W03

Fekete Máté

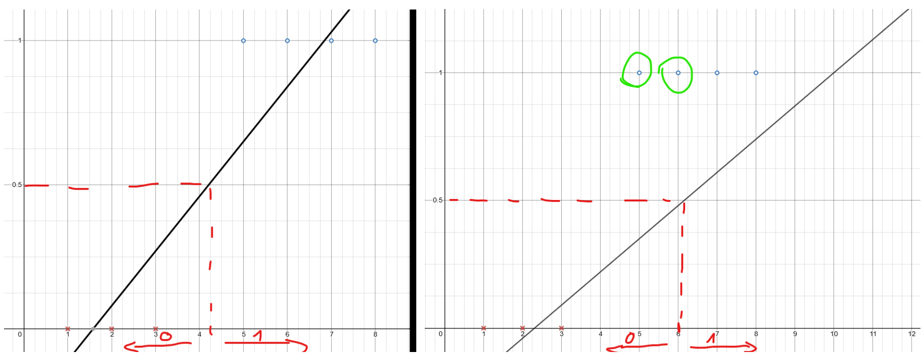
2020 Április

Tartalom jegyzék

1 Osztályozás

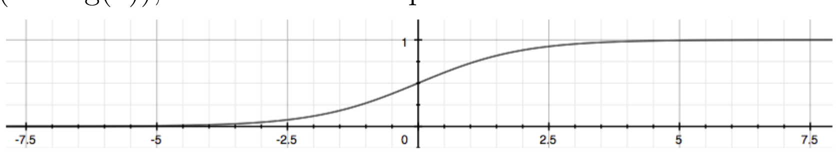
Egy eldöntendő kérdést próbálunk modellezni.
Pl.: $x^{(i)}$ valamilyen jellemzői egy email-nek, $y^{(i)}$ pedig pedig 1 ha spam, 0 ha nem az.
Megpróbálhatnánk lineráis regresszióval egy egyenest illeszteni az adatkörre, majd egy határértéket megadva azt feltételezni, hogy ami balra van 0 kimenet, jobbra pedig 1-et ad.

A probléma ezzel a megvalósítással az, hogy egy szélsőséges érték felboríthatja a hipotézis függvényünket, illetve h(x) nem csak a [0,1] intervallumon vehet fel értékeket.



A két ábra közötti különbség egyetlen plusz tanuló példa, viszont mivel így már más egyenest illeszkedik jobban az adatkörre, a zölddel bekaríkázott esetekben valószínűleg rosszul fog működni az algoritmus.

Ennek kiküszöbölésére a sigmoid vagy logisztikus függvényt használhatjuk (Jele: g(x)), amit az alábbi kép mutat be.



$h_\theta(x) = g(\theta^T x)$
$z = \theta^T x$
$g(z) = \frac{1}{1+e^{-z}}$

A függvény értékkészlete így már a [0,1] intervallumba esik, így sokkal jobban használható osztályozásra.

$h_\theta(x)$ annak az esélyét fogja visszadni, hogy az adott példának mekkora eséllyel 1 a kimenete $(P(y = 1|x; \theta))$.

y=1, ha h(x)≥0.5, azaz z≥0
y=0, ha h(x)<0.5, azaz z<0

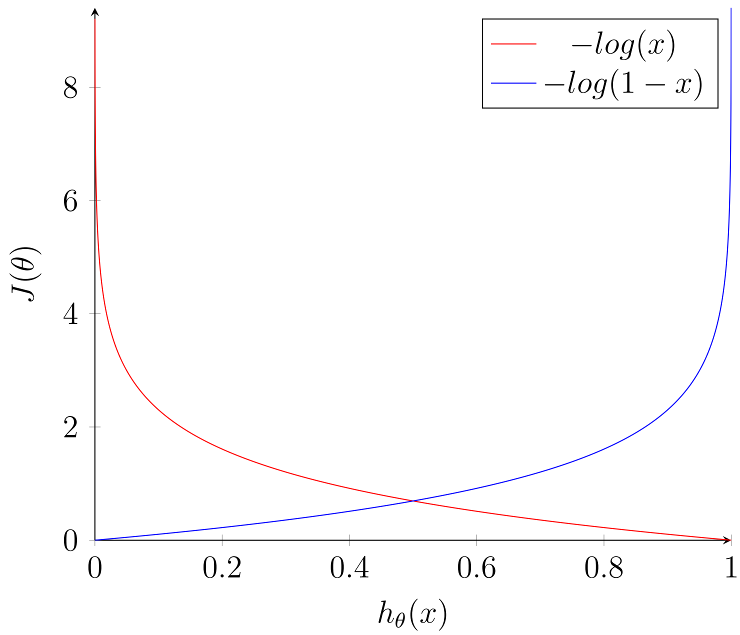
Döntési felület:
az a sík, ami elválasztja a pozitív(1) és negatív(0) példákat.

Ha nem egy egyenes a megfelelő elválasztó az y értékek között, adhatunk a g(z) sigmoid függvénynek nem lineáris paramétert is, pl. ami egy kört ír le.

2 Költség függvény

A lineáris regressziónál használt költség függvény túl sok lokális minimumot adna vissza, viszont mi inkább egy konvex függvényt szeretnénk.

$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h\theta(x^{(i)}), y^{(i)})$
$Cost(h_\theta(x), y) = -log(h_\theta(x))$ ha y = 1
$Cost(h_\theta(x), y) = -log(1 - h_\theta(x))$ ha y = 0



Ha y = 1, de h(x) ≈ 0, akkor mivel -log(x) a végtelenbe konvergál a 0 irányába, a költség függvényünk is a végtelenbe fog tartani, ezzel jelezve, hogy nagyon mellé lőtt az algoritmus. Ellenkező esetben pedig a 0-ba konvergál -log(x), azaz minél közelebb van y-hoz h(x), annál kisebb lesz a költség függvény értéke, ami nekünk pont a megfelelő működést jelenti.

2.1 Egyszerűsített költség függvény

$Cost(h_\theta(x), y) = -ylog(h_\theta(x)) - (1-y)log(1-h_\theta(x))$

Így ha y = 0, az első tag kiesik, ha y = 1 akkor pedig a második, a maradék pedig pont a fentebb említett egyenletek egyike lesz.

A teljes J költség függvényünk tehát a következő lesz:

$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}log(h_\theta(x^{(i)})) + (1-y^{(i)})log(1-h_\theta(x^{(i)}))]$

vagy vektorizálva:

$h = g(X\theta)$
$J(\theta) = \frac{1}{m} - y^T log(h) - (1-y)^T log(1-h)$

2.2 Gradiens módszer logisztikus reggeszióhoz

A képlet továbbra is a következő, csak a költség függvényünk változott meg

$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$

Vektorizálva:

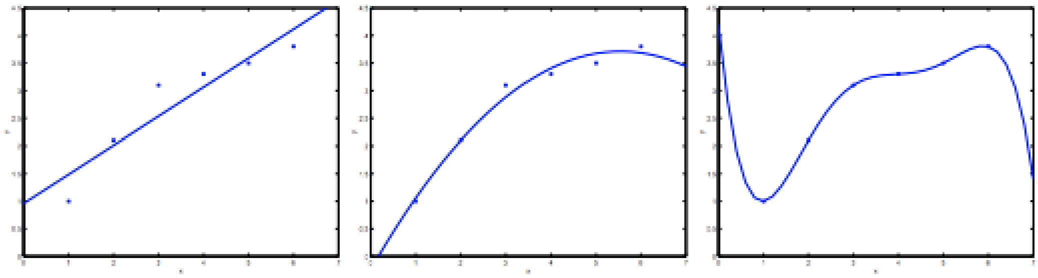
$\theta := \theta - \frac{\alpha}{m}X^T(g(X\theta) - y)$

3 Osztályozás több osztállyal

Pl. email mappákba sorolása (munka,család,spam)
y = 0,1...n

Felbontjuk a problémát n+1 alproblémára, mindegyiknél azt feltételezzük, hogy y az egyik osztályban van, így mindegyik alproblémában y = 0,1, mert vagy tényleg abban az osztályban van vagy nem.
Végül azt választjuk, aminek legnagyobb valószínűsége volt.

4 Túltanulás



A bal oldali ábrán az elsőfokú hipotézis függvényünk nem igazán illeszkedik jól a tanuló példákra, ezt alultanulásnak (underfitting) hívjuk.
A középső példához már hozzáadtunk egy saját x^2 jellemzőt, amivel láthatóan jobban illeszkedik a modellünk, így gondolhatnánk, hogy újabb jellemzők felvétele mindig pontosítja a hipotézis függvényt.
Ennek cáfolata a jobb szélső példa, ahol egy ötödfokú polinom is szerepel a képletben, ezzel szinte tökéletesen illeszkedik a tanuló példáinkra, viszont szemmel láthatóan nem igazán alkalmas arra, hogy egy új példa értékét jósoljuk meg vele. Ezt hívjuk Túltanulásnak (overfitting).

A túltanáls a következő módokon orvosolható:

- Csökkentsük a jellemzők számát
  - Válasszuk ki melyikeket szeretnénk megtartani
  - Használjunk modell szelekciós algoritmust
- Regularizáció
  - Hagyjuk meg a jellemzőket, de bűntessük azokat, amelyek túl nagy hatással vannak
  - A regularizáció akkor működik jól, ha sok olyan változónk van, amik csak kissebb hatással vannak a költségre

Tfh. túltanulás problémánk van és inkább négyzeteshez hasonló függvényt szeretnénk illeszteni e helyett:
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Ha az akarjuk hogy kisebb hatással legyen a $\theta_3 x^3$ és $\theta_4 x + 4$ tag, módosítanunk kell a költség függvényünket:

$min\theta \frac{1}{2m} \sum\limits_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000 * \theta_3^2 + 1000 * \theta_4^2$

Ahhoz, hogy ez a függvény minimális legyen, a θ_3 és θ_4 tagoknak közel 0-nak kell lenni.
Viszont ha túl sok jellemzőnk van, nem biztos hogy tudni fogjuk melyikek vannak nagy hatással, ezért regularizáljuk mindet.

$min\theta \frac{1}{2m} \sum\limits_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum\limits_{j=1}^{n} \theta_j^2$

Ahol λ a regularizációs paraméter, ami meghatározza hogy az egyes θ paraméterek költségei mennyire arányosak.