Pēteris Račinskis
pr20015

Mājas darbs 5: SVM, naive Bayes

Uzdevums 1

Piebilde par kerneļiem: nelineāriem kerneļiem SVM rezultāts dažādu programmatūras pakotņu dokumentācijā (un WEKA izdrukā) parasti ir dots kā funkcija no atbalsta vektoriem (t.i., treniņa datu kopas punktiem ar nenulles svaru) xi:

$$f(x) = \sum_i \alpha_i \gamma_i K(\bar{x_i}, \bar{x}) + b \geq 0$$

, kur alfa, gamma - apgūtais svars un dotā zīme atbalsta vektoram. Nelineārai kerneļa funkcijai nevar izteikt vienu vektoru w mazāk-dimensionālajā telpā, kur pastāv treniņa kopas un neklasificētie datu punkti -

$$\alpha\gamma K(\bar{x_i}, \bar{x}) \neq K(\alpha\gamma\bar{x_i}, \bar{x}) \neq K(\alpha\gamma\bar{x_i}) \cdot K(\bar{x}) \Rightarrow$$
$$\Rightarrow \sum_i \alpha_i \gamma_i K(\bar{x_i}, \bar{x}) \neq K(\bar{w}^T, \bar{x})$$

Atšķirīga ir arī optimizācijas problēmas uzstādne. Dažas saites, no kurām ņemtas formulas lapas apakšā.[1] Pāri palikušo atbalsta vektoru skaits WEKA rezultātā tiek uzrādīts ("support vectors") un nosaka modeļa izmērus.

a. SVM - Ionosphere rezultāti ar dažādām kernel funkcijām

Apskatītie kerneļi - polinomiālais (pakāpē 1, 2), RBF, PUK.

- Lineārais (pol. pakāpē 1): vienkāršākais kerneļa veids un sākotnējais problēmas formulējums, kuru var izteikt arī kā vienkāršu vektoru skalāro reizinājumu. Ar noklusējuma parametriem iegūst 88.6% precizitāti. Atbalsta vektors - viens, jo visi atbalsta vektori tiek lineāri kombinēti vektorā *w*.
- Kvadrātiskais (pol. pakāpē 2): lineāri kombinēt treniņa datu kopas vektorus vairs nevar, taču ar noklusējuma parametriem iegūta 90.6% precizitāte, izmantojot 74 nenulles atbalsta vektorus.
- RBF: radiāli eksponenciāla funkcija. Ar noklusējuma iestatījumiem iegūtā precizitāte ir sliktāka, taču šajā gadījumā pats kernelis ir parametrisks. Pie sākotnējā gamma = 0.01 precizitāte ir tikai 76% ar 249 atbalsta vektoriem, taču palielinot to var samazināt atbalsta vektoru skaitu - 111 ar 93.4% precizitāti pie gamma = 0.35. Tālāk palielinot gamma līdz 0.5 var iegūt precizitāti 94.9% ar 127 vektoriem. Tālāk

[1] https://www.robots.ox.ac.uk/~az/lectures/ml/lect3.pdf
https://ai6034.mit.edu/wiki/images/SVM_and_Boosting.pdf
https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.391.6045&rep=rep1&type=pdf

palielinot šo lielumu precizitāte atkal samazinās un atbalsta vektoru skaits atkal pieaug.

- PUK: Pīrsona VII sadalījuma kernelis ar diviem parametriem - sigma un omega. Pēc noklusējuma - abi ir vienādi ar 1. Rezultāts - 94% precizitāte ar 221 vektoru. Rupjos soļos mainot tikai sigmu, var sasniegt 95.2% ar 162 vektoriem pie vērtības 2. Mainot tikai omegu, var sasnieg 94.6% ar 236 vektoriem pie 0.5. Abiem ir šķietami pretēja ietekme, tāpēc tika pamēģināts abus pacelt uz 5, un rezultātā tika saglabāta 92% precizitāte tikai šoreiz ar vien 137 vektoriem.

b. SVM - kernel funkciju parametri

Parametri kerneļiem:

- Lineārais, kvadrātiskais - abi definēti zem polinomiālā kerneļa, kur vienīgais acīmredzami saprotamais parametrs ir polinoma pakāpe. Gadījumā 1 - vienkāršs vektoru skalārais reizinājums.
- RBF: parametrs gamma ir konstante, ar ko reizināts vektoru starpības garuma kvadrāts eksponentē. T.i., liela gamma - vērtība strauji tuvojas nullei. Maza gamma, vērtība krītas lēnāk un lielāka nozīme ir tālākiem datu punktiem.
- PUK: sigma nosaka sadalījuma platumu, omega ļauj kropļot tā formu - tuvu normālajam pie lieliem omega, ar “resnākām astēm” pie maziem. Detalizēti aprakstīti pēdējā saitē iepriekšējās lapas apakšā.

Pēteris Račinskis
pr20015

Uzdevums 2

a. Rezultāti

Izmantotas 3 metodes:

- naive bayes: 95.281%
- naive bayes multinomial text: 92.285%
- naive bayes updateable: 95.281%
- naive bayes (updateable) with kernel: 95.944%

b. Atšķirības starp metodēm

“NaiveBayes” ir lekcijas materiālā aprakstītā algoritma versija, kas spēj apstrādāt nepārtrauktus datus. “NaiveBayesUpdateable” ir tā paša algoritma implementācija, bet ar iespēju rezultējošo modeli papildināt/atjaunināt (“update”). “NaiveBayesMultinomialText” vienkārši atmet tos parametrus, kas nav diskrēti (t.i., kas nav uzdoti teksta vai teksta kopu(?) formā). “useKernelEstimator” iespēja aizstāj normāli sadalīta lieluma pieņēmumu, apstrādājot nepārtrauktus datus, ar neparametrisku “kernel density estimator” funkciju, kas izmanto kerneli, lai salīdzinātu punktu ar jau esošajiem, nevis varbūtības blīvuma funkciju.