Învățare Automată - Laboratorul 3
Arbori de Decizie

Tudor Berariu
tudor.berariu@gmail.com
Laboratorul AIMAS
Facultatea de Automatică și Calculatoare
Universitatea Politehnica București

8 martie 2016

1 Scopul laboratorului

Scopul laboratorului îl reprezintă învățarea conceptului de *arbore de decizie* și înțelegerea și implementarea algoritmului **ID3**.

2 Arbori de decizie

2.1 Problema de rezolvat

Problema de rezolvat în acest laborator este una de învățare supervizată: *fiind dat un set de date* $\mathbf{X}$ *ce conține exemple descrise printr-un set de atribute discrete* $\mathcal{A}$ *și etichetate cu câte o clasă dintr-o mulțime cunoscută* $\mathcal{C}$*, să se construiască un model pentru clasificarea exemplelor noi.*

2.2 Ce este un *arbore de decizie*?

Un *arbore de decizie* este un astfel de clasificator ce aproximează funcții discrete. Într-un *arbore de decizie* orice nod care nu este frunză conține un test pentru un atribut, având câte un arc (și implicit un subarbore) pentru fiecare valoare posibilă a atributului. Fiecare nod frunză este etichetat cu o clasă.

Pentru a clasifica un obiect nou se pornește din rădăcina arborelui și din fiecare nod se coboară pe arcul corespunzător valorii atributului pe care o are obiectul dat. Atunci când se ajunge într-un nod frunză, clasa acestuia va deveni și clasa exemplului.

3 Construirea arborilor de decizie

Construirea arborilor de decizie reprezintă un exemplu de învățare simbolică inductivă. Procesul de învățare este ghidat către construirea unor arbori cât mai mici. Pentru asta, la fiecare pas se va alege un atribut cât mai *bun* (care discriminează între clasele posibile / aduce cât mai multă informație despre clasa obiectelor).

Pentru a măsura omogenitatea (impuritatea) unui set de date $\mathbf{X}$ se folosește *entropia*:

$$Entropy(\mathbf{X}) = -\sum_{c \in \mathcal{C}} \frac{|\mathbf{X}_c|}{|\mathbf{X}|} \log_2 \frac{|\mathbf{X}_c|}{|\mathbf{X}|} \tag{1}$$

unde $\mathbf{X}_c = \{\mathbf{x} \in \mathbf{X} | class(\mathbf{x}) = c\}$.

Revenind la alegerea unui atribut pentru un nod nou în arborele de decizie, aceasta se va face preferând acel atribut care aduce un *câștig informațional* mai mare. Câștigul informațional corespunzător unui atribut A reprezintă scăderea entropiei provocată de partiționarea setului de date pe baza acelui atribut.

$$Gain(\mathbf{X}, A) = Entropy(\mathbf{X}) - \sum_{v \in values(A)} \frac{\mathbf{X}_v}{\mathbf{X}} Entropy(\mathbf{X}_v) \tag{2}$$

unde $\mathbf{X}_v = \{\mathbf{x} \in \mathbf{X} | x_A = v\}$.

3.1 Algoritmul ID3

Algoritmul **ID3**primește un set de date $\mathbf{X}$ (obiecte descrise prin valori pentru un set de atribute $\mathcal{A}$ și etichetate cu o clasă din $\mathcal{C}$). Algoritmul construiește recursiv un arbore de decizie pe baza acestuia.

Algoritmul funcționează astfel:

1. dacă toate exemplele din $\mathbf{X}$ aparțin unei singure clase c, atunci se construiește un singur nod frunză etichetat cu acea clasă;
2. dacă nu mai există atribute, atunci se construiește un nod frunză etichetat cu cea mai frecventă clasă din $\mathbf{X}$
3. altfel
   (a) se alege atributul $A* \in \mathcal{A}$ care aduce cel mai mare câștig informațional și se construiește un nod corespunzător acestuia;

$$A* = \underset{A \in \mathcal{A}}{\operatorname{argmax}}\, Gain(\mathbf{X}, A) \tag{3}$$

   (b) pentru fiecare valoare posibilă v a atributului $A*$ se construiește $\mathbf{X}_v$ (eliminând atributul $A*$ din $\mathcal{A}$);
   (c) pentru fiecare valoare posibilă v a atributului $A*$ se adaugă o muchie în arborele de decizie din nodul curent către subarborele construit aplicând recursiv algoritmul ID3 pentru $\mathbf{X}_v$.

4 Cerințe

Să se implementeze algoritmul **ID3** și să se testeze pe setul de date dat.