LATVIJAS UNIVERSITĀTE
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE
MATEMĀTIKAS NODAĻA

STATISTISKĀS UN DATU IZRACES METODES KLASIFIKĀCIJAS UZDEVUMOS

DIPLOMDARBS

Autors: **Anastasija Tetereva**

Stud. apl. at05001

Darba vadītājs: docents Dr. Math. Jānis Valeinis

RĪGA 2010

Saturs

Anotācija

Pēdējos gados klasifikācijas problemātika kļuvusi ļoti aktuāla lēmumu pieņemšanas dažādās sfērās. Šo uzdevumu var risināt gan ar statistikas, gan ar datu izraces (angliski data mining) palīdzību. Šī darba mērķis ir noskaidrot, vai datu izraces algoritmi spēj konkurēt ar statistiskām metodēm. Darbā ir aprakstīta lineāra diskriminantu analīze (LDA),kodolu diskriminantu analīze (KDA), kā arī klasifikācijas koki (CRT) un vienslāņa neironu tīkli (NNET). Statistiskajiem klasifikatoriem ir aprakstīts diskriminantu funkcijas un diskriminācijas robežu iegūšanas process, datu izraces modeļiem ir apskatīti klasifikatoru būvēšanas algoritmi, kā arī metodes, ar kuru palīdzību var izvairīties no pārliekas pielāgošanās datiem. Darba nobeigumā ir apskatīti modeļu salīdzināšanas paņēmieni. Lai empīriski salīdzinātu klasiskās metodes ar datu izraces metodēm, tika veiktas simulācijas programmā R.

Atslēgas vārdi: datu izrace, klasifikators, diskriminantu analīze, klasifikācijas koki, neironu tīkli, kopējā precizitāte.

Annotation

In recent years classification problem has become a topical question in different field of decision making. Such kind of tasks can be solved using both statistical and data mining techniques. The goal of this thesis is to elucidate whether the data mining algorithms can be considered as competitors of statistical methods. Linear and kernel discriminant analysis, classification trees and neural networks are described in the thesis. It is explained how to get discriminant function and discrimination borders for statistical techniques and how to construct data mining classifiers avoiding unnecessary adaptation to data. Finally, model assessment and selection are discussed. The thesis contains empirical comparison of the classical statistical techniques and data mining algorithms in terms of simulated examples. Simulations were fulfilled, using statistical software R.

Key words: data mining, classifier, discriminant analysis,classification trees, neural networks, overall accuracy.

1. Ievads

Datu izrace (angliski data mining) ir process, kurā no liela apjoma datiem tiek iegūtas jaunas, netriviālas, praktiski derīgas sakarības, kas nepieciešamas lēmumu pieņemšanai visdažādākajās darbības sfērās. Datu izraces pamatā ir likumsakarību, kas raksturīgas datu izlasēm, pētīšana. Pats termins radies 1978.gadā un sākot ar 90.gadu pirmo pusi, datu izraces metodes ir guvušas plašu ievērību. Līdz tam datu analīzi veica ar statistisko metožu palīdzību. Attīstoties tādām nozarēm kā tēlu atpazīšana, mākslīgais intelekts, datu bāžu teorija un mašīnapmācība (angliski machine learning), izmantojot statistiskās metodes ir izveidojusies jauna plaša nozare - datu izrace.

Tradicionālas statistiskās datu analīzes metodes galvenokārt balstās uz iepriekš formulētiem teorētiskiem pieņēmumiem, bet datu izraces pamatā ir "ne uzreiz pamanāmu" likumsakarību meklēšana. Ja salīdzina datu izraci, statistiku un mašīnapmācību, tad statistika pamatā bāzējas uz teoriju, mašīnapmācība bāzējas uz apmācību, bet datu izrace integrē teoriju un apmācību. Ja statistika koncentrējas uz teorētisko rezultātu iegūšanu, bet mašīnapmācība - uz apmācības aģentu darbības uzlabošanu, tad datu izrace ir koncentrēta uz vienotu datu analīzes procesu, kas ietver datu attīrīšanu, apmācību, rezultātu integrāciju un vizualizāciju.

Datu izraces uzdevumi pēc savas būtības tiek dalīti divos tipos: aprakstošie un prognozējošie. Aprakstošie modeļi tiek veidoti ar mērķi atrast paraugus, kas kopumā raksturotu esošus datus. Tie kalpo eksistējošo īpašību atklāšanai, nevis jaunu īpašību prognozēšanai. Prognozējošais modelis ir orientēts uz datu vērtību prognozēšanu, ņemot vērā objektu aprakstošās vērtības un iepriekš, no datiem iegūtas zināšanas. Prognozēšanas process notiek divos posmos - likumsakarību atrašana un atrasto likumsakarību izmantošana, lai prognozētu nezināmās vērtības. Prognozēšanas uzdevumu apakšuzdevums ir klasifikācija - objektu piederības noteikšana iepriekš definētām grupām. Klasificēšanas metodes tiek plaši pielietotas medicīnā, banku industrijā, mārketingā un daudzās citās sfērās. Piemē-

ram, intensīvās terapijas nodaļā atrodas grupa ar pacientiem, kuriem, iespējams, ir sepse (asins saindēšanās). Jāatrod klasifikators, ar kura palīdzību, ņemot vērā tikai ārējos un vienkāršākos simptomus, var pēc iespējas precīzāk pateikt, vai cilvēks ir slims. Laboratorijas tests dod precīzus rezultātus, bet ārstēšana ir jāsāk pēc iespējas ātrāk, negaidot analīžu rezultātus. Daudzi klasifikācijas paņēmieni tiek piedāvāti bankām, lai ar maksimālu precizitāti novērtētu, vai klients, kuram tiek izsniegts kredīts, izrādīsies maksātspējīgs. Mārketingam ir aktuāls uzdevums no visiem uzņēmuma klientiem izvēlēties tiešas pārdošanas komunikācijai tikai tos klientus, kuri labprāt piekritīs konkrētam piedāvājumam.

Statistikas valodā runājot klasifikācijas uzdevums var tikt aprakstīts sekojoši - pieņemsim, ka ir dota novērojumu (datu) kopa $\{\mathbb{X}, G\}$, kura sastāv no atsevišķiem ierakstiem (angliski example, record) $(\mathbf{X}_j, G_j)$, $j = \overline{1, N}$, kur $\mathbf{X}_j \in \mathbb{K}^p$ ir pazīme vai faktors (angliski attribute, feature). $\mathbb{K}^p$ var būt, piemēram, $\mathbb{R}^p$, $\mathbb{N}^p$, $\mathbb{Z}^p$. G_j ir ieraksta X_j īstā grupa. Tālāk darbā īsto klasi apzīmēsim ar G, bet to novērtējumu ar $\widehat{G}(X)$. Gan G, gan $\widehat{G}(X)$ pieņem vērtības no visu iespējamo klašu jeb grupu (angliski class) kopas $\Gamma = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_g\}$. Katrs konkrēts ieraksts pieder kādai no grupām. Klasifikācijas uzdevums ir izveidot attēlojumu (klasifikatoru, modeli) $\widehat{G}(X) : \mathbb{K}^p \to \Gamma$, kurš pazīmju kopai piekārtos grupu, kad tā nav zināma. Piemēram, ir doti dati par trim bankas klientiem, kuru vecums ir 25, 50 un 75 gadi, bet vidējie mēneša ienākumi attiecīgi ir 100, 1500 un 300 lati. Visiem klientiem tika izsniegts kredīts. Vēlāk izrādījās, ka tikai otrais klients ir izpildījis savas kredītsaistības. Šajā piemērā ierakstu skaits ir $N = 3$, iespējamo grupu kopa $\Gamma = \{\textit{maksātspējīgs}, \textit{maksātnespējīgs}\}$. Pazīmes X ir *vecums* un *vidējie mēneša ienākumi*. $X_1 = (25, 100)$, $X_2 = (50, 1500)$, $X_3 = (75, 300)$, $G_1 = G_3 = \{\textit{maksātnespējīgs}\}$, $G_2 = \{\textit{maksātspējīgs}\}$. Uzdevums ir atrast klasifikatoru $\widehat{G}(X)$, kurš ļaus prognozēt, vai klients atdos kredītu.

Dotā diplomdarba uzdevums ir izpētīt visizplatītākās statistiskās un datu izraces klasifikācijas metodes, kā arī to salīdzināšanas paņēmienus. Lai sasniegtu darba mērķi tika izvirzīti sekojoši uzdevumi:

1. Iepazīties ar lineāro un kodolu diskriminantu analīzi;
2. Apskatīt klasifikācijas koku un vienslāņa neironu tīklu metodes;
3. Veikt metožu salīdzināšanu, veicot simulācijas un pielietojot butstrapa un krosvalidācijas metodes.

Darbs sastāv no četrām galvenajām nodaļām un tām atbilstošām apakšnodaļām. Pirmā nodaļa ir veltīta statistisko diskriminācijas metožu - lineārās diskriminantu analīzes un kodolu diskriminantu analīzes - aprakstam un ilustrācijai. Turpmākais darbs ir veltīts datu izraces algoritmiem. Otrā nodaļā tiek apskatīti klasifikācijas koki un vienslāņa neironu tīkli. Pēdējā nodaļā ir parādīts, kā, izmantojot butstrapa un krosvalidācijas metodes, var salīdzināt statistiskos un datu izraces klasifikatorus. Visi teorētiskie rezultāti un metodes ir ilustrētas ar simulēto datu piemēriem, kuri palīdz saprast ne tikai dažādu metožu iespējas pielietošanai praksē, bet arī to priekšrocības un trūkumus. Pielikumā ir dotas praktisko uzdevumu programmas.

Darbā iekļautās teorijas izklāsts pamatā ir balstīts uz Hastie [1], Ripley [2], Duda [3], Bishop [4] un McLachlan [5] grāmatām. Datorprogrammas tika izveidotas ar paketes R palīdzību.

2. Statistiskie klasifikatori

2.1. Lineārā diskriminantu analīze

Diskriminantu analīze ir metožu kopums, kas ļauj risināt objektu atšķirības (diskriminācijas) problēmu, vadoties pēc objektu parametriem. Tā ļauj pētīt atšķirības starp divām un vairākām objektu grupām. Diskriminantu analīzes procedūras var būt sadalītas divās grupās: pirmā procedūra ļauj interpretēt atšķirības starp jau eksistējošām grupām, otrā - klasificē jaunus objektus, izmantojot jau izveidotas klasifikācijas pazīmes. Diskriminantu analīzes ideja ir piekārtot konkrētam ierakstam klasi ar vislielāko nosacīto varbūtību, tādā veidā minimizējot kļūdainas klasificēšana varbūtību. Katrai kļūdainai klasificēšanai, kad $\widehat{\mathcal{G}} \neq \mathcal{G}$ tiek piekārtotas izmaksas $\lambda(\widehat{\mathcal{G}}|\mathcal{G})$. Lai vienkāršotu apzīmējumus, šīs nodaļas ietvaros $G = \mathcal{G}$ vietā rakstīsim $\mathcal{G}$, bet $\widehat{G}(X) = \mathcal{G}$ apzīmēsim ar $\widehat{\mathcal{G}}$.

Definīcija 1. [3] Sagaidāmās izmaksas i-tās grupas klasificēšanā, kuras tiek sauktas par nosacīto risku, ir funkcija

$$R(\widehat{\mathcal{G}}_i|x) = \sum_{j=1}^{g} \lambda(\widehat{\mathcal{G}}_i|\mathcal{G}_j)P(\mathcal{G}_j|x), \tag{2.1.1}$$

kur $\mathcal{G}$ ir novērojuma īstā klase, $\widehat{\mathcal{G}}$ ir novērojuma prognozējama klase, bet $P(\mathcal{G}|x)$ ir varbūtība, ka x īstā klase ir $\mathcal{G}$.

Definīcija 2. [3] Kopējais nosacītais risks ir

$$\sum_{i=1}^{g} R(\widehat{\mathcal{G}}_i|x). \tag{2.1.2}$$

Par labu klasifikatoru tiek uzskatīts klasifikators, kurš minimizē kopējo nosacīto risku. Tā kā pareizās klasificēšanas izmaksas vienmēr ir mazākas par kļūdainas klasificēšanas

izmaksām, vienkāršības dēļ pieņemsim, ka

$$\lambda(\widehat{\mathcal{G}}_i|\mathcal{G}_j) = \begin{cases} 0, \, i = j, \\ 1, \, i \neq j. \end{cases}$$

Tad i-tās grupas nosacītais risks izskatīsies sekojoši

$$R(\widehat{\mathcal{G}}_i|x) = \sum_{j=1}^{g} \lambda(\widehat{\mathcal{G}}_i|\mathcal{G}_j) P(\mathcal{G}_j|x) = \sum_{i \neq j} P(\mathcal{G}_j|x) = 1 - P(\mathcal{G}_i|x).$$

Ir acīmredzami, ka kopējo nosacīto risku minimizēs klasifikators, kurš piekārtos ierakstam grupu $\mathcal{G}_i$, ja $P(\mathcal{G}_i|x) > P(\mathcal{G}_j|x)$. Katra novērojuma nosacītā varbūtība var tikt aprēķināta pēc Beijesa formulas [6]

$$P(\mathcal{G}_i|x) = \frac{p(x|\mathcal{G}_i)\pi(\mathcal{G}_i)}{\sum_{j=1}^{g} p(x|\mathcal{G}_j)\pi(\mathcal{G}_j)}, \tag{2.1.3}$$

kur $\pi(\mathcal{G}_j)$ ir piederības pie klases $\mathcal{G}_j$ varbūtība, $p(x|\mathcal{G}_j)$ ir klases $\mathcal{G}_j$ sadalījuma blīvuma funkcija.

Definīcija 3. [7] Par grupas $\mathcal{G}_i$ diskriminantu funkciju sauc tās nosacīto varbūtību $P(\mathcal{G}_i|x)$, kura ir definēta (2.1.3).

Teorēma 1. *Pieņemsim, ka visu grupu $\mathcal{G}_i$, $i = 1, ..., g$, sadalījumi atbilst daudzdimensiju normālajam sadalījumam ar matemātisko cerību μ_i un kovariāciju matricu Σ_i, t.i. $p(x|\mathcal{G}_i) \sim N(\mu_i, \Sigma_i)$, un kovariāciju matricas ir vienādas diagonālmatricas $\Sigma_i = \Sigma = \sigma^2 \mathbb{I}$, tad i-tās grupas diskriminantu funkcija*

$$g_i(x) = \omega_i^t x + \omega_{io},$$

kur $\omega_i = \frac{\mu_i^t}{\sigma^2}$, $\omega_{io} = -\frac{1}{2\sigma^2}\mu_i^t\mu_i + \ln \pi(\mathcal{G}_i)$, ω_i^t ir vektora ω_i transponētais vektors.

Pierādījums. Novērojums tiek piekārtots grupai, kurā diskriminantu funkcija sasniedz savu maksimumu, minimizējot kopējo risku (2.1.1). Tās izvēle nav unikāla. Ja diskriminantu funkcija tiek pareizināta ar pozitīvu konstanti vai tai tiek pielietota monotoni augoša transformācija, klasificēšanas rezultāts paliks nemainīgs. Sākotnēji diskriminantu funkcija tiek definēta kā grupas nosacītā varbūtība, kura tiek aprēķināta pēc Beijesa formulas (2.1.3), tālāk to pareizina ar $\sum_{j=1}^{g} p(x|\mathcal{G}_j)\pi(\mathcal{G}_j)$ un logaritmē. Rezultātā iegūst, ka

$$g_i(x) = \ln p(x|\mathcal{G}_i) + \ln \pi(\mathcal{G}_i).$$

Ņemot vērā, ka

$$p(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp\left[-\frac{1}{2}(x-\mu)^t\Sigma^{-1}(x-\mu)\right],$$

kur $p(x)$ ir sadalījuma blīvuma funkcija, $|\Sigma|$ ir kovariāciju matricas Σ determinants, p ir telpas $\mathbb{R}^p$ dimensija, Σ^{-1} ir matricas Σ inversā matrica,
diskriminatu funkcija ir vienāda ar

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t\Sigma^{-1}(x-\mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}|\Sigma| + \ln \pi(\mathcal{G}_i).$$

Ja visu grupu kovāriāciju matricas ir diagonālmatricas un ir vienādas savā starpā, tad

$$g_i(x) = -\frac{||x-\mu_i||^2}{2\sigma^2} + \ln \pi(\mathcal{G}_i) = -\frac{1}{2\sigma^2}\left[x^tx - 2\mu_i^tx + \mu_i^t\mu_i\right] + \ln \pi(\mathcal{G}_i),$$

kur $||x|| = \sqrt{x^tx} = \sqrt{x_1^2 + x_2^2 + ... + x_p^2}$ ir norma Eiklīda telpā $\mathbb{R}^p$. Vienīgais loceklis, kurš ir atkarīgs no x kvadrātiski, var tikt atņemts no diskriminantu funkcijas kā konstante, jo fiksētam x būs vienāds visu grupu diskriminantu funkcijām un neietekmēs lēmuma pieņemšanu. Tādējādi,

$$g_i(x) = \frac{\mu_i^t}{\sigma^2}x - \frac{1}{2\sigma^2}\mu_i^t\mu_i + \ln \pi(\mathcal{G}_i) = \omega_i^tx + \omega_{io}.$$

□

Piezīme 2. *Lineārās diskriminantu funkcijas novērtējums ir iegūstams, aizvietojot izteiksmē (2.2.1) matemātisko cerību μ_i un dispersiju σ^2 ar to novērtējumiem $\overline{x} = \frac{\sum_{l=1}^N x_l}{N}$ un $S^2 = \frac{\sum_{l=1}^N (x_l-\overline{x})^2}{N-1}$.*

Piezīme 3. *Lineāru diskriminantu analīzes novērtētas grupas vērtība $G(x) = \underset{i}{argmax} g_i(x)$, kur $g_i(x)$ ir (2.2.1).*

Apgalvojums 4. *[3] Robeža starp klasēm i un j tiek meklēta kā vienādojuma*

$$\frac{(\mu_i-\mu_j)}{\sigma^2}x - \frac{||\mu_i||^2 - ||\mu_j||^2}{2\sigma^2} + \ln\frac{\pi(\mathcal{G}_i)}{\pi(\mathcal{G}_j)} = 0$$

atrisinājums.

Pierādījums. Klases novērtējums tiek meklēts kā $G(x) = \underset{i}{\text{argmax}} g_i(x)$. Ja $g_i(x) > g_j(x)$, tad novērojumam tiek piekārtota klase $\mathcal{G}_i$, ja $g_i(x) < g_j(x)$, tad klase $\mathcal{G}_j$. Taisne $g_i(x) = g_j(x)$ jeb $g_i(x) - g_j(x) = 0$, kur $g_i(x)$ un $g_j(x)$ ir definēti (2.2.1) ir diskriminācijas robeža, kura atdala klasi $\mathcal{G}_i$ no $\mathcal{G}_j$. $g_i(x) - g_j(x) = \frac{\mu_i^t}{\sigma^2}x - \frac{1}{2\sigma^2}\mu_i^t\mu_i + \ln \pi(\mathcal{G}_i) - (\frac{\mu_j^t}{\sigma^2}x - \frac{1}{2\sigma^2}\mu_j^t\mu_j + \ln \pi(\mathcal{G}_j)) = \frac{(\mu_i-\mu_j)}{\sigma^2}x - \frac{||\mu_i||^2-||\mu_j||^2}{2\sigma^2} + \ln\frac{\pi(\mathcal{G}_i)}{\pi(\mathcal{G}_j)} = 0$.

Piezīme 5. *Divu klašu gadījumā ir pieņemts runāt ne par divām diskriminantu funkcijām, bet par vienu* $g(x) = g_1(x) - g_2(x)$ *un piekārtot novērojumu klasei 1, ja* $g(x) > 0$ *un klasei 2 pretējā gadījumā.*

2.2. Lineārās diskriminantu analīzes ilustrācija

Lai ilustrētu diskriminantu analīzes pielietojumu praksē un vēlāk salīdzinātu to ar citiem klasifikatoriem pat gadījumos, kad pieņēmumi netiek izpildīti, tika konstruēti 5 piemēri. Visi piemēri ir izveidoti no datiem ar divām pazīmēm, t.i. tie nāk no divdimensionālā sadalījuma. Dimensiju skaits netika palielināts, lai piemērus varētu ilustrēt ar attēliem, novērojot, kādas diskriminantu robežas konstruē katrs klasifikators. Grupu kopa visos piemēros sastāv no diviem elementiem un ir vienāda ar $\Gamma = \{A, B\}$. Pirmajā piemērā (*Anorm*) dati tika simulēti no divdimensiju normālā sadalījuma. A un B grupas teorētiskie sadalījumi ir attiecīgi $A \sim N((1,5)^t; diag(1^2, 2^2))$ un $B \sim N((2,2)^t; diag(1^2, 2^2))$, kur $diag(a, b)$ ir 2×2 diagonālmatrica ar elementiem a un b uz diagonāles. Šis piemērs tika konstruēts tā, lai izpildītos visi lineārās diskriminantu analīzes pieņēmumi. Tādā veidā darbā tiks empīriski pētīts, vai datu izraces klasifikatori varēs dod labāku precizitāti par LDA, kad visi LDA pieņēmumi ir spēkā. Lai saprastu, kā strādā metode gadījumā, kad dati nav sadalīti nepārtraukti, tika izveidots piemērs *Adiskr*, kurā pirmais faktors ir sadalīts normāli $x \sim N(0; 2)$, otrais ir uzdots ar sadalījuma rindu, kura ir attēlota 2.1. tabulā. Šis piemērs dos iespēju saprast, kādi klasifikatori zaudē savu spēku, kad dati nav sadalīti nepārtraukti. Īpaši interesanti ir analizēt piemēru *Adiskr* kodolu diskriminantu analīzes kontekstā, kur grupu sadalījumi tiks novērtēti pēc datiem ar kodolu gludināšanu, tomēr ar pieņēmumu par sadalījuma blīvuma funkciju nepārtrauktību. *Adiskr* varētu sagādāt grūtības arī neironu tīkliem, jo diskriminantu funkcija tiek uzdota kā nepārtraukta funkcija. *Adiskr* datu dalījums divās grupās ir uzdots ar (2.2.1), kur $z \sim N(0; 1)$ ir troksnis, kurš tiek pieskaitīts dalījuma robežai, lai padarītu klasifikācijas uzdevumu sarežģītāku, t.i. lai robeža starp grupām nav izsakāma kā funkcija. Izlases dalījums divās grupās ir

2.1. tabula: *Adiskr* izlasē izmantotā diskrēta gadījuma lieluma sadalījums.

Vērtība	1	2	3	4	5	6
Varbūtība	0.5	0.3	0.1	0.05	0.03	0.02

uzdots ar likumu (2.2.1), kur $z \sim N(0;1)$.

$$\mathcal{G}_{diskr} = \begin{cases} A, \text{ ja } x - y \geq z - 2, \\ B, \text{ ja } x - y < z - 2. \end{cases} \tag{2.2.1}$$

$$\mathcal{G}_{lin} = \begin{cases} A, \text{ ja } x + y \geq z, \\ B, \text{ ja } x + y < z. \end{cases} \tag{2.2.2}$$

Trīs nākamie piemēri -*Alin*, *Akv* un *Ael* - tiek veidoti, simulējot datus no $x \sim N(0;2)$, $y \sim exp(1)$ un $z \sim N(0;1)$. Apvienojot x un y tiek izveidots divdimensionāls sadalījums, kurš ir sadalīts grupās 3 veidos, mēģinot imitēt dažādas datu struktūras. Šie piemēri ir konstruēti, lai izpētītu, kādas datu robežas vislabāk atpazīst viens vai cits klasifikators. Lai apgrūtinātu klasifikatora uzdevumu, katru reizi robežai tiks pieskaitīts troksnis z. *Alin* piemērs ir izveidots tā, lai neizpildītos pieņēmums par daudzdimensionālo normālo sadalījumu, bet klases tomēr ir atdalāmas ar hiperplaknēm. (2.2.2) parāda, kādā veidā *Alin* piemērā simulētie dati tiek sadalīti divās grupās - A un B. Akv imitē grupas, kuras ir atdalāmas ar kvadrātisko robežu. Šis piemērs ir interesants ar to, ka viena grupa atrodas otras grupas iekšā. Dalījums grupās tiek nodrošināts ar likumu (2.2.3). Pēdējais piemērs ļauj novērtēt, kā uzvedas klasifikators, kad viena no grupām satur mazāk datu nekā otra. Robeža starp klasēm atkal nav novērtējama kā lineāra. Dalījums grupās A un B ir uzdots ar vienādojumu (2.2.4)

$$\mathcal{G}_{kv} = \begin{cases} A, \text{ ja } -0.8(x + y - 1)^2 + 2 \geq z, \\ B, \text{ ja } -0.8(x + y - 1)^2 + 2 < z. \end{cases} \tag{2.2.3}$$

$$\mathcal{G}_{el} = \begin{cases} A, \text{ ja } 0.5x^2 + 3y^2 \geq z + 3, \\ B, \text{ ja } 0.5x^2 + 3y^2 < z + 3. \end{cases} \tag{2.2.4}$$

Visi pieci piemēri un to dalījums grupās pie izlases apjoma $N = 300$ ir redzamas 2.1. attēlā. Lai varētu pārliecināties par lineārās diskriminantu analīzes pielietošanas iespēju, jāveic hipotēžu, par grupu sadalījumu atbilstību normālajam un kovariāciju matricu vienādību, pārbaudi. Normalitātes pārbaudei katras izlases katrai grupai jākonstruē $q - q$ grafiks. Grafiku konstruēšanā tika izmantots

Apgalvojums 6. *[8] Ja gadījuma lieluma sadalījums atbilst daudzdimensiju normālajam sadalījumam, tad Mahalanobisa attāluma starp novērojumiem un matemātisko cerību kvadrāts ir sadalīts pēc χ^2 likuma ar p brīvības pakāpēm (p ir dimensiju skaits), t.i.*

$$D_M{}^2 = (x - \mu)^t \Sigma^{-1} (x - \mu) \overset{x \sim N(\mu;\Sigma)}{\sim} \chi_p^2.$$

$q-q$ grafiki izlasēm ar apjomu $N=300$ ir parādīti 2.2. attēlā. Kovariāciju matricu vienādība tika pārbaudīta ar Box's M testa palīdzību, kas apgalvo, ka

$$M=\gamma\sum_{i=1}^{q}(n_i-1)\log|S_i^{-1}S| \overset{\Sigma_i=\Sigma,\forall i=\overline{1,q}}{\sim} \chi^2_{p(p+1)(q-1)/2},$$

kur p ir pazīmju skaits, q ir grupu skaits, S_i it i-tās grupas kovariāciju matricas novērtējums, bet S ir grupu *pooled* kovariāciju matricas novērtējums, $\gamma=1-\frac{2p^2+3p-1}{6(p+1)(q-1)}\{\sum_{i=1}^{q}\frac{1}{n_i-1}-\frac{1}{N-q}\}$, N ir novērojumu skaits, n_i ir novērojumu skaits i-tajā grupā. Testa p-vērtības izlasei ar apjomu $N=300$ ir attēlotas 2.2. tabulā. Var secināt, ka tikai pirmajai izlasei

2.2. tabula: Izlašu Box's M testa p-vērtības.

Izlase	p-vērtība
$Anorm$	3.96×10^{-1}
$Adiskr$	4.70×10^{-6}
$Alin$	1.97×10^{-12}
Akv	1.41×10^{-53}
Ael	3.96×10^{-1}

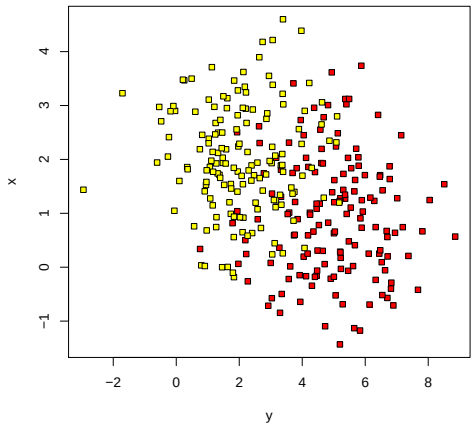
($Anorm$) pie nozīmības līmeņa $\alpha=0.01$ nevar noraidīt hipotēzi, ka izpildās lineārās diskriminantu analīzes pieņēmumi. Neskatoties uz šo faktu, visiem 5 piemēriem tika pielietota diskriminantu analīze un apskatītas modeļa kopējās precizitātes izmaiņas. Runājot par kāda konkrēta modeļa precizitāti, ņem vērā kopējo precizitāti, kuras novērtējums ir

$$OA=\frac{1}{N}\sum_{i=1}^{N}I(G_i=\widehat{G}(X_i)), \tag{2.2.5}$$

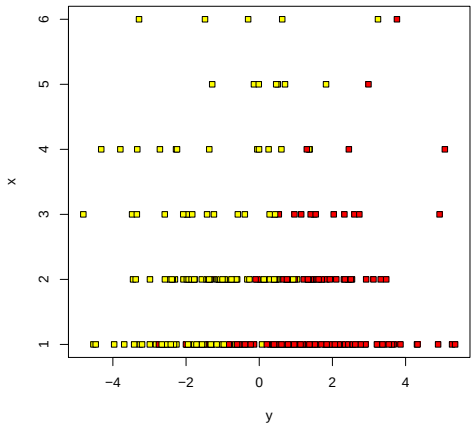
un katras grupas precizitāti atsevišķi, kas tiek novērtēta ar

$$A_k=\frac{1}{n_k}\sum_{i=1}^{n_k}I(G_i=\widehat{G}(X_i)), \tag{2.2.6}$$

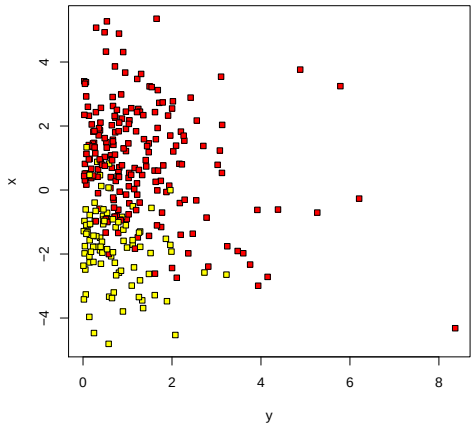
kur $I(G_i=\widehat{G}(X_i))$ ir īstās un prognozētās klases sakritības indikatorfunkcija, N ir izlases apjoms, n_k ir k-tās klases apjoms izlasē. Lineārās diskriminantu analīzes klasifikatori visām 5 izlasēm ar apjomu 300 ir parādīti 2.3. attēlā. Ir zināms, ka izlases apjomam tiecoties uz bezgalību, kopējā precizitāte tiecas uz īsto klasifikatora precizitāti (. [9]. Lai varētu salīdzināt lineārās diskriminantu analīzes precizitāti visos konstruētos piemēros, tika izveidotas 5 izlases ar apjomu $N=100000$. Pielietojot lineāro diskriminantu analīzi, tika izrēķināta kopējā un grupu precizitāte, kurai pie dotās izlases apjoma ir jābūt ļoti



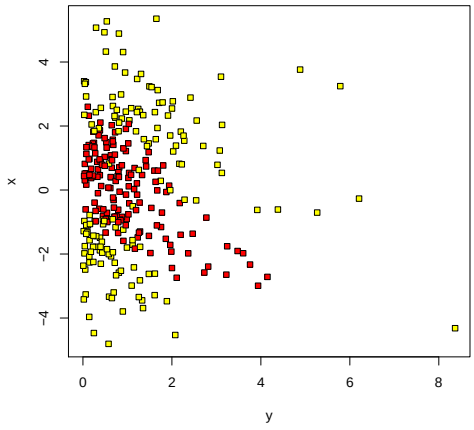
(a) *Anorm* $(N = 300,\ n_1 = 150,\ n_2 = 150)$



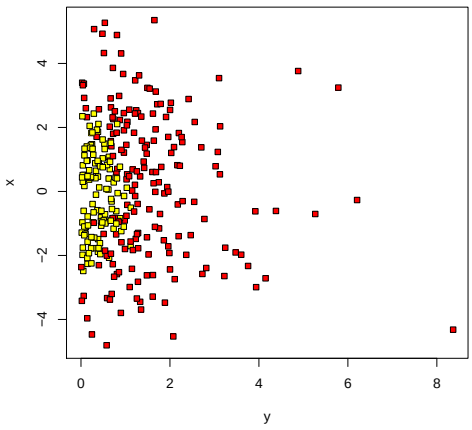
(b) *Adiskr* $(N = 300,\ n_1 = 152,\ n_2 = 148)$



(c) *Alin* $(N = 300,\ n_1 = 201,\ n_2 = 99)$

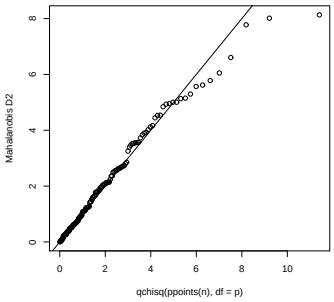


(d) *Akv* $(N = 300,\ n_1 = 148,\ n_2 = 152)$

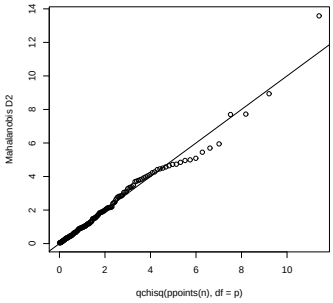


(e) *Ael* $(N = 300,\ n_1 = 182,\ n_2 = 118)$

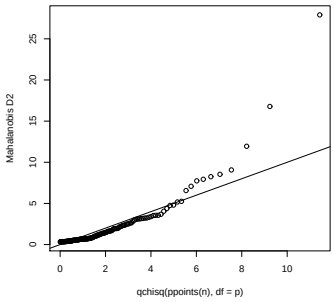
2.1. att.: Ģenerēto izlašu izkliedes grafiki.



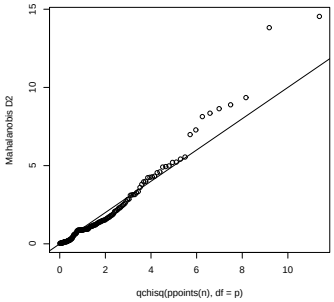
(a) $Anorm(\mathrm{A})$



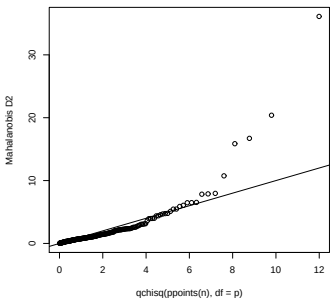
(b) $Anorm(\mathrm{B})$



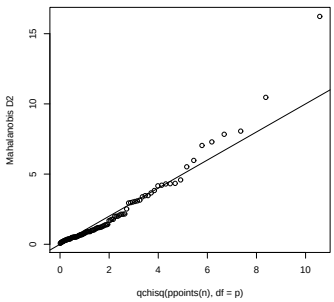
(c) $Adiskr(\mathrm{A})$



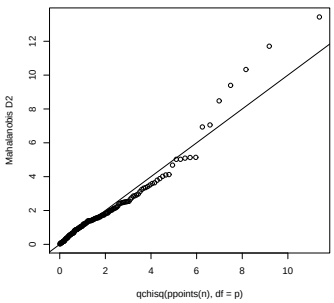
(d) $Adiskr(\mathrm{B})$



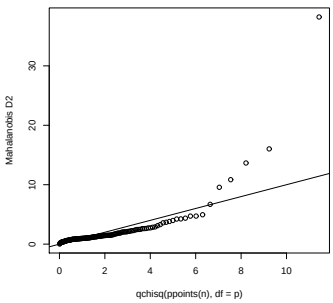
(e) $Alin(\mathrm{A})$



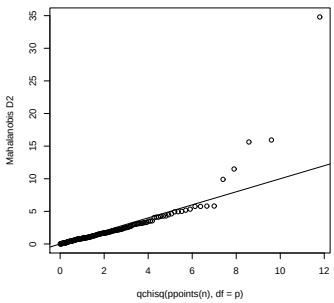
(f) $Alin(\mathrm{B})$



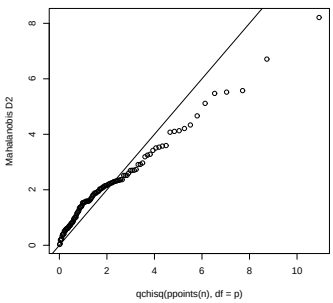
(g) $Akv(\mathrm{A})$



(h) $Akv(\mathrm{B})$

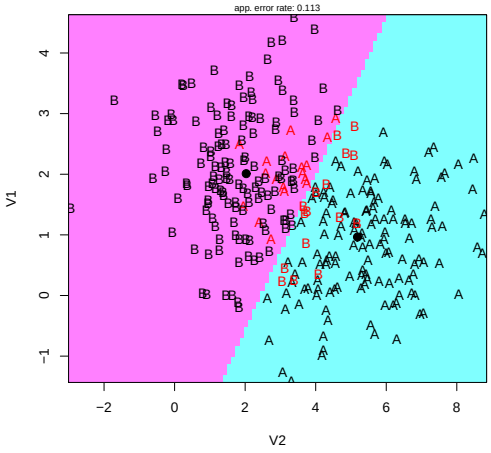


(i) $Ael(\mathrm{A})$

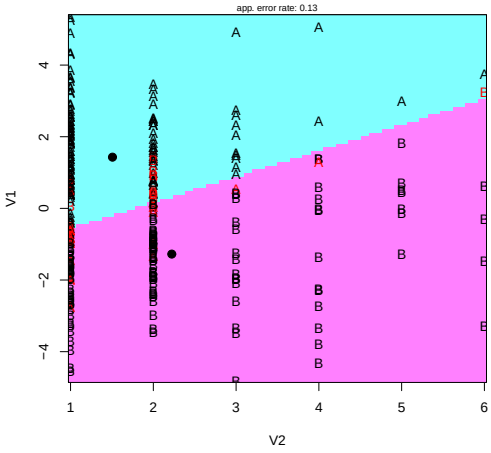


(j) $Ael(\mathrm{B})$

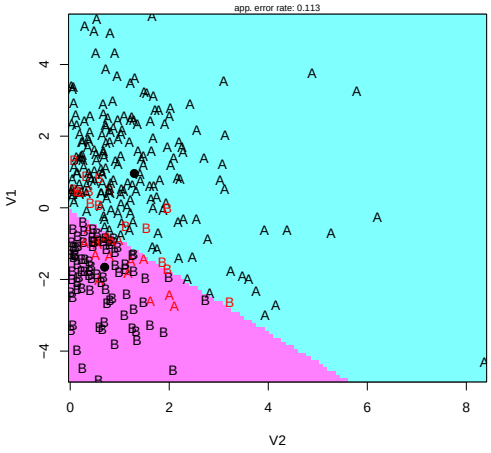
2.2. att.: Ģenerēto izlašu grupu $q - q$ grafiki.



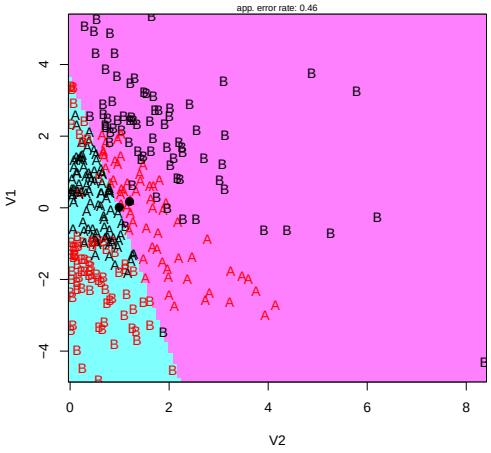
(a) *Anorm*



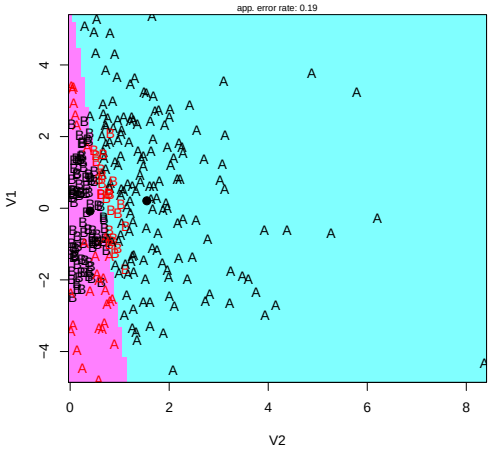
(b) *Adiskr*



(c) *Alin*



(d) *Akv*



(e) *Ael*

2.3. att.: Izlašu dalījums grupās, pielietojot lineāro diskriminantu analīzi.

2.3. tabula: *Anorm* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - lineārā diskriminantu analīze. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.8518	0.003281	0.8715	0.002350	0.8617	0.000341
N =50	0.8705	0.001465	0.8669	0.001645	0.8687	0.000134
N =100	0.8746	0.001178	0.8707	0.001258	0.8727	0.000077
N =200	0.8757	0.000413	0.8773	0.000401	0.8765	0.000016
N =500	0.8773	0.000216	0.8776	0.000229	0.8775	0.000015
N =1000	0.8794	0.000127	0.8793	0.000133	0.8793	0.000011
N =100000	0.8804	0.000001	0.8797	0.000001	0.8800	0.000001

tuvu īstai precizitātei. Tika pamanīts, ka pie tik liela datu apjoma nav atšķirības starp precizitātes novērtēšanu no modelēšanas datiem un pārbaudes datiem, tāpēc tā tika novērtēta pēc tiem pašiem datiem, kuri tika izmantoti konstruējot diskriminācijas robežu. Procedūra tika atkārtota 1000 reizes. Izrādījās, ka precizitātes novērtējums palika gandrīz nemainīgs. Tādā veidā novērtēta precizitāte acīmredzami ir visaugstākā precizitāte, kura var tikt sasniegta dotajiem sadalījumiem.

Skaidrs, ka izmantojot klasifikatora konstruēšanai izlases ar mazāko apjomu, precizitāte nebūs tik augsta. Lai saprastu, pie kādiem izlases apjomiem metode strādā pietiekoši labi, visiem 5 piemēriem tika ģenerēti dati ar apjomu $M = 100000$, no tiem ņemtas izlases ar apjomiem $N = 30, 50, 100, 200, 500, 1000$. Dotajā gadījumā datu nav pietiekoši daudz, tāpēc klasifikatora konstruēšanai tika izmantoti N dati, bet precizitātes novērtēšanai atlikušie $M - N$ dati. Šādi novērtēta precizitāte arī būs pietiekoši tuva konkrēta klasifikatora īstai precizitātei, tikai vairs nebūs tik augsta kā pie $N = 100000$. Mērķis ir izdarīt secinājumu par izlases apjoma ietekmi, tāpēc procedūra jāatkārto pietiekoši daudz reizes un jāizrēķina vidējā kopējā un grupu precizitāte, t.i. $\widehat{OA} = \frac{1}{1000}\sum_{j=1}^{1000} OA_j$ un $\widehat{A_k} = \frac{1}{1000}\sum_{j=1}^{1000} A_{kj}$, un to dispersijas novērtējumi, t.i. $S(OA) = \frac{1}{1000}\sum_{j=1}^{1000}(OA_j - \widehat{OA})^2$ un $S(A_k) = \frac{1}{1000}\sum_{j=1}^{1000}(A_{kj} - \widehat{A_k})^2$. Darbā tika izvēlēts atkārtojumu skaits vienāds ar 1000.

2.3., 2.4., 2.5., 2.6. un 2.7. tabulās ir atspoguļoti datu simulāciju rezultāti, kas parāda, ka lineārā diskriminantu analīze dod labus rezultātus un var tikt veiksmīgi izmantota pat gadījumos, kad pieņēmumi par normalitāti un kovariāciju matricu vienādību netiek

2.4. tabula: *Adiskr* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - lineārā diskriminantu analīze. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.8697	0.001987	0.8497	0.002063	0.8603	0.000113
N =50	0.8734	0.000565	0.8567	0.000706	0.8656	0.000026
N =100	0.8741	0.000360	0.8601	0.000407	0.8675	0.000014
N =200	0.8746	0.000330	0.8594	0.000345	0.8675	0.000015
N =500	0.8758	0.000166	0.8583	0.000164	0.8676	0.000015
N =1000	0.8752	0.000077	0.8600	0.000100	0.8681	0.000013
N =1000000	0.8936	0.000000	0.8430	0.000000	0.8702	0.000000

2.5. tabula: *Alin* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - lineārā diskriminantu analīze. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.8108	0.003648	0.8900	0.002297	0.8382	0.000737
N =50	0.8323	0.001496	0.8848	0.001181	0.8504	0.000261
N =100	0.8299	0.000705	0.8945	0.000583	0.8521	0.000104
N =200	0.8343	0.000334	0.8932	0.000341	0.8545	0.000055
N =500	0.8339	0.000182	0.8958	0.000155	0.8548	0.000035
N =1000	0.8364	0.000097	0.8948	0.000088	0.8556	0.000027
N =100000	0.9118	0.000001	0.7927	0.000009	0.8702	0.000001

2.6. tabula: *Akv* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - lineārā diskriminantu analīze. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.6519	0.014485	0.4488	0.005286	0.5505	0.002982
N =50	0.6169	0.007753	0.4846	0.004145	0.5509	0.002030
N =100	0.6194	0.005784	0.4837	0.004020	0.5514	0.001794
N =200	0.6265	0.004839	0.4789	0.004398	0.5528	0.001360
N =500	0.6298	0.000757	0.4765	0.004528	0.5533	0.000879
N =1000	0.6255	0.000956	0.4948	0.003797	0.5602	0.000667
N =1000000	0.6435	0.000026	0.4794	0.000029	0.5617	0.000054

2.7. tabula: Ael piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - lineārā diskriminantu analīze. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.6657	0.005069	0.9286	0.003468	0.7763	0.000701
N =50	0.6502	0.002127	0.9553	0.001008	0.7786	0.000273
N =100	0.6624	0.001767	0.9509	0.000865	0.7837	0.000195
N =200	0.6644	0.000704	0.9548	0.000351	0.7863	0.000076
N =500	0.6638	0.000334	0.9576	0.000147	0.7861	0.000044
N =1000	0.6609	0.000157	0.9602	0.000072	0.7847	0.000028
N =100000	0.7534	0.000002	0.8611	0.000004	0.7987	0.000001

izpildīti, bet robežas starp klasēm ir lineāras vai tuvas lineārām. Tāpēc LDA precizitāte *Anorm*, *Alin* un *Adiskr* piemēru gadījumā ir pietiekoši augsta, ko nevar teikt par *Akv* un *Ael* piemēriem. *Ael* izlasēm mazākā klase bieži vien tiek ignorēta un visi ieraksti pieskaitīti pie grupas ar lielāku apjomu, bet *Akv* izlases vispār nevar tikt sadalītas precīzi, jo piemēra konstrukcija ir tāda, ka viena klase atrodas citas klases iekšā un nevar tikt atdalīta ar vienu hiperplakni. Jāatzīmē arī tas, ka piemēriem *Anorm*, *Adiskr* un *Alin* precizitāte nav tik ļoti atkarīga no izlases apjoma kā *Akv* un *Ael*. *Akv* un *Ael* gadījumos precizitātes novērtējumu vidējās vērtības izkliede ir lielāka nekā pirmajiem trim piemēriem. Interesants ir fakts, ka, izlases apjomam pieaugot, grupu precizitāte svārstās, bet kopējā precizitāte visu laiku aug.

2.3. Kodolu diskriminantu analīze

Kā tika minēts iepriekšējā nodaļā, diskriminantu funkcija tiek primāri definēta kā grupas nosacīta varbūtība (2.1.3). Daudzos gadījumos grupu sadalījuma blīvuma funkcijas izskatās pārāk sarežģīti, lai pastāvētu iespēja teorētiski noteikt diskriminantu funkcijas vienādojumu. Duong [10] piedāvā cīnīties ar šo problēmu, ar kodolu gludināšanas palīdzību. Diskriminantu funkcijas vietā joprojām tiek izmantota nosacītā varbūtība, vienīgā atšķirība ir tā, ka teorētiskā sadalījuma blīvuma funkcija tiek aizstāta ar tās kodolu gludināto versiju.

Definīcija 4. [11] Par funkcijas $p(x)$ novērtējumu ar kodolu gludināšanu sauc

$$\tilde{p}(x) = \frac{1}{N}\sum_{i=1}^{N} K_H(x - X_i), \tag{2.3.1}$$

kur $K_H(x) = |H|^{-1}K(H^{-1}x)$, K ir kodols, H ir joslas platumu matrica, H^{-1} ir joslas platumu matricas inversā matrica, bet $|H^{-1}|$ ir tās determinants.

Definīcija 5. [11] Par kodolu $K(x)$, $x \in \mathbb{R}^p$ sauc funkciju, kurai izpildās

$$\int_{\mathbb{R}^p} K(x)\mathrm{dx} = 1. \tag{2.3.2}$$

Piezīme 7. *Parasti par kodolu tiek izvēlēta simetriska sadalījuma blīvuma funkcija.*
Tāpat kā vienas dimensijas gadījumā vislielākā nozīme ir joslas platuma H izvēlei. Daudzi autori, piemēram [12], piedāvā izmantot kodola vietā daudzdimensiju standartizēto normālo sadalījumu

$$K(x) = (2\pi)^{-1}exp(-\frac{1}{2}x^t x).$$

Definīcija 6. [8]
Par kodola diskriminantu funkciju sauc

$$\tilde{g}(x) = \tilde{p}(x|\mathcal{G}_i)\pi(\mathcal{G}_i),$$

kur $\tilde{p}(x)$ ir sadalījuma blīvuma funkcijas novērtējums ar kodolu gludināšanu (2.3.1) un $\pi(\mathcal{G}_i)$ ir piederības pie klases $\mathcal{G}_i$ varbūtība. Kā jau tika minēts iepriekš, svarīgākais uzdevums ir pareizi izvēlēties joslas platumu matricu. To risina ar daudziem paņēmieniem: ievietošanas metodes, piemēram, "rule of thumb" (pieņem, ka dati ir normāli sadalīti, novērtē kovariāciju matricu un pareizina ar koeficientiem no tabulām), butstraps, krosvalidācija u.t.t. Plašāku aprakstu sniedz literatūras avoti [13], [14] un [12]. Darbā tiks izmantota ar krosvalidācijas un ievietošanas metodēm iegūta joslas platumu matrica. Apskatīsim krosvalidācijas metodi. Šī metode minimizē integrēto kvadrātisko kļūdu.

Definīcija 7. [12] Ja $p(x)$ ir kāda nezināma funkcija, bet $\tilde{p}(x)$ ir tās novērtējums, tad par funkcijas $\tilde{p}(x)$ integrēto kvadrātisko kļūdu sauc

$$ISE(H) = \int_{R^p} \{\tilde{p}(x) - p(x)\}^2 \mathrm{dx}$$

Apgalvojums 8. *[12] Joslas platumu matrica ar krosvalidācijas metodi tiek meklēta kā*

$$H_{ISE} = \underset{H\in\mathcal{H}}{argmin} ISE(H) =$$

$$= \underset{H\in\mathcal{H}}{argmin} \frac{1}{N^2|H|} \sum_{i=1}^{N}\sum_{j=1}^{N} K \star K\{H^{-1}(X_j - X_i)\} - \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} K_H(X_j - X_i),$$

kur $\mathcal{H}$ ir visu pozitīvi definitu simetrisku matricu telpa, H ir joslas platumu matrica, K ir kodols (2.3.2), $K \star K(u)$ ir divu funkciju konvolūcija

$$K \star K(u) = \int K(u-v)K(v)dv.$$

Pierādījums. $ISE(H) = \int \{\tilde{p}(x) - p(x)\}^2 dx = \int \tilde{p}^2(x)dx - 2\int \tilde{p}(x)p(x)dx + \int p^2(x)dx$
Pirmais saskaitāmais var tikt izrēķināts, izmantojot datus, trešais nav atkarīgs no H un tāpēc var tikt ignorēts, bet otrais ir $\mathbb{E}(\tilde{p}(x))$ (pamatots [7]) un tiek novērtēts ar "vienu-izmest-ārā" krosvalidāciju kā

$$\widehat{\mathbb{E}}(\tilde{p}(x)) = \frac{1}{N}\sum_{i=1}^{N} \tilde{p}_{-i}(X_i),$$

kur $\tilde{p}_{-i}$ blīvuma funkcijas novērtējums, ko iegūst atmetot i-to novērojumu un ir vienāds ar

$$\tilde{p}_{-i}(x) = \frac{1}{N-1}\sum_{i=1, i\neq j}^{N} K_H(X_j - x).$$

Pirmais saskaitāmais var tikt aprēķināts no datiem (skatīt [12]) kā

$$\int \tilde{p}^2(x)dx = \frac{1}{N_2|H|}\sum_{j=1}^{N}\sum_{j=1}^{N} K \star K\{H^{-1}(X_j - X_i)\}.$$

Tādējādi,

$$ISE(H) = \frac{1}{N^2|H|}\sum_{j=1}^{N}\sum_{j=1}^{N} K \star K\{H^{-1}(X_j - X_i)\} - \frac{2}{N(N-1)}\sum_{j=1}^{N}\sum_{j=1, j\neq i}^{N} K_H(X_j - X_i).$$

□

Atšķirībā no krosvalidācijas metodes, ievietošanas metode minimizē funkcijas novērtējuma vidējo integrēto kvadrātisko kļūdu un izvēlas $H_{MISE} = \underset{H\in\mathcal{H}}{\operatorname{argmin}} MISE(H)$.

Definīcija 8. [12]
Ja $p(x)$ ir kāda nezināma funkcija, bet $\tilde{p}(x)$ ir tās novērtējums, tad par funkcijas $\tilde{p}(x)$ vidējo integrēto kvadrātisko kļūdu sauc

$$MISE(H) = \mathbb{E}\int_{R^p} \{\tilde{p}(x) - p(x)\}^2 \mathrm{dx}.$$

Tāda matricas H izvēle ir grūti realizējama praksē, tāpēc funkcijas $\tilde{p}(x)$ kļūdas mērīšanai izmanto asimptotisko vidējo integrēto kvadrātisko kļūdu.

Definīcija 9. [12]

Ja $p(x)$ ir kāda nezināma funkcija, bet $\tilde{p}(x)$ ir tās novērtējums, tad par funkcijas $\tilde{p}(x)$ asimptotisko vidējo integrēto kvadrātisko kļūdu sauc

$$AMISE(H) = n^{-1}(4\pi)^{-p/2}|H|^{-1/2} + \frac{1}{4}(vechH)^T \Psi_4 (vechH),$$

kur Ψ_r ir matrica, kuras elementi $\psi_r = \int_{R^p} p^{(r)}(x)p(x)dx$, $r = (r_1, r_2, ..., r_p)$, $|r| = \sum_{i=1}^{p} r_i$ un $p^{(r)}(x) = \frac{d^{|r|}}{d_{x_1}^{r_1},...,d_{x_p}^{r_p}} p(x)$. $vechH$ ir operators, kurš pārveido matricu par vektoru, ņemot vērā tikai tos elementus, kuri atrodas uz matricas diagonāles un virs tās. Piemēram, divu dimensiju gadījumā, ja matrica $H = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix}$, tad $vechH = (h_{11}, h_{12}, h_{22})^t$.

H izvēlas tā, lai minimizētu $AMISE(H)$ novērtējumu, t.i. $\widehat{H}_{AMISE} = \underset{H \in \mathcal{H}}{\operatorname{argmin}} \widehat{AMISE}(H)$, kurš ir vienāds ar

$$\widehat{AMISE}(H) = n^{-1}(4\pi)^{-p/2}|H|^{-1/2} + \frac{1}{4}(vechH)^T \widehat{\Psi_4} (vechH).$$

Ņemot vērā, ka $\psi_r = \mathbb{E}p^r(x)$, $x \sim p(x)$, vislabākais ψ_r novērtējums ir $\widehat{\psi}_r = \frac{1}{N}\sum_{i=1}^{N} \tilde{p}^{(r)} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} K_C^{(r)}(X_i - X_j)$, kur $\tilde{p}$ ir blīvuma funkcijas kodolu novērtējums (2.3.1) ar joslas platumu matricu C.

Nepieciešams tikai izvēlēties piemērotu C, ko sauc par pilota joslas platumu matricu. Lai varētu meklēt C analītiski, tiek pieņemts, ka šī matrica ir diagonālmatrica $c^2 I$. Lai koriģētu neprecizitātes, kuras rodas ierobežojot matricas C vispārējo formu, joslas platumu C^* meklēšanai pielieto divu veidu datu transformācijas [15]:

skalēto

$$X^* = S^{-1/2}X, \tag{2.3.3}$$

kur S ir izlases kovariāciju matricas novērtējums un S_D ir diagonālmatrica, kas sastāv no dispersiju novērtējumiem, vai sfērisko

$$X^* = S_D^{-1/2}X. \tag{2.3.4}$$

C tiek iegūta ar inversām transformācijām $C = S^{1/2}C^*S^{1/2}$ un $C = S_D^{1/2}C^*S_D^{1/2}$. Optimālo C^* meklē minimizējot asimptotisko vidējo kvadrātisko kļūdu summu, kuras analītiskais izskats ir atrodams [15].

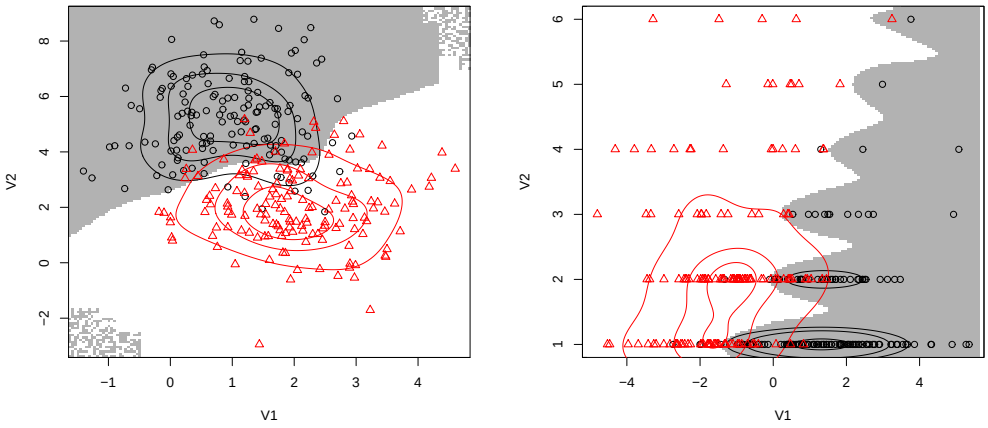
2.8. tabula: *Anorm* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - kodolu diskriminantu analīze. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.8577	0.002780	0.8559	0.003043	0.8568	0.000345
N =50	0.8699	0.001763	0.8598	0.002064	0.8648	0.000154
N =100	0.8708	0.000747	0.8727	0.000832	0.8718	0.000045
N =200	0.8757	0.000477	0.8718	0.000495	0.8737	0.000027
N =500	0.8772	0.000281	0.8770	0.000267	0.8771	0.000013
N =1000	0.8775	0.000185	0.8768	0.000180	0.8771	0.000012
N =100000	0.9149	0.000001	0.8014	0.000002	0.8749	0.00000

2.4. Kodolu diskriminantu analīzes ilustrācija

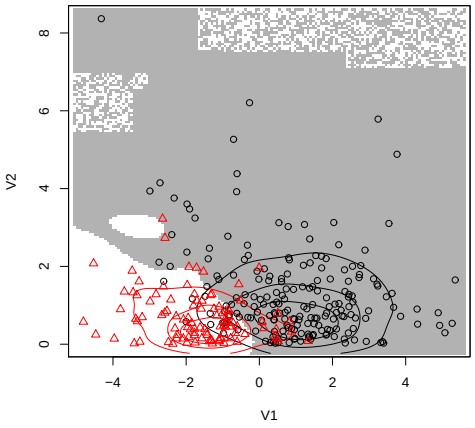
Lai parādītu, kā strādā kodolu diskriminantu analīze, izmantosim iepriekš izmantotus 5 piemērus: *Anorm*, *Adiskr*, *Alin*, *Akv* un *Ael*. Lai pielietotu diskriminantu analīzes kodolu variantu nav nepieciešamas pārbaudīt pieņēmumus par kovariāciju matricu vienādību un grupu sadalījumu atbilstību normālajam. Diskriminantu funkcijas tiek iegūtas, izmantojot nosacīto varbūtību novērtējumu ar kodolu gludināšanu. Robežas starp klasēm, analoģiski kā tas tika darīts lineārajā diskriminantu analīzē, tiek iegūtas atskaitot vienu diskriminantu funkciju no otras. Lai izvēlētos joslas platumu matricas novērtēšanas veidu, konstruēsim modeļus, izmantojot krosvalidācijas metodi un ievietošanas metodes ar sfērisko (2.3.4) un skalēto (2.3.3) transformācijām priekš pilota matricas novērtēšanas, un novērtēsim to grupu un kopējo precizitāti, simulējot datus. Abas ievietošanas metodes deva vienādus, ļoti tuvus krosvālidācijai, rezultātus. Tomēr, pielietojot krosvalidāciju, dažām izlasēm ir iegūstams neliels precizitātes uzlabojums. Šo apsvērumu dēļ modeļa konstruēšanai un rezultātu salīdzināšanai tiks izmantota tieši šī metode.

Kodolu diskriminantu analīzes pielietošana piemēriem (*Anorm*, *Adiskr*, *Alin*, *Akv*, *Ael*) ir redzama 2.4. attēlā, kurā ir attēloti gan diskriminācijas robežas, gan grupu sadalījuma blīvuma funkciju novērtējumu, ar kodolu gludināšanu, līmeņlīnijas, izlasēm ar apjomu $N = 300$. 2.8., 2.9., 2.10., 2.11. un 2.12. tabulās ir atspoguļotas precizitātes novērtējumu vidējās vērtības un empīriskās dispersijas pie dažādiem izlases apjomiem. Tāpat kā lineāras diskriminantu analīzes gadījumā, katru reizi tika simulēti $M = 100000$ dati, no tiem

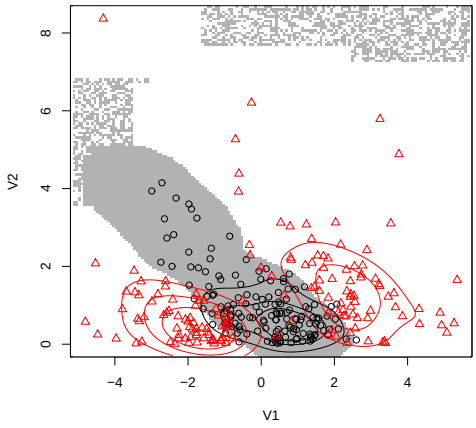


(a) *Adiskr*

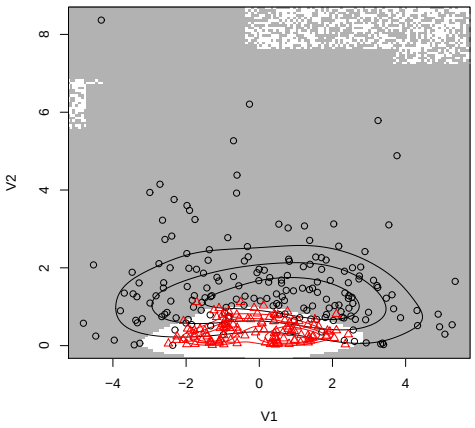
(b) *Adiskr*



(c) *Alin*



(d) *Akv*



(e) *Ael*

2.4. att.: Izlašu dalījums grupās, pielietojot kodolu diskriminantu analīzi.

2.9. tabula: *Adiskr* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - kodolu diskriminantu analīze. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.8943	0.001508	0.7553	0.008388	0.8292	0.001079
N =50	0.9176	0.000762	0.7342	0.005967	0.8317	0.000789
N =100	0.9430	0.000355	0.7003	0.003783	0.8297	0.000459
N =200	0.9564	0.000201	0.6718	0.002736	0.8236	0.000358
N =500	0.9636	0.000060	0.6594	0.001187	0.8218	0.000182
N =1000	0.9691	0.000046	0.6547	0.000798	0.8230	0.000112
N =1000000	0.9601	0.000018	0.7108	0.000102	0.8447	0.000008

2.10. tabula: *Alin* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - kodolu diskriminantu analīze. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.7829	0.005015	0.8584	0.004524	0.8090	0.001528
N =50	0.8146	0.002157	0.8675	0.001413	0.8329	0.000619
N =100	0.8208	0.001330	0.8797	0.001036	0.8411	0.000235
N =200	0.8286	0.000843	0.8819	0.000680	0.8469	0.000143
N =500	0.8321	0.000341	0.8881	0.000294	0.8510	0.000065
N =1000	0.8396	0.000204	0.8856	0.000193	0.8547	0.000041
N =100000	0.9130	0.000009	0.7999	0.000128	0.8736	0.000005

2.11. tabula: *Akv* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - kodolu diskriminantu analīze. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.8931	0.004737	0.7008	0.008964	0.7972	0.001866
N =50	0.9126	0.002420	0.7410	0.004443	0.8269	0.000899
N =100	0.9316	0.000684	0.7868	0.001863	0.8591	0.000259
N =200	0.9388	0.000226	0.8169	0.000643	0.8778	0.000115
N =500	0.9370	0.000150	0.8444	0.000240	0.8908	0.000020
N =1000	0.9341	0.000073	0.8543	0.000099	0.8943	0.000013
N =100000	0.9351	0.000002	0.8694	0.000045	0.9025	0.000009

2.12. tabula: Ael piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - kodolu diskriminantu analīze. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =50	0.7634	0.005027	0.9573	0.001714	0.8450	0.001042
N =50	0.7670	0.002521	0.9692	0.000798	0.8521	0.000494
N =100	0.8017	0.001341	0.9715	0.000283	0.8731	0.000297
N =200	0.8241	0.000567	0.9732	0.000165	0.8867	0.000128
N =500	0.8546	0.000254	0.9664	0.000099	0.9012	0.000046
N =1000	0.8654	0.000095	0.9652	0.000036	0.9067	0.000020
N =100000	0.9103	0.000040	0.9371	0.000015	0.9216	0.000015

paņemta izlase ar apjomu N, no $M-N$ datiem novērtēta precizitāte, procedūra atkārtota 1000 reizes. Pie $N = 100000$ precizitāte tika novērtēta, izmantojot modelēšanas datus.

Var pamanīt, ka kodolu diskriminantu analīze nedod rezultātu uzlabojumu gadījumos, kad starp datiem pastāv lineārā atkarībā vai ir spēkā lineārās diskriminantu analīzes pieņēmumi, t.i. *Anorm*, *Adiskr*, *Alin* izlasēm. Dažreiz tā ir pat nedaudz neprecīzāka par lineāro diskriminantu analīzi. Izlasēm *Akv* un *Ael*, kur dalījums grupās nav lineārs, diskriminantu analīzes kodolu gludināšanas versija dod manāmu rezultātu uzlabojumu, proti, kopējās precizitātes palielināšanu.

3. Datu izraces klasifikatori

3.1. Klasifikācijas koki

Klasifikācijas koks, datu ieguvē, ir plaši izmantota metode. Tā ir neparametriska un ļoti viegli interpretējama, tāpēc īpašu atzinību ir ieguvusi medicīnā, mārketingā un lēmumu pieņemšanas teorijā. Koka uzdevums ir sadalīt pazīmju telpu daudzdimensiju taisnstūros, kuru skaldnes ir paralēlas koordinātu asīm. Visi ieraksti, kuri atrodas vienā taisnstūrī, tiek piekārtoti kādai konkrētai klasei. Lēmumu koks T ir grafs ar vairākām virsotnēm, ko apzīmēsim ar t, kas savā starpā ir saistītas ar lokiem. Pastāv trīs tipa virsotnes: sakne(t_0), iekšējais mezgls un lapa. Koka sakne − virsotne, kurai nav ieejošu loku un ir viens vai vairāki izejoši loki. Iekšējais mezgls − virsotne, kurai ir viens ieejošs loks un vismaz divi izejoši loki. Koka lapa − virsotne, kurai ir viens ieejošais loks un nav izejošo loku. 3.1. attēlā ir parādīts klasifikācijas koka piemērs un telpas R^2 dalījums taisnstūros, ar koka palīdzību. Koka konstruēšanai tika izmantota *Alin* piemēra izlase ar apjomu $N = 300$. $\{V1, V2\}$ ir novērojumu pazīmju kopa. Dotajā gadījumā pazīme $V1$ tiek pirmā izmantota datu dalīšanai. Pirmajā solī ieraksti, kuriem $V1 < (-0.392241)$ tiek pieskaitīti pie grupas B, bet visi pārējie pie grupas A.

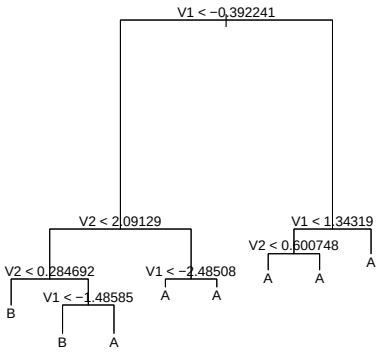
Parasti teorijā tiek apskatīti koki, kuriem katrai virsotnei ir ne vairāk kā divi izejošie loki. Pārējie gadījumi var tikt reducēti uz šādu koku tipu. Visi n_k ieraksti, kuri atrodas kādā konkrētā lapā t_k, tiek piekārtoti klasei, kura maksimizē ierakstu daļu dotajā lapā.

Definīcija 10. [1] Ar klasifikācijas koku iegūtais īstās grupas G novērtējums $\widehat{G}$ lapā t_k ir vienāds ar

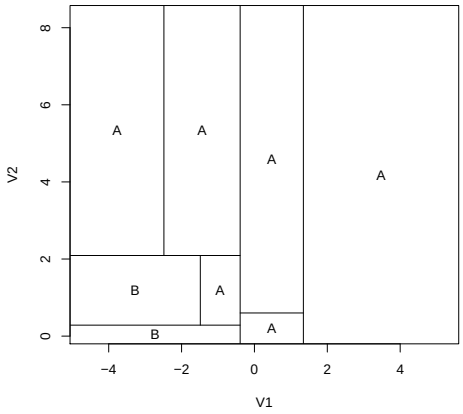
$$\widehat{G}_{t_k} = \underset{i=1,\ldots,g}{\operatorname{argmax}} \widehat{p}_{ki}, \tag{3.1.1}$$

kur

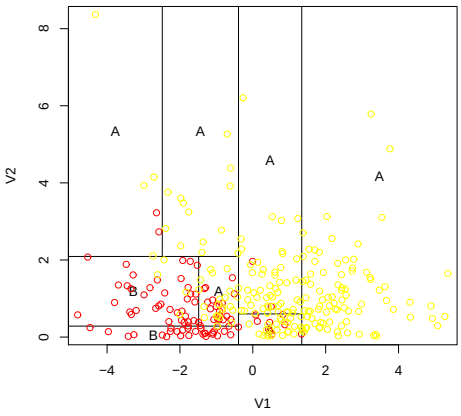
$$\widehat{p}_{ki} = \frac{1}{n_k} \sum_{j=1}^{n_k} I(G_j = \widehat{G}(X_j)).$$



(a) *Alin* klasifikācijas koks T_{lin}



(b) Telpas R^2 dalījums apgabalos ar T_{lin}



(c) *Alin* izlase kopā ar T_{lin} dalījumu

(3.1.2)

Koka konstruēšana notiek rekursīvi, katrā mezglā izvēloties gan nākamo dalīšanas pazīmi, gan tās vērtību. Izvēle notiek, minimizējot kādu informācijas kritēriju.

Definīcija 11. [?] Par nepareizās klasifikācijas kļūdu sauc

$$ME = \frac{1}{n_k} \sum_{j=1}^{n_k} I(G_j \neq \widehat{G}(X_j)) = 1 - \widehat{p}_{ki},$$

kur $\widehat{p}_{ki}$ ir definēts (10), $I(G_j \neq \widehat{G}(X_j))$ ir īstās un prognozētās klases nesakritības indikatorfunkcija.

Definīcija 12. [16] Par Gini indeksu sauc

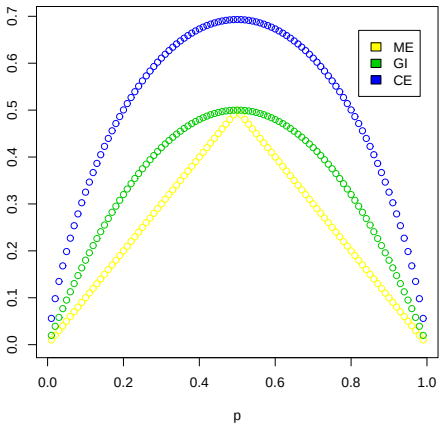
$$GI = \sum_{i=1}^{g} \widehat{p}_{ki}(1 - \widehat{p}_{ki}).$$

Definīcija 13. [1] Par kross-entropiju sauc

$$CE = -\sum_{i=1}^{g} \widehat{p}_{ik} \log \widehat{p}_{ki}.$$

Divu grupu gadījumā šie lielumi pieņem vienkāršāku formu: $ME = 1 - \max(p, 1-p)$, $GI = 2p(1-p)$ un $CE = -p \ln p - (1-p) \ln (1-p)$. Uzzīmējot visu trīs funkciju atkarību no p, var pamanīt, ka tie sasniedz savu maksimumu pie vienādas p vērtības un to izskats ir ļoti līdzīgs, tāpēc bieži vien visu informācijas mēru lietošana dod vienādus rezultātus (skatīt 3.2. attēlu).

Kļūdas minimizēšana notiek sekojošā veidā - katrā mezglā algoritms aprēķina kādu informācijas mēru katram ierakstam (kalkulācijas notiek pa visām pazīmēm un ierakstiem, kuri atrodas šajā mezglā). Pazīme un ieraksts, kuri minimizē informācijas mēra vērtību tiek izvēlēti kā datu dalījuma slieksnis. Jāpievērš uzmanība tam, ka viena un tā pati pazīme var tikt izmantota datu dalīšanai vairākas reizes. Koku algoritmu trūkums ir tāds, ka koks nekad neatgriežas iepriekšējos mezglos, tādā veidā kļūdas minimizācija notiek lokāli katrā mezglā, nevis pa visu koku. Ir acīmredzami, ka algoritms var strādāt līdz brīdim, kad būs panākta nulles kļūda, t.i. katra lapa saturēs tikai vienas grupas ierakstus. Taču tad modelis būs pārāk labi pielāgots datiem un nedos labus pārbaudes rezultātus, tāpēc



3.2. att.: Informācijas mēru atkarība no p divu grupu gadījumā.

katrs mezgls kļūst par lapu tikai tad, kad spēkā ir kāds speciāls nosacījums. Algoritma apstāšanās nosacījumi katrā mezglā var būt dažādi, piemēram, ir sasniegts iepriekš noteikts informācijas mēra izmaiņas slieksnis un turpmāka dalīšana nedod būtisku informācijas mēra palielināšanu, koka lielums ir sasniedzis maksimāli pieļaujamo, visi ieraksti mezglā pieder vienai klasei vai ierakstu skaits mezglā ir sasniedzis iepriekš noteiktu minimumu. Parasti vidējās informācijas mēra izmaiņas tiek ņemtas vērā.

Definīcija 14. [2] Par informācijas mēra i izmaiņām lapā t_k sauc

$$\Delta i(t_k) = i(t_k) - \sum_{j=1}^{2} \pi_j \cdot i(t_{k+j}),$$

kur $\pi_j = \frac{|t_{k+j}|}{|t_k|}$ ir t_k mezgla ierakstu daļa, kura ir iekļauta t_{k+j} apakšmezglā, $i(t_k)$ ir mezgla t_k informācijas mērs un $i(t_{k+j})$ir apakšmezgla t_{k+j} informācijas mērs.

Ir spēkā sekojošs rezultāts:

Apgalvojums 9. *[2] Ja informācijas mēra funkcija i ir stingri ieliekta, tad informācijas mēra izmaiņas Δi ir nenegatīvas.*

Pierādījums. Izmantojot Jensena nevienādību, iegūstam

$$\sum_{j=1}^{2} \pi_j \cdot i(t_{k+j}) \leq i(\sum_{j=1}^{2} \pi_j \cdot (t_{k+j})) = i(t_k)$$

Šis apgalvojums apliecina augstāk teikto - koks var tikt audzēts bezgalīgi, kamēr nesasniegs nulles kļūdu. Tāpēc, lai izvairītos no modeļa pārmērīgas pielāgošanās datiem, tiek izmantota koku apgriešana (prunning) [17]. Tā tiek realizēta, konstruējot maksimāli liela

izmēra koku T^{max} un minimizējot lielumu, kurš ņem vērā ne tikai kopējo kļūdas līmeni, bet arī koka izmēru (mezglu skaitu).

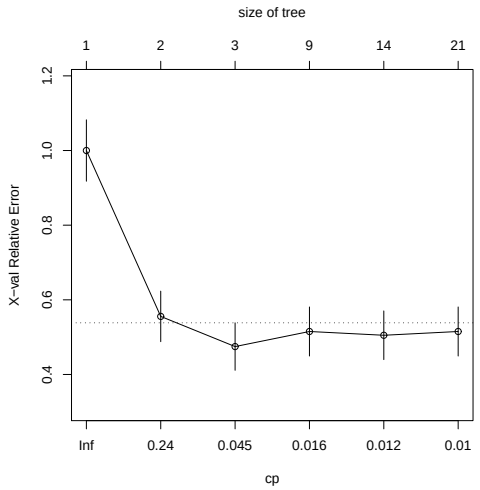
Definīcija 15. [1] Par sabalansēta (angliski penalized) koka T riska funkciju sauc

$$R_\alpha(T) = R(t) + \alpha \cdot |T|,$$

kur R(2.1.2) ir pirmajā nodaļā definēta klasifikatora riska funkcija, $|T|$ ir koka T izmērs.

Par apgriezto koku sauc koku $T^p = \underset{T \in \tau}{\operatorname{argmin}} R_\alpha(T)$, kur τ ir koka T^{max} visu apakškoku kopa. Galvenais uzdevums ir piemeklēt piemērotu sabalansēšanas lielumu α. Dažreiz piedāvā ņemt $\alpha = 2(d-1)$, kur d ir parametru skaits. Eksistē arī citi α novērtēšanas paņēmieni, kuri var būt apskatīti, piemēram, [2]. Vienkāršības dēļ ļoti bieži ņem $\alpha = 0$, tad kritērijs 15 vienkārši minimizē klasifikatora risku. Vienīgā problēma ir tāda, ka riska novērtēšana, izmantojot modelēšanas datus, noved pie pārāk optimistiskiem rezultātiem, bet datu apjoms ne vienmēr atļauj sadalīt izlasi modelēšanas un pārbaudes daļās. Lai novērtētu risku precīzāk, izmanto krosvalidācijas metodi, kura tiks aprakstīta 3.nodaļā. Ilustrēsim koka apgriešanu ar piemēru, izmantojot *Alin* izlasi.

Piemērs 1. Apskatīsim kā procentuāli mainās, ar krosvalidācijas palīdzību novērtēts kopējais kļūdas līmenis, atkarībā no koka izmēra. Rezultāti ir parādīti 4.4. attēlā. Palielinot koka izmēru no 1 līdz 2 ir novērots diezgan būtisks relatīvās kļūdas samazinājums. Tālāk relatīvā kļūda sāk svārstīties, sasniedzot savu lokālo minimumu pie koka izmēra, kas vienāds ar 3. Tālākais koka izmēra palielinājums noved pie kļūdas svārstībām, tāpēc visracionālāk kā klasifikatoru lietot koku ar izmēru 2.



3.3. att.: Kopējās kļūdas procentuālās izmaiņas atkarībā no koka izmēra.

Pielietojot klasifikācijas kokus uz datu piemēriem, izvēlēsimies koku izmēru, kurš minimizē krosvalidācijas kļūdu. Datu dalīšanai mezglos kā informācijas mēru izmantosim Gini indeksu.

3.2. Klasifikācijas koku ilustrācija

Lai ilustrētu klasifikācijas koku metodi, izmantosim iepriekš minētus piemērus: *Anorm*, *Adiskr*, *Alin*, *Akv* un *Ael*. Lai pielietotu datu izraces metodes nav nepieciešams pārbaudīt jebkādus pieņēmumus. Klasifikators tiek būvēts neparametriski, balstoties tikai un vienīgi uz sakarībām, kuras ir novērotas datos. Kā tika minēts iepriekš, koku konstruēšanai tiks izmantots Gini informācijas mērs, koki tiks apgriezti, pielietojot krosvalidāciju. Klasifikācijas koku pielietošana izlasēm ar apjomu $N = 300$ no piemēriem *Anorm*, *Adiskr*, *Alin*, *Akv* un *Ael* ir parādīta 3.4. attēlā, kurā ir redzamas gan diskriminācijas robežas, gan datu dalījums grupās. 3.1., 3.2., 3.3., 3.4. un 3.5. tabulās ir atspoguļotas grupu un kopējās precizitātes novērtējumu vidējās vērtības izmaiņas, atkarībā no izlases apjoma. Simulācijas tika veiktas tieši tādā pašā veidā kā lineārās un kodolu diskriminantu analīzes gadījumā. Jāatzīst, ka izlases apjomam, klasifikācijas koku gadījumā, ir lielāka nozīme nekā diskriminantu analīzes gadījumā. Tas ir saprotams, jo klasifikators neveic nekādu pieņēmumu par datu sadalījumu, bet mēģina vienkārši mācīties no to struktūras. Jo vairāk datu, jo vieglāk ir uzbūvēt kvalitatīvu klasifikatoru. Taču šī īpašība nevar tikt uzskatīta par nopietnu metodes trūkumu, jo datu izraces problemātikā parasti nav novērojams datu trūkums (izņemot datu izraces pielietošanu medicīnā). Turklāt, precizitāte nav pārāk slikta pat pie izlases apjoma $N = 50$, tā vienkārši dažos gadījumos ievērojami atšķiras no maksimālās precizitātes.

Apskatot piemērus ļoti uzskatāmi var redzēt klasifikācijas koku trūkumu. Klasifikācijas koki, kuri tiek konstruēti, izmantojot sabalansētas izlases, parāda vienādu precizitāti abām klasēm, bet nesabalansētām izlasēm, vienas grupas − tās, kura izlasē ir vairākumā − precizitāte ir manāmi lielāka par otrās grupas precizitāti. Tas nozīmē, ka klasifikācijas kokam ir raksturīgi ignorēt grupu ar mazāku apjomu, kas dabiski izriet no grupas novērtējuma definīcijas 3.3.2. Šī koku īpašība ļoti uzskatāmi parādās piemērā *Alin*, kur pirmā grupa lielākoties ir vairākumā. Ja kāda grupa ir pārstāvēta ar nelielu ierakstu skaitu, tad tā būs mazākumā arī daudzās lapās. Neskatoties uz metodes trūkumu, klasifikācijas koku precizitāte ir ļoti tuva kodolu diskriminantu analīzes precizitātei. Pārsteidzoši

3.1. tabula: *Anorm* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - klasifikācijas koki. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.8344	0.009665	0.8195	0.010015	0.8270	0.001345
N =50	0.8432	0.005518	0.8302	0.006817	0.8367	0.000680
N =100	0.8486	0.002612	0.8456	0.002969	0.8471	0.000212
N =200	0.8544	0.001698	0.8534	0.001783	0.8539	0.000103
N =500	0.8616	0.001287	0.8586	0.001293	0.8601	0.000070
N =1000	0.8645	0.001001	0.8634	0.001003	0.8639	0.000051
N =100000	0.8784	0.000008	0.8414	0.000003	0.8599	0.000000

3.2. tabula: *Adiskr* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - klasifikācijas koki. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.8129	0.006191	0.7922	0.005871	0.8032	0.001033
N =50	0.8255	0.004622	0.8211	0.004564	0.8234	0.000520
N =100	0.8306	0.003412	0.8374	0.003813	0.8337	0.000354
N =200	0.8344	0.002892	0.8457	0.002262	0.8397	0.000216
N =500	0.8396	0.002408	0.8518	0.001626	0.8452	0.000180
N =1000	0.8422	0.002365	0.8544	0.001605	0.8479	0.000156
N =100000	0.8935	0.000103	0.8225	0.000109	0.8607	0.000006

ir tas, ka izlasei *Anorm*, koka klasifikators deva gandrīz tādu pašu rezultātu kā lineārā vai kodolu diskriminantu analīze. Tādējādi, koku klasifikatori var pilnvērtīgi konkurēt ar diskriminantu analīzi precizitātes ziņā. To izmantošana dažos gadījumos ir lietderīgāka nekā statistisko klasifikatoru izmantošana. Viens no iemesliem varētu būt koku vieglā interpretācija. Apskatot koku, uzreiz var ļoti uzskatāmi redzēt, kāda pazīme visvairāk ietekmē ierakstu dalīšanu grupās — datu dalīšanai kā pirmā tiks izmantota šī pazīme. Var arī redzēt, kāda pazīmes vērtība var tikt izmantota, lai pirmā tuvinājumā sadalītu ierakstus vairākās grupās.

3.3. tabula: *Alin* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - klasifikācijas koki. Atkārtojumu skaits - 1000.

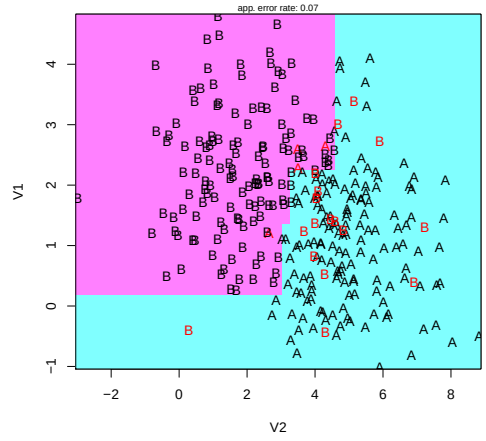
	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.7856	0.010589	0.8119	0.012763	0.7947	0.002742
N =50	0.7898	0.007251	0.8308	0.007103	0.8040	0.001334
N =100	0.8098	0.004182	0.8373	0.004000	0.8193	0.000707
N =200	0.8288	0.002810	0.8384	0.003002	0.8321	0.000461
N =500	0.8276	0.001425	0.8620	0.001731	0.8391	0.000207
N =1000	0.8472	0.000943	0.8484	0.001326	0.8476	0.000109
N =100000	0.9126	0.000026	0.7686	0.000039	0.8623	0.000001

3.4. tabula: *Akv* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - klasifikācijas koki. Atkārtojumu skaits - 1000.

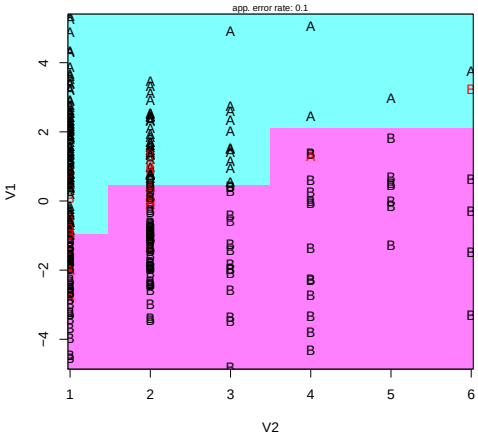
	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.7934	0.013262	0.7036	0.018508	0.7485	0.003689
N =50	0.8059	0.006382	0.7482	0.006067	0.7770	0.001198
N =100	0.8319	0.003189	0.7986	0.003093	0.8152	0.000641
N =200	0.8626	0.001569	0.8099	0.001627	0.8362	0.000320
N =500	0.8762	0.001037	0.8324	0.001009	0.8543	0.000152
N =1000	0.8844	0.000586	0.8357	0.000773	0.8601	0.000114
N =100000	0.8960	0.000022	0.8614	0.000023	0.8788	0.000001

3.5. tabula: Ael piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - klasifikācijas koki. Atkārtojumu skaits - 1000.

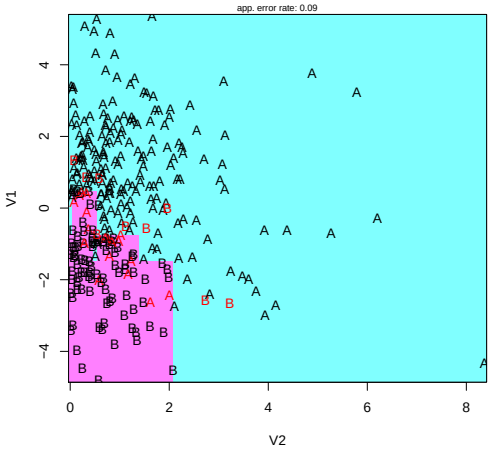
	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =50	0.8296	0.006845	0.8824	0.005121	0.8518	0.001596
N =100	0.8792	0.002831	0.8891	0.002026	0.8834	0.000624
N =200	0.8921	0.001217	0.9034	0.001419	0.8968	0.000224
N =500	0.9002	0.000473	0.9116	0.000719	0.9049	0.000035
N =1000	0.8964	0.000487	0.9177	0.000576	0.9052	0.000030
N =100000	0.9038	0.000005	0.9201	0.000023	0.9106	0.000001



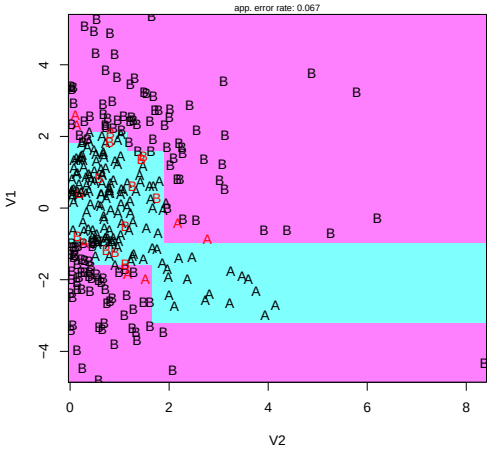
(a) *Adiskr*



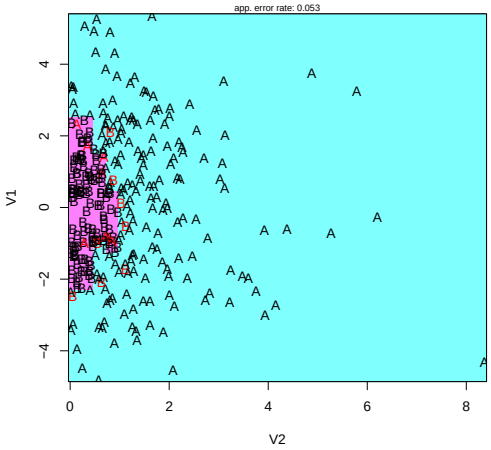
(b) *Adiskr*



(c) *Alin*

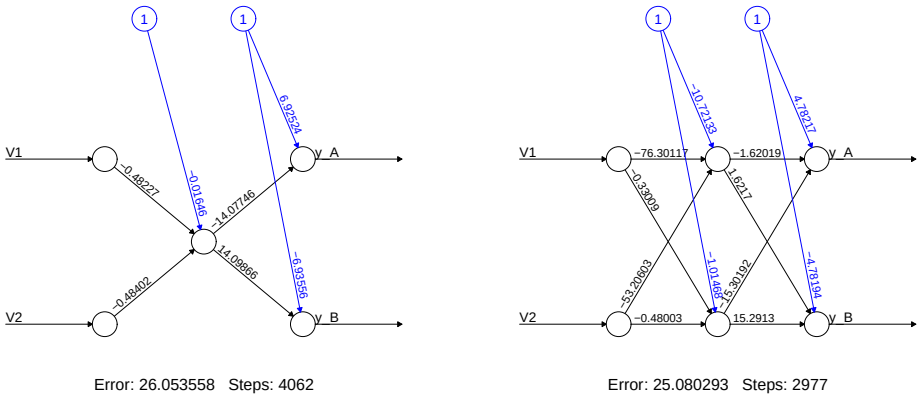


(d) *Akv*



(e) *Ael*

3.4. att.: Izlašu dalījums uz grupām, pielietojot klasifikācijas koku metodi.



(a) Neirona tīkls ar 1 neironu starpslānī (b) Neirona tīkls ar 2 neironiem starpslānī

3.5. att.: Vienslāņa neirona tīkls *Alin* izlases datiem.

3.3. Neironu tīkli

Mākslīgais neirons imitē bioloģiskā neirona īpašības. Neirons ir smadzeņu struktūras elements, kas nodrošina informācijas uztveršanu un nodošanu elektroķīmiskā impulsa veidā. Neirons sastāv no divu tipu atzariem: aksona un dendritiem. Neirons uztver signālus no citiem neironiem pa dendritiem un nodod signālus pa aksonu. Sinaps ir funkcionālais mezgls starp diviem neironiem, tas mainās atkarībā no procesu aktivitātes, kuros tas piedalās. Sinapsa apmācības ideja ir neironu tīklu uzbūves pamatā.

Neironu tīkli ir ļoti spēcīga modelēšanas metode, kura ļauj atspoguļot komplicētas sakarības. Pēc savas būtības tie ir nelineāras funkcijas un ļauj modelēt sakarības ar ļoti lielu mainīgo skaitu. Neirona ieejā tiek padots signālu kopums $X_1, X_2, ..., X_p$, katrs no tiem ir kāda cita neirona izejas signāls vai sākotnējie dati. Katrs signāls tiek pareizināts ar sinaptiskajam spēkam atbilstošu svaru $w_i, i = \overline{1,p}$. Ieejas signālu svērtā summa tiek pakļauta nelineāras aktivizācijas funkcijas iedarbībai, rezultātā veidojas izejas signāls. Vienslāņa neirona tīkla piemērs ar vienu un diviem neironiem vidējā slānī ir atspoguļots 3.5. attēlā. Piemērs tika konstruēts, izmantojot izlases *Alin* ar apjomu $N = 300$ datus.

Neironu tīklu darbības ideja ir radusies no slavenās Kolmogorova teorēmas.

Teorēma 10. *(Andrej Kolmogorovs, 1957. g. [18])*
$\forall p \geq 2$, $p \in Z_+$ *eksistē tādas intervālā* $I = [0;1]$ *definētas nepārtrauktas reāla argumenta funkcijas* φ_{qr} $(r = 1, ..., p,\ q = 1, ..., 2p)$, *ka katrai hiperkubā* $I = [0;1]^p$ *definētai nepār-*

trauktai reālo argumentu funkcijai $f(x_1, x_2, ..., x_p)$ *eksistē tādas intervālā* $[0; 1]$ *definētas nepārtrauktas reāla argumenta funkcijas* Φ_q $(q = 1, ..., 2p)$, *ka*

$$f(x_1, x_2, ..., x_p) = \sum_{q=0}^{2p} \Phi_q(\sum_{r=1}^{p} \varphi_{qr}(x_r)).$$

Pierādījums. Teorēmas konstruktīvs pierādījums var tikt atrasts [18].

Ir skaidrs, ka ar transformāciju palīdzību jebkuru apgabalu var saspiest uz $I = [0; 1]^p$ hiperkubu, tāpēc teorēmas apgalvojumu var pārformulēt un teikt, ka jebkuru daudzdimensiju nepārtraukto funkciju var izteikt kā viendimensiju nepārtraukto funkciju summu. Galvenais uzdevums ir pareizi piemeklēt atbilstošās viendimensiju funkcijas, kas vispārīgi runājot, nav viegli. Ir vēl daži iemesli, kāpēc Kolmogorova teorēmai ir tikai teorētiska nozīme - tajā tiek piedāvāts iepriekš uzdot funkciju f, bet neironu tīklu gadījumā tieši šī funkcija ir izpētes objekts, turklāt, nezināmo koeficientu skaits tiek iepriekš nofiksēts, bet funkcijas paliek nezināmas. Uz doto brīdi eksistē pierādījums, ka teorēma nav spēkā, gadījumā, ja prasīts tiktu funkciju φ un ϕ gludums. Bet bez šī nosacījuma tīkla apmācība ir gandrīz nerealizējams uzdevums, jo, lai varētu apmācīt neirona tīklu, ir jāeksistē atvasinājumam šīm funkcijām. Tāpēc neironu tīklu kontekstā funkciju φ vietā parasti izmanto ar svara koeficientu pareizinātas logistiskās $\varphi(x) = \frac{1}{1+e^{-x}}$ jeb tangensa $\varphi(x) = \tan(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ funkcijas, Φ vietā visbiežāk pielieto svērto $\Phi(x) = \text{sgn}(x)$ vai softmax funkciju $\Phi(x_i) = \frac{e^{x_i}}{\sum_i e^{x_i}}$. Praksē grūtības sagādā arī tas, ka, trenējot tīklu, nav iespējams noteikt optimālo priekš noteiktās precizitātes, funkciju skaitu. Tam ir jābūt uzdotam iepriekš. Ripley [2] piedāvā pārformulēt Kolmogorova teorēmu, lai rezultātu varētu izmantot neironu tīklu kontekstā.

Teorēma 11. *Katra nepārtraukta funkcija* $f(x_1, x_2, ..., x_p) : R^p \to R^g$ *kompaktā kopā var tikt vienmērīgi novērtēta ar* $y(x_1, x_2, ..., x_p)$ *kā*

$$y(x_1, x_2, ..., x_p) = \Phi(\sum_{j=1}^{q} \Omega_j \varphi \sum_{\mu=1}^{p} (\omega_{\mu j} x_\mu + \omega_{0j}) + \Omega_0), \tag{3.3.1}$$

kur φ *ir logistiska funkcija, bet* $\Phi \equiv 1$ *ir identitātes funkcija,* $\{\Omega_j, \omega_{\mu j}\}$, $j = \overline{0, q}$, $\mu = \overline{0, p}$ *ir nezināmo parametru, kuri var pieņemt reālās vērtības, kopa.*

Pierādījums. No sākuma, apskatīsim gadījumu, kad $p = q = 1$. Jebkura nepārtraukta funkcija f var būt vienmērīgi novērtēta intervālā $[a; b]$ ar lēcienveida funkciju, kuras solis

ir mazāks par $\varepsilon/2$. Lai konstruētu šādu novērtējumu, $\forall x \in [a{:}b]$ definē vaļējo intervālu $I(x) = (l(x); u(x))$, kur $l(x) = \max\{y < x : |f(y) - f(x)| \geq \varepsilon/4\}$ un $u(x) = \min\{y > x : |f(y) - f(x)| \geq \varepsilon/4\}$. $I(x)$ tiek sadalīta uz $I(x_i)$ galīgo summu. Tālāk sašķirot x_i dilstošā secībā un intervālā $[l_i, u_i) = [u(x_{i-1}), u(x_i))$ pieņem, ka $g = f(x_i)$. Tad $|\Delta g(l_i)| = |f(x_{i-1}) - f(x_i)| \leq |f(x_{i-1}) - f(l_i)| + |f(x_i) - f(l_i)| \leq \frac{2\varepsilon}{4} = \frac{\varepsilon}{2}$. Ir acīmredzami, ka ar logistisko funkciju summa var pēc patikas tuvu novērtēt trepjveida funkciju. Katra funkcija $\prod_{i=1}^{d} \cos(\omega_i x + \psi_i)$ var tikt izteikta kā $\sum_j a_j \cos(\omega_j' x + \psi_j')$. Katrs šīs summas loceklis ir nepārtraukta funkcija, tātad var būt novērtēta ar iepriekš konstruēto trepjveida funkciju. Slavenais Furjē rezultāts saka (skatīt [2]), ka katra nepārtraukta funkcija $f : R^d \to R$ var tikt novērtēta ar trigonometrisko polinomu palīdzību. Nofiksē kompaktu kopu K un $\varepsilon > 0$. Katra f komponente f_i ir nepārtraukta, tātad var piemeklēt funkciju g_i, lai $\sup\limits_{x \in K} |f_i(x) - g_i(x)| < \varepsilon/\sqrt{d}$ un $\sup\limits_{x \in K} ||f(x) - g(x)|| < \varepsilon$. □

Lai pielietotu neirona tīklus klasificēšanas problemātikai, no datiem ir jānovērtē g funkcijas y_A, $A = 1, ..., g$, viena priekš katras grupas. Funkcijas vērtība ir vienāda ar 1, ja ieraksts pieder klasei A un ar 0 pratējā gadījumā. Lai funkciju vērtības varētu interpretēt, kā piederības varbūtību pie konkrētas klases, tās tiek pakļautas softmax funkcijas iedarbībai.

Definīcija 16. [19] Ar neirona tīkla iegūtais īstās grupas G novērtējums $\widehat{G}$ ir vienāds ar

$$\widehat{G}(X) = \underset{A=1,...,g}{\operatorname{argmax}}\, y_A, \tag{3.3.2}$$

kur

$$y_A(x_1, x_2, ..., x_p) = \Phi(\sum_{\Delta=1}^{q} \Omega_{A\Delta} \varphi \sum_{\mu=1}^{p} (\omega_{\mu\Delta} x_\mu + \omega_{0\Delta}) + \omega_{A0}),$$

$\varphi(x) = \frac{1}{1+e^{-x}}$, $\Phi(x_i) = \frac{e^{x_i}}{\sum_i e^{x_i}}$, $\{\Omega_{A\Delta}, \omega_{\mu\Delta}\}$, $\Delta = \overline{0, q}$, $\mu = \overline{0, p}$ ir nezināmo parametru kopa. Nezināmie parametri pieņem reālās vērtības un tiek saukti par svariem.

Atgriezīsimies pie 3.5. attēla. Izlasē *Alin* ir ieraksti no klasēm A un B, tātad tīkla mērķis ir atrast grupu indikatorfunkciju novērtējumus y_A un y_B. Pazīmju kopa sastāv no divām pazīmēm $\{V1, V2\}$. Nezināmo parametru kopa pirmajā gadījumā ir

$$\Theta = \{\Omega_{A0}, \Omega_{B0}, \Omega_{A1}, \Omega_{B1}, \omega_{01}, \omega_{11}, \omega_{21}\},$$

kura tiek novērtēta ar

$$\Theta_0 = \{6.9, -6.9, -14.1, 14.1, -0.0, -0,5, -0.5\}.$$

Otrajā gadījumā, kad neironu skaits starpslānī tiek palielināts līdz diviem neironiem, palielinās arī nezināmo parametru kopa

$$\Theta = \{\Omega_{A0}, \Omega_{B0}, \Omega_{A1}, \Omega_{B1}, \Omega_{A2}, \Omega_{B2}, \omega_{01}, \omega_{02}, \omega_{11}, \omega_{12}, \omega_{21}, \omega_{22}\},$$

kura tiek novērtēta

$$\Theta_0 = \{4.8, -4.8, -1.6, 1.6, -15.3, -15.3, -10.7, -1.0, -76.3, -0.3, -53.2, -0.5\}.$$

Lai apmācītu neirona tīklu ir jādefinē kļūdas funkciju. Izmanto vai nu atlikumu kvadrātu summu, vai nu krosentropiju.

Definīcija 17. [19] Par krosentropiju sauc

$$R(\Theta) = -\sum_{i=1}^{N}\sum_{A=1}^{g} f_A(X_i) \ln y_A(X_i), \tag{3.3.3}$$

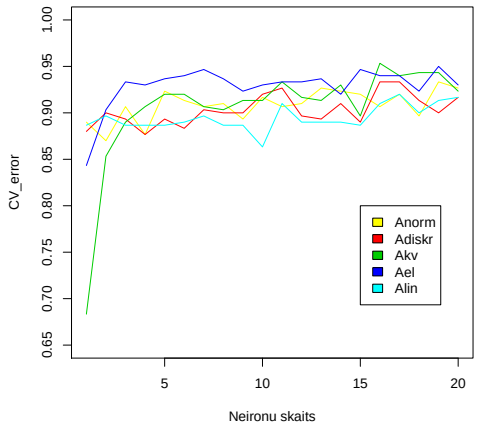
kur $\Theta = \{\Omega_{A\Delta}, \omega_{\mu\Delta}\}$, $\Delta = \overline{0,q}$, $\mu = \overline{0,p}$.
Par atlikumu kvadrātu summu sauc

$$R(\Theta) = \sum_{i=1}^{N}\sum_{A=1}^{g}(f_A(X_i) - y_A(X_i))^2. \tag{3.3.4}$$

Neirona tīkla apmācības būtība ir parametru kopas Θ novērtēšana, minimizējot kļūdu $R(\Theta)$. Eksistē ļoti daudz paņēmienu, kā minimizē kļūdu, piemēram, gradienta metode. Piešķirot parametru kopai Θ kaut kādu sākuma vērtību Θ_0, aprēķina $R(\Theta_0)$, tālāk aprēķina $grad(R)_{\Theta_0}$. Tā kā funkcija dilst gradientam pretējā virzienā, būtu labi nākamajā iterācijā pāriet tieši šajā virzienā, ņemot $\Theta_1 = \alpha \cdot grad(R)_{\Theta_0}$, kur α ir pietiekami mazs, lai "nepaskrietu" minimumam garām, bet ne tik mazs, lai līdz minimumam būtu jāveic pārāk liels soļu skaits. Punkts $grad(R)_{\Theta_0} = 0$ ir punkts, kurā funkcija varētu sasniegt savu minimumu. Ja $R(\Theta)$ vērtība joprojām neatbilst precizitātes nosacījumiem, procedūra jāsāk no jauna. Metode pēc būtības ir daudzargumentu funkcijas minimuma meklēšana, kas vispārīgi runājot, ir darbietilpīgs process. Šajā darbā parametru kopas novērtēšana notiks ar kļūdas atpakaļizplatīšanas (angliski backpropagation of error) metodes palīdzību, kura sīkāk ir aprakstīta [20].

Lai sāktu trenēt neirona tīklu ir jādefinē neironu skaits starpslāni. Ja neironu skaits būs pārāk liels, tad tīklam būs laba iespēja pielāgoties datiem un tuvoties nulles kļūdai, taču tāds tīkls nedos labus rezultātus uz pārbaudes datiem. Lai noteiktu optimālo neironu

skaitu, atkal pielieto krosvalidāciju. Parasti sadala datus 10 daļās, 9 no kurām izmanto tīkla apmācībai un 1 pārbaudei, atkārto procedūru 10 reizes un izskaitļo vidējo kopējo kļūdas līmeni. Izvēlās tādu neironu skaitu, kurš minimizē krosvalidācijas kļūdu. Krosvalidācijas kļūdas atkarība no neironu skaita starpslānī izlasēm *Anorm*, *Adiskr*, *Alin*, *Akv* un *Ael* ir parādīta 3.6. attēlā. Sākuma posmā neironu skaita palielināšana tiešām dod manāmu precizitātes uzlabošanu, taču pēc tam kļūdas līmenis stabilizējas un turpmākai neironu skaita palielināšanai nav jēgas.



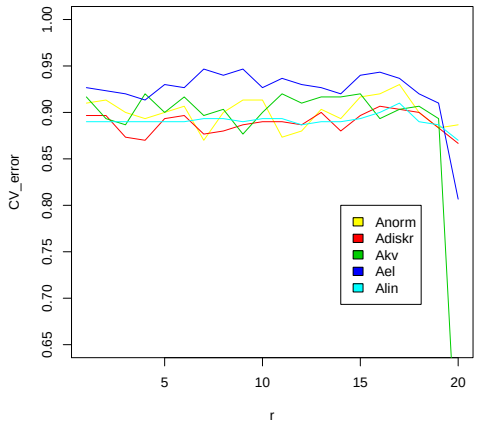
3.6. att.: Kopējās precizitātes atkārība no neironu skaita starslānī.

Ir pierādīts, ka uzlabot neironu tīklu darbību palīdz svara koeficientu samazināšanas parametra ieviešana. Tas padara funkcijas novērtējumu ar neironu tīklu palīdzību gludāku un līdz ar to ne tik ļoti pielāgotu datiem. Ar šī koeficienta palīdzību koriģē kļūdas funkcijas $R(\Theta)$ (3.3.3) vai (3.3.4) vērtību.

Definīcija 18. Par koriģēto kļūdas funkciju sauc

$$\tilde{R}(\Theta) = R(\Theta) + \lambda \frac{1}{2} \sum_i \omega_i,$$

kur ω_i ir svara koeficienti pa visu tīklu un λ ir svara koeficientu samazināšanas parametrs. 3.7. attēlā ir parādīta visu 5 izlašu krosvalidācijas kļūdas atkarība no parametra λ. Tīkla starpslānis satur 5 neironus. Krosvalidācijas kļūda ir atkarīgā no parametra λ. Tika novērots rezultāta uzlabojums, izmantojot $\lambda \neq 0$. Turpmākais parametra palielinājums nedod būtisku rezultāta uzlabojumu, kļūda sāk svārstīties un kopējā precizitāte strauji dilst pie parametra vērtības, kas vienāda ar 1. Dotajā gadījumā jebkāda $\lambda \in (0; 1)$ dod līdzīgus rezultātus. Darbā visi turpmākie aprēķini tiks veikti pie $\lambda = 1 \cdot 10^{-6}$.



3.7. att.: Kopējās kļūdas atkarība no λ, kur $\lambda = 1 \cdot 10^{r-20}$.

3.6. tabula: *Anorm* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - neironu tīkli. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.8358	0.005074	0.8210	0.006455	0.8284	0.001140
N =50	0.8426	0.004539	0.8315	0.004454	0.8370	0.000648
N =100	0.8509	0.003315	0.8622	0.004255	0.8565	0.000320
N =200	0.8698	0.001340	0.8641	0.001395	0.8669	0.000100
N =500	0.8758	0.000395	0.8765	0.000414	0.8761	0.000018
N =1000	0.8760	0.000254	0.8787	0.000263	0.8774	0.000013
N =100000	0.8798	0.000002	0.8803	0.000002	0.8801	0.000000

3.4. Neironu tīklu ilustrācija

Neironu tīklu pielietošanas ilustrācijai, tāpat kā iepriekšējā nodaļā, izmantosim *Anorm*, *Adiskr*, *Alin*, *Akv* un *Ael* piemērus. Neironu tīkli ir neparametriskā klasifikācijas metode, tāpēc nav nepieciešams pārbaudīt pieņēmumus. Lai pielāgotu neironu tīklus klasifikācijas uzdevumam, tiek definētas g funkcijas $f_A = I(G)$, $A = \overline{1, g}$. Respektīvi, funkcijas vērtība ir vienāda ar 1, ja ieraksts pieder klasei A un ir nulle pretējā gadījumā. Katru reizi svaru samazināšanas koeficients un neironu skaits tiks izvēlēti, minimizējot krosvalidācijas kļūdu. Pie nebūtiska precizitātes uzlabojuma, priekšroka tiks dota metodei ar mazāko parametru skaitu.

Neironu tīklu pielietošana izlasēm ar apjomu $N = 300$ no piemēriem *Anorm*, *Adiskr*,

3.7. tabula: *Adiskr* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - neironu tīkli. Atkārtojumu skaits - 1000.

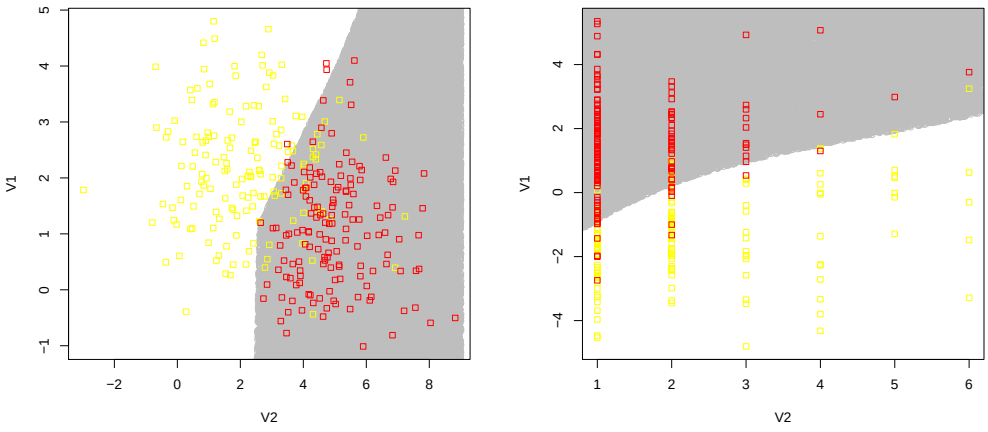
	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.8256	0.005127	0.8297	0.004146	0.8275	0.001132
N =50	0.8454	0.002117	0.8445	0.002622	0.8450	0.000333
N =100	0.8583	0.001652	0.8554	0.001666	0.8569	0.000092
N =200	0.8594	0.001270	0.8601	0.001184	0.8598	0.000071
N =500	0.8652	0.000575	0.8613	0.000515	0.8633	0.000033
N =1000	0.8671	0.000228	0.8662	0.000278	0.8667	0.000012
N =100000	0.8875	0.000026	0.8499	0.000011	0.8701	0.000009

3.8. tabula: *Alin* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - neironu tīkli. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.7971	0.006832	0.8268	0.007342	0.8074	0.002409
N =50	0.8215	0.002832	0.8391	0.002567	0.8276	0.000751
N =100	0.8228	0.002558	0.8650	0.002324	0.8374	0.000508
N =200	0.8406	0.001284	0.8632	0.001337	0.8484	0.000206
N =500	0.8438	0.000648	0.8785	0.000581	0.8555	0.000105
N =1000	0.8510	0.000354	0.8779	0.000387	0.8599	0.000053
N =100000	0.9116	0.000008	0.7981	0.000066	0.8720	0.000004

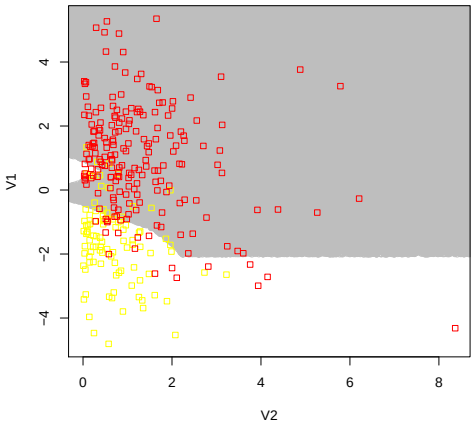
3.9. tabula: *Akv* piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - neironu tīkli. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =30	0.8397	0.007220	0.8338	0.004751	0.8368	0.002266
N =50	0.8573	0.005637	0.8512	0.002166	0.8543	0.001130
N =100	0.9033	0.001448	0.8576	0.000934	0.8805	0.000218
N =200	0.9115	0.000812	0.8670	0.000501	0.8892	0.000098
N =500	0.9205	0.000229	0.8742	0.000200	0.8974	0.000016
N =1000	0.9240	0.000117	0.8742	0.000107	0.8992	0.000012
N =100000	0.9272	0.000009	0.8745	0.000038	0.9010	0.000006

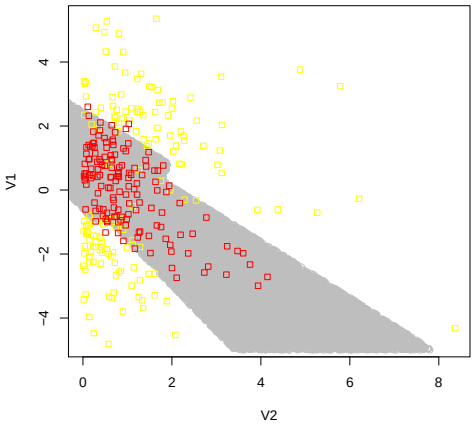


(a) *Adiskr*

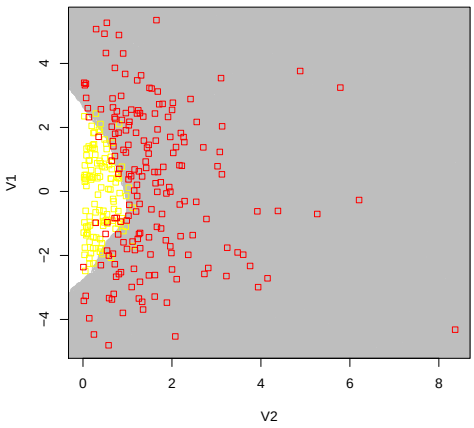
(b) *Adiskr*



(c) *Alin*



(d) *Akv*



(e) *Ael*

3.8. att.: Izlašu dalījums uz grupām, pielietojot neironu tīklus.

3.10. tabula: Ael piemēra grupu un kopējās precizitātes novērtējumu vidējā vērtība un empīriskā dispersija pie dažādiem izlašu apjomiem. Metode - neironu tīkli. Atkārtojumu skaits - 1000.

	$\overline{A_A}$	$S(A_A)$	$\overline{A_B}$	$S(A_B)$	$\overline{OA}$	$S(OA)$
N =50	0.8600	0.004083	0.8986	0.002433	0.8762	0.001089
N =100	0.8885	0.001353	0.9071	0.001128	0.8964	0.000225
N =200	0.9041	0.000440	0.9192	0.000633	0.9105	0.000062
N =500	0.9118	0.000164	0.9270	0.000238	0.9182	0.000018
N =1000	0.9120	0.000072	0.9339	0.000083	0.9210	0.000011
N =100000	0.9242	0.000016	0.9191	0.000013	0.9221	0.000013

Alin, *Akv* un *Ael* ir atrodama 3.8. attēlā, kurā ir attēloti gan diskriminācijas robežas, gan dati un to grupas.3.6., 3.7., 3.8., 3.9. un 3.10. tabulās ir atspoguļotas grupu un kopējās precizitātes novērtējumu vidējās vērtības un empīriskās dispersijas izmaiņas atkarībā no izlases apjoma.

Kopumā spriežot, neirona tīklu klasifikatori tiešām dod pārsteidzoši labus rezultātus. Ja pareizi izvēlēties neironu skaitu un regulēšanas parametru, tie izveido tuvas lineārām diskriminānciju robežas, gadījumos, kad dalījums grupās ir lineārs, tajā pašā laikā labi atpazīst nelineārās sakarības. Tas nozīmē, ka nav jāuztraucas par pārlieku pielāgošanos datiem. Šī metode dod labāku rezultātu par lineāro diskriminantu analīzi, pat *Anorm* izlasei, kura tika izveidota speciāli, lai izpildītos visi lineārās diskriminantu analīzes pieņēmumi. Izlasei *Ael* neironu tīkli dod labāku rezultātu par visām iepriekšējām metodēm. Runājot par pārejām izlasēm, rezultāti nav sliktāki par augstāk aplūkotajām metodēm. Vienīgais neironu tīklu trūkums ir tas, ka metode pēc būtības ir kā melnā kaste, tai nav acīmredzamas interpretācijas, parametru skaitam un svaru samazināšanas koeficientam ir jābūt uzdotiem iepriekš. Turklāt svaru novērtēšana notiek diezgan lēni, algoritms ir laikietilpīgs. Visu šo iemeslu dēļ, metode nav ieguvusi plašu pielietojumu praksē. Taču gadījumos, kad ir jāmodelē sarežģītas sakarības ar lielu parametru skaitu, grūti atrast metodi, kura darbotos tikpat efektīvi, kā neironu tīkli.

4. Kopējās precizitātes novērtēšana

Kā tika novērots augstāk apskatītajos piemēros, bieži vien dažādas diskriminācijas metodes (izņemot lineāro diskriminantu analīzi) piedāvā līdzīgas klasificēšanas robežas un līdz ar to dod līdzīgu precizitāti. Šī iemesla dēļ praksē bieži vien grūtības sagādā nevis metodes izvēle, bet sagaidāmā kļūdas līmeņa novērtēšana. Šajā nodaļā tiks apskatītas universālās novērtēšanas metodes, kuras nebalstās uz pieņēmumiem par grupu sadalījuma blīvuma funkcijām un var tikt pielietotas gan statistiskajiem, gan datu izraces modeļiem. Metodes tiks ilustrētas ar *Akv* piemēra palīdzību, pielietojot tās lineārai un kodolu diskriminantu analīzei, klasifikācijas kokiem un neironu tīkliem. Šis piemērs tika izvēlēts tāpēc, ka tas izrādījās izaicinošs lineārai diskriminantu analīzei un deva vislielākās precizitātes atšķirības dažādām metodēm.

Definīcija 19. Par grupas k kopējo precizitāti sauc varbūtību, ka klasifikatora novērtēta grupa sakritīs ar īsto grupu, t.i.

$$TA_k = P(G = k|\widehat{G}(X) = k),$$

kur G ir X īstā grupa, bet $\widehat{G}(X)$ ir tās novērtējums.

Definīcija 20. Par klasifikatora kopējo precizitāti sauc

$$TOA = \sum_{k=1}^{g} \pi_k \cdot TA_k,$$

kur π_k ir piederības klasei k varbūtība.

Grupas precizitātes un kopējās precizitātes novērtējumu formulas ir dotas (2.2.6) un (2.2.5). Visvienkāršākais un praksē visbiežāk pielietojams precizitātes novērtēšanas veids ir novērtēt kopējo precizitāti no tiem pašiem datiem, no kuriem tika būvēts modelis. Skaidrs, ka šī metode nav pārāk precīza un dod ļoti optimistisku novērtējumu, īpaši datu izraces metodēm, ja nav ieviesti modeļa samazināšanas parametri un klasifikators ir

pielāgots konkrētai datu kopai. Lai saprastu, vai dažādu paņēmienu precizitātes novērtējumi tuvojas īstajai precizitātei, klasifikatori tika būvēti, izmantojot *Akv* izlasi ar apjomu $N = 300$, bet precizitāte tika novērtēta, ģenerējot datus pēc *Akv* likuma, bet ar ļoti lielu apjomu - 100000. Tāds kļūdas novērtējums var tikt uzskatīts par ļoti tuvu īstās precizitātes novērtējumu. 4.1. tabulā ir parādīta precizitāte, kas novērtēta no modelēšanas datiem, kura, kā var redzēt no tabulas 4.2., ir pārāk optimistiska salīdzinājumā ar "īsto" precizitāti. Jāpiebilst, ka starpība starp modelēšanas un pārbaudes precizitāti nav pārāk ievērojama statistiskiem klasifikatoriem, taču datu izraces metodēm modelēšanas kļūda tiešām dod pārāk optimistiskus rezultātus.

4.1. tabula: *Akv* izlases ar apjomu $N = 300$ grupu un kopējās precizitātes novērtējums no modelēšanas datiem.

	LDA	CRT	NNET	KDA
A_A	0.6014	0.9527	0.9324	0.9527
A_B	0.4803	0.9145	0.8618	0.8421
OA	0.5400	0.9333	0.8967	0.8967

4.2. tabula: *Akv* izlases ar apjomu $N = 300$ grupu un kopējās precizitātes "īstā" vērtība.

	LDA	CRT	NNET	KDA
A_A	0.6224	0.8782	0.9273	0.9419
A_B	0.4728	0.8193	0.8677	0.8420
OA	0.5477	0.8488	0.8976	0.8920

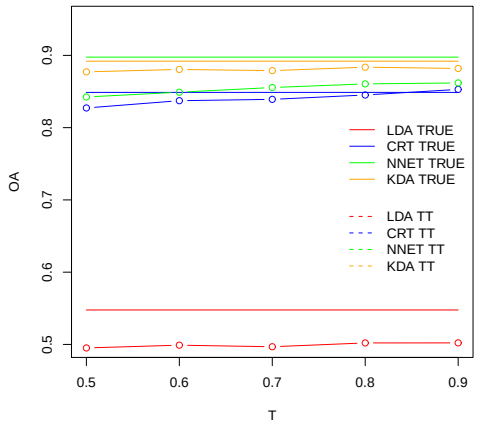
Cits, arī ļoti vienkāršs paņēmiens, ir sadalīt datus modelēšanas un pārbaudes daļās. Dažādos avotos tiek rekomendētas dažādas dalīšanas proporcijas [21]. Sākot no 0.5 datu modelēšanai un 0.5 pārbaudei līdz 0.9 modelēšanai un 0.1 pārbaudei. Šī metode ir izmantojama tikai gadījumos, kad ir pietiekoši daudz datu. Nepilna pieejamas datu kopas izmantošana ir šīs metodes trūkums.

Definīcija 21. Par kopējās precizitātes novērtējumu, sadalot datu kopu modelēšanas un pārbaudes daļās, sauc

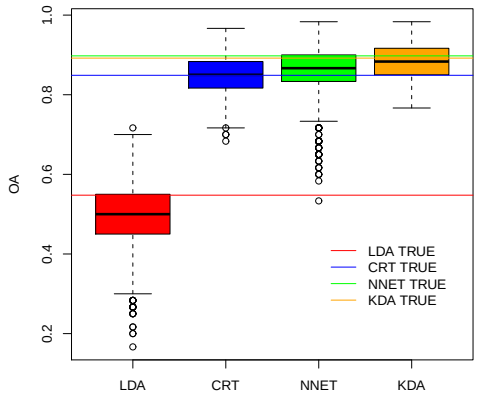
$$TT = OA(\widehat{G}^{TRAIN}(X^{TEST})),$$

kur $\widehat{G}^{TRAIN}$ ir klasifikators, kurš ir izveidots, izmantojot modelēšanas datu daļu, bet

$\widehat{G}^{TRAIN}(X^{TEST})$ ir klasificēšanas rezultāts, pielietojot šo klasifikatoru pārbaudes datu daļai.

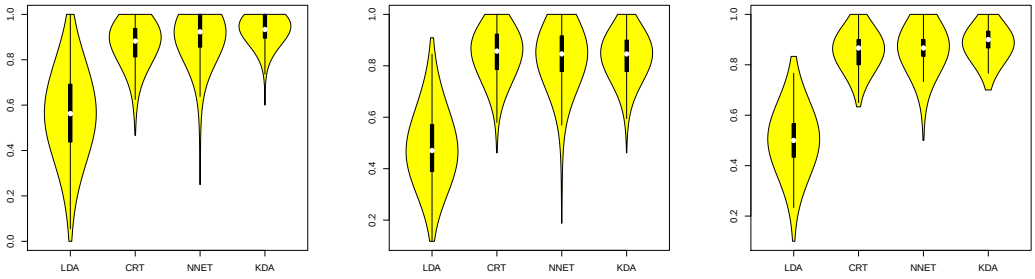


4.1. att.: Modelēšanas un pārbaudes precizitātes novērtējumu atkarība no datu dalīšanas proporcijas un "īstā" precizitāte.



4.2. att.: Modelēšanas un pārbaudes precizitātes novērtējumu kastes diagrammas pie $T = 0.20$.

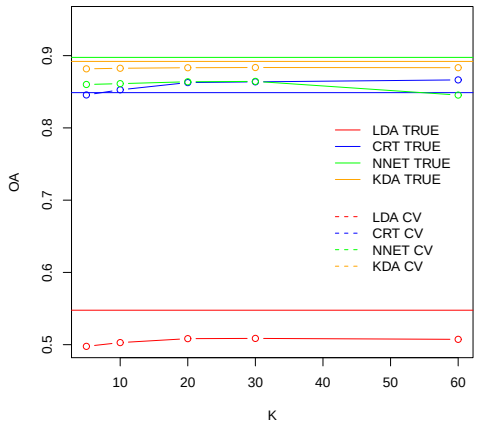
Piemērs 2. Darba gaitā izlasei *Akv* ar apjomu $N = 300$ tika aprēķināta modelēšanas un pārbaudes precizitāte, sadalot datus dažādās proporcijās 1000 reizes un aprēķinot vidējo precizitāti. Precizitāte tika novērtēta 4 dažādiem klasifikatoriem - lineārai diskriminantu analīzei, kodolu diskriminantu analīzei, klasifikācijas kokiem un neironu tīkliem. Grafiks 4.1. rāda kā mainās modelēšanas un pārbaudes precizitāte, palielinot modelēšanas datu daļu (T), un cik tā ir tuva īstai precizitātei. Ar *LDA TRUE*, *CRT TRUE*,



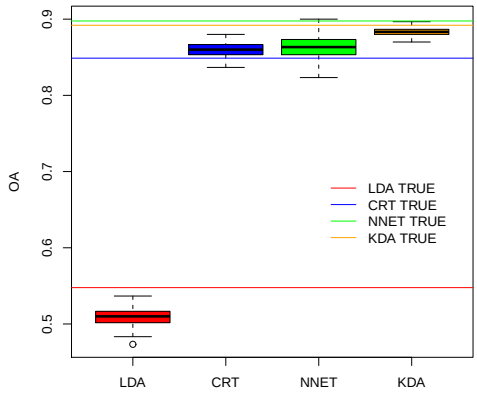
(a) A_A (b) A_B (c) OA

4.3. att.: Vijoles veida grafiki grupu un kopējās precizitātes novērtējumiem pie $T = 0.20$.

$NNET\ TRUE$, $KDA\ TRUE$ ir apzīmēta metožu "īstā" precizitāte dotajām piemēram pie dotā izlases apjoma, bet ar $LDA\ TT$, $CRT\ TT$, $NNET\ TT$, $KDA\ TT$ modelēšanas un pārbaudes precizitāte. Šajā piemērā ir novērots, ka modelēšanas un pārbaudes precizitāte pieaug, pieaugot modelēšanas daļai, nepārsniedzot īsto precizitāti (izņemot klasifikācijas koku gadījumu). Pie tam pieaugums datu izraces klasifikatoriem notiek straujāk nekā statistiskajiem klasifikatoriem. Daļēji to var izskaidrot tā, ka klasifikators strādā labāk, ja izmanto vairāk informācijas no datiem. Lineārās diskriminantu analīzes gadījumā dalīšanas proporcija ne pārāk stipri ietekmē precizitātes novērtējumu, bet kodolu diskriminantu analīzes un, īpaši izteikti, klasifikācijas koku un neironu tīklu gadījumā kopējās precizitātes novērtējuma vidēja vērtība aug, ja aug datu modelēšanas daļa. Aug arī novērtējuma dispersija, ko var redzēt. Tas varētu būt izskaidrojams sekojošā veidā - parametriskā metode mazāk pielāgojas datiem, tāpēc nav tik ļoti būtiski, cik daudz datu tiek izmantots trenēšanai un cik daudz pārbaudei. Jo vairāk informācijas no datiem tiek izmantots, konstruējot neparametrisko modeli, jo precīzāks ir modelis. Vadoties no rezultātiem, iegūtiem, analizējot modelēšanas un pārbaudes kopējās precizitātes novērtējumu, var teikt, ka kodolu diskriminantu analīze izlasei Akv nodrošina vislielāko klasificēšanas precizitāti. 4.1. attēlā ir parādītas klasifikatoru modelēšanas un pārbaudes precizitātes kastes diagrammas kopā ar īsto precizitāti gadījumā, kad pārbaudes daļa sastāda 20% no izlases apjoma, atkārtojumu skaits 1000. Grupu un kopējās precizitātes novērtējumu sadalījumi ir attēloti 4.3. ar vijoles veida grafiku palīdzību. Vijoles veida grafiks ir kastes grafiks, kuram sānos ir pievienots sadalījuma blīvuma funkcijas novērtējums ar kodolu gludināšanu.



4.4. att.: Krosvalidācijas precizitātes novērtējumu atkarība no krosvalidācijas daļu skaita un īstā precizitāte.



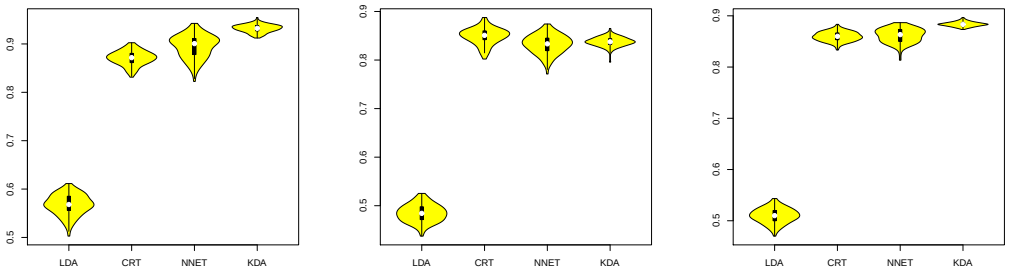
4.5. att.: Krosvalidācijas precizitātes novērtējumu kastes diagrammas pie $K = 20$.

Nākamā metode ir krosvalidācijas pielietošana kopējās precizitātes novērtēšanai. Tā ir īstenojama sekojošā veidā: datu bāzi sadala K vienādas daļās, konstruē klasifikatoru, izmantojot $K - 1$ daļas, bet kļūdu novērtē no 1 palikušās daļas. Klasifikatora būvēšanas procedūru atkārto K reizes (katru reizi izmetot citu datu daļu) tad aprēķina vidējo precizitāti, kura arī ir K-daļu krosvalidācijas kopējās precizitātes novērtējums.

Definīcija 22. [23] Par K-daļu krosvalidācijas precizitātes novērtējumu sauc

$$CV = \frac{1}{K} \sum_{k=1}^{K} OA(\widehat{G}^{-k}(X^k)),$$

kur $\widehat{G}^{-k}$ ir klasifikators, kurš ir izveidots, izmetot no datiem k-to daļu, bet $\widehat{G}^{-k}(X^k)$ ir



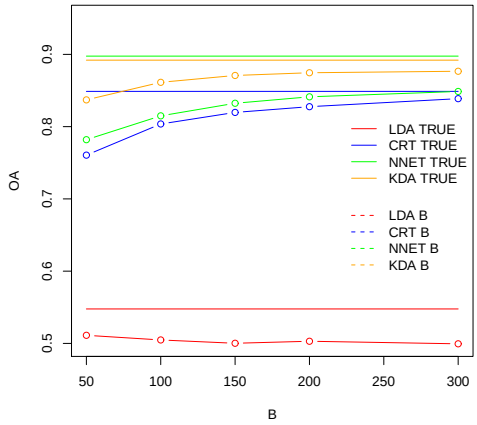
(a) A_A (b) A_B (c) OA

4.6. att.: Vijoles veida grafiki grupu un kopējās precizitātes krosvalidācijas novērtējumiem pie $K = 20$.

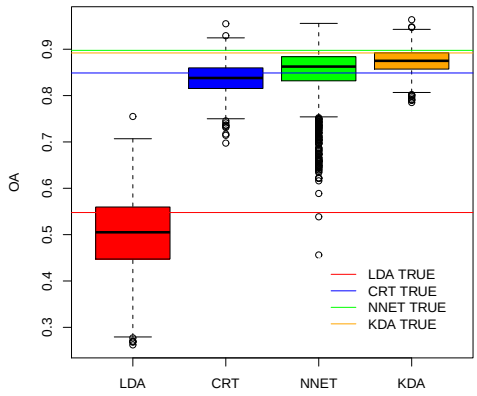
klasificēšanas rezultāts, pielietojot šo klasifikatoru k-tai datu daļai.

Piemērs 3. Novērtēsim kopējo precizitāti, kura tika iegūta, pielietojot lineāro diskriminantu analīzi, kodolu diskriminantu analīzi, klasifikācijas kokus un neironu tīklus izlasei *Akv* ar apjomu $N = 300$. Mainot K vērtības, tiek novērots (attēls 4.4.), ka kopējās precizitātes novērtējuma svārstības sāk samazināties pie $K = 20$. Ar $LDA\ TRUE$, $CRT\ TRUE$, $NNET\ TRUE$, $KDA\ TRUE$ ir apzīmēta metožu "īstā" precizitāte dotajām piemēram pie dotā izlases apjoma, bet ar $LDA\ CV$, $CRT\ CV$, $NNET\ CV$, $KDA\ CV$ modelēšanas un pārbaudes precizitāte. Neparametriskiem klasifikatoriem tiek novērots tāds pats efekts kā augstāk aplūkotajai precizitātes novērtēšanas metodei. Ja palielināt K, tad precizitātes novērtējums aug, jo katru reizi modeļa veidošanai tiek izmantots lielāks datu skaits. Par krosvalidācijas metodes trūkumu var uzskatīt to, ka precizitāte tiek novērtēta ļoti daudz reizes un process prasa daudz laika. Taču, kā novērots piemēros, šī metode ļauj ievērojami samazināt novērtējuma dispersiju. Kastes diagrammas 4.5. attēlā parāda, ka novērtējuma izkliede ir daudz mazāka nekā modelēšanas un pārbaudes precizitātes novērtēšanas metodei. Grupu un kopējās precizitātes novērtējumu sadalījumi pie atkārtojumu skaita vienāda ar 1000 ir attēloti ar vijoles veida grafikiem 4.6. attēlā. Krosvalidācijas novērtējums liecina, ka izlasei *Akv* visefektīvākās metodes ir kodolu diskriminantu analīze un neironu tīkli. Abi klasifikatori dod ļoti līdzīgu krosvalidācijas kopējās precizitātes novērtējumu. Jāatzīmē, ka pieaugot daļu skaitam, pieaug arī precizitātes novērtējums. Tas ir spēkā visiem klasifikatoriem, izņemot neironu tīklus, kuriem novērtējums sasniedz savu maksimumu pie $K = 30$. Klasifikāciju koku precizitātes

novērtējums ir pārāk optimistisks un ir lielāks par īsto precizitāti visiem $K > 5$. Pārējo klasifikatoru precizitātes novērtējums ir mazāks par īsto precizitāti visiem K.



4.7. att.: Butstrapa precizitātes novērtējumu atkarība no butstrapa izlases apjoma un īstā precizitāte.

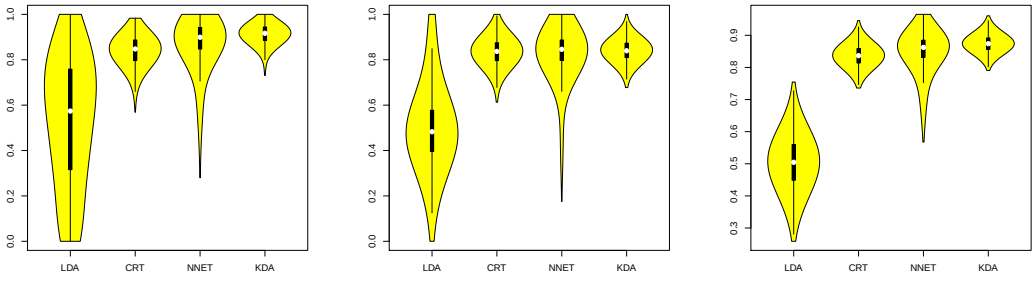


4.8. att.: Butstrapa precizitātes novērtējumu kastes diagrammas pie $B = 300$.

Pēdējā apskatītā metode ir kopējās precizitātes novērtēšana ar butstrapu. No datu kopuma X, kurš satur N novērojumus, tiek uz labu laimi izvēlēta butstrapa (ar atkārtojumiem, izlases apjoms parasti tiek izvēlēts vienāds ar datu kopas apjomu) izlase X^B, kuru izmantojot tiek būvēts klasifikators. Kopējās precizitātes novērtējums tiek aprēķināts, izmantojot datus no X, kas nebija iekļauti X^B.

Definīcija 23. Par butstrapa kopējās precizitātes novērtējumu sauc

$$B = OA(\widehat{G}^B(X^{-B})),$$



(a) A_A (b) A_B (c) OA

4.9. att.: Vijoles veida grafiki grupu un kopējās precizitātes butstrapa novērtējumiem pie $B = N = 300$.

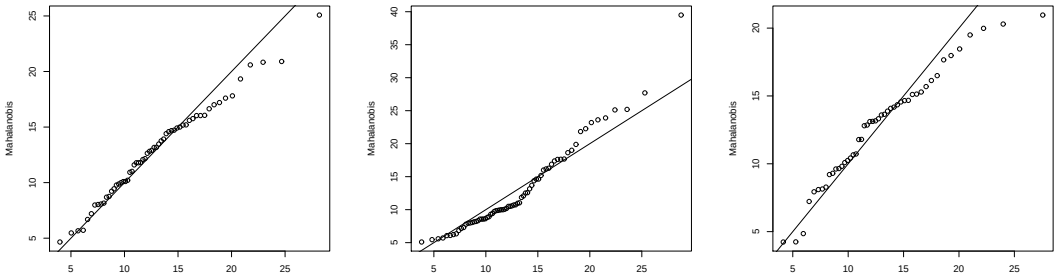
kur $\widehat{G}^B$ ir klasifikators, kurš ir izveidots, izmantojot no butstrapa izlasi, bet $\widehat{G}^B(X^{-B})$ ir klasificēšanas rezultāts, pielietojot šo klasifikatoru datiem, kuri nav iekļauti butstrapa izlasē.

Piemērs 4. Pielietojot butstrapu *Akv* izlasei ar apjomu $N = 300$ dažādiem klasifikatoriem, iegūstam rezultātus, kuri ir parādīti 4.7. attēlā. Ar $LDA\ TRUE$, $CRT\ TRUE$, $NNET\ TRUE$, $KDA\ TRUE$ ir apzīmēta metožu "īstā" precizitāte dotajām piemēram pie dotā izlases apjoma, bet ar $LDA\ B$, $CRT\ B$, $NNET\ B$, $KDA\ B$ modelēšanas un pārbaudes precizitāte. Parasti klasifikācijas uzdevumos ņem butstrapa izlases apjomu vienādu ar sākotnējās izlases apjomu. Lai pārbaudītu tādas pieejas racionalitāti, butstrapa precizitātes novērtējuma vidējā vērtība (atkārtojumu skaits 1000) tika aprēķināta pie izlases apjoma $B = 50, 100, 150, 200, 300$. Empīriskie novērojumi parādīja, ka precizitātes novērtējums aug ar butstrapa izlases apjoma palielināšanu, bet visu laiku ir zemāks par īsto precizitāti. Tāpat kā visas iepriekšējās precizitātes novērtēšanas metodes, butstraps izvēlas kodolu diskriminantu analīzi un neironu tīklus kā visprecīzākos. Butstrapa novērtējuma dispersija būtiski pārsniedz krosvalidācijas novērtējuma dispersiju (attēli 4.9. un 4.8.).

Analizējot precizitātes novērtējumu *Akv* piemērā, var secināt, ka visas metodes izvēlas vienus un tos pašus klasifikatorus. Gan butstrapa, gan modelēšanas un pārbaudes novērtējuma dispersija ir lielākā par krosvalidācijas novērtējuma dispersiju. Krosvalidācijas novērtējums ir arī vistuvākais īstai precizitātei. Piemērs parādīja, ka novērtējuma uzvedība ir atkarīgā ne tikai no novērtēšanas metodes, bet arī no izmantotā klasifikatora.

Visiem, izņemot klasifikācijas kokus, apskatītajiem modeļiem novērtējums bija mazāks par īsto precizitāti. Klasifikācijas koku precizitātes novērtējums, izmantojot pārbaudes datus un krosvalidāciju, bija lielāks par īsto precizitāti. Var pamanīt, ka augot modelēšanā izmantotai datu daļai, precizitātes novērtējums pieaug. Īpaši izteikti raksturīgs tas ir datu izraces metodēm.

5. Pielietojumi



(a) A (b) B (c) C

5.1. att.: Grupu $q - q$ grafiki vīna klasifikācijas uzdevumam.

Klasifikācijas uzdevumi ir sastopami daudzās dzīves sfērās - biznesā, ekonomikā, medicīnā, mārketingā, lēmumu pieņemšanā u.t.t. Šis nodaļas ietvaros tiks aprakstīts ne pārāk tipisks, bet ļoti interesants klasifikācijas uzdevums - vīna kvalitātes noteikšana pēc tā ķīmiskā sastāva. Ir zināms, ka vīna garša, tātad arī cena, ir atkarīga ne tikai no vīna šķirnes, bet arī no ražas gada un reģiona. Parasti, lai noteiktu vīna kategoriju, aicina ekspertus, kuri nosaka to pēc degustācijas. Ekspertu pakalpojumi ir dārgi un viņu viedoklis ne vienmēr ir objektīvs. Tātad, labāk ir uzticēt vīna klasifikāciju kādam algoritmam. Klasifikācijas uzdevumam tiks lietota *Institute of Pharmaceutical and Food Analysis and*

5.1. tabula: Testēšanas un pārbaudes vidējās precizitātes novērtējums, tā empīriskā dispersija un kvantiles vīna klasifikācijas uzdevumam.

	$\overline{OA}$	$S(OA)$	5% kvantile	95% kvantile
LDA	0.9862	0.0003	0.9655	1.0000
CRT	0.9157	0.0025	0.8372	0.9767
NNET	0.8384	0.0063	0.6930	0.9000

5.2. tabula: Butstrapa vidējās precizitātes novērtējums, tā empīriskā dispersija un kvantiles vīna klasifikācijas uzdevumam.

	$\overline{OA}$	$S(OA)$	5% kvantile	95% kvantile
LDA	0.9794	0.0003	0.9524	1.0000
CRT	0.9006	0.0019	0.8209	0.9643
NNET	0.8220	0.0006	0.7045	0.9154

5.3. tabula: Krosvalidācijas vidējās precizitātes novērtējums, tā empīriskā dispersija un kvantiles vīna klasifikācijas uzdevumam.

	$\overline{OA}$	$S(OA)$	5% kvantile	95% kvantile
LDA	0.9874	0.0000	0.9777	0.9944
CRT	0.8964	0.0002	0.8764	0.9215
NNET	0.8340	0.0001	0.8212	0.8460

Technologies datu bāze (skatīt [24]), kura satur 178 novērojumus par vienu, Itālijā ražotu, sarkanā vīna šķirni. Degustācijas laikā vīni tika sadalīti trijās klasēs: A (59 novērojumi), B (71 novērojums) un C (48 novērojumi). Paralēli tika veikta ķīmiskā un fiziskā analīze, kura saturēja 13 parametrus: alkohola saturs ($V2$), ābolu etiķa saturs ($V3$), potašas saturs ($V4$), sārmainība ($V5$), magnija saturs ($V6$), kopējais fenolu saturs ($V7$), flavanoidu saturs ($V8$), neflavanoida izcelsmes fenolu saturs ($V9$), proantocianinu saturs ($V10$), krāsas intensitāte ($V11$), krāsa ($V12$), OD280/OD315 šķīdība ($V13$) un prolina saturs ($V14$).

Uzdevums ir atrast klasifikatoru, kurš ļaus pēc vīna ķīmiskā sastāva noteikt tā kategoriju, paredzēt cilvēku viedokli par konkrētu vīnu un pareizi izvēlēties cenu. Vai tiešām tas, kāds vīns garšo pircējiem visvairāk, ir izskaidrojams ar to ķīmisko sastāvu? Dotai problemātikai tika pielietotas trīs metodes: LDA, CRT un NNET. Kodolu diskriminantu analīze nevar tikt pielietota tik lielam parametru skaitam. LDA pielietošanai ir nepieciešams pārbaudīt hipotēzi par grupu sadalījumu atbilstību daudzdimensiju normālajam sadalījumam un grupu kovariāciju matricu vienādību. Normalitāte tika pārbaudīta, konstruējot $q - q$ grafikus (5.1. attēls) un veicot Box'M testu (p-vērtība $1.35 \cdot 10^{35}$). Var secināt, ka LDA pieņēmumi neizpildās. Neskatoties uz to, LDA metode tiks pielietota vīna klasificēšanas problemātikai, jo iepriekšējās nodaļās empīriski tika parādīts, ka LDA

5.4. tabula: LDA robežu funkciju koeficienti.

Parametrs	$g_1(V)$	$g_2(V)$
$V2$	-0.403399781	0.8717930699
$V3$	0.165254596	0.3053797325
$V4$	-0.369075256	2.3458497486
$V5$	0.154797889	-0.1463807654
$V6$	-0.002163496	-0.0004627565
$V7$	0.618052068	-0.0322128171
$V8$	-1.661191235	-0.4919980543
$V9$	-1.495818440	-1.6309537953
$V10$	0.134092628	-0.3070875776
$V11$	0.355055710	0.2532306865
$V12$	-0.818036073	-1.5156344987
$V13$	-1.157559376	0.0511839665
$V14$	-0.002691206	0.0028529846

var dod labus rezultātus pat gadījumos, kad pieņēmumi nav spēkā. Ja novērtēt precizitāti, izmantojot modelēšanas datus, LDA un NNET dod 100% precizitāti, bet CRT 98.13% precizitāti. Darbā tika parādīts, ka precizitātes novērtējums no modelēšanas datiem dod pārāk optimistiskus rezultātus. Tāpēc precizitātes novērtēšanai tika pielietotas trīs augstāk aprakstītās metodes: modelēšana un pārbaude, butstraps un krosvalidācija. Modelēšanas un pārbaudes precizitātes novērtējums tika aprēķināts, izmantojot 80% datu modelēšanai un 20% pārbaudei. Procedūra tika atkārtota 1000 reizes. Vidējais precizitātes novērtējums, tā dispersija un 5% un 95% empīriskās kvantīles ir redzamas 5.1. tabulā. Butstrapu precizitātes novērtējumam tika izmantota butstrapa izlase ar apjomu 178. Procedūras atkārtojumu skaits ir 1000 reizes. Rezultāti ir attēloti 5.2. tabulā. Krosvalidācijas precizitātes novērtējums tika iegūts, atkārtojot 10 daļu krosvalidācijas procedūru 1000 reizes. Precizitātes vidējās vērtības novērtējums, empīriskā dispersija un kvantiles ir parādītas 5.3. tabulā.

Visi precizitātes novērtējumi dod līdzīgus rezultātus, atšķirīgs ir tas, ka butstraps dod visplatākos precizitātes ticamības intervālus, bet krosvalidācija visšaurākos. Visas trīs metodes izvēlējās LDA kā vislabāko klasifikatoru. Precizitāte tiešām ir pārsteidzoši laba - pie vispesimistiskākā varianta LDA dod 95% precizitāti. Izrādījās, ka cilvēku gaume attiecībā uz vīnu tiešām var tikt izskaidrota ar tā ķīmisko sastāvu. Telpu $\mathbb{R}^{13}$ var sadalīt apgabalos A, B un C ar divu diskriminantu hiperplakņu $g_1(V)$ un $g_2(V)$ palīdzību. Abu funkciju koeficienti ir atrodami 5.4. tabulā.

6. Secinājumi

Darbā tika aplūkotas un teorētiski pamatotas divas statistiskās klasifikācijas metodes - lineāra un kodolu diskriminantu analīze, kā arī divas datu izraces metodes - klasifikācijas koki un neironu tīkli. Lineārai diskriminantu analīzei tika parādīts kā tiek iegūtas diskriminācijas funkcijas un diskriminācijas robežas. Ar simulāciju palīdzību tika parādīts, ka metode strādā pietiekoši labi pat gadījumos, kad lineārās diskriminantu analīzes pieņēmumi par grupu kovariāciju matricu vienādību un sadalījumu atbilstību normālajam neizpildās, bet klases ir atdalāmas ar hiperplakni.

Kodolu diskriminantu analīze dod iespēju konstruēt diskriminācijas funkcijas bez pieņēmumiem par grupu sadalījumiem, bet novērtējot tos ar kodolu gludināšanu. Empīriski tika novērots, ka metode dod ļoti labus, salīdzinājumā ar lineāro diskriminantu analīzi, rezultātus, neatkarīgi no joslas platumu matricas novērtēšanas metodes (darbā apskatītas krosvalidācijas un ievietošanas metodes). Vienīgais klasifikatora trūkums ir tas, ka joslas platumu matricas novērtēšana ir ietilpīgs darba un laika process, tāpēc grūti realizējams, kad dimensiju skaits ir lielāks par seši. Šī iemesla dēļ metode nav guvusi plašu atzinību praksē.

Klasifikācijas koki un neironu tīkli arī dod labus rezultātus, tuvus kodolu diskriminantu analīzei, pie daudz mazāka resursu patēriņa. Tāpēc datu izraces metodes var uzskatīt par statistisko metožu nopietniem konkurentiem. Klasifikācijas koku priekšrocība ir viegla interpretācija un pārskatāmība. Darbā ir aprakstīti trīs klasifikācijas robežas meklēšanas metodes, t.i. minimizējot Gini indeksu, entropiju vai kross-entropiju. Tika apskatīta un ilustrēta metode, ar kuras palīdzību var veiksmīgi cīnīties ar klasifikatora pārmērīgu pielāgošanos datiem - koku apgriešana.

Jāatzīst, ka neskatoties uz klasifikācijas koku priekšrocībām un plašu atzinību medicīnā un mārketingā, neironu tīkli ir spēcīgāka metode, kura tiek veiksmīgi izmantota pie liela dimensiju skaita un ir spējīga atpazīt sarežģītas datu struktūras. Trūkums ir to sa-

režģītā interpretācija un tas, ka parametru skaitam ir jābūt uzdotam iepriekš. Lai atrastu optimālo neironu skaitu, tika pielietota krosvalidācija. Koeficientu samazināšanas parametra ieviešana palīdzēja pat diskrēta sadalījuma gadījumā panākt augstu klasifikatora precizitāti.

Metožu empīriskai salīdzināšanai tika konstruēti 5 pretpiemēri, ar kuru palīdzību var novērot klasifikatoru priekšrocības un trūkumus. LDA nedod labus rezultātus, kad grupas nav atdalāmas lineāri vai viena grupa atrodas otras grupas iekšā. Kodolu diskriminantu analīze varētu būt atzīta par vislabāko klasifikācijas uzdevuma risināšanas paņēmienu, ja būtu spējīga klasificēt novērojumus ar lielu parametru skaitu. Klasifikācijas koku trūkums ir jūtīgums pret datu nesabalansētību. Bet neironu tīklu konverģence dažreiz ir apgrūtināta ar diskrēto vērtību parametriem. Kopumā var secināt, ka nevar izdalīt kādu no metodēm kā visstiprāko, lēmums par izmantojamo algoritmu ir jāpieņem, vadoties no klasifikācijas uzdevuma konteksta. Darbā tika parādīts, ka datu izraces metožu precizitāte ir vairāk atkarīga no izlases apjoma nekā statistisko metožu precizitāte. Ja pielieto metodes reāliem datiem, atšķirības starp klasifikatoru veidiem nav tik krasas.

Darbā tika izskatīti trīs klasifikatora precizitātes novērtēšanas metodes: modelēšanas un pārbaudes, krosvalidācija un butstraps. Tika izpētīta modelēšanas un pārbaudes precizitātes atkarība no tā, cik liela datu daļa tiek izmantota modeļa konstruēšanai, parādīts, kādā veidā butstrapa precizitāte mainās atkarībā no butstrapa izlases apjoma, kā arī novērotas krosvalidācijas precizitātes izmaiņas, atkarībā no datu dalīšanas daļu skaita. Tika empīriski parādīts, ka visas metodes dod priekšroku vienam un tam pašam klasifikatoru veidam, taču precizitātes novērtējumu empīriskās dispersijas ir dažādas. Krosvalidācijas novērtējuma dispersija ir ievērojami mazāka par modelēšanas un pārbaudes un butstrapa novērtējumu dispersijām. Tāpēc krosvalidācija varētu būt pievilcīgāka precizitātes ticamības intervālu konstruēšanai. Tika parādīts, ka precizitātes novērtējums ir atkarīgs ne tikai no novērtēšanas metodes, bet arī no klasifikatora veida. Vistuvākais īstās precizitātes novērtējums ir novērojams kodolu diskriminantu analīzei. Datu izraces klasifikatoru novērtējumi izrādījās vairāk nekā statistiskie klasifikatori atkarīgi no tā, kāda datu daļa tiek izmantota modeļa konstruēšanai un kāda precizitātes novērtēšanai. Jo lielāka datu daļa ir iesaistīta klasifikatora meklēšanai, jo tā precizitātes novērtējums ir lielāks. Neskatoties uz to, precizitātes novērtējumi visām, izņemot klasifikācijas kokus, metodēm ir zemāki par īsto precizitāti. Klasifikācijas koku precizitāte tiek novērtēta pārāk optimistiski ar mode-

lēšanas un pārbaudes metodi, kad pārbaudes datu apjoms ir pietiekoši maz. To pašu var teikt par krosvalidācijas metodi.

Statistisko un datu izraces klasifikatoru pētīšana var turpināties vairākos virzienos. Pirmkārt, ir interesanti, kādas metodes būtu visefektīvākās, lai konstruētu klasifikatora kopējās precizitātes ticamības intervālus. Otrā uzmanības cienīga problemātika ir klasifikatoru apvienošanas algoritmi - parametru telpa tiek sadalīta apgabalos, kuros katram klasifikatoram ir savs svars, jo precīzāk klasifikators strādā konkrētajā apgabalā, jo lielāka būs tā ietekme lēmuma pieņemšanā. Treškārt, var tikt pētītas metodes trokšņaino datu atdalīšanai. Diskriminācijas robežas tuvumā, kur dati īsti nav atdalāmi, tie netiek klasificēti vai tiek piekārtoti pie konkrētas klases ar ļoti mazu varbūtību. Visbeidzot, līdz šim nav īsti skaidrs, kā korekti pielietot datu izraces metodes gadījumā, kad jāminimizē ne tikai kļūda, bet arī zaudējumi, īpaši, ja izmaksu funkcija nav konstante. Jebkurā gadījumā šī tematika ir aktuāla un ar plašām pielietošanas iespējām praksē.

Izmantotā literatūra un avoti

[1] T.Hastie, R.Tibshirani, and J.Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.*, pages 1–533. Springer, New York, 2001.

[2] B.D.Ripley. *Pattern Recognition and Neural Networks.*, pages 1–403. Cambridge University Press, New York, 2004.

[3] R.O.Duda, P.E.Hart, and D.G.Stork. *Pattern Classification.*, pages 1–738. John Wiley & Sohns, New York, 2001.

[4] C.M.Bishop. *Pattern Recognition and Machine Learning.*, pages 1–703. Springer, Singapore, 2006.

[5] G.J.McLachlan. *Discriminant Analysis and Statistical Pattern Recognition.*, pages 1–526. John Wiley & Sohns, 2004.

[6] C.P.Robert. *The Bayesian Choise.*, pages 1–96. Springer, New York, 2007.

[7] W.Haerdle and L.Simar. *Applied Multivariate Statistics.*, pages 1–486. Springer, New York, 2003.

[8] J.E.Gentle, W.Haerdle, and Y.Mori. *Handbook of Computational Statistics: Concepts and Methods.*, pages 517–621. Springer, Berlin, 2004.

[9] L.Devroye, L.Gyorfi, and G.Lugosi. *A Probabilistic Theory of Pattern Recognition.*, pages 1–56. Springer, New York, 1996.

[10] T.Duong. Kernel density estimation and kernel discriminant analysis for multivariate data in r. *Journal of Statistical Software*, 2007.

[11] B.W.Silverman. *Density Estimation for Statistics and Data Analysis.*, pages 76–95. Chapman and Hall, New York, 1998.

[12] W.Haerdle, M.Mueller, S.Sperlich, and A.Werwatz. *Nonparametric and Semiparametric Models.*, pages 1–300. Springer, Heidelberg, 2004.

[13] J.Lu, K.N.Plataniotis, A.N.Venetsanopoulos, and J.Wang. An efficient kernel discriminant analysis method. *Pattern Recognition*, 2005.

[14] T.Duong and M.L.Hazelton. Plug-in bandwidth matrices for bivariate kernel density estimation. *Nonparametric Statistics*, 2003.

[15] Y.Zhang, M.L.King, and R.J.Hyndman. Bandwidth selection for multivariate kernel density. estimation using mcmc. Technical report, Monach University, Department of Econometrics and Business Statistics, 2004.

[16] K.J.Archer. An r package for deriving a classification tree for predicting an ordinal response. *Journal of Statistical Software*, 2010.

[17] W.N.Venables and B.D.Ripley. *Modern Applied Statistics with S.* 2002.

[18] J.Braun and M.Griebel. On a constructive proof of kolmogorov superposition theorem. Bonn, 2007.

[19] C.M.Bishop. *Neural Networks for Pattern Recognition.*, pages 1–496. Clarendon Press, Oxford, 1995.

[20] M.M.Gupta, L.Jin, and N.Homma. *Static and Dynamic Neural Networks.*, pages 1–164. John Wiley & Sohns, New Jersey, 2003.

[21] D.Michie, D.J.Spiegelhalter, and C.C.Taylor. *Machine Learning, Neural and Statistical Classification.*, pages 1–290. Ellis Horwood Limited, 1994.

[22] V.N.Vapnik. *The Nature of Statistical Learning Theory.*, pages 123–171. Springer, New York, 2000.

[23] B.Efron. *The Jackknife, the Bootstrap and Other Resampling Plans.*, pages 29–45. Capital City Press, Vermont, 1982.

[24] P.Cortez, A.Cerdeira, F.Almeida, T.Matos, and J.Reis. Modeling wine preferences by data mining from physicochemical properties. University of Minho, 2009.

A Pielikumi

A1. Programma piemēru konstruēšanai

```
N<-150
sigma <- matrix(c(1,0,0,2), ncol=2)
set.seed(3)
A <- rmvnorm(n=N, mean=c(1,5), sigma=sigma)
set.seed(5)
B <- rmvnorm(n=N, mean=c(2,2), sigma=sigma)
a<-rep(1,N)
b<-rep(2,N)
grupa<-c(a,b)
matrica<-matrix(ncol=2,nrow=N*2)
matrica[,1]<-t(c(A[,1],B[,1]))
matrica[,2]<-t(c(A[,2],B[,2]))
A_norm<-matrix(c(matrica,grupa),ncol=3)
A_norm<-as.data.frame(A_norm)
A_norm[,3][A_norm[,3]=="1"]<-"A"
A_norm[,3][A_norm[,3]=="2"]<-"B"
A_norm[,3]<-factor(A_norm[,3])
plot(A_norm[,2],A_norm[,1], pch=22,bg=c("red", "yellow")
[unclass(A_norm$V3)],xlab="y",ylab="x")
points(A_norm[,2],A_norm[,1], pch=22,bg=c("red", "yellow")
[unclass(A_norm$V3)],xlab="y",ylab="x")

set.seed(3)
```

```
x<-rnorm(300,0,2)
set.seed(5)
y<-rexp(300,1)
set.seed(20)
noise<-rnorm(300,0,1)
z<-c(1:length(x))
probs<-c(0.5,0.3,0.1,0.05,0.03,0.02)
set.seed(35)
d<-rdiscrete(300, probs, values = 1:length(probs))
B<-matrix(c(x,d,z),ncol=3)
A_diskr<-as.data.frame(B, row.names = NULL)
A_diskr[,3][A_diskr[,1]-A_diskr[,2]>=noise-2]<-"A"
A_diskr[,3][A_diskr[,1]-A_diskr[,2]<noise-2]<-"B"
A_diskr[,3]<-factor(A_diskr[,3])
plot(A_diskr[,1],A_diskr[,2], pch=22,bg=c("red", "yellow")
[unclass(A_diskr$V3)],xlab="y",ylab="x")
dim(A_kub[A_kub[,3]=="A",])

A<-matrix(c(x,y,z),ncol=3)
A_lin<-as.data.frame(A, row.names = NULL)
A_lin[,3][A_lin[,1]+A_lin[,2]>=noise]<-"A"
A_lin[,3][A_lin[,1]+A_lin[,2]<noise]<-"B"
A_lin[,3]<-factor(A_lin[,3])
plot(A_lin[,2],A_lin[,1], pch=22,bg=c("red", "yellow")
[unclass(A_lin$V3)],xlab="y",ylab="x")

A_kv<-as.data.frame(A, row.names = NULL)
A_kv[,3][(-0.8*(A_kv[,1]+A_kv[,2]-1)^2+2)>=noise]<-"A"
A_kv[,3][(-0.8*(A_kv[,1]+A_kv[,2]-1)^2+2)<noise]<-"B"
A_kv[,3]<-factor(A_kv[,3])
plot(A_kv[,2], A_kv[,1],pch=22,bg=c("red", "yellow")
[unclass(A_kv$V3)],xlab="y",ylab="x")
```

```
A_kub<-as.data.frame(A, row.names = NULL)
A_kub[,3][((0.5*A_kub[,1]^2+3*A_kub[,2]^2))>=3+noise]<-"A"
A_kub[,3][((0.5*A_kub[,1]^2+3*A_kub[,2]^2))<3+noise]<-"B"
A_kub[,3]<-factor(A_kub[,3])
plot(A_kub[,2],A_kub[,1], pch=22,bg=c("red", "yellow")
[unclass(A_kub$V3)],xlab="y",ylab="x")

x <- as.matrix(A_kub[A_kub[,3]=="B",1:2]) # n x p numeric matrix
center <- colMeans(x) # centroid
n <- nrow(x); p <- ncol(x); cov <- cov(x);
d <- mahalanobis(x,center,cov) # distances
qqplot(qchisq(ppoints(n),df=p),d,
ylab="Mahalanobis D2")
abline(a=0,b=1)
```

A2. Programma simulāciju veikšanai

```
##############
#####LDA#####
##############
model.lda2 <- lda(A_norm[,1:2],A_norm[,3], prior = c(1,1)/2)
#Accuracy
ct <- table(A_norm[,3], predict(model.lda2, A_norm[,1:2]))
prop.table(ct, 1)
diag(prop.table(ct, 1))
# total percent correct
model.lda1.acc<-sum(diag(prop.table(ct)))
### Test accuracy
N<-100000
A_test<-c();B_test<-c();AB_test<-c()
for (i in 1:1000)
{
```

```
sigma <- matrix(c(1,0,0,2), ncol=2)
A <- rmvnorm(n=N, mean=c(1,5), sigma=sigma)
B <- rmvnorm(n=N, mean=c(2,2), sigma=sigma)
a<-rep(1,N)
b<-rep(2,N)
grupa<-c(a,b)
matrica<-matrix(ncol=2,nrow=N*2)
matrica[,1]<-t(c(A[,1],B[,1]))
matrica[,2]<-t(c(A[,2],B[,2]))
TEST<-matrix(c(matrica,grupa),ncol=3)
TEST<-as.data.frame(TEST)
TEST[,3][TEST[,3]=="1"]<-"A"
TEST[,3][TEST[,3]=="2"]<-"B"
TEST[,3]<-factor(TEST[,3])
ct <- table(TEST[,3], predict(model.lda2, TEST[,1:2])$class)
A_test[i]<-prop.table(ct, 1)[1,1]
B_test[i]<-prop.table(ct, 1)[2,2]
AB_test[i]<-sum(diag(prop.table(ct)))
}
mean(A_test);mean(B_test;mean(AB_test)
vioplot(A_test,B_test,AB_test,col=7,ylim=c(0.5,1))
#Plots
par(mfrow=c(1,1))
plot(model.lda2, dimen=1) # fit from lda
plot(model.lda2, dimen=1, type="density") # fit from lda
partimat(V3~.,data=A_kub,method="lda",xlim=c(-2,8),main="")
##############
#####CRT#####
##############
dati.tr1<- rpart(A_norm[,3] ~.,A_norm[,1:2],
control=rpart.control(minsplit=1))
dati.tr<-prune(dati.tr1, dati.tr1$
```

```
cptable[which.min(dati.tr1$cptable[,"xerror"]),"CP"])
plot(dati.tr1);text(dati.tr);formula(dati.tr);summary(dati.tr)
ct <- table(A_norm[iz2,3],
predict(dati.tr,A_norm[iz2,1:2],type="class"))
prop.table(ct, 1)
diag(prop.table(ct, 1))
model.tr.acc<-sum(diag(prop.table(ct)))
model.tr.acc

### Test accuracy
A_test<-c();B_test<-c();AB_test<-c()
N<-100000
for (i in 1:1000)
{
sigma <- matrix(c(1,0,0,2), ncol=2)
set.seed(i)
A <- rmvnorm(n=N, mean=c(1,5), sigma=sigma)
set.seed(I-i)
B <- rmvnorm(n=N, mean=c(2,2), sigma=sigma)
a<-rep(1,N)
b<-rep(2,N)
grupa<-c(a,b)
matrica<-matrix(ncol=2,nrow=N*2)
matrica[,1]<-t(c(A[,1],B[,1]))
matrica[,2]<-t(c(A[,2],B[,2]))
TEST<-matrix(c(matrica,grupa),ncol=3)
TEST<-as.data.frame(TEST)
TEST[,3][TEST[,3]=="1"]<-"A"
TEST[,3][TEST[,3]=="2"]<-"B"
TEST[,3]<-factor(TEST[,3])
ct <- table(TEST[,3], predict(dati.tr,TEST[,1:2],type="class"))
A_test[i]<-prop.table(ct, 1)[1,1]
```

```
B_test[i]<-prop.table(ct, 1)[2,2]
AB_test[i]<-sum(diag(prop.table(ct)))
}
vioplot(A_test,B_test,AB_test,col=7,names=c("A", "B", "AB"))
mean(A_test);mean(B_test);mean(AB_test)
###Plots
partimat(V3~.,data=A_norm,method="rpart",
main="",control=rpart.control(1,0.000001))
##########
###NNET###
##########
target<-class.ind(A_norm[iz1,3])
dati.nnet <- nnet(A_norm[iz1,1:2],target, size=3, rang = 0.1,
decay = 5e-4, maxit = 20000,linout = FALSE, entropy = FALSE,
softmax = TRUE,censored = FALSE)
A_norm_nnet<-c();
A_norm_nnet[A_norm[,3]=="A"]<-1
A_norm_nnet[A_norm[,3]=="B"]<-2
pred<-c();
pred<-predict(dati.nnet, A_norm[2,1:2])
ct<-table(A_norm_nnet[sort()],apply(pred,1,which.is.max))
prop.table(ct, 1)[1,1]
prop.table(ct, 1)[2,2]
sum(diag(prop.table(ct)))
### Test accuracy
N<-100000
A_test<-c();B_test<-c();AB_test<-c()
for (i in 1:1000)
{
sigma <- matrix(c(1,0,0,2), ncol=2)
set.seed(i)
A <- rmvnorm(n=N, mean=c(1,5), sigma=sigma)
```

```
B <- rmvnorm(n=N, mean=c(2,2), sigma=sigma)
a<-rep(1,N)
b<-rep(2,N)
grupa<-c(a,b)
matrica<-matrix(ncol=2,nrow=N*2)
matrica[,1]<-t(c(A[,1],B[,1]))
matrica[,2]<-t(c(A[,2],B[,2]))
TEST<-matrix(c(matrica,grupa),ncol=3)
TEST<-as.data.frame(TEST)
TEST[,3][TEST[,3]=="1"]<-"A"
TEST[,3][TEST[,3]=="2"]<-"B"
TEST[,3]<-factor(TEST[,3])
pred<-c()
pred<-predict(dati.nnet, TEST[1:2])
ct<-table(TEST[,3],apply(pred,1,which.is.max))
A_test[i]<-prop.table(ct, 1)[1,1]
B_test[i]<-prop.table(ct, 1)[2,2]
AB_test[i]<-sum(diag(prop.table(ct)))

vioplot(A_test,B_test,AB_test,col=7)
mean(A_test);mean(B_test);mean(AB_test)
#plot
T<-matrix(c(runif(100000,-2,6),runif(100000,-3,9)),ncol=2)
pred<-c()
pred<-max.col(predict(dati.nnet,T))
pred[pred=="1"]<-"A"
pred[pred=="2"]<-"B"
T<-matrix(c(T[,1],T[,2],pred),ncol=3)
plot(T[T[,3]=="A",2],T[T[,3]=="A",1],col=8,xlim=c(min(A_norm[,2]),
max(A_norm[,2])),ylim=c(min(A_norm[,1]),max(A_norm[,1])),xlab="V2",
ylab="V1")
points(A_norm[A_norm[,3]=="B",2],A_norm[A_norm[,3]=="B",1],col=7,pch=22)
```

```
points(A_norm[A_norm[,3]=="A",2],A_norm[A_norm[,3]=="A",1],col=10,pch=22)
#########
###KDA###
#########
dati.kda<-A_norm[,1:2]
dati.kda.gr<-A_norm[,3]
Hpi1 <- Hkda(x = dati.kda, x.group = dati.kda.gr, bw = "scv",
pre = "scaled")
dati.kda$class<-c()
dati.kda$class<-kda(x = dati.kda, x.group = dati.kda.gr,
y=A_norm[,1:2], Hs = Hpi1)
ct<-table(A_norm[,3], dati.kda$class)
prop.table(ct,1)
diag(prop.table(ct, 1))
model.tr.acc<-sum(diag(prop.table(ct)))
### Test accuracy
N<-100000
A_test<-c();B_test<-c();AB_test<-c()
for (i in 1:1000)
{
sigma <- matrix(c(1,0,0,2), ncol=2)
A <- rmvnorm(n=N, mean=c(1,5), sigma=sigma)
B <- rmvnorm(n=N, mean=c(2,2), sigma=sigma)
a<-rep(1,N)
b<-rep(2,N)
grupa<-c(a,b)
matrica<-matrix(ncol=2,nrow=N*2)
matrica[,1]<-t(c(A[,1],B[,1]))
matrica[,2]<-t(c(A[,2],B[,2]))
TEST<-matrix(c(matrica,grupa),ncol=3)
TEST<-as.data.frame(TEST)
TEST[,3][TEST[,3]=="1"]<-"A"
```

```
TEST[,3][TEST[,3]=="2"]<-"B"
TEST[,3]<-factor(TEST[,3])
dati.kda$class<-c()
dati.kda$class<-kda(x = dati.kda, x.group = dati.kda.gr,
y=TEST[,1:2], Hs = Hpi1)
ct<-table(TEST[,3], dati.kda$class)
A_test[i]<-prop.table(ct, 1)[1,1]
B_test[i]<-prop.table(ct, 1)[2,2]
AB_test[i]<-sum(diag(prop.table(ct)))

vioplot(A_test,B_test,AB_test,col=7,names=c("A", "B", "AB") )
mean(A_test)
mean(B_test)
mean(AB_test)
#zimejums
## bivariate example
ir <- dati.kda[,1:2]
ir.gr <- dati.kda.gr
kda.fhat <- kda.kde(ir, ir.gr, Hs=Hpi1)
plot(kda.fhat, cont=0, partcol=4:6)
plot(kda.fhat,ir,ir.gr, drawlabels=FALSE, drawpoints=TRUE)
## univariate example
dia <- dati.kda[,1]
dia.gr <- dati.kda.gr
hs <- hkda(x=dia, x.gr=dia.gr)
kda.fhat <- kda.kde(dia, dia.gr, hs=hs)
kda(dia, dia.gr, y=dia, hs=hs)
plot(kda.fhat,dia,dia.gr)
hist(dia[dia.gr=="A"],freq=F,breaks=20,col=2,xlim=c(min(dia),
max(dia)),ylim=c(0,1))
hist(dia[dia.gr=="B"],freq=F,add=TRUE,breaks=20,col=3)
```

A3. Programma precizitātes novērtēšanai

```
######################
##### TRAIN TEST #####
######################
TTTLDA_A<-c();TTTLDA_B<-c();TTTLDA_ABC<-c()
simulacijas<-c(270,240,210,180,150)
for (i in 1:(length(simulacijas)))
{
e1.lda<-c();e2.lda<-c();e4.lda<-c()
e1.crt<-c();e2.crt<-c();e4.crt<-c()
e1.nnet<-c();e2.nnet<-c();e4.nnet<-c()
e1.kda<-c();e2.kda<-c();e4.kda<-c()

for (k in 1:1000)
{
boot<-sample(300,simulacijas[i],replace=FALSE)
train<-boot
test<-rep(1:300)[-unique(boot)]
###LDA
model.lda1 <- lda(A_kv[train,1:2],A_kv[train,3])
ct <- table(A_kv[test,3], predict(model.lda1, A_kv[test,1:2])$class)
e1.lda[k]<-prop.table(ct,1)[1,1]
e2.lda[k]<-prop.table(ct,1)[2,2]
e4.lda[k]<-sum(diag(prop.table(ct)))
###CRT
dati.tr1<- rpart(A_kv[train,3] ~.,
A_kv[train,1:2],control=rpart.control(minsplit=1))
dati.tr<-prune(dati.tr1, dati.tr1$cptable
[which.min(dati.tr1$cptable[,"xerror"]),"CP"])
ct <- table(A_kv[test,3], predict(dati.tr,A_kv[test,1:2],type="class"))
e1.crt[k]<-prop.table(ct,1)[1,1]
e2.crt[k]<-prop.table(ct,1)[2,2]
```

```
e4.crt[k]<-sum(diag(prop.table(ct)))
###NNET
dati.nnet <- nnet(A_kv[train,1:2],class.ind(A_kv[,3])[train,], size=3,
rang = 0.1, decay =1*10^{-6}, maxit = 20000,saftmax=TRUE)
pred<-max.col(predict(dati.nnet, A_kv[test,1:2]))
pred[pred=="1"]<-"A"
pred[pred=="2"]<-"B"
ct <- table(A_kv[test,3],pred)
if (length(pred[pred=="A"])>0)
e1.nnet[k]<-prop.table(ct,1)["A","A"] else e1.nnet[k]<-0
if (length(pred[pred=="B"])>0)
e2.nnet[k]<-prop.table(ct,1)["B","B"] else e2.nnet[k]<-0
e4.nnet[k]<-sum(diag(prop.table(ct)))
###KDA
Hpi1 <- Hkda(x = A_kv[train,1:2], x.group = A_kv[train,3],
bw = "scv",pre = "sphere")
dati.kda.class<-kda(x = A_kv[train,1:2], x.group = A_kv[train,3],
y=A_kv[test,1:2], Hs = Hpi1)
ct<-table(A_kv[test,3], dati.kda.class)
e1.kda[k]<-prop.table(ct,1)[1,1]
e2.kda[k]<-prop.table(ct,1)[2,2]
e4.kda[k]<-sum(diag(prop.table(ct)))
}
TTTLDA_A[((i-1)*8+1):(8*i)]<-c(mean(e1.lda),
var(e1.lda),mean(e1.crt),var(e1.crt),
mean(e1.nnet),var(e1.nnet),mean(e1.kda),var(e1.kda))
TTTLDA_B[((i-1)*8+1):(8*i)]<-c(mean(e2.lda),
var(e1.lda),mean(e2.crt),var(e2.crt),
mean(e2.nnet),var(e2.nnet),mean(e2.kda),var(e2.kda))
TTTLDA_ABC[((i-1)*8+1):(8*i)]<-c(mean(e4.lda),
var(e4.lda),mean(e4.crt),var(e4.crt),
mean(e4.nnet),var(e4.nnet),mean(e4.kda),var(e4.kda))
```

```
}
#####################
##### BOOTSTRAP #####
#####################
sim<-c(50,100,150,200,300)
rezultats_kv_boot<-c()
for (j in 1:length(sim))
{
e1.lda<-c();e2.lda<-c();e4.lda<-c()
e1.crt<-c();e2.crt<-c();e4.crt<-c()
e1.nnet<-c();e2.nnet<-c();e4.nnet<-c()
e1.kda<-c();e2.kda<-c();e4.kda<-c()

for (k in 1:1000)
{
boot<-sample(300,sim[j],replace=TRUE)
train<-boot
test<-rep(1:300)[-unique(boot)]
###LDA
model.lda1 <- lda(A_kv[train,1:2],A_kv[train,3])
ct <- table(A_kv[test,3],
predict(model.lda1, A_kv[test,1:2])$class)
e1.lda[k]<-prop.table(ct,1)[1,1]
e2.lda[k]<-prop.table(ct,1)[2,2]
e4.lda[k]<-sum(diag(prop.table(ct)))
###CRT
dati.tr1<- rpart(A_kv[train,3] ~.,
A_kv[train,1:2],control=rpart.control(minsplit=1))
dati.tr<-prune(dati.tr1, dati.tr1$
cptable[which.min(dati.tr1$cptable[,"xerror"]),"CP"])
ct <- table(A_kv[test,3],
predict(dati.tr,A_kv[test,1:2],type="class"))
```

```
e1.crt[k]<-prop.table(ct,1)[1,1]
e2.crt[k]<-prop.table(ct,1)[2,2]
e4.crt[k]<-sum(diag(prop.table(ct)))
###NNET
dati.nnet <- nnet(A_kv[train,1:2],class.ind(A_kv[,3])
[train,], size=3, rang = 0.1, decay =1*10^{-6},
maxit = 20000,saftmax=TRUE)
pred<-max.col(predict(dati.nnet, A_kv[test,1:2]))
pred[pred=="1"]<-"A"
pred[pred=="2"]<-"B"
ct <- table(A_kv[test,3],pred)
if (length(pred[pred=="A"])>0) e1.nnet[k]
<-prop.table(ct,1)["A","A"] else e1.nnet[k]<-0
if (length(pred[pred=="B"])>0) e2.nnet[k]
<-prop.table(ct,1)["B","B"] else e2.nnet[k]<-0
e4.nnet[k]<-sum(diag(prop.table(ct)))
###KDA
Hpi1 <- Hkda(x = A_kv[train,1:2], x.group = A_kv[train,3],
bw = "plugin",pre = "sphere")
dati.kda.class<-kda(x = A_kv[train,1:2],x.group=A_kv[train,3],
y=A_kv[test,1:2], Hs = Hpi1)
ct<-table(A_kv[test,3], dati.kda.class)
e1.kda[k]<-prop.table(ct,1)[1,1]
e2.kda[k]<-prop.table(ct,1)[2,2]
e4.kda[k]<-sum(diag(prop.table(ct)))
}
rezultats_kv_boot[(24*(j-1)+1):(24*j)]<-c(
mean(e1.lda),var(e1.lda),mean(e1.crt),var(e1.crt),
mean(e1.nnet),var(e1.nnet),mean(e1.kda),var(e1.kda),
mean(e2.lda),var(e2.lda),mean(e2.crt),var(e2.crt),
mean(e2.nnet),var(e2.nnet),mean(e2.kda),var(e2.kda),
mean(e4.lda),var(e4.lda),mean(e4.crt),var(e4.crt),
```

```
mean(e4.nnet),var(e4.nnet),mean(e4.kda),var(e4.kda)
)
}
############################
##### CROSS-VALIDATION #####
############################
sim<-c(5,10,20,30,50)
rezultats_kv_cv<-c()
for (j in 1:length(sim))
{
LDA_A<-c();LDA_B<-c();LDA_ABC<-c()
CRT_A<-c();CRT_B<-c();CRT_ABC<-c()
NNET_A<-c();NNET_B<-c();NNET_ABC<-c()
KDA_A<-c();KDA_B<-c();KDA_ABC<-c()
for (i in 1:1000)
{
e1.lda<-c();e2.lda<-c();e4.lda<-c()
e1.crt<-c();e2.crt<-c();e4.crt<-c();
e1.nnet<-c();e2.nnet<-c();e4.nnet<-c();
e1.kda<-c();e2.kda<-c();e4.kda<-c();

boot<-sample(300,300,replace=FALSE)
for (k in 1:sim[j])
{
train<-boot[-(((300/sim[j])*(k-1)+1):((300/sim[j])*k))]
###LDA
model.lda1 <- lda(A_kv[train,1:2],A_kv[train,3])
ct <- table(A_kv[-train,3], predict(model.lda1,
A_kv[-train,1:2])$class)
e1.lda[k]<-prop.table(ct,1)[1,1]
e2.lda[k]<-prop.table(ct,1)[2,2]
e4.lda[k]<-sum(diag(prop.table(ct)))
```

```
###CRT
dati.tr1<- rpart(A_kv[train,3] ~.,A_kv[train,1:2],
control=rpart.control(minsplit=1))
dati.tr<-prune(dati.tr1, dati.tr1$
cptable[which.min(dati.tr1$cptable[,"xerror"]),"CP"])
ct <- table(A_kv[-train,3], predict
(dati.tr,A_kv[-train,1:2],type="class"))
e1.crt[k]<-prop.table(ct,1)[1,1]
e2.crt[k]<-prop.table(ct,1)[2,2]
e4.crt[k]<-sum(diag(prop.table(ct)))
###NNET
dati.nnet <- nnet(A_kv[train,1:2],class.ind(A_kv[,3])
[train,], size=3, rang = 0.1, decay =1*10^{-6},
maxit = 20000,saftmax=TRUE)
dati.nnet$convergence
pred<-max.col(predict(dati.nnet, A_kv[-train,1:2]))
pred[pred=="1"]<-"A"
pred[pred=="2"]<-"B"
ct <- table(A_kv[-train,3],pred)
if (length(pred[pred=="A"])>0) e1.nnet[k]
<-prop.table(ct,1)["A","A"] else e1.nnet[k]<-0
if (length(pred[pred=="B"])>0) e2.nnet[k]
<-prop.table(ct,1)["B","B"] else e2.nnet[k]<-0
e4.nnet[k]<-sum(diag(prop.table(ct)))
###KDA
Hpi1 <- Hkda(x = A_kv[train,1:2], x.group =
A_kv[train,3], bw = "plugin",pre = "sphere")
dati.kda.class<-kda(x = A_kv[train,1:2],
x.group = A_kv[train,3],
y=A_kv[-train,1:2], Hs = Hpi1)
ct<-table(A_kv[-train,3], dati.kda.class)
e1.kda[k]<-prop.table(ct,1)[1,1]
```

```
e2.kda[k]<-prop.table(ct,1)[2,2]
e4.kda[k]<-sum(diag(prop.table(ct)))
}
LDA_A[i]<-mean(e1.lda);LDA_B[i]<-
mean(e2.lda);LDA_ABC[i]<-mean(e4.lda);
CRT_A[i]<-mean(e1.crt);CRT_B[i]<-
mean(e2.crt);CRT_ABC[i]<-mean(e4.crt);
NNET_A[i]<-mean(e1.nnet);NNET_B[i]<-
mean(e2.nnet);NNET_ABC[i]<-mean(e4.nnet);
KDA_A[i]<-mean(e1.kda);KDA_B[i]<-
mean(e2.kda);KDA_ABC[i]<-mean(e4.kda);
}
rezultats_kv_cv[(24*(j-1)+1):(24*j)]<-c(
mean(LDA_A),var(LDA_A),mean(CRT_A),var(CRT_A),mean(NNET_A),
var(NNET_A),mean(KDA_A),var(KDA_A),
mean(LDA_B),var(LDA_B),mean(CRT_B),var(CRT_B),mean(NNET_B),
var(NNET_B),mean(KDA_B),var(KDA_B),mean(LDA_ABC),
var(LDA_ABC),mean(CRT_ABC),var(CRT_ABC),mean(NNET_ABC),
var(NNET_ABC),mean(KDA_ABC),var(KDA_ABC)
)
}
```

Diplomdarbs “Statistiskās un datu izraces metodes klasifikācijas uzdevumos” izstrādāts LU Fizikas un Matemātikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Anastasija Tetereva

_____ _____

(paraksts) (datums)

Rekomendēju darbu aizstāvēšanai.
Vadītājs: docents Dr. Math. Jānis Valeinis

_____ _____

(paraksts) (datums)

Recenzents: docente Dr. math. Nadežda Siņenko

_____ _____

(paraksts) (datums)

Darbs iesniegts Matemātikas nodaļā _____

(datums)

_____

(darbu pieņēma)

Darbs aizstāvēts gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____, vērtējums_____

(datums)

Komisijas sekretārs/-e: Ingrīda Uļjane _____

(paraksts)