Capitole Speciale de Informatică

Curs 7: Clasificarea documentelor în spațiu vectorial

8 noiembrie 2018

Pentru o colecție $\mathcal{D}$ de N documente cu termeni t din un vocabular V, an definit

- df_t =numărul de documente în care apare t
- cf_t =numărul de apariții a lui t în colecția $\mathcal{D}$
- $tf_{t,d}$ =numărul de apariții a lui t în documentul $d \in \mathcal{D}$
- $idf_t = \log \frac{N}{df_t}$ (frecvența inversă de document)
- $tf\text{-}idf_{t,d} = tf_{t,d} \cdot idf_t$

Dacă q este o interogare, măsura de scor de potrivire între q și d este

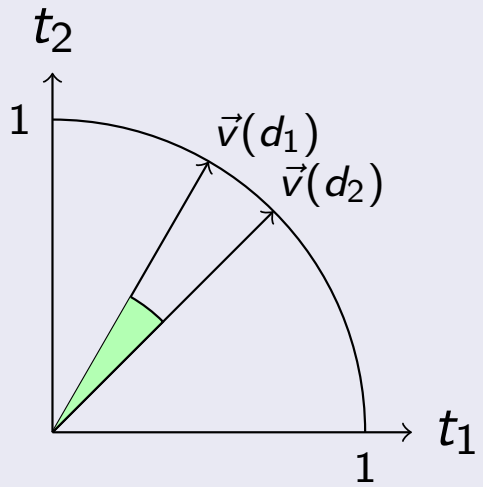
$$scor(q, d) = \sum_{t \in q} tf\text{-}idf_{td}$$

Fie $V = \{t_1, \ldots, t_N\}$ vocabularul de termeni ai documentelor din $\mathcal{D}$

- $d \in \mathcal{D} \mapsto \vec{V}(d) = \langle w(t_1, d), \ldots, w(t_N, d) \rangle$
  unde $w(t_i, d)$ reprezintă greutatea (sau ponderea) termenului t_i în documentul d. Se presupune implicit că
  $$w(t_i, d) = tf\text{-}idf_{t_i,d}$$
- $d \in \mathcal{D} \mapsto \vec{v}(d) = \vec{V}(d)/|\vec{V}(d)|$
  unde $|\vec{V}(d)| = \sqrt{w(t_1, d)^2 + \ldots + w(t_N, d)^2}$ este lungimea euclideană a vectorului $\vec{V}(d)$.
- Similaritatea cosinusoidală a două documente d_1, d_2 este
  $$sim(d_1, d_2) = \vec{v}(d_1) \cdot \vec{v}(d_2)$$
  unde produsul scalar dintre doi vectori este
  $$\langle v_1, \ldots, v_N \rangle \cdot \langle v'_1, \ldots, v'_N \rangle = \sum_{i=1}^{N} v_i \, v'_i.$$

Exemplu ilustrat

$V = \{t_1, t_2\}$



Cosinusul unghiului verde dintre $\vec{v}(d_1)$ și $\vec{v}(d_2)$ reprezintă similaritatea dintre d_1 și d_2. Se observă că

$$-1 \leq sim(d_1, d_2) \leq 1.$$

Clasificarea documentelor reprezentate vectorial

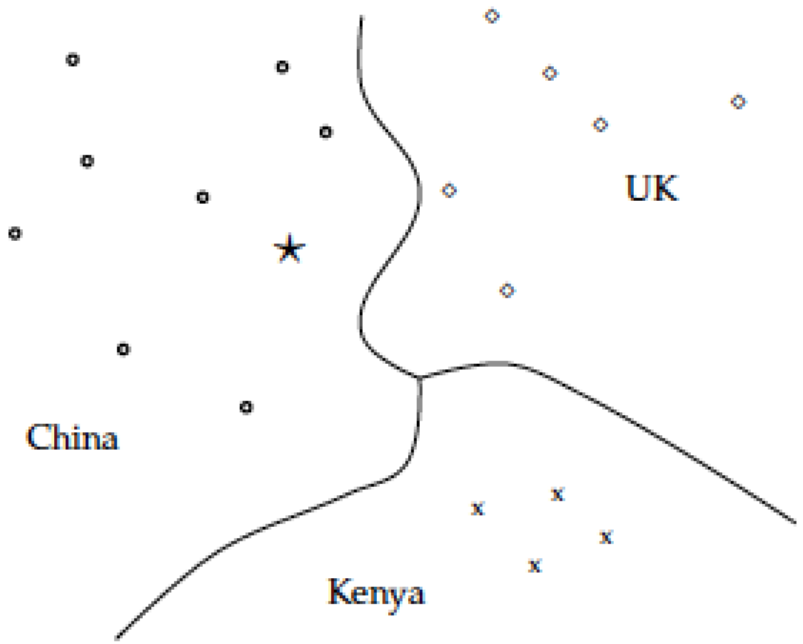
În acest curs vom presupune că fiecare document d este reprezentat ca un vector $\vec{v}(d) = (v_1, \ldots, v_N) \in \mathbb{R}^N$ unde

- $V = \{t_1, t_2, \ldots, t_N\}$ este mulțimea de termeni din vocabular (inclusiv și termenii din d)
- v_i este greutatea termenului t în d. De obicei, $v_i = tf\text{-}idf_{t,d}$

Observații preliminare

- O clasă de documente ocupă o anumită zonă de puncte în spațiul vectorial.
- Tehnicile de clasificare în spațiu vectorial sunt aplicabile dacă are loc ipoteza de contiguitate: **documentele din aceeași clasă formează o regiune contiguă, iar regiunile de clase diferite nu se suprapun.**

Documente din clasele China ($\circ$), Kenya ($\times$), UK ($\diamond$)



- **Se dau**
  - o mulțime finită de clase $\mathbb{C}$ și o mulțime finită de antrenare $\mathbb{D} = \{\langle d_i, c_i \rangle \mid 1 \leq i \leq n\}$ unde d_i sunt documente și $c_1, \ldots, c_n \in \mathbb{C}$
  - un document test d
- **Să se decidă** clasa de documente c la care aparține d.

De acum încolo vom considera că

- $\mathcal{D} := \{d \mid \langle d, c \rangle \in \mathbb{D}\}$ și
- $\mathcal{D}_c := \{d \mid \langle d, c \rangle \in \mathbb{D}\}$.

Remarcă: $\mathcal{D} = \bigcup_{c \in \mathbb{C}} \mathcal{D}_c$

Clasificarea Rocchio

1. Pentru fiecare clasă de documente $\mathcal{D}_c$ se calculează centrul de masă

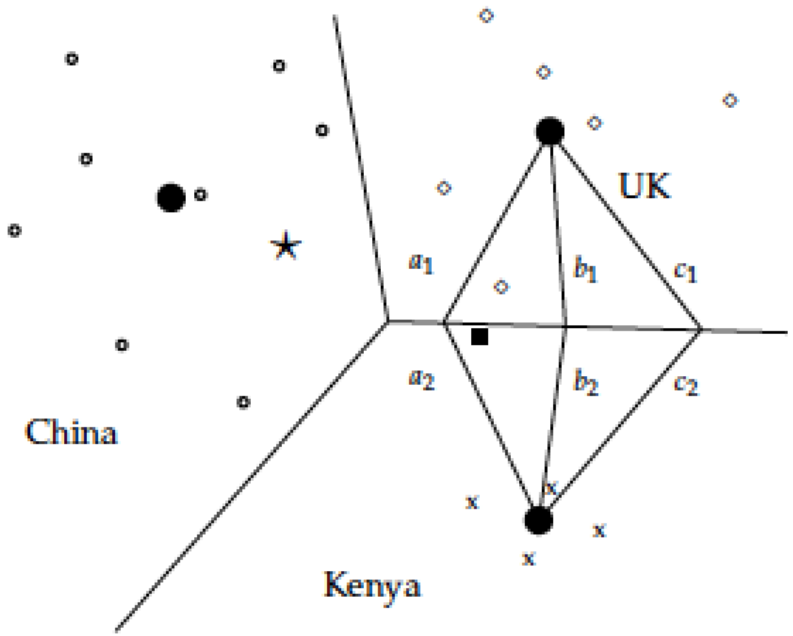
$$\vec{\mu}_c = \frac{1}{|\mathcal{D}_c|} \sum_{d \in \mathcal{D}_c} \vec{v}(d)$$

2. Granița dintre două clase c și c' în spațiul vectorial este mulțimea de puncte la distanță egală față de $\vec{\mu}_c$ și $\vec{\mu}_{c'}$
   - În general, granița dintre clasele c și c' este un hiperplan definit de mulțimea de puncte $\vec{x}$ pentru care are loc ecuația $\vec{w} \cdot \vec{x} = b$, unde
     - $\vec{w} \in \mathbb{R}^N$ este un vector N-dimensional perpendicular pe hiperplanul despărțitor:

       $\vec{w} = \vec{\mu}_c - \vec{\mu}_{c'}$
     - $b \in \mathbb{R}$ este o constantă:

       $b = \vec{v} \cdot \vec{w} = \frac{|\vec{\mu}_c|^2 - |\vec{\mu}_{c'}|^2}{2}$ unde $\vec{v} = \frac{1}{2}(\vec{\mu}_c + \vec{\mu}_{c'})$

Clasificarea Rocchio
Exemplu ilustrat



InvataRocchio($\mathbb{C}, \mathbb{D}$)
1 **for each** $c \in \mathbb{C}$ **do**
2 $\quad \vec{\mu}_c = \frac{1}{|\mathcal{D}_c| \sum_{d \in \mathcal{D}_c} \vec{v}(d)}$
3 **return** $\{\vec{\mu}_{c_1}, \ldots, \vec{\mu}_{c_M}\}$ unde $\mathbb{C} = \{c_1, \ldots, c_M\}$

AplicaRocchio($\{\vec{\mu}_1, \ldots, \vec{\mu}_M\}, d$)
1 $\quad$ **return** *arg* $\min_{c_j} |\vec{\mu}_{c_j} - \vec{v}(d)|$

Complexitatea clasificării Rocchio:

mod	complexitate în timp
învățare	$\Theta(\|\mathbb{D}\| L_{\text{medie}} + \|\mathbb{C}\| \cdot \|V\|)$
testare	$\Theta(L_a + \|\mathbb{C}\| \cdot M_a) = \Theta(\|\mathbb{C}\| \cdot M_a)$

unde V este vocabularul de termeni, L_{medie} este lungimea medie a unui document (ca secvență de termeni), L_a este lungimea documentului test, și M_a este nr. de termeni distincți ai documentului test.

Algoritmul lui Rocchio

Exemplu ilustrat de învățare și testare

	docID	cuvinte în document	în $c = China$?
mț. de antrenare	1	Chinese Beijing Chinese	da
	2	Chinese Chinese Shanghai	da
	3	Chinese Macao	da
	4	Tokyo Japan Chinese	nu
document test	5	Chinese Chinese Chinese Tokyo Japan	?

Vectorii și centroizii pentru aceste documente sunt:

	greutăți de termeni					
vector	Chinese	Japan	Tokyo	Macao	Beijing	Shanghai
$\vec{d}_1$	0	0	0	0	1.0	0
$\vec{d}_2$	0	0	0	0	0	1.0
$\vec{d}_3$	0	0	0	1.0	0	0
$\vec{d}_4$	0	0.71	0.71	0	0	0
$\vec{d}_5$	0	0.71	0.71	0	0	0
$\vec{\mu}_c$	0	0	0	0.33	0.33	0.33
$\vec{\mu}_{\overline{c}}$	0	0.71	0.71	0	0	0

În acest exemplu, hiperplanul de separare dintre clasele $c = China$ și $\overline{c} = \overline{China}$ este $\vec{w} \cdot \vec{x} = b$ unde

$$\begin{aligned} \vec{w} &\approx (0, -.71, -.71, 1/3, 1/3, 1/3)^T \\ b &= -1/3 \end{aligned}$$

Funcționează bine pentru clase contigue care sunt sfere de raze aproximativ egale.

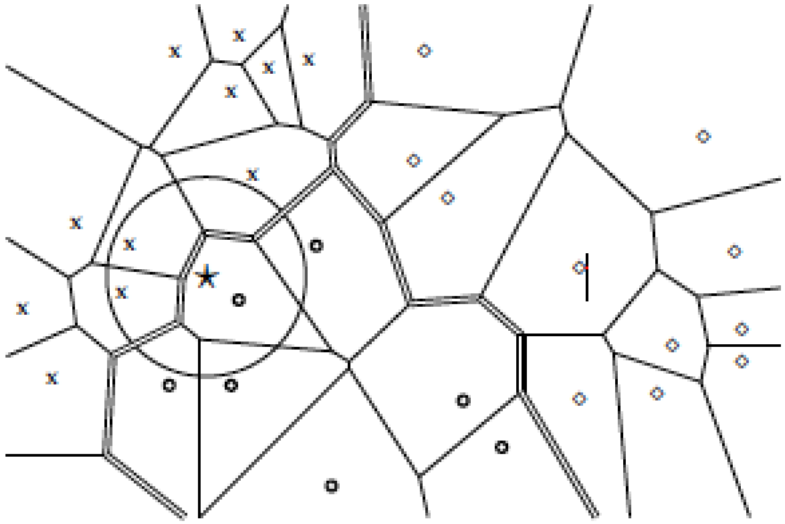
- Când sunt doar două 2 clase, de ex. *China* și complementul ei, clasa $\overline{China}$.
  China ocupă o regiune relativ mică
  $\Rightarrow$ ipoteza că clasele au raze egale trebuie revizuită, de exemplu:
  $d \in c$ dacă și numai dacă $|\vec{\mu}_c - \vec{v}(d)| < |\vec{\mu}(\overline{c}) - \vec{\mu}(d)| - b$ pentru un $d > 0$.

Metoda celor mai apropiați k vecini (kNN)

Un document nou d se atribuie clasei c_j dacă, dintre cei mai apropiați k vecini ai lui $\vec{v}(d)$ în spațiul vectorial, majoritatea fac parte din clasa c_j.

- În general, regiunile claselor sunt poligoane convexe în spațiul vectorial
- Pentru $k = 1$, regiunile formează un mozaic Voronoi, format din regiuni numite celule Voronoi

EXEMPLU: Regiuni determinate de 3 tipuri de documente: $\times$, $\circ$, $\diamond$



- Diagrama Voronoi este formată din celule Voronoi: fiecare celulă conține un singur punct $\vec{v}(d)$
- Granițele de decizie dintre cele 3 regiuni sunt liniile duble

- 1NN nu este robustă: clasificarea fiecărui document test depinde de un singur document de antrenare (cel mai apropiat vecin), care ar putea fi etichetat incorect.
- Clasificarea devine mai robustă pentru $k > 1$. Valori frecvent folosite pt k sunt 3, 5, sau între 50 și 100.
  - Alegerea unei valori bune pentru k se face adesea în etapa de învățare.
- Există și o versiune probabilistă a metodei de clasificare kNN: probab. apartenenței la clasa c este proporția de vecini din clasa c dintre cei mai apropiați k vecini. (Vezi slide-ul următor)

Metoda kNN–varianta probabilistă

Învățare (antrenare) și testare

InvataKNN($\mathbb{C}, \mathbb{D}$)
1 $\mathbb{D}' := \text{Preproceseaza}(\mathbb{D})$
2 $k :=$ Selecteaza-K($\mathbb{C}, \mathbb{D}'$)
3 **return** $\mathcal{D}', k$

AplicaKNN($\mathbb{C}, \mathbb{D}', k, d$)
1 $S_k := \text{CalculeazaCeiMaiApropiatiVecini}(\mathbb{D}', k, d)$
2 **for** fiecare $c_j \in \mathbb{C}$ **do**
3 $\quad p_j := |S_k \cap c_j|/k$
4 **return** $argmax_j\ p_j$

InvataKNN($\mathbb{C}, \mathbb{D}$)
1 $\mathbb{D}' := \text{Preproceseaza}(\mathbb{D})$
2 $k :=$ Selecteaza-K($\mathbb{C}, \mathbb{D}'$)
3 **return** $\mathcal{D}', k$

AplicaKNN($\mathbb{C}, \mathbb{D}', k, d$)
1 $S_k := \text{CalculeazaCeiMaiApropiatiVecini}(\mathbb{D}', k, d)$
2 **for** fiecare $c_j \in \mathbb{C}$ **do**
3 $\quad p_j := |S_k \cap c_j|/k$
4 **return** $argmax_j\ p_j$

OBSERVAŢIE

Timpul de testare al metodei kNN este $\Theta(|\mathbb{D}| \cdot M_{mediu} \cdot M_a)$ unde M_{mediu} este nr. mediu de termeni într-un document din colecție.

- avantaj: depinde liniar de $|\mathbb{D}|$, numărul de elemente din mulțimea de antrenare.

Un clasificator în două clase este liniar dacă apartenența unui document d la o clasă se decide comparând o combinație liniară componentelor (sau trăsăturilor) lui $\vec{v}(d)$ cu o valoare-prag (engl. *threshold*). În caz contar, clasificatorul este neliniar.

- În general, un clasificator liniar al unui vector $\vec{x} = (x_1, \ldots, x_M)$ în două clase operează în felul următor:

  AplicaClasificatorLiniar($\vec{w}, b, \vec{x}$)
  1 $scor := \vec{w} \cdot \vec{x} = \sum_{i=1}^{M} w_i \cdot x_i$
  2 **if** $scor > b$ **then return** 1
  3 **else return** 0

- Hiperplanul $\vec{w} \cdot \vec{x} = b$ se numește **hiperplan de decizie**.

Vom vedea că

- Metodele Bayes naiv și Rocchio sunt clasificatori liniari.
- Metoda kNN este clasificator neliniar.

Fie $\vec{\mu}_{c_1} = (a_1, \ldots, a_M)$ și $\vec{\mu}_{c_2} = (b_1, \ldots, b_M)$ centroizii celor două clase. Clasificatorul Rocchio pentru $\vec{x} = (x_1, \ldots, x_M)$ returnează 1 dacă $dist(\vec{\mu}_{c_1}, \vec{x}) > dist(\vec{\mu}_{c_1}, \vec{x})$, și 0 în caz contrar.

- $dist(\vec{\mu}_{c_1}, \vec{x}) > dist(\vec{\mu}_{c_2}, \vec{x})$ dacă și numai dacă $dist(\vec{\mu}_{c_1}, \vec{x})^2 > dist(\vec{\mu}_{c_2}, \vec{x})^2$, adică
  $0 < \sum_{i=1}^{M}(a_i - x_i)^2 - \sum_{i=1}^{M}(b_i - x_i)^2 =$
  $2\sum_{i=1}^{M}(b_i - a_i)x_i + \sum_{i=1}^{M}(a_i^2 - b_i^2)$

$\Rightarrow$ Clasificatorul Rocchio pentru $\vec{x} = (x_1, x_2, \ldots, x_M)$ returnează 1 dacă și numai dacă $\vec{w} \cdot \vec{x} > b$ unde

- $\vec{w} = \vec{\mu}_{c_2} - \vec{\mu}_{c_1} = (b_1 - a_1, b_2 - a_2, \ldots, b_M - a_M)$
- $b = \frac{1}{2}\sum_{i=1}^{M}(b_i^2 - a_i^2) = (|\vec{\mu}_{c_2}|^2 - |\vec{\mu}_{c_1}|^2)/2$

Reamintim faptul că metoda Bayes naivă pentru două clase complementare c și $\overline{c}$ operează în felul următor:

- Învață din mulțimea $\mathbb{D}$ estimările $\hat{P}(c)$, $\hat{P}(\overline{c})$, și $\hat{P}(t \mid c)$, $\hat{P}(t \mid \overline{c})$ pentru toți termenii t ce apar în documentele din $\mathbb{D}$
- Dacă secvența de termeni a documentului test d este $[t_1, \ldots, t_{n_d}]$, calculează

$$\hat{P}(d \mid c) = \hat{P}(c) \cdot \sum_{k=1}^{n_d} \hat{P}(t_k \mid c) \quad \text{și} \quad \hat{P}(d \mid \overline{c}) = \hat{P}(\overline{c}) \cdot \sum_{k=1}^{n_d} \hat{P}(t_k \mid \overline{c})$$

și clasifică d în clasa c dacă și numai dacă $\hat{P}(d \mid c) > \hat{P}(d \mid \overline{c})$.

$\Rightarrow$ $d \in c$ dacă și numai dacă $\log \frac{A}{B} > 0$, adică

$$0 < \log \frac{\hat{P}(c)}{\hat{P}(\overline{c})} + \sum_{k=1}^{n_d} \log \frac{\hat{P}(t_k \mid c)}{\hat{P}(t_k \mid \overline{c})}$$

$d \in c$ dacă și numai dacă

$$0 < \log \frac{\hat{P}(c)}{\hat{P}(\overline{c})} + \sum_{k=1}^{n_d} \log \frac{\hat{P}(t_k \mid c)}{\hat{P}(t_k \mid \overline{c})}$$

$$= \log \frac{\hat{P}(c)}{\hat{P}(\overline{c})} + \sum_{i=1}^{M} \log \frac{\hat{P}(t_i \mid c)}{\hat{P}(t_i \mid \overline{c})} \cdot x_i$$

unde

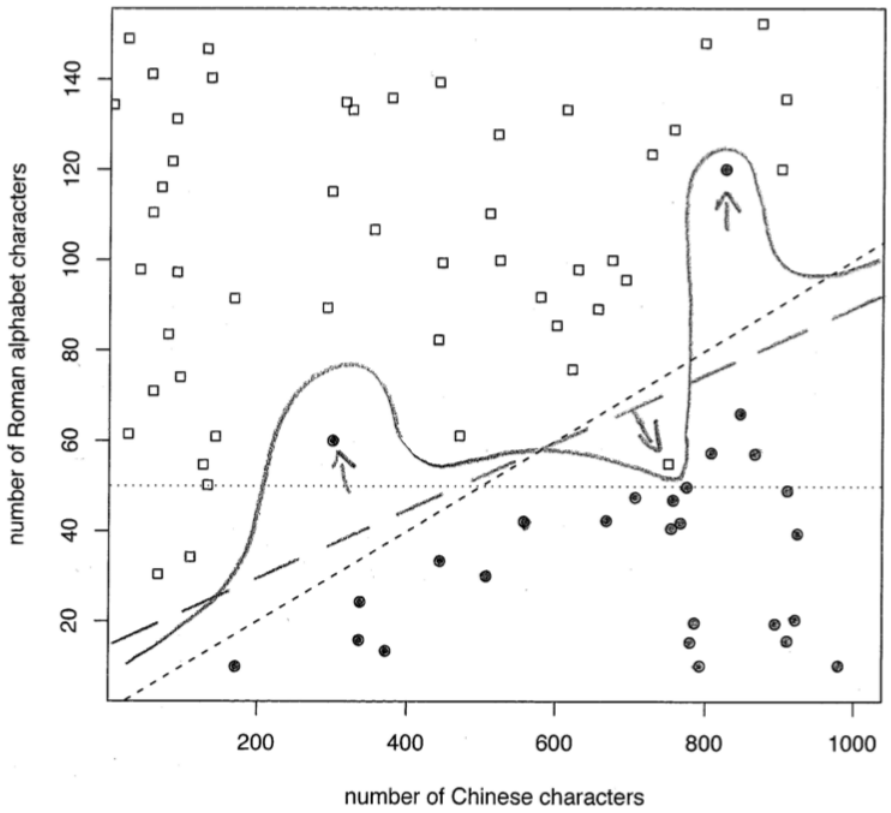
- $\{t_1, \ldots, t_M\}$ este vocabularul tuturor termenilor din documente din $\mathbb{D}$ și d
- x_i este numărul de apariții al lui t_i în d

$\Rightarrow$ Bayes naiv este clasificator liniar cu

$$b = -\log \frac{\hat{P}(c)}{\hat{P}(\overline{c})} \quad \text{și} \quad w_i = \log \frac{\hat{P}(t_i \mid c)}{\hat{P}(t_i \mid \overline{c})}$$

- Documentele d pentru care $P(d \mid c) = P(d \mid \overline{c})$ sunt reprezentate de puncte într-un hiperplan, numit **granița** clasei c cu $\overline{c}$.
  - Metodele de învățare liniară calculează hiperplanuri care sunt *aproximări* ale acestei granițe.
- Un document ,,zgomot" este un document d clasificat greșit în exemplele de învățare $\mathbb{D}$
  - În general, documentele ,,zgomot" afectează negativ rezultatul învățării $\Rightarrow$ șanse mai mari de clasificări eronate.

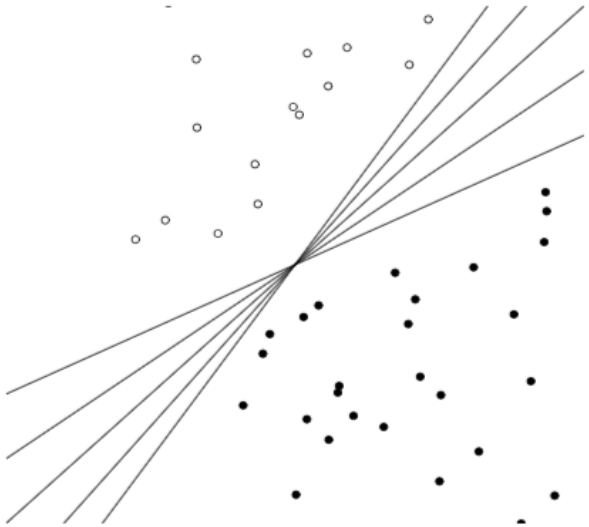
SCENARIU: Clasificarea paginilor web scrise în chineză sau nu. Paginile web scrise doar în chineză sunt reprezentate cu •, iar cele care au și caractere din alfabetul latin sunt marcate cu □. Granița dintre clase separă corect cele 2 tipuri de documente, cu excepția a 3 documente zgomot.



Separabilitate liniară

Două clase sunt linar separabile dacă există un hiperplan care le separă.

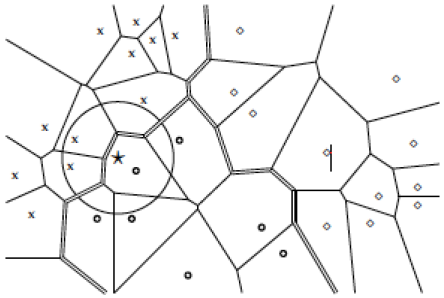
- În general, dacă două clase sunt liniar separabile, atunci există o infinitate de separatori liniari.



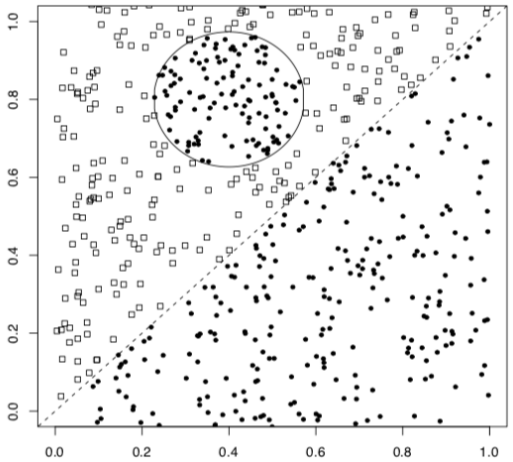
$\Rightarrow$ Problemă: cum putem defini un criteriu pentru a alege un separator liniar cât mai bun?

Clasificatori neliniari

kNN este clasificator neliniar: de exemplu, este evident că granița dintre clase deteminată de kNN în figura de mai jos nu ese o linie, ci o secvență de segmente liniare pe porțiuni scurte:



În anumite cazuri, clasificatorii liniari nu sunt adecvați, de ex.

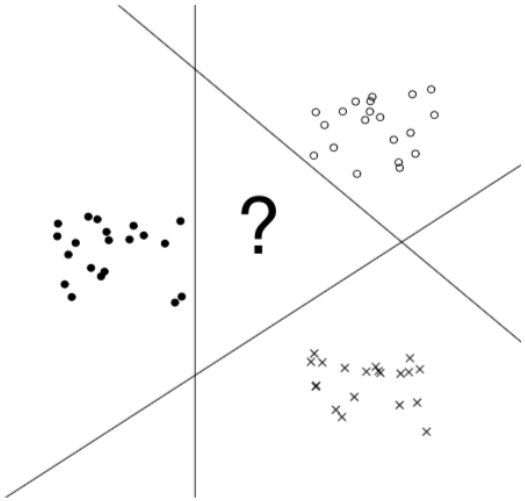


Dacă clasele nu sunt mutual exclusive (au documente în comun), vorbim despre **clasificare multi-etichetă** sau **clasificare multi-valoare**.

- Un clasificator liniar pentru $J > 2$ clase $c_1, \ldots, c_J$ care nu sunt mutual exclusive se poate construi astfel:
  - se construiește câte un clasificator liniar pentru fiecare cdin perechile de clase $c_k, \overline{c}_k$, $k = 1..J$
  - se aplica separat fiecare clasificator. Decizia unui clasificator este inependentă de deciziile celorlalți clasificatori.

Dacă clasele sunt mutual exclusive (nu au documente în comun), vorbim despre **clasificare multinomială** sau **clasificare multi-clasă**. Este necesar să definim o funcție de decizie γ de la mulțimea de documente la $\mathbb{C} = \{c_1, \ldots, c_J\}$.

Problemă: Cum putem combina clasificatori liniari de 2 clase pentru a obține un clasificator de J clase?



Putem proceda astfel:

1. Construim un clasificator liniar pentru clasele c_k și $\overline{c}_k$ pentru $k = 1..J \Rightarrow J$ separatori liniari.
2. Pentru un document test d, aplicăm toți cei J clasificatori
3. Alegem unul din criteriile următoare pentru a alege clasa c din care să facă parte documentul d
   - c să aibe scorul maxim (presupunem că fiecare clasificator calculează un scor de apartenență la clasa respectivă).
   - c să aibe valoarea maximă de confidență.
   - probabilitatea de apartenență la c sa fie cea mai mare.

Compromisul tendință-variație (1)

Metodele de învățare Γ prezentate până acum au urmărit să minimizeze erorile de clasificare a documentelor din mulțimea de teste T, și s-au bazat pe premisa că T și $\mathcal{D} := \{d \mid \langle d, c\rangle \in \mathbb{D}\}$ sunt mulțimi generate cu aceeeași distribuție $P(\langle d, c\rangle)$ unde d este un doc. și c este clasa lui d.

- Metodele Bayes naiv și Bernoulli sunt modele generative care caută

$$\text{argmax}_{c\in\mathbb{C}} P(c|d) = \text{argmax}_{c\in\mathbb{C}} \frac{P(d|c)\cdot P(c)}{P(d)}$$

  aplicând tehnici diferite de estimare a lui $P(d|c)$.

Altă metodă de evaluare a lui Γ este să minimizeze eroarea de calcul al lui $P(c|d)$ cu $\gamma(d)$. Mai precis, se dorește să se minimizeze *eroarea medie pătrată* (engl. mean squared error):

$$MSE(\gamma) := E_d[\gamma(d) - P(c \mid d)]^2 = \sum_d (\gamma(d) - P(c \mid d))^2 \cdot P(d)$$

unde E_d este valoarea medie în raport cu $P(d)$.

Compromisul tendință-variație (2)

Eroarea de învățare a lui γ cu Γ este

$$\begin{aligned}\text{eroare-invatare}(\Gamma) &= E_{\mathbb{D}}(MSE(\Gamma(\mathbb{D}))) \\ &= E_{\mathbb{D}}E_d(\Gamma(\mathbb{D})(d) - P(c|d))^2\end{aligned}$$

În general, $E[x-\alpha]^2 = (Ex-\alpha)^2 + E[x-Ex]^2$.
Pentru cazul special $x = \Gamma(\mathbb{D})(d)$ și $\alpha = P(c \mid d)$ obținem

$$\begin{aligned}E_{\mathbb{D}}E_d(\Gamma(\mathbb{D})(d) - P(c|d))^2 =& E_d E_{\mathbb{D}}(\Gamma(\mathbb{D})(d) - P(c|d))^2 \\ =& E_d[(E_{\mathbb{D}}\Gamma(\mathbb{D})(d) - P(c|d))^2 \\ &+ E_{\mathbb{D}}(\Gamma(\mathbb{D})(d) - E_{\mathbb{D}}\Gamma(\mathbb{D})(d))^2] \\ =& E_d[\text{tendința}(\Gamma, d) + \text{varianța}(\Gamma, d)]\end{aligned}$$

$$\begin{aligned}\text{unde} \quad \text{tendința}(\Gamma, d) &= (P(c|d) - E_{\mathbb{D}}\Gamma(\mathbb{D})(d))^2 \\ \text{varianța}(\Gamma, d) &= E_{\mathbb{D}}(\Gamma(\mathbb{D})(d) - E_{\mathbb{D}}\Gamma(\mathbb{D})(d))^2\end{aligned}$$

Compromisul tendință-variație (2)
Concluzii

- tendința este pătratul diferenței dintre $P(c|d)$ și predicția clasificatorului învățat pentru $P(c|d)$. Tendința este
  - **mare** când Γ produce clasificatori cu multe erori de clasificare
  - **mică** atunci când (1) Γ produce clasificatori cu puține erori de clasificare, sau (2) mț. diferite de antrenare produc erori pe documente diferite, sau (3) mț. diferite de antrenare produc erori pozitive și negative de clasificare pe unele documente, dar media erorii lor de clasificare tinde la 0.
- varianța este variația predicției clasificatorului învățat: media pătratelor diferențelor dintre $\Gamma(\mathbb{D})(d)$ și valoarea medie $E_{\mathbb{D}}\Gamma(\mathbb{D})(d)$. Varianța este
  - **mare** când mulțimi diferite de antrenare produc clasificatori f. diferiți.
  - **mică** dacă variații ale mulțimii de antrenare au efecte minore asupra dlasificatorilor calculați.

Compromisul tendință-variație (2)
Concluzii

- Metodele de învățare liniară au
  - variație mică pentru că mț. diferite de antrenare produc hiperplanuri de decizie similare.
- Metoda kNN are variație mare
- Eroarea de învățare=tendință+variație.
  În general, nu putem minimiza simultan și tendința și variația.

  $\Rightarrow$ când alegem o metodă de învățare, avem de ales dacă vrem să minimizăm tendința sau variația.

Bibliografie

1. *Vector space classification* (Cap. 14) din
   Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze: *An Introduction to Information Retrieval*.
   Ediție online (c) 2009 Cambridge UP.
   `http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf`