2. Regresia liniară

Reprezentarea modelului de regresie liniară
Funcţia de cost pătratică
Algoritmul de minimizare al gradientului

În acest curs vom crea un algoritm de predicţie pentru preţul unei case, dându-se ca şi caracteristică de intrare preţul terenului. Posibile exemple de perechi suprafaţă teren - preţ casă, ce compun baza de date de antrenare (training set), sunt redate în tabelul 2.1.

Suprafaţă teren [mp]	Preţ [1000 EUR]
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

Tab. 2.1 Exemple de perechi de antrenare suprafaţă teren - preţ casă.

Distribuţia preţurilor caselor, în funcţie de suprafaţa terenului, este redată în figura 2.1.

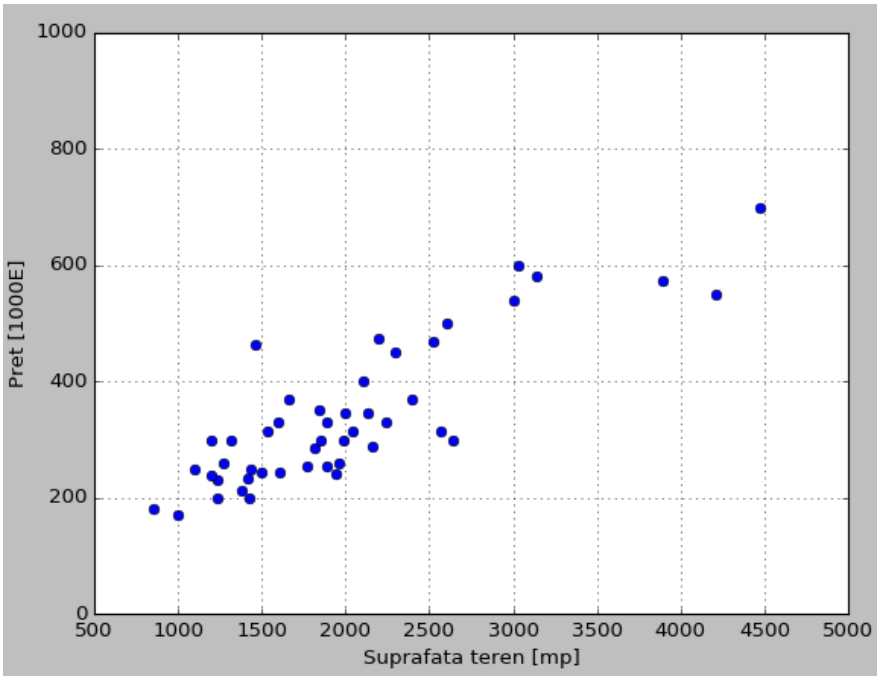

Fig. 2.1 Distribuţia perechilor suprafaţă teren - preţ casă din baza de date de antrenare.

2.1 Reprezentarea modelului

În cele ce urmează, vom utiliza următoarele notaţii:

- $x^{(i)}$: vectorul caracteristicilor de intrare (features); în cazul exemplului curent caracteristica de intrare este suprafaţa terenului;
- $y^{(i)}$: valoarea de ieşire - preţul casei;
- $(x^{(i)}, y^{(i)})$: exemplu de antrenare;
- $(x^{(i)}, y^{(i)}); i = 1, ..., m$: baza de date de antrenare (training set), compunsă din m exemple;
- i: index către elementele din training set;
- $\mathbb{X}$: spaţiul caracteristicilor de intrare;
- $\mathbb{Y}$: spaţiul caracteristicilor de ieşire;

În exemplul curent, atât spaţiul caracteristicilor de intrare, cât şi spaţiul caracteristicilor de ieşire aparţin mulţimii numerelor reale: $\mathbb{X} = \mathbb{Y} = \mathbb{R}$.

Obiectivul algoritmului de învăţare este de a învăţa funcţia $h_\theta(x) : \mathbb{X} \to \mathbb{Y}$, în aşa fel încât ipoteza $h_\theta(x)$ să reprezinte un predictor "optim" pentru valoarea corespondentă a lui y [4]. Coeficienţii θ, denumiţi *parametrii modelului*, sunt valorile ce trebuiesc învăţate prin algoritmul de învăţare.

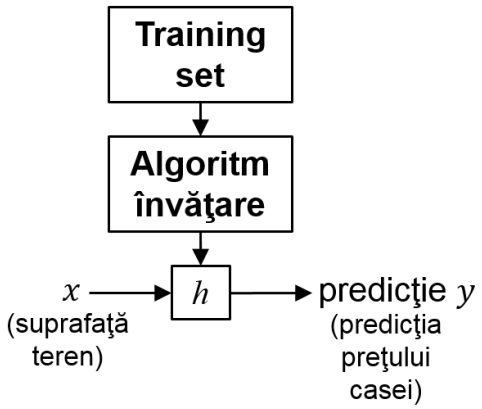

Fig. 2.2 Schema bloc al unui algoritm de învăţare.

Atunci când variabila pe care dorim să o prezicem este continuă ($y \in \mathbb{R}$), cum este cazul exemplului de faţă, problema de învăţare este denumită *regresie*. În cazul în care y poate lua doar câteva valori discrete ($y \in \{0, 1, ..., k\}$) (de exemplu, atunci când dorim să prezicem dacă proprietatea este o casă $y = 0$ sau un apartament $y = 1$), problema de învăţare se numeşte *clasificare*.

Pentru exemplul curent vom folosi o funcţie de aproximare liniară:

$$h_\theta(x) = \theta_0 + \theta_1 x, \tag{2.1}$$

unde x reprezintă suprafaţa terenului.

În cazul în care, pe lângă suprafaţa terenului, dispunem de caracteristici (features)

adiţionale ce pot ajuta la predicţia preţului (de exemplu, numărul de dormitoare), funcţia de aproximare poate fi rescrisă astfel:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2, \tag{2.2}$$

unde $x = \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix}$ este în acest caz un vector de caracteristici compus din x_1 (este suprafaţa terenului) şi x_2 (numărul de dormitoare), iar $x_0 = 1$.

Ecuaţia 2.2 a modelului poate fi generalizată sub forma:

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x, \tag{2.3}$$

unde n este numărul de caracteristici (features).

2.2 Funcţia de cost

Acurateţea funcţiei de aproximare $h_\theta(x)$ se măsoară cu ajutorul unei funcţii de cost $J(\theta)$. Aceasta ia în considerare diferenţa medie dintre perechile de caracteristici de intrare x - valoarea y a ieşirii. În cazul modelului 2.1 utilizat în acest curs, funcţia de cost are forma:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right)^2 = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2, \tag{2.4}$$

unde m este numărul de exemple de antrenare, iar $h_\theta(x_i) - y_i$ reprezintă diferenţa dintre valoarea prezisă $h_\theta(x_i)$ şi valoarea reală y_i. Datorită exponentului, funcţia poartă denumirea de *funcţie de eroare pătratică*, sau *funcţie de eroare medie*. Cu alte cuvinte, $J(\theta)$ exprimă cât de bine sunt aproximate exemplele de antrenare de către $h_\theta(x)$, aşa cum este ilustrat în figura 2.3.

Obiectivul algoritmului de învăţare este de a determina parametrii θ_0 şi θ_1, în aşa fel încât $h_\theta(x)$ să fie cât mai aproape de y pentru exemplele de antrenare (x, y). Acest lucru se întamplă atunci când funcţia de cost are valoarea minimă:

$$\min_{\theta_0,\theta_1} J(\theta_0, \theta_1) = \min_{\theta_0,\theta_1} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2. \tag{2.5}$$

2.3 Algoritmul de minimizare al gradientului

În acestă fază avem definit modelul ipotezei şi o modalitate de măsurare a gradului de potrivire a modelului pe datele de antrenare. În următorul pas vom estima parametrii modelului utilizând algoritmul de minimizare al gradientului.

În figura 2.4 este redată o posibilă formă a funcţiei de cost $J(\theta)$, în funcţie de parametrii θ_0 şi θ_1. Coeficienţii θ_0 şi θ_1 se află pe axele x şi y, iar valoarea lui $J(\theta)$ pe axa z. Fiecare poziţie 3D din forma funcţiei reprezintă valoarea lui $J(\theta)$ pentru o combinaţie de valori θ_0

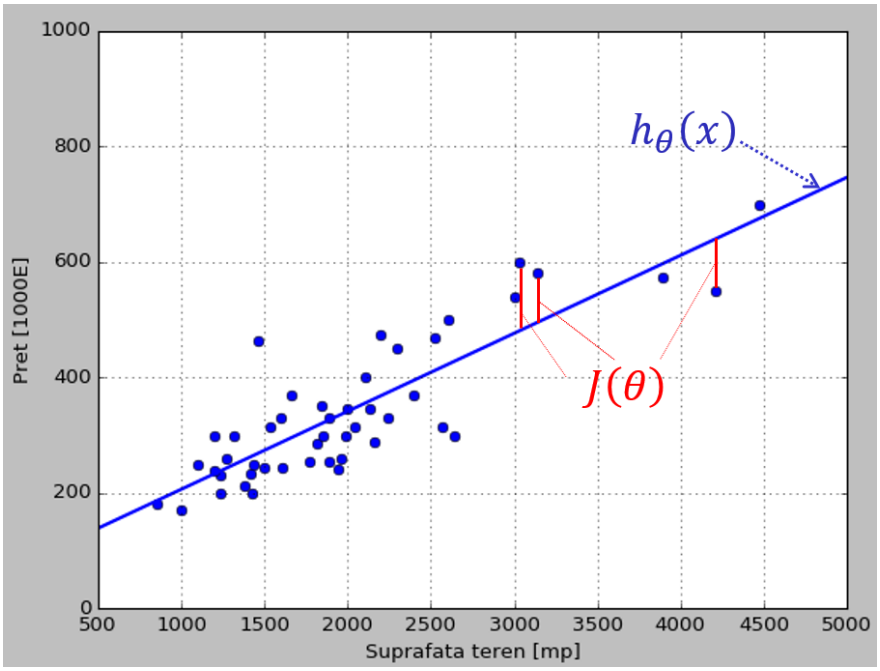

Fig. 2.3 Diferenţele dintre ipoteza $h_\theta(x)$ şi valoarea reală a lui y, exprimate prin intermediul funcţiei de cost $J(\theta)$.

şi θ_1. Două perechi de parametrii optimi se găsesc la cele două minime ale funcţiei, indicate de săgeţiile roşii.

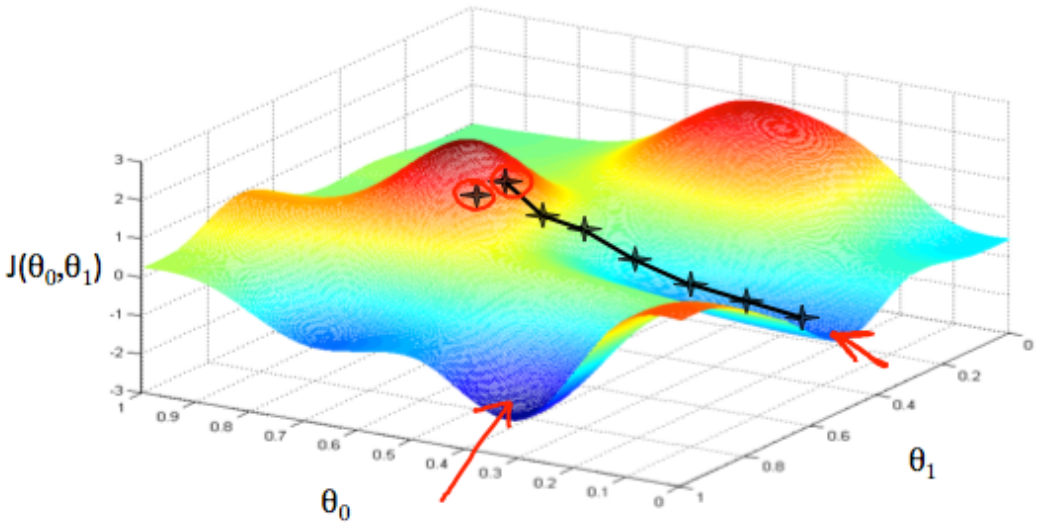

Fig. 2.4 Posibilă formă a funcţiei de cost $J(\theta)$.

Modalitatea de determinare a minimului lui $J(\theta)$ se bazează pe calculul derivatei funcţiei de cost (linia tangentă la valoarea funcţiei). Panta tangentei este derivata la punctul respectiv, ce ne va indica direcţia de deplasare către minimul funcţiei. Algoritmul va efectua paşi către valori inferioare ale funcţiei de cost prin coborâri cât mai abrupte. Mărimea fiecărui pas este determinată de parametrul α, denumit *rată de învăţare* (eng. learning rate).

Spre exemplu, distanţa dintre fiecare "stea" din graficul de mai sus reprezintă un pas

determinat de parametrul α. Un α de valoare mică va produce paşi de dimensiune mai mică. Analog, o valoare mare pentru α va produce paşi de dimensiuni mai mari. Direcţia în care va fi efectuat un pas este dată de derivata parţială a lui $J(\theta_0, \theta_1)$. În funcţie de poziţia din graf de unde se porneşte căutarea minimului, putem ajunge la diferite puncte de minim. În figura 2.4 sunt redate două asemenea puncte de minim.

Pseudocodul algoritmului de minimizare al gradientului are următoarea formă:

Repetă până la convergenţă: {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

}

unde $j = 0, 1$ reprezintă indexul caracteristicilor. După fiecare iteraţie a algoritmului, parametrii θ trebuiesc actualizaţi simultan.

Următorul pas în estimarea parametrilor θ este calculul derivatei lui $J(\theta_0, \theta_1)$, dată în relaţia următoare pentru un singur exemplu de antrenare:

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \left(h_\theta(x) - y\right)^2 && (2.6)\\
&= 2 \cdot \frac{1}{2} \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(h_\theta(x) - y\right) && (2.7)\\
&= \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{n} \theta_i x_i - y \right) && (2.8)\\
&= \left(h_\theta(x) - y\right) x_j. && (2.9)
\end{aligned}$$

Astfel, ecuaţia 2.6 devine:

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=0}^{m} \left(\left(h_\theta(x) - y\right) x_j \right). \tag{2.10}$$

Bibliografie

[1] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[2] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[3] M. Lutz, *Learning Python*, 2nd Ed. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2003.

[4] A. Ng, "Stanford cs229 - machine learning," 2008.

[5] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd Ed. Pearson Education, 2003.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc., 2017.