13M051MU Mašinsko učenje
1. domaći zadatak 2024/25
Generalizovani linearni modeli

1 Regresija

U datoteci `data-reg.csv` su podaci za jedan regresioni problem. Format datoteke je sledeći:

- prvih 5 kolona su prediktori;
- poslednja kolona je ciljna promenljiva;
- svaka vrsta predstavlja jedan obučavajući primer.

Model koji treba da implementirate određen je zbirom S cifara vašeg broja indeksa, po modulu 3:

S mod 3	Model
0	grebena (*ridge*) regresija
1	LASSO regresija
2	lokalno ponderisana linearna regresija

Na primer, za broj indeksa 20**/3221 je

$$S = 3 + 2 + 2 + 1 = 8, \quad S \bmod 3 = 2$$

pa je metod lokalno ponderisana linearna regresija. Kod grebene i LASSO regresije za hipotezu usvojite polinom 2. reda.

Rešenje podrazumeva kompletan kod i izveštaj sa sledećim elementima:

1) grafikom koji demonstrira kako ste izabrali hiper-parametar vašeg modela,
2) izabranom vrednošću hiper-parametra,
3) procenom vrednosti korena srednje-kvadratne greške vašeg modela.

2 Klasifikacija

Poslednja kolona u datoteci `data-class.csv` je oznaka klase. Ostale kolone sadrže vrednosti prediktora. Studenti sa parnim brojem (*ne* godinom) indeksa rade sa logističkom regresijom, a oni sa neparnim brojem rade sa softmaks klasifikatorom. Koristite samo originalne prediktore (bez polinomijalne ekspanzije prediktora). U ovom slučaju, modeli su dovoljno jednostavni, tako da nije potreba regularizacija.

2.1 Logistička regresija (LR)

Pošto je u pitanju problem sa više od 2 klase, treba primeniti pristup tipa "jedan-protiv-ostalih":

- projektuje se po jedan klasifikator za svaku klasu, koji treba da prepozna da li je primer iz te klase ili iz neke (bilo koje) od preostalih;
- novi primer se pušta kroz sva 3 klasifikatora, a konačna odluka se donosi poređenjem vrednosti hipoteza.

2.2 Softmax

U nastavku je izvođenje izraza za gradijent kriterijumske f-je. Prirodni parametar multinomijalne raspodele je

$$\boldsymbol{\eta} = \begin{bmatrix} \ln \phi_1/\phi_k & \cdots & \ln \phi_{k-1}/\phi_k \end{bmatrix}^\mathsf{T}$$

gde je $\phi_i = p(y = i \,|\, \mathbf{x}; \boldsymbol{\theta})$. Usvajamo $\eta_i = \boldsymbol{\theta}_i^\mathsf{T}\mathbf{x}$, pa je

$$\phi_i = \frac{e^{\boldsymbol{\theta}_i^\mathsf{T}\mathbf{x}}}{1 + \sum_{j=1}^{k-1} e^{\boldsymbol{\theta}_j^\mathsf{T}\mathbf{x}}}, \quad i = 1, \ldots, k-1,$$
$$\phi_k = \frac{1}{1 + \sum_{j=1}^{k-1} e^{\boldsymbol{\theta}_j^\mathsf{T}\mathbf{x}}}.$$

Definišimo $\boldsymbol{\theta}_k = \mathbf{0}$ i matricu parametara

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\theta}_1 & \cdots & \boldsymbol{\theta}_k \end{bmatrix}.$$

Sada je

$$\phi_i = \frac{e^{\boldsymbol{\theta}_i^\mathsf{T}\mathbf{x}}}{\sum_{j=1}^{k} e^{\boldsymbol{\theta}_j^\mathsf{T}\mathbf{x}}}, \quad i = 1, \ldots, k.$$

Izraz za log-verodostojnost je

$$J(\boldsymbol{\Theta}) = l(\boldsymbol{\Theta}) = \sum_{i=1}^{m} \ln \phi_{y^{(i)}} = \sum_{i=1}^{m} \boldsymbol{\theta}_{y^{(i)}}^\mathsf{T}\mathbf{x}^{(i)} - \ln\left(\sum_{j=1}^{k} e^{\boldsymbol{\theta}_j^\mathsf{T}\mathbf{x}^{(i)}}\right)$$

Gradijent po $\boldsymbol{\theta}_l$ je

$$\nabla_{\boldsymbol{\theta}_l} J(\boldsymbol{\Theta}) = \sum_{i=1}^{m} \left(\mathbb{1}_{y^{(i)}=l} - \frac{e^{\boldsymbol{\theta}_l^\mathsf{T}\mathbf{x}^{(i)}}}{\sum_{j=1}^{k} e^{\boldsymbol{\theta}_j^\mathsf{T}\mathbf{x}^{(i)}}}\right)\mathbf{x}^{(i)}$$

Jedna iteracija gradijentnog uspona data je sa

$$\boldsymbol{\theta}_l \leftarrow \boldsymbol{\theta}_l + \alpha \nabla_{\boldsymbol{\theta}_l} J(\boldsymbol{\Theta}), \quad l = 1, \ldots, k-1$$

($\boldsymbol{\theta}_k$ se ne ažurira i uvek ostaje $\mathbf{0}$).

2.3 Numerička optimizacija

Za numeričku maksimizaciju verodostojnosti koristite stohastički gradijentni uspon sa "mini-šaržama". Eksperimentišite sa stopom učenja α i veličinom mini-šarže m_{mb}, i probajte da odredite kombinaciju koja daje najbržu konvergenciju. Opciono, možete eksperimentisati sa naprednim tehnikama (inercija, AdaGrad, RMSProp, Adam). Ilustrujte konvergenciju sa pet grafika:

- jedan sa "optimalnom" kombinacijom vrednosti α^* i m_{mb}^*,
- dva sa α^* i prevelikim, odnosno premalim m_{mb},
- dva sa m_{mb}^* i prevelikim, odnosno premalim α.

Na x-osama ovih grafika treba da bude prikazan broj obučavajućih primera kroz koje je algoritam "prošao"[1]: u jednoj iteraciji "konzumirate" jednu mini-šaržu, pa će x-koordinata prve tačke na grafiku biti m_{mb} itd. Na y-osama

[1] Jedna iteracija sa velikim m_{mb} može biti numerički skuplja od nekoliko iteracija sa manjim m_{mb}; ako bi x-osa prikazivala redni broj iteracije, ne bismo imali fer poređenje.

grafika treba da bude log-verodostojnost. Kod logističke regresije imate 3 modela; prikažite log-verodostojnosti za svaki od njih na istom grafiku (različitim bojama), ili samo za jedan od njih—odlučite sami šta je ilustrativnije.

Na kraju izveštaja navedite ostvarenu tačnost modela na trening i test skupu.

Napomene

Pretporuka je da se domaći zadatak radi u Python-u, a toleriše se i Matlab/Octave. Nije dozvoljeno korišćenje gotovih f-ja za linearnu, polinomijalnu, grebenu, LASSO i lokalno ponderisanu regresiju, logističku regresiju, softmax, gradijentni spust, niti za validaciju, osim u cilju provere ispravnosti sopstvenih rezultata. Sve f-je treba da napišete sami, koristeći samo konstrukte za rad sa vektorima i matricama, optimizaciju, generisanje slučajnih brojeva, crtanje grafika (Python moduli numpy, scipy, matplotlib) i sl.

Rešenje predajete putem Teams-a do naznačenog datuma. Opciono, kôd i izveštaj možete objediniti u Jupyter Notebook; u tom slučaju pošaljite ipynb i njegovu html verziju sa generisanim rezultatima. Sve fajlove spakujte u zip datoteku i nazovite je `ime-prezime-mu-dz1.zip`. Upustvo za predaju zadatka putem MS Teams-a možete naći ovde.