

LAPI – Laboratorul de
Analiza și Prelucrarea
Imaginilor



Universitatea
POLITEHNICA din
București



Facultatea de Electronică,
Telecomunicații și
Tehnologia Informației

TACAI - Tehnici de Analiză și Clasificare Automată a Informației

Note de laborator

Dr.ing. Ionuț Mironică

Conf.dr.ing. Bogdan Ionescu

Laborator 3

Cuprins:

- Fuzionarea descriptorilor
- Analiza componentelor principale (PCA)
- Exerciții

I. Fuzionarea descriptorilor

Introducere

- În cele mai multe dintre cazuri, pentru reprezentarea conținutului multimedia este necesară combinarea mai multor tipuri de descriptori.
- De exemplu, conținutul unui document video poate fi reprezentat atât pe baza structurii temporale, cât și folosind descriptori de mișcare, descriptori audio.
- Strategiile de fuziune a datelor se bazează pe ipoteza conform căreia o decizie obținută pe baza mai multor descriptori poate oferi performanțe superioare unei decizii bazate pe un singur tip de descriptor.

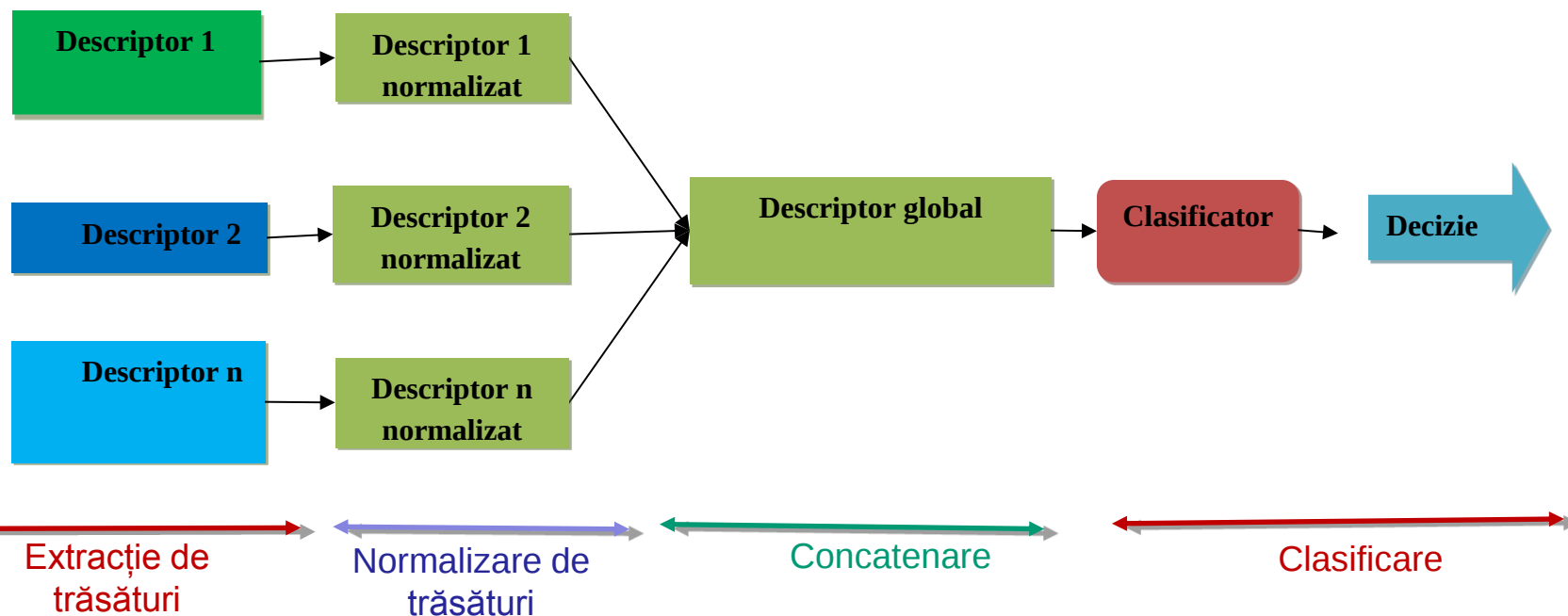
I. Fuzionarea descriptorilor

Introducere

- Tehnicile de tip "early fusion" realizează agregarea datelor „timpuriu” în lanțul de prelucrare, înainte de a fi folosite la indexare sau în alte procese de analiză.
- Fuziunea datelor are loc în spațiul de caracteristici (și constă practic în concatenarea propriu-zisă a tuturor descriptorilor fără a ține cont de redundanța acestora.

I. Fuzionarea descriptorilor

Schemă fuzionare (early fusion)



I. Fuzionarea descriptorilor

Normalizarea descriptorilor

- De exemplu, dacă obiectul multimedia X este descris de descriptorii de conținut $desc1 = \{a1, a2, \dots, an\}$, $desc2 = \{b1, b2, \dots, bm\}$ și respectiv $desc3 = \{c1, c2, \dots, cl\}$, descriptorul agregat este dat de concatenarea valorilor și anume $desc-f = \{a1, \dots, an, b1, \dots, bm, c1, \dots, cl\}$.
- Descriptori diferiți tind să aibă intervale de variație diferite ale valorilor, de la normalizări diferite, de exemplu valori între $[0; 1]$ sau $[a; b]$ (unde a și b sunt două valori cunoscute) până la intervale de valori variabile și care depind de tipul datelor.

I. Fuzionarea descriptorilor

Normalizarea min-max

Fie $desc1 = \{a11, a12, ..., a1n\}$ descriptorul unui obiect multimedia.
Descriptorul se va normaliza cu formula:

$$a_i = \frac{a_i - \min\{a_i\}}{\max\{a_i\} - \min\{a_i\}}$$

unde $\min\{a_i\}$ și $\max\{a_i\}$ reprezintă operatorii ce returnează valoarea minimă și respectiv maximă a tuturor valorilor descriptorilor (pentru toate obiectele multimedia considerate) pentru atributul a_i .

Normalizarea min-max asigură o normalizare a valorilor în intervalul $[0; 1]$.

I. Fuzionarea descriptorilor

Normalizarea z-score

Fie $desc1 = \{a11, a12, ..., a1n\}$ descriptorul unui obiect multimedia. Descriptorul se va normaliza cu formula:

$$a_i = \frac{a_i - medie\{a_i\}}{\sigma\{a_i\}}$$

unde operatorii $medie\{a_i\}$ și $\sigma\{a_i\}$ returnează valoarea medie și respectiv abaterea pătratică medie a tuturor valorilor descriptorilor pentru atributul a_i .

Normalizarea z-score realizează o normalizare a valorilor pe o distribuție de medie zero și dispersie unu.

II. Analiza componentelor principale

Introducere

În algoritmi de recunoaștere a formelor, selecția caracteristicilor relevante poate reprezenta o alegere decisivă pentru proiectarea oricărui clasificator.

Selecția caracteristicilor poate fi văzută și ca un proces de „compresie de date”, fiind asimilată cu o transformare liniară din spațiul inițial al observațiilor într-un spațiu cu mai puține dimensiuni. O astfel de transformare este necesară deoarece poate conserva o mare parte din informație și permite aplicarea unor algoritmi în timp real, algoritmi eficienți fiind doar în spații cu dimensiuni reduse.

II. Analiza componentelor principale

Transformata Karhunen-Loève

Transformata Karhunen-Loève (Analiza Componentelor Principale - **PCA**) este o metodă liniară de selecție a caracteristicilor.

Schemă algoritm

- se calculează matricea de covariație a descriptorilor de intrare;
- se calculează valorile proprii și se ordonează descrescător:
 $\lambda_1 > \lambda_2 > \dots > \lambda_n$
- se calculează vectorii proprii asociați valorilor proprii: $[\Phi_1, \Phi_2, \dots, \Phi_n]$
- se construiește matricea K a vectorilor proprii: $K = [\Phi_1, \Phi_2, \dots, \Phi_n]$

II. Analiza componentelor principale

Transformata Karhunen-Loève

Schemă algoritm (continuare)

- numărul "m" de trăsături care se rețin după transformarea unitară se obține din valoarea maximă admisibilă a erorii medii pătratice de estimare.
- fiecărei descriptor de intrare X i se va aplica transformarea Karhunen-Loève:
$$Y = KX$$
- Se vor reține numai $m < n$ componente ale lui Y .

Pentru implementarea PCA se poate utiliza biblioteca Matlab Prtools (<http://prtools.org/>) sau Weka (secțiunea Preprocess).

II. Analiza componentelor principale

Weka - PCA

Choose

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter: Choose **None** Apply

Current relation:
Relation: pima_diabetes
Instances: 768
Attributes: 9

Attributes:
All None Invert Pattern

No.	Name
1	☒ preg
2	☐ plas
3	☐ pres
4	☐ skin
5	☐ insu
6	☐ mass
7	☐ pedi
8	☐ age
9	☐ class

Remove

Selected attribute:
Name: preg
Missing: 0 (0%)
Distinct: 17
Type: Numeric
Unique: 2 (0%)

Statistic	Value
Minimum	0
Maximum	17
Mean	3.845
StdDev	3.37

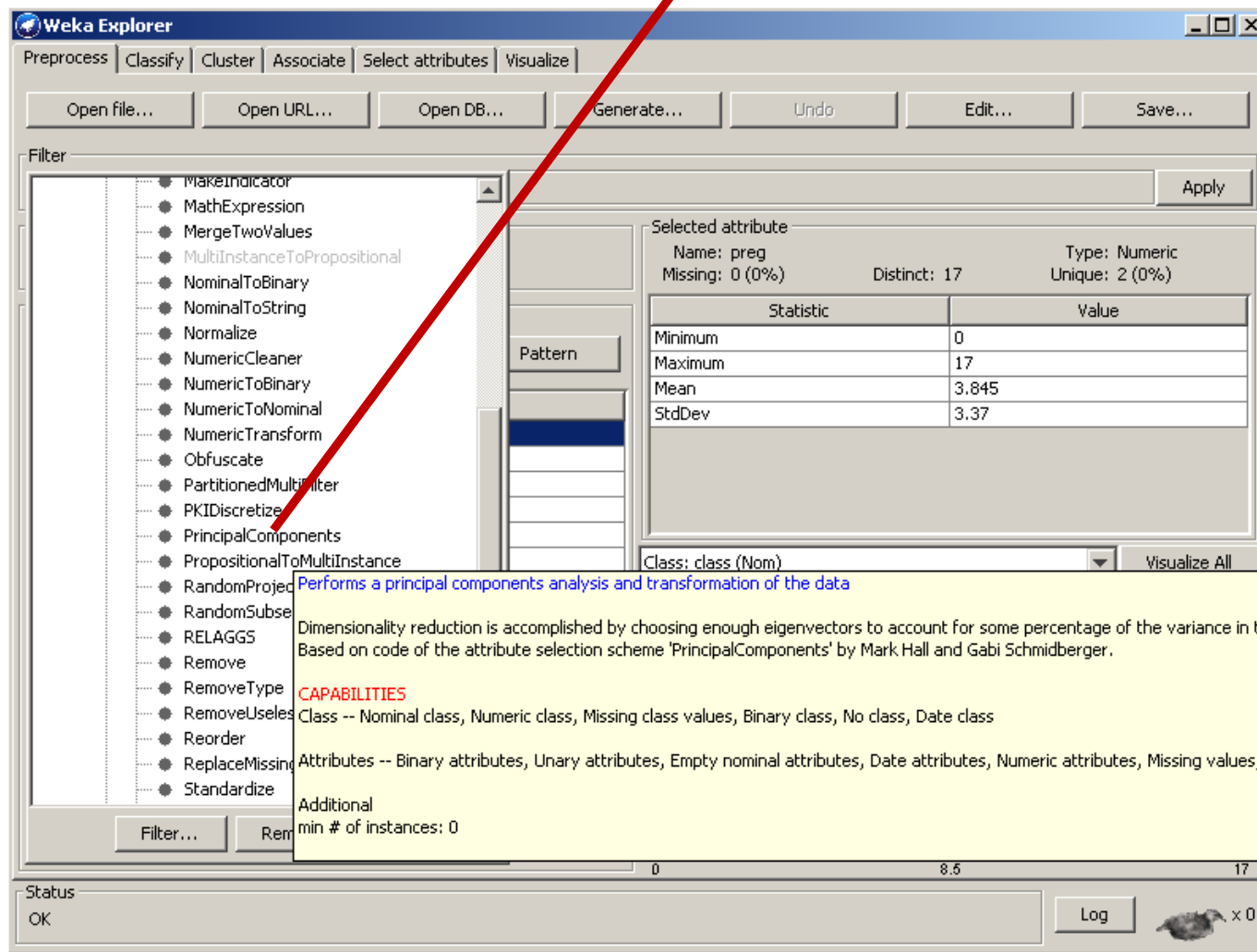
Class: class (Nom) Visualize All

Status: OK Log x 0

II. Analiza componentelor principale

Weka - PCA

PrincipalComponents



Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter

- MakeIndicator
- MathExpression
- MergeTwoValues
- MultiInstanceToPropositional
- NominalToBinary
- NominalToString
- Normalize
- NumericCleaner
- NumericToBinary
- NumericToNominal
- NumericTransform
- Obfuscate
- PartitionedMultiFilter
- PKIDiscretize
- PrincipalComponents**
- PropositionalToMultiInstance
- RandomProject
- RandomSubspace
- RELAGGS
- Remove
- RemoveType
- RemoveUseless
- Reorder
- ReplaceMissing
- Standardize

Selected attribute

Name: preg
Missing: 0 (0%)
Distinct: 17
Type: Numeric
Unique: 2 (0%)

Statistic	Value
Minimum	0
Maximum	17
Mean	3.845
StdDev	3.37

Class: class (Nom)

Visualize All

Performs a principal components analysis and transformation of the data

Dimensionality reduction is accomplished by choosing enough eigenvectors to account for some percentage of the variance in the original data -- default 0.95 (95%). Based on code of the attribute selection scheme 'PrincipalComponents' by Mark Hall and Gabi Schmidberger.

CAPABILITIES

Class -- Nominal class, Numeric class, Missing class values, Binary class, No class, Date class

Attributes -- Binary attributes, Unary attributes, Empty nominal attributes, Date attributes, Numeric attributes, Missing values, Nominal attributes

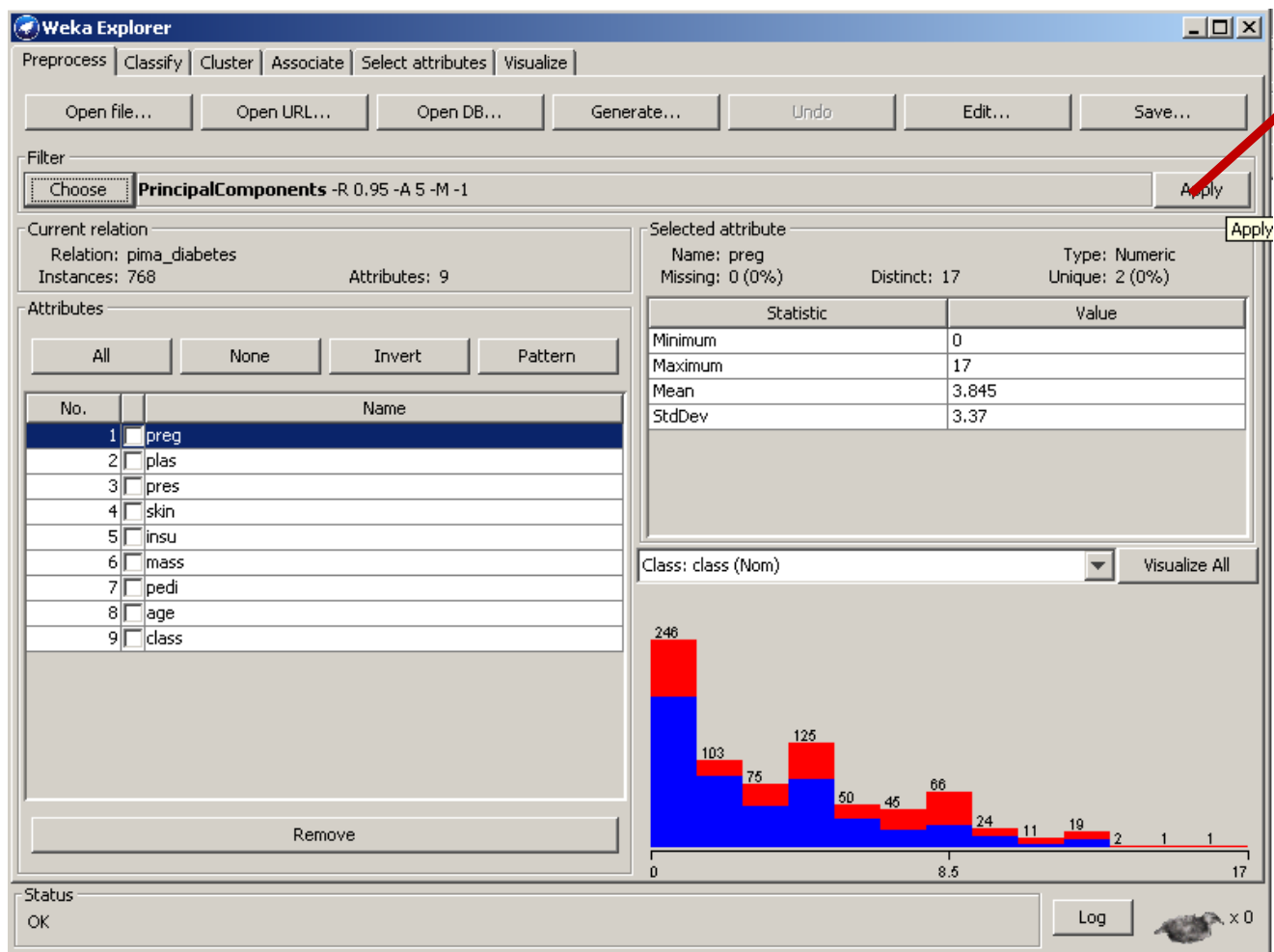
Additional
min # of instances: 0

Status
OK

Log

II. Analiza componentelor principale

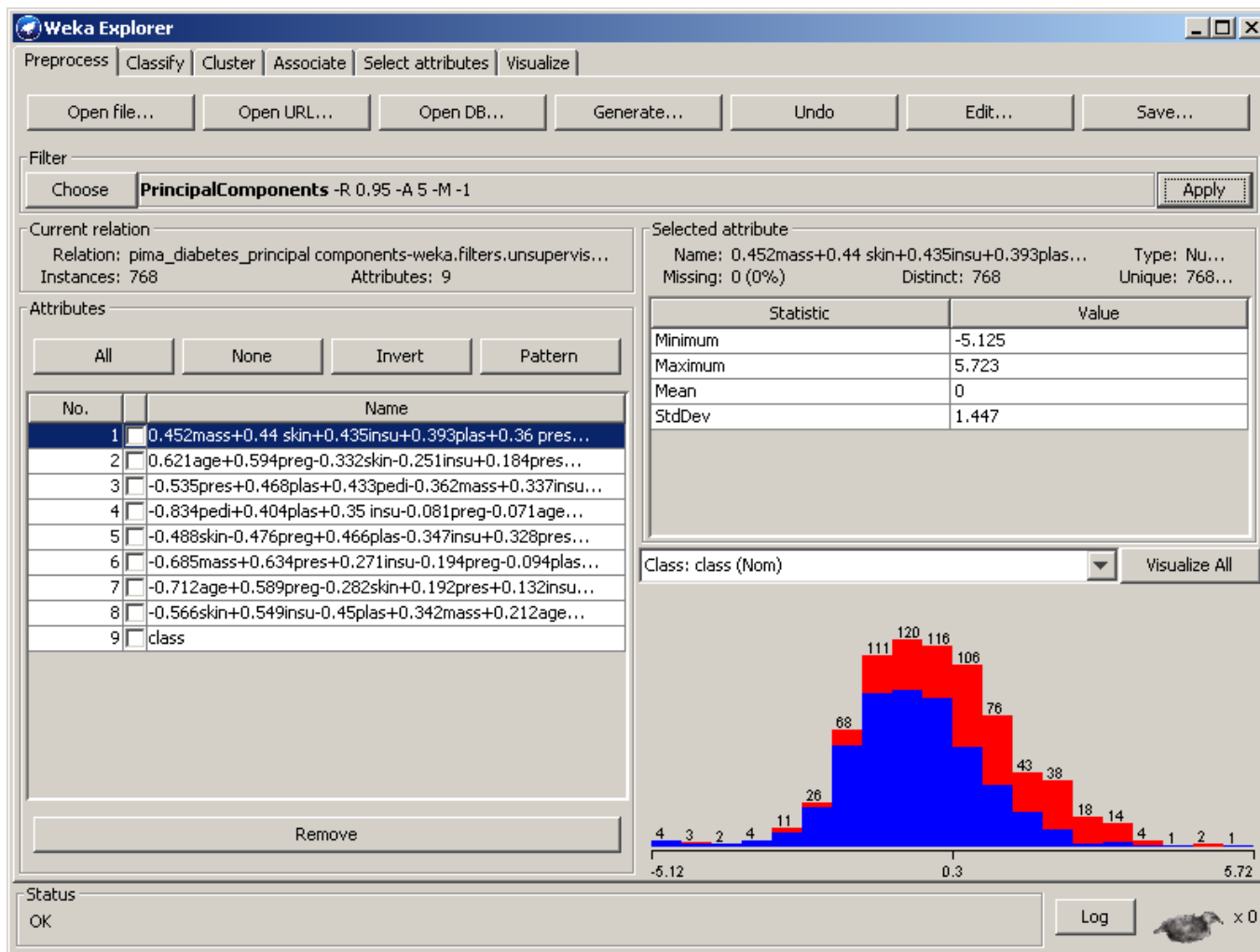
Weka - PCA



Apply

II. Analiza componentelor principale

Weka - PCA



III. Exerciții

Exercițiul 1

Pentru cele două baze de date (texturi și imagini naturale):

- Realizați un program în Matlab care calculează rând pe rând cei trei descriptori (Color Moments, LBP și HoG) și îi concatenează într-un singur descriptor.
- Apoi se încarcă în Weka și se testează rând pe rând performanța clasificatorilor ZeroR, OneR, arbori de decizie (J48), Nearest Neighbor (IBK) și Naive Bayes.
- Rețineți performanțele într-un fișier text.

III. Exerciții

Exercițiul 1 - continuare

■ Schemă algoritm Matlab:

(imagePaths, labels) = Citește fișier de configurare (train.txt)

descriptors %inițializare matrice pentru export în fișier arff

pentru fiecare imagine din bază (contor i)

- im = citeșteImagine(i)

- desc1 = CalculDescriptor1(im);

- desc2 = CalculDescriptor2(im);

- desc3 = CalculDescriptor3(im);

- desc = concat(desc1, desc2, desc3);

- Adaugă variabilei descriptors -» concat (desc, labels(i))

Salvează variabila descriptors în fișier arff

III. Exerciții

Exercițiul 2

Pentru cele două baze de date (texturi și imagini naturale):

- Realizați un program în Matlab care calculează rând pe rând cei trei descriptori (Color Moments, LBP și HoG), îi **normalizează** (folosind min-max sau z-score) și îi concatenează într-un singur descriptor.
- Apoi se încarcă în Weka și se testează rând pe rând performanța clasificatorilor ZeroR, OneR, arbori de decizie (J48), Nearest Neighbor (IBK) și Naive Bayes.
- Rețineți performanțele într-un fișier text.
- Comparați performanțele cu cele obținute în exercițiul anterior.

III. Exerciții

Exercițiul 2 - continuare

■ Schemă algoritm Matlab:

(imagePaths, labels) = Citește fișier de configurare (train.txt)

descriptors %inițializare matrice pentru export în fișier arff

pentru fiecare imagine din bază (contor i)

- im = citeșteImagine(i)

- desc1 = Normalizează(CalculDescriptor1(im));

- desc2 = Normalizează(CalculDescriptor2(im));

- desc3 = Normalizează(CalculDescriptor3(im));

- desc = concat(desc1, desc2, desc3);

- Adaugă variabilei descriptors -» concat (desc, labels(i))

Salvează variabila descriptors în fișier arff

III. Exerciții

Exercițiul 3

Pentru baza de date de imagini naturale:

- Realizați un program în Matlab care calculează descriptorul HoG.
- Încărcați fișierul în Weka și aplicați o transformare PCA (care să rețină rând pe rând 30/%, 60% și 90% din descriptor).
- Testați rând pe rând performanța clasificatorilor ZeroR, OneR, arbori de decizie (J48), Nearest Neighbor (IBK) și Naive Bayes.
- Rețineți performanțele într-un fișier text.
- Comparați performanțele cu cele obținute în cazul în care folosim doar descriptorul HoG singur.

IV. Exerciții

Exerciții suplimentare

- Să se facă salvarea în mod automat din Matlab a fișierului arff;
- Să se rezolve exercițiul 3 prin utilizarea bibliotecii prtools pentru aplicarea algoritmului PCA.

Spor!