Învățare Automată - Laboratorul 3

Arbori de decizie. Păduri aleatoare

Tudor Berariu

Facultatea de Automatică și Calculatoare

Scopul laboratorului

- Scopul laboratorului îl reprezintă înțelegerea conceptului de *arbore de decizie*, precum și implementarea unor clasificatori pe baza acestui model.

- În cadrul laboratorului veți:
  1. construi un arbore de clasificare aleator;
  2. construi un arbore de decizie construit cu algoritmul ID3;
  3. construi păduri de arbori;
  4. compara metodele de mai sus.

Problema de rezolvat

Clasificare

Problema pe care o veți rezolva în cadrul acestui laborator este una de învățare supervizată: fiind dat un set de date $\mathbf{X}$ ce conține exemple descrise printr-o mulțime 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.

Seturile de date

Testați pe următoarele seturi de date:

- `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`
- `https://archive.ics.uci.edu/ml/datasets/Chess+%28King-Rook+vs.+King-Pawn%29`

Arbori de decizie. Păduri

Arbori de decizie

Un arbore de decizie este un 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.

Predicția pe baza arborilor de decizie

Pentru a clasifica un exemplu se pleacă din nodul rădăcină, coborând pe muchiile corespunzătoare valorilor pe care le are acesta pentru atributele întâlnite pe drum. Clasa întâlnită în nodul frunză reprezintă predicția arborelui.

Păduri de arbori

Predicția unei păduri de arbori este clasa majoritară între predicțiile independente ale tuturor arborilor.

Cerința 1: Arbori aleatori

- Implementați o funcție recursivă `random_tree` care construiește arbori de decizie de adâncime d astfel pe baza unui set de date $\mathbf{X}$ și a unei mulțimi de atribute $\mathcal{A}$ astfel:
  - Dacă $d = 0$, atunci se construiește un nod frunză cu clasa majoritară din $\mathbf{X}$.
  - Dacă $d > 0$, atunci se alege aleator un atribut a_i din $\mathcal{A}$ și se construiește câte un subarbore pentru fiecare valoare v_j a atributului a_i apelând `random_tree` pentru $d - 1$, $\mathbf{X}_{i/j} = \{\mathbf{x} \in \mathbf{X} | a_i(\mathbf{x}) = v_j\}$ și $\mathcal{A} \setminus \{a_i\}$.

Cerința 2: Algoritmul ID3

- Implementați o funcție recursivă `id3` care construiește arbori de decizie pe baza unui set de date $\mathbf{X}$ și a unei mulțimi de atribute $\mathcal{A}$ astfel:
  - Dacă toate exemplele din $\mathbf{X}$ aparțin unei singure clase c, atunci se construiește un nod frunză etichetat cu acea clasă c.
  - Dacă nu mai există atribute, atunci se construiește un nod frunză etichetat cu cea mai frecventă clasă din $\mathbf{X}$.
  - Altfel,
    1. se alege atributul a^* care aduce cel mai mare câștig informațional și se construiește un nod test corespunzător acestuia;

$$a^* = \underset{a_i \in \mathcal{A}}{\operatorname{argmax}} \left[entropy\left(\mathbf{X}\right) - \sum_{v_j \in vals(a_i)} \frac{|\mathbf{X}_{i/j}|}{|\mathbf{X}|} entropy\left(\mathbf{X}_{i/j}\right) \right]$$

$$entropy\left(\mathbf{X}\right) = -\sum_{c \in \mathcal{C}} \frac{|\mathbf{X}_c|}{|\mathbf{X}|} \log_2 \left(\frac{|\mathbf{X}_c|}{|\mathbf{X}|} \right)$$

    2. pentru fiecare valoare posibilă v_j a lui a^* se construiește un subarbore apelând recursiv `id3` pentru $\mathbf{X}_j = \{\mathbf{x} \in \mathbf{X} | a^*(\mathbf{x}) = v_j\}$ și $\mathcal{A} \setminus \{a^*\}$.

Cerința 3: Păduri de arbori aleatori

- Implementați în clasificator de tip pădure de arbori aleatori construind n arbori de adâncime maximă d fiecare dintre aceștia pornind de la o submulțime aleatoare a lui $\mathbf{X}$. Folosiți funcția `random_tree` de la cerința 1.
- Porniți de la $n = 100$, $d = 3$ și submulțimi formate din 50% din elementele lui $\mathbf{X}$ alese la întâmplare și experimentați cu acești hiperparametri.
- Pentru predicția clasei pentru obiecte noi alegeți clasa majoritară.
- Comparați rezultatele obținute folosind un singur arbore construit cu `id3` și o pădure de arbori aleatori. Discuție: zgomot, overfitting.

Cerința 4: păduri de arbori id3

- Modificați algoritmul `id3` astfel încât să construiască arbori de adâncime maximă d.
- Implementați un clasificator de tip pădure de arbori `id3`.