TARTU ÜLIKOOL

Arvutiteaduse instituut

Informaatika õppekava

Liina Anette Pärtel

Märgendite silumine klassifikaatorite logistilisel kalibreerimisel

Bakalaureusetöö (9 EAP)

Juhendaja: Meelis Kull, PhD

Tartu 2019

Märgendite silumine klassifikaatorite logistilisel kalibreerimisel

Lühikokkuvõte: Antud töös kirjeldati põhjalikult tõenäosuslike klassifikaatorite kalibreerimist logistilise kalibreerimise ehk Platti skaleerimise meetodiga. Töö käigus viidi läbi eksperimendid otsimaks logistilise kalibreerimise meetodi silumismäära väärtusest paremat väärtust. Eksperimendid viidi läbi sünteetilistel andmestikel, varieerides nii klasside suurust kui klassijaotust. Eksperimentide tulemustest sai teada, et Platti skaleerimise meetodis valitud silumismäär pole ühelgi vaadeldaval andmestikul optimaalne. Veel leiti, et optimaalne silumismäär sõltub lisaks klasside suurusest ka mudeli veamäärast.

Võtmesõnad: masinõpe, mudeli kalibreerimine, Platti skaleerimine, logistiline regressioon, tõenäosuslik klassifikaator

CERCS: P176 Tehisintellekt

Label smoothing in logistic calibration of classifiers

Abstract: This thesis gives a detailed overview of calibrating probabilistic classifiers with logistic calibration also known as Platt scaling. Experiments were carried out to find better label smoothing parameter values than what is used in logistic calibration. Experiments were carried out on toy datasets, varying the size and distribution of classes. The results also show that the current label smoothing parameter formula for Platt scaling is not the optimal value for any of the chosen datasets. It is also noteworthy that the optimal label smoothing parameter depends on both class size and error rate.

Keywords: machine learning, model calibration, Platt scaling, logistic regression, probabilistic classifier

CERCS: P176 Artificial intelligence

Sisukord

Sissejuhatus

Viimasel ajal on masinõppe kasutamine erinevates valdkondades kõvasti kasvanud (Langley, 2011). Näiteks kasutatakse masinõpet tervishoius täpsemate diagnooside andmisel, panganduses pettuste leidmisel ja jaekaubanduses kliendile sobivate toodete soovitamisel (Machine Learning: What it is and why it matters, n.d.). Sealjuures antakse arvutitele aina rohkem vabadust iseseisvalt oma ennustuste põhjal otsuseid langetada. Muuhulgas isejuhtivate elektriautode tootmisega tegelev firma Tesla kasutab oma autodes masinõpet, mille abil saadud ennustuste põhjal saab autopiloodi süsteem teha erinevaid toiminguid: leida sihtkohta jõudmiseks optimaalne teekond, vahetada ise sõiduridu ja parkida auto iseseisvalt parkimiskohale (Tesla Inc., n.d.).

Kõige levinumad masinõppe ülesanded on regressioon ja klassifikatsioon. Regressiooni puhul on tegu reaalarvulise ennustuse tegemisega, klassifikatsiooni puhul andmetepunktide klassideks jaotamine. Näiteks binaarse ehk kahe klassiga klassifikatsiooni puhul võib meetodit kasutada meilide seas spämmi tuvastamisel – klassifitseerija mudel peab otsustama, kas sissetulnud e-kiri kuulub klassi rämspost või mitte-rämpspost. Samuti peab eelmainitud isejuhtiva auto programm enne sõitmise alustamist olema kindel, et tee on vaba. Selleks, et õnnetust vältida, tahame, et mudel oleks veendunud oma ennustuses. Seda ei saa me kontrollida ainult ennustatud klassi teades. Õigete ja täpsete otsuste tegemiseks on tarvis lisaks ennustatud klassile teada ka mudeli hinnangut oma ennustusele – kui suure tõenäosusega on just see vastus õige (Flach, 2012).

Enamik klassifitseerimismeetoditest väljastavad vajadusel tõenäosushinnangu ennustusele, kaasa arvatud naiivne Bayes (Naive Bayes), otsustuspuu (Decision Tree) ja logistiline regressioon (Zadrozny & Elkan, 2001). Mõned neist meetoditest on aga tõenäosuste väljastamisel liiga enesekindlad. Näiteks kui mudel ennustab kümnel andmepunktil positiivsesse klassi kuulumise tõenäosuseks 0.9, kuid tegelikult kuulub neist ainult seitse punkti positiivsesse klassi, siis loetakse mudelit liiga enesekindlaks. Seda saab aga leevendada Platti skaleerimise meetodit ehk logistilist kalibreerimist kasutades.

Käesolev töö uurib, kas Platti skaleerimine töötab ning leida erinevate eksperimentide tulemusena, kas see meetod on parim variant liigse enesekindluse leevendamiseks.

1. Põhimõisted ja definitsioonid

Antud peatükk kirjeldab Platti skaleerimise olemust ja selle mõistmiseks vajalikke definitsioone ja mõisteid.

1.1 Tõenäosuslik klassifikaator

Klassifitseerimine on andmepunktide etteantud klassideks jaotamine, kus mudeli treenimiseks kasutatav andmestik on formaadis (x, y), kus $x \epsilon \mathrm{X}$ on andmepunkt ja y selle andmepunkti tegelik klass (Flach, 2012). Tõenäosuslik klassifikaator on klassifikaator, mis andmepunkti klassi ennustuse asemel annab mingisse klassi kuulumise tõenäosuse (Flach, 2012). Enamjaolt ei piisa täpsete ennustuste tegemiseks lihtsalt tõenäosusest, vaid on vaja, et mudeli ennustused oleksid ka piisavalt hästi kalibreeritud (Zadrozny & Elkan, 2001).

Mudeli kalibreerimine on meetod, millega saab lineaarse klassifikatsiooni mudeli tulemused muuta mingisse klassi kuulumise tõenäosusteks (Flach, 2012). Samuti kasutatakse kalibreerimist juhul, kui esialgne mudel pole kalibreeritud. Mudel on hästi kalibreeritud juhul, kui ennustatud tõenäosuste jaotus on võrdne vaadeldud klasside jaotustega (Kull, Silva Filho, & Flach, 2017). Näiteks vaadeldes kõiki andmepunkte, kus ennustatud positiivsesse klassi kuulumise tõenäosus on 0.9, peaks täpselt 90% nendest punktidest kuuluma positiivsesse klassi. Näiteks logistiline regressioon on õppimisalgoritm, mille kasutamisel saadud mudel on suhteliselt hästi kalibreeritud.

1.2 Ühetunnuseline logistiline regressioon

Ühetunnuseline (*univariate*) logistiline regressioon on tõenäosuslik klassifikaator, mida võib seletada kui lineaarset klassifikaatorit, mis on logistiliselt kalibreeritud (Flach, 2012). Logistilise regressiooni algoritm võtab aluseks mingi lineaarse klassifikaatori mudeliga saadud ennustused ning sobitab neile logistilise funktsiooni, mis näitab positiivsesse klassi kuulumise tõenäosust.

Logistilise regressiooni abil logistilise funktsiooni sobitamine andmestikule ja positiivsesse klassi kuulumise tõenäosus on defineeritud valemiga (Flach, 2012):

$$P(y = 1|\mathrm{X}) = \frac{1}{1 + \ \mathrm{e}^{-(AX+B)}}$$

kus X – tunnuse väärtus ning parameetrid A ja B leitakse logaritmkadu (vt peatükk 1.3) minimeerides valemiga

$$\min -\sum_{i} t_i \log(p_i) + (1 - t_i) \log(1 - p_i)$$

Logistilist regressiooni saab sobitada mitmetunnuselistele andmestikele, kuid siin töös kasutatakse mudeli kasutamiseks ühte tunnust. Sel juhul peab mudel treenima kaks parameetrit: logistilise funktsiooni tõus ja vabaliige.

Võtame 40-andmepunktilise andmestiku, kus on 20 positiivset punkti, mille tunnuse väärtused on genereeritud normaaljaotusest keskväärtusega $\mu = 0.5$ ja standardhälbega $\sigma = 1$, ja 20 negatiivset punkti, mille tunnuse väärtused on genereeritud normaaljaotusest keskväärtusega $\mu =$ -0.5 ja standardhälbega $\sigma = 1$. Joonisel 1 on kujutatud need andmepunktid ning nende tunnuse väärtuste normaaljaotused.

Arvutame sellele andmestikule sobiva Bayesi-optimaalse ehk parima võimaliku tõenäosusliku klassifikaatori. Bayesi-optimaalne klassifikaator on defineeritud valemiga

$$p^*(x) = \frac{P(X = x|Y = 1)}{P(X = x|Y = 0) + P(X = x|Y = 1)}$$

Kui eeldada, et

$$P(X = x|Y = 1) = N(x|\mu_1, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu_1)^2}{2\sigma^2}}$$

$$P(X = x|Y = 0) = N(x|\mu_0, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu_0)^2}{2\sigma^2}}$$

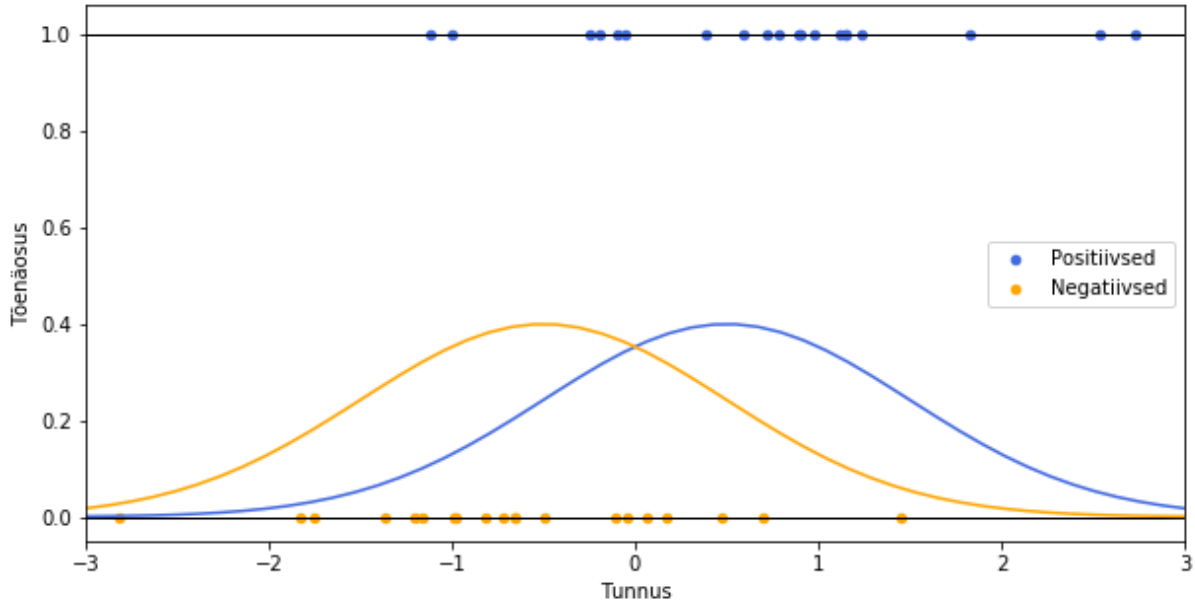
siis saab Bayesi-optimaalse tõenäosusliku klassifikaatori välja arvutada valemiga

$$p^*(x) = \frac{P(X = x|Y = 1)}{P(X = x|Y = 0)+P(X = x|Y = 1)} = \frac{e^{-\frac{(x-\mu_1)^2}{2\sigma^2}}}{e^{-\frac{(x-\mu_1)^2}{2\sigma^2}} + e^{-\frac{(x-\mu_0)^2}{2\sigma^2}}}$$

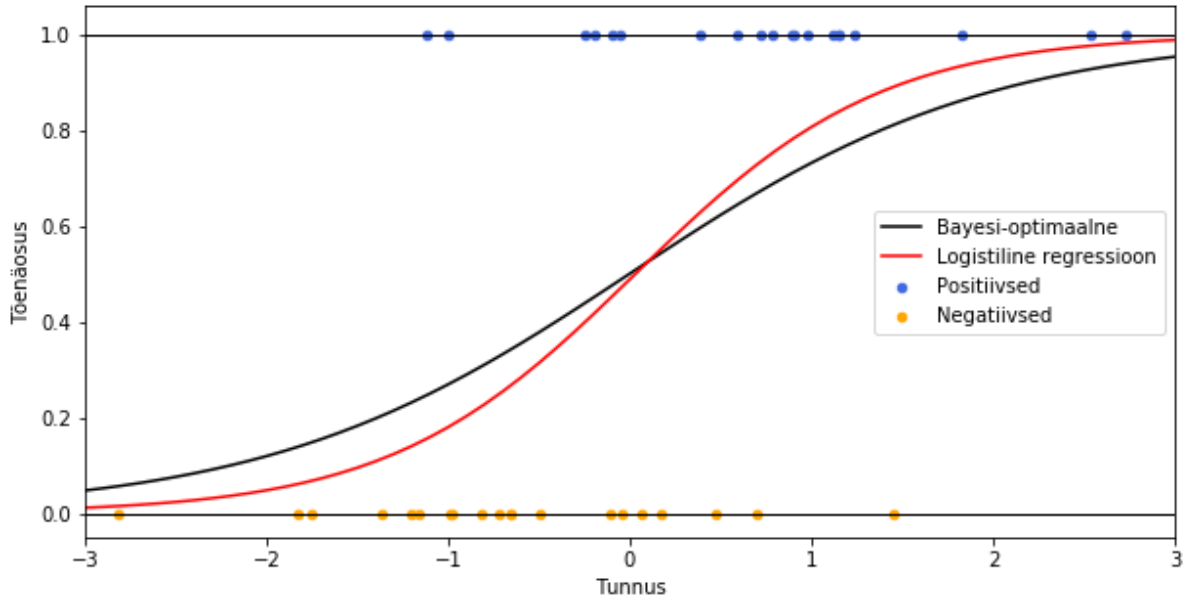
$$= \frac{1}{1 + e^{-\left(\frac{\mu_1-\mu_0}{\sigma^2}x + \frac{\mu_0^2-\mu_1^2}{2\sigma^2}\right)}} = \frac{1}{1 + e^{-(wx+b)}}$$

Meie näites, kus ühe normaaljaotuse parameetrid on $\mu_1 = 0.5, \sigma = 1$ ja teise omad $\mu_0 = -0.5, \sigma = 1$, on $w = \frac{\mu_1 - \mu_0}{\sigma^2} = \frac{0.5-(-0.5)}{1^2} = 1$ ja $b = \frac{\mu_0^2 - \mu_1^2}{2\sigma^2} = \frac{(-0.5)\text{^}2-(0.5)\text{^}2}{2*1^2} = 0$. Seega on Bayesi-optimaalse klassifikaatori valem meie näite puhul $p^*(x) = \frac{1}{1+e^{-x}}$.

Leiame ka logistilise regressiooni mudeliga tõenäosusfunktsiooni. Leiame mõlema tõenäosusfunktsiooni väärtused punktidel -3-st 3-ni ja kujutame neid joonisel 2.



Joonis 1. Normaaljaotused ja nendest genereeritud andmepunktid



Joonis 2. Bayesi-optimaalne mudel ja treenitud mudel

Ülaltoodud jooniselt võib näha, et logistilise regressiooni mudeliga saadud tõenäosusfunktsioon on järsem kui Bayesi-optimaalne funktsioon, see tähendab, et positiivsetele punktidele antakse kõrgem tõenäosus positiivsesse klassi kuuluda kui seda teeks Bayesi-optimaalne funktsioon. Sama kehtib ka negatiivsete punktide kohta. See näitab, et logistilise regressiooni mudel on liiga enesekindel oma ennustustes ehk üle sobitunud.

Logistilise regressiooni asemel on ka võimalik kasutada normaaljaotuste hindamise meetodit – LDA ehk Linear Discriminant Analysis. Selle meetodi korral tahetakse leida, milliste parameetritega normaaljaotustesse andmepunktid kuuluvad ning sobitada neile Bayesi-optimaalne klassifikaator (Pohar, Blas, & Turk, 2004). Siiski annab logistiline regressioon praktikas paremaid tulemusi kui LDA, kuna LDA eeldab, et klassid on normaaljaotusega, logistiline regressioon otseselt seda eeldust klasside jaotuse kohta ei tee (Pohar, Blas, & Turk, 2004).

1.3 Logaritmiline kadu

Logaritmiline kadu on üks mudeli headuse mõõtmise viisidest. Logaritmiline kadu vaatab iga andmepunkti puhul selle asemel, kas ennustatud märgend ja tõeline märgend on võrdsed, seda, kui „kaugel“ ennustatud märgend asub õigest märgendist.

Logaritmiline kadu on defineeritud

$$c = -\frac{1}{\mathrm{n}}\sum_{\mathrm{i}=0}^{\mathrm{n}}\sum_{\mathrm{j}=0}^{\mathrm{m}} \mathrm{y_{ij}} log(\mathrm{p_{ij}})$$

kus n – andmepunktide arv, m - klasside arv, y_{ij} – klassi j i-nda andmepuntki tegelikud märgendid, p_{ij} – klassi j i-nda andmepunkti ennustatud märgendid.

Binaarse ehk kaheklassilisel andmestikul saab logaritmilise kao leida valemiga

$$c = -\frac{1}{\mathrm{n}}\sum_{1}^{\mathrm{n}}[y_i log(p_i) + (1 - y_i) log(1 - p_i)]$$

Oletame, et mudel annab ühe andmepunkti positiivsesse klassi kuulumise tõenäosuseks 0.49, kuid selle punkti tegelik märgend on 0. Kasutades mudeli hindamiseks täpsust eeldades, et klasside vaheline piir on $p = 0.5$, saame mudeli kohta teada vaid seda, et see ennustas sel andmepunktil õigesti. Kasutades aga logaritmilist kaofunktsiooni, näeme ära ka selle, et mudel on väga lähedal valesti ennustamisele. Seega eeldame, et mudelit on vaja veel optimeerida.

Selle näite põhjal arvutatud logaritmiline kadu $c = -log(0.51) \approx$ -1.708. Oletame aga, et teine mudel annab andmepunkti tõenäosuseks 0.51, aga punkti tegelik märgend on 0. Selle mudeli täpsus on küll 0, kuna ta ennustas ainsa punkti märgendi valesti, kuid logaritmiline kadu $c = -log(0.51) \approx$ -1.708 on eelmise mudeli omaga sama. See näitab, et tegelikult mõlemad mudelid on sama ebakindlad, mis siis, et ühe mudeli täpsus on 0 ja teisel 1.

1.4 Logistilise regressiooni kasutamine kalibreerimiseks

Logistilist regressiooni saab kasutada lineaarsete klassifikaatoritega saadud klassiennustuste teisendamiseks klassi kuulumise tõenäosuseks. Platt oma artiklis (Platt, 1999) tutvustab oma logistilise regressiooni alammeetodit tugivektor-masinate mudelil (*support vector machine* ehk *SVM)*, ning ka populaarne Pythoni masinõppe raamistik scikit-learn leiab SVM meetodi puhul tõenäosused just Platti meetodit kasutades (scikit-learn). Platti kirjeldatud meetod võtab SVM meetodist saadud ennustused ja treenib neil logistilise regressiooni mudeli. Sel juhul kasutame logistilist regressiooni märgendite ennustuste tõenäosusteks muutmisel

Logistilist regressiooni saab aga ka teiste tõenäosuslike klassifikaatori ennustuste peal kasutada. Näiteks naiivse Bayesi meetod on tihti andmestikul ülesobitunud, kuna see seab (reaalses elus enamjaolt mitte tõese) eelduse – kõik andmestiku tunnused on omavahel sõltumatud (Flach, 2012). Sel juhul võib logistiline regressioon anda meile paremini kalibreeritud ja vähem ülesobitunud mudeli.

1.5 Ülesobitamine

Ülesobitamine on ennustamisel mudeli liigne sõltuvus treeningandmetest, millest tulenevalt on treeningandmetel tulemused väga head aga testandmetel tunduvalt halvemad (Flach, 2012).

Siin ja edaspidi: tegusõna *ülesobituma* on passiivne ja *ülesobitama* on aktiivne. Näiteks masinõppija ülesobitas mudeli, mille tulemusena on mudel nüüd ülesobitunud.

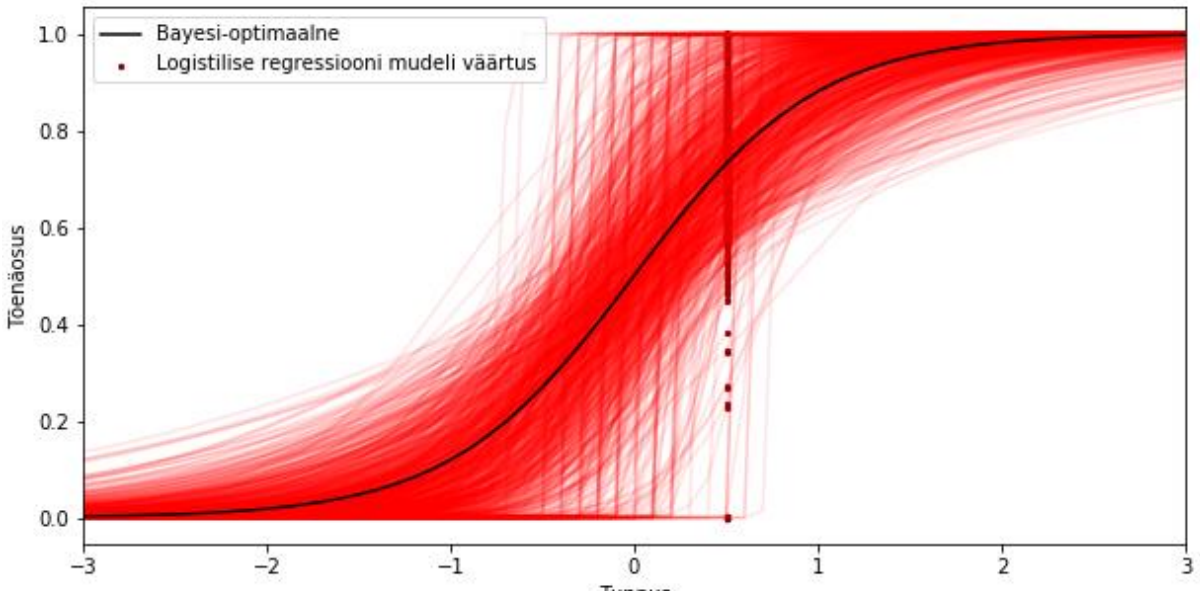
Ülesobitamine võib olla tingitud erinevatest asjaoludest:

1) liiga keeruka mudeli kasutamine. Näiteks tehisnärvivõrk, mis on hea mittelineaarsete sõltuvuste sobitamiseks, kuid on lihtsa lineaarse andmestiku peal eeldatavasti üle sobitunud (Hawkins, 2004). Ülesobitumise leevendamiseks tuleb valida lihtsam mudel.

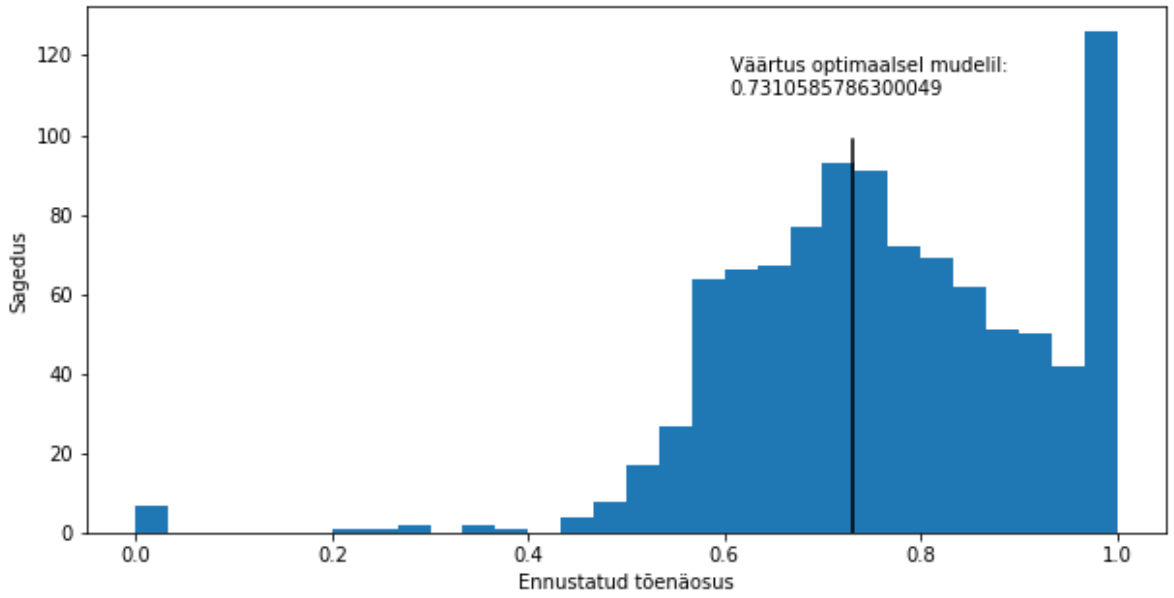
2) liiga paljude parameetrite peal treenimine. Kahe parameetriga seletataval andmestikul on üle sobitunud mudel, mille treenimiseks on kasutatud rohkem kui kaht parameetrit (Hawkins, 2004). Ülesobitumise leevendamiseks tuleb leida kõige tähtsamad parameetrid ning kasutada ainult neid.
3) mudeli kao minimeerimine treeningandmetel. Treeningandmetel minimaalse kaoga mudel ei pruugi olla parim testandmetel. Ülesobitumise leevendamiseks saab kasutada erinevaid meetodeid, millest hakkame rääkima järgnevates peatükkides.

Proovime leida, kas logistiline regressioon keskmiselt ülesobitub. Selle jaoks treenime 1000 logistilise regressiooni mudelit, kus treeningandmeteks on 10 positiivset ja 10 negatiivset andmepunkti, mis on genereeritud normaaljaotustest vastavalt keskväärtustega 0.5 ja -0.5 ja standardhälbega 1. Leiame punktis 0.5 nende mudelite keskmise tõenäosuse sellele punktile. Kujutame need mudelid ja punkti 0.5 väärtused joonisel 3.

Tekitame nende väärtuste järgi histogrammi, mis on näidatud joonisel 4. Leiame ka Bayesi-optimaalse mudeli ennustuse sellele punktile ning arvutame, mitmel korral on logistilise regressiooni mudeliga väärtus olnud väiksem kui optimaalne väärtus ja mitmel korral suurem. Andmetelt arvutame välja, et väärtus on olnud optimaalsest väiksem 432 korral ja suurem 568 korral, ning kõikide logistilise regressiooni ennustuste keskmine on 0.7616, mis on 0.0305 võrra suurem kui optimaalne väärtus selles punktis, milleks on $\frac{1}{1+e^{-0.5}}$ =0.73106. Selle järgi saab öelda, et keskmiselt on logistiline regressioon ülesobitunud – suurem osa mudelitest pakub andmepunktile suuremat tõenäosust kui optimaalne mudel.



Joonis 3. 1000 samade parameetritega treenitud logistilise regressiooni mudelit



Joonis 4. Histogramm punkti 0.5 väärtuste sageduste kohta

Kuna logistilise regressiooni meetodil on kombeks ülesobituda just ülaltoodud loetelu kolmanda punkti tõttu, on selle leevendamiseks parim viis kasutada andmete regulariseerimist või märgendite silumist.

1.6 Regulariseerimine

Regulariseerimine on meetod ülesobitumise leevendamiseks, muutes andmepunktide tunnuste tähtsuseid. Tavaliselt proovitakse need tähtsused ehk kaalud minimeerida (Flach, 2012). See vähendab tõenäosust, et mudel õpib ära liiga keerulise tõenäosusfunktsiooni. Enamjaolt kasutatakse masinõppes kahte populaarsemat regulariseerimismeetodit – L1-normi ja L2-normi regulariseerimine (Flach, 2012). Logistilise regressiooni mudel kasutab ülesobitamise leevendamiseks regulariseerimise asemel aga hoopis märgendite silumist.

1.7 Märgendite silumine

Kui peaks juhtuma, et andmestikus on mõned etteantud märgendid valed, võib see mudeli ennustusi päris palju mõjutada. Selle vältimiseks saab märgenditele lisada müra ehk ebakindlust, et antud märgend on tegelikult ka õige (Flach, 2012). Märgendite silumine muudab igal treeningandmepunktil tõelise märgendi väärtuse $y_{neg} = 0\ ja\ y_{pos} = 1$ vastavalt $y_{neg} = \varepsilon_{neg}$ ja $y_{\text{pos}} = \varepsilon_{pos}$, kus $\varepsilon_{neg}\ ja\ \varepsilon_{pos}$ – mingid väiksed konstandid ehk silumismäärad (Goodfellow, Bengio, & Courville, 2016).

Oletame, et meil on kaheklassiline andmestik, kus positiivsete klasside märgendiks on 1 ja negatiivsete omaks 0. Võtame $\varepsilon_{neg} = 0.02$ ja $\varepsilon_{pos} = 1 - \varepsilon_{neg}$. Sel juhul on nüüd positiivsete andmepunktide uueks väärtuseks $y_{\text{pos}} = 1 - 0.02 = 0.98$ ja $y_{neg} = 0.02$. Neid uusi märgendeid saab mõista nii, et me anname iga märgendi kohta väikse võimaluse, et tegelikult on see märgend hoopis vastupidine. Näiteks $y_{\text{pos}} = 0.98$ tähendab seda, et me oleme 98% kindlad, et tegemist on positiivse märgendiga, kuid me lubame endile ka 2% eksimisvõimalust. Platti skaleerimise meetod kasutab märgendite silumist, kus silumismäär on defineeritud Platti leitud valemiga.

1.8 Platti skaleerimine

Platti skaleerimine on logistilise regressiooni mudeli sobitamine lineaarse klassifikaatori mudelilt saadud ennustustele (Flach, 2012). Platti skaleerimise meetodis aga leitakse logistilise regressiooni valemi parameetrid valemiga

$$\min -\sum_{\text{i}} \text{y}_\text{i} \log(\text{p}_\text{i}) + (1 - \text{y}_\text{i}) \log(1 - \text{p}_\text{i})$$

kus y_i on klassi i märgendi uus väärtus pärast andmestikule märgendite silumise rakendamist.

Positiivsete klasside uueks märgendiks pakub Platt valemit (Platt, 1999)

$$\text{y}_{\text{pos}} = \frac{N_+ + 1}{N_+ + 2}$$

kus N_+ on positiivsete eksemplaride arv ja negatiivse klassi tõenäosuseks

$$\text{y}_{\text{neg}} = \frac{1}{\text{N}_- + 2}$$

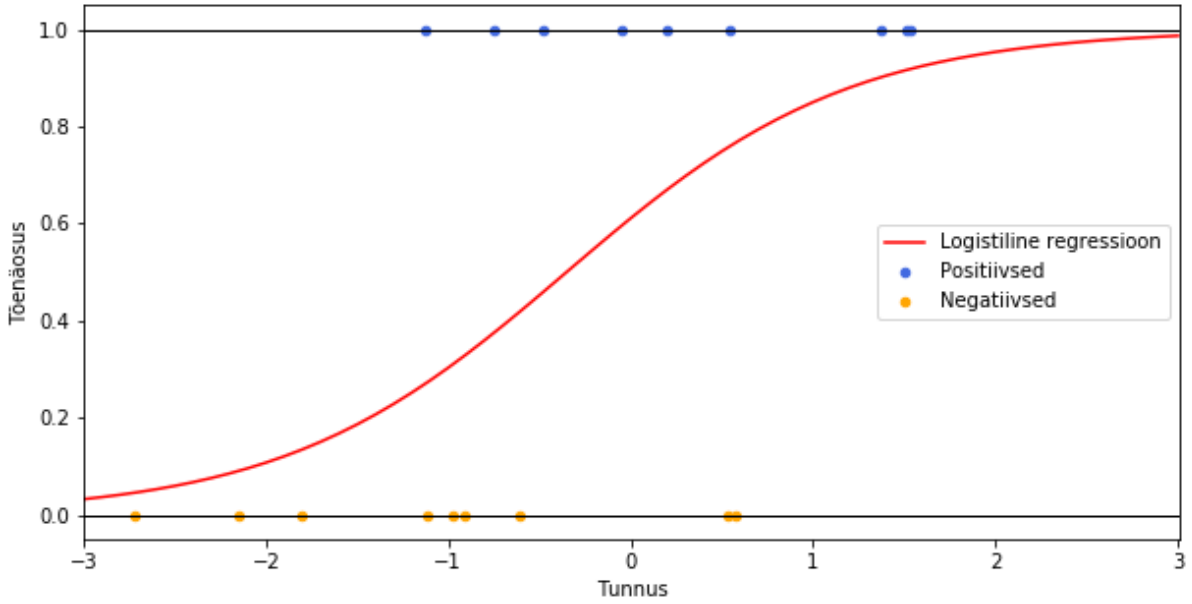
kus N_- on negatiivsete eksemplaride arv.

Andmete hulga lähenedes lõpmatusele läheneb positiivse klassi silutud tõenäosus ühele ja negatiivse klassi tõenäosus nullile (Flach, 2012).

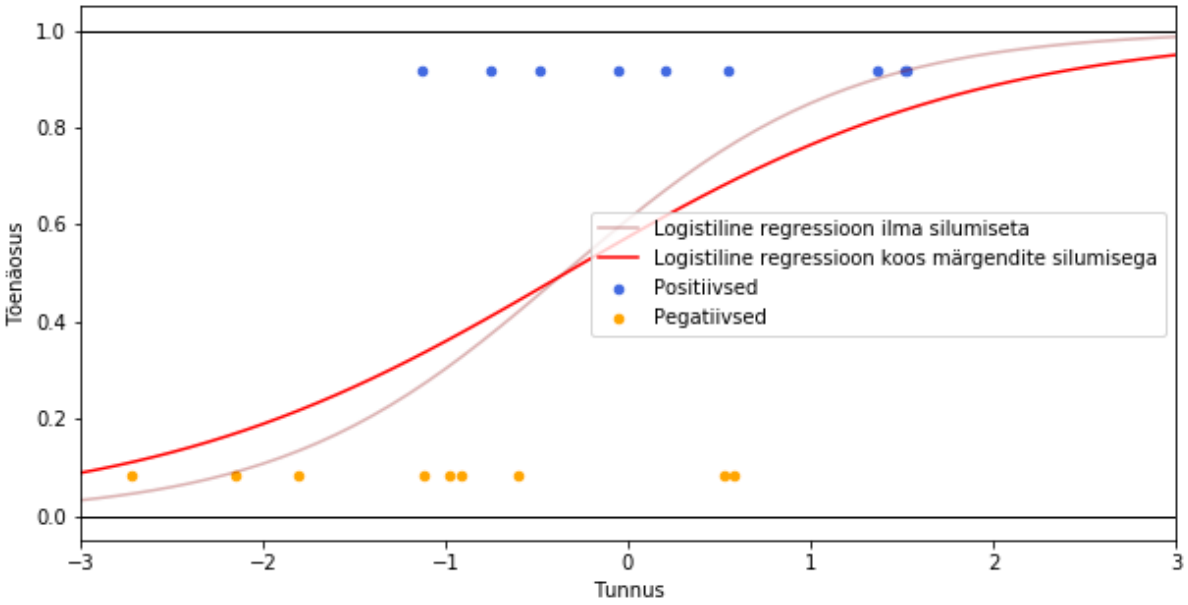
Genereerime taaskord 20 treeningandmepunkti, millest 10 positiivset ja 10 negatiivset punkti pärinevad normaaljaotustest keskväärtusega vastavalt 0.5 ja -0.5 ja standardhälbega 1.

Treenime ka treeningpunktidel logistilise regressiooni mudeli ning näitame selle tõenäosusfunktsiooni joonisel 5.

Nüüd kasutame märgendite silumist Platti leitud silumismäära valemiga. Seega positiivsete klasside tõenäosuseks on nüüd $\frac{11}{12} = \sim 0.917$ ja negatiivsete klasside tõenäosuseks $\frac{1}{12} = \sim 0.083$. Sobitame ka nüüd andmestikule logistilise regressiooni mudeli ning näitame seda joonisel 6.



Joonis 5. Andmestik ilma märgendite silumiseta



Joonis 6. Andmestik märgendite silumisega Platti valitud silumismääraga

Võrreldes mõlemat joonist on näha, et märgendite silumist kasutamata on logistiline funktsioon järsem, mis viitab sellele, et logistilise regressiooni mudel vähemalt sel andmestikul ilma märgendite silumiseta on enesekindlam kui märgendite silumisega, mis viitab ülesobitamisele.

2. Platti skaleerimise uurimine

2.1 Töö eesmärk

Töö eesmärk on tehisandmetel vaadelda, kas Platti skaleerimise meetodit kasutades leevendame ülesobitumist treeningandmetel ning uurida, kas Platti skaleerimise meetodis silumismäära jaoks valitud valem on parim või saab leida mingi paremini sobiva väärtuse silumismäärale.

2.2 Ülevaade meetoditest ja lähenemisest

2.2.1 Kasutatud tarkvara

Eksperimentide koodi kirjutamiseks ja jooksutamiseks kasutan programmeerimiskeelt Python versioon 3.7. Samuti kasutan Pythoni populaarset masinõppe raamistikku scikit-learn, kuna seal on juba defineeritud mõned vajalikud meetodid, näiteks etteantud keskväärtuse ja standardhälbega normaaljaotusest punktide genereerimine ning õigete märgendite ja ennustuste põhjal logistilise kao arvutamine.

Logistilise regressiooni mudeli loomiseks ja treenimiseks koos märgendite silumisega kasutan Mari-Liis Allikivi poolt selleks tööks antud lähtekoodi keeles Python.

2.2.2 Andmete genereerimine

Eksperimentide läbi viimiseks otsustasime kasutada sünteetilisi andmeid, mis on genereeritud normaaljaotustest, mille parameetrid anname ise ette. Sel juhul saab normaaljaotuste keskväärtuse ja standardhälbe abil välja arvutada Bayesi-optimaalse tõenäosusliku klassifikaatori, mida saab võrrelda treenitud mudelitega. Samuti saab niimoodi palju erinevaid katseid läbi viia: saab muuta andmestiku suurust ja klasside jaotust ning saab muuta normaaljaotuste, kust andmed on genereeritud, parameetreid.

Treeningandmed genereerin kahest normaaljaotusest: positiivse märgendiga punktide jaoks kasutan etteantud keskväärtust, negatiivsete punktide jaoks kasutan etteantud keskväärtuse vastandarvu. Standardhälve σ=1 on kogu aeg sama mõlemal jaotusel.

$$x_{pos} \in N(\mu, 1), \;\; x_{neg} \in N(-\mu, 1)$$

Positiivse klassi suuruse defineerin muutujas *suurus*, negatiivse klassi hulk sõltub ka klasside suuruste suhtest, mis on defineeritud muutujas *suhe*.

$$\mathrm{N}_{pos} = suurus, \;\; \mathrm{N}_{neg} = suurus * suhe$$

Katseid teen erinevate keskväärtuste, klasside suuruste ja suhetega: keskväärtused on μ = 0.5, 1.0, 1.5, 2.0, suuruste hulk on defineeritud $suurus = 10, 15, 20, \dots, 200$ ja suhted $suhe = 1.0, 1.5, 2.0$. Otsustasime katsetada erinevaid silumismäärasid vahemikust $\varepsilon = 0.002, 0.004, \dots, 0.2$. Seejuures on $\varepsilon_{pos} = \varepsilon_{neg}$, käosolevas töös ei uuri olukorda, kus $\varepsilon_{pos} \neq \varepsilon_{neg}$.

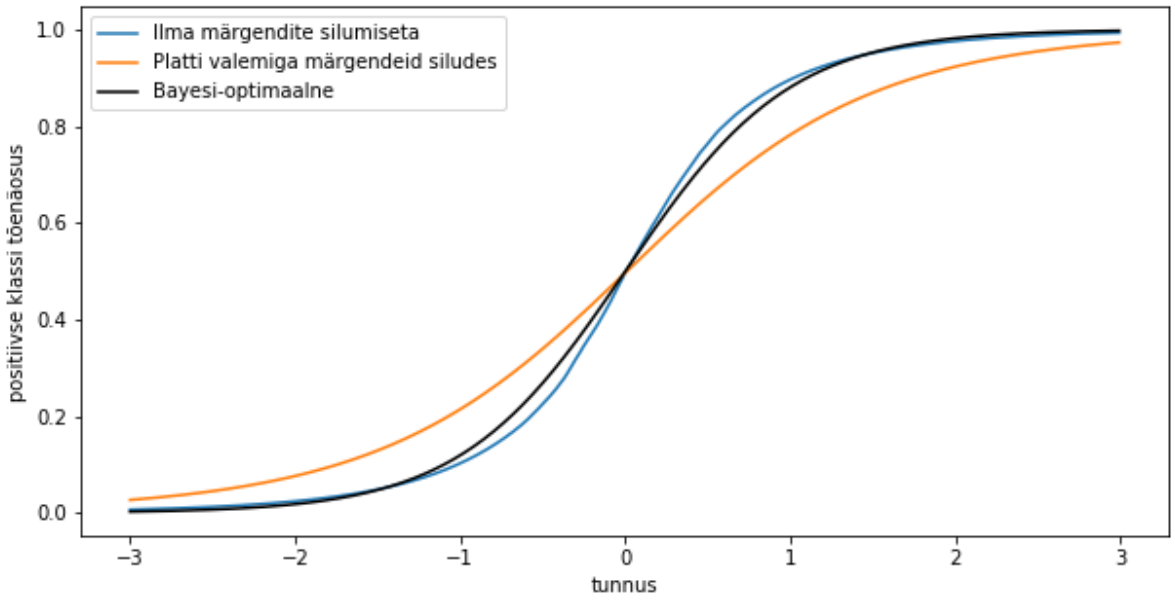
Mudelite testimiseks genereerin treenimisandmetega sama keskväärtustega normaaljaotustest 10 000 positiivset ja 10 000 negatiivset. See andmehulk jääb samaks igal iteratsioonil ja iga treeninghulga suuruse puhul.

2.3 Tulemuste analüüs

2.3.1 Märgendite silumise vajadus Platti skaleerimist kasutades

Esimese asjana otsustati uurida, kas märgendite silumine on Platti skaleerimise meetodi juures üldse vajalik, võrreldes silumata andmestikuga treenimisel saadud logistilist funktsiooni Platti valitud silumismääraga silutud andmestikuga saadud funktsiooniga. Samuti saab võrrelda logaritmilist kadu treening- ja testandmestikul, et näha, kas kumbki mudel on ülesobitunud.

Võrdluse jaoks genereerin 10 positiivset ja 10 negatiivset treeningandmepunkti normaaljaotustest keskväärtustega vastavalt 0.5 ja -0.5. Nende normaaljaotuste järgi saan peatükis 1.2 defineeritud valemi järgi arvutada Bayesi-optimaalse logistilise funktsiooni. Samuti treenin igal iteratsioonil nii ilma märgendite silumiseta kui silumisega mudelid. Lõpuks leian iga punkti $x \epsilon [-3,3]$ keskmise ennustuse kõigi silumiseta ja silumisega mudelite peale iga keskväärtuse kohta. Kõikide punktide keskmised on näidatud joonisel 7.



Joonis 7. Silumiseta ja silumisega mudelite keskmised tõenäosusfunktsioonid andmestikul suurusega 10

Jooniselt 7 on näha, et Platti skaleerimisega treenitud mudeli tõenäosusfunktsioon on palju laugem kui märgendite silumist mittekasutav mudel. Sel näitel on ilma silumata mudel ainult pisut enesekindlam kui Bayesi-optimaalne mudel, Platti meetodiga saadakse aga liiga lame funktsioon. Siit võib järeldada, et mingisugustel andmestikel teeb Platti skaleerimine liiga palju märgendite silumist. Sellest aga tekib küsimus, kas leidub mõni muu silumismäär, mis oleks Platti meetodist parem.

2.3.2 Platti valitud silumismäära võrdlemine teiste silumismääradega

Platti skaleerimise meetodi silumismäärast parema silumismäära leidmiseks leian iga vaadeldava silumismäära kaod mitu korda ning leian lõpuks nende keskmised väärtused. Iga ε kao leidmiseks ei piisa vaid ühest iteratsioonist, kuna treenimiseks kasutatavate punktide arv on suhteliselt väike ning juhuslik ja nende genereerimisel ja treenimisel tekkiv mudel võib iga kord olla väga erinev.

Kõigepealt leian iga keskväärtuse ja suhte variatsiooniga neile vähima keskmise logaritmilise kaoga silumismäär. Selle jaoks leiame logaritmilise kao kõigi suuruste kohta igal iteratsioonil kolmel juhul:

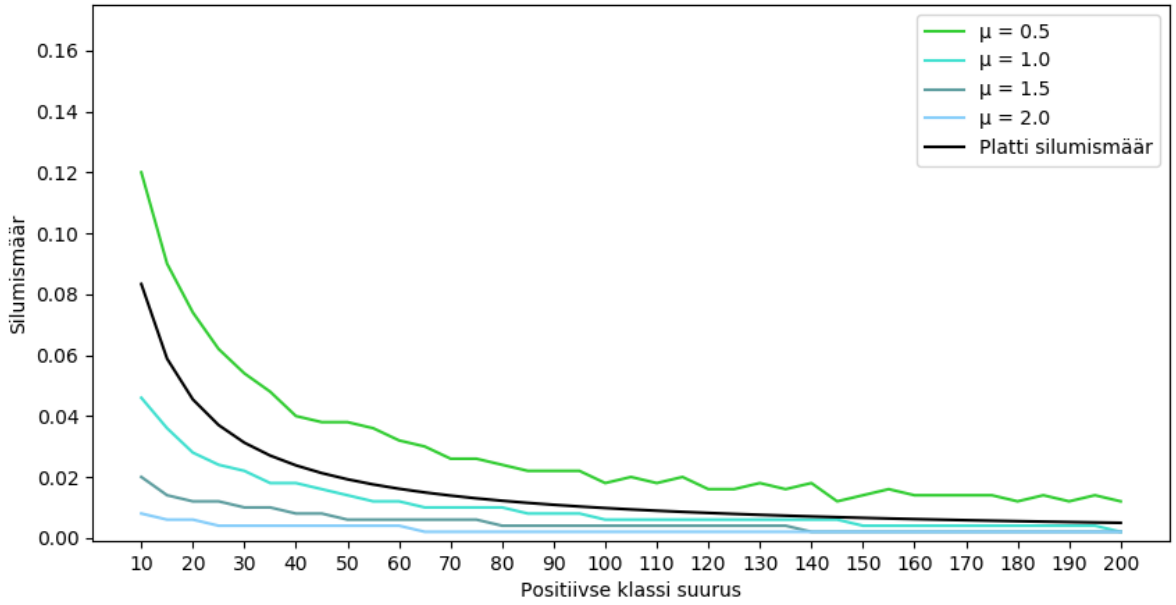
a) Platti skaleerimist kasutamata,
b) kasutades Platti skaleerimist Platti valitud silumismäärade valemiga,
c) kasutades Platti skaleerimist iga testitava silumismääraga.

Juhuslikkuse minimeerimiseks jooksutan kadude arvutamist 1000 korda, kuna siis võib eeldada, et silumismäärade keskmised kaod on koondunud mingisse punkti, mis on keskmiselt kõige iseloomulikum sellele ε-le ja andmestiku suurusele. Iteratsioonide lõppedes arvutan igale klassi suurusele vastava keskmiselt parima ehk minimaalse kaoga silumismäära. Algoritm 1 näitab pseudoalgoritmina ülalkirjeldatut.

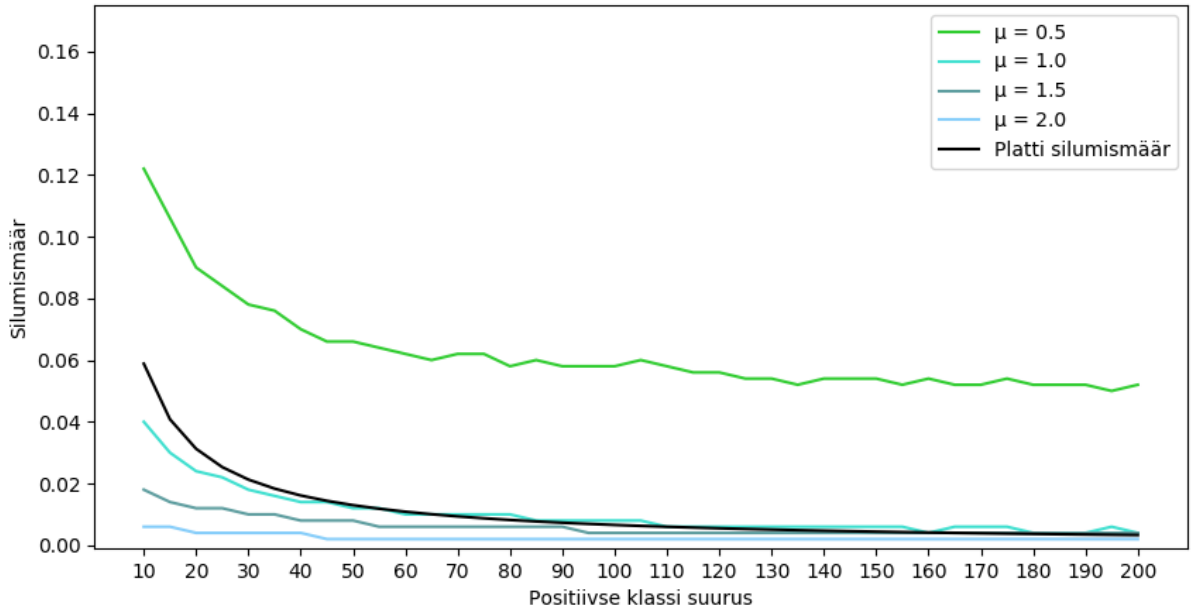
```
iga vaadeldava kombinatsiooni (γ, n_pos, n_neg) korral:
   genereeritakse testandmestik:
        • 10000 korda x_pos^test ∈ N(γ, 1)
        • 10000 korda x_neg^test ∈ N(-γ, 1)
   1000 korda:
        • genereeritakse treeningandmestik:
            ▪ n_pos korda x_pos^train ∈ N(γ, 1)
            ▪ n_neg korda x_neg^train ∈ N(-γ, 1)
        • iga vaadeldava silumismäära ε korral:
            ▪ treenitakse logistilise regressiooni mudel
            ▪ leitakse treenitud mudeli kadu
   leitakse iga silumismäära ε korral keskmine väärtus kadu
   leitakse ε, mille korral on keskmine kadu kõige väiksem
```

Algoritm 1. Parima silumismäära leidmine

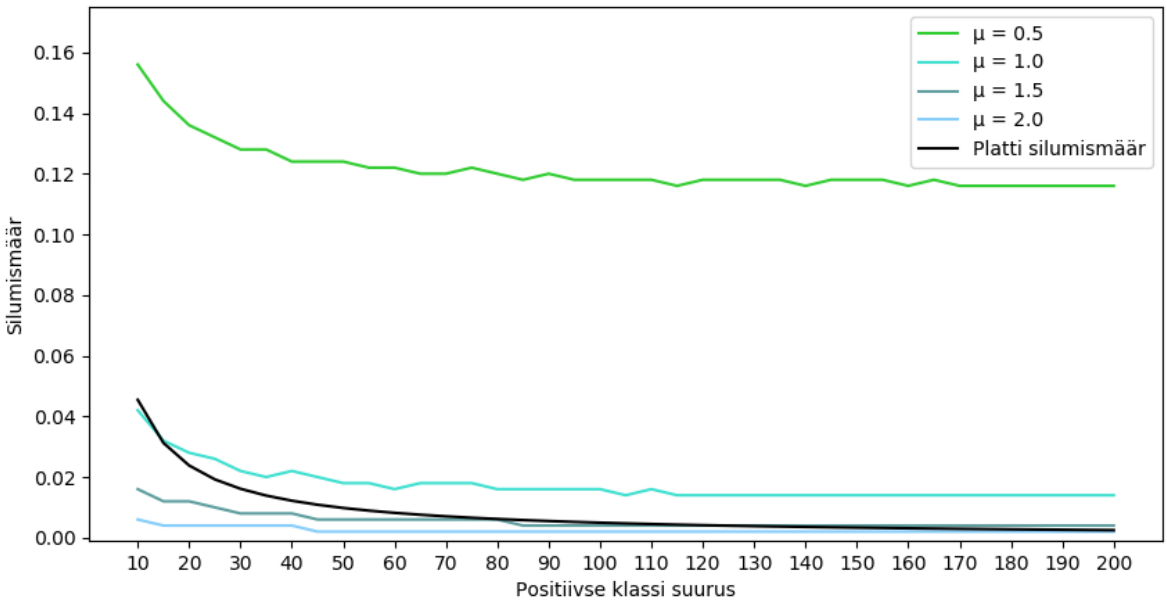
Joonistel 9-11 on näidatud iga klassi suurustega kõikidele keskväärtustele ja suurustele vastavad parimad silumismäärad olenevalt klassi suuruste suhtest. Kõigilt joonistelt on näha, et Platti valitud silumismäära valem pole ühegi valitud keskväärtuse ega suuruse suhte juures optimaalne, sest ühegi vaadeldava andmestiku optimaalsed silumismäärad ei kattu Platti valemiga saadu määradega. See näitab, et Platti valem ei ole parim võimalik.



Joonis 2. Parimad silumismäärad klasside suhtega 1.0

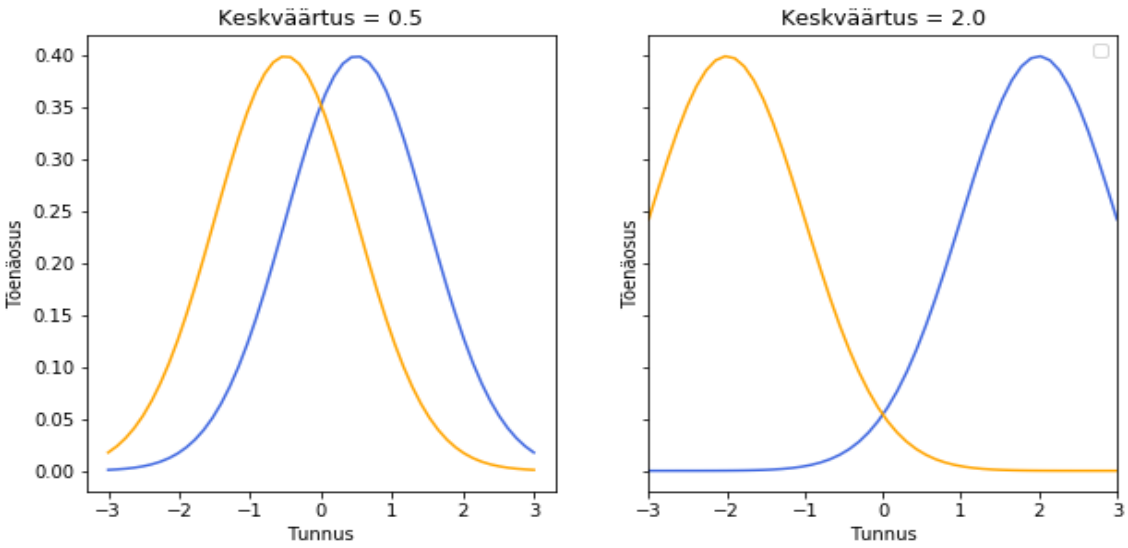


Joonis 10. Parimad silumismäärad klasside suhtega 1.5



Joonis 3. Parimad silumismäärad klasside suhtega 2.0

Kõigilt joonistelt on näha, et silumismäärad varieeruvad erinevate keskväärtuste puhul päris palju. Näiteks keskväärtus 0.5 ja suuruste suhte 1.0 parimad silumismäärad koonduvad umbes 0.015 juurde, suhtega 1.5 koonduvad juba 0.06 juurde ja suhtega 2.0 koonduvad umbes 0.120 juurde. See näitab, et mida lähemal ja segunenud on klassid, seda rohkem on vaja märgendeid siluda ja seda suurem on silumismäär. See on ka loogiline, kuna nii lähedal olevatel normaaljaotustel on paljudel punktidel võimalus olla pärit nii ühest jaotusest kui ka teisest. Joonisel 12 on näha, et normaalajaotustel keskväärtustega $\mu = 0.5$ ja $\mu = -0.5$ on ala, kus asetsev punkt võib pärineda mõlemast jaotusest, suurem kui keskväärtustega $\mu = 2$ ja $\mu = -2$. Seega on ka vaja anda suurem tõenäosus punkti vastandklassi kuulumisele.

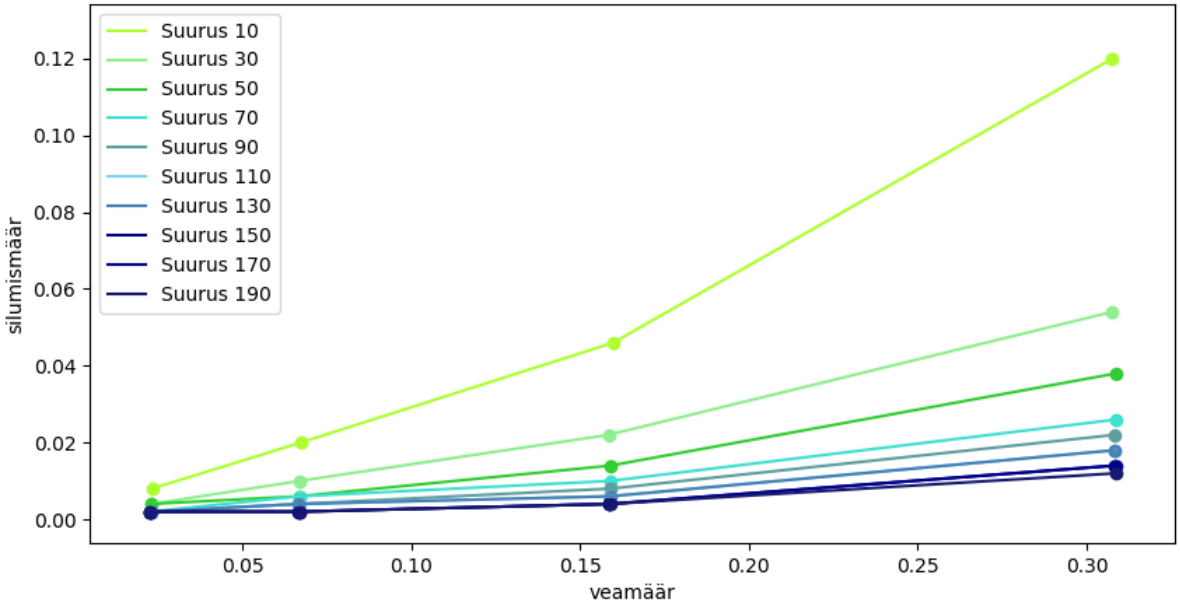


Joonis 12. Erineva keskväärtusega normaaljaotused ja nende lõikumisalad

Võrreldes jooniseid 9-11, on ka näha, et optimaalne silumismäär sõltub iga keskväärtuse puhul lisaks andmestiku suurusele ka andmestiku suhtest. Käesolevas töös vaadati ainult juhte, kus negatiivne klass on suurem kui positiivne, kuna Platti valitud meetod kasutab silumismäära valikul just negatiivse klassi suurust. Ajapuudusel ei jõutud vaadelda juhte, kus positiivse klassi suurus on suurem kui negatiivse oma. Seepärast saab joonistelt teada selle, et negatiivse klassi suuruse suurenemisega kasvab ka optimaalse silumismäära väärtus. Näiteks keskväärtus 0.5 puhul koondub suhtega 1.0 optimaalne silumismäär väärtusesse 0.012, suhtega 1.5 juba väärtusesse 0.052 ja suhtega 2.0 väärtusesse 0.116. Seega suureneb sel keskväärtusel suhet kahekordistades optimaalne silumismäär peaaegu 10 korda. Keskväärtuse 1.0 korral muutub see 7 korda, keskväärtuse 1.5 korral 2 korda. Keskväärtuse 2.0 korral jääb see samaks. Seega sõltub silumismäära väärtus lisaks andmestiku suurusest ja andmestiku jaotuste keskväärtustest ka negatiivse ja positiivse klassi suhtest.

Tavaliselt reaalsetel ehk mitte-tehislikel andmestikel me ei tea midagi nende jaotuste kohta, kust punktid pärinevad. Seepärast on meil vaja leida võrdlemiseks mingi mõõde, mida saab leida nii meie tehisandmetel kui tulevikus reaalsetel andmete. Üheks selliseks mõõteks saab olla veamäär, kuna seda saab mõõta ka teadmata jaotustega andmestikel. Meie saame oma normaaljaotustel välja arvutada treeningandmestiku kaomäära, leides, mitu tegelikult positiivset punkti jäävad kahe normaaljaotuse lõikumiskohast vasakule.

Leiame veamäärad iga normaaljaotuse kohta. Suhteks võtame suhe = 1.0. Kanname tulemused joonisele 13, kus vaadeldavad klasside suurused on eristatud värvidega. Vaatleme ainult osa enne defineeritud suurustest, kuna muidu oleks joonis liiga kirju ja eristamatu.



Joonis 12. Veamäär ja optimaalne silumismäär vaadeldavate suuruste korral

Jooniselt 12 on võimalik välja lugeda silumismäära suuruse sõltuvus veamäära suurusest – veamäära kasvades tõuseb ka optimaalne silumismäär. Kuna veamäär sõltub sellest, kui segunenud klassid on, mis sõltub omakorda sellest, kui lähedal on andmete genereerimiseks kasutatud jaotused, siis on see tulemus loogiline, kuna sõltuvuse jaotuste keskväärtuste ja silumismäärade vahel juba leidsime. Selle joonise järgi saaksime juba proovida leida mingisuguse veamäära ja klasside suurusega defineeritud valemi, millega saab leida optimaalset silumismäära ka siin töös defineerimata keskväärtustega normaalajotustest genereeritud andmestikel ja reaalsetel andmetel. Valemi sobitamine jäi töö raamidest välja.

Kokkuvõte

Antud töös kirjeldati põhjalikult tõenäosuslike klassifikaatorite kalibreerimist logistilise kalibreerimise ehk Platti skaleerimise meetodiga. Töö käigus viidi läbi eksperimendid otsimaks logistilise kalibreerimise meetodi silumismäära väärtusest paremat väärtust.

Käesoleva töö eesmärgiks oli leida, kas logistilise kalibreerimise meetodis defineeritud silumismäär annab parimaid tulemusi. Katsete tulemustest saab näha, et mingitel andmestikel annab logistiliselt kalibreeritud logistilise regressiooni mudel treeningandmetel halvemaid tulemusi kui kalibreerimata mudel. Samuti sai eksperimentidest näha, et ühelgi vaadeldaval andmestikul ei olnud logistilise kalibreerimise silumismäär optimaalne. Seega võib arvata, et tegelikult leidub Platti skaleerimise meetodi silumismäära valemist mõni parem ja väiksema keskmise kaoga valem. Samuti oli joonistelt näha, et optimaalne silumismäär sõltus lisaks andmehulga suurusest, mis on sisse arvestatud ka Platti valitud silumismäära valemis, ka mudeli veamäärast.

Töö edasiarendusena oleks võimalik proovida leida uus valem logistilise kalibratsiooni silumismäära arvutamiseks. Selle jaoks tuleb töös kirjeldatud katseid läbi viia veel rohkemate andmestikega ning varieerida rohkem katsete parameetreid. Valemi leidmiseks saab näiteks kasutada masinõpet, mis leiaks optimaalse silumismäära jaoks vajalike tunnuste kaalud.

Viidatud kirjandus

Flach, P. (2012). *Machine Learning: The Art and Science of Algorithms That Make Sense of Data.* Cambridge University Press.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* MIT press.

Hawkins, D. M. (2004). The Problem of Overfitting. *Journal of chemical information and computer sciences, 44*(1), 1-12.

Kull, M., Silva Filho, T. M., & Flach, P. (2017). Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electron. J. Statist, 11*(2), 5052-5080.

Langley, P. (2011). The changing science of machine learning. *Machine Learning, 82*(3), 275-279.

Machine Learning: What it is and why it matters. (kuupäev puudub). Kasutamise kuupäev: 10. Mai 2019. a., allikas Analytics, Business Intelligence and Data Management: https://www.sas.com/en_us/insights/analytics/machine-learning.html

Platt, J. (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Advances in large margin classifiers, 10*(3), 61-74.

Pohar, M., Blas, M., & Turk, S. (2004). "Comparison of logistic regression and linear discriminant analysis: a simulation study. *Metodološki zvezki, 1*(1), 143.

scikit-learn. (kuupäev puudub). Kasutamise kuupäev: 8. Mai 2019. a., allikas https://scikit-learn.org/stable/modules/svm.html#scores-and-probabilities

Tesla Inc. (kuupäev puudub). *Autopilot*. Kasutamise kuupäev: 10. Mai 2019. a., allikas Electric Cars, Solar Panels & Clean Energy Storage: https://www.tesla.com/en_GB/autopilot

Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates. *ICML*.

Lisad

1. Repositoorium

Töös eksperimentide jooksutamiseks ja graafikute joonistamiseks kirjutatud lähtekood on avalikult üleval aadressil

https://github.com/liinaanette/Platti-skaleerimise-uurimine

2. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Liina Anette Pärtel**,

1) annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

   Märgendite silumine klassifikaatorite logistilisel kalibreerimisel,

   mille juhendaja on Meelis Kull,

   reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2) Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3) Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4) Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Liina Anette Pärtel

10.05.2019