Directii de Cercetare in Explorarea Datelor

Prof. Dan A. Simovici

University of Massachusetts Boston

Departmentul de Informatica

Boston, Massachusetts, USA

Ce este Explorarea Datelor (Data Mining)?

ED: Procesul de identificare a unor fapte si proprietati ale datelor.

- ED foloseste a varietate de discipline informatice: base de date, inteligenta artificiala, logica si statistica.
- ED este aplicata frecvent unor volume mari de date; exista numeroase grupuri de date de volum relativ redus care creaza probleme dificile.

Ce incercam sa descoperim cu ED?

Probleme majore:

- Descoperirea asocierilor intre obiecte.
- Gruparea obiectelor in multimi de obiecte similare. (*clustering*)
- Classificarea obiectelor bazata pe proprietatile lor
- Evaluarea intersului faptelor si proprietatilor descoperite.
- Preparatea datelor (curatire, discretizare, etc.).

Cine are nevoie de ED?

- banci si cei care acorda credit;
- medici si biologi care incearca sa descopere cauzele bolilor si sa formuleze diagnostice;
- organizatii guvernamentale care incearca sa neutralizeze raufacatori;
- informaticieni care dirijeaza retele informatice si dezvolta algoritmi pentru cercetarea internetului;
- ecologi si biologi intersati sa descopere surse de poluare,
- si multi altii...

Ce cunostinte practice sunt necesare?

- base de date relationale; SQL si folosirea lui in C++, Java, si alte limbaje;
- algoritmi care lucreaza cu o varietate de structuri de date;
- gestionarea depozitelor de date (data warehousing);
- cunosterea pachetelor de programe principale: Clementine, SAS, WEKA, etc.

Ce cunostinte teoretice sunt necesare?

- diverse arii de matematica:

Clustering	spatii metrice
	algebra lineara si analiza functionala
Clasificare	teoria informatiei
	grafuri
Reguli de asociere	latici

- teoria complexitatii : NP- si #P-completitudine
- teoria informatiei;
- probabilitati si statistica.

Baze de date si data mining

Tycho Brache (1546–1601)	colector de date multe date, dar n-a extras legile astronomice
Johannes Kepler (1571-1630)	minier de date

Clustering

Important for:

- condensarea datelor (prezentarea concisa a datelor);
- identificarea tendintelor in date.

A. K. Jain (1999): “nu exista un algorithm pentru clustering care este universal applicabil in decoperirea oricarei structuri prezente in multimi multidimensionale de date”

Un exemplu de algorithm - clusterizarea incrementala

- date nominale
- clustering incremental

Caracteristica principala: Clustering-ul incremental

formeaza grupuri aduagind in mod successiv obiecte la grupuri (clusters) sau formind noi grupuri.

Date numerice si date nominale

- Date numerice:
  - inaltime: 1.82m, 1.25m, ...
  - temperatura: 38, 41, 54
- Date nominale:
  - culoare: rosu, verde, albastru,...
  - forma: patrat, romb, trapez,cerc,...

Distante pot fi definite in mod natural intre obiecte care au atribute numerice (folosing diferite metrici din R^n).

Dificultati cu datele nominale

Lipsa unei distante “naturale”: singura distanta ce se poate introduce este distanta Hamming, unde $d(o, o')$ este numarul de atribute in care o si o' sunt diferite.

Istoric

Algoritmi de grupare incrementala:

- Hartigan (1975)
- Fisher (1987) : COBWEB

Aplicatii

- F. Can et al.: baze de date de documente (1993–1995)
- Langford: detectarea focarelor de infectie din spitale (2001)
- J. Lin: serii temporale
- M. Charikar: regasirea dinamica a informatiei
- M. Ester: magazii de date (data warehouses)

Interesul clusteringului incremental

- Folosirea memoriei principale este minima
- Cerintele de timp cresc linear cu numarul de obiecte (scalable algorithm)

Sisteme de obiecte (SO)

Un sistem de obiecte este o pereche $\mathcal{S} = (S, H)$, unde

- S este o multime numita multimea de obiecte ale sistemului $\mathcal{S}$,
- $H = \{A_1, \ldots, A_m\}$ este o multime de functii definite pe S.

A_i (numit un atribut al lui $\mathcal{S}$) este o functie $A_i : S \longrightarrow E_i$, unde E_i este domeniul lui A_i.

Partitii

O partitie pe o multime S este o colectie nevida de parti ale lui S indexata de o multime I, $\pi = \{B_i | i \in I\}$ asa fel incit:

- $\bigcup_{i \in I} B_i = S$, si
- $i \neq j$ implica $B_i \cap B_j = \emptyset$.

B_i sunt *blocurile partitiei* π. Multimea partitiilor lui S este notata cu $\mathsf{PART}(S)$.

Laticea Partitiilor

$\pi \leq \sigma$ daca fiecare block B al partitiei π este inclus intr-un block al partitiei σ.

Daca $\pi, \pi' \in \mathsf{PART}(S)$ exists o partitie minimala π_1 astfel ca $\pi \leq \pi_1$ si $\pi' \leq \pi_1$; de asemenea, exista cea mai mare partitie π_2 pentru care $\pi_2 \leq \pi$ si $\pi_2 \leq \pi'$. Prima partitie se noteaza cu $\pi \vee \pi'$; a doua cu $\pi \wedge \pi'$.

Partitii generate de atribute

Un atribut A al sistemului $\mathcal{S} = (S, H)$ generereaza o partitie $\pi^A \in parts(S)$: doua obiecte apartin aceluiasi bloc al partitiei π^A daca au aceiasi proiectie pe A.
B_a^A: blocul lui π^A care consta din obiectele lui S care au componenta pe A egala cu a.
In baze de date relationale π^A se obtine folosind optiunea **group by** A al frazei **select** in standard SQL.

Partitii generate de multimi de attribute

T

tid	$\cdots$	L	$\cdots$
t_1	$\cdots$	a_1	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_i	$\cdots$	a_i	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_n	$\cdots$	a_n	$\cdots$

L generate o partitie
a multimii de obiecte
$t_j \equiv_L t_k$ daca si numai daca $t_j[L] = t$

Notam cu π_L partitia
generata de L

Partitii si dependente functionale

$$T$$

tid	$\cdots$	L	$\cdots$	K	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_i	$\cdots$	a_i	$\cdots$	b_i	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_j	$\cdots$	a_j	$\cdots$	b_j	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_n	$\cdots$	a_n	$\cdots$	b_n	$\cdots$

T satisface dependenta functionala$L \longrightarrow K$ daca $a_i = a_j$ implica $b_i = b_j$ pentru $i, j,$ adica, $t_i \equiv_L t_j$ implica $t_i \equiv_K t_j,$ adica, $\pi_L \leq \pi_K$

Clusterizari ca partitii

O clusterizare a unui sistem de obiecte $\mathcal{S} = (S, H)$ este o partitie κ a multimii de obiecte S.
Scopul nostru: determinarea grupajelor κ pornind de la legaturile lor cu partitile induse de atribute π^A.

Valuari si Metrici

- $v : \mathsf{PART}(S) \longrightarrow \mathbb{R}$ definita de $v(\pi) = \sum_{i=1}^{n} |B_i|^2$, unde $\pi = \{B_1, \ldots, B_n\}$ este o valuare inferioara pe $\mathsf{PART}(S)$:

$$v(\pi \vee \sigma) + v(\pi \wedge \sigma) \geq v(\pi) + v(\sigma) \tag{1}$$

  pentru $\pi, \sigma \in \mathsf{PART}(S)$.
- Pentru fiecare valuare inferioara v, functia $d : (\mathsf{PART}(S))^2 \longrightarrow \mathbb{R}$ data de $d(\pi, \sigma) = v(\pi) + v(\sigma) - 2v(\pi \wedge \sigma)$ este o distanta pe $\mathsf{PART}(S)$.

Criteriul de Optimalitate

Se cauta o grupare $\kappa = \{C_1, \ldots, C_n\} \in \mathsf{PART}(S)$ astfel ca distanta totala de la κ la partitiile atributelor:

$$D(\kappa) = \sum_{i=1}^{n} d(\kappa, \pi^{A_i})$$

sa fie minima.

Grupaje si partitiile atributelor

$$d(\kappa, \pi^A) = \sum_{i=1}^{n} |C_i|^2 + \sum_{j=1}^{m_A} |B^A_{a_j}|^2 - 2 \sum_{i=1}^{n} \sum_{j=1}^{m_A} |C_i \cap B^A_{a_j}|^2,$$

AMICA

(A Metric Incremental Clustering Algorithm)
Fie $t \notin S$, si fie $Z = S \cup \{t\}$. Urmatoarele situatii pot avea loc:

1. obiectul t este adagat unui grup(cluster) existent C_k, sau
2. un nou grup, C_{n+1} este creat care consista doar din t.

Relativ la π^A, t se adauga blocului $B^A_{t[A]}$.

Obiectul se adauga unui cluster existent

$$\begin{aligned}\kappa_{(k)} &= \{C_1, \ldots, C_{k-1}, C_k \cup \{t\}, C_{k+1}, \ldots, C_n\} \\ \pi^{A'} &= \{B^A_{a_1}, \ldots, B^A_{t[A]} \cup \{t\}, \ldots, B^A_{a_{m_A}}\}\end{aligned}$$

$$d(\kappa_{(k)}, \pi^{A'}) - d(\kappa, \pi^A) = 2|C_k \oplus B^A_{t[A]}|.$$

Cresterea minima a $d(\kappa_{(k)}, \pi^{A'})$ este data de:

$$\min_k \sum_A 2|C_k \oplus B^A_{t[A]}|.$$

Obiectul formeaza un nou cluster

$$
\begin{aligned}
\kappa' &= \{C_1, \ldots, \ldots, C_n, \{t\}\} \\
\pi^{A'} &= \{B^A_{a_1}, \ldots, B^A_{t[A]} \cup \{t\}, \ldots, B^A_{a_{m_A}}\}
\end{aligned}
$$

$$d(\kappa', \pi^{A'}) - d(\kappa, \pi^A) = 2|B^A_{t[A]}|.$$

Directie de actionare

$$D(\kappa') - D(\kappa) = \begin{cases} 2 \cdot \sum_A |C_k \oplus B^A_{t[A]}| & \text{in Case 1} \\ 2 \cdot \sum_A |B^A_{t[A]}| & \text{in Case 2.} \end{cases}$$

Daca $\min_k \sum_A |C_k \oplus B^A_{t[A]}| < \sum_A |B^A_{t[A]}|$ se adauga t la clusterul C_k pentru care $\sum_A |C_k \oplus B^A_{t[A]}|$ este minima; altfel se creaza un nou cluster cu un singur obiect.

Dificultatile grupajului incremental

- Algoritmii de grupare incrementala sunt afectate, in general, de ordinea de prelucrare a obiectelor.
- Fiecare algoritm procedeaza intr-o maniera "hill-climbing" care produce minime locale (si nu globale).

Limitarea efectului ordonarii obiectelor

Am folosit tehnica “not-yet” introdusa de Roure si Talavera:

NOT-YET: Un nou grupaj este creat numai daca conditia

$$r(t) = \frac{\sum_A |B^A_{t[A]}|}{\min_k \sum_A |C_k \oplus B^A_{t[A]}|} < \alpha,$$

este satisfacuta, adica, numai daca effectul adaugarii obiectului t asupra distantei totale $r(t)$ este suficient de semnificativ.

$\alpha \leq 1$ este un parametru dat de utilizator (daca $\alpha = 1$ obiectele nu sunt trimise la buffer).

Algorithmul AMICA

Intrari: Setul de date S si α

Iesiri: clustering $C_1, \ldots, C_{\mathsf{nc}}$

Metoda:

$\mathtt{nc} = 0; \ell = 1;$

while $S \neq \emptyset$ do

select an object t; $S = S - \{t\}$;

if $\sum_A |B^A_{t[A]}| \leq \alpha \min_{1 \leq k \leq \mathsf{nc}} \sum_A |C_k \oplus B^A_{t[A]}|$

then

$\mathtt{nc}$ ++; create a new single-object cluster $C_{\mathsf{nc}} = \{t\}$;

else

$r(t) = \sum_A |B^A_{t[A]}| / \min_{1 \leq k \leq \mathsf{nc}} \sum_A |C_k \oplus B^A_{t[A]}|$

if $r(t) > 1$

then $k = \mathtt{arg\ min}_k \sum_A |C_k \oplus B^A_{t[A]}|$

add t to cluster C_k;

else /* this means $\alpha < r(t) \leq 1$ */

place t in NOT-YET buffer;

end if;

Experimente cu date produse sintetic

- Date sintetice: produse de un algoritm care genereaza obiecte cu componente reale grupate in jurul unui numar dat de centre.
- Datele au fost discretizate folowing un numar specific de intervale de discretizare, ceea ce ne permite sa tratam datele ca date nominale.
- Am experimentat cu citeva multimi de date cu un numar crescind de obiecte, cu un numar crescind de dimensiuni, folsing citeva permutari ale obiectelor.
- Toate experimentele folosesc $\alpha = 0.95$.

Stabilitatea Grupurilor

- Experiment executat pe o baza de date care consta din 10,000 de objecte (grupate in jurul a 6 centroizi)
- O prima aplicare a algoritmului genereaza 11 grupuri.
- Cele mai multe obiecte (9895) sunt concentrate in 6 grupuri, ceea ce reprezinta o buna aproximare a grupurilor “naturale” produse de algoritmul de generare.

AMICA este relativ imuna la permutari

Initial		Permutatare Aleatoare		
Cluster	Mar.	Cluster	Mar.	Distributie (cluster original)
1	1548	1	1692	1692 (2)
2	1693	2	1552	1548 (1), 3 (3), 1 (2)
3	1655	3	1672	1672 (5)
4	1711	4	1711	1711 (4)
5	1672	5	1652	1652 (3)
6	1616	6	1616	1616 (6)
7	1	7	85	85 (8)
8	85	8	10	10 (9)
9	10	9	8	8 (10)
10	8	10	1	1 (11)
11	1	11	1	1 (7)

Scalabilitate

Numar de obiecte	Timp pt. 3 permutari (ms)			Timp mediu (ms)
2000	131	140	154	141.7
5000	410	381	432	407.7
10000	782	761	831	794.7
20000	1103	1148	1061	1104

Setul de date CIUPERCI

- Setul de date contine 8124 descrieri de ciuperci si este tipic folosit pentru probleme de clasificare.
- Algoritmii de clasificare incearca sa determine daca un tip de ciuperca este comestibil sau otravitor.
- Atributul (otravitor/comestibil) este eliminat si AMICA a fost aplicat la setul de date fara acest atribut.

Rezultate experimentale

Cl. no.	O/C	Total	Procentul grupului dominant
1	825/2752	3577	76.9%
2	8/1050	1058	99.2%
3	1304/0	1304	100%
4	0/163	163	100%
5	1735/28	1763	98.4%
6	0/7	7	100%
7	0/192	192	100%
8	36/16	52	69%
9	8/0	8	100%

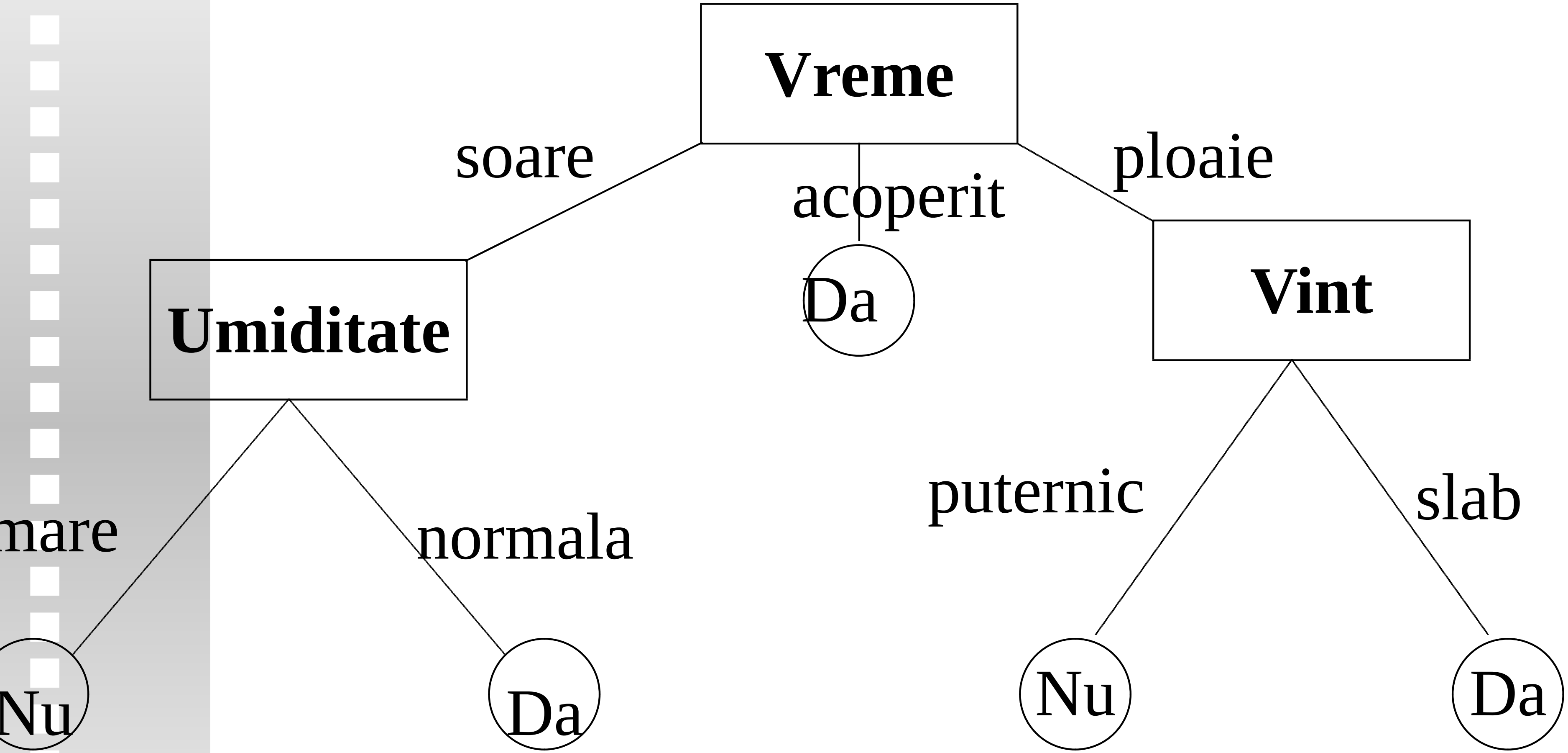
Stabilitate la Permutari

C_i	Grupuri Calculate Permutatare aleatoare									
	C'_1	C'_2	C'_3	C'_4	C'_5	C'_6	C'_7	C'_8	C'_9	C'_{10}
	3540	1797	1095	192	1296	8	36	7	137	16
3577	3540	0	37	0	0	0	0	0	0	0
1058	0	0	1058	0	0	0	0	0	0	0
1304	0	8	0	0	1296	0	0	0	0	0
163	0	26	0	0	0	0	0	0	137	0
1763	0	1763	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	7	0	0
192	0	0	0	192	0	0	0	0	0	0
52	0	0	0	0	0	0	36	0	0	16
8	0	0	0	0	0	8	0	0	0	0

Probleme inrudite

- Continuarea studiului experimental cu alte valori ale factorului “not-yet” α.
- Combinarea algoritmului AMICA cu tehnici speciale de discretizare pentru extinderea algoritmului la date cu caracter mix,
- Grupare incrementala in varianta “Semi-supervised” bazata pe AMICA.
- IC aplicat la date de tip “stream”

Arbori de decizie



Cum classifica arborii de decizie

$$(\textbf{Vreme} = \text{soare}, \textbf{Temperatura} = \text{cald},$$
$$\textbf{Umiditate} = \text{mare}, \textbf{Vint} = \text{puternic})$$

Orice arbore de decizie este reprezentat de o disjunctie de conjunctii:

$$((\textbf{Vreme} = \text{soare} \wedge (\textbf{Umiditate} = \text{normala})$$
$$\vee(\textbf{Vreme} = \text{acoperit})$$
$$\vee((\textbf{Vreme} = \text{ploaie} \wedge (\textbf{Vint} = \text{slab})$$

Entropia lui Shannon

$$X : \begin{pmatrix} a_1 & \cdots & a_n \\ p_1 & \cdots & p_n \end{pmatrix}, \text{ where } p_1 + \cdots + p_n = 1.$$

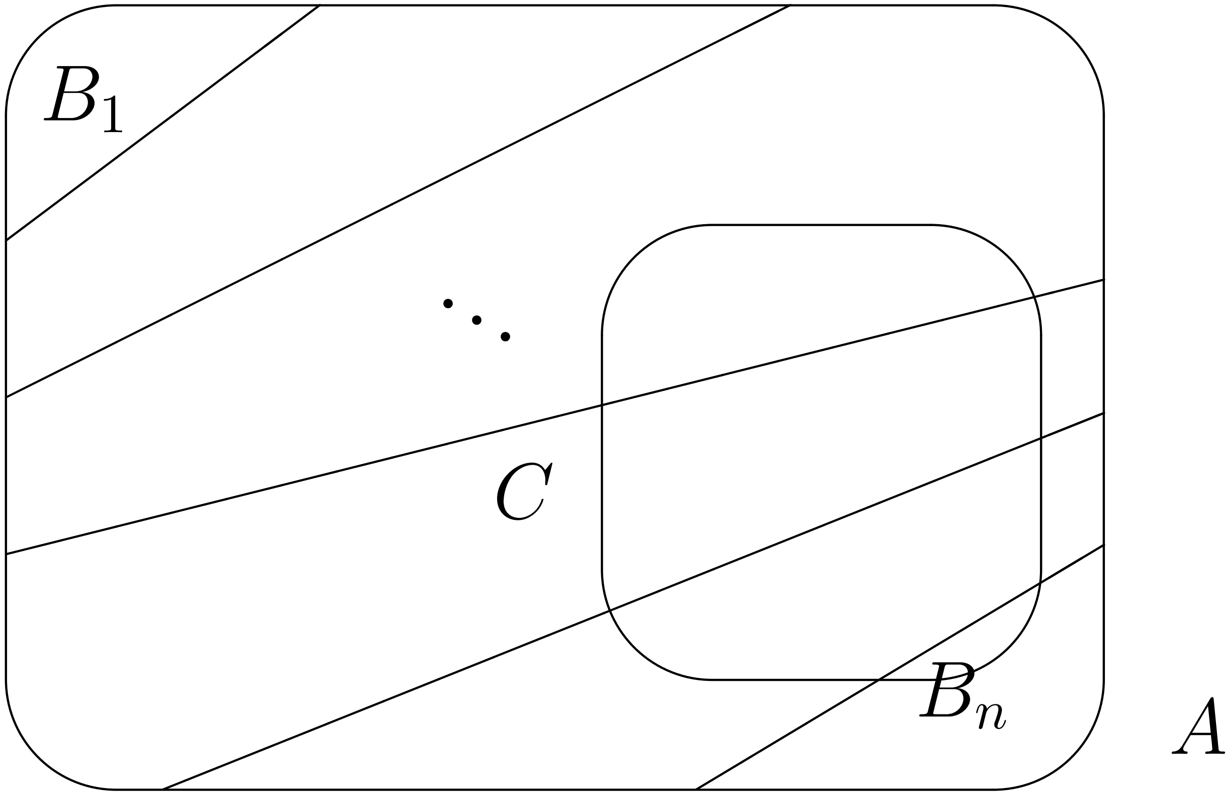
Entropia lui X este
$\mathcal{H}(X) = p_1 \log_2 \frac{1}{p_1} + \cdots + p_n \log_2 \frac{1}{p_n}$.
Daca $\pi = \{B_1, \ldots, B_n\}$ este o partitie a multimii A atunci entropia lui π este:

$$\mathcal{H}(\pi) = -\sum_{i=1}^{n} \frac{|B_i|}{|A|} \log_2 \frac{|B_i|}{|A|}.$$

Urma unei partitii

Fie $\pi = \{B_1, \ldots, B_n\}$ a partitie a multimii A si $C \subseteq A$.

Urma partitiei π pe C este $\pi_C = \{B_i \cap C | B_i \cap C \neq \emptyset\}$



Entropia Conditionala a Partitiilor

$$\text{Fie } \begin{aligned} \pi &= \{B_1, \ldots, B_n\} \\ \sigma &= \{C_1, \ldots, C_m\} \end{aligned}$$

doua partitii alse multimii A. The entropia conditionala a lui π prin σ este:

$$\mathcal{H}(\pi|\sigma) = \sum_{j=1}^{m} \frac{|C_j|}{|C|} \mathcal{H}(\pi_{C_j})$$

Cistigul lui π relativ la σ este:

$$\mathrm{Gain}(\pi, \sigma) = \mathcal{H}(\pi) - \mathcal{H}(\pi|\sigma)$$

Partitii si Arbori de Decisie

Alegera atributului de separare (splitting attribute) intr-un arbore de decizie se face in (ID3, sau C5.1 - Quinlan) folosind cistigul informational:
Fie K este atributul care defineste clasa, atunci alegerea atributului de separare A se face maximizind

$$\text{Gain}(\pi_K, \pi_A) = \mathcal{H}(\pi_K) - \mathcal{H}(\pi_K | \pi_A)$$

(Quinlan's ID3 or C4.5,...)

Zile favorabile pt. tenis

Zi	Vreme	Temp.	Umid.	Vint	Tenis
z1	soare	cald	rid	slab	nu
z2	soare	cald	rid	tare	nu
z3	acoperit	cald	rid	slab	da
z4	ploaie	mod	rid	slab	da
z5	ploaie	rece	nor	slab	da
z6	ploaie	rece	nor	tare	nu
z7	acoperit	rece	nor	tare	da
z8	soare	mod	rid	slab	nu
z9	soare	rece	nor	slab	da
z10	ploaie	mod	nor	slab	da
z11	soare	mod	nor	tare	da
z12	acoperit	mod	rid	tare	da
z13	acoperit	cald	nor	slab	da
z14	ploaie	mod	rid	tare	nu

$$\mathcal{H}(\pi_{tenis}) = -\frac{5}{14}\log\frac{5}{14} - \frac{9}{14}\log\frac{9}{14} = 0.940$$

Continuarea Exemplului

Penrtu **vreme**:

$$
\begin{aligned}
C_{soare} &= \{z1, z2, z8, z9, z11\} \\
C_{acoperit} &= \{z3, z7, z12, z13\} \\
C_{ploaie} &= \{z4, z5, z6, z10, z14\}
\end{aligned}
$$

Urmele partitiei π_{tenis}:

$$
\begin{aligned}
\pi_{tenis_{C_{soare}}} &= \{\{z1, z2, z8\}, \{z9, z11\}\} \\
\pi_{tenis_{C_{acoperit}}} &= \{\{z3, z7, z12, z13\}\} \\
\pi_{tenis_{C_{ploaie}}} &= \{\{z6, z14\}, \{z4, z5, z10\}\}
\end{aligned}
$$

Urmele partitiei π_{tenis}:

$$\begin{aligned}
\pi_{tenis_{C_{soare}}} &= \{\{z1, z2, z8\}, \{z9, z11\}\} \\
\pi_{tenis_{C_{acoperit}}} &= \{\{z3, z7, z12, z13\}\} \\
\pi_{tenis_{C_{ploaie}}} &= \{\{z6, z14\}, \{z4, z5, z10\}\}
\end{aligned}$$

$$\begin{aligned}
\mathcal{H}(\pi_{tenis_{C_{soare}}}) &= -\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\frac{2}{5} = 1.116 \\
\mathcal{H}(\pi_{tenis_{C_{acoperit}}}) &= -\frac{4}{4}\log\frac{4}{4} = 0 \\
\mathcal{H}(\pi_{tenis_{C_{ploaie}}}) &= -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\frac{3}{5} = 1.116
\end{aligned}$$

Calcule Similare

$$\begin{aligned} \text{Gain}(\pi_{tenis}, \pi_{vreme}) &= 0.247 \\ \text{Gain}(\pi_{tenis}, \pi_{umiditate}) &= 0.151 \\ \text{Gain}(\pi_{tenis}, \pi_{vint}) &= 0.048 \\ \text{Gain}(\pi_{tenis}, \pi_{vreme}) &= 0.029 \end{aligned}$$

Atributul de scindare: **vreme**

Probleme generate de crieriul de cistig

- Alegera atributului de scindare este pur locala. Arborele care rezulta nu este optimal in mod necesar.
- Arborii care resulta pot avea multe virfuri terminale, ceea ce provoaca o fragmentare excesiva a datelor.

Metrici si arbori de decizie

López de Mántaras introduce o distanta bazata pe entropia Shannon.

$$d(\pi, \sigma) = \mathcal{H}(\pi|\sigma) + \mathcal{H}(\sigma|\pi).$$

Un nou criteriu de alegere a attributului de scindare:

$$A = \text{ arg } \min d(\pi_K, \pi_A)$$

Suma a doua partitii

Daca $M \cap P = \emptyset$ si

$$\begin{aligned} \pi &= \{B_1, \ldots, B_m\} \in \mathsf{PART}(M), \\ \sigma &= \{C_1, \ldots, C_n\} \in \mathsf{PART}(P), \end{aligned}$$

definim $\pi + \sigma$ ca partitia multimii $M \cup P$:

$$\pi + \sigma = \{B_1, \ldots, B_m, C_1, \ldots, C_n\}.$$

Daca M, P, Q sunt disjuncte si
$\pi \in \mathsf{PART}(M), \sigma \in \mathsf{PART}(P), \tau \in \mathsf{PART}(Q)$, atunci

$$\pi + (\sigma + \tau) = (\pi + \sigma) + \tau.$$

Axiomatizarea Entropiei Generalizate

Fie $\Phi : \mathbb{R}^2_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$ o functie continua, unde $\Phi(x, y) = \Phi(y, x)$, $\Phi(x, 0) = x$ pentru $x, y \in \mathbb{R}_{\geq 0}$ si $\beta \in \mathbb{R}$, $\beta > 0$.
Sistemul de axiome (Φ, β) pentru $\mathcal{H} : \mathsf{PART}(A) \longrightarrow \mathbb{R}_{\geq 0}$ consta din

(P1) Daca $\pi, \pi' \in \mathsf{PART}(A)$, $\pi \leq \pi'$, atunci $\mathcal{H}(\pi') \leq \mathcal{H}(\pi)$.

(P2) Daca A, B sunt doua multimi finite, $|A| \leq |B|$, atunci $\mathcal{H}(\iota_A) \leq \mathcal{H}(\iota_B)$.

(P3) Pentru A, B, $A \cap B = \emptyset$, $\pi \in \mathsf{PART}(A)$ si $\sigma \in \mathsf{PART}(B)$ avem:

$$\begin{aligned} &\mathcal{H}(\pi + \sigma) \\ &= \left(\frac{|A|}{|A| + |B|} \right)^{\beta} \mathcal{H}(\pi) + \left(\frac{|B|}{|A| + |B|} \right)^{\beta} \mathcal{H}(\sigma) \\ &\quad + \mathcal{H}(\{A, B\}). \end{aligned}$$

(P4) Daca $\pi \in \mathsf{PART}(A)$ si $\sigma \in \mathsf{PART}(B)$, atunci

$$\mathcal{H}(\pi \times \sigma) = \Phi(\mathcal{H}(\pi), \mathcal{H}(\sigma)).$$

- β determina o entropie $\mathcal{H}_\beta$ pina la un factor constant. β determina si functia Φ.
- Daca $\beta \neq 1$ atunci for a partition $\pi = \{A_1, \ldots, A_n\} \in \mathsf{PART}(A)$ we have:

$$\mathcal{H}_\beta(\pi) = \frac{k}{\beta - 1} \left(1 - \sum_{j=1}^{n} \left(\frac{|A_j|}{|A|} \right)^\beta \right),$$

unde k este o constanta astfel ca $k(\beta - 1) > 0$.

- Daca $\beta \neq 1$ avem $\Phi(x, y) = x + y - \frac{1}{k}xy$ pentru $x, y \in \mathbb{R}_{\geq 0}$.
- Daca $\beta = 2$ avem indexul Gini:

$$\mathcal{H}_2(\pi) = c\left(1 - \sum_{j=1}^{n}\left(\frac{|A_j|}{|A|}\right)^2\right).$$

- Cazul limita $\beta \to 1$ da entropia Shannon, adica

$$\mathcal{H}_1(\pi) = -c\sum_{j=1}^{n}\frac{|A_j|}{|A|}\log_2\frac{|A_j|}{|A|}.$$

si $\Phi(x, y) = x + y$ for $x, y \in \mathbb{R}_{\geq 0}$.

Entropia conditional data de (Φ, β)-entropy $\mathcal{H}$ este $\mathcal{H}_\beta : \mathsf{PART}^2 \longrightarrow \mathbb{R}_{\geq 0}$:

$$\mathcal{H}_\beta(\pi|\sigma) = \sum_{j=1}^{n} \left(\frac{|C_j|}{|A|}\right)^\beta \mathcal{H}_\beta(\pi_{C_j}),$$

unde $\pi, \sigma \in \mathsf{PART}(A)$ si $\sigma = \{C_1, \ldots, C_n\}$.
$\mathcal{H}_\beta(\pi|\omega_A) = \mathcal{H}_\beta(\pi)$.

Daca $\pi \in \mathsf{PART}(A)$ avem:

- $\mathcal{H}(\pi) = 0$ daca si numai daca $\pi = \omega_A$.
- Daca $\pi, \sigma \in \mathsf{PART}(A)$ avem $\mathcal{H}_\beta(\pi|\sigma) = 0$ daca si numai daca $\sigma \leq \pi$.

- Fie $\pi, \sigma, \sigma' \in \mathsf{PART}(A)$. Daca $\sigma \leq \sigma'$ atunci $\mathcal{H}_\beta(\pi|\sigma) \leq \mathcal{H}_\beta(\pi|\sigma')$ for $\beta > 0$.
- Fie $\pi, \sigma \in \mathsf{PART}(A)$ si $\beta > 0$. Avem $\mathcal{H}_\beta(\pi|\sigma) \leq \mathcal{H}_\beta(\pi)$.
- Daca $\pi, \pi', \sigma \in \mathsf{PART}(A)$ astfel ca $\pi \leq \pi'$ atunci $\mathcal{H}_\beta(\pi|\sigma) \geq \mathcal{H}_\beta(\pi'|\sigma)$.
- Pentru $\beta \geq 1$ avem $\mathcal{H}_\beta(\pi \wedge \sigma) \leq \mathcal{H}_\beta(\pi) + \mathcal{H}_\beta(\sigma)$.

Daca $\beta \geq 1$ si $\pi, \sigma, \tau \in \mathsf{PART}(A)$ avem inegalitatea:

$$\mathcal{H}_\beta(\pi|\sigma) + \mathcal{H}_\beta(\sigma|\tau) \geq \mathcal{H}_\beta(\pi|\tau).$$

Rezultatul nostru generalizeaza resultatul lui López de Mántaras:
Daca $\beta \geq 1$ fie $d_\beta : \mathsf{PART}(A)^2 \longrightarrow \mathbb{R}_{\geq 0}$ definita de $d_\beta(\pi, \sigma) = \mathcal{H}_\beta(\pi|\sigma) + \mathcal{H}_\beta(\sigma|\pi)$ for $\pi, \sigma \in \mathsf{PART}(A)$.
d_β este o metrica pe $\mathsf{PART}(A)$.

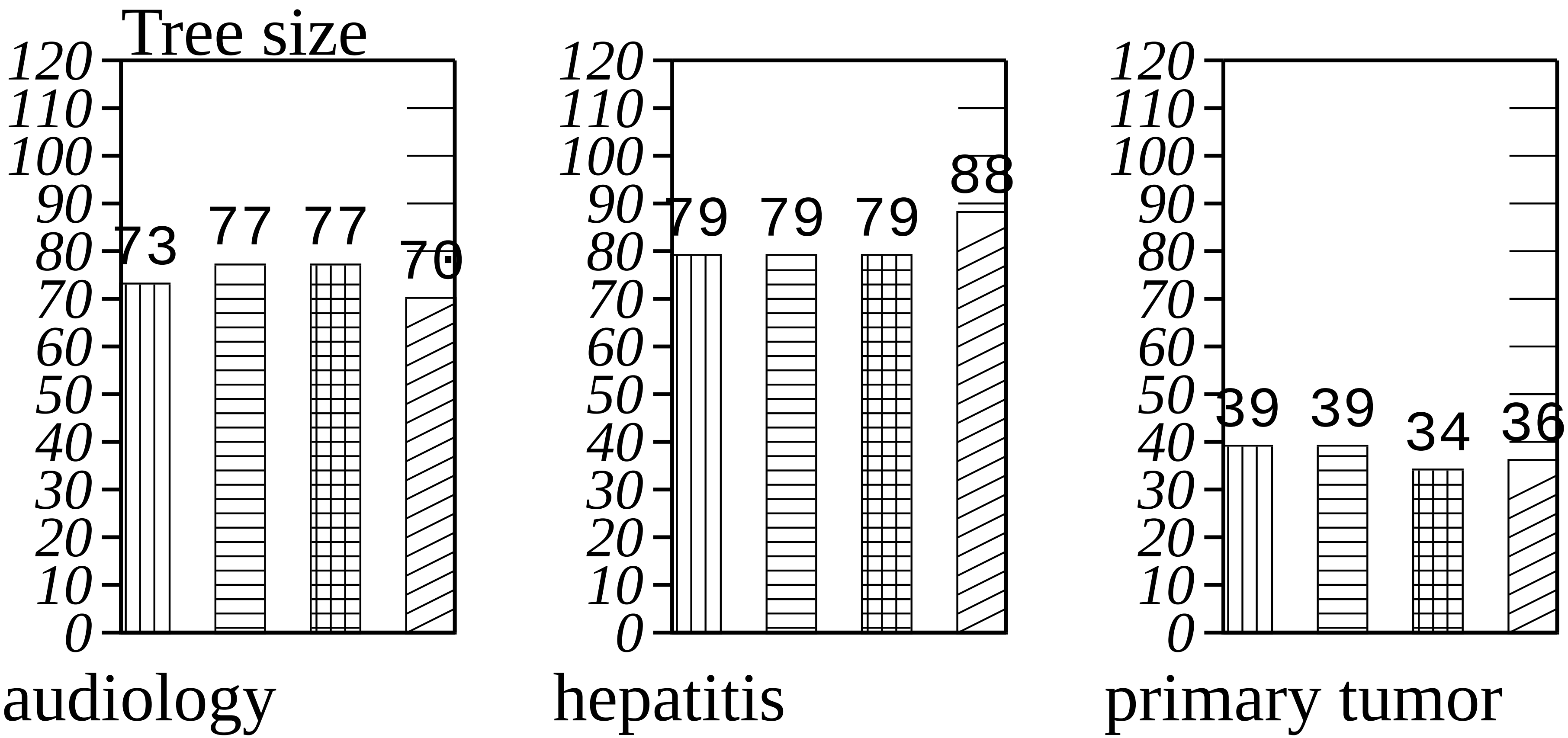
O noua alegere a atributului de scindare:

$$A = \text{arg } \min d(\pi_K, \pi_A)$$

O noua problema: alegerea cea mai buna a parametrului β pentru o multime de date depinde de properietatile ei statistice.

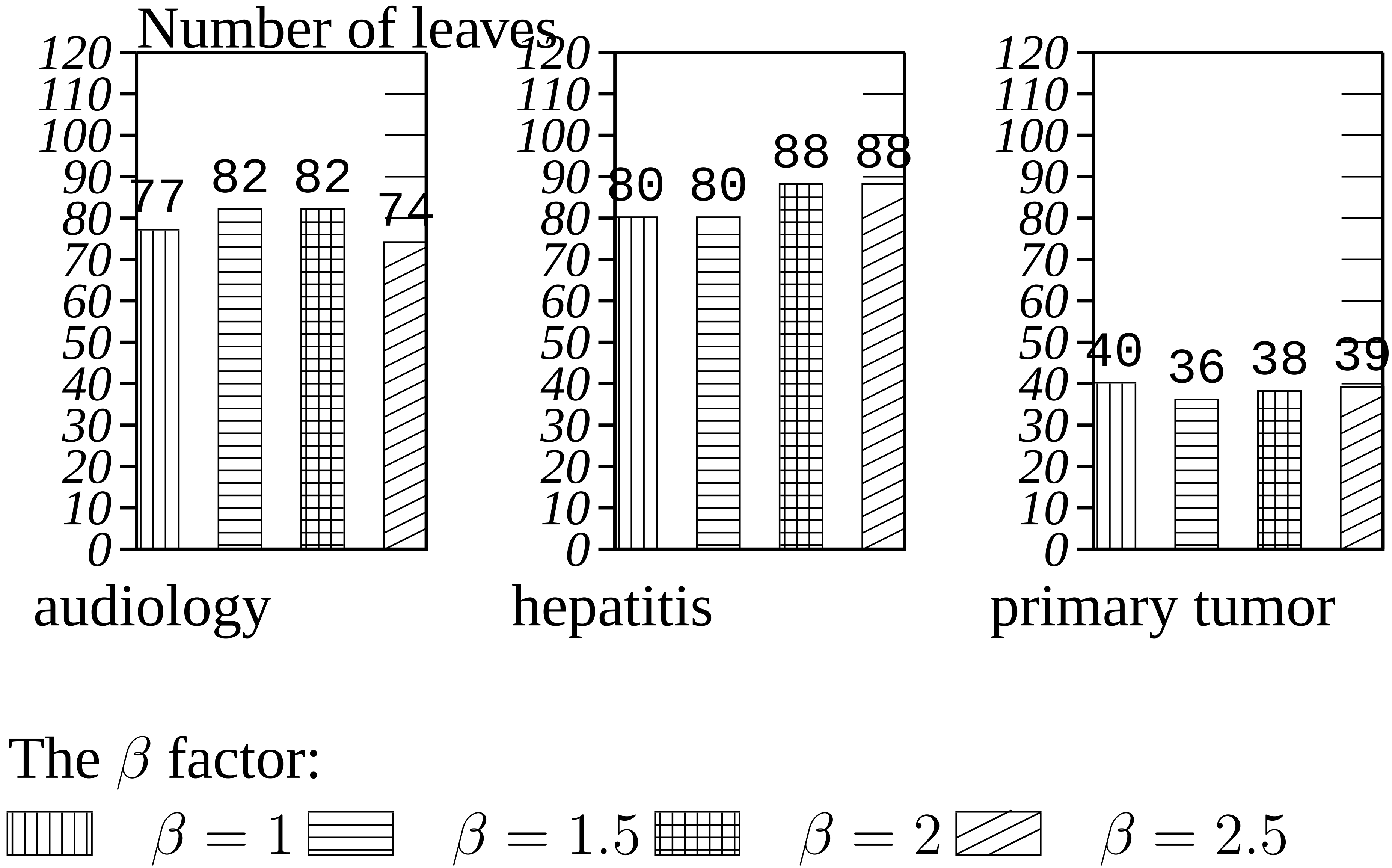
- Am experimentat cu 33 baze de date din colectia UCI.
- Fiecate experiment a folosit o 5-validere incrucisata; media a fost obtinuta pentru 5 experimente.

- Dimensiunea si numarul de virfuri terminale descreste pentru 18 din cele 33 baze de date si creste pentru celelalte 15.
- Cea mai importanta reducere a fost obtinuta pentru `primary-tumor`, unde numarul total de noduri a fost redus cu 37% pentru $\beta = 2.5$, iar numarul de noduri terminale a fost redus cu 38.8% in comparatie cu algoritmul standard (C5.0).



The β factor:

$\beta = 1$ $\beta = 1.5$ $\beta = 2$ $\beta = 2.5$


Number of leaves
120
110
100
90
80
70
60
50
40
30
20
10
0
77
82
82
74
80
80
88
88
40
36
38
39
audiology
hepatitis
primary tumor
The β factor:
$\beta = 1$
$\beta = 1.5$
$\beta = 2$
$\beta = 2.5$

Unde ne putem informa despre DM?

- Conferinte principale:
  - KDD (USA)
  - PKDD (Europa)
  - PAKDD (Asia si Australia)
  - ICDM (anul acesta la Brighton, UK)
  - ICML
- TKDE (IEEE), Journal of Data Mining
- KDNuggets
- Internetul (CiteSeer)