Învăţare automată

1. **(Exemple de funcţii care măsoară similaritatea dintre două imagini oarecare; demonstrarea unor proprietăţi)**

În învăţarea automată, *funcţiile de similaritate* joacă un rol foarte important atât în clasificarea automată (vedeţi de exemplu algoritmul k-NN) cât şi în clusterizare.

Considerăm X o mulţime formată din imagini dreptunghiulare de dimensiuni arbitrare, în care fiecare pixel este reprezentat sub forma unui număr întreg din mulţimea $\{0, \ldots, 255\}$. Fie $k_1 : X \times X \rightarrow \mathbb{R}$ o *funcţie de similaritate* a imaginilor, definită astfel: $k_1(x, x') =$ numărul de zone pătratice (engl., pixel patches), de dimensiune 16×16 pixeli, care apar atât în imaginea x cât şi în imaginea x'.[1]

a. Demonstraţi că funcţia k_1 are proprietatea următoare: există $d \in \mathbb{N}^*$ şi o funcţie („mapare“) $\phi : X \rightarrow \mathbb{R}^d$ astfel încât $k_1(x, x') = \phi(x) \cdot \phi(x')$ pentru orice x şi $x' \in X$.[2] **(Am notat cu $\cdot$ produsul scalar al vectorilor.)**

b. Fie acum o altă *funcţie de similaritate*:

$$k_2(x, x') = \begin{cases} 1 & \text{dacă } k_1(x, x') \geq 1, \quad \text{adică, există cel puţin un “patch”} \\ & \qquad\qquad\qquad\quad\ \text{(zonă pătratică) comun(ă) pentru } x \text{ şi } x' \\ 0 & \text{în caz contrar.} \end{cases}$$

Demonstraţi că funcţia k_2 are proprietatea următoare: există o mulţime de imagini $x_1, \ldots, x_n$ şi un vector-coloană $f \in \mathbb{R}^n$ astfel încât $f^\top G f < 0$, unde $\top$ reprezintă operaţia de transpunere, iar G este matricea de tip $n \times n$ având elementele $G(i, j) \stackrel{def.}{=} k_2(x_i, x_j)$, pentru $i, j = 1, \ldots, n$.[3]

***Sugestie*: Puteţi face demonstraţia în manieră constructivă, adică alegând în mod convenabil numărul n şi „construind“ / definind imaginile $x_1, \ldots, x_n$ astfel încât să existe $f \in \mathbb{R}^n$ cu propietatea $f^\top G f < 0$.**

[1] *Precizare*: Nu se va lua în considerare poziţia în care apare o astfel de *zonă pătratică*, ci doar *imaginea* / coloritul ei ca atare. O astfel de *zonă* poate să apară în mai multe poziţii într-o aceeaşi imagine, însă funcţia $k_1(x, x')$ va „contoriza“ o dată o anumită *zonă pătratică* dacă ea apare [nu are importanţă de câte ori anume] atât în x cât şi în x'.

[2] Altfel spus, funcţia k_1 este *funcţie-nucleu.*

[3] Altfel spus, funcţia k_2 *nu* este pozitiv semidefinită, deci *nu* este funcţie-nucleu (conform teoremei lui Mercer).

2. **(Funcţii de cost: o proprietate foarte utilă pentru mai mulţi algoritmi de învăţare automată care produc separatori liniari)**

Fie o funcţie $L : \mathbb{R}^2 \to \mathbb{R}$. Vom spune că L este ***funcţie de cost*** sau ***funcţie de pierdere*** (engl., loss function) dacă ea este $i.$ nenegativă şi $ii.$ convexă în raport cu primul argument, pentru orice valoare fixată a celui de-al doilea argument.

Spre *exemplu*,

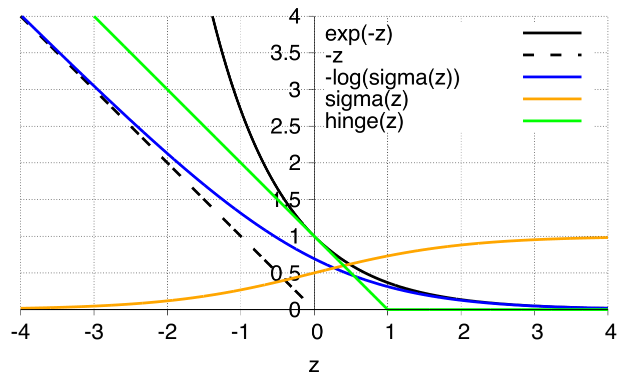
– pentru regresia liniară şi pentru perceptronul liniar se foloseşte – cu rol de funcţie de cost – pătratul erorii, adică $L(r, y) = (y - r)^2$,

– pentru regresia logistică se foloseşte ***funcţia de cost logistică*** $L(r, y) = \ln(1 + e^{-yr})$,

– pentru maşini cu vectori-suport (SVM) se foloseşte ***funcţia de cost hinge*** $L(r, y) = \max\{0, 1 - yr\}$, iar

– pentru AdaBoost, se foloseşte ***funcţia de cost*** [*negativ*] ***exponenţială*** $L(r, y) = e^{-yr}$.

(În figura alăturată, am folosit notaţia $z = yr$.)



Fie de asemenea numerele d şi $n \in \mathbb{N}^*$ şi elementele $x_1, \ldots, x_n \in \mathbb{R}^d$ (numite în învăţarea automată ***instanţe neetichetate***), precum şi $y_1, \ldots, y_n \in \mathbb{R}$ (desemnând fie clase, fie — ca în cazul regresiei liniare şi al perceptronului liniar — valori reale oarecare pentru funcţia de învăţat).

Este de remarcat faptul următor (foarte util din punct de vedere teoretic, dar şi din punct de vedere practic): unii algoritmi clasici de învăţare automată — precum cei menţionaţi mai sus — pot fi văzuţi ca rezolvând [câte] o ***problemă de optimizare*** (cu termen de regularizare de normă L_2) de forma următoare:

$$\min_{w \in \mathbb{R}^d} \left(\frac{1}{n} \sum_{i=1}^{n} L(w \cdot x_i, y_i) + \frac{\lambda}{2} \|w\|^2 \right), \tag{1}$$

unde $\lambda \geq 0$, operatorul $\cdot$ desemnează produsul scalar al vectorilor, operatorul $\| \ \|$ desemnează norma euclidiană ($\|w\| \stackrel{def.}{=} \sqrt{w^2}$), iar L este o anumită funcţie de cost.

Demonstraţi că orice soluţie w^* a ***problemei de optimizare*** (1) poate fi scrisă sub forma următoare:

$$w^* = \sum_{i=1}^{n} \alpha_i x_i \text{ pentru anumite valori } \alpha_i \in \mathbb{R}. \tag{2}$$

Observaţie: Conform relaţiei (2), vectorul-soluţie $w*$ pentru problema de optimizare (1) se „reprezintă“ ca o combinaţie liniară de instanţele $x_1, \ldots, x_n$.

Atenţie: Veţi rezolva problema în *cazul particular* când $\lambda > 0$, iar funcţia de cost $L : \mathbb{R}^2 \to \mathbb{R}$ este derivabilă în raport cu primul argument, pentru orice valoare fixată a celui de-al doilea argument. Puteţi, eventual, să folosiţi următoarele formule de calcul cu ***derivate vectoriale*** care generalizează bine-cunoscutele reguli de derivare pentru funcţii de o variabilă reală: $\frac{\partial}{\partial w}(a \cdot w) = a$ şi $\frac{\partial}{\partial w} w^2 = 2w$, pentru orice a şi w din $\mathbb{R}^d$.

ÎNVĂȚARE AUTOMATĂ

① X – mulțime imagini dreptunghiulare

$k_1(x, x')$ = nr. zone pătratice (16x16 pixeli) care apar în x și x'

a) $\phi : X \to R^d$ ← vector

Presupunem că avem un set de zone pătratice:

$P = \{p_1, p_2, \dots, p_d\}$, unde p_i = zonă pătratică 16x16

$\phi(x) = (\phi_1(x), \phi_2(x), \dots, \phi_d(x))$, unde:

$$\phi_1(x) = \begin{cases} 1, & \text{dacă } p_i \in x \\ 0, & \text{dacă } p_i \notin x \end{cases}$$

Observăm că: $\phi_i(x) \cdot \phi_i(x') = 1 \iff$ zona $p_i \in x$ și $p_i \in x'$

$$\Rightarrow \boxed{k_1(x, x') = \sum_{i=1}^{d} \phi_i(x) \cdot \phi_i(x') = \phi(x) \cdot \phi(x')}$$

b) Alegem $n = 3$

$\Rightarrow \rho^T = (a_1 \ \ a_2 \ \ a_3)$

$$\Rightarrow G = \begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix}$$

$$\Rightarrow \rho^T G \rho = (a_1 \ a_2 \ a_3) \begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix} (a_1 \ a_2 \ a_3)$$

$$= \begin{pmatrix} a_1 + a_2 x + a_3 y & a_1 x + a_2 + a_3 z & a_1 y + a_2 z + a_3 \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{pmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$$= 3\left((a_1 + a_2 x + a_3 y)a_1 + (a_1 x + a_2 + a_3 z)a_2 + (a_1 y + a_2 z + a_3)a_3\right)$$

$$= 3\left(a_1^2 + a_1 a_2 x + a_1 a_3 y + a_1 a_2 x + a_2^2 + a_2 a_3 z + a_1 a_3 y + a_2 a_3 z + a_3^2\right)$$

$$= 3\left(a_1^2 + a_2^2 + a_3^2 + 2a_1 a_2 x + 2a_2 a_3 z + 2a_1 a_3 y\right)$$

Alegem $f^T = (1 \; 1 \; -1)$

$$\Rightarrow f^T G f = 3(1 + 1 + 1 + 2x - 2z - 2y)$$

$$\Rightarrow f^T G f = 9 + 6x - 6y - 6z$$

Alegem $(x, y, z) = (0, 1, 1)$ $\Rightarrow f^T G f = 9 + 0 - 6 - 6 = -3 < 0$

$$\boxed{\Rightarrow f^T G f < 0}$$

② Demonstrați că $(\forall) w^*$ poate fi scrisă ca $w^* = \sum_{i=1}^{n} \alpha_i x_i$

Dem:

$$\min_{w \in \mathbb{R}^d} \left\{ \frac{1}{m} \sum_{i=1}^{m} L(w \cdot x_i, y_i) + \frac{\lambda}{2} \|w\|^2 \right\}, \quad \lambda > 0, \quad L\text{- loss func.}$$

$$f(w) = \frac{1}{m} \sum_{i=1}^{m} L(w \cdot x_i, y_i) + \frac{\lambda}{2} \|w\|^2$$

$$\boxed{\text{Pt } f(w): \nabla f = \left(\frac{\partial f}{\partial w} \right)}$$

$$\nabla f(w) = \frac{1}{m} \sum_{i=1}^{m} \nabla L(w \cdot x_i, y_i) \cdot x_i + \lambda w$$

$$\nabla_w L(w \cdot x_i, y_i) = \frac{\partial L(w \cdot x_i, y_i)}{\partial (w \cdot x_i)} \cdot \nabla_w (w \cdot x_i)$$

Știim că $\nabla_w (w \cdot x_i) = x_i$

$$\Rightarrow \nabla_w L(w \cdot x_i, y_i) = \frac{\partial L(w \cdot x_i, y_i)}{\partial (w \cdot x_i)} x_i$$

Notăm cu $g_i = \dfrac{\partial L(w \cdot x_i, y_i)}{\partial (w \cdot x_i)}$

$$\Rightarrow \nabla_w L(w \cdot x_i, y_i) = g_i x_i$$

$$\Rightarrow \nabla f(w) = \frac{1}{m} \sum_{i=1}^{m} g_i x_i + \lambda w$$

La punctul de minim w^*, derivata tb să fie 0:

$$\nabla f(w^*) = 0$$

$$\Leftrightarrow \frac{1}{n}\sum_{i=1}^{n} g_i x_i + \lambda w^* = 0$$

$$\Leftrightarrow \lambda w^* = -\frac{1}{n}\sum_{i=1}^{n} g_i x_i$$

$$\Leftrightarrow w^* = -\frac{1}{\lambda n}\sum_{i=1}^{n} g_i x_i$$

Deci că putem scrie $\boxed{w^* = \sum_{i=1}^{n} \alpha_i x_i}$, unde $\alpha_i = -\frac{g_i}{\lambda n}$