Curs 10:

Modele de regresie neliniară

Structura

- Motivaţie
- Reminder: Corelaţii, coeficient de corelaţie
- Reminder: Regresie liniară
- Modele neliniare
  - Modele liniare generalizate, modele aditive
  - Arbori de regresie
  - Reţele RBF
  - Modele bazate pe cei mai apropiați vecini
  - Modele bazate pe vectori suport

Motivaţie

Problema: Pornind de la caracteristici cunoscute ale unei maşini (e.g. Nr cilindri, cai putere, greutate, model etc) se doreşte estimarea consumului de combustibil (e.g. exprimat prin "miles per gallon")

Exemplu [autoMpg.arff de la http://archive.ics.uci.edu/ml/datasets.html]

@relation autoMpg

@attribute cylinders { 8, 4, 6, 3, 5} @attribute displacement real

@attribute horsepower real @attribute weight real @attribute acceleration real

@attribute model { 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82}

@attribute origin { 1, 3, 2}

@attribute class real

@data

8,307,130,3504,12,70,1,18

8,350,165,3693,11.5,70,1,15

4,113,95,2372,15,70,3,24

6,198,95,2833,15.5,70,1,22

6,199,97,2774,15.5,70,1,18

Motivație

Problema: Pornind de la caracteristici cunoscute ale unei maşini (e.g. Nr cilindri, cai putere, greutate, model etc) se doreşte estimarea consumului de combustibil (e.g. exprimat prin "miles per gallon")

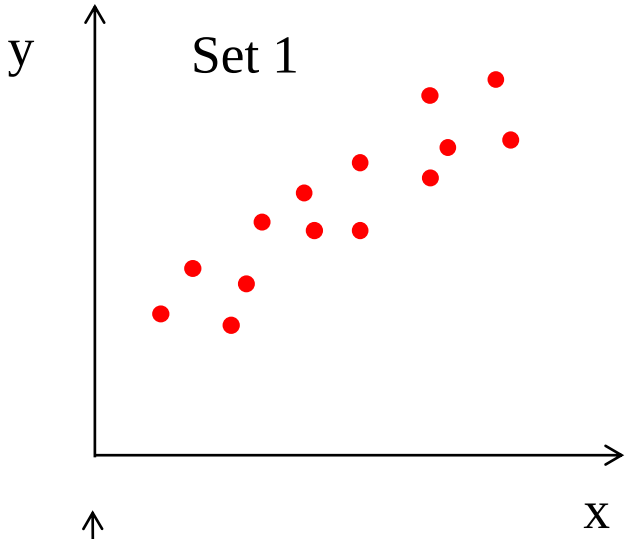
Exemplu [autoMpg.arff de la http://archive.ics.uci.edu/ml/datasets.html]

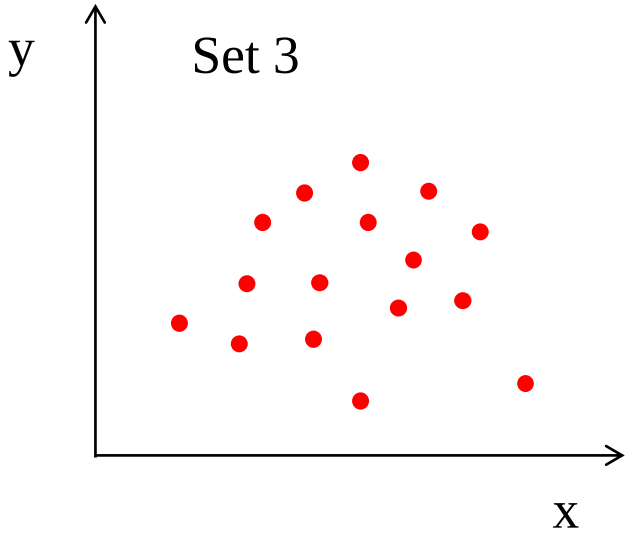
@relation autoMpg
@attribute cylinders { 8, 4, 6, 3, 5} @attribute displacement real
@attribute horsepower real @attribute weight real @attribute acceleration real
@attribute model { 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82}
@attribute origin { 1, 3, 2}
@attribute class real
@data
8,307,130,3504,12,70,1,18
8,350,165,3693,11.5,70,1,15
4,113,95,2372,15,70,3,24
6,198,95,2833,15.5,70,1,22
6,199,97,2774,15.5,70,1,18

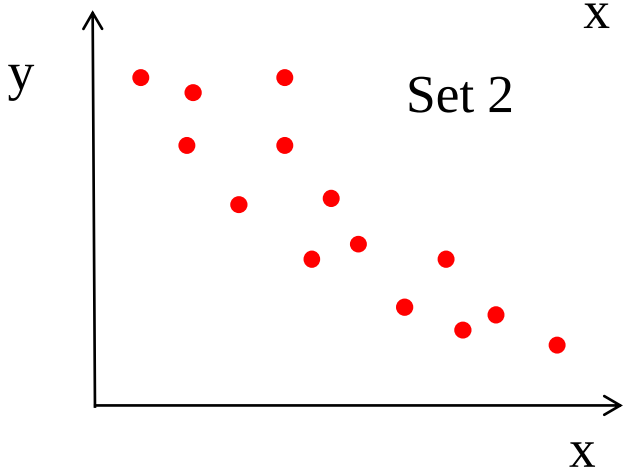
Se caută o relaţie care să descrie dependenţa dintre consumul de combustibil (atributul class în setul de date) şi caracteristicile maşinii (primele 7 atribute din setul de date)

Un exemplu mai simplu

Câteva seturi de date generate artificial



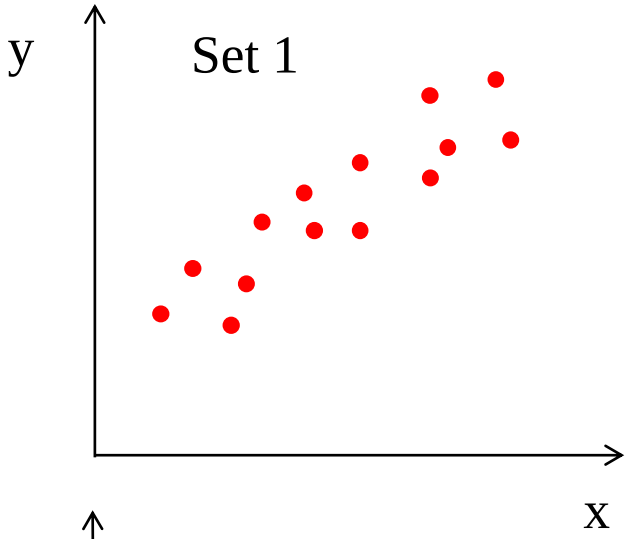


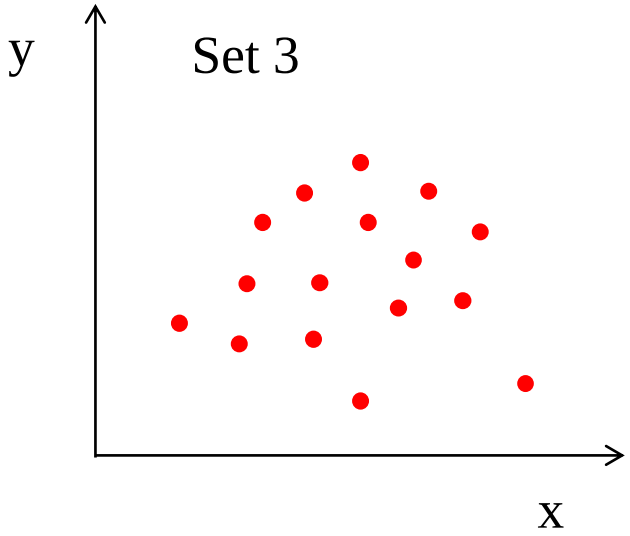


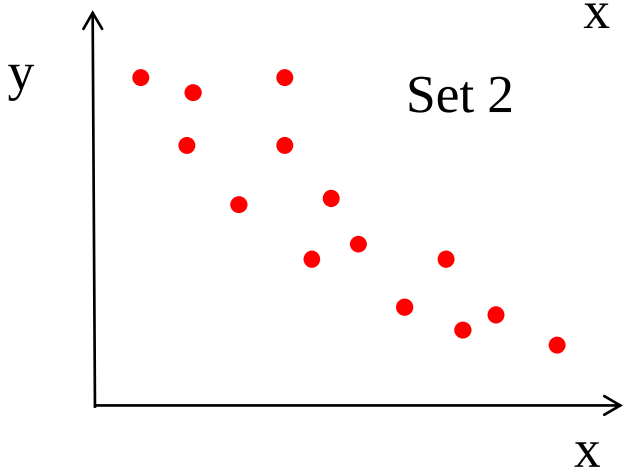
Ce se poate spune despre datele din fiecare set?

Un exemplu mai simplu

Câteva seturi de date generate artificial



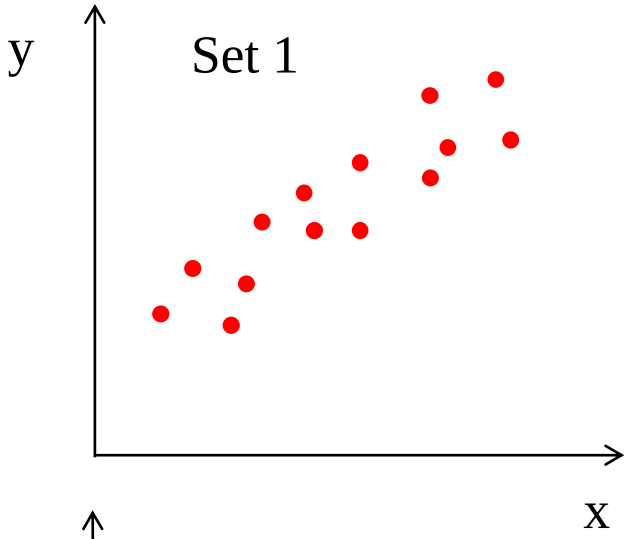


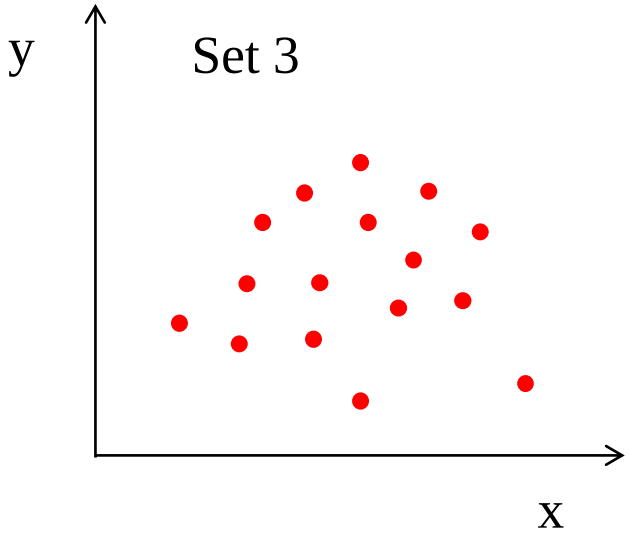


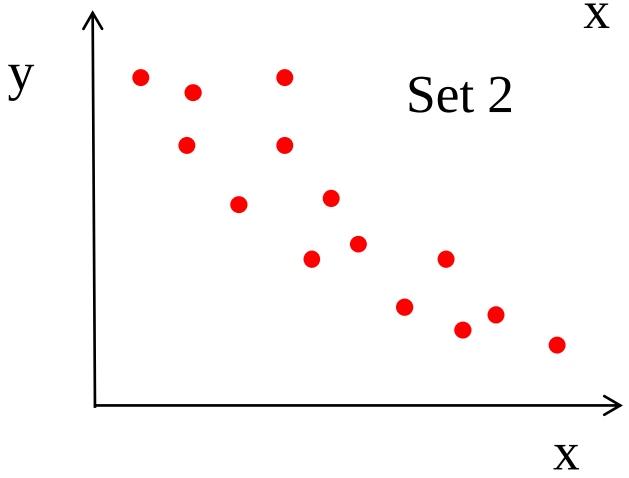
Set 1: datele par să fie corelate pozitiv
= dacă x creşte atunci şi y creşte

Un exemplu mai simplu

Câteva seturi de date generate artificial



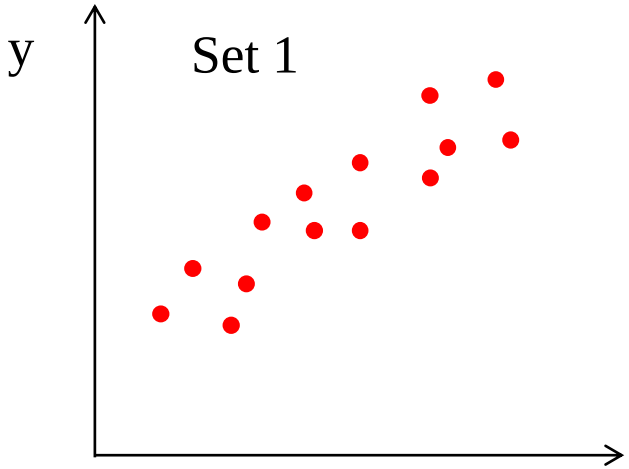


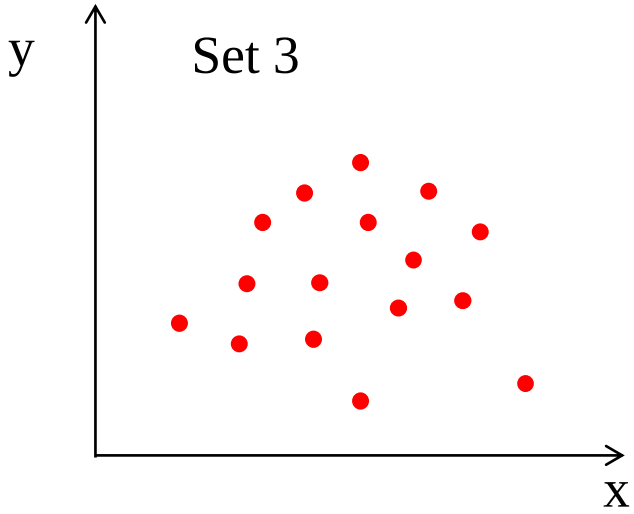


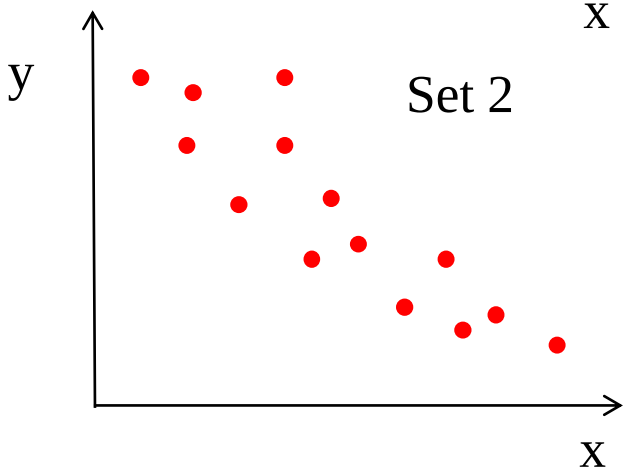
Set 2: datele par să fie corelate negaitiv
= dacă x creşte atunci y scade

Un exemplu mai simplu

Câteva seturi de date generate artificial







Set 3: datele par să nu fie corelate (e doar un nor de puncte)

Intrebări:

- Cum poate fi măsurat gradul de corelaţie?
- Ce tip de corelaţie există?

Reminder: Coeficient de corelaţie

Cum poate fi măsurat gradul de corelaţie?

[reminder – Probabilităţi şi Statistică]

- De exemplu folosind coeficientul de corelatie Pearson – exprimă gradul de corelaţie liniară dintre cele două variabile (numerice)

$$R(X,Y) = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - avg(X))(y_i - avg(Y))}{stdev(X)\,stdev(Y)}$$

$$stdev(X) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - avg(X))^2}$$

$$stdev(Y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - avg(Y))^2}$$

$$avg(X) = \frac{1}{n}\sum_{i=1}^{n} x_i,\ avg(Y) = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Obs: -1<=R(X,Y)<=1

- R(X,Y) apropiat 1: corelaţie liniară pozitivă
- R(X,Y) apropiat to -1: corelaţie liniară negativă
- R(X,Y) apropiat to 0: nu sunt corelate liniar (poate exista corelaţie neliniară între X şi Z)

Reminder: Regresie liniară

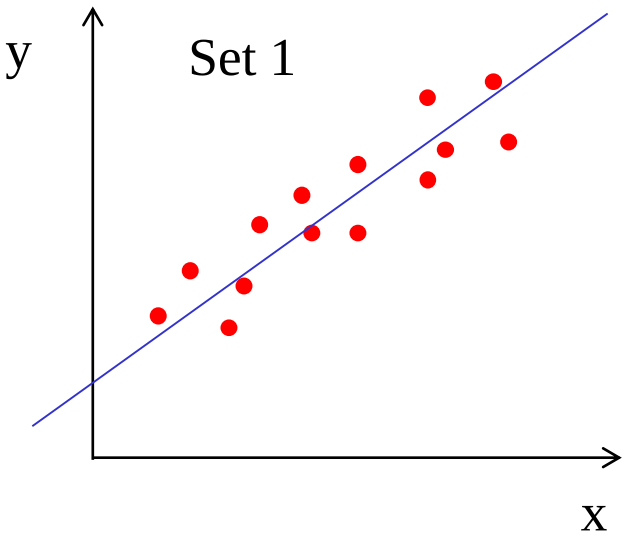
Ce tip de corelaţie ? [reminder – Statistică]

Cazul cel mai simplu: Dependenţa liniară dintre două variabile scalare:

$$Y=w_1X+w_0$$

- X= variabila predictor (independentă, intrare, explicativă)
- Y= variabila prognozată (dependentă, răspuns, explicată)

Scopul regresiei liniare: estimarea parametrilor w_1 şi w_0 a.î. valorile asociate variabilelor X (i.e. $x_1,x_2,\ldots, x_n$) şi Y (i.e. $y_1,y_2,\ldots, y_n$) sunt bine explicate de către funcţia liniară, i.e. suma pătratelor erorilor este minimizată



$$SSE(w_1,w_0)=\sum_{i=1}^{n}(y_i-(w_1x_i+w_0))^2$$

$$=\sum_{i=1}^{n}(y_i-\overline{w}\overline{x_i})^2$$

$$(\overline{w}=(w_1,w_0),\ \overline{x_i}=(x_i,1)^T)$$

Vector linie Vector coloană

Reminder:Regresie liniară simplă

Reminder: algebra liniară

$$w = (w_1, w_0), D = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ 1 & 1 & \dots & 1 \end{pmatrix}^T, y = (y_1, y_2, \dots, y_n)^T$$

$$SSE(w) = \left\| y - Dw^T \right\|^2 = (y - Dw^T)^T (y - Dw^T)$$

$$= y^T y - 2wD^T y + wD^T Dw^T$$

Determinarea vectorului w care minimizează SSE(w) este echivalentă cu determinarea punctului critic al lui SSE, adică rezolvarea următoarelor ecuaţii în raport cu w:

$$D^T Dw^T = D^T y \Rightarrow w^T = (D^T D)^{-1} D^T y = D^+ y$$

$$D^+ = (D^T D)^{-1} D^T \text{ este pseudoinversa lui } D$$

Reminder:Regresie liniară multiplă

Obs: abordarea poate fi extinsă în cazul mai multor variabile predictor (e.g. Setul autoMPG)

$$w = (w_1, w_2, ..., w_d, w_0), D = \begin{pmatrix} x_{11} & x_{12} & ... & x_{1n} \\ x_{21} & x_{22} & ... & x_{2n} \\ ... & ... & ... & ... \\ x_{d1} & x_{d2} & ... & x_{dn} \\ 1 & 1 & ... & 1 \end{pmatrix}^T, y = (y_1, y_2, ..., y_n)^T$$

$$SSE(w) = \left\| y - Dw^T \right\|^2 = (y - Dw^T)^T (y - Dw^T)$$

$$= y^T y - 2wD^T y + wD^T Dw^T$$

$$D^T Dw^T = D^T y \Rightarrow w^T = (D^T D)^{-1} D^T y$$

Regresie liniară - regularizare

Obs: dacă matricea D^TD este singulară (inversa nu poate fi calculată) atunci funcţia obiectiv (SSE) este modificată prin adăugarea unui termen de regularizare care va modifica matricea în aşa fel încât să se obţină o matrice inversabilă).

Exemple:

- Regularizare Tikhonov (ridge regression)

$$SSE'(w) = SSE(w) + \lambda \|w\|^2$$

$$w = (D^T D + \lambda I)^{-1} D^T y$$

$$I = (d+1) \times (d+1) \text{ matrice identitate}$$

Obs:

- Termenul de penalizare “descurajează” valorile mari ale parametrilor
- Parametrul termenului de regularizare (lambda) poate fi ales în manieră adaptivă folosind validare încrucişată

Regresie liniară - regularizare

Obs: dacă matricea D^TD este singulară (inversa nu poate fi calculată) atunci funcţia obiectiv (SSE) este modificată prin adăugarea unui termen de regularizare care va modifica matricea în aşa fel încât să se obţină o matrice inversabilă.

Exemple:

- Regularizare Lasso

$$SSE'(w) = SSE(w) + \lambda \sum_{i=1}^{d} | w_i |$$

Obs:

- In acest caz problema de optimizare se rezolvă folosind metode numerice
- Este utilă în cazul problemelor cu multe variabile dintre care o mare parte sunt irelevante (specific pt “sparse models”) – faciliteaza “setarea” pe 0 a unora dintre coeficienți acționând ca o tehnică de selecție a variabilelor

Analiza calității unui model

- Context:
  - Set de date: $\{(x_1,y_1), \ldots, (x_n,y_n)\}$
  - Model: y=F(x;w), w = coeficienți adaptivi (trebuie estimați)

- Eroare medie pătratică (Mean Squared Error – MSE)

$$MSE(x,y,w) = \frac{1}{n}\sum_{i=1}^{n}\big(y_i - F(x_i,w)\big)^2$$

- Coeficient de determinare (R-squared) = raportul dintre varianța explicată de model și varianța totală (valori între -1 și 1); cu cât valoarea este mai mare cu atât este mai mai adecvat modelul

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\big(y_i - F(x_i,w)\big)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

($\bar{y}$ = media valorilor de ieșire din setul de date)

Analiza calității unui model

Context:

- Considerăm două modele M1 și M2 construite pornind de la același set de date – fiecare dintre modele are un alt număr de parametri (k1 parametri în M1, k2 parametri în M2)
- Scopul urmărit este identificarea celui mai adecvat model (cel care asigură compromisul dintre calitatea aproximării și complexitatea modelului; complexitatea este exprimată prin numărul de parametri)
- AIC = Akaike Information Criterion (depinde de log-verosimilitatea modelului si numărul de parametri – cu cât valoarea este mai mică cu atât este considerat mai bun modelul)

$$AIC(x, y;\ w) = -2\ln\big(L(x, y, w)\big) + 2k$$

Obs: în cazul în care se consideră că zgomotul din date are distribuție normală, funcția de verosimilitate poate fi descrisă prin:

$$L(x, y;\ w) = \prod_{i=1}^{n} \exp\left(-\frac{\big(y_i - F(x_i, w)\big)^2}{2\sigma_i}\right)$$

Dincolo de modele liniare

Avantaje ale modelelor liniare:

- Suport teoretic și metode eficiente de estimare a parametrilor
- Ușor de interpretat și de analizat influența fiecăreia dintre variabilele predictor

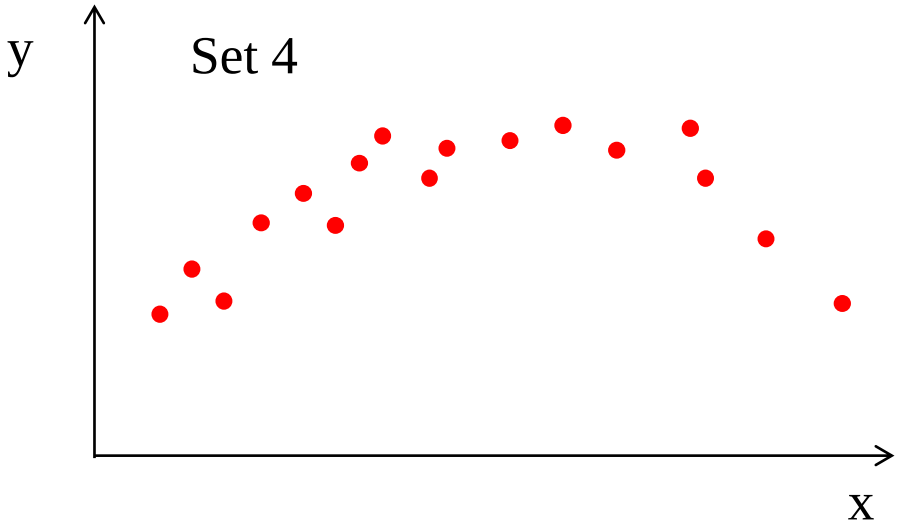
Dezavantaje ale modelelor liniare:

- Nu permit modelarea dependenței neliniare dintre variabila de ieșire și variabilele predictor (nu sunt suficient de flexibile)

Dincolo de modele liniare

Cum se abordează cazul în care dependenţa dintre variabila prezisă şi cele predictor nu este liniară?

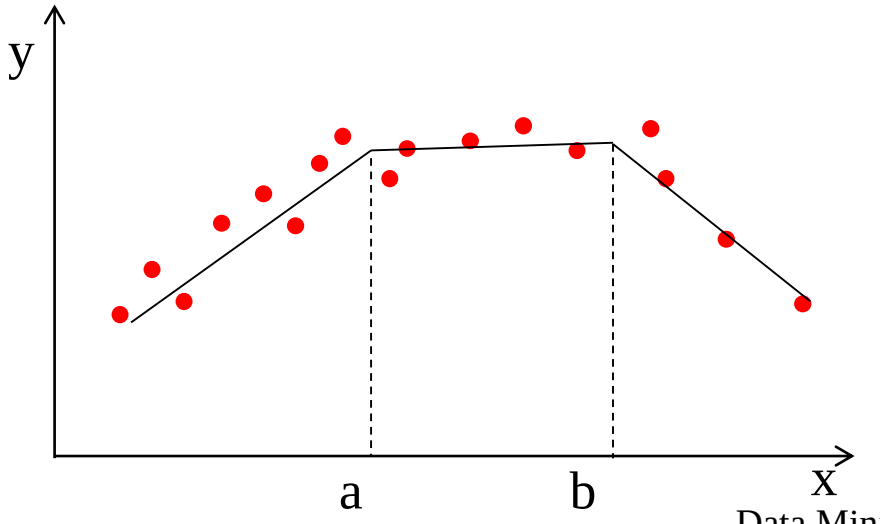
Sunt necesare alte modele



Dincolo de modele liniare

Idee:

- O dependenţă neliniară poate fi modelată prin mai multe funcţii liniare (câte una pentru fiecare regiune)
- Procesul de regresie constă din două etape:
  - Identificarea regiunilor prin partiţionarea spaţiului variabilelor predictor
  - Identificarea modelului de regresie (liniar) pt fiecare dintre regiuni



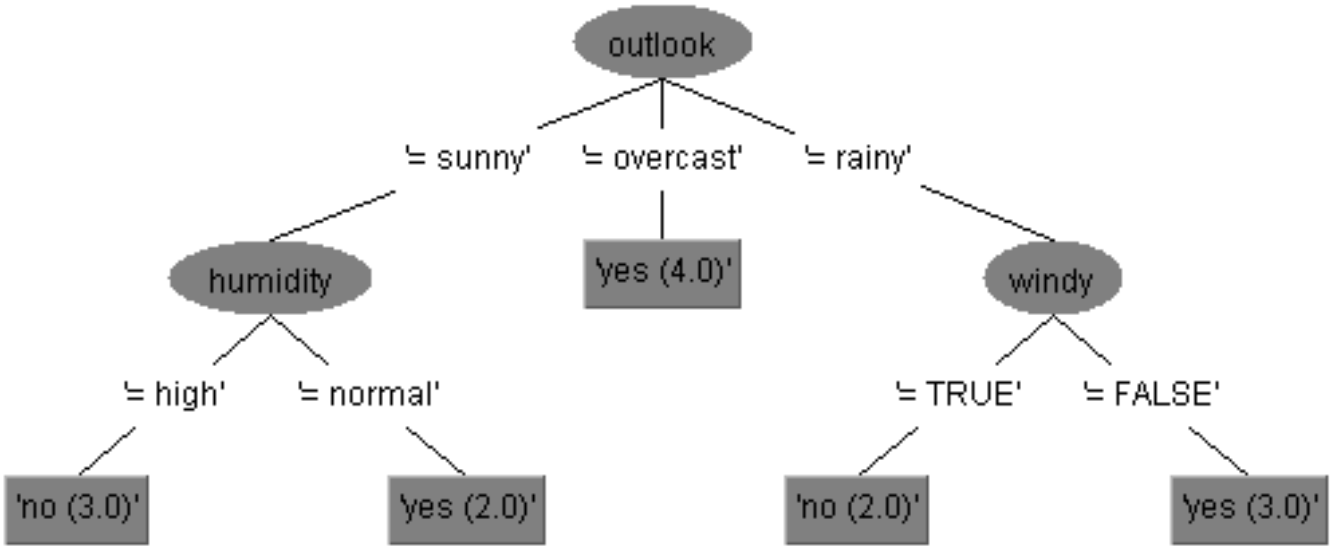
Intrebare:

Cu care dintre modelele de clasificare este similar un astfel de model?

Arbori de regresie

Reminder:

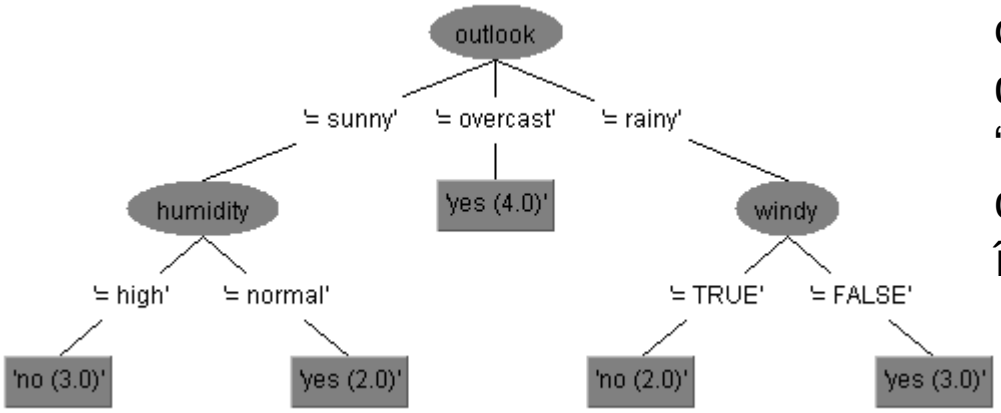
Arbori de decizie= arbore în care nodurile interne conţin condiţii referitoare la variabilele predictor iar cele frunză conțin informaţii privind variabila de ieșire (în cazul arborilor de clasificare variabila de ieșire este discretă şi nodurile frunză conţin indicatori de clasă)



Arbori de regresie

Reminder:

Arbori de decizie= arbore în care nodurile interne conţin condiţii referitoare la variabilele predictor iar cele frunză conțin informaţii privind variabila de ieșire (în cazul arborilor de clasificare variabila de ieșire este discretă şi nodurile frunză conţin indicatori de clasă)



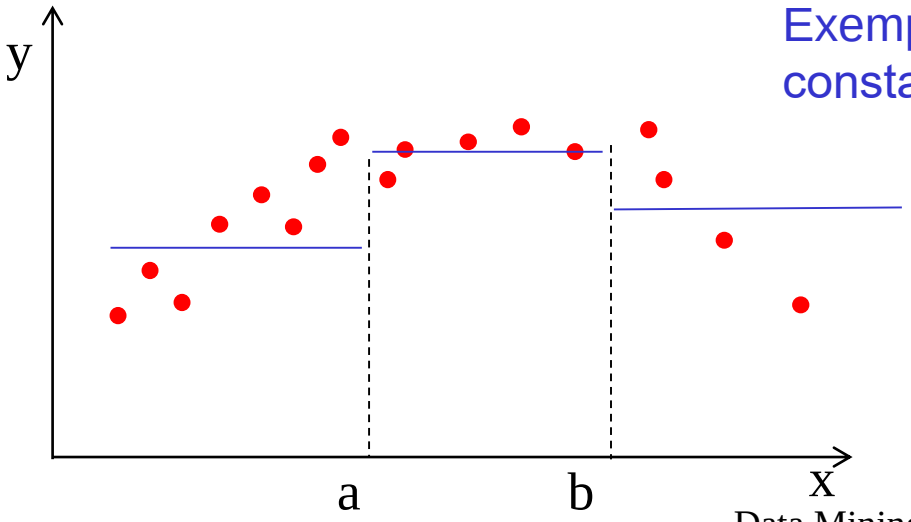
Intrebare:

- Dar dacă variabila prezisă este continuă? (ex: în locul unui răspuns de tipul da/nu în cazul problemei “weather-play” ar fi o valoare [0,1] care ar exprima un nivel de decizie între 0 (nu) şi 1 (da)

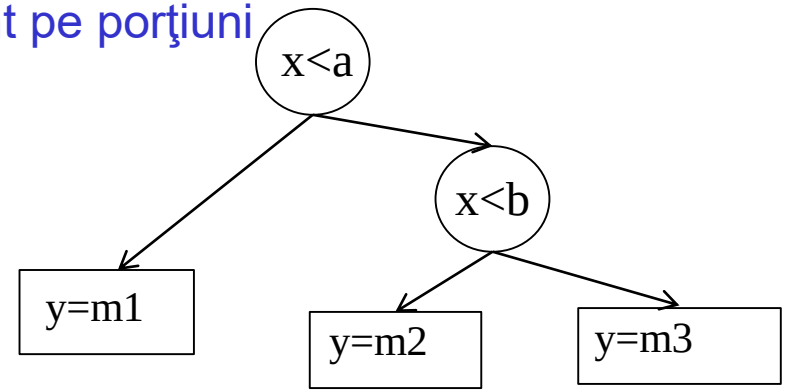
Arbori de regresie

Ideea principală:

- Se utilizează un proces similar de partiţionare a spaţiului de decizie ca şi în cazul arborilor de clasificare (criteriul de ramificare este corelat cu eroarea medie pătratică)
- Pt variabile predictor continue condiţia de ramificare este: variabila < valoare sau variabila > valoare sau variabila în [min,max]
- Pentru fiecare regiune se calculează media valorilor de ieșire (y) corespunzătoare datelor din regiunea respectivă



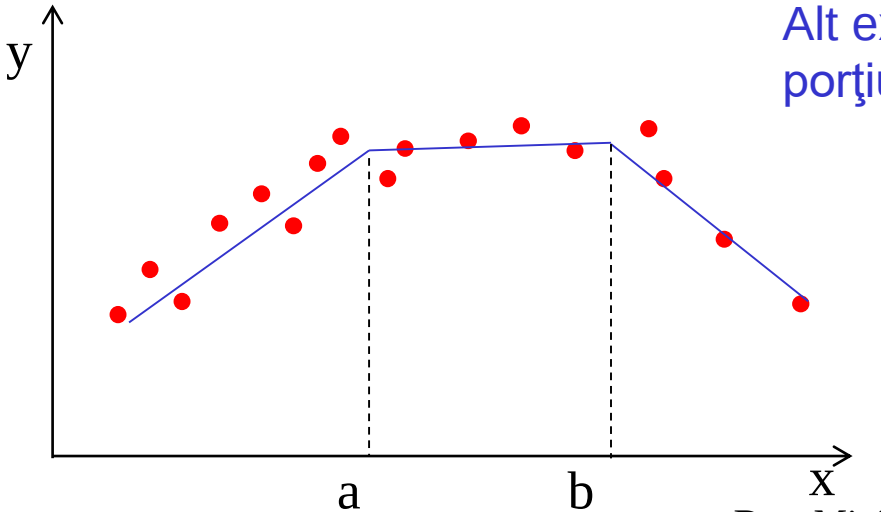
Exemplu foarte simplu -> model constant pe porţiuni



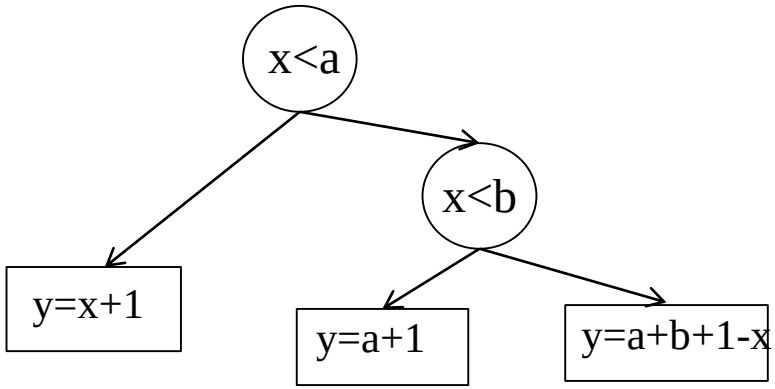
Arbori de regresie

Ideea principală:

- Se utilizează un proces similar de partiţionare a spaţiului de decizie ca şi în cazul arborilor de clasificare (criteriul de ramificare este corelat cu eroarea medie pătratică)
- Pt variabile predictor continue condiţia de ramificare este: variabila < valoare sau variabila > valoare sau variabila în [min,max]
- Se deduce un model de regresie (de exemplu liniar) pt fiecare dintre regiunile identificate prin procedura de ramificare



Alt exemplu -> model liniar pe porţiuni



Arbori de regresie

Construcție: Recursive Binary Splitting

- se pornește cu nodul de start (corespunde întregului set de date) și se aplică recursiv o strategie greedy pt a selecta variabila predictor j și valoarea de secționare s care minimizează suma pătratelor erorilor corespunzătoare datelor care aparțin celor două regiuni disjuncte definite de s:

$$R_1 = \{x | x_j < s\} \quad \text{si} \quad R_2 = \{x | x_j \geq s\}$$

$$MSE(j,s) = \sum_{x_i \in R_1(j,s)} (y_i - \hat{y}_{R1})^2 + \sum_{x_i \in R_2(j,s)} (y_i - \hat{y}_{R2})^2$$

- $\hat{y}_{R1}$ și $\hat{y}_{R2}$ sunt medii ale variabilelor de ieșire corespunzătoaare datelor ce aparțin lui R1, respectiv R2
- Procesul de partiționare este aplicat recursiv până când nodurile frunză ajung să corespundă unui subset de date suficient de mic (de exemplu cel mult 5 elemente)
- Se aplică o strategie de simplificare (pruning - similară cu cea utilizată în cazul arborilor de clasificare): arborii sunt înlocuiți cu noduri frunză atât timp cât MSE nu crește semnificativ)

Dincolo de modele liniare

Trecerea de la dependența liniară la dependența neliniară se poate realiza:

- în raport cu variabilele (modelul rămâne liniar în raport cu parametrii de estimat → se pot folosi în continuare tehnicile de estimare a coeficienților de la regresia liniară):
  - modele liniare generalizate: se aplică transformare neliniară asupra variabilei răspuns corespunzătoare modelului liniar
  - modele aditive: se aplică transformări neliniare asupra variabilelor predictor
- atât în raport cu variabilele cât și cu parametrii de estimat:
  - rețele neuronale: se aplică transformări neliniare într-o manieră ierarhică

Modele liniare generalizate

Context statistic: $y=w_1x_1+ ... +w_dx_d +w_0 + z$ (z este variabilă aleatoare cu repartiție normală N(0,sigma)) - y e variabilă aleatoare cu distribuție normală

Idee: y este modelată printr-o variabilă aleatoare care are media $f(w_1x_1+ ... +w_dx_1 +w_0)$ – unde f este o funcție neliniară

Principalele elemente ale unui GLM (Generalized Linear Model):

- Funcţia de medie (mean function): f
- Funcţia de legătură (link function): f^{-1}
- Distribuţia de probabilitate a variabilei de ieșire (distribution)

Funcție medie (f)	Funcție legătura (f^{-1})	Distribuție
f(u)=u	identitate	normală
f(u)=-1/u	inversa	Exponențială, Gamma
f(u)=exp(u)	Log	Poisson
f(u)=1/(1+exp(-u))	Logit	Bernoulli

Modele liniare generalizate

Idee: y este modelată printr-o variabilă aleatoare care are media $f(w_1x_1 + ... + w_dx_d + w_0)$ – unde f este o functie neliniară

Principalele elemente ale unui GLM (Generalized Linear Model):

- Funcţia de medie (mean function): f
- Funcţia de legătură (link function): f^{-1}
- Distribuţia de probabilitate a variabilei de ieșire (distribution)

Funcție medie (f)	Funcție legătura (f^{-1})	Distribuție
f(u)=u	identitate	normală
f(u)=-1/u	inverse	Exponențială, Gamma
f(u)=exp(u)	Log	Poisson
f(u)=1/(1+exp(-u))	Logit	Bernoulli

Regresie clasică (metoda celor mai mici pătrate)

Modele liniare generalizate

Idee: y este modelată printr-o variabilă aleatoare care are media $f(w_1x_1 + \ldots + w_dx_d + w_0)$ – unde f este o functie neliniară

Principalele elemente ale unui GLM (Generalized Linear Model):

- Funcţia de medie (mean function): f
- Funcţia de legătură (link function): f^{-1}
- Distribuţia de probabilitate a variabilei de ieșire (distribution)

Funcție medie (f)	Funcție legătura (f^{-1})	Distribuție
f(u)=u	identitate	normală
f(u)=-1/u	inverse	Exponențială, Gamma
f(u)=exp(u)	Log	Poisson
f(u)=1/(1+exp(-u))	Logit	Bernoulli

Regresie Logistică (utilizată pentru probleme de clasificare)

Modele aditive

Idee: se conservă proprietatea de liniaritate în raport cu parametrii de estimat

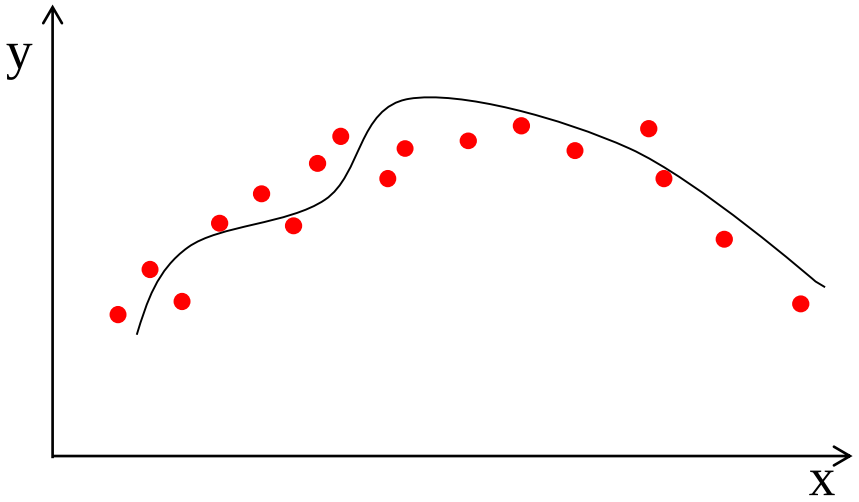
- Se extinde modelul clasic de regresie liniară considerând atribute transformate prin intermediul unor funcţii neliniare

$$y=w_0+w_1h_1(x)+w_2h_2(x)+\ldots+w_mh_m(x)$$

(x e un vector, h_i e o funcţie ce asociază un scalar sau un vector argumentului său)

Caz particular 1. Modele polinomiale (d=1): $y= w_0+w_1x+w_2x^2+\ldots+w_mx^m$

(x este un scalar)



Obs: modelele polinomiale sunt mai flexibile decât cele liniare și sunt utile în practică cu condiția să nu fie de grad prea mare (dacă gradul e mai mare decât 4 modelul captează inclusiv zgomotul din date)

Modele aditive

Caz particular 2. Transformare neliniară a efectelor variabilelor predictor

$$y=w_0+w_1h_1(x_1)+w_2h_2(x_2)+\ldots+w_dh_d(x_d)$$

(d reprezintă numărul variabilelor predictor)

Obs:

- Modelele aditive de tipul celui de mai sus se bazează pe ipoteza că influențele variabilelor predictor asupra variabilei de ieșire sunt necorelate

- Pentru a lua în considerare interacțiunile dintre variabilele predictor se pot introduce termeni "combinați" de tipul: $h(x_i, x_j)= x_ix_j$

Modele globale și modele locale

Modelare globală: același tip de dependență pentru toate valorile variabilelor predictor.

Exemple:

- Modelul clasic de regresie liniară
- Modelele polinomiale

Modelare locală: fiecărei regiuni din domeniul variabilelor predictor îi poate corespunde un alt model.

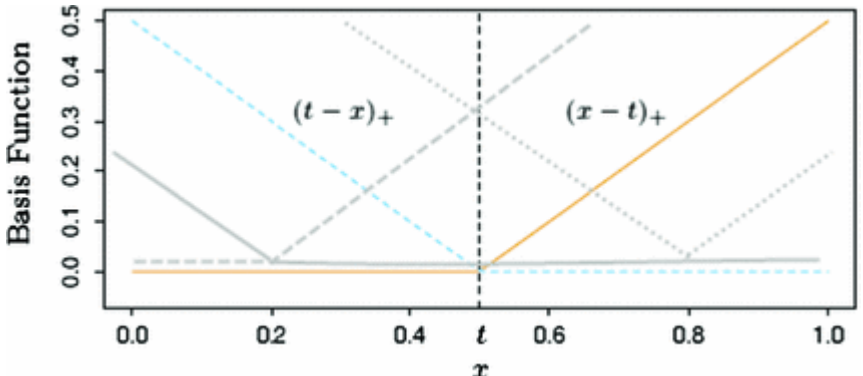
Exemple:

- Arbori de regresie – valoare constantă asociată fiecărei regiuni
- Modele de tip spline – modele polinomiale diferite alese astfel încât să fie asigurată continuitate (și netezime) în punctele de trecere dintr-o regiune în alta (knots)
- MARS (Multivariate Adaptive Regression Splines) – utilizează ca funcții de bază funcții liniare pe porțiuni:

  max(0,x-t), max(0,t-x)

(t = punct de trecere intre regiuni)

[Hastie, Tibshirani, Friedman – The Elements of Statistical Learning, 2017]



Modele globale și modele locale

Obs: regiunile corespunzătoare modelelor locale pot fi

- Disjuncte (ca în cazul arborilor de regresie sau a modelelor de tip spline)
- Suprapuse – modele bazate pe funcții de bază radiale
    - h_i sunt funcţii care iau valori nenule pe tot domeniul, însă sunt semnificative doar pentru regiuni limitate din spaţiul variabilelor predictor
    - dacă aceste funcţii au simetrie radială (de exemplu funcţii gaussiene) se ajunge la reţelele de tip RBF (pot fi interpretate ca un caz particular de reţele neuronale)

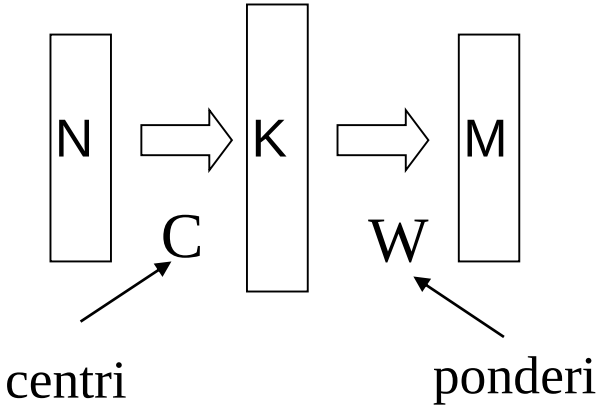
Rețele cu funcții radiale

- RBF - “Radial Basis Function”:

- Arhitectura:
  - Două nivele de unități funcționale

  - Funcții de agregare:
    - Unități ascunse: distanța dintre vectorul de intrare și cel al ponderilor corespunzătoare unității ascunse

    - Unități de ieșire: suma ponderată



$$G(X, C^k) = \left\| X - C^k \right\| = \sqrt{\sum_{i=1}^{N} (x_i - c_i^k)^2}$$

Funcții de transfer (activare):

- nivelul ascuns: funcții cu simetrie radială

- nivelul de ieșire: funcții liniare

Rețele cu funcții radiale

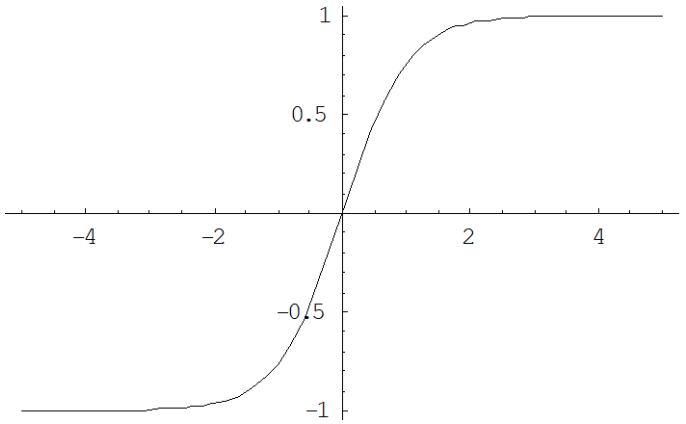
Diferența față de rețelele feedforward clasice:

Funcții de transfer:

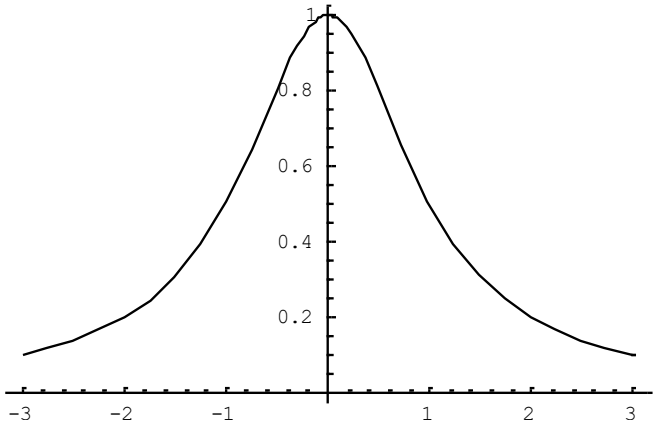
FF: funcții sigmoidale

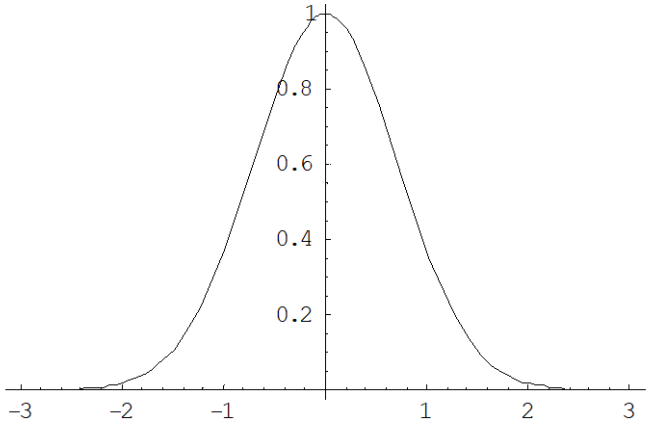
RBF: funcții cu simetrie radială

Funcție sigmoidală



Funcții cu simetrie radială



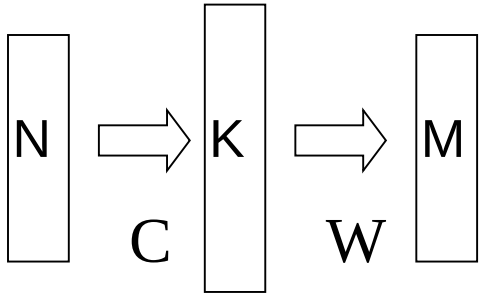


Rețele cu funcții radiale

Funcționare:

$$y_i = \sum_{k=1}^{K} w_{ik}\, g(\|X - C^k\|) - w_{i0},\ i = \overline{1, M}$$

$$y_i = \sum_{k=1}^{K} w_{ik} z_k - w_{i0},\ z_k = g(\|X - C^k\|)$$



Matrice centri Matrice ponderi

Parametrii C^k pot fi interpretați ca prototipuri (centri) asociați unităților ascunse: vectorii de intrare X apropiați lui C^k vor conduce la o valoarea de ieșire semnificativă pe când cei îndepărtați vor conduce la o valoare de ieșire nesemnificativă; la construirea răspunsului rețelei vor contribui doar unitățile a căror centri sunt suficient de similari cu data de intrare

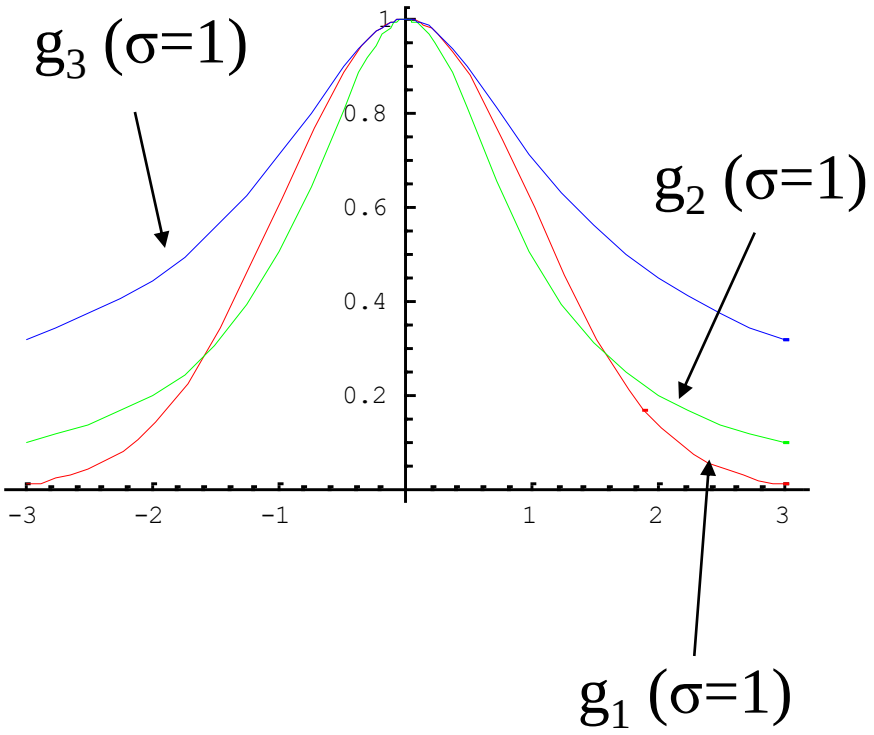
Rețele cu funcții radiale

Exemple de funcții radiale:

$$g_1(u) = \exp(-u^2/(2\sigma^2))$$

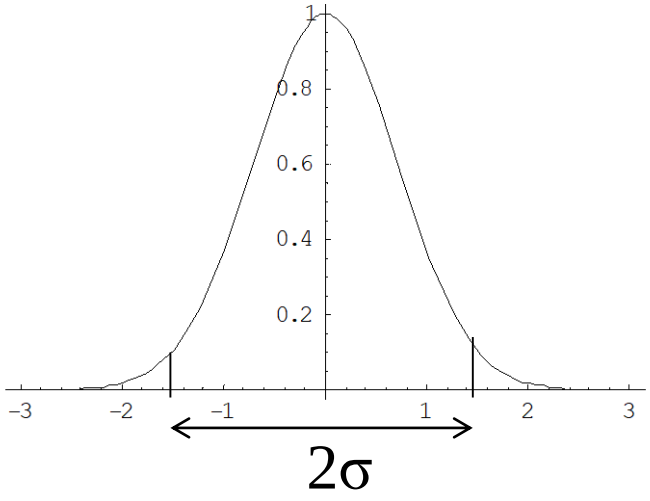
$$g_2(u) = 1/(u^2+\sigma^2)$$

$$g_3(u) = 1/\sqrt{u^2+\sigma^2}$$

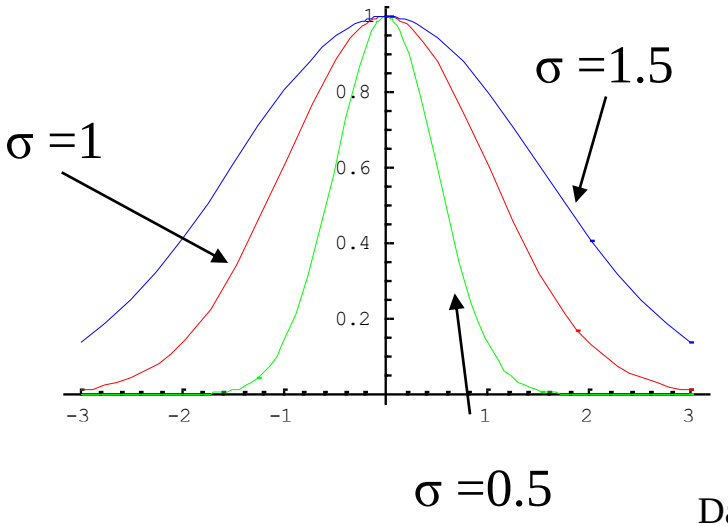


Rețele cu funcții radiale

- Fiecare unitate ascunsă este "sensibilă" la semnalele de intrare provenite dintr-o regiune a spațiului de intrare aflată în vecinatatea centrului. Aceasta regiune este denumită câmp receptiv
- Dimensiunea câmpului receptiv depinde de σ

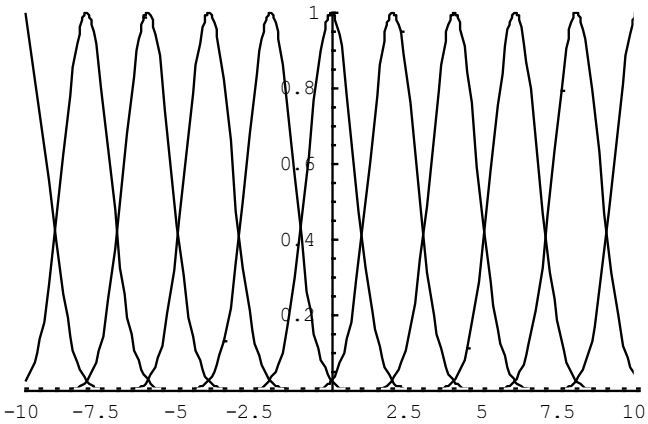


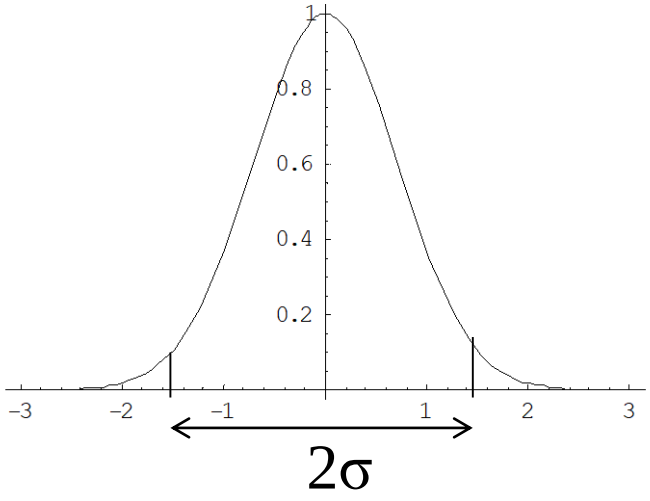
$$g(u) = \exp\left(-\frac{u^2}{2\sigma^2}\right)$$

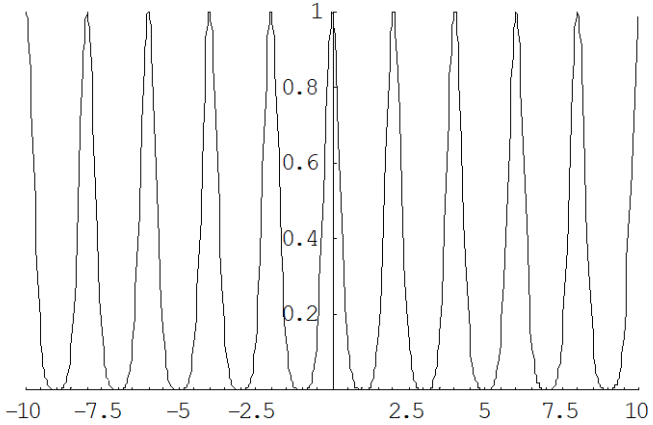


Rețele cu funcții radiale

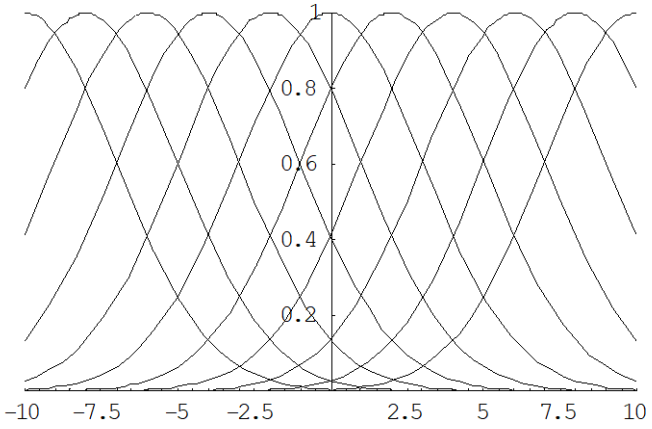
Influența lui σ: $g(u) = \exp\left(-\frac{u^2}{2\sigma^2}\right)$







subacoperire

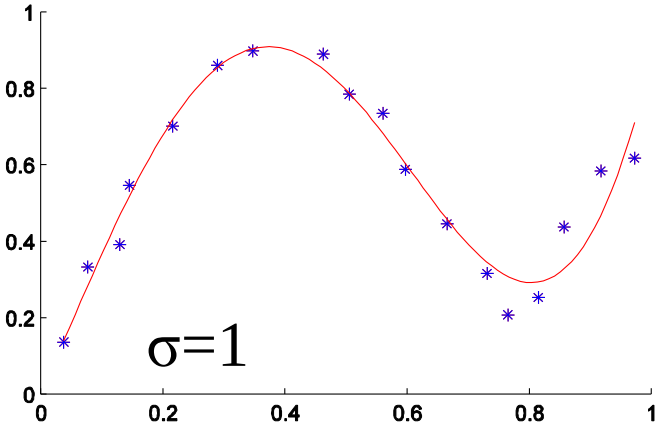


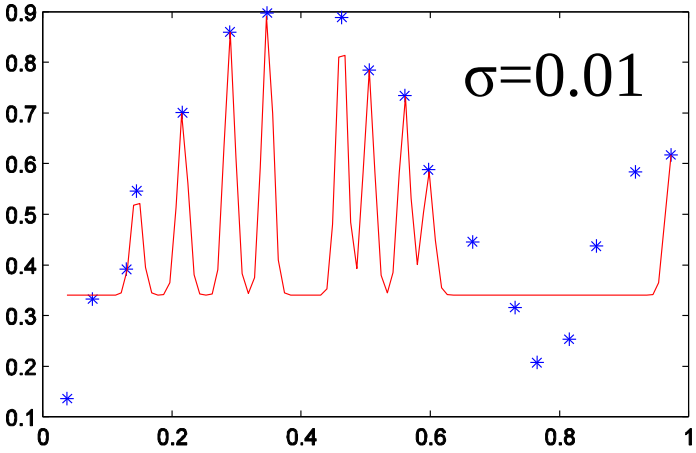
supraacoperire

Rețele cu funcții radiale

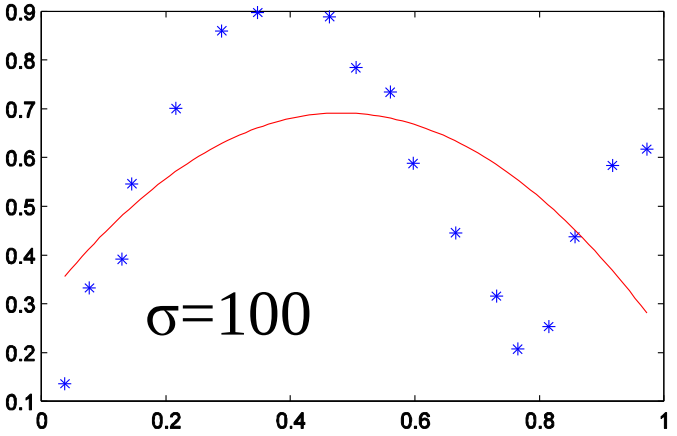
- O bună acoperire a domeniului datelor de intrare de către câmpurile receptive ale funcțiilor radiale de transfer este esențială pentru calitatea aproximării
- Valori prea mici conduc la incapacitatea de a produce rezultate pentru întreg domeniul datelor
- Valori prea mari nu surprind variabilitatea datelor

Acoperire adecvată





Subacoperire

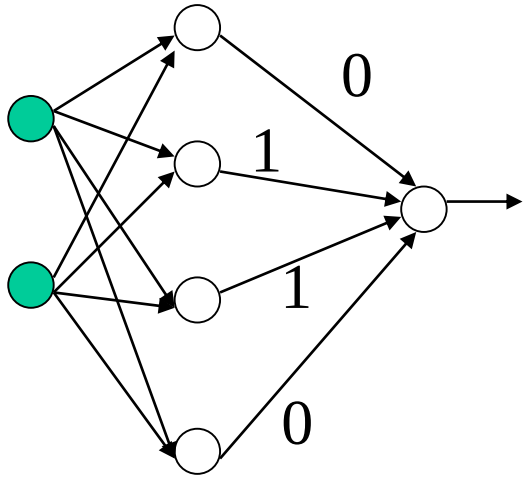


Supraacoperire

Rețele cu funcții radiale

Exemplu (caz particular) : rețea RBF pentru reprezentarea lui XOR

- 2 unități de intrare
- 4 unități ascunse
- 1 unitate de ieșire



Centrii:

u.a. 1: (0,0)

u.a. 2: (1,0)

u.a. 3: (0,1)

u.a. 4: (1,1)

Ponderi:

w1: 0

w2: 1

w3: 1

w4: 0

Funcție de activare:

g(u)=1 if u=0

g(u)=0 if u<>0

Aceasta abordare nu poate fi aplicată pentru probleme generale de aproximare

Rețele cu funcții radiale

Invățare:

Set de antrenare: $\{(x^1,d^1), \ldots, (x^L,d^L)\}$

Etape:

(a) Stabilirea parametrilor corespunzatori nivelului ascuns: centrii C și parametrii σ

(b) Determinarea parametrilor W (problemă de optimizare pătratică)

Obs: Invățarea de tip RBF elimină o parte dintre dezavantajele algoritmului BP: convergența lentă, blocarea în minime locale (întrucât se ajunge la rezolvarea unei probleme mai simple de optimizare) etc.

Rețele cu funcții radiale

Invățare:

Set de antrenare: $\{(x^1,d^1), \ldots, (x^L,d^L)\}$

(a) Stabilirea parametrilor corespunzători nivelului ascuns: centrii C și parametrii σ

(a) K=L (nr centri = nr exemple), $C^k=x^k$

(b) K<L : centrii se stabilesc

(a) prin selecție aleatoare dintre exemplele din setul de antrenare

(b) prin selecție sistematică dintre exemplele din setul de antrenare (Orthogonal Least Squares)

(c) prin utilizarea unui algoritm de grupare (poate permite și estimarea numărului de centri) – in acest caz centrii nu vor face neapărat parte din setul de antrenare

Rețele cu funcții radiale

Orthogonal Least Squares:

- Selecție incrementală a centrilor astfel încât eroarea să fie micșorată cât mai mult
- Noul centru este ales astfel încât să fie ortogonal pe spațiul generat de către centrii deja selectați (procesul este bazat pe metoda de ortogonalizare Gram-Schmidt)
- Abordarea este corelată cu regresia de tip “ridge”

Rețele cu funcții radiale

Grupare (clustering):

- Se urmărește identificarea a K clase în setul de date de antrenare $\{X_1,\ldots,X_L\}$ astfel încât datele din fiecare clasă să fie suficient de similare pe când datele din clase diferite să fie suficient de diferite

- Fiecare clasă va avea un reprezentant (e.g. media datelor din clasă) care va fi considerat centrul clasei

- Algoritmii pentru determinarea reprezentanților clasei sunt cunoscuți sub numele de algoritmi partiționali (realizează o partiționare a spațiului de intrare)

Algoritm clasic: K-means

Rețele cu funcții radiale

Varianta incrementală:

- Se pornește cu un număr mic de centri inițializați aleator

- Se parcurge setul de antrenare:
  - Dacă există un centru suficient de similar cu data de intrare atunci componentele centrului respectiv se modifică pentru a asigura asimilarea datei de intrare în clasa aferentă centrului.
  - Dacă data de intrare este diferită semnificativ de toți centrii atunci este adăugat un nou centru (echivalent cu adăugarea unei noi unități ascunse) care este inițializat chiar cu data de intrare analizată

Obs: necesită definirea unor valori prag care sa permită cuantificarea a ceea ce reprezintă suficient de similar/diferit

Rețele cu funcții radiale

Antrenare incrementală pentru rețele RBF

$K = K_0$

$C_i^k = select(\{X_1^i, \ldots, X_L^i\}), i = 1..N; k = 1..K$

$t = 0$

REPEAT

  FOR l = 1, L DO

    determina $k^* \in \{1, \ldots, K\}$ astfel incat $d(X^l, C^{k^*}) \leq d(X^l, C^k)$ pt orice k

    IF $d(X^l, C^{k^*}) < \delta$ THEN $C^{k^*} := C^{k^*} + \eta \cdot (X^l - C^{k^*})$

    ELSE $K = K + 1; C^K = X^l$

$t = t + 1$

$\eta = \eta_0 t^{-\alpha}$

UNTIL $t > t_{\max}$ OR $\eta < \varepsilon$

Rețele cu funcții radiale

Estimarea lărgimilor câmpurilor receptive.

- Reguli euristice:

$$\sigma = \frac{d_{\max}}{\sqrt{2K}},\ d_{\max} = \text{distanta maxima dintre centri}$$

$$\sigma_k = \gamma d(C^k, C^j), C^j = \text{centrul cel mai apropiat de } C^k, \gamma \in [0.5,1]$$

$$\sigma_k = \frac{1}{m}\sum_{j=1}^{m} d(C^k, C^j), C^1, ..., C^m : \text{cei mai apropiati } m \text{ centri de } C^j$$

  - Proces iterativ intercalat:
    - Fixează valorile σ și optimizează valorile centrilor
    - Fixează valorile centrilor și optimizează valorile σ

Rețele cu funcții radiale

Determinarea ponderilor conexiunilor dintre nivelul ascuns și nivelul de ieșire:

- Problema este similară cu cea a antrenării unei rețele cu un singur nivel de unități cu funcții liniare de activare sau cu cea a estimării parametrilor unui model liniar de regresie

$$E(W)=\frac{1}{2}\sum_{l=1}^{L}\sum_{i=1}^{M}\left(d_i^l-\sum_{k=1}^{K}w_{ik}g_k^l\right)^2,\ g_k^l=g(\left\|x^l-C^k\right\|)$$

$$E(W)=\frac{1}{2}\sum_{l=1}^{L}\left\|d^l-Wg^l\right\|^2=\frac{1}{2}\sum_{l=1}^{L}(d^l-Wg^l)^T(d^l-Wg^l)$$

$$\nabla E(W)=-\sum_{l=1}^{L}(g^l)^T(d^l-Wg^l)=0$$

$$G^TGW=G^Td$$

$$W=(G^TG)^{-1}G^Td$$

Rețele cu funcții radiale

Determinarea ponderilor conexiunilor dintre nivelul ascuns și nivelul de ieșire:

- Problema este similară cu cea a antrenării unei rețele cu un singur nivel de unități cu funcții liniare de activare
- Algoritm: Widrow-Hoff (caz particular al algoritmului BackPropagation)

- Initializare:
  - $w_{ij}(0):=rand(-1,1)$ (ponderile sunt inițializate aleator în [-1,1]),
  - p:=0 (contor de iterații)
- Proces iterativ

REPEAT
  FOR l:=1,L DO
    Calculeaza $y_i(l)$ si $delta_i(l)=d_i(l)-y_i(l)$, i=1,M
    Ajusteaza ponderile: $w_{ik}:=w_{ik}+eta*delta_i(l)*z_k(l)$
  ENDFOR
  Calculează E(W) pentru noile valori ale ponderilor
  p:=p+1
UNTIL E(W)<E* OR p>pmax

Comparație: RBF vs. BP

Rețele de tip RBF:

- 1 nivel ascuns
- Funcții de agregare bazate pe distanțe (pt. nivelul ascuns)
- Funcții de activare cu simetrie radială (pt. nivelul ascuns)
- Unități de ieșire cu funcție liniară
- Antrenare separată a parametrilor adaptivi
- Similare cu tehnicile de aproximare locală

Rețele de tip BackPropagation(BP):

- Mai multe nivele ascunse
- Funcții de agregare bazate pe suma ponderată
- Funcții de activare sigmoidale (pt. nivelul ascuns)
- Unități de ieșire liniare sau neliniare
- Antrenare simultană a parametrilor adaptivi
- Similare cu tehnicile de aproximare globală

Regresie de tip Nearest Neighbour

Ideea principală:

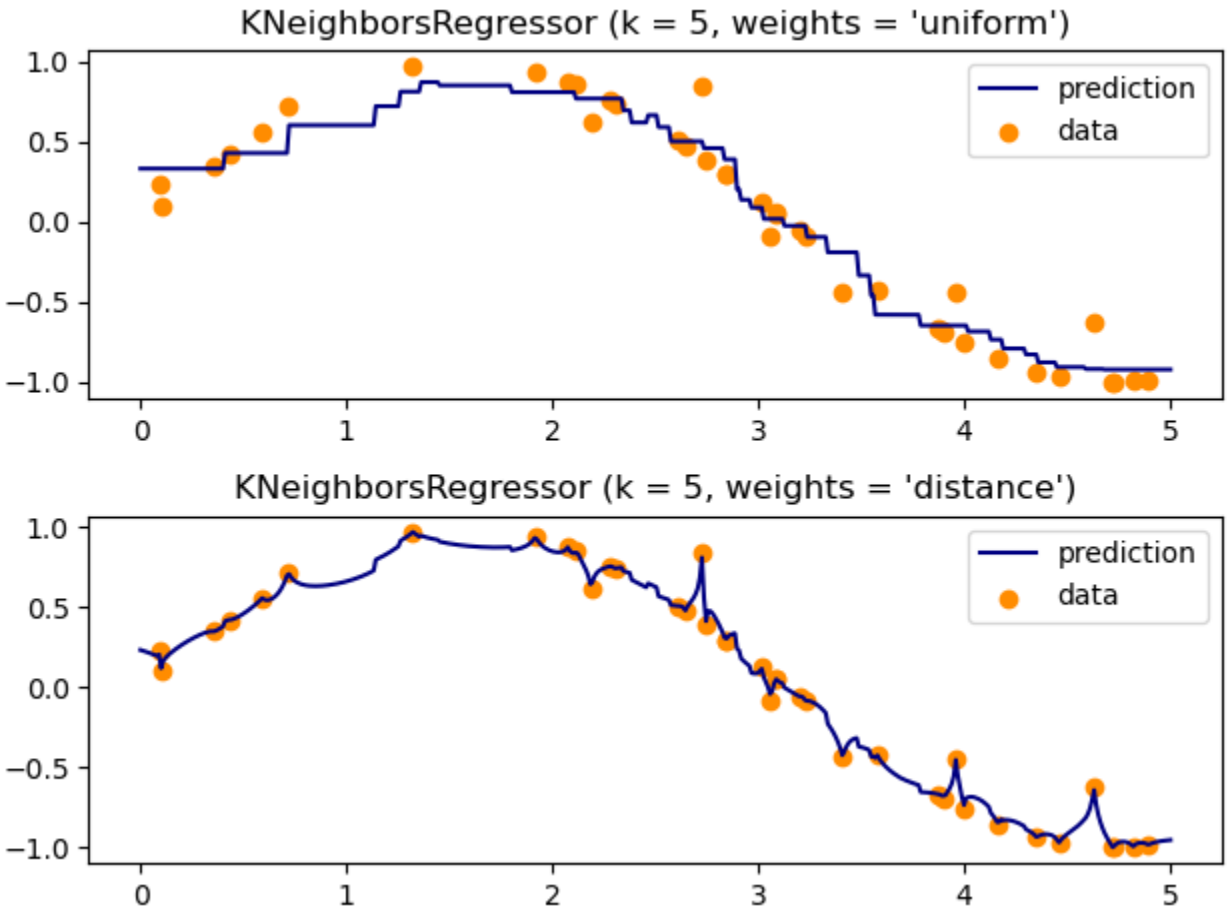
- pentru a estima valoarea asociată unui argument x (vector de variabile predictor) se determină cei mai apropiați vecini
  - Cei mai apropiați k vecini ai lui x
  - Toți vecinii aflați într-o vecinătate de rază R a lui x
- se calculează media (simplă sau ponderată) a valorii corespunzătoare vecinilor

Obs: ponderea corespunzătoare unui vecin poate fi stabilită folosind inversul distanței dintre elementul vecin și cel pentru care se dorește estimarea

Regresie de tip Nearest Neighbour

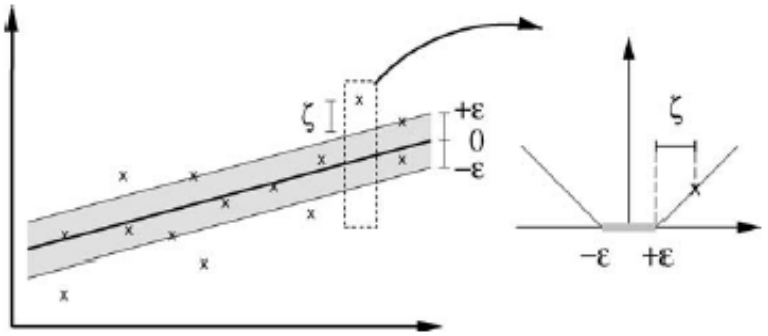
Exemplu:

(https://scikit-learn.org/stable/auto_examples/neighbors/plot_regression.html)



Support Vector Regression

- Idee:
  - Se caută o funcție f cât mai plată (cât mai apropiată de o funcție constantă) astfel încât cât mai multe dintre valorile y din setul de antrenare să se afle la o diferență mai mică decât o valoare eps prespecificată



[Smola, Scholkopf – A Tutorial on Support Vector Regression, 2004]

$$\text{minimize} \quad \frac{1}{2}\|w\|^2$$

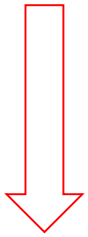
$$\text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon \end{cases}$$

Support Vector Regression

- Introducere variabile de “relaxare” (slack variables):

$$\text{minimize} \quad \frac{1}{2}\|w\|^2$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon \end{cases}$$



$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*)$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

- Parametrul C controlează compromisul între “platitudinea” functiei f și măsura în care se acceptă deviație mai mare decât eps
- Obs: numărul de exemple este notat cu l

[Smola, Scholkopf – A Tutorial on Support Vector Regression, 2004]

Support Vector Regression

- Problema de optimizare cu restricții se poate rezolva aplicând tehnica multiplicatorilor lui Lagrange (ca în cazul clasificatorilor de tip SVM)

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*)$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w, x_i\rangle - b \le \varepsilon + \xi_i \\ \langle w, x_i\rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases}$$

Problema dual:

$$L := \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*) - \sum_{i=1}^{\ell}(\eta_i\xi_i + \eta_i^*\xi_i^*)$$

$$- \sum_{i=1}^{\ell}\alpha_i(\varepsilon + \xi_i - y_i + \langle w, x_i\rangle + b)$$

$$- \sum_{i=1}^{\ell}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle w, x_i\rangle - b)$$

Obs: problema se reduce la a minimiza L în raport cu w, b, ξ și multiplicatorii α, η

$$f(x) = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)\langle x_i, x\rangle + b$$

[Smola, Scholkopf – A Tutorial on Support Vector Regression, 2004]

Support Vector Regression

- Problema de optimizare cu restricții se poate rezolva aplicând tehnica multiplicatorilor lui Lagrange (ca în cazul clasificatorilor de tip SVM)

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*)$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w, x_i\rangle - b \le \varepsilon + \xi_i \\ \langle w, x_i\rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases}$$

Problema duală:

$$L := \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*) - \sum_{i=1}^{\ell}(\eta_i\xi_i + \eta_i^*\xi_i^*)$$

$$- \sum_{i=1}^{\ell}\alpha_i(\varepsilon + \xi_i - y_i + \langle w, x_i\rangle + b)$$

$$- \sum_{i=1}^{\ell}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle w, x_i\rangle - b)$$

Obs: problema se reduce la a minimiza L în raport cu w, b, ξ și multiplicatorii α, η

Modelul de regresie

sau altă fcție nucleu (regresie neliniară)

$$f(x) = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)\langle x_i, x\rangle + b$$

[Smola, Scholkopf – A Tutorial on Support Vector Regression, 2004]

Sumar

- Modelele liniare de regresie au avantajul că sunt bine fundamentate matematic și ușor de interpretat
- Modelele neliniare în raport cu variabilele predictor, dar liniare în raport cu coeficienții permit utilizarea tehnicilor de estimare a parametrilor folosite în cazul modelelor liniare; performanța depinde de tipul de funcții de bază utiilizate
- O parte dintre modelele de clasificare au corespondent și în cazul problemelor de regresie:
  - Arbori de regresie
  - Regresie bazată pe cel mai apropiat vecin
  - Modele de regresie bazate pe vectori suport
- Cele mai flexibile modele (dar și mai dificil de interpretat) sunt cele bazate pe rețele neuronale

Curs următor

- Analiza seriilor de timp
  - Pre-procesarea seriilor de timp
  - Predicţie
  - Identificare şabloane
  - Grupare şi clasificare
  - Detecţie anomalii