

# Prelucrarea Imaginilor

SEA - 2020

Laurențiu Frangu, Laurențiu Baicu

# Cuprins

1. Introducere, sisteme de achiziție și prelucrare a imaginilor
2. Preprocesare și îmbunătățire
3. Segmentarea și transformări morfologice
4. Extragerea trăsăturilor și măsurări pe imagine
- 5. Recunoașterea formelor**
6. Aplicații

# Recunoa terea formelor

- Abordarea determinist
- Abordarea statistic

# Recunoașterea formelor

Recunoașterea formelor	Clasificare	abordare decizional-teoretică	statistică	parametrică	statistică
				neparametrică	
			deterministă	parametrică	deterministă
				neparametrică	
		abordare structural-lingvistică			
	Descriere	abordare structural-lingvistică			

Abordări

# Clasificarea determinist

- Introducere
- Utilizarea funcției distanță
- Metoda celui mai apropiat vecin (NN)
- Metoda celor mai apropiați K vecini (KNN)
- Metoda funcțiilor de potențial
- Clasificator de distanță minimă
- Discriminantul liniar al lui Fisher
- Analiza structurii datelor

# Introducere

- Recunoaștere, clasificare
- Aplicații
- Caracteristici (proprietăți măsurabile)
- Clasă (grup de obiecte)
- Clasificator (metoda de recunoaștere sau de discriminare între clase)
- Clasificare supervizată, instruire, set de instruire, exemple antioane

# Introducere

- Aplicații frecvente
  - text
  - amprente
  - vorbitorul
  - vocea (mesajul)
  - substanțe chimice
  - parfumuri
- Caracteristici
  - descriptori de form sau de imagini
  - descriptori de form
  - componente spectrale
  - componente spectrale sau serii de timp
  - linii în spectru
  - frecvențe de rezonanță

# Introducere

- Surse de informație pentru recunoaștere
  - forma geometrică (scriere, amprente, obiectele manipulate de robot, modele grafice)
  - sunete (voce, vorbitor, zgomotele mașinilor)
  - vectori de stare ai sistemelor tehnice (degradarea izolației, rezonanță mecanică)
  - vectori de stare ai sistemelor economice (contextul economic, riscul împrumutului)
  - spectrograme (substanțe chimice)
- Abordarea vectorială : fiecare individ este reprezentat printr-un vector



# Introducere

- Clase
  - grup de vectori cu proprietăți comune
  - recunoașterea formei = recunoașterea clasei careia îi aparține individul (vectorul) observat
  - dacă recunoaștem cifrele, 10 clase (a 11-a clasă poate fi cea a simbolurilor care nu sînt cifre)
  - dacă recunoaștem persoana care a scris, atîtea clase cîtî autori posibili

# Introducere

- Clasificator
  - algoritm prin care decidem cărei clase îi aparține individul (vectorul) observat
  - divide spațiul formelor în  $c$  domenii distincte ( $c$  este numărul de clase)
  - divizarea poate fi explicit (metode parametrice) sau implicit (metode neparametrice)

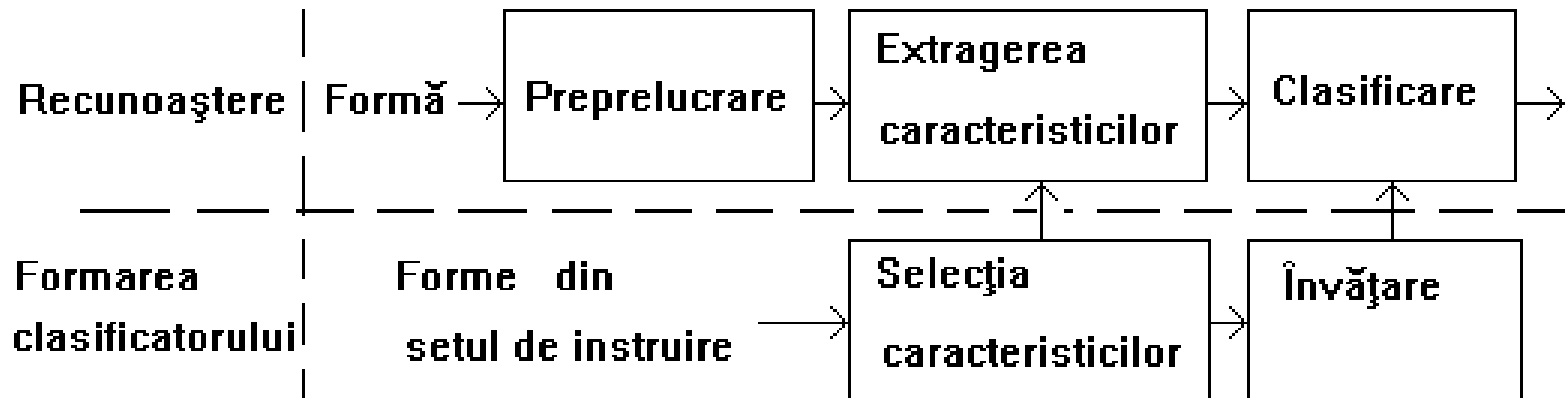
# Introducere

- Clasificare supervizat
  - cazul în care există un număr de indivizi, a căror apartenență la clase este cunoscută
  - formează setul de instruire
  - atribuirea la clase a fost făcută de un expert (instructorul)
- Instruire
  - utilizarea vectorilor din setul de instruire (exemplare) pentru alegerea algoritmului potrivit și determinarea parametrilor săi

# Introducere

- Etapele procesului de recunoaștere
  - analiza structurii datelor
  - transformarea spațiului formelor, în noile coordonate (dacă e cazul)
  - instruire (identificarea parametrică, dacă e cazul)
  - examinare (validare)
  - exploatare (clasificarea indivizilor nou observați)

# Introducere



# Introducere

- Instruire, examinare (validare), exploatare
  - este ales algoritmul de recunoaștere, din etapa de analiz
  - sînt identificați parametrii (dacă e cazul)
  - performanța clasicatorului este evaluată, în etapa de examinare, folosind o parte din setul de instruire
  - se repetă instruirea, dacă performanța este nesatisfăcătoare
  - dacă algoritmul a fost validat, poate fi folosit pentru recunoașterea indivizilor nou observați (etapa de exploatare)

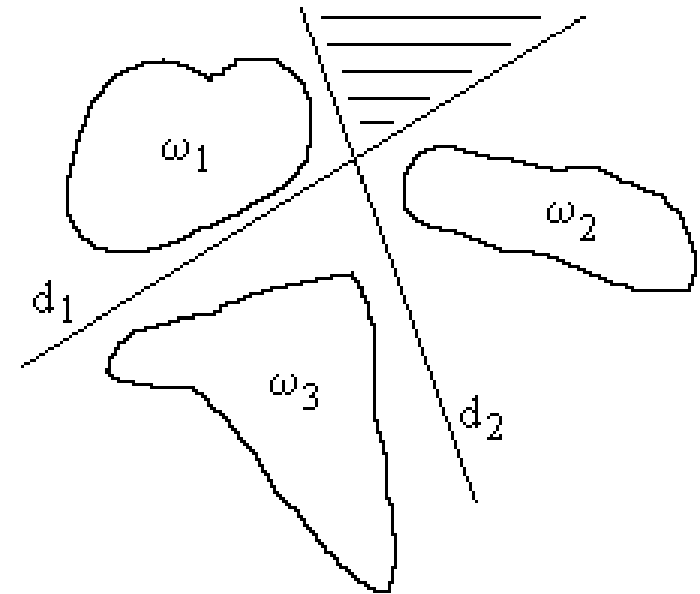
# Introducere

- Noțiunea de funcții discriminante
  - set de  $c$  funcții, al căror argument este vectorul observat
  - proprietate: funcția cu valoarea maximă decide la ce clasă este atribuit vectorul
- Exemplu  $f_i(x)$   $i = 1..c$   
decizie în favoarea clasei  $i$ , dac  $f_i(x) = \max_j f_j(x)$

# Introducere

- Noțiunea de frontieră
  - set de funcții, al căror argument este vectorul observat
  - proprietate: separă domeniile a două clase adiacente

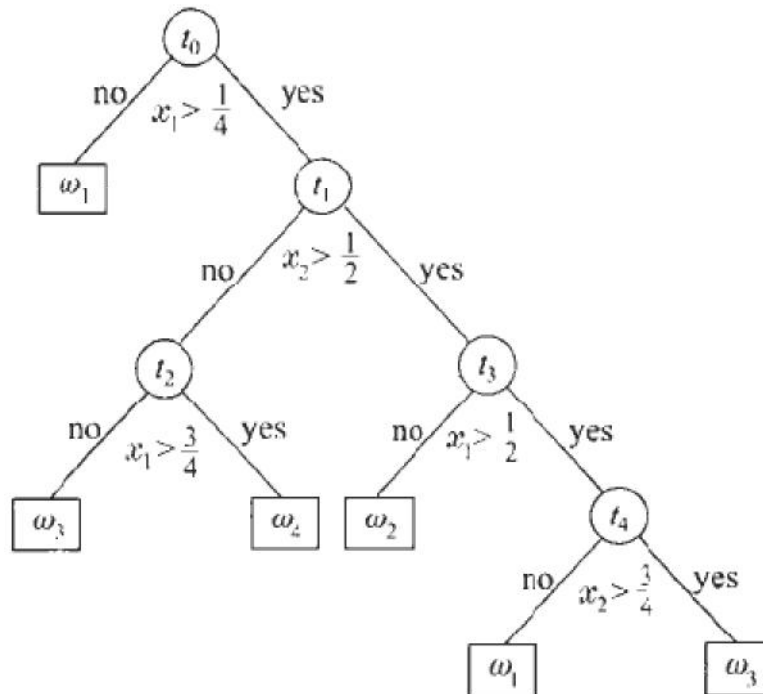
- Exemplu  $g_{ij}(x) = 0$ 
  - dac  $g_{ij}(x) > 0 \rightarrow x \in \check{S}_i$
  - dac  $g_{ij}(x) < 0 \rightarrow x \in \check{S}_j$





# Introducere

- Noțiunea de arbore de decizie
  - mai mulți pași pentru decizie, criterii diferite



# Introducere

- Estimarea parametrilor clasificatorului
  - numai pentru clasificatori parametrici
  - informația prelucrată din setul de instruire
  - după validarea algoritmului, setul de parametri va fi folosit pentru recunoașterea indivizilor nou observați (etapa de exploatare)

# Funcții distanță

- Proprietăți:
- $f(x, y) \geq 0$  pentru orice pereche  $x, y$  din  $R^d$
- $f(x, x) = 0$  pentru orice  $x$  din  $R^d$
- $f(x, y) = f(y, x)$  pentru orice pereche  $x, y$  din  $R^d$
- $f(x, y) + f(y, z) \geq f(x, z)$  pentru orice triplet  $x, y, z$  din  $R^d$

# Funcții distanță

- Exemple:
  - Euclidian
  - Minkowsky
  - Manhattan (*city block*)
  - Max
  - Hamming

# Funcții distanță

- Exemple:
  - Minkowsky

$$f(x, y) = \left( \sum_{i=1}^d |x_i - y_i|^n \right)^{\frac{1}{n}}$$

# Funcții distanță

- Exemple:
  - Minkowsky
  - Euclidian

$$f(x, y) = \left( \sum_{i=1}^d |x_i - y_i|^n \right)^{\frac{1}{n}}$$

$$f(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

# Funcții distanță

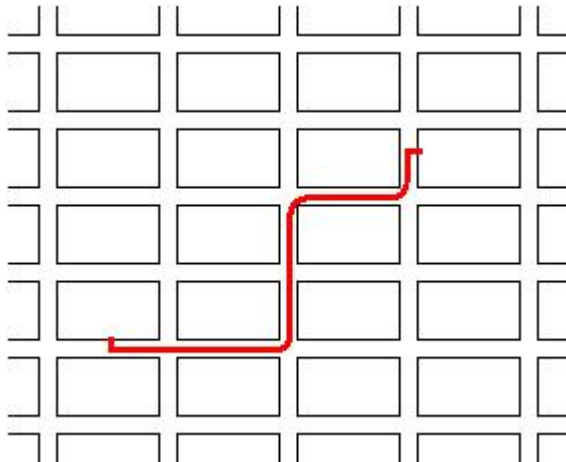
- Exemple:

- Minkowsky
- Euclidian
- Manhattan (*city block*)

$$f(x, y) = \left( \sum_{i=1}^d |x_i - y_i|^n \right)^{\frac{1}{n}}$$

$$f(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

$$f(x, y) = \sum_{i=1}^d |x_i - y_i|$$



# Funcții distanță

- Exemple în spațiu continuu:

- Minkowsky
- Euclidian
- Manhattan (*city block*)
- Max

$$f(x, y) = \left( \sum_{i=1}^d |x_i - y_i|^n \right)^{\frac{1}{n}}$$

$$f(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$$

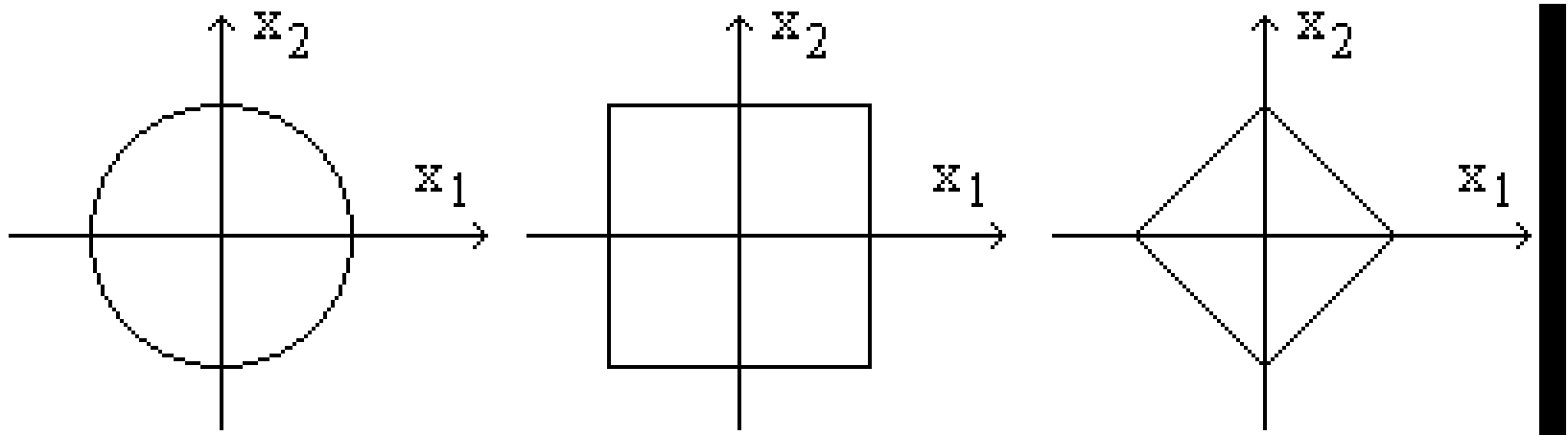
$$f(x, y) = \sum_{i=1}^d |x_i - y_i|$$

$$f(x, y) = \text{Max}_{i=1, d} |x_i - y_i|$$



# Funcții distanță

- Exemple în spațiu continuu:
  - sfere 2D, distanțele euclidiană, Max, Manhattan



# Funcții distanță

- Exemplu în spațiu discret:
  - Hamming

$$f(x, y) = \sum_{i=1}^d XOR(x_i, y_i)$$

# Cei mai apropiați $K$ vecini

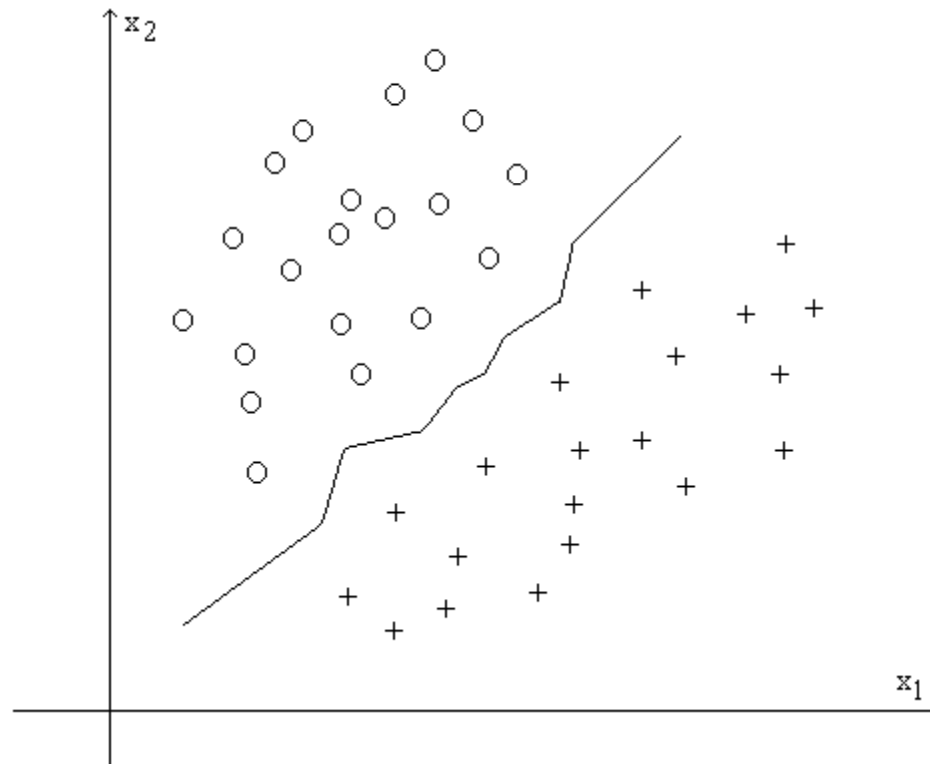
- Metod simpl de clasificare neparametric
- $K=1$ : cel mai apropiat vecin
  - determinist , neparametric
  - regula de clasificare
  - necesită o funcție distanță (ca orice metodă determinist )
  - frontier neliniar (dar liniară pe porțiuni)
  - exemplu în  $R^2$

# Cei mai apropiați $K$ vecini

- Regula celui mai apropiat vecin (NN)
  - caută, în setul de instruire, cel mai apropiat vecin al vectorului observat
  - atribuie vectorul la clasa acelui vecin
- Regula celor mai apropiați  $K$  vecini (KNN)
  - caută, în setul de instruire, cei mai apropiați  $K$  vecini ai vectorului observat
  - atribuie vectorul la clasa cea mai frecventă, dintre cei  $K$  vecini

# Cei mai apropiați $K$ vecini

- Regula celui mai apropiat vecin (NN)
  - frontieră liniară pe porțiuni



# Cei mai apropiați K vecini

- Regula celor mai apropiați K vecini
  - numerele reprezentanților fiecărei clase în setul de K e antioane formează un set de funcții discriminante
  - presupunem numărul de e antioane vecine cu  $x$ , care aparțin lui  $i : k_i$
  - restricție: 
$$\sum_{i=1}^c k_i = K$$
  - familia de funcții discriminante:  $f_i(x) = k_i$

# Cei mai apropiați $K$ vecini

- Proprietăți
  - nu necesită cunoștințe anterioare despre proprietățile claselor
  - simplă, nu necesită exprimare parametrică
  - la fiecare operație de recunoaștere, explorează întregul set de instruire
  - timpul de calcul proporțional cu dimensiunea setului de instruire și cu numărul de coordonate
  - frontiera pentru regula NN: liniară pe porțiuni
  - frontiera pentru regula KNN: polinom de ordin mai mare, pe porțiuni
  - frontiera rezultă implicit (nu este în formă explicită)

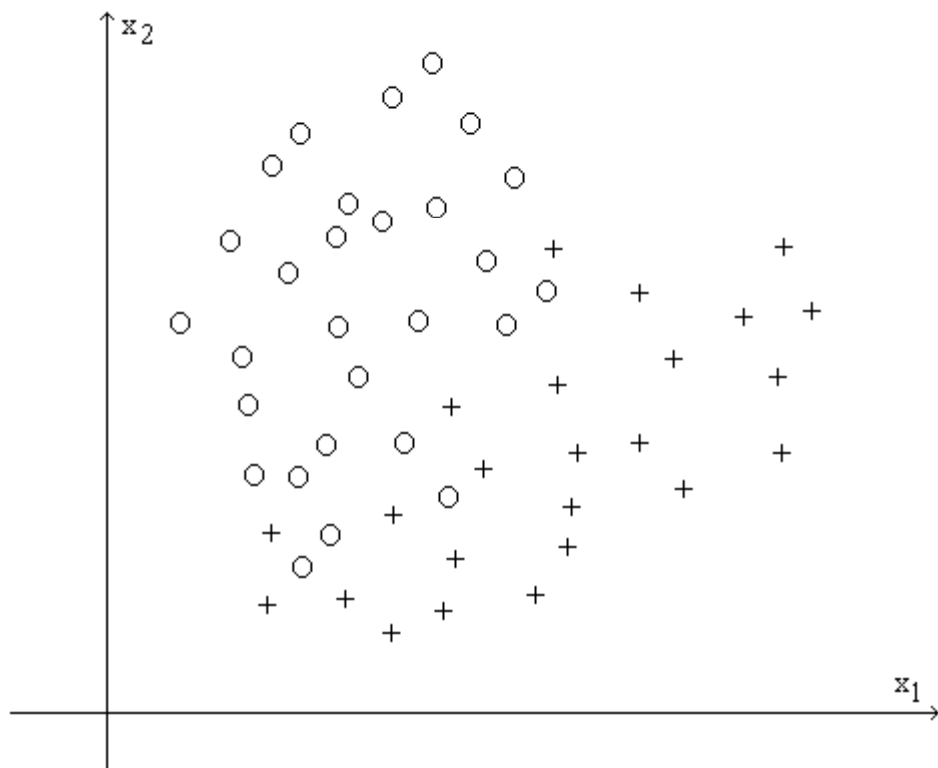
# Cei mai apropiați $K$ vecini

- Proprietăți
  - valori mici ale lui  $K$  produc o frontieră variantă , influențată de zgomot
  - valori mari ale lui  $K$  produc o frontieră prea netedă , incapabilă să urmeze forma reală a frontierei
  - o valoare mare a lui  $K$  este potrivit pentru un set de instruire mare
  - timpul de calcul proporțional cu  $K*n$
  - $K$  trebuie ales în funcție de setul de instruire



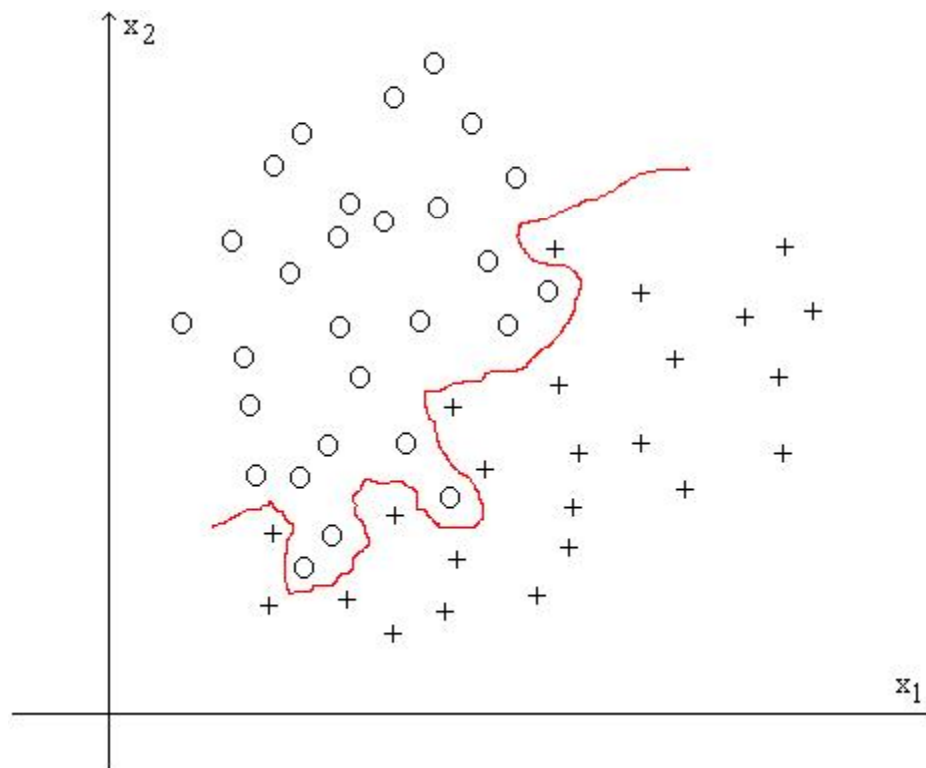
# Cei mai apropiați $K$ vecini

- Regula celui mai apropiat vecin (NN)
  - frontieră influențată de zgomot

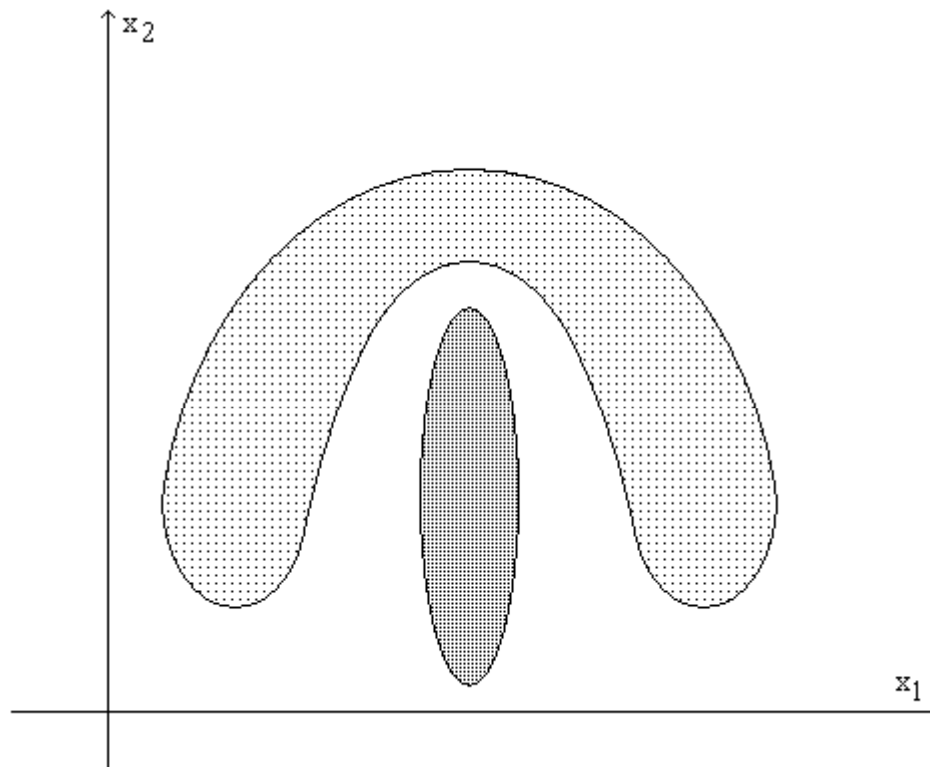


# Cei mai apropiați $K$ vecini

- Regula celui mai apropiat vecin (NN)
  - frontieră influențată de zgomot

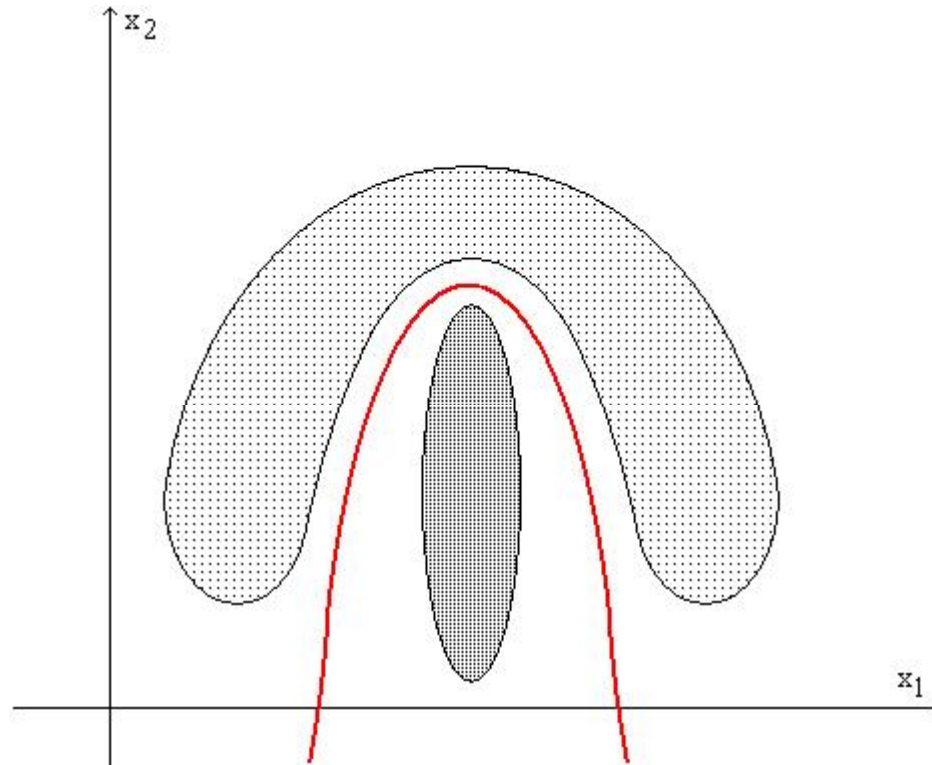


# Cei mai apropiați $K$ vecini



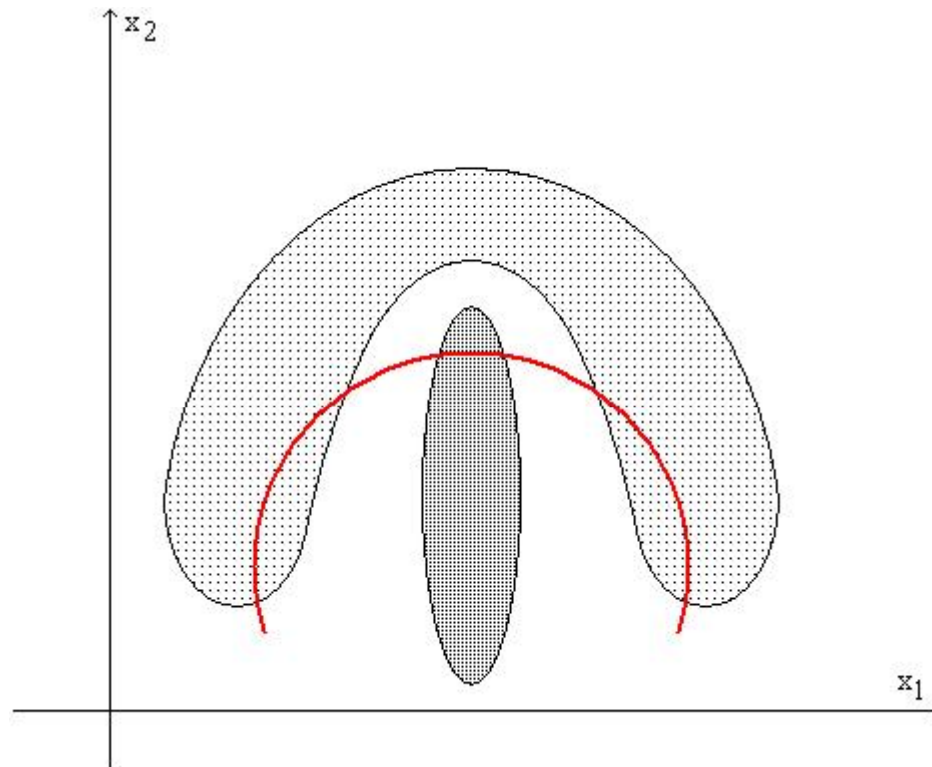
- KNN polinom de grad mai mare de 2

# Cei mai apropiați $K$ vecini



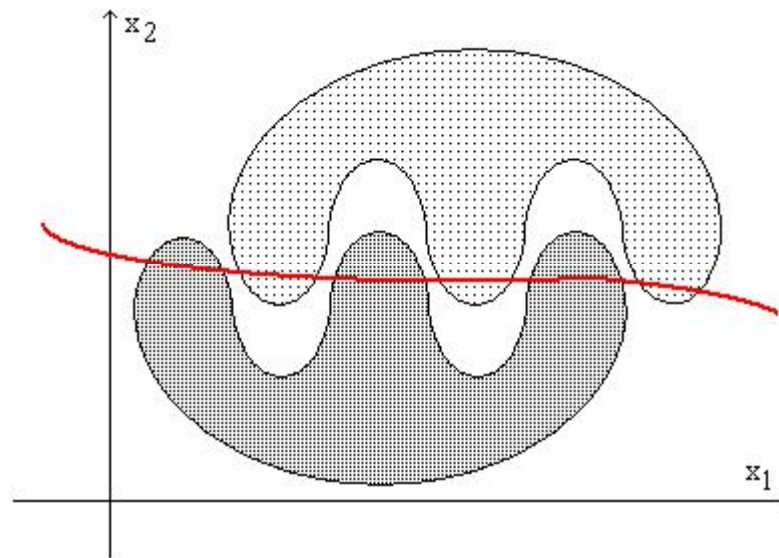
- KNN, valoare bună a lui  $K$

# Cei mai apropiați $K$ vecini



- KNN, valoare prea mare a lui  $K$

# Cei mai apropiați $K$ vecini



- KNN, valoare prea mare a lui  $K$

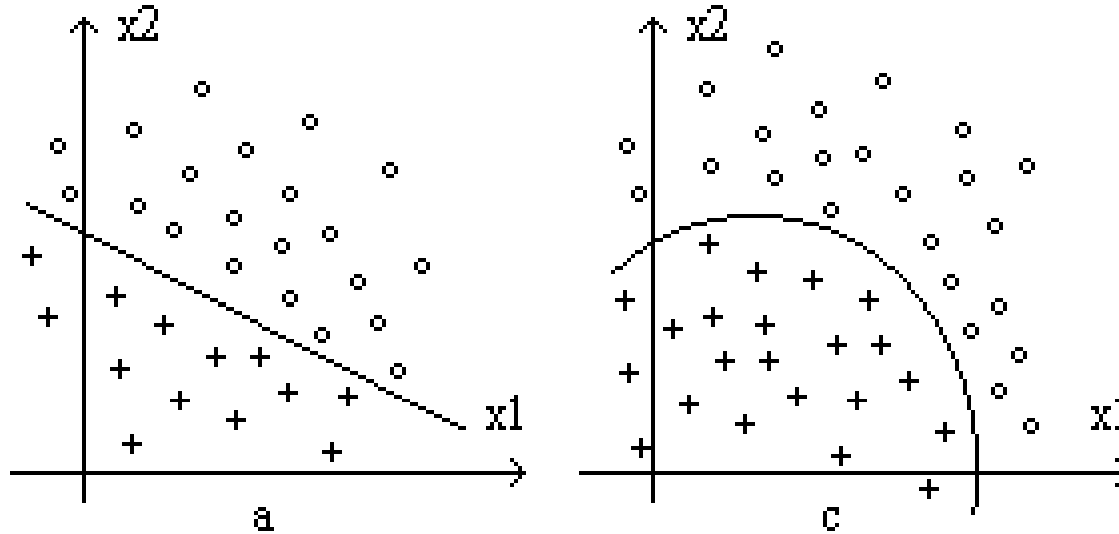
# Clasificator parametric

- Folosește descriere parametrică a funcțiilor discriminante și a frontierelor
- Reduce masiv timpul de calcul
- Forme uzuale ale funcțiilor: polinoame de grad 1 (liniar) sau 2 (parabolic)

$$f(x) = w^t x + w_0 \qquad f(x) = \sum_{i=1}^d w_i x_i + w_0$$

$$f(x) = x^t B x + w^t x + w_0 \qquad f(x) = \sum_{i=1}^d b_{ii} x_i^2 + \sum_{i=1}^{d-1} \sum_{j=i+1}^d b_{ij} x_i x_j + \sum_{i=1}^d w_i x_i + w_0$$

# Clasificator parametric



$$f(x) = w^t x + w_0$$

$$f(x) = \sum_{i=1}^d w_i x_i + w_0$$

$$f(x) = x^t B x + w^t x + w_0$$

$$f(x) = \sum_{i=1}^d b_{ii} x_i^2 + \sum_{i=1}^{d-1} \sum_{j=i+1}^d b_{ij} x_i x_j + \sum_{i=1}^d w_i x_i + w_0$$



# Clasificator de distanță minimă

- Parametric

- presupune c fiecare clas ocup un domeniu cvasisferic, în spațiul formelor,  $R^d$
- calculeaz centrul fiec rei clase (prototipul clasei)
- N.B. Ec. 4.12 din carte este gre it 
$$R_i = \frac{1}{n_i} \sum_{k=1}^{n_i} x_{ik}$$
- regula de atribuire: aloc vectorul la clasa celui mai apropiat prototip
- forma funcțiilor discriminante:  $f_i(x) = -d(x, R_i)$

# Clasificator de distanță minimă

- Proprietăți
  - clasificator parametric, liniar
  - parametrii sînt coordonatele prototipurilor
  - parametrii sînt calculați off-line
  - Calcule on-line foarte rapide
- Demonstrație pentru liniaritate

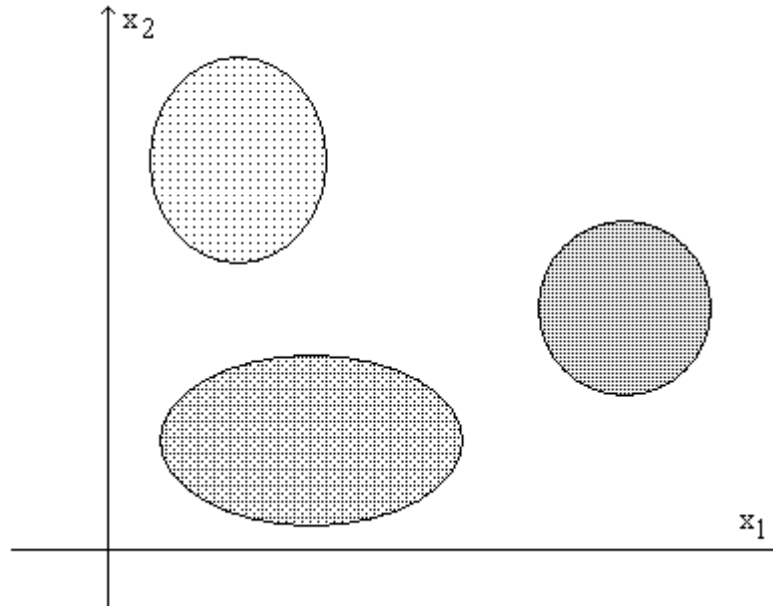
$$f_i(x) = -d^2(x, R_i)$$

$$f_i(x) = -(x - R_i)^t (x - R_i) = -x^t x + 2R_i^t x - R_i^t R_i$$

$$f_i(x) = 2R_i^t x - R_i^t R_i$$

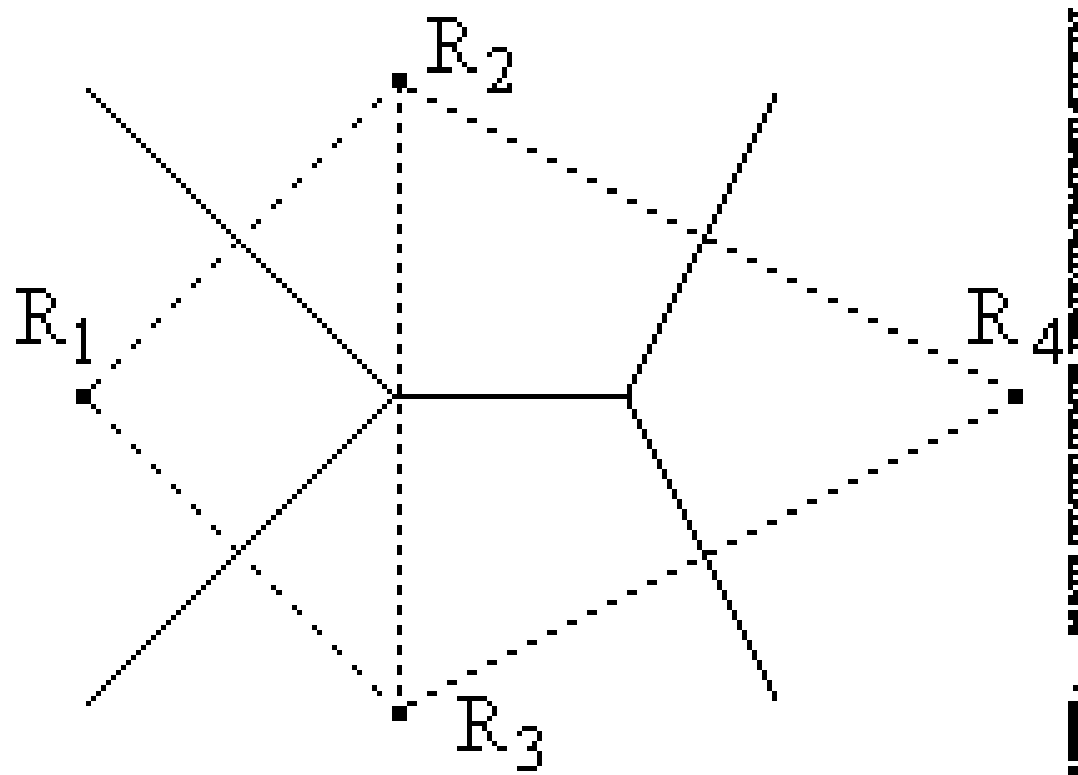
# Clasificator de distanță minimă

- Exemplu: potrivit



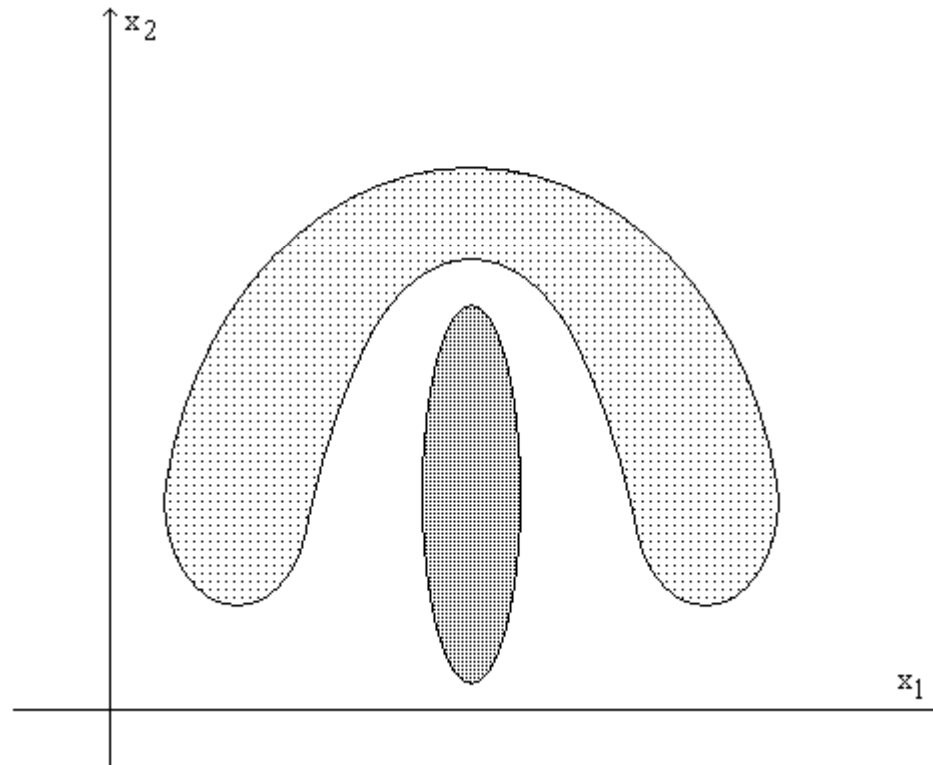
# Clasificator de distanță minimă

- Exemplu: frontiere



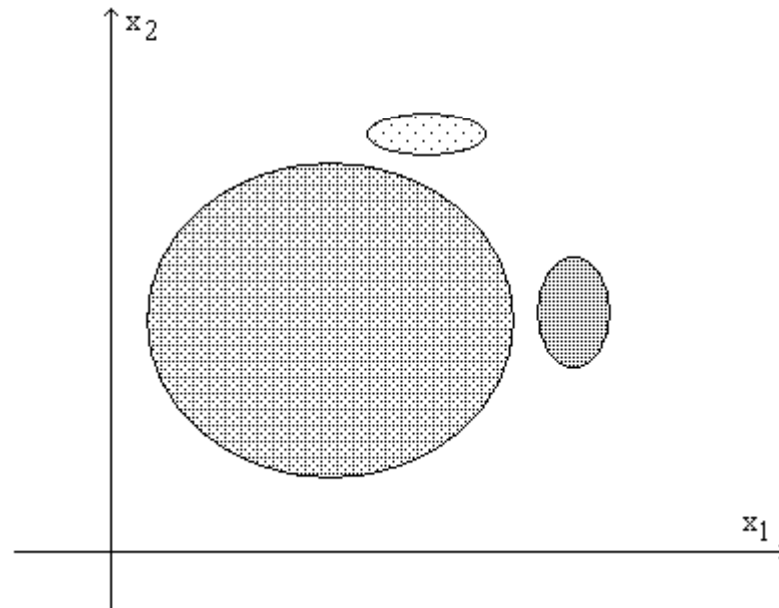
# Clasificator de distanță minimă

- Exemplu: nepotrivit



# Clasificator de distanță minimă

- Exemplu: nepotrivit



# Comparație algoritmi

## K vecini

- Etapă off-line:
- Etapă on-line:
- Calculează  $N$  distanțe
- Ordonează irul (primii  $K$  vecini)
- Decide în favoarea clasei majoritare dintre  $K$  vecini

## Distanță minimă

- Etapă off-line:
- Calculează  $c$  vectori medii (centrele claselor)
- Etapă on-line:
- Calculează  $c$  distanțe
- Decide în favoarea clasei cu distanța minimă

# Comparație algoritmi

## K vecini

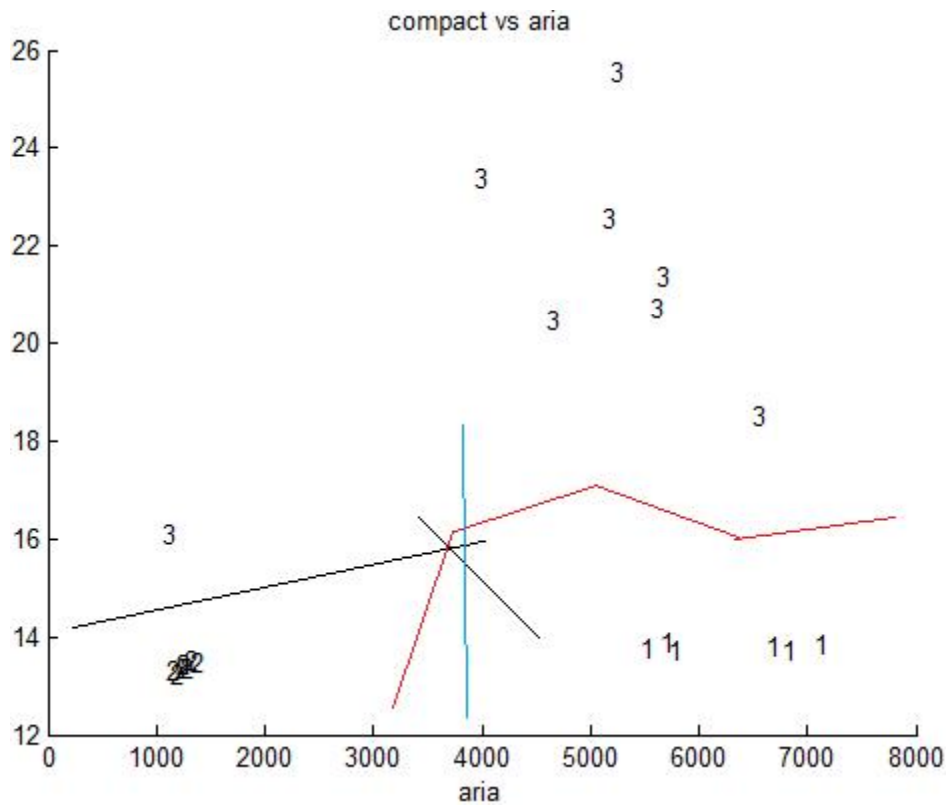
- Neparametric
- Nu sînt necesare cunoștințe anterioare
- Volum mare de calcule on-line: distanțe  $N*d$ , ordonare  $N*K$

## Distanță minimă

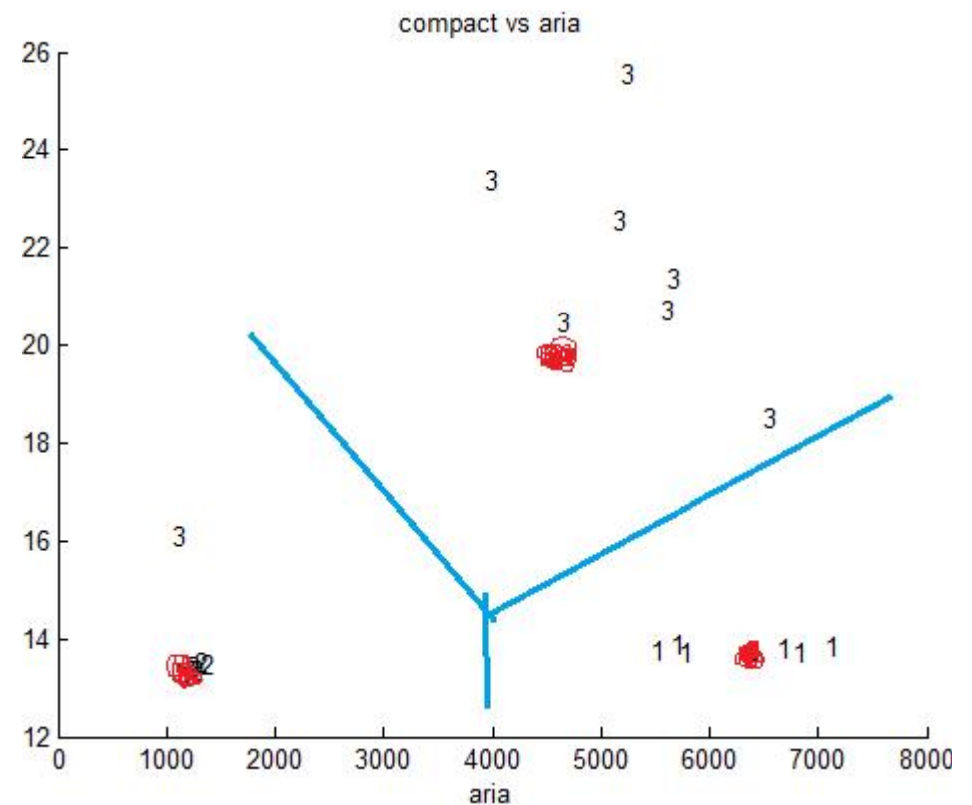
- Parametric
- Necesară cunoașterea grupării claselor în setul de instruire (cvasi sferice, comparabile)
- Volum moderat de calcule off-line: mediere  $N*d$
- Volum mic de calcule on-line



# Comparație algoritmi



Lauren iu Frangu, 2020



Prelucrarea Imaginilor, SEA

49

# Cuprins

1. Introducere, sisteme de achiziție și prelucrare a imaginilor
2. Preprocesare și îmbunătățire
3. Segmentarea și transformări morfologice
4. Extragerea trăsăturilor și măsurile pe imagine
5. Recunoașterea formelor
6. Aplicații