Tema nr. 8

Fie $F : \mathbb{R}^n \longrightarrow \mathbb{R}$ o funcţie reală, $F(x) = F(x_1, x_2, \ldots, x_n)$. Să se aproximeze un punct de minim (local sau global) al funcţiei F folosind metoda gradientului descendent. Să se testeze diversele metode de calcul a ratei de învăţare descrise în acest text. Să se calculeze gradientul funcţiei F folosind formula analitică şi formula aproximativă. Să se compare soluţiile obţinute folosind cele două moduri de calcul a gradientului funcţiei F, din punctul de vedere al numărului de iteraţii efectuate pentru obţinerea soluţiilor (pentru aceeaşi precizie $\epsilon > 0$). Testaţi toate funcţiile listate la sfârşitul acestui text.

Bonus: Aplicaţi algoritmii de minimizare pentru rezolvarea problemei de regresia logistică descrisă in fişierul RL (pag. 65).

Obţineţi **10pt** daca folosiţi funcţia de log-verosimilitate şi gradientul ei deja calculate.

Obţineţi **15pt** dacă construiţi funcţia de log-verosimilitate şi gradientul ei pornind de la tabelul cu date.

$$\max\{F(x); x \in V\} \quad = \quad -\min\{-F(x); x \in V\} \quad ,$$

$$\operatorname{argmax}\{F(x); x \in V\} \quad = \quad \operatorname{argmin}\{-F(x); x \in V\}$$

Minimizarea funcţiilor

Fie $F : \mathbb{R}^n \longrightarrow \mathbb{R}$ o funcţie reală de două ori derivabilă , $F \in C^2(\mathbb{R}^n)$, pentru care vrem să aproximăm soluţia x^* a problemei de minimizare:

$$\min\{F(x); x \in V\} \quad \longleftrightarrow \quad F(x^*) \leq F(x) \quad \forall x \in V \tag{1}$$

unde $V = \mathbb{R}^n$ (x^* este punct de minim global) sau $V = S(\bar{x}, r)$, sfera de centru $\bar{x}$ şi rază r (punct de minim local). Se numeşte *punct critic* pentru funcţia F, un punct $\tilde{x}$ care este rădăcină a sistemului de ecuaţii:

$$\nabla F(\tilde{x}) = 0 \quad , \quad \nabla F(x) = \begin{pmatrix} \dfrac{\partial F}{\partial x_1}(x) \\ \vdots \\ \dfrac{\partial F}{\partial x_i}(x) \\ \vdots \\ \dfrac{\partial F}{\partial x_n}(x) \end{pmatrix} . \tag{2}$$

Se ştie că pentru funcţiile de două ori derivabile, punctele de minim ale funcţiei F se găsesc printre punctele critice. Un punct critic este punct de minim dacă matricea hessiană este pozitiv semidefinită:

$$H(\tilde{x}) = \left(\frac{\partial^2 F}{\partial x_i \partial x_j}(\tilde{x}) \right)_{i,j=1,\dots,n} \quad , \quad \big(H(\tilde{x})z, z\big)_{\mathbb{R}^n} \geq 0 \quad \forall z \in \mathbb{R}^n.$$

Metoda gradientului descendent

Punctul de minim al funcţiei F se aproximează construind un şir de vectori $\{x^k\} \subseteq \mathbb{R}^n$ care, în anumite condiţii, converge la punctul de minim x^* căutat. Convergenţa şirului depinde de alegerea primului element al şirului, x^0.

Elementul $k+1$ al şirului, x^{k+1}, se construieşte pornind de la elementul precedent, x^k, astfel:

$$x^{k+1} = x^k - \eta_k \nabla F(x^k) \ , \ k = 0, 1, \dots \ , \ x^0 - \text{ ales random} \tag{3}$$

Elementul η_k poartă numele de rată de învăţare sau pasul iteraţiei.

Strategii de alegere a ratei de învăţare

1. $\eta_k = \eta \ , \ \forall k \ (\eta = 10^{-3}, 10^{-4}, ...)$. O rată de învăţare constantă prea mare poate face ca punctul de minim să nu poată fi găsit, iar o valoare prea mică pentru rata de învăţare are dezavantajul unui cost de calcul mare.

2. Un mod de a rezolva problemele care apar în cazul ratei de învăţare constante este de a considera o valoare variabilă, în funcţie de contextul local. Metoda descrisă mai jos poartă denumirea de ajustare de tip *backtracking* a lungimii pasului/ratei de învăţare (*backtracking line search*). Această metodă funcţionează pentru funcţii convexe.

   Se considera $\beta \in (0,1)$ fixat (de obicei se alege $\beta = 0.8$). La fiecare pas rata de învăţare se calculează astfel:

   $\eta = 1;$
   $p = 1;$
   while $F(x^k - \eta \nabla F(x^k)) > F(x^k) - \frac{\eta}{2}\|\nabla F(x^k)\|^2$ && $p < 8$
   $\quad \eta = \eta\,\beta;$
   $\quad p++;$

Observaţie importantă: Alegerea vectorului iniţial, x^0, poate determina convergenţa sau divergenţa şirului $\{x^k\}$ la punctul de minim x^*. De obicei, o alegere a datelor iniţiale în vecinătatea lui x^* asigură convergenţa $x^k \longrightarrow x^*$ pentru $k \to \infty$.

Nu este necesară memorarea întregului şir $\{x^k\}$ ci avem nevoie doar de 'ultimul' element x^{k_0} calculat. Se consideră că un element x^{k_0} aproximează punctul de minim căutat, x^*, $x^{k_0} \approx x^*$ (x^{k_0} este ultimul element al şirului care se calculează) atunci când diferenţa dintre două elemente succesive ale şirului devine suficient de mică, i.e.,

$$\|x^{k_0} - x^{k_0-1}\| \leq \epsilon \tag{4}$$

unde ϵ este precizia cu care vrem să aproximăm soluţia x^*.

Prin urmare, o schemă posibilă de aproximare a soluţiei x^* este următoarea:

Schema de calcul

se aleg random elementele vectorului iniţial x ;
$k = 0$;
do
{
- calculează $\nabla F(x)$;
- calculează rata de învăţare η folosind una din cele 2 metode;
- $x = x - \eta \nabla F(x)$;
- $k = k + 1$;

}
while ($\eta \|\nabla F(x)\| \geq \epsilon$ şi $k \leq k_{\max}$ şi $\eta \|\nabla F(x)\| \leq 10^{10}$)
if ($\eta \|\nabla F(x)\| \leq \epsilon$) $x \approx x^*$;
else *"divergenţă"* ; //(de încercat schimbarea datelor iniţiale)

O valoare posibilă pentru $k_{\max}$ este 30000 şi $\epsilon > 10^{-5}$.

Pentru a calcula valoarea gradientului funcţiei F într-un punct oarecare se va folosi formula analitică de calcul a gradientului (funcţie declarată în program) şi de asemenea se va folosi următoarea formulă aproximativă:

$$\nabla F(x) \approx \big(G_i(x,h)\big)_{i=1,\ldots,n} \quad , \quad \frac{\partial F}{\partial x_i}(x) \approx G_i(x,h)$$

unde

$$\frac{\partial F}{\partial x_i}(x) \approx G_i(x,h) = \frac{-F_1 + 8F_2 - 8F_3 + F_4}{12h} \ , \ \forall i = 1,\ldots,n$$

cu $h = 10^{-5}$ sau 10^{-6} (poate fi considerat parametru de intrare) şi:

$$\begin{aligned} F_1 &= F(x_1,\ldots,x_{i-1},x_i+2h,x_{i+1},\ldots,x_n),\\ F_2 &= F(x_1,\ldots,x_{i-1},x_i+h,x_{i+1},\ldots,x_n),\\ F_3 &= F(x_1,\ldots,x_{i-1},x_i-h,x_{i+1},\ldots,x_n),\\ F_4 &= F(x_1,\ldots,x_{i-1},x_i-2h,x_{i+1},\ldots,x_n). \end{aligned}$$

Exemple

$l(w_0,w_1) = -\ln\left(1+\exp\left(w_0-w_1\right)\right) + w_0 + w_1 - \ln\left(1+\exp\left(w_0+w_1\right)\right)$

$\nabla l(w_0,w_1) = \begin{pmatrix} -\sigma(w_0-w_1)+\sigma(-w_0-w_1) \\ \sigma(w_0-w_1)+\sigma(-w_0-w_1) \end{pmatrix}, \quad \sigma(z) = \dfrac{1}{1+\exp(-z)}.$

$F(x_1,x_2) = x_1^2 + x_2^2 - 2x_1 - 4x_2 - 1, \nabla F(x_1,x_2) = \begin{pmatrix} 2x_1-2 \\ 2x_2-4 \end{pmatrix}, x_1^* = 1, \ x_2^* = 2.$

$F(x_1,x_2) = 3x_1^2 - 12x_1 + 2x_2^2 + 16x_2 - 10, \nabla F(x_1,x_2) = \begin{pmatrix} 6x_1-12 \\ 4x_2+16 \end{pmatrix}, x_1^* = 2, x_2^* = -4.$

$F(x_1,x_2) = x_1^2 - 4x_1x_2 + 5x_2^2 - 4x_2 + 3, \nabla F(x_1,x_2) = \begin{pmatrix} 2x_1-4x_2 \\ -4x_1+10x_2-4 \end{pmatrix}, x_1^* = 4, x_2^* = 2.$

$F(x_1,x_2) = x_1^2x_2 - 2x_1x_2^2 + 3x_1x_2 + 4, \nabla F(x_1,x_2) = \begin{pmatrix} 2x_1x_2 - 2x_2^2 + 3x_2 \\ x_1^2 - 4x_1x_2 + 3x_1 \end{pmatrix}, x_1^* = -1, x_2^* = 0.5.$