-se ca funcție de evaluare chiar funcția obiectiv. Evident, se alege cromozomul x_3 , deoarece evaluarea „returnează” cea mai mare valoare. În al doilea caz, se consideră funcția obiectiv $f : [-1, 2] \rightarrow R$, dată de $f(x) = x \cdot \sin(10 \cdot \pi \cdot x) + 1$, care trebuie de asemenea maximizată, considerându-se o reprezentare binară cu 22 de biți (e.g. $x_1 = 0,637197 \sim 1000101110110101000111$; $x_2 = -0,958973 \sim 000000111000000001000$; $x_3 = 1,627888 \sim 1110000000111111000101$). La fel ca mai înainte, se consideră funcția de evaluare egală cu funcția obiectiv. După evaluarea celor trei cromozomi, cel mai bun se dovedește a fi x_3 , care „returnează” cea mai mare valoare.

C. Populația

Rolul *populației* este acela de a înmagazina reprezentările tuturor soluțiilor posibile ale unei probleme, ea fiind o mulțime de copii multiple de genotipuri. În timp ce indivizii sunt obiecte statice, ce nu se schimbă (adaptează) pe parcursul timpului, populația este nucleul evoluției, ea „evoluând” în timp. Diversitatea regăsită într-o populație ilustrează numărul de soluții diferite prezente la un moment dat, cunoscându-se diferite măsuri pentru ea (e.g. numărul valorilor de performanță diferite, numărul fenotipurilor/genotipurilor diferite, entropie etc.).

Definirea unei populații poate să fie deosebit de simplă –doar specificarea numărului indivizilor compoziți, sau mult mai complicată –definirea unei structuri spațiale, relativă la o distanță sau o relație de vecinătate între indivizi. În timp ce operatorii de variație acționează asupra unui individ sau asupra a doi indivizi (componentele unei populații), operatorii de selecție (selectarea părinților și a supraviețuitorilor) acționează asupra populației, luând în considerație întreaga populație curentă, operațiunile executându-se asupra a ceea ce există la un moment dat (e.g. cei mai performanți/cei mai neperformanți indivizi ai actualei populații sunt aleși pentru o anumită operație (obținerea unei noi generații/înlocuire)).

O populație numeroasă este utilă în explorarea spațiului soluțiilor posibile, mai ales atunci când suntem interesați mai mult de optimul global decât de cele locale. În același timp însă, o populație mare implică costuri mai mari în timp și memorie. Să mai amintim aici că, de cele mai multe ori, mărimea populației rămâne constantă pe tot parcursul evoluției, o alegere de aproximativ 100 de indivizi fiind ușuală în multe aplicații.

C₁. Mecanismul de selecție a părinților

Unul dintre principalele aspecte privind „evoluția” ce caracterizează GA se referă la procesul de selecție a „părinților”/„perechilor” (*parent/mating selection*), permisând celor mai performanți indivizi din populația actuală să genereze noua populație. Prin „părinte” înțelegem un individ din actuala populație, care a fost selectat pentru a fi transformat (prin operatorii de variație) în vederea creării noi generații (descendenților). Împreună cu mecanismul de selecție a supraviețuitorilor, selecția părinților reprezintă „motorul” evoluției indivizilor de la o generație la alta. În principiu, selecția părinților se face de manieră probabilistă, permisând celor mai performanți indivizi să fie implicați în producerea noii generații. Fără a intra în detalii, prezentăm, în continuare, câteva dintre cele mai cunoscute tipuri de selecție a părinților:

- Selecția proporțională (*fitness proportional selection -FPS*), în care probabilitatea ca un individ x_i să fie selectat este $p_i = \frac{f(x_i)}{F}$;
- Selecția prin ordonare (*ranking selection*) se bazează pe sortarea populației pe baza valorilor obținute la evaluare și alocarea unei probabilități de selecție fiecărui individ în concordanță cu rangul său în ierarhia obținută prin sortare. Modul de trecere de la rangul individului în ierarhie la probabilitatea de selecție este arbitrar, putând fi realizat, de exemplu, prin aplicații liniare sau exponențiale (*linear/exponential ranking*);
- Selecția turnir (*tournament selection*) se utilizează mai ales în cazuri în care este dificil de obținut informație despre întreaga populație (e.g. populație foarte numeroasă). În acest caz, selecția de tip turnir ia în considerație competitorii (de unde și termenul de selecție prin competiție sau concurs), nefiind necesară informația globală despre populație, ci doar compararea directă a oricăror doi cromozomi (turnir binar) și selectarea celui mai performant. În general însă, se poate considera un număr mai mare de competitori deodată, de exemplu k , definind astfel și mărimea turnirului (*tournament size*). Un pseudo-cod pentru acest tip de selecție a n părinți are forma următoare [45]:

```

BEGIN

    set current_member = 1;
    WHILE (current_member ≤ n) DO
        Pick k individuals randomly, with or without replacement;
        Select the best of these k comparing their fitness values;
        Denote this individual as i;
        set mating_pool [current_member] = i;
        set current_member = current_member + 1;

    END WHILE
END

```

Probabilitatea p de selecție a celui mai performant competitor al concursului se poate alege în mod determinist, și anume $p = 1$ (turnir determinist), dar există și versiunea probabilistă cu $p < 1$ (turnir stochastic).

- Alte selecții (e.g. ruleta (*Monte Carlo*), selecția universală stochastică (*stochastic universal sampling* –SUS), selecția elitistă etc.) pot fi luate în considerație. Prezentăm mai jos pseudocodurile pentru selecțiile de tip ruletă și SUS [45]. Pentru aceasta, presupunând că există o anumită secvență $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ a întregii populații (obținută prin ordonare sau în mod aleatoriu), se calculează valorile $a_i = \sum_1^i p_i$, unde p_i reprezintă probabilitățile de selecție.

A. *Ruleta Monte Carlo*):

```

BEGIN

    set current_member = 1;
    WHILE (current_member ≤ n) DO
        Pick a random value r uniformly from [0, 1];
        set i = 1;
        WHILE (ai < r) DO
            set i = i + 1;
        END WHILE
        set mating_pool [current_member] = parents [i];
        set current_member = current_member + 1;

    END WHILE
END

```

B. Selecția universală stochastică:

```
BEGIN
    set current_member = i = 1;
    Pick a random value r uniformly from [0, 1/n];
    WHILE (current_member ≤ n) DO
        WHILE (r < a) DO
            set mating_pool [current_member] = parents [];
            set r = r + 1/n;
            set current_member = current_member + 1;
        END WHILE
        set i = i + 1;
    END WHILE
END
```

C₂. Mecanismul de selecție a supraviețuitorilor

Mecanismul de selecție a supraviețuitorilor (înlocuirea), cunoscut și ca selecția datorată mediului, este procesul prin care se diferențiază indivizii pe baza calității lor. În principiu, este similar cu mecanismul de selectare a părinților, dar operează pe alt palier al ciclului evolutiv. Astfel, după selectarea părinților și obținerea descendenților acestora, este nevoie de un mecanism de selecție a celor care vor forma generația următoare, deoarece volumul populației rămâne, în principiu, constant. Decizia în această privință se bazează pe valoarea funcției de evaluare pentru fiecare individ în parte, fiind ușor favorizați cei cu calități superioare, deși se ia în considerație de multe ori și conceptul de vârstă. Să notăm că, spre deosebire de mecanismul de selectare a părinților, care este îndeobște stochastic, mecanismul de selectare a supraviețuitorilor este deseori determinist.

În continuare, vom prezenta câteva strategii de selectare a supraviețuitorilor, propuse de-a lungul timpului.

- Strategia bazată pe vârstă (*age-based strategy*), utilizată în cadrul schemelor simple GA, constă, în principiu, în faptul că orice individ „supraviețuiește” doar un anumit număr de generații al GA, după care este înlocuit (e.g. un ciclu în cazul în care numărul părinților este egal cu cel al descendenților ($n = m$), situație în care se înlocuiesc toți părinții cu urmașii lor). Strategia aceasta se poate implementa și când $m < n$, sau în cazul în care un singur

descendent este creat și introdus în populație în fiecare ciclu. Există și varianta înlocuirii unui părinte, ales în mod aleatoriu;

- Strategia bazată pe evaluare (*fitness-based strategy*) se referă la mecanismul, bazat pe evaluarea indivizilor, prin care se aleg n elemente din totalul populație de $n + m$ părinți și descendenți care vor forma generația următoare, metodele amintite la punctul **C₁** rămânând valabile și în acest caz.

Remarcă: Atunci când se abordează conceptul GA, se vorbește fie despre reprezentarea soluțiilor posibile în vederea construirii unei populații de indivizi (care vor fi modificați în timp, producând descendenți ce vor moșteni o parte din caracteristicile părinților, diferind totuși mai mult sau mai puțin de aceștia, astfel încât noi soluții potențiale vor intra în procesul de evaluare), fie de modul deosebit de supraviețuire al competitorilor în vederea selectării pentru reproducere, pe baza performanțelor lor individuale. Pentru acest al doilea aspect, menționăm două modele populaționale cunoscute în literatură.

- ◆ Modelul *generațional* (*generational model*), în care fiecare generație începe cu o populație numărând n indivizi, din care se selectează o mulțime de reproducere (*mating pool*) ce conține n părinți (numărul părinților este același cu mărimea populației), după care se obțin m ($= n$) descendenți prin aplicarea operatorilor de variație și a evaluării. Astfel, după fiecare generație, întreaga populație este înlocuită cu descendenții, ce vor forma „generația următoare” (*next generation*);
- ◆ Modelul staționar (*steady-state model*), în care, spre deosebire de modelul anterior, nu se mai înlocuiește întreaga populație dintr-o dată, ci doar un număr de m ($< n$) indivizi vechi sunt înlocuiți de m indivizi noi, descendenții; raportul $r = m/n$ se numește rata de înlocuire (*generational gap*). Amintim aici algoritmul GENITOR al lui Whitley [175], [176], în care cei mai puțin performanți m indivizi sunt înlocuiți (de obicei, alegând $m = 1$).

D. Operatorii de variație (recombinare, mutație)

Rolul operatorilor de variație este acela de a crea noi indivizi, plecând de la cei deja existenți. După selectarea, într-un fel sau altul, a indivizilor care vor crea generația următoare, următorul pas în ciclul evolutiv constă în chiar mecanismele prin care aceasta este obținută. Pentru fiecare descendent ce trebuie creat este nevoie fie de o pereche de „părinți” ce au fost în prealabil selectați pentru „reproducere”, fie de un singur individ selectat pentru a suferi o

modificare (mutație). Noul descendent (noua soluție), fie că a fost obținut din doi părinți cu ajutorul operatorului de recombinare/încrucișare, fie dintr-un singur individ cu ajutorul operatorului de mutație, va „moșteni” unele din caracteristicile înaintașilor, fiind diferit mai mult sau mai puțin de aceștia. Mai departe, vor fi selectați noi părinți, și astfel noi și noi generații sunt produse, astfel încât performanța medie va fi îmbunătățită de la o generație la alta, deoarece prin mecanismul de selecție cei mai „buni” indivizi sunt lăsați să se reproducă. Să amintim faptul că operatorii de variație depind de reprezentarea cromozomială.

D₁. Recombinarea (încrucișarea)

În domeniul algoritmilor genetici, operatorul de recombinare/încrucișare (*recombination/crossover operator*) este un operator binar de variație, care se aplică, ușual, la doi părinți (cromozomi) producând unul sau doi descendenți ce vor moșteni caracteristici combinate de la cei doi părinți, fiind considerat de mulți ca cel mai important mod de a crea diversitate, devansând astfel mutația. Principiul care stă la baza sa, plecând de la metafora biologică evolutivă, stipulează că prin încrucișarea a doi indivizi cu caracteristici bune (pentru scopul propus), descendenții acestora vor avea probabil caracteristici cel puțin la fel de bune, obținute prin combinarea acestora. Operatorul de recombinare este, în general, un operator stochastic, aplicându-se pe baza unei rate de încrucișare (*crossover rate*) p_c , de obicei aleasă în intervalul $[0.5, 1]$ și care determină șansa ca unei perechi alese de părinți să i se aplice acest operator. Mecanismul este simplu: în principiu se aleg doi părinți și un număr aleator din intervalul $[0, 1]$, care se compară cu rata p_c . După aceea, dacă acesta este mai mic decât rata de încrucișare, doi descendenți vor fi creați din cei doi părinți prin încrucișare, în celălalt caz ei vor fi creați prin copierea părinților. Astfel, descendenții fie vor fi indivizi noi, cu caracteristici moștenite prin recombinare de la părinți, fie clone ale părinților. Să remarcăm că, în general, se pot alege p părinți care să dea naștere la q descendenți, totuși schema de încrucișare $p = 2, q = 2$ fiind predominantă.

Vom prezenta mai jos, succint, unele din cele mai cunoscute metode de încrucișare, în funcție de maniera de reprezentare cromozomială.

- Operatorii de recombinare în cazul reprezentării binare sunt, de regulă, de tipul ($p = 2, q = 2$), cu toate că s-au propus și generalizări:
 - *Încrucișarea cu un punct de tăietură (one-point crossover)*, în care se alege un număr aleator k din mulțimea $\{1, 2, \dots, l-1\}$, unde l este lungimea codării (i.e. lungimea cromozomului), după care cei doi părinți se divizează în acest punct (i.e. secvențele cromozomiale corespunzătoare) și se obțin cei doi descendenți prin recombinarea segmentelor astfel obținute.

Vom exemplifica acest procedeu, considerând cromozomii (a, a, a, b, b, a, b) și (b, b, a, b, a, a, b) și $k = 3$. Descendenții astfel obținuți vor fi: (a, a, a, b, a, a, b) și (b, b, a, b, b, a, b) .

- *Încrucișarea cu N-tăieturi (N-point crossover)*, care se obține din cea precedentă, divizarea făcându-se în mai mult de două locații din secvența cromozomială. Practic, se aleg N numere aleatoare din mulțimea $\{1, 2, \dots, l-1\}$, se segmentează părinții în locațiile respective, după care se obțin descendenții prin cuplarea după o anumită regulă a segmentelor obținute prin divizare. De exemplu, plecând de la cei doi cromozomi de mai sus și alegând două tăieturi $k_1 = 3$ și $k_2 = 5$ (două tăieturi), vom putea obține următorii doi descendenți: (a, a, a, b, a, a, b) și (b, b, a, b, b, a, b) .
- *Încrucișarea uniformă (uniform crossover)*, care, spre deosebire de cele anterioare care segmentează părinții, tratează fiecare genă individual, alegând în mod aleator căruia părinte îi aparține. Practic, se generează o secvență de l numere aleatoare (reamintim că l este lungimea codării), uniform repartizate în intervalul $[0, 1]$. După aceea, pentru primul descendent, pentru fiecare poziție în parte a codării cromozomiale, se compară componenta corespunzătoare a secvenței cu un parametru fixat p (de obicei egal cu 0,5) și se decide că gena respectivă va fi moștenită de la primul părinte, dacă valoarea este mai mică decât p , sau de la cel de-al doilea părinte, în celălalt caz. Al doilea descendent este creat, în mod natural, prin aplicația inversă (i.e. se alege valoarea corespunzătoare de la celălalt părinte). De exemplu, pentru cromozomii anteriori, fie secvența aleatoare, uniform repartizată $[0.47, 0.53, 0.64, 0.27, 0.31, 0.78, 0.83]$ și $p = 0.5$. Atunci, primul descendent va avea forma (a, b, a, b, b, a, b) , iar al doilea (b, a, a, b, a, a, b) .
- Operatorii de recombinare în cazul reprezentării întregi sunt, de regulă, aceiași ca și în cazul reprezentării binare.
- Operatorii de recombinare în cazul reprezentării reale pot fi fie cei prezentați anterior în cadrul reprezentării binare sau întregi (cazul recombinării discrete *discrete recombination*), fie cei de tip recombinare aritmetică (*arithmetic recombination*), în care, considerând părinții \mathbf{x} , \mathbf{y} și descendental \mathbf{z} , valoarea genei a i -a este dată de combinația convexă $z_i = \alpha \cdot x_i + (1 - \alpha) \cdot y_i$, $\alpha \in [0, 1]$. De obicei parametrul α are valoarea intrinsecă $\frac{1}{2}$ (i.e. mijlocul segmentului $[\mathbf{x}, \mathbf{y}]$), cu toate că, teoretic, ar trebui ales aleatoriu în intervalul $[0, 1]$. Vom prezenta succint, trei tipuri de astfel de încrucișare.

- *Recombinarea aritmetică simplă*, în care se alege un punct de încrucișare k din mulțimea $\{1, 2, \dots, l-1\}$ și descendenții se construiesc astfel: primul va avea primele k gene ale primului părinte, valorile pentru restul lor fiind calculate după formula: $z_i = \alpha \cdot x_i + (1-\alpha) \cdot y_i$, $i = k+1, k+2, \dots, l$, în timp ce al doilea descendent se obține în același mod, dar alegând părinții invers. Pentru exemplificare, fie părinții $\mathbf{x} = (1.2, 2.5, 3.4, 4.5)$, și $\mathbf{y} = (0.3, 2.1, 7.6, 0.5)$ și $k = 2$, $\alpha = \frac{1}{2}$. Obținem, astfel: $(1.2, 2.5, 5.5, 2.5)$, $(0.3, 2.1, 5.5, 2.5)$.
- *Recombinarea aritmetică unică*, în care se alege un punct de încrucișare k din mulțimea $\{1, 2, \dots, l-1\}$ și descendenții se construiesc astfel: primul va avea primele $(k-1)$ gene ale primului părinte, gena a k -a fiind calculată după formula: $z_k = \alpha \cdot x_k + (1-\alpha) \cdot y_k$, restul fiind, în continuare, moștenite de la primul părinte, în timp ce al doilea descendent se obține în același mod, dar alegând părinții invers. Considerând, din nou, cazul anterior, obținem: $(1.2, 2.3, 3.4, 4.5)$, $(0.3, 2.3, 7.6, 0.5)$.
- *Recombinarea aritmetică totală*, în care înlocuirea genelor se face în totalitate (schimbare completă), după următoarele formule: $z_i = \alpha \cdot x_i + (1-\alpha) \cdot y_i$, $i = 1, 2, \dots, l$ pentru primul descendent, și $z'_i = \alpha \cdot y_i + (1-\alpha) \cdot x_i$, $i = 1, 2, \dots, l$, pentru al doilea descendent. Menționăm că acest mod de recombinare este cel mai des utilizat în practică. Pentru exemplul anterior, obținem: $(0.75, 2.3, 5.5, 2.5)$, $(0.75, 2.3, 5.5, 2.5)$.
- Operatorii de recombinare în cazul reprezentării prin permutări formează un caz mai special, deoarece aici contează, în principiu, o anumită ordine, în consecință nefiind indicat mixajul arbitrar al genelor, ideea de bază fiind prezervarea în descendenți a informației comune a celor doi părinți. Cele mai cunoscute metode utilizate sunt următoarele (pentru detalii, vezi [45]):
 - *Încrucișarea aplicată parțială* (*partially mapped crossover – PMX*), propusă de către Goldberg și Lingle [60], inițial pentru problema comis-voiajorului (TSP), dar extinsă ulterior și pentru alte probleme de proximitate/vecinătate (*adjacency-type problems*). În principiu, PMX transmite informații privind atâtordonarea cât și valorile de la părinți la descendenți, o porțiune din secvența corespunzătoare unui părinte fiind aplicată peste o porțiune din secvența celuilalt părinte (aplicarea parțială -PM), restul informației fiind interschimbătă. De exemplu, să considerăm părinții $\mathbf{x} = (1, 2, 3 | 5, 4, 6, 7 | 8, 9)$ și $\mathbf{y} = (4, 5, 2 | 1, 8, 7, 6 | 9, 3)$, în care cifrele pot indica etichetele

orașelor pentru TSP, pentru care se aleg două puncte de încrucișare în mod aleatoriu, de exemplu 3 și 7 (marcate cu bare verticale). În prima etapă, avem: (*, *, *, 1, 8, 7, 6, *, *) și (*, *, *, 5, 4, 6, 7, *, *), copiindu-se subsecvența primului părinte în descendantul al doilea și subsecvența celui de-al doilea părinte în primul descendant, obținându-se în același timp și aplicația 1~5, 8~4, 7~6 și 6~7. Apoi, în cei doi descendenți se copiază elementele din cei doi părinți în locurile rămase libere, având grijă să nu se repete valorile. În cazul în care o valoare se repetă, aceasta va fi înlocuită cu cea corespunzătoare aplicației de mai sus. În cazul de față, următorul pas duce la (*, 2, 3, 1, 8, 7, 6, *, 9) pentru primul descendant și la (*, *, 2, 5, 4, 6, 7, 9, 3) pentru al doilea, semnul ,*’ indicând faptul că o copiere automată ar fi dus la suprapunere (e.g. 1 în cazul primului descendant, copiat pe primul loc, s-ar fi suprapus cu 1 deja existent). Atunci, procedând după regula de mai sus, deoarece 1~5, pe primul loc vom pune 5, iar în ceea ce privește ,*’ de pe locul 8 în care ar fi trebuit pus 8, vom pune 4, deoarece avem corespondența 8~4. În final, obținem primul descendant (5, 2, 3, 1, 8, 7, 6, 4, 9) și al doilea, procedând analog, (8, 1, 2, 5, 4, 6, 7, 9, 3).

- *Încrucișarea extremităților (edge crossover -EX)*, introdusă de Whitley [174], în care crearea descendenților se bazează pe considerarea extremităților prezente într-unul sau mai mulți părinți, utilizând așa-numitul tabel margine/adiacență (*edge/adjacency table*), ce conține, pentru fiecare element din permutare, lista celorlalte elemente conectate la acesta (i.e. vecine cu el) în cei doi părinți. În general, acest mod de recombinare are ca scop moștenirea a cât mai multor extremități parentale posibil, eficiența sa în probleme combinatorice (e.g. TSP) fiind dovedită cu prisosință. Au apărut mai multe variante îmbunătățite ale acestei proceduri, cea mai des utilizată fiind cea cunoscută cu numele de *edge-3*, pe care o și prezentăm mai jos. Să notăm că un singur descendant este creat cu ajutorul acestui operator. Pe scurt, algoritmul utilizat în acest caz este următorul: (i) se construiește tabelul de adiacență; (ii) se alege aleator o genă ca genă curentă (element curent) și se îndepărtează din tabel toate referințele la aceasta; (iii) se examinează lista adiacențelor corespunzătoare genei curente. Astfel, dacă există o extremitate comună celor doi părinți (semnalizată prin semnul ,+’), se alege aceasta ca fiind următorul element. Altfel, din lista adiacență curentă, se alege acel element care are lista sa adiacență cea

mai scurtă. Legăturile se tăie (sunt rupte) în mod aleatoriu. Odată ce un element a fost considerat, referințele la el sunt îndepărtate de pe listele de adiacență ale celorlalte elemente; (iv) se repetă pasul (iii) până când se termină de completat întreaga secvență, sau un element care nu are intrări în lista sa de adiacență a fost utilizat, caz în care se alege un nou element de start, dacă nu s-au epuizat toate elementele. Pentru a înțelege mai bine algoritmul, vom prezenta un exemplu referitor la cazul TSP. Fie cei doi părinți: $x = (g, d, m, h, b, j, f, i, a, k, e, c)$ și $y = (c, e, k, a, g, b, h, i, j, f, m, d)$, unde literele a, b, ..., m reprezintă cele 12 orașe luate în considerare. În acest caz, tabelul de adiacență este prezentat mai jos (un '+' indică faptul că extremitatea se găsește în ambii părinți):

Oraș	Lista extremităților	Oraș	Lista extremităților
a	+k, g, i	g	a, b, c, d
b	+h, g, j	h	+b, i, m
c	+e, d, g	i	h, j, a, f
d	+m, g, c	j	+f, i, b
e	+k, +c	k	+e, +a
f	+j, m, i	m	+d, f, h

Se alege, aleatoriu, să zicem orașul ,a' ca prim element al secvenței, după care orașul ,k' este ales, deoarece extremitatea (a, k) apare în ambii părinți. Apoi, orașul ,e' este ales din lista extremităților lui ,k', deoarece este singurul rămas în lista lui ,k'. Se repetă procedura, obținându-se astfel secvența parțială (a, k, e, c), de la care nu mai avem alegere deterministă de a continua completarea secvenței. Orașul ,c' are extremități orașele ,d' și ,g', fiecare având două extremități neutilizate. Se va alege, aleatoriu, orașul ,d', de exemplu, pentru continuarea procedurii. Construcția deterministă continuă astfel până la poziția 7, obținând secvența parțială (a, k, e, c, d, m). La această poziție se va alege din nou, aleatoriu, între ,f' și ,h', alegând, de exemplu, orașul ,h', după care, urmând procedura

deterministă, se ajunge la secvența (a, k, e, c, d, m, h, b, g), punct din care nu mai putem continua aşa, deoarece nu mai există extremități rămase în lista lui „g” (situație numită și *eșec* al recombinării de tip *edge-3*). În acest caz, se procedează la utilizarea aşa-numitelor elemente terminale. Un element se numește *terminal* dacă apare la unul dintre capetele unei secvențe parțiale, când toate extremitățile acestuia sunt moștenite de la părinti. Terminalul este *viu* (*live*) dacă acel element încă are intrări în lista sa, altfel fiind considerat *mort* (*dead*). În principiu, când în această procedură apare un *eșec*, există un terminal „viu” în secvență pentru continuarea procesului de obținere a descendantului, altfel trebuie reinitiată procedura de la început. În cazul de față, ne vom reîntoarce la elementul „a” (care a reprezentat și punctul de start), care mai are o intrare neutilizată încă în listă. Procedura în acest caz este de inversare a secvenței parțiale, pentru a putea continua cu „a”. Plecând de la secvența parțială inversată (g, b, h, m, d, c, e, k, a) vom adăuga orașul „i” rămas pe lista lui „a”. Continuând în maniera obișnuită și nemaivând *eșecuri*, vom obține descendantul dorit sub forma (g, b, h, m, d, c, e, k, a, i, f, j).

- *Încruzișarea de ordine (order crossover -OX)* se aplică în cazul problemelor de tip permutări bazate pe ordine, copiind în descendant un segment ales aleator din primul părinte, cu scopul de a transmite, în același timp, și informația privind ordinea relativă din al doilea părinte. Pe scurt, procedura se bazează pe alegerea aleatorie a două puncte de încruzișare atât în primul cât și în al doilea părinte, după care se copiază în primul descendant segmentul din primul părinte cuprins între cele două puncte. Apoi, plecând de la al doilea punct de încruzișare pentru al doilea părinte, se copiază elementele neutilizate deja, în ordinea în care apar în acesta, începând cu partea de final a secvenței descendantului spre partea de început a sa (manieră toroidală). Al doilea descendant se obține în mod similar, inversând părintii. Vom exemplifica acest tip de încruzișare cu un caz TSP, în care $\mathbf{x} = (j, h, d, e, f, i, g, c, b, a)$ și $\mathbf{y} = (h, g, e, b, c, j, i, a, d, f)$, iar cele două puncte de încruzișare sunt 3 și 6, deci secvența din primul părinte, copiată în primul descendant este (e, f, i). Pentru început obținem pentru primul descendant forma: (*, *, *, e, f, **i**, *, *, *, *). Completând apoi, „de la coadă la cap” primul descendant cu elementele neutilizate din al doilea părinte, se obține secvența primului descendant (b, c, j, e, f, **i**, a, d, h, g). Al doilea descendant va fi dat de (e, f, **i**, b, c, j, g, a, h, d).

- *Încrucișarea ciclică (cycle crossover -CX)*, care are ca scop prezervarea a cât mai multă informație despre poziția absolută în care apar elementele. CX nu utilizează, ca alți operatori de recombinare, puncte de încrucișare. Tehnic vorbind, mai întâi se copiază primul element din primul părinte în primul descendant (pe prima poziție), după care se caută primul element din al doilea părinte, se găsește poziția lui în primul părinte și se copiază pe această poziție în primul descendant. Apoi, se caută elementul din aceeași poziție în al doilea părinte și se găsește în primul părinte și.a.m.d. Procesul se continuă atâtă timp cât nu se găsește un element deja copiat, moment în care se termină un ciclu (început cu primul element din primul părinte), restul pozițiilor din primul descendant fiind completate cu elementele de pe aceleași poziții din al doilea părinte. Pentru o mai bună înțelegere a modului de operare a acestei încrucișări, să considerăm $\mathbf{x} = (1, 2, 3, 4, 5, 6, 7, 8, 9)$, $\mathbf{y} = (4, 1, 2, 8, 7, 6, 9, 3, 5)$, care pot fi considerate și ca etichetele orașelor pentru TSP. Vom începe cu primul element din \mathbf{x} , pe care îl copiem în primul descendant pe primul loc $(1, *, *, *, *, *, *, *, *)$ și vom căuta apoi elementul din \mathbf{y} care se găsește pe aceeași poziție (prima), adică 4, care se va copia în descendant în poziția a 4-a, $(1, *, *, 4, *, *, *, *, *)$, deoarece aceasta este poziția sa în \mathbf{x} . Mai departe, în \mathbf{y} , pe aceeași poziție (i.e. a 4-a), avem pe 8, deci avem $(1, *, *, 4, *, *, *, 8, *)$, deoarece 8 se găsește în primul părinte pe poziția a 8-a. Se continuă cu $(1, *, 3, 4, *, *, *, 8, *)$, $(1, 2, 3, 4, *, *, *, 8, *)$. În acest moment, selectarea lui 2 va implica selectarea lui 1, care deja a fost copiat, deci vom încheia ciclul în schema de încrucișare. În consecință, vom folosi al doilea părinte pentru completarea locurilor libere, deci $(1, 2, 3, 4, 7, 6, 9, 8, 5)$. Similar, plecând de la \mathbf{y} , obținem $(4, 1, 2, 8, 5, 6, 7, 3, 9)$.
- Operatorii de recombinare cu părinți mulți (*multiparent recombination*) extind operatorii de variație cu doi părinți –cazul încrucișării– sau cu un părinte –cazul mutației–, considerând scheme cu 3, 4 și mai mulți părinți, scheme ce nu-și mai găsesc echivalentul în biologie, dar sunt fezabile din punct de vedere matematic și, în plus, s-au dovedit eficiente în multe cazuri (vezi [45]).

D₂. Mutăția

Spre deosebire de recombinare (încrucișare), care era un operator *binar* (exceptând cazul multi-parental), operatorul de *mutație* este unul *unar*, inspirat de mutația biologică (i.e. procesul de schimbări ale materialului

genetic, în spăția ADN și ARN, cauzat de erori de copiere ale acestuia în timpul diviziunii celulare), fiind considerat pe locul doi în ierarhia operatorilor de variație din cadrul GA. El se aplică unui singur cromozom, rezultând un mutant ușor modificat al acestuia, numit tot *descendent (offspring)*. Un operator de mutație este întotdeauna stochastic, deoarece mutantul obținut depinde de rezultatele unor alegeri aleatoare. Rolul său în contextul GA (diferit de alte cazuri de EA), este acela de a preveni situația ca populația de cromozomi să devină prea asemănătoare, implicând astfel încetinirea sau chiar stoparea procesului de evoluție, obiectiv îndeplinit prin procurarea de „sânge proaspăt” în procesul de evoluție. Din punct de vedere tehnic, mutația modifică una sau mai multe gene cu o anumită probabilitate –rata de mutație.

La fel ca și în cazul operatorului de recombinare, și în cazul mutației procedura depinde de modul de codare (reprezentare) utilizat. În cele ce urmează vom trece în revistă succint cele mai cunoscute metode.

- Operatorul de mutație în cazul reprezentării binare consideră, în principiu, fiecare genă separat, și permite fiecărui bit să basculeze (*bit-flipping*) între 0 și 1 cu o probabilitate scăzută p_m (*bitwise mutation*). Dacă lungimea codării este l , atunci, deoarece fiecare bit are aceeași probabilitate de mutație p_m , rezultă că, în medie, un număr de $l \cdot p_m$ valori vor fi schimbate. Un exemplu în acest sens este mutația cromozomului (0, 1, 1, 1, 1), reprezentând, de exemplu, codarea binară a numărului 15, în care presupunem că, generând 5 numere aleatorii (pseudo-aleatorii), numerele de pe pozițiile a 3-a și a 5-a sunt mai mici decât rata de mutație p_m . Atunci $(0, 1, 1, 1, 1) \rightarrow (0, 1, 0, 1, 0)$. Problema care apare în cazul acestui tip de mutație este modul în care se alege rata de mutație p_m .
- Cei mai utilizați operatori de mutație în cazul reprezentării întregi sunt următorii doi:
 - *Resetarea aleatoare (random resetting/random choice)*, extinde mutația anterioară de tip *bit-flipping*, în sensul că, cu o probabilitate p_m , o valoare nouă este aleasă în mod aleatoriu din domeniul valorilor pentru fiecare poziție; se aplică mai ales în cazul în care genele codează atributele categoriale (e.g. nord, sud, est, vest; verde, albastru, galben, roșu etc.);
 - *Mutația deformare (creep mutation)* se aplică mai ales în cazul atributelor ordinale, constând în adunarea sau scăderea la fiecare genă a unei valori (pozitive) mici, cu o anumită probabilitate p , nedepășind, evident, domeniul corespunzător al valorilor. În acest mod, cu o probabilitate semnificativă, valoarea unei gene va fi schimbată cu o valoare similară (obținută printr-o „deformare” –*creep*). În cazul acestei mutații este nevoie de o anumită repartiție care să controleze modul de

alegere a valorilor pentru fiecare poziție, repartiție ce este determinată de anumiți parametri. În consecință, trebuie rezolvată problema alegerii acestor parametri, pentru eficientizarea procesului. Să menționăm că, adesea, se utilizează două mutații în tandem.

- Operatorii de mutație utilizați în cazul reprezentării reale schimbă, aleatoriu, valorile fiecărei gene în cadrul domeniului acestora de valori după schema $(x_1, x_2, \dots, x_l) \rightarrow (x'_1, x'_2, \dots, x'_l)$, $x_i, x'_i \in D_i$, unde D_i este domeniul valorilor pentru gena a i -a. Prezentăm, în continuare, două tipuri uzuale de astfel de operatori.
 - *Mutația uniformă (uniform mutation)*, se bazează pe alegerea uniformă (repartiția uniformă) a valorilor genelor x_i, x'_i în domeniul D_i , fiind, practic, analogă mutațiilor *bit-flipping* (cazul binar) sau *random resetting* (cazul întregilor).
 - *Mutația neuniformă (nonuniform mutation)*, se bazează, la fel ca și mutația *creep* din cazul întregilor, pe „deformarea” valorii fiecărei gene prin adăugarea unei mici cantități, obținută dintr-o repartiție normală (gaussiană), de medie zero și deviație standard (SD) arbitrară, aleasă de utilizator, având grijă ca noua valoare a genei să aparțină domeniului valorilor sale. Tinând cont de faptul că în cazul acestei repartiții, de exemplu 95% din valori se găsesc în intervalul de încredere $I_{95\%}$, definit de $(-1.96 \times SD, 1.96 \times SD)$, rezultă că majoritatea modificărilor vor fi de mică amploare, pentru o valoare convenabilă a dispersiei. Deoarece repartiția care generează mutația este determinată de la început, acest tip de mutație mai este cunoscut și ca mutație neuniformă cu repartiție fixă.
- Să remarcăm că se poate utiliza și mutația de tip *bit-flipping* (cazul codării binare) pentru reprezentări reale. De exemplu [120], în cazul problemei maximizării funcției obiectiv $f : [-1, 2] \rightarrow R$, dată de $f(x) = x \cdot \sin(10 \cdot \pi \cdot x) + 1$, considerând pentru operația de mutație cromozomul $x_3 = 1,627888 \sim (1110000000|11111000101)$ în care înlocuim, de exemplu, gena a 10-a, care are valoarea 0 cu 1, obținem mutația $x'_3 = 1,630818 \sim (1110000001|11111000101)$, care produce o valoare mai mare a funcției de evaluare, deci mutantul este mai bun decât originalul.
- Operatorii de mutație utilizați în cazul reprezentării prin permutări nu mai pot acționa, ținând cont de contextul reprezentării,

independent asupra fiecărei gene. Vom menționa, pe scurt, patru dintre cei mai comuni astfel de operatori.

- *Mutația schimb (swap mutation)*, în care se aleg, aleatoriu, două poziții (gene) în cromozom, și se schimbă între ele valorile acestora. De exemplu, plecând de la cromozomul (1, 2, 3, 4, 5, 6, 7, 8, 9) și alegând pozițiile (genele) a 4-a și a 7-a, se obține mutantul (1, 2, 3, 7, 5, 6, 4, 8, 9).
- *Mutația inserare (insert mutation)* operează în felul următor: se aleg două gene, aleatoriu, după care una din ele se inserează lângă cealaltă. De exemplu, plecând de la cazul anterior, se obține (1, 2, 3, 4, 5, 6, 7, 8, 9) \rightarrow (1, 2, 3, 4, 7, 5, 6, 8, 9).
- *Mutația amestec (scramble mutation)* operează astfel: se consideră fie întreaga secvență a cromozomului, fie o submulțime a sa, aleasă aleatoriu, după care pozițiile genelor corespunzătoare acesteia se amestecă (*scrambled'egg -omletă*). De exemplu, plecând de la cromozomul (1, 2, 3, 4, 5, 6, 7, 8, 9) și alegând, de exemplu, subsecvența (4, 5, 6, 7), obținem mutantul (1, 2, 3, 6, 7, 4, 5, 8, 9).
- *Mutația inversiune (inversion mutation)* operează astfel: se aleg, în mod aleatoriu, două poziții în cromozom, după care subsecvența astfel obținută (definită de cele două poziții) se copiază în ordine inversă. De exemplu, considerând în cromozomul (1, 2, 3, 4, 5, 6, 7, 8, 9) pozițiile 4 și 7, și aplicând inversiunea, se obține mutantul (1, 2, 3, 7, 6, 5, 4, 8, 9).

E. Parametrii algoritmului

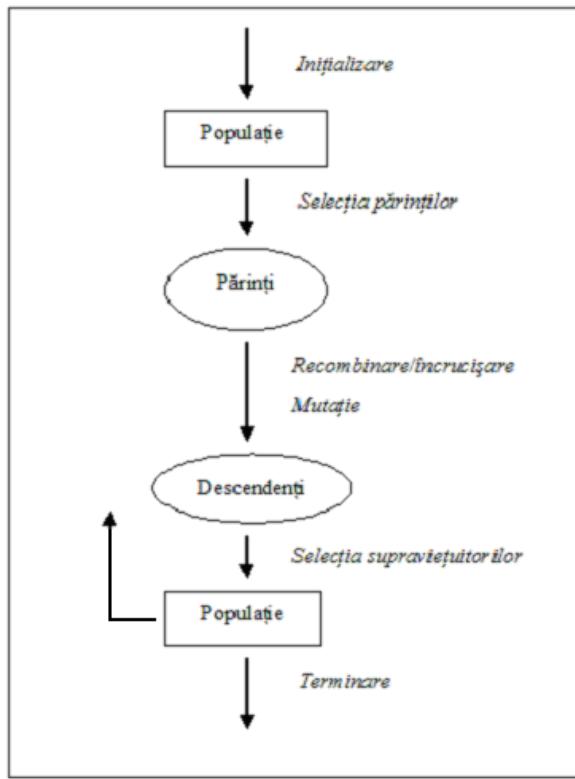
Orice algoritm genetic implică, aşa cum bine s-a observat mai sus, un set de parametri care apar în construcția sa: dimensiunea/topologia populației, probabilitățile de aplicare a operatorilor de variație (probabilitatea de încrucișare, probabilitatea de mutație), numărul total de generații etc. Este un fapt notoriu că problema setării (alegerii) acestor parametri este crucială pentru o performanță bună a algoritmului. De exemplu, atunci când se construiește un algoritm genetic, se specifică faptul că s-a ales, de pildă, reprezentarea binară, încrucișarea uniformă, mutația de tip *bit-flipping*, selecția turnir și modelul generational pentru populație. Ca să se poată trece la construirea efectivă a sa, este nevoie să se specifică valorile parametrilor corespunzători (e.g. dimensiunea populației, probabilitățile de încrucișare p_c și de mutație p_m , mărimea turnirului k). Se cunosc, în principiu, două abordări privind alegerea celor mai bune valori ale parametrilor pentru ca algoritmul să lucreze eficient:

- ◆ Ajustarea parametrilor (*parameter tuning*), adică găsirea unor valori bune pentru parametri (pe baza unor experimente cu diferite valori), înainte de rularea algoritmului și nemodificarea lor pe parcursul rulării algoritmului –parametri statici.
- ◆ Controlul (adaptarea) parametrilor (*parameter control*), reprezentă alternativa la metoda anterioară, presupunând începerea rulării algoritmului cu o anumită setare a parametrilor, care va fi schimbăță pe parcursul rulării acestuia –parametri dinamici.

Se observă că este mai bine, în ciuda dificultăților inerente, dacă se utilizează metoda adaptării parametrilor în locul fixării lor încă de la început. În afară de un consum considerabil de timp și resurse computaționale pentru găsirea unor valori convenabile ale parametrilor înainte de rularea algoritmului, s-a observat că, în principiu, fiecare problemă necesită o anumită setare a lor, chiar dacă diferite tipuri de probleme cer setări specifice, și că fixarea valorii parametrilor, odată pentru totdeauna, intră în contradicție cu dinamica inerentă oricărui GA. Există multe căi de a rezolva aceasta problemă (e.g. extinderea de la o constantă p la o funcție $p(t)$, utilizarea chiar a GA pentru optimizarea setării parametrilor, utilizarea factorului uman în modificarea parametrilor, utilizând informația curentă etc.), cheia succesului constând în alegerea strategiei corespunzătoare problemei ce trebuie rezolvată, alegere bazată atât pe cunoașterea profundă a problemei cât și a unor metode potrivite de adaptare a parametrilor, prezente în literatura de specialitate (vezi, de exemplu, [8], [43], [142]).

5.8.2 Arhitectura unui algoritm genetic

În paragraful precedent am făcut o scurtă trecere în revistă a componentelor unui algoritm genetic, insistând acolo unde am crezut de cuvînță asupra unor detalii tehnice. A sosit acum momentul să „asamblăm” elementele constitutive ale unui asemenea algoritm într-o structură coerentă, care să ilustreze modul lor specific de operare. Schematic, arhitectura oricărui algoritm genetic (algoritm evolutiv, în general) poate fi reprezentată ca mai jos.



Această arhitectură simplă se poate implementa sub forma următoarei structuri [119], [120].

```

BEGIN
   $t \leftarrow 0$ 
  initialize  $P(t)$ 
  evaluate  $P(t)$ 
  WHILE (NOT termination condition) DO
    BEGIN
       $t \leftarrow t + 1$ 
      select  $P(t)$  from  $P(t - 1)$ 
      alter  $P(t)$ 
      evaluate  $P(t)$ 
    END
  END

```

Plecând de la această structură, prezentăm mai jos schema clasică a unui GA [40].

1. Se alege un mecanism de selecție și se inițializează populația $P(0)$.
2. Se pune $t \leftarrow 0$.
3. Se evaluatează cromozomii populației $P(t)$ și se reține cel mai performant dintre ei.
4. Se aplică operatorul de selecție de n ori (n reprezintă dimensiunea populației inițiale), cromozomii selectați formând o populație intermediară P^1 de aceeași dimensiune.
5. Se aplică operatorul de recombinare/încrucișare asupra populației P^1 , descendenții obținuți formând populația P^2 împreună cu cromozomii din P^1 , rămași după îndepărțarea părinților aleși pentru încrucișare.
6. Se aplică operatorul de mutație asupra populației P^2 , rezultând o nouă generație de cromozomi $P(t + 1)$.
7. Se pune $t \leftarrow t + 1$.

Dacă $t \leq N$, unde N reprezintă numărul maxim de generații, atunci se revine la pasul 3, altfel cel mai performant cromozom este soluția problemei și STOP.

Remarcă: 1. Primul pas în implementarea oricărui algoritm genetic este reprezentat de generarea unei populații inițiale. În cazul clasic, fiecare membru al acestei populații este o secvență binară, numită fie „genotip” [99], fie „cromozom” [139].
2. Execuția unui algoritm genetic implică două stadii. În primul rând, plecând de la populația curentă, se aplică acesteia selecția, în vederea creării populației intermediere. În al doilea rând, se aplică operatorii de variație (recombinare/încrucișare și mutație) populației intermediere în vederea creării populației următoare.
3. După aplicarea selecției, recombinării/încrucișării și mutației, populația următoare este supusă evaluării. Astfel, procesul de trecere de la populația curentă la populația următoare, pe baza evaluării, selecției, recombinării/încrucișării și mutației, reprezintă o generație în execuția algoritmului.
4. Atât între pașii de inițializare și evaluare (prima fază), cât și între pașii de modificare și evaluare (faza a doua), se poate introduce o procedură de optimizare locală a populației $P(t)$, obținând o schemă GA modificată.

Notă: Algoritmii evolutivi, în general, și cei genetici, în particular, posedă o caracteristică generală privind modul în care operează. Astfel, la început, după inițializare, indivizii sunt „împrăștiați” în mod aleatoriu în întregul spațiu de căutare a soluției optime. După câteva generații, prin aplicarea operatorilor de selecție și variație, situația se schimbă și indivizii încep să „urce dealul” (în cazul unei probleme de maximizare), abandonând „văile”. În sfârșit, spre finalul căutării, toți indivizii se „îngrămadesc” pe câteva „vârfuri de deal”, iar dacă aceste vârfuri reprezintă maxime locale și nu maximul global, este posibil ca populația să „urce pe un deal greșit”. Plecând de la această comparație, ajungem la definirea unei probleme majore a căutării: „echilibrul între exploatarea celor mai buni indivizi disponibili la un moment dat și explorarea robustă a spațiului de căutare”. Cu toate că nu există definiții universal acceptate, putem spune că:

- *Explorarea (exploration)* reprezintă generarea de indivizi noi în regiuni necunoscute ale spațiului de căutare, descoperind astfel arii promițătoare de cercetare și câștigând, în consecință, informații despre problema de rezolvat;
- *Exploatarea (exploitation)* reprezintă concentrarea căutării în vecinătatea soluțiilor deja performante, optimizând o arie cunoscută ca promițătoare, prin utilizarea de informație disponibilă.

În acest context, în ceea ce privește operatorii de variație, putem considera recombinarea ca explorativă, în timp ce mutația poate fi considerată exploataativă, existând o colaborare și, în același timp, „competiție” între ei pentru o căutare optimă.

5.8.3. Aplicații

Este dificil să prezentăm o listă cât de cât completă a aplicațiilor algoritmilor genetici, în particular, și a celor evolutivi, în general. O scurtă navigare pe Internet va furniza o multitudine de aplicații în cele mai variate domenii ca, de exemplu: ingererie, artă, economie, biologie, genetică, cercetări operaționale, robotică, științe sociale, fizică, chimie, informatică etc.

O listă cu unele din cele mai cunoscute aplicații este prezentată mai jos, cu toate că este departe de a crea o imagine globală asupra câmpului de aplicații posibile:

- ◆ Proiectarea automată în diferite domenii: mecatronică, materiale compozite, echipamente industriale, rețele de calculatoare, aeronautică etc.;
- ◆ Cinetica chimică, optimizarea structurii moleculare;
- ◆ Criptografie;
- ◆ Robotică;

- ◆ Analiza lingvistică;
- ◆ Teoria jocurilor;
- ◆ Optimizarea în rețelele comunicații (e.g. comunicațiile GSM);
- ◆ Ingineria software;
- ◆ TSP (travelling salesman problem);
- ◆ Probleme de programare a unor activități (căi ferate, funcționarea unor mașini-unelte, roboți în rețea, instituții etc.);
- ◆ Afaceri și domenii conexe (finanțe, marketing, planificarea activității, decizii etc.);
- ◆ Sociologie (procesarea imaginilor și recunoașterea formelor, simularea evoluției normelor de comportament etc.);
- ◆ Modelarea calibrării migrației populaționale;
- ◆ Data Mining (procesul de clustering, selectarea atributelor, clasificare, predicție, descoperirea de reguli etc.);
- ◆ Inteligență Artificială (e.g. proiectare pentru rețelele neuronale artificiale, optimizări în cadrul GA etc.).

Prezentarea GA a fost datorată, în primul rând, utilității lor în cadrul procesului de Data Mining, ei reprezentând un instrument important în acest domeniu. În afară de aceasta, GA reprezintă un domeniu fascinant de cercetare, dinamica sa imitând procesele evolutive naturale, și de aceea încercarea de a familiariza cititorul cu elementele de bază ale GA este utilă, alături de prezentarea în cadrul acestei cărți a celuilalt domeniu de sorginte naturală, NN.

În continuare, vom prezenta, pentru exemplificare, trei moduri în care pot fi aplicați GA, în particular, și EA, în general, în probleme de clustering, clasificarea datelor și optimizări NN. Prezentăm mai jos, sintetic, etapele unui asemenea proces în cazul celor trei aplicații menționate mai sus.

- ◆ *Aplicație în probleme de clustering* [88]. Presupunem că trebuie să clusterizăm datele disponibile în trei clustere distincte, utilizând mecanismul GA. O abordare posibilă poate fi structurată în următorii doi pași.
 - *Pasul 1:* Se începe cu gruparea aleatorie a datelor. Funcția de evaluare din cadrul algoritmului va determina dacă un set de date corespunde unuia sau altuia dintre clustere, evaluând cât de bine se potrivește acesta celorlalte elemente din clusterul respectiv. Un exemplu de astfel de funcție poate fi orice funcție ce stabilește nivelul de similaritate între elementele dintr-un grup.
 - *Pasul 2:* Se utilizează acum operatorii de variație din cadrul GA. Astfel, dacă o secvență de date, evaluată cu ajutorul funcției de adecvare, este găsită că se potrivește suficient de bine într-un anumit cluster, ea va supraviețui și va fi copiată în acesta. În caz

contrar, acea secvență va fi încrucișată cu altă secvență pentru crearea unui descendent mai bun și.a.m.d.

- ◆ *Aplicații în probleme de clasificare.* Clasificatorii evolutivi disponibili momentan sunt foarte complicați și dificil de aplicat, deoarece utilizează sisteme complicate de asignare de credite care penalizează sau recompensează regulile bune, precum și scheme foarte complicate ale întregului sistem. O sarcină importantă ar fi, în acest context, aceea de a dezvolta motoare evolutive de găsire a regulilor, mult mai puțin complexe și, în același timp, oferind o acuratețe competitivă. Una dintre abordări preia punctul de vedere al *școlii Michigan* privind reprezentarea – fiecare cromozom codifică o singură regulă în formă conjunctivă, de tipul *if... then*, iar întreaga populație reprezintă mulțimea completă de reguli. Se înlocuiește însă sistemul de acordare de credite printr-un mecanism intenționat pentru favorizarea apariției și menținerii de subpopulații de dimensiune variabilă, care sunt legate de diferite puncte de optim, numit *cromodinamica genetică* [41]. Clasificatorul astfel construit a fost aplicat pentru detecția spam-ului [144]. Cercetarea a implicat dezvoltarea mai departe a unor noi algoritmi în cadrul cromodinamicii genetice, algoritmi validați mai întâi în optimizarea mai multor funcții [148], cel mai performant fiind utilizat pentru clasificare, problema reală considerată pentru testare fiind reprezentată de diagnoza diabetului indienilor Pima [146], [149]. Cea mai recentă abordare asupra clasificatorilor evolutivi este aceea a construirii unei tehnici bazate pe co-evoluția cooperativă. Această abordare a fost validată pe trei probleme reale: diagnoza diabetului indienilor Pima [150], plantele iris [145], [147] și detectarea spam-ului [151].
- ◆ *Aplicație în proiectarea PNN.* Așa cum am arătat și mai înainte, GA se pot utiliza și în probleme de proiectare a instrumentelor DM (e.g. rețele neuronale, GA), în scopul optimizării modului în care acestea rezolvă problemele abordate. Prezentăm, pe scurt, o aplicație GA în problema căutării parametrului σ din cadrul unui PNN [68] (vezi și subparagraful 5.3.3.). În principiu, se consideră că parametrul de ajustare σ reprezintă un cromozom cu o singură genă, aparținând domeniului valorilor, definit pe baza intervalului de încredere 99,7% al mediilor valorilor distanțelor dintre elementele unei clase. Utilizându-se diferite tipuri de operatori de variație, se caută acel cromozom care maximizează funcția obiectiv, definită de numărul cazurilor corect clasificate.

Prezența algoritmilor genetici (evolutivi) pe Internet este astăzi de necontestat. Plecând de la website-uri dedicate prezentării și implementării lor în diferite situații, ca, de exemplu:

- Illinois Genetic Algorithms Laboratory –ILLiGAL (<http://www.illigal.ge.uiuc.edu/about.html>),

- Kanpur Genetic Algorithms Laboratory –KanGAL (<http://www.iitk.ac.in/kangal/>),
- Evolutionary Computation Laboratory (George Mason University) –EClab (<http://cs.gmu.edu/~eclab/>),
- The MathWorks-Genetic Algorithm and Direct Search Toolbox (<http://www.mathworks.com/products/gads/>),

există website-uri dedicate universului calcului evolutiv, ca, de exemplu:

- International Society for Genetic and Evolutionary Computation - ISGEC (<http://www.isgec.org/>);
- Special Interest Group for Genetic and Evolutionary Computation - SIGEVO (<http://www.sigego.org/>);
- Colorado State Genetic Algorithms Group (<http://www.cs.colostate.edu/~genitor/>)
- Machine Learning and Genetic Algorithms group –M&EC University of Torino (<http://www.di.unito.it/~mluser/>).

De asemenea, în afara referințelor bibliografice prezente în această carte sau în literatura de specialitate, se pot găsi pe Internet o mare varietate de website-uri prezentând tutoriale, note de curs, articole științifice și, în general, o sumedenie de contribuții în cele mai diverse arii de interes, toate dovedind puternica vitalitate a acestui domeniu.

Vom încheia această succintă trecere în revistă a algoritmilor genetici (evolutivi, în general), amintind o remarcă interesantă privind utilitatea EA în aplicațiile din viața reală [47] (vezi și [119]: „*Algoritmii evolutivi se aseamănă mult cu un briceag (swiss army knife): un dispozitiv care poate fi aplicat la rezolvarea unor probleme variate din viață, dar cu un anumit cost. Astfel, pentru fiecare caz în care poate fi folosit, există de obicei câte un dispozitiv special proiectat pentru acea problemă și care, evident, este mai bun decât briceagul... Totuși, dacă nu se știe exact ce sarcină trebuie înălțată, natura flexibilă a briceagului îl face să fie ales el în loc de alt instrument... Posedarea unui briceag permite rezolvarea unei mari varietăți de probleme, rapid și eficient, chiar dacă ar exista un mijloc mai bun...*”

6. PERFORMANȚA CLASIFICĂRII

O bună parte a acestei cărți a prezentat fundamentele procesului de clasificare, proces deosebit de important în domeniul Data Mining. Este timpul acum să ne ocupăm și de anumite aspecte privind modul de a evalua performanța diferitelor modele de clasificare (și decizie).

6.1. Costul și acuratețea clasificării

Una din problemele importante în procesul de clasificare este calcularea *costului* clasificării și estimarea *acurateții* ei. Pentru a fi cât mai sugestivi, vom prezenta sintetic aceste aspecte, utilizând cele două tabele de mai jos, ce rezumă rezultatele unui proces de clasificare. Să reamintim că matricea costurilor este aleasă de utilizator, indicând ponderile acordate clasificării corecte/eroneate a cazurilor și depinde de contextul problemei.

MATRICE COST		CLASE ANTICIPATE	
CLASE EFECTIVE		Clasa = Da	Clasa = Nu
	Clasa = Da	p	q
	Clasa = Nu	r	s

CLASIFICARE		CLASE ANTICIPATE	
CLASE EFECTIVE		Clasa = Da	Clasa = Nu
	Clasa = Da	a	b
	Clasa = Nu	c	d

Formulele aferente calculării costului și acurateții clasificării sunt următoarele:

$$Cost = p \times a + q \times b + r \times c + s \times d,$$

$$Acuratete = \frac{a + d}{a + b + c + d}.$$

Exemplu:

Să considerăm un proces de clasificare ale cărui performanțe sunt sintetizate în următoarele trei tabele. Astfel, primul tabel se referă la costurile atribuite clasificării corecte/eroneate a cazurilor, costuri stabilite prealabil procesului de clasificare. Tabelele următoare sintetizează clasificările obținute prin aplicarea a doi clasificatori (modelul M_1 și modelul M_2).

MATRICE COST		CLASE ANTICIPATE	
CLASE EFECTIVE		Clasa = Da	Clasa = Nu
	Clasa = Da	-1	100
	Clasa = Nu	1	0

CLASIFICARE (model M_1)		CLASE ANTICIPATE	
CLASE EFECTIVE		Clasa = Da	Clasa = Nu
	Clasa = Da	150	40
	Clasa = Nu	60	250

CLASIFICARE (model M_2)		CLASE ANTICIPATE	
CLASE EFECTIVE		Clasa = Da	Clasa = Nu
	Clasa = Da	250	45
	Clasa = Nu	5	200

După efectuarea calculelor, obținem următorii parametri măsurând performanțele celor doi clasificatori:

$$Cost_{M_1} = -1 \times 150 + 100 \times 40 + 1 \times 60 + 0 = 3910,$$

$$Cost_{M_2} = -1 \times 250 + 100 \times 45 + 1 \times 5 + 0 = 4255,$$

$$Acuratete_{M_1} = \frac{400}{500} = 80\%,$$

$$Acuratete_{M_2} = \frac{450}{500} = 90\%.$$

Problema care se pune după calcularea celor doi parametri, măsurând performanțele a două sau mai multe modele, este modul cum pot fi utilizati în alegerea celui mai performant model (vezi și paragraful 6.4.). Aceasta rămâne o problemă deschisă, depinzând de condițiile concrete în care se utilizează clasificarea. În principiu însă, trebuie stabilit un echilibru între costuri și acuratețe, adică un compromis acceptabil pentru alegerea clasificatorului optim pentru problema dată.

În același context al investigării performanțelor unui anumit model de clasificare, cât și în ceea ce privește compararea mai multor clasificatori în vederea alegerii celui mai performant, vom introduce încă două noțiuni importante. Senzitivitatea unui model de clasificare (binară: +/- sau echivalent DA/NU) este un parametru care exprimă un anumit aspect privind performanța clasificatorului. Astfel, *senzitivitatea* unui clasificator este proporția cazurilor care au un rezultat pozitiv al clasificării în raport cu toate cazurile pozitive testate, adică:

$$\text{Senzitivitatea} = \frac{\text{numarul cazurilor adevarat pozitive}}{\text{numarul cazurilor adevarat pozitive} + \text{numarul cazurilor fals negative}}$$

sau, formal:

$$\text{Senzitivitatea} = \frac{a}{a+b}.$$

Notă: O senzitivitate de 100% înseamnă că toate cazurile negative sunt recunoscute ca atare ($b = 0$, deci nu există cazuri fals negative).

În teoria recuperării informației (*information retrieval*) senzitivitatea mai este numită și *rapel (recall)*.

Senzitivitatea de una singură nu este suficientă, pentru că 100% senzitivitate este un rezultat fără sens atât din punct de vedere statistic cât și al cercetării. Vom folosi de aceea încă un indicator, și anume specificitatea. În teste binare, *specificitatea* reprezintă proporția cazurilor adevărat negative în raport cu toate cazurile negative testate, adică:

$$\text{Specificitatea} = \frac{\text{numarul cazurilor adevarat negative}}{\text{numarul cazurilor adevarat negative} + \text{numarul cazurilor fals pozitive}}$$

sau, formal:

$$\text{Specificitatea} = \frac{d}{c + d}.$$

Notă: Se observă că o specificitate de 100% înseamnă că toate cazurile pozitive sunt recunoscute ca atare ($c = 0$, deci nu există cazuri fals pozitive).

Remarcăm, ca și mai sus, că este nevoie și de studiul senzitivității pentru a evita analizele lipsite de conținut.

Legătura dintre acești parametri de performanță, cunoșcuți și ca *metriki* de evaluare a performanței unor modele de clasificare (decizie), poate fi realizată luând în considerare așa-numitul *tabel al confuziei* (*table of confusion*), prezentat mai jos.

		CLASE ANTICIPATE	
CLASE EFECTIVE		Clasa = Da	Clasa = Nu
CLASE EFECTIVE	Clasa = Da	a <i>Adevărat pozitiv (AP)</i>	b <i>Fals negativ (FN)</i>
	Clasa = Nu	c <i>Fals pozitiv (FP)</i>	d <i>Adevărat negativ (AN)</i>

Atunci avem:

$$\text{Acuratete} = \frac{a + d}{a + b + c + d} = \frac{AP + AN}{AP + AN + FP + FN}.$$

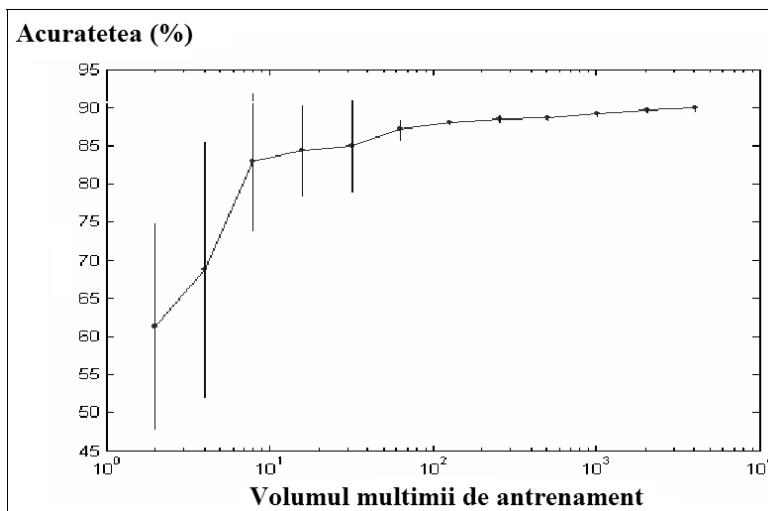
6.2. Curbe de învățare

Metoda *curbelor de învățare* (*learning-curve*) este o altă abordare ce ține de domeniul învățării automate (în particular, clasificarea) și care constă în măsurarea performanței unor algoritmi în cazul unor seturi mari de date. Această metodă se bazează pe următoarea observație simplă: „costul computațional al învățării (antrenării) unui model crește semnificativ odată cu creșterea volumului mulțimii de antrenament, în timp ce acuratețea modelului

are o creștere încetinită pe măsura creșterii volumului mulțimii de antrenament". Astfel, această metodă încearcă să „monitorizeze" atât creșterea costului cât și a performanței în funcție de utilizarea unor mulțimi de antrenament din ce în ce mai mari, astfel încât să se poată stopa procesul de antrenament atunci când costul depășește avantajele (acurateții), identificând deci volumul optim al mulțimii de antrenament.

Exemplu:

Figura de mai jos ilustrează modul de „lucru" al curbelor de învățare (măsurarea accurateții ca funcție de volumul mulțimii de antrenament).



Prezentăm mai jos câteva metode mai cunoscute de măsurare a accurateții, utilizând setul disponibil de date:

- *Holdout* Se rezervă 2/3 din obiecte pentru antrenament și 1/3 pentru testare);
- Sub-eșantionarea la întâmplare (*random subsampling*), echivalentă cu holdout-uri repetate;
- Validarea încrucișată (*cross validation*), care se referă la următorii pași:
 - Partiționarea în k submulțimi disjuncte;
 - Aplicarea tehnicii k -fold: antrenarea pe $(k-1)$ partiții, pentru test rămânând doar una;

- *Leave-one-out*, atunci când $k = n$.
- Eșantionare stratificată (*stratified sampling*), adică eșantionarea fiecărei submulțimi (strat) în mod independent.
- *Bootstrap* (nume ce vine de la cuvântul *bootstrap* = „urechea” din spatele ghetelor, folosită pentru a le scoate din picioare) -aluzie la *Baronul Münchhausen* care a reușit să iasă dintr-o mlaștină trăgându-se de propriul păr (!)- se referă la eșantionarea cu înlocuirea din eșantionul original (*bootstrap resampling*).

6.3. Curvele ROC

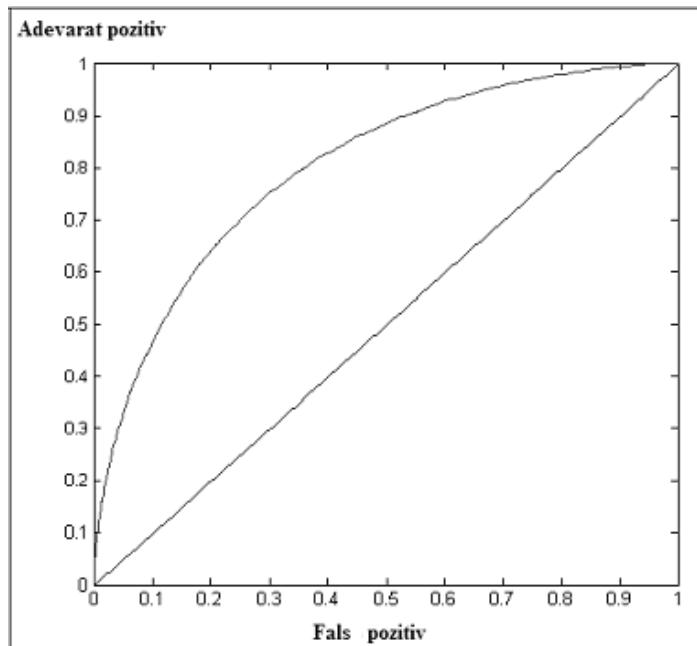
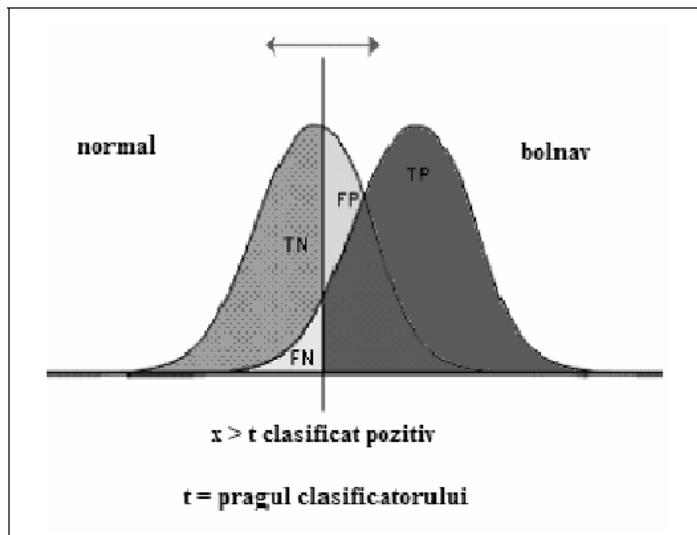
În continuare, vom prezenta câteva noțiuni de bază privind curvele ROC (*Receiver Operating Characteristic*), larg utilizate în evaluare rezultatelor unei predicții (prognoze). Vom prezenta schematic cele mai importante aspecte privind curvele ROC.

- ◆ Dezvoltate puternic în anii 1950 în domeniul detecției semnalelor, curvele ROC au fost utilizate mai întâi în studiul sistemelor discriminator (*discriminator systems*) pentru detecția semnalelor radio în prezența zgomotului. Acest lucru se petreceea în anii celui de-al Doilea Război Mondial, după atacul de la Pearl Harbor (7 Decembrie 1941).
- ◆ Cercetarea inițială a fost motivată de dorința de a determina cum au putut operatorii de la US RADAR să nu detecteze avioanele japoneze ce atacau flota din Pacific.
- ◆ După această primă fază, cercetările au continuat în anii 1950 în direcția analizei zgomotului prezent în semnale.
- ◆ Curvele ROC s-au dovedit de utilă în evaluarea rezultatelor din domeniul învățării automate, la fel ca și în evaluarea motoarelor de căutare de pe Internet.
- ◆ Sunt folosite, de asemenea, intens, în cercetările medicale (e.g. epidemiologie).
- ◆ Curvele ROC reflectă compromisul între alarmele reale și cele false.

Concret, graficul unei curbe ROC reprezintă pe axa ordonatelor (y) rata cazurilor *Adevărat pozitive* (AP), în timp ce pe axa absciselor (x) este reprezentată rata cazurilor *Fals pozitive* (FP), pe măsură ce „pragul” de discriminare este modificat continuu (la fel ca și repartiția eșantionului sau matricea cost). Astfel, se poate alege pragul de discriminare optim, fiecare

punct de pe curba ROC corespunzând performanței modelului pentru o astfel de alegere. Rezultă de aici că metoda curbelor ROC este una dinamică.

Figurile de mai jos ilustrează sintetic modul de lucru al curbelor ROC.



Observăm, din figura de mai sus, că avem următoarele trei situații extreme:

- $(AP, FP) = (0, 0)$: declară totul ca aparținând clasei cazurilor „negative”;
- $(AP, FP) = (1, 1)$: declară totul ca aparținând clasei cazurilor „pozitive”;
- $(AP, FP) = (1, 0)$: predicția ideală.

Astfel, putem trage următoarele concluzii:

- Metoda de predicție cea mai performantă posibilă (i.e. discriminarea optimă) implică un grafic constituit din punctul extrem – stânga sus – în spațiul ROC, adică toate cazurile adevărat pozitive identificate și niciun caz fals pozitiv identificat;
- Un predictor (clasificator) ce operează absolut la întâmplare va produce o linie dreaptă la 45 de grade (prima bisectoare), plecând din punctul – stânga jos – și ajungând în punctul – dreapta sus – astfel încât, pe măsură ce „pragul” de discriminare este deplasat, vom avea un număr egal de cazuri adevărat și fals pozitive;
- Rezultatele care corespund regiunii de sub această linie de „nondiscriminare” sugerează un predictor ce produce în mod semnificativ rezultate greșite.

Exemplu:

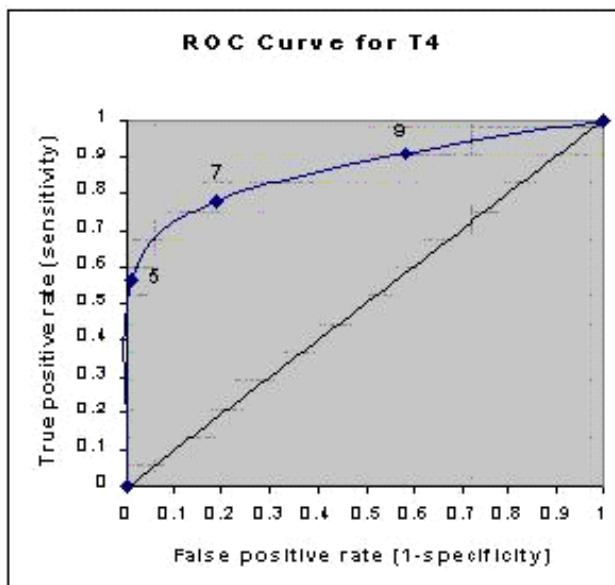
Să considerăm datele tabelate mai jos [62] privind pacienți suspectați de hipotiroidism.

T4 value	Hypothyroid	Euthyroid
≤ 5	18	1
5.1 - 7	7	17
7.1 - 9	4	36
≥ 9	3	39
Total:	32	93

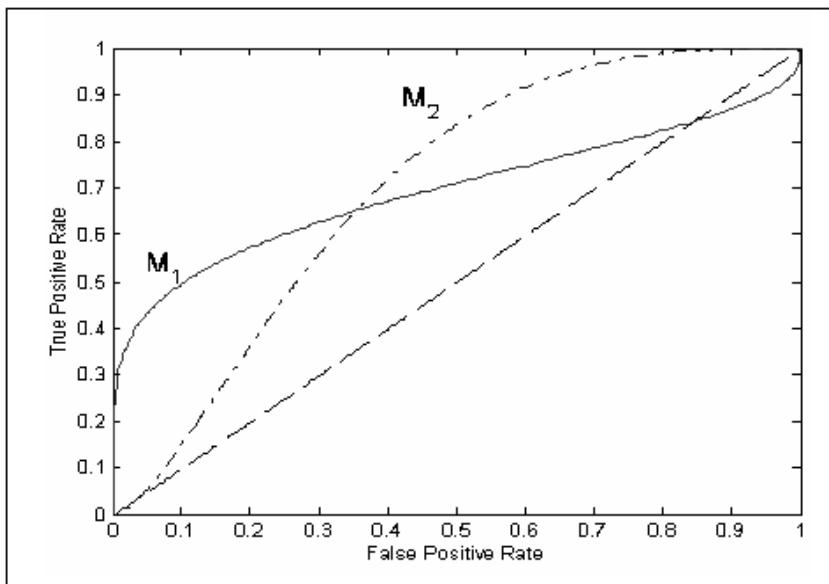
Aplicând metoda curbelor ROC în acest caz, obținem următoarele rezultate, tabelate mai jos.

Prag discriminare T4	Adevărat pozitiv	Fals pozitiv
5	0.56	0.01
7	0.78	0.19
9	0.91	0.58

Curba ROC pentru acest caz este ilustrată în figura de mai jos.



Așa cum am spus mai sus, curbele ROC se utilizează în compararea performanțelor a două sau mai multe modele. Figura de mai jos ilustrează sugestiv această funcție a curbelor ROC.

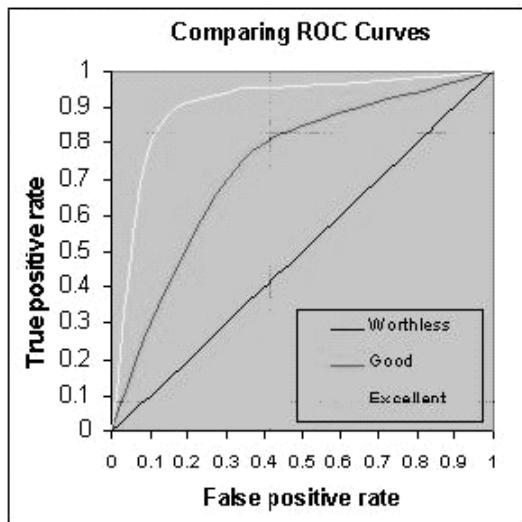


Din figura de mai sus se observă că niciun model nu este în mod semnificativ mai performant decât celălalt.

În concluzie, putem sublinia următoarele caracteristici ale utilizării curbelor ROC:

- ◆ Pun în evidență echilibrul dintre sensibilitate și specificitate (o creștere în sensibilitate va implica o scădere în specificitate);
- ◆ Cu cât curba ROC este mai apropiată de extremitatea stângă și de partea de sus, cu atât acuratețea predicției (clasificări/decizie) este mai bună;
- ◆ Cu cât curba ROC este mai apropiată de diagonală principală, cu atât acuratețea este mai slabă.
- ◆ Acuratețea este măsurată de aria aflată sub curba ROC. Astfel, aria egală cu 1 reprezintă modelul perfect, în timp ce aria de 0,5 înseamnă cel mai puțin performant model.

Figura de mai jos sintetizează aceste concluzii.



Intuitiv, putem stabili o scară cu 5 trepte (A-E), de exemplu, privind gradul de acuratețe, ținând cont de utilizarea curbelor ROC (i.e. aria de dedesubtul curbei). Astfel, avem:

- (A) $0,90 - 1,00$ = Predicție excelentă;
- (B) $0,80 - 0,90$ = Predicție bună;
- (C) $0,70 - 0,80$ = Predicție satisfăcătoare;
- (D) $0,60 - 0,70$ = Predicție nesatisfăcătoare;
- (E) $0,50 - 0,60$ = Predicție eronată.

6.4. Comparația statistică a performanțelor clasificării

În final, vom prezenta, pe scurt, o abordare statistică privind compararea performanțelor a două modele. Să presupunem deci că dispunem de două modele M_1 și M_2 , astfel încât:

- Modelul M_1 are acuratețea de 85%, obținută pe o mulțime de testare conținând 30 instanțe;
- Modelul M_2 are acuratețea de 75%, obținută pe o mulțime de testare conținând 5000 instanțe.

Întrebarea, naturală pe care ne-o punem în acest caz este: „Care model este mai performant?”. Greu de răspuns, intuitiv. Pe de-o parte, primul model are

acuratețea mai mare, dar pe de altă parte a fost testat pe o mulțime mai mică. Pentru a răspunde corect la această întrebare, avem nevoie de instrumente statistice, mai precis de *testul asupra diferenței a două proporții (difference between two proportions test)*, utilizat pentru calculul nivelului de semnificație privind diferența între două proporții (rapoarte). Algoritmul atașat testului este următorul:

Algoritm pentru compararea performanțelor (t-test)

Input:

- Se consideră prima proporție (raport) p_1 , corespunzătoare primului eșantion de volum N_1 ;
- Se consideră a doua proporție (raport) p_2 , corespunzătoare celui de-al doilea eșantion de volum N_2 ;

Nivelul de semnificație statistică P se calculează utilizând statistică t , corespunzătoare comparației, dată de:

$$|t| = \sqrt{\frac{N_1 \cdot N_2}{N_1 + N_2} \cdot \frac{|p_1 - p_2|}{\sqrt{p \cdot q}}},$$

$$\text{unde } p = \frac{p_1 \cdot N_1 + p_2 \cdot N_2}{N_1 + N_2}, \quad q = 1 - p.$$

Se utilizează tabelul statisticii t (Student) cu $(N_1 + N_2 - 2)$ grade de libertate, pentru calculul nivelului corespunzător P .

Output:

- Modelul cel mai performant.

Aplicând algoritmul de mai sus în cazul concret al comparației celor două modele, obținem nivelul de semnificație $P = 0.20$, ceea ce indică faptul că cele două modele sunt comparabile în ceea ce privește acuratețea deciziei (nicio diferență statistic semnificativă, deoarece $P > 0.05$). Pentru amănunte, a se revedea și testul „*t-test for independent samples*”.

Remarcă: Dacă în exemplul de mai sus considerăm că mulțimea de testare pentru primul model M_1 conține 100 instanțe în loc de doar 30 (deci o mai bună testare), atunci nivelul de semnificație corespunzător va fi $P = 0.02 (< 0.05)$, deci primul model este semnificativ mai performant decât al doilea.

În ceea ce privește intervalul de încredere pentru acuratețe, se utilizează repartitia binomială $B(N, p)$, bazată pe experimentele Bernoulli, independente și repetate, în sensul că, date fiind a predicții (decizii/clasificări) corecte pentru N instanțe în mulțimea de testare, acuratețea reală, care este egală cu raportul $\frac{a}{N}$, va fi repartizată binomial (p reprezintă acuratețea adevărată (teoretică) a modelului). Mai mult, în cazul în care N depășește pragul de 30 instanțe, această repartitie se poate aproxima suficient de bine cu o repartitie normală (gaussiană) de medie p și dispersie $p(1 - p)/N$.

Se știe că, pentru un nivel dat $\alpha \in (0,1)$, avem:

$$P\left(Z_{\alpha/2} < \frac{x - p}{\sqrt{p(1-p)/N}} < Z_{1-\alpha/2}\right) = 1 - \alpha,$$

deci, notând acuratețea reală obținută din testare cu ac , obținem intervalul (p_1, p_2) de încredere de nivel $(1 - \alpha)$ pentru adevărată acuratețe p :

$$p_{1,2} = \frac{2 \times N \times ac + Z_{\alpha/2}^2 \pm \sqrt{Z_{\alpha/2}^2 + 4 \times N \times ac - 4 \times N \times ac^2}}{2(N + Z_{\alpha/2}^2)}$$

Dacă, de exemplu, la testarea unui model obținem o acuratețe de 80% pe o mulțime de antrenament de 100 instanțe, atunci intervalul de încredere de nivel 95% (cel ușual) al acurateței va fi (71.1%, 86.6%).

O altă abordare statistică pentru compararea performanțelor a două modele este cea care se bazează pe estimarea intervalului de încredere al diferenței erorilor obținute pe mulțimile de testare corespunzătoare. Concret, fie modelele M_1, M_2, \dots, M_k , testate pe mulțimile de testare independente T_1, T_2, \dots, T_k , de volume N_1, N_2, \dots, N_k , obținându-se erorile e_1, e_2, \dots, e_k . Se poate arăta că, pentru n_i suficient de mari, erorile vor fi normal repartizate. Pentru fiecare pereche de două modele M_i și M_j , se calculează diferența $d = |e_i - e_j|$, care va fi normal repartizată $N(d_a, \sigma_a)$, unde d_a reprezintă adevărată diferență (teoretică), iar σ_a are expresia:

$$\sigma_a^2 = \sigma_i^2 + \sigma_j^2 \approx \frac{e_i(1-e_i)}{N_i} + \frac{e_j(1-e_j)}{N_j}.$$

În aceste condiții, intervalul de încredere de nivel $(1 - \alpha)$ va fi dat de:

$$d_{a,1,2} = d \pm Z_{\alpha/2} \cdot \sigma_a.$$

De exemplu, să presupunem că pentru modelul M_1 avem $N_1 = 30$ instanțe și $e_1 = 15\%$, iar pentru modelul M_2 avem $N_2 = 5000$ instanțe și $e_2 = 25\%$. Atunci $d = 10\%$ și, în consecință, $\sigma_a = 0.0043$, deci, în final, obținem că intervalul de

încredere de nivel 95% este dat de (0.10 ± 0.128) , diferența nefiind statistic semnificativă (nu conține valoarea zero), lucru verificat și folosind testul diferenței a două proporții. Astfel putem răspunde și la dilema prezentată la începutul acestui paragraf.

Remarcă: Toate testele statistice utilizate mai sus se referă la domeniul *inferenței statistice*, care se bazează pe utilizarea de eșantioane pentru estimarea parametrilor adevărați (teoretici) ai unor populații statistice (e.g. adevărata acuratețe). În aceste condiții, intervalele de încredere de un anumit nivel vor estima domeniul valorilor în care se va găsi, cu o anumită probabilitate (dată de nivelul de încredere), parametrul căutat. Reamintim că orice valoare din aceste intervale este echiprobabilă, de obicei alegându-se mijlocul intervalului (media).

INDEX

- abaterea medie pătratică*, 72
acuratețea, 27, 292
ADALINE, 190, 197
algoritmul back-propagation, 197
algoritm compararea
 performanțelor, 303
algoritm evolutiv, 264
algoritm genetic, 265
 exploatarea 288
 explorarea, 288
algoritmul k-means, 260
algoritmul k-nearest neighbor, 232
analiza canonica, 130
analiza componentelor principale, 147
analiza discriminant, 131
analiza factorială, 124
analiza supraviețuirii, 114
anomalie, 140
aproximare (mulțimi rough)
 inferioară, 238
 superioară, 238
aproximarea B-inferioară, 241
aproximarea B-superioară, 241
arbore de clasificare, 149
asimetria, 98
Bayes, formula, 43, 170
 regula de decizie, 202
căștigul de informație, 159
clasificarea, 24
clasificare naivă Bayes, 172
clustering, 30
 ierarhic, 247, 257
 neierarhic, partitional, 247, 259
coeficient de corelație, 77
costuri de clasificare greșită, 164, 292
covarianță, 77
cromozom, 268
cuantilă, 68
cuartilă, 68
curbă de învățare, 295
curbă ROC, 297
dată, 56
 numerică, 61
 categorială, 62
 cenzurată, 63, 115
decilă, 68
deviația standard, 72
dispersia, 71
distanța Chebychev, 253
distanța cosinus, 253
distanța Euclidiană, 253
distanța fuzzy, 255
distanța Hamming, 253
distanța Jaccard, 254
distanța Mahalanobis, 254
distanța Manhattan 253
distanța Minkowski, 253
distanța Tanimoto, 254
elbow criterion, 33
entropia, 159
excesul, 100
fasonarea unui arbore, 165
feedforward, 191
formula probabilității totale, 169
frontiera de decizie, 183
funcția de activare, 179
 adecvare/performanță, evaluare
265, 269
 gaussiană, 181
 Heaviside, 180
 liniară, 180
 liniară pe porțiuni, 180
 logistică, 181
 sigmoidă, 181
funcția discriminant, 219
funcție eroare, 188

- funcția de supraviețuire*, 115
funcția hazard, 116
genă, 268
greedy algorithm, 153
histograma, 87
hiperplan de separație, 218
Hopfield, 199
indexul GINI, 155
initializarea (GA), 268
interval de încredere, 72
învățare
 nesupervizată, 17
 supervizată, 17, 25
Kohonen, 198
MADALINE, 198
marginea de separație, 215, 219
mașini cu suport vectorial, 214
mașini cu nucleu, 216
matricea costurilor, 292
matricea covariantei, 128
matricea de discernabilitate, 237, 240
măsura de clasificare greșită, 161
măsură de similaritate, 252
media, 69
mediana, 69
model aditiv, 117
model de ajustare, 121
model autopredictiv, 122
modelul ARIMA (Box-Jenkins), 122
modelul de regresie Cox, 114
model explicativ, 122
model populational (GA)
 generațional, 274
 staționar, 274
modul, 70
mulțime
 antrenament, 25
 crisp, 242
 rough, 237, 242
 testare, 25, 39
mutația, 265, 281
 amestec, 284
 bit-flipping, 282
 deformare, 282
 inserare, 284
 inversiune, 284
 neuniformă, 283
 resetarea aleatoare, 282
 schimb, 284
 uniformă, 283
neuron, 176
nucleul produs-scalar, 218
Occam's Razor, 166
OLAP, 132
operatori de variație, 274
overfitting, 165
parametru
 cost, 202
 de ajustare, 202
 pierdere, 202
 „părinte”, 271
percentilă, 68
perceptron, 181
 algoritmul de convergență, 187
 algoritm de învățare, 182
 criteriul perceptronului, 189
 cu un singur strat, 197
 multi-strat, 197
 teorema de convergență, 184
performanța totală, 270
populația, 270
problemă de tip XOR, 224
probabilitate apriorică, 170, 202
probabilitate posteroiară, 170
probabilități prealabile, 164
programarea evolutivă, 266
programarea genetică, 266
radial basis function, 198
recombinarea (încruzișarea), 265, 275
 aplicată parțială, 277
 aritmetică simplă, 277
 aritmetică totală, 277
 aritmetică unică, 277
 ciclică, 281
 cu un punct de tăietură, 275
 cu N-tăieturi, 276

- extremităților*, 278
- de ordine*, 280
- uniformă*, 276
- redus relativ*, 243
- regiune de frontieră*, 237, 238, 242
- regresia*, 36, 79
- regresie liniară multiplă*, 103
- regresia logistică*, 111
- regula delta*, 197
- reguli de asociere*, 33, 227
 - acuratețea*, 229
 - puterea de acoperire*, 229
- relație de indiscernabilitate*, 239
- relație indusă*, 239
- repartiția de probabilitate*, 67
- repartiția exponențială*, 101
- repartiția gamma*, 101
- repartiția log-normală*, 102
- repartiția marginală*, 73
- repartiția Weibull*, 102
- reprezentarea (GA)*, 268
- rețea neuronală*,
 - feedforward cu un singur strat*, 192
 - feedforward multistrat*, 192
 - învățare*, 193
 - auto-organizată*, 196
 - consolidată*, 195
 - supervizată*, 194
 - recurentă*, 193, 199
 - probabilistă*, 201
 - algoritm de antrenare*, 205
- robustețea*, 27
- scalabilitatea*, 27
- selecția*
 - evaluare*, 273
 - ordonare*, 271
 - proporțională*, 271
- ruleta (Monte Carlo)*, 272
- turnir*, 271
- universală stochastică*, 272
- vârstă*, 273
- senzitivitatea*, 294
- serie dinamică*, 118
- serie temporală*, 118
- sistem de decizie*, 239
- sistem de informații*, 238
- soluție candidat*, 265
- specificitatea*, 294
- strat*, 192
 - ascuns*, 192
- strategie evolutivă*, 266
- tabel al confuziei*, 295
- taxonomia*, 24
- spațiul caracteristicilor*, 217
- TSP (travelling salesman problem)*, 153, 268
- underfitting*, 165
- univers (RS)*, 239
- validare*, 45
- variabile de relaxare*, 222, 225
- variabilă predictor*, 79
- variabilă răspuns*, 79
- varianța*, 71
- vector input*, 179
- vector pondere*, 179
- vector suport*, 215, 220
- verosimilitate*, 145, 170

BIBLIOGRAFIE

1. **Abonyi, J., Feil, B.**, *Cluster analysis for data mining and system identification*, Birkhäuser, 2007.
2. **Ackley, D.**, *A connectionist machine for genetic hillclimbing*, Kluwer Academic Publishers, 1987.
3. **Adam, J.M.**, *Data Mining for association rules and sequential patterns: sequential and parallel algorithms*, Springer, 2001.
4. **Agrawal, R., Imielinski, T., Swami, A.**, *Mining association rules between sets of items in large databases*. Proc. 1993 ACM SIGMOD Conference, Washington, 207-216, 1993.
5. **Agrawal, R., Srikant, R.**, *Fast Algorithms for Mining Association Rules in Large Databases*, Proc. 20th International Conference on Very Large Data Bases, 487-499, Morgan Kaufmann, 1994.
6. **Agrawal, R., Srikant, R.**, *Mining sequential patterns*, Proc. 11th ICDE 1995, Taipei, Taiwan, March 1995.
7. **Agrawal, R., Srikant, R.**, *Mining sequential patterns: Generalizations and performance improvements* (In P. M. G. Apers, M. Bouzeghoub, and G. Gardarin, eds.), EDBT 1996, pp. 3-17, 1996.
8. **Aldrich, J.H., Nelson, F.D.**, *Linear probability, logit and probit models*, Sage University Papers, 1985.
9. **Allen, D.M.**, *Mean square error of prediction as a criterion for selecting variables*, Technometrics, 13, pp. 469-475, 1971.
10. **Altman, D.G.**, *Practical statistics for medical research*, Chapman and Hall, 1991.
11. **Altman, D.G., Bland, J.M.**, *Diagnostic tests. 1: Sensitivity and specificity*, BMJ, 308, 1552 (11 June), 1994.
12. **Altman, D.G., Bland, J.M.**, *Diagnostic tests 2: Predictive values*, BMJ, 309, 102 (9 July), 1994.

13. **Andersen, R.**, *Modern Methods for Robust Regression*, Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-152, Thousand Oaks, CA: Sage, 2008.
14. **An, L., Tong, L.**, *Similarity, Boolean Reasoning and Rule Induction*, Proc. Second International Symposium on Intelligent Information Technology Application vol. 1, 7-12, 2008.
15. **Anderson, J. A.**, *An introduction to Neural Networks*, MIT Press, 1995.
16. **Aris, R.**, *Mathematical modelling techniques*, Pitman, 1978.
17. **Aspray, W., Burks, A.**, *Papers of John von Neumann on computing and computer theory*, Charles Babbage Institute Reprint Series for the history of computing, Vol. 12, Cambridge, MA: MIT Press, 1986.
18. **Bäck, T.**, *Self-adaptation*. Ch. 21, pp. 188-211 (Bäck T., Fogel D.B., Michalewicz Z., Eds. *-Evolutionary computation 2: advanced algorithms and operators*), Institute of Physics Publishing, 2000.
19. **Bäck, T., Fogel D.B., Michalewicz Z.** (Eds), *Evolutionary computation 1: basic algorithms and operators*, Institute of Physics Publishing, Bristol, 2000
20. **Baker, J.E.**, *Reducing bias and inefficiency in the selection algorithm*, Proc. 2nd International Conference on Genetic Algorithms and Their Applications (Grefenstette J.J. Ed.), Lawrence Erlbaum, Hillsdale, New Jersey, 14-21, 1987.
21. **Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D.**, *Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications*, Morgan Kaufmann, 1998.
22. **Barricelli, N.A.**, *Esempi numerici di processi di evoluzione*, Methodos, pp. 45-68, 1954.
23. **Barto, A.G., Sutton, R.S., Anderson, C.W.**, *Neuronlike adaptive elements that can solve difficult learning control problems*, IEEE Transactions on Systems, Man, and Cybernetics, SMC-13, pp. 834-846, 1983.
24. **Bates, D.M., Watts, D.G.**, *Nonlinear Regression Analysis and Its Applications* (Wiley Series in Probability and Statistics), Wiley, 1988.

25. **Bayes, Th.**, *An Essay towards solving a Problem in the Doctrine of Chances*, Philosophical Transactions of the Royal Society of London 53: 370-418, 1763.
26. **Bădulescu F., Gorunescu F.**, *Informatică oncologică: Metode statistico-informatice în oncologie*, Ed. Didactică și Pedagogică, București, 2003.
27. **Beachley, N.H., Harrison, H.L.**, *Introduction to dynamic systems analysis*, Harper & Row, 1978.
28. **Belciug, S., Gorunescu, F.**, *A hybrid neural network/genetic algorithm applied to the breast cancer detection and recurrence*, Expert Systems, 2010 (under revision).
29. **Bender, E.A.**, *An introduction to mathematical modeling*, John Wiley, 1978.
30. **Berkhin, P.**, *Survey of clustering data mining techniques*, Technical report, Accrue Software, San Jose, California, 2002. (<http://citeseer.nj.nec.com/berkhin02survey.html>).
31. **Berry, M., Linoff, G.**, *Mastering Data Mining*, John Wiley & Sons, 2000.
32. **Bezdek, J.C.**, *Pattern recognition with fuzzy objective function algorithms*, Plenum Press, New York, 1981.
33. **Beyer, H.G.**, *The theory of evolution strategies*, Springer, 2010.
34. **Bishop, C.**, *Neural networks for pattern recognition*, Oxford University Press, 1995.
35. **Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.K.**, *Occam's razor*, Information Processing Letters (Elsevier North-Holland, Inc.), 377-380, 1987.
36. **Booker, L., Fogel, D., Whitley, D., Angeline, P., Eiben, A.**, *Recombination*, Handbook of Evolutionary Computation, 1997.
37. **Boring, S.**, *Sensitivity, specificity, and predictive value*, Clinical Methods: The History, Physical, and Laboratory Examinations (3rd Ed.), (Walker H.K., Hall W.D., Hurst J.W. -Eds.), Ch. 6, Butterworths, 1990.

38. **Borji, A., Hamidi, M.**, *Evolving a fuzzy rule-base for image segmentation*, World Academy of Science, Engineering and Technology, 28, 4-9, 2007.
39. **Boser, B., Guyon, I., Vapnik, V.N.**, *A training algorithm for optimal margin classifiers*, Proc. 5th Annual Workshop on Computational Learning Theory, San Mateo, California, USA, pp. 144-152, 1992.
40. **Box, G.E.P., Jenkins, G.M.**, *Time series analysis: Forecasting and control*. San Francisco: Holden-Day, 1976.
41. **Box G.E.P.**, *Evolutionary operation: a method of increasing industrial productivity*, Applied Statistics, 6, pp. 81-101, 1957.
42. **Boyce B.R., Meadow C.T., Kraft D.H.**, *Measurement in information science*, Academic Press, 1994.
43. **Brause, R., Langsdorf, T. Hepp, M.**, *Neural Data Mining for Credit Card Fraud Detection*, Proc. 11th IEEE International Conference on Tools with Artificial Intelligence, 103, 1999.
44. **Breiman L., Friedman J. H., Olshen R. A., Stone C. J.**, *Classification and regression trees*, Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software, 1984.
45. **Bremermann H.J.**, *Optimization through evolution and recombination*, Self-Organizing Systems, 1962, (M.C. Yovits, G.T. Jacobi, and G.D. Goldstein Eds.), Spartan Books, Washington DC, pp. 93-106, 1962.
46. **Brown, M.P., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C.W., Furey, T.S., Ares, M. Jr., Haussler, D.**, *Knowledge-based Analysis of Microarray Gene Expression Data by Using Support Vector Machines*, Proc. of the National Academy of Sciences USA, 97(1), 262-267, 2000.
47. **Bull, L., Kovacs, T.**, *Foundations of Learning Classifier Systems: An Introduction*, Studies in Fuzziness and Soft Computing, 183, Springer, 2005.
48. **Bull, L., Bernardo-Mansilla, E., Holmes, J.**, *Learning Classifier Systems in Data Mining: An Introduction*, Studies in Computational Intelligence, 125, Springer, 2008.

49. **Burbridge, R., Trotter, M., Buxton, B., Holden, S.,** *Drug Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis*, Computers and Chemistry, 26(1), 5-14, 2001.
50. **Burges, C.,** *A tutorial on Support Vector Machines for pattern recognition*, Data Mining and Knowledge Discovery, 2, 121–167, Kluwer Academic Publishers, Boston, 1998.
51. **Cabena P., Hadjinian P., Stadler R., Verhees J., Zanasi A.,** *Discovering Data Mining from Concept to Implementation*, Prentice Hall, 1997.
52. **Cacoullos, T.,** *Estimation of a multivariate density*”, Ann. Inst. Stat. Math (Tokyo), 18, 179-189, 1966.
53. **Cazzaniga, M., Salerno, F., Borroni, G., Ceriani, R., Stucchi, G., Guerzoni, P., Casiraghi, M.A., Tommasini, M.,** *Prediction of asymptomatic cirrhosis in chronic hepatitis C patients: accuracy of artificial neural networks compared with logistic regression models*, Eur. J. Gastroenterol. Hepatol., vol. 21(6), pp. 681-687, Jun, 2009.
54. **Chaudhuri S., Dayal U.,** *Data Warehousing and OLAP for Decision Support*, ACM SIGMOD Int. Conf on Management of Data, ACM Press, Tucson, Arizona, 1997.
55. **Chen, H., Wang, X.F., Ma, D.Q., Ma, B.R.,** (2007) *Neural network-based computer-aided diagnosis in distinguishing malignant from benign solitary pulmonary nodules by computed tomography*, Chin. Med. J. (Engl.), 120(14), 1211-1215, 2007.
56. **Cheng, C-H., Chen, Jr-S.,** *Diagnosing cardiovascular disease using an enhanced rough sets approach*, Applied Artificial Intelligence, 23(6), 487-499, 2009.
57. **Cheng, J-H., Chen, H-P., Cheng, K-L.,** *Business failure prediction model based on grey prediction and rough set theory*, WSEAS Transactions on Information Science and Applications, 6(2), 329-339, 2009.
58. **Cherkassky V, Mulier F,** *Learning from data: concepts, theory and methods*, Wiley, 1998.

59. **Chi, C.L., Street, W.N., Wolberg, W.H.**, *Application of artificial neural network-based survival analysis on two breast cancer datasets*, AMIA Annu. Symp. Proc., 11, 130-134, 2007.
60. **Chiu, J.S., Wang, Y.F., Su, Y.C., Wei, L.H., Liao, J.G., Li, Y.C.** *Artificial neural network to predict skeletal metastasis in patients with prostate cancer*, J. Med. Syst., 33(2), 91-100, Apr 2009.
61. **Christensen, T., Neuberger, J., Crowe, J., et al.**, Beneficial effect of azathioprine and prediction of prognosis in primary biliary cirrhosis: final results of an international trial, *gastroenterology*, 89, 1084-1091, 1985.
62. **Clark P., Niblett, T.**, *The CN2 Induction Algorithm*, Machine Learning, 3(4), pp. 261-283, 1989.
63. **Cliff N., Krus D.J.**, *Interpretation of canonical variate analysis: Rotated vs. unrotated solutions*, *Psychometrika*, 41(1), pp. 35-42, 1976.
64. **Cochran, W. G.**, *Sampling Techniques*, John Wiley & Sons, 1977.
65. **Cock De, M., Cornelis, C., Kerre, E.E.**, *Fuzzy Rough Sets: Beyond the Obvious*, Proc. FUZZIEEE2004 (2004 IEEE International Conference on Fuzzy Systems), 1, 103–108, 2004.
66. **Codd, E.F., Codd, S.B., Salley, C.T.** "Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate", Codd & Date, Inc., 1993.
67. **Cohen W.**, *Fast Effective Rule Induction*, Proceedings of the 12th Intl. Conf. on Machine Learning, Tahoe City, CA, USA, pp. 115-123, 1995.
68. **Cooley, W.W., Lohnes, P.R.**, *Multivariate data analysis*. New York: Wiley, 1971.
69. **Cortes C., Vapnik V.N.**, *Support vector networks*, Machine Learning, Vol. 20, pp. 273-297, 1995.
70. **Cost S., Salzberg S.**, *A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features*, Machine Learning, 10(1), pp. 57-78, 1993.
71. **Cover T.M.**, *Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition*, IEEE Transactions on Electronic Computers, Vol. EC-14, pp. 326-334, 1965.

72. **Cox D.R., Miller D.**, *The theory of stochastic processes*, Methuen, 1965.
73. **Cox D.R.**, *Regression models and life tables*, J. Roy. Statist. Soc. B, 34, pp. 187-202, 1972.
74. **Cristianini, N., Shawe_Taylor, J.**, *An introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000.
75. **Darlington, R.B.**, *Regression and linear models*, New York: McGraw-Hill, 1990.
76. **Darwiche, A.**, *Modeling and reasoning with Bayesian Networks*, Cambridge University Press, 2009.
77. **Darwin, C.**, *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life* (1st ed.), London: John Murray, 1859.
78. **Dasarathy B.V.**, (Eds.) *Nearest Neighbour Norms: Nearest Neighbour Pattern Classification Techniques*, IEEE Computer Society Press, 1991.
79. **Davis, L.**, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.
80. **De Jong, K.**, *An analysis of the behaviour of a class of genetic adaptive systems*, PhD thesis, University of Michigan, 1975.
81. **De Jong, K.**, *Evolutionary computation*, MIT Press, 2002.
82. Delgado, M., M., Pegalajar, M.C., Cuellar, M.P., *Memetic evolutionary training for recurrent neural networks: an application to time-series prediction*, Expert Systems, 23(2), 99-115, 2006.
83. **Domingos, P.**, *The Role of Occam's Razor in Knowledge Discovery*, Data Mining and Knowledge Discovery, 3, 409-425, 1999.
84. **Draper N.R., Smith H.**, *Applied regression analysis* (2nd ed.), John Wiley, 1981.
85. **Drucker, H., Burges, Ch.J.C., Kaufman, L., Smola, A., Vladimir Vapnik, V.**, *Support Vector Regression Machines*, Advances in Neural Information Processing Systems 9, NIPS 1996, 155-161, MIT Press, 1997.

86. **Dubois, D., Prade, H.**, *Rough fuzzy sets and fuzzy rough sets*, *International Journal of General Systems*, 17, 191–209, 1990.
87. **Dubois, D., Prade, H.**, *Putting rough sets and fuzzy sets together*, Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory, 203-232, Kluwer Academic Publishers, Dordrecht 1992.
88. **Duda R.O., Hart P.E.**, *Pattern classification and scene analysis*, John Wiley, 1973.
89. **Duda R.O., Hart P.E., Stork D.H.**, *Pattern Classification* (2nd ed.), Wiley Interscience, 2001.
90. **Dumitrescu D., Costin H.**, *Rețele neuronale*, Teora, București, 1996.
91. **Dumitrescu D.**, *Algoritmi genetici și strategii evolutive – aplicatii în Inteligența Artificială și în domenii conexe*, Editura Albastră Cluj-Napoca, 2000.
92. **Dumitrescu D., Stoean C.**, *Genetic Chromodynamics Metaheuristic for Multimodal Optimization*, WSEAS Transactions on Information Science and Application, Issue 8, Volume 3, pp. 1444-1452, 2006.
93. **Dunteman, G.H.**, *Principal Components Analysis*, Series: Quantitative applications in the social science, Sage University Paper, Vol. 69, 1989.
94. **Dym C.L., Ivey E.S.**, *Principles of mathematical modeling*, Academic Press, 1980.
95. **Dzubera, J., Whitley, D.**, *Advanced correlation analysis of operators for the Traveling Salesman Problem*, Proc. 3rd Conference on Parallel Problem Solving from Nature/International Conference on Evolutionary Computation, Lectures Notes In Computer Science (Davidor Y., Schwefel H-P., Männer R. –Eds.), 866, 68-77, Springer, 1994.
96. **Eiben, A.E., Schippers, C.A.**, *On evolutionary exploration and exploitation*, *Fundamenta Informaticae* 35(1-4), 35-50, 1998.
97. **Eiben A.E., Hinterding R., Michalewicz Z.**, *Parameter control in evolutionary algorithms*, *IEEE Transactions on Evolutionary Computation*, 3(2), pp. 124-141, 1999.

98. **Eiben A.E.**, *Multiparent recombination in evolutionary computing* (Gosh A., Tsutsui S. -Eds.) –*Advances in evolutionary computation: theory and applications*), Springer Verlag, 175-192, 2003.
99. **Eiben A.E., Smith J.E.**, *Introduction to Evolutionary Computing*, Springer, 2003.
100. **Eiben, A.E., Michalewicz, Z., Schoenauer, M., Smith, J.E.**, *Parameter Control in Evolutionary Algorithms*, Studies in Computational Intelligence, 54, Springer 2007.
101. **Egan, J.P.**, Signal detection theory and ROC analysis, Series in Cognition and Perception. Academic Press, New York, 1975.
102. **El-Darzi E., Gorunescu M., Gorunescu F.**, *A non-homogeneous kinetic model in the patients flow modelling*, Proceedings The First East European Conference on Health Care Modelling and Computation, -HCMC2005, Craiova, Romania, August 31st-September 2nd (Eds. F. Gorunescu, M. Gorunescu, E. El-Darzi), Medical University Press, pp. 99-106, 2005.
103. **El-Darzi, E., Abbi, R., Vasilakis, C., Gorunescu, F., Gorunescu, M., Millard, P.**, *Length of stay-based clustering methods for patient grouping*, Intelligent Patient management (Studies in Computational Intelligence (McClean S., Millard P., El-Darzi E., Nugent C. Eds.), 189, Springer, 2009.
104. **El-Gamal, M.A., Mohamed, M.D.**, *Ensembles of Neural Networks for Fault Diagnosis in Analog Circuits*, Journal of Electronic Testing: Theory and Applications, 23(4), 323 – 339, 2007.
105. **Elsner, U.**, *Static and Dynamic Graph Partitioning. A comparative study of existing algorithms*, Logos, 2002.
106. **English T.M.**, *Evaluation of evolutionary and genetic optimizers: no free lunch* (Fogel L.J., Angeline P.J., Bäck Eds. –*Proceedings of the 5th Annual Conference on Evolutionary Programming*), MIT Press, Cambridge MA, pp. 163-169, 1996.
107. **Everitt, B., Landau, S., Leese, M.**, *Cluster analysis*, Wiley, 2001.
108. **Fausett, L.V.** *Fundamentals of Neural Networks: architectures, algorithms and applications*, Prentice Hall, 1993.

109. **Fawcett, T.**, *An introduction to ROC analysis*, Pattern Recognition Letters 27, 861–874, 2006.
110. **Fayyad U., Piatetsky-Shapiro G., Smyth P., Uthurusamy R.**, *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, 1996.
111. **Fibak, J., Pawlak, Z., Slowinski, K., Slowinski, R.**, *Rough sets based decision algorithm for treatment of duodenal ulcer by HSV*, Bull. Polish Acad. Sci. Biological Sci. 34/10-12, 227- 246, 1986.
112. **Fisher D.**, *Improving Inference through Conceptual Clustering*, Proceedings of 1987 AAAI Conferences, Seattle Washington, pp. 461-465, 1987.
113. **Fisher, D.**, *Knowledge acquisition via incremental conceptual clustering*, Machine Learning, 2, 139–172, 1987.
114. **Fisher R.A.**, *On the mathematical foundations of theoretical statistics*. Philosophical Transactions of the Royal Society, A, 222, pp. 309-368, 1922.
115. **Fisher R.A.** "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188, 1936.
116. **Foucart T., Bensaber A., Garnier R.**, *Méthodes pratiques de la Statistique*, Masson, 1987.
117. **Fraser A.S.**, *Simulation of Genetic Systems by Automatic Digital Computers. I. Introduction*. Australian Journal of Biological Sciences vol. 10, pp. 484-491, 1957.
118. **Frawley W., Piatetsky-Shapiro G., Matheus C.**, *Knowledge Discovery in Databases: An Overview*, AI Magazine, Fall 1992.
119. **French, M.N., Krajewski, W.F., Cuykendall, R.R.**, *Rainfall forecasting in space and time using a neural network*, Journal of Hydrolgy, 137(1-4), 1-31, 1992.
120. **Freund J.**, *Modern elementary statistics* (11th ed.), Pearson/Prentice Hall, 2004.
121. **Fukunaga K.**, *Introduction to statistical pattern recognition*, Academic Press, 1990.

122. **Gao, X., Zhao, R., Qi, C., Shi, Z.**, *A rough set method to treat ECG signals for predicting coronary heart disease*, Journal of Biomedical Engineering, 25(5), 1025-1028, 2008.
123. **Gan, G., Ma, C., Wu, J.**, *Data clustering: Theory, algorithms, and applications*, SIAM, 2007.
124. **Garcia-Orellana, C.J., Gallardo-Caballero, R., Macias-Macias, M., Gonzalez-Velasco, H.**, *SVM and neural networks comparison in mammographic CAD*, Conf. Proc. IEEE Eng. Med. Bio. Soc., 3204-3207, 2007.
125. **Garcia-Laencina, P.J., Sancho-Gomez, J.L., Figueiras-Vidal, A.R., Verleysen, M.**, *K nearest neighbours with mutual information for simultaneous classification and missing data imputation*, Neurocomputing, 72 (7-9), 1483-1493, 2009.
126. **Gaver D.P, Kobayashi H., Sedgewick R.**, *Probability Theory and Computer Science*, International Lecture Series in Computer Science (Louchard G., Latouche G. -Eds.), Academic Press, 1983.
127. **Gilchrist W.**, *Statistical modeling*, John Wiley, 1984.
128. **Glasziou, P., Del Mar, C., Salisbury, J.**, *Evidence-based medicine workbook*, BMJ Publishing Group, 2003.
129. **Glesner, M., Pöchmüller, W.**, *Neurocomputers: An Overview of Neural Networks in VLSI*, Chapman & Hall, 1994.
130. **Glück, R., Klimov, A.V.**, *Occam's Razor in Metacomputation: the Notion of a Perfect Process Tree*, Proc. Third International Workshop on Static Analysis (Lecture Notes In Computer Science, Vol. 724), 112-123, Springer-Verlag, 1993.
131. **Goh, C., Law, R., Mok, H.**, *Analyzing and forecasting tourism demand: A rough sets approach*, Journal of Travel Research, 46(3), 327-338, 2008.
132. **Goldberg D.E.**, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Massachusetts, 1989.
133. **Goldberg D.E., Lingle R.**, *Alleles, loci, and the travelling salesman problem* (Grefenstette J.J., Ed. -*Proceedings of 1st International Conference on Genetic Algorithms and Their Applications*), Lawrence Erlbaum, Hillsdale, New Jersey, pp. 154-159, 1985.

134. **Goldberg D.E.**, *Sizing Populations for Serial and Parallel Genetic Algorithms*, Addison-Wesley, 1989.
135. **Goldstein B.J., Mushlin A.I.**, *Use of single thyroxine test to evaluate ambulatory medical patients for suspected hypothyroidism*. J. Gen. Intern. Med., 2, pp. 20-24, 1987.
136. **Gordon A. D.**, *Classification* (2nd ed.), Chapman and Hall, 1999.
137. **Gorsuch, R.L.**, *Factor analysis* (2nd ed.), Lawrence Erlbaum Assoc. Inc., Hillsdale, NJ, 1983.
138. **Gorunescu F., Prodan A.**, *Modelare stochastică și simulare*, Editura Albastră Cluj-Napoca, 2001.
139. **Gorunescu F., Găman A., Găman G., Gorunescu M., Ciurea T., Ciurea P.**, *A dynamic forecasting model predicting the long-term response in the therapy of myelodysplastic syndromes*, Proceedings International Conference “Leukemia towards the cure 2002” –Miami, USA, 19-22 September, 2002, Leukemia and Lymphoma J.–special issue, pp. 86. 2002.
140. **Gorunescu F., Găman G., Găman A., Gorunescu M., Ciurea T., Ciurea P.**, *A classification and regression tree technique applied in acute leukemia*, Proceedings International Conference “Leukemia towards the cure 2002” –Miami, USA, 19-22 September, 2002, Leukemia and Lymphoma J.–special issue, pp. 77, 2002.
141. **Gorunescu F.**, *Architecture of Probabilistic Neural Networks: estimating the adjustable smoothing factor*, Proceedings of the 4th International Conference on Artificial Intelligence and Digital Communications, Research Notes in Artificial Intelligence and Digital Communications (Subseries of Research Notes in Computer Science), 104, pp. 56-62, 2004.
142. **Gorunescu F., Gorunescu M., El-Darzi E., Gorunescu S.**, *An Evolutionary Computational Approach to Probabilistic Neural Network with Application to Hepatic Cancer Diagnosis*, Proceedings The 18th IEEE International Symposium on Computer-Based Medical Systems – IEEE CBMS 2005- Dublin, Ireland, June 23-24, IEEE Computer Science Press (A. Tsymbal and P. Cunningham - Eds.), 461-466, 2005a.
143. **Gorunescu F.**, *Benchmarking Probabilistic Neural Network algorithms*, Proceedings of the 6-th International Conference on Artificial Intelligence and Digital Communications, Thessaloniki, Greece, 18-21 August 2006,

(Research Notes in Artificial Intelligence and Digital Communications - Subseries of Research Notes in Computer Science), 106, pp.1-7, 2006.

144. **Gorunescu F.**, *Clustering evolutiv: o abordare euristică a grupării optime a pacienților*, Craiova Medicală, Vol. 5, Nr. 2, pp. 350-352, 2003a.
145. **Gorunescu F., Gorunescu M.**, *A classification trees support system for hospitals admissions modelling. Alternative genetic algorithm based approach*, Proceedings of the 2nd National Conference on Artificial Intelligence and Digital Communications, Research Notes in Artificial Intelligence and Digital Communications (Subseries of Research Notes in Computer Science), 102, pp. 29-36, 2002.
146. **Gorunescu F., Gorunescu M.**, *A heuristic approach in cancer research using evolutionary computation*, International Journal of Computer Research (Nova Science), vol. 12 (4), pp. 59-65, 2005b.
147. **Gorunescu F., Gorunescu M., El-Darzi E., Ene M.**, *A Probabilistic Neural Network-based method for hepatic diseases diagnosis*, Buletinul Stiințific, Universitatea din Pitești, Seria Matematică și Informatică, Nr. 11, pp. 59-66, 2005c.
148. **Gorunescu F., Gorunescu M., El-Darzi E., Ene M.**, *Comparing classifiers: Probabilistic Neural Networks vs. k-means clustering*, Proceedings of the 5th International Conference on Artificial Intelligence and Digital Communications, Research Notes in Artificial Intelligence and Digital Communications (Subseries of Research Notes in Computer Science, Annals University of Craiova), 105, pp. 6-11, 2005d.
149. **Gorunescu F., Gorunescu M., El-Darzi E., Ene M., Gorunescu S.**, *Performance enhancement approach for Probabilistic Neural Networks*, Proceedings The First East European Conference on Health Care Modelling and Computation, -HCMC2005, Craiova, Romania, August 31st-September 2nd (F. Gorunescu, M. Gorunescu, E. El-Darzi - Eds.), Medical University Press, pp. 142-148, 2005e.
150. **Gorunescu F., Gorunescu M., El-Darzi E., Ene M., Gorunescu S.**, *Statistical Comparison of a Probabilistic Neural Network Approach in Hepatic Cancer Diagnosis*, Proceedings Eurocon2005 –IEEE International Conference on “Computer as a tool”, Belgrade, Serbia, November 21-24, pp. 237-240, 1-4244-0049-X/05/\$20.00 ©2005 IEEE, 2005f.
151. **Gorunescu F., Gorunescu M., El-Darzi E., Gorunescu S., Revett K., Khan A.**, *A Cancer Diagnosis System Based on Rough Sets and*

Probabilistic Neural Networks, Proceedings The First East European Conference on Health Care Modelling and Computation, -HCMC2005, Craiova, Romania, August 31st-September 2nd (F. Gorunescu, M. Gorunescu, E. El-Darzi - Eds.), Medical University Press, pp. 149-159, 2005g.

152. **Gorunescu F., Gorunescu M., Gorunescu R.,** *A genetic algorithm approach to cancer research*, Proceedings 1st MEDINF International Conference on Medical Informatics & Engineering, 9-11 Oct. 2003, Craiova Medicala –International issue, Vol 5, Sup. 3, pp. 129-132, 2003.
153. **Gorunescu F., Gorunescu M., Gorunescu R.,** *A metaheuristic GAs method as a decision support for the choice of cancer treatment*, Siberian Journal of Numerical Mathematics (Siberian Branch of the Russian Academy of Science –Novosibirsk), Vol. 7, No. 4, pp. 301-307, 2004.
154. **Gorunescu F.,** *K-means clustering: o abordare euristică în eficientizarea tratamentelor*, Craiova Medicală, Vol. 5, Nr. 3, pp. 421-423, 2003b.
155. **Gorunescu F.,** *Measuring similarities: an application to the chromosomes supervised selection*, Proceedings of the 3rd International Conference on Artificial Intelligence and Digital Communications, Research Notes in Artificial Intelligence and Digital Communications (Subseries of Research Notes in Computer Science,), 103, pp. 56-66, 2003c.
156. **Gorunescu F., Săftoiu A., Gorunescu M., Ciurea T.,** *Analiza discriminant în evaluarea tratamentului cu interferon a hepatitei cronice C*, Craiova Medicală, Vol. 3, Nr. 1, pp. 11-13, 2001.
157. **Gorunescu F., Săftoiu A., Gorunescu M., Ciurea T.,** *Logistic modeling in the assesment of the interferon treatment of chronic hepatitis C*, Craiova Medicală, Vol. 2, Nr. 3, pp. 169-172, 2000.
158. **Gorunescu F., Tiță I., Gorunescu M.,** *Forecasting the growth of phytolacca americana sowed in the fall*, Phytologia Balcanica -Sofia, Vol. 9(1), pp. 93-95, 2003.
159. **Gorunescu M. , Gorunescu F., Ene M., El-Darzi E.,** *A heuristic approach in hepatic cancer diagnosis using a probabilistic neural network-based model*, Proceedings of The 11th Applied Stochastic Models and Data Analysis –ASMDA 2005, Brest, France, 17-21 May, 2005, (J. Janssen and P. Lenca - Eds.), pp. 1016-1025, 2005.

160. **Gorunescu, M., Gorunescu, F., Revett, K.,** *Investigating a Breast Cancer Dataset Using a Combined Approach: Probabilistic Neural Networks and Rough Sets*, Proc. 3rd ACM International Conference on Intelligent Computing and Information Systems ICICIS07 - March, 15-18, 2007, Cairo, Egypt, Police Press, 246-249, 2007.
- Gorunescu, F., Gorunescu, M., Săftoiu, A., Vilmann, P., Belciug, S.:** *Competitive/Collaborative Neural Computing System for Medical Diagnosis in Pancreatic Cancer Detection*. Expert Systems (The Journal of Knowledge Engineering), Willey&Blackwell, DOI: 10.1111/j.1468-0394.2010.00540.x, (2010). (a)
- Gorunescu, F., El-Darzi, E., Belciug, S., Gorunescu, M.:** *Patient grouping optimization using a hybrid Self-Organizing Map and Gaussian Mixture Model for length of stay-based clustering system*. Proc. IEEE International Conference on Intelligent Systems –IS’2010, 173--178, (2010). , London, UK, 2010.
161. **Gourieroux Ch., Monfort A.,** *Time series and dynamic models*, Cambridge University Press, 1997.
162. **Gower J. C., Legendre P.,** *Metric and Euclidean properties of dissimilarity coefficients*, Journal of Classification 3, pp. 5-48, 1986.
163. **Green, D.M., Swets, J.M.,** *Signal detection theory and psychophysics*. New York: John Wiley and Sons Inc., 1966.
164. **Groth R.,** *Data Mining. A hands-on approach for business professionals*, Prentice Hall, 1998.
165. **Grubbs F.,** *Procedures for Detecting Outlying Observations in Samples*, Technometrics, Vol. 11, No. 1, pp. 1-21, 1969.
166. **Grünwald, P.,** *The Minimum Description Length principle*, MIT Press, June 2007.
167. **Grzymala J., Siddhave S.,** *Rough set Approach to Rule Induction from Incomplete Data*, Proceeding of the IPMU’2004, The 10th International Conference on information Processing and Management of Uncertainty in Knowledge-Based System, 2004.
168. **Gurney, K.,** *An introduction to Neural Networks*, CRC Press, 1997.

169. **Guyon, I., Weston, J., Barnhill, S., Vapnik, V.**, *Gene Selection for Cancer Classification using Support Vector Machines*, Machine Learning, 46(1-3), 2002.
170. **Guyon, I., Elisseeff, A**: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157-1182, 2003.
171. **Halkidi, M., Batistakis, Y., Vazirgiannis, M.**, *On clustering validation techniques*, Journal of Intelligent Information Systems, 17(2/3), 107–145, Kluwer Academic Publishers, 2001
172. **Hammer, B.**, *Learning with Recurrent Neural Networks* (Lecture Notes in Control and Information Sciences), Springer, 2000.
173. **Han J, Kamber M.**, *Data Mining – Concepts and Techniques* (2nd Ed.), Morgan Kaufmann Publishers, 2006.
174. **Han, J., Pei, J., Yin, Y., Mao, R.**, *Mining frequent patterns without candidate generation*, Data Mining and Knowledge Discovery 8, 53-87, 2004.
175. **Hand D., Mannila H., Smyth P.**, *Principles of Data Mining*, MIT Press, Cambridge, MA, 2001.
176. **Hand, D., Yu, K.**, *Idiot's Bayes - not so stupid after all?*, International Statistical Review. 69 (3), 385-399, 2001.
177. **Hanley, J.A., McNeil, B.J.**, *The meaning and use of the area under a receiver operating characteristic (ROC) curve*, Radiology 143, 29– 36, 1982.
178. **Hansen, L.P., Sargent, T.J.**, *Robustness*, Princeton University Press, 2007.
179. **Harris S.**, Total Information Awareness official responds to criticism, *Government Executive.com*, January 31, 2003, <http://www.govexec.com/dailyfed/0103/013103h1.htm>
180. **Hassan, Md. R, Hossain, M.M., Bailey, J., Ramamohanarao, K.**, *Improving k-Nearest Neighbour classification with distance functions based on Receiver Operating Characteristics*, Proc. 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I, Lecture Notes In Artificial Intelligence, 5211, 489-504, Springer, 2008.

181. **Hassanien, A., Ali, J.,** *Image classification and retrieval algorithm based on rough set theory*, S. Afr. Comput. J., 30, 9-16, 2003.
182. **Hassanien, A.,** *Rough set approach for attribute reduction and rule generation: A case of patients with suspected breast cancer*, Journal of the American Society for Information Science and Technology, 55(11), 954-962, 2004.
183. **Hassanien, A., Abraham, A., Peters, J., Schaefer, G., Henry, C.,** *Rough sets and near sets in medical imaging: a review*, IEEE Transactions on Information Technology in Biomedicine, 13(6), 955-968, 2009.
184. **Hastie T., Tibshirani R., Friedman J.,** *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Verlag, 2001.
185. **Hastie T., Tibshirani R.,** *Generalized additive models (with discussion)*, Statistical Science 1, 297-318, 1986.
186. **Hastie T., Tibshirani R.,** *Generalized additive models*, Chapman & Hall, 1990.
187. **Haupt, R., Haupt, S.E.,** *Practical genetic algorithms*, Wiley, 2004.
188. **Haykin S.,** *Neural Networks. A comprehensive foundation* (2nd Ed.), Prentice Hall, 1999.
189. **Hertz J., Krogh A., Palmer R.G.,** *Introduction to the theory of neural computation*, Redwood City, CA: Addison Wesley, 1991.
190. **Heyer, L.J., Kruglyak, S., Yoosoph, S.,** *Exploring expression data: identification and analysis of coexpressed genes*, Genome Res., 9(11), 1106-1115, 1999.
191. **Hill, T., Lewicki, P.,** *Statistics, methods and applications. A comprehensive reference for science, industry, and data mining*, StatSoft, 2006.
192. **Hincapie, J.G., Kirsch, R.F.,** *Feasibility of EMG-based neural network controller for an upper extremity neuroprosthesis*, IEEE Trans. Neural. Syst. Rehabil. Eng., 17(1), 80-90, Feb 2009.
193. **Hinton, G., Sejnowski, T. J. (Editors),** *Unsupervised Learning: Foundations of Neural Computation*, MIT Press, 1999.

194. **Hoffmann A.**, *Paradigms of artificial intelligence*, Springer Verlag, 1998.
195. **Holland J.H.**, *Adaptation in Natural and Artificial Systems*, MIT Press, 1992 (First edition: University of Michigan Press, Ann Arbor, 1975).
196. **Holland, J.H.**, *Adaptation*, Progress in Theoretical Biology (R. Rosen, F.M. Snell Eds.), 4, Plenum, 1976.
197. **Holte R.C.**, *Very Simple Classification Rules Perform Well on Most Commonly Used Datasets*, Machine Learning, 11, pp 63-90, 1993.
198. **Hopfield, J. J.**, *Neural networks and physical systems with emergent collective computational abilities*, Proc. National Academy of Sciences of the USA, 79(8), 2554-2558, 1982.
199. **Hoppner,F., Klawonn, F., Kruse, R., T. Runkler, T.**, *Fuzzy Cluster Analysis*, Wiley, Chichester, 1999.
200. **Hoskinson A.**, *Creating the ultimate research assistant*, Computer (IEEE Computer Society), Vo. 38, Nr. 11, pp. 97-99, 2005.
201. **Hotelling H.**, *Relations between two sets of variates*, Biometrika, 28, pp. 321-377, 1936.
202. **Hua, S., Sun, Z.**, *Support Vector Machine Approach for Protein Subcellular Localization Prediction*, Bioinformatics, 17(8), 721-728, 2001.
203. **Huang, C., Dorsey, E., Boose, M.A.**, *Life Insurer Financial Distress Prediction: A Neural Network Model*, Journal of Insurance Regulation, Winter, 13(2), 131-167, 1994.
204. **Huber P.J.**, *Robust Statistics*, John Wiley, 1981.
205. **Huet, S., Bouvier, A., Poursat, M.-A., Jolivet, E.**, *Statistical tools for nonlinear regression*, 2nd Edition (Springer Series in statistics), Springer, 2004.
206. **Hunt E.B., Marin J., Stone P.T.**, *Experiments in Induction*. Academic Press, New York, 1966.

207. **Hwang, J-N., Choi, J., Oh, S. Marks, R.**, *Query-based learning applied to partially trained multilayer perceptron*, IEEE Transactions on Neural Networks, 2(1), 131-136, 1991.
208. **Ince, E.A., Ali, S.A.**, *Rule based segmentation and subject identification using fiducial features and subspace projection methods*, Journal of Computers, 2(4), 68-75, 2007.
209. **Jackson, J.E.**, *A User's Guide to Principal Components*, Wiley series in probability and mathematical statistics, Wiley-IEEE, 2003.
210. **Jain A.K, Dubes R.C.**, *Algorithms for Clustering Data*, New Jersey, Prentice Hall, 1988.
211. **Jambu M.**, *Exploratory and multivariate data analysis*, Academic Press, 1991.
212. **Jensen, F.**, *Bayesian networks and decision graphs*, Statistics for Engineering and Information Science, Springer, 2001.
213. **Jiang, Z., Yamauchi, K., Yoshioka, K., Aoki, K., Kuroyanagi, S., Iwata, A., Yang, J.. Wang, K.**, *Support Vector Machine-Based Feature Selection for Classification of Liver Fibrosis Grade in Chronic Hepatitis C*, Journal of Medical Systems 20, 389–394, 2006.
214. **Jin., H., Lu, Y.**, *A non-inferiority test of areas under two parametric ROC curves*, Contemporary Clinical Trials, (30)4, 375-379, 2009.
215. **Joachims, T.**, *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*, Springer, 2002.
216. **Jolliffe, I.T.**, *Principal component analysis* (2nd Ed.), Springer series in statistics, Springer, 2002.
217. **Kahn B., Strong D., Wang R.**, *Information Quality Benchmarks: Product and Service Performance*, Communications of the ACM, April 2002. pp. 184-192, 2002.
218. **Kalamatianos, D., Liatsis, P., Wellstead, P.E.**, (2006) *Near-infrared spectroscopic measurements of blood analytes using multi-layer perceptron neural networks*, Conf. Proc. IEEE Eng. Med. Biol. Soc., 3541-3544, 2006.

219. **Kaufman L., Rousseeuw P. J.**, *Finding groups in data: an introduction to cluster analysis*, John Wiley & Sons, 2005.
220. **Khan, A.U., Bandopadhyaya, T.K., Sharma, S.**, *Classification of Stocks Using Self Organizing Map*, International Journal of Soft Computing Applications, 4, 19-24, 2009.
221. **Kim Jae-On, Mueller C.**, *Factor Analysis. Statistical Methods and Practical Issues*, Series 07-014, Sage Publications, Beverly Hills/London, 1978.
222. **Kim Jae-On, Mueller C.**, *Introduction to factor analysis*, Series 07-013, Sage Publications, Beverly Hills/London, 1978.
223. **Kitajima, M., Hirai, T., Katsuragawa, S., Okuda, T., Fukuoka, H., Sasao, A., Akter, M., Awai, K., Nakayama, Y., Ikeda, R., Yamashita, Y., Yano, S., Kuratsu, J., Doi, K.**, *Differentiation of common large sellar-suprasellar masses effect of artificial neural network on radiologists' diagnosis performance*, Acad. Radiol., 16(3), 313-320, Mar 2009.
224. **Kohonen T.**, *An introduction to neural computing*, Neural Networks, Vol. 1, pp. 3-16, 1988.
225. **Kohonen T.**, *Self-Organizing Maps* (3rd extended Ed.), Springer Series in Information Sciences, Vol. 30, Springer, 2001.
226. **Komorowski J., Polkovski, L., Skowron A.**, *Rough Sets: A tutorial*, Lecture Notes for ESSLLI'99: 11th European Summer School in Language, Logic and Information, Utrecht, Holland, 1999.
227. **Komorowski J., Øhrn A., Skowron A.**, *The ROSETTA Rough Set Software System*, Handbook of Data Mining and Knowledge Discovery (W. Klösgen and J. Zytkow -Eds.), ch. D.2.3, Oxford University Press, 2002.
228. **Koski, T., Noble, J.**, *Bayesian Networks: An Introduction*, Wiley Series in Probability and Statistics, 2009.
229. **Koza, J.**, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.
230. **Kowalczyk, W., Z. Piasta, Z.**, *Rough sets inspired approach to knowledge discovery in business databases*, In: The Second Pacific Asian

Conference on Knowledge Discovery Data Mining, (PAKDD'98), Melbourne, Australia, 1998.

231. **Kumar, A., Agrawal, D.P. Joshi, S.D.**, *Multiscale rough set data analysis with application to stock performance modeling*, Intelligent Data Analysis, 8(2), 197-209, 2004.
232. **Kuo, S.J., Y.H. Hsiao, Y.H., Y.L. Huang, Y.L., Chen, D.R.**, *Classification of benign and malignant breast tumors using neural networks and three-dimensional power Doppler ultrasound*, Ultrasound Obstet. Gynecol., 32 (1), 97-102, 2008.
233. **Lai, K.C., Chiang, H.C., Chen, W.C., Tsai, F.J., L. B. Jeng, L.B.**, *Artificial neural network-based study can predict gastric cancer staging*, Hepatogastroenterology, 55(86-87), 1859-1863, Sep-Oct 2008.
234. **Langdon, W., Poli, R.**, *Foundations of genetic programming*, Springer, 2002.
235. **Larose D.**, *Discovering Knowledge in Data: An Introduction to Data Mining*, John Wiley, 2006.
236. **Lasko, T.A., Bhagwat, J.G., Zou, K.H., Ohno-Machado, L.**, *The use of receiver operating characteristic curves in biomedical informatics*, Journal of Biomedical Informatics, 38(5), 404–415, 2005.
237. **Larose, D.**, *Data mining methods and models*, Wiley-Interscience, 2006.
238. **Lebowitz, M.**, *Experiments with incremental concept formation*, Machine Learning, 2, 103–138, 1987.
239. **Levine M.**, *Canonical analysis and factor comparison*, Sage Publications, Beverly Hills/London, 1977.
240. **Liang K.H., Krus D.J., Webb J.M.**, *K-fold crossvalidation in canonical analysis*, Multivariate Behavioral Research, 30, pp. 539-545, 1995.
241. **Lin, T.Y., Cercone, N.**, (Eds.) *Rough sets and data mining. Analysis of imprecise data*, Kluwer Academic Publishers, 1997.
242. **Lin, T.Y., Tremba, J.**: Attribute Transformations on Numerical Databases. Lecture Notes in Computer Science, Vol. 1805, Springer, 2000.

243. **Lin, T.Y.**: Attribute transformations for data mining. I: Theoretical explorations. *International journal of intelligent systems*. **17** (2), 213-222, 2002.
244. **Lin, T.Y., Yiyu Y. Yao, Y.Y., Zadeh, L.**, (Eds.) *Data Mining, Rough Sets and Granular Computing*, Physica-Verlag Heidelberg, 2002.
245. **Lindeman, R.H., Merenda, P.F., Gold, R.** (1980). *Introduction to bivariate and multivariate analysis*, New York: Scott, Foresman & Co., 1980.
246. **Lippmann, R.P.**, *An introduction to computing with neural nets*, IEEE ASSP Magazine, Vol. 4, pp. 4-22, 1987.
247. **Liu, H., Hussain, F., Tan, C.L., Dash, M.**: Discretization: An Enabling Technique. *Data Mining and Knowledge Discovery*. **6** (4), 393-423 (2002)
248. **Liu, H., Hiroshi, M.**: Feature Selection for Knowledge Discovery and Data Mining. *Kluwer Academic Publishers* (1998a)
249. **Liu, H., Hiroshi, M. (Eds.)**: Feature Extraction, Construction and Selection: A Data Mining Perspective. *Springer International Series in Engineering and Computer Science* (1998b)
250. **Loève, M.**, *Probability Theory I* (4th Edition), Springer-Verlag, 1977.
251. **Luo, F-L., Unbehauen, R.**, *Applied Neural Networks for Signal Processing*, Cambridge University Press, 1999.
252. **Maglogiannis, I., Sarimveis, H., Kiranoudis, C.T., Chatziiannou, A.A., Oikonomou, N., Aidinis, V.**, *Radial basis function neural networks classification for the recognition of idiopathic pulmonary fibrosis in microscopic images*, *IEEE Trans. Inf. Technol. Biomed.*, **12**(1), 42-54, 2008.
253. **Maqsood, I., Abraham, A.**, *Weather analysis using ensemble of connectionist learning paradigms*, *Applied Soft Computing*, **7**(3), 995-1004, 2007.
254. **Mandic, D., Chambers, J.**, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*, Wiley, 2001.

255. **Maung, Z.M., Mikami, Y.**, *A rule-based syllable segmentation of Myanmar Text*, Proc. IJCNLP-08 Workshop on NLP for Less Privileged Languages, 51–58, 2008.
256. **McLachlan, G.J., Peel, D.**, *Finite Mixture Models*, New York, John Wiley & Sons, 2000.
257. **Martinez W.L., Martinez A.L.**, *Exploratory Data Analysis with MATLAB*, CRC Press, 2004.
258. **Martinez, L.C., da Hora, D.N., Palotti, J.R., Meira, W., Pappa, G.**, *From an artificial neural network to a stock market day-trading system: a case study on the BM&F BOVESPA*, Proc. 2009 international joint conference on Neural Networks, 3251-3258, 2009.
259. **McCulloch W.S., Pitts W.A.**, *A logical calculus of the ideas immanent in nervous activity*, Bulletin of Mathematics and Biophysics, 5, pp. 115-133, 1943.
260. **McLachlan, G., Krishnan, T.**, *The EM Algorithm and extensions*, 2nd Ed., Wiley, 2008.
261. **McLaren, C.E., W.P. Chen, W.P., Nie, K., M.Y. Su, M.Y.**, *Prediction of malignant breast lesions from MRI features: a comparison of artificial neural network and logistic regression techniques*, Acad. Radiol., 16(7), 842-851, 2009.
262. **Meinel, L.A., Stolpen, A.H., Berbaum, K.S., Fajardo, L.L., Reinhardt, J.M.**, *Breast MRI lesion classification: improved performance of human readers with a backpropagation neural network computer-aided diagnosis (CAD) system*. J. Magn. Reson. Imaging, 25(1) : 89-95, 2007.
263. **Menzies T., Hu Y.**, *Data Mining For Very Busy People*, IEEE Computer, Volume 36, Issue 11, pp. 22-29 2003.
264. **Metz, C.E.**, *Basic principles of ROC analysis*, Sem. Nuc. Med, 8, 283-298, 1978.
265. **Michalewicz Z., Fogel D.B.**, *How to solve it: Modern Heuristics*, Springer-Verlag, 2004.
266. **Michalewicz Z.**, *Genetic Algorithms + Data Structures = Evolution Programs* (3rd Ed.), Springer-Verlag, 1996.

267. **Michalski, R.S.**, *Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts*, *International Journal of Policy Analysis and Information Systems*, 4, 219–244, 1980.
268. **Michalski, R.S., Stepp, R.E.**, *Learning from observation: Conceptual clustering*, *Machine Learning: An Artificial Intelligence Approach* (Michalski, R. S., Carbonell, J. G.; Mitchell, T. M. Eds.), Palo Alto, CA: Tioga, 331–363, 1983.
269. **Michel, A.N., Liu, D.**, *Qualitative analysis and synthesis of recurrent neural networks*, Pure and Applied Mathematics, CRC Press, 2002.
270. **Mihoc Gh., Urseanu V.**, *Sondaje și estimații statistice*, Ed. Tehnică, 1977.
271. **Miller, B.L., Goldberg, D.E.**, *Genetic Algorithms, Tournament Selection and the Effects of Noise*, *Complex Systems* 9, 193-212, 1996.
272. **Minsky M.L., Papert S.A.**, *Perceptrons*, Cambridge, MA: MIT Press, 1969 (Expanded Edition, 1987).
273. **Mirzaaghazadeh, A., Motameni, H., Karshenas, M., Nematzadeh, H.**, *Learning flexible neural networks for pattern recognition*, World Academy of Science, Engineering and Technology 33, 88-91, 2007.
274. **Mitchell, T.M.**, *Machine learning*, McGraw Hill, 1997.
275. **Mitchell, M.**, *An introduction to genetic algorithms (complex adaptive systems)*, MIT Press, 1998.
276. **Moore, D.S., McCabe, G.P.**, *Introduction to the Practice of Statistics* (3rd ed.), New York: W. H. Freeman, 1999.
277. **Montana, D., Davis, L.** *Training feedforward networks using genetic algorithms*, Proc. International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1989.
278. **Mrozek, A., L. Plonka, L.**, *Rough sets in image analysis*, *Foundations of Computing Decision Sciences* 18(3-4), 259 – 273, 1993.
279. **Mulaik, S.A.**, *Foundations of Factor Analysis* (2nd Ed.), Chapman & Hall/CRC Statistics in the Social and Behavioral Scie), 2009.

280. **Muller, K.R., Smola, A.J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V.**, *Predicting Time Series with Support Vector Machines*, Lecture Notes in Computer Science, 1327, Proceedings of the International Conference on Artificial Neural Networks, 999–1004, 1997.
281. **Mustapha, N., Jalali, M., Jalali, M.**, *Expectation Maximization clustering algorithm for user modeling in Web usage mining systems*, European Journal of Scientific Research, 32(4), 467-476, 2009.
282. **Neubaer, A.**, *Adaptive non-uniform mutation for genetic algorithms*, Computational Intelligence Theory and Applications, Lecture Notes in Computer Science, Springer, 1997
283. **Nguyen, H.S., Nguyen, S.H.**, *Pattern extraction from data*, Fundamenta Informaticae, 34, 129-144, 1998.
284. **Nguyen, H.D., Yoshihara, I., Yasunaga, M.**, *Modified edge recombination operators of genetic algorithms for the traveling salesman problem*, Industrial Electronics Society, IECON 2000, 26th Annual Conference of the IEEE, 4, 2815-2820, 2000.
285. **Nigsch, F., Bender, A., van Buuren, B., Tissen, J., Nigsch, E., Mitchell, J.B.O.**, *Melting Point Prediction Employing k-nearest Neighbor Algorithms and Genetic Parameter Optimization*, Journal of Chemical Information and Modeling, 46(6), 2412–2422, 2006.
286. **Norton, P.G., Dunn, E.V.**, Snoring as a risk factor for disease: an epidemiological survey, *Br. Med. J.*, 291, 630-632, 1985.
287. **Obe, O.O, Shangodoyin, D.K.**, *Artificial neural network based model for forecasting sugar cane production*, Journal of Computer Science 6(4), 439-445, 2010.
288. **Ogulata, S.N., Sahin, C., Erol, R.**, *Neural network-based computer-aided diagnosis in classification of primary generalized epilepsy by EEG signals*, *J. Med. Syst.*, 33(2), 107-112, Apr. 2009.
289. **Øhrn A., Komorowski J.**, *ROSETTA: A Rough Set Toolkit for Analysis of Data*, Proceedings Third International Joint Conference on Information Sciences, Fifth International Workshop on Rough Sets and Soft Computing (RSSC'97), Durham, NC, USA, March 1-5, Vol. 3, pp. 403-407, 1997.

290. **Øhrn A.**, *ROSETTA Technical Reference Manual*, Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2000.
291. OLAP Council (2002). *OLAP and OLAP Server Definitions*, available from <http://www.olapcouncil.org/research/glossary.htm>, -April 2003.
292. **Oliver, J., Wallace, C.S.**, *Inferring decision graphs*, Proc. of the 1992 Aust. Joint Conf. on Artificial Intelligence, pp.361-367, 1992.
293. **Oliver, I.M., Smith, D.J., Holland, J.**, *A study of permutation crossover operators on the travelling salesman problem*, Proc. 2nd International Conference on Genetic Algorithms and Their Applications (Grefenstette J.J. -Ed.), Lawrence Erlbaum, Hillsdale, New Jersey, 224-230, 1987
294. **O'Neill, S., Leahy, F., Pasterkamp, H., Tal, A.**, *The effects of chronic hyperinflation, nutritional status, and posture on respiratory muscle strength in cystic fibrosis*, Am. Rev. Respir. Dis., 128, 1051-1054, 1983.
295. **Osaki S.**, *Applied stochastic system modeling*, Springer, 1992.
296. **Osuna, E., Freund, R., Girosi, F.**, *Training Support Vector Machines: An Application to Face Detection*, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 130–136, 1997.
297. **Park, L-J.**, *Learning of Neural Networks for Fraud Detection Based on a Partial Area Under Curve*, Lecture Notes in Computer Science, Springer, 3497, 922-927, 2005.
298. **Parzen, E.**, *On estimation of a probability density function and mode*, Ann. Math. Stat., 33, 1065-1076, 1962.
299. **Pawlak Z.**, *Rough Sets*, International Journal of Computer and Information Sciences, Vol. 11, pp. 341-356, 1982.
300. **Pawlak Z.**, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishing, Dordrecht, 1991.
301. **Pearson, K.**, Contributions to the mathematical theory of evolution, II: Skew variation in homogeneous material. *Philosophical Transactions of the Royal Society of London*, 186, 343-414, 1895.
302. **Pearson, K.**, Das Fehlgesetz und seine Verallgemeinerungen durch Fechner und Pearson. A Rejoinder. *Biometrika*, 4, 169-212, 1905.

303. **Pearson, K.**, On the generalized probable error in multiple normal correlation, *Biometrika*, 6, 59-68, 1908.
304. **Pedrycz, W., Skowron, A.**, *Fuzzy and rough sets, Handbook of data mining and knowledge discovery*, Oxford University Press, Inc., New York, 2002.
305. **Pendaharkar, P.C.**, *A comparison of gradient ascent, gradient descent and genetic-algorithm-based artificial neural networks for the binary classification problem*, Expert Systems, 24(2), 65-86, 2007.
306. **Pepe, M.S.**, *The statistical evaluation of medical tests for classification and prediction*, Oxford University Press, 2004.
307. **Piatetsky-Shapiro, G.**, *Discovery, analysis, and presentation of strong rules* (in *Knowledge Discovery in Databases*, Piatetsky-Shapiro, G. and Frawley, W., Eds.,), AAAI/MIT Press, Cambridge, MA, 1991.
308. **Polkowski, L., Skowron, A.**, (Eds.) *Rough Sets in Knowledge Discovery 1: Methodology and Applications*, Physica-Verlag, Heidelberg, 1998.
309. **Polkowski, L., Skowron, A.**, (Eds.) *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*, Physica-Verlag, Heidelberg, 1998.
310. **Polkowski, L., Tsumoto, S., Lin, T.Y.**, (Eds.) *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems* (Studies in Fuzziness and Soft Computing), Physica-Verlag, Heidelberg, 2000.
311. **Polkowski, L.**, *Rough Sets*, Physica-Verlag, Heidelberg, 2002.
312. **Pontil, M., Verri, A.**, *Support Vector Machines for 3D Object Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(6), 637-646, 1998.
313. **Price R., Shanks G.**, *A Semiotic Information Quality Framework*, Proc. IFIP International Conference on Decision Support Systems (DSS2004): Decision Support in an Uncertain and Complex World, Prato, 2004.
314. **Provost, F., Fawcett, T.**, *Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions*,

- Proc. 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97). AAAI Press, Menlo Park, CA, 43–48, 1997.
315. **Provost, F., Fawcett, T., Kohavi, R.**, *The case against accuracy estimation for comparing induction algorithms*, Proc. ICML-98 (Shavlik, J. -Ed.), Morgan Kaufmann, San Francisco, CA, pp. 445–453, 1998.
316. **Quinlan, J.R.**, *Induction of decision trees*, Machine Learning, 1, pp. 81–106, 1986.
317. **Quinlan, J.R., Rivest, R.L.**, *Inferring decision trees using the minimum description length principle*, Information and Computation, 80(3), 227–248, 1989.
318. **Quinlan, J. R.**, C4.5: *Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.
319. **Radzikowska, A.M., Kerre, E.E.**, *A comparative study of fuzzy rough sets*, Fuzzy Sets and Systems, 126, 137–156, 2002.
320. **Raj, V.S.**, *Better performance of neural networks using functional graph for weather forecasting*, Proc. 12th WSEAS international conference on Computers, 826-831, 2008.
321. **Rechenberg, I.**, *Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution* (PhD thesis, 1971); reprinted by Frommann-Holzboog, 1973.
322. **Revett K., Gorunescu F., Gorunescu M., El-Darzi E., Ene M.**, *A Breast Cancer Diagnosis System: A Combined Approach Using Rough Sets and Probabilistic Neural Networks*, Proceedings Eurocon2005 –IEEE International Conference on “Computer as a tool”, Belgrade, Serbia, November 21-24, pp. 1124-1127, 1-4244-0049-X/05/\$20.00 ©2005 IEEE, 2005.
323. **Revett K., Gorunescu F., Gorunescu M., Ene M.**, *Mining A Primary Biliary Cirrhosis Dataset Using Rough Sets and a Probabilistic Neural Network*, Proceedings of the IEEE Intelligent Systems 2006 –IS06, Westminster, London, pp. 284-289, IEEE 2006.
324. **Revett, K., Gorunescu, F., Salem, A.B.**, *Feature Selection in Parkinson’s Disease: A Rough Sets Approach*, Proc. International Multiconference on Computer Science and Information Technology - IMCSIT 2009, Mrągowo, Poland, 4, 425 – 428, 2009.

325. **Ripley, B.D.**, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.
326. **Rissanen, J.**, *Modeling by the shortest data description*, Automatica 14, 465-471, 1978.
327. **Rissanen, J.**, *Information and Complexity in Statistical Modeling*, Springer, 2007.
328. ***, **ROC graphs: Notes and practical considerations for researchers**, HP Labs, Tech. Rep. HPL-2003-4, 2003
329. **Roemer, V.M., Walden, R.**, *Sensitivity, specificity, receiver-operating characteristic (ROC) curves and likelihood ratios for electronic foetal heart rate monitoring using new evaluation techniques*, Z. Geburtshilfe Neonatol., 214(3), 108-118, 2010.
330. **Rokach, L., Maimon, O.**, *Data mining with decision trees*, Series in Machine Perception and Artificial Intelligence, 69, World Scientific, 2007
331. **Romesburg H.C.**, *Cluster Analysis for Researchers* (2nd Ed.), Krieger Pub. Co, 2004.
332. **Rosenblatt F.**, *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*, Washington DC: Spartan, 1962.
333. **Rosenblatt F.**, *The Perceptron: A probabilistic model for information storage and organization in the brain*, Psychoanalytic Review, 65, pp. 386-408, 1958.
334. **Ryan, T.**, *Modern Regression Methods, Solutions Manual* (2nd edition), Wiley Series in Probability and Statistics, Wiley, 2009.
335. **Saad, E.W., Choi, J.J., Vian, J.L., Wunsch, D.C.**, *Query-based learning for aerospace applications*, IEEE Trans. Neural Netw., 14(6), 1437-48, 2003.
336. **Sandage A.**, EDWIN HUBBLE 1889-1953, JRASC Vol. 83, No.6, 1989.
337. **Dos Santos, W.P., R.E. de Souza, R.E., dos Santos Filho, P.B.**, *Evaluation of Alzheimer's disease by analysis of MR images using multilayer perceptrons and Kohonen SOM classifiers as an alternative to the ADC maps*, Conf. Proc. IEEE Eng. Med. Biol. Soc., 2118-2121, 2007.

338. Săftoiu A., Ciurea T., Gorunescu F., Rogoveanu I., Gorunescu M., Georgescu C., *Stochastic modeling of the tumor volume assessment and growth patterns in hepatocellular carcinoma*, Journal of Oncology/Bulletin du Cancer, vol. 91, n° 6, pp. 162-166, 2004.
339. Săftoiu A., Gorunescu F., Rogoveanu I., Ciurea T., Ciurea P., *A Dynamic Forecasting Model to Predict the Long-Term Response During Interferon Therapy in Chronic Hepatitis C*, World Congresses of Gastroenterology, Austria, Vienna, September 6-11, Digestion (Karger), 59, (3), pp. 349, 1998.
340. Săftoiu, A., Vilmann, P., Gorunescu, F., Gheonea, D.I., Gorunescu, M., Ciurea, T., Popescu, G.L., Iordache, A., Hassan, H., Iordache, S., *Neural network analysis of dynamic sequences of EUS elastography used for the differential diagnosis of chronic pancreatitis and pancreatic cancer*, Gastrointestinal Endoscopy, Mosby- Elsevier, 68(6), 1086-1094, 2008.
341. Sanzogni, L., Kerr, D., *Milk production estimates using feed forward artificial neural networks*, Computer and Electronics in Agriculture, 32(1), July 2001.
342. Schaffer J.D., *Some Effects of Selection Procedures on Hyperplane Sampling by Genetic Algorithms*, Genetic Algorithms and Simulated Annealing (Davis L. -Ed.), Pitman, 1987.
343. Schaffer, J.D., Whitley, D., Eshelman, L., *Combination of Genetic Algorithms and Neural Networks: The state of the art*, Combination of Genetic Algorithms and Neural Networks, IEEE Computer Society, 1992.
344. Schlkopf, B., Smola, A., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (Adaptive Computation and Machine Learning), MIT Press, 2001.
345. Schwefel, H.P., *Numerische Optimierung von Computer-Modellen* (PhD thesis, 1974);rReprinted by Birkhäuser, 1977.
346. Seber, G.A.F., Wild, C.J., *Nonlinear Regression* (Wiley Series in Probability and Statistics), Wiley, 1989.
347. Senthilkumaran, N., Rajesh, R., *A Study on Rough Set Theory for Medical Image Segmentation*, International Journal of Recent Trends in Engineering, 2(2), 236-238, 2009.

348. **Shafer, J., Agrawal, R., Mehta, M.**, *SPRINT: A scalable parallel classifier for Data Mining*, Proc. 22nd VLDB Conference Mumbai (Bombay), India, pp. 544-555, 1996.
349. **Shakhnarovich G., Darrell T., Indyk P.**, (Eds.) *Nearest-Neighbor Methods in Learning and Vision*, MIT Press, 2005.
350. **Shank, D.B., McClendon, R.W., Paz, J., Hoogenboom, G.**, *Ensemble artificial neural networks for prediction of dew point temperature*, Applied Artificial Intelligence, 22(6), 523-542, 2008.
- 351. Shannon A., Riecan B., Langova-Orozova D., Kerre E., Sotirova E., Gorunescu F., Petrounias I., Trelewicz J., Atanassov K., Krawczak M., Chountas P., Melo-Pinto, Georgiev P., Kim T., Tasheva V.,** *Generalized net model with intuitionistic fuzzy estimations of the process of obtaining of scientific titles and degrees*, Notes on Intuitionistic Fuzzy Sets (Special issue –Proc. 9th International Conference on Intuitionistic Fuzzy Sets), Vol. 11, pp. 95-114, 2005.

Shvaytser, H., Peleg, S.: *Fuzzy and probability vectors as elements of vector space*. Information Sciences 36, 231--247 (1985)

Shu, C., Burn, D.H.: *Artificial neural network ensembles and their application in pooled flood frequency analysis*. Water Resources Research (2004) 40(9), W09301, doi: 10.1029/2003WR002816, 1-10 (Abstract)

Shuai, J-J., Li, H-L.: *Using rough set and worst practice DEA in business failure prediction*. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Lecture Notes in Computer Science 3642, Springer (2005)

Skowron, A.: *Boolean reasoning for decision rules generation*. Lecture Notes in Computer Science 689, 295--305 (1993)

Slezak, D.: *Approximate Entropy Reducts*. Fundamenta Informaticae 53(3,4), 365--390 (2002)

Slowinski, R., Zopounidis, C.: *Applications of the rough set approach to evaluation of bankruptcy risk*. International J. Intelligent Systems in Accounting, Finance and Management 4/1, 27--41 (1995)

Smith, T.F.: *Occam's razor*. Nature 285(5767), pp. 620 (1980)

Smith, J.E., Fogarty, T.C.: *Operator and parameter adaptation in genetic algorithms.* Soft Computing 1(2), 81--87 (1997)

Song, T., Jamshidi, MM., Lee, R.R., Hung, M.: *A modified probabilistic neural network for partial volume segmentation in brain MR image.* IEEE Trans. Neural Networks 18(5), 1424--1432 (2007)

Song, J.H., Venkatesh, S.S., Conant, E.A., Arger, P.H., Sehgal, C.M.: *Comparative analysis of logistic regression and artificial neural network for computer-aided diagnosis of breast masses.* Acad. Radiol. 12(4), 487--495 (2005)

Sorjamaa, A., Hao, J., Lendasse, A.: *Mutual Information and k-Nearest Neighbors Approximator for Time Series Prediction.* Lecture Notes in Computer Science 3697, 553--558 (2005)

Spackman, K.A.: *Signal detection theory: Valuable tools for evaluating inductive learning.* Proc. 6-th International Workshop on Machine Learning, 160--163, Morgan Kaufman, San Mateo, CA (1989)

Spearman, C.: *"General intelligence" objectively determined and measured.* American Journal of Psychology 15, 201--293 (1904)

Spearman, C.: *The nature of intelligence and the principles of cognition.* London: Macmillan (1923)

Spearman, C.: *The abilities of man.* London: Macmillan (1927)

Specht, D.F.: *Probabilistic neural networks for classification mapping or associative memory.* Proc. IEEE International Conference on Neural Networks 1, 525--532 (1988)

Specht, D.F.: *Probabilistic neural networks.* Neural Networks 3, 110--118 (1990)

Specht, D.F.: *Probabilistic neural networks and the polynomial adaline as complementary techniques for classification.* IEEE Trans. on Neural Networks 1(1), 111--121 (1990)

Specht, D.F.: *A Generalized Regression Neural Network.* IEEE Transactions on Neural Networks 2(6), 568--576 (1991)

Srisawat, A., Phienthrakul, T., Kijksirikul, B.: *SV-kNNC: An Algorithm for Improving the Efficiency of k-Nearest Neighbor.* In: Yang, Q., Webb, G. (eds.)

PRICAI 2006: Trends in Artificial Intelligence, Lecture Notes in Computer Science 4099, 975--979 (2006)

Stefansky, W.: *Rejecting Outliers in Factorial Designs*. Technometrics 14, 469--479 (1972)

Steinwart, I., Christmann, A.: *Support Vector Machines*. Information Science and Statistics, Springer (2008)

Stevens, J.: *Applied multivariate statistics for the social sciences*. Hillsdale, NJ: Erlbaum (1986)

Stoean, R., Preuss, M., Stoean, C., El-Darzi E., Dumitrescu D.: *An Evolutionary Approximation for the Coefficients of Decision Functions within a Support Vector Machine Learning Strategy*. Foundations on Computational Intelligence 1, 83--114 (2009)a.

Stoean, C., Preuss, M., Gorunescu, R., Dumitrescu, D.: *Elitist generational genetic chromodynamics - a new radii-based evolutionary algorithm for multimodal optimization*. 2005 IEEE Congress on Evolutionary Computation (CEC 2005), 1839--1846, IEEE Computer Society Press (2005)

Stoean, C., Preuss, M., Dumitrescu, D., Stoean, R.: *Cooperative Evolution of Rules for Classification*. IEEE Post-proceedings Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2006), 317--322, IEEE Press (2006)

Stoean, R., Preuss, M., Stoean, C., El-Darzi E., Dumitrescu D.: *Support Vector Machine Learning with an Evolutionary Engine*. Journal of the Operational Research Society, Special Issue: Data Mining and Operational Research: Techniques and Applications 60(8), 1116--1122 (2009)b.

Stoean, R., Stoean, C., Lupsor, M., Stefanescu, H., Badea, R.: *Evolutionary-Driven Support Vector Machines for Determining the Degree of Liver Fibrosis in Chronic Hepatitis C*. Artificial Intelligence in Medicine, Elsevier (2010) (*in press*).

Sunay, A.S., Cunedioglu, U., Ylmaz, B.: *Feasibility of probabilistic neural networks, Kohonen self-organizing maps and fuzzy clustering for source localization of ventricular focal arrhythmias from intravenous catheter measurements*. Expert System 26(1), 70--81 (2009)

Sutton, R.S., Barto, A.G.: *Reinforcement learning*. MIT Press (1999)

Suykens, J.A.K., Vandewalle, J.P.L., De Moor, B.L.R.: *Artificial neural networks for modelling and control of non-linear systems*. Kluwer Academic Publishers (1996)

Swets, J.A.: *The relative operating characteristic in psychology. A technique for isolating effects of response bias finds wide use in the study of perception and cognition*. Science 182(4116), 990--1000 (1973)

Swets, J.A., Dawes, R.M., Monahan, J.: *Better decisions through science*. Scientific American 283, 82--87 (2000)

Swiniarski, R.: *Rough sets Bayesian methods applied to cancer detection*. In: Polkowski, L., Skowron, A. (eds.) Proc. First International Conference on Rough Sets and Soft Computing (RSCTC'98), Warszawa, Poland, LNAI 1424, 609--616, Springer-Verlag (1998)

Syswerda, G.: *Uniform crossover in genetic algorithms*. In: Scaffer, J.D. (ed.) Proc. 3rd International Conference on Genetic Algorithms, 2--9, Morgan Kaufmann (1989)

Tan Pang-Ning, Steinbach, M., Kumar, V.: *Introduction to Data Mining*. Addison-Wesley (2005)

352. Tasseva V., Atanassov K., Matveev M., El-Darzi E., Chountas P., Gorunescu F., *Modelling the flow of patient through intensive care unit using generalized net*, Proceedings The First East European Conference on Health Care Modelling and Computation, -HCMC2005, Craiova, Romania, August 31st-September 2nd (F. Gorunescu, M. Gorunescu, E. El-Darzi- Eds.), Medical University Press, pp. 290-299, 2005.

Tay, F., Shen, L.: *Economic and financial prediction using rough sets model*. European Journal of Operational Research 141(3), 641--659 (2002)

Thomson, S.K.: *Sampling* (2nd ed.). Wiley Series in Probability and Statistics (2002)

Thuraisingham, B.: *Data Mining: Technologies, Techniques, Tools, and Trends*. CRC Press (1999)

Thurstone, L.L.: *Multiple factor analysis*. Psychological Review 38, 406--427 (1931)

353. **Tiță I., Gorunescu F.**, *Considerations on species genesis for Vicia L. using the chromosomal complement study and statistical analysis*, Phytologia Balcanica-Sofia, Vol. 7(3), pp. 341-348, 2001.

Titterington, D.M., Smith, A.F.M., Makov, U.E.: *Statistical Analysis of Finite Mixture Distributions*. John Wiley and Sons, New York (1985)

Tong, S., Chang, E.: *Support Vector Machine Active Learning for Image Retrieval*. Proc. of the 9th ACM international conference on Multimedia 9, 107-118 (2001)

Toussaint, G.T.: *Geometric proximity graphs for improving nearest neighbor methods in instance-based learning and data mining*. International Journal of Computational Geometry and Applications 15(2), 101--150 (2005)

Tukey, J.: *Exploratory Data Analysis*. Addison-Wesley (1977)

Unnikrishnan, N., Mahajan., A., Chu, T.: *Intelligent system modeling of a three-dimensional ultrasonic positioning system using neural networks and genetic algorithms*. Proc. Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 217(5), 367--377 (2003)

Vajda, P., Eiben, A.E., Hordijk, W.: *Parameter Control Methods for Selection Operators in Genetic Algorithms*. Lecture Notes in Computer Science 5199, Springer (2008)

Vapnik, V.N., Chervonenkis, Ya: *On the uniform convergence of relative frequencies of events to their probabilities*. Theoretical Probability and its applications 17, 264--280 (1971)

Vapnik, V.N.: *Estimation of dependencies based on empirical data* (translated by S. Kotz). Springer (1982)

Vapnik, V.N., Golowich S.E., Smola A.J.: *Support Vector Method for Function Approximation, Regression Estimation and Signal Processing*. NIPS 1996, Denver, USA, 281--287 (1996)

Vapnik, V.N.: *Statistical learning theory*. Wiley (1998)

Vapnik, V.N.: *The nature of statistical learning theory*. Springer Verlag (1995)

Ward, J.H.: *Hierarchical grouping to optimize an objective function*. Journal of the American Statistical Association 58(301), 236--244 (1963)

Velleman, P., Hoaglin, D.: *The ABC's of EDA: Applications, Basics, and Computing of Exploratory Data Analysis*. Duxbury (1981)

Wand, Y., Wang, R.: *Anchoring Data Quality Dimensions in Ontological Foundations*. Communications of the ACM, 86--95 (1996)

Wang, J., Neskovic, P., Cooper, L.N. *Improving nearest neighbor rule with a simple adaptive distance measure*,. Pattern Recognition Letters 28, 207--213 (2007)

Westphal, C., Blaxton, T.: *Data Mining Solutions*. John Wiley (1998)

Whitley, D.: *Permutations*. In: Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.) *Evolutionary computation 1: basic algorithms and operators*, Institute of Physics Publishing, Bristol, Ch. 33.3, pp. 274-284 (2000)

Whitley, D., Kauth, J.: *Genitor: A different genetic algorithm*. Proc. of the Rocky Mountain Conference on Artificial Intelligence, 118--130 (1988)

Whitley, D.: *The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best*. Proc. of the Third International Conference on Genetic Algorithms, 116--123. Morgan Kaufman (1989)

Whitley, D., Hanson, T.: *Optimizing neural networks using faster more accurate genetic search*. Proc. 3rd International Conference on Genetic Algorithms, 391--396. Morgan Kaufmann (1989)

Whitley, D., Starkweather, T., Fuquay, D'A.: *Scheduling problems and traveling salesman: The genetic edge recombination operator*. Proc. 3rd International Conference on Genetic algorithms, 133--140. Morgan Kaufmann (1989)

Whitley, D., Starkweather, T., Shaner, D.: *Traveling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination*. Handbook of Genetic Algorithms, Van Nostrand (1990)

Whitley, D.: *Genetic Algorithms in Engineering and Computer Science*. In: Periaux, J., Winter, G. (eds.) John Wiley and Sons Ltd. (1995)

Widrow, B., Hoff, M.E.: *Adaptive switching circuits*. IRE WESCON Convention Record, New York (reprinted in Anderson and Rosenfeld, 1988), 4, 96--104 (1960)

354. **Widrow B., Lehr M.A.**, *30 years of adaptive neural networks: perceptron, perceptron, madaline, and backpropagation*. Proceedings of the IEEE, 78 (9), pp. 1415-1442, 1990.
355. **Wikinews**, “U.S. Army intelligence had detected 9/11 terrorists year before”,
http://en.wikinews.org/w/index.php?title=U.S._Army_intelligence_had_detected_9/11_terrorists_year_before%2C_says_officer&oldid=130741.
356. **Wishart, G.C., Warwick ,J., Pitsinis, V., Duffy, S., Britton, P.D.**, *Measuring performance in clinical breast examination*, Br. J. Surg., 97(8), 1246-1252, 2010.
357. **Witten I.H., Eibe, F.**, *Data Mining: Practical Machine Learning Tools and Techniques* (2nd Ed.), Morgan Kaufmann, 2005.
358. **Wood S.**, *Generalized additive models. An introduction with R*, Chapman & Hall/CRC, 2006.
359. **Yao, J., Herbert, J.**, *Financial time-series analysis with rough sets*, Applied Soft Computing, 9(3), 1000-1007, 2009.
360. **Ye N., Farley T.**, *A scientific approach to cyberattack detection*, Computer (IEEE Computer Society), Vo. 38, Nr. 11, pp. 55-61, 2005.
361. **Ye N., Chen Q.**, *Computer intrusion detection through EWMA for autocorrelated and uncorrelated data*, IEEE Trans. Reliability, vol. 52, no. 1, pp. 73-82, 2003.
362. **Ye N.**, *Mining computer and network security data*, The Handbook of Data Mining, (Ye N.- Ed.), Lawrence Erlbaum Assoc., pp. 617-636, 2003.
363. **Yeh, W., Huang, S.W., Li, P.C.**, *Liver Fibrosis Grade Classification with B-mode Ultrasound*, Ultrasound in Medicine and Biology 29(9) 1229–1235, 2003.
364. **Zadeh, L.**, *Fuzzy sets*, Information and Control, 8, 338-353, 1965.
365. **Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.**, *New algorithms for fast discovery of association rules*, Technical Report TR651, 1997.
366. **Zaknich A.**, *Neural networks for intelligent signal processing*, World Scientific, Series in Innovative Intelligence, Vol. 4, 2003.

367. **Zhang, C., Zhang, S.**, Association rule mining: models and algorithms (Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence), Springer, 2002.
368. **Zhong N.**, *Using Rough Sets with Heuristics for Feature Selection*, Journal of Intelligent Information Systems, 16, pp. 199-214, Kluwer Academic Publishers, 2001.
369. **Ziarko W.**, *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer Verlag, 1994.
370. **Zur, R.M., Pesce, L.L., Jiang, Y.**, *The effect of two priors on Bayesian estimation of "Proper" binormal ROC curves from common and degenerate datasets*, Acad Radiol., 17(8), 969-79, 2010.
371. **Zweig, M.H., Campbell, G.**, *Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine*, Clinical chemistry, 39(8), 561–577, 1993.