W06

Fekete Máté

2020 Április

Tartalomjegyzék

1 Hibakeresés az algoritmusban

Tfh. implementáltunk egy reguláris regressziós modellt, ami a lakás árakat jósolja.

$J(\theta) = \frac{1}{2m}[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2]$

Viszont amikor új példákon tesztelnénk a hipotézist kiderül, hogy nagyon nagy hibákat vét. A következőket próbálhatjuk meg:

- Szerezzünk több tanuló példát
- Próbáljuk meg egy kisebb jellemző halmazon
- Próbáljuk meg több jellemzővel
- Adjunk hozzá polinomiális jellemzőket (pl. x_1^2)
- Csökkentsük a λ értéket
- Növeljük a λ értéket

De melyikbe érdemes időt fektetni?

2 A hipotézis értékelése

Osszuk fel az adatkörünket tanuló és tesztelő halmazra. Általában a 70% tanuló, 30% tesztelő arány megfelelő.
Tanuló példáink továbbra is: $(x^{(1)}, y^{(1)}) \dots (x^{(m)}, y^{(m)})$
Tesztelő példák pedig: $(x_{test}^{(1)}, y_{test}^{(1)}) \dots (x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$
Ahol m_{test} a tesztelő példáink száma.

2.1 Teszt halmaz hibája

Lineráis regresszióhoz:
$J_{test}(\theta) = \frac{1}{2m_{test}}[\sum_{i=1}^{m_{test}}(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$ (szimpla négyzetes hiba)
Osztályozáshoz: (0/1) Félreosztályozási hiba
$err(h_\theta(x), y) = \begin{cases} 1 \text{ ha } h_\theta(x) \geq 0.5 \text{ és } y = 0 \text{ VAGY } h_\theta(x) < 0.5 \text{ és } y = 1 \\ 0 \text{ egyébként} \end{cases}$

Ez után az átlagos hiba a teszt halmazra:
$\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} err(h_\theta(x_{test}^{(i)}, y_{test}^{(i)})$

3 Modell választás

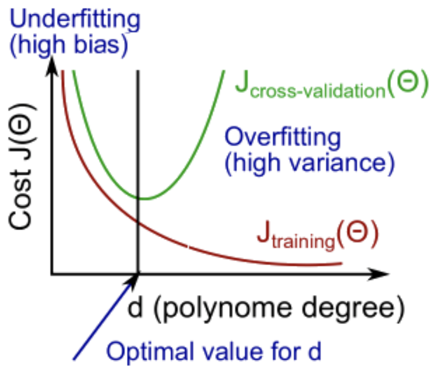
Honnan tudjuk, hogy a hipotézis függvényünket mekkora fokszámának kell lennie, hogy jól illeszkedjen az adatkörünkre?
Azt már tudjuk, hogy a tanuló halmazra való illeszkedést hiába mérjük, mert nem biztosítja azt, hogy új példákon is jól működjön (lásd túltanítás).
Ugyan arra a problémára nézzünk 1 és 10 közötti fokszámú hipotézis függvényeket.
Számoljuk ki mindegyikre a megfelelő Θ paramétert, ezt jelölje $\Theta^{(i)}$, ahol i a fokszám.
Számoljuk ki mindegyikre a teszt halmaz hibáját, majd válasszuk azt, amelyiknél ez a legkisebb volt.

Probléma: az utolsó lépés miatt a teszt halmaz már nem jó mérce a modellünkhöz, hiszen ugyan az történt mint az alap halmazunknál, mivel optimalizáltunk rá, már hiába teszteljük, nem biztos hogy új adatra is jó lesz.

Megoldás: Osszuk eggyel több halmazra.
Így 3 halmazunk lesz, a tanuló, **validációs(cross validation)** és teszt.
Jelölésükben a fentebb láttott alapján (a validációst cv alsó indexszel különböztetjük meg), a hiba értékük simán a négyzetes hiba a megfelelő paraméterekkel (pl. validációnál m_{cv}, $(x_c v^{(i)}, y_c v^{(i)})$, stb.)

Innentől a következő módon értékelhetjük a modellünket:
Ugyan úgy kiszámljuk a Θ értékeket minden fokszámra, de ez után a validációs hibák alapján vesszük a minimálist, majd a teszt hibával megmérhetjük, hogy mennyire működhet jól új példákon.

4 Alultanulás vs Túltanulás



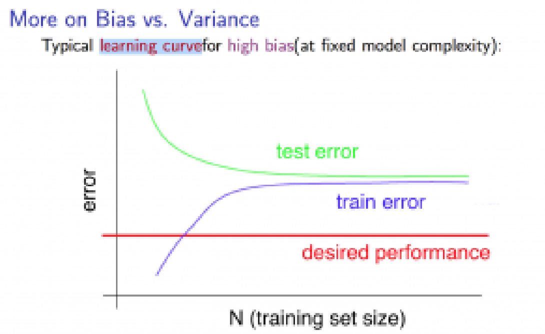
Az alultanulás felismerhető ebből, ha $J_{train}(\Theta)$ és $J_{cv}(\Theta)$ értéke is magas, ilyenkor nagyobb fokszámú polinomot kell használnunk.
Ha $J_{train}(\Theta)$ alacsony, viszont $J_{cv}(\Theta)$ lényegesen magasabb (tehát a tanuló halmazra nagyon jól illeszkedik, viszont új példákra rossz értéket ad), akkor valószínűleg túltanulás a problémánk.

4.1 Regularizáció

Túltanulás esetén a λ regularizációs együttható optimalizálása is hasonló módon történik, annyi különbséggel, hogy ha $J_{train}(\Theta)$ alacsony és $J_{cv}(\Theta)$ magas, akkor túl kicsit a λ értékünk (a tanuló halmaz továbbra is túl jól illeszkedik, nem csökkentettük eléggé a magasabb fokú jellemzők befolyását).
Ha $J_{train}(\Theta)$ és $J_{cv}(\Theta)$ is magas, akkor λ értéke túl nagy, gyakorlatilag egy konstans függvény felé normalizálja a változóinkat, ami nyilván nem túl hasznos.

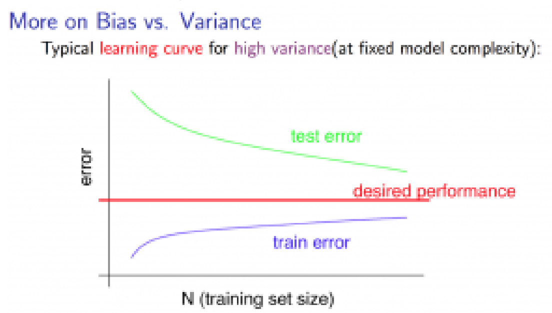
5 Tanulási görbe

Az algoritmusunknak csak egy kisebb tanuló halmazt adunk, majd ezt növeljük. Alultanult algoritmus:



Mivel a tanuló és teszt hiba viszonylag gyorsan közel kerül egymáshoz, nagyobb tanuló halmaz alkalmazása nem javít az algoritmuson.

Túltanult algoritmus:



A tanuló és teszt hiba között egészen addig nagy lesz a különbség, ameddig a jelenleg túl nagy fokú polinomunk már nem tud teljesen pontosan illeszkedni a tanuló halmazra, viszont ez általában csak viszonylag nagy halmazra igaz, sokat javíthat a növelése.

6 Hibakeresés az algoritmusban II

A következtetéseket levonva:

- Szerezzünk több tanuló példát - túltanulás ellen
- Próbáljuk meg egy kisebb jellemző halmazon - túltanulás ellen
- Próbáljuk meg több jellemzővel - alultanulás ellen
- Adjunk hozzá polinomiális jellemzőket (pl. x_1^2) - alultanulás ellen
- Csökkentsük a λ értéket - alultanulás ellen
- Növeljük a λ értéket - túltanulás ellen