

# GENERAREA AUTOMATĂ A REGULILOR PRIN ÎNVĂȚARE INDUCTIVĂ

*Studiu comparativ*

dr. ing. Mihaela M. Oprea

Universitatea Petrol-Gaze Ploiești Catedra de Informatică

**Rezumat:** Succesul unui sistem expert depinde în bună măsură de existența unei baze de cunoștințe complete, coerente și neredundante. Generarea regulilor din baza de cunoștințe se poate realiza cu ajutorul algoritmilor de învățare inductivă. Lucrarea prezintă un studiu comparativ al aplicării unor astfel de algoritmi, de învățare inductivă, ID3, C4.5, ILA, DCL și RITIO.

**Cuvinte cheie:** învățare inductivă, arbori de decizie, generarea regulilor, sistem expert.

## 1. Introducere

Principala problemă care poate apare în cadrul etapei de achiziție a cunoștințelor unui sistem expert constă în transferul cât mai fidel al cunoștințelor experților umani în baza de cunoștințe. Astfel, succesul sistemului expert va depinde în bună măsură de existența unei baze de cunoștințe complete, coerente și neredundante. Întrucât regulile de producție reprezintă o modalitate directă, expresivă și flexibilă de a exprima cunoașterea în multe domenii, vom prezenta un studiu comparativ al principalilor algoritmi de învățare inductivă, care permit generarea automată a regulilor pornind de la un set de exemple de instruire.

În secțiunea 2, prezentăm succint algoritmii de învățare inductivă, care au fost analizați în cadrul acestui studiu, ID3, C4.5, ILA, DCL, RITIO. Avantajele generării directe a setului de reguli în raport cu inducerea arborelui de decizie sunt discutate în secțiunea 3. Rezultatele studiului comparativ sunt detaliate în secțiunea 4. În ultima secțiune, prezentăm principalele concluzii.

## 2. Algoritmi de învățare inductivă

### 2.1. Problema învățării inductive

Una dintre cele mai eficiente metode de achiziție automată a cunoașterii o reprezintă învățarea inductivă. Învățarea bazată pe arbori de decizie, programarea logică inductivă, învățarea cu ajutorul rețelelor neuronale, algoritmi genetici sunt exemple de metode de învățare inductivă, care operează pornind de la o mulțime de exemple de instruire, ce reprezintă istoricul deciziilor anterioare.

Formularea problemei învățării inductive, așa cum este prezentată în [1], este următoarea:

Se dă o mulțime completă de exemple de instruire:  $M = \{ \langle x_1, f(x_1) \rangle, \dots, \langle x_n, f(x_n) \rangle \}$ , unde  $f(x_i)$  este valoarea funcției  $f$  corespunzătoare instanței  $x_i$  și un spațiu al ipotezelor,  $H$ , din care sunt selectate ipoteze. Scopul problemei este de a genera o ipoteză  $h$  care să fie consistentă cu exemplele din mulțimea  $M$ .

Algoritmii de învățare inductivă se pot clasifica în două clase mari: clasa algoritmilor care se bazează pe arbori de decizie (ID3, ASSISTANT86, GID3, C4.5) și clasa algoritmilor care nu utilizează arbori de decizie (ILA, DCL, RITIO). În secțiunile următoare, prezentăm succint algoritmi din ambele clase.

### 2.2. Algoritm ID3

Algoritmii tip ID3 [1] împart mulțimea  $M$  de instruire în submulțimi omogene fără a face referire la clasa submulțimii. ID3 caută atributul cel mai relevant în raport cu toate exemplele, chiar dacă unele valori ale aceluia atribut pot fi irelevante. Algoritmii utilizează entropia ca o metodă de restrângere a spațiului de căutare a ipotezelor. ID3 este un algoritm *greedy* care dezvoltă arborele de sus în jos, la fiecare nod selectând atributul care clasifică cel mai bine exemplele locale de instruire. Cel mai bun atribut este cel care are *câștigul maxim de informație*. Dacă definim *entropia* ca o măsură a impurității dintr-o colecție de exemple de instruire, putem defini o măsură a eficacității unui atribut  $A$  în clasificarea datelor de instruire. Măsura numită *câștigul de informație* reprezintă reducerea prevăzută a entropiei, cauzată de partiționarea exemplilor în conformitate cu acest atribut.

$$Castig(M, A) = Entropie(M) - \sum_{v \in Valori(A)} \frac{|Mv|}{|M|} Entropie(Mv)$$

unde  $Valori(A)$  este mulțimea tuturor valorilor posibile ale atributului  $A$ , iar  $M_v$  este submulțimea lui  $M$  pentru care atributul  $A$  are valoarea  $v$ .

ID3 realizează o căutare a unei ipoteze care se potrivește cu exemplele de instruire. Spațiul ipotezelor,  $H$ , este, în acest caz, mulțimea tuturor arborilor de decizie. Algoritmul execută o căutare tip hill-climbing care pornește cu un arbore vid, considerând în mod progresiv ipoteze din ce în ce mai elaborate în căutarea unui arbore de decizie, care să clasifice corect toate datele de instruire.

### 2.3. Algoritmul C4.5

Algoritmul C4.5 [2] este o extensie a algoritmului ID3, care permite procesarea datelor incerte cu prețul creșterii ratei erorii de clasificare. Inițial, algoritmul produce un arbore de decizie pe care, mai apoi, îl simplifică eliminând toate condițiile care nu sunt necesare. Astfel, C4.5 generează un arbore simplificat, din care sunt extrase regulile. Rata erorii de clasificare poate fi zero sau diferită de zero, în timp ce, în cazul algoritmului ID3, aceasta este zero. C4.5 generează un număr redus de reguli de producție, însă, aceste reguli pot eșua în încercarea lor de a clasifica corect toate exemplele din mulțimea de instruire.

### 2.4. Algoritmul ILA

Algoritmul ILA (Inductive Learning Algorithm) [3] generează setul de reguli în mod iterativ. La fiecare iterație caută o regulă care acoperă un număr cât mai mare de exemple de instruire ale unei singure clase. După ce a găsit o regulă, ILA elimină din mulțimea de instruire acele exemple acoperite de obicei și adaugă regula la sfârșitul setului de reguli. Exemplele eliminate sunt, de fapt, marcate pentru a fi ignorate la următoarea iterație. Astfel, se obține un set ordonat de reguli. ILA se concentrează pe depistarea valorilor relevante ale atributelor prin eliminarea condițiilor care nu sunt necesare.

### 2.5. Algoritmul DCL

Algoritmul DCL (Disjunctive Concept Learning) [4] este o versiune îmbunătățită a algoritmului ILA. Practic, DCL generează reguli de producție cu operatori AND/OR pornind de la mulțimea de exemple de instruire. Algoritmul permite atât învățarea conceptelor disjunctive, cât și a celor conjunctive. Principalele avantaje ale acestui algoritm sunt generarea unui număr mic de reguli, în raport cu majoritatea algoritmilor de învățare inductivă și clasificarea unui număr mult mai mare de exemple necunoscute, în raport cu algoritmi de învățare a conceptelor conjunctive. Algoritmul DCL extrage același număr de reguli ca și ILA, dar adaugă disjuncții la partea stângă a regulilor generate. Generarea regulilor se realizează lucrând separat cu mulțimea exemplilor pozitive de instruire și, respectiv, cu mulțimea exemplilor negative de instruire.

### 2.6. Algoritmul RITIO

Caracteristica principală a algoritmului RITIO (Rule Induction without decision Tree constructIO) [5] o reprezintă eliminarea atributelor în ordinea descrescătoare a irelevanței. Similar cu algoritmi care construiesc arbori de decizie (tip ID3) utilizează măsura entropiei ca o modalitate de a restrânge spațiul de căutare a ipotezei, dar, spre deosebire de aceștia, limbajul ipotezelor folosește structura de regulă și, astfel, RITIO generează regulile fără să construiască arborii de decizie. Rezultatele experimentale au confirmat faptul că, și în cazul bazelor de date industriale (cu zgomot, adică cu valori greșite ale atributelor, cu date contradictorii, incomplete sau redundate), algoritmul RITIO generează reguli care au o precizie predictivă mare.

## 3. Generarea regulilor versus inducerea arborelui de decizie

Una dintre formele importante de reprezentare, utilizată în cadrul învățării inductive, este arborele de decizie, într-un proces cunoscut drept inducerea top-down a arborilor de decizie. O altă reprezentare importantă



este dată de structura de regulă **IF-THEN**. Structurile tip arbore de decizie, așa cum sunt induse de algoritmi din clasa ID3, produc fragmentarea bazei de date ori de câte ori este testat un atribut de aritate mare. Aceasta diminuează înțelegerea ipotezelor conceptului indus. Mai mult, aceste structuri au tendința de a repeta subarbori, atunci când exprimă concepte disjunctive de forma  $(atribut_1 = A) \text{ OR } (atribut_2 = B)$ , această problemă fiind cunoscută sub numele de problema replicării. Ca o consecință a acestor două probleme, arborii de decizie tind să se extindă foarte mult în majoritatea cazurilor. Cu toate că există anumite soluții care au fost propuse (de exemplu, [6] în cazul problemei replicării, [7] și [8] în cazul simplificării arborilor de decizie), generarea directă a regulilor are, pe lângă avantajul de a evita crearea arborilor de decizie, și avantajul de a produce o descriere mai compactă a conceptelor în cazul majorității problemelor reale.

În continuare, considerăm problema mult simplificată, a analizei proiectelor de investiții și vom studia doi algoritmi reprezentativi ai celor două clase de algoritmi de învățare inductivă, ID3 și ILA. Atributele utilizate sunt: *Niv\_invest* (nivelul investiției), *Risc\_global* (riscul global al proiectului), *Timp\_rev* (timpul de recuperare a investiției), *Rentabilitatea* (rentabilitatea fondurilor proprii), iar mulțimile de valori posibile corespunzătoare: {scăzut, mediu, ridicat}, {scăzut, mediu, ridicat}, {f\_scurt, scurt, mediu, lung} și respectiv, {mică, medie, importantă}. Anumite exemple analizate și prezentate în cadrul acestei lucrări includ doar unele atribute și, uneori, doar unele valori posibile ale acestora. Parametrul *Decizie* se referă la decizia luată în urma analizei, proiect acceptat (*Da*), respectiv, proiect respins (*Nu*).

În tabelul 1 este prezentată mulțimea  $M_1$  a exemplelor de instruire în cazul analizei proiectelor de investiții.

**Tabelul 1. Mulțimea  $M_1$  a exemplelor de instruire în cazul analizei proiectelor de investiții**

| Ex. | Risc global | Rentabilitatea | Timp rev. | Nivel invest | Decizie |
|-----|-------------|----------------|-----------|--------------|---------|
| 1.  | ridicat     | importantă     | lung      | ridicat      | Nu      |
| 2.  | ridicat     | importantă     | lung      | scăzut       | Nu      |
| 3.  | scăzut      | importantă     | lung      | ridicat      | Da      |
| 4.  | mediu       | medie          | lung      | ridicat      | Da      |
| 5.  | mediu       | mică           | scurt     | ridicat      | Da      |
| 6.  | mediu       | mică           | scurt     | scăzut       | Nu      |
| 7.  | scăzut      | mică           | scurt     | scăzut       | Da      |
| 8.  | ridicat     | medie          | lung      | ridicat      | Nu      |
| 9.  | ridicat     | mică           | scurt     | ridicat      | Da      |
| 10. | mediu       | medie          | scurt     | ridicat      | Da      |
| 11. | ridicat     | medie          | scurt     | scăzut       | Da      |
| 12. | scăzut      | medie          | lung      | scăzut       | Da      |
| 13. | scăzut      | importantă     | scurt     | ridicat      | Da      |
| 14. | mediu       | medie          | lung      | scăzut       | Da      |

După aplicarea algoritmului ILA asupra mulțimii  $M_1$  obținem următoarele reguli:

- R1: IF Risc\_global = scăzut THEN Decizie = Da.
- R2: IF Risc\_global = ridicat AND Timp\_rev = lung THEN Decizie = Nu.
- R3: IF Risc\_global = mediu AND Nivel\_invest = scăzut THEN Decizie = Nu.
- R4: IF Risc\_global = mediu AND Nivel\_invest = ridicat THEN Decizie = Da.
- R5: IF Risc\_global = ridicat AND Timp\_rev = scurt THEN Decizie = Nu.

Aceste reguli sunt identice cu cele obținute atunci când s-a aplicat un algoritm tip ID3. Se poate observa că regulile generate nu conțin condiții care nu sunt necesare.

În figura 1, este prezentat arborele de decizie indus din tabelul 1 cu ajutorul algoritmului ID3. Arborele de decizie simplificat este prezentat în figura 2.

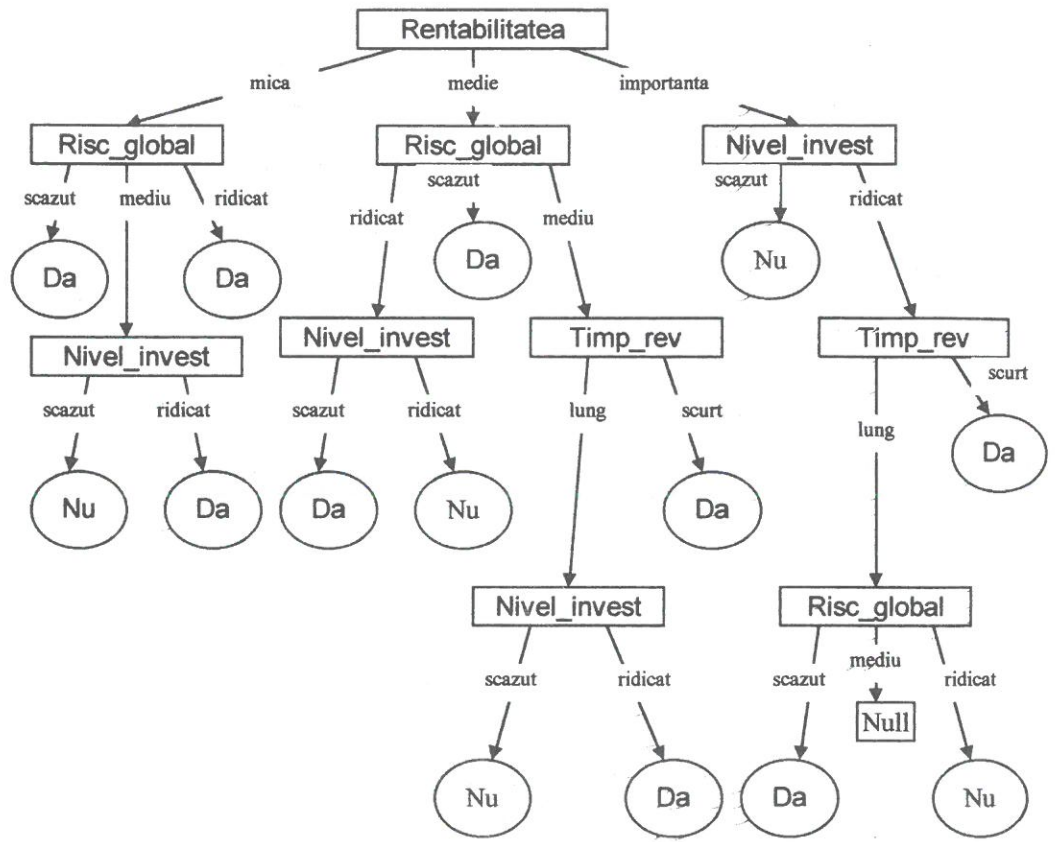


Figura 1. Arborele de decizie indus din tabelul 1 (ID3)

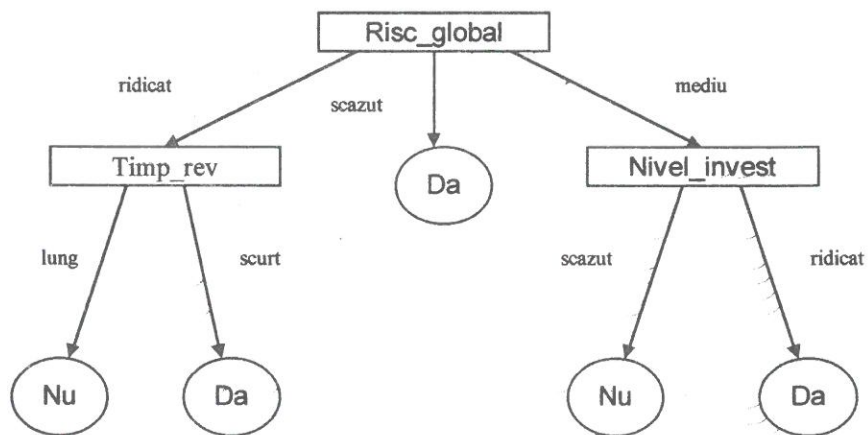


Figura 2. Arborele de decizie simplificat

În tabelul 2 este prezentat un alt set de exemple de instruire. Acest set realizează o clasificare a proiectelor de investiții care sunt analizate. Problema a fost simplificată luându-se în considerare doar trei parametri.



**Tabelul 2. Mulțimea  $M_2$  a exemplilor de instruire în cazul problemei de clasificare a proiectelor de investiții**

| Ex. | Nivel investiții | Timp rev. | Risc global | Clasa |
|-----|------------------|-----------|-------------|-------|
| 1.  | ridicat          | scurt     | mediu       | IP1   |
| 2.  | ridicat          | mediu     | scăzut      | IP2   |
| 3.  | mediu            | mediu     | scăzut      | IP2   |
| 4.  | scăzut           | mediu     | scăzut      | IP2   |
| 5.  | ridicat          | mediu     | mediu       | IP1   |
| 6.  | ridicat          | lung      | ridicat     | IP0   |
| 7.  | ridicat          | lung      | mediu       | IP3   |
| 8.  | scăzut           | lung      | mediu       | IP3   |
| 9.  | scăzut           | lung      | ridicat     | IP0   |
| 10. | mediu            | lung      | scăzut      | IP2   |
| 11. | scăzut           | scurt     | mediu       | IP1   |

Regulile generate după aplicarea celor doi algoritmi, ID3 și ILA sunt prezentate în tabelul 3. Ambii algoritmi au generat același număr de reguli (5), dar regula 3 generată de ILA este mai simplă decât cea generată de ID3. Aceasta se datorează faptului că ILA elimină condițiile care nu sunt necesare, de exemplu:  $Risc\_global=mediu$ .

**Tabelul 3. Regulile generate de ID3 și ILA**

| Algoritm   | Nr. regulă | Regula   |
|------------|------------|--|
| ID3<br>ILA | 1.         | IF Risc_global = scăzut THEN Clasa = IP2.<br>IF Risc_global = scăzut THEN Clasa = IP2.   |
| ID3<br>ILA | 2.         | IF Risc_global = ridicat THEN Clasa = IP0.<br>IF Risc_global = ridicat THEN Clasa = IP0.                                       |
| ID3<br>ILA | 3.         | IF Timp_rev = short AND Risc_global = mediu THEN Clasa = IP1.<br>IF Timp_rev = scurt THEN Clasa = IP1.                         |
| ID3<br>ILA | 4.         | IF Timp_rev = mediu AND Risc_global = mediu THEN Clasa = IP1.<br>IF Timp_rev = mediu AND Risc_global = mediu THEN Clasa = IP1. |
| ID3<br>ILA | 5.         | IF Timp_rev = lung AND Risc_global = mediu THEN Clasa = IP3.<br>IF Timp_rev = lung AND Risc_global = mediu THEN Clasa = IP3.   |

Scurta analiză prezentată confirmă faptul că, deși algoritmi de învățare inductivă, care se bazează pe arbori de decizie, sunt eficienți și pot lucra cu un număr mare de exemple de instruire, totuși, ei nu extrag cele mai generale reguli.

#### 4. Rezultate experimentale

Analiza preliminară (prezentată în [9]), care a inclus algoritmi ID3, C4.5, ILA și DCL, a relevat comportarea mult mai bună a algoritmilor ILA și DCL în raport cu ID3 și C4.5. În lucrarea de față, prezentăm rezultatele unei analize extinse, care include trei algoritmi din clasa algoritmilor care nu utilizează arbori de decizie, ILA, DCL și RITIO. Experimentele au fost realizate pe diferite mulțimi de exemple de instruire. O parte din experimente au avut drept scop extragerea regulilor pentru baza de cunoștințe a unui sistem bazat pe cunoștințe, destinat îmbunătățirii protecției mediului, care este prezentat în [10]. În această secțiune, ne vom concentra asupra experimentelor realizate pe mulțimi de exemple simple, de instruire în cazul problemei anterior menționate, de analiză a proiectelor de investiții. Trebuie specificat că analiza proiectelor de investiții este mult simplificată, numărul de parametri analizați fiind redus.

În tabelul 4 este prezentată mulțimea  $M_3$  a exemplelor de instruire, iar în tabelul 5 sunt prezentate regulile generate de către algoritmi ILA, DCL și RITIO.

**Tabelul 4. Mulțimea  $M_3$  a exemplelor de instruire în cazul analizei proiectelor de investiții**

| Nr. ex. | Nivel investiții | Risc global | Timp rev. | Decizie |
|---------|------------------|-------------|-----------|---------|
| 1.      | mediu            | scăzut      | scurt     | Da      |
| 2.      | scăzut           | ridicat     | lung      | Nu      |
| 3.      | scăzut           | ridicat     | f. scurt  | Da      |
| 4.      | ridicat          | ridicat     | lung      | Nu      |
| 5.      | ridicat          | mediu       | mediu     | Da      |
| 6.      | ridicat          | ridicat     | mediu     | Nu      |
| 7.      | ridicat          | mediu       | f. scurt  | Da      |

**Tabelul 5. Regulile extrase de ILA, DCL și RITIO**

| Algoritm                    | Nr. regulă | Regula   |
|-----------------------------|------------|--|
| ILA<br>DCL<br>RTIO          | 1.         | IF Risc_global=mediu THEN Decizie=Da.<br>IF Risc_global=mediu THEN Decizie=Da.<br>IF Risc_global=mediu THEN Decizie=Da.  |
| ILA<br>DCL<br><br>RITIO     | 2.         | IF Nivel_invest=mediu THEN Decizie=Da.<br>IF (Nivel_invest=mediu) OR (Risc_global=scăzut)<br>OR (Timp_rev=scurt)<br>THEN Decizie=Da.<br>IF Nivel_invest=mediu THEN Decizie=Da.   |
| ILA<br>DCL<br>RITIO         | 3.         | IF Timp_rev=f_scurt THEN Decizie=Da.<br>IF Timp_rev=f_scurt THEN Decizie=Da.<br>IF Timp_rev=f_scurt THEN Decizie=Da.   |
| ILA<br>DCL<br>RITIO         | 4.         | IF Timp_rev=lung THEN Decizie=Nu.<br>IF Timp_rev=lung THEN Decizie=Nu.<br>IF Timp_rev=lung THEN Decizie=Nu.  |
| ILA<br><br>DCL<br><br>RITIO | 5.         | IF Nivel_invest=ridicat AND Risc_global=ridicat<br>THEN Decizie=Nu.<br>IF (Nivel_invest=ridicat AND Risc_global=ridicat)<br>OR (Timp_rev=mediu AND Risc_global=ridicat)<br>THEN Decizie=Nu.<br>IF Nivel_invest=ridicat AND Risc_global=ridicat<br>THEN Decizie=Nu. |

Analizând rezultatele prezentate în tabelul 5, observăm că toți algoritmi generează același număr de reguli, iar algoritmi ILA și RITIO generează, în cazul mulțimii  $M_3$  de instruire, același set de reguli. O altă observație se referă la faptul că algoritmi ILA și RITIO elimină condițiile care nu sunt necesare, generând astfel reguli mai simple decât algoritmul DCL. De exemplu, sunt eliminate condițiile '*Timp\_rev=mediu*' și '*Timp\_rev=scurt*'.

Un algoritm bun de generare a regulilor trebuie să permită și clasificarea exemplelor necunoscute. În tabelul 6, sunt prezentate ratele erorii de clasificare pentru trei mulțimi de test cu exemple necunoscute, obținute prin aplicarea algoritmilor ID3, C4.5, ILA, DCL și RITIO. Așa cum era de așteptat, cel mai bun comportament l-au avut algoritmi DCL și RITIO. DCL care poate genera reguli disjunctive a permis cea mai bună clasificare a exemplelor necunoscute, în cazul mulțimilor de test  $S_1$  și  $S_2$ , iar RITIO a avut un comportament foarte bun, în cazul tuturor celor trei mulțimi de test, explicația fiind datorată caracteristicii acestui algoritm, de a elimina atributele în ordinea descrescătoare a irelevanței lor. Algoritmi C4.5 și ILA au generat rate apropiate ale erorii de clasificare, ambii eliminând condițiile care nu sunt necesare și generând, astfel, reguli mai simple decât ceilalți algoritmi.



Tabelul 6

|                  | ID3   | C4.5  | ILA   | DCL          | RITIO        |
|------------------|-------|-------|-------|--------------|--------------|
| Mulțimea de test |       |       |       |              |              |
| S <sub>1</sub>   | 48.7% | 34.1% | 32.7% | <b>28.5%</b> | 29.8%        |
| S <sub>2</sub>   | 37.3% | 24.7% | 26.5% | <b>18.5%</b> | 21.7%        |
| S <sub>3</sub>   | 7.12% | 4.31% | 4.26% | 1.71%        | <b>1.43%</b> |

În tabelul 7, sunt prezentate rezultatele obținute pentru două mulțimi ale exemplilor de instruire, în cazul problemei de clasificare a proiectelor de investiții și, respectiv, al problemei de protecție a mediului [10]. ILA și C4.5 au generat mai puține reguli cu mai puține condiții decât cele generate de către ceilalți algoritmi. Astfel, pot clasifica mai multe exemple necunoscute din setul de test. În cazul mulțimii de instruire pentru protecția mediului, DCL a obținut cel mai mic număr de reguli, iar C4.5 cel mai mic număr mediu de condiții, în schimbul posibilei eșuări a regulilor în cazul clasificării exemplilor necunoscute (rata erorii de clasificare este 15.7%). Algoritmul RITIO a avut cea mai bună comportare atât din punct de vedere al numărului de reguli generate, cât și al numărului mediu de condiții.

Trebuie remarcat faptul că toate rezultatele experimentale precum și concluziile studiului sunt dependente și de natura datelor din mulțimile de test.

Tabelul 7

| Mulțimea de instruire  | Algoritm | Nr. de reguli generate | Nr. mediu de condiții |
|------------------------|----------|------------------------|-----------------------|
| proiecte de investiții | ID3      | 5                      | 1.6                   |
|                        | C4.5     | 5 (0%) rata erorii     | 1.4                   |
|                        | ILA      | 5                      | 1.4                   |
|                        | DCL      | 5                      | 2.26                  |
|                        | RITIO    | 5                      | 1.4                   |
| protecția mediului     | ID3      | 14                     | 2.17                  |
|                        | C4.5     | 10 (15.7%) rata erorii | <b>1.8</b>            |
|                        | ILA      | 12                     | 2.04                  |
|                        | DCL      | 9                      | 2.21                  |
|                        | RITIO    | 11                     | 2.04                  |

## 5. Concluzii

Generarea automată a regulilor prin învățare inductivă necesită identificarea cu precizie a unui set complet de atribute relevante. Algoritmul RITIO, care a avut, în medie, cea mai bună comportare în cadrul întregului studiu comparativ prezentat, realizează această identificare prin eliminarea atributelor în ordinea descrescătoare a irelevanței. Algoritmul DCL a avut și el o comportare bună, permițând generarea celui mai mic număr de reguli, în raport cu ceilalți algoritmi. Performanțele relativ identice ale algoritmilor C4.5 și ILA sunt umbrite de rata mare a erorii de clasificare, prezentă în anumite situații în cazul algoritmului C4.5.

Concluzia finală a experimentelor efectuate până în prezent relevă faptul că, în general, rata erorii de clasificare este mai mică, în cazul algoritmilor de învățare inductivă, care nu se bazează pe arbori de decizie decât în cazul celor care utilizează arborii de decizie pentru generarea regulilor, iar algoritmi cu cea mai bună comportare sunt RITIO și DCL.

## Bibliografie

1. MITCHELL, T.: Machine Learning, McGraw-Hill, 1997.
2. QUINLAN, J. R.: C4.5: Programs for Machine Learning. Philadelphia, PA: Morgan Kaufmann, 1993.
3. TOLUN, M., S. ABU-SOUD: ILA: An Inductive Learning Algorithm for Rule Extraction. În: Expert Systems with Applications, 14, 1998, pp. 361-370.
4. TOLUN, M., S. ABU-SOUD: A Disjunctive Concept Learning Algorithm for Rule Generation. În: Proc. of the 17th IASTED International Conference Applied Informatics, ACTA Press, 1999, pp. 665-667.

5. **WU, X., D. URPANI, J. SYKES:** Rule Induction Without Decision Tree Construction. În: Proc. of the 12th European Conf. on Artificial Intelligence (ECAI), John Wiley&Sons, 1996, pp. 463-467.
6. **YANG, D., G. BLIX, L.A. RENDELL:** The Replication Problem: A Constructive Induction Approach. În: Proc. of European Working Session on Learning, Springer-Verlag, 1991, pp. 44-61.
7. **BRESLOW, L., D. AHA:** Symplifying Decision Trees: A survey. În: The Knowledge Engineering Review, Vol.12:1, 1997, pp. 1-40.
8. **BOHANEK M., I. BRATKO:** Trading Accuracy for Simplicity in Decision Trees. În: Machine Learning, No. 15, 1994, pp. 223-250.
9. **OPREA, M.:** Învățarea inductivă a regulilor. În: Revista Informatica Economică, nr. 1 (17), 2001, pp. 114-118.
10. **OPREA, M.:** Toward the Development of a Knowledge-based System for Environmental Protection. În: Proc. of the 8th International Conference Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU), Madrid, Vol. II, 2000, pp. 1003-1008.