

Învatarea inductiva a regulilor

Conf.dr.ing. Mihaela M. OPREA
Catedra de Informatica, Universitatea Ploiesti

Învatarea inductiva reprezinta o metoda eficienta de achizitie automata a cunoasterii într-un sistem expert. Lucrarea prezinta o analiza a rezultatelor preliminare obtinute prin aplicarea a patru algoritmi de învățare inductiva, ID3 si C4.5, algoritmi bazati pe arbori de decizie si respectiv, ILA si DCL, algoritmi care nu se bazeaza pe arbori de decizie. Algoritmii au fost utilizati pentru generarea automata a unui set de reguli pornind de la o multime de exemple de instruire.

Cuvinte cheie: învățare inductiva, arbori de decizie, extragerea regulilor, sistem expert.

Introducere

Procesul de achizitie a cunoasterii reprezinta o etapa extrem de importanta în dezvoltarea unui sistem expert. Aceasta etapa presupune realizarea unui transfer al cunoasterii de la expertii în domeniu catre baza de cunostinte a sistemului expert. Întrucât regulile de productie reprezinta o modalitate directa, expresiva si flexibila de exprimare a cunoasterii în multe domenii, vom prezenta o analiza a principalelor metode de învățare inductiva care permit generarea automata a regulilor pornind de la un set de exemple de instruire.

În sectiunea 2 este descrisa succint problema învățarii inductive, precum si unele metode de învățare inductiva din clasa algoritmilor bazati pe arbori de decizie (ID3, C4.5) si respectiv, a celor care nu utilizeaza arbori de decizie (ILA, DCL). Un studiu comparativ al celor patru algoritmi este detaliat în sectiunea 3. În ultima sectiune sunt prezentate principalele concluzii ale analizei.

Învatarea inductiva

Învatarea inductiva reprezinta una dintre cele mai eficiente metode utilizate pentru achizitia automata a cunoasterii. Învatarea bazata pe arbori de decizie, învățarea cu ajutorul rețelilor neuronale, programarea logica inductiva, algoritmi genetici (ex. [4]) sunt exemple de metode de învățare inductiva care

opereaza pornind de la o multime de exemple de instruire ce reprezinta, de fapt, istoricul unor decizii anterioare. Învatarea inductiva poate realiza învățarea conceptelor conjunctive si/sau disjunctive. Majoritatea algoritmilor de învățare inductiva permit învățarea conceptelor conjunctive, adica, generarea regulilor care contin în partea stânga numai conjunctii. Unul din algoritmii eficienti care permit învățarea atât a conceptelor conjunctive cât si a celor disjunctive este algoritmul DCL [7].

Problema învățarii inductive

Formularea problemei învățarii inductive așa cum este prezentata în [2] este următoarea: Se da o multime completa de exemple de instruire:

$D = \{ \langle x_1, f(x_1) \rangle, \dots, \langle x_n, f(x_n) \rangle \}$, unde $f(x_i)$ este valoarea functiei f corespunzătoare instanței x_i si un spatiu al ipotezelor, H , din care sunt selectate ipoteze. Scopul problemei este de a genera o ipoteza h care sa fie consistenta cu exemplele din multimea D .

Algoritmii de învățare inductiva se pot clasifica în doua clase: clasa algoritmilor care se bazeaza pe arbori de decizie si clasa algoritmilor care nu utilizeaza arbori de decizie. În sectiunile urmatoare vom face o scurta prezentare a unor algoritmi de învățare inductiva din ambele clase.

Învatarea inductiva bazata pe arbori de decizie

Metodele de învățare inductivă care se bazează pe arbori de decizie generează arbori de decizie pornind de la mulțimea exemplelor de instruire. Aceste metode sunt preferate întrucât sunt eficiente și pot lucra cu un număr mare de exemple de instruire [1]. Totuși, acest tip de metode nu produce întotdeauna cele mai generale reguli de producție.

Cei mai cunoscuți algoritmi care preiau la intrare o mulțime de exemple și produc un arbore de decizie care este consistent cu exemplele sunt: ID3, ID4, ASSISTANT86, GID3, C4.5.

Algoritmii tip ID3 [2] împart mulțimea 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 acelui atribut pot fi irelevante. Algoritmul utilizează entropia ca o metodă de restrângere a spațiului de căutare a ipotezelor. ID3 este un algoritm greedy care dezvoltă arbori 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 într-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 exemplelor în conformitate cu acest atribut.

$$\text{Castig}(D, A) = \text{Entropie}(D) - \sum_{v \in \text{Valori}(A)} \frac{|D_v|}{|D|} \text{Entropie}(D_v)$$

unde $\text{Valori}(A)$ este mulțimea tuturor valorilor posibile ale atributului A , iar D_v este submulțimea lui D 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 urcarea dealului (hill-climbing) care porneste 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 datele de instruire.

În figura 1 este prezentat un arbore de decizie care estimează prețul unei locuințe în funcție de o serie de factori cum ar fi, amplasarea locuinței, vârsta persoanei care cumpără locuința, condiția socială a aceluia, zona de plasare a locuinței în cazul în care amplasamentul este într-un oraș. Practic, arborele de decizie realizează o clasificare a locuințelor în funcție de prețul lor discretizat. Fiecare cale care porneste din rădăcina și

merge până la o frunză reprezintă o regulă care inferează apartenența la o clasă. De exemplu, din figura 1 putem deduce următoarea regulă:

IF amplasare=oras AND zona=buna AND condiție=excelentă THEN preț=ridicată.

Algoritmul C4.5 [5] 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 esua în încercarea lor de a clasifica corect toate exemplele din mulțimea de instruire.

Învatarea inductiva care nu se bazeaza pe arbori de decizie

Principalii algoritmi de învățare inductivă care nu se bazează pe arbori de decizie sunt: ILA, DCL, RITIO, AQ [2] etc. Acești algoritmi extrag regulile de producție direct din mulțimea exemplilor de instruire.

Algoritmul ILA (Inductive Learning Algorithm) [6] generează setul de reguli în mod iterativ. La fiecare iterație caută o regulă care acopere un număr cât mai mare de exemple de instruire ale unei singure clase. După ce a

gasit o regulă, ILA elimină din mulțimea de instruire acele exemple acoperite de regulă și adaugă regulă 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.

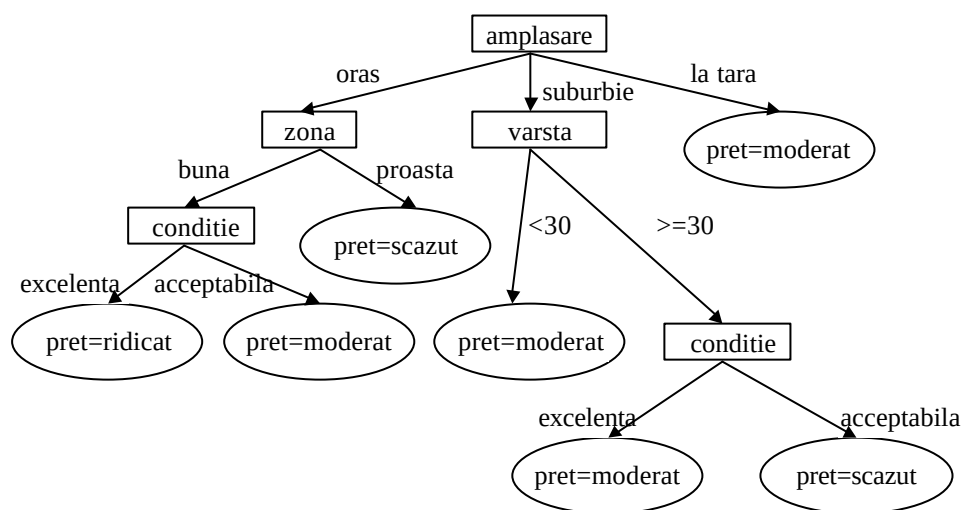


Fig. 1. Arbore de decizie

Algoritmul DCL (Disjunctive Concept Learning) [7] este o versiune îmbunătățită a algoritmului ILA. Practic, DCL generează reguli de producție cu operatori AND/OR pomind de la mulțimea de exemple de instruire. Algoritmul permite și învățarea conceptelor disjunctive nu numai 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.

Rezultate experimentale

Prezentăm rezultatele unei analize preliminare a celor patru algoritmi, ID3, C4.5, ILA și DCL. 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 [3]. În această secțiune ne vom concentra asupra experimentelor realizate pe mulțimi de exemple simple, de instruire în cazul rezolvării problemei de analiză a proiectelor de investiții. Trebuie specificat faptul că în aceste experimente analiza proiectelor de investiții este mult simplificată, numărul de parametri analizați fiind redus.

Tabelul 1Multimea M_1 a exemplelor de instruire în cazul analizei proiectelor de investitii

Nr.ex.	Nivel_invest	Risc_global	Timp_rev	Decizie
1.	mediu	scazut	scurt	Da
2.	scazut	ridicat	lung	Nu
3.	scazut	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

În tabelul 1 este prezentata multimea M_1 a exemplelor de instruire. În acest exemplu *Niv_invest* (nivelul investitiei), *Risc_global* (riscul global al proiectului), *Timp_rev* (timpul de recuperare a investitiei) sunt atribute cu urmatoarele multimi de valori posibile {scazut, mediu, ridicat}, {scazut, mediu, ridicat} si respectiv, {f_scurt, scurt, mediu, lung}. Setul de reguli generate de cei patru algoritmi pornind de la multimea M_1 de exemple este prezentat în tabelul 2. Parametrul *Decizie* se refera la decizia luata în urma analizei, proiect acceptat (*Da*), respectiv, proiect respins (*Nu*).

Analizând rezultatele prezentate în tabelul 2 observam ca toti algoritmi genereaza acelasi numar de reguli, iar algoritmi C4.5 si ILA genereaza acelasi set de reguli. Aceasta ultima observatie trebuie sa fie amendata cu o

corectie care se refera la rata erorii de clasificare. În tabelul 3 vom evidentia aceasta rata a erorii de clasificare. O alta observatie se refera la faptul ca algoritmi C4.5 si ILA elimina conditiile care nu sunt necesare, generând astfel reguli mai simple decât algoritmi ID3 si DCL. De exemplu, sunt eliminate conditiile '*Timp_rev=mediu*' si '*Timp_rev=scurt*'

Un algoritm bun de extragere a regulilor trebuie sa genereze regulile care sa permita si clasificarea exemplelor necunoscute. În tabelul 3 sunt prezentate ratele erorii de clasificare pentru doua seturi de test cu exemple necunoscute, obtinute prin aplicarea celor patru algoritmi. Asa cum era de asteptat, cel mai bun comportament l-a avut algoritmul DCL care include disjunctii ce permit o mai buna clasificare a exemplelor necunoscute.

Tabelul 2

Regulile extrase de ID3, C4.5, ILA si DCL

Algoritm	Nr.regula	Regula
ID3	1.	IF Risc_global=mediu AND Timp_rev=mediu THEN Decizie=Da.
C4.5		IF Risc_global=mediu THEN Decizie=Da.
ILA		IF Risc_global=mediu THEN Decizie=Da.
DCL		IF Risc_global=mediu THEN Decizie=Da.
ID3	2.	IF Timp_rev=scurt THEN Decizie=Da.
C4.5		IF Nivel_invest=mediu THEN Decizie=Da.
ILA		IF Nivel_invest=mediu THEN Decizie=Da.
DCL		IF (Nivel_invest=mediu) OR (Risc_global=scazut) OR (Timp_rev=scurt) THEN Decizie=Da.
ID3	3.	IF Timp_rev=f_scurt THEN Decizie=Da.
C4.5		IF Timp_rev=f_scurt THEN Decizie=Da.
ILA		IF Timp_rev=f_scurt THEN Decizie=Da.

DCL		IF Timp_rev=f_scurt THEN Decizie=Da.
ID3	4.	IF Timp_rev=lung THEN Decizie=Nu.
C4.5		IF Timp_rev=lung THEN Decizie=Nu.
ILA		IF Timp_rev=lung THEN Decizie=Nu.
DCL		IF Timp_rev=lung THEN Decizie=Nu.
IDA	5.	IF Risc_global=ridicat AND Timp_rev=mediu THEN Decizie=Nu.
C4.5		IF Nivel_invest=ridicat AND Risc_global=ridicat THEN Decizie=Nu.
ILA		IF Nivel_invest=ridicat AND Risc_global=ridicat THEN Decizie=Nu.
DCL		IF (Nivel_invest=ridicat AND Risc_global=ridicat) OR (Timp_rev=mediu AND Risc_global=ridicat) THEN Decizie=Nu.

Tabelul 3

	ID3	C4.5	ILA	DCL
Setul de test				
S ₁	61.8%	45.1%	32.7%	28.5%
S ₂	47.3%	28.9%	26.7%	18.5%

Concluzii

Analiza preliminară a celor patru algoritmi de învățare inductivă, ID3, C4.5, ILA și DCL releva faptul că, în general, comportamentul lor a fost bun și s-a atins un anumit grad de generalitate a regulilor extrase. Algoritmul DCL, care permite și generarea unor reguli cu condiții disjunctive, a avut cea mai bună comportare. Trebuie remarcat faptul că toate concluziile analizei sunt dependente de problema analizată și de natura datelor din seturile de test. De asemenea, performanța relativ identică a algoritmilor C4.5 și ILA (conform tabelului 2) este umbră de rată erorii de clasificare prezentată în tabelul 3. Concluzia finală a experimentelor efectuate până în prezent, este aceea că, în general, rata erorii de clasificare este mai mică în cazul algoritmilor care nu se bazează pe arbori de decizie, decât în cazul celor care utilizează arborii de decizie pentru generarea regulilor.

Bibliografie

- [1] L. Breslow, D. Aha, Simplifying decision trees: A survey, *The Knowledge Engineering Review*, Vol.12:1, 1997, 1-40.
- [2] T. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
- [3] M. Oprea, Toward the Development of a Knowledge-based System for Environmental Protection, *Proceedings of the 8th Internat. Conf. IPMU*, Vol.II, 2000, 1003-1008.
- [4] M. Oprea, Some Remarks about Rules Learning by Genetic Algorithms Approach, *Proceedings of the 4-th Internat. Symposium of Economic Informatics*, 1999, INFOREC Printing House, Bucharest, 722-727.
- [5] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Philadelphia, PA: Morgan Kaufmann, 1993.
- [6] M. Tolun, S. Abu-Soud, ILA: an inductive learning algorithm for rule extraction, *Expert Systems with Applications*, 14, 1998, 361-370.
- [7] M. Tolun, S. Abu-Soud, A Disjunctive Concept Learning Algorithm for Rule Generation, *Proceedings of the 17th IASTED International Conference Applied Informatics*, ACTA Press, 1999, 665-667.