TARTU ÜLIKOOL

MATEMAATIKA-INFORMAATIKATEADUSKOND

Matemaatilise statistika instituut

Matemaatilise statistika eriala

Jaak Sõnajalg

Tutvumine AdaBoostiga

Bakalaureusetöö (6 EAP)

Juhendaja: Jüri Lember

Tartu 2013

Sisukord

Sissejuhatus **4**

1 Klassifitseerimisteooria **6**

1.1 Probleemipüstitus 6

1.2 Kaofunktsioon 7

1.3 Risk 8

1.4 Bayesi klassifitseerija 9

2 Vapnik-Tšervonenkise teooria **13**

2.1 Andmetest õppimine 13

2.2 Empiirilise riski minimeerimine 14

2.3 Vapnik-Tšervonenkise dimensioon 17

2.4 Riskihinnangud Vapnik-Tšervonenkise dimensiooni kaudu 20

3 AdaBoost **24**

3.1 ϕ-risk 24

3.2 Algoritmi kirjeldus 30

3.3 Eksponentsiaalne kadu ja AdaBoost 33
3.4 AdaBoosti treeningviga 37
3.5 Marginaalilävega treeningviga 39
3.6 Hinnang AdaBoosti riskile 42

4 Katsed AdaBoostiga **45**

4.1 Andmete genereerimine, Bayesi klassifitseerija ja Bayesi risk . . . 46
4.2 AdaBoosti rakendamine 48
4.3 Treeningvea hinnangud 50
4.4 Marginaalilävega treeningviga ja selle hinnang 51
4.5 Riski hinnang 54
4.6 AdaBoost teiste rusikareeglite korral 56

Kokkuvõte (inglise keeles) **60**

Kirjandus **61**

Sissejuhatus

AdaBoost on *boosting*-meetodite perre kuuluv tehisõppe algoritm. *Boosting*-meetodite nimetus tuleneb nende omadusest parandada või võimendada (ingl *boost*) suhteliselt lihtsate klassifitseerijate klassifitseerimisomadusi. *Boosting*-algoritmid kasutavad kergesti rakendatavaid ja juhuslikust arvamisest vaid veidi paremate omadustega klassifitseerijaid ja konstrueerivad neist ühe, väga hea klassifitseerija. AdaBoosti kirjeldasid esmakordselt Yoav Freund ja Robert Schapire 1996. aastal [5].

Töö esimeses peatükis tutvustatakse mitmeid klassifitseerimisteooria põhimõisteid nagu näiteks kaofunktsioon ja risk. Järgmine peatükk tutvustab andmetest õppimise põhimõtet. Kolmas peatükk keskendub AdaBoostile. Lisaks algoritmi kirjeldusele esitatakse seal mitmed hinnangud AdaBoosti väljundi riskile. Neljandas peatükis kirjeldatakse autori poolt läbi viidud katseid AdaBoostiga.

Allikmaterjalidele on töös viidatud nurksulgude abil. Nurksulgudes on number, mis vastab allikmaterjali järjekorranumbrile töö lõpus esitatud kasutatud kirjanduse loetelus. Mõnele konkreetsele faktile viitamise korral on nurksulgudes ka leheküljenumber või -numbrid, kust fakti viidatud allikmaterjalist leiab. Järelduste tõestused ja näited, mille juures viited allikmaterjalidele puuduvad, on autor lahendanud iseseisvalt.

Neljandas peatükis kirjeldatud simulatsioonide tarbeks tarvilike programmide

koostamiseks on kasutatud statistikatarkvara R. Töös esitatud jooniste kujundamisel on kasutatud pilditöötlustarkvara GIMP 2.

Peatükk 1

Klassifitseerimisteooria

Esimeses peatükis tutvustatakse klassifitseerimisteooria põhimõisteid ja nendevahelisi seoseid.

1.1 Probleemipüstitus

Klassifitseerimisprobleemiks nimetame olukorda, kus soovime tundmatut objekti liigitada ühte etteantud klassidest [9, lk 6]. Klassifitseerimisel on eesmärgiks liigitada tundmatu objekt klassi, mille esindajate omadustega tundmatu objekti teadaolevad omadused enim sarnanevad.

Näide 1.1. Klassifitseerimisprobleemi näitena võib vaadelda taime liigi kindlakstegemist taimede välimääraja abil. Siin kasutame leitud taimele omaseid väliseid tunnuseid nagu värv, mõõtmed ja lehtede kuju, et taime liigiline kuuluvus kindlaks teha. Välimäärajat osavalt kasutades peaksime teada saama, millisele taimeliigile on leitud taime tunnused omased.

Esitame klassifitseeritava objekti kirjelduse tunnusvektorina x. Tunnusvektorit x

vaatleme kui juhusliku suuruse X realisatsiooni. Tähistame juhusliku suuruse X võimalike väärtuste hulga $\mathcal{X}$. Tunnusvektorisse kuuluvad tunnused võivad olla nii kvalitatiivsed kui kvantitatiivsed [8, lk 10].

Näide 1.2. Kui soovime teada saada, kas meie ees olev selgroogne on imetaja, lind, kala, sisalik või kahepaikne, võivad tunnusvektori x moodustada erinevad selgroogseid iseloomustavad omadused nagu näiteks kehatemperatuur, lennuvõime olemasolu ja paljunemisviis. Selgroogsete klassifitseerija seab igale sellisele tunnusvektorile vastavusse klassi hulgast {„imetaja", „lind", „kala", „sisalik", „kahepaikne"}.

Tähistame objekti klassikuuluvuse y. Käsitleme klassikuuluvust y juhusliku suuruse Y realisatsioonina. Olgu $\mathcal{Y}$ juhusliku suuruse Y võimalike väärtuste hulk. Eeldame, et $\mathcal{Y}$ on lõplik hulk, seega võime igale hulka $\mathcal{Y}$ kuuluvale klassile vastavusse seada ühe täisarvu. Antud töös vaatleme ainult kaheklassilist klassifitseerimisprobleemi. Nagu hiljem AdaBoosti algoritmiga tutvudes näeme, on AdaBoosti puhul klasside tähistamiseks loomulik kasutada täisarve -1 ja 1, st

$$\mathcal{Y} \coloneqq \{-1, 1\}.$$

Klassifitseerijat võime vaadelda kui operaatorit, mille argumendiks on objekti tunnusvektor ja väärtuseks üks arv hulgast $\mathcal{Y}$. Klassifitseerija matemaatiline kuju on seega

$$g : \mathcal{X} \mapsto \mathcal{Y}.$$

1.2 Kaofunktsioon

Klassifitseerimisveaks nimetame seda, kui klassifitseerija g seab objektile vastavusse vale klassi, st $g(X) \neq Y$. Klassifitseerija g poolt tehtud otsuste kvaliteedile

hinnangu andmiseks tutvume kaofunktsiooni mõistega. Kaofunktsioon

$$L : \mathcal{Y} \times \mathcal{Y} \mapsto [0, \infty)$$

seab klasside paarile (i, j) vastavusse mittenegatiivse reaalarvu. Seda reaalarvu tõlgendame kahjuna, mille toob endaga kaasa klassi i kuuluva objekti pidamine klassi j esindajaks [9, lk 9].

Kaofunktsiooni kasutatakse otsuse täpsuse mõõtmiseks. Kui meie eesmärgiks on juhusliku suuruse Y väärtuse kindlakstegemine, siis on kaofunktsiooni sisendiks tõene väärtus Y ja klassifitseerija g poolt tunnusvektorit X kasutades tehtud otsus $g(X)$. Kaofunktsiooni väljundiks on mittenegatiivne reaalarv, mis peegeldab klassifitseerija g poolt tehtud otsuse kvaliteeti.

On loomulik eeldada, et $L(i, i) = 0$, korrektse klassifitseerimisega mingisugust kahju ei kaasne. Seda eeldust rahuldab klassifitseerimisülesannete korral kasutatav sümmeetriline ehk 0-1 kaofunktsioon [9, lk 13]

$$L(Y, g(X)) = I_{Y \neq g(X)} = \begin{cases} 0 & \text{kui } Y = g(X) \\ 1 & \text{mujal.} \end{cases} \tag{1.1}$$

Sümmeetriline kaofunktsioon seab kõigile klassifitseerimisvigadele vastavusse võrdse kahju.

1.3 Risk

Klassifitseerija g sisendit tunnusvektorit x võime käsitleda d-dimensionaalse juhusliku vektorina. Fikseeritud tunnusvektori korral on ka objekti klass juhuslik, võrdsete tunnusvektoritega objektid võivad olla erinevatest klassidest. Näiteks võib 173-sentimeetrine ja 63-kilogrammine inimene olla nii mees kui ka naine. Juhusliku vektori (X, Y) jaotust ja kaofunktsiooni mõistet kasutades saame defineerida kriteeriumi hindamaks klassifitseerija headust.

Definitsioon 1.1 (vt [10, lk 11])**.** *Klassifitseerija g risk on keskmine kahju üle tunnusvektori ja klasside:*

$$R(g) \coloneqq E_{X,Y} L\big(Y, g(X)\big). \tag{1.2}$$

Risk näitab, kui hea on klassifitseerija g kaofunktsiooni L suhtes. Nimetus „risk" viitab viisile, kuidas seda matemaatilist suurust tõlgendame. Me soovime, et klassifitseerija g risk oleks võimalikult väike. Iga klassifitseerija g korral on riskifunktsiooni väljundiks reaalarv. Reaalarvude hulk on lineaarselt järjestatud hulk, seetõttu saame kõiki klassifitseerijaid riskifunktsiooni kasutades võrrelda. Mitmest klassifitseerijast parima välja valimiseks leiame, millise klassifitseerija risk on juhusliku vektori (X, Y) jaotuse korral väikseim.

Diskreetse juhusliku suuruse keskväärtuse definitsiooni kasutades saame leida klassifitseerija g riski $R(g)$ sümmeetrilise kaofunktsiooni (1.1) korral:

$$\begin{aligned} R_{0-1}(g) &= E(I_{Y \neq g(X)}) = \\ &= 0 \cdot P\big(Y = g(X)\big) + 1 \cdot P\big(y \neq g(X)\big) = P\big(Y \neq g(X)\big). \end{aligned} \tag{1.3}$$

Sümmeetrilise kaofunktsiooni korral on klassifitseerija g risk võrdne klassifitseerimisvea tegemise tõenäosusega.

1.4 Bayesi klassifitseerija

Vaatleme klassifitseerija g riski. Tähistame tõenäosuse, et objekt tunnusvektoriga $x \in \mathcal{X}$ kuulub klassi 1:

$$\eta(x) \coloneqq P(Y = 1 | X = x). \tag{1.4}$$

Kõik objektid kuuluvad täpselt ühte klassi. Kuna klasse on kaks, avaldub tõenäosus, et objekt tunnusvektoriga x kuulub klassi -1:

$$P(Y = -1|X = x) = 1 - \eta(x).$$

Definitsioon 1.2 (vt [10, lk 11])**.** *Olgu tunnusvektor $x \in \mathcal{X}$ fikseeritud. Klassifitseerija g tinglik risk tunnusvektori x korral on keskmine kahju, mis objekti, mille tunnusvektor on x, klassifitseerimisega kaasneb:*

$$R\big(g(x)|x\big) \coloneqq E_{Y|X}\big[L\big(Y, g(x)\big)|X = x\big]. \tag{1.5}$$

Klassifitseerija g riski $R(g)$ saame leida, kui keskmistame tingliku riski üle tunnusvektori x jaotuse:

$$\begin{aligned} R(g) &= E_{X,Y} L\big(Y, g(X)\big) = \\ &= E_X\big\{E_{Y|X}\big[L\big(Y, g(X)\big)|X\big]\big\} = \\ &= E_X R(g(X)|X). \end{aligned} \tag{1.6}$$

Defineerime klassifitseerija, mille risk on nii väike kui võimalik.

Definitsioon 1.3 (vt [10, lk 12])**.** *Klassifitseerijat, mis minimeerib tingliku riski (1.5) iga tunnusvektori $x \in \mathcal{X}$ korral, nimetame Bayesi klassifitseerijaks. Tähistame Bayesi klassifitseerija g^*.*

Näiteme, et nii defineeritud klassifitseerija minimeerib riski $R(g)$ üle kõigi klassifitseerijate hulga $\mathcal{G}$. Definitsiooni järgi rahuldab Bayesi klassifitseerija iga $x \in \mathcal{X}$ korral seost

$$g^*(x) \coloneqq \arg\min_{g \in \mathcal{G}} R(g(x)|x), \tag{1.7}$$

kus minimeerimine käib üle kõigi klassifitseerijate hulga $\mathcal{G}$. Seega kehtib iga klassifitseerija g ja tunnusvektori x korral:

$$R\big(g^*(x)|x\big) \leqslant R\big(g(x)|x\big).$$

Keskväärtuse monotoonsuse omadusest ja võrdustereast (1.6) saame, et iga klassifitseerija g korral kehtib:

$$R(g^*) = E_X\left[R(g^*(X)|X)\right] \leqslant E_X\left[R(g(X)|X)\right] = R(g).$$

Seega minimeerib Bayesi klassifitseerija g^* riski üle kõigi klassifitseerijate hulga $\mathcal{G}$,

$$R(g^*) = \inf_{g\in\mathcal{G}} R(g). \tag{1.8}$$

Bayesi klassifitseerija riski nimetatakse Bayesi riskiks. Tähistame Bayesi riski $R^* := R(g^*)$.

Eespool nägime (vt (1.3)), et 0-1 kaofunktsiooni korral on klassifitseerija risk võrdne valesti klassifitseerimise tõenäosusega. Klassifitseerija g tinglik risk on 0-1 kaofunktsiooni ja fikseeritud $x \in \mathcal{X}$ korral võrdne klassifitseerimisvea tegemise tõenäosusega x korral:

$$R(g(x)|x) = E_{Y|X}[I_{Y=g(X)}|X = x] = P(Y \neq g(X)|X = x).$$

Et tinglikku riski iga $x \in \mathcal{X}$ korral minimeerida, seab Bayesi klassifitseerija 0-1 kaofunktsiooni korral tunnusvektorile x vastavusse kõige tõenäosema klassi. Kuna iga objekt kuulub täpselt ühte klassi, on tunnusvektori x korral klassi 1 kuulumise tõenäosus $\eta(x)$ suurem klassi -1 kuulumise tõenäosusest parajasti siis, kui $\eta(x) > \frac{1}{2}$. Bayesi klassifitseerija on sümmeetrilise kaofunktsiooni korral kujul:

$$g^*(x) = \begin{cases} 1 & \text{kui } \eta(x) > \frac{1}{2}, \\ -1 & \text{kui } \eta(x) \leqslant \frac{1}{2}. \end{cases} \tag{1.9}$$

Bayesi klassifitseerija klassifitseerib kõik objektid tunnusvektoriga x klassi, millesse kuulumise tinglik tõenäosus on x korral suurem. Seega on tõenäosus, et objekt tunnusvektoriga x kuulub klassi, millesse kuulumise tõenäosus on x korral

väiksem, võrdne klassifitseerimisvea tegemise tõenäosusega. Objekti, mille tunnusvektor on x, klassifitseerimisel vea tegemise tõenäosus on sümmeetrilise kaofunktsiooni korral võrdne tingliku riskiga $R(g(x)|x)$. Seega avaldub Bayesi klassifitseerija tinglik risk 0-1 kaofunktsiooni ja $x \in \mathcal{X}$ korral:

$$R(g^*(x)|x) = \min\{\eta(x), 1 - \eta(x)\}.$$

Bayesi risk avaldub seetõttu (vt (1.6)):

$$R^* = E_X \min\{\eta(X), 1 - \eta(X)\}. \tag{1.10}$$

Peatükk 2

Vapnik-Tšervonenkise teooria

Teises peatükis tutvume andmetest õppimise põhimõtetega. Teooria tugineb suuresti vene teadlaste Vladimir Vapniku ja Aleksei Tšervonenkise tööle, mistõttu nimetatakse seda tehisõppe osa Vapnik-Tšervonenkise teooriaks, lühendatult ka VC teooriaks.

2.1 Andmetest õppimine

Seni näidatud esitused riskile ja Bayesi klassifitseerijale on arvutuseeskirjadena kasutatavad vaid juhul, kui teame juhusliku vektori (X, Y) jaotust. Praktilist huvi pakkuvate klassifitseerimisprobleemide korral me objektide jaotust ei tea. Tehisõppe ülesannet, mille korral soovitakse järele aimata või tuvastada [16, lk 21] seost tunnusvektori väärtuste hulga $\mathcal{X}$ ja klasside hulga $\mathcal{Y}$ vahel, nimetatakse suunatud õppeks (ingl *supervised learning*) [8, lk 2]. Õppimine seisneb olemasolevate andmete põhjal järelduste tegemises. Õppijale (ingl *learner*), milleks on tavaliselt arvutiprogramm, antakse ette andmestik, mida nimetame treeningvalimiks. Iga

treeningvalimisse

$$\mathcal{D}_n = (x_i, y_i), \quad i = 1, ..., n$$

kuuluva objekti kohta teame me nii selle tunnusvektorit x_i, kui ka seda, millisesse klassi objekt kuulub, $y_i \in \mathcal{Y}$. Eeldame, et objektid on genereeritud sõltumatult samast jaotusest, mis võib meile tundmatu olla. Suunatud õppe korral kasutame treeningvalimi objekte, et konstrueerida klassifitseerija, mis kõikide treeningvalimi objektidega samast jaotusest objektide klassifitseerimisel võimalikult vähe eksiks. Suunatud õpet kutsutakse „suunatuks", kuna väljundiks olev klassikuuluvus suunab andmetest õppimise protsessi [8, lk 2].

Suunatud õppest erineb suunamiseta õpe (ingl *unsupervised learning*). Seal otsitakse andmestikusiseseid seoseid, mingit varem teada olevat klasside hulka seda protsessi suunamas pole [8, lk 438].

Treeningandmeid kasutades saame treeningvalimist sõltuva klassifitseerija

$$g_n : (\mathcal{X} \times \mathcal{Y})^n \times \mathcal{X} \to \mathcal{Y},$$

mille risk

$$R(g_n) = E_{X,Y} L\big(Y, g_n\left(\mathcal{D}_n, X\right)\big)$$

sõltub samuti andmetest.

2.2 Empiirilise riski minimeerimine

Defineerime klassifitseerija g empiirilise riski:

$$R_n(g) \coloneqq \frac{1}{n} \sum_{i=1}^{N} L\big(y_i, g(x_i)\big). \tag{2.1}$$

Treeningvalimi korral arvutatud empiiriline risk on keskmine kaofunktsiooni väärtus, mis treeningvalimi elementide klassifitseerimisega kaasneb. Sümmeetrilise kaofunktsiooni korral on klassifitseerija empiiriline risk võrdne klassifitseerimisvigade osakaaluga,

$$R_n(g) = \frac{1}{n} \sum_{i=1}^{N} I_{y_i \neq g(x_i)}. \tag{2.2}$$

Nimetame sümmeetrilise kaofunktsiooni ja treeningvalimi korral arvutatud empiirilist riski treeningveaks. Klassifitseerija treeningviga on võrdne klassifitseerija poolt treeningvalimi objektide klassifitseerimisel tehtud vigade osakaaluga.

Fikseeritud klassifitseerimisprobleemi korral on parim klassifitseerija selline, mis riski üle kõikide klassifitseerijate hulga minimeerib (vt (1.8)). Sageli pole (X, Y) jaotus teada, klassifitseerija konstrueerimiseks saame kasutada treeningvalimit $\mathcal{D}_n$. Kui tunnuste seas on kasvõi üks pidev tunnus, pole kahe objekti tunnusvektorid kunagi võrdsed ja me saame alati leida klassifitseerija, mis treeningvalimi objektide klassifitseerimisel kunagi ei eksi. Selline klassifitseerija klassifitseerib treeningvalimi $\mathcal{D}_n$ objekte küll eksimatult, ent mõne teise samast jaotusest valimi korral on tema klassifitseerimisvigade arv suur. Seda, kui klassifitseerija kirjeldab treeningvalimi juhuslikkusest tulenevaid iseärasusi liiga täpselt, nimetatakse ülesobitumuseks [9, lk 30-31].

Treeningvea üle kõikide klassifitseerijate hulga minimeerimisel pole erilist mõtet. Seetõttu fikseeritakse suunatud õppe korral alati mõni kõikide klassifitseerijate hulga alamhulk, mille seast hiljem parimat klassifitseerijat otsitakse. „Andmetest õppimine on protsess, mis seisneb antud funktsioonide klassist sobiva funktsiooni väljavalimises.” [16, lk 21]

Klassifitseerijate klassist $\mathcal{G}$ parima klassifitseerija g_n välja valimisel empiirilise

riski põhjal otsuse tegemist,

$$\hat{g}_n = \arg\min_{g \in \mathcal{G}} R_n\big(g\big), \tag{2.3}$$

nimetame empiirilise riski minimeerimise printsiibiks (ERM-printsiibiks) [10, lk 24]. Sümmeetrilise kaofunktsiooni korral minimeeritakse ERM-printsiibil funktsioonide klassist $\mathcal{G}$ klassifitseerija valimisel treeningviga.

Vaatleme funktsioonide klassist $\mathcal{G}$ treeningandmete põhjal valitud klassifitseerijat g_n. Võrdleme klassifitseerija g_n klassifitseerimisomadusi parima võimaliku klassifitseerija, Bayesi klassifitseerija omadustega. Selleks uurime klassifitseerija g_n teoreetilise riski ja Bayesi riski vahet $R(g_n) - R^*$ ja lahutame selle kaheks osaks, hindamisveaks ja lähendamisveaks [9, lk 30]:

$$R(g_n) - R^* = \big(R(g_n) - \inf_{g \in \mathcal{G}} R(g)\big) + \big(\inf_{g \in \mathcal{G}} R(g) - R^*\big). \tag{2.4}$$

Hindamisviga

$$R(g_n) - \inf_{g \in \mathcal{G}} R(g)$$

kirjeldab, kui palju erineb g_n klassifitseerimisomadustelt parimast klassi $\mathcal{G}$ kuuluvast klassifitseerijast. Hindamisvea suurus sõltub treeningandmetest ja klassifitseerija g_n klassist $\mathcal{G}$ valimise meetodist.

Lähendamisviga

$$\inf_{g \in \mathcal{G}} R(g) - R^*$$

kirjeldab, kui hästi klassi $\mathcal{G}$ liikmed Bayesi klassifitseerijat imiteerida suudavad. Lähendamisviga on klassi $\mathcal{G}$ omadus ja selle suurus treeningandmetest ei sõltu.

Võib jääda mulje, et klassifitseerijate hulka $\mathcal{G}$ kuuluvate klassifitseerijate arvu ja kahte tüüpi vigade vahel on selge seos: kui $\mathcal{G}$ on suur, on lähendamisviga väike ja hindamisviga suur; kui $\mathcal{G}$ on väike, on lähendamisviga suur ja hindamisviga väike. Kui hulka $\mathcal{G}$ kuulub ainult üks funktsioon, on hindamisviga võrdne nulliga. Lähendamisviga võrdub nulliga siis, kui klass $\mathcal{G}$ on kõikide klassifitseerijate

hulk. Sellisel juhul kuulub klassi $\mathcal{G}$ ka Bayesi klassifitseerija, mistõttu võrreldakse hindamisvea leidmisel andmete põhjal leitud klassifitseerija g_n riski Bayesi riskiga, st hindamisviga võib olla suur. Tegelikult sõltub klassifitseerijate hulga $\mathcal{G}$ puhul hindamis- ja lähendamisvea suurus hoopis järgmises alapeatükis tutvustatavast omadusest.

2.3 Vapnik-Tšervonenkise dimensioon

Klassifitseerijate hulga $\mathcal{G}$ klassifitseerimisomadusi kirjeldab selle liikmete arvust paremini tema komplekssus. Komplekssus iseloomustab klassifitseerijate klassi mitmekesisust [10, lk 27]. Üheks klassifitseerijate klassi komplekssuse mõõduks on klassi Vapnik-Tšervonenkise dimensioon ehk VC dimensioon. VC dimensiooni definitsiooni esitamiseks tutvume eraldavuskordaja (ingl *shatter coefficient*) mõistega.

Olgu $\mathcal{G}$ klassifitseerijate klass. Olgu $\mathcal{D}_n = \big((x_1, y_1), ..., (x_n, y_n)\big)$ objektide hulk, mille liikmed on sõltumatud ja samast jaotusest. Iga klassifitseerija $g \in \mathcal{G}$ puhul kehtib

$$\big(g(x_1), ..., g(x_n)\big) \in \{-1, 1\}^n.$$

Sellise klassikuuluvustest 1 ja -1 koosneva n-mõõtmelise vektori moodustamiseks on 2^n erinevat moodust. Seda, kui palju erinevaid järjendeid n objekti korral klassifitseerijate $g \in \mathcal{G}$ põhjal moodustada saab, kirjeldab järgnevalt defineeritav eraldavuskordaja mõiste.

Definitsioon 2.1 (vt [12, lk 2])**.** *Klassi $\mathcal{G}$ n-ndaks eraldavuskordajaks nimetame suurust*

$$\mathcal{S}_{\mathcal{G}}(n) \coloneqq \max_{x_1,...,x_n \in \mathcal{X}} \big|\big(g(x_1), ..., g(x_n)\big) \in \{-1, 1\}^n, g \in \mathcal{G}\big|, \tag{2.5}$$

kus $|\cdot|$ *on hulka kuuluvate elementide arv.*

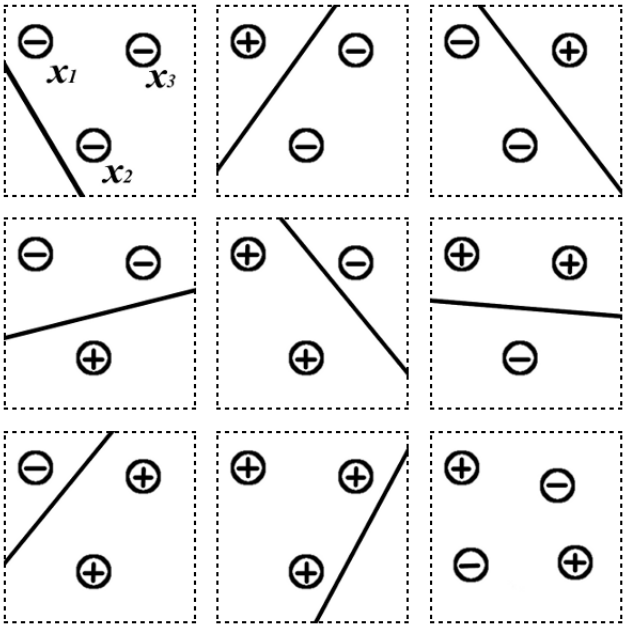
Hulga $\mathcal{G}$ n-s eraldavuskordaja $\mathcal{S}_{\mathcal{G}}(n)$ näitab, milline on suurim arv erinevaid klassikuuluvuste kombinatsioone, mis on võimalik hulga $\mathcal{G}$ liikmete poolt n-elemendilistele hulka $\mathcal{X}$ kuuluvate liikmetega punktihulkadele vastavusse seada. Kui klassifitseerijate hulga $\mathcal{G}$ liikmete abil on võimalik punktidele $(x_1, ..., x_n)$ vastavusse seada 2^n erinevat klassikuuluvuste kombinatsiooni, siis ütleme, et $\mathcal{G}$ eraldab täielikult (ingl *shatters* [16, lk 147]) punktihulga $(x_1, ..., x_n)$.

Definitsioon 2.2 (vt [12, lk 2])**.** *Klassifitseerijate hulga $\mathcal{G}$ Vapnik-Tšervonenkise (VC) dimensioon on suurim täisarv n, mille korral kehtib $\mathcal{S}_{\mathcal{G}}(n) = 2^n$.*

Klassifitseerijate hulga $\mathcal{G}$ VC dimensioon on vähemalt n, kui leidub vähemalt üks punktide hulk $(x_1, ..., x_n)$, mille korral saab $\mathcal{G}$ liikmete abil moodustada kõikvõimalikud alamhulgad hulgast $\{-1, 1\}^n$. Klassi $\mathcal{G}$ VC dimensioon on väiksem kui n, kui ühegi n-elemendilise punktihulga $(x_1, ..., x_n)$ korral pole $\mathcal{G}$ elementide abil võimalik moodustada kõiki hulga $\{-1, 1\}^n$ erinevaid alamhulki.

Näide 2.1. Olgu $d = 2$ ja olgu tunnused X_1 ja X_2 pidevad. Olgu $\mathcal{G}$ klassifitseerijate hulk, mille moodustavad kõik ruumi $\mathbb{R}^2$ sirged. Sirget kasutame klassifitseerijana sellisel moel, et ühele poole sirget jäävatele objektidele seatakse vastavusse klassikuuluvus -1 ja teisele poole jäävatele objektidele klassikuuluvus 1. Joonisel 2.1 on näidatud, et kolme punkti $x_1, x_2, x_3 \in \mathbb{R}^2$ puhul on hulka $\mathcal{G}$ kuuluvate klassifitseerijate abil võimalik tekitada 8 erinevat järjendit $\big(g(x_1), g(x_2), g(x_3)\big)$, st kõigi sirgete hulga $\mathcal{G}$ kolmas eraldavuskordaja on 8, $\mathcal{S}_{\mathcal{G}}(3) = 2^3 = 8$.

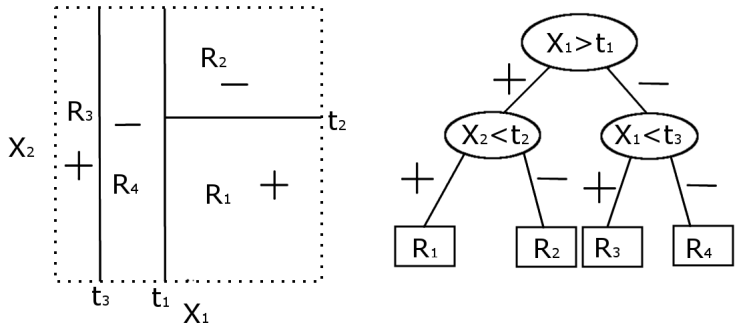
Joonise viimaselt alamjoonisel on kujutatud nelja punkti, millele vastavad klassikuuluvused, mida ühegi sirge abil neile vastavusse seada ei saa. Saab näidata [3, lk 125], et ei leidu ühtegi neljaliikmelist punktihulka, mille sirgete hulk täielikult eraldaks. Seega on sirgete hulga VC dimensioon 3.



Joonis 2.1. Kahemõõtmelises ruumis on kõigi sirgete hulga VC dimensioon 3.

Näide 2.2. Olgu $d = 2$ ja olgu tunnused X_1 ja X_2 reaalväärtuselised. Vaatleme klassifitseerijatena ainult neid sirgeid, mis on kummagi teljega paralleelsed. Jagame juhuslike vektorite (X_1, X_2) kõikvõimalikest väärtustest koosneva hulga $\mathcal{X}$ teljega paralleelse sirgega kaheks piirkonnaks. Seejärel jagame ühe või mõlemad tekkinud piirkondadest teljega paralleelse sirgega omakorda kaheks osaks. Tekkinud piirkonnad võime omakorda ühe teljega paralleelse sirgega kaheks jagada jne. Sellisel moel teljega paralleelseid sirgeid klassifitseerijatena kasutades defineerime kahendpuu-klassifitseerijate hulga $\mathcal{P}$.

Olgu $\mathcal{P}_m$ kõigi selliste kahendpuu-tüüpi klassifitseerijate hulk, millel on maksimaalselt $m + 1$ taset ja mille sõlmpunktides olevad tingimused sõltuvad ainult ühest tunnusest. Igale $\mathcal{P}_m$ liikmele vastab tunnusvektori väärtuste ruumi $\mathcal{X}$ maksimaalselt 2^m-ks telgedega paralleelsete piiridega piirkonnaks jaotamise tulemus.



Joonis 2.2. Rekursiivselt piirkondadeks jagamine ja sellele vastav kahendpuu.

Sellise kahendpuu sõlmed on analoogsed sirgetega, mis on ühe teljega paralleelsed. Kui kahendpuul on $m+1$ taset, siis on sellel maksimaalselt 2^m-1 sõlmpunkti. Kasutame klassifitseerijate hulga $\mathcal{P}_m$ VC-dimensiooni ülemise tõkkena järgmist hinnangut, mis sõltub hulga $\mathcal{P}_m$ liikmete maksimaalsest sõlmpunktide arvust (vt [13]):

$$V_{\mathcal{P}_m} \leqslant 3 \cdot (2^m - 1). \tag{2.6}$$

Näiteks pole maksimaalselt kolme tasemega kahendpuu-klassifitseerijate klassi $\mathcal{P}_2$ VC dimensioon suurem kui 9. Klassi $\mathcal{P}_2$ illustreerib joonis 2.2.

2.4 Riskihinnangud Vapnik-Tšervonenkise dimensiooni kaudu

Meid huvitab, mis kaasneb teoreetilise juhusliku vektori (X, Y) jaotuse asemel treeningvalimi kasutamisega. Järgnev teoreem annab hinnangu ja ülemise usaldusPiiri teoreetilise riski $R(g)$ ja treeningvalimilt arvutatud riski $R_n(g)$ vahele.

Teoreem 2.1 (vt [10, lk 34])**.** *Olgu $\mathcal{G}$ klassifikaatorite klass, mille VC dimensioon*

V on lõplik. Kaheklassilise klassifitseerimisprobleemi ja sümmeetrilise kaofunktsiooni korral kehtivad riskihinnangud

$$E\big(\sup_{g\in\mathcal{G}}|R_n(g)-R(g)|\big)\leqslant 2\sqrt{\frac{V\ln(n+1)+\ln 2}{n}} \tag{2.7}$$

ja

$$P\big(\sup_{g\in\mathcal{G}}|R_n(g)-R(g)|>\epsilon\big)\leqslant 8(n+1)^V e^{-n\epsilon^2/32}, \tag{2.8}$$

kus $\epsilon>0$.

Võrratust (2.8) nimetatakse Vapnik-Tšervonenkise võrratuseks.

Järeldus 2.1 (vt [10, lk 35])**.** *Olgu $\hat{g}_n$ ERM-printsiibil hulgast $\mathcal{G}$ valitud klassifitseerija. Siis kehtivad võrratused*

$$ER(\hat{g}_n)-\inf_{g\in\mathcal{G}}R(g)\leqslant 4\sqrt{\frac{V\ln(n+1)+\ln 2}{n}} \tag{2.9}$$

ja

$$P\big(R(\hat{g}_n)-\inf_{g\in\mathcal{G}}R(g)>\epsilon\big)\leqslant 8(n+1)^V e^{-n\epsilon^2/128} \tag{2.10}$$

Tõestus.

$$\begin{aligned}
ER(\hat{g}_n)-\inf_{g\in\mathcal{G}}R(g) &= ER(\hat{g}_n)-ER_n(\hat{g}_n)+ER_n(\hat{g}_n)-\inf_{g\in\mathcal{G}}R(g)= \\
&= E\left(R(\hat{g}_n)-R_n(\hat{g}_n)\right)+E\left(R_n(\hat{g}_n)-\inf_{g\in\mathcal{G}}R(g)\right)\leqslant \\
&\leqslant E\left[\sup_{g\in\mathcal{G}}\left(R(g)-R_n(g)\right)\right]+E\left[\sup_{g\in\mathcal{G}}\left(R_n(\hat{g}_n)-R(g)\right)\right]\leqslant \\
&\leqslant E\left[\sup_{g\in\mathcal{G}}\left(R(g)-R_n(g)\right)\right]+E\left[\sup_{g\in\mathcal{G}}\left(R_n(g)-R(g)\right)\right]\leqslant \\
&\leqslant E\left(\sup_{g\in\mathcal{G}}|R(g)-R_n(g)|\right)+E\left(\sup_{g\in\mathcal{G}}|R_n(g)-R(g)|\right)= \\
&= 2E\big(\sup_{g\in\mathcal{G}}|R_n(g)-R(g)|\big)\leqslant \\
&\overset{(2.7)}{\leqslant} 4\sqrt{\frac{V\ln(n+1)+\ln 2}{n}}
\end{aligned}$$

Saadud tulemust kasutame teise võrratuse kehtivuse näitamiseks:

$$\begin{aligned} P\big(R(\hat{g}_n) - \inf_{g\in\mathcal{G}} R(g) > \epsilon\big) &\leqslant P\big(2\sup_{g\in\mathcal{G}} |R_n(g) - R(g)| > \epsilon\big) = \\ &= P\left(\sup_{g\in\mathcal{G}} |R_n(g) - R(g)| > \frac{\epsilon}{2}\right) = \\ &\overset{(2.8)}{\leqslant} 8(n+1)^V e^{-n\left(\frac{\epsilon}{2}\right)^2/32} = \\ &= 8(n+1)^V e^{-n\epsilon^2/128}. \end{aligned}$$

□

Vapnik-Tšervonenkise võrratusest (2.1) tuleneb ka järgmine, nn PAC-hinnang. Järgnevas esitatud ja tõestatud võrratus annab ülemise usalduspiiri suvalisele hulka $\mathcal{G}$ kuuluva klassifitseerija riskile.

Järeldus 2.2 (vt [10, lk 35])**.** *Suvalise $g_n \in \mathcal{G}$ korral kehtib tõenäosusega $1-\delta$ võrratus:*

$$R(g_n) \leqslant R_n(g_n) + 2\sqrt{\frac{8\big(V\ln(n+1) - \ln\delta + \ln 8\big)}{n}} \tag{2.11}$$

Tõestus. Peame hindama, kui tõenäoselt antud võrratus kehtib, ehk tarvis on hinnata tõenäosust

$$P\left(R(g_n) - R_n(g_n) \leqslant 2\sqrt{\frac{8\big(V\ln(n+1) - \ln\delta + \ln 8\big)}{n}}\right).$$

Selleks hindame tõenäosust, et võrratus ei kehti:

$$\begin{aligned} &P\left(R(g_n) - R_n(g_n) > 2\sqrt{\frac{8\big(V\ln(n+1) - \ln\delta + \ln 8\big)}{n}}\right) \leqslant \\ &\leqslant P\left(\sup_{g\in\mathcal{G}} |R_n(g) - R(g)| \geqslant 2\sqrt{\frac{8\big(V\ln(n+1) - \ln\delta + \ln 8\big)}{n}}\right) \leqslant \end{aligned}$$

$$\overset{(2.8)}{\leqslant} 8(n+1)^V \exp\left[\frac{-n\left(2\sqrt{\frac{8\left(V\ln(n+1)-\ln\delta+\ln 8\right)}{n}}\right)^2}{32}\right] =$$

$$= 8(n+1)^V \exp\left[\frac{-n\cdot 4\cdot 8\left(V\ln(n+1)-\ln\delta+\ln 8\right)}{n\cdot 32}\right] =$$

$$= 8(n+1)^V \exp\left[-\left(\ln(n+1)^V-\ln\delta+\ln 8\right)\right] =$$

$$= 8(n+1)^V \exp\left[\ln\frac{\delta}{8(n+1)^V}\right] = \delta.$$

Saime, et võrratuse mittekehtimise tõenäosus ei ole suurem kui δ. Seega kehtib võrratus (2.11) tõenäosusega $1-\delta$. □

Peatükk 3

AdaBoost

Kolmandas peatükis tutvume esmalt ϕ-riski teooriaga. Seejärel tutvume kõige populaarsema [8, lk 299] *boosting*-algoritmi AdaBoostiga.

3.1 ϕ-risk

Kogu selles alapeatükis esitatud mõttekulg tugineb artiklile [1].

Soovime treeningvalimi $\mathcal{D}_n = \big((x_1, y_1), ..., (x_n, y_n)\big)$ põhjal konstrueerida võimalikult täpset klassifitseerijat g. Selleks oleks loomulik minimeerida empiirilist riski 0-1 kaofunktsiooni korral ehk leida funktsioon $g \in \mathcal{G}$, mille korral oleks treeningvalimi klassifitseerimisel tehtud klassifitseerimisvigade osakaal (2.2) võimalikult väike. Osutub, et empiirilise riski minimeerimise probleem on paljude mittetriviaalsete funktsiooniklasside $\mathcal{G}$ korral arvutuslikult keerukas. ERM-printsiibi rakendamise keerukusele viitab see, et seal kasutav 0-1 kaofunktsioon pole kumer. Sellest tulenevalt kasutatakse tehisõppes, sh *boosting*-meetodite juures kumeraid nn surrogaat-kaofunktsioone $\phi\big(yf(x)\big)$. Uurime järgnevas, millised tagajärjed 0-1

kaofunktsiooni asemel kumera ja seega arvutuslikult märksa tõhusama funktsiooni kasutamisega kaasnevad.

Olgu $\mathcal{F}$ reaalväärtuseliste funktsioonide klass hulgal $\mathcal{X}$, $f : \mathcal{X} \mapsto \mathbb{R}$. Hulgale $\mathcal{F}$ vastab klassifitseerijate klass $\mathcal{G}$, kus iga $f \in \mathcal{F}$ defineerib klassifitseerija $\mathrm{sgn}(f)$,

$$\mathcal{G} = \{\mathrm{sgn}(f) : f \in \mathcal{F}\}$$

Olgu ϕ mittenegatiivne funktsioon, st

$$\phi : \mathbb{R} \mapsto [0, \infty).$$

Funktsiooni f ϕ-risk on defineeritud järgnevalt:

$$R_\phi(f) \coloneqq E_{X,Y}\, \phi\big(Y f(X)\big). \tag{3.1}$$

Funktsiooni f riskiks nimetame klassifitseerija $\mathrm{sgn}(f)$ riski 0-1 kaofunktsiooni korral (vt (1.3)):

$$R(f) \coloneqq R\big(\mathrm{sgn}(f)\big) = P\big(Y \neq \mathrm{sgn}(f(X))\big). \tag{3.2}$$

Olgu f_ϕ^* funktsioon, mis minimeerib ϕ-riski üle kõigi funktsioonide. Funktsioonile f_ϕ^* vastav risk on minimaalne ϕ-risk, $R_\phi^* \coloneqq R_\phi(f_\phi^*)$. Meid huvitab, milline on klassifitseerija $\mathrm{sgn}(f_\phi^*)$ ja Bayesi klassifitseerija g^* vaheline seos.

Defineerime $\eta \in [0, 1]$ korral funktsiooni $H(\eta)$:

$$H(\eta) \coloneqq \inf_{\alpha \in \mathbb{R}} \big(\eta\phi(\alpha) + (1-\eta)\phi(-\alpha)\big), \tag{3.3}$$

ja funktsiooni

$$H^-(\eta) \coloneqq \inf_{\alpha:\, \alpha(2\eta-1) \leqslant 0} \big(\eta\phi(\alpha) + (1-\eta)\phi(-\alpha)\big). \tag{3.4}$$

Kehtib $H^-(\eta) \geqslant H(\eta)$ - suuruse $H(\eta)$ leidmisel minimeeritakse üle kõikvõimalike argumentide hulga, $H^-(\eta)$ leidmisel arvestatakse ainult teatud tingimusele

vastavate argumentidega. Veel märkame, et $H(\frac{1}{2}) = H^-(\frac{1}{2})$, sest punktis $\eta = \frac{1}{2}$ pole tingimus $\alpha(2\eta - 1) \leqslant 0$ kitsendav. Defineerime suuruste $H(\eta)$ ja $H^-(\eta)$ kaudu funktsiooni ϕ iseloomustava omaduse.

Definitsioon 3.1 ([1, lk 7])**.** *Öeldakse, et funktsioon ϕ on kalibreeritud, kui iga $\eta \neq \frac{1}{2}$ korral kehtib:*

$$H^-(\eta) > H(\eta). \tag{3.5}$$

Järgmine teoreem pakubki kriteeriumi, mis seob funktsiooni f iseloomustava ϕ-riski tema poolt defineeritud klassifitseerija $\operatorname{sgn}(f)$ riskiga.

Teoreem 3.1 (vt [1, lk 8])**.** *Olgu $f \in \mathcal{F}$. Kui funktsioon ϕ on kalibreeritud ja kumer, siis leidub mittenegatiivne ja kumer funktsioon Ψ nii, et $\Psi(t) = 0$ parajasti siis, kui $t = 0$ ja*

$$\Psi\big(R(f) - R^*\big) \leqslant R_\phi(f) - R_\phi^*. \tag{3.6}$$

Esitame ja tõestame teoreemis 3.1 esitatud tingimustele vastava funktsiooni Ψ kasuliku omaduse.

Järeldus 3.1. *Olgu a_n mittenegatiivsete liikmetega arvjada. Siis $\Psi(a_n) \to 0$ parajasti siis, kui $a_n \to 0$.*

Tõestus. Tarvilikkus. Kehtigu $\Psi(a_n) \to 0$. Oletame vastuväiteliselt, et jada (a_n) ei koondu nulliks. Siis leidub $\epsilon > 0$, mille korral leidub iga naturaalarvu N korral naturaalarv m, mis rahuldab võrratust $m \geqslant N$ ja millele vastava a_m puhul kehtib $a_m \geqslant \epsilon$. Seega saame moodustada jada (a_n) osajada (a_{n_k}), mille kõigi liikmete korral kehtib $a_{n_k} \geqslant \epsilon$. Vaatleme jada $\big(\Psi(a_{n_k})\big)$. Kuna funktsioon Ψ on kumer ja mittenegatiivne ning on võrdne nulliga vaid argumendi 0 korral, siis on Ψ mittenegatiivsete argumentide korral kasvav funktsioon ja kehtib $\Psi(\epsilon) > 0$. Jada (a_n) osajada (a_{n_k}) moodustamise viisist lähtub, et jada $\big(\Psi(a_{n_k})\big)$ kõigi liikmete korral

kehtib $\Psi(a_{n_k}) \geqslant \Psi(\epsilon)$. Poollõiku $\big[0, \Psi(\epsilon)\big)$ ei kuulu ühtegi jada $\big(\Psi(a_{n_k})\big)$ liiget, seega ei koondu jada $\big(\Psi(a_{n_k})\big)$ nulliks. Kuna jada $\big(\Psi(a_n)\big)$ koondub nulliks, peaks nulliks koonduma ka selle jada iga osajada. Jõudsime vastuoluni.

Piisavus. Kehtigu $a_n \to 0$. Kuna funktsioon Ψ on kumer, siis on see ka pidev. Seega järeldub koondumisest $a_n \to 0$ koondumine $\Psi(a_n) \to \Psi(0)$. Funktsioon Ψ on võrdne nulliga parajasti siis, kui selle argument on null, seega $\Psi(a_n) \to 0$. □

Teoreemist 3.1 lähtub, et kui meil õnnestub leida funktsioon f, mis minimeerib kalibreeritud ja kumera ϕ korral ϕ-riski üle kõigi reaalväärtuseliste funktsioonide hulga, siis minimeerib see funktsioon ka riski (3.2), st funktsiooni f põhjal defineeritud klassifitseerija $\operatorname{sgn}(f)$ on Bayesi klassifitseerija. Kui leidub reaalväärtuseliste funktsioonide jada (f_n), mille puhul kehtib kumera ja kalibreeritud ϕ korral $R_\phi(f_n) \to R^*_\phi$, siis kehtib $\Psi\big(R(f_n) - R^*\big) \to 0$, millest saame järeldust 3.1 kasutades, et kehtib $R(f_n) \to R^*$. Seega on kumera ja kalibreeritud ϕ korral ϕ-riski üle kõikide funktsioonide hulga minimeerimine samaväärne riski üle kõikide klassifitseerijate hulga minimeerimisega.

Kui funktsioon ϕ on kumer, siis saame selle kalibreerituse üle otsustada järgmise teoreemi abil.

Teoreem 3.2 ([1, lk 7])**.** *Olgu ϕ kumer funktsioon. Siis on ϕ kalibreeritud parajasti siis, kui see on punktis 0 diferentseeruv ja kehtib $\phi'(0) < 0$.*

Kuna juhusliku vektori (X, Y) jaotus pole meile tavaliselt teada, kasutame ka ϕ-riski hindamiseks empiirilist riski. Defineerime funktsiooni f empiirilise ϕ-riski:

$$\hat{R}_\phi(f) \coloneqq \frac{1}{n} \sum_{i=1}^{n} \phi\big(y_i f(x_i)\big). \tag{3.7}$$

Näide 3.1. Uurime, kas funktsioon e^{-t} on kalibreeritud. Olgu $\phi(t) = e^{-t}$. Siis minimaalne tinglik ϕ-risk avaldub:

$$H(\eta) = \inf_{\alpha \in \mathbb{R}} C_\eta(\alpha),$$

kus

$$C_\eta(\alpha) \coloneqq \eta e^{-\alpha} + (1-\eta)e^{\alpha}$$

Suuruse $H(\eta)$ leidmiseks tuleb meil leida funktsiooni $C_\eta(\alpha)$ alumine raja. Funktsioon $C_\eta(\alpha)$ on rangelt kumerate funktsioonide summa ja on seega ka ise kogu oma määramispiirkonnas rangelt kumer. Seega on sellel funktsioonil maksimaalselt üks statsionaarne punkt. Kui statsionaarne punkt leidub, on see miinimum. Kui miinimum leidub, on see ühtlasi funktsiooni alumiseks rajaks. Leiame funktsiooni $C_\eta(\alpha)$ tuletise:

$$C'_\eta(\alpha) = -\eta e^{-\alpha} + (1-\eta)e^{\alpha}.$$

Statsionaarse punkti leidmiseks võrdsustame tuletise nulliga:

$$\begin{aligned}
& -\eta e^{-\alpha} + (1-\eta)e^{\alpha} = 0 \quad \Longrightarrow \\
\Longrightarrow \quad & -\eta + (1-\eta)e^{2\alpha} = 0 \quad \Longrightarrow \\
\Longrightarrow \quad & e^{2\alpha} = \frac{\eta}{1-\eta} \quad \Longrightarrow \\
\Longrightarrow \quad & \alpha^*(\eta) \coloneqq \alpha = \frac{1}{2}\ln\frac{\eta}{1-\eta}.
\end{aligned}$$

Nüüd saame lihtsustada:

$$H(\eta) = C_\eta\big(\alpha^*(\eta)\big) = \eta\frac{\sqrt{1-\eta}}{\sqrt{\eta}} + (1-\eta)\frac{\sqrt{\eta}}{\sqrt{1-\eta}} = 2\sqrt{\eta(1-\eta)}.$$

Suuruse $H^-(\eta)$ leidmiseks paneme tähele, et kui $\eta < \frac{1}{2}$, siis kehtib $(2\eta - 1) < 0$, st $H^-(\eta)$ leidmisel nõutava erimärgilisuse tingimuse rahuldamiseks peaks kehtima $\alpha \geqslant 0$. Paneme tähele, et $\eta < \frac{1}{2}$ korral kehtib $\alpha^*(\eta) < 0$. Kuna funktsiooni $C_\eta(\alpha)$

ainus statsionaarne punkt piirkonda $[0, \infty)$ ei kuulu, saavutab funktsioon $C_\eta(\alpha)$ range kumeruse tõttu piirkonnas $[0, \infty)$ minimaalse väärtuse punktis $\alpha = 0$. Siis

$$H^-\eta = C_\eta(0) = 1.$$

Kui $\eta > \frac{1}{2}$, leiame sarnaselt arutledes, et $H^-(\eta) = C_\eta(0) = 1$

Kui $\eta = \frac{1}{2}$, kehtib alati $H(\eta) = H^-(\eta)$. Seega kehtib $H^-\frac{1}{2} = \alpha^*(\frac{1}{2})$ millest saame, et kehtib $H^-(\frac{1}{2}) = 1$.

Seega kehtib iga $\eta \in [0, 1]$ korral:

$$H^-(\eta) \equiv 1.$$

Funktsioon $H(\eta) = 2\sqrt{\eta(1-\eta)}$ on nõgus ja saavutab oma maksimumi punktis $\frac{1}{2}$. Kui $\eta \neq \frac{1}{2}$, kehtib $H(\eta) < 1 = H^-(\eta)$, st definitsiooni (3.1) järgi on $\phi(t) = e^{-t}$ kalibreeritud funktsioon.

Funktsioon e^{-t} on kumer, seega saab selle kalibreerituse üle ka teoreemi (3.2) alusel otsustada. Leiame funktsiooni tuletise punktis 0:

$$\phi'(0) = -e^0 = -1.$$

Leidsime, et kehtib $\phi'(0) < 0$, seega on funktsioon e^{-t} kalibreeritud.

Näide 3.2. Leiame ka ühe funktsiooni, mis pole kalibreeritud. Võtame $\phi(t) = e^t$. Siis $\phi(t)$ on kumer, mistõttu saame sellele rakendada teoreemi (3.2). Leiame funktsiooni $\phi(t)$ tuletise punktis 0:

$$\phi'(0) = e^0 = 1.$$

Seega kehtib $\phi'(0) > 0$. Teoreemist (3.2) saame, et funktsioon e^t pole kalibreeritud. Funktsiooni $\phi(t) = e^t$ puhul on selge, miks selle poolt defineeritud ϕ-riski minimeerimisel leitava funktsiooni f põhjal defineeritud klassifitseerija $\mathrm{sgn}(f)$

riski $R(g)$ ei minimeeri. Suurused $yf(x)$, mida ϕ-riski arvutamisel kasutame, on suured siis, kui funktsiooni f poolt defineeritud klassifitseerija otsus $\mathrm{sgn}\big(f(x)\big)$ ja objekti tegelik klassikuuluvus y ühtivad. Funktsioon e^t on aga kasvav funktsioon. Järelikult vastaks funktsiooni e^t korral õigetele klassifitseerimistele suurem väärtus, kui klassifitseerimisvigadele. Sellise funktsiooni põhjal defineeritud ϕ-riski minimeerimisel leitud klassifitseerija käitumisviis on vastupidine sellele, mida healt klassifitseerijalt ootame.

3.2 Algoritmi kirjeldus

AdaBoosti tööpõhimõtet kirjeldab algoritm 1 [6, lk 2].

Olgu $\mathcal{H}$ klassifitseerijate hulk, millelt nõuame, et sellest treeningandmete põhjal ERM-printsiibil leitud klassifitseerija annaks juhuslikust klassifitseerimisest kasvõi natuke paremaid tulemusi [11, lk 125]. Selline klassifitseerijate hulk ei pruugi treeningandmete põhjal leida kuigi head klassifitseerijat, teisalt teeb $\mathcal{H}$ lihtsus selle treeningandmetele kergesti rakendatavaks. Seda $\mathcal{H}$ omadust kasutab AdaBoost - hulka $\mathcal{H}$ kuuluvaid klassifitseerijaid rakendatakse treeningvalimile korduvalt. Iga kord, kui treeningvalimile $\mathcal{H}$ liiget rakendatakse, modifitseeritakse treeningvalimit allpool kirjeldatud moel. Nii saadakse klassifitseerijate järjend $h_1, ..., h_T$. Lõpuks moodustakse klassifitseerijatest $h_1, ..., h_T$ funktsioon

$$f_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x), \tag{3.8}$$

mille põhjal defineeritakse klassifitseerija

$$g_T(x) \coloneqq \mathrm{sgn}(f_T). \tag{3.9}$$

Suurused $\alpha_1, ..., \alpha_T$ arvutatakse AdaBoosti algoritmi poolt. Iga α_t näitab, kui suur on sellele vastava klassifitseerija h_t mõju lõpliku otsuse tegemise juures. Mida

Eelnevalt oleme välja valinud lihtsate klassifitseerijate hulga $\mathcal{H}$.

Sisend : Treeningvalim $(x_1, y_1), ..., (x_n, y_n)$ ja iteratsioonide arv T.

Alusta kaaludega $w_i^{(1)} = \frac{1}{n}, \quad i = 1, ..., n.$

Iga $t = 1, ..., T$ **korral:**

1. Leia ERM-printsiibil $h_t \in \{-1, 1\}$ kasutades treeningvalimit kaaludega $w_i^{(t)}$.
2. Leia kaalutud treeningviga $\epsilon_t = \sum_{i=1}^{n} w_i^{(t)} I_{y_i \neq h_t(x_i)}$.
3. Arvuta $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$
4. Arvuta $w_i^{(t+1)} \leftarrow w_i^{(t)} e^{-\alpha_t y_i h_t(x_i)} \quad \forall i = 1, ..., n;$
   normaliseeri, et kehtiks $\sum_{i=1}^{n} w_i^{(t+1)} = 1.$

Väljund: $f_T(x) \coloneqq \sum_{i=1}^{t} \alpha_t h_t(x)$

Algoritm 1: Diskreetne AdaBoost

täpsem on klassifitseerija h_t, seda suurem on sellele vastav α_t. Eeldame, et kui lihtsate klassifitseerijate hulka $\mathcal{H}$ kuulub klassifitseerija h, siis kuulub hulka $\mathcal{H}$ ka sellega täpselt vastupidiseid otsuseid tegev klassifitseerija $-h$. Sellisel juhul kehtib ERM-printsiibil leitud klassifitseerija h_t korral alati $\epsilon_t \leqslant \frac{1}{2}$, mistõttu kehtib alati $\alpha_t \geqslant 0$. Seetõttu on loomulik nimetada suurust α_t klassifitseerija h_t kaaluks. Klassifitseerija g_T kasutamist võime vaadelda kaalutud enamushääletusena: T-liikmeline „komitee" [8, lk 299] väljastab otsuse, mille paika panemisel on suurem mõju neil, kelle otsusel on suurem kaal, ent lõppotsuse vormimisel arvestatakse ometi kõigi arvamusega.

Igal algoritmi tsüklisammul seatakse igale objektile (x_i, y_i), $i = 1, ..., n$ vastavusse kaal $w_i^{(t+1)}$, $i = 1, ..., n$. Enne esimest tsüklisammu väärtustatakse kõigi objektide kaalud $w_i^{(1)} = \frac{1}{n}$, st klassifitseerija h_1 treenitakse esialgse treeningvalimi põhjal. Igal järgneval sammul $t = 2, ..., T$ kasutatakse klassifitseerija h_{t-1}

otsuste põhjal modifitseeritud kaalusid. Sammul t arvutatakse sammu $t+1$ jaoks kehtivad kaalud $w_i^{(t+1)}$ põhimõttel, et h_t poolt valesti klassifitseeritud objektidele (x_i, y_i) vastaks sammul $t+1$ endisest suurem kaal ja h_t poolt korrektselt klassifitseeritud objektidele vastaks järgmisel sammul väiksem kaal. Täpsemalt, objektile (x_i, y_i) vastavale kaalule omistatakse sammul t väärtus

$$w_i^{(t+1)} \leftarrow \begin{cases} w_i^{(t)} e^{-\alpha_t} & \text{kui } h_t(x_i) = y_i, \\ w_i^{(t)} e^{\alpha_t} & \text{kui } h_t(x_i) \neq y_i. \end{cases}$$

Kuna klassifitseerijate kaalude α_t kehtib alati $\alpha_t \geqslant 0$, siis kehtib iga t ja i korral $w_i^{(t)} e^{-\alpha_t} \leqslant w_i^{(t)}$ ja $w_i^{(t)} e^{\alpha_t} \geqslant w_i^{(t)}$. Seega omandab raskesti klassifitseeritav ja seetõttu klassifitseerijate h_t poolt tihti valesti klassifitseeritav objekt iteratsioonide arvu t kasvades aina suurema kaalu, mistõttu on sellel objektil iga järgmise klassifitseerija treenimise juures suurem mõju.

On näidatud [2], et sobivalt treeningvalimi mahtu n silmas pidades valitud iteratsioonide arvu T ja lõpliku VC dimensiooniga $\mathcal{H}$ korral kehtib AdaBoosti väljundi korral

$$R(g_n) \xrightarrow{n} R^* \quad \text{p.k.} \tag{3.10}$$

Kui klassifitseerimisalgoritmi korral kehtib koondumine kujul (3.10), siis öeldakse, et see algoritm on tugevalt mõjus [9, lk 26]. Seega, kui iteratsioonide arv valitakse teataval treeningvalimi mahust sõltuval viisil, siis on AdaBoost tugevalt mõjus.

AdaBoosti juures kasutatavaid lihtsaid klassifitseerijaid $h \in \mathcal{H}$ nimetatakse ka rusikareegliteks (ingl *weak learner*). Kaheklassilisel juhul kasutatakse rusikareeglitena sageli kahendpuid (vt näidet (2.2)).

Sageli kasutatakse igal iteratsioonisammul ERM-printsiibil optimaalseima klassifitseerija $\hat{h}_t \in \mathcal{H}$ otsimise asemel mõnda lähenemist, mis annab kergema vaevaga mõne klassifitseerijale $\hat{h}_t$ piisavalt lähedaste omadustega klassifitseerija. Näiteks,

kui rusikareeglite hulgana $\mathcal{H}$ kasutatakse näites (2.2) tutvustatud kahendpuid, leitakse igal AdaBoosti iteratsioonisammul sobivaim kahendpuu-tüüpi klassifitseerija Gini indeksi abil [8, lk 317-318].

3.3 Eksponentsiaalne kadu ja AdaBoost

Näitame, et AdaBoost minimeerib teataval moel empiirilist ϕ-riski üle klassi $\mathcal{F}$, kus

$$\phi(t) = e^{-t}$$

ja

$$\mathcal{F} = \{\sum_{t=1}^{T} \alpha_t h_t : \alpha_t \in \mathbb{R}^+, h_t \in \mathcal{H}\},$$

kus $\mathcal{H}$ on meie poolt välja valitud lihtsate klassifitseerijate hulk. Eespool näitasime, et funktsioon e^{-t} on kalibreeritud (vt näidet 3.1). Seega omab funktsiooni e^{-t} põhjal defineeritud ϕ-riski minimeerimine mõtet.

Empiirilise ϕ-riski

$$\hat{R}_\phi = \frac{1}{n} \sum_{i=1}^{n} e^{-y_i f(x_i)} \tag{3.11}$$

minimeerimine üle funktsioonide klassi $\mathcal{F}$ võib olla väga keerukas [8, lk 304]. Sellise $f \in \mathcal{F}$, millel põhinev klassifitseerija $\mathrm{sgn}(f)$ on parimate klassifitseerimisomadustega, hindamiseks kasutab AdaBoost lähenemist, kus t-ndal tsüklisammul lisatakse klassifitseerijate järjendile $h_1, ..., h_{t-1}$ klassifitseerija h_t, mis on t-nda tsüklisammu alguses kehtivate valimikaalude $w_i^{(t)}$ juures optimaalseim. Seejuures ei muudeta sellele eelnevatel sammudel leitud klassifitseerijaid $h_1, ..., h_{t-1}$ ja neile vastavaid kaalusid $\alpha_1, ..., \alpha_{t-1}$.

Teoreem 3.3 (vt [8, lk-d 305-306])**.** *AdaBoost leiab igal sammul klassifitseerija*

$h_t \in \mathcal{H}$ ja kordaja α_t nii, et

$$(\alpha_t, h_t) = \arg\min_{\alpha,h} \sum_{i=1}^{n} \exp\big[-y_i\big(f_{t-1}(x_i) + \alpha h(x_i)\big)\big], \tag{3.12}$$

kus

$$f_{t-1} = \sum_{s=1}^{t-1} \alpha_s h_s,\ f_0 = 0.$$

Tõestus. (vt [8, lk-d 305-306], [10, lk 121]) Veendume, et AdaBoost käitub igal sammul tõesti nii nagu võrduses (3.12) näidatud. Selleks paneme tähele, et võrduse (3.12) võime avaldada

$$(\alpha_t, h_t) = \arg\min_{\alpha,h} \sum_{i=1}^{n} W_i^{(t)} \exp\big(-\alpha y_i h(x_i)\big), \tag{3.13}$$

kus $W_i^{(t)} = \exp\big(-y_i f_{t-1}(x_i)\big)$. Suurus $W_i^{(t)}$ ei sõltu t-ndal iteratsioonisammul leitavast parameetrist α_t ja klassifitseerijast h_t. Suurus $W_i^{(t)}$ sõltub varem leitud funktsioonist f_{t-1}. Võrdleme selliselt defineeritud suurust $W_i^{(t)}$ AdaBoosti algoritmi t-ndasse tsüklisammu sisenemisel objektile (x_i, y_i) vastava kaaluga $w_i^{(t)}$.

Selleks paneme tähele, et t-ndal iteratsioonisammul saab i-nda objekti kaal väärtuse

$$w_i^{(t+1)} \leftarrow \frac{1}{Z_t} w_i^{(t)} e^{-\alpha_t y_i h_t(x_i)}, \tag{3.14}$$

kus Z_t on normeeriv konstant, mis tagab, et t-ndal iteratsioonisammul objektide jaoks arvutatud kaalude summa oleks 1,

$$Z_t \coloneqq \sum_{i=1}^{n} w_i^{(t+1)} e^{-\alpha_t y_i h_t(x_i)}. \tag{3.15}$$

AdaBoosti t-ndasse tsüklisammu sisenedes objektile i vastav kaal $w_i^{(t)}$ avaldub (3.14) järgi eelnevatel iteratsioonisammudel leitud suuruste kaudu

$$w_i^{(t)} = \frac{e^{-\alpha_{t-1} y_i h_{t-1}(x_i)} e^{-\alpha_{t-2} y_i h_{t-2}(x_i)} \ldots e^{-\alpha_1 y_i h_1(x_i)}}{Z_{t-1} Z_{t-2} \ldots Z_1} w_i^{(1)} =$$

$$= \frac{e^{-\alpha_{t-1}y_i h_{t-1}(x_i) - \alpha_{t-2}y_i h_{t-2}(x_i) - \ldots - \alpha_1 y_i h_1(x_i)}}{Z_{t-1}Z_{t-2}\ldots Z_1 n} =$$
$$= \frac{e^{-y_i f_{t-1}(x_i)}}{Z_{t-1}Z_{t-2}\ldots Z_1 n}, \tag{3.16}$$

kus $w_i^{(1)} = \frac{1}{n}$ on objekti i kaal enne AdaBoosti esimesse tsüklisammu sisenemist. Võrduses (3.13) esinevad kordajad $W_i^{(t)}$ saab nüüd neile vastavate objektide t-ndasse iteratsioonisammu sisenemisel kehtivate kaalude kaudu avaldada

$$W_i^{(t)} = \exp\bigl(-y_i f_{t-1}(x_i)\bigr) = Z_{t-1}Z_{t-2}\ldots Z_1 n w_i^{(t)}. \tag{3.17}$$

Kõik suurused $W_i^{(t)}$ on leitavad objektide kaalusid $w_i^{(t)}$ ühe ja sama kordajaga korrutamise teel, see kaalude omavahelisi suhteid ei muuda.

Näitame nüüd, et võrdust (3.13) rahuldava $h_t \in \mathcal{H}$ leidmiseks piisab iga $\alpha > 0$ korral sellest, kui lahendada

$$h_t = \arg\min_h \sum_{i=1}^{n} w_i^{(t)} I_{y_i \neq h(x_i)}, \tag{3.18}$$

ehk (3.13) minimeerimine on samaväärne kaalutud treeningvea minimeerimisega. Selles veendumiseks avaldame võrduses (3.13) minimeeritava suuruse järgnevalt:

$$\begin{aligned}
\sum_{i=1}^{n} W_i^{(t)} e^{-\alpha y_i h(x_i)} &= e^{-\alpha} \sum_{y_i = h(x_i)} W_i^{(t)} + e^{\alpha} \sum_{y_i \neq h(x_i)} W_i^{(t)} = \\
&= e^{-\alpha} \sum_{i=1}^{n} W_i^{(t)} I_{y_i = h(x_i)} + e^{\alpha} \sum_{i=1}^{n} W_i^{(t)} I_{y_i \neq h(x_i)} = \\
&= e^{-\alpha} \sum_{i=1}^{n} W_i^{(t)} \left(1 - I_{y_i \neq h(x_i)}\right) + e^{\alpha} \sum_{i=1}^{n} W_i^{(t)} I_{y_i \neq h(x_i)} = \\
&= \left(e^{\alpha} - e^{-\alpha}\right) Z_{t-1}\ldots Z_1 n \sum_{i=1}^{n} w_i^{(t)} I_{y_i \neq h(x_i)} + e^{-\alpha} \sum_{i=1}^{n} W_i^{(t)}.
\end{aligned}$$

Nii avaldatud kujul suuruse minimeerimine ei sõltu seal kaalutud treeningvea ees olevast kordajast ja klassifitseerijast h mitte sõltuvast liidetavast, seega on võrdust (3.13) rahuldava $h_t \in \mathcal{H}$ leidmine iga α korral tõesti kaalutud treeningvea minimeerimisega samaväärne.

Fikseerime $h_t \in \mathcal{H}$, mis on defineeritud seosega (3.18), st h_t on leitud kaalutud treeningandmete põhjal ERM-printsiibil. Leiame nüüd kordaja α, mis rahuldab seost (3.13), ehk leiame

$$\alpha_t = \arg\min_{\alpha} \sum_{i=1}^{n} w_i^{(t)} \exp\big(-\alpha y_i h_t(x_i)\big), \tag{3.19}$$

Selleks vaatleme minimeeritavat summat argumendi α funktsioonina:

$$f(\alpha) = \sum_{i=1}^{n} w_i^{(t)} \exp\big(-\alpha y_i h_t(x_i)\big).$$

Paneme tähele, et nii defineeritud $f(\alpha)$ on rangelt kumerate funktsioonide summa ja on seega ka ise kogu oma määramispiirkonnas rangelt kumer. Seega pole funktsioonil $f(\alpha)$ üle ühe ekstreemumi ja kui ekstreemum leidub, on see miinimum. Miinimumi leidmiseks leiame $f(\alpha)$ tuletise $f'(\alpha)$,

$$f'(\alpha) = \sum_{i=1}^{n} -y_i h_t(x_i) w_i^{(t)} \exp\big(-\alpha y_i h_t(x_i)\big),$$

ning võrdsustame selle nulliga. Saame:

$$\begin{aligned}
&\sum_{i=1}^{n} -y_i h_t(x_i) w_i^{(t)} \exp\big(-\alpha y_i h_t(x_i)\big) = 0 \quad \Longrightarrow \\
\Longrightarrow \quad & e^{\alpha} \sum_{h_t(x_i)\neq y_i} w_i^{(t)} + e^{-\alpha} \sum_{h_t(x_i)=y_i} w_i^{(t)} = 0 \quad \Longrightarrow \\
\Longrightarrow \quad & e^{\alpha}\epsilon_t - e^{-\alpha}(1-\epsilon_t) = 0 \quad \Longrightarrow \\
\overset{|\cdot e^{\alpha}}{\Longrightarrow} \quad & e^{2\alpha}\epsilon_t - (1-\epsilon_t) = 0 \quad \Longrightarrow \\
\Longrightarrow \quad & e^{2\alpha}\epsilon_t = 1-\epsilon_t \quad \Longrightarrow \\
\overset{\ln(\cdot)}{\Longrightarrow} \quad & 2\alpha + \ln\epsilon_t = \ln(1-\epsilon_t) \quad \Longrightarrow \\
\Longrightarrow \quad & \alpha = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t},
\end{aligned}$$

kus ϵ_t tähistab h_t kaalutud treeningviga. Tähistame funktsiooni $f(\alpha)$ miinimumi $\alpha_t \coloneqq \alpha$.

Eespool näitasime, et seoses (3.13) esinevad suurused $W_i^{(t)}$ on proportsionaalsed objektide kaaludega $w_i^{(t)}$. Kuna ka klassifitseerija h_t ja kordaja α_t leidsime samal moel, kui algoritmis 1, on ka t-ndal iteratsioonisammul arvutatavad suurused $W_i^{(t+1)}$ proportsionaalsed kaaludega $w_i^{(t+1)}$. Seega on AdaBoosti tsüklisamm võrdväärne seose (3.12) lahendamisega. □

Näitasime, et AdaBoost hindab empiirilist ϕ-riski (3.11) minimeerivat funktsiooni f_T järk-järgult.

3.4 AdaBoosti treeningviga

AdaBoosti abil leitud funktsiooni f_T poolt defineeritud klassifitseerija $\mathrm{sgn}(f_T)$ treeningviga avaldub

$$R_n\big(\mathrm{sgn}(f_T)\big) = \frac{1}{n}\sum_{i=1}^{n} I_{\mathrm{sgn}(f_T(x_i))\neq y_i}. \tag{3.20}$$

Uurime, kuidas AdaBoosti treeningviga iteratsioonide arvu t kasvades käitub. Järgmine teoreem annab AdaBoosti väljundi treeningveale iteratsioonide arvust sõltuva ülemise tõkke.

Teoreem 3.4 ([5, lk 127])**.** *Olgu $f_T = \alpha_1 h_1 + ... + \alpha_T h_T$ AdaBoosti väljund ja $\epsilon_1, ..., \epsilon_T$ rusikareeglite kaalutud treeningvead. Siis kehtib $\mathrm{sgn}(f_T)$ treeningvea jaoks hinnang*

$$R_n\big(\mathrm{sgn}(f_T)\big) \leqslant \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)}. \tag{3.21}$$

Tõestus. (vt [10, lk 124]) Kasutame seost (3.17), saame:

$$Z_T Z_{T-1} ... Z_1 n w_i^{(T+1)} = \exp\big(-y_i f_T(x_i)\big).$$

Summeerime võrduse mõlemad pooled üle treeningvalimi, saame

$$Z_T ... Z_1 n \sum_{i=1}^{n} w_i^{(T+1)} = \sum_{i=1}^{n} \exp\big(-y_i f_T(x_i)\big) \quad \Longrightarrow$$
$$\Longrightarrow \quad Z_T ... Z_1 n = \sum_{i=1}^{n} \exp\big(-y_i f_T(x_i)\big),$$

sest kaalud $w_i^{(T+1)}$ on normeeritud nii, et nende summa oleks võrdne ühega. Avaldame nüüd normeeriva konstandi Z_t kaalutud treeningvea ϵ_t kaudu.

$$\begin{aligned}
Z_t &= \sum_{i=1}^{n} w_i^{(t)} e^{-\alpha_t y_i h_t(x_i)} = \\
&= e^{-\alpha_t} \sum_{y_i = h_t(x_i)} w_i^{(t)} + e^{\alpha_t} \sum_{y_i \neq h_t(x_i)} w_i^{(t)} = \\
&= e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t = \\
&= \exp\left(-\frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}\right)(1-\epsilon_t) + \exp\left(\frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}\right)\epsilon_t = \\
&= \frac{\sqrt{\epsilon_t}}{\sqrt{1-\epsilon_t}}(1-\epsilon_t) + \frac{\sqrt{1-\epsilon_t}}{\sqrt{\epsilon_t}}\epsilon_t = \\
&= 2\sqrt{\epsilon_t(1-\epsilon_t)}. \qquad (3.22)
\end{aligned}$$

Esitatud seoseid kasutades saame nüüd AdaBoosti väljundi põhjal defineeritud klassifitseerija g_T treeningviga hinnata:

$$\begin{aligned}
R_n(f_T) &= \frac{1}{n}\sum_{i=1}^{n} I_{g_T(x_i)\neq y_i} \leqslant \\
&\leqslant \frac{1}{n}\sum_{i=1}^{n} \exp\big(-y_i f_T(x_i)\big) = \\
&= \prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)}.
\end{aligned}$$

kus esimese võrratuse õigsuses veendumiseks paneme tähele, et indikaatorfunktsioon $I_{g_T(x_i)\neq y_i}$ on samaväärne indikaatorfunktsiooniga $I_{y_i f_T(x_i)\leqslant 0}$. See funktsioon on eksponentfunktsiooniga $\exp\big(-y_i f_T(x_i)\big)$ võrdne vaid siis, kui $y_i f_T(x_i) = 0$; mujal kehtib $\exp\big(-y_i f_T(x_i)\big) > I_{y_i f_T(x_i)\leqslant 0}$. □

Defineerime

$$\gamma_t \coloneqq 1 - 2\epsilon_t, \tag{3.23}$$

mis kirjeldab, kui palju on h_t klassifitseerimisomadustelt juhuslikust klassifitseerijast parem, $\epsilon_t = \frac{1}{2} - \frac{1}{2}\gamma_t$. Avaldame selle suuruse kaudu AdaBoosti väljundi põhjal defineeritud klassifitseerija treeningvea ülemise tõkke:

$$R_n\big(\operatorname{sgn}(f_T)\big) \leqslant \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)} = \prod_{t=1}^{T} \sqrt{1-\gamma_t^2} \leqslant e^{-\frac{1}{2}\sum_{t=1}^{T}\gamma_t^2}. \tag{3.24}$$

Eelneva võrratusterea viimase võrratuse tõestamiseks veendume, et iga t korral kehtib

$$\sqrt{1-\gamma_t^2} \leqslant e^{-\frac{1}{2}\gamma_t^2}. \tag{3.25}$$

Juhul, kui h_t kaalutud treeningviga ϵ_t on võrdne nulliga, kehtib $\gamma_t^2 = 1$ ja võrratus (3.25) kehtib. Juhul, kui $\epsilon_t > 0$, võime kasutada iga positiivse ω korral kehtivat naturaallogaritmi omadust $\ln\omega \geqslant 1 - \frac{1}{\omega}$. Võttes $\omega \coloneqq 1 - \gamma_t^2$ saame:

$$\sqrt{1-\gamma_t^2} = e^{\frac{1}{2}\ln(1-\gamma_t^2)} = e^{-\frac{1}{2}\ln\frac{1}{1-\gamma_t^2}} \leqslant e^{-\frac{1}{2}(1-(1-\gamma_t^2))} = e^{-\frac{1}{2}\gamma_t^2}.$$

Võrratustereast (3.24) järeldub, et kui kehtib $\sum_{t=1}^{T}\gamma_t^2 \to \infty$, siis läheneb klassifitseerija $\operatorname{sgn}(f_T)$ treeningviga iteratsioonide arvu kasvades nullile. Treeningvea nulliks koondumiseks on piisav, kui leidub selline $\gamma > 0$, et iga γ_t korral kehtib $\gamma_t \geqslant \gamma$. AdaBoosti-eelsete *boosting*-algoritmide puhul pidi parameetri γ suurus teada olema juba enne õppimisprotsessi algust. Selle asemel „kohaneb" AdaBoost kaalutud treeningvigadega ϵ_t. AdaBoost on selles mõttes „kohanemisvõimeline" *boosting* (ingl *adaptive boosting*), millest tuleneb ka algoritmi nimi [7].

3.5 Marginaalilävega treeningviga

Kõik AdaBoosti algoritmis esinevad klassifitseerijate h_t kaalud on mittenegatiivsed, $\alpha_t \geqslant 0$. Seetõttu võime funktsiooni f_T normaliseerida, klassifitseerija

$g_T = \mathrm{sgn}(f_T)$ otsused sellest ei muutu. Defineerime:

$$\beta_t \coloneqq \frac{\alpha_t}{\sum\limits_{u=1}^{T} \alpha_u}. \tag{3.26}$$

Niimoodi funktsiooni f_T normaliseerides võime öelda, et AdaBoosti väljund kuulub alati hulka

$$co_T(\mathcal{H}) = \left\{ \sum_{t=1}^{T} \beta_t h_t : \beta_t \geqslant 0, \sum_{t=1}^{T} \beta_t = 1, h_t \in \mathcal{H} \right\}. \tag{3.27}$$

Defineerime i-nda objekti marginaali [10, lk 124]:

$$\rho_i(f_T) \coloneqq \frac{y_i f_T(x_i)}{\sum\limits_{t=1}^{T} \alpha_t} = y_i \left(\sum_{t=1}^{T} \beta_t h_t(x_i) \right). \tag{3.28}$$

Objekti marginaal AdaBoosti väljundi f_T korral kätkeb endas informatsiooni funktsiooni f_T põhjal defineeritud klassifitseerija $\mathrm{sgn}(f_T)$ poolt objekti klassifitseerimisel tehtud otsuse õigsuse ja selle otsuse kindluse kohta. Klassifitseeerija $\mathrm{sgn}(f_T)$ klassifitseerib i-nda objekti õigese klassi parajasti siis, kui marginaal $\rho_i(f_T)$ on positiivne. Marginaali puhul kehtib alati $\rho_i \in [-1, 1]$. AdaBoosti väljundi f_T korral leitud i-nda objekti marginaali absoluutväärtus näitab, millise häälteenamusega „komitee" f_T i-nda objekti klassifitseerimisel otsustas.

Defineerime $0 \leqslant \theta \leqslant 1$ korral funktsiooni f marginaalilävega empiirilise riski [11, lk 128]:

$$R_n^\theta(f) \coloneqq \frac{1}{n} \sum_{i=1}^{n} I_{\rho_i(f) \leqslant \theta}. \tag{3.29}$$

Nimetame suurust θ marginaaliläveks. Marginaalilävega empiirilise riski $R_n^\theta(f)$ korral seatakse kõigile lävest θ väiksematele marginaalidele vastavusse ühikuline kahju. Nimetame treeningvalimi põhjal arvutatud marginaalilävega empiirilist riski marginaalilävega treeningveaks.

Esitame teoreemis 3.4 esitatud hinnangust üldisema hinnangu.

Teoreem 3.5 ([11, lk 128])**.** *Olgu $f_T \in co_T(\mathcal{H})$ AdaBoosti väljund ja $\epsilon_1, ..., \epsilon_T$ rusikareeglite kaalutud treeningvead. Siis kehtib iga $\theta \geqslant 0$ korral funktsiooni f_T marginaalilävega treeningvea jaoks hinnang*

$$R_n^\theta(f_T) \leqslant \prod_{t=1}^{T} (1-\gamma_t)^{\frac{1-\theta}{2}} (1+\gamma_t)^{\frac{1+\theta}{2}}. \tag{3.30}$$

Tõestus. (vt [11, lk-d 128-129]) Alustuseks tuletame meelde, et t-ndal iteratsioonisammul i-ndale objektile vastavusse seatava kaalu võib esitada (vt (3.16)):

$$w_i^{(t+1)} = \frac{e^{-y_i f_t(x_i)}}{n \prod_{t=1}^{T} Z_t}, \tag{3.31}$$

Treeningvalimi i-nda objekti marginaali definitsioonist saame:

$$\begin{aligned}
\rho_i(f_T) \leqslant \theta \quad &\Longleftrightarrow \quad y_i f_T(x_i) \leqslant \theta \sum_{t=1}^{T} \alpha_t \quad \Longleftrightarrow \\
&\Longleftrightarrow \quad \exp\left(y_i f_T(x_i)\right) \leqslant \exp\left(\theta \sum_{t=1}^{T} \alpha_t\right) \quad \Longleftrightarrow \\
&\Longleftrightarrow \quad 1 \leqslant \exp\left(\theta \sum_{t=1}^{T} \alpha_t\right) \exp\left(-y_i f_T(x_i)\right),
\end{aligned}$$

millest järeldub:

$$\exp\left(\theta \sum_{t=1}^{T} \alpha_t - y_i f_T(x_i)\right) \geqslant I_{\rho_i(f_T) \leqslant \theta}. \tag{3.32}$$

Vastesitatud võrratusest saame marginaalilävega treeningvea definitsioonist (3.29) ja seosest (3.31):

$$\begin{aligned}
R_n^\theta(f_T) &= \frac{1}{n} \sum_{i=1}^{n} I_{\rho_i(f_T) \leqslant \theta} \leqslant \\
&\leqslant \frac{1}{n} \sum_{i=1}^{n} \exp\left(\theta \sum_{t=1}^{T} \alpha_t\right) \exp\left(-y_i f_T(x_i)\right) = \\
&= \frac{1}{n} \exp\left(\theta \sum_{t=1}^{T} \alpha_t\right) \left(n \prod_{t=1}^{T} Z_t\right) \sum_{i=1}^{n} w_i^{(t+1)} =
\end{aligned}$$

$$= \left(\prod_{t=1}^{T} Z_t \right) \exp \left(\theta \sum_{t=1}^{T} \alpha_t \right),$$

kus viimase võrduse saamiseks paneme tähele, et $\sum_{i=1}^{n} w_i^{(t+1)} = 1$. Seost $\epsilon_t = \frac{1}{2} - \frac{1}{2}\gamma_t$ (vt (3.23)) ja α_t definitsiooni kasutades avaldame:

$$\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t} = \ln \frac{\sqrt{1+\gamma_t}}{\sqrt{1-\gamma_t}}.$$

Tuletame veel meelde, et normeeriva konstandi Z_t võime avaldada (vt (3.22)):

$$Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)} = \sqrt{1-\gamma_t^2}.$$

Nüüd jõuamegi soovitud tulemuseni:

$$\begin{aligned} R_n^{\theta}(f_T) &\leqslant \left(\prod_{t=1}^{T} Z_t \right) \exp \left(\theta \sum_{t=1}^{T} \alpha_t \right) = \\ &= \prod_{t=1}^{T} \sqrt{1-\gamma_t^2} \exp \left(\sum_{t=1}^{T} \ln \frac{(1+\gamma_t)^{\frac{\theta}{2}}}{(1-\gamma_t)^{\frac{\theta}{2}}} \right) = \\ &= \prod_{t=1}^{T} \left((1-\gamma_t)^{\frac{1}{2}} (1+\gamma_t)^{\frac{1}{2}} \right) \prod_{t=1}^{T} \left(\frac{(1+\gamma_t)^{\frac{\theta}{2}}}{(1-\gamma_t)^{\frac{\theta}{2}}} \right) = \\ &= \prod_{t=1}^{T} (1-\gamma_t)^{\frac{1-\theta}{2}} (1+\gamma_t)^{\frac{1+\theta}{2}}. \end{aligned}$$

□

Erijuhul $\theta = 0$ saame vastselt tõestatud hinnangust eespool esitatud hinnangu (3.21).

3.6 Hinnang AdaBoosti riskile

Defineerime $0 \leqslant \theta \leqslant 1$ korral funktsiooni [11, lk 127]

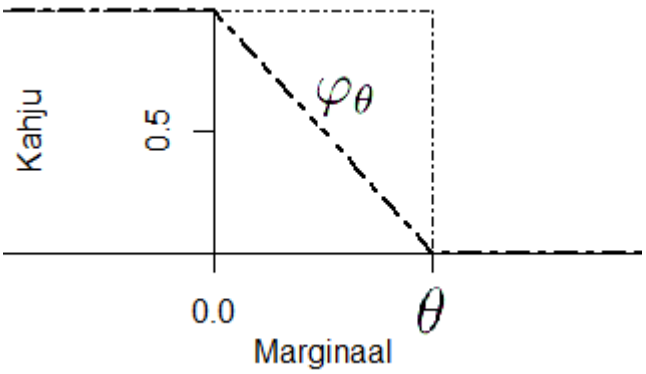
$$\varphi_\theta(t) = \begin{cases} 1 & \text{kui } t \leqslant 0, \\ 1 - \frac{t}{\theta} & \text{kui } 0 < t \leqslant \theta, \\ 0 & \text{kui } t > \theta. \end{cases} \tag{3.33}$$

Vaatleme empiirilist ϕ-riski (vt (3.7)), kus $\phi(t) = \varphi_\theta(t)$:

$$A_n^\theta(f) = \frac{1}{n}\sum_{i=1}^{n} \varphi_\theta\big(y_i f(x_i)\big). \tag{3.34}$$

Paneme tähele, et kehtib võrratus

$$A_n^\theta(f) \leqslant R_n^\theta(f).$$



Joonis 3.1. Funktsioonid φ_θ ja $I_{(-\infty,\theta)}$.

Tõepoolest, marginaaliga treeningvea R_n^θ (vt (3.29)) arvutamisel seatakse kõigile suurusest θ väiksema marginaaliga objektidele vastavusse ühikuline kahju; suuruse A_n^θ arvutamisel kaasneb lävest θ väiksema aga nullist suurema marginaaliga objektidega ühest väiksem „karistus" (vt joonist 3.1). Paneme veel tähele, et marginaaliläve $\theta = 0$ korral on suurused $A_n^\theta(f)$ ja $R_n^\theta(f)$ võrdsed klassifitseerimisvigade osakaalu ehk empiirilise riskiga 0-1 kaofunktsiooni korral (vt (2.2)).

Järgmine teoreem annab ülemise tõkke AdaBoosti väljundi f_T põhjal defineeritud klassifitseerija $\operatorname{sgn}(f_T)$ riskile.

Teoreem 3.6 ([10, lk 125])**.** *Olgu $\theta > 0$. Siis kehtib iga täisarvu n ja iga $f \in co_T(\mathcal{H})$ korral n-elemendisele treeningvalimi puhul tõenäosusega vähemalt $1 - \delta$*

hinnang

$$R\big(\mathrm{sgn}(f)\big) \leqslant A_n^\theta(f) + \frac{8}{\theta}\sqrt{\frac{2V_{\mathcal{H}}\ln(n+1)}{n}} + \sqrt{\frac{ln\frac{2}{\delta}}{2n}}, \tag{3.35}$$

kus $V_{\mathcal{H}}$ on lihtsate klassifitseerijate klassi $\mathcal{H}$ VC dimensioon.

Arvestades, et kehtib $A_n^\theta(f) \leqslant R_n^\theta(f)$, saame suuruse $A_n^\theta(f)$ hindamiseks kasutada teoreemis 3.5 esitatud hinnangut.

Teoreemis 3.6 esitatud võrratuse (3.35) parema poole teine ja kolmas liidetav lähenevad fikseeritud suuruse δ, lõpliku $V_{\mathcal{H}}$ ja fikseeritud marginaaliläve θ korral treeningvalimi kasvades nullile.

Peatükk 4

Katsed AdaBoostiga

Viimases peatükis uurime, kuidas tuli AdaBoost toime ühe konkreetse kaheklassilise klassifitseerimisprobleemiga. AdaBoosti andmetele rakendamiseks kasutati statistikatarkvara R paketti *ada* [4]. Pakett *ada* kasutab rusikareeglitena näites 2.2 kirjeldatud kahendpuu-tüüpi klassifitseerijaid, mis on teostatud statistikatarkvara R paketina *rpart* [15]. Pakett *rpart* võimaldab määrata rusikareeglitena kasutatavate kahendpuude maksimaalse sügavuse.

Genereeritud andmetele rakendati AdaBoosti kolme erineva rusikareeglite klassi korral. Rusikareeglite klassidena olid kasutusel maksimaalselt kolmetasemeliste kahendpuu-klassifitseerijate hulk $\mathcal{P}_2$, maksimaalselt viietasemeliste kahendpuu-klassifitseerijate hulk $\mathcal{P}_4$ ja kõigi kahendpuu-klassifitseerijate hulk $\mathcal{P}$.

4.1 Andmete genereerimine, Bayesi klassifitseerija ja Bayesi risk

Andmed genereeriti Hastie jt tehisõppe-alases raamatus kirjeldatud viisil [8, lk 17]. Nimetatud raamatus kasutatakse sel moel genereeritud andmeid läbivalt, andmetele rakendatakse raamatus mitmesuguseid klassifitseerimismeetodeid, ent mitte *boosting*-algoritme.

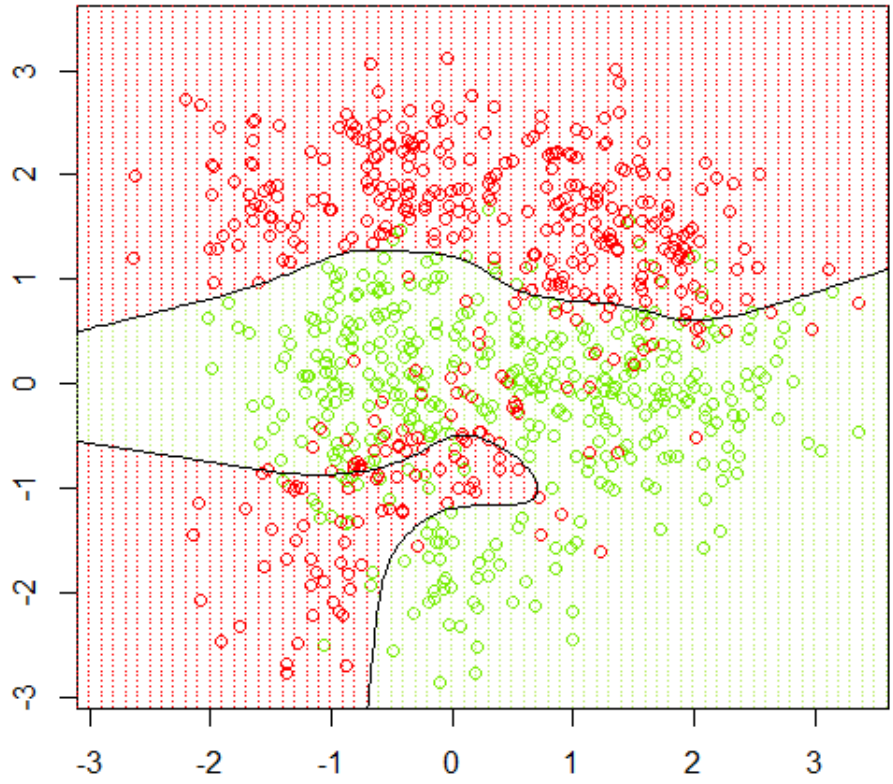
Katsealune klassifitseerimisprobleem oli kaheklassiline. Klassideks olid „roheline" ja „punane", st $\mathcal{Y} = \{$„roheline", „punane"$\}$. Klassid „roheline" ja „punane" kodeeriti vastavalt -1 ja 1. Tunnusvektor x oli kahemõõtmeline, tunnused tähistati X_1 ja X_2. Andmete genereerimisel kasutati kõigepealt kahemõõtmelist normaaljaotust, kus tunnuse X_1 keskväärtus oli 1, tunnuse X_2 keskväärtus 0 ja kovariatsioonimaatriksiks ühikmaatriks $\boldsymbol{I}$. Sellisest jaotusest genereeriti kümme vektorit, st genereeriti kahemõõtmelised vektorid $\boldsymbol{\mu_1}, ..., \boldsymbol{\mu_{10}}$, mille puhul kehtis:

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \boldsymbol{I} \right]. \tag{4.1}$$

Seejärel kasutati neid vektoreid klassi „roheline" esindajate genereerimisel. Täpsemalt, loodi 400 objekti klassikuuluvusega $y = -1$, mille tunnusvektor x genereeriti kahemõõtmelisest normaaljaotusest $\mathcal{N}_2(\boldsymbol{\mu_k}, \boldsymbol{I}/5)$, kus keskväärtus $\boldsymbol{\mu_k}$ valiti iga objekti jaoks vektorite $\boldsymbol{\mu_1}, ..., \boldsymbol{\mu_{10}}$ seast juhuslikult.

Klassi „punane" esindajate genereerimiseks kasutati samasugust lähenemist. Keskväärtustena kasutatud vektorite $\boldsymbol{\mu_{11}}, ..., \boldsymbol{\mu_{20}}$ genereerimiseks kasutati kahemõõtmelist normaaljaotust, mille tunnuste keskväärtused olid võrreldes jaotusega (4.1) vahetatud, st vektorite $\boldsymbol{\mu_{11}}, ..., \boldsymbol{\mu_{20}}$ puhul kehtis:

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \boldsymbol{I} \right].$$



Joonis 4.1. Treeningvalim ja Bayesi otsustuspiirkonnad

„Punaseid" punkte genereeriti sama palju, kui „rohelisi". Seega genereeriti 400 objekti, mille korral $y = 1$ ja tunnusvektor x oli jaotusest $\mathcal{N}_2(\boldsymbol{\mu_k}, \boldsymbol{I}/5)$, kus keskväärtus $\boldsymbol{\mu_k}$ valiti iga objekti jaoks juhuslikult vektorite $\boldsymbol{\mu_{11}}, ..., \boldsymbol{\mu_{20}}$ seast.

Nii saadi 800 objektiga treeningvalim, millesse kuulus võrdselt „punase" ja „rohelise" klassi esindajaid. Kuna juhusliku vektori (X, Y) jaotus oli teada, sai leida Bayesi klassifitseerija otsustuspiirkonnad (vt joonist 4.1). Bayesi klassifitseerija otsustuspiirkondade vahelisel piiril kehtib $\eta(x) = \frac{1}{2}$.

Monte-Carlo meetodil leitud hinnang Bayesi riskile oli $0{,}159$. Riski hindamiseks

genereeriti kolm valimit, mis kõik koosnesid 50000-st „punasest" ja 50000-st „rohelisest" punktist. Kõigi kolme valimi korral leiti Bayesi klassifitseerija empiiriline risk. Hinnang Bayesi riskile leiti kolme empiirilise riski aritmeetilise keskmise arvutamise teel.

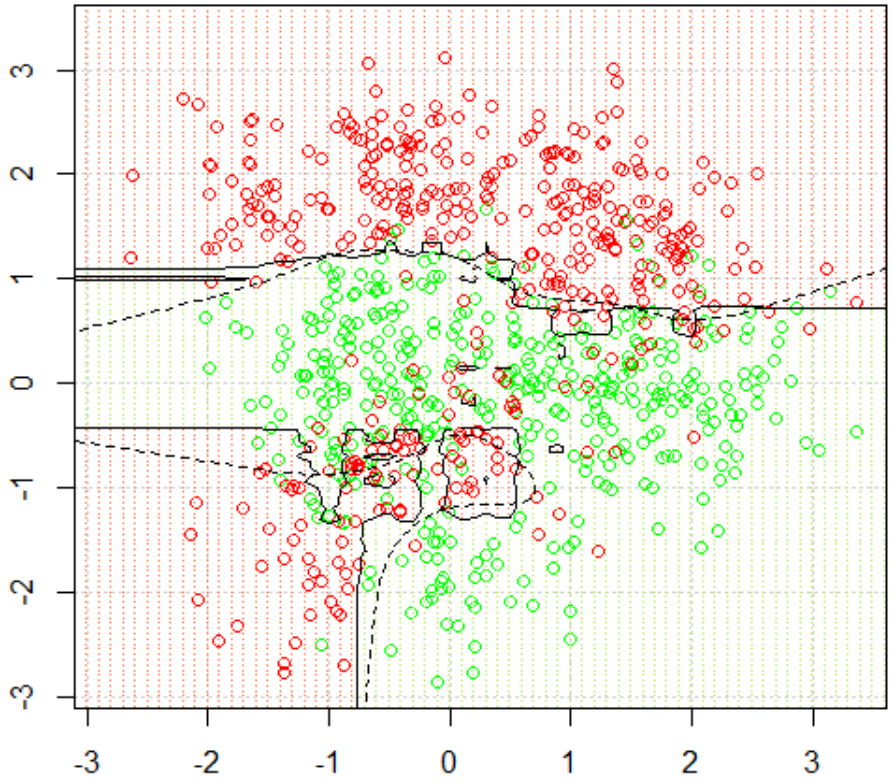
4.2 AdaBoosti rakendamine

Treeningvalimile rakendati esmalt AdaBoosti algoritmi, kus rusikareeglina $\mathcal{H}$ oli kasutusel kõigi maksimaalselt kolmetasemeliste kahendpuude hulk $\mathcal{P}_2$. Kuna meid huvitab treeningvea nulliks koondumise kiirus ja AdaBoosti väljundi riski käitumine pärast treeningvea nulliks koondumist, lisati klassifitseerijate järjendile $h_1, ..., h_t$ klassifitseerijaid $h \in \mathcal{H}$ senikaua kui treeningviga nulliks koondus. Esimest korda koondus treeningviga nulliks 2443-ndal iteratsioonisammul. Sellest lähtudes pikendati klassifitseerijate järjendit veel, kuni jõuti 5000-st rusikareeglist h_t koosneva funktsioonini $f_{5000}(x) = \sum_{t=1}^{5000} \alpha_t h_t(x)$. Nii saadi klassifitseerija $g_{5000}(x) = \operatorname{sgn}(f_{5000}(x))$, mille otsustuspiirkonnad on kujutatud joonisel 4.2. Jooniselt näeme, et AdaBoosti väljundi otsustuspiirkonnad matkivad Bayesi klassifitseerija otsustuspiirkondi küllaltki hästi.

Seejärel genereeriti samal moel treeningvalimist suurem testvalim. Mõlema klassi jaotusest genereeriti 10000 objekti. Nii saadi valim $x_1, ..., x_{20000}$, mille objektide tunnusvektorid olid genereeritud sõltumatult. Klassifitseerijat g_{5000} rakendati testvalimile. Testvalimilt arvutatud empiirilist riski (testviga) võime kasutada AdaBoosti väljundi põhjal defineeritud klassifitseerija riski hinnanguna.

Funktsioonide

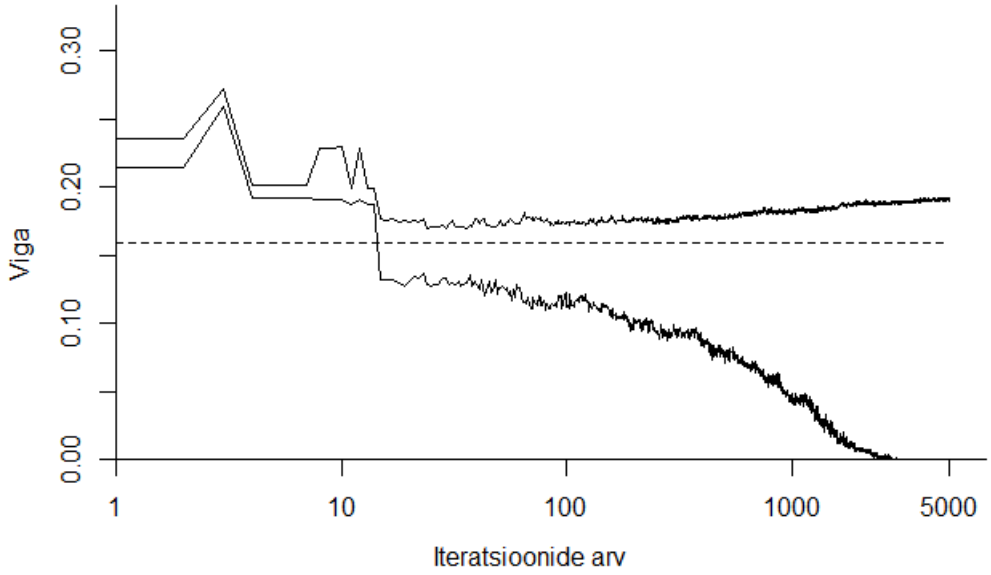
$$f_t = \sum_{i=1}^{t} \alpha_t h_t,\ t = 1, ..., 5000$$



Joonis 4.2. Treeningvalim ja AdaBoosti otsustuspiirkonnad. Iteratsioonide arv T on 5000, rusikareeglite klass on $\mathcal{P}_2$. Katkendliku joonega on tähistatud Bayesi klassifitseerija otsustuspiir.

põhjal defineeritud klassifitseerijate $\mathrm{sgn}(f_t)$ treeningvea ja testvea iteratsioonisammude arvust sõltuvat muutumist on kujutatud joonisel 4.3. Testviga oli väikseim küllaltki väikese iteratsioonide arvu korral - testvea miinimum $0{,}170$ saavutati 31-l iteratsioonisammul. Sealt edasi hakkas testviga vähehaaval kasvama: 100-ndal iteratsioonisammul oli testviga $0{,}174$, 1000-ndal sammul $0{,}181$, 2500-ndal sammul $0{,}189$ ning lõpuks, pärast 5000-ndat sammu $0{,}191$.

Tavaliselt kaasneb klassifitseerija keerukuse kasvu ja sellest tuleneva treeningvea kahanemisega märgatav testvea kasv [8, lk 38], mis on tingitud ülesobitumusest. AdaBoost suudab ülesobitumusest sageli hoiduda. Ka vaatlusaluse näite korral



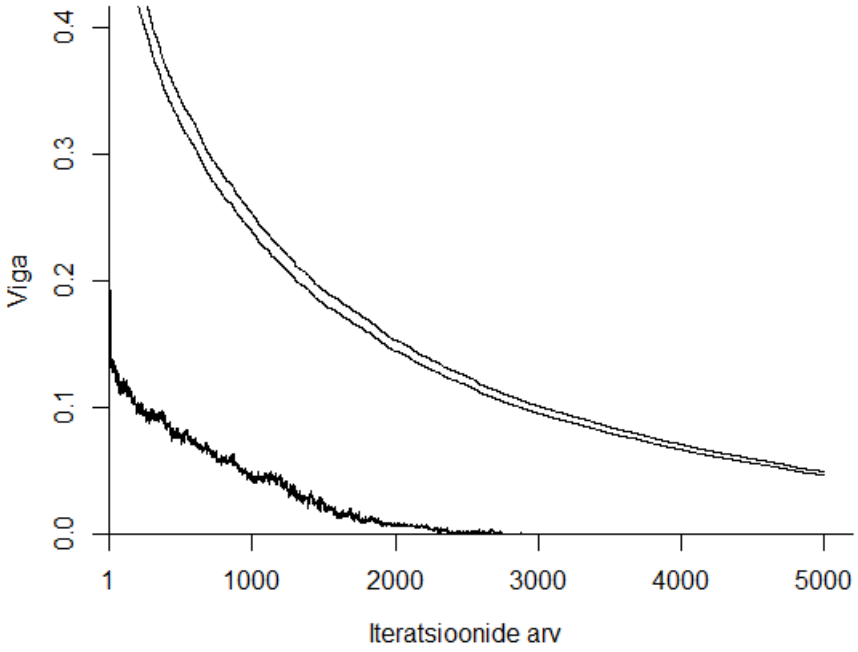
Joonis 4.3. Testvea (ülemine pidev joon) ja treeningvea (alumine pidev joon) sõltuvus iteratsioonide arvust rusikareeglite hulga $\mathcal{P}_2$ korral. Katkendliku joonega on tähistatud Bayesi risk. Iteratsioonide arvu skaala on logaritmiline.

ei esinenud märkimisväärset testvea kasvu isegi pärast seda, kui treeningviga oli juba nulliks koondunud.

4.3 Treeningvea hinnangud

Seejärel kontrolliti teoreemis 3.4 esitatud hinnangut AdaBoosti väljundi põhjal defineeritud klassifitseerija treeningveale:

$$R_n\big(\operatorname{sgn}(f_T)\big) \leqslant \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)}. \tag{4.2}$$



Joonis 4.4. Treeningviga ja treeningvea hinnangud (4.2) (keskmine joon) ja (4.3) (ülemine joon) rusikareeglite hulga $\mathcal{P}_2$ korral.

Joonisel 4.4 on igal sammul t võrreldud klassifitseerija $\operatorname{sgn}(f_t)$ treeningviga selle ülemise tõkkega (4.2). Lisaks on joonisele kantud treeningvea hinnang

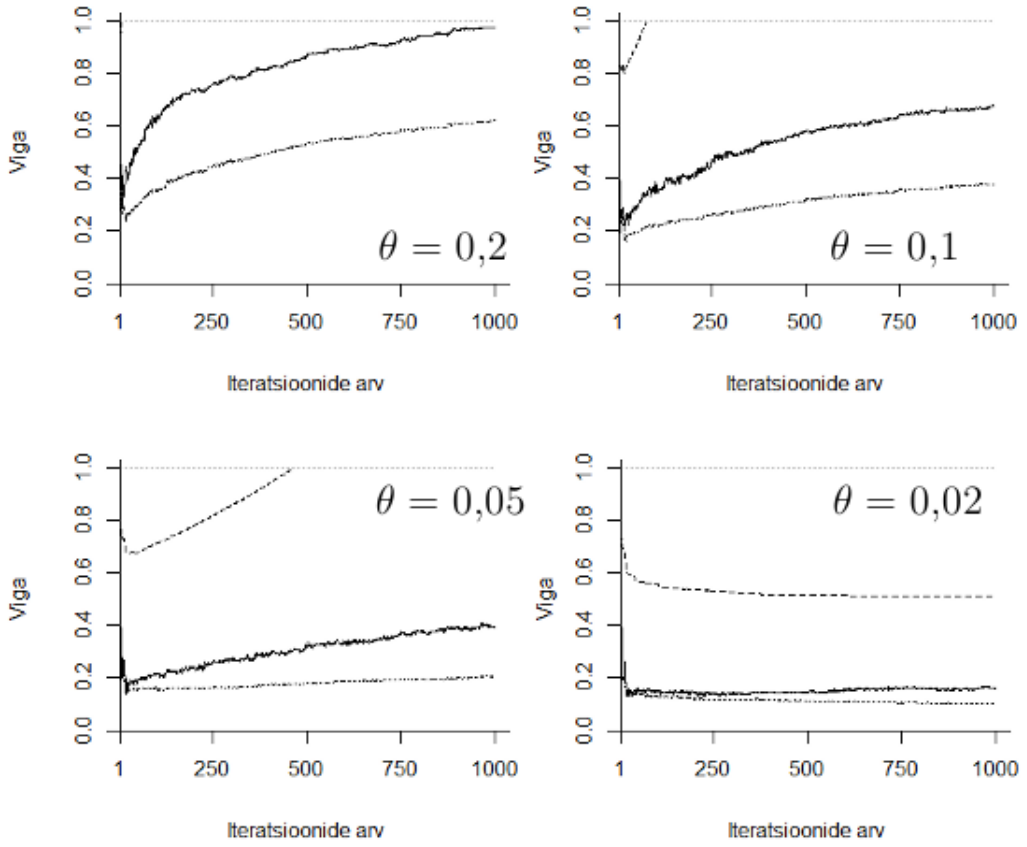
$$R_n(f_T) \leqslant \exp\left[-\frac{1}{2}\sum_{u=1}^{t}\gamma_u^2\right]. \tag{4.3}$$

Jooniselt näeme, et hinnangud treeningveale pole kuigi täpsed. Samas matkivad treeningvea hinnangute kõverad treeningvea joone kuju küllaltki hästi.

4.4 Marginaalilävega treeningviga ja selle hinnang

Uurime, kui täpne on teoreemis 3.5 esitatud hinnang marginaalilävega treeningveale:

$$R_n^{\theta}(f_T) \leqslant \prod_{t=1}^{T}(1-\gamma_t)^{\frac{1-\theta}{2}}(1+\gamma_t)^{\frac{1+\theta}{2}}, \tag{4.4}$$

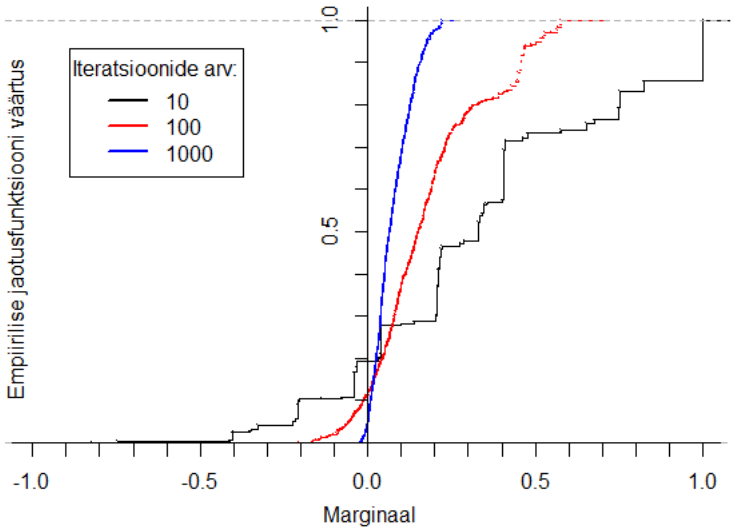


Joonis 4.5. Marginaalilävega treeningviga (keskmine joon), selle hinnang (ülemine joon) ja suurus $A_n^\theta(f_t)$ (alumine joon) rusikareeglite klassi $\mathcal{P}_2$ ja marginaaliläve θ nelja erineva väärtuse korral.

kust $\gamma_t = 1 - 2\epsilon_t$.

Marginaalilävega treeningviga ja selle hinnang arvutati esimesel tuhandel iteratsioonisammul leitud funktsioonide f_t jaoks marginaaliläve θ nelja erineva väärtuse korral. Joonisel 4.5 on nelja erineva θ korral kujutatud marginaaliga treeningviga $R_n^\theta(f_t)$, selle hinnangu (3.30) ja suuruse $A_n^\theta(f_t)$ (vt (3.34)) muutumist iteratsioonisammudel $t = 1, ..., 1000$.

Näeme, et hinnangu (4.4) väärtus ei ületanud 1000 esimese iteratsioonisammu



Joonis 4.6. Treeningvalimi objektide marginaalide empiiriline jaotus rusikareeglite klassi $\mathcal{P}_2$ korral 10 (kõige laugem joon), 100 ja 1000 (kõige järsem joon) iteratsioonisammu järel.

jooksul ühte ainult kõige väiksema valitud marginaaliläve, $\theta = 0{,}02$ korral. Läve $\theta = 0{,}2$ korral oli marginaalilävega treeningvea $R_n^\theta(f_t)$ ülemise tõkke (4.4) väärtus ühest suurem juba 12-ndal iteratsioonisammul.

Ka hinnatav marginaalilävega treeningviga oli kolme suurema marginaaliläve θ korral kasvav. Vaid väikseima läve $\theta = 0{,}02$ korral ei kaasnenud iteratsioonide arvu kasvuga marginaalilävega treeningvea kasv.

Uuriti, miks marginaalilävega treeningviga suuremate lävede korral kasvas. Selleks leiti treeningvalimi objektide marginaalide (vt (3.28)) empiirilise jaotusfunktsiooni

$$F_n(\theta) = \frac{\#(\rho_i | \rho_i \leqslant \theta)}{n}$$

väärtused 10-nda, 100-nda ja 1000-nda iteratsioonisammu järel (vt joonist 4.6).

Iteratsioonisammude arvu suurenemisel vähenes nullist väiksemate marginaalide

osakaal, st treeningviga vähenes, mida nägime ka jooniselt 4.3. Iteratsioonisammude kasvades klassifitseeris AdaBoosti väljundi f_t põhjal defineeritud klassifitseerija $\mathrm{sgn}(f_t)$ üha enam treeningvalimi objekte korrektselt, ent kõigi objektide korral tehti otsus napi häälteenamusega. Treeningvalimi objektide marginaalid kogunesid iteratsioonisammude arvu kasvades üha kitsamasse nullpunktist suuremate väärtustega vahemikku, mistõttu läheneski joonisel 4.5 suuremate marginaalilävede θ korral marginaalilävega treeningviga ühele.

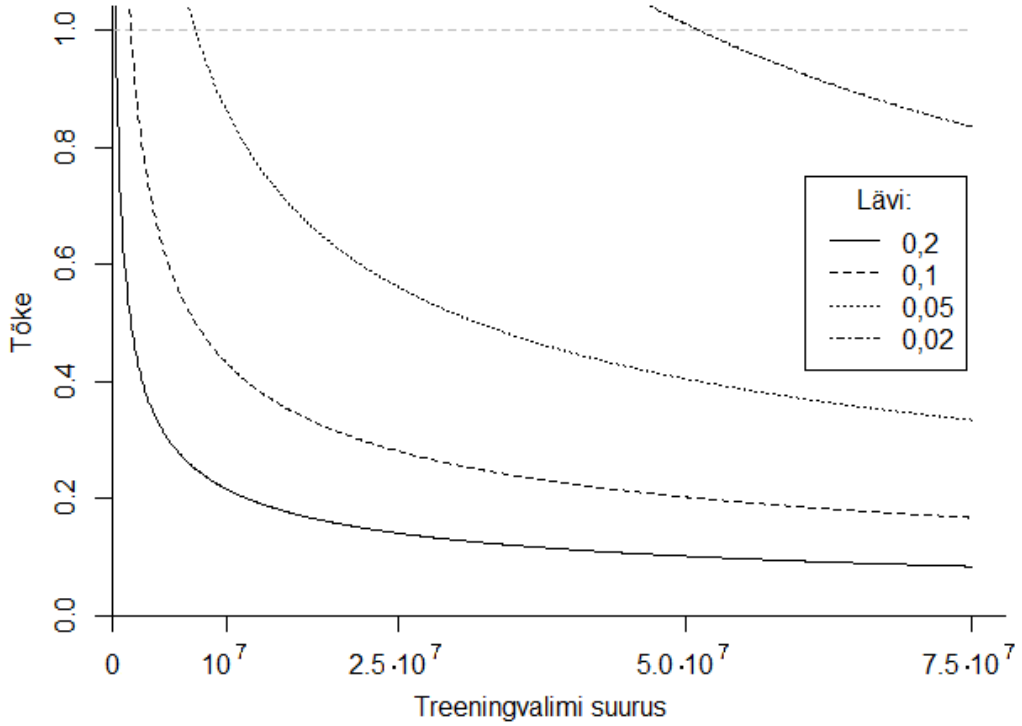
4.5 Riski hinnang

Uurime teoreemis 3.6 esitatud hinnangut AdaBoosti väljundi f_t poolt defineeritud klassifitseerija riskile. Teoreem annab iga hulka $co_T(\mathcal{H})$ kuuluva funktsiooni f põhjal defineeritud klassifitseerija riski $R\big(\mathrm{sgn}(f)\big)$ ja suuruse $A_n^\theta(f)$ vahele järgmise, tõenäosusega $1-\delta$ kehtiva ülemise tõkke (vt (3.35)).

$$R\big(\mathrm{sgn}(f)\big) - A_n^\theta(f) \leqslant \frac{8}{\theta}\sqrt{\frac{2V_{\mathcal{H}}\ln(n+1)}{n}} + \sqrt{\frac{ln\frac{2}{\delta}}{2n}}. \tag{4.5}$$

Antud hinnang sõltub suurusest δ, mida võime käsitleda olulisuse nivoona, lävest θ, klassifitseerijate hulga $\mathcal{H}$ VC dimensioonist $V_{\mathcal{H}}$ ja treeningvalimi mahust n. Fikseerime $\delta = 0{,}05$. Kasutame rusikareeglite hulga $\mathcal{P}_2$ VC dimensiooni hindamiseks võrratust (2.6). Võrratusest järeldub, et kehtib $\mathcal{V}_{\mathcal{P}_2} \leqslant 9$. Joonisel 4.7 on kujutatud, kuidas nelja läve θ väärtuse korral võrratuse (4.5) parem pool treeningvalimi suurusest n sõltub.

Näeme, et võrratuse (4.5) parem pool on funktsiooni $f \in co_T(\mathcal{H})$ põhjal defineeritud klassifitseerija riski $R\big(\mathrm{sgn}(f)\big)$ ja suuruse $A_n^\theta(f)$ vahe hindamisel väga ebatäpne. Selleks, et võrratuse (4.5) parem pool ühest väiksem oleks, peaks treeningvalim olema väga suur. Läve $\theta = 0{,}2$ korral on võrratuse (4.5) parem pool ühest



Joonis 4.7. Riski ja suuruse A_n^θ vahe hinnangu sõltuvus treeningvalimi suurusest nelja erineva marginaaliläve θ korral.

väiksem alles juhul, kui treeningvalimi maht on lähedane 400000-le või suurem. Väiksemate läve θ väärtuste korral on tarvis veelgi suuremat treeningvalimit.

Riski $R\big(\mathrm{sgn}(f)\big)$ hindamisel liidetakse võrratuse (4.5) paremale poolele veel mittenegatiivne suurus $A_n^\theta(f)$. Järeldame, et vaadeldav hinnang AdaBoosti väljundi põhjal defineeritud klassifitseerija riskile on väga ebatäpne.

4.6 AdaBoost teiste rusikareeglite korral

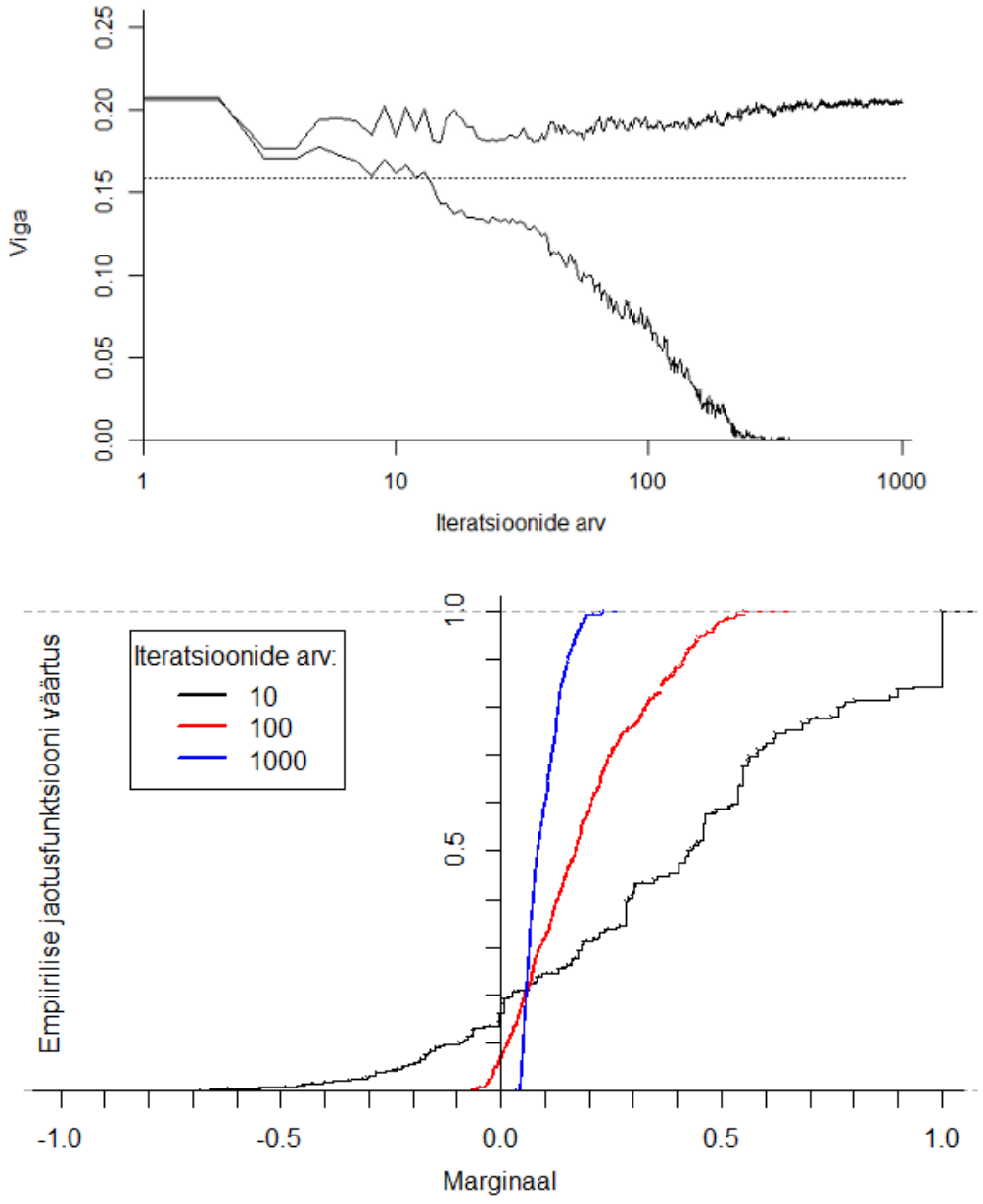
Eelneva treeningvalimiga identsel moel genereeriti kaks uut treeningvalimit. Neist ühele rakendati AdaBoosti, kus rusikareeglite klassiks $\mathcal{H}$ oli valitud kõigi maksimaalselt viietasemeliste kahendpuude klass $\mathcal{P}_4$; teisele rakendati AdaBoosti, kus rusikareeglite klassiks $\mathcal{H}$ oli kõigi kahendpuude klass $\mathcal{P}$. Samuti genereeriti eelneva testvalimiga identsel moel kaks testvalimit. Ühele neist rakendati AdaBoosti, kus rusikareeglite hulgaks $\mathcal{P}_4$, väljundit (vt joonist 4.8); teisele rakendati kõigi kahendpuude klassi $\mathcal{P}$ põhjal konstrueeritud AdaBoosti väljundit (vt joonist 4.9).

Jooniselt 4.9 näeme, et treeningvalimi objektide marginaalide jaotusfunktsiooni saba nihkus rusikareeglite hulga $\mathcal{P}$ korral reaaltelje positiivses suunas isegi pärast seda, kui funktsiooni f_t põhjal defineeritud klassifitseerija $\mathrm{sgn}(f_t)$ treeningviga oli nulliks koondunud. Seda AdaBoosti omadust on peetud üheks põhjuseks, miks AdaBoosti korral sageli ülesobitumust ei esine [14].

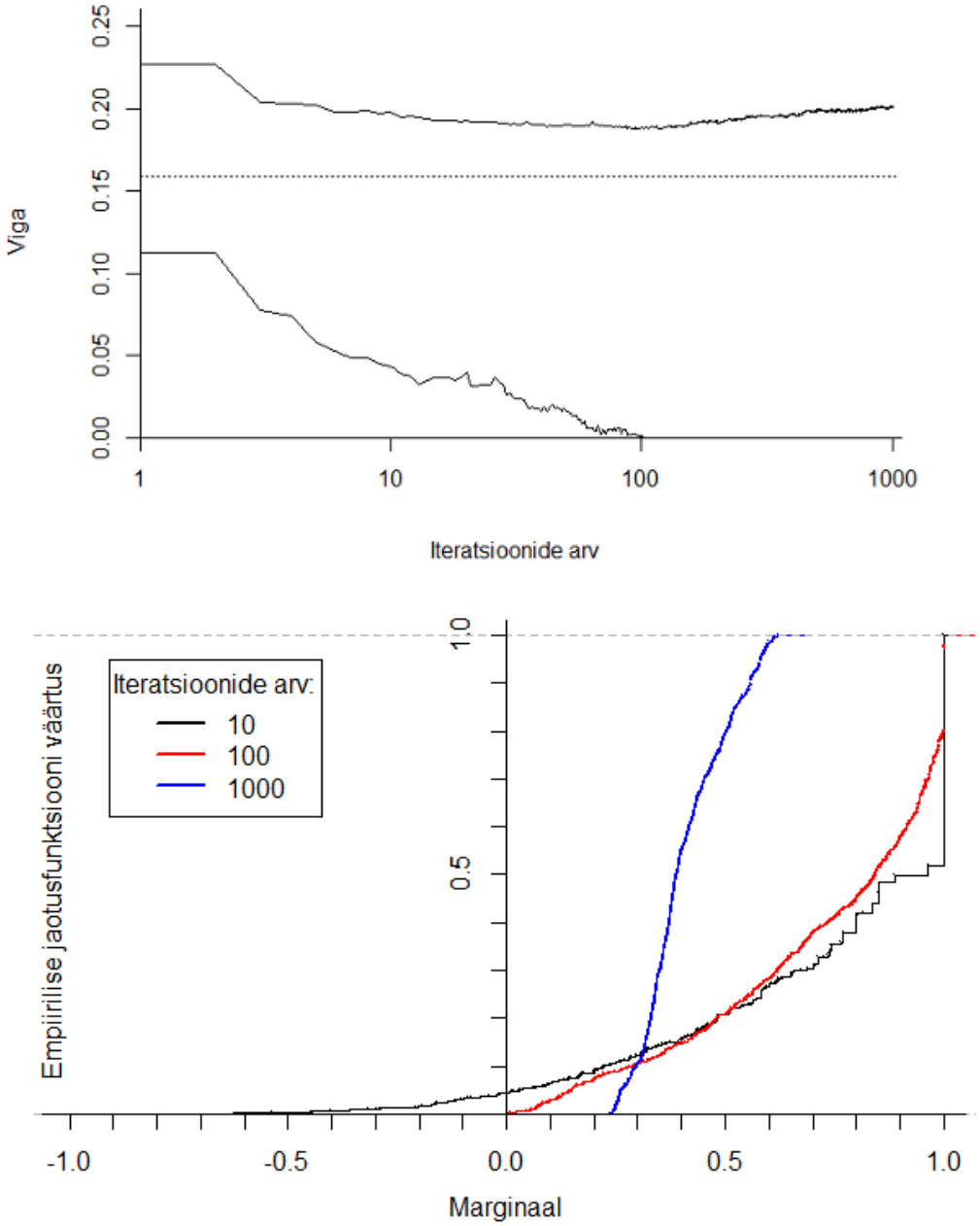
Jooniste 4.3, 4.8 ja 4.9 võrdlemisel näeme, et mida suurem oli rusikareeglitena kasutatavate kahendpuu-klassifitseerijate maksimaalne tasemete arv $m + 1$, seda kiiremini koondus treeningviga nulliks. Eespool nägime, et rusikareeglite klassi $\mathcal{P}_2$ korral koondus treeningviga esimest korda nulliks 2500-ndaks iteratsioonisammuks. Klassi $\mathcal{P}_4$ korral koondus AdaBoosti väljundi põhjal defineeritud klassifitseerija treeningviga nulliks 300-ndaks iteratsioonisammuks, kõikide kahendpuude klassi $\mathcal{P}$ korral oli treeningviga võrdne nulliga juba 100-ndal iteratsioonisammul.

Rusikareeglitena kasutatavate kahendpuude tasemete arvu ja AdaBoosti väljundi põhjal defineeritud klassifitseerija testvea vahel võis täheldada hoopis vastupidist seost. Mida lihtsamad olid rusikareeglitena kasutatavad kahendpuud, seda

väiksem oli testvea miinimum. Rusikareeglite klasside $\mathcal{P}_4$ ja $\mathcal{P}$ korral erines klassifitseerija $\mathrm{sgn}(f_{1000})$ testviga küllaltki vähe esimesel iteratsioonisammul ERM-printsiibil leitud klassifitseerija h_1 testveast. Osutubki, et AdaBoosti väljundi põhjal defineeritud klassifitseerija klassifitseerimisomadusi silmas pidades on kasulik rusikareeglitena kasutatavate kahendpuude mõõtmetele teatavad piirangud seada [8, lk-d 323-324].



Joonis 4.8. Treening- ja testviga ning marginaalide jaotused rusikareeglite hulga $\mathcal{P}_4$ korral.



Joonis 4.9. Treening- ja testviga ning marginaalide jaotused rusikareeglite hulga $\mathcal{P}$ korral.

Introduction to AdaBoost

Bachelor thesis

Jaak Sõnajalg

Summary

Boosting has been referred to as one of the most powerful learning ideas introduced during the past decades. Although boosting can be used in case of both regression and classification problem we focus our attention in this paper solely on classification. The purpose of this thesis is to outline the main qualities of boosting methods and to introduce the most popular boosting algorithm AdaBoost.

This thesis is organized as follows. Chapter 1 includes some of the basic definitions of statistical learning such as loss function and risk. Chapter 2 discusses the idea of learning from data and introduces the basic principles of what is known as Vapnik-Chernonenkis theory. Chapter 3 is devoted to boosting. Description of the AdaBoost algorithm is given. We also assess some of the properties of AdaBoost, e.g. risk bounds.

In Chapter 4 the AdaBoost algorithm is put to use. We observe how AdaBoost performs when applied to simulated training data. We calculate some risk bounds

and use the results to evaluate the precision of the risk bounds by comparing them with the actual risk.

Kirjandus

[1] Bartlett, P.L., M.I. Jordan, J.D. McAuliffe, 2006. Convexity, Classification, and Risk Bounds, *Journal of the American Statistical Association*, 101 (473), lk-d 138-156.

[2] Bartlett, P.L., M. Traskin, 2007. AdaBoost is consistent, *Journal of Machine Learning Research* , 8, lk-d 2347-2368.

[3] Burges, C, 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), lk-d 121-167.

[4] Culp , M., K. Johnson, G. Michailidis, 2013. *Package 'ada'*. [www-materjal] Kättesaadav aadressil <http://cran.r-project.org/web/packages/ada/ada.pdf> [viimati vaadatud 04.04.2013].

[5] Freund, Y., R. E. Schapire, 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences,* 55 (1), lk-d 119-139.

[6] Friedman, J., T. Hastie, R. Tibshirani, 2000. Additive Logistic Regression: a Statistical View of Boosting, *Annals of Statistics,* 28(2).

[7] Freund, Y., R. E. Schapire, 1999. A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5), lk-d 771-780.

[8] Hastie, T, R. Tibshirani, R. Friedman, 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. väljaanne. Springer.

[9] Lember, J., 2008. Tehisõpe, *MTMS.02.035 Tehisõpe*, Tartu Ülikool. Kättesaadav aadressil <http://www.ms.ut.ee/ained/Tehis\%F5pe/tehisope8.pdf> [viimati vaadatud 04.04.2013].

[10] Lember, J., 2012. Introduction to statistical learning (lecture notes), *Statistical learning*, Aalto Ülikool. Kättesaadav aadressil <https://math.aalto.fi/en/research/stochastics/events/lember/AFrame.pdf> [viimati vaadatud 04.05.2013].

[11] Meir, R., G. Rätsch, 2003. An Introduction to Boosting and Leveraging. *Advanced Lectures on Machine Learning*, lk-d 118-183. Springer.

[12] Nowak, R. Statistical Learning Theory, *ECE 901*, University of Wisconsin-Madison. Kättesaadav aadressil <http://nowak.ece.wisc.edu/SLT09/lecture18.pdf> [viimati vaadatud 21.04.2013].

[13] Nowak, R. Statistical Learning Theory, *ECE 901*, University of Wisconsin-Madison. Kättesaadav aadressil <http://nowak.ece.wisc.edu/SLT09/lecture20.pdf> [viimati vaadatud 28.04.2013].

[14] Schapire, R.E., Y. Freund, P. Bartlett, 1998. Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. *Annals of Statistics*, 26(5), lk-d 1651-1686.

[15] Therneau, T.M., E.J. Atkinson, 2013. *An Introduction to Recursive Partitioning Using the RPART Routines*. Kättesaadav aadressil <http://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf> [viimati vaadatud 04.04.2013].

[16] Vapnik, V.N, 1998. *Statistical Learning Theory*. Wiley.

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Jaak Sõnajalg
(*autori nimi*)
(sünnikuupäev: 19. mai 1991)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
Tutvumine AdaBoostiga,
(*lõputöö pealkiri*)

mille juhendaja on Jüri Lember,
(*juhendaja nimi*)

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 6. mail 2013