Logistična regresija

Gregor Majcen (63070199)

21. maj 2012

1 Uvod

Cilj sedme domače naloge je implementacija logistične regresije z regularizacijo. Kaj točno to pomeni, je podrobneje opisano v naslednjih poglavjih. Uspešnost naše implementacije je potrebno dokazati na realnih podatkih (iz prejšnje naloge) in sicer s 5-kratnim prečnim preverjanjem. Da dokažemo, kako močno je regularizacija pomembna bomo to tudi testirali in primerjali rezultate.

2 Podatki in opis problemske domene

Podatki, ki jih preiskujemo so s področja kemoinformatike. Podatke so bolj podrobno opisani ze v prejšnji nalogi, ampak izpostavimo najpomembnejše. Imamo 1776 atributov, 3751 primerov in en binaren razred. Kot zelo pomemben podatek je, da so atributi normalizirani. V peti domači nalogi smo ugotovili, da pomaga pri hitrosti iskanju minimuma linearne regresije. Za bolj točno računanje smo dodali se dodatnih n*1776 atributov, ki so vsi prejšnji $attr^2$, $attr^3$, ..., $attr^n$. S tem smo simulirali višjo stopnjo polinoma. Zaradi prevelikega prileganja podatkov je tu potrebna še regularizacija.

3 Metoda

Logistična regresija je metoda za klasifikacijo diskretnih problemov. Je zelo podobna linearni regresiji, ki smo jo implementirali v eni izmed prejšnjih nalog. Edina razlika je v naši hipotezi h.

Pri linearni regresiji je $h_\theta(x) = \theta^T x$, kar pa tu ne pride več v poštev, saj ni razloga, zakaj bi hoteli imeti ciljne vrednosti večje od 1 ali manjše od 0. To popravimo s tako imenovano *sigmoidno funkcijo* $g(z) = \frac{1}{1+e^{-z}}$. Sigmoidna funkcija je vse kar potrebujemo: je monotono naraščujoča, zavzema vrednosti $(0, 1)$ in ima zelo lep odvod: $g'(z) = g(z)(1 - g(z))$.

Naša nova funkcija za hipotezo je torej $h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$

Ker je sedaj razred binaren, lahko tudi *verjetje* (ang. likelihood) zelo poenostavimo. Kot vemo že od prejšnjič, mora biti verjetje čimvečje. Za lažje računanje raje vzamemo logaritem

verjetja:

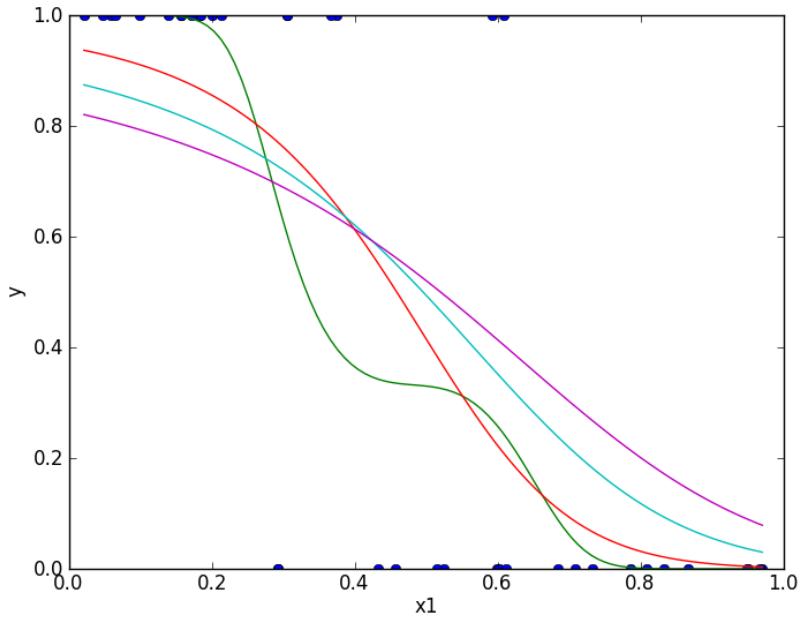
$$l(\theta) = \sum_{i=1}^{m} y^{(i)} * log(h(x^{(i)})) + (1 - y^{(i)}) * log(1 - h(x^{(i)}))$$

in njegov odvod

$$\frac{\partial}{\partial \theta_j} l(\theta) = (y - h_\theta(x)) x_j$$

Cenovna funkcija, ki jo potrebujemo za naš algoritem je $-l(\theta)$ in jo je potrebno minimizirati. Posledično se tudi pri odvodu doda negativen predznak. $-\frac{\partial}{\partial \theta_j} l(\theta)$ je torej naša gradientna funkcija. S pomočjo že znanega in kar hitrega algoritma L-BFGS poiščemo minimum cenovne funkcije in kot rezultat dobimo optimalne θ.

Zaradi simuliranja višje stopnje polinoma se podatki preveč prilagajajo. Zato je potrebna regularizacija. Naši cenovni funkciji prištejemo še $\lambda * \sum_{i=1}^{n} \theta_i^2$, kjer je λ poljuben parameter. Posledično je potrebno spremeniti tudi gradientno funkcijo. Tu pa prištejemo $\lambda * \sum_{i=1}^{n} 2 * \theta_i$. Na sliki 1 je lepo razvidno, kako se funkcije prilagajajo.



Slika 1: Sigmoidna funkcija pri različnih vrednosti λ

4 Rezultati

Logistično regresijo sem testiral s 5-kratnim prečnim preverjanjem in izračunal logLoss oceno.

Stopnja polinoma	λ	logLoss
2	0	2.62
1	0.01	0.50
1	0.005	0.52
2	0.01	0.51
2	0.001	0.58
3	0.01	0.52
3	0.001	0.62

Tabela 1: tabela rezultatov

V tabeli 1 je tudi lepo razvidno, da brez regularizacije enostavno ne gre. S pomočno poskušanj sem dobil najboljše rezultate pri $\lambda = 0.01$ in originalnih podatkih. V tabeli so prikazani le pomembnejši rezultati.

5 Izjava o izdelavi domače naloge

Domačo nalogo in pripadajoče programe sem izdelal sam.