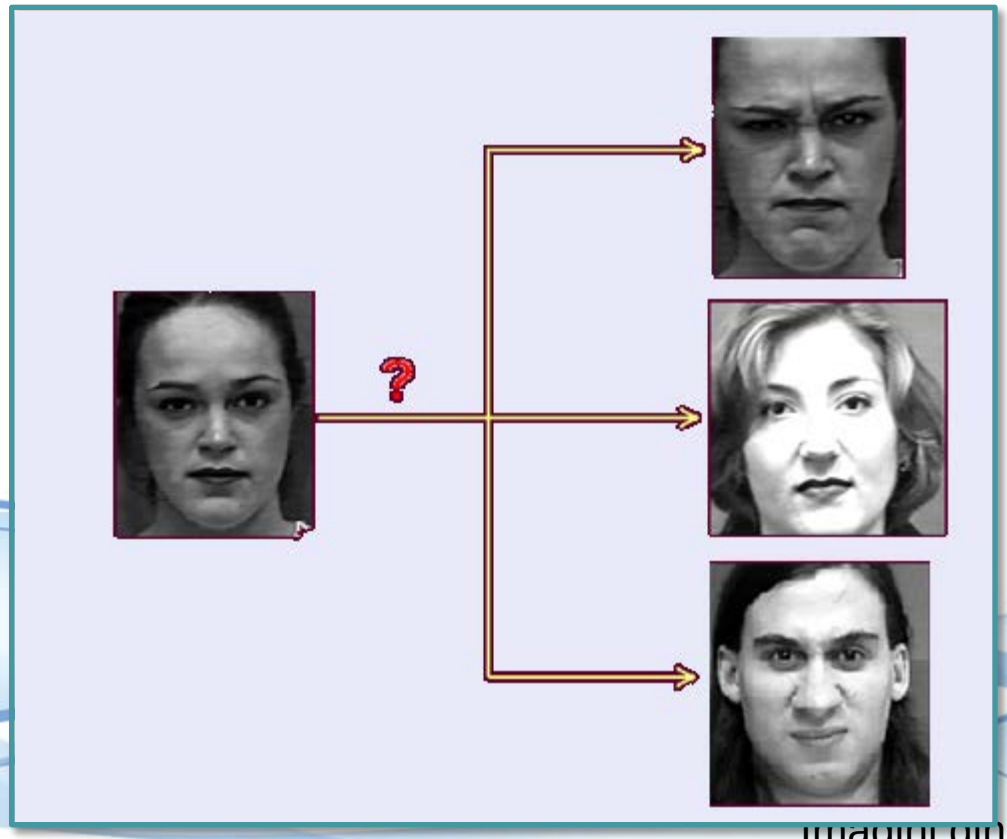
Metode de recunoastere generica

Interfatare vizuala om-masina

Corneliu Florea

Recunoasterea fetelor

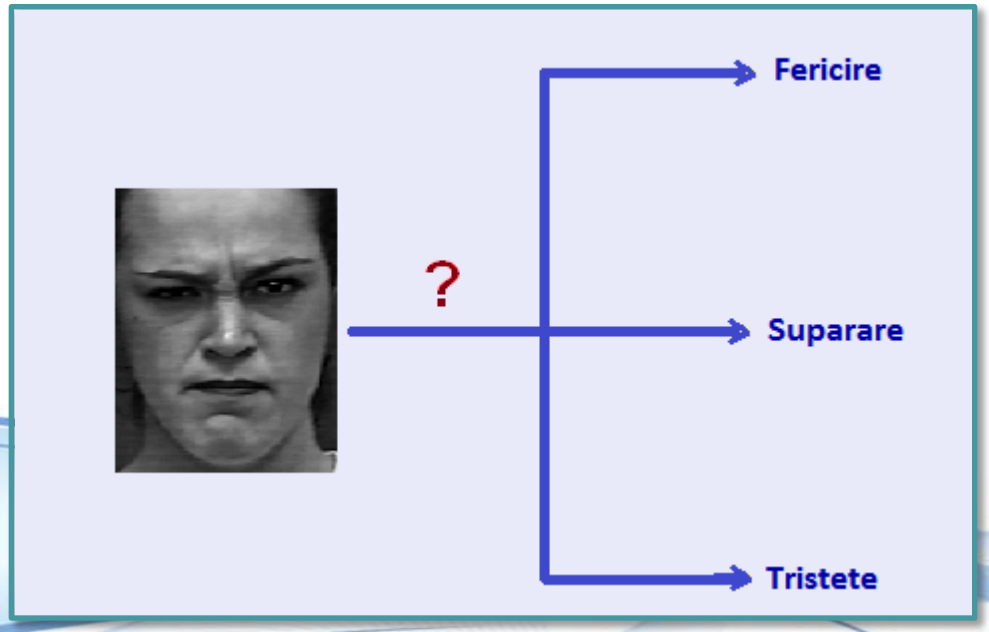
Problema: identificarea identitatii unei persoane.
Setul de persoane este finit si cunoscut



Imagini din baza Cohn-Kanade

Recunoasterea emotiei

Problema: identificarea emotiei unei persoane.
Setul de emotii posibile este finit si cunoscut

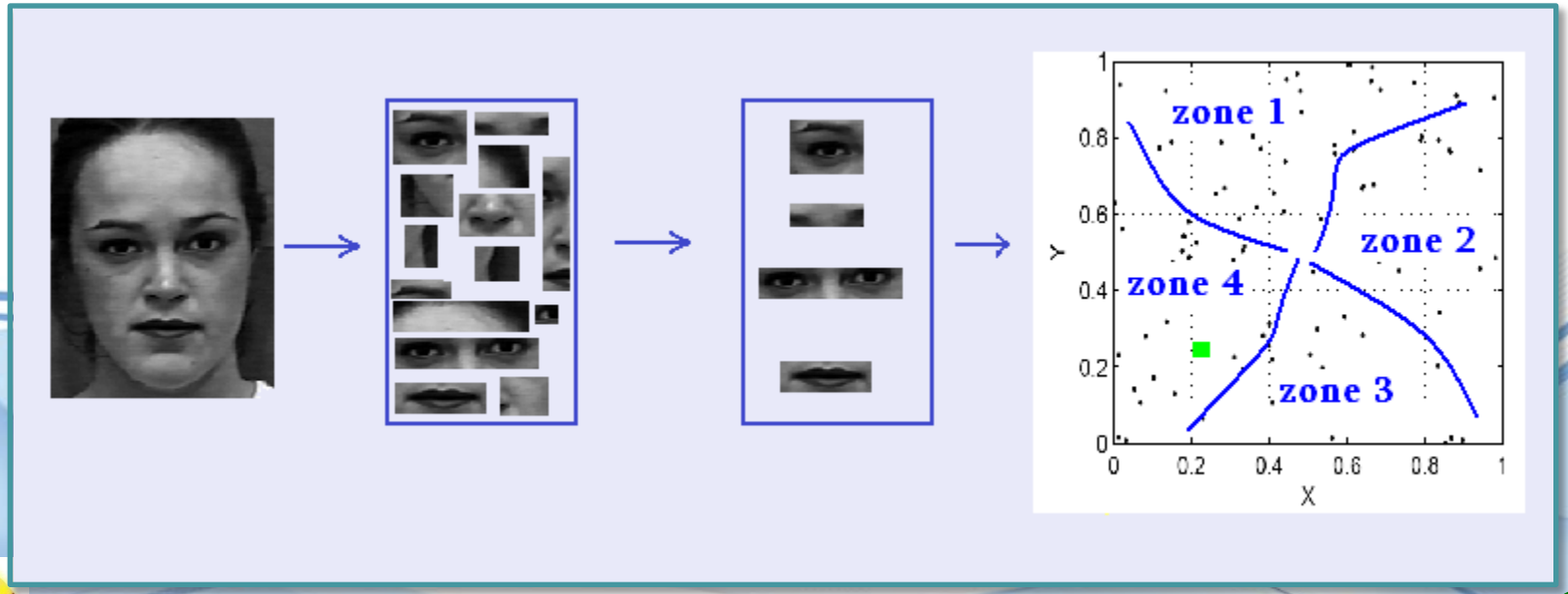


Imagini din baza Cohn-Kanade

Problematica clasificarii

Etape:

- Extragere de trasaturi semnificative problemei
- Reducerea numarului de trasaturilor
- Clasificare:
  - Binara
  - Cu valori multiple



Parametri derivati din segmentare

- Poate cea mai uzuala metoda de clasificare
- Se segmenteaza zona de interes
  - De obicei se alege o zona mai mare
- Se filtreaza zona segmentata
  - Se filtreaza valorile extrase
  - Se filtreaza morfologic
- Se calculeaza diferite valori extrase din rezultat
- Pe o baza de antrenament se identifica valorile utile si se compune regula de decizie finala – divizarea in domenii

Parametri din segmentare: Exemplu

- Problema : Clasificare de gen
- Se segmenteaza pe culoare pielea
- Se construiesc parametri:
  - Raportul de compacticitate, ρ
    - Femeile au fata mai rotunda
  - Numarul de zone de fundal in masca, N_H
    - Barbatii au par facial
  - Dimensiunea (Aria) ochilor, A_E
    - Femeile au ochi mai mari





Parametri din segmentare: Exemplu

- Se construiesc parametri:
  - Fara suport fizic
    - Aria cercului circumscris / Aria dreptunghiului circumscris, R_A
    - Saturatia medie a fetei , S_{avg}
    - Varianta pe planul U, σ^2_U





- Pe baza unei baze de antrenament se construiesc **reguli de decizie**:
  - Decizia este luata pe baza unei reguli compuse

Parametri din segmentare: Exemplu

- Regula de decizie finala:

$R_F = (\rho < 0.5 \;||\; \rho > 0.5\,) \;\&\; (N_H > 24) \;\&\; (A_E < 15) \;\&$
$\&\; (R_A < 0.437) \;\&\; (S_{avg} > 23 \;||\; S_{avg} < 5) \;\&$
$\&\; (10 < \sigma^2_U < 34.1245)$



R_F < T - Barbat

$R_F \geq$ T - Femeie

Se pot obtine rezultate foarte bune dar cu :

- **Multa munca manuala**
- **Baza de date foarte buna**

Extragere automata de trasaturi

Trasatura – scalar/vector care exprima o caracteristica locala a unei imagini/obiect

Extragerea de trasaturi poate fi:

- Spatiala – convolutie cu diferite nuclee
- Spectrala – bancuri de filtre de tip FTJ, FTB, FTS
- Mixta:
  - Filtre trece jos Gaussiene
  - Filtre Gabor

Extragere de trasaturi

Cand se aleg trasaturile trebuie avut in vedere:

- Suficienta - caracteristicile trebuie să aibă informaţii suficiente despre imagine şi nu trebuie să solicite cunoştinţe specifice domeniului pentru extragerea lor.
- Usor de calculat - abordarea să fie fezabilă pentru o colecţie de imagini de mari dimensiuni şi recuperare rapidă
  - Trasaturi “slabe” – usor de calculat, multe
  - Trasaturi “puternice” – putine, complexe.
- Compatibilitate cu SVU - trebuie să fie legate de caracteristici umane perceptuale.

Domeniul de analiza

Domeniu spatial:

- Caut valori constante de o anumita intensitate
- Structuri cu o distributie fixa
  - Trasaturi de tip Haar; Local Binary Pattern

Domeniu spectral:

- Se obtine cu ajutorul transformatei Fourier/ DCT
- Caut variatii

Domeniu mixt

- Caut anumite valori combinate cu anumite variatii
  - Filtre Gabor

Imagine integrala

- Provine din Grafica – "Summed-area tables"
- Varianta 2D a conceputului ***prefix sums***
  - Fie un vector **f** =[f_1, f_2, f_3, …]
  - Se construieste suma cumulativa

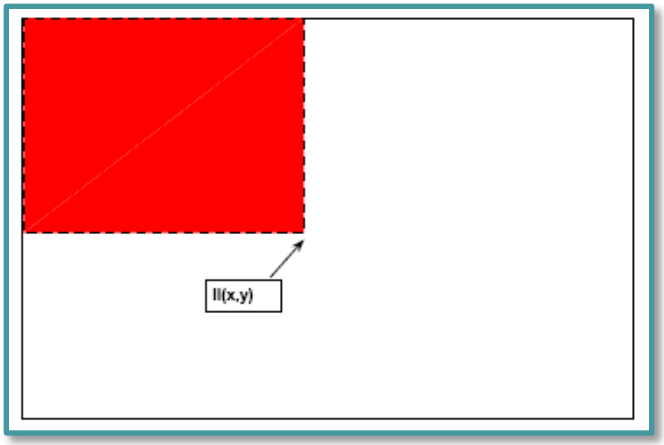
F = [F_1=f_1, F_2=f_1+f_2, F_3 = f_1+f_2+f_3, …]

  - Suma elementelor consecutive din vectorul initial, dintre doi indici i_1 si i_2 este:

$$f_{i_1} + f_{i_1+1} + \ldots + f_{i_2} = F_{i_2} - F_{i_1-1}$$

Imaginea integrala

- Cum implementam acest principiu in 2D?



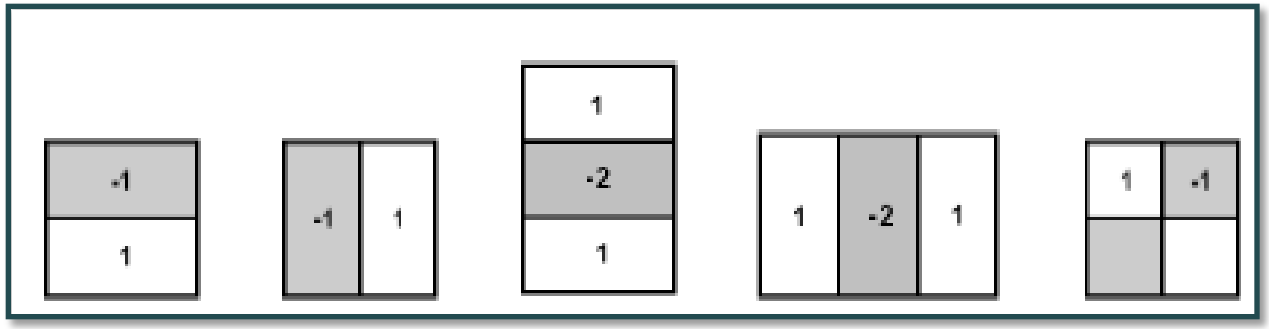
$$II(x,y) = \sum_{i=1}^{x}\sum_{j=1}^{y} I(i,j)$$

- La ce bun?

$$\sum_{i=x_1}^{x_2}\sum_{j=y_1}^{y_2} I(i,j) = II(x_2,y_2) + II(x_1-1,y_1-1)$$
$$- II(x_2,y_1-1) + II(x_1-1,y_2)$$

Trasaturi Haar

- Provin din transformata Wavelet de tip Haar

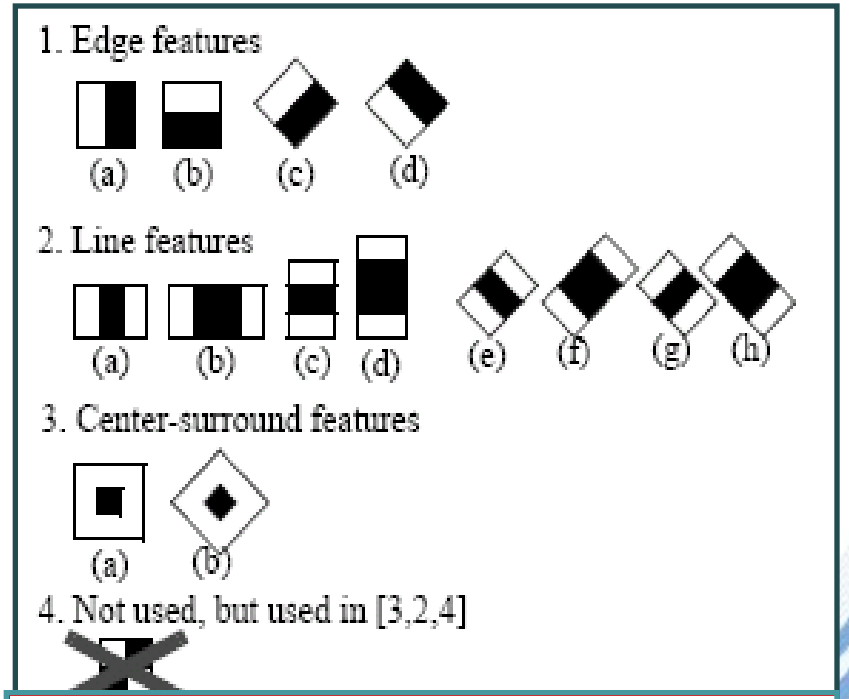


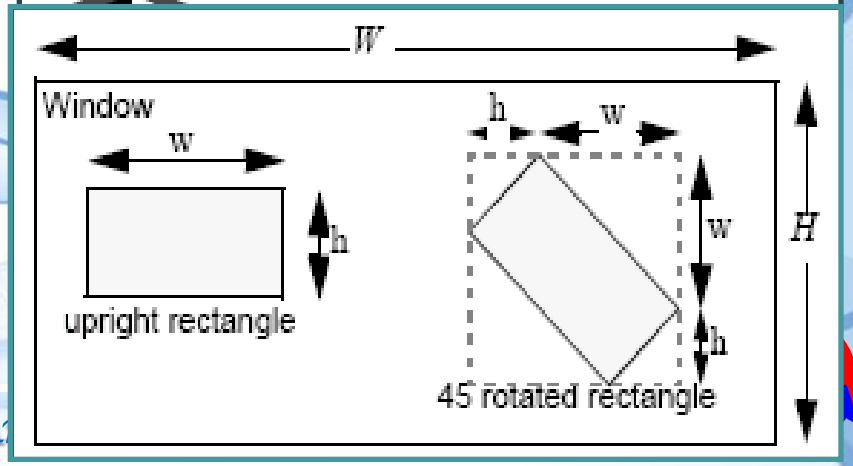
- Dupa Viola si Jones se numesc trasaturi Haar
- Se calculeaza eficient cu Imaginea Integrala

Trasaturi Haar extinse

Ex: Trasaturi Haar extinse

- Lienhart propune 45 trasaturi dreptunghiulare rotate
- O trasatura este representata de:
  - Tipul trasaturii: 1(a), 2(b);
  - Locatia (x, y)
  - Dimensiunea trasaturii (w, h)
- Pentru o fereastra 24x24 exista 117,941 trasaturi diferite





Trasaturi Haar asimetrice

- **Ramirez** propune trasaturi asimetrice
- O trasatura este representata de:
  - Tipul trasaturii: 1,2,…6;
  - Locatia (x, y)
  - Parametri
- Pentru o fereastra 24x24 exista $2*10^8$ trasaturi diferite

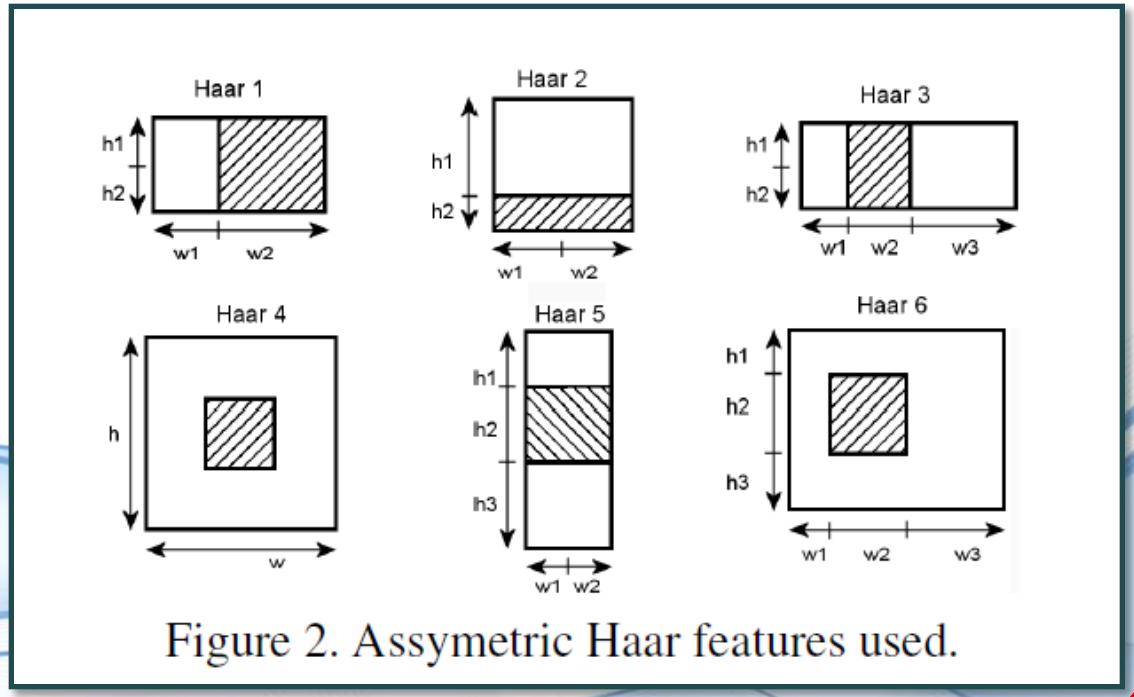


Figure 2. Assymetric Haar features used.

Trasaturi Census

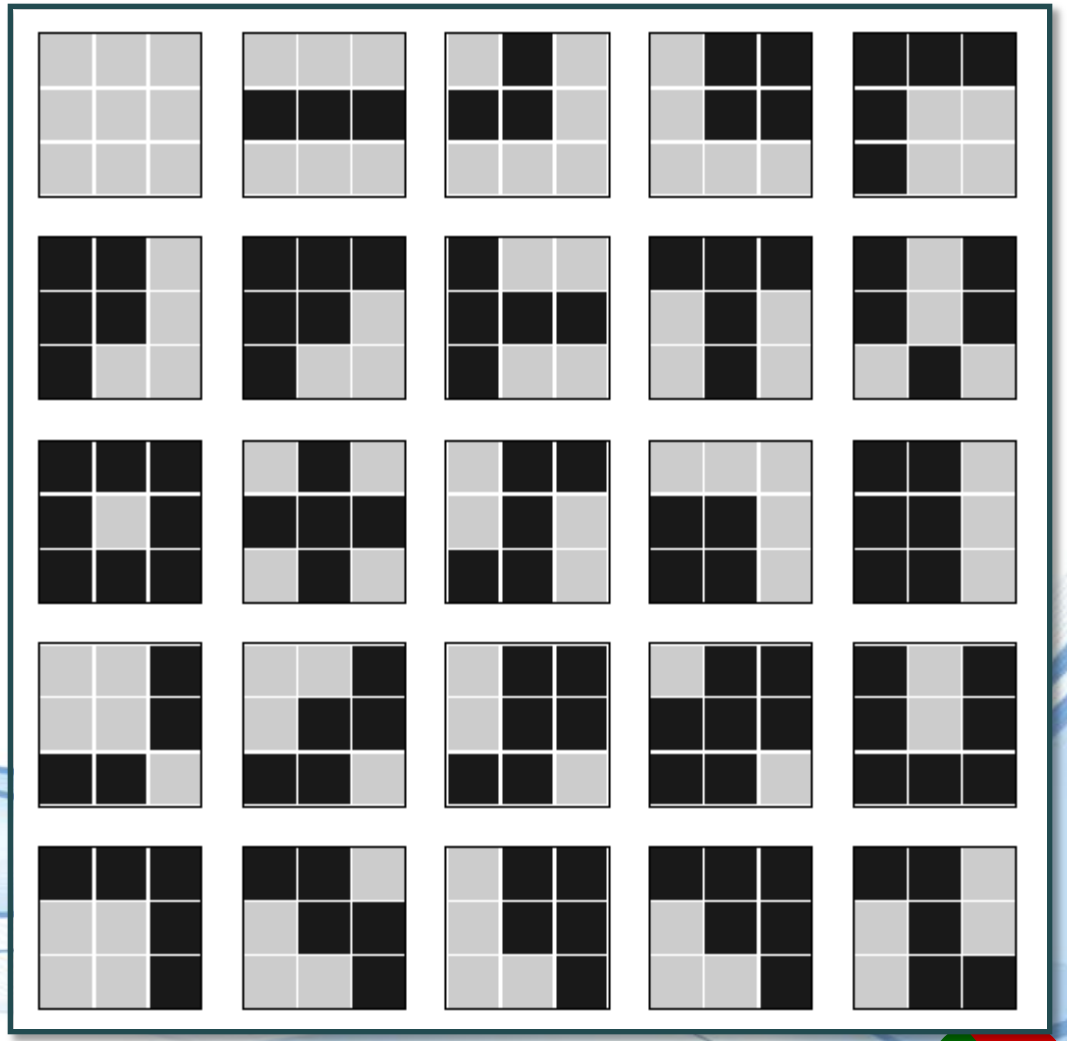
- Zabih si Woodfill introduc transformarea Census:

$$C(x) = \bigodot_{y \in N} \zeta(I(x), I(y))$$

  - Se considera o vecinatate (ce exclude punctul) curent:
  - ζ(A,B) functie de comparatie
    - la valoarea 1 daca A>B
    - la valoarea 0 daca A<B

Trasaturi Census

- Kublbeck propune utilizare trasaturilor census in detectie:
  - Aproximeaza o succesiune de Haar
  - Sensibile pentru textura



Vecinatati

Texture Unit

- Introdus de catre He si Wang in 1990
- Descriptor primar de textura
- Se defineste pe o ferestra 3x3 in care
  - Valoarea curenta este V_0
  - Valorile din vecinatate sunt V_1,…, V_8:

V_1	V_2	V_3
V_4	V_0	V_5
V_6	V_7	V_8

Texture Unit se defineste a fi o matrice 3x3 cu valorile E_1,…, E_8

$$E_k = \begin{cases} -1, V_k < V_0 \\ 0, V_k = V_0 \\ 1, V_k > V_0 \end{cases}$$

Fuzzy Texture Unit

- Exista variante:
  - Fuzzy Texture Unit definit pe baza functiilor de apartenenta, μ

$$E_k = \begin{cases} \mu_0(V_k - V_0) \\ \mu_1(V_k - V_0) \\ \mu_2(V_k - V_0) \end{cases}$$

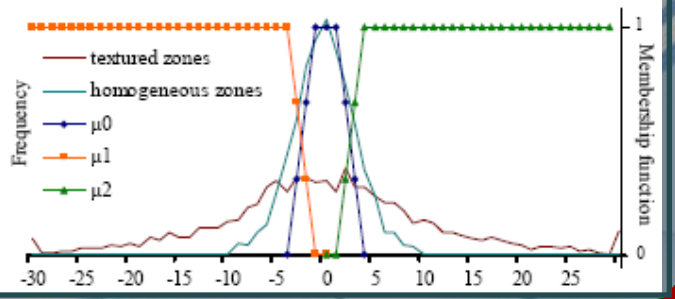
  - Fuzzy Texture Unit simplificat

$$E_k = \begin{cases} -1, V_k - V_0 > T \\ 0, |V_k - V_0| \le T \\ 1, V_0 - V_k > T \end{cases}$$

$$\mu_0 = \begin{cases} 1 & if \quad x \le -4 \\ \frac{-(x+1)}{3} & if \quad -4 < x < -1 \\ 0 & if \quad x \ge -1 \end{cases}$$

$$\mu_1 = \begin{cases} 0 & if \quad x \le 4 \\ \frac{x+4}{3} & if \quad -4 < x < -1 \\ 1 & if \quad -1 \le x \le 1 \\ \frac{4-x}{3} & if \quad 1 < x < 4 \\ 0 & if \quad x \ge 4 \end{cases}$$

$$\mu_2 = \begin{cases} 0 & if \quad x \le 1 \\ \frac{x-1}{3} & if \quad 1 < x < 4 \\ 1 & if \quad x \ge 4 \end{cases}$$



Texture Unit - exemplu

- Texture Unit simplificat
  - T=5:
  - Fereastra 7x7

194	197	193	199	192	199	195
189	191	194	201	192	197	197
196	190	187	198	194	192	191
198	198	191	**195**	199	198	196
187	193	201	194	193	193	195
183	189	204	200	197	196	200
193	192	190	188	201	196	192

0	0	0	0	0	0	0
-1	0	0	1	0	0	0
0	0	-1	0	0	0	0
0	0	0	**0**	0	0	0
-1	0	1	0	0	0	0
-1	-1	1	0	0	0	0
0	0	0	-1	1	0	0

Texture Unit - Codare

- Reprezentarea Texture Unit:
  - Doi biti (trei valori) pentru fiecare punct al ferestrei
  - Fereastra *NxN*
  - Sunt necesari $2(N^2-1)$ biti
- Nu se poate mai simplu?
- Putem utiliza un singur bit pentru codare:
  - Local Binary Pattern

Local Binary Pattern

- **Local Binary Pattern** (LBP) este un descriptor de textura care codifica primitivele locale (muchii, curbe, pete, zone uniforme) intr-o histograma de trasaturi
- Se defineste pe o ferestra 3x3 in care
  - Valoarea curenta este g_0
  - Valorile din vecinatate sunt $g_1,\ldots, g_8$:

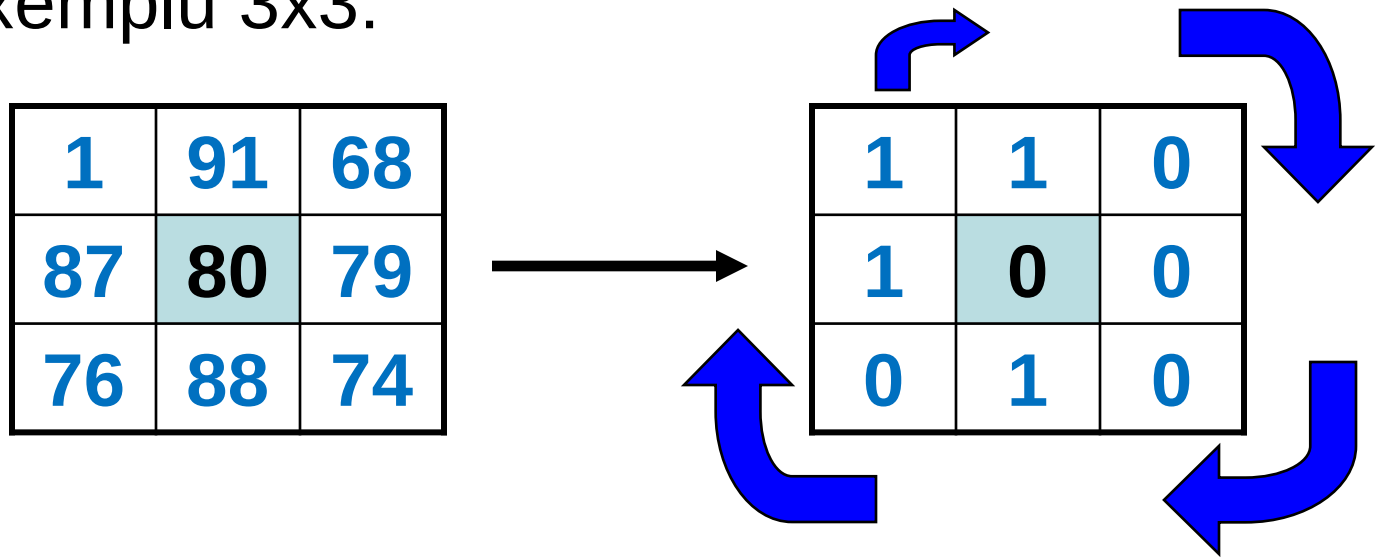
g_1	g_2	g_3
g_4	g_0	g_5
g_6	g_7	g_8

- Fereastra rezultanta este:

$$e_i = \begin{cases} 0, & g_i < g_0 \\ 1, & g_i \geq g_0 \end{cases}$$

Local Binary Pattern

- Exemplu 3x3:



- Cele 8 valori calculate formeaza cei 8 biti ai unui octet:
  - Se incepe din (0,0) si se parcurge circular
    - Se fixeaza directia
    - **11000101**

Local Binary Pattern

- Numarul se reprezinta in baza 10:

$128 \cdot 1+64 \cdot 1+32 \cdot 0+16 \cdot 0+8 \cdot 0+4 \cdot 1+2 \cdot 0+1 \cdot 1 = 201$

- Proprietati:
  - Invariant la orice schimbare monotona a nivelului de gri
  - Usor de calculat
- De multe ori este folosit in conjunctie cu contrastul *C*:

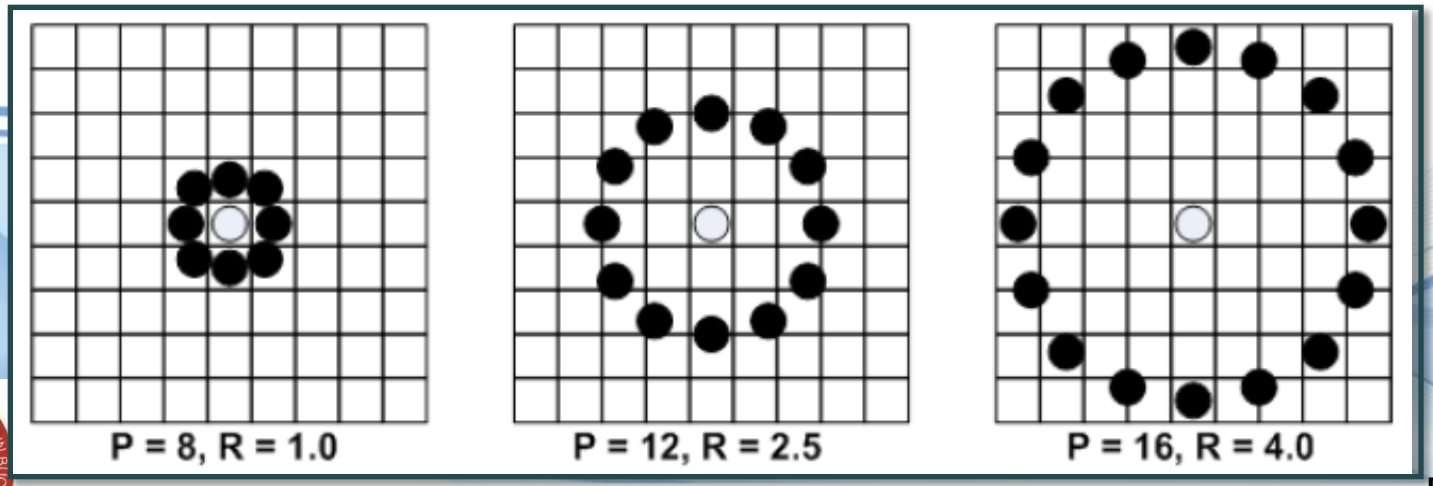
86	91	68
87	80	79
76	88	74

1	1	0
1	0	0
0	1	0

C = (86+91+87+88)/4 – -(68+79+74+76)/4 = = -13.75

LBP- multirezolutie

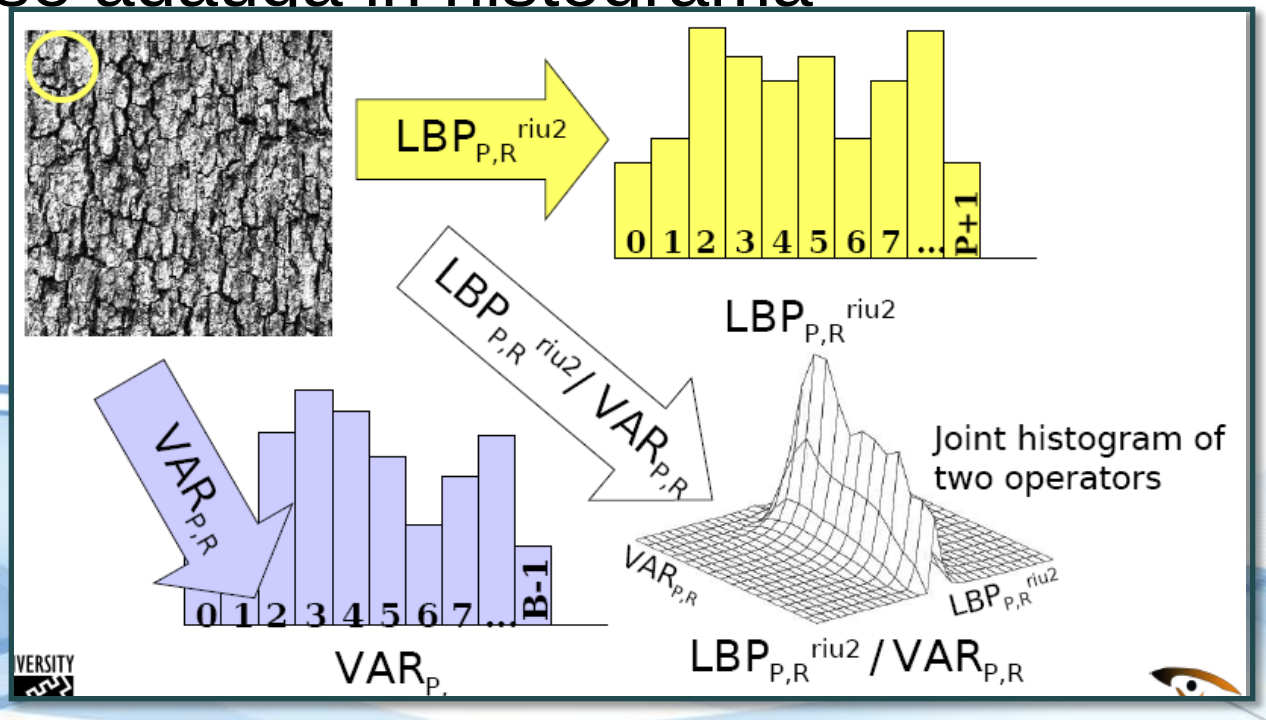
- Ferestrele de calcul sunt parametrice in functie de numarul de pixeli (biti), *P* si de raza, *R*
  - Daca nu se nimereste intr-un pixel, valoarea pt LBP se obtine prin interpolare (bi-lineara)
- Invariante la rotatie (in jurul punctului central)
- Invariante la variatia nivelului de gri



[Pietikäinen]

LBP- caracterizarea unei imagini

- Caracterizarea ↔ histograma trasaturilor (LBP si/sau C)
- Pentru fiecare pixel se determina LBP-ul si se adauga in histograma



[Pietikäinen]

Histogram of Oriented Gradient (HOG)

- Trasatura localizata spatial, utila in special pentru texturi. Introdusa de Dalal&Triggs, [Dalal]
- **Ideea** : forma unui obiect poate fi descrisa cu ajutorul distributiei intensitatii gradientilor respectiv a directiei muchiilor.
- Imaginea se imparte in celule. Fiecare celule are o histograma de orientari.

- **Descriptorul HOG** = **celule** x **histograme**

Algoritmul HOG

Algoritm:

1. Calculul gradientului
2. Histograma de orientari
3. Blocul descriptiv
4. Normalizarea blocurilor
5. Constructia clasificatorului final

Antrenare

HOG : Calculul gradientului

1. Calculul gradientului
   - Se bazeaza, de obicei pe baza convolutiei cu:
     - **Nuclee centrate**

-1	**0**	1

-1
0
1

     - Nuclee necentrate

-1	1

-1
1

     - Sobel

-1	0	1
-2	**0**	2
-1	0	1

-1	-2	-1
0	**0**	0
1	2	1

HOG : Histograma de orientari

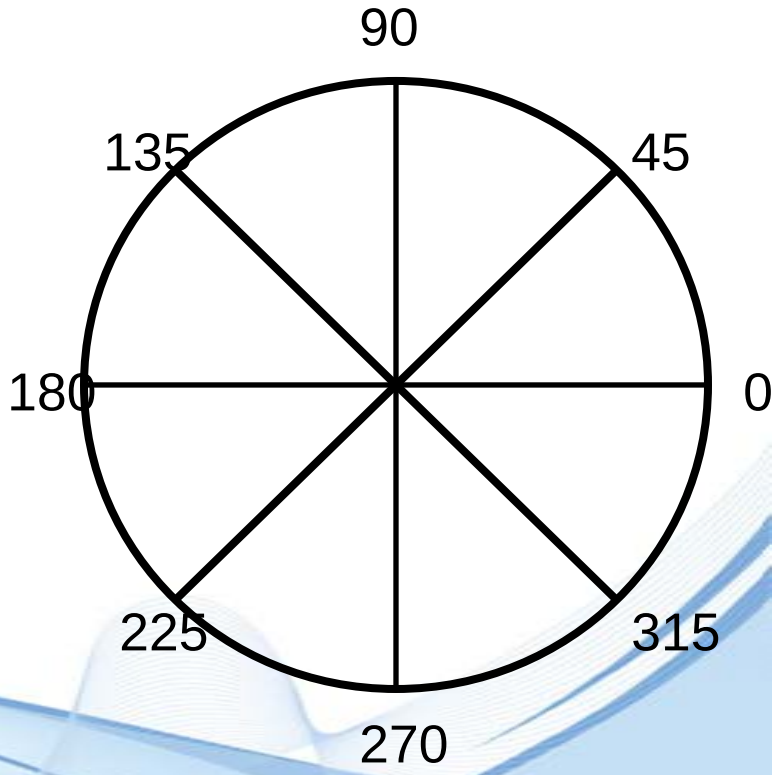
Orientarea se calculeaza cu precizie continua

Se interpoleaza la un numar de pozitii predefinite:

- In domeniu fara semn [0,180]
- In domeniu cu semn [0,360]

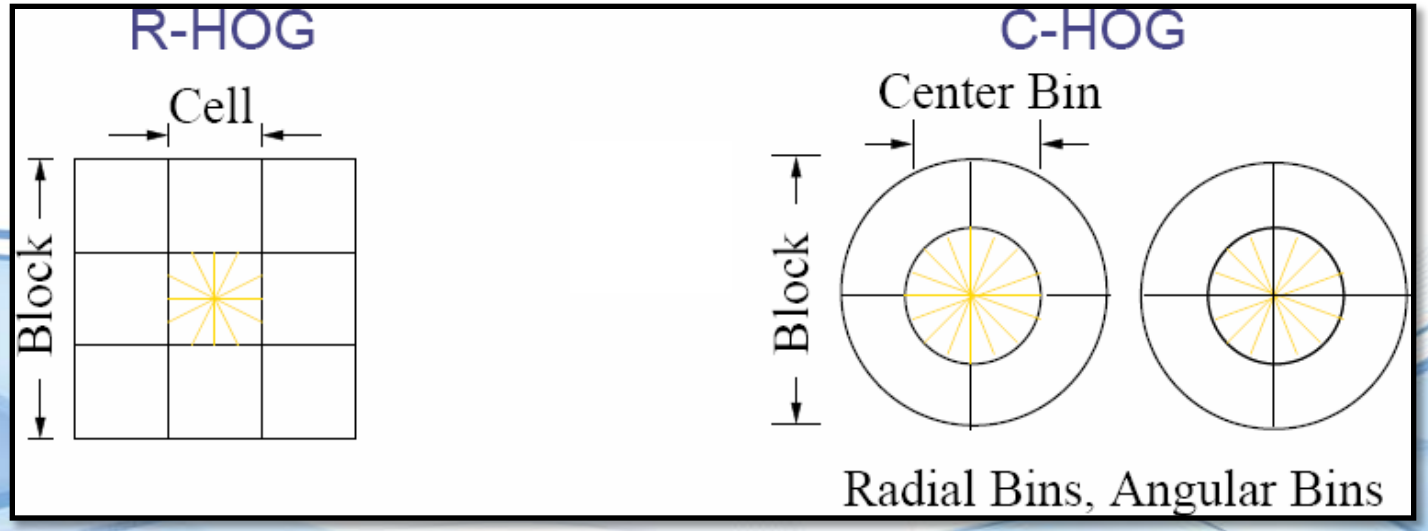
Un gradient voteaza la ambii vecini:

- In functie de **distanta** pana la ei
- Votul este proportional cu **magnitudinea** gradientului
- Se poate aplica o functie magnitudinii



HOG :Blocul descriptiv

- Imaginea se imparte in blocuri, partial suprapuse
- Fiecare bloc este alcatuit din celule
- Pe fiecare bloc se aplica **normalizarea contrastului**
- Celulele pot fi **rectangulare** sau radiale (log-polare)
- Histograma se calculeaza pe fiecare celula



HOG :Normalizarea blocurilor

- Exista deja un descriptor pe fiecare bloc, v
- Pe fiecare bloc se aplica **normalizarea contrastului descriptorului:**
  - Norma L_2
  - trunchere :
  - Norma L_1

$$v = \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon}}$$

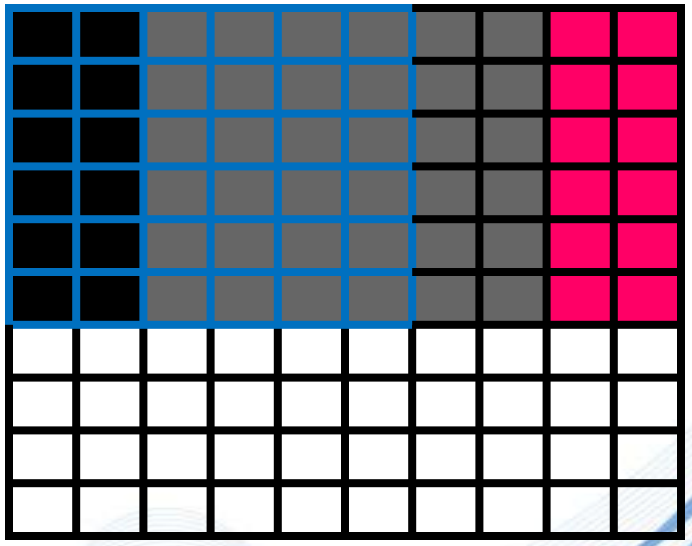
Sa nu impartim la 0

Norma L_2

Descriptorul final este vectorul tutoror componentelor din toate celulele normalizate din toate blocurile din fereastra de interes

HOG :Exemplu

- Se utilizeaza nuclee centrate
- Celula
  - Imaginea este impartita in celule : fiecare pixel apartine unei celule si numai uneia
  - Ex: are 6 x 6 pixeli
- Blocul
  - Suparapuse sau distantate
  - Ex: are 3 x 3 celule
- 9 directii de orientare in [0 , 180]
- Norma L2 cu trunchere in [0, 0.2]



Recapitulare – transformata Fourier 2D

Transformata directa:

$$v(k,l) = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u(m,n) \exp\left(-\frac{2\pi j(kl+mn)}{N} \right)$$

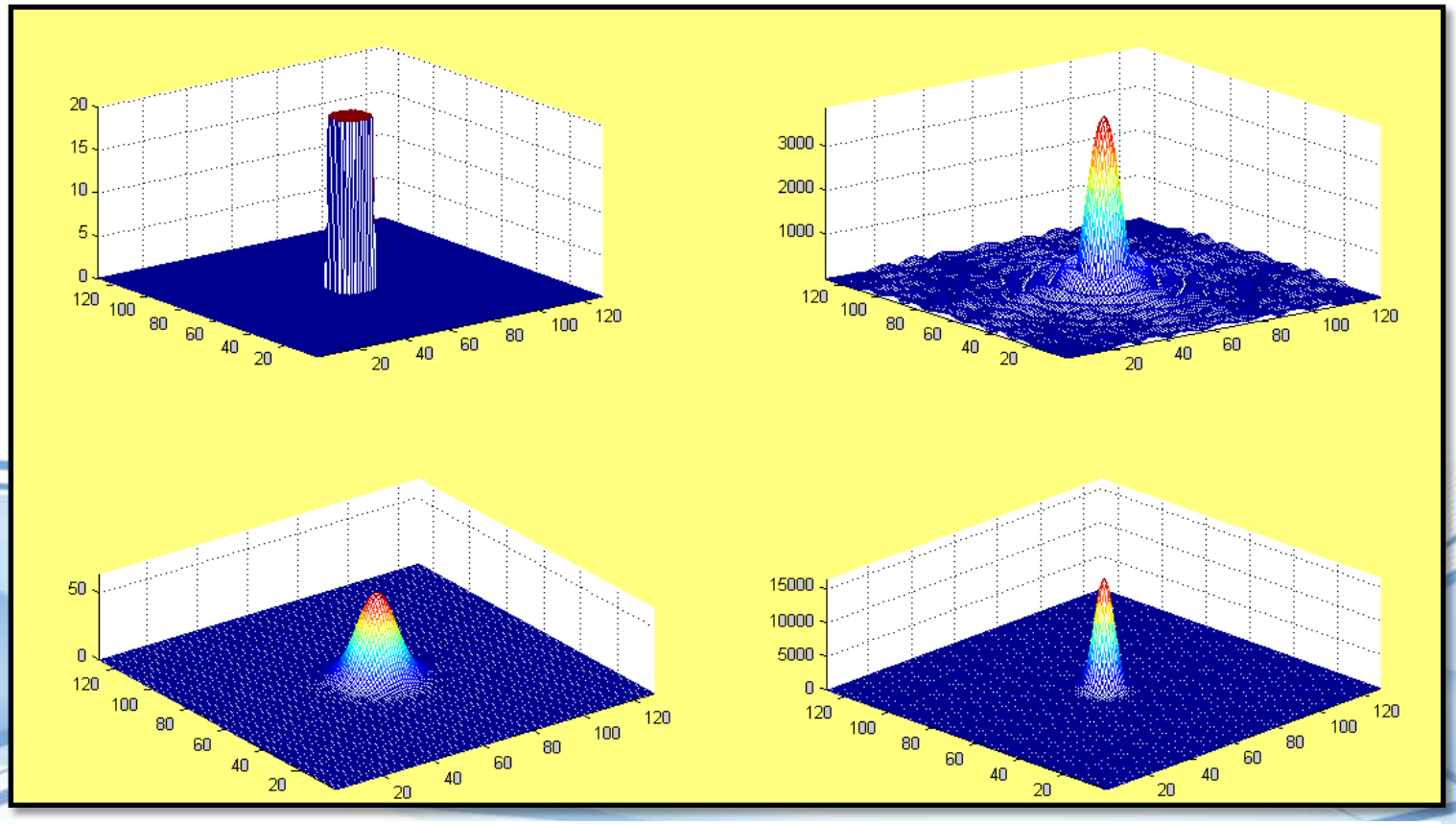
Transformata inversa:

$$u(m,n) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} v(k,l) \exp\left(\frac{2\pi j(kl+mn)}{N} \right)$$

Exemple

Box (spatial) ↔ *Sinc* (spectral)

Clopot Gaussian (spatial) ↔ *Clopot Gaussian* (spectral)



Filtru Gabor 1D

Filtru Gabor – produsul dintr-o infasuratoare gaussiana si o purtatoare sinusoidala complexa

$$g(t) = ke^{j\theta}w(at)s(t)$$

Unde:

$$w(t) = e^{-\pi t^2}$$

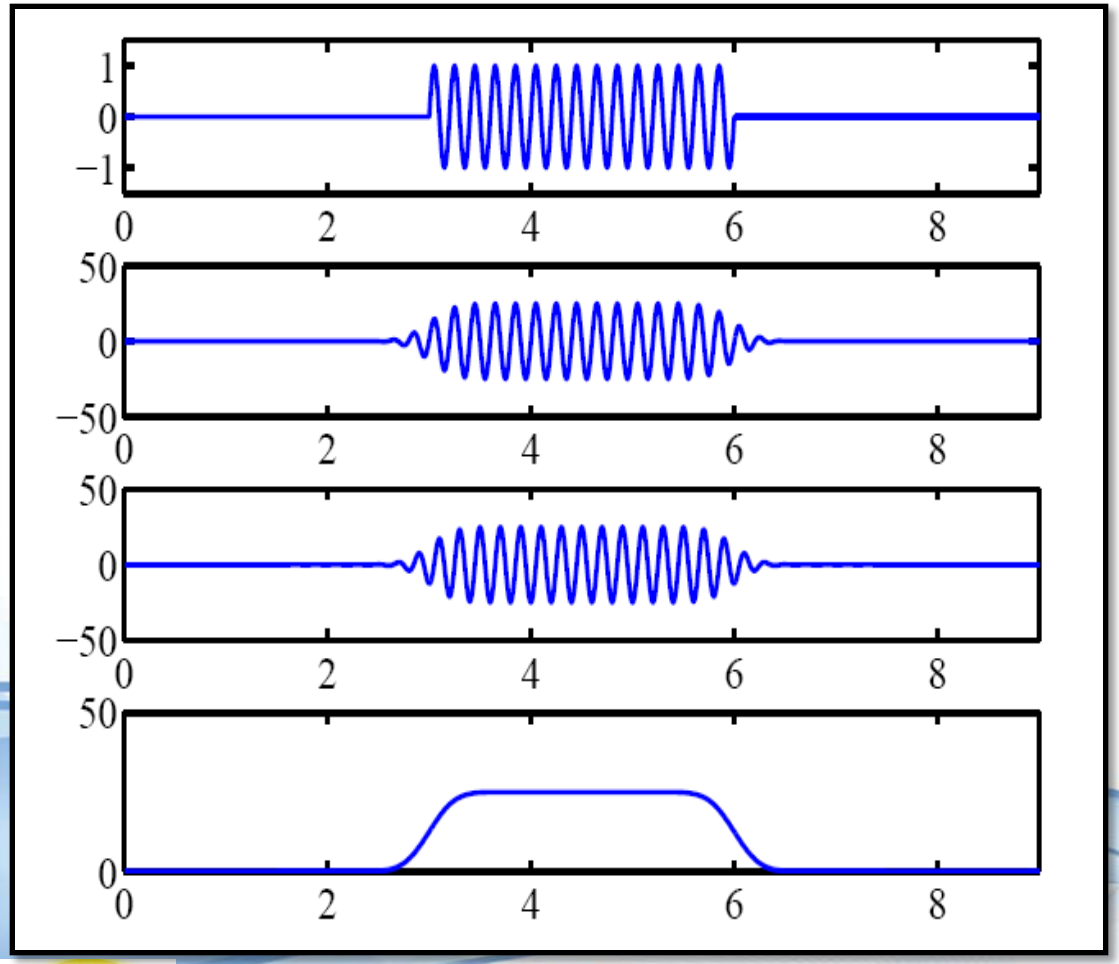
$$s(t) = e^{j(2\pi f_0 t)}$$

$$e^{j\theta}s(t)e^{j(2\pi f_0 t)} = \sin(2\pi f_0 t + \theta) + j\cos(2\pi f_0 t + \theta)$$

Aici k, θ, f_0 sunt parametrii filtrului.

Scalar de iesire – energia filtrului

Filtru Gabor 1D - exemplu



Semnal de intrare

Iesirea filtrului Gabor (purtatoarea cosinusoidala)

Iesirea filtrului Gabor (purtatoarea sinusoidala)

Energia filtrului Gabor

[Movellan]

Filtru Gabor 2D

$$g(x, y) = s(x, y)\, w_r(x, y)$$

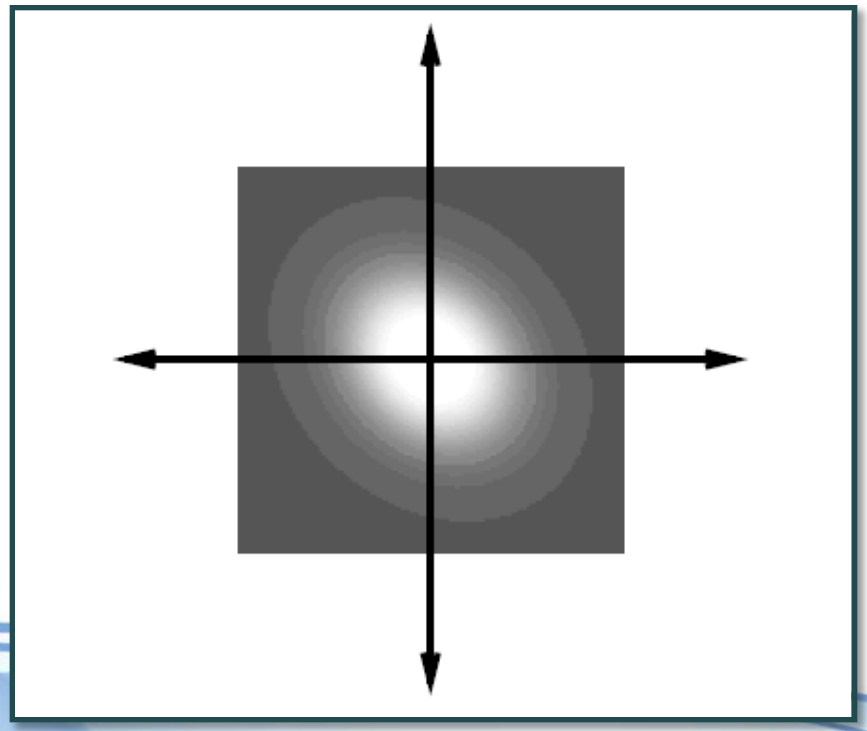
$$s(x, y) = exp\left(j\left(2\pi F_0 (x \cos \omega_0 + y \sin \omega_0) + P\right)\right)$$

$$w_r(x, y) = K\, exp\left(-\pi\left(a^2 (x - x_0)_r^2 + b^2 (y - y_0)_r^2\right)\right)$$

Unde:

- K : Amplitudinea infasuratoarei Gaussiene
- (a, b) : Factorii de scala pe cele doua axe ale Gaussienei
- θ: Unghiul de inclinare al filtrului.
- (x_0, y_0) : Punctul central al filtrului
- F_0: amplitudinea infasuratoarei
- ω_0: frecventa (pulsatia infasuratoarei)
- P : Faza purtatoarei

Filtru Gabor 2D



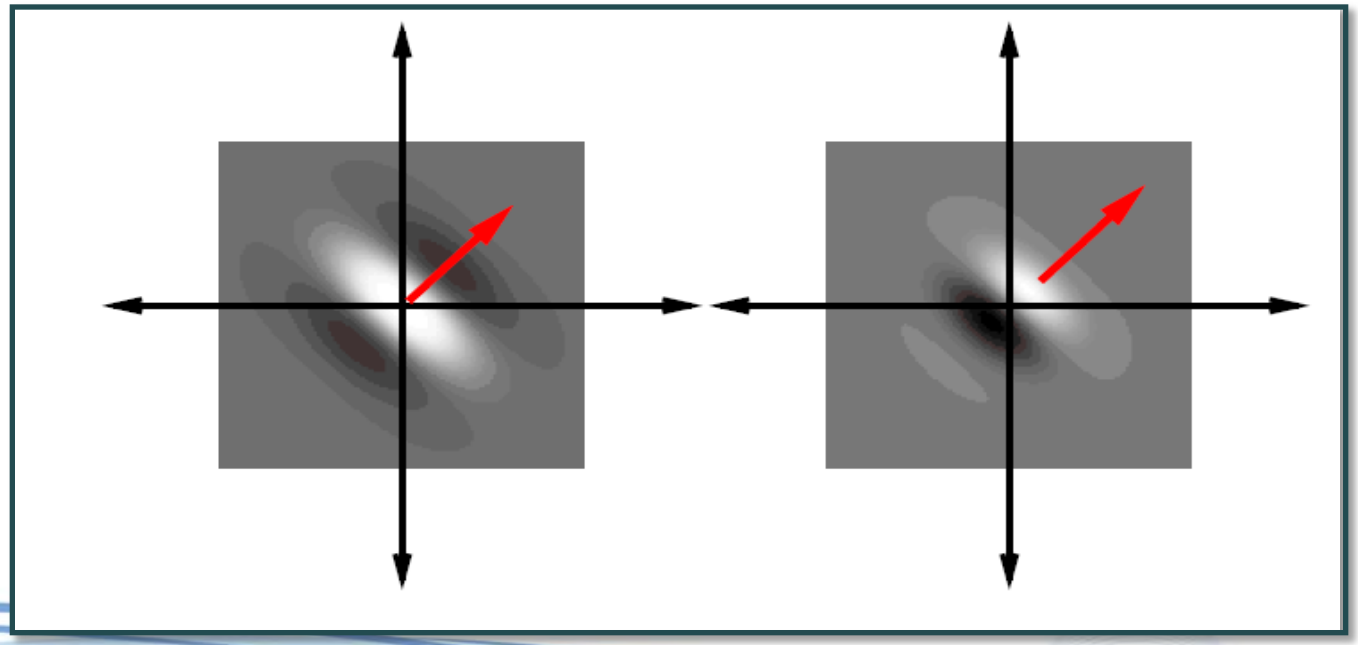
[Movellan]

Infasuratoarea Gaussiana:

- $x_0 = y_0 = 0$
- $a = 1/50$ pixeli
- $b = 1/40$ pixeli
- $\theta = 45^o$

Filtru Gabor 2D

Partea reala si imaginara a unui filtru Gabor in domeniul spatial:



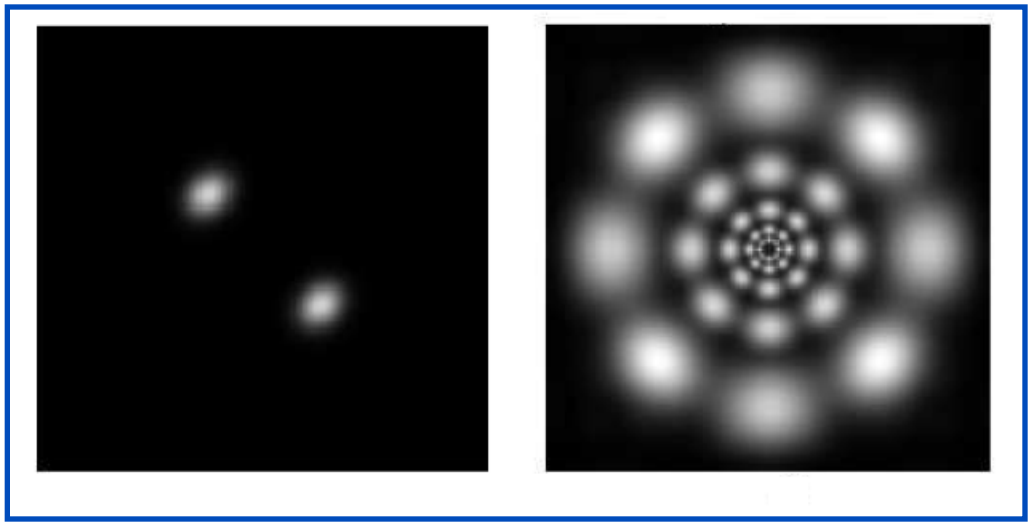
Imaginile sunt 128x128. Parametrii folositi sunt:

- $x_0 = y_0 = 0$
- $a = 1/50$ pixeli, $b = 1/40$ pixeli,
- $\theta = -45^o$, $F_0 = \sqrt{2}/80$ ciclii/pixel,
- $\omega_0 = 45\ ^o/s$, $P = 0^o$.

[Movellan]

Bancuri de filtre Gabor

Pentru extragerea de trasaturi se folosesc bancuri de filtre Gabor

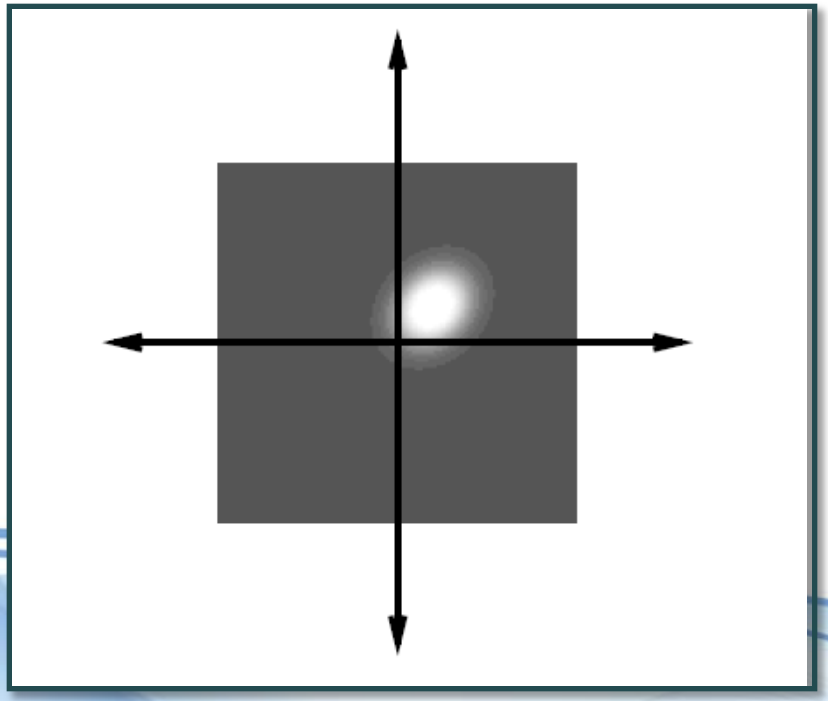


Amplitudinea functiei de transfer a unui filtru Gabor bi-dimensional cu orientare de 45^{o} si frecvente centrale nenule;

Amplitudinea unei functii de transfer a unui banc de filtre Gabor bi-dimensionale (4 orientarii si 5 scale);

Filtru Gabor 2D

Transformata Fourier a filtrului Gabor



Imaginile sunt 128x128.

Parametrii folositi sunt:

- $x_0 = y_0 = 0$
- $a = 1/50$ pixeli, $b = 1/40$ pixeli,
- $\theta = -45^o$, $F_0 = \sqrt{2}/80$ ciclii/pixel,
- $\omega_0 = 45\ ^o/s$, $P = 0^o$.

[Movellan]

MPEG-7

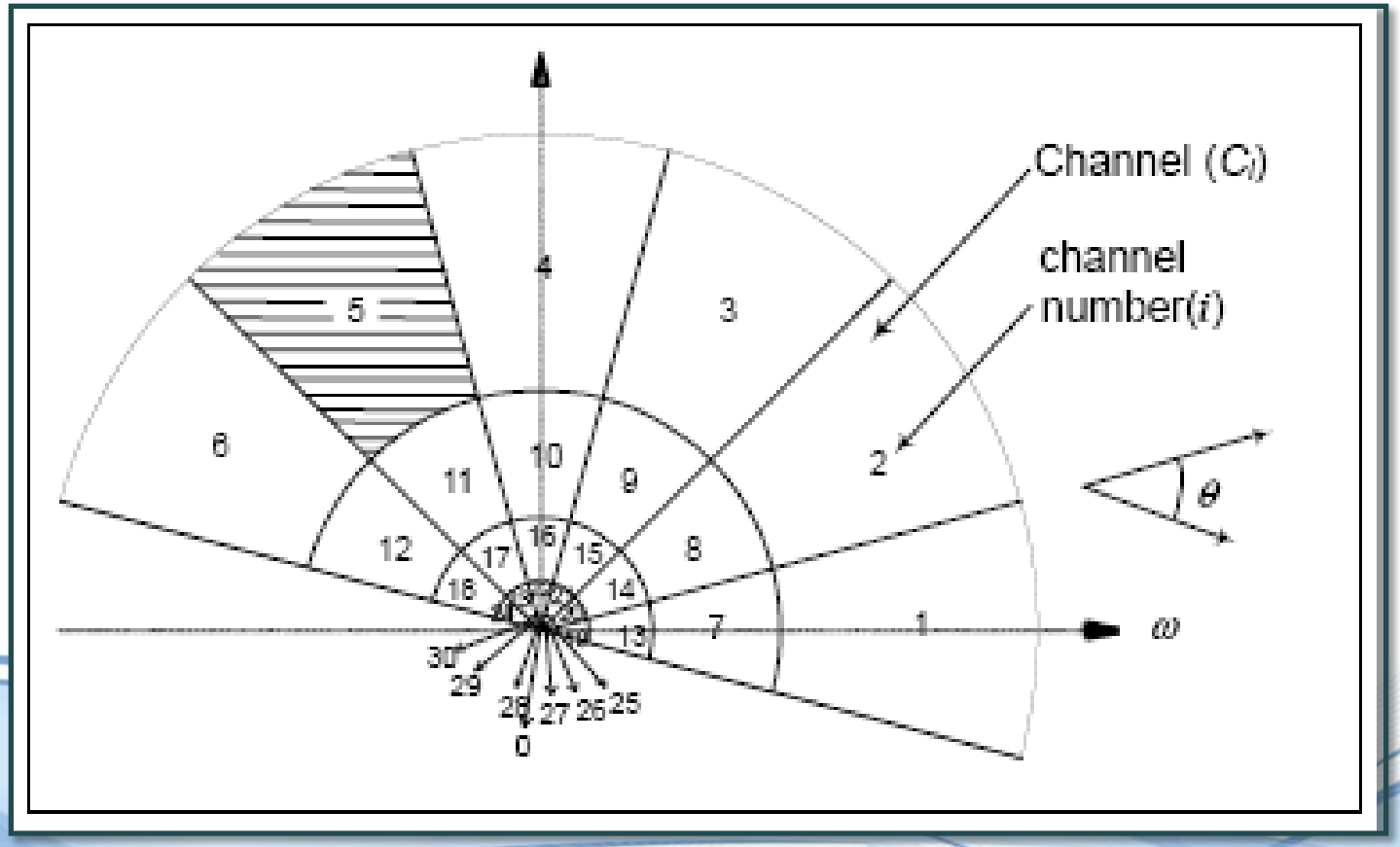
MPEG:

- Motion Picture Experts Group
- Elaborare de standarde pentru transmisie audio si transmisie si compresie video
- Include descriptori de textura:
  - Homogeneous Texture - ***3.2.3.1*** Banc de filtre Gabor
  - Edge Histogram
  - Texture Browsing

Filtre Gabor in MPEG-7

- Similaritate cu SVU – descompunere in benzi spectrale [ManRo]
  - Benzi simetrice fata de centru in coordonate polare
  - Benzi de latime constanta pe fiecare directie angulara
- Bancul de filtre Gabor este specificat:
  - Directii unghiulare spatiate cu 30 de grade – 6 directii
  - Frecventa centrala $\omega_s = \omega_0 2^{-s}$, $s=\{0,1,2,3,4\}$; $\omega_0 = 3/4$
  - Latimea de banda $B_s = B_0 2^{-s}$, $s=\{0,1,2,3,4\}$; $\omega_0 = 1/2$

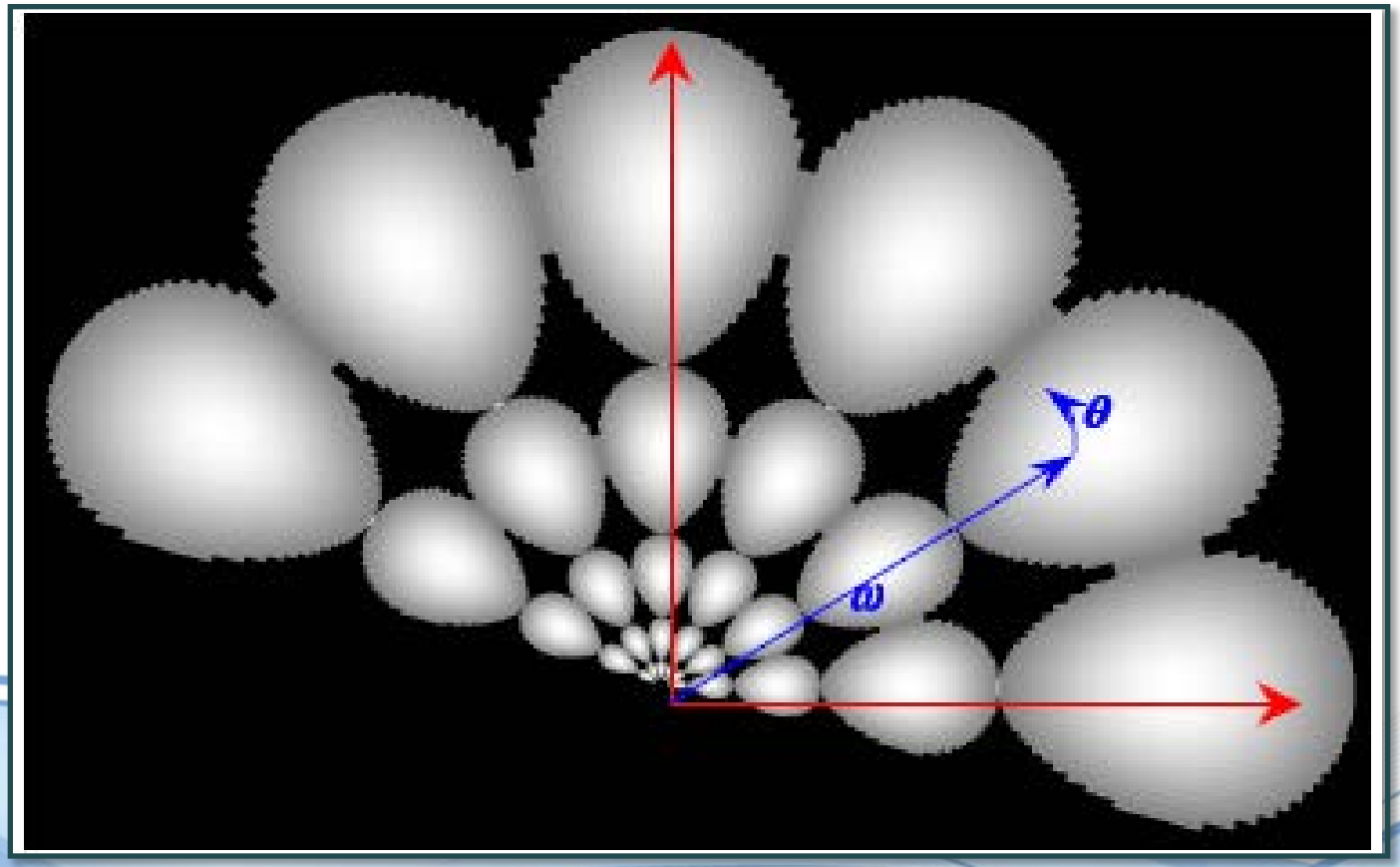
Filtre Gabor in MPEG-7



Impartirea spatiului de frecventa

[ManRo]

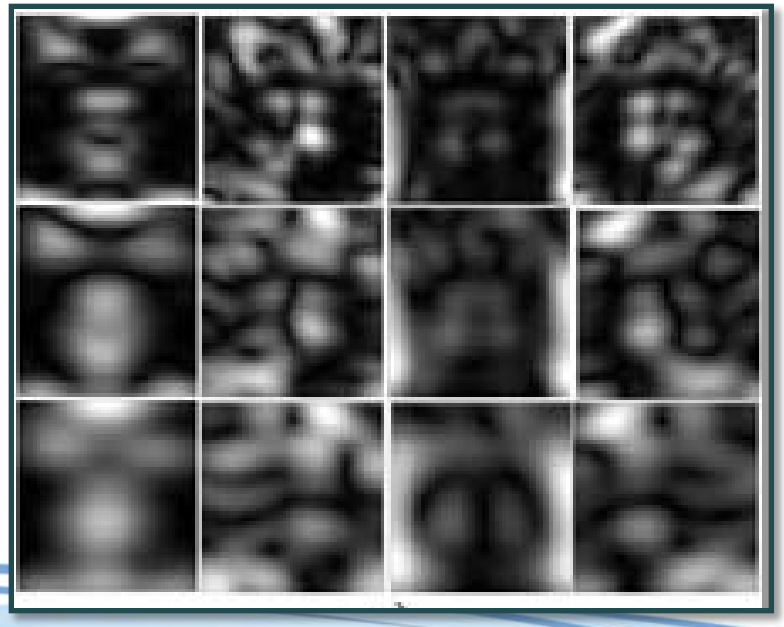
Filtre Gabor in MPEG-7



Bancul de filtre Gabor in coordonate polare;

[ManRo]

Exemplu – recunoasterea emotiilor

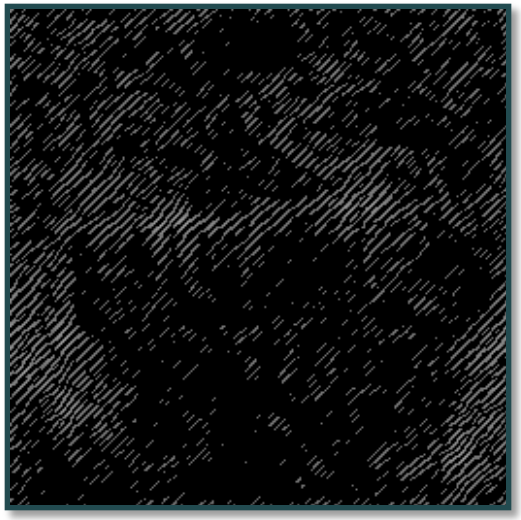
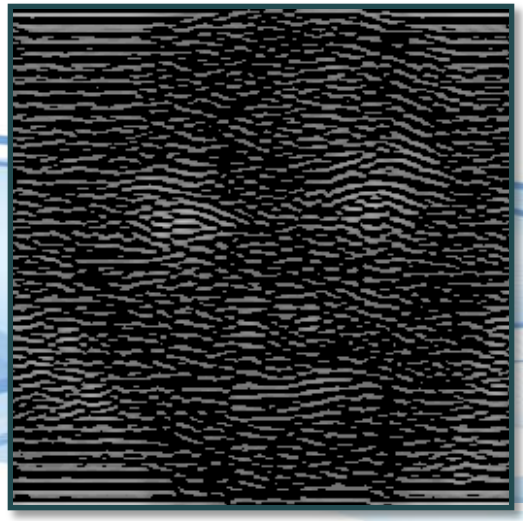
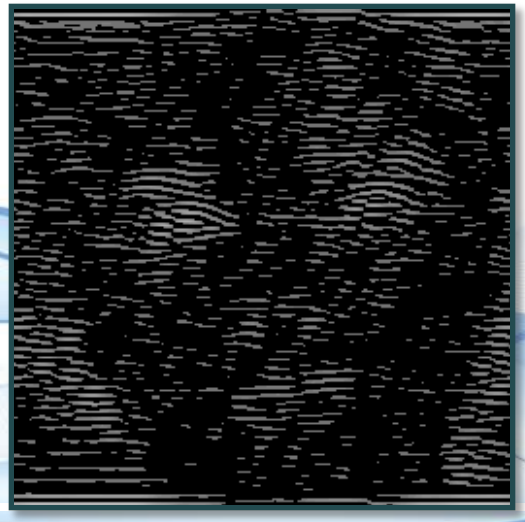


Magnitudinea imaginilor rezultate in urma unui filtrarii cu un banc de 12 filtre Gabor

[Buciu08]

Exemplu: fata umana

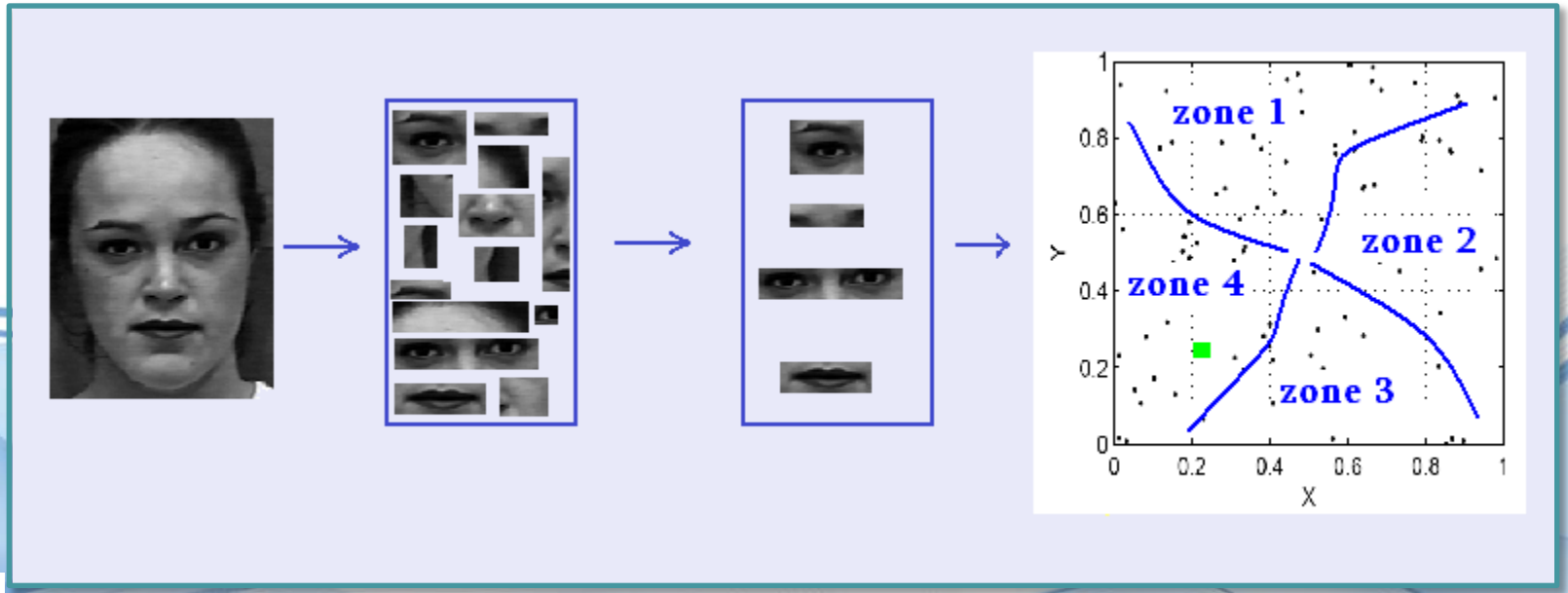


Problematica clasificarii

Etape:

- Extragere de trasaturi semnificative problemei
- Reducerea numarului de trasaturilor
- Clasificare



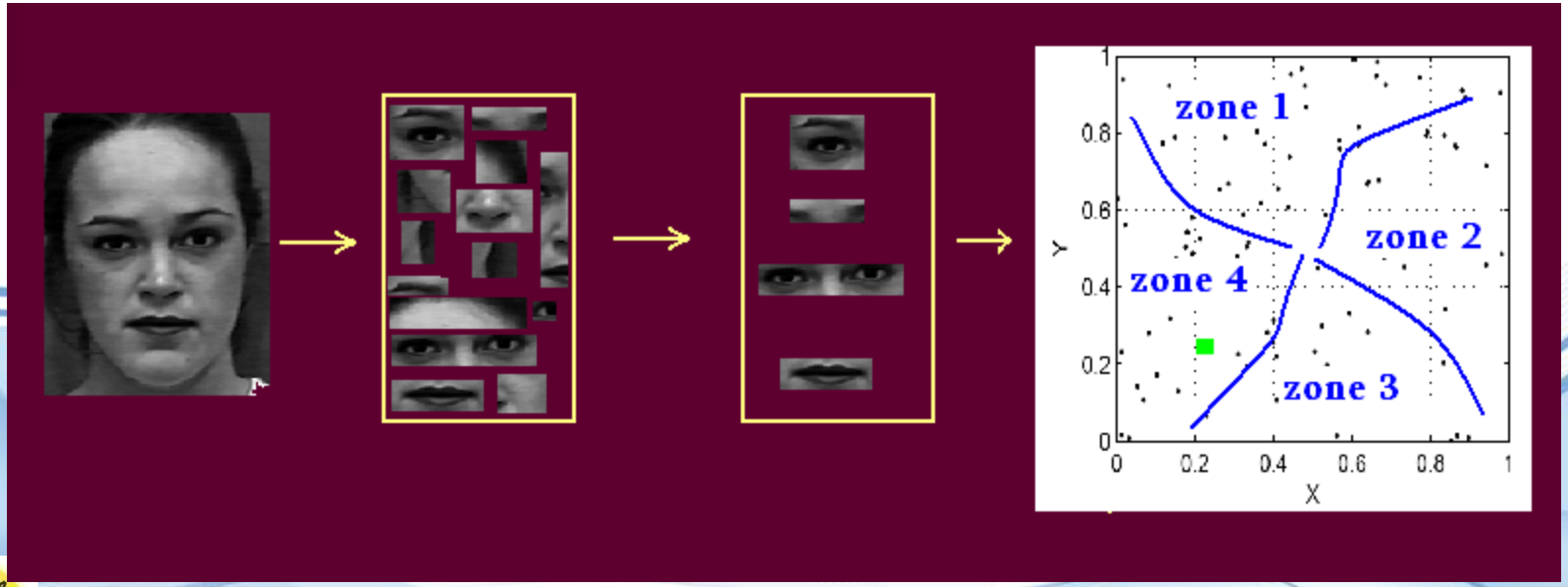
Selectia trasaturilor

- Algoritmii de extragere de trasaturi furnizeaza prea multa informatie
- Reducerea informatiei (selectia trasaturilor):
  - Metode univariate – se considera o singura trasatura la un moment dat
    - Ordonarea trasaturilor
    - ADABoost
  - Metode multi-variate – mai multe trasaturi:
    - Metode supervizate (“wrapper methods”)
    - Metode ne-supervizate (“filter methods”):
      - Analiza pe Componente Principale
      - Analiza pe Componente Independente

Problematica clasificarii

Etape:

- Extragere de trasaturi semnificative problemei
- Reducerea numarului trasaturilor
- Clasificare



Clasificarea

- Tipuri de metode de clasificare :
  - Dupa control:
    - Nesupervizata
    - Supervizata
  - Dupa abordare:
    - Probabilistica – maximizam o probabilitate:
      - Maximum Likelihood
      - Inferenta Bayesiana
    - Minimizam o functie cost
      - Retele neurale
      - SVM - **RVM**
      - Ansambluri de arbori
      - Etc.

[Fick]

Clasificare nesupervizata

- Partiţioneaza un set de obiecte în grupuri de obiecte similare. Fiecare grup se numeşte cluster (ciorchine)
- Clustering este folosit pentru a "detecta grupuri" într-un set de date.
- Algoritmii de clustering se bazează pe o funcţie de potrivire construita cu ajutorul unei distanţe
  - de obicei se încearcă să se creeze clustere compacte (stranse)



[Fick]



Arbore de Clasificare/Regresie

- Un model predictive care:
  - foloseste un set de reguli binare
  - calculeaza o valoare tinta
- Poate fi folosit:
  - pentru clasificare (variabile categorice: **A, B, C, D**)
  - pentru regresie (variabile continue **A.1415**)
- Definit de functia obiectiv

Constructia unui arbore

Antrenare:

- Intr-un nod:
  - Care variabila este folosita pentru comparatie?
  - Care este valoarea pragului?
- Decizia de oprire: Nod terminal sau continua divizarea?
- Atribuie nodului terminal o clasa

Constructia unui arbore de regresie

Exemplu:

- Functia obiectiv: eroarea patratica medie

$$MSE = \sum_{i=1}^{N} \left(Y_i - [Y_i] \right)^2$$

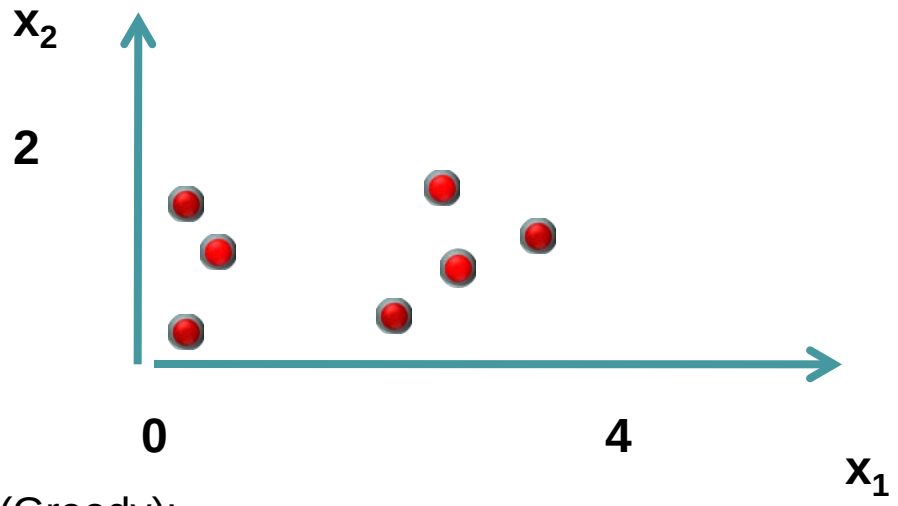
Y_i valoarea asteptata (data prin baza de antrenament)

$[Y_i]$ Valoarea prezisa de arbore

Arbore de regresie - exemplu

Set de antrenament

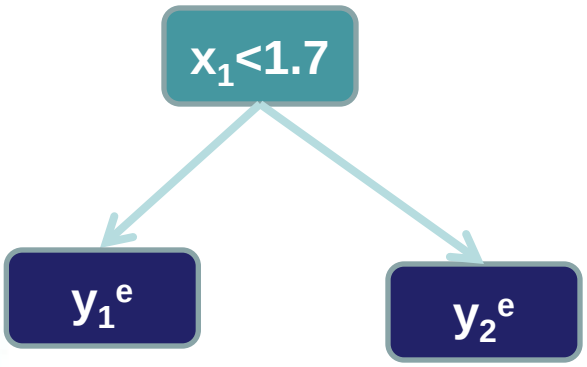
Obj	x_1	x_2	y
X_1	0.14	1.6	0.23
X_2	3.7	1.4	1.90
X_3	2.4	0.6	3.56
X_N	0.15	0.87	1.5

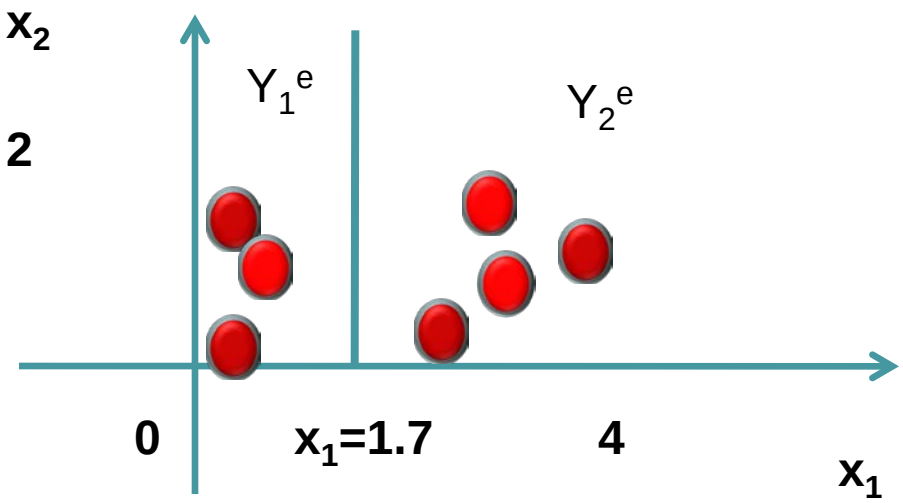


SPLIT (Greedy):
MinMSE = RealMAX
For each dimension d = $x_1 \ldots x_2$
For val = min($d_1 \ldots d_{N-1}$): max($d_1 \ldots d_{N-1}$
Split between val_{d_i} and val_{d_i+1}
Predicted value = mean of values in split
Compute MSE. If less than MinMSE, store
end
End
Use the dimension and val that lead to MinMSE

Arbore de regresie - exemplu

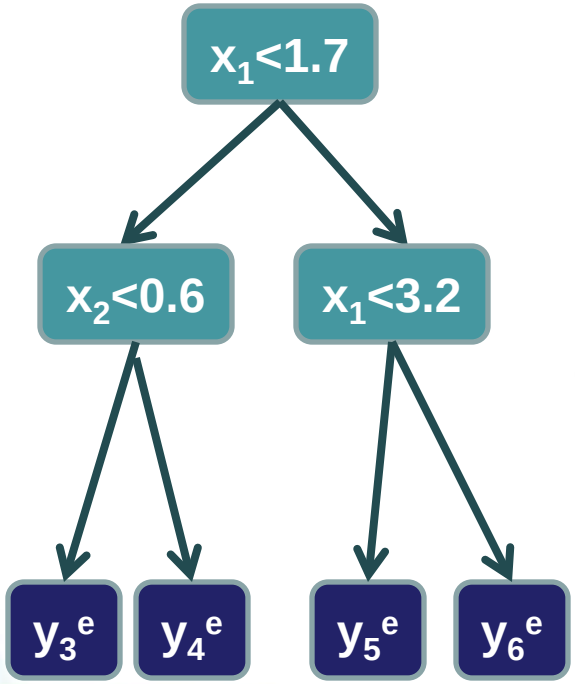
Y_1^e = media lui $Y_{1:3}$
Y_2^e = media lui $Y_{4:7}$

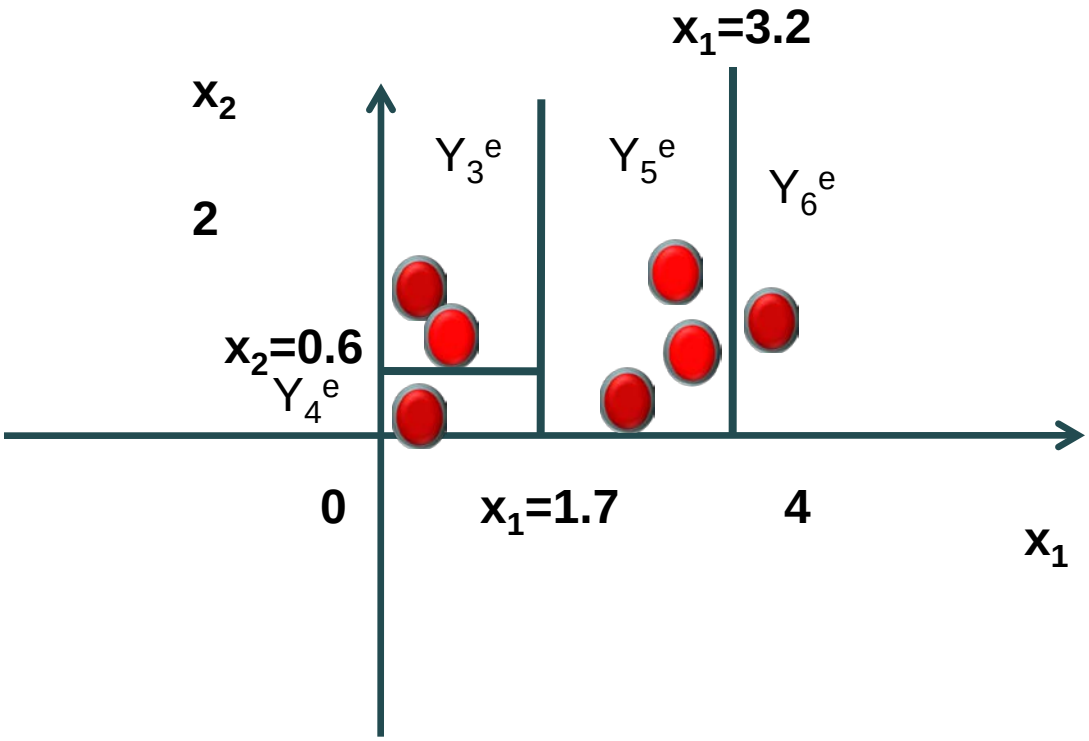




$$MSE = \frac{1}{7}\left(\sum_{i=1}^{3} (y_i - y_1^e)^2 + \sum_{i=4}^{7} (y_i - y_2^e)^2 \right)$$

Arbore de regresie - exemplu





$$MSE = \frac{1}{7}\left(\sum_{i=N_1}^{N_2-1}\left(y_i - y_3^e\right)^2 + \sum_{i=N_2}^{N_3-1}\left(y_i - y_4^e\right)^2 + \sum_{i=N_3}^{N_4-1}\left(y_i - y_5^e\right)^2 + \sum_{i=N_4}^{N}\left(y_i - y_6^e\right)^2 \right)$$

Arbore de regresie - exemplu

- Cand ne oprim?
  - MSE < ?
    - Overfitting vs . Generalization

    (Suprainvatare vs Generalizare)
  - Max Tree depth
    - Overfitting vs . Generalization

Clasificare

- Masuri obiectiv:
- Index GINI (index de impuritate)

$$GINI(X) = 1 - \sum_{i=1}^{N} (p_i)^2$$

- P_i frecventa relative a clasei i in X

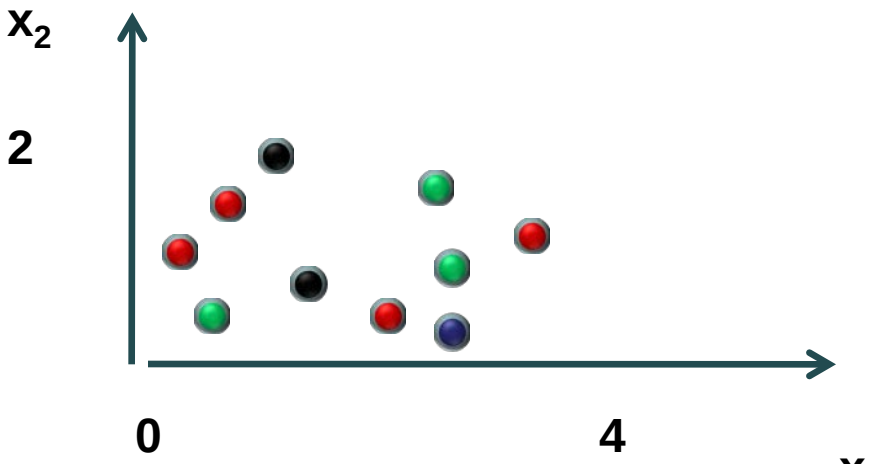
X – set de date

- Valorile mai bune pentru GINI sunt cele mai mici
- Masoara dezechilibrul social

Arbore de clasificare

Date de antrenament

Obj	x_1	x_2	y
X_1	0.14	1.6	3
X_2	3.7	1.4	1
X_3	2.4	0.6	2
X_N	0.15	0.87	3

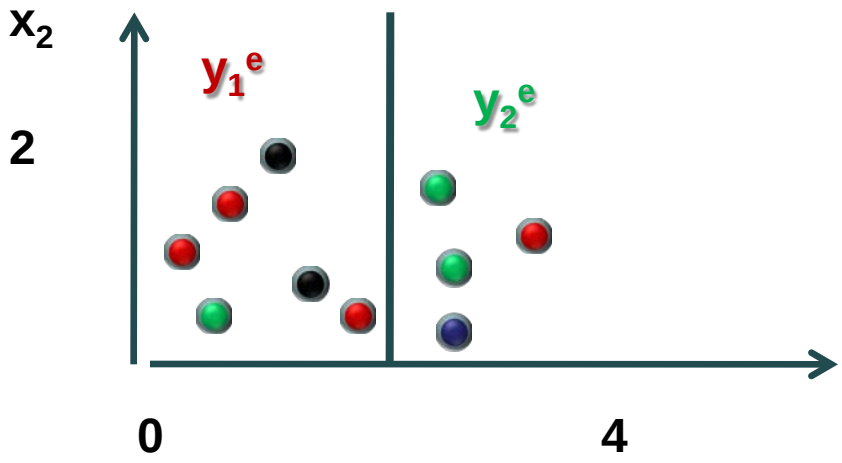


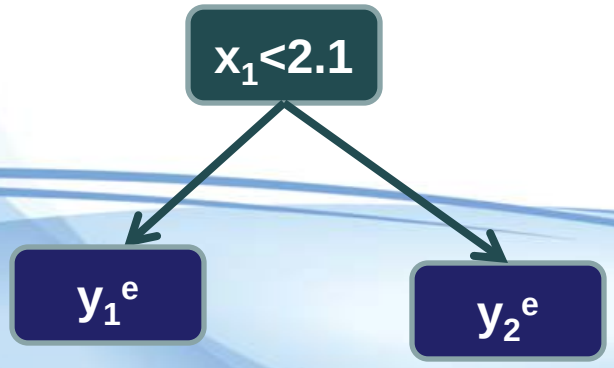
SPLIT (Greedy):
MinGINI = RealMAX
For each dimension d = $x_1 \ldots x_2$
For val = min($d_1 \ldots d_{N-1}$): max($d_1 \ldots d_{N-1}$
Split between val_{d_i} and val_{d_i+1}
Subset value = the **majority** of values in subset
Compute GINI. If less than MinGINI, store
end
End
Use the dimension and val that lead to MinGINI

Arbore de clasificare

Y_1^e = majoritatea lui $Y_{1:6}$

Y_2^e = majoritatea lui $Y_{7:10}$





$$GINI_{split}(X) = \frac{N_1}{N} GINI(X_1) + \frac{N_2}{N} GINI(X_2)$$

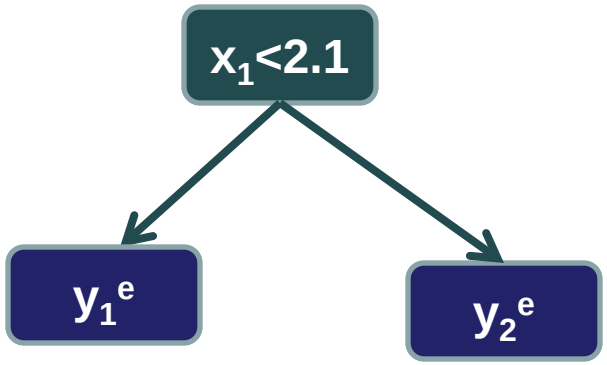
X – full dataset

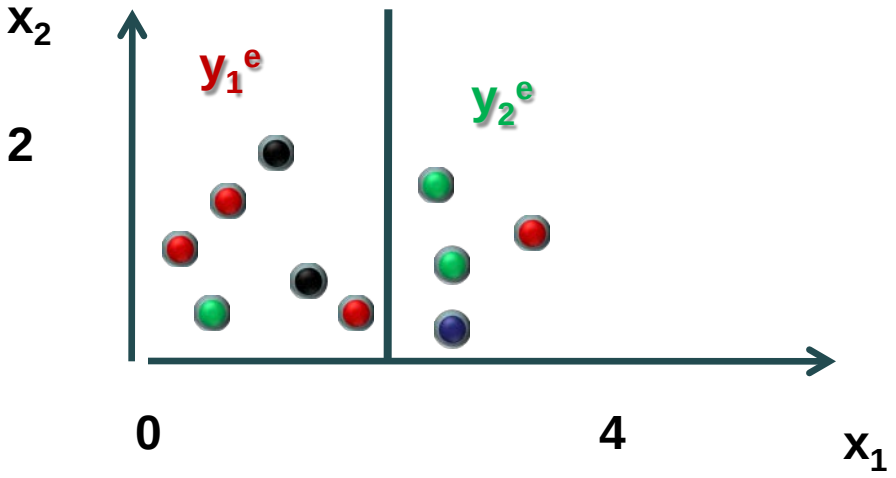
X_1 , X_2 subsets after split

Arbore de clasificare

Y_1^e = majoritatea lui $Y_{1:6}$

Y_2^e = majoritatea lui $Y_{7:10}$



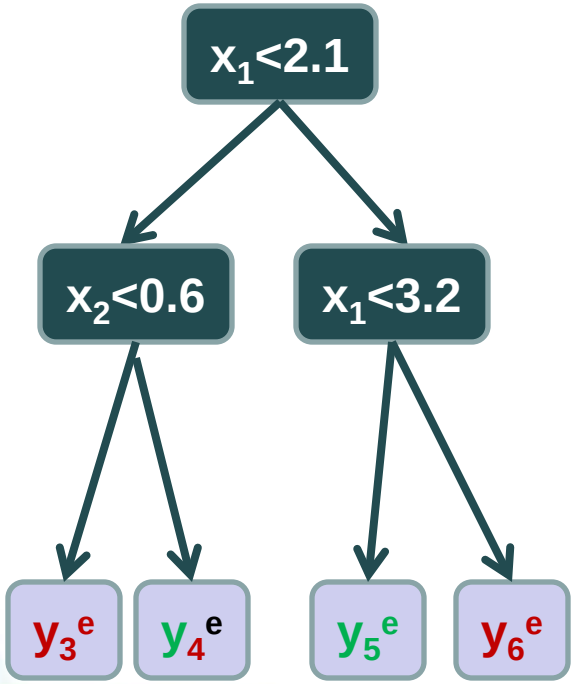


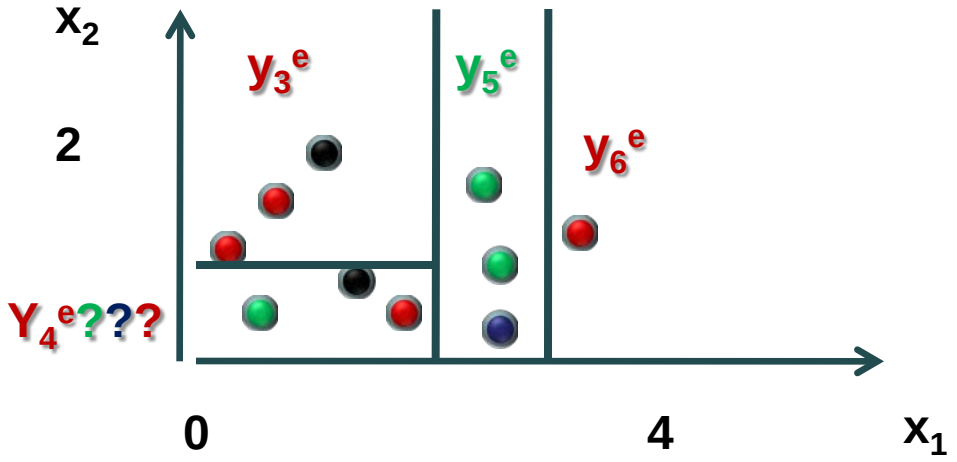
$$GINI_{split}(X) = \frac{6}{10} GINI(X_1) + \frac{4}{10} GINI(X_2) = 0.385 + 0.25 = 0.635$$

$$GINI(X_1) = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{2}{6}\right)^2 - \left(\frac{1}{6}\right)^2 = 1 - 0.3589 = 0.6411$$

$$GINI(X_2) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{1}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = 1 - 0.375 = 0.625$$

Arbore de clasificare





$$GINI_{split}(X_1) = \frac{3}{6} GINI(X_3) + \frac{3}{6} GINI(X_4) = 0.56125$$

$$GINI(X_3) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = 1 - 0.4356 - 0.1089 = 0.4555$$

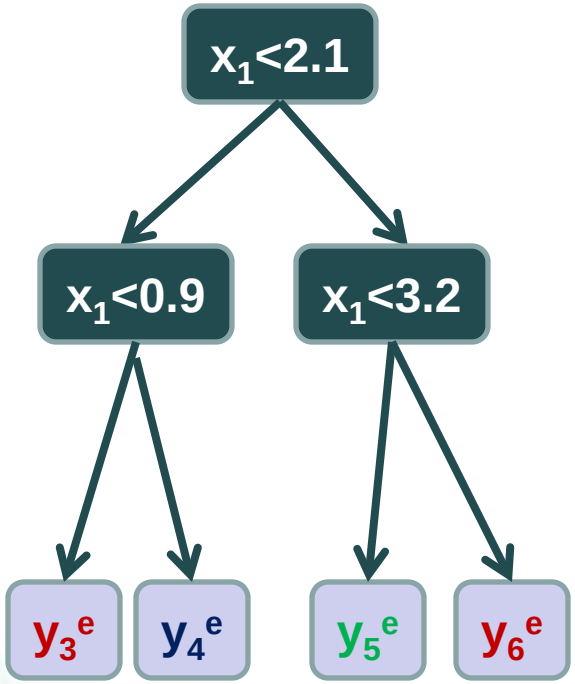
$$GINI(X_4) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = 0.667$$

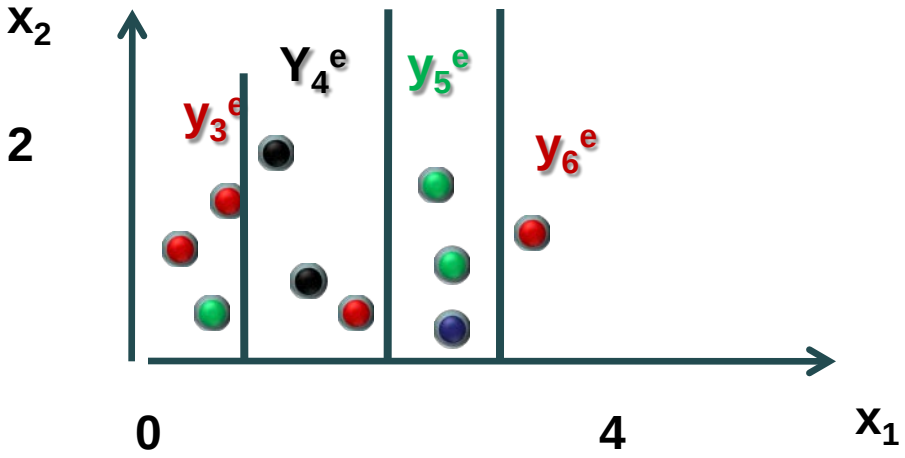
$$GINI_{split}(X_2) = \frac{3}{4} GINI(X_5) + \frac{1}{4} GINI(X_6) = 0.3415$$

$$GINI(X_5) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = 0.4555$$

$$GINI(X_6) = 1 - (1)^2 - (0)^2 = 0$$

Arbore de clasificare





$$GINI_{split}(X_1) = \frac{3}{6} GINI(X_3) + \frac{3}{6} GINI(X_4) = 0.2275$$

$$GINI(X_3) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = 1 - 0.4356 - 0.1089 = 0.4555$$

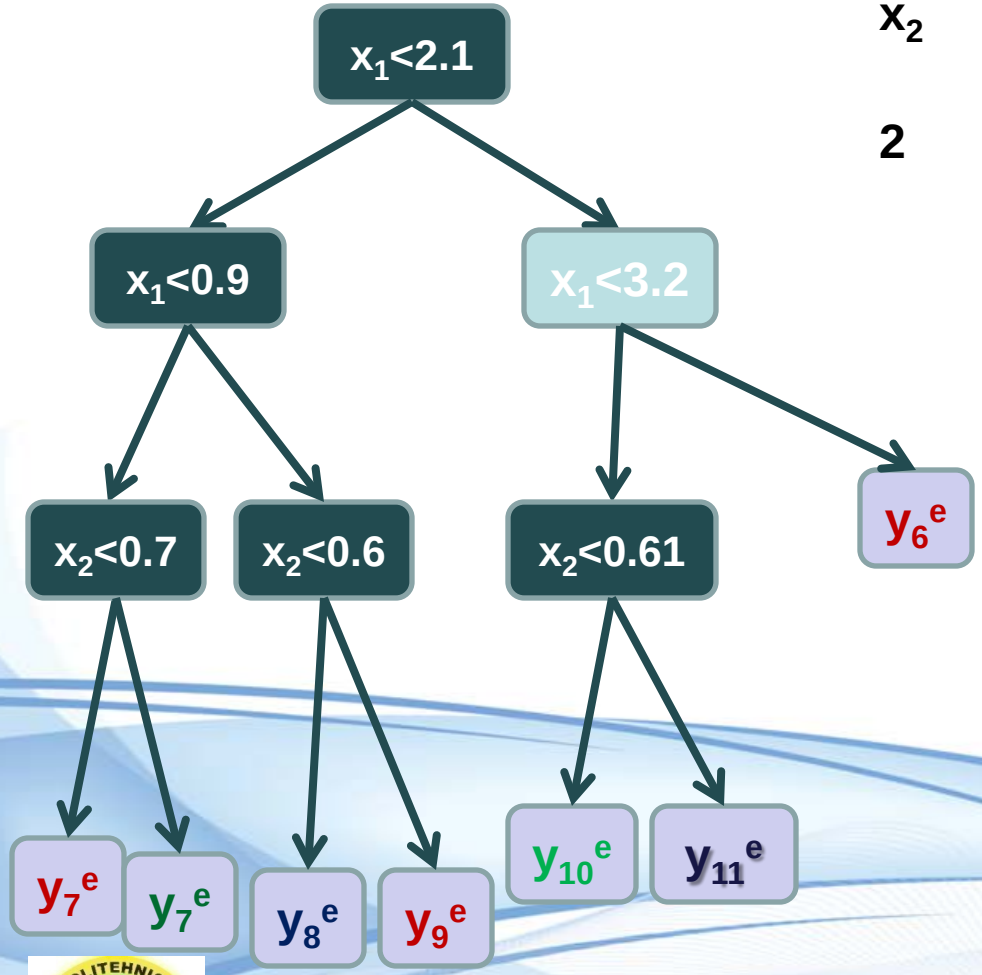
$$GINI(X_4) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.4555$$

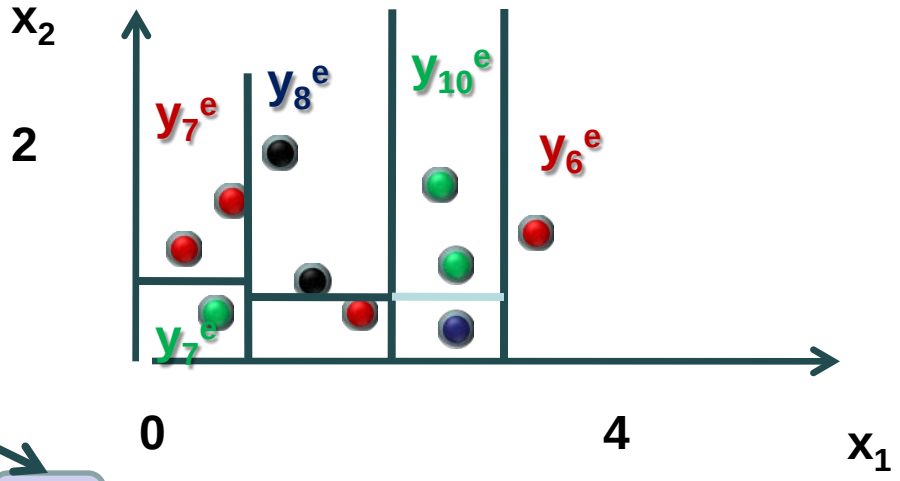
$$GINI_{split}(X_1) = \frac{3}{4} GINI(X_5) + \frac{1}{4} GINI(X_6) = 0.3415$$

$$GINI(X_5) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = 0.4555$$

$$GINI(X_6) = 1 - (1)^2 - (0)^2 = 0$$

Arbore de clasificare

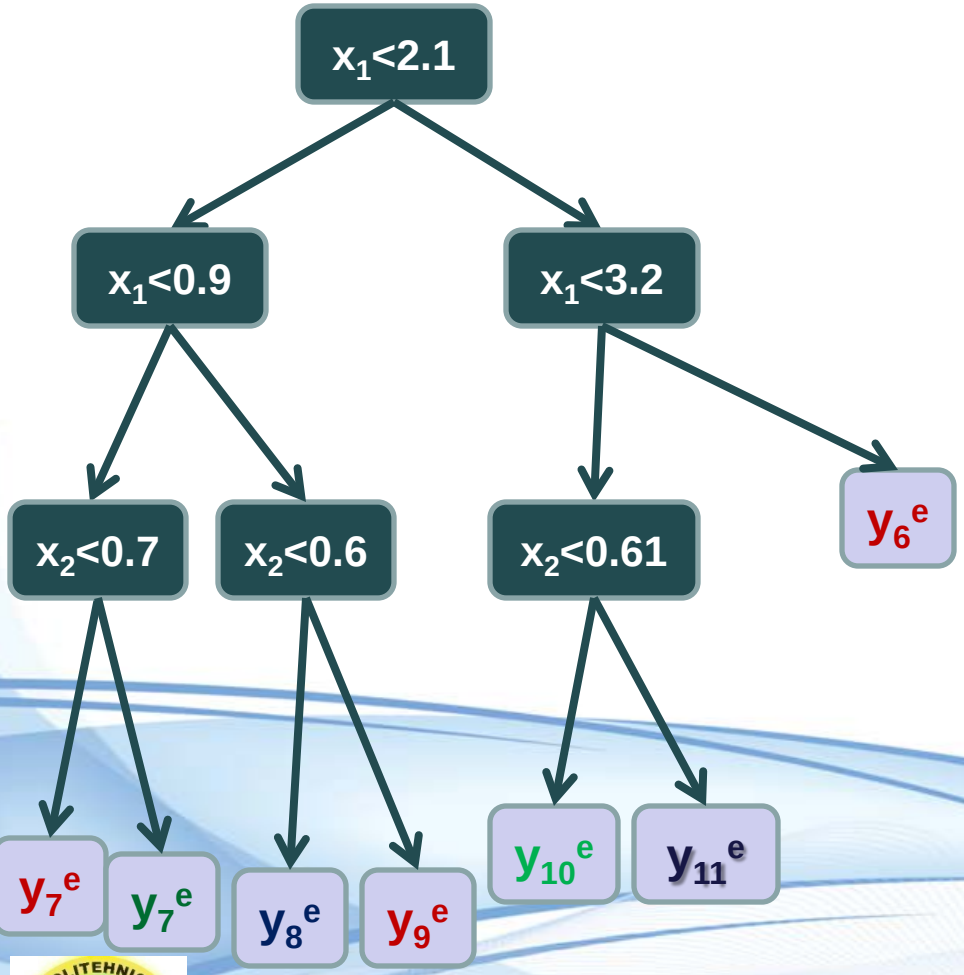


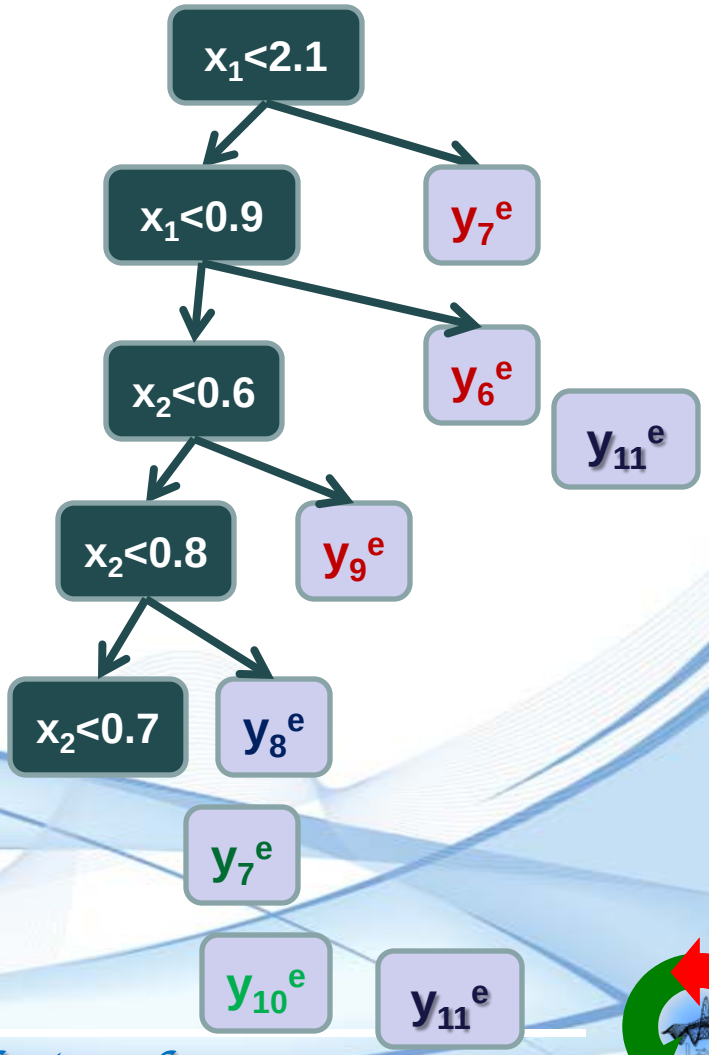


$$GINI_{split}(X_3) = 0$$
$$GINI_{split}(X_4) = 0$$
$$GINI_{split}(X_5) = 0$$

Un nod pe clasa !!!

Arbori buni versus arbori slabi





Ansamblu de arbori

- Folosesc mai multi arbori
  - Sisteme foarte puternice
- **Bootstrapping**:
  - Alege f= ?[%] din setul de antrenament pentru a construe un arbore
  - Esantionare (alegere) cu inlocuire
  - Repeta pentru N arbori

Tree ensemble

- Bagging
  - Bootstrapping and aggregating
  - Classifier: the majority of votes
  - Regression : mean
- Boosting:
  - Weighted sum of tree result

Random forest

Ansamblu de arbori

SPLIT (Greedy):
MinGINI = RealMAX
For **each** dimension d = $x_1 \ldots x_N$
For val = min($d_1 \ldots d_{N-1}$): max($d_1 \ldots d_{N-1}$)
Split between val_{d_i} and val_{d_i+1}
Subset value = the majority of values
Compute GINI.
If less then MinGINI, store
end
End
Use dimension and val that lead to MinGINI

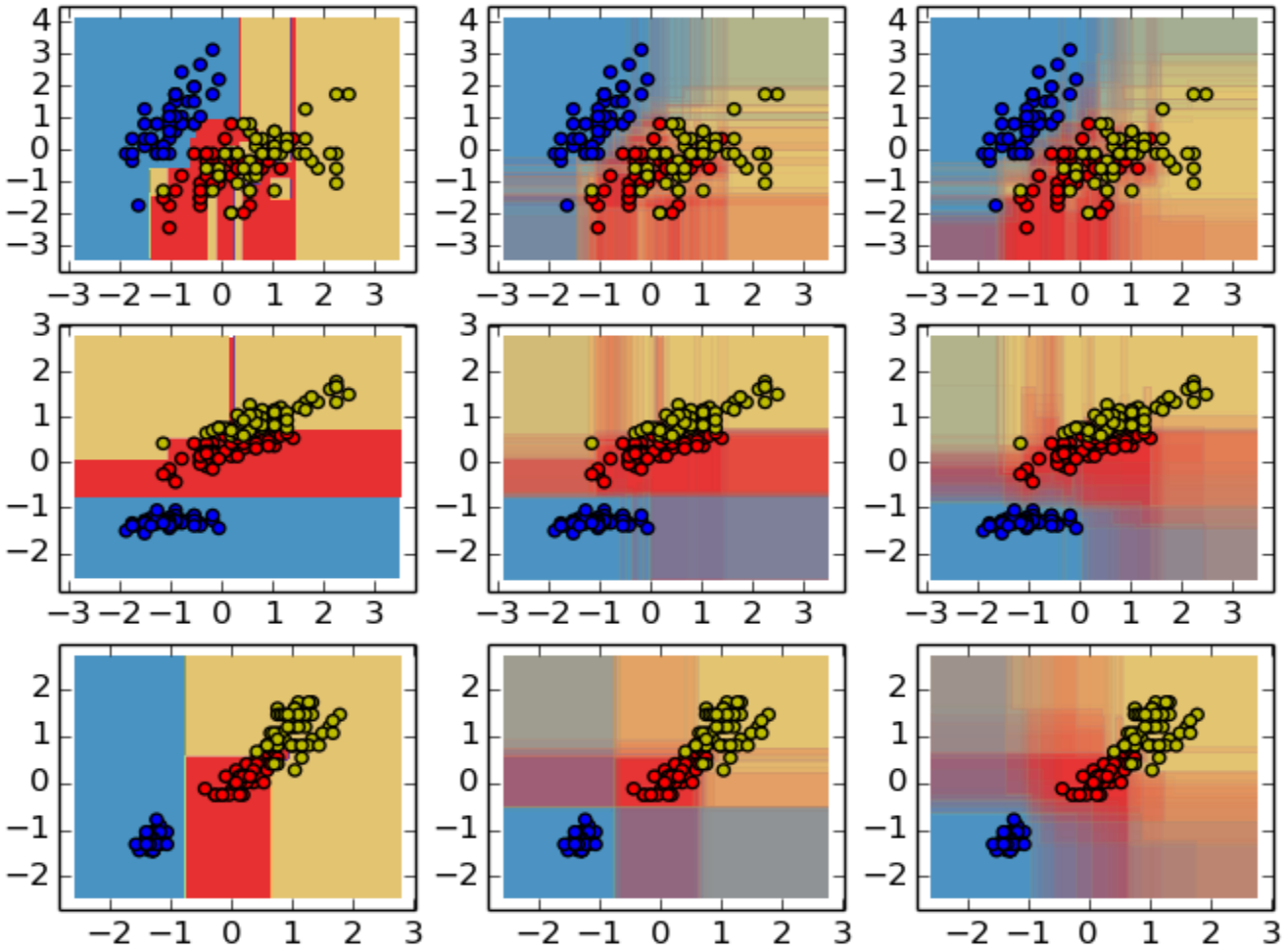
Ansamblu de arbori standard

SPLIT (Greedy):
MinGINI = RealMAX
For **randomly selected N_1** dimensions from $x_1 \ldots x_N$
For val = min($d_1 \ldots d_{N-1}$): max($d_1 \ldots d_{N-1}$)
Split between val_{d_i} and val_{d_i+1}
Subset value = the majority of values
Compute GINI.
If less then MinGINI, store
end
End
Use the dimension and val that lead to MinGINI

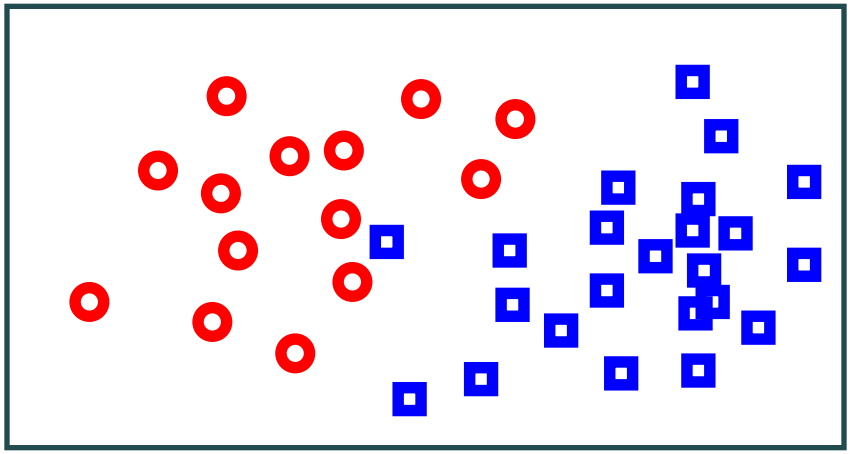
Random Forest

Rezultatul random forest

Decision surfaces of a decision tree, of a random forest, and of an extra-trees classifier



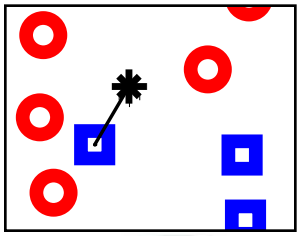
Regula “Cel mai apropiat vecin”



Sa consideram problema partitionarii in doua clase, unde fiecare esantion (data) are doua componente (*x*,*y*).

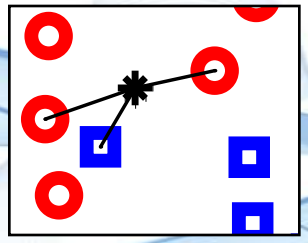
Un punct **q** va fi atribuit in clasa celui mai apropiat (distanta minima) vecin.

k = 1



Se determina cei mai apropiati **k** vecini iar clasa se atribuie ca fiind clasa majoritatii

k = 3



Problema: Metoda necesita o functie buna de distanta

[Fick]

Regula “Cel mai apropiat vecin”

Cost computational mare:

- Pentru a determina cel mai apropiat vecin al unui punct de interes *q*, trebuie să calculam distanţa până la toate cele *N* exemple din setul de antrenare

- Putem accelera procesul prin:

  - Pre-sortarea exemplelor de antrenare în structurile logice – arbori de cautare
  - Calculam doar o distanţă aproximativă
  - Elimina datele redundante (condensare)



[Fick]



Regula “Cel mai apropiat vecin”

Stocarea:

- Trebuiesc stocate toate datele din setul de antrenament
  - Reducem memoria: Eliminam datele redundante
  - Crestem memoria: Arbori de sortare

Date multi-dimensionale:

- Numarul de pasi de antrenare creste exponenţial cu dimensiunea
- Costul de calcul creste

Tehnici de reducere a setului de antrenare / referinta



[Fick]



Algoritmul RSC

Algoritmul RSC (Representative supervised clustering):

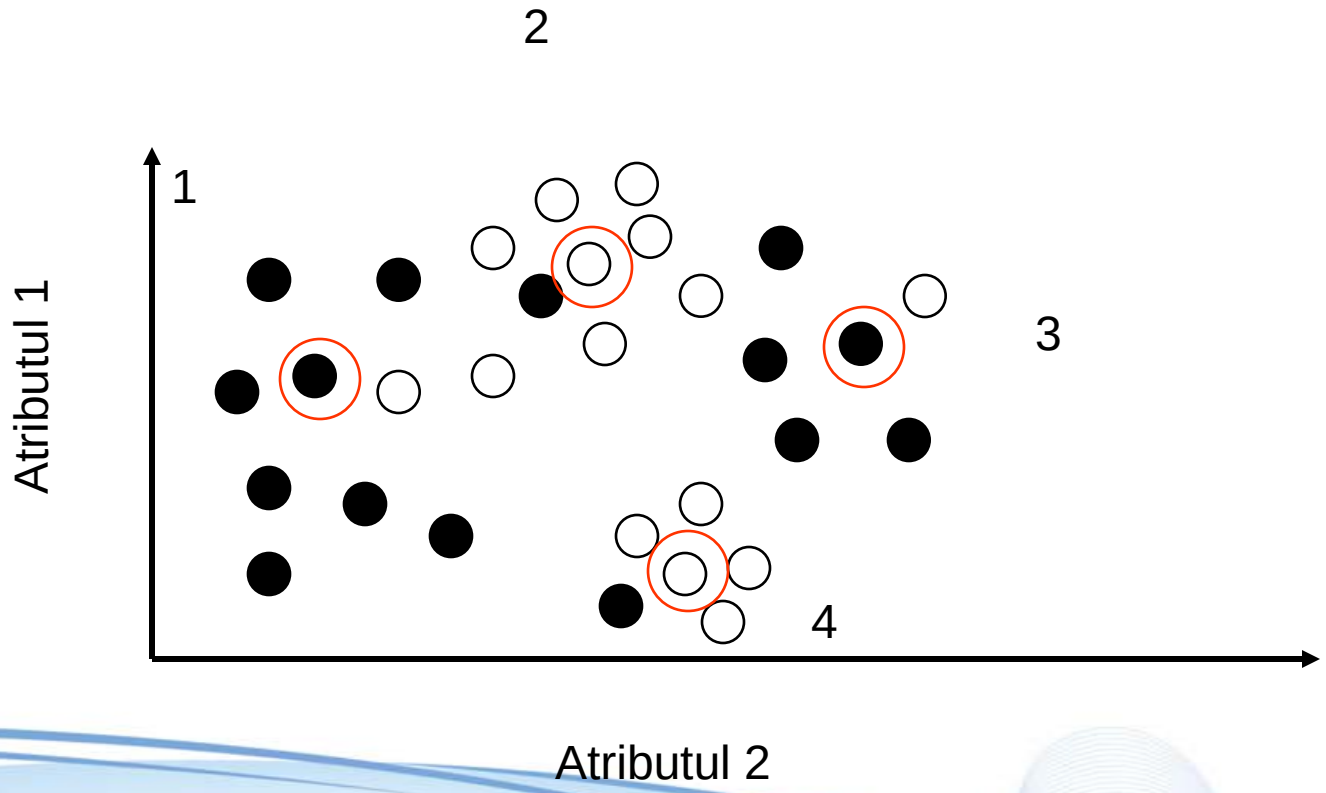
- Are ca scop gasirea unui grup redus de obiecte (numite **reprezentanti**) dintre toate obiectele din setul de date existent care sa reprezinte cat mai bine setul din care provine
  - Fiecare reprezentant va corespunde unei clase

- Obiectele care raman in setul de date vor fi asignate cluster-ului de al carui raeprezentant sunt cele mai apropiate.

Observatie: Algoritmul *K-means (ISODATA)* este un algoritm de tip RSC.

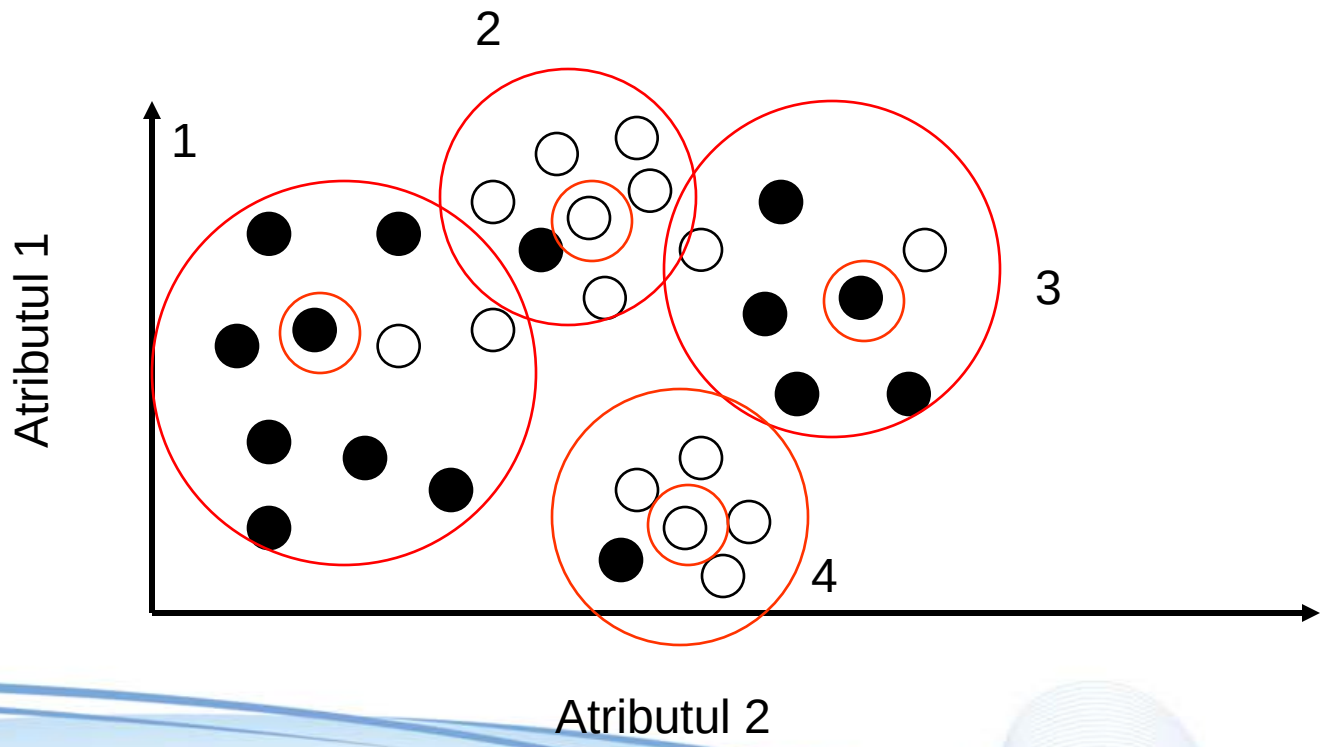
[Fick]

Reprezentanti



[Fick]

Alegerea reprezentantilor



Scopul RSC: Determinarea unui subset O_R al lui O astfel incat clasificarea datelor utilizand obiectele din O_R ca reprezentanti sa minimizeze o functie de cost *q(X).*

[Fick]

Algoritmul Greedy Hill Climbing

REPETA de *r* ORI

Repr := un set aleator de reprezentanti

CAT TIMP NU S-A TERMINAT, EXECUTA

1. Creaza o noua solutie *S* prin adaugarea unui singur nou membru in setul de reprezentanti in *Repr* si elimina un reprezentant din *Repr*
2. Determina elementul s din S pentru care functia de cost *q(s)* este minima (daca sunt mai multe alege aleator una)
3. DACA *q(s)<q(Repr*) ATUNCI *Repr:=s*
   ALTFEL DACA *q(s)=q(Repr)* SI |*s*|>|*Repr*| ATUNCI Repr:=s
   ALTFEL termina si intorce *Repr* ca solutie pentru aceasta incercare.

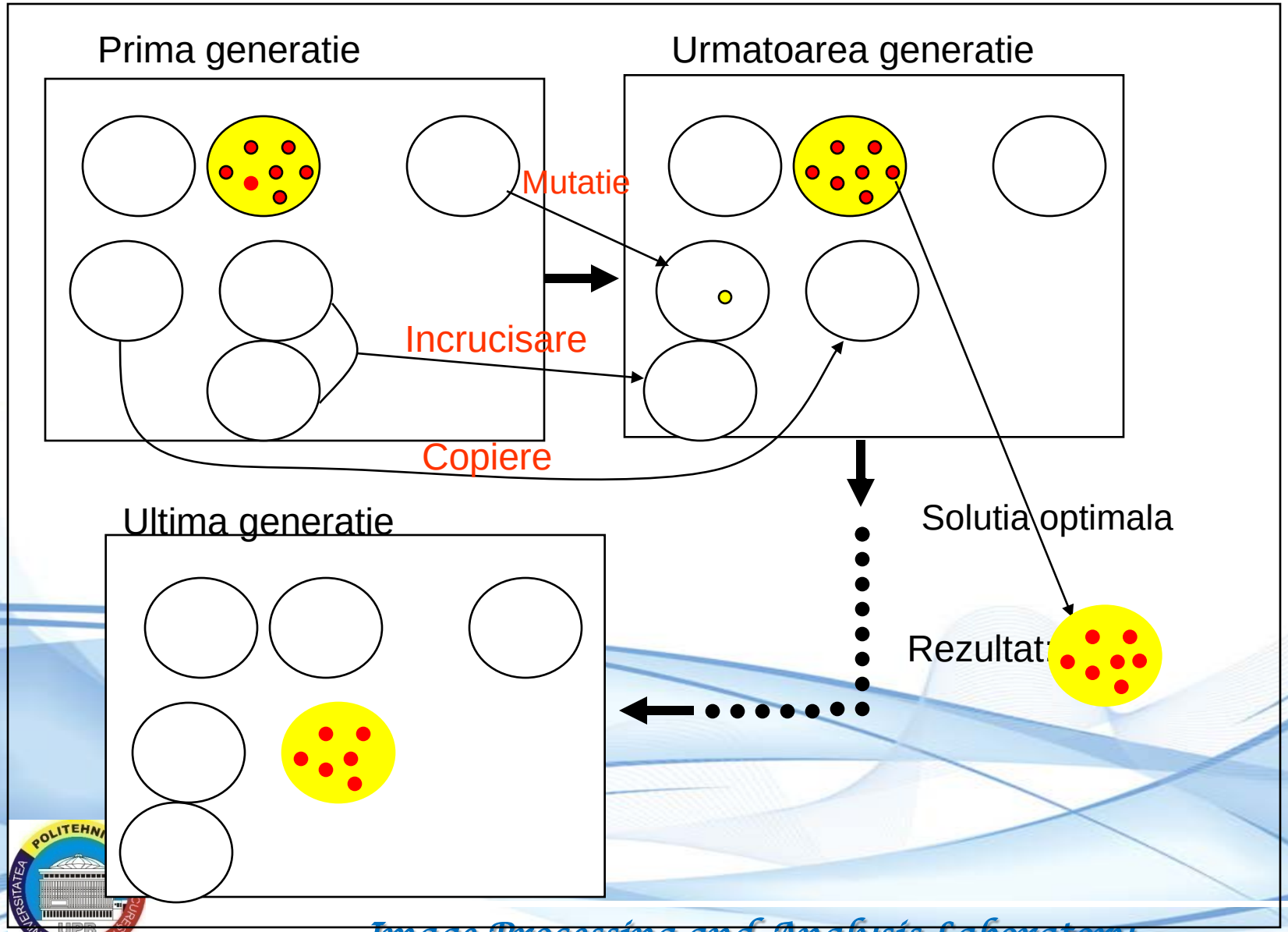
Raporteza cea mai buna solutie obtinuta in urma celor *r* incercari.

Observatie:

- Rezultatul este cea mai buna solutie obtinuta in ***r*** incercari

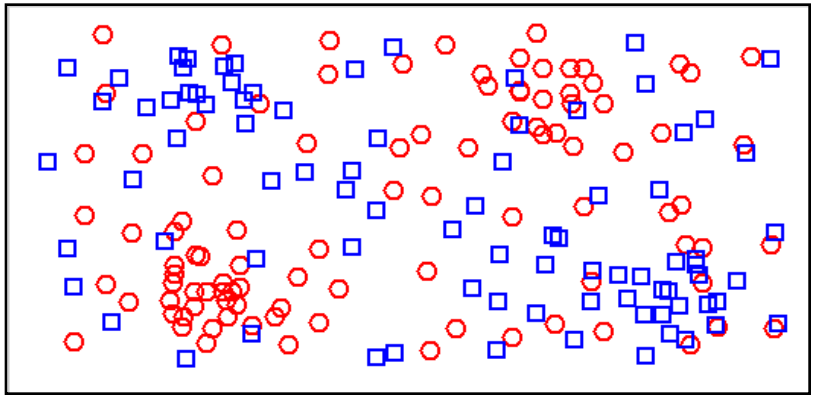
[Fick]

Calcul evolutional



[Fick]

Reducerea setului de date - editarea



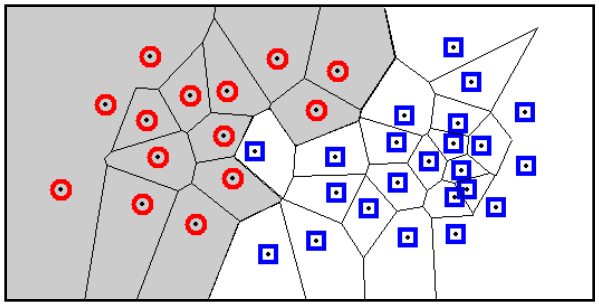
- Datele de antrenare pot conţine puncte de zgomot sau clase suprapuse
- Editarea : eliminarea punctelor de zgomot şi producerea de granite clare - puncte departe de limitele deciziei
- Scopul principal : sa sporeasca precizia clasificatorului
  Scopul secundar : cresterea vitezei unui clasificator k-NN

[Fick]

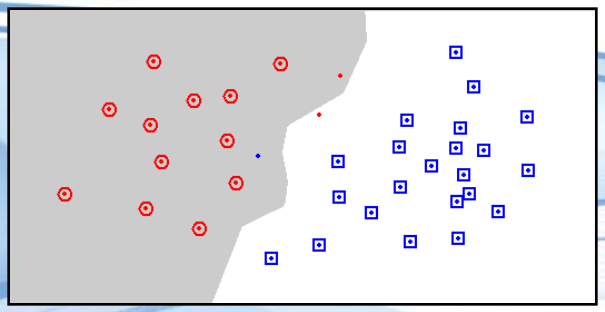
Editarea de tip Wilson

- Introdusa de Wilson in 1972 [Wilson]
- Se elimina punctele care sunt in dezacord cu majoritatea din cei mai apropiati k vecini

Exemplu simplu

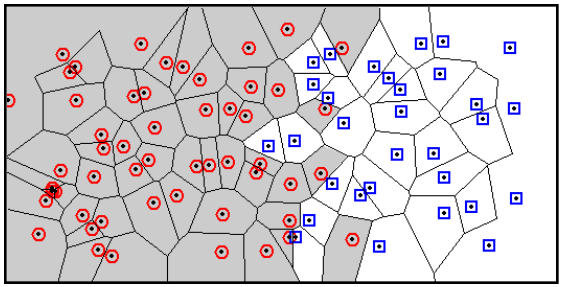


Date originale

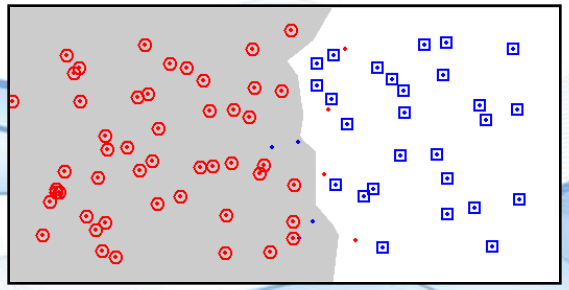


Editare Wilson cu k=7

Clase suprapuse



Date originale

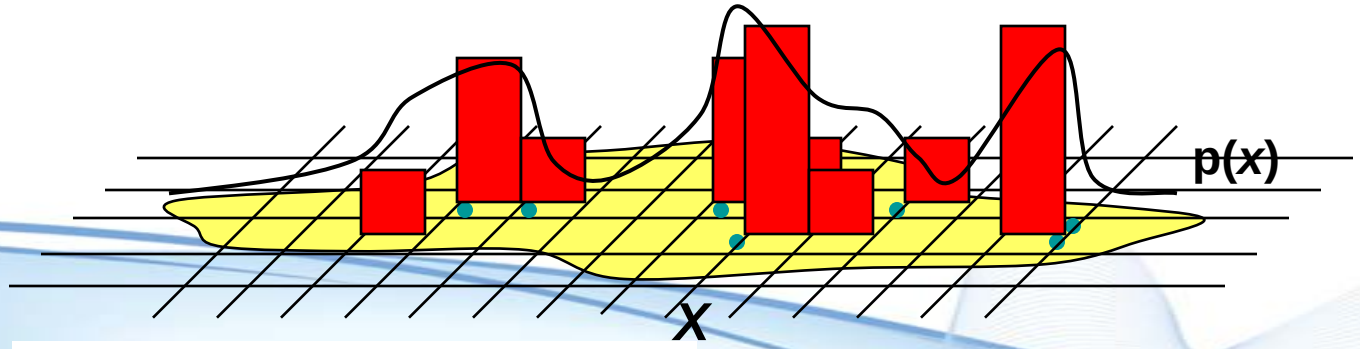


Editare Wilson cu k=7

[Fick]

Editarea de tip Monte Carlo

- Exista foarte multe date – probabil redundante
- Metoda pentru a reduce volumul datelor
- Fiind dat un set de date *X* caracterizat de o distributie p(*x*)
- Extragem N esantioane din setul de date
- Vom aproxima distributia pe baza setului de N date



$$p_N(x) = \frac{1}{N}\sum_{i=1}^{N} 1(x^{(i)} = x) \xrightarrow[N\to\infty]{} p(x)$$

An Introduction to Markov Chain Monte Carlo, Teg Grenager

Reguli compuse de calcul

- Cascade de conditii:
  - Agregate cu SI
    - Rata de detectie finala este produsul ratelor de detectie individuale
    - Utilitate: Clasificatori individuali relaxati
    - Viteza: Primele conditii sa elimine cea mai mare parte a negativelor
  - Agregate cu SAU
    - Rata de detectie este suma a ratelor de dectie individuale
    - Utilitate: Clasificatori individuali restrictivi

Estimarea performantelor

- Metoda de evaluare a performantelor trebuie sa fie parte a proiectarii clasificatorului si nu sa il preceada
  - Evaluarea comparabila in complexitate cu clasificatorul
- Aspecte:
  - Evaluarea performantelor
  - Imbunatatirea performantelor

Evaluarea performantelor

- Aspecte care trebuie avute in vedere cand se construieste o regula de clasificare:
  - Puterea de discriminare (cat de bine separa datele invizibile)
  - Exactitate (cat de precis estimeaza apartenenta rezultatului)
  - Receiver operating characteristic (ROC) trebuie avut in vedere pentru clasificatorii binari

Discriminarea

- De obicei se masoara prin rata de eroare (de clasificare incorecta):
  - **Aparenta** – numarul de vectori din setul de antrenament clasificati incorect

$$e_A = \frac{1}{N}\sum_{i=1}^{N} Q(v_i) \quad Q(v) = \begin{cases} 1 & \text{Clasificat gresit} \\ 0 & \text{Clasificat corect} \end{cases}$$

Dimensiunea setului de antrenament

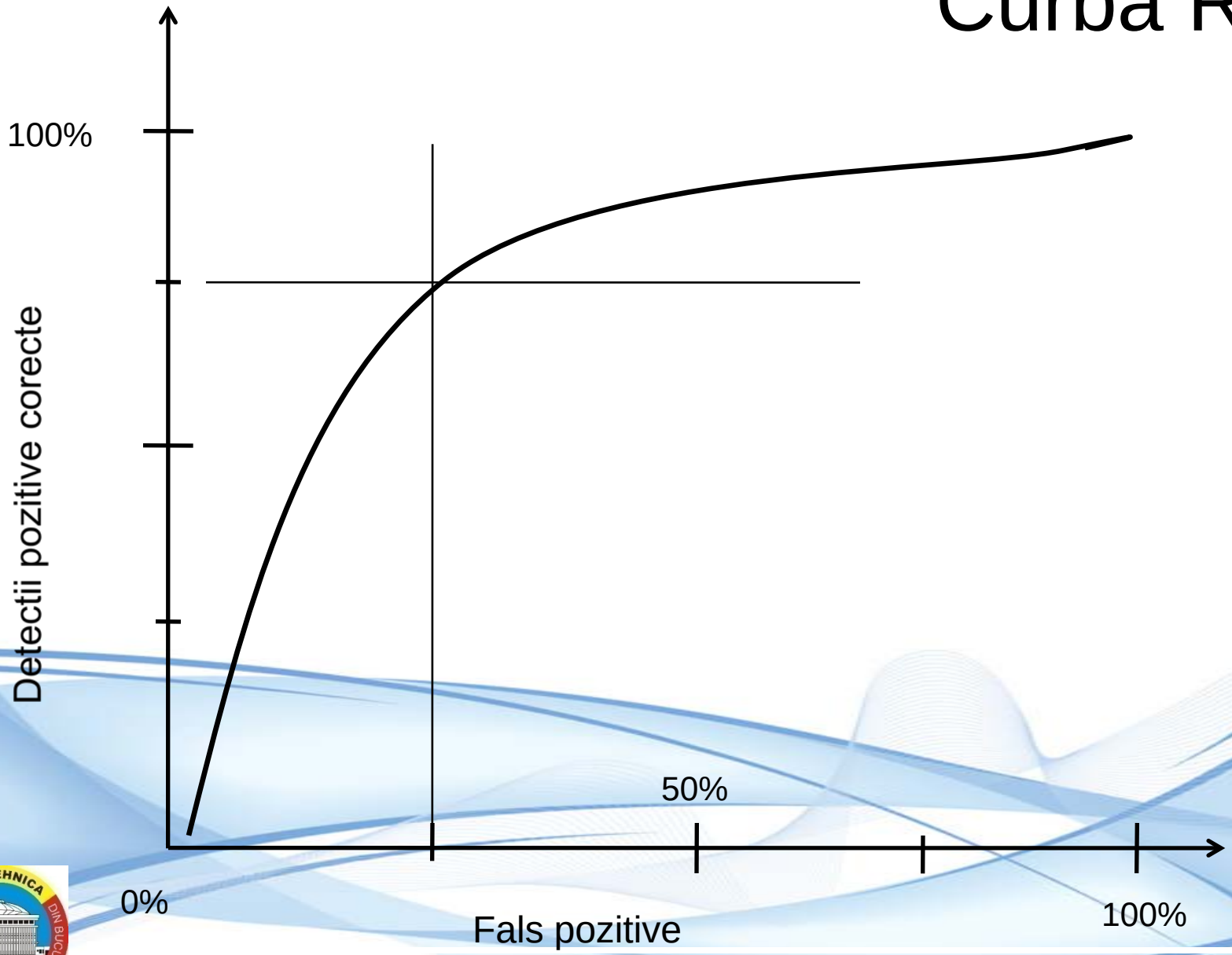
Vectorul de iesire

  - **Reala** – eroarea aparenta calculata pe un set infinit

Curba ROC

- In contextul unui clasificator binar se definesc:
  - Detectii corecte (true positive) – indivizi **facand parte** din clasa tinta identificati **corect**
  - Falsuri pozitive – indivizi care **nu fac** parte din clasa tinta dar au fost identificati **incorect ca facand** parte din ea
  - Falsuri negative – indivizi care **fac** parte din clasa tinta dar au fost ratati si identificati incorect ca facand parte din complementul ei
- Curba Receiver operating characteristic (**ROC**) are pe o axa valorile ratei de detectie (adevarat pozitive) si pe celalta falsurile positive
  - Praguri relaxate – rata de dectie 100% , dar totul e in clasa tinta
  - Praguri stranse – nu scapa nici un fals, dar nici nu se detecteaza nimic

Curba ROC



Referinţe bibliografice

Analiza pe Componente Independente:

- [Agrawal] Amit Agrawal “Independent Component Analysis (ICA) and Factor Analysis (FA)”, ENEE Seminar, Sept 23 www.umiacs.umd.edu/~shaohua/enee698a_f03/ica_fa.ppt
- [Hyvarinen] A. Hyvarinen, E. Oja, Independent Component Analysis: A tutorial, April 1999 (http://www.cis.hut.fi/projects/ica/)

Clasificare:

- [Fick] Christoph F. Eick, “Supervised Clustering - Algorithms and Applications” www2.cs.uh.edu/~ceick/kdd/Lafayette05.ppt
- [Wilson] Wilson, D.L. “Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. Syst. Man Cybernet. 2, pp. 408-421, 1972.

Pattern Recognition in general si evaluarea performantelor:

- [Webb] Andrew Webb, “Pattern Recognition”, John Wiley and sons, 2002

Referinţe bibliografice

Texture Unit:

- [He] He, D. and Wang, L., "*Texture unit, texture spectrum and texture analysis*," IEEE Trans. on Geoscience and Remote Sensing 28, pp.509-512 Iulie, 1990.

Local Binary Pattern:

- [Ojala1]T. Ojala et al, "*A comparative study of texture measures with classification based on feature distributions*", Pattern Recognition, vol. 29, pp. 51-59, 1996.
- [Ojala2] T. Ojala et al, "*Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 24, pp. 971-981, 2002.
- [Ahonen] Timo Ahonen et al, "*Face Recognition with Local Binary Patterns*", ECCV 2004, LNCS 3021, pp. 469-481, 2004
- [Pietikäinen] Matti Pietikäinen, "*Face Analysis Using Local Binary Patterns*" http://www.ee.oulu.fi/mvg

Histogram of Oriented Gradient:

- [Dalal] Navneet Dalal, Bill Triggs, "Histograms of Oriented Gradients for Human Detection", International Conference on Computer Vision and Pattern Recognition, Vol 2, pp. 886-893

Referinţe bibliografice

Filtre Gabor:

- [Movellan] Javier R. Movellan “Tutorial on Gabor Filters”
- [Buciu08] Ioan Buciu, I. Nafornita, I. Pitas, “Facial Expression Recognition under Noisy Environment Using Gabor Filters”, Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara, Seria Electronica si Telecomunicatii, Tom 53(67), Fascicola 1-2, 28
- [ManRo] Y. Man Ro, M. Kim, H. K. Kang, B.S. Manjunath, and J. Kim “MPEG-7 Homogeneous Texture Descriptor” IEEE Transaction on Circuits and Systems for Video Technology, vol. 11, no. 6, June 2001

Analiza pe Componente Principale:

- [Narasimhan] S. Narasimhan,” Computer Vision” – lecture 19, Columbia University http://www.cs.cmu.edu/afs/cs/academic/class/15385-s06/lectures/ppts/

.

Multumesc pentru atentie!

In ceea ce priveste intrebarile… va ascult ?