

1. ZADATAK

Q1: Koliko iznosi širina margine?

A1: Širina margine iznosi 2 (vidljivo sa slika, a i izračunato)

Q2: Koji primjeri su potporni vektori i zašto?

A2: Primjeri [5,2], [5,4], [3,2]; zato što leže na margini.

Q3: Zašto stršeća vrijednost ne utječe na SVM?

A3: Na SVM općenito ne utječu stršeće vrijednosti (neaktivna ograničenja), već samo oni primjeri koje se nalaze na margini (aktivna ograničenja).

Q4: Kako se linearan SVM nosi s linearno neodvojivim skupom podataka?

A4: Linearan SVM s mekom marginom dozvoljava da neki primjeri budu na pogrešnoj strani granice, ali to kažnjavamo tim više što su primjeri nalaze dublje na pogrešnoj strani granice. Parametar C određuje kompromis između veličine margine i ukupne kazne. Veći C dovodi do većeg kažnjavanja pogrešne klasifikacije, što za posljedicu ima složeniji model.

Q5: Zašto SVM ipak uspijeva pronaći nekakvu granicu kod linearno neodvojivog problema, iako koristimo linearnu jezgru?

A5: Zbog meke margine, kao što je objašnjeno u A4.

3. ZADATAK

Q1: Razlikuje li se površina pogreške na skupu za učenje i skupu za ispitivanje? Zašto?

A1: Razlikuju se, što je očekivano. Model treniramo na skupu za učenje te se on dobro prilagodi podacima za učenje te je stoga pogreška na skupu za učenje mala. Pogreška na skupu za ispitivanje je veća jer su to modelu novi, neviđeni podaci.

Q2: U prikazu površine pogreške, koji dio površine odgovara prenaučenosti, a koji podnaučenosti? Zašto?

A2: Na grafovima u 3.b) : Područje dolje lijevo (mali gamma, mali C) odgovara podnaučenosti; i pogreška učenja i ispitna pogreška su velike (na grafovima crne). Područje gore desno (veliki gamma, veliki C) je pogreška učenja mala (bijela), a ispitna pogreška velika (crna), što je karakteristika prenaučenosti.

Q3: Kako broj dimenzija n utječe na površinu pogreške, odnosno na optimalne hiperparametre (C^*, γ^*) ?

A3: Za veći broj dimenzija (featurea) dobijemo manje optimalne parametre C i gamma; to je očekivano jer je model lako prenaučiti za složeniji (višedimenzionalni) skup podataka. Stoga biramo manji C i manju gamma kako bismo spriječili prenaučenost modela.

Q4: Preporuka je da povećanje vrijednosti za γ treba biti popraćeno smanjenjem vrijednosti za C. Govore li vaši rezultati u prilog toj preporuci? Obrazložite.

A4: Govore. S grafa u 3.b), ispitna pogreška na skupu 1 (skup s 2

featurea) vidljivo je da je "bijeliji" (manja pogreška) dio grafa raspoređen dijagonalno – tamo gdje je C velik, a γ mali imamo relativno bijelo područje, te možemo vidjeti dijagonalno kako se C smanjuje, a γ raste da područje ostaje relativno bijelo. [Q2 i Q4 ako ne skužiš iz ovog objasnim ti sutra prije labosa, puno je lakše objasniti direktno uz sliku jer su uz to i vezana pitanja]

Q5: Podrazumijevana vrijednost parametara je $C=1$ i $\gamma=1/n$. Bi li te vrijednosti bile optimalne u ovom slučaju?

A5: U mome slučaju (kod tebe su vjerojatno drugačiji optimalni C i γ) ne bi. C mi je veći od 1 za oba skupa.

4. ZADATAK

Q1: Kako radi ovo skaliranje?

A1: MinMax skaliranje prebacuje sve značajke u raspon $[0,1]$.

Q2: Dobiveni histogrami su vrlo slični. U čemu je razlika?

A2: Razlika je što su, očekivano, kod skaliranja obje značajke u rasponu $[0, 1]$, dok su neskalinane vrijednosti (a) podzadatak) u drugačijim rasponima.

Q3: Kako radi ovo skaliranje?

A3: StandardScaler oduzima od podatka oduzima srednju vrijednost te to dijeli sa standardnom devijacijom, proces koji se naziva standardizacijom. Tada su svi podaci centrirani (srednja vrijednost je 0) i imaju relativno skaliranu varijancu. [ovo s varijancom ne kužim skroz, odgovor je s wikipedije]

Q4: Dobiveni histogrami su vrlo slični. U čemu je razlika?

A4: Razlika je što su ovdje podaci centrirani oko 0, dok kod neskalinanih podataka to nije bio slučaj.

Q5: Jesu li rezultati očekivani? Obrazložite.

A5: Jesu. Modeli naučeni na skaliranim podacima imaju veću točnost od modela naučenog na neskalinanim podacima.

Q6: Bi li bilo dobro kada bismo funkciju `'fit_transform'` primijenili na cijelom skupu podataka? Zašto? Bi li bilo dobro kada bismo tu funkciju primijenili zasebno na skupu za učenje i zasebno na skupu za ispitivanje? Zašto?

A6: Nemam pojma i ne znam odakle da shvatim.

5. ZADATAK

Q1: Kako k utječe na izgled granice između klasa? A broj primjera N ?

A1: Vidljivo sa slika:: S većim k izgled granice je jednostavniji, što je očekivano budući da s porastom k pada složenost modela. S porastom broja primjera N izgled granice ostaje sličan.

6. ZADATAK

Q1: Kojem području odgovara prenaučenosť, a kojem podnaučenost

modela? Zašto?

A1: Lijevo prenaučenos (mali k), desno podnaučenos (veliki k). Jer za mali k šum znatno utječu na predikciju. Ako je npr. $k=1$ i imamo 1 primjer koji je zbog šuma završio na suprotnoj strani, svi novi primjeri koji su njemu najbliži susjedi će biti krivo klasificirani. Dok ako je $k=2+$, svi primjeri oko njega će biti ispravno klasificirani. [opet možda malo teško za shvatiti samo iz teksta, ako ne skužiš objasnim sutra na papiru].

Q2: Je li uvijek moguće doseći pogrešku od 0 na skupu za učenje?

A2: Ako je $k=1$ pogreška će uvijek biti 0 na skupu za učenje, ali ako je $k>1$ neće. Ako je $k=1$, primjer za učenje će biti najbliži susjed samome sebi pa će se ispravno klasificirati. Dok ako je $k>1$, možemo imati sličnu situaciju kao opisanu u prošlom pitanju pa bi se primjer krivo klasificirao. [opet, lakše je na papiru ako ne shvatiš].

7. ZADATAK

Q1: Zašto je ovaj problem tako izražen kod algoritma k -nn?

A1: K -nn algoritam za klasifikaciju koristi euklidsku udaljenost dvaju primjera. Problem nastaje kada imamo jednu dimenziju mnogo veću od druge. Npr., imamo 2 primjera za učenje, jedan $[1000, 0.1]$, drugi $[1005, 0.2]$. Dolazi primjer za ispitivanje $[1010, 0.11]$. Euklidska udaljenost od prvog primjera je $\sqrt{(1010-1000)^2 + (0.11-0.1)^2} = 100.0001$. Euklidska udaljenost od drugog primjera je $\sqrt{(1010-1005)^2 + (0.2-0.11)^2} = 25$. Nestosito. Taj će primjer biti klasificiran kao primjer $[1005, 0.2]$ jer je njemu apsolutno gledano bliže. Ali relativno gledano, on je zapravo trebao biti klasificiran kao primjer $[1000, 0.1]$ – jer je 0.11 puno bliže 0.1 nego 0.2, a između 1000 i 1005 relativno gledano nema velike razlike. [opet je sjeban objasniti tekstom, a zapravo je dosta jednostavno pa ti objasnim sutra ako ne skužiš iz ovog].

Q2: Zašto nebitne značajke ovoliko utječu na performanse modela?

A2: Ako imamo 5 nebitnih značajki i 1 bitnu značajku te se naš ispitni primjer skroz poklapa u nebitnim značajkama s nekim primjerom za učenje, on će vjerojatno biti klasificiran kao taj primjer, iako možda imamo neki primjer za učenje s kojim se poklapa u bitnoj značajci.

Primjer: prva značajka je bitna, ostale 4 su nebitne; primjeri za učenje su $[1\ 2\ 3\ 4\ 5]$ i $[-5\ 10\ 11\ 12\ 13]$. Dođe primjer $[1\ 10\ 11\ 12\ 13]$. On će biti klasificiran kao primjer $[-5\ 10\ 11\ 12\ 13]$ jer je njemu euklidski gledano bliže, a s njim se ne poklapa u bitnoj značajci, za razliku od poklapanja u bitnoj značajci kod prvog primjera za učenje. I to je problem jer mi želimo klasificirati primjer prema bitnim značajkama.