

Facultatea de Electronică, Telecomunicații și Tehnologie Informației
AIM AI Multimedia Lab
<https://www.aimultimedialab.ro/>
Universitatea Politehnică din București

Tehnici de analiză și clasificare automată a informației

Prof. dr. ing. Bogdan IONESCU
<https://bionescu.aimultimedialab.ro/>

București, 2022

1

Plan Curs

- M1. Introducere (concept, aplicații)
- M2. Prelucrarea și reprezentarea datelor de intrare
- M3. Tehnici de clasificare ne-supervizată ("clustering")
- M4. Tehnici de clasificare supervizată ("classification")
- M5. Evaluarea performanței clasificatorilor

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

2

> M4. Tehnici de clasificare supervizată ("classification")

- 4.1. [Introducere]
- 4.2. [k-NN]
- 4.3. [Support Vector Machines]
- 4.4. [Arbori de decizie]

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

3

Clasificare supervizată (classification) - principiu

classification = partiționarea datelor de intrare în mulțimi similare pe baza unor exemple a priori de astfel de partiții (date de antrenare);

date de intrare

date de antrenare

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

4

Clasificare supervizată (classification) – principiu (cont.)

classification = partiționarea datelor de intrare în mulțimi similare pe baza unor exemple a priori de astfel de partiții (date de antrenare);

date de intrare

partiționare

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

5

Clasificare supervizată (classification) – principiu (cont.)

classification = partiționarea datelor de intrare în mulțimi similare pe baza unor exemple a priori de astfel de partiții (date de antrenare);

date de intrare

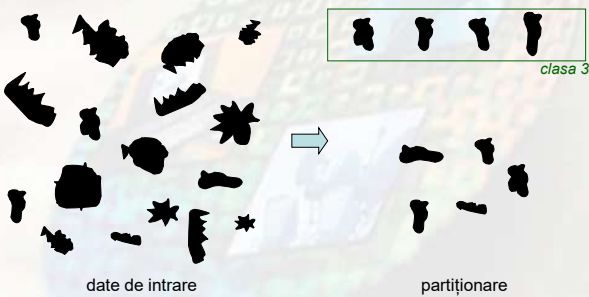
partiționare

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

6

Clasificare supervizată (classification) - principiu (cont.)

classification = partiționarea datelor de intrare în mulțimi similare pe baza unor exemple a priori de astfel de partiții (date de antrenare);



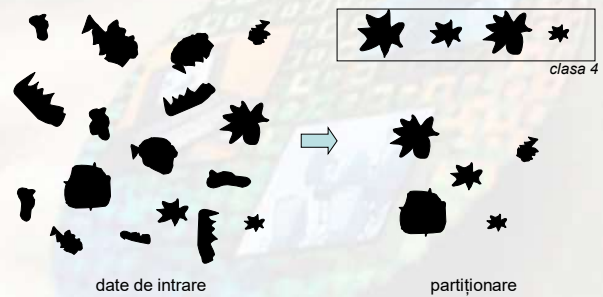
Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

7

7

Clasificare supervizată (classification) - principiu (cont.)

classification = partiționarea datelor de intrare în mulțimi similare pe baza unor exemple a priori de astfel de partiții (date de antrenare);



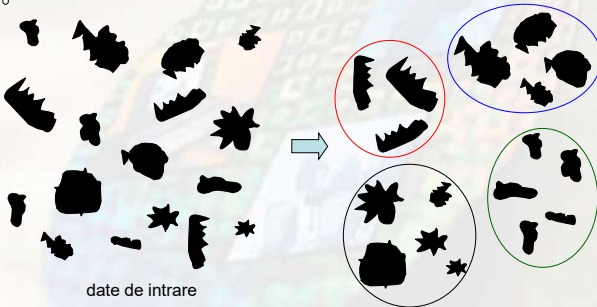
Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

8

8

Clasificare supervizată (classification) - principiu (cont.)

classification = partiționarea datelor de intrare în mulțimi similare pe baza unor exemple a priori de astfel de partiții (date de antrenare);



Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

9

9

k-NN

datele de intrare sunt clasificate pe baza unui vot majoritar cu privire la clasa de apartenență a celor mai apropiați k vecini;

> date de intrare:

- date de antrenare:

$$[Y_1 \in C_1, Y_2 \in C_2, \dots, Y_m \in C_m], Y_j = [y_{j,1}, \dots, y_{j,n}], C_j \in \{1, \dots, C\}$$

- date de clasificat:

$$X_i = [x_{i,1}, \dots, x_{i,n}], i = 1, \dots, n;$$

> algoritm:

p1. se alege o valoare pentru k (ex. 1, 3, 5 etc);

> antrenare:

p2. sunt stocate datele etichetate (lazy classifier);

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

10

10

k-NN (cont.)

> algoritm (cont.):

> clasificare:

p3. pentru fiecare instanță de clasificat, X_i , se calculează distanța către toate datele de antrenare, $Y_j, j=1, \dots, m$;

p4. se determină cele mai apropiate k date de antrenare;

p5. instanța X_i este clasificată ca aparținând clasei predominante din cele k date deja etichetate (vot majoritar);

p6. procesul se repetă până când sunt clasificate toate instanțele de intrare.

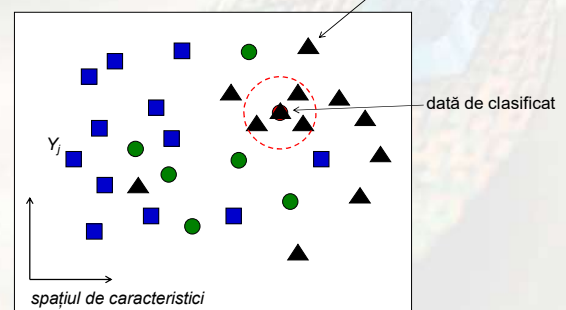
Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

11

11

k-NN (cont.)

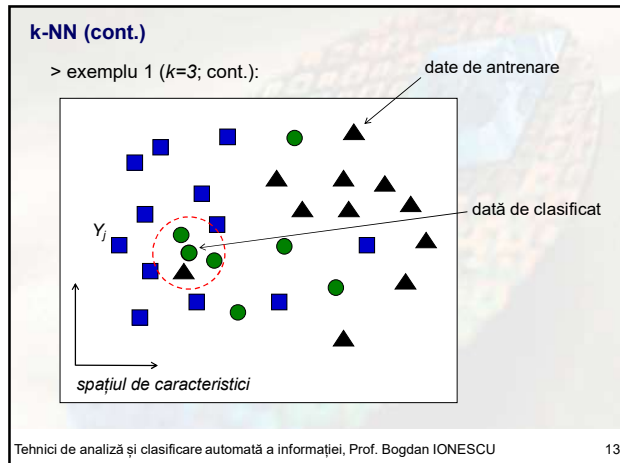
> exemplu 1 ($k=3$):



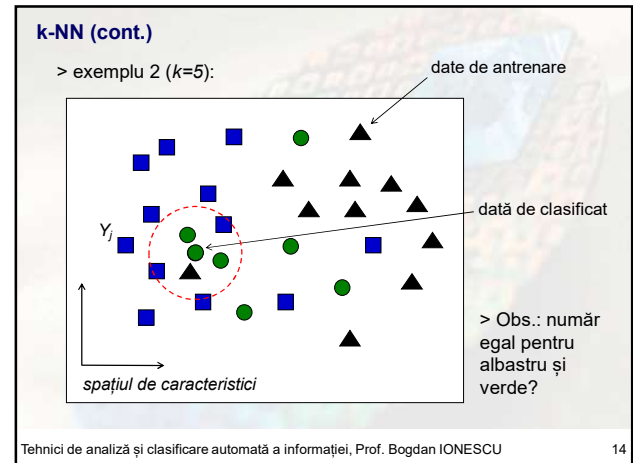
Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

12

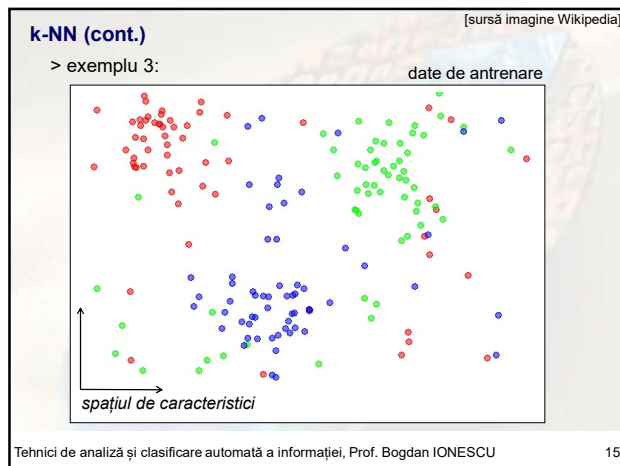
12



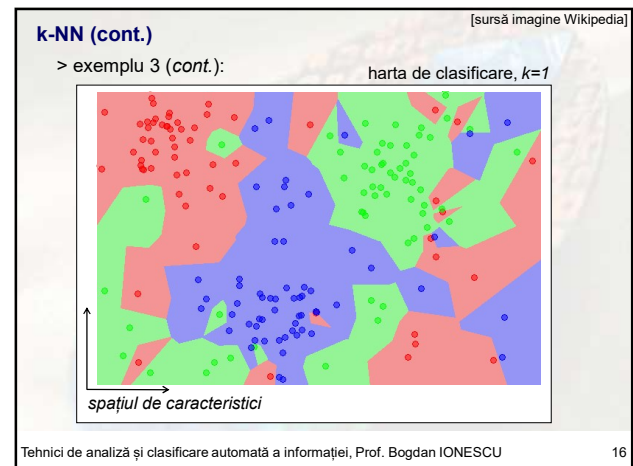
13



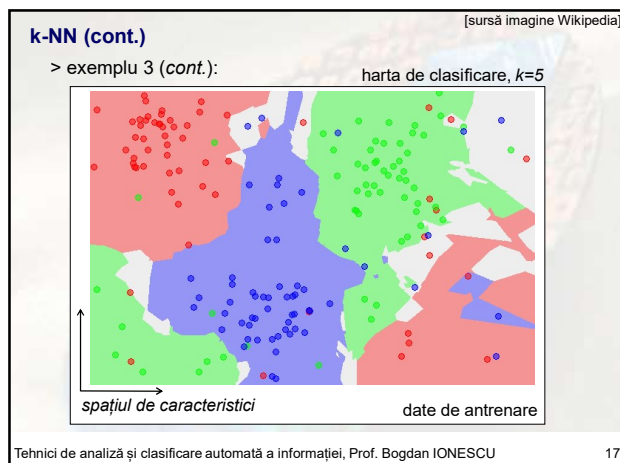
14



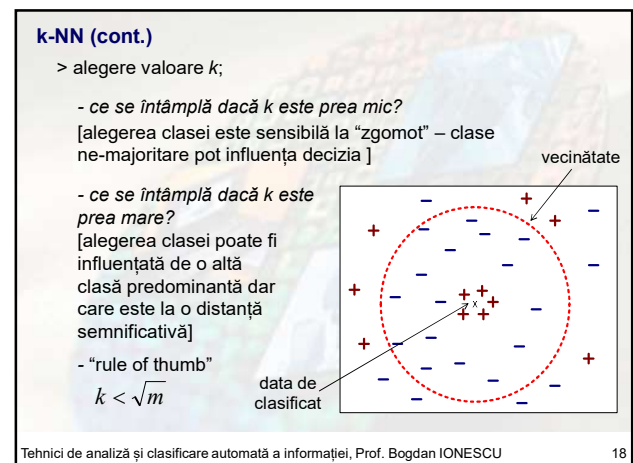
15



16



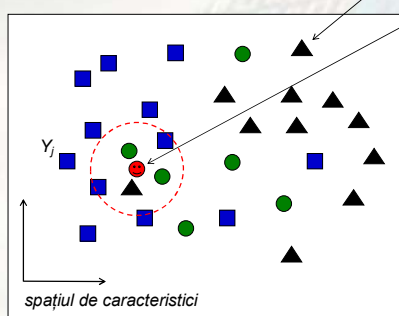
17



18

k-NN (cont.)

> îmbunătățire decizie (exemplu 2, $k=5$);



date de antrenare

dată de clasificat

> număr egal de clase
"majoritare" ?

> *îmbunătățire*:
ponderarea contribuției
vecinilor la votul
majoritar:

$$w = \frac{1}{d(X_i, Y_j)^2}$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

19

19

k-NN (cont.)

> avantaje:

- poate fi aplicat datelor de orice tip de distribuție (ex. nu neapărat linear separabile);
- simplu și intuitiv;
- eficient când numărul de date de antrenare este foarte mare.

> dezavantaje:

- alegerea valorii lui k ;
- complexitate matematică ridicată - nu există o etapă preliminară de antrenare (care poate fi realizată offline);
- pentru o bună performanță necesită un număr semnificativ de exemple (avantaj și dezavantaj).

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

20

20

Support Vector Machines

☞ Datele de intrare sunt împărțite în două clase prin optimizarea ecuației unui hiperplan astfel încât distanța la date este maximă;

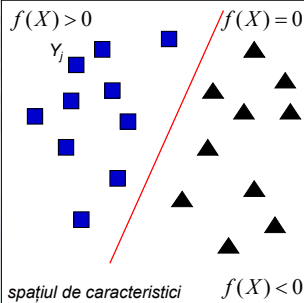
> date de intrare:

- date de antrenare:
 $[Y_1 \in C_1, \dots, Y_m \in C_m]$,
 $C_j \in \{+1, -1\}$, $Y_j = [Y_{j,1}, \dots, Y_{j,n}]$;

- date de clasificat:
 $X = [X_{i,1}, \dots, X_{i,n}]$, $i = 1, \dots, n$;

- *clasificator liniar*: se determină o funcție liniară:

$$f(X) = \begin{cases} > 0 & \text{clasa}(+1) \\ < 0 & \text{clasa}(-1) \end{cases}$$



Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

21

21

Support Vector Machines (cont.)

- *clasificator liniar*: se determină o funcție liniară (cont.):

$$f(X) = \begin{cases} > 0 & \text{clasa}(+1) \\ < 0 & \text{clasa}(-1) \end{cases}$$

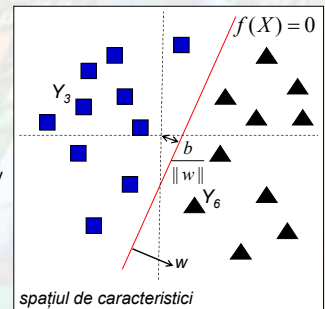
- funcția reprezintă ecuația unui hiperplan:

$$f(X) = w^T \cdot X + b$$

unde w și b reprezintă vectorul normal la hiperplan și respectiv decalajul față de origine;

$$f(Y_3) = w^T \cdot Y_3 + b > 0$$

$$f(Y_6) = w^T \cdot Y_6 + b < 0$$



Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

22

22

Support Vector Machines (cont.)

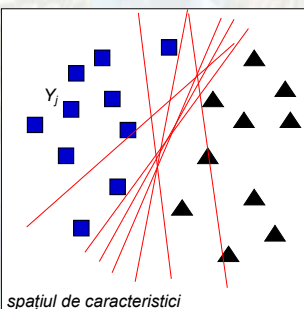
- *formulare clasificator*: având la dispoziție datele de antrenare, să se determine parametrii w și b ai hiperplanului care separă cel mai bine datele;

- clasificarea datelor noi se face prin:

$$f'(X_i) = \text{sgn}(w^T \cdot X_i + b)$$

$$\text{sgn}(x) = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \end{cases}$$

- cum determinăm hiperplanul care separă cel mai bine datele?



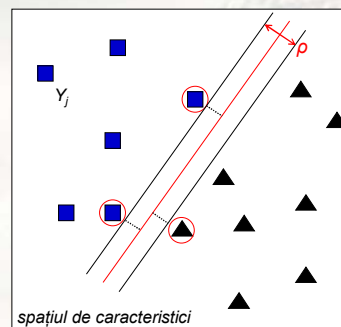
Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

23

23

Support Vector Machines (cont.)

- alegere hiperplan;



- datele de antrenare cele mai apropiate de hiperplan se numesc *vectori suport*;

- distanța dintre vectorii suport definesc o margine p ;

- *soluție*: separarea datelor se face cu hiperplanul ce maximizează pe p ;

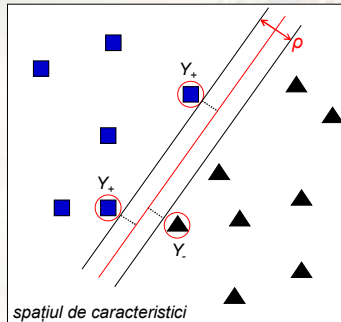
Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

24

24

Support Vector Machines (cont.)

- alegere hiperplan (cont.);



- dacă:

$$w^T \cdot X + b = 0$$

atunci și:

$$c(w^T \cdot X + b) = 0$$

astfel, putem normaliza valorile astfel încât:

$$w^T \cdot Y_+ + b = +1$$

$$w^T \cdot Y_- + b = -1$$

unde Y_+ și respectiv Y_- sunt vectorii suport din planul + și respectiv -;

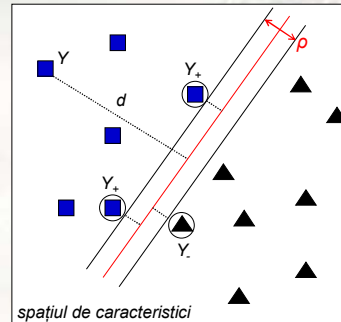
Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

25

25

Support Vector Machines (cont.)

- alegere hiperplan (cont.);



- cum determinăm valoarea lui ρ ?

$$\rho = \frac{w^T \cdot Y_+ + b}{\|w\|} - \frac{w^T \cdot Y_- + b}{\|w\|} = \frac{2}{\|w\|}$$

- care este distanța de la un vector Y la hiperplan?

$$d = \frac{w^T \cdot Y + b}{\|w\|}$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

26

26

Support Vector Machines (cont.)

- clasificatorul rezultat (formulare matematică):

- având m date de antrenare, $\{(Y_j, c_j)\}$, cu $j=1, \dots, m$, $Y_j = [y_{j,1}, \dots, y_{j,n}]$ și $c_j \in \{-1, +1\}$, acestea sunt separate de un hiperplan de margine ρ ;

- pentru fiecare set $\{(Y_j, c_j)\}$, avem:

$$w^T \cdot Y_j + b \leq -\frac{\rho}{2} \quad \text{dacă } c_j = -1$$

$$w^T \cdot Y_j + b \geq \frac{\rho}{2} \quad \text{dacă } c_j = +1$$

$$\Rightarrow c_j (w^T \cdot Y_j + b) \geq \frac{\rho}{2}$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

27

27

Support Vector Machines (cont.)

- clasificatorul rezultat (formulare matematică; cont.):

- pentru fiecare vector suport, Y_j^s , inegalitate devine egalitate:

$$c_j (w^T \cdot Y_j^s + b) = \frac{\rho}{2}$$

- astfel, distanța de la Y_j^s la hiperplan devine:

$$d = \frac{c_j (w^T \cdot Y_j^s + b)}{\|w\|} = \frac{1}{\|w\|}$$

- deci marginea ρ este:

$$\rho = \frac{2}{\|w\|}$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

28

28

Support Vector Machines (cont.)

- clasificatorul rezultat (formulare matematică; cont.):

- în aceste condiții antrenarea clasificatorului poate fi formulată ca:

să se determine w și b astfel încât să fie **maximizat** ρ , cu condiția că pentru toate datele de antrenare $\{(Y_j, c_j)\}$: $c_j (w^T \cdot Y_j + b) \geq 1$

normalizare la $\rho/2$

- și mai departe reformulată ca (minimizare):

să se determine w și b astfel încât să fie **minimizat** $\|w\|^2 = w^T w$, cu condiția că pentru toate $\{(Y_j, c_j)\}$: $c_j (w^T \cdot Y_j + b) \geq 1$

normalizare la $\rho/2$

= o problemă de optimizare pătratică (bine studiată în literatură);

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

29

29

Support Vector Machines (cont.)

- clasificatorul rezultat (formulare matematică; cont.):

să se determine w și b astfel încât să fie **minimizat** $\|w\|^2 = w^T w$, cu condiția că pentru toate $\{(Y_j, c_j)\}$: $c_j (w^T \cdot Y_j + b) \geq 1$

normalizare la $\rho/2$

- soluție folosind multiplicatorii Lagrange:

să se determine $\alpha_1, \dots, \alpha_m$ astfel încât să maximizăm:

$$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j c_i c_j Y_i^T Y_j$$

cu următoarele ipoteze:

$$(1) \sum_j \alpha_j c_j = 0$$

$$(2) \alpha_i \geq 0 \text{ pentru } \forall \alpha_i$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

30

30

Support Vector Machines (cont.)

- clasificatorul rezultat (formulare matematică; cont.):

să se determine $\alpha_1, \dots, \alpha_m$ astfel încât să maximizăm:

$$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j c_i c_j Y_i^T Y_j$$

cu următoarele ipoteze:

(1) $\sum_j \alpha_j c_j = 0$

(2) $\alpha_i \geq 0$ pentru $\forall \alpha_i$

$$\Rightarrow w = \sum_j \alpha_j c_j Y_j$$

$$\Rightarrow b = c_k - \sum_j \alpha_j c_j Y_j^T Y_k \quad \forall \alpha_k > 0$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

31

31

Support Vector Machines (cont.)

- SVM liniar "hard margin":

$$w = \sum_j \alpha_j c_j Y_j$$

$$b = c_k - \sum_j \alpha_j c_j Y_j^T Y_k \quad \forall \alpha_k > 0$$

- fiecare valoare α non-nulă indică un vector suport;

- clasificatorul este dat de:

$$f_{linSVM}(X_i) = \sum_j \alpha_j c_j Y_j^T X_i + b$$

unde $X_i = [x_{i,1}, \dots, x_{i,n}]$, $i=1, \dots, n$, sunt datele de clasificat;

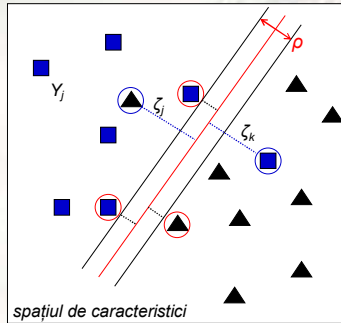
Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

32

32

Support Vector Machines (cont.)

- SVM "soft margin":



- ce putem face în această situație?

- o variantă este aceea de a adăuga un set de variabile (soft) care să-mi permită să reduc clasificările greșite, dificile sau afectate de zgomot
= "soft margin" SVM;

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

33

33

Support Vector Machines (cont.)

- SVM "soft margin" (cont.):

- formulare "hard margin":

să se determine w și b astfel încât să fie **minimizat** $\|w\|^2 = w^T w$, cu condiția că pentru toate $\{(Y_j, c_j)\}$: $c_j (w^T \cdot Y_j + b) \geq 1$

normalizare la $p/2$

- formulare "soft margin":

să se determine w și b astfel încât să fie **minimizat** $w^T w + C \sum_j \zeta_j$ cu condiția că pentru toate $\{(Y_j, c_j)\}$: $c_j (w^T \cdot Y_j + b) \geq 1 - \zeta_j$, $\zeta_j' \geq 0$

normalizare la $p/2$

- parametrul C poate fi văzut ca o modalitate de a controla adaptarea excesivă la datele de antrenare ("overfitting"): compromis între maximizare margine și adaptare la date.

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

34

34

Support Vector Machines (cont.)

- SVM "soft margin" (cont.):

- soluție folosind multiplicatorii Lagrange (aceeași formulare):

să se determine $\alpha_1, \dots, \alpha_m$ astfel încât să maximizăm:

$$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j c_i c_j Y_i^T Y_j$$

cu următoarele ipoteze:

(1) $\sum_j \alpha_j c_j = 0$

(2) $\alpha_i \geq 0$ pentru $\forall \alpha_i$

$$\Rightarrow w = \sum_j \alpha_j c_j Y_j$$

$$\Rightarrow b = c_k (1 - \zeta_k) - \sum_j \alpha_j c_j Y_j^T Y_k \quad \forall \alpha_k > 0$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

35

35

Support Vector Machines (cont.)

- SVM "soft margin" (cont.):

$$w = \sum_j \alpha_j c_j Y_j$$

$$b = c_k (1 - \zeta_k) - \sum_j \alpha_j c_j Y_j^T Y_k \quad \forall \alpha_k > 0$$

- fiecare valoare α non-nulă indică un vector suport;

- clasificatorul este dat de:

$$f_{softSVM}(X_i) = \sum_j \alpha_j c_j Y_j^T X_i + b$$

unde $X_i = [x_{i,1}, \dots, x_{i,n}]$, $i=1, \dots, n$, sunt datele de clasificat;

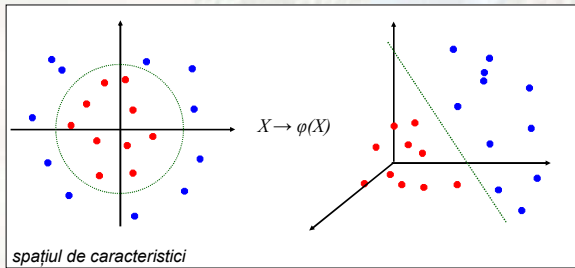
Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

36

36

Support Vector Machines (cont.)

- ce se întâmplă dacă datele nu sunt liniar separabile?
[transformare a spațiului astfel încât separabilitatea să fie posibilă]



[sursă Machine Learning Group, Univ. of Texas]

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

37

37

Support Vector Machines (cont.)

- abordare neliniară;

$$f_{linSVM}(X_i) = \sum_j \alpha_j c_j (Y_j^T X_i) + b$$

unde $X_i = [x_{i,1}, \dots, x_{i,n}]$, $i=1, \dots, n$, sunt datele de clasificat;

$$b = c_k - \sum_j \alpha_j c_j (Y_j^T Y_k)$$

- "kernel trick" (vezi și M3, k-means) - produsele $X^T X$ sunt transformate printr-o funcție neliniară:

$$K(Y_j, Y_k) = Y_j^T Y_k \quad \Rightarrow \quad K(Y_j, Y_k) = \varphi(Y_j)^T \varphi(Y_k)$$

[liniar] [neliniar]

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

38

38

Support Vector Machines (cont.)

- abordare neliniară: "kernel trick" (cont.);
- funcțiile nucleu trebuie să fie semi-pozitiv definite și simetrice;
- exemple de funcții uzuale folosite pentru SVM:

$$K(Y_j, Y_k) = Y_j^T Y_k \quad \text{liniar}$$

$$K(Y_j, Y_k) = (1 + Y_j^T Y_k)^p \quad \text{polinomial}$$

$$K(Y_j, Y_k) = e^{-\frac{\|Y_j - Y_k\|^2}{2\sigma^2}} \quad \text{Gaussiană (Radial Basis Function)}$$

$$K(Y_j, Y_k) = 1 - 2 \sum_{i=1}^n \frac{(y_{j,i} - y_{k,i})^2}{(y_{j,i} + y_{k,i})} \quad \text{Chi-Square}$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

39

39

Support Vector Machines (cont.)

- abordare neliniară: "kernel trick" (cont.);

să se determine $\alpha_1, \dots, \alpha_m$ astfel încât să maximizăm:

$$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j c_i c_j K(Y_i, Y_j)$$

cu următoarele ipoteze:

$$(1) \sum_j \alpha_j c_j = 0$$

$$(2) \alpha_i \geq 0 \text{ pentru } \forall \alpha_i$$

- clasificatorul este dat de:

$$f_{nelinSVM}(X_i) = \sum_j \alpha_j c_j K(Y_j, X_i) + b$$

unde $X_i = [x_{i,1}, \dots, x_{i,n}]$, $i=1, \dots, n$, sunt datele de clasificat;

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

40

40

Support Vector Machines (cont.)

- ce se întâmplă dacă datele de clasificat sunt multi-clasă?
(SVM este nativ un clasificator binar)
[formăm un clasificator multiclasă]

> date de intrare:

- date de antrenare:

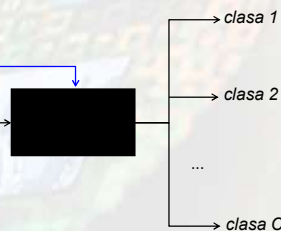
$$\{(Y_j, c_j)\}, j=1, \dots, m,$$

$$Y_j = [y_{j,1}, \dots, y_{j,n}],$$

$$c_j \in \{1, \dots, C\};$$

- date de clasificat:

$$X_i = [x_{i,1}, \dots, x_{i,n}], i=1, \dots, n;$$



Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

41

41

Support Vector Machines (cont.)

- clasificare multi-clasă (cont.):

abordare 1 - one-vs.-all

- sunt creați C clasificatori SVM (câte unul pentru fiecare clasă);

- clasificatorii sunt antrenați cu datele de antrenare modificate ca, exemplu SVM₂:

$$\{(Y_j + 1)\} \text{ dacă } c_j = 2$$

$$\{(Y_j - 1)\} \text{ altfel}$$

cum luăm decizia de clasificare?
 $\arg \max_{c=1, \dots, C} \{f_{SVM_c}(X_i)\}$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

42

42

Support Vector Machines (cont.)

- clasificare multi-clasă (cont.):

abordare 2 - **one-vs.-one**

- sunt creați C' clasificatori SVM, câte unul pentru fiecare combinație de două clase = $\frac{(C-1)C}{2}$ x SVM;

- clasificatorii sunt antrenați doar cu datele de antrenare aferente celor două clase;

cum luăm decizia de clasificare? vot majoritar

$\{(Y_p, 1)\} \rightarrow \{(Y_p, +1)\}$
 $\{(Y_p, 2)\} \rightarrow \{(Y_p, -1)\}$

$X_i \rightarrow \text{SVM}_{1,2} \rightarrow \begin{cases} \text{clasa } +1 \text{ (real } 1) \\ \text{sau} \\ \text{clasa } -1 \text{ (real } 2) \end{cases}$ +1 vot

$X_i \rightarrow \text{SVM}_{1,3} \rightarrow \begin{cases} \text{clasa } +1 \text{ (real } 1) \\ \text{sau} \\ \text{clasa } -1 \text{ (real } 3) \end{cases}$ +1 vot

$\dots$

$X_i \rightarrow \text{SVM}_{(C-1),C} \rightarrow \begin{cases} \text{clasa } +1 \text{ (real } C-1) \\ \text{sau} \\ \text{clasa } -1 \text{ (real } C) \end{cases}$ etc

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU 43

43

Arbori de decizie (Decision Trees)

Datele sunt clasificate prin asocierea observațiilor (date de antrenare) cu o serie de concluzii privind valorile acestora (predicție), ceea ce conduce la o reprezentare arborescentă;

- punerea problemei, un exemplu simplu:

- să presupunem că avem posibilitatea să realizăm patru activități de weekend:
 {"cumpărături", "film", "tenis", "nimic"}
 (reprezintă clasele);

- aceste activități depind de o serie de variabile:
 "vreme" $\in$ {"vânt", "ploaie", "soare"}
 "buget" $\in$ {"bogat", "sărac"}
 "vizită părinți" $\in$ {"da", "nu"}
 (reprezintă atributele datelor);

[Simon Colton, lectures, 2004]

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU 44

44

Arbori de decizie (Decision Trees; cont.)

- punerea problemei, un exemplu simplu (cont.):

- pe baza cunoștințelor actuale putem asocia valorile atributelor unor decizii de activități (clase): ~ **etapă de antrenare**;

nodurile arborelui reprezintă **atributele**;

ramurile arborelui reprezintă **relaționarea valorilor atributelor**;

frunzele arborelui reprezintă **clasele**;

[Simon Colton, lectures, 2004]

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU 45

45

Arbori de decizie (Decision Trees; cont.)

- punerea problemei, un exemplu simplu (cont.):

- pentru o serie de valori noi ale atributelor, folosind arborele creat putem lua o decizie: ~ **etapă de clasificare**;

Sâmbătă 16 Mai:
 "vizită părinți" = "nu"
 "vreme" = "soare"

Duminică 17 Mai:
 "vizită părinți" = "nu"
 "vreme" = "vânt"
 "buget" = "sărac"

[Simon Colton, lectures, 2004]

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU 46

46

Arbori de decizie (Decision Trees; cont.)

- antrenarea arborelui (metoda ID3 - Iterative Dichotomiser 3);

- cum selectăm atributele care să fie asociate nodurilor și cum alegem ordinea (prioritatea) acestora?

• **entropie**: având la dispoziție un sistem binar de clasificare și un set de exemple S în care p_+ % exemple sunt clasificate în clasa1 (pozitive) și respectiv p_- % în clasa2 (negative) atunci:

$$\text{entropy}(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

> este o măsură a "purității" datelor pentru o colecție de exemple (puritate = datele sunt fie toate în clasă sau clasa este goală);

$p \rightarrow 0 \Rightarrow \log_2(p)$ mare, negativ; $p \log_2(p) \approx 0$

$p \rightarrow 1 \Rightarrow \log_2(p)$ mic, ≈ 0 ; $p \log_2(p) \approx 0$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU 47

47

Arbori de decizie (Decision Trees; cont.)

- antrenarea arborelui (metoda ID3; cont.);

• **entropie**: în cazul a C clase, unde setul de exemple S are p_i % exemple clasificate în clasa c_i , atunci:

$$\text{entropy}(S) = \sum_{i=1}^C -p_i \log_2(p_i)$$

• **câștig informațional** ("information gain"): pentru un atribut A , cu mulțimea de valori posibile $\{A\}$, notând cu S_a subsetul de exemple din S în care atributul A are valoarea a , atunci:

$$\text{gain}(S, A) = \text{entropy}(S) - \sum_{a \in \{A\}} \frac{|S_a|}{|S|} \text{entropy}(S_a)$$

unde operatorul $|\cdot|$ returnează numărul de elemente al unui set.

> este o măsură a reducerii entropiei datorată valorii lui A ;

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU 48

48

Arbori de decizie (Decision Trees; cont.)

- antrenarea arborelui (metoda ID3; cont.);

> date de intrare:

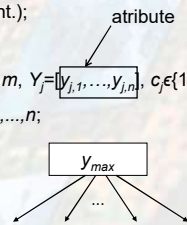
- date de antrenare $S: \{(Y_j, c_j)\}, j=1, \dots, m, Y_j = [y_{j,1}, \dots, y_{j,n}], c_j \in \{1, \dots, C\}$;
- date de clasificat: $X_i = [x_{i,1}, \dots, x_{i,n}], i=1, \dots, n$;

> algoritm:

> antrenare:

p1. atributul $y=y_{max}$ pentru care avem valoarea maximă a information gain, $gain(S, y)$, relativ la S este ales drept rădăcină;

p2. pentru fiecare valoare posibilă a lui y_{max} (din mulțimea $\{y_{max}\}$) creăm o ramură;



Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

49

49

Arbori de decizie (Decision Trees; cont.)

- antrenarea arborelui (metoda ID3; cont.);

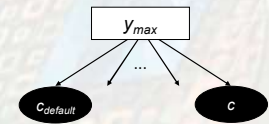
> algoritm (cont.):

> antrenare (cont.):

p3. pentru fiecare ramură calculăm S_v (cu v valoarea asociată ramurii);

p4. dacă S_v este mulțimea vidă atunci determinăm clasa $C_{default}$ care are cele mai multe exemple în setul de antrenare S ; aceasta definește frunza ce închide această ramură;

p5. dacă S_v conține doar date dintr-o clasă c atunci definim cu această clasă frunza ce închide ramura;



Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

50

50

Arbori de decizie (Decision Trees; cont.)

- antrenarea arborelui (metoda ID3; cont.);

> algoritm (cont.):

> antrenare (cont.):

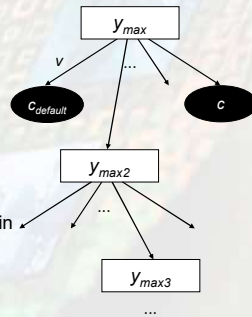
p6. dacă S_v nu este conform p4 și p5 atunci:

- y_{max} este eliminat din lista potențialelor atribute care definesc noduri;

- determinăm atributul $y=y_{max2}$ pentru care avem information gain maxim relativ la S_v , $gain(S_v, y)$;

- acesta crează un nod nou;

- se repetă algoritmul cu pasul 2.



Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

51

51

Arbori de decizie (Decision Trees; cont.)

- antrenarea arborelui (metoda ID3; cont.);

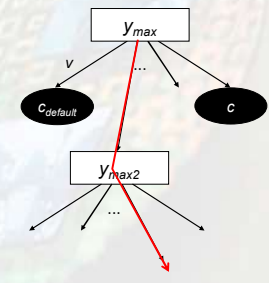
> algoritm (cont.):

> clasificare:

- pentru un vector X de intrare nu trebuie decât să parcurg arborele în funcție de valorile atributelor;

- clasa de apartenență a lui X este dată de frunza arborelui la care ajung în final;

- procesul de clasificare este imediat.



Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

52

52

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric:

- date de intrare (setul S):

nr.	vreme	vizită părinți	buget	decizie
#1	soare	da	bogat	film
#2	soare	nu	bogat	tenis
#3	vânt	da	bogat	film
#4	ploaie	da	sărac	film
#5	ploaie	nu	bogat	nimic
#6	ploaie	da	sărac	film
#7	vânt	nu	sărac	film
#8	vânt	nu	bogat	cumpărături
#9	vânt	da	bogat	film
#10	soare	nu	bogat	tenis

[Simon Colton, lectures, 2004]

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

53

53

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric (cont.):

- p1: alegere nod rădăcină:

$$entropy(S) = \sum_{i=1}^C -p_i \log_2(p_i) \quad C \in \{\text{film, tenis, cumpărături, nimic}\}$$

$$entropy(S) = -p_{\text{film}} \log_2(p_{\text{film}}) - p_{\text{tenis}} \log_2(p_{\text{tenis}}) - p_{\text{cumpărături}} \log_2(p_{\text{cumpărături}}) - p_{\text{nimic}} \log_2(p_{\text{nimic}})$$

$$= -\frac{6}{10} \log_2\left(\frac{6}{10}\right) - \frac{2}{10} \log_2\left(\frac{2}{10}\right) - \frac{1}{10} \log_2\left(\frac{1}{10}\right)$$

$$- \frac{1}{10} \log_2\left(\frac{1}{10}\right) = 1.571$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

54

54

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric (cont.):

$A \in \{\text{vreme, vizită părinți, buget}\}$

- p1: alegere nod rădăcină (cont.):

$$\text{gain}(S, A) = \text{entropy}(S) - \sum_{a \in \{A\}} \frac{|S_a|}{|S|} \text{entropy}(S_a)$$

$$\text{gain}(S, \text{vreme}) = 1.571 - \frac{3}{10} \text{entropy}(S_{\text{soare}}) - \frac{4}{10} \text{entropy}(S_{\text{vânt}}) - \frac{3}{10} \text{entropy}(S_{\text{ploaie}})$$

$$\text{entropy}(S_{\text{ploaie}}) = -\frac{2}{3} \log_2\left(\frac{2}{3}\right) - 0 - 0 - \frac{1}{3} \log_2\left(\frac{1}{3}\right) = 0.918$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

55

55

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric (cont.):

- p1: alegere nod rădăcină (cont.):

$$\text{gain}(S, \text{vreme}) = 1.571 - \frac{3}{10} \text{entropy}(S_{\text{soare}}) - \frac{4}{10} \text{entropy}(S_{\text{vânt}}) - \frac{3}{10} \text{entropy}(S_{\text{ploaie}})$$

$$\text{entropy}(S_{\text{ploaie}}) = 0.918$$

$$\text{entropy}(S_{\text{vânt}}) = 0.811 \Rightarrow \text{gain}(S, \text{vreme}) = 0.7$$

$$\text{entropy}(S_{\text{soare}}) = 0.918$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

56

56

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric (cont.):

- p1: alegere nod rădăcină (cont.):

$$\text{gain}(S, \text{vizita_parinti}) = 1.571 - \frac{5}{10} \text{entropy}(S_{\text{da}}) - \frac{5}{10} \text{entropy}(S_{\text{nu}})$$

$$\text{entropy}(S_{\text{da}}) = 0$$

$$\text{entropy}(S_{\text{nu}}) = 1.922$$

$$\Rightarrow \text{gain}(S, \text{vizita_parinti}) = 0.61$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

57

57

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric (cont.):

- p1: alegere nod rădăcină (cont.):

$$\text{gain}(S, \text{buget}) = 1.571 - \frac{7}{10} \text{entropy}(S_{\text{bogat}}) - \frac{3}{10} \text{entropy}(S_{\text{sarac}})$$

$$\text{entropy}(S_{\text{bogat}}) = 1.842$$

$$\text{entropy}(S_{\text{sarac}}) = 0$$

$$\Rightarrow \text{gain}(S, \text{buget}) = 0.2816$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

58

58

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric (cont.):

- p1: alegere nod rădăcină (cont.):

$$\text{gain}(S, \text{vreme}) = 0.7$$

$$\text{gain}(S, \text{vizita_parinti}) = 0.61$$

$$\text{gain}(S, \text{buget}) = 0.2816$$

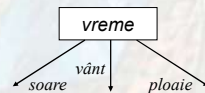
- p2: creăm ramurile pentru nodul vreme;

- p3: calculăm:

$$S_{\text{ploaie}} = \{\#4, \#5, \#6\}, |S_{\text{ploaie}}| = 3;$$

$$S_{\text{soare}} = \{\#1, \#2, \#10\}, |S_{\text{soare}}| = 3;$$

$$S_{\text{vânt}} = \{\#3, \#7, \#8, \#9\}, |S_{\text{vânt}}| = 4.$$



Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

59

59

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric (cont.):

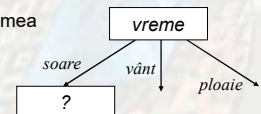
- p4: $S_{\text{ploaie}}, S_{\text{soare}}, S_{\text{vânt}}$ sunt mulțimea vidă? nu;

- p5: $S_{\text{ploaie}}, S_{\text{soare}}, S_{\text{vânt}}$ conțin date doar dintr-o clasă? nu;

- p6: determinăm atributul (altul decât vreme) pentru care obținem informația gain maxim:

$$\text{entropy}(S_{\text{soare}}) = -p_{\text{film}} \log_2(p_{\text{film}}) - p_{\text{tenis}} \log_2(p_{\text{tenis}}) - p_{\text{cumparatur_i}} \log_2(p_{\text{cumparatur_i}}) - p_{\text{nimic}} \log_2(p_{\text{nimic}})$$

- valorile sunt calculate pe S_{soare} doar;



Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

60

60

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric (cont.):

- p6: determinăm atributul pentru care obținem information gain maxim (cont.):

$$\begin{aligned} \text{entropy}(S_{\text{soare}}) &= \\ &= -p_{\text{film}} \log_2(p_{\text{film}}) - p_{\text{tenis}} \log_2(p_{\text{tenis}}) - \\ &= -p_{\text{cumparatur i}} \log_2(p_{\text{cumparatur i}}) - p_{\text{nimic}} \log_2(p_{\text{nimic}}) \\ \text{entropy}(S_{\text{soare}}) &= -\frac{1}{3} \log_2\left(\frac{1}{3}\right) - \frac{2}{3} \log_2\left(\frac{2}{3}\right) - 0 - 0 \\ &= 0.918 \end{aligned}$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

61

61

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric (cont.):

- p6: determinăm atributul pentru care obținem information gain maxim (cont.):

$$\begin{aligned} \text{gain}(S_{\text{soare}}, \text{vizita parinti}) &= \\ &= 0.918 - \frac{1}{3} \text{entropy}(S_{\text{da}}) - \frac{2}{3} \text{entropy}(S_{\text{nu}}) \\ \text{entropy}(S_{\text{da}}) &= 0 \\ \text{entropy}(S_{\text{nu}}) &= 0 \end{aligned} \Rightarrow \text{gain}(S_{\text{soare}}, \text{vizita parinti}) = 0.918$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

62

62

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric (cont.):

- p6: determinăm atributul pentru care obținem information gain maxim (cont.):

$$\begin{aligned} \text{gain}(S_{\text{soare}}, \text{buget}) &= \\ &= 0.918 - \frac{3}{3} \text{entropy}(S_{\text{bogat}}) - \frac{0}{3} \text{entropy}(S_{\text{sarac}}) \\ \text{entropy}(S_{\text{bogat}}) &= 0.918 \\ \text{entropy}(S_{\text{sarac}}) &= 0 \end{aligned} \Rightarrow \text{gain}(S_{\text{soare}}, \text{buget}) = 0$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

63

63

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric (cont.):

- p6: determinăm atributul pentru care obținem information gain maxim (cont.):

$$\begin{aligned} \text{gain}(S_{\text{soare}}, \text{vizita parinti}) &= 0.918 \\ \text{gain}(S_{\text{soare}}, \text{buget}) &= 0 \end{aligned}$$

- p2: creăm ramurile pentru nodul vizită părinți;

- p3: calculăm:

$$\begin{aligned} S_{\text{da} \cap \text{soare}} &= \{\#1\}, |S_{\text{da}}| = 1; \\ S_{\text{nu} \cap \text{soare}} &= \{\#2, \#10\}, |S_{\text{nu}}| = 2. \end{aligned}$$

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

64

64

Arbori de decizie (Decision Trees; cont.)

> exemplu numeric (cont.):

- p4: $S_{\text{da}}, S_{\text{nu}}$ sunt mulțimea vidă? nu;
- p5: $S_{\text{da}}, S_{\text{nu}}$ conțin date doar dintr-o clasă?
 - da, S_{da} este în categoria film;
 - da, S_{nu} este în categoria tenis;
- ... pentru valorile vânt și ploaie se procedează similar ...



Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

65

65

> Sfârșit M4

Tehnici de analiză și clasificare automată a informației, Prof. Bogdan IONESCU

66

66