Curs 5:

Clasificarea datelor (III)

kNearest Neighbours și Naive Bayes

Structura

- Clasificatori bazați pe instanțe
  - Modelul celor mai apropiați vecini (k-Nearest Neighbours)
- Clasificatori bazaţi pe modele probabiliste
  - Modelul Bayesian naiv (Naive Bayes)

Clasificatori bazaţi pe instanţe

Ideea principală: datele similare aparţin aceleiaşi clase

(raționamentul bazat pe analogie este utilizat în multe domenii – exemplu: ce boală a avut un pacient cu simptome similare?)

- Modelul de clasificare constă tocmai din setul de antrenare
  - Procesul de antrenare constă doar în stocarea datelor din set
- Clasificarea unei noi date constă în:
  - Se calculează similaritatea (sau disimilaritatea) dintre noua dată şi cele din setul de antrenare şi se identifică exemplarele cele mai apropiate
  - Se alege clasa cea mai frecvent întâlnită în subsetul celor mai similare exemple

Clasificatori bazaţi pe instanţe

Ideea principală: datele similare aparţin aceleiaşi clase

Obs:

- Astfel de clasificatori sunt consideraţi leneşi (“lazy”) deoarece faza de antrenare nu presupune nici un efort de calcul (întregul efort este amânat pentru faza de clasificare)

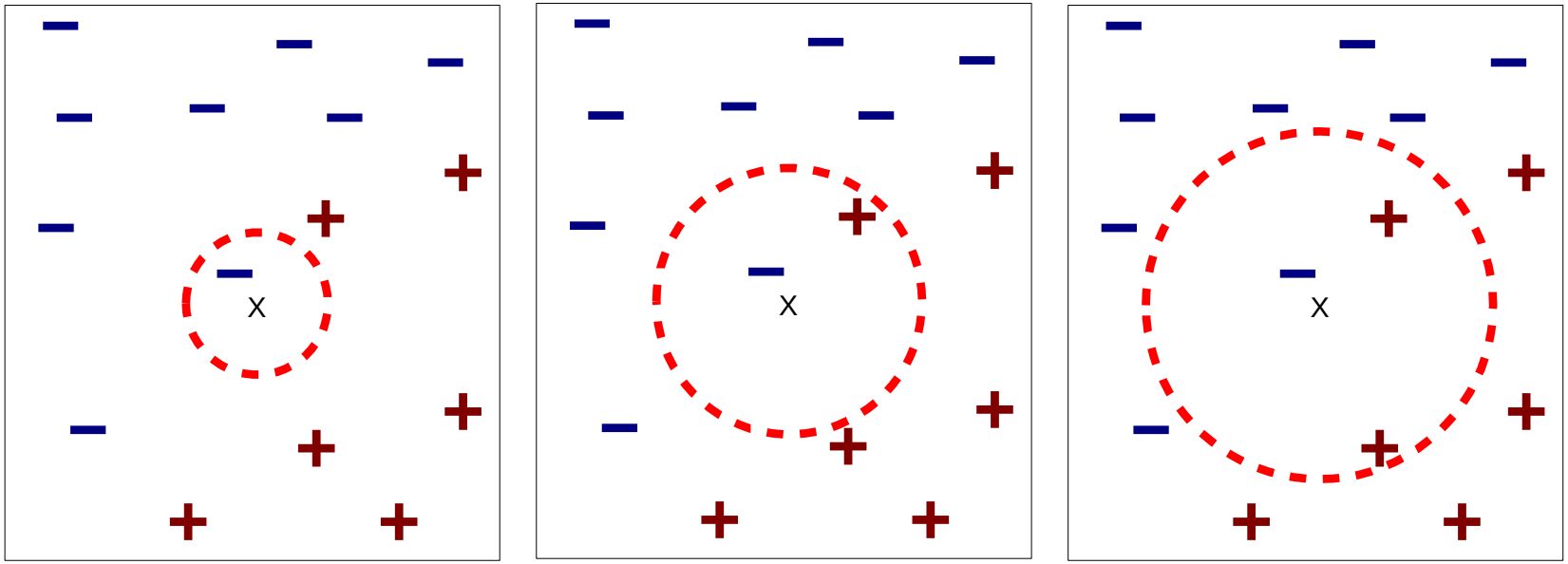
- Cei mai populari clasificatori din această categorie sunt cei bazaţi pe principiul celui/celor mai apropiat/apropiaţi vecin/vecini (k-Nearest Neighbour)

- Aplicații:
  - sisteme de recomandare
  - diagnoza medicală

Clasificatori bazaţi pe instanţe

kNN – k Nearest Neighbour

- Pt fiecare dată de clasificat:
  - Determină cele mai apropiate (mai similare) k exemple din setul de antrenare
  - Identifică cea mai frecventă clasă



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

[Tan, Steinbach, Kumar; Introduction to Data Mining, slides, 2004]

Clasificatori bazaţi pe instanţe

kNN – k Nearest Neighbour

- Pt fiecare dată de clasificat:
  - Determină cele mai apropiate (mai similare) k exemple din setul de antrenare
  - Identifică cea mai frecventă clasă

Performanţa clasificatorilor de tip kNN depinde de:

- Măsura de similaritate/ disimilaritate
  - Se alege în funcţie de tipurile atributelor şi de proprietăţile problemei
- Valoarea lui k (numărul de vecini)
  - Cazul cel mai simplu: k=1 (nu e indicat în cazul datelor afectate de zgomot)

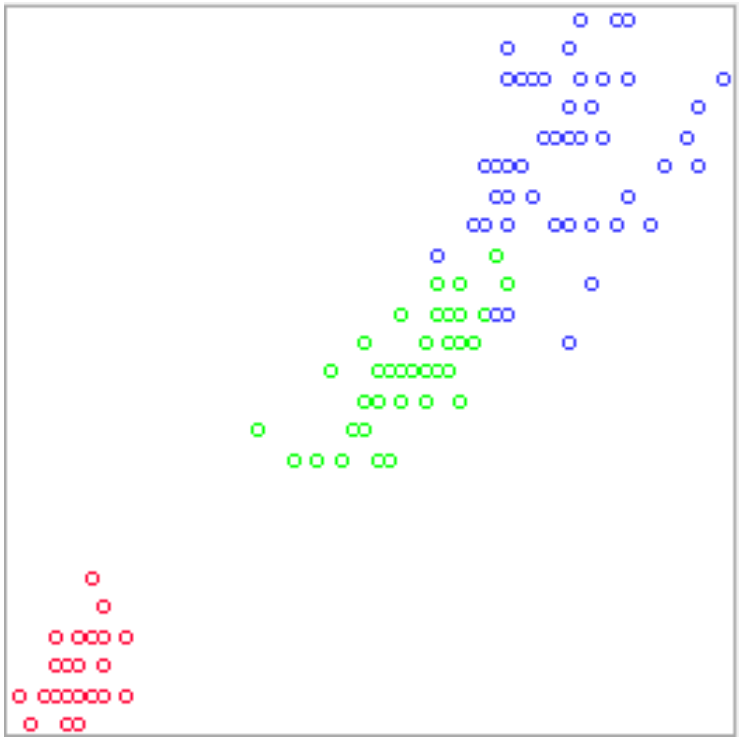
Obs: kNN induce o partiţionare în regiuni a spaţiului datelor; regiunile nu sunt calculate explicit ci sunt implicit determinate de măsura de similaritate (precum şi de valoarea lui k)

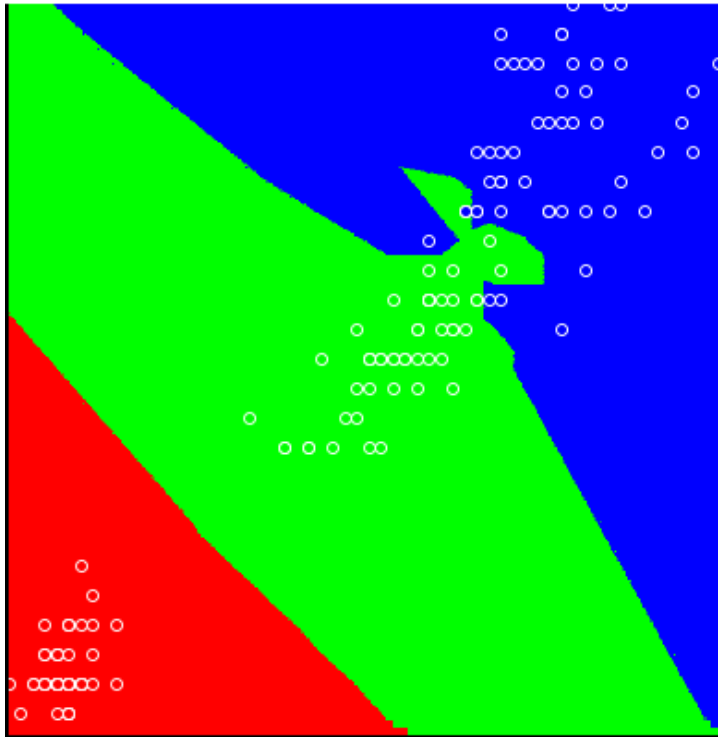
Clasificatori bazaţi pe instanţe

1NN = Nearest Neighbor bazat pe cel mai apropiat vecin (şi distanţa euclidiană)

Ilustrarea regiunilor. Dataset: iris2D (“petal length” and “petal width”).

Plot: Weka->Visualization->BoundaryVisualizer





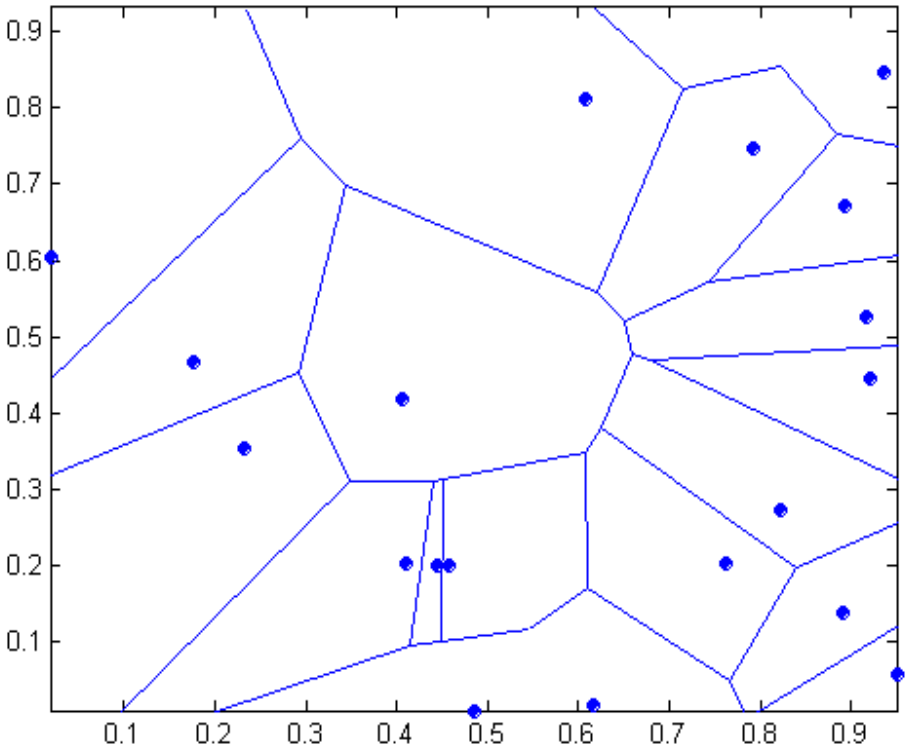
Clasificatori bazaţi pe instanţe

1NN = Nearest Neighbor bazat pe cel mai apropiat vecin (şi distanţa euclidiană)

1NN induce o partiţionare a spaţiului datelor (e.g. în 2D aceasta corespunde unei diagrame Voronoi)

Obs:

Fiecare instanţă din setul de antrenare (punctele din imagine) corespunde unei regiuni care cuprinde datele aflate în vecinătatea acelei instanţe



[Tan, Steinbach, Kumar; Introduction to Data Mining, slides, 2004]

Măsuri de similaritate/ disimilaritate

Considerăm două entităţi (e.g. vectori de date, serii de timp etc) A and B

- O măsură de similaritate, S, asociază perechii (A,B) un număr, S(A,B) , care este cu atât mai mare cu cât A şi B sunt mai similare
- O măsură de disimilaritate, D, asociază perechii (A,B) un număr, D(A,B), care este cu atât mai mare cu cât A şi B sunt mai diferite

(obs: o măsură de disimilaritate nu satisface neapărat toate proprietățile matematice ale unei metrici, de exemplu poate să nu satisfacă inegalitatea triunghiului)

Alegerea măsurii depinde de:

- Tipul atributelor
- Numărul de atribute
- Distribuţia datelor
- Particularitățile problemei

Măsuri de similaritate/ disimilaritate

Atribute numerice

Cele mai populare măsuri de disimilaritate:

- Distanţa euclidiană
- Distanţa Manhattan

Obs:

- Distanţa euclidiană este invariantă în raport cu rotaţii
- Dacă nu toate atributele au aceeaşi importanţă (sau dacă nu sunt scalate adecvat) atunci e indicat să se folosească varianta ponderată

  ($w_i(a_i-b_i)^2$ în loc de $(a_i-b_i)^2$)

$$d_p(A,B) = \sqrt[p]{\sum_{i=1}^{n}(a_i - b_i)^p} \quad \text{(Minkowski, } L_p\text{)}$$

$$d_E(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \quad \text{(Euclidean, p = 2)}$$

$$d_M(A,B) = \sum_{i=1}^{n}|a_i - b_i| \quad \text{(Manhattan, p = 1)}$$

$$d_\infty(A,B) = \max_{i=\overline{1,n}} |a_i - b_i| \quad (p = \infty)$$

Obs.

- Ponderile pot fi determinate folosind tehnici de preprocesare
- Pentru a evita utilizarea ponderilor datele pot fi în prealabil scalate sau standardizate

Măsuri de similaritate/ disimilaritate

Atribute numerice

Cele mai populare măsuri de disimilaritate:

- Distanţa euclidiană
- Distanţa Manhattan

$$d_p(A,B) = \sqrt[p]{\sum_{i=1}^{n}(a_i - b_i)^p} \quad \text{(Minkowski, } \mathrm{L_p}\text{)}$$

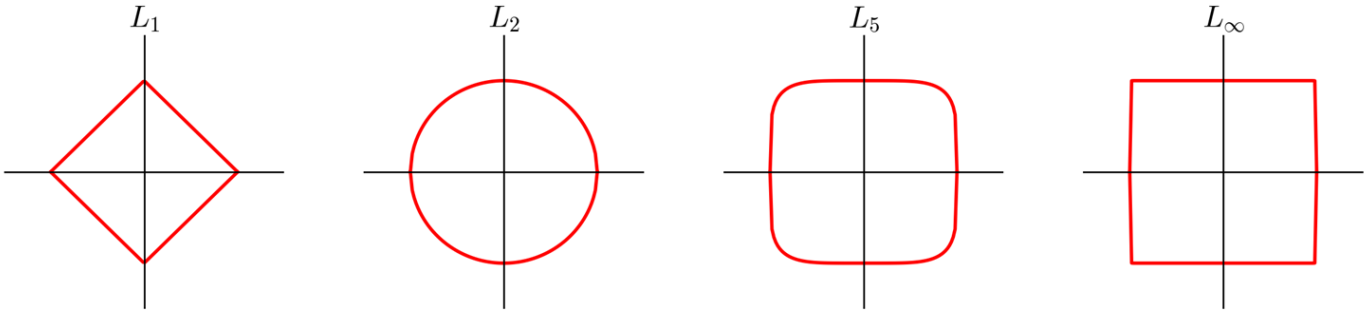
$$d_E(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \quad \text{(Euclidean, p = 2)}$$

$$d_M(A,B) = \sum_{i=1}^{n}|a_i - b_i| \quad \text{(Manhattan, p = 1)}$$

$$d_\infty(A,B) = \max_{i=\overline{1,n}} |a_i - b_i| \quad (\mathrm{p} = \infty)$$

Diferențe între măsurile de disimilaritate

(vizualizarea punctelor egal distanțate de originea sistemului de axe de coordonate)



Source: Lecture 19 - Data Science, Steven Skiena, Stony Brook University

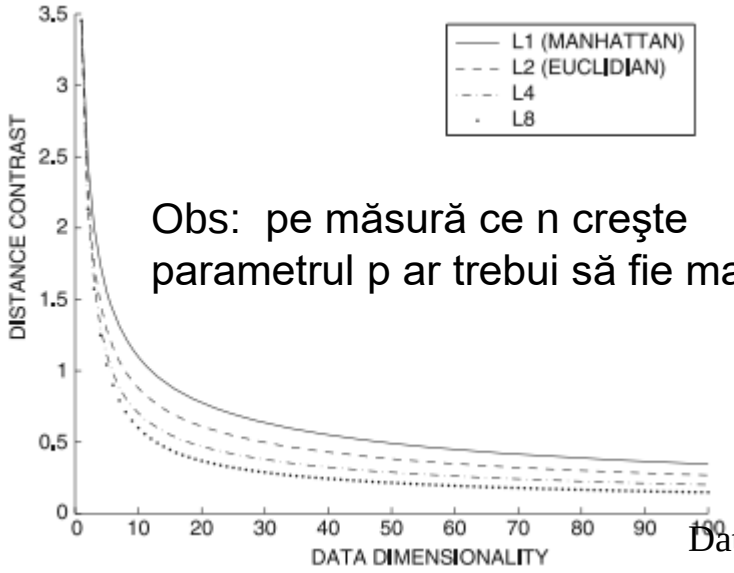
Măsuri de similaritate/ disimilaritate

Aspecte practice – problema dimensiunii (dimensionality curse):

- Puterea de discriminare a acestor distanţe scade pe măsură ce nr de atribute (n) creşte →

  pt date cu multe atribute clasificatorii bazaţi pe distanţe devin inefectivi

- Se recomandă reducerea dimensionalității (de ex. prin PCA sau transformări neliniare – tSNE, UMAP)



Obs: pe măsură ce n creşte parametrul p ar trebui să fie mai mic

$$d_p(A,B) = \sqrt[p]{\sum_{i=1}^{n}(a_i - b_i)^p} \quad \text{(Minkowski, } L_p\text{)}$$

$$d_E(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \quad \text{(Euclidean, p = 2)}$$

$$d_M(A,B) = \sum_{i=1}^{n}|a_i - b_i| \quad \text{(Manhattan, p = 1)}$$

$$d_\infty(A,B) = \max_{i=\overline{1,n}}|a_i - b_i| \quad (p = \infty)$$

Distance contrast = $\frac{d_{max} - d_{min}}{\sigma}$

d_{max}, d_{min}= cea mai mare si cea mai mica dintre distante

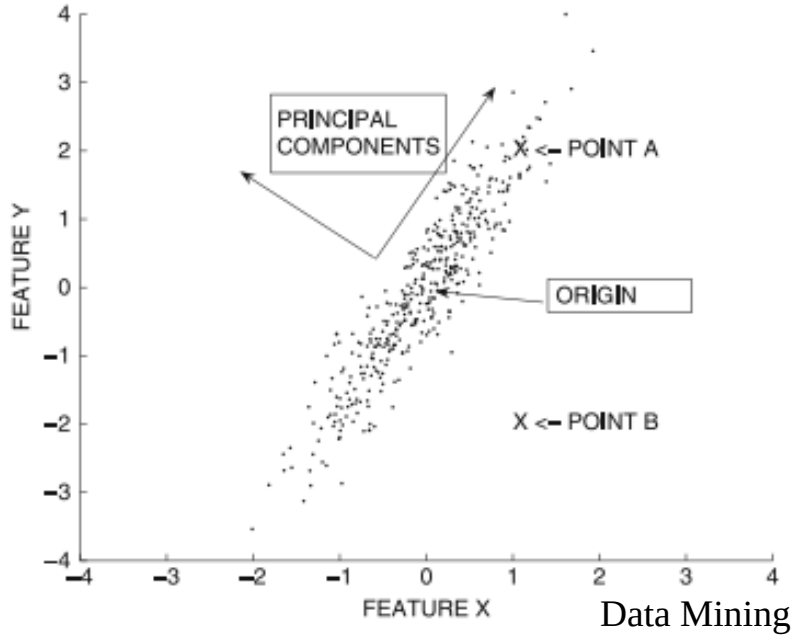
σ = abaterea standard a distantelor

[Aggarwal, Data Mining Textbook, 2015]

Măsuri de similaritate/ disimilaritate

Aspecte practice – impactul distribuţiei datelor

Intrebare: Care punct e mai aproape de origine? A sau B?



[Aggarwal, Data Mining Textbook, 2015]

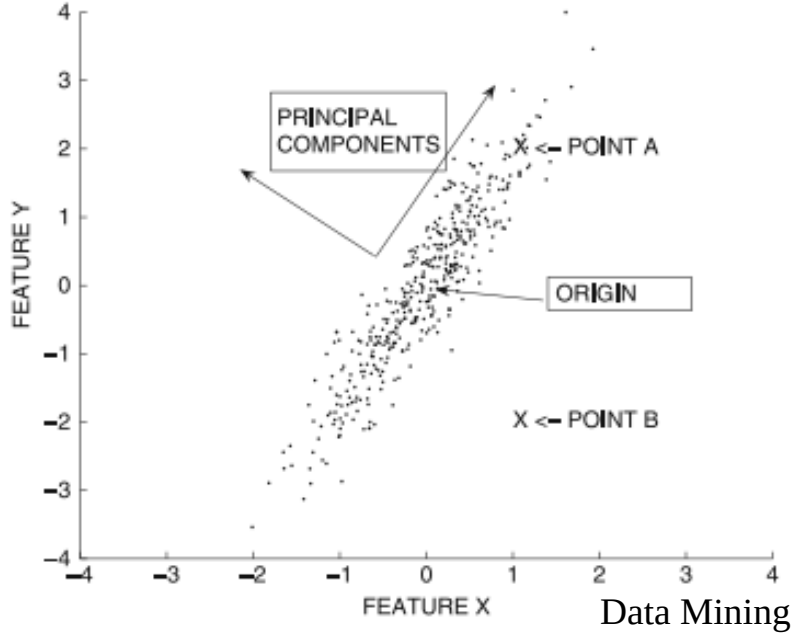
Măsuri de similaritate/ disimilaritate

Aspecte practice – impactul distribuţiei datelor

Intrebare: Care punct e mai aproape de origine? A sau B?

R: d(O,A) = d(O,B) (distanţe euclidiene egale). Luând în considerare distribuţia datelor: A este mai apropiat de O decât B

Altă întrebare: cum poate fi inclusă distribuţia datelor în calculul distanţei?



Distanta Mahalanobis

$d_{Mah}(A,B) = \sqrt{(A-B)^T C^{-1}(A-B)}$

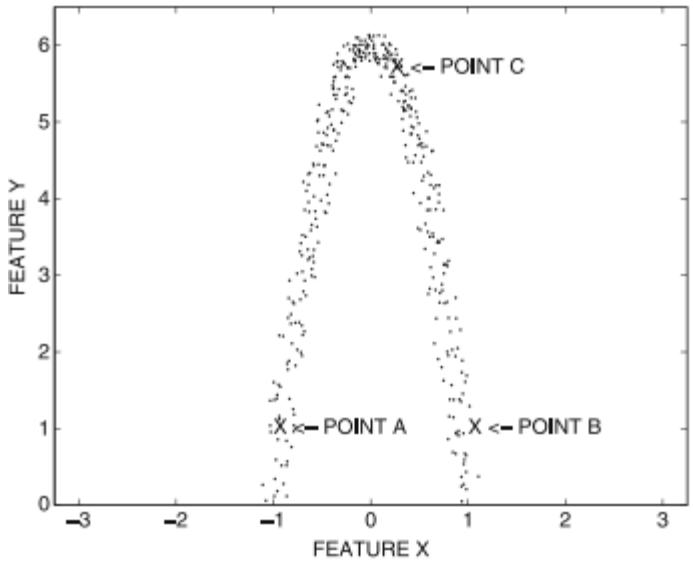
C^{-1} = inversa matricii de covarianta

[Aggarwal, Data MiningTextbook, 2015]

Măsuri de similaritate/ disimilaritate

Aspecte practice – impactul distribuţiei datelor

Intrebare: este distanţa dintre A şi B mai mică decât distanţa dintre B şi C?



[Aggarwal, Data Mining Textbook, 2015]

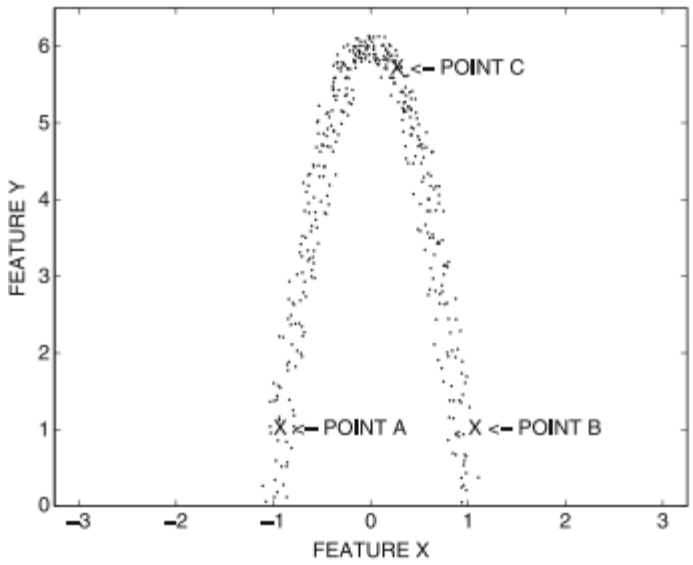
Măsuri de similaritate/ disimilaritate

Aspecte practice – impactul distribuţiei datelor

Intrebare: este distanţa dintre A şi B mai mică decât distanţa dintre B şi C?

R: da, dacă ignorăm distribuţia datelor şi folosim distanţa euclidiană

Totuşi, distribuţia datelor nu poate fi ignorată întrucât este cea care furnizează contextul problemei, iar în acest context d(A,B)>d(B,C)



Distanţa geodesică:

- Se construieşte un graf ce are în noduri punctele iar muchiile unesc nodurile vecine (ex: cei mai apropiaţi k vecini)
- Calculează distanţa dintre două puncte ca fiind cea mai scurtă cale în graf

[Aggarwal, Data Mining Textbook, 2015]

Măsuri de similaritate/ disimilaritate

Atribute numerice – măsură de similaritate

- Măsura cosinus: $sim(A,B)=A^TB/(||A||\ ||B||)$ (produsul scalar dintre A şi B împărţit la produsul normelor)

Remarcă:

- In cazul vectorilor normalizaţi (||A||=||B||=1) similaritatea e maximă când distanţa euclidiană este minimă:

$$d_E^2(A,B) = (A-B)^T(A-B) = A^TA - 2A^TB + B^TB$$

$$= 2(1 - A^TB) = 2(1 - sim(A,B))$$

Măsuri de similaritate/ disimilaritate

Atribute nominale

Abordare 1: Transformarea atributelor nominale în atribute numerice (prin binarizare = one hot encoding) şi utilizarea măsurilor de similaritate/disimilaritate pentru vectori binari:

- Disimilaritate: distanţa Hamming = distanţa Manhattan: $d_H(A,B)=d_M(A,B)$

- Jaccard similarity:

$$J(A,B) = \frac{\sum_{i=1}^{n} a_i b_i}{\sum_{i=1}^{n}(a_i^2 + b_i^2 - a_i b_i)} = \frac{card(S_A \cap S_B)}{card(S_A \cup S_B)}$$

Obs: S_A şi S_B sunt submulţimi ale mulţimii globale cu n atribute care corespund vectorilor de apartenenţă A şi B.

Măsuri de similaritate/ disimilaritate

Atribute nominale

Abordare 2: Utilizează măsuri locale de similaritate (între valorile atributelor)

$$S(A,B) = \sum_{i=1}^{n} S(a_i, b_i)$$

$$S(a_i, b_i) = \begin{cases} 1 & \text{if } a_i = b_i \\ 0 & \text{if } a_i \neq b_i \end{cases}$$

Obs: similarităţile mai puţin frecvente pot fi considerate mai relevante decât cele frecvente

$$S(a_i, b_i) = \begin{cases} 1/f^2(a_i) & \text{if } a_i = b_i \\ 0 & \text{if } a_i \neq b_i \end{cases}$$

$f(a_i) = \text{frecventa valorii } a_i \text{ in setul de date} \quad \text{(pt atributul i)}$

Măsuri de similaritate/ disimilaritate

Atribute mixte: se combină măsurile corespunzătoare celor două tipuri de atribute (utilizând ponderi specifice)

$$S(A,B) = \lambda S_{numerical}(A,B) + (1-\lambda) S_{nominal}(A,B)$$

Alte tipuri de date:

- Siruri (e.g. text sau secvenţe biologice) – se utilizează distanţa de editare (distanța Levenshtein)
- Concepte (e.g. noduri într-o ontologie) – distanţe bazate pe cele mai scurte căi în grafuri sau arbori
- Grafuri (e.g. reţele sociale sau biologice) – ponderea structurilor (tiparelor) similare în cele două structuri

kNN: alegerea lui k

Performanţa clasificatorilor de tip kNN depinde de numărul de vecini

Cazuri extreme:

- k=1 - clasificatorul nu este robust (erorile din setul de date influenţează răspunsul clasificatorului)

- k=N - e echivalent cu ZeroR fiind bazat doar pe modul de distribuire a datelor în clase

Cum se alege k?

- Abordare de tip trial-and-error: se încearcă diferite valori şi se alege valoarea care maximizează performanţa

kNN: cost

Clasificarea unei noi instanţe necesită calculul a N distanţe (sau măsuri de similaritate pt un set de date cu N elemente care au n atribute precum şi selecţia celor mai mici k distanţe ➔ O(Nn+kN) (costul de calcul a similarității / disimilarității poate fi diferit de Nn – depinde de structura datelor)

Dacă N e mare această prelucrare poate fi costisitoare (întrucât trebuie efectuată pentru fiecare instanţă care trebuie clasificată)

Abordări posibile:

- Crearea unei structuri de indexare a datelor din setul de antrenare care permite identificarea celor mai apropiaţi vecini într-un mod eficient (arbori k-d = generalizare a arborilor binari de căutare pentru date k-dimensionale; obs: în notațiile noastre k=n si nu are legătură cu numărul de vecini)
- Reducerea numărului de date din setul de antrenare prin gruparea lor în clustere şi înlocuirea fiecărui cluster cu un singur prototip
- Selecţia unor prototipuri din set

Modele probabiliste de clasificare

Exemplu: Presupunem că ne interesează să estimăm probabilitatea ca un pacient care are simptomul S să aibă boala T

- Probabilitatea de estimat: P(T|S) = probabilitatea evenimentului T condiţionată de evenimentul S

- Presupunem că se cunosc:

  - P(S) – probabilitatea ca simptomul sa fie prezent (evidence)
  - P(T) – prevalența bolii T estimată pe baza unor studii populaționale (e o măsură a frecvenţei de apariţie a bolii) (prior)
  - P(S|T) – se estimează pe baza cunoştinţelor medicale (cât de frecvent este simptomul S în cazul bolii T) (likelihood)

- Regula de calcul (regula probabilităţii condiţionate – formula lui Bayes):

$$P(T|S)=P(S|T)P(T)/P(S) = likelihood*prior/evidence$$

- Cum se analizează cazul în care nu e un singur simptom S, ci mai multe simptome $S_1,S_2,\ldots,S_n$?

Modele probabiliste de clasificare

Exemplu: Presupunem că ne interesează să estimăm probabilitatea ca un pacient care are simptomele $S_1,S_2,\ldots,S_n$ să aibă boala T

- Probabilitatea de estimat: $P(T| S_1,S_2,\ldots,S_n)$
- Se foloseşte regula Bayes:
  - $P(T| S_1,S_2,\ldots,S_n)=P(S_1,S_2,\ldots,S_n|T)P(T)/P(S_1,S_2,\ldots,S_n)$
- Ipoteză simplificatoare: simptomele $(S_1,S_2,\ldots,S_n)$ corespund unor evenimente independente în condițiile în care clasa este cunoscută (această ipoteză nu este întotdeauna adevărată însă poate fi acceptată în anumite situaţii practice)
- Intrucât $P(S_1,S_2,\ldots,S_n)$ nu depinde de T, dacă se dorește doar să se determine clasa (T pozitiv sau T negativ) este suficient să se considere că $P(T| S_1,S_2,\ldots,S_n)$ este proporțional cu $P(S_1|T)\,P(S_2|T)\ldots P(S_n|T)P(T)$

Clasificatorul Naïve Bayes

Problema de clasificare:

- Pentru o dată $D_i=(a_{i1},a_{i2},\ldots,a_{in})$ se pune problemă determinării clasei căreia îi aparţine

Ideea principală

- Estimează $P(C_k| D_i)=P(a_{i1},a_{i2},\ldots,a_{in}|C_k)P(C_k)/ P(a_{i1},a_{i2},\ldots,a_{in})$ pt fiecare k din {1,2,…,K} şi selectează probabilitatea maximă; aceasta va indica cărei clase îi aparţine, cel mai probabil, data; întrucât $P(a_{i1},a_{i2},\ldots,a_{in})$ este aceeași indiferent de clasă, probabilitatea de la numitor nu influențează decizia (se poate considera egală cu 1)

- Ipoteză simplificatoare: atributele sunt independente (acesta este motivul pentru care clasificatorul este denumit "naiv")

- $P(C_k| D_i)= P(a_{i1}|C_k)\ P(a_{i2}|C_k)\ldots P(a_{in}|C_k)P(C_k)$

- Estimarea probabilităţii de clasificare necesită cunoaşterea lui $P(a_{i1}|C_k)$, $P(a_{i2}|C_k)$, …, $P(a_{in}|C_k)$ şi $P(C_k)$
- Aceste probabilităţi pot estimate pe baza setului de antrenare (ca frecvenţe relative) – această estimare corespunde procesului de învăţare specific clasificatorului Naïve Bayes

Clasificatorul Naïve Bayes

Exemplu:

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	**play** Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

P(C1)=P(no)=5/14 P(C2)= P(yes)=9/14

A1: outlook

P(sunny|C1)=P(sunny,C1)/P(C1)
=(3/14)/(5/14)=3/5

P(sunny|C2)=P(sunny,C2)/P(C2)
=(2/14)/(9/14)=2/9

P(overcast|C1)=P(overcast,C1)/P(C1)
=0

P(overcast|C2)=P(overcast,C2)/P(C2)
=(4/14)/(9/14)=4/9

P(rainy|C1)=P(rainy,C1)/P(C1)
=(2/14)/(5/14)=2/5

P(rainy|C2)=P(rainy,C2)/P(C2)
=(3/14)/(9/14)=3/9

Clasificatorul Naïve Bayes

Exemplu:

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	**play** Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

P(C1)=P(no)=5/14 P(C2)= P(yes)=9/14

A2: temperature

P(hot|C1)=P(hot,C1)/P(C1)=2/5

P(hot|C2)=P(hot,C2)/P(C2)=2/9

P(mild|C1)=P(mild,C1)/P(C1)=2/5

P(mild|C2)=P(mild,C2)/P(C2)=4/9

P(cool|C1)=P(cool,C1)/P(C1)
=(2/14)/(5/14)=1/5

P(cool|C2)=P(cool,C2)/P(C2)=2/9

Clasificatorul Naïve Bayes

Exemplu:

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	**play** Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

P(C1)=P(no)=5/14 P(C2)= P(yes)=9/14

A3: humidity

P(high|C1)=P(high,C1)/P(C1)=4/5

P(high|C2)=P(high,C2)/P(C2)=3/9

P(normal|C1)=P(normal,C1)/P(C1)=1/5

P(normal|C2)=P(normal,C2)/P(C2)=6/9

Clasificatorul Naïve Bayes

Exemplu:

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	**play** Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

P(C1)=P(no)=5/14 P(C2)= P(yes)=9/14

A4: windy

P(FALSE|C1)=P(FALSE,C1)/P(C1)=2/5

P(FALSE|C2)=P(FALSE,C2)/P(C2)=6/9

P(TRUE|C1)=P(TRUE,C1)/P(C1)=3/5

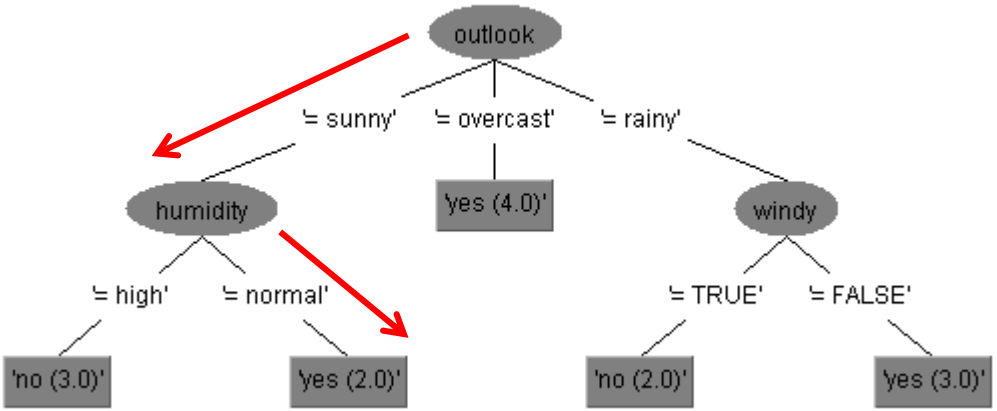
P(TRUE|C2)=P(TRUE,C2)/P(C2)=3/9

Clasificatorul Naïve Bayes

Exemplu:

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	**play** Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no



D=(outlook=sunny, temperature=mild, humidity=normal, windy=False)
P(C1|D)=P(sunny|C1)*P(mild|C1)*P(normal|C1)*P(FALSE|C1)*P(C1)/P(D)=
=3/5*2/5*1/5*2/5*5/14/P(D)=60/8750/P(D) = 0.006875/P(D)

P(C2|D)=P(sunny|C2)*P(mild|C2)*P(normal|C2)*P(FALSE|C2)*P(C2)/P(D)=
=2/9*4/9*6/9*6/9*9/14/P(D)=2592/91854=0.028219/P(D) ➔ yes

Clasificatorul Naïve Bayes

Exemplu:

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Obs: dacă pt o anumită valoare de atribut (a_{ij}) şi o anumită clasă C_k nu există exemplu în setul de antrenare, atunci $P(a_{ij}| C_k)=0$ şi (datorită ipotezei de independenţă) pt orice instanţă având valoarea a_{ij} pt atributul A_i , probabilitatea să aparţină clasei C_k este 0.

Această situaţie poate să apară în special în cazul claselor mici.

Tratarea acestor situaţii prin regula de “netezire”:

$P(a_{ij}| C_k)=(count(a_{ij},C_k)+alpha)/(count(C_k)+m_i*alpha)$

alpha = parametru de netezire Laplace (exemplu: alpha=1)

m_i= nr de valori distincte ale atributului A_i

Clasificatorul Naïve Bayes

Obs:

- Acest model poate fi aplicat direct atributelor discrete şi se bazează pe unul din următoarele modele probabiliste:
  - Binomial (pentru atribute binare)
  - Multinomial (pentru atribute discrete)
- In cazul atributelor numerice care iau valori într-un domeniu continuu există două abordări principale:
  - Atributele sunt discretizate înainte de utilizarea clasificatorului (performanţa acestuia depinde de procesul de discretizare)
  - Se folosesc modele probabiliste continue (e.g. Gaussian) cu parametri estimaţi pe baza setului de antrenare

Rețele Bayesiene pentru clasificare

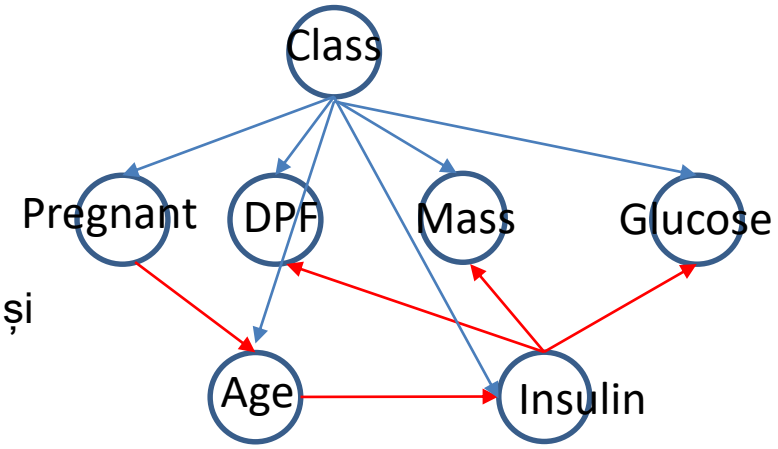
Motivație:

- Ipoteza referitoare la independența atributelor poate să fie prea restrictivă
  - modelul Naive Bayes corespunde unei structuri arborescente cu un singur nivel
  - interacțiunile dintre atribute pot fi modelate utilizând rețele Bayesiene (grafuri orientate aciclice):
    - Nod = atribut
    - Arc = corespunde interacțiunii dintre atribute

Exemplu (Pima dataset)

[Friedman et al, Bayesian Network Classifiers, 1997]

- Naïve Bayes:
  - săgeți albastre (interacțiunea dintre atributul de clasă și atributele predictive)
- Rețea Bayesiană:
  - săgeți albastre + săgeți roșii (interacțiuni între atribute predictive)



Rețele Bayesiene pentru clasificare

- Tree Augmented Naive Bayes (TAN) – [Friedman et al, Bayesian Network Classifiers, 1997]

- Idee
  - Se calculează informația mutuală dintre oricare două atribute și se utilizează ca pondere pentru a defini o rețea de interacțiuni între atribute
  - Se determină arborele de acoperire de pondere maximă (maximum weight spanning tree)
  - Se stabilesc orientările muchiilor astfel încât acestea să pornească de la nodul corespunzător atributului “clasa”

- Implementări
  - R – bnlearn (https://www.bnlearn.com/examples/classifiers/)
  - Python – pyAgrum (https://pyagrum.readthedocs.io/en/latest/skbnClassifier.html)

Sumar

- Clasificatori bazați pe instanțe (k-Nearest Neighbour)
  - Avantaj: Proces de antrenare simplu (doar stocarea exemplelor)
  - Dezavantaj:
    - Costul clasificării poate fi mare dacă nu sunt utilizate structuri eficiente de căutare (pt eficientizare se pot utiliza arbori k-d sau tabele de hashing care țin cont de similaritate)
    - Performanța depinde de măsura de similaritate și de valoarea lui k

Sumar

- Clasificatori de tip Naive Bayes
  - Avantaj:
    - Ușor de construit (se bazează doar pe calcule de frecvențe)
    - Nu necesită un volum mare de date de antrenate
    - Sunt eficienți în faza de clasificare
  - Dezavantaj:
    - In varianta naivă nu țin cont de interacțiunile dintre atribute
    - Valorile estimate ale probabilităților trebuie tratate cu grijă (nu este recomandat să fie utilizate ca atare ci doar în contextul deciziei legat de clasă și care este bazată doar pe compararea valorilor)

Curs următor

Modele funcționale (bazate pe transformări ale datelor folosind diverse funcții)

- Rețele neuronale (Feedforward Neural Networks – Multilayer Peceptrons)
- Modele bazate pe vectori suport (Support Vector Machines)