Vilniaus universitetas
Matematikos ir informatikos fakultetas

Algirdas Mačiulis

Duomenų tyrimas

Paskaitų konspektas

2011

Turinys

Įvadas

Duomenų tyrimo (kartais dar vadinamo duomenų kasyba) tikslas - atskleisti objektyviai egzistuojančius dėsningumus, sąryšius bei vidinę struktūrą didelėse įvairios prigimties duomenų aibėse. Didžioji dalis kurso skirta tokių dėsningumų paieškos metodams ir algoritmams nagrinėti.

Negalima nubrėžti labai aiškios ribos tarp duomenų tyrimo ir daugiamatės statistikos. Todėl dalis šiame kurse nagrinėjamų temų, pavyzdžiui, kai kurie klasifikavimo metodai, regresija, klasterizacija lengviau bus suprantamos skaitytojams, jau susipažinusiems su pagrindinėmis algebros bei matematinės statistikos sąvokomis.

Visos kurso temos suskirstytos į penkis skyrius. Pirmajame skyriuje pateikiamos pagrindinės tikimybių teorijos bei informacijos teorijos sąvokos ir rezultatai. Šį skyrių gali praleisti skaitytojai, kuriems žinomos sąvokos: atsitiktinis įvykis ir atsitiktinis dydis, sąlyginė tikimybė, atsitiktinių dydžių pasiskirstymo funkcija, vidurkis, dispersija, koreliacija, entropija, tarpusavio informacija. Antrajame skyriuje aptariamos duomenų rūšys, galimos jų transformacijos. Apžvelgiamos ir pavyzdžiais iliustruojamos duomenų tyrimo uždavinių klasės. 3 - 5 skyriuose nagrinėjami konkretūs duomenų tyrimo uždavinių sprendimo metodai ir algoritmai.

Praktiniuose užsiėmimuose patartina naudoti specializuotą programinę įrangą. Paminėsime tris programų paketus.

SAS Enterprise Miner -
http://www.sas.com/technologies/analytics/datamining/miner/

SPSS Modeler -
http://www-01.ibm.com/software/analytics/spss/products/modeler/

Weka - http://www.cs.waikato.ac.nz/ml/weka/

Pirmieji du - mokami. Weka yra atviro Java kodo programų paketas, kurio aprašymą galima rasti [10, 11] knygose.

1 Pagrindinės tikimybių teorijos ir informacijos teorijos sąvokos

1.1 Tikimybės. Atsitiktiniai dydžiai ir jų skirstiniai

1.1.1 apibrėžimas. *Tegu Ω yra baigtinė arba skaiti aibė, $\mathcal{P}(\Omega)$ visų jos poaibių sistema, $P(\omega)$, $\omega \in \Omega$, neneigiami skaičiai, tenkinantys sąlygą*

$$\sum_{\omega \in \Omega} P(\omega) = 1.$$

Tikimybe vadinsime funkciją $P : \mathcal{P}(\Omega) \to [0, 1]$, kiekvienam $A \subset \Omega$ apibrėžiamą lygybe

$$P(A) = \sum_{\omega \in A} P(\omega).$$

Porą (Ω, P) vadinsime diskrečiąja tikimybine erdve, o Ω - elementariųjų įvykių aibe.

Diskrečioji tikimybinė erdvė vadinama baigtine, kai jos elementariųjų įvykių aibės elementų skaičius $|\Omega|$ yra baigtinis.

Jei eksperimentas yra nusakomas tikimybine erdve (Ω, P) , tai aibės Ω elementai ω dar vadinami jo *elementariosiomis baigtimis*. Tada bet kuri to eksperimento baigtis A, sudaryta iš elementariųjų baigčių, vadinama *atsitiktiniu įvykiu* tikimybinėje erdvėje (Ω, P). Kitaip sakant, atsitiktiniu įvykiu laikysime bet kurį aibės Ω poaibį. Kaip įprasta, būtinąjį įvykį žymėsime Ω, negalimąjį, t.y. neturintį palankių elementariųjų baigčių, žymėsime tuščios aibės simboliu $\emptyset$, įvykiui A priešingą įvykį - $\overline{A}$.

1.1.1 pavyzdys. (*Klasikinis tikimybės apibrėžimas.*) Tai yra baigtinė tikimybinė erdvė, kurioje visi elementarieji įvykiai vienodai galimi. Taigi, jei

$$\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\},$$

tai

$$P(\omega_1) = P(\omega_2) = \cdots = P(\omega_n) = \frac{1}{n}.$$

Todėl pagal apibrėžimą įvykio $A = \{\omega_{i_1}, \omega_{i_2}, \ldots, \omega_{i_k}\} \subset \Omega$ tikimybė bus

$$P(A) = \sum_{j=1}^{k} P(\omega_{i_j}) = \frac{k}{n}.$$

Kitaip sakant, įvykio tikimybė yra lygi jam palankių elementariųjų baigčių skaičiaus ir visų elementariųjų baigčių skaičiaus santykiui.

1.1.2 pavyzdys. Metamos trys simetriškos monetos. Raskime įvykio $A =$\{atsivertė bent vienas herbas\} tikimybę. Šiuo atveju galime sudaryti tokią elementariųjų įvykių aibę

$$\Omega = \{\omega_0, \omega_1, \omega_2, \omega_3\},$$

čia $\omega_i =$\{atsivertė i herbų\}. Tada $A = \{\omega_1, \omega_2, \omega_3\}$.
Atrodytų, kad $P(A) = \frac{3}{4}$. Tačiau patyrę monetų mėtytojai pastebės, kad taip nėra. Iš tikrųjų ne visi įvykiai ω_i yra vienodai galimi. Todėl klasikinis tikimybės apibrėžimas čia netinka, o eksperimento sąlygas atitinkančios elementariųjų įvykių tikimybės yra

$$P(\omega_0) = P(\omega_3) = \frac{1}{8}, \quad P(\omega_1) = P(\omega_2) = \frac{3}{8}.$$

Taigi

$$P(A) = P(\omega_1) + P(\omega_2) + P(\omega_3) = \frac{7}{8}.$$

1.1.2 apibrėžimas. *Įvykiai A ir B vadinami nesuderinamais, kai $P(A \cap B) = 0$. Jei $A \cap B = \emptyset$, tai A ir B vadinami nesutaikomais.*

Pastebėsime, kad bet kurie nesutaikomi įvykiai yra ir nesuderinami. Diskrečiojoje tikimybinėje erdvėje, neturinčioje nulinės tikimybės elementariųjų įvykių, šios dvi sąvokos ekvivalenčios.

1.1.3 pavyzdys. Dėžėje yra k baltų ir m juodų rutulių. Atsitiktinai be grąžinimo traukiame du rutulius. Nagrinėsime įvykius $A =$\{pirmasis rutulys baltas\} ir $B =$\{antrasis rutulys baltas\}. Tegu

$$\Omega = \{\omega_{bb}, \omega_{bj}, \omega_{jb}, \omega_{jj}\},$$

čia elementariųjų įvykių ω indeksai žymi ištraukto rutulio spalvą. Pavyzdžiui, $\omega_{bj} =$ \{pirmasis rutulys baltas, o antras - juodas\}. Tada

$$A = \{\omega_{bb}, \omega_{bj}\}, \quad B = \{\omega_{bb}, \omega_{jb}\}, \quad A \cap B = \{\omega_{bb}\} \neq \emptyset.$$

Matome, kad įvykiai A ir B yra sutaikomi. Rasime jų tikimybes. Kadangi

$$P(\omega_{bb}) = \frac{k(k-1)}{(k+m)(k+m-1)}, \quad P(\omega_{jj}) = \frac{m(m-1)}{(k+m)(k+m-1)},$$

$$P(\omega_{bj}) = P(\omega_{jb}) = \frac{k \cdot m}{(k+m)(k+m-1)}\,,$$

tai

$$P(A) = P(\omega_{bb}) + P(\omega_{bj}) = P(\omega_{bb}) + P(\omega_{jb}) = P(B) = \frac{k}{k+m}\,.$$

Tačiau įvykiai A ir B ne visada bus suderinami, nes

$$P(A \cap B) = P(\omega_{bb}) = \frac{k(k-1)}{(k+m)(k+m-1)} = 0\,,$$

kai $k \leq 1$.

Tegu $A,\ B,\ B_1,\ A_1, A_2, \ldots$ yra atsitiktiniai įvykiai diskrečiojoje tikimybinėje erdvėje (Ω, P). Priminsime pagrindines tikimybės savybes, išplaukiančias iš jos apibrėžimo.

1. $P(\emptyset) = 0, \quad P(\Omega) = 1.$

2. Jei $A \subset B$, tai $P(A) \leq P(B)$.

3. Jei A ir B nesuderinami ir $A_1 \subset A,\ B_1 \subset B$, tai įvykiai A_1 ir B_1 taip pat bus nesuderinami.

4. Jeigu įvykiai $A_1, A_2, \ldots$ poromis nesuderinami, t.y. $P(A_i \cap A_j) = 0$ visiems natūraliesiems $i \neq j$, tai
$$P(\bigcup_i A_i) = \sum_i P(A_i)\,.$$

5. Jei $A \subset B$, tai $P(B \setminus A) = P(B) - P(A)\,.$

6. $P(\overline{A}) = 1 - P(A)\,.$

7. $P(A \cup B) = P(A) + P(B) - P(A \cap B)\,.$

Tikimybių teorijoje vartojama ir abstrakti tikimybinės erdvės sąvoka. Bendruoju atveju Ω gali būti bet kuri netuščia aibė. Ką tuomet laikyti atsitiktiniais įvykiais? Kai Ω turi be galo daug skirtingų elementų, tai begalinės jų sąjungos bei sankirtos gali būti tokie Ω poaibiai, kad, įvedant tikimybės sąvoką, kils dideli matematiniai sunkumai. Todėl atsitiktinių įvykių aibe laikoma tik tam tikra, pakankamai "turtinga" Ω paibių sistema $\mathcal{F}$, turinti tokias savybes:

i) $\Omega \in \mathcal{F}$,

ii) $A \in \mathcal{F} \Rightarrow \overline{A} \in \mathcal{F}$,

iii) $A_1, A_2, \ldots \in \mathcal{F} \Rightarrow A_1 \cup A_2 \cup \ldots \in \mathcal{F}$.

Paibių sistema $\mathcal{F}$ vadinama atsitiktinių įvykių σ (sigma) algebra. Tada tikimybe vadinama funkcija $P : \mathcal{F} \to [0, 1]$, jei

i) $P(\Omega) = 1$;

ii) jei $A_i \in \mathcal{F}$ ir $A_i \cap A_j = \emptyset$ visiems natūraliesiems $i \neq j$, tai $P(A_1 \cup A_2 \cup \ldots) = P(A_1) + P(A_2) + \ldots$.

Taip apibrėžta tikimybė tenkina visas anksčiau suformuluotas savybes ir neprieštarauja 1.1.1 apibrėžimui. Mes neakcentuosime σ algebros vaidmens. Kalbėdami apie atsitiktinius įvykius, turėsime omenyje, kad jie priklauso tam tikrai σ algebrai. Pastebėsime, kad $\mathcal{P}(\Omega)$ yra σ algebra.

Sąlyginė tikimybė. Dažnai galimybė įvykti vienam įvykiui priklauso nuo to, ar įvyksta kitas įvykis. Tarkime norime rasti įvykio A tikimybę, žinodami, kad įvyko įvykis B. Tokia tikimybė vadinama įvykio A *sąlygine* tikimybe ir žymima $P(A|B)$ (skaitoma "tikimybė, kad įvyks A su sąlyga, kad įvyko B" arba "įvyks A , jeigu įvyko B"). Jei $P(B) > 0$, tai

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

Prisiminę 1.1.3 pavyzdžio eksperimentą, kai $k > 0$, nesunkiai rasime, kad ištraukus baltą rutulį, tikimybė vėl ištraukti baltą bus

$$P(B|A) = \frac{k-1}{k+m-1}.$$

Kaip matome, šiuo atveju $P(B|A) \neq P(B)$. Tai rodo, kad įvykio B tikimybė priklauso nuo to, ar įvyko įvykis A. Tokie įvykiai vadinami *priklausomais.*

Įvykių nepriklausomumas - tai viena iš svarbesniųjų tikimybių teorijos sąvokų. Pateiksime griežtesnį jos apibrėžimą. Iš sąlyginės tikimybės apibrėžimo išplaukia vadinamoji *tikimybių daugybos teorema* :

$$P(A \cap B) = P(B)P(A|B) = P(A)P(B|A). \tag{1.1}$$

Kai A ir B yra nepriklausomi įvykiai, turėsime $P(A|B) = P(A)$ ir $P(B|A) = P(B)$. Todėl, atsižvelgę į (1.1), gauname tokį įvykių nepriklausomumo apibrėžimą.

1.1.3 apibrėžimas. *Įvykius A ir B vadinsime nepriklausomais, jeigu*

$$P(A \cap B) = P(A)P(B).$$

Didesnio įvykių skaičiaus nepriklausomumas apibrėžiamas sudėtingiau. Pavyzdžiui, įvykiai A, B ir C vadinami nepriklausomais, jei teisingos visos keturios lygybės

$$\begin{aligned} P(A \cap B) &= P(A)P(B), \quad P(A \cap C) = P(A)P(C), \\ P(B \cap C) &= P(B)P(C), \quad P(A \cap B \cap C) = P(A)P(B)P(C). \end{aligned}$$

Nereikia šios sąvokos painioti su nesutaikomumu. Nepriklausomi įvykiai nebūtinai nesutaikomi. Dažnai įvykių nepriklausomumas pastebimas intuityviai. Tačiau intuicija gali ir suklaidinti.

1.1.4 pavyzdys. Metame lošimo kauliuką. Nagrinėsime įvykius

A={atsivertė lyginis skaičius akučių} ,

B={atsivertė ne mažiau kaip 4 akutės} ,

C={atsivertė daugiau kaip 4 akutės} .

Ar šie įvykiai priklausomi?

Apskaičiuosime įvykių tikimybes. Nesunku suprasti, kad

$$A = \{2, 4, 6\}, \quad B = \{4, 5, 6\}, \quad C = \{5, 6\}$$

$$A \cap B = \{4, 6\}, \quad A \cap C = \{6\}.$$

Todėl

$$P(A)P(C) = \frac{3}{6} \cdot \frac{2}{6} = \frac{1}{6} = P(A \cap C).$$

Taigi įvykiai A ir C nepriklausomi. Tačiau

$$P(A)P(B) = \frac{3}{6} \cdot \frac{3}{6} \neq \frac{2}{6} = P(A \cap B).$$

Todėl gauname, kad įvykiai A ir B , o tuo pačiu ir visi trys įvykiai A, B ir C , yra priklausomi.

Pilnosios tikimybės ir Bajeso formulės. Tarkime, kad slaptas pranešimas užšifruotas raidėmis a, b, c ir žinoma, kad paprastai pusę šifruoto teksto sudaro raidės a, o raidė b

sutinkama dvigubai dažniau nei c. Be to, kol pasiekia adresatą, vidutiniškai 10% raidžių b bei 5% raidžių c iškraipomos ir virsta raidėmis a. Kokia tikimybė, kad tryliktas adresato gauto šifruoto teksto simbolis bus raidė a ?

Atsakymas būtų aiškus, jeigu žinotume koks buvo tryliktas šifro simbolis. Tačiau kaip išspręsti šį uždavinį to nežinant? Atsakymą padės rasti *pilnosios tikimybės formulė*:

$$P(A) = \sum_{i \in I} P(H_i) P(A|H_i)\,, \tag{1.2}$$

čia $H_i,\ (i \in I)$ - baigtinė arba skaiti poromis nesuderinamų įvykių šeima, tenkinanti sąlygą

$$P(\bigcup_{i \in I} H_i) = 1.$$

Pilnosios tikimybės formulė teigia, kad *apriorinę* įvykio A tikimybę galima rasti, žinant *aposteriorines* (sąlygines) A tikimybes, esant sąlygoms H_i, ir tų sąlygų susidarymo tikimybes. Esant toms pačioms prielaidoms kaip ir pilnosios tikimybės formulėje, galime rasti ir hipotezių aposteriorines tikimybes $P(H_j|A)$:

$$P(H_j|A) = \frac{P(H_j)P(A|H_j)}{\sum\limits_{i \in I} P(H_i)P(A|H_i)}\,. \tag{1.3}$$

(1.3) lygybė vadinama Bajeso hipotezių tikrinimo formule. Ja galime remtis tokioje sprendimų priėmimo situacijoje. Tarkime, žinome, jog įvyko vienas įvykis iš poromis nesuderinamų įvykių šeimos H_i $(i \in I)$ (teisinga viena iš kelių hipotezių) Kuris iš įvykių įvyko - nežinome, tačiau turime "netiesioginę" informaciją: įvyko įvykis A. Tarkime, reikia nuspręsti, kuria hipoteze H_i vadovautis, priimant sprendimą apie tolimesnius veiksmus. Mažiausia tikimybė suklysti bus tada, jei savo sprendimą grįsime ta hipoteze, kuriai $P(H_i|A)$ yra didžiausia.

1.1.5 pavyzdys. Išspręsime suformuluotą uždavinį apie iškraipytą šifrą. Kadangi viskas priklauso nuo to koks buvo neiškraipyto šifro tryliktas simbolis, tai atitinkamai ir parinksime hipotezes H_1, H_2, H_3 :

H_1={tryliktas siunčiamo šifro simbolis buvo raidė a}\,, $\quad P(H_1) = \frac{1}{2}$;

H_2={tryliktas siunčiamo šifro simbolis buvo raidė b}\,, $\quad P(H_2) = \frac{1}{3}$;

H_3={tryliktas siunčiamo šifro simbolis buvo raidė c}\,, $\quad P(H_3) = \frac{1}{6}$;

Tegul A=\{tryliktas gauto šifro simbolis yra raidė a\}. Tuomet, pagal uždavinio sąlygas,

$$P(A|H_1) = 1\,, \quad P(A|H_2) = 0,1\,, \quad P(A|H_3) = 0,05\,.$$

Pritaikę pilnosios tikimybės formulę, gauname

$$\begin{aligned} P(A) &= P(H_1)P(A|H_1) + P(H_2)P(A|H_2) + P(H_3)P(A|H_3) \\ &= \frac{1}{2}\cdot 1 + \frac{1}{3}\cdot\frac{1}{10} + \frac{1}{6}\cdot\frac{1}{20} = \frac{13}{24}\,. \end{aligned}$$

Galime formuluoti ir kitą, dažnai žymiai aktualesnį, klausimą. Tarkime, kad vienaip ar kitaip adresatas sugebėjo perskaityti tą nelemtą tryliktąjį gauto šifro simbolį - tai buvo raidė a. Kokia tikimybė, kad ji nėra iškraipyta? Kitaip sakant, mus dominanti tikimybė yra $P(H_1|A)$. Ją rasime, pasinaudoję Bajeso formule. Pastebėsime, kad trupmenos vardiklis (1.3) lygybėje, pagal pilnosios tikimybės formulę, yra lygus $P(A)$. Todėl

$$P(H_1|A) = \frac{P(H_1)P(A|H_1)}{P(A)} = \frac{\frac{1}{2}\cdot 1}{\frac{13}{24}} = \frac{12}{13}\,.$$

Bernulio eksperimentai. Bernulio eksperimentų schema nusakoma taip: eksperimentą atlikus vieną kartą, jo sėkmės tikimybė lygi p. Atliekame n nepriklausomų eksperimentų. Sėkmių skaičių pažymėkime S_n. Kokia tikimybė, kad eksperimentas pavyks k kartų, t.y. $S_n = k$? Atsakymas į šį klausimą toks:

$$P(S_n = k) = \binom{n}{k} p^k (1-p)^{n-k}\,, \quad k = 0, 1, \ldots, n. \tag{1.4}$$

Bernulio schema yra vienodų ir nepriklausomų statistinių eksperimentų matematinis modelis. Ją naudojant skaičiuojamos tikimybės, susijusios su nepriklausomų vienodų bandymų seka, kai kiekviename bandyme galimos tik dvi baigtys.

1.1.6 pavyzdys. Informacija perduodama triukšmingu kanalu, kuris vidutiniškai iškraipo 1% visų siunčiamų bitų. Kokia tikimybė, kad baite bus ne daugiau dviejų iškraipytų bitų? Šiuo atveju sėkmė - gauti iškraipytą bitą. Pagal sąlygą tokios "sėkmės" tikimybė kiekvienu atveju yra $p = 0,01$. Mums reikalinga tikimybė, kad "sėkmių" skaičius po 8 bandymų būtų ne didesnis už 2. Pasinaudoję (1.4) lygybe, gausime

$$\begin{aligned} P(S_8 \le 2) &= P(S_8 = 0) + P(S_8 = 1) + P(S_8 = 2) \\ &= \binom{8}{0} 0,01^0 0,99^8 + \binom{8}{1} 0,01^1 0,99^7 + \binom{8}{2} 0,01^2 0,99^6 \approx 0,999946 \end{aligned}$$

Atsitiktiniai dydžiai. Apibrėždami atsitiktinius įvykius, kalbėjome apie eksperimentus ir jų elementariąsias baigtis. Praktiškai beveik visada susiduriame su skaitiniais stebimojo dydžio matavimais, t.y. su kokio nors atsitiktinio dydžio reikšmėmis.

1.1.4 apibrėžimas. *Tarkime, kad* (Ω, P) *yra diskrečioji tikimybinė erdvė. Atsitiktiniu dydžiu šioje erdvėje vadinama realioji funkcija* $X : \Omega \to \mathbb{R}$.

Taigi atsitiktinis dydis nusako taisyklę, pagal kurią kiekvienam elementariajam įvykiui priskiriama skaitinė reikšmė. Diskrečiosios tikimybinės erdvės atsitiktinių dydžių reikšmių aibė yra baigtinė arba skaiti. Tokie atsitiktiniai dydžiai X vadinami *diskrečiaisiais* ir dažniausiai nusakomi *reikšmių skirstiniu*, nurodant galimas reikšmes x_i ir jų tikimybes

$$p_i = P(X = x_i) = \sum_{\omega \in \Omega \,:\, X(\omega) = x_i} P(\omega)\,, \quad i = 1, 2, \ldots\,.$$

Pavyzdžiui sėkmių skaičius S_n, atlikus n Bernulio eksperimentų, yra diskretus atsitiktinis dydis, kurio reikšmių aibė $\{0, 1, 2, \ldots, n\}$, o tikimybės nusakomos (1.4) formule. Tokį skirstinį turintis atsitiktinis dydis vadinamas *binominiu* ir žymimas $S_n \sim \mathcal{B}(n, p)$.
Kai elementariųjų įvykių aibė Ω nėra skaiti, atsitiktinio dydžio reikšmių aibė gali būti labai "gausi" ir net nesunumeruojama. Pavyzdžiui, matuojant kliento aptarnavimo laiką, priklausomai nuo matavimo vienetų, rezultatas gali būti bet kuris tam tikro intervalo taškas. Šiuo atveju prasminga kalbėti ne apie pavienių reikšmių tikimybes, bet apie reikšmių priklausymo nurodytam intervalui tikimybę.

1.1.5 apibrėžimas. *Atsitiktinis dydis* X*, kurio patekimo į intervalą* $[a, b]$ *tikimybė skaičiuojama pagal formulę*

$$P(a \le X \le b) = \int_a^b p(x)\,dx\,, \quad p(x) \ge 0\,,$$

vadinamas absoliučiai tolydžiuoju dydžiu, o funkcija $p(x)$ *vadinama jo tankiu.*

Pastebėsime, kad bet kokiam absoliučiai tolydžiam atsitiktiniam dydžiui X ir realiajam skaičiui a "taškinė" tikimybė $P(X = a)$ lygi 0, o tankio funkcija $p(x)$ tenkina sąlygą

$$\int_{-\infty}^{\infty} p(x)\,dx = 1\,.$$

Kita vertus, pasirodo, kad bet kuri neneigiama funkcija funkcija $p(x)$, tenkinanti pastarąją lygybę, gali būti laikoma kažkokio atsitiktinio dydžio tankiu.

Dažnai atsitiktiniai dydžiai (tiek diskretieji, tiek tolydieji) nusakomi specialia - *pasiskirstymo funkcija.* Atsitiktinio dydžio X pasiskirstymo funkcija $F(x)$ yra

$$F(x) = P(X < x)\,, \quad x \in \mathbb{R}\,.$$

Absoliučiai tolydžiojo atsitiktinio dydžio pasiskirstymo funkciją ir tankį sieja lygybės:

$$\begin{aligned} F'(x) &= p(x)\,, \\ F(x) &= \int\limits_{-\infty}^{x} p(u)\,du\,. \end{aligned}$$

Kai nagrinėjami keli atsitiktiniai dydžiai, pasidaro svarbi jų tarpusavio priklausomybė. Natūralu pavadinti atsitiktinius dydžius X_1 ir X_2 nepriklausomais, kai su bet kuriais realiųjų skaičių poaibiais B_1, B_2 įvykiai $\{\omega : X_1(\omega) \in B_1\}$ ir $\{\omega : X_2(\omega) \in B_2\}$ yra nepriklausomi, kitaip sakant, jei

$$P(X_1 \in B_1,\, X_2 \in B_2) = P(X_1 \in B_1)P(X_2 \in B_2)\,.$$

Diskrečiųjų atsitiktinių dydžių atveju pakanka pareikalauti, kad visiems realiesiems x, y būtų tenkinama lygybė

$$P(X_1 = x,\, X_2 = y) = P(X_1 = x)P(X_2 = y)\,.$$

Priminsime kai kurias atsitiktinių dydžių skaitines charakteristikas. Pradėsime nuo vidurkio, nusakančio vidutinę atsitiktinio dydžio reikšmę. Diskrečiojo atsitiktinio dydžio skirstinį patogiausia užrašyti lentele

X	x_1	x_2	x_3	$\ldots$
P	p_1	p_2	p_3	$\ldots$

Žinoma, $p_1 + p_2 + \cdots = 1\,, \quad p_i \geq 0$. Taip nusakyto atsitiktinio dydžio *vidurkiu* vadinama suma

$$\mathbf{E}X = x_1p_1 + x_2p_2 + x_3p_3 + \ldots\,.$$

Jeigu X turi tankį $p(x)$, tai vidurkis apibrėžiamas kaip integralas

$$\mathbf{E}X = \int_{-\infty}^{\infty} xp(x)\,dx\,.$$

Galima apibrėžti ir atsitiktinio dydžio funkcijos vidurkį. Dikrečiojo ir tolydžiojo dydžių atvejais funkcijos vidurkis atitinkamai yra

$$\mathbf{E}f(X) = f(x_1)p_1 + f(x_2)p_2 + f(x_3)p_3 + \dots\,, \quad \mathbf{E}f(X) = \int_{-\infty}^{\infty} f(x)p(x)\,dx\,.$$

Suformuluosime pagrindines vidurkių savybes. Tegu $X, X_1, X_2, \dots, X_n$ yra atsitiktiniai dydžiai, turintys baigtinius vidurkius.

1. Su bet kokiomis konstantomis $c_1, c_2, \dots, c_n$ teisinga lygybė

$$\mathbf{E}\Big(\sum_{i=1}^{n} c_i X_i\Big) = \sum_{i=1}^{n} c_i \mathbf{E}X_i\,.$$

2. Jei $X_1 \le X_2$, tai $\mathbf{E}X_1 \le \mathbf{E}X_2$.
3. $|\mathbf{E}X| \le \mathbf{E}|X|$.
4. Jei X_1, X_2 yra nepriklausomi, tai

$$\mathbf{E}(X_1 \cdot X_2) = \mathbf{E}(X_1) \cdot \mathbf{E}(X_2)\,.$$

Kaip jau buvo minėta, vidurkis parodo vidutinę atsitiktinio dydžio X reikšmę. Jo sklaidą apie vidurkį aprašo *dispersija*

$$\mathbf{D}X = \mathbf{E}(X - \mathbf{E}X)^2\,.$$

Kvadratinė šaknis iš dispersijos vadinama *standartiniu nuokrypiu* $\sigma(X) = \sqrt{\mathbf{D}X}$.
Paminėsime keletą dispersijos savybių, laikydami, kad atsitiktiniai dydžiai
$X, X_1, X_2, \dots, X_n$ turi baigtines dispersijas.

1. $\mathbf{D}X \ge 0\,.$
2. $\mathbf{D}X = \mathbf{E}X^2 - (\mathbf{E}X)^2\,.$
3. $\mathbf{D}(X_1 + X_2) = \mathbf{D}X_1 + \mathbf{D}X_2 + 2cov(X_1, X_2)\,,$ čia

$$cov(X_1, X_2) = \mathbf{E}(X_1 - \mathbf{E}X_1)(X_2 - \mathbf{E}X_2) = \mathbf{E}X_1X_2 - \mathbf{E}X_1\mathbf{E}X_2$$

yra atsitiktinių dydžių X_1 ir X_2 *kovariacija*.

4. Jei $X_1, X_2, \ldots, X_n$ yra nepriklausomi atsitiktiniai dydžiai, tai su bet kokiomis konstantomis $c_1, c_2, \ldots, c_n$ teisinga lygybė

$$\mathbf{D}\Big(\sum_{i=1}^{n} c_i X_i\Big) = \sum_{i=1}^{n} c_i^2 \mathbf{D} X_i \,.$$

1.1.7 pavyzdys. Rasime binominio skirstinio, nusakyto (1.4) tikimybėmis, vidurkį ir dispersiją. Tegul $X_i = 1$, jei i-tasis Bernulio eksperimentas buvo sėkmingas, ir $X_i = 0$ - priešingu atveju. Tada sėkmių skaičius po n eksperimentų bus n nepriklausomų, vienodai pasiskirsčiusių atsitiktinių dydžių suma

$$S_n = X_1 + X_2 + \cdots + X_n \,.$$

Be to, visiems $i = 1, 2, \ldots, n$ atsitiktinio dydžio X_i skirstinys yra

X_i	0	1
P	$1-p$	p

Todėl

$$\begin{aligned} \mathbf{E}X_i &= \mathbf{E}X_i^2 = p \,, \\ \mathbf{D}X_i &= \mathbf{E}X_i^2 - (\mathbf{E}X_i)^2 = p(1-p) \,. \end{aligned}$$

Dabar jau nesunkiai randame nepriklausomų atsitiktinių dydžių X_i sumos vidurkį ir dispersiją

$$\begin{aligned} \mathbf{E}S_n &= \sum_{i=1}^{n} \mathbf{E}X_i = np \,, \\ \mathbf{D}S_n &= \sum_{i=1}^{n} \mathbf{D}X_i = np(1-p) \,. \end{aligned}$$

1.2 Įvykių sistemos

Tarkime $\mathcal{A} = \{A_i : i \in I\}$ yra diskrečiosios tikimybinės erdvės (Ω, P) įvykių šeima, I - baigtinė arba skaiti indeksų aibė.

1.2.1 apibrėžimas. *Jei* $P(A_i \cap A_j) = 0$ *visiems* $i, j \in I$, $i \neq j$ *ir*

$$P(\bigcup_{i \in I} A_i) = 1\,,$$

tai $\mathcal{A}$ *vadinsime tikimybinės erdvės* (Ω, P) *įvykių sistema.*

Pastebėsime, kad poromis nesuderinamiems įvykiams A_i

$$P(\bigcup_{i \in I} A_i) = \sum_{i \in I} P(A_i)\,.$$

Todėl antrojoje apibrėžimo sąlygoje sąjungos tikimybę pakeitę atitinkamų tikimybių suma, gautume ekvivalentišką įvykių sistemos apibrėžimą. Taip pat aišku, kad jei A_i poromis nesutaikomi ir

$$\bigcup_{i \in I} A_i = \Omega\,,$$

tai $\mathcal{A}$ - įvykių sistema. Atvirkščias teiginys teisingas tik, kai tikimybinėje erdvėje nėra nulinės tikimybės elementariųjų įvykių, t.y. $P(\omega) > 0$ visiems $\omega \in \Omega$.

1.2.2 apibrėžimas. *Tarkime* $\mathcal{A} = \{A_i : i \in I\}$ *ir* $\mathcal{B} = \{B_j : j \in J\}$ *yra tikimybinės erdvės* (Ω, P) *įvykių sistemos.* $\mathcal{A}$ *yra sistemos* $\mathcal{B}$ *apvalkalas, jei kiekvienam* $j \in J$ *galima rasti tokį* $i \in I$ *, kad* $P(A_i \cap B_j) = P(B_j)$*. Tokiu atveju sakysime, kad sistema* $\mathcal{B}$ *yra tikslesnė už sistemą* $\mathcal{A}$*.*

Kitaip sakant, kiekvienai tikslesnės sistemos aibei visada atsiras ją dengianti "grubesnės" sistemos aibė.

1.2.1 teorema. *Jei* $\mathcal{B}$ *yra tikslesnė už sistemą* $\mathcal{A}$*, tai visiems* $i \in I,\ j \in J$

$$P(A_i \cap B_j) = P(B_j) \quad \textit{arba} \quad P(A_i \cap B_j) = 0.$$

Įrodymas. Kai $P(B_j) = 0$, šis teiginys akivaizdus. Tarkime, kad $P(B_j) \neq 0$ ir $P(A_i \cap B_j) \neq P(B_j)$. Lieka įsitikinti, kad tokiu atveju būtinai $P(A_i \cap B_j) = 0$. Pagal apvalkalo apibrėžimą, aibėje I galima rasti tokį $i_0 \neq i$, kad

$$P(A_{i_0} \cap B_j) = P(B_j).$$

Vadinasi

$$B_j = A_{i_0} \cap B_j \cup N\,, \quad P(N) = 0.$$

Taigi

$$P(A_i \cap B_j) = P(A_i \cap (A_{i_0} \cap B_j) \cup (A_i \cap N)) \leq P(A_i \cap A_{i_0}) + P(N) = 0.$$

□

1.2.3 apibrėžimas. *Tarkime $\mathcal{A} = \{A_i : i \in I\}$ ir $\mathcal{B} = \{B_j : j \in J\}$ yra tikimybinės erdvės (Ω, P) įvykių sistemos. Jų jungtinė sistema $\mathcal{A} \wedge \mathcal{B}$ nusakoma lygybe*

$$\mathcal{A} \wedge \mathcal{B} = \{A_i \cap B_j : (i,j) \in I \times J\}.$$

Akivaizdu, kad $\mathcal{A} \wedge \mathcal{B}$ tikslesnė ir už $\mathcal{A}$ ir už $\mathcal{B}$. Bet pasirodo, kad ji yra pati "grubiausia" iš visų tikslesnių už $\mathcal{A}$ ir $\mathcal{B}$. Teisingas toks teiginys.

1.2.2 teorema. *Jei sistema $\mathcal{C}$ yra tikslesnė už $\mathcal{A}$ ir $\mathcal{B}$, tai ji tikslesnė ir už $\mathcal{A} \wedge \mathcal{B}$.*

Įrodymas. Iš tikrųjų, iš teiginio prielaidos išplaukia, kad kiekvienam $C \in \mathcal{C}$ galima rasti tokius A_i ir B_j kad

$$P(A_i \cap C) = P(B_j \cap C) = P(C).$$

Todėl

$$B_j \cap C = C \setminus N\,, \quad N \subset C\,, \quad P(N) = 0.$$

Dabar gauname

$$\begin{aligned} P(A_i \cap B_j \cap C) &= P(A_i \cap (C \setminus N)) = P((A_i \cap C) \setminus (A_i \cap N)) \\ &= P(A_i \cap C) - P(A_i \cap N) = P(C). \end{aligned}$$

Pastaroji lygybė ir įrodo, kad $\mathcal{C}$ yra tikslesnė už $\mathcal{A} \wedge \mathcal{B}$. □

1.2.4 apibrėžimas. *Tikimybinės erdvės (Ω, P) įvykių sistemos $\mathcal{A} = \{A_i : i \in I\}$ ir $\mathcal{B} = \{B_j : j \in J\}$ vadinamos nepriklausomomis, jei*

$$P(A_i \cap B_j) = P(A_i)P(B_j)$$

visiems $(i,j) \in I \times J$.

1.2.1 pavyzdys. Tarkime X ir Y diskretieji atsitiktiniai dydžiai erdvėje (Ω, P), o $\{x_i \in \mathbb{R} : i \in I\}$ ir $\{y_j \in \mathbb{R} : j \in J\}$ - jų galimų reikšmių aibės. Apibrėžkime įvykius $A_i = \{X = x_i\}$ ir $B_j = \{Y = y_j\}$. Nesunku pastebėti, kad $\mathcal{A} = \{A_i : i \in I\}$ ir $\mathcal{B} = \{B_j : j \in J\}$ yra tikimybinės erdvės (Ω, P) įvykių sistemos. Jos bus nepriklausomos tada ir tik tada, kai nepriklausomi yra jas generuojantys atsitiktiniai dydžiai X ir Y.

1.3 Informacija ir entropija

Tarkime A yra tikimybinės erdvės (Ω, P) atsitiktinis įvykis, kurio tikimybė $P(A) = p$. Kiek informacijos gauname, įvykus šiam įvykiui ? Nekalbėsime apie gautos informacijos prasmę ar naudą. Mūsų tikslas - apibrėžti kiekybinį informacijos matą, priklausantį nuo įvykio tikimybės p. Įykio A prigimtis, skaičiuojant *informacijos kiekį* $I(A)$, nėra svarbi. Todėl informacijos kiekį apibrėšime kaip kintamojo p funkciją ir dažnai rašysime $I(A) = I(p)$. Jai kelsime tokius reikalavimus :

1. Informacija turi būti apibrėžta ir neneigiama, t.y. $I(p) \geq 0$, visiems $p \in (0,\ 1]$.

2. Nežymiai pakitus įvykio tikimybei, informacijos kiekis taip pat turėtų pasikeisti nedaug. Kitaip sakant funkcija $I(p)$ turi būti tolydi.

3. Funkcija $I(p)$ turi griežtai monotoniškai mažėjanti, t.y. kuo įvykio tikimybė mažesnė, tuo didesnį informacijos kiekį jam įvykus gauname. Jei pastarasis reikalavimas pasirodė keistas, panagrinėkite du atsitiktinius įvykius: A_1 ={ateinančių metų liepos septintą dieną Vilniuje snigs} ir A_2 ={ateinančių metų liepos septintoji Vilniuje bus saulėta}. Kurio įvykio tikimybė didesnė ir, kuriam įvykus, daugiau sužinotumėte apie Lietuvos klimato pokyčius?!

4. Įvykus dviem nepriklausomiems įvykiams, gautos informacijos kiekis turėtų būti lygus jų informacijų sumai. Prisiminę, kad dviems nepriklausomiems įvykiams A ir B tikimybė įvykti kartu yra $P(A \cap B) = P(A)P(B)$, turėsime tokį reikalavimą informacijos kiekio funkcijai: $I(p \cdot q) = I(p) + I(q)$ visiems $p, q \in (0,\ 1]$.

Pasirodo šie, iš pirmo žvilgsnio, paprasti reikalavimai vienareikšmiškai nusako juos tenkinančią funkciją.

1.3.1 teorema. *Funkcija $I(p)$ tenkina 1-4 sąlygas tada ir tik tada, kai egzistuoja $b > 1$, jog*

$$I(p) = \log_b \frac{1}{p}\,.$$

Įrodymas. Logaritminė funkcija aišku tenkina minėtus reikalavimus. Belieka įsitikinti, kad 1-4 sąlygas tenkinanti funkcija $I(p)$ yra būtinai logaritminė.

Tegul m ir n bet kokie natūralieji skaičiai. Iš 4 sąlygos išplaukia, kad

$$I(p^n) = I(p \cdot p^{n-1}) = I(p) + I(p^{n-1}) = I(p) + I(p) + I(p^{n-2}) = \cdots = nI(p)\,.$$

Todėl

$$I(p) = I((p^{1/m})^m) = mI(p^{1/m})$$

ir

$$I(p^{1/m}) = \frac{1}{m} I(p)\,.$$

Taigi, visiems teigiamiems racionaliesiems skaičiams $\frac{n}{m}$

$$I(p^{n/m}) = I((p^{1/m})^n) = \frac{n}{m} I(p)\,.$$

Dėl funkcijos $I(p)$ tolydumo iš čia išplaukia, kad

$$I(p^a) = aI(p)$$

visiems realiesiems $a \geq 0$. Todėl visiems $p \in (0,\ 1]$

$$I(p) = I\left(\left(\frac{1}{\mathrm{e}}\right)^{-\ln p}\right) = -I\left(\frac{1}{\mathrm{e}}\right) \ln p\,. \tag{1.5}$$

Kadangi funkcija $I(p)$ yra griežtai mažėjanti ir neneigiama, tai $I\left(\frac{1}{\mathrm{e}}\right) > 0$ ir galima rasti $b > 1$, kad

$$I\left(\frac{1}{\mathrm{e}}\right) = \frac{1}{\ln b}\,.$$

Iš čia ir (1.5) galutinai gauname

$$I(p) = -\frac{\ln p}{\ln b} = \log_b \frac{1}{p}\,.$$

□

Dabar jau galime apibrėžti įvykio informaciją. Logaritmo pagrindo pasirinkimas apsprendžia tik jos matavimo vienetus. Laikysime, kad informacijos vienetą gauname, įvykus įvykiui, kurio tikimybė $\frac{1}{2}$. Tada $b = 2$.

1.3.1 apibrėžimas. *Informacijos kiekiu, gaunamu įvykus įvykiui A, kurio tikimybė $p > 0$, vadinsime dydį*

$$I(A) = I(p) = \log_2 \frac{1}{p},$$

o jo matavimo vienetus - bitais.

Kartais naudojami ir kiti informacijos kiekio vienetai.

Logaritmo pagrindas (b)	*Informacijos kiekio vienetas*
2	bitas
3	tritas
e	natas
10	hartlis

Sąryšiai tarp šių matavimo vienetų nusakomi lygybėmis

$$1\ bitas\ = \log_3 2\ trito\ = \ln 2\ nato\ = \lg 2\ hartlio\,.$$

Beje, čia bitas ne atsitiktinai sutampa su dvejetainio skaičiaus skaitmens pavadinimu.

1.3.1 pavyzdys. Metame simetrišką monetą. Eksperimento baigčių S={atsivertė skaičius} ir H={atsivertė skaičius} tikimybės yra $P(S) = P(H) = 0,5$. Todėl bet kurios baigties atveju gaunamas $I(S) = I(H) = \log_2 2 = 1$ bitas informacijos. Jei moneta metama n kartų, tai bet kurią eksperimento baigtį galime nusakyti n dvejetainių skaitmenų, pavyzdžiui

$$\underbrace{011101110\ldots01101}_{n}$$

Čia 0 ir 1 žymi įvykius S ir H. Tokios baigties tikimybė yra 2^{-n}, o gaunamas informacijos kiekis

$$I\left(\frac{1}{2^n}\right) = \log_2 2^n = n\,,$$

t.y. lygiai tiek, kiek bitų užima informacija apie eksperimento rezultatą.

Pastebėsime, kad būtino įvykio informacija $I(\Omega) = 0$. Kitaip sakant, kai įvyksta tai "kas ir turėjo įvykti", mes nieko naujo nesužinome. Tačiau, jei įvyktų tai "kas įvykti negali", turėtume "labai daug" naujos informacijos. Toks pastebėjimas pateisina informacijos kiekio apibrėžimo papildymą nulinės tikimybės įvykiams. Taigi, jei $P(A) = 0$, tai

$$I(A) = \lim_{p \to 0} I(p) = \infty \, .$$

Galima apibrėžti ir dviejų įvykių sąlyginę informaciją.

1.3.2 apibrėžimas. *Tegul A ir B yra tikimybinės erdvės (Ω, P) atsitiktiniai įvykiai ir $P(B) > 0$. Įvykio A su sąlyga B informacija vadinsime dydį*

$$I(A|B) = \log_2 \frac{1}{P(A|B)} = -\log_2 \frac{P(A \cap B)}{P(B)} \, .$$

Pastebėsime, kad $I(A|B) = I(A)$ tada ir tik tada, kai įvykiai A ir B yra nepriklausomi. Aptarėme pavienio atsitiktinio įvykio informacijos kiekio sąvoką. Dabar pabandysime nusakyti įvykių sistemos informaciją. Pradėsime nuo pavyzdžio.

1.3.2 pavyzdys. Tegul galimos bandymo baigtys yra $A_1, A_2, \ldots A_n$, o jų tikimybės atitinkamai $p_1, p_2, \ldots p_n$. Kiek informacijos gausime atlikę tokį bandymą? Norėdami atsakyti į šį klausimą, galime samprotauti taip. Atlikus N tokių nepriklausomų bandymų, baigtis A_i pasikartos apytiksliai $N \cdot p_i$ kartų ir kiekvieną kartą gaunamos informacijos kiekis bus

$$I(A_i) = \log_2 \frac{1}{p_i} \, .$$

Taigi po visos bandymų serijos sukauptas informacijos kiekis bus

$$I_N \approx \sum_{i=1}^{n} N p_i \log_2 \frac{1}{p_i} \, .$$

Todėl vidutinis informacijos kiekis, gaunamas atlikus vieną bandymą, yra

$$\frac{I_N}{N} \approx \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i} \, .$$

Pastebėsime, kad dešinėje šios apytikslės lygybės pusėje esantis reiškinys yra lygus vidutinei įvykių $A_1, A_2, \ldots A_n$ informacijai.

1.3.3 apibrėžimas. *Tegul* $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ *yra tikimybinės erdvės* (Ω, P) *įvykių su tikimybėmis* $p_i = P(A_i)$ *sistema. Įvykių sistemos* $\mathcal{A}$ *entropija vadinsime dydį*

$$H(\mathcal{A}) = H(p_1, p_2, \ldots, p_n) = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i}\,.$$

Čia ir analogiškose sumose toliau visi dėmenys $p_i \log_2 \frac{1}{p_i}$ arba $p_i \log_2 p_i$ yra lygūs 0 , kai $p_i = 0$. Skaitytojams, kuriems toks susitarimas atrodo įtartinas, priminsime, kad

$$\lim_{p\to 0} p \log_2 \frac{1}{p} = \lim_{p\to 0} p \log_2 p = 0\,.$$

Aiškinantis įvairius sąryšius, kartais patogu interpretuoti entropiją kaip dydį, reiškiantį neapibrėžtumą, kurį jaučiame, nežinodami kuris iš sistemos $\mathcal{A}$ įvykių įvyks. Panagrinėkime dviejų įvykių su tikimybėmis p ir $1-p$ sistemą. Jos entropiją žymėsime $h(p) = H(p, 1-p)$. Taigi binarinės entropijos funkcija

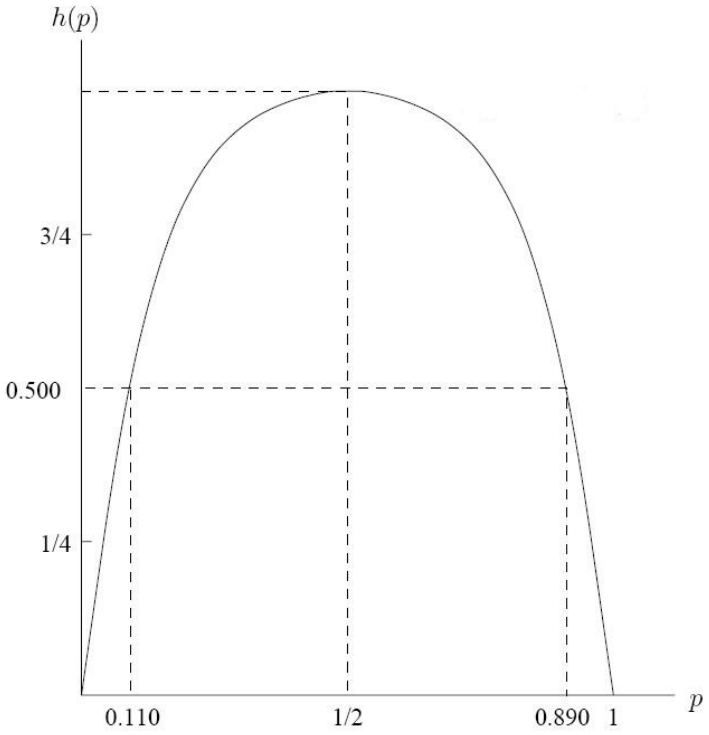
$$h(p) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}\,. \tag{1.6}$$

Šios funkcijos grafikas pavaizduotas 1.1 paveiksle. Matome, kad didžiausia entropijos reikšmė lygi $h\left(\frac{1}{2}\right) = 1$. Tai ir rodo, kad sunkiausiai prognozuojama dviejų įvykių sistema yra ta, kurioje abu įvykiai yra vienodai galimi, t.y. $p = \frac{1}{2}$. Ir priešingai - kai $p = 0$ arba $p = 1$, jokio neapibrėžtumo nėra, nes iš anksto aišku kuris iš dviejų įvykių įvyks. Tad nenuostabu, kad šiuo atveju entropija lygi $h(0) = h(1) = 0$.

Įvykus kokiam nors įvykiui B, sistemos $\mathcal{A}$ entropija gali pasikeisti. Pavyzdžiui, analizuodami prekybos centro duomenų bazės įrašus, pagal pirkėjo krepšelį bandome nuspėti pirkėjo amžių. Jei jis pirko tik duonos, tai aišku, nelabai ką tegalime pasakyti apie jo amžių. Bet, kai tarp jo pirkinių atrandame alų ir skrudintą duoną, neapibrėžtumas pirkėjo amžiaus atžvilgiu ženkliai sumažėja. Likusio neapibrėžtumo laipsnį nusako sąlyginė entropija

$$H(\mathcal{A}|B) = \sum_{i=1}^{n} P(A_i|B) I(A_i|B) = \sum_{i=1}^{n} P(A_i|B) \log_2 \frac{1}{P(A_i|B)}\,. \tag{1.7}$$

Tokių entropijų vidutinė reikšmė leidžia įvertinti neapibrėžtumą sistemos $\mathcal{A}$ atžvilgiu, kuris lieka, gavus informaciją apie kitą tos pačios tikimybinės erdvės įvykių sistemą $\mathcal{B} = \{B_1, B_2, \ldots, B_m\}$.



1.1 pav. Binarinės entropijos funkcija

1.3.4 apibrėžimas. *Dydį*

$$H(\mathcal{A}|\mathcal{B}) = \sum_{j=1}^{m} P(B_j) H(\mathcal{A}|B_j)$$

vadinsime sąlygine $\mathcal{A}$ entropija $\mathcal{B}$ atžvilgiu, o entropijos pokytį

$$I(\mathcal{A}, \mathcal{B}) = H(\mathcal{A}) - H(\mathcal{A}|\mathcal{B})$$

vadinsime sistemų $\mathcal{A}$ ir $\mathcal{B}$ tarpusavio informacija.

Sistemų $\mathcal{A}$ ir $\mathcal{B}$ jungtinės sistemos $\mathcal{A} \wedge \mathcal{B}$ entropiją žymėsime $H(\mathcal{A}, \mathcal{B})$. Prisiminę 1.2.3 apibrėžimą, turėsime

$$H(\mathcal{A}, \mathcal{B}) = H(\mathcal{A} \wedge \mathcal{B}) = \sum_{i=1}^{n} \sum_{j=1}^{m} P(A_i \cap B_j) \log_2 \frac{1}{P(A_i \cap B_j)} \,. \tag{1.8}$$

Šis apibrėžimas akivaizdžiai apibendrinamas ir didesniam įvykių sistemų skaičiui $k \geq 2$

$$H(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k) = H(\mathcal{A}_1 \wedge \mathcal{A}_2 \wedge \cdots \wedge \mathcal{A}_k) \,.$$

Nagrinėjant entropijų savybes ir sąryšius, mums pravers vienas pagalbinis teiginys. Vektorių $(x_1, x_2, \ldots, x_n)$, sudarytą iš neneigiamų realiųjų skaičių, vadinsime tikimybiniu vektoriumi (kitaip: diskrečiuoju skirstiniu), jei

$$\sum_{i=1}^{n} x_i = 1\,.$$

1.3.2 lema (Gibbs'o nelygybė)**.** *Tegul $b > 1$. Tada bet kokiems tikimybiniams vektoriams $(x_1, x_2, \ldots, x_n)$ ir $(y_1, y_2, \ldots, y_n)$ teisinga nelygybė*

$$\sum_{i=1}^{n} x_i \log_b \left(\frac{y_i}{x_i}\right) \leq 0\,. \tag{1.9}$$

Nelygybė virsta lygybe tada ir tik tada, kai vektoriai $(x_1, x_2, \ldots, x_n)$ ir $(y_1, y_2, \ldots, y_n)$ sutampa.

Įrodymas. Kaip jau buvo minėta, kai $x_i = 0$, tai i - tasis nagrinėjamos sumos dėmuo taip pat lygus 0. Be to, pastebėsime, kad lemos teiginys yra teisingas, jei kuriam nors i, $x_i > 0$ ir $y_i = 0$, nes tada $x_i \log_b \left(\frac{y_i}{x_i}\right) = -\infty$. Todėl, nesiaurindami bendrumo, galime nagrinėti tik tokius vektorius, kuriems $y_i > 0$, jeigu $x_i > 0$. Taigi mums lieka įrodyti lemos teiginį, nelygybę (1.9) užrašius šitaip:

$$\sum_{i=1}^{n}{}^{*} x_i \log_b \left(\frac{y_i}{x_i}\right) \leq 0\,.$$

Čia simbolis * prie sumos ženklo reiškia, kad sumuojama tik pagal tas i reikšmes, kurioms $x_i > 0$. Vadinasi, ir visi y_i šioje sumoje yra teigiami. Padauginę pastarosios nelygybės abi puses iš teigiamo skaičiaus $\ln b$, pereisime prie natūraliųjų logaritmų

$$\sum_{i=1}^{n}{}^{*} x_i \ln \left(\frac{y_i}{x_i}\right) \leq 0\,. \tag{1.10}$$

Kadangi funkcija $y = \ln x$ yra iškila aukštyn, o jos grafiko liestinė taške $x = 1$ yra tiesė $y = x - 1$, tai

$$\ln x \leq x - 1$$

visiems $x > 0$. Be to, nelygybė virsta lygybe tik, kai $x = 1$. Iš čia išplaukia

$$\sum_{i=1}^{n}{}^{*} x_i \ln \left(\frac{y_i}{x_i}\right) \leq \sum_{i=1}^{n}{}^{*} x_i \left(\frac{y_i}{x_i} - 1\right) = \sum_{i=1}^{n}{}^{*} y_i - \sum_{i=1}^{n}{}^{*} x_i \leq 0\,.$$

Pastebėsime, kad abi pastarosios nelygybės, turi virsti lygybėmis, jei (1.10) suma lygi 0. Tačiau taip gali atsitikti tik, kai $x_i = y_i$ visiems $i = 1, 2, \ldots, n$. Lema įrodyta. □

Aptarsime pagrindines entropijos savybes. Pirmiausiai išsiaiškinsime kokios sistemos turi didžiausias ir kokios mažiausias entropijas.

1.3.3 teorema. *Įvykių sistemos* $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ *entropija tenkina nelygybes*

$$0 \leq H(\mathcal{A}) \leq \log_2 n\,. \tag{1.11}$$

Mažiausią reikšmę $H(\mathcal{A}) = 0$ *ji įgyja tada ir tik tada, kai sistema sudaryta iš įvykių, kurių tikimybės yra* 0 *arba* 1. *Didžiausią entropiją* $H(\mathcal{A}) = \log_2 n$ *turės tos ir tik tos sistemos, kuriose visi įvykiai yra vienodai galimi, t.y.* $P(A_i) = \frac{1}{n}$ *visiems* $i = 1, 2, \ldots, n$.

Įrodymas. Pagal apibrėžimą entropija yra neneigiamų dėmenų suma

$$H(\mathcal{A}) = \sum_{i=1}^{n} P(A_i) \log_2 \frac{1}{P(A_i)}\,.$$

Todėl aišku, kad $H(\mathcal{A}) \geq 0$. Be to, tokia suma lygi 0 tada ir tik tada, kai visi dėmenys lygūs 0. Vadinasi, $P(A_i) = 0$ arba $P(A_i) = 1$ visiems $i = 1, 2, \ldots, n$. Iš tikrųjų tik viena iš šių tikimybių bus lygi 1, nes sistemą sudarančių įvykių tikimybių suma visada yra 1.

Antrąją nelygybę įrodysime skirtumui $H(\mathcal{A}) - \log_2 n$ pritaikę 1.3.2 lemą. Gausime

$$\begin{aligned} H(\mathcal{A}) - \log_2 n &= \sum_{i=1}^{n} P(A_i) \log_2 \frac{1}{P(A_i)} - \sum_{i=1}^{n} P(A_i) \log_2 n \\ &= \sum_{i=1}^{n} P(A_i) \log_2 \left(\frac{1/n}{P(A_i)} \right) \leq 0\,. \end{aligned}$$

Pastaroji nelygybė išplaukia iš (1.9) nelygybės, pasirinkus $x_i = P(A_i)$ ir $y_i = 1/n$. Iš čia pagal 1.3.2 lemą gauname ir paskutinį teoremos teiginį apie didžiausią entropiją. □

Pastaba. Įvykių sistemos entropija nepasikeistų iš sistemos pašalinus nulinės tikimybės įvykius (jei tokių yra). Todėl (1.11) nelygybėje n galima pakeisti teigiamą tikimybę turinčių sistemos $\mathcal{A}$ įvykių skaičiumi.

Atrodo, kad kuo sudėtingesnė ir daugiau įvykių turi sistema, tuo didesnė jos entropija. Tačiau tiesioginės priklausomybės tarp įvykių skaičiaus sistemoje ir jos entropijos dydžio, aišku, nėra.

1.3.3 pavyzdys. Tegul $\mathcal{A} = \{A_1, A_2, A_3, A_4\}$, $\mathcal{B} = \{B_1, B_2, B_3, B_4, B_5\}$ ir

$$\begin{aligned} P(A_1) &= P(A_2) = P(A_3) = P(A_4) = \frac{1}{4}; \\ P(B_1) &= P(B_2) = P(B_3) = P(B_4) = \frac{1}{16}, \quad P(B_5) = \frac{3}{4}. \end{aligned}$$

Apskaičiuojame sistemų $\mathcal{A}$ ir $\mathcal{B}$ entropijas

$$\begin{aligned} H(\mathcal{A}) &= 4 \cdot \frac{1}{4} \cdot \log_2 4; \\ H(\mathcal{B}) &= 4 \cdot \frac{1}{16} \cdot \log_2 16 + \frac{3}{4} \cdot \log_2 \frac{4}{3} \approx 1,3113. \end{aligned}$$

Kaip matome, $H(\mathcal{A}) > H(\mathcal{B})$. Gautąją nelygybę galime interpretuoti taip: numatyti, kuris iš įvykių įvyks, sistemoje $\mathcal{A}$ yra sunkiau, nei sistemoje $\mathcal{B}$.

Kai kurioms įvykių sistemų klasėms jų entropijų santykis visada nusakomas vienareikšmiškai. Pavyzdžiui, galime palyginti sistemos ir jos apvalkalo entropijas.

1.3.4 teorema. *Jei įvykių sistema* $\mathcal{B} = \{B_1, B_2, \ldots, B_m\}$ *yra tikslesnė už sistemą* $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$, *tai*

$$H(\mathcal{A}) \leq H(\mathcal{B}).$$

Įrodymas. Kadangi $\mathcal{B}$ yra tikslesnė už $\mathcal{A}$, tai pagal 1.2.1 teoremą

$$P(A_i) = \sum_{j=1}^{m} P(A_i \cap B_j) = \sum_{j \in J_i} P(B_j), \quad i = 1, 2, \ldots, n.$$

Čia J_i - poromis nesikertantys aibės $\{1, 2, \ldots, m\}$ poaibiai, tenkinantys sąlygą

$$\bigcup_{i=1}^{n} J_i = \{1, 2, \ldots, m\}.$$

Todėl

$$\begin{aligned} H(\mathcal{A}) &= -\sum_{i=1}^{n} P(A_i) \log_2 P(A_i) = -\sum_{i=1}^{n} \left(\sum_{j \in J_i} P(B_j) \right) \log_2 \left(\sum_{j \in J_i} P(B_j) \right) \\ &\leq -\sum_{i=1}^{n} \sum_{j \in J_i} P(B_j) \log_2 P(B_j) = -\sum_{j=1}^{m} P(B_j) \log_2 P(B_j) = H(\mathcal{B}) \end{aligned}$$

□

Prisiminkime sąlyginės entropijos $H(\mathcal{A}|B)$ apibrėžimą (1.7). Klausimas: ar, įvykus kokiam nors įvykiui B, sistemos $\mathcal{A}$ entropija sumažėja, kitaip sakant, ar galima tvirtinti, kad visada $H(\mathcal{A}|B) \leq H(\mathcal{A})$? Neigiamą atsakymą į šį klausimą pagrindžia toks pavyzdys.

1.3.4 pavyzdys. Tegul X ir Y yra atsitiktiniai dydžiai, įgyjantys reikšmes 0 ir 1, o jų bendrasis dvimatis skirstinys nusakytas lentele

$X\backslash Y$	0	1	$P(X=i)$
0	0,25	0,25	0,5
1	0	0,5	0,5
$P(Y=j)$	0,25	0,75	1

Nagrinėsime dvi įvykių sistemas

$$\mathcal{A} = \{\ \{X=0\},\ \{X=1\}\ \} \quad \text{ir} \quad \mathcal{B} = \{\ \{Y=0\},\ \{Y=1\}\ \}\,.$$

Kaip įprasta tokiais atvejais, sistemų $\mathcal{A}$ ir $\mathcal{B}$ entropijas žymėsime tiesiog $H(X)$ ir $H(Y)$. Prisiminę binarinės entropijos funkcijos $h(p)$ apibrėžimą (1.6), turėsime

$$\begin{aligned} H(X) &= h(0,5) = 1\,, \\ H(Y) &= h(0,25) \approx 0,811\,. \end{aligned}$$

Apskaičiuojame sąlygines tikimybes

$$\begin{aligned} P(Y=0|X=0) &= \frac{P(Y=0,\ X=0)}{P(X=0)} = \frac{0,25}{0,5} = 0,5\,, \\ P(Y=1|X=0) &= 1 - P(Y=0|X=0) = 0,5\,. \end{aligned}$$

Analogiškai gauname, kad

$$P(Y=0|X=1) = 0 \quad \text{ir} \quad P(Y=1|X=1) = 1\,.$$

Todėl, pagal (1.7) formulę, sąlyginės Y entropijos, kai žinomos atsitiktinio dydžio X reikšmės, bus

$$\begin{aligned} H(Y|X=0) &= h(0,5) = 1\,, \\ H(Y|X=1) &= h(0) = 0\,. \end{aligned}$$

Vidutinė šių entropijų reikšmė pagal 1.3.4 apibrėžimą yra

$$\begin{aligned} H(Y|X) &= P(X=0)\cdot H(Y|X=0) + P(X=1)\cdot H(Y|X=1) \\ &= \frac{1}{2}\cdot 1 + \frac{1}{2}\cdot 0 = \frac{1}{2}\,. \end{aligned}$$

Matome, kad $H(Y|X=1) < H(Y) < H(Y|X=0)$, tačiau $H(Y|X) < H(Y)$.

Šiame pavyzdyje vidutinė sąlyginė entropija $H(Y|X)$ nusako likusį (sumažėjusį) neapibrėžtumą Y atžvilgiu po to, kai gauta informacija apie atsitiktinį dydį X. Pasirodo toks sumažėjimas nėra atsitiktinis.

1.3.5 teorema. *Visoms įvykių sistemoms $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ ir $\mathcal{B} = \{B_1, B_2, \ldots, B_m\}$ jų tarpusavio informacija yra neneigiama:*

$$I(\mathcal{A}, \mathcal{B}) \geq 0\,.$$

Be to, $I(\mathcal{A}, \mathcal{B}) = 0$ tada ir tik tada, kai sistemos $\mathcal{A}$ ir $\mathcal{B}$ yra nepriklausomos.

Įrodymas. Kadangi įvykiai su nulinėmis tikimybėmis neturi įtakos $I(\mathcal{A}, \mathcal{B})$ reikšmei, tai tarsime, kad $P(A_i)\cdot P(B_j) > 0$ visiems $i = 1, 2, \ldots, n;\quad j = 1, 2, \ldots, m$.
Pagal apibrėžimą

$$\begin{aligned} H(\mathcal{A}|\mathcal{B}) &= \sum_{j=1}^{m} P(B_j) H(\mathcal{A}|B_j) = \sum_{j=1}^{m}\sum_{i=1}^{n} P(B_j)P(A_i|B_j)\log_2 \frac{1}{P(A_i|B_j)} \\ &= \sum_{j=1}^{m}\sum_{i=1}^{n} P(A_i \cap B_j)\log_2\left(\frac{P(B_j)}{P(A_i\cap B_j)}\right)\,. \end{aligned} \tag{1.12}$$

Iš įvykių sistemos apibrėžimo išplaukia, kad visiems $i = 1, 2, \ldots, n$

$$P(A_i) = \sum_{j=1}^{m} P(A_i \cap B_j)\,.$$

Todėl

$$H(\mathcal{A}) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i\cap B_j)\log_2\frac{1}{P(A_i)}\,.$$

Prisiminę 1.3.4 apibrėžimą ir įstatę gautas išraiškas, turėsime

$$-I(\mathcal{A}, \mathcal{B}) = H(\mathcal{A}|\mathcal{B}) - H(\mathcal{A}) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(A_i\cap B_j)\log_2\left(\frac{P(A_i)P(B_j)}{P(A_i\cap B_j)}\right) \leq 0\,. \tag{1.13}$$

Pastaroji nelygybė išplaukia iš 1.3.2 lemos, pritaikius ją tikimybiniams vektoriams $(x_1, x_2, \ldots, x_{mn})$ ir $(y_1, y_2, \ldots, y_{mn})$, kurių komponentės yra

$$x_{(i-1)m+j} = P(A_i \cap B_j) \quad \text{ir} \quad y_{(i-1)m+j} = P(A_i)P(B_j)\,, \quad i = 1, 2, \ldots, n\,, \quad j = 1, 2, \ldots, m\,.$$

Pagal tą pačią lemą gauname, kad (1.13) nelygybė virsta lygybe tada ir tik tada, kai visiems i ir j

$$P(A_i \cap B_j) = P(A_i)P(B_j)\,,$$

kitaip sakant, kai sistemos $\mathcal{A}$ ir $\mathcal{B}$ yra nepriklausomos. Teorema įrodyta. □

Iš ką tik įrodytos teoremos išplaukia, kad bet kokioms įvykių sistemoms $\mathcal{A}$ ir $\mathcal{B}$

$$H(\mathcal{A}|\mathcal{B}) \leq H(\mathcal{A})\,,$$

o šių entropijų lygybė yra sistemų $\mathcal{A}$ ir $\mathcal{B}$ nepriklausomumo kriterijus.
Jungtinės sistemos entropijos $H(\mathcal{A}, \mathcal{B})$ reikšmė priklauso ne tik nuo ją sudarančių sistemų, bet ir nuo jų priklausomumo laipsnio.

1.3.6 teorema. *Įvykių sistemoms $\mathcal{A}$ ir $\mathcal{B}$ teisingi sąryšiai*

$$H(\mathcal{A}, \mathcal{B}) = H(\mathcal{B}) + H(\mathcal{A}|\mathcal{B})\,, \tag{1.14}$$

$$H(\mathcal{A}, \mathcal{B}) = H(\mathcal{A}) + H(\mathcal{B}) - I(\mathcal{A}, \mathcal{B})\,. \tag{1.15}$$

Įrodymas. Pasiremsime kai kuriais 1.3.5 teoremos įrodymo tarpiniais rezultatais. Kadangi visiems $j = 1, 2, \ldots, m$

$$P(B_j) = \sum_{i=1}^{n} P(A_i \cap B_j)\,,$$

tai lygybę (1.12) galime parašyti taip

$$\begin{aligned} H(\mathcal{A}|\mathcal{B}) &= \sum_{j=1}^{m}\sum_{i=1}^{n} P(A_i \cap B_j) \log_2 \left(\frac{1}{P(A_i \cap B_j)}\right) - \sum_{j=1}^{m} \left(\sum_{i=1}^{n} P(A_i \cap B_j)\right) \log_2 \left(\frac{1}{P(B_j)}\right) \\ &= H(\mathcal{A}, \mathcal{B}) - H(\mathcal{B})\,. \end{aligned}$$

Iš čia gauname (1.14) lygybę.

Pagal tarpusavio informacijos apibrėžimą 1.3.4

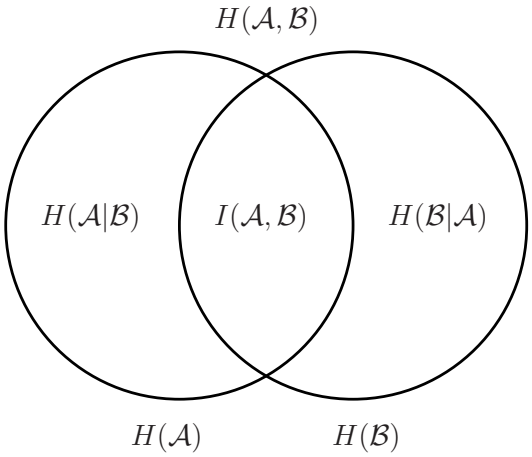
$$H(\mathcal{A}|\mathcal{B}) = H(\mathcal{A}) - I(\mathcal{A}, \mathcal{B})\,.$$

Įstatę šią sąlyginės entropijos išraišką į (1.14), gauname (1.15) lygybę ir tuo pačiu baigiame teoremos įrodymą. □

Pastebėsime, kad $H(\mathcal{A},\mathcal{B}) = H(\mathcal{B},\mathcal{A})$. Todėl iš ką tik įrodytos (1.15) lygybės išplaukia, kad simetriška yra ir tarpusavio informacija, t.y.

$$I(\mathcal{A},\mathcal{B}) = I(\mathcal{B},\mathcal{A}).$$

Dviejų sistemų entropiją, sąlyginių entropijų bei tarpusavio informacjos sąryšius patogu vaizduoti 1.2 paveiksle pateikiama diagrama.



1.2 pav. Entropija ir informacija

Iš 1.3.5 teoremos ir (1.15) lygybės gauname tokį jungtinės sistemos entropijos įvertį

$$H(\mathcal{A},\mathcal{B}) \leq H(\mathcal{A}) + H(\mathcal{B}).$$

Pritaikius indukciją, pastarąją nelygybę nesudėtinga apibendrinti didesniam įvykių sistemų skaičiui.

1.3.7 teorema. *Jei $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$ yra tos pačios diskrečiosios tikimybinės erdvės įvykių sistemos, tai*

$$H(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k) \leq H(\mathcal{A}_1) + H(\mathcal{A}_2) + \cdots + H(\mathcal{A}_k).$$

Ši nelygybė virsta lygybe tada ir tik tada, kai $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$ yra nepriklausomos sistemos.

1.4 Atsitiktinių dydžių entropijos

Su diskrečiojo atsitiktinio dydžio entropijos sąvoka mes jau buvome susidūrę, nagrinėdami 1.3.4 pavyzdį. Tad visiškai suprantamas bus toks apibrėžimas.

1.4.1 apibrėžimas. *Diskrečiojo atsitiktinio dydžio X, įgyjančio reikšmes $x_1, x_2, \dots$, entropija $H(X)$ yra lygi įvykių sistemos, sudarytos iš įvykių $A_i = \{X = x_i\}$, $i = 1, 2, \dots$, entropijai, t.y.,*

$$H(X) = \sum_i P(A_i) \log_2 \frac{1}{P(A_i)} .$$

Todėl diskrečiojo atsitiktinio dydžio entropijos interpretacija bei savybės niekuo nesiskiria nuo 1.3 skyrelyje aptartos įvykių sistemos entropijos. Pateiksime keletą iš galimų, su entropijos sąvoka susijusių, uždavinių.

1.4.1 pavyzdys. Tegul T yra koks nors tekstas. Pavyzdžiui,

$$T = abrakadabra$$

Kaip matome, tekste sutinkamos penkios skirtingos raidės. Tad, norėdami parašyti dvejetainį T kodą, kiekvienai raidei turėsime skirti po tris bitus. Pavyzdžiui,

$$a \mapsto 000,\ b \mapsto 001,\ d \mapsto 010,\ k \mapsto 011,\ r \mapsto 100 . \tag{1.16}$$

Vadinasi, viso teksto kodo ilgis bus 33 bitai. Ar įmanoma sukonstruoti trumpesnį vienareikšmiškai dekoduojamą kodą?

Galime manyti, kad tekstas T yra sudarytas iš 11 atsitiktinio dydžio X realizacijų. Kitaip sakant, atsitiktinis dydis X reiškia atsitiktinai pasirinktą T raidę. Pagal raidžių dažnius tekste sudarome atsitiktinio dydžio X skirstinį

X	a	b	d	k	r
P	$\frac{5}{11}$	$\frac{2}{11}$	$\frac{1}{11}$	$\frac{1}{11}$	$\frac{2}{11}$

Apskaičiavę X entropiją (kartais ji dar vadinama teksto T entropija), sužinosime vidutinį informacijos kiekį, tenkantį vienai teksto raidei. Taigi

$$H(X) = \frac{5}{11} \log_2 \frac{11}{5} + 2 \cdot \frac{2}{11} \log_2 \frac{11}{2} + 2 \cdot \frac{1}{11} \log_2 11 \approx 2,0404 .$$

Tai yra vadinamasis *Šenono rėžis* vidutiniam vienos raidės kodo bitų skaičiui. Kitaip sakant, nėra tokio vienareikšmiškai dekoduojamo kodo, kuriuo pakeitus (1.16) kodą, tekstą T atvaizduotume trumpesne nei $11 \cdot H(X) \approx 22,44$ bitų seka. Tačiau ne visiems tekstams Šenono rėžis yra pasiekiamas. Tie skaitytojai, kurie šiek tiek žino duomenų kompresijos algoritmus, supras kaip konstruojamas tekstui T geriausią rezultatą duodantis kodas (beje ne vienintelis):

$$a \mapsto 0,\ b \mapsto 100,\ d \mapsto 110,\ k \mapsto 111,\ r \mapsto 101\,.$$

Jis tekstą T užkoduoja 23 bitų seka.

- Jei norite įsitikinti, kad Šenono rėžis yra pasiekiamas, pabandykite užkoduoti tekstą $T' = abrakadabrakaada$.

1.4.2 pavyzdys. Atliekamas tyrimas, siekiant nustatyti kokie faktoriai įtakoja galimybę susirgti tam tikra liga. Ligos požymį žymėsime kintamuoju Y, įgyjančiu reikšmes S ir N (serga, neserga). Nagrinėjami trys faktoriai:

X_1 – paciento lytis, galimos reikšmės $\{M, V\}$;

X_2 – rūkymas, galimos reikšmės $\{taip, ne\}$;

X_3 – kraujospūdis, galimos reikšmės $\{mažas, normalus, didelis\}$.

100 pacientų tyrimo duomenys pateikti 1.1 lentelėje. Kuris faktorius labiausiai įtakoja polinkį susirgti? Norėdami atsakyti į šį klausimą, turime išsiaiškinti, kaip pasikeičia Y entropija, vieno ar kito požymio atžvilgiu. Kitaip sakant, turime palyginti tris tarpusavio informacijas

$$I(Y, X_i) = H(Y) - H(Y|X_i)\,, \quad i = 1, 2, 3.$$

Pirmiausiai rasime $H(Y)$. Atsitiktinio dydžio Y skirstinys yra

Y	N	S
P	$0,44$	$0,56$

Vadinasi,

$$H(Y) = h(0,44) \approx 0,989587521$$

Paciento lytis (X_1)	*Rūkymas* (X_2)	*Kraujospūdis* (X_3)	*Ligos požymis* (Y)	Pacientų skaičius
M	*ne*	*mažas*	N	5
M	*ne*	*mažas*	S	2
M	*ne*	*normalus*	N	10
M	*ne*	*didelis*	S	6
M	*taip*	*mažas*	N	4
M	*taip*	*mažas*	S	2
M	*taip*	*normalus*	N	8
M	*taip*	*didelis*	N	1
M	*taip*	*didelis*	S	8
V	*ne*	*normalus*	N	8
V	*ne*	*didelis*	S	10
V	*ne*	*mažas*	N	2
V	*taip*	*mažas*	S	7
V	*taip*	*normalus*	N	6
V	*taip*	*normalus*	S	5
V	*taip*	*didelis*	S	16

1.1 lentelė. Ligą įtakojantys faktoriai

Skaičiuosime $H(Y|X_1)$. Pasinaudoję (1.7) formule, pagal 1.1 lentelės duomenis gausime

$$\begin{aligned} H(Y|X_1=M) &= P(Y=N|X_1=M)\log_2\frac{1}{P(Y=N|X_1=M)} \\ &+ P(Y=S|X_1=M)\log_2\frac{1}{P(Y=S|X_1=M)} \\ &= h\left(\frac{28}{46}\right)\approx 0,965636133 \end{aligned}$$

Analogiškai

$$H(Y|X_1=V)=h\left(\frac{16}{54}\right)\approx 0,876716289$$

Dabar, prisiminę 1.3.4 apibrėžimą, randame $H(Y|X_1)$

$$\begin{aligned} H(Y|X_1) &= P(X_1=M)H(Y|X_1=M)+P(X_1=V)H(Y|X_1=V) \\ &= 0,46\cdot h\left(\frac{28}{46}\right)+0,54\cdot h\left(\frac{16}{54}\right)\approx 0,917619417 \end{aligned}$$

Analogiškai skaičiuojame ir likusias entropijas

$$H(Y|X_2 = ne) = h\left(\frac{25}{43}\right) \approx 0,980798365;$$
$$H(Y|X_2 = taip) = h\left(\frac{19}{57}\right) \approx 0,918295834;$$
$$H(Y|X_2) = 0,43 \cdot h\left(\frac{25}{43}\right) + 0,57 \cdot h\left(\frac{19}{57}\right) \approx 0,945171922;$$
$$H(Y|X_3 = mažas) = h\left(\frac{11}{22}\right) = 1;$$
$$H(Y|X_3 = normalus) = h\left(\frac{32}{37}\right) \approx 0,571354974;$$
$$H(Y|X_3 = didelis) = h\left(\frac{1}{41}\right) \approx 0,165427034;$$
$$H(Y|X_3) = 0,22 \cdot 1 + 0,37 \cdot h\left(\frac{32}{37}\right) + 0,41 \cdot h\left(\frac{1}{41}\right) \approx 0,499226424;$$

Dabar jau galime rasti ieškomąsias tarpusavio informacijas

$$I(Y, X_1) \approx 0,071968104\,,$$
$$I(Y, X_2) \approx 0,044415599\,,$$
$$I(Y, X_3) \approx 0,490361097\,.$$

Kaip matome, diagnozuojant ligą, labiausiai informatyvus yra kraujospūdžio dydis. Tuo tarpu paciento lytis ir jo įprotis rūkyti galimybę susirgti įtakoja nežymiai.

Pastaba. 1.1 lentelėje pateikti duomenys yra fiktyvūs ir skirti tik aptariamų sąvokų iliustracijai. Todėl suformuluotos išvados jokiu būdu nereiškia, kad rūkymas nekenkia sveikatai!

Tolydžiųjų atsitiktinių dydžių entropija apibrėžiama kitaip. Skiriasi ir jos interpretacija.

1.4.2 apibrėžimas. *Tolydžiojo atsitiktinio dydžio X su tankio funkcija $p_X(x)$ entropija $H(X)$ yra lygi*

$$H(X) = -\int_{-\infty}^{\infty} p_X(x) \log_2 p_X(x)\, dx\,.$$

Iš karto reikia pažymėti, kad toldžiojo atsitiktinio dydžio entropijos negalima interpretuoti kaip vidutinės informacijos, nes šiuo atveju $P(X = x) = 0$, nepriklausomai nuo tankio

funkcijos $p_X(x)$ reikšmės. Be to, toldžiojo atsitiktinio dydžio entropija gali būti ir neigiama. Panagrinėkime tokį pavyzdį.

1.4.3 pavyzdys. Tegul X yra normalusis (kitaip: Gauso) atsitiktinis dydis su vidurkiu a ir dispersija σ^2, $\sigma > 0$. Tokio atsitiktinio dydžio tankio funkcija yra

$$p_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-a)^2}{2\sigma^2}\right\}.$$

Rasime jo entropiją. Pagal apibrėžimą

$$\begin{aligned} H(X) &= \int_{-\infty}^{\infty} p_X(x) \left(\frac{(x-a)^2}{2\sigma^2}(\log_2 \mathrm{e}) + \log_2(\sqrt{2\pi}\sigma)\right) dx \\ &= \frac{\log_2 \mathrm{e}}{2\sigma^2} \int_{-\infty}^{\infty} (x-a)^2 p_X(x)\, dx + \log_2(\sqrt{2\pi}\sigma) \int_{-\infty}^{\infty} p_X(x)\, dx \\ &= \frac{\log_2 \mathrm{e}}{2\sigma^2} \cdot \sigma^2 + \log_2(\sqrt{2\pi}\sigma) \cdot 1 = \frac{1}{2}\log_2(2\mathrm{e}\pi\sigma^2). \end{aligned}$$

Matome, kad normaliojo atsitiktinio dydžio entropija priklauso tik nuo jo dispersijos σ^2 ir yra neigiama, kai $\sigma^2 < \frac{1}{2\mathrm{e}\pi}$.

Įrodysime dar vieną įdomią normaliojo atsitiktinio dydžio savybę. Tuo tikslu prisiminkime (1.9) nelygybę. Panašiai įrodoma ir jos "tolydžioji" versija

$$\int_{-\infty}^{\infty} p_X(x) \log_b\left(\frac{p_Y(x)}{p_X(x)}\right) dx \le 0. \tag{1.17}$$

Čia $b > 1$, o $p_X(x)$ ir $p_Y(x)$ - bet kokios tankio funkcijos. Pasinaudoję šia nelygybe, įrodysime tokį teiginį.

1.4.1 teorema. *Jei absoliučiai tolydus atsitiktinis dydis X turi baigtinę dispersiją $\mathbf{D}X = \sigma^2$, $\sigma > 0$, tai jo entropija*

$$H(X) \le \frac{1}{2}\log_2(2\mathrm{e}\pi\sigma^2).$$

Įrodymas. Atsitiktinio dydžio X vidurkį pažymėsime $\mathbf{E}X = a$. Tegul Y yra normalusis atsitiktinis dydis, turintis tokius pat vidurkį ir dispersiją, kaip ir atsitiktinis dydis X. Tada jo tankio funkcijos logaritmas

$$\log_2 p_Y(x) = -\frac{(x-a)^2}{2\sigma^2}(\log_2 \mathrm{e}) - \log_2(\sqrt{2\pi}\sigma). \tag{1.18}$$

Atsitiktiniams dydžiams X ir Y pritaikysime (1.17) nelygybę. Pasirinkę $b = 2$, ją galime parašyti taip

$$-\int_{-\infty}^{\infty} p_X(x) \log_2 p_X(x)\, dx \le -\int_{-\infty}^{\infty} p_X(x) \log_2 p_Y(x)\, dx\,.$$

Kairėje pastarosios nelygybės pusėje esantis reiškinys, pagal apibrėžimą, yra atsitiktinio dydžio X entropija $H(X)$. Vadinasi,

$$H(X) \le -\int_{-\infty}^{\infty} p_X(x) \log_2 p_Y(x)\, dx\,.$$

Įrašę $\log_2 p_Y(x)$ išraišką (1.18), gausime

$$\begin{aligned} H(X) &\le \int_{-\infty}^{\infty} p_X(x) \left(\frac{(x-a)^2}{2\sigma^2} (\log_2 \mathrm{e}) + \log_2(\sqrt{2\pi}\sigma) \right) dx \\ &= \frac{\log_2 \mathrm{e}}{2\sigma^2} \int_{-\infty}^{\infty} (x-a)^2 p_X(x)\, dx + \log_2(\sqrt{2\pi}\sigma) \int_{-\infty}^{\infty} p_X(x)\, dx \\ &= \frac{\log_2 \mathrm{e}}{2\sigma^2} \cdot \mathbf{D}X + \log_2(\sqrt{2\pi}\sigma) \cdot 1 = \frac{1}{2} \log_2(2\mathrm{e}\pi\sigma^2)\,. \end{aligned}$$

□

Iš 1.4.1 teoremos išplaukia, kad tarp visų absoliučiai tolydžių atsitiktinių dydžių, turinčių baigtinę dispersiją σ^2, didžiausią entropiją $\frac{1}{2}\log_2(2\mathrm{e}\pi\sigma^2)$ turi normalusis atsitiktinis dydis.

2 Pagrindiniai duomenų tyrimo uždaviniai ir sąvokos

Pradinę pažintį su duomenų tyrimo pagrindinėmis sąvokomis ir sprendžiamais uždaviniais pradėsime nuo paprasčiausių pavyzdžių.

2.1 Pirmieji pavyzdžiai

Norėdami neformaliai supažindinti su pradine duomenų tyrimo terminologija, pateiksime keletą paprastų duomenų rinkinių. Jie mums pravers ir vėliau nagrinėjamų metodų iliustracijai.

2.1.1 Duomenys apie orą

Stebimos oro sąlygos, kurioms esant "Žvaigždžių" komanda sutinka žaisti parodomąsias rungtynes. 2.1 lentelėje pateikiamas duomenų rinkinys (*imtis*), sudarytas iš 14 *įrašų*. Kaip matome, kiekvieną įrašą sudaro penkios *kintamųjų* arba *atributų* reikšmės, nusakančios tam tikras įrašo charakteristikas. Šiuo atveju pirmųjų keturių kintamųjų *Oras, Temperatūra, Drėgnumas, Vėjuota* reišmės įtakoja kintamojo *Žaisti* įgyjamą reikšmę. Pačios kintamųjų reikšmės čia labiau atspindi kokybinius (kitaip: *kategorinius*) esamos meteorologinės situacijos parametrus, o ne kiekybinius. Pavyzdžiui *Drėgnumas* gali būti *didelis* arba *normalus*, *Temperatūra* nusakoma "pagal savijautą" - *karšta, šilta, vėsu.*
Dabar pagal turimus duomenis pabandykime sukonstruoti *taisykles*, kurios leistų nuspėti "Žvaigždžių" elgesį, esant bet kokioms oro sąlygoms. Kitaip sakant, reikia *"išmokti"* apskaičiuoti kintamojo *Žaisti* reikšmę, priklausomai nuo likusiųjų kintamųjų reikšmių. Taisyklės galėtų būti tokios

`Jei` *Oras=saulėta* `ir` *Drėgnumas=didelis* `tai` *Žaisti=ne*
`Jei` *Oras=lietinga* `ir` *Vėjuota=TRUE* `tai` *Žaisti=ne*
`Jei` *Oras=debesuota* `tai` *Žaisti=taip*
`Jei` *Drėgnumas=normalus* `tai` *Žaisti=taip*
`Kitais atvejais` *Žaisti=taip*

Pastebėsime, kad šios taisyklės teisingai klasifikuoja visus įrašus, jei taikomos tokia tvarka

	Oras	*Temperatūra*	*Drėgnumas*	*Vėjuota*	*Žaisti*
1	saulėta	karšta	didelis	FALSE	ne
2	saulėta	karšta	didelis	TRUE	ne
3	debesuota	karšta	didelis	FALSE	taip
4	lietinga	šilta	didelis	FALSE	taip
5	lietinga	vėsu	normalus	FALSE	taip
6	lietinga	vėsu	normalus	TRUE	ne
7	debesuota	vėsu	normalus	TRUE	taip
8	saulėta	šilta	didelis	FALSE	ne
9	saulėta	vėsu	normalus	FALSE	taip
10	lietinga	šilta	normalus	FALSE	taip
11	saulėta	šilta	normalus	TRUE	taip
12	debesuota	šilta	didelis	TRUE	taip
13	debesuota	karšta	normalus	FALSE	taip
14	lietinga	šilta	didelis	TRUE	ne

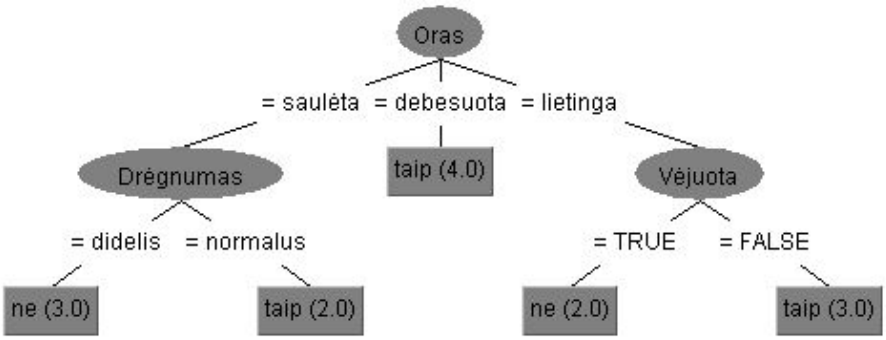
2.1 lentelė. Duomenys apie orą

kaip užrašytos. Tas pačias taisykles kita tvarka pritaikius, galima gauti kitokį rezultatą. Pavyzdžiui, atskirai paimta tasyklė

`Jei` *Drėgnumas=normalus* `tai` *Žaisti=taip*

ne visada bus teisinga.

Dažnai turimą imtį patogiau klasifikuoti ne pagal taisyklių seką, o žymiai vaizdesne priemone - *sprendimų medžiu*. Siekdami suprasti kada "Žvaigždės" sutinka žaisti, o kada ne, pagal 2.1 lentelės duomenis sudarykime 2.1 paveiksle pavaizduotą sprendimų medį. Pirmiausiai tikrinama kintamojo *Oras* reikšmė. Pirmosios trys medžio šakos atitinka tris galimas reikšmes. Jei *Oras* = *debesuota*, "Žvaigždės" žaidžia. Tai atspindi medžio lapas pirmame lygyje. Priešingu atveju, tikrinamos kintamųjų *Drėgnumas* arba *Vėjuota* reikšmės, o galutinius sprendimus matome pavaizduotus antro lygio lapuose. Lapuose parašyti skaičiai rodo kiek imties įrašų tenkina visas sąlygas nuo šaknies iki atitinkamo lapo. Kaip matome, visi 14



2.1 pav. Sprendimų medis duomenims apie orą

įrašų klasifikuojami teisingai. Tam net neprireikė informacijos apie kintamojo *Temperatūra* reikšmę.

Patikslinkime mūsų meteorologinius stebėjimus. Kintamuosius *(Oras, Temperatūra, Drėgnumas, Vėjuota, Žaisti)* pažymėkime (X_1, X_2, X_3, X_4, Y) ir išmatuokime oro temperatūrą bei drėgnumą. Rezultatai pateikti 2.2 lentelėje.

Dabar kintamieji X_2 ir X_3 jau bus *kiekybiniai* (arba *skaitiniai*) , nes jų reikšmės nusako temperatūrą laipsniais Celsijaus skalėje ir drėgnumą procentais. Kokybinio kintamojo Y reikšmių aibė yra $\{taip,\ ne\}$. Taisyklės, pagal kurias klasifikuojami įrašai kintamojo Y atžvilgiu, galėtų būti tokios

`Jei` $X_1 = saulėta$ `ir` $X_3 > 84$ `tai` $Y = ne$

`Jei` $X_1 = lietinga$ `ir` $X_4 = TRUE$ `tai` $Y = ne$

`Kitais atvejais` $Y = taip$

Be klasifikacijos dažnai dar nagrinėjamos ir vadinamosios *asociaciacijos* taisyklės, nusakančios galimus sąryšius tarp įvairių kintamųjų. Pavyzdžiui, pagal 2.1 lentelės duomenis galime "išmokti" tokias taisykles

`Jei` *Temperatūra=vėsu* `tai` *Drėgnumas=normalus*

	X_1	X_2	X_3	X_4	Y
1	saulėta	29	85	FALSE	ne
2	saulėta	27	90	TRUE	ne
3	debesuota	28	86	FALSE	taip
4	lietinga	21	96	FALSE	taip
5	lietinga	20	80	FALSE	taip
6	lietinga	18	70	TRUE	ne
7	debesuota	18	65	TRUE	taip
8	saulėta	22	95	FALSE	ne
9	saulėta	21	70	FALSE	taip
10	lietinga	24	80	FALSE	taip
11	saulėta	24	70	TRUE	taip
12	debesuota	22	90	TRUE	taip
13	debesuota	27	75	FALSE	taip
14	lietinga	22	91	TRUE	ne

2.2 lentelė. Patikslinti duomenys apie orą

`Jei` *Oras=saulėta* `ir` *Žaisti=ne* `tai` *Drėgnumas=didelis*

`Jei` *Vėjuota=FALSE* `ir` *Žaisti=ne* `tai` *Oras=saulėta* `ir` *Drėgnumas=didelis*

Aišku, galima sugalvoti ir daugiau (pabandykite !) asociacijos ir klasifikacijos taisyklių, kurios būtų teisingos visiems minėtų imčių įrašams. Didelėms imtims tai padaryti nėra taip paprasta. Tada konstruojamos taisyklės ar sprendimų medžiai teisingai klasifikuojantys didesniąją imties įrašų dalį.

2.1.2 Regos korekcija

Gydytojai rekomenduoja koreguoti regėjimą kontaktiniais lęšiais, tik esant tam tikroms sąlygoms. 2.3 lentelėje pateikiami duomenys apie galimybę skirti vienokius ar kitokius kontaktinius lęšius, priklausomai nuo paciento *amžiaus*, *regėjimo*, *astigmatizmo* ir *ašarų kiekio*.

	Amžius	*Regėjimas*	*Astigmatizmas*	*Ašarų kiekis*	*Lęšiai*
1	jaunas	trumparegis	ne	sumažėjęs	neskirti
2	jaunas	trumparegis	ne	normalus	minkšti
3	jaunas	trumparegis	taip	sumažėjęs	neskirti
4	jaunas	trumparegis	taip	normalus	kieti
5	jaunas	toliaregis	ne	sumažėjęs	neskirti
6	jaunas	toliaregis	ne	normalus	minkšti
7	jaunas	toliaregis	taip	sumažėjęs	neskirti
8	jaunas	toliaregis	taip	normalus	kieti
9	vidutinis	trumparegis	ne	sumažėjęs	neskirti
10	vidutinis	trumparegis	ne	normalus	minkšti
11	vidutinis	trumparegis	taip	sumažėjęs	neskirti
12	vidutinis	trumparegis	taip	normalus	kieti
13	vidutinis	toliaregis	ne	sumažėjęs	neskirti
14	vidutinis	toliaregis	ne	normalus	minkšti
15	vidutinis	toliaregis	taip	sumažėjęs	neskirti
16	vidutinis	toliaregis	taip	normalus	neskirti
17	vyresnis	trumparegis	ne	sumažėjęs	neskirti
18	vyresnis	trumparegis	ne	normalus	neskirti
19	vyresnis	trumparegis	taip	sumažėjęs	neskirti
20	vyresnis	trumparegis	taip	normalus	kieti
21	vyresnis	toliaregis	ne	sumažėjęs	neskirti
22	vyresnis	toliaregis	ne	normalus	minkšti
23	vyresnis	toliaregis	taip	sumažėjęs	neskirti
24	vyresnis	toliaregis	taip	normalus	neskirti

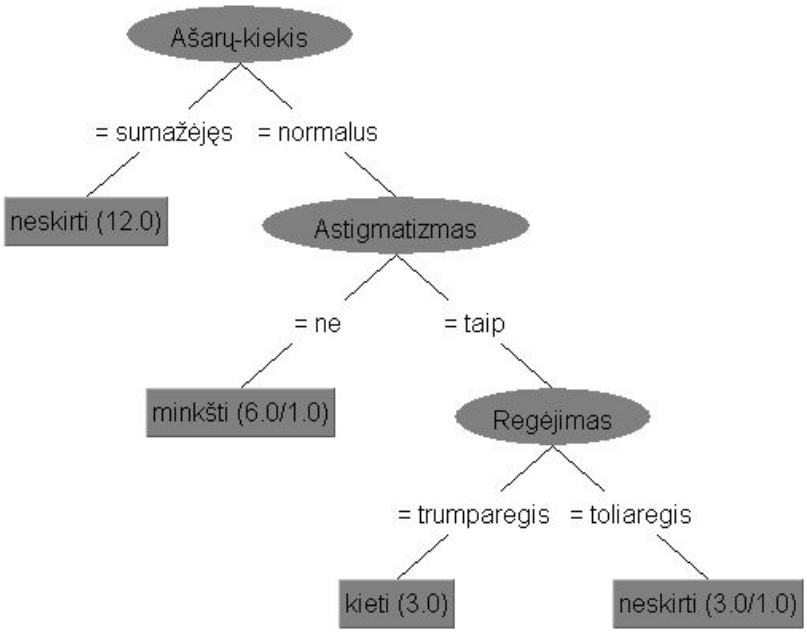
2 .3 lentelė. Duomenys apie regos korekciją

Perspėjimas: pavyzdys yra iliustracinis, todėl žemiau formuluojamų rekomendacijų nereikėtų laikyti alternatyva jūsų gydytojo nuomonei !

Iš viso yra 24 įrašai, nusakantys visas galimas šių parametrų reikšmių kombinacijas ($3 \times 2 \times 2 \times 2 = 24$). Tačiau jų pateikimo būdas vizualiai sunkiai aprėpiamas. O jeigu kintamųjų skaičius būtų didesnis ir įrašų turėtume ne 24, o tarkime 10000 ? Kitaip sakant, reikalingos kuo paprastesnės *taisyklės*, leidžiančios teisingai klasifikuoti visus ar bent jau didesnę dalį imties įrašų pagal kintamojo *Lešiai* reikšmes. Pavyzdžiui, tokia paprasta taisyklė

`Jei` *Ašarų kiekis=sumažėjęs* `tai` *Lęšiai=neskirti*

teisingai klasifikuoja 12 įrašų, bet likusieji lieka visai neklasifikuoti. Aišku galimas ir kitas kraštutinumas - parašyti taisyklių seką iš 24 sąlygų, kuri visiškai atitiks turimus duomenis. Tačiau taikyti tokią "taisyklę" būtų, ko gero, daugiau vargo, nei studijuoti pradinę lentelę. Kaip jau matėme, kita priemonė tokiam uždaviniui spręsti yra sprendimų medis. Vienas iš galimų, šio pavyzdžio duomenis atitinkantis, sprendimų medis pavaizduotas 2 .2 paveiksle.



2 .2 pav. Sprendimų medis regos korekcijos duomenims

Pastebėsime, kad jis teisingai klasifikuoja 22 įrašus iš 24. Tokių medžių konstrukcija bus nagrinėjama vėliau.

2.1.3 Irisų klasifikacija

Turbūt žymiausias ir klasikiniu tapęs klasifikavimo uždavinio pavyzdys priklauso amerikiečių statistikui R.A.Fišeriui. 1936 metais jis pateikė duomenis apie tris augalų rūšis: *Iris-setosa, Iris-versicolor, Iris-virginica.* Buvo matuojami keturi kiekvieno augalo parametrai: *taurėlapio ilgis, taurėlapio plotis, vainiklapio ilgis, vainiklapio plotis* (centimetrais). Gauta imtis, sudaryta iš 150 įrašų (po 50 kiekvienos rūšies irisų). Dalis šios imties pateikiama 2.4 lentelėje.

Skaitiniai kintamieji X_1, X_2, X_3, X_4 žymi minėtus augalo parametrus, o kokybinis kintamasis Y nusako augalo rūšį. Klausimas kaip *nepriklausomi kintamieji* X_1, X_2, X_3, X_4 įtakoja iriso rūšį, t.y. priklausomybę vienai iš trijų *klasių*, nusakomų *priklausomo kintamojo* Y reikšmėmis. Išnagrinėjus visus 150 imties įrašų, galima būtų "išmokti" , pavyzdžiui, tokias irisų klasifikavimo taisykles:

`Jei` $X_3 < 2,45$ `tai` $Y = Iris - setosa$

`Jei` $X_2 < 2,10$ `tai` $Y = Iris - versicolor$

`Jei` $X_2 < 2,45$ `ir` $X_3 < 4,55$ `tai` $Y = Iris - versicolor$

`Jei` $X_2 < 2,95$ `ir` $X_4 < 1,35$ `tai` $Y = Iris - versicolor$

`Jei` $X_3 > 2,45$ `ir` $X_3 < 4,45$ `tai` $Y = Iris - versicolor$

`Jei` $X_1 > 5,85$ `ir` $X_3 < 4,75$ `tai` $Y = Iris - versicolor$

`Jei` $X_2 < 2,55$ `ir` $X_3 < 4,95$ `ir` $X_4 < 1,55$ `tai` $Y = Iris - versicolor$

`Jei` $X_3 > 2,45$ `ir` $X_3 < 4,95$ `ir` $X_4 < 1,55$ `tai` $Y = Iris - versicolor$

`Jei` $X_1 > 6,55$ `ir` $X_3 < 5,05$ `tai` $Y = Iris - versicolor$

`Jei` $X_2 < 2,75$ `ir` $X_4 < 1,65$ `ir` $X_1 < 6,05$ `tai` $Y = Iris - versicolor$

`Jei` $X_1 > 5,85$ `ir` $X_1 < 5,95$ `ir` $X_3 < 4,85$ `tai` $Y = Iris - versicolor$

`Jei` $X_3 > 5,15$ `tai` $Y = Iris - virginica$

`Jei` $X_4 > 1,85$ `tai` $Y = Iris - virginica$

`Jei` $X_4 > 1,75$ `ir` $X_2 < 3,05$ `tai` $Y = Iris - virginica$

`Jei` $X_3 > 4,95$ `ir` $X_4 < 1,55$ `tai` $Y = Iris - virginica$

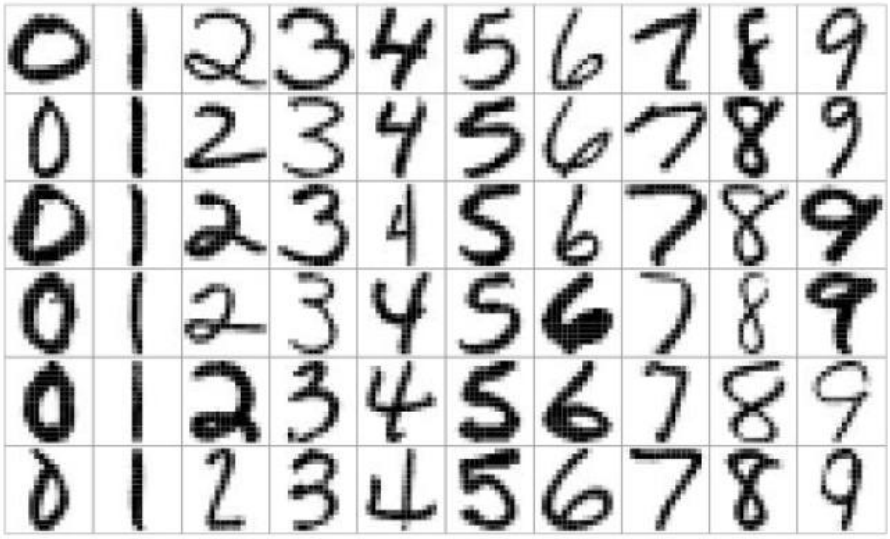
	Taurėlapio ilgis (X_1)	Taurėlapio plotis (X_2)	Vainiklapio ilgis (X_3)	Vainiklapio plotis (X_4)	Iriso rūšis (Y)
1	5,1	3,5	1,4	0,2	Iris-setosa
2	4,9	3,0	1,4	0,2	Iris-setosa
3	4,7	3,2	1,3	0,2	Iris-setosa
4	4,6	3,1	1,5	0,2	Iris-setosa
5	5,0	3,6	1,4	0,2	Iris-setosa
...	...	...	...	...	...
50	5,0	3,3	1,4	0,2	Iris-setosa
51	7,0	3,2	4,7	1,4	Iris-versicolor
52	6,4	3,2	4,5	1,5	Iris-versicolor
53	6,9	3,1	4,9	1,5	Iris-versicolor
54	5,5	2,3	4,0	1,3	Iris-versicolor
55	6,5	2,8	4,6	1,5	Iris-versicolor
...	...	...	...	...	...
100	5,7	2,8	4,1	1,3	Iris-versicolor
101	6,3	3,3	6,0	2,5	Iris-virginica
102	5,8	2,7	5,1	1,9	Iris-virginica
103	7,1	3,0	5,9	2,1	Iris-virginica
104	6,3	2,9	5,6	1,8	Iris-virginica
105	6,5	3,0	5,8	2,2	Iris-virginica
...	...	...	...	...	...
150	5,9	3,0	5,1	1,8	Iris-virginica

2 .4 lentelė. Irisų rūšys

Kaip matome, šios taisyklės yra labai komplikuotos. Vėliau nagrinėsime specialius klasifikavimo taisyklių konstravimo metodus, leidžiančius gauti tokią pat informaciją suteikiančias, tačiau žymiai kompaktiškesnes taisykles.

2.1.4 Rankraščio atpažinimas

Skanuojant ranka rašytus skaitmenis, pavyzdžiui pašto indeksus ant voko, asmens kodus dokumentuose ar anketose, dėl rankraščio ir rašymo priemonių skirtumų galimos klaidos juos atpažįstant. Tarkime, kad po tam tikrų transformacijų skaneris kiekvieną ranka rašytą skaitmenį išsaugo kaip nespalvotą 16 x 16 = 256 pikselių piešinį. Keliasdešimt tokių piešinių pavyzdžių matome 2 .3 paveiksle.



2 .3 pav. Ranka rašytų skaitmenų pavyzdžiai

Kiekvieno pikselio šviesumas kinta nuo 0 iki 255 todėl aprašomas skaitiniu kintamuoju $X_i \in \{0, 1, \ldots, 255\}$, $(i = 1, 2, \ldots, 256)$. Taigi kiekvienas imties įrašas

$$(X_1, X_2, \ldots, X_{256}, Y)$$

turės 256 nepriklausomus kintamuosius, kurių reikšmės apsprendžia 257 - ojo (priklausomo) kategorinio kintamojo Y reikšmę. Čia Y nusako klasę, kuriai priklauso visas piešinys t.y. koks skaitmuo jame pavaizduotas. Tad vėl turime spręsti klasifikavimo uždavinį ir $Y \in \{0, 1, \ldots, 9\}$. Tačiau, jei skaitmuo parašytas labai neaiškiai, reikėtų numatyti pakartotinio

skanavimo arba "rankinio" identifikavimo galimybę. Tam reikėtų klasių skaičių padidinti iki vienuolikos, leidžiant kai kuriuos įrašus priskirti klasei "*neperskaitau*". Tai sumažintų klaidos tikimybę.

2.1.5 Skaitinė prognozė

Nors irisų imtyje visi nepriklausomieji kintamieji buvo skaitiniai, tačiau klasifikavimas buvo atliekamas pagal kategorinį kintamąjį. 2.5 lentelėje pateikiama dalis duomenų apie įvairias (hipotetines) kompiuterio konfigūracijas, siekiant išsiaiškinti kaip priklauso jo našumas, išreikštas sąlyginiais vienetais, nuo įvairių sistemos parametrų.

	Takto ilgis (ns)	Pagr.atm. min (Kb)	Pagr.atm. max(Kb)	Spart. atm.(Kb)	Kanalai min	Kanalai max	Našumas
	X_1	X_2	X_3	X_4	X_5	X_6	Y
1	125	256	6000	256	16	128	199
2	29	8000	32000	32	8	32	253
3	29	8000	32000	32	8	32	253
4	29	8000	32000	32	8	32	253
5	29	8000	16000	32	8	16	132
6	26	8000	32000	64	8	32	290
7	23	16000	32000	64	16	32	381
8	23	16000	32000	64	16	32	381
9	23	16000	64000	64	16	32	749
10	23	32000	64000	128	32	64	1238
...	...	...	...	...	...	...	...
207	125	2000	8000	0	2	14	41
208	480	512	8000	32	0	0	47
209	480	1000	4000	0	0	0	25

2.5 lentelė. Kompiuterio darbo našumo duomenys

Laikydami, kad našumas (priklausomas kintamasis Y) tiesiškai priklauso nuo kintamųjų

$X_1, X_2, X_3, X_4, X_5, X_6$, galėtume gauti, pavyzdžiui, tokį kintamojo Y įvertį:

$$\hat{Y} = -66,481 + 0,066X_1 + 0,014X_2 + 0,0066X_3 + 0,494X_4 - 0,172X_5 + 1,20117X_6\,.$$

Tai yra vadinamasis *tiesinės regresijos* modelis, o gautoji lygybė kartais dar vadinama *tiesinės regresijos lygtimi*. Kaip randami šios lygties koeficientai bei kaip atrodo kitokie regresijos modeliai kalbėsime vėliau.

2.2 Duomenys ir jų atributai

Pereisime prie kiek nuoseklesnio dėstymo. Vieno populiacijos objekto stebėjimo (matavimo) rezultatas vadinamas *įrašu*. Visi įrašai sudaro *imtį*, o jų kiekis vadinamas *imties dydžiu*. Tuo atveju, kai matuojame tik vieną atributą (pvz., ūgį), įrašas turės vienintelę komponentę, o stebimasis dydis (ir pati imtis) vadinami vienmačiais. Jei mums rūpi kelios stebimojo objekto charakteristikos (pvz., lytis, ūgis ir svoris), tai įrašą sudarys p komponenčių (paminėtuoju atveju p=3), o pats stebimasis dydis (ir imtis) vadinami p-mačiais. Pavyzdžiui, Fišerio irisų imtis iš 2.1.3 skyrelio yra penkiamatė ($p = 5$), o jos dydis lygus 150.

Komponentes sudaro įvairių tipų *kintamieji*. Dažniausiai sutinkami *skaitiniai* (kitaip: kardinalieji ar kiekybiniai) kintamieji, jie dar skirstomi į tolydžiuosius (temperatūra, tūris, laikas,...) ir diskrečiuosius (vaikų šeimoje skaičius, avarijų ar klientų per dieną skaičius,...). Bet kuris iš šių tipų skirstomas dar į dvi grupes - santykinius (svoris, ūgis ir pan.; prasmę turi ne tik svorių skirtumas, bet ir santykis) ir intervalinius (kitaip: skirtuminius) (temperatūra, IQ, data; skirtumai turi prasmę, tačiau santykiai - ne). Dydžio priskyrimas vienai ar kitai grupei iš esmės priklauso nuo to, galime ar ne įvesti prasmingą nulį. Santykiniams kintamiesiams teiginys pavidalo: Jonas (60 kg) yra du kartus sunkesnis už mažąjį Augustą (30 kg), yra teisingas (nes, kokius svorio vienetus beįvestume, sąvoka "svoris lygus nuliui" reiškia tą patį), tačiau teigti, kad šiandien ($+20°C$) yra dvigubai šilčiau negu vakar ($+10°C$) yra neteisinga, nes, pvz., Farenheito skalėje šios temperatūros užrašomos kaip $+68°F$ ir $+50°F$. Panašiai yra su intelektualumo koeficientu IQ (nes psichologai nesutaria, ką reiškia IQ=0) ar su data (paprastai atskaitos pradžia (pvz., Kristaus gimimas) yra susitarimo reikalas). Antra vertus, teiginys, kad futbolo rungtynių kėlinys trunka du su puse

karto ilgiau negu krepšinio, yra teisingas (laiko momentų (ar temperatūrų) skirtumas yra santykinis kintamasis, nes sąvoka "įvykis truko 0 laiko" yra vienareikšmiškai suprantamas). Kitą dydžių klasę sudaro *kokybiniai* (kitaip: *kategoriniai*) kintamieji, kurie savo ruožtu skirstomi į *ranginius* ir *vardinius*.

Ranginiai (kitaip: tvarkos arba ordinalieji (lot. ordinatio - sutvarkymas)) kintamieji (pvz., varžybose užimta vieta, išsimokslinimas, ...). Tarkime, kad keturių komandų turnyre komandos U, V ir Z po pirmojo rato surinko atitinkamai 45, 16 ir 18 taškų.

	1-as ratas	2-as ratas	Galutinis rezultatas
U	45	28	73
	A	S	A
V	16	30	46
	B	A	S
Z	18	12	30
	S	B	B

Kadangi 0 yra natūrali vertinimo skalės pradžia, taškų skaičius yra skaitinis santykinis kintamasis. Antra vertus, sporto esmė yra kuo aukštesnė vieta, todėl komandas galima išdėstyti pagal užimtą vietą. Kitais žodžiais, U yra 1-ji, V - 3-ji, o Z - 2-ji komanda, tačiau dabar skaičiai 1, 2, ir 3 yra komandos vieta arba rangas. Tiesą sakant, tai netgi ne skaičiai, o simboliai, komandas mes galime pavadinti auksine, sidabrine arba bronzine (A, S ir B). Jei tartume, kad pirmenybėse buvo ir antrasis ratas, kuriame komandos surinko atitinkamai 28, 30 ir 12 taškų, tai aišku, kad jų taškus galima (ir reikia) sudėti, tačiau simbolių (ranginių kintamųjų) A, S ir B suma prasmės neturi.

Prisiminkime 2.1 lentelėje pateiktą oro sąlygų stebėjimo imtį. Visi penki kintamieji šioje imtyje yra kategoriniai. Tačiau tik *Temperatūra* ir *Drėgnumas* bus ranginiai. Tuo tarpu likusieji trys yra vardiniai (arba nominalieji (nominus - lot. vardas)) kintamieji, nes šiuo atveju jokio natūralaus išdėstymo "didėjimo tvarka" nėra. Kiti vardinių kintamųjų pavyzdžiai galėtų būti : akių spalva, socialinė grupė, automobilio gamintojo vardas ir t.t. Taip pat vardinis yra ir 2.1.4 skyrelyje skaitmenų atpažinimo uždavinyje nagrinėjamas kintamasis Y. Nors $Y \in \{0, 1, \ldots, 9\}$, tačiau jo negalime laikyti ranginiu, nes pavyzdžiui, tai kad $Y = 5$

šiuo atveju reiškia, kad piešinyje buvo "nupieštas" penketas. Kitaip sakant, tai yra tam tikros klasės, kuriai priklauso piešinys, kodas.

Mes susipažinome su įvairiais kintamųjų tipais. Atkreipsime dėmesį, kad skaičius prasmingų operacijų, kurias galime atlikti su kintamaisiais, priklauso nuo jų tipo. Pavyzdžiui, nėra jokios prasmės skaičiuoti vardinio kintamojo vidurkį, net jei galimos tokio kintamojo reikšmės užkoduotos skaičiais. 2.6 lentelėje išvardintos leistinos operacijos įvairiems kintamiesiems.

Kintamojo tipas		Leistinos operacijos
Kategorinis	Vardinis	Įrašų, patekusių į kiekvieną kategoriją, skaičiaus radimas
	Ranginis	Įrašų, turinčių konkretų rangą, skaičiaus radimas. Rangų palyginimas (santykiai "daugiau", "mažiau")
Skaitinis	Intervalinis	Sudėtis, atimtis, daugyba, dalyba iš skaičiaus
	Santykinis	Visos matematinės operacijos

2.6 lentelė. Kintamųjų tipai ir leistinos operacijos

2.3 Pradinė duomenų analizė ir jų transformacijos

Pradiniai duomenys beveik niekada nebūna "gražūs" ir iš karto tinkami naudojimui. Todėl prieš pradedant spręsti vienokį ar kitokį duomenų tyrimo uždavinį, duomenis reikia paruošti taip, kad jie būtų tinkami numatomam tyrimo uždaviniui ir jo sprendimo algoritmui. Aptarsime dažniausiai pasitaikančias problemas ir galimus jų sprendimo būdus.

2.3.1 Duomenų transformacijos

Galimų reikšmių skaičiaus atžvilgiu, paprasčiausi yra kategoriniai kintamieji. Jų reikšmės, priklausomai nuo duomenų prigimties, yra labai įvairios. Tačiau, kaip žinia, bet koks matematinis aparatas labiausiai "pritaikytas" dirbti su skaičiais. Todėl, turėdami kategorinį

kintamąjį X, įgyjantį k ($k \geq 1$) skirtingų reikšmių, galime galvoti, kad

$$X \in \{1, 2, \ldots, k\}.$$

Prisiminkime oro sąlygų imtį 2.2 lentelėje. Juk kiekvienam matematikui ir, gal būt, daugumai duomenų tyrėjų užrašas $X_1 \in \{1, 2, 3\}$ yra žymiai malonesnis ir aiškesnis nei

$$X_1 \in \{\textit{saulėta},\ \textit{debesuota},\ \textit{lietinga}\}\,.$$

Dvi reikšmes įgyjantį kintamąjį galima koduoti binariniu kodu : 0 ir 1 . Beje, kartais statistiniuose tyrimuose k -reikšmis kategorinis kintamasis X transformuojamas į vadinamąjį fiktyvų (angl. dummy) kintamąjį X', kurio reikšmės yra k - ženkliai binariniai žodžiai, sudaryti iš $k-1$ nulio ir vieno vieneto. Pavyzdžiui, jau minėtas oro sąlygas nusakantis kintamasis X_1 būtų koduojamas taip

X_1	X_1'
saulėta	100
debesuota	010
lietinga	001

Skaitinių kintamųjų transformacijos priklauso nuo naudojamų duomenų tyrimo metodų. Dažnai algoritmai, besiremiantys atstumo p - matėje erdvėje skaičiavimu, reikalauja, kad skaitinių kintamųjų reikšmės būtų standartizuotos, kitaip sakant, priklausytų, pavyzdžiui, intervalui $[-1; 1]$ arba $[0; 1]$.

Tarkime, kad n yra imties dydis, o $x_1, x_2, \ldots, x_n$ yra šioje imtyje stebėtos skaitinio kintamojo X reikšmės. Yra įvairių X reikšmių standartizavimo būdų. Pateiksime tris iš jų.

1. **Dešimtainis standartizavimas**. Apibrėžkime sveiką neneigiamą skaičių K:

$$K = \min\left\{k\ :\ k \geq 0\,,\ 10^{-k} \max_{1 \leq i \leq n} |x_i| \leq 1\ \right\}.$$

   Tada standartizuotos reikšmės x_i' randamos, pastumiant kablelį per K pozicijų į kairę. Taigi

$$x_i' = x_i\, 10^{-K}\,.$$

   Akivaizdu, kad taip transformuotos reikšmės priklausys intervalui $[-1; 1]$.

2. **Min-max standartizavimas**. Dešimtainio standartizavimo trūkumas gali pasireikšti, kai visos reikšmės x_i' yra "sustumiamos" į palyginti mažą intervalo $[-1;1]$ dalį. Pavyzdžiui, jei visi $x_i \in [100;200]$, tai $x_i' \in [0,1;0,2]$. Tokios reikšmių koncentracijos leidžia išvengti min-max transformacija. Pažymėkime

$$m = \min_{1\le i\le n} x_i \quad \text{ir} \quad M = \max_{1\le i\le n} x_i\,.$$

Tada

$$x_i' = \frac{x_i - m}{M - m}\,.$$

Taip standartizuotos reikšmės priklausys intervalui $[0;1]$. Norint jas "išsklaidyti" intervale $[-1;1]$, pastarąją lygybę reikėtų pakeisti tokia

$$x_i' = \frac{2x_i - M - m}{M - m}\,.$$

3. ***z* - standartizavimas**. Tai labiausiai paplitęs standartizavimas, kuris reiškia vadinamųjų z reikšmių skaičiavimą. Tarkime, kad $\bar{x}$ yra kintamojo X imties vidurkis, o s - imties standartinis nuokrypis :

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i\,,$$

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}\,.$$

Tuomet z reikšmės skaičiuojamos pagal formulę

$$z_i = \frac{x_i - \bar{x}}{s}\,.$$

Šiai transformacijai būdinga tai, kad bet kurios duomenų aibės $\{x_1, x_2, \ldots, x_n\}$ z reikšmių vidurkis visada lygus 0, o standartinis nuokrypis visada lygus 1 : $\bar{z} = 0$, $s_z = 1$.

Gali kilti klausimas, kodėl vienaip ar kitaip duomenys transformuojami prieš pradedant juos tirti. Ar nebūtų paprasčiau, jei reikalingos transformacijos būtų tiesiog sudedamoji atitinkamų duomenų tyrimo algoritmų dalis ? Atsakymui į šį klausimą, galima paminėti du argumentus. Visų pirma tokia pat transformacija reikalinga keliems skirtingiems duomenų

tyrimo metodams. Todėl kiekvieną kartą ją skaičiuoti iš naujo neracionalu. Dar svarbesnis yra kitas motyvas. Duomenų tyrimo metu ir testuojant gautas išvadas, dažnai tenka dirbti tiek su visa imtimi, tiek su dalimi jos įrašų. Kad kiekvieną kartą negautume vis kitaip standartizuotus duomenis, atitinkamų transformacijų parametrai, kartą juos suskaičiavus, turėtų būti saugomi kartu su tyrimo rezultatais.

2.3.2 Tolydžiųjų kintamųjų diskretizavimas

Iš skaitinių tolydžiųjų kintamųjų galima gauti ranginius, o iš šių - vardinius kintamuosius, tačiau taip prarandama dalis informacijos. Pavyzdžiui, tegul kintamasis yra ūgis. Renkame informaciją apie studentų vaikinų ūgį. Mums tereikia žinoti, ar vaikinas žemaūgis (iki 160cm), ar vidutinio ūgio ([160; 180]), ar aukštaūgis (per 180 cm). Šiuo atveju ūgis jau nusakomas ranginiu kintamuoju, turinčiu tris leistinas reikšmes: *žemaūgis*,*vidutinio ūgio* ir *aukštaūgis*. Informacijos nuostoliai bus tuo mažesni, kuo "teisingiau" suskaidysime pradinį reikšmių intervalą į 3 dalis (jei iš anksto apsispręsta grupuoti studentus į 3 ūgio kategorijas).

Aptarsime vieną iš galimų panašaus uždavinio sprendimo būdų. Tarkime, kad X yra vienas iš tolydžių nepriklausomų kintamųjų, o Y - priklausomas kategorinis kintamasis, turintis k kategorijų. Galime laikyti, kad $Y \in \{1, 2, \ldots, k\}$, o imties įrašai yra $\{(x_i, y_i)\,,\ i = 1, 2, \ldots, N\}$. Diskretizuosime kintamąjį X , transformuodami jį į kategorinį kintamąjį, turintį ne daugiau kaip k_X kategorijų $(2 \leq k_X < N)$.

Pirmiausiai visą kintamojo X leistinų reikšmių intervalą skaidome į nesikertančius intervalus taip, kad skirtingos reikšmės x_i priklausytų skirtingiems intervalams. Tarkime, kad I_1 ir I_2 yra bet kurie du gretimi dalinimo intervalai. Toliau, pasirinkę reikšmingumo lygmenį $0 < \alpha < 1$, tikriname ar statistiškai reikšmingas skirtumas tarp kintamojo Y skirstinių, kai X priklauso intervalams I_1 ir I_2. Tikrinimui naudosime χ^2 kriterijų.

Pažymėkime n_{mj} skaičių įrašų, kuriems $X \in I_m$ ir $Y = j$ ($m = 1, 2\,;\ j = 1, 2, \ldots, k$), t.y.

$$n_{mj} = \sum_{\substack{i=1 \\ x_i \in I_m\,,\, y_i = j}}^{N} 1\,.$$

Be to, tegul $n_{\cdot j} = n_{1j} + n_{2j}$,

$$n_{m\cdot} = \sum_{j=1}^{k} n_{mj} \ .$$

ir

$$n = n_{1\cdot} + n_{2\cdot} = \sum_{j=1}^{k} n_{\cdot j} \ .$$

Kitaip sakant, n yra įrašų, kuriems $X \in I_1 \cup I_2$, skaičius. Visi šie žymėjimai ir jų tarpusavio sąryšiai atsispindi 2 .7 lentelėje.

$X \backslash Y$	1	2	...	k	Σ
I_1	n_{11}	n_{12}	...	n_{1k}	$n_{1\cdot}$
I_2	n_{21}	n_{22}	...	n_{2k}	$n_{2\cdot}$
Σ	$n_{\cdot 1}$	$n_{\cdot 2}$	...	$n_{\cdot k}$	n

2 .7 lentelė. Porinė dažnių lentelė

Kriterijaus funkcija (kitaip dar vadinama kriterijaus statistika) yra

$$\chi^2 = \sum_{m=1}^{2} \sum_{j=1}^{k} \frac{(n_{mj} - E_{mj})^2}{E_{mj}} \ ,$$

čia E_{mj} - tikėtinieji dažniai. Jie apskaičiuojami pagal formulę

$$E_{mj} = \frac{n_{m\cdot}\, n_{\cdot j}}{n} \ .$$

Laisvės laipsnių skaičius lygus $k - 1$.

Jei dažnių lentelėje kuris nors iš $n_{m\cdot}$ ar $n_{\cdot j}$ lygus 0, laikysime, kad E_{mj} lygus kokiam nors mažam teigiamam skaičiui, tarkime $E_{mj} = 0,1$. Tuo siekiama išvengti dalybos iš 0 , skaičiuojant χ^2.

Jei gautoji χ^2 reikšmė yra mažesnė už χ^2 skirstinio su $k - 1$ laisvės laipsniu α lygmens kritinę reikšmę $\chi^2_\alpha(k-1)$, tai galime tvirtinti, kad kintamojo Y skirstinys nepriklauso nuo to kuriam iš intervalų I_1 ar I_2 priklauso X reikšmės. Todėl intervalus I_1 ir I_2 galime sujungti ir vienetu sumažinti turimų intervalų skaičių.

Praktiškai procedura atliekama taip. Suskaičiuojamos χ^2 reikšmės visoms gretimų intervalų poroms. Tegul I_1 ir I_2 yra gretimi intervalai, kuriems ši reikšmė buvo mažiausia, tarkime $\chi^2 = u$. Toliau nagrinėjame du galimus atvejus.

a) Jei

$$u < \chi^2_\alpha(k-1) \ ,$$

tai intervalus I_1 ir I_2 sujungiame ir vėl skaičiuojame χ^2 reikšmes tik jau mažesniam intervalų skaičiui. Tai kartojame tol, kol intervalų skaičius taps ne didesnis už pageidaujamą kategorijų kiekį k_X. Pastebėsime, kad kiekviename žingsnyje (aišku, išskyrus pirmąjį) reikės perskaičiuoti ne daugiau, kaip dvi χ^2 reikšmes.

b) Kuriame nors žingsnyje gauname, kad

$$u \geq \chi^2_\alpha(k-1)$$

ir turimas intervalų skaičius vis dar didesnis už k_X. Tada, sumažinę reikšmingumo lygmenį α, padidiname $\chi^2_\alpha(k-1)$ ir tęsiame procedūrą (žr. punktą a))

Jei reikia diskretizuoti kelis kintamuosius, ši procedūra kartojama, kiekvienam iš jų parenkant tinkamus parametrus k_X ir α . Pastebėsime, kad $\chi^2_\alpha(k-1)$ reikšmės randamos lentelėse arba skaičiuojamos, panaudojant specialias statistinių paketų funkcijas. Panagrinėkime pavyzdį.

Pavyzdys.

Buvo tiriamas naujos gydymo metodikos efektyvumas pacientams, kurių amžius iki 60 metų. Rezultatą nusako kintamasis Y . Jei po gydymo kurso užfiksuotas pagerėjimas, tai $Y = 1$. Priešingu atveju $Y = 2$. 2.8 lentelėje pateikti duomenys apie 12 stebėtų ligonių. Pasirinkę reikšmingumo lygmenį $\alpha = 0,1$ ir jo nekeisdami, suskaidysime X leistinų reikšmių intervalą $(0; 60]$ taip, kad gautume kuo mažesnį diskretizuoto kintamojo kategorijų skaičių. Turime, kad Y kategorijų skaičius $k = 2$. Tada χ^2 skirstinio kritinių reikšmių lentelėje randame $\chi^2_{0,1}(1) = 2,706$.

Pradinius intervalo skaidymo taškus pasirinksime, imdami vidurius tarp dviejų gretimųjų kintamojo X reikšmių. Taigi pradinis skaidinys turės 12 intervalų : $(0; 2]$, $(2; 5], \ldots, (52,5; 60]$. Vienuolikai gretimų intervalų porų apskaičiuojame χ^2 reikšmes. Mažiausia bus kai

	Paciento amžius (m.) X	Gydymo rezultatas Y
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

2 .8 lentelė. Gydymo rezultatai

$I_1 = (7,5;\ 8,5]$, $I_2 = (8,5;\ 10]$. Pažiūrėkime kaip ji randama. Šiuos intervalus atitinkanti porinė dažnių lentelė yra

$X\backslash Y$	1	2	Σ
$I_1 = (7,5;\ 8,5]$	$n_{11} = 1$	$n_{12} = 0$	$n_{1\cdot} = 1$
$I_2 = (8,5;\ 10]$	$n_{21} = 1$	$n_{22} = 0$	$n_{2\cdot} = 1$
Σ	$n_{\cdot 1} = 2$	$n_{\cdot 2} = 0$	$n = 2$

Pagal šiuos duomenis apskaičiuojame tikėtinuosius dažnius. Turėsime

$$E_{11} = E_{21} = \frac{1 \cdot 2}{2} = 1$$

ir $E_{12} = E_{22} = 0,1$, nes $n_{\cdot 2} = 0$. Todėl

$$\chi^2 = \frac{(1-1)^2}{1} + \frac{(0-0,1)^2}{0,1} + \frac{(1-1)^2}{1} + \frac{(0-0,1)^2}{0,1} = 0,2\,.$$

Kadangi $\chi^2 < 2,706$, tai sujungę nagrinėjamus intervalus, vietoje I_1 ir I_2 turėsime vieną naują intervalą $(7,5;\ 10]$.

Toliau tęsdami intervalų jungimo procesą, ateisime iki trijų intervalų skaidinio :

$$(0;\ 10],\ (10;\ 42],\ (42;\ 60]\ .$$

Ar galima kuriuos nors iš jų sujungti? Suskaičiuosime dvi šio skaidinio gretimų intervalų poras atitinkančias χ^2 reikšmes.

1. Skaičiavimai intervalams $(0;\ 10]$ ir $(10;\ 42]$.

$X\backslash Y$	1	2	Σ
$I_1 = (0;\ 10]$	$n_{11} = 4$	$n_{12} = 1$	$n_{1\cdot} = 5$
$I_2 = (10;\ 42]$	$n_{21} = 1$	$n_{22} = 3$	$n_{2\cdot} = 4$
Σ	$n_{\cdot 1} = 5$	$n_{\cdot 2} = 4$	$n = 9$

$$E_{11} \approx 2,78\,,\ E_{12} \approx 2,22\,,\ E_{21} \approx 2,22\,,\ E_{22} \approx 1,78\,.$$

Todėl $\chi^2 \approx 2,72$.

2. Skaičiavimai intervalams $(10;\ 42]$ ir $(42;\ 60]$.

$X\backslash Y$	1	2	Σ
$I_1 = (10;\ 42]$	$n_{11} = 1$	$n_{12} = 3$	$n_{1\cdot} = 4$
$I_2 = (42;\ 60]$	$n_{21} = 3$	$n_{22} = 0$	$n_{2\cdot} = 3$
Σ	$n_{\cdot 1} = 4$	$n_{\cdot 2} = 3$	$n = 7$

$$E_{11} \approx 2,29\,,\ E_{12} \approx 1,71\,,\ E_{21} \approx 1,71\,,\ E_{22} \approx 1,29\,.$$

Iš čia gauname $\chi^2 \approx 3,96$.

Kaip matome, abiem atvejais χ^2 reikšmės didesnės už kritinę reikšmę $\chi^2_{0,1}(1) = 2,706$. Tai rodo, kad tarp šių intervalų yra esminiai skirtumai ir jungti juos nerekomenduojama.

Tiriamą imtį dabar galime užrašyti dažnių lentelės pavidalu

Paciento	Gydymo rezultatas	
Amžius	**1**	**2**
jaunas	4	1
vidutinis	1	3
vyresnis	3	0

Čia pacientai skirstomi į tris amžiaus grupes, nusakomas gautais metų intervalais (0; 10], (10; 42], (42; 60] . Žinoma, dalies informacijos, lyginant su 2.8 lentele, netekome. Pavyzdžiui, vidutiniam pacientų amžiui skaičiuoti pastaroji lentelė nebetiktų. Tačiau ji lengviau vizualiai suvokiama.

2.3.3 Trūkstamosios reikšmės

Dažnai dėl tam tikrų priežasčių neįmanoma išmatuoti kai kurių vieno ar kelių objektų parametrų. Tai reiškia, kad kai kuriuose įrašuose nėra vieno ar kelių kintamųjų reikšmių. Tai vadinamosios *trūkstamosios reikšmės* (praleistieji stebėjimai). Jei tokių įrašų dalis visoje imtyje nedidelė, galima juos paprasčiausiai išmesti arba, atliekant skaičiavimus, ignoruoti. Priklausomai nuo to, kokiomis programinėmis priemonėmis atliekamas tyrimas.
Tačiau kartais ir trūkstamos reikšmės (arba jų skaičius) yra informatyvios. Pavyzdžiui, paprašius įvertinti politiko populiarumą 10 balų skalėje, 1% apklaustųjų jį įvertino dešimtukais, o 99% nurodė, kad tokio politiko nežino (t.y. turime net 99% trūkstamų reikšmių). Taigi ar šis politikas gali laikyti save populiariu ? Šiuo atveju, matyt, reikėtų kitaip planuoti apklausą. Pavyzdžiui, galima išplėsti populiarumo vertinimo skalę iki 11, laikant, kad nepažįstamo politiko populiarumas lygus 0. Kita vertus, jei imtis nedidelė ir yra galimybė, reikėtų pabandyti atstatyti trūkstamas reikšmes, pasitelkiant į pagalbą ir duomenų savininkus ar tyrimo užsakovus.
Pagaliau paprasčiausias (bet nebūtinai geriausias) būdas eliminuoti trūkstamas reikšmes, automatiškai pakeičiant jas tam tikromis konstantomis. Galimi, pavyzdžiui, tokie variantai.

1. Visas trūkstamas vieno kintamojo reikšmes keičiame viena konstanta. Šios konstantos reikšmė labai priklauso tiek nuo tiriamų duomenų prigimties, tiek nuo tyrimo metodų.

2. Visas trūkstamas vieno skaitinio kintamojo reikšmes keičiame jo vidurkiu.

3. Jei įrašai jau yra kokiu nors būdu klasifikuoti, tai trūkstamas vieno skaitinio kintamojo reikšmes keičiame jo vidutine reikšme toje klasėje, kuriai priklauso nagrinėjamas įrašas.

Šie paprasti problemos sprendimo būdai atrodo viliojančiai. Tačiau jais piktnaudžiauti nevertėtų. Keičiant vieno ar, juo labiau, kelių kintamųjų trūkstamas reikšmes, duomenys yra iškraipomi. Viena konstanta keičiant visų kintamųjų trūkstamas reikšmes, galime gauti taip vadinamas *išskirtis* t.y. mažai tikėtinas arba net visiškai neįmanomas kai kurių kintamųjų reikšmes. Pavyzdžiui, nutarę visų skaitinių kintamųjų trūkstamas reikšmes pakeisti 0, galime "atrasti", kad kai kurie piliečiai visai nieko nesveria arba jų svoris normalus, bet ūgis lygus 0.

Bet kuriuo atveju, trūkstamų reikšmių pakeitimas dažniausiai remiasi prielaida, kad tos trūkstamos reikšmės "nedaro didelės įtakos" tyrimo rezultatui. Ar tikrai taip, reikėtų įsitikinti sprendžiant tą pati uždavinį skirtingais metodais, taikant skirtingus trūkstamų reikšmių eliminavimo būdus ar net atsisakant kintamųjų, turinčių daug nežinomų reikšmių. Beje, išskirtys gali atsirasti ne tik eliminuojant trūkstamas reikšmes. Tai gali būti paprasčiausia duomenų įvedimo ar anketos pildymo klaida, pvz. nurodant kliento amžių, parašyta 96 vietoje 69. Bet kartais išskirtys atspindi kokį nors objektyvų reiškinį. Tai liudija toks pavyzdys. Analizuojant prekybos tinklo nuolaidų kortelių duomenis, buvo pastebėta nedidelė grupė klientų, perkančių neįtikėtinai daug. Atidžiau ištyrus šias išskirtis, paaiškėjo, kad visos kortelės priklauso to paties prekybos tinklo parduotuvių kasininkams. Jie paprasčiausiai "paskolindavo" savo korteles jų neturintiems klientams, siekdami gauti daugiau premijinių taškų.

Kategorinio kintamojo išskirtimi paprastai laikoma reikšmė, nepriklausanti jo leistinų reikšmių aibei. Sunkiau apibūdinti skaitinio kintamojo išskirtį. Pavyzdžiui, kokia kreditinės kortelės apyvarta laikytina išskirtimi ? Intuityviai aišku, kad tai priklauso ne tik nuo apyvartos dydžio, bet ir nuo tiriamos populiacijos. Juk auksinių kortelių savininkų imtyje ir studentų imtyje išskirtinės apyvartos dydis turėtų skirtis.

Todėl dažniausiai išskirtimis laikomos tos skaitinio kintamojo reikšmės x_i, kurių z reikšmės absoliučiuoju didumu didesnės už 3 : $|z_i| > 3$. Toks apibrėžimas grindžiamas tikimybių

teorijoje gerai žinoma Čebyšovo nelygybe. Beje, kai kurie autoriai siūlo laikyti "įtartinomis" jau tas reikšmes x_i, kurioms $|z_i| > 2$.

Taigi tiek trūkstamas reikšmes, tiek išskirtis reikia analizuoti atidžiai, gerai suprantant turimų duomenų prigimtį. Pageidautina, kad šiame procese dalyvautų ir duomenų savininkai arba tos srities ekspertai.

2.3.4 Objektų artumo matai

Skiriamos dvi objektų artumo matų rūšys: *atstumai* ir *panašumo* (kitaip: *asociatyvumo*) *koeficientai*.

2.3.1 apibrėžimas. *Atstumu tarp dviejų objektų* $\boldsymbol{a}$ *ir* $\boldsymbol{b}$ *vadinsime neneigiamą skaitinę funkciją* $d(\boldsymbol{a}, \boldsymbol{b})$, *nusakančią kiek skirtingi* $\boldsymbol{a}$ *ir* $\boldsymbol{b}$. *Objektai yra panašūs, jei atstumas tarp jų mažas.*

2.3.2 apibrėžimas. *Dviejų objektų* $\boldsymbol{a}$ *ir* $\boldsymbol{b}$ *panašumo koeficientu vadinsime skaitinę funkciją* $s(\boldsymbol{a}, \boldsymbol{b})$, *nusakančią kiek panašūs* $\boldsymbol{a}$ *ir* $\boldsymbol{b}$. *Kuo didesnis panašumo koeficientas, tuo panašesniais vadinami objektai.*

Daugelis atstumų yra metrikos.

2.3.3 apibrėžimas. *Atstumas* $d(\boldsymbol{a}, \boldsymbol{b})$ *yra metrika, jei jis tenkina sąlygas:*

1) simetriškumo: $d(\boldsymbol{a}, \boldsymbol{b}) = d(\boldsymbol{b}, \boldsymbol{a})$;

2) trikampio nelygybės: $d(\boldsymbol{a}, \boldsymbol{b}) \leq d(\boldsymbol{a}, \boldsymbol{c}) + d(\boldsymbol{c}, \boldsymbol{b})$;

3) netapačių objektų atskiriamumo: $d(\boldsymbol{a}, \boldsymbol{b}) = 0 \iff \boldsymbol{a} = \boldsymbol{b}$.

Artumo mato pasirinkimas labai priklauso nuo matuojamų objektų atributų tipo, matavimo skalės bei sprendžiamo uždavinio. Tarsime, kad objektas nusakomas imties įrašo atributų reikšmėmis. Todėl toliau neskirsime sąvokų "objektų artumas" ir "įrašų artumas"[1]. Pradžioje aptarsime paprastų vieno atributo įrašų artumo matus. Tegul dviejų įrašų atributo reikšmės yra u ir v. Dažniausiai sutinkami atstumai ir panašumo koeficientai, priklausomai nuo atributo tipo, pateikiami 2.9 lentelėje Nesunku įsitikinti, kad kad visi šioje lentelėje

[1] Iš tikrųjų objektas ir jo atributų reikšmių rinkinys aišku nėra tas pats. Pavyzdžiui gali sutapti dviejų žmonių ūgis, bet ne patys žmonės.

Atributo tipas	Atstumas	Panašumo koeficientas
Vardinis	$d(u,v) = \begin{cases} 0\,, & \text{jei } u = v\,, \\ 1\,, & \text{jei } u \neq v\,. \end{cases}$	$s(u,v) = 1 - d(u,v)$
Ranginis (galimos reikšmės vaizduojamos skaičiais $0, 1, 2, \ldots, M$)	$d(u,v) = \dfrac{\lvert u - v \rvert}{M}$	$s(u,v) = 1 - d(u,v)$
Skaitinis	$d(u,v) = \lvert u - v \rvert$	$s(u,v) = \dfrac{1}{1 + d(u,v)}$
		$s(u,v) = \mathrm{e}^{-d(u,v)}$
		$s(u,v) = 1 - \dfrac{d(u,v) - d_{\min}}{d_{\max} - d_{\min}}$

2 .9 lentelė. Vieno atributo reikšmių u ir v artumo matai

paminėti atstumai yra metrikos.

Kai objektą nusako k skaitinių atributų, tenka matuoti atstumus tarp vektorių $\mathbf{u} = (u_1, u_2, \ldots, u_k)$ ir $\mathbf{v} = (v_1, v_2, \ldots, v_k)$. Dažniausiai naudojami Minkovskio metrikos

$$d(\mathbf{u}, \mathbf{v}) = \left(\sum_{i=1}^{k} |u_i - v_i|^q \right)^{1/q}, \quad q \geq 1\,, \tag{2 .1}$$

atskiri atvejai.

- $q = 1$. *Manheteno (blokinis) atstumas*

$$d(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{k} |u_i - v_i|\,. \tag{2 .2}$$

  Kai $\mathbf{u}$ ir $\mathbf{v}$ yra binariniai vektoriai, ši metrika dar vadinama *Hamingo atstumu* ir reiškia nesutampančių bitų skaičių.

- $q = 2$. *Euklido metrika*

$$d(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_{i=1}^{k} |u_i - v_i|^2}\,. \tag{2 .3}$$

- $q = \infty$. *Čebyšovo (maksimumo) atstumas*

$$d(\mathbf{u}, \mathbf{v}) = \max_i |u_i - v_i| \,. \tag{2.4}$$

Skaičiuojant Čebyšovo atstumą, naudojama tik dalis turimos informacijos apie objektus. Kada tai pateisinama? Tarkime, grupei keliautojų atliekamas suderinamumo testas. Norima sudaryti grupeles (klasterius), į kurias patektų panašių charakterių (matuojamų atributų) žmonės, t.y. visi jų skirtumai nebūtų ryškūs. Tinkamiausias šiuo atveju panašumo kriterijus yra mažas Čebyšovo atstumas.

Vienas iš metrinių atstumų matų trūkumų - nevienoda skirtingai matuojamų atributų įtaka. Kintamieji, kurių sklaidos charakteristikos įgyja dideles reikšmes, gali nustelbti mažai įvairuojančių kintamųjų įtaką. Tarkime, turime du vektorius $(1;\ 200)$ ir $(1,1;\ 500)$. Euklido atstumas tarp jų yra

$$\sqrt{0,1^2 + 300^2} \approx 300,0002 \,.$$

Atstumą faktiškai nulemia antroji vektorių koordinatė. Vienas iš būdų šio trūkumo išvengti - užuot naudojus pačius atributus, imti jų standartizuotąsias reikšmes (žr. 2.3.1 skyrelį). Kitas būdas - turint tokius duomenis, naudoti mąstelių skirtumus kompensuojančią metriką. Viena iš tokių - Mahalanobio atstumas

$$d_M(\mathbf{u}, \mathbf{v}) = \sqrt{(\mathbf{u} - \mathbf{v})\mathbf{\Sigma}^{-1}(\mathbf{u} - \mathbf{v})^T} \,, \tag{2.5}$$

čia $\mathbf{\Sigma}^{-1}$ yra atributų kovariacijų matricos

$$\mathbf{\Sigma} = (\sigma_{ij})_{k \times k}$$

atvirkštinė matrica. Priminsime, kad i - tojo ir j - atributų kovariacija apibrėžiama taip:

$$\sigma_{ij} = \mathrm{cov}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{n-1} \sum_{l=1}^{n} (x_{li} - \bar{x}_i)(x_{lj} - \bar{x}_j) \,, \tag{2.6}$$

čia $\mathbf{x}_i = (x_{1i}, x_{2i}, \ldots, x_{ni})^T$ i - tojo atributo reikšmių stulpelis,

$$\bar{x}_i = \frac{1}{n} \sum_{l=1}^{n} x_{li} \,,$$

$i = 1, 2, \ldots, k$.

Kai kovariacijų matrica yra vienetinė, Mahalanobio atstumas sutampa su Euklido metrika. Atstumas tarp diskrečiųjų vektorių paprastai reiškiamas *nesutapimų metrika*

$$d_\Delta(\mathbf{u},\mathbf{v}) = \frac{1}{k}\sum_{\substack{i=1\\ u_i\neq v_i}}^{k} 1\,. \tag{2.7}$$

Aptarsime ir keletą vektorių panašumo matų. Vektorių skaliarinę sandaugą ir vektoriaus ilgį žymėsime

$$\mathbf{u}\cdot\mathbf{v} = \sum_{i=1}^{k} u_i v_i\,, \quad \|\mathbf{v}\| = \sqrt{\mathbf{v}\cdot\mathbf{v}}\,.$$

Be to, pažymėsime $\bar{\mathbf{v}}$ - vidurkių vektorių, sudarytą iš k vienodų koordinačių $\bar{v}$

$$\bar{\mathbf{v}} = (\bar{v},\bar{v},\ldots,\bar{v})\,, \quad \bar{v} = \frac{1}{k}\sum_{i=1}^{k} v_i\,.$$

Keletas dažniausiai naudojamų panašumo koeficiento išraiškų pateikiama 2.10 lentelėje. Palyginsime binarinių vektorių suderinamumo ir Žakardo panašumo koeficientus. Jei $\mathbf{u}$ ir $\mathbf{v}$ yra binariniai vektoriai, tai 0 ir 1 išsidėstymą jų koordinatėse nusako dažnių lentelė

$u_i \backslash v_i$	0	1
0	k_{00}	k_{01}
1	k_{10}	k_{11}

Čia k_{lm} reiškia koordinačių, tenkinančių sąlygas $u_i = l\,,\ v_i = m$, skaičių. Aišku, kad

$$k_{00} + k_{01} + k_{10} + k_{11} = k\,,$$

o sutampančių koordinačių skaičius yra lygus $k_{00} + k_{11}$. Todėl suderinamumo koeficientas

$$s_{\text{sud}}(\mathbf{u},\mathbf{v}) = 1 - d_\Delta(\mathbf{u},\mathbf{v}) = \frac{k_{00}+k_{11}}{k}\,.$$

Skaičiuojant Žakardo panašumo koeficientą $s_J(\mathbf{u},\mathbf{v})$, sutampantys nuliai nėra svarbūs

$$s_J(\mathbf{u},\mathbf{v}) = \frac{k_{11}}{k_{01}+k_{10}+k_{11}}\,.$$

Pavadinimas	$s(\mathbf{u}, \mathbf{v})$ formulė
Suderinamumo	$1 - d_\Delta(\mathbf{u}, \mathbf{v})$
Žakardo	$\dfrac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 - \mathbf{u} \cdot \mathbf{v}}$
Vektorių kampo kosinusas	$\dfrac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$
Koreliacijos	$\dfrac{(\mathbf{u} - \bar{\mathbf{u}}) \cdot (\mathbf{v} - \bar{\mathbf{v}})}{\|\mathbf{u} - \bar{\mathbf{u}}\| \, \|\mathbf{v} - \bar{\mathbf{v}}\|}$

2 .10 lentelė. Vektorių $\mathbf{u}$ ir $\mathbf{v}$ panašumo koeficientai $s(\mathbf{u}, \mathbf{v})$

2.3.1 pavyzdys. Parduotuvė prekiauja 10 pavadinimų prekėmis. Dviejų pirkėjų "krepšeliai" vaizduojami binariniais vektoriais

$$\begin{aligned} \mathbf{u} &= (0, 1, 0, 1, 0, 0, 0, 0, 0, 0) \\ \mathbf{v} &= (1, 0, 1, 0, 0, 0, 0, 0, 0, 0)\,. \end{aligned}$$

Tai rodo, kad pirmasis pirkėjas pirko tik antro ir ketvirto, o antrasis - pirmo ir trečio pavadinimo prekes. Kaip matome, pirkėjų poreikiai skirtingi. Tačiau suderinamumo koeficientas rodo ką kita

$$s_{\text{sud}}(\mathbf{u}, \mathbf{v}) = 1 - d_\Delta(\mathbf{u}, \mathbf{v}) = 1 - 0,4 = 0,6\,.$$

Tokį rezultatą sąlygoja daug sutampančių nulių. Bet prekės, kurios nesudomino nė vieno pirkėjo, nėra svarbios (pagalvokite kokį rezultatą gautume, jei parduotuvės asortimentas būtų ne 10, o pavyzdžiui, 10000 pavadinimų). Žakardo koeficientas šiuo atveju objektyvesnis, nes jis ignoruoja sutampančius nulius:

$$s_J(\mathbf{u}, \mathbf{v}) = \frac{0}{2 + 2 - 0} = 0\,.$$

Tokie atributai, kuriems svarbios yra tik nenulinės reikšmės, kartais dar vadinami *asimetriniais*.

Koreliacijos koeficientai paprastai naudojami kaip atsitiktinių dydžių (atributų) panašumo matai. Kartais jais remiantis vertinamas objektų (įrašų) panašumas. Statistikoje įprastas $\mathbf{u}$ ir $\mathbf{v}$ koreliacijos koeficiento apibrėžimas yra

$$r(\mathbf{u},\mathbf{v}) = \frac{\mathrm{cov}(\mathbf{u},\mathbf{v})}{s_{\mathbf{u}} s_{\mathbf{v}}}\,,$$

čia kovariacija $\mathrm{cov}(\mathbf{u},\mathbf{v})$ randama pagal (2.6) formulę, o $s_{\mathbf{v}}$ ir $s_{\mathbf{u}}$ yra vektorių standartiniai nuokrypiai

$$s_{\mathbf{u}} = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(u_i-\bar{u})^2}\,, \quad s_{\mathbf{v}} = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(v_i-\bar{v})^2}\,.$$

Nesunku įsitikinti, kad

$$r(\mathbf{u},\mathbf{v}) = \frac{(\mathbf{u}-\bar{\mathbf{u}})\cdot(\mathbf{v}-\bar{\mathbf{v}})}{\|\mathbf{u}-\bar{\mathbf{u}}\|\,\|\mathbf{v}-\bar{\mathbf{v}}\|}\,.$$

Koreliacijos koeficientas visada priklauso intervalui $[-1;\ 1]$. Jei jis lygus $+1$ arba -1 , tai vektoriai yra tiesiškai priklausomi

$$\mathbf{v} = a\mathbf{u} + b\,,$$

be to krypties koeficientas a ir $r(\mathbf{u},\mathbf{v})$ turės vienodus ženklus. Tačiau, jei $r(\mathbf{u},\mathbf{v}) = 0$, tai dar nereiškia, kad tarp vektorių nėra jokios priklausomybės. Pavyzdžiui, imkime vektorius

$$\begin{aligned}\mathbf{u} &= (-2,-1,0,1,2)\,,\\ \mathbf{v} &= (4,1,0,1,4)\,,\end{aligned}$$

kurių koordinatės susietos lygybe $v_i = u_i^2$. Tačiau, nežiūrint to, $r(\mathbf{u},\mathbf{v}) = 0$. Kitaip sakant, koreliacijos koeficiento lygybė nuliui rodo, kad nėra tiesinės priklausomybės.

Jau aptarėme homogeniškų vektorių panašumo matus, t.y. kai visos vektoriaus komponentės yra vieno tipo atributų reikšmės. Deja dažniausiai imties atributai būna skirtingų tipų. Tad kaip matuoti atstumą tarp heterogeniškų vektorių? Natūralus problemos sprendimas būtų toks: pasirinkti iš 2.9 lentelės tinkamą matą kiekvienam atributui (vektoriaus koordinatei) ir iš jų sudaryti bendrą matą (pavyzdžiui, apskaičiuojant vidutinę reikšmę). Esant galimybei, patartina kiekvieno atributo matą taip normuoti, kad jis priklausytų intervalui $[0;\ 1]$. Be to, galima įvesti papildomus svorio koeficientus, leidžiančius atsižvelgti į galimą atributų asimetriją ir pageidaujamą įtaką bendrajam matui. Gautume tokią heterogeniškų vektorių $\mathbf{u}$ ir $\mathbf{v}$ artumo mato konstrukciją.

1. Pasirenkame i -tojo atributo atstumą $d_i(\mathbf{u}, \mathbf{v}) = d(u_i, v_i) \in [0;\ 1]$ arba panašumo koeficientą $s_i(\mathbf{u}, \mathbf{v}) = s(u_i, v_i) \in [0;\ 1]$.

2. Apibrėžiame i -tojo atributo asimetrijos ir trūkstamų reikšmių indikatorių δ_i . Jis lygus 0, jei i - tasis atributas asimetrinis ir $u_i = v_i = 0$ arba kai kuriame nors vektoriuje trūksta i - tosios koordinatės. Kitais atvejais $\delta_i = 1$.

3. Pasirenkame i -tojo atributo svorį $w_i \geq 0$ taip, kad visų svorių suma būtų lygi 1

$$\sum_{i=1}^{k} w_i = 1\,.$$

4. Randame atstumą tarp vektorių $\mathbf{u}$ ir $\mathbf{v}$

$$d(\mathbf{u}, \mathbf{v}) = \left(\sum_{i=1}^{k} \delta_i\right)^{-1} \sum_{i=1}^{k} w_i \delta_i d_i(\mathbf{u}, \mathbf{v}) \tag{2.8}$$

arba jų panašumo koeficientą

$$s(\mathbf{u}, \mathbf{v}) = \left(\sum_{i=1}^{k} \delta_i\right)^{-1} \sum_{i=1}^{k} w_i \delta_i s_i(\mathbf{u}, \mathbf{v}) \tag{2.9}$$

Skaitiniams vektoriams svorius analogiškai galima įvesti ir Minkovskio metrikoje (2.1)

$$d(\mathbf{u}, \mathbf{v}) = \left(\sum_{i=1}^{k} w_i |u_i - v_i|^q\right)^{1/q}, \quad q \geq 1\,. \tag{2.10}$$

Jau matėme, kad objektų artumo mato pasirinkimas labai priklauso nuo imties atributų tipo. Tačiau šį pasirinkimą įtakoja ir kiti faktoriai: sprendžiamo uždavinio specifika, naudojama programinė įranga, ankstesnė panašių uždavinių sprendimo patirtis ir t.t. Todėl gali tekti pabandyti įvairus artumo matus, norint rasti tinkamiausią.

2.4 Duomenų tyrimo uždavinių tipai

Turėdami vienokius ar kitokius duomenis, priklausomai nuo poreikio, galime formuluoti ir spręsti įvairius duomenų tyrimo uždavinius.

Kontroliuojamo mokymo uždaviniai.

Tarkime imties įrašai sudaryti iš nepriklausomo kintamojo X ir priklausomo kintamojo Y

reikšmių: $(x_1, y_1), \ldots, (x_n, y_n)$. Beje kintamieji gali būti ir daugiamačiai. Tada žymėsime $X = (X_1, \ldots, X_k)$, o priklausomas kintamasis $Y = (Y_1, \ldots, Y_m)$. Analogiškai $x_i = (x_{i1}, \ldots, x_{ik})$ ir $y_i = (y_{i1}, \ldots, y_{im})$ žymėsime i - tojo imties įrašo reikšmes. Priklausomai nuo kintamojo Y tipo, skiriami tokie kontroliuojamo mokymo uždaviniai

- **Klasifikavimas.** Kintamasis Y - kategorinis. Iš turimų imties duomenų reikia "išmokti" visoje populiacijoje nustatyti Y reikšmę pagal žinomą X reikšmę . Kitaip sakant, reikia "išmokti" nustatyti kuriai kintamojo Y kategorijai priklauso bet kuris populiacijos objektas, kai žinoma tik jo X reikšmė. 2.1.1 pavyzdyje (žr. 2.2 lentelė) kintamasis $X = (X_1, X_2, X_3, X_4)$ nusako oro sąlygas, pagal kurias reikia nuspręsti kuriai Y kategorijai priklauso konkreti meteorologinė situacija, t.y., rungtynės įvyks ar ne. Panašiai 2.1.3 pavyzdyje pagal vainiklapio ir taurėlapio matmenis irisai klasifikuojami į tris kategorijas (rūšis); 2.1.4 pavyzdyje pagal 256 pikselių šviesumą nustatomas piešinyje pavaizduotas skaitmuo. Aišku, kad ne visada pavyks visą populiaciją klasifikuoti tiksliai arba sukonstruotas absoliučiai teisingas klasifikatorius (pavyzdžiui klasifikacijos taisyklių seka arba sprendimų medis) bus per daug sudėtingas. Todėl iš tikrųjų yra konstruojamas kintamojo Y įvertis

$$\hat{Y} = f(X)\,, \tag{2.11}$$

stengiantis, kad kuo didesnei imties įrašų daliai jis būtų teisingas, t.y. $f(x_i) = y_i$. Kitaip sakant, turėdami įrašus (x_i, y_i) , mes galime *kontroliuoti* gautojo įverčio patikimumą.

2.1.2 pavyzdyje įvertis, nusakomas 2.2 paveiksle pavaizduotu sprendimų medžiu, yra teisingas 22 įrašams iš 24. Pastebėsime, kad šis pavyzdys skiriasi nuo kitų dar ir tuo, kad 2.3 lentelėje pateikiamos visos galimos keturmačio nepriklausomo kintamojo reikšmės. Todėl čia pagrindinis uždavinys - suteikti turimiems duomenims kuo aiškesnę struktūrą, nurodant aiškią ir patikimą įverčio funkciją f .

- **Skaitinė prognozė.** Uždavinys panašus į klasifikavimo uždavinį, tik šiuo atveju priklausomas kintamasis Y - skaitinis. Todėl, konstruojant įvertį (2.11), svarbios yra prielaidos apie funkcijos f analizines savybes ir kintamojo X komponenčių įtakos

svorį. Kiek įvertis atitinka tikrąjį kintamąjį Y nusako vadinamoji klaidų (nuostolių) funkcija $L(Y, f(X))$. Šios funkcijos parinkimas taip pat įtakoja galutinį rezultatą.

2.1.5 pavyzdyje visi kintamieji yra skaitiniai ir daroma prielaida, kad f yra tiesinė nepriklausomų kintamųjų funkcija

$$f(X) = \beta_0 + \sum_{j=1}^{6} \beta_j X_j \,.$$

Koeficientai $\beta = (\beta_0, \beta_1, \ldots, \beta_6)$ rasti vadinamuoju mažiausių kvadratų metodu. Kitaip sakant, pasirinkus klaidų funkciją

$$L(Y, f(X)) = (Y - f(X))^2 \,,$$

buvo minimizuojama suma

$$S(\beta) = \sum_{i=1}^{209} L(y_i, f(x_i)) \,.$$

Taigi, tiek spręsdami klasifikavimo, tiek skaitinės prognozės uždavinį, tai ko "išmokstame" (t.y. sukonstruojame įvertį $\hat{Y}$) galime "pasitikrinti" palygindami gautą rezultatą su imtyje turimais įrašais. Tačiau yra ir tokių uždavinių, kuriuos tenka spręsti be "mokytojo pagalbos".

Nekontroliuojamo mokymo uždaviniai.

Šiuo atveju nėra priklausomo kintamojo, todėl imtis susideda tik iš kintamojo X reikšmių $x_1, \ldots, x_n$. Beje kintamojo X dimensija k gali būti net didesnė už imties dydį n. Išskirsime tokius nekontroliuojamo mokymo uždavinius

- **Asociacijos taisyklių konstravimas.** Paprasčiausios asociacijos taisyklės buvo sukonstruotos 2.1.1 pavyzdyje. Buvo stengiamasi 2.1 lentelėje įžvelgti kokius nors dėsningumus, t.y., kokias nors "dėsningas" kintamųjų reikšmes.

  Paprastai šis uždavinys kyla analizuojant didelės dimensijos kategorinių kintamųjų imtis. Tipiškas pavyzdys - vadinamasis pirkėjo krepšelio uždavinys: analizuojant prekybos centro pardavimų duomenis, stengiamasi nustatyti kokios prekės dažniausiai perkamos kartu.

- **Klasterinė analizė.** Tikslas : turimus imties įrašus suskirstyti į tam tikras "natūralias" grupes (klasterius). Klasterių skaičius gali ir nebūti žinomas iš anksto.

  Grįžkime prie 2.1.3 pavyzdžio. Tarkime, kad mes turime taurėlapių ir vainiklapių matmenis, bet nežinome kokios rūšies buvo augalai. Tada vietoje 2.4 lentelės turėtume duomenis, pateiktus 2.11 lentelėje.

	Taurėlapio ilgis (X_1)	Taurėlapio plotis (X_2)	Vainiklapio ilgis (X_3)	Vainiklapio plotis (X_4)
1	5,1	3,5	1,4	0,2
2	4,9	3,0	1,4	0,2
3	4,7	3,2	1,3	0,2
4	4,6	3,1	1,5	0,2
5	5,0	3,6	1,4	0,2
...	...	...	...	...
50	5,0	3,3	1,4	0,2
51	7,0	3,2	4,7	1,4
52	6,4	3,2	4,5	1,5
53	6,9	3,1	4,9	1,5
54	5,5	2,3	4,0	1,3
55	6,5	2,8	4,6	1,5
...	...	...	...	...
100	5,7	2,8	4,1	1,3
101	6,3	3,3	6,0	2,5
102	5,8	2,7	5,1	1,9
103	7,1	3,0	5,9	2,1
104	6,3	2,9	5,6	1,8
105	6,5	3,0	5,8	2,2
...	...	...	...	...
150	5,9	3,0	5,1	1,8

2.11 lentelė. Irisų matmenys

Turimos imties klasterinė analizė turėtų "išmokyti" mus atsakyti į tris klausimus:

1) Kelių rūšių irisai buvo matuojami?

2) Kurios rūšies buvo kiekvienas iš 150 jau išmatuotų augalų ?

3) Kaip nustatyti bet kurio, naujai išmatuoto, iriso rūšį?

Aišku, kad šio ar kurio nors kito klasterizacijos uždavinio sprendimas ir atsakymas priklausys ir nuo to kaip bus matuojamas dviejų imties įrašų atstumas. Kitaip sakant, ką reiškia "panašios kintamųjų reikšmės". Tai ypač aktualu, kai dalis kintamųjų yra kategoriniai.

Skirtingai nuo klasifikacijos ir skaitinės prognozės, šis apmokymas nėra "kontroliuojamas" imties įrašais. Todėl patikrinti gautus rezultatus galima tik atsižvelgiant į tai, kiek jie atitinka realią praktiką.

3 Kontroliuojamo mokymo uždaviniai: klasifikavimas

3.1 Kontroliuojamas mokymas ir klasifikavimas

Tarkime imties įrašai sudaryti iš nepriklausomo kintamojo X ir priklausomo kintamojo Y reikšmių: $(x_1, y_1), \ldots, (x_n, y_n)$. Bendriausiu atveju kontroliuojamo mokymo uždavinys yra pagal šiuos duomenis sukonstruoti taisykles, leidžiančias kuo patikimiau prognozuoti Y reikšmę bet kuriai nepriklausomo kintamojo X reikšmei. Dažniausiai Y priklauso nuo daugelio parametrų. Tada nepriklausomas kintamasis yra daugiamatis $X = (X_1, \ldots, X_k)$, o jo komponentės kartais vadinamos atributais. Kontroliuojamo mokymo uždavinio sprendimas gali būti pavaizduotas tokia schema

$$\begin{array}{c}\text{Atributai}\\ (X_1, \ldots, X_k)\end{array} \quad \Longrightarrow \quad \boxed{\begin{array}{l}\text{Sukonstruotos}\\ \text{taisyklės}\end{array}} \quad \Longrightarrow \quad \begin{array}{l}\text{Kintamojo}\\ Y \text{ reikšmė}\end{array}$$

Priklausomai nuo kintamojo Y tipo, skirsime *klasifikavimo* ir *skaitinės prognozės* uždavinius

Kai priklausomas kintamasis Y yra kategorinis, tada pagal jo reikšmę imties įrašas (x_i, y_i) priskiriamas klasei, kuriai $Y = y_i$. Todėl Y vadinamas klasės kintamuoju, o klasių skaičių apsprendžia jo galimų reikšmių aibės dydis. 3.1 lentelėje pateikiami duomenys apie kai kurių stuburinių gyvūnų klasifikaciją. Zoologijoje skiriamos penkios stuburinių klasės: žinduoliai, paukščiai, žuvys, ropliai ir varliagyviai. Klasė, kuriai priskiriamas gyvūnas, priklauso nuo daugelio charakteristikų (atributų): kūno temperatūros, reprodukcijos būdo, galimybės skraidyti ir t.t. Lentelėje pateikiamos šešių atributų $X = (X_1, X_2, \ldots, X_6)$ ir klasės kintamojo Y reikšmės. Šiame pavyzdyje visi kintamieji yra kategoriniai. Bendruoju atveju klasifikavimo uždavinyje, skirtingai nuo regresijos, priklausomas kintamasis būtinai kategorinis. Tuo tarpu atributai gali būti ir skaitiniai.

3.1.1 apibrėžimas. *Tegul A_X yra nepriklausomų kintamųjų $X = (X_1, X_2, \ldots, X_k)$ galimų reikšmių aibė, o baigtinė aibė A_Y sudaryta iš visų galimų klasės kintamojo Y reikšmių. Klasifikavimo uždavinys yra pagal imties duomenis sukonstruoti funkciją*

$$f: \quad A_X \mapsto A_Y \, .$$

Ši funkcija vadinama klasifikavimo modeliu.

Pavadinimas	Kraujo tipas (X_1)	Odos danga (X_2)	Gyva-vedis (X_3)	Gyvena vandenyje (X_4)	Skraido (X_5)	Turi kojas (X_6)	Klasė (Y)
žmogus	šiltas	plaukai	taip	ne	ne	taip	žinduolis
pitonas	šaltas	žvynai	ne	ne	ne	ne	roplys
lašiša	šaltas	žvynai	ne	taip	ne	ne	žuvis
banginis	šiltas	plaukai	taip	taip	ne	ne	žinduolis
varlė	šaltas	nėra	ne	kartais	ne	taip	varliagyvis
komodo varanas	šaltas	žvynai	ne	ne	ne	taip	roplys
šikšnosparnis	šiltas	plaukai	taip	ne	taip	taip	žinduolis
balandis	šiltas	plunksnos	ne	ne	taip	taip	paukštis
katė	šiltas	kailis	taip	ne	ne	taip	žinduolis
tigrinis ryklys	šaltas	žvynai	taip	taip	ne	ne	žuvis
vėžlys	šaltas	žvynai	ne	kartais	ne	taip	roplys
pingvinas	šiltas	plunksnos	ne	kartais	ne	taip	paukštis
dygliuotis	šiltas	dygliai	taip	ne	ne	taip	žinduolis
ungurys	šaltas	žvynai	ne	taip	ne	ne	žuvis
salamandra	šaltas	nėra	ne	kartais	ne	taip	varliagyvis

3 .1 lentelė. Stuburiniai gyvūnai

Klasifikavimo modelio pagalba sprendžiami uždaviniai gali būti tokie.

1. Turimų duomenų klasifikavimas. Klasifikavimo modelis naudojamas kaip priemonė, leidžianti nustatyti kriterijus, kuriais remiantis, objektas priskiriamas vienai ar kitai klasei. Pavyzdžiui, būtų naudinga (ne tik biologui), 3.1 lentelėje pateiktų duomenų pagrindu, nustatyti kokie požymiai apsprendžia ar tiriamasis gyvūnas yra žinduolis, paukštis, žuvis, roplys ar varliagyvis.

2. Nežinomų duomenų prgnozė. Turėdami naujo objekto (imties įrašo) atributų reikšmes, klasifikavimo modelio pagalba priskiriame jį vienai iš galimų klasių. Pavyzdžiui, anksčiau nesutiktas gyvūnas *šiurpusis nuodadantis* pasižymi tokiomis savybėmis

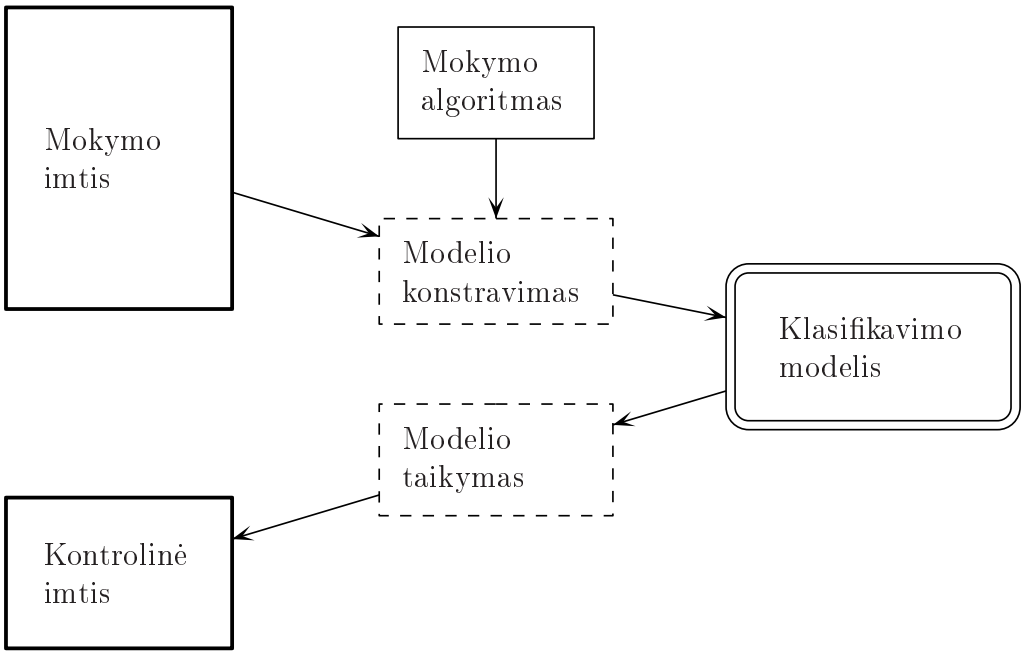
Pavadinimas	Kraujo tipas (X_1)	Odos danga (X_2)	Gyvavedis (X_3)	Gyvena vandenyje (X_4)	Skraido (X_5)	Turi kojas (X_6)	Klasė (Y)
šiurpusis nuodadantis	šaltas	žvynai	ne	ne	ne	taip	?

Kokiai klasei jis priklauso ? Atsakymą gali duoti 3.1 lentelėje pateiktų duomenų pagrindu sukonstruotas klasifikavimo modelis.

Klasifikavimo modeliai dažniausiai taikomi binarinių arba vardinių kintamųjų imtims. Ranginių kintamųjų atveju jie yra mažiau efektyvūs, nes , kaip taisyklė, neatsižvelgia į natūralią rangų tvarką (pvz. pradinis išsilavinimas yra mažesnis už universitetinį).
Klasifikavimo modelio konstravimo (kitaip: mokymo algoritmo) pagrindas yra imtis. Reikia rasti tokį modelį kuris teisingai atspindėtų sąryšį tarp imties atributų ir klasės kintamojo. Modelis turi kuo tiksliau atitikti turimus imties duomenis ir tuo pačiu teisingai nustatyti klases, kurioms priskirtini nauji įrašai. Kitaip sakant, geras modelis turi sugebėti apibendrinti tai, ko "išmoko" iš imties duomenų. Bendroji klasifikavimo modelio konstravimo schema pavaizduota 3.1 paveiksle.
Pirmiausiai turima duomenų aibė skaidoma į dvi dalis. Didesnioji imties įrašų su žinomomis klasės kintamojo reikšmėmis dalis patenka į *mokymo imtį*, o likusieji įrašai sudaro *kontrolinę imtį*. Pagal mokymo imties duomenis, naudojant vienokį ar kitokį mokymo algoritmą, konstruojamas klasifikavimo modelis. Po to jis taikomas kontrolinės imties įrašams. Modelio patikimumą nusako teisingai ir neteisingai klasifikuotų kontrolinės imties įrašų skaičių santykis. Iš šių skaičių sudaryta lentelė vadinama *nesutapimų matrica*. 3.2 lentelėje pavaizduota binarinio klasifikavimo modelio nesutapimų matrica. Čia n_{ij} yra kontrolinės imties įrašų, priklausančių klasei i, bet priskirtų klasei j, skaičius. Neteisingai klasifikuotų kontrolinės imties įrašų dalis

$$e = \frac{n_{01} + n_{10}}{n_{00} + n_{01} + n_{10} + n_{11}}$$

vadinama modelio *klaidos koeficientu*. Dauguma mokymo algoritmų, konstruodami klasifikavimo modelius, siekia minimizuoti klaidos koeficientą.



3 .1 pav. Klasifikavimo modelio konstravimas

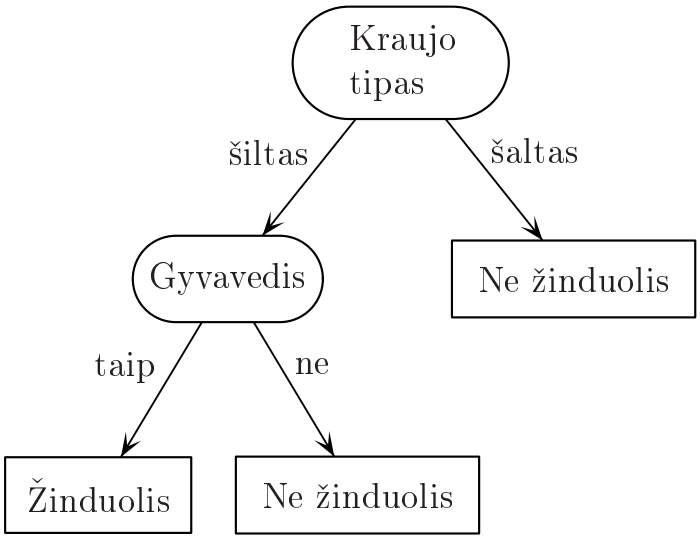
		Prognozuojama klasė	
		0	**1**
Tikroji klasė	**0**	n_{00}	n_{01}
	1	n_{10}	n_{11}

3 .2 lentelė. Binarinio klasifikavimo nesutapimų matrica

3.2 Sprendimų medžiai

Viena iš paprastesnių klasifikavimo modelio (trumpiau sakant, klasifikatoriaus) formų yra *sprendimų medis*. Dėl savo paprastumo ir vaizdumo sprendimų medžiai gana plačiai naudojami. Kaip jie atrodo ? Iliustracijai pasitelksime pavyzdį apie stuburinių gyvūnų klasifikavimą (žr. 3 .1 lent.). Uždavinį šiek tiek supaprastinsime. Visus stuburinius skirstysime ne į penkias, o tik į dvi skirtingas kategorijas: žinduolius ir ne žinduolius. Kitaip sakant, turėsime binarinį klasės kintamąjį.

Tarkime, aptiktas iki šiol nežinomas gyvūnas. Žinduolis jis ar ne ? Aišku, kad tai priklauso nuo atsakymų į keletą klausimų apie šį gyvūną. Pirmas klausimas galėtų būti apie jo kūno temperatūrą. Jei gyvūnas šaltakraujis, jis tikrai ne žinduolis. Priešingu atveju jis yra paukštis arba žinduolis ir todėl šį savotišką žaidimą "Taip-Ne" tęsiame toliau. Skaitytojas, kuris yra žaidęs tokį žaidimą, jau suprato, kad norint greitai laimėti, reikia protingai formuluoti klausimus. Žiūrint į 3 .1 lentelės duomenis, tinkamas klausimas būtų: ar paslaptingasis gyvūnas yra gyvavedis ? Taip - žinduolis, ne - paukštis. Nelabai vykusio klausimo pavyzdys: ar gyvūnas turi kojas ? Akivaizdu, kad išgirdus atsakymą "taip", tektų žaidimą tęsti toliau. Kaip matome, klasifikavimo problemą galima spręsti atsakant į eilę gerai apgalvotų klausimų apie nagrinėjamo įrašo atributų reikšmes. Be to, kiekvienas sekantis klausimas priklauso nuo atsakymo į prieš tai užduotą klausimą. Kitaip sakant, klausimai - atsakymai sudaro hierarchinę struktūrą, kurią galima pavaizduoti *sprendimų medžiu*. Jo šaknis ir visos vidinės viršūnės atspindi pateiktus klausimus, šakos atitinka galimus atsakymus, o lapuose randasi klasės kintamojo reikšmės. Tuo būdu, perėję klausimų - atsakymų grandinę nuo šaknies iki lapo atrandame klasę, kuriai priklauso nagrinėjamas įrašas. 3 .2 paveiksle pavaizduotas aukščiau išnagrinėto pavyzdžio sprendimų medis, klasifikuojantis gyvūnus į žinduolius ir ne žinduolius.



3 .2 pav. Žinduolių klasifikavimo sprendimų medis

Sukonstravus sprendimų medį, pati klasifikavimo procedura tampa labai paprasta. Pavyzdžiui, anksčiau jau sutiktas šiurpusis nuodadantis labai nesunkiai klasifikuojamas kaip ne žinduolis, nes yra šaltakraujis.
Lieka atsakyti į du "paprastus" klausimus. Kaip suformuluoti gerus klausimus ir kaip įvertinti klasifikatoriaus patikimumą? Toliau apie tai ir pakalbėsime.

3.2.1 Sprendimų medžių konstravimas

Galimų sprendimų medžių skaičius yra eksponentinis atributų kiekio atžvilgiu. Todėl rasti optimalų medį tiesioginio perrinkimo būdu didesniam kintamųjų skaičiui yra praktiškai neišsprendžiama problema. Tačiau yra efektyvūs algoritmai, leidžiantys konstruoti, nors ir ne visada optimalius, tačiau pakankamai tikslius sprendimų medžius. Paprastai tokie algoritmai "augina" medį, kiekviename žingsnyje optimaliai parinkdami kintamuosius tolesniam imties skaidymui. Reikia pažymėti, kad optimalumo kriterijai gali būti įvairūs ir pasirenkami, priklausomai nuo sprendžiamo uždavinio. Vienas iš tokių yra, dar 1966 metais paskelbtas ir klasikiniu tapęs, E.B.Hunt'o algoritmas. Jo pagrindu buvo konstruojami daugelis vėlesnių algoritmų, pavyzdžiui, ID3, C4.5, CART. Šiame skyrelyje aptarsime ir pavyzdžiais iliustruosime bendruosius Hunt'o algoritmo principus.

Hunt'o algoritmas

Hunt'o algoritmas konstruoja sprendimų medį rekursiškai skaidydamas mokymo imtį į mažesnes dalis. Tarkime D_T yra su medžio viršūne T susijusi mokymo imties įrašų aibė, o A_Y - visų klasių aibė. Kitaip sakant, A_Y yra sudaryta iš visų galimų priklausomo (klasės) kintamojo Y reikšmių. Algoritmą sudaro du žingsniai.

H1. Jei visi aibėje D_T esantys įrašai priklauso vienai klasei $y_T \in A_Y$, tai viršūnė T yra lapas, žymintis klasę y_T.

H2. Jei aibėje D_T yra įrašų, priklausančių skirtingoms klasėms, tai T tampa vidine medžio viršūne, kurios vaikams priskiriami aibės D_T poaibiai. Poaibių skaičius ir sudėtis priklauso nuo pasirenkamos atributų reikšmių tikrinimo sąlygos, t.y. nuo suformuluoto klausimo. Toliau algoritmas kartojamas kiekvienam viršūnės T vaikui.

Antrąjame žingsnyje atributų tikrinimo sąlygų parinkimas nedetalizuojamas ir priklauso nuo konkrečios algoritmo modifikacijos. Šią problemą aptarsime vėliau. o kol kas algoritmą iliustruosime tokiu pavyzdžiu.

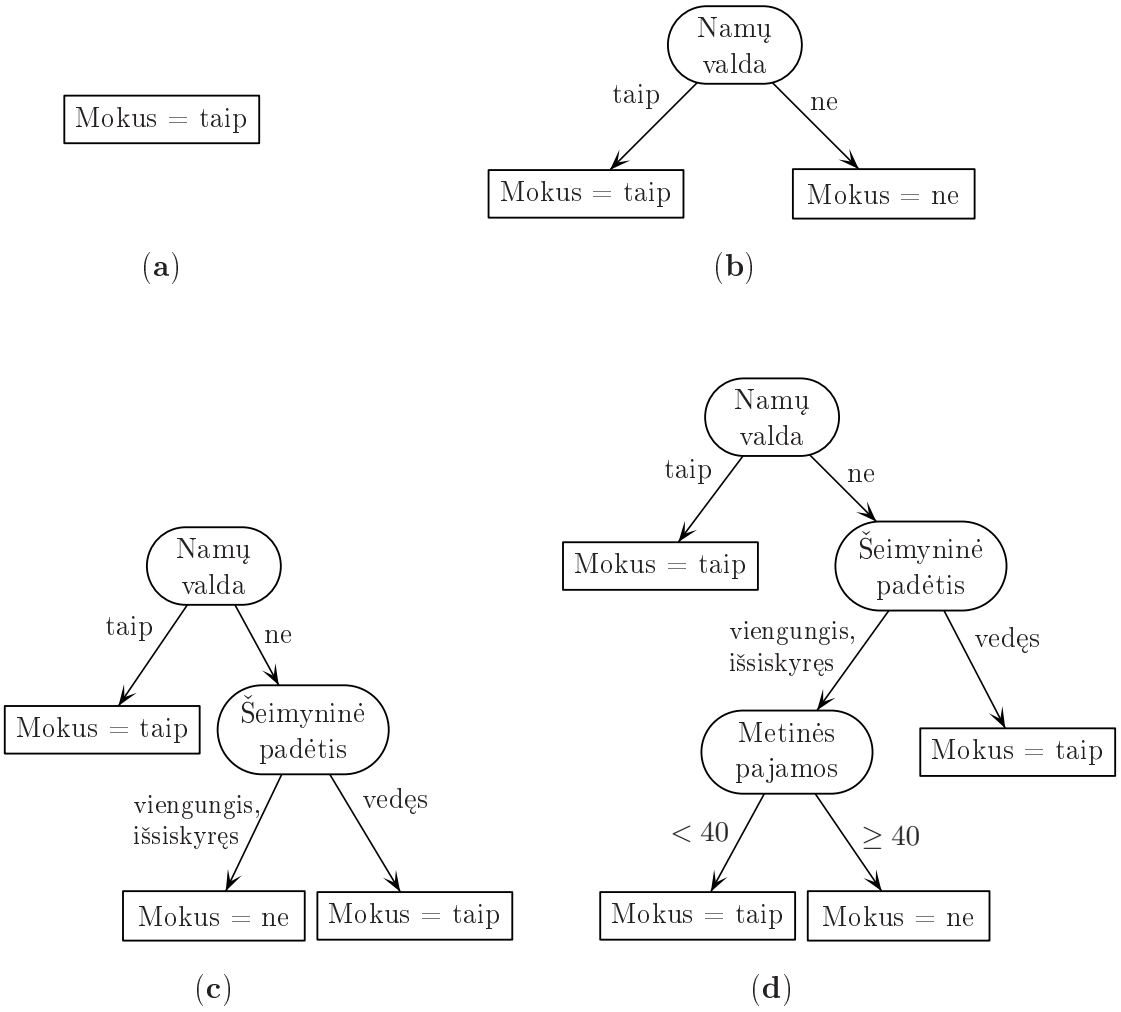
3.2.1 pavyzdys. Prieš skirdamas paskolą, bankas analizuoja ar potencialus klientas bus mokus, t.y. laiku mokės periodines įmokas. Šiuo atveju mokymo imtį sudaro banko turima informacija apie 10 ankstesnių jo klientų.

	Namų valda (X_1)	Šeimyninė padėtis(X_2)	Metinės pajamos (tūkst.Lt)(X_3)	Mokus klientas(Y)
1	taip	vedęs	65	taip
2	ne	vedęs	50	taip
3	taip	viengungis	35	taip
4	taip	vedęs	60	taip
5	ne	išsiskyręs	47	ne
6	ne	viengungis	30	taip
7	taip	išsiskyręs	110	taip
8	ne	viengungis	42	ne
9	taip	vedęs	37	taip
10	ne	viengungis	45	ne

3 .3 lentelė. Banko klientai

3 .3 lentelėje pateikiama tokia informacija apie banko klientus: turi ar neturi savo vardu registruotą namą arba butą, šeimyninė padėtis, metinės pajamos, mokumas (ar tvarkingai grąžino paskolą).
Algoritmas pradeda darbą nuo pradinio medžio, sudaryto iš vienos viršūnės, pažymėtos `Mokus=taip` (žr.3 .3(a) paveikslą).

Tai reiškia, kad dauguma buvusių klientų atsiskaitė tvarkingai. Tačiau medį reikia išplėsti, nes mokymo imtyje yra abiems klasėms priklausančių įrašų. Todėl visi klientai dalijami į dvi dalis pagal kintamojo X_1 (`Namų valda`) galimas reikšmes, kaip pavaizduota 3 .3(b) paveiksle. Kodėl būtent pagal X_1 ? Kaip jau buvo minėta, atributų parinkimo problemą



3 .3 pav. Hunt'o algoritmas sprendimų medžio konstrukcijai

nagrinėsime vėliau. Dabar tiesiog manysime, kad čia siūlomi imties skaidiniai yra geriausi. Toliau analizuojame abu šaknies vaikus. 3 .3 lentelėje matome, kad visi namų valdos savininkai sėkmingai grąžino paskolą. Todėl kairysis šaknies vaikas lieka lapu, atitinkančiu klasę $Y =$`taip` (žr. 3 .3(b) pav.). Tuo tarpu dešinysis vaikas skaidomas toliau, rekursiškai vykdant Hunt'o algoritmo H1 ir H2 žingsnius, kol gaunamas medis, kurio kiekvienas lapas atitinka tik vienos klasės įrašus. Tai pavyko padaryti po dviejų žingsnių. Po kiekvieno žingsnio kintantys medžiai pavaizduoti 3 .3(c) ir (d) paveiksluose.

Kad Hunt'o algoritmo aprašymas būtų pilnas, liko atskirai aptarti dvi išskirtines situacijas.

1. Kuriai nors viršūnei T, nėra ją atitinkančių įrašų, t.y. $D_T = \emptyset$. Tokiu atveju T

paskelbiama lapu, atitinkančiu klasę, dažniausiai sutinkamą aibėje $D_{T'}$, čia T' žymi viršūnės T tėvą.

2. Visi aibės D_T įrašai, turi vienodas nepriklausomų kintamųjų reikšmes, bet priklauso ne vienai klasei. Tai reiškia, kad H2 žingsnyje situacija pradėtų kartotis. Tokiu atveju T paskelbiama lapu, atitinkančiu klasę, dažniausiai sutinkamą aibėje D_T.

Bet kuris sprendimų medį konstruojantis mokymo algoritmas būtinai turi išspręsti tokias dvi problemas.

1. Kaip skaidyti mokymo imtį? Turi būti parinktas objektyvus ir pagrįstas skaidinio "gerumo matas", kiekviename žingsnyje leidžiantis pasirinkti geriausią skaidinį. Be to, formuluojant sąlygas, būtina atsižvelgti į atributų tipų skirtumus.

2. Kada sustoti? Pagal Hunt'o algoritmą, viršūnė nebeskaidoma tik kai visi jos įrašai priklauso vienai klasei arba turi vienodas atributų reikšmes. Tačiau, labai "šakotas" medis ne visada yra geras. Todėl reikalingi kriterijai, leidžiantys anksčiau sustabdyti medžio auginimo procesą.

3.2.2 Skaidymo būdai ir jų palyginimas

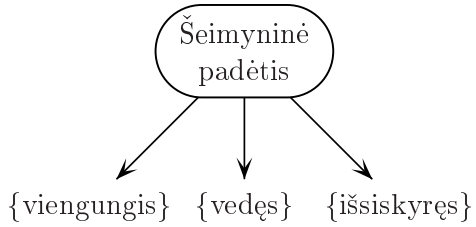
Kaip matėme, pagrindinis sprendimų medžio konstravimo algoritmo elementas yra mokymo imties įrašų skaidymas į dalis, priklausomai nuo pasirinktų atributų galimų reikšmių. Todėl tiek pčios sąlygos formulavimas, tiek galimi atsakymai (o tuo pačiu ir skaidomos viršūnės vaikų skaičius) priklauso nuo atributų tipo.

Binariniai kintamieji. Tai paprasčiausias kintamasis, kartais dar vadinamas "taip-ne" atributu. Medžio viršūnę skaidant tokio kintamojo atžvilgiu, visada gaunami du vaikai, vaizduojantys du galimus atsakymus. 3.2.1 pavyzdyje tokio tipo yra atributas X_1 ir klasės kintamasis Y, o medžio viršūnės skaidymas pagal X_1 pavaizduotas 3 .3(b) paveiksle.

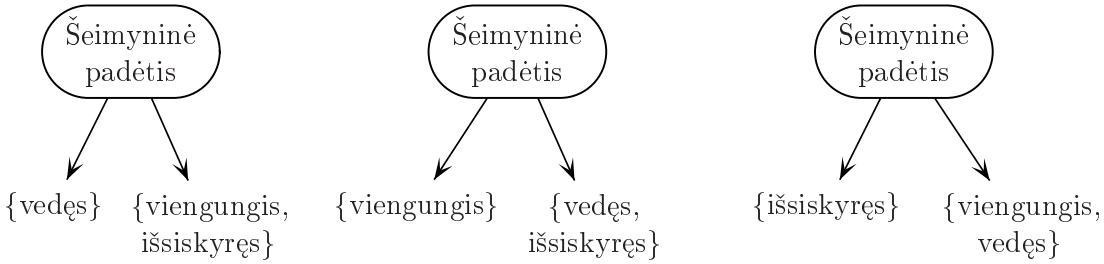
Vardiniai kintamieji. Jei vardinio kintamojo galimų reikšmių skaičius yra k, tai jo atžvilgiu turimą įrašų aibę galima suskaidyti $B_k - 1$ būdų. Čia B_k yra kombinatorikoje žinomi Belo skaičiai, nesunkiai randami iš rekurentinio sąryšio

$$B_{k+1} = \sum_{m=0}^{k} \binom{k}{m} B_m \,, \quad B_0 = 1 \,.$$

Iš viso turėsime $2^{k-1}-1$ binarinių skaidinių, o likusieji bus daugianariai. Pavyzdžiui, 3.2.1 pavyzdyje vardinis kintamasis X_2 (`Šeimyninė padėtis`) įgyja $k=3$ skirtingas reikšmes: `viengungis, vedęs, išsiskyręs`. Todėl X_2 atžvilgiu atitinkamą medžio viršūnę galima išskaidyti 4 būdais: vienas skaidinys yra daugianaris, o kiti 3 - binariniai (žr. 3.4(a),(b) paveikslus). Kai kurie algoritmai, pavyzdžiui CART, dirba tik su binariniais vardinių kintamųjų skaidiniais.



(**a**) Daugianaris skaidinys



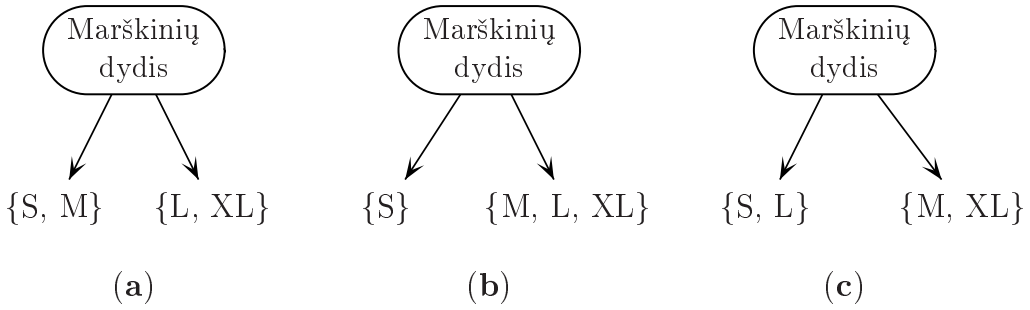
(**b**) Binariniai skaidiniai

3.4 pav. Skaidiniai pagal vardinį kintamąjį

Ranginiai kintamieji. Panašiai kaip ir vardiniai kintamieji, ranginiai atributai gali generuoti tiek binarinius tiek daugianarius skaidinius. Tik šiuo atveju, grupuojant reikšmes, dažniausiai atsižvelgiama į jų natūralią tvarką.

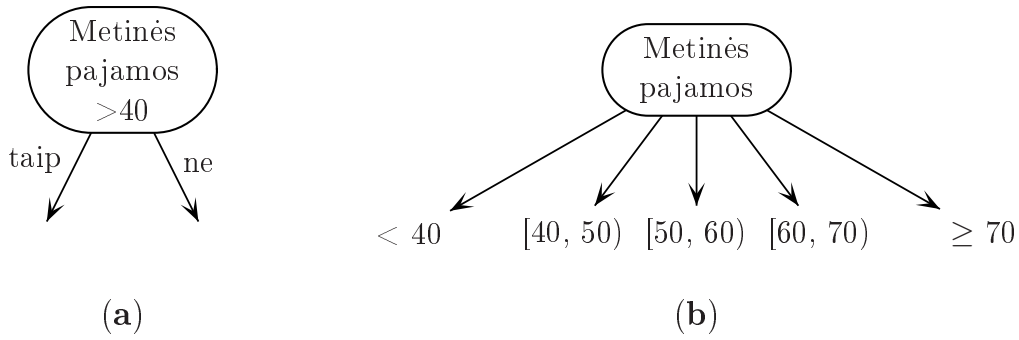
3.5 paveiksle pavaizduoti trys galimi imties įrašų grupavimo būdai ranginio kintamojo `Marškinių dydis` atžvilgiu. Natūrali tokio atributo reikšmių tvarka yra `S, M, L, XL`. Kaip matome, tik grupavimas, pavaizduotas 3.5(c) paveiksle, šią tvarką pažeidžia.

Tolydūs kintamieji. Mokymo imties įrašai grupuojami tolydžiojo kintamojo X atžvilgiu,



3 .5 pav. Ranginio kintamojo reikšmių grupavimas

prieš tai jį diskretizavus. Bendruoju atveju diskretizavimo procedūra buvo išnagrinėta 2.3.2 skyrelyje. Jei reikalingas binarinis skaidinys, galima paprasčiausiai tikrinti sąlygas $X < x$ arba $X \geq x$, tinkamai parinkus x. 3 .6 paveiksle pavaizduoti du skaidiniai kintamojo X_3 (`Metinės pajamos`) atžvilgiu (žr. 3 .3 lent.).



3 .6 pav. Skaidiniai pagal tolydųjį kintamąjį

Norėdami palyginti galimus skaidinius, sieksime nustatyti kuris iš jų suteikia daugiau aiškumo klasės kintamojo Y galimų reikšmių atžvilgiu, kitaip sakant, kada *neapibrėžtumas* yra mažiausias. Galimi įvairūs neapibrėžtumo matai, priklausantys nuo atstiktinio dydžio Y skirstinio. Paprastai lyginami neapibrėžtumo pokyčiai iki ir po padalinimo. Geriausiu pripažįstamas tas skaidinys, kuris labiausiai sumažina neapibrėžtumą.

Tegul T yra sprendimų medžio viršūnė, o galimų klasių aibė $A_Y = \{y_0, y_1, \ldots, y_{c-1}\}$.

Žymėsime

$$P_T(i) = P_T(Y = y_i)$$

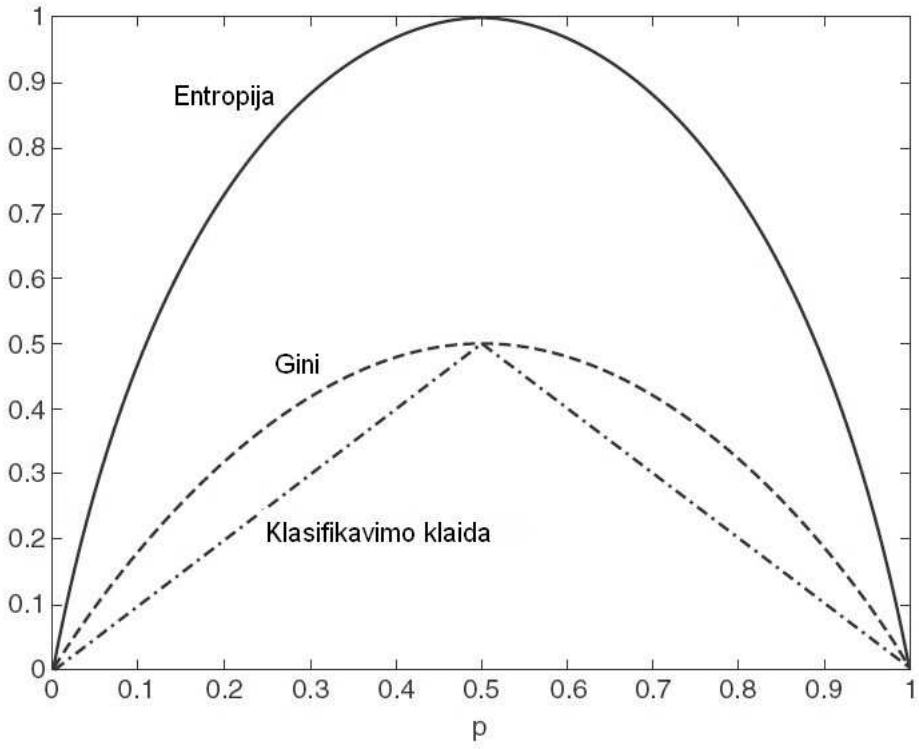
klasės y_i santykinį dažnį viršūnę T atitinkančioje įrašų aibėje D_T. Nagrinėsime tris neapibrėžtumą viršūnėje T apibūdinančius dydžius: entropiją $H_T(Y)$, Gini indeksą $G_T(Y)$ ir klasifikavimo klaidą $E_T(Y)$:

$$H_T(Y) = -\sum_{i=0}^{c-1} P_T(i) \log_2 P_T(i)\,, \tag{3.1}$$

$$G_T(Y) = 1 - \sum_{i=0}^{c-1} P_T^2(i)\,, \tag{3.2}$$

$$E_T(Y) = 1 - \max_{0 \le i \le c-1} P_T(i)\,. \tag{3.3}$$

Sprendžiant binarinio klasifikavimo uždavinį, šie neapibrėžtumo matai yra ekvivalentūs.



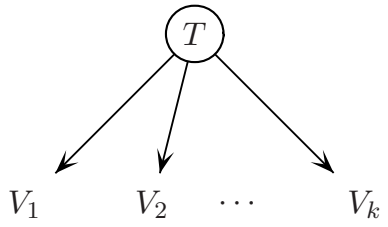
3.7 pav. Neapibrėžtumo charakteristikų palyginimas binariniam klasifikavimui

Jų grafikai pavaizduoti 3.7 paveiksle. Šiuo atveju tai yra kintamojo p funkcijos, čia

$$p = P_T(0) = 1 - P_T(1)$$

Kaip ir reikėjo tikėtis, didžiausias neapibrėžtumas gaunamas, kai abiejų klasių dažniai sutampa ($p = 0,5$). Kai visi įrašai priklauso vienai klasei (p lygu 0 arba 1), jokio neapibrėžtumo nelieka.

Tarkime, kad viršūnė T skaidoma kurio nors kintamojo X atžvilgiu ir įgyja k vaikų $V_1, V_2, \ldots, V_k$ (žr. 3 .8 pav.). Viršūnę T atitinkančių įrašų skaičių žymėsime $N(T)$. Aišku,



3 .8 pav. k - naris skaidinys

kad

$$N(T) = N(V_1) + N(V_2) + \cdots + N(V_k).$$

Norėdami nusakyti gautąjį neapibrėžtumo pokytį, pažymėkime $F_V(Y)$ pasirinktąjį neapibrėžtumo matą viršūnėje V. Pavyzdžiui, tai gali būti bet kuri iš (3 .1) - (3 .3) funkcijų. Tada vidutinis neapibrėžtumo pokytis bus matuojamas dydžiu

$$I_T(Y, X) = F_T(Y) - \sum_{j=1}^{k} \frac{N(V_j)}{N(T)} F_{V_j}(Y)\,. \tag{3 .4}$$

Kai $F_T(Y) = H_T(Y)$, prisiminę sąlyginės entropijos apibrėžimą 1.3.4, iš (3 .4) gausime, kad

$$I_T(Y, X) = H_T(Y) - H_T(Y|X)\,. \tag{3 .5}$$

Kitaip sakant, tai yra atsitiktinių dydžių X ir Y tarpusavio informacija viršūnėje T. Todėl šią viršūnę skaidome pagal atributą $\hat{X}$, suteikiantį daugiausiai informacijos apie klasės kintamąjį Y , t.y.

$$\hat{X} = \operatorname*{argmax}_X I_T(Y, X)\,.$$

Kadangi

$$\operatorname*{argmax}_X I_T(Y, X) = \operatorname*{argmin}_X H_T(Y|X)\,,$$

tai dažniausiai yra minimizuojama (3 .4) išraiškoje esanti suma, kuri reiškia vidutinį neapibrėžtumą po išskaidymo.

Prisiminkime 3.2.1 pavyzdį.

3.2.2 pavyzdys. Pažiūrėkime ar teisingai buvo pasirinktas pirmojo skaidymo kintamasis (X_1), konstruojant 3.2.1 pavyzdžio sprendimų medį. Paprastumo dėlei 3 .4 lentelėje kintamųjų X_1 ir Y reikšmes `ne, taip` užkoduosime 0 ir 1, o kintamojo X_2 reikšmes `viengungis, vedęs, išsiskyręs` pakeisime į 1, 2 ir 3 atitinkamai. Taip užrašyti banko klientų duomenys pavaizduoti 3 .4 lentelėje.

Kl.kodas	X_1	X_2	X_3	Y
1	1	2	65	1
2	0	2	50	1
3	1	1	35	1
4	1	2	60	1
5	0	3	47	0
6	0	1	30	1
7	1	3	110	1
8	0	1	42	0
9	1	2	37	1
10	0	1	45	0

3 .4 lentelė. Modifikuoti banko klientų duomenys

Konstruosime binarinį sprendimų medį pagal Gini indeksą. Kadangi iš 10 įrašų tik 3 priklauso nulinei klasei ($Y = 0$), tai šaknies T Gini indeksas

$$G_T(Y) = 1 - 0,3^2 - 0,7^2 = 0,42\,.$$

Binarinius skaidinius pagal kintamuosius X_1 ir X_2 atitinkantys skaičiavimai pateikiami 3 .9 paveiksle.

Geriausią binarinį skaidinį tolydžiojo kintamojo X_3 atžvilgiu rasime tikrindami sąlygą $X_3 < x$. Dalinimo tašką x rasime imdami tarpinius taškus tarp dviejų gretimų X_3 reikšmių 3 .4 lentelėje. Rinksimės tą x, kuriam skaidinys turės mažiausią Gini indeksą. Skaičiavimai ženkliai palengvėja, kai mokymo imties duomenys iš anksto surūšiuojami pagal kintamojo

	X_1	
	0	1
Y = 0	3	0
Y = 1	2	5
$G_V(Y)$	0,48	0
$G_T(Y\|X_1)$	0,24	
$I_T(Y,X_1)$	0,18	

	X_2	
	{1, 2}	{3}
Y = 0	2	1
Y = 1	6	1
$G_V(Y)$	0,375	0,5
$G_T(Y\|X_2)$	0,4	
$I_T(Y,X_2)$	0,02	

	X_2	
	{1, 3}	{2}
Y = 0	3	0
Y = 1	3	4
$G_V(Y)$	0,5	0
$G_T(Y\|X_2)$	0,3	
$I_T(Y,X_2)$	0,12	

	X_2	
	{2, 3}	{1}
Y = 0	1	2
Y = 1	5	2
$G_V(Y)$	0,27778	0,5
$G_T(Y\|X_2)$	0,366666667	
$I_T(Y,X_2)$	0,053333333	

3 .9 pav. Binariniai skaidiniai pagal kategorinius kintamuosius

X_3 reikšmes. Rezultatai pateikiami 3 .10 paveiksle. Kaip matome, mažiausias Gini indeksas, o tuo pačiu ir didžiausias neapibrėžtumo sumažėjimas, gaunamas, kai $x = 48$.

Palyginę 3 .9 ir 3 .10 lentelėse matomus skaičiavimus, galime tvirtinti, kad binarinį sprendimų medį reikia pradėti konstruoti nuo skaidinio pagal X_1, nes

$$I_T(Y, X_1) > \max I_T(Y, X_2) = \max I_T(Y, X_3)\,.$$

Tai ir yra pavaizduota 3 .3(b) paveiksle.

Kai sprendimų medis nėra binarinis, renkantis skaidinį, reikia atsižvelgti ir į kintamojo reikšmių skaičių. Gali atsitikti taip, kad (3 .1) - (3 .3) neapibrėžtumo matai tinkamiausiais pripažins skaidinius pagal daugiausiai skirtingų reikšmių įgyjančius kintamuosius. Pavyzdžiui, jei 3 .4 lentelės įrašų skaidymui pasirinksime atributą $X_0 =$ "*Kl.kodas*", gausime, kad visiems $j = 1, 2, \ldots, 10$

$$H_{V_j}(Y) = G_{V_j}(Y) = E_{V_j}(Y) = 0\,,$$

Y	1	1	1	0	0	0	1	1	1	1
X_3	30	35	37	42	45	47	50	60	65	110

x	25		32		36		40		43		46		48		55		62		90		120	
	<	>=	<	>=	<	>=	<	>=	<	>=	<	>=	<	>=	<	>=	<	>=	<	>=	<	>=
Y = 0	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
Y = 1	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
$G_Y(Y)$	0	0,42	0	0,44	0	0,47	0	0,49	0,38	0,44	0,48	0,32	0,5	0	0,49	0	0,47	0	0,44	0	0,42	0
$G_T(Y\|X_3)$	0,42		0,4		0,375		0,34286		0,41667		0,4		**0,3**		0,34286		0,375		0,4		0,42	
$I_T(Y,X_3)$	0		0,02		0,045		0,07714		0,00333		0,02		**0,12**		0,07714		0,045		0,02		0	

3 .10 pav. Binariniai skaidiniai pagal tolygųjį kintamąjį

nes kiekvieną vaiką atitiks tik vienas įrašas. Todėl

$$I_T(Y, X_0) > I_T(Y, X_1).$$

Tačiau toks medis visiškai netinkamas būsimų įrašų klasifikavimui, nes kliento kodas yra unikalus. Kai kurie algoritmai, pavyzdžiui C4.5, neapibrėžtumo pokyčiui matuoti vietoje tarpusavio informacijos (3 .5), naudoja *santykinę tarpusavio informaciją*

$$\tilde{I}_T(Y, X) = \frac{I_T(Y, X)}{H(V)}\,.$$

Čia $H(V)$ žymi viršūnės T skaidinio, pavaizduoto 3 .8 paveiksle, entropiją

$$H(V) = -\sum_{j=1}^{k} \frac{N(V_j)}{N(T)} \log_2 \frac{N(V_j)}{N(T)}\,.$$

Tokia modifikacija leidžia neutralizuoti per daug susmulkinto skaidinio įtaką. Pavyzdžiui, jau minėtiems banko klientų duomenims turėsime

$$\begin{aligned}
I_T(Y, X_0) &= H_T(Y) - H_T(Y|X_0) = h(0,3) - 0 \approx 0,8813\,,\\
\tilde{I}_T(Y, X_0) &= \frac{I_T(Y, X_0)}{\log_2 10} \approx 0,2653\,,\\
I_T(Y, X_1) &= H_T(Y) - H_T(Y|X_1) = h(0,3) - (0,5\,h(0,4) + 0,5\,h(0)) \approx 0,3958\,,\\
\tilde{I}_T(Y, X_1) &= \frac{I_T(Y, X_1)}{\log_2 2} \approx 0,3958\,.
\end{aligned}$$

Renkamės skaidinį pagal X_1, nes $\tilde{I}_T(Y, X_1) > \tilde{I}_T(Y, X_0)$.

Apibendrindami tai kas pasakyta, pateiksime sprendimų medį konstruojančio algoritmo schemą. Pavadinkime jį `AuginkMedi` (žr. 3.11 pav.). Algoritmas rekursyviai konstruoja medį, pagal mokymo imties įrašus E ir jų atributų aibę F . Jis renkasi geriausią skaidinį (7

```
AuginkMedi(E, F)
 1.  if stopSąlyga(E, F) = True then
 2.     lapas = naujaViršūnė()
 3.     lapas.vardas = klasifikavimas()
 4.     return lapas
 5.  else
 6.     viršūnėT = naujaViršūnė()
 7.     viršūnėT.sąlyga= geriausiasSkaidinys(E, F)
 8.     V = {v | v yra galimas viršūnėT.salyga tikrinimo rezultatas}
 9.     for v ∈ V do
10.        E_v = {e | viršūnėT.salyga(e)= v  ∧  e ∈ E}
11.        vaikas= AuginkMedi(E_v, F)
12.        medis papildomas viršūne vaikas, kurios tėvas viršūnėT,
           jas jungianti briauna (vaikas→viršūnėT) žymima v
13.     end for
14.  end if
15.  return viršūnėT
```

3.11 pav. Sprendimų medį konstruojantis algoritmas

žingsnis) ir skaido jau turimo medžio lapus (11 ir 12 žingsniai), kol netenkinama pabaigos sąlyga (1 žingsnis). Konkreti algoritmo realizacija priklauso nuo keturių čia naudojamų procedūrų aprašymo.

1. Funkcija `naujaViršūnė()` prijungia prie jau turimo medžio naują viršūnę, tarkime, *nviršūnė*. Jei tai yra lapas, jis žymi klasę *nviršūnė.vardas*. Priešingu atveju *nviršūnė.sąlyga* reiškia skaidinio, kurį vaizduoja *nviršūnė*, sąlygą.

2. Funkcija `geriausiasSkaidinys()` randa sąlygą, pagal kurią turi būti skaidomi mokymo imties įrašai. Kaip jau minėjome, tai priklauso nuo pasirinkto neapibrėžtumo

mato. Čia dažnai naudojami entropija (3 .1) ir Gini indeksas (3 .2).

3. Funkcija `klasifikavimas`() randa lapą žyminčią klasę. Dažniausiai tai yra klasė, kuriai priklauso dauguma lapą *lapas* atitinkančių mokymo imties įrašų:

$$lapas.vardas = y_{i_{\max}}\,, \quad i_{\max} = \underset{i}{\operatorname{argmax}}\, P_{lapas}(i)\,.$$

Kartais dažniai $P_{lapas}(i)$ dar naudojami įvertinti tikimybėms, kad viršūnei *lapas* priskirtas įrašas yra klasėje y_i.

4. Funkcija `stopSąlyga`() naudojama medžio auginimo procesui sustabdyti. Tai reikėtų daryti, jei visi likę įrašai priklausytų vienai klasei arba turėtų vienodas atributų reikšmes. Kartais procesas stabdomas ir anksčiau. Pavyzdžiui, kai likusių įrašų skaičius pasidaro mažesnis už tam tikrą, iš anksto nustatytą ribą.

Mes aptarėme algoritmą, kuris kiekvienoje viršūnėje skaido medį pagal vieną kurį nors atributą X_j. Tačiau galimi ir kitokie skaidymo būdai. Bendruoju atveju viršūnėje T_i galima formuluoti sąlygą S_i priklausančią nuo daugelio atributų $S_i = S_i(X_1, X_2, \ldots, X_k)$. Tada jau imties įrašus dalijančios "linijos" nebebus lygiagrečios "koordinatinėms ašims". Todėl optimalių sąlygų S_i paieška kiekvienoje viršūnėje darosi labai sudėtinga. Šį uždavinį galima supaprastinti įvedant papildomus sudėtinius atributus. Iš anksto apsisprendus kokios sudėtinės salygos bus formuluojamos konstruojant medį, prie turimų atributų $X_1, X_2, \ldots, X_k$ prijungiamos kai kurios jų funkcijos $X' = F(X_1, X_2, \ldots, X_k)$. Pavyzdžiui, jei imties įrašai gerai atskiriami lyginant atributų X_1 ir X_7 reikšmes tarpusavyje, tai įvedę naują kintamąjį

$$X' = X_1 - X_7\,,$$

galėsime tikrinti sąlygas $(X' = 0)\,,\ (X' < 0)$ ir t.t. Kitaip sakant, galėsime taikyti mūsų jau aptartus algoritmus. Tokios metodikos trūkumas akivaizdus - imtyje atsiranda perteklinė informacija, nes nauji kintamieji yra jau esamų atributų funkcijos. Tačiau šį trūkumą "kompensuos" geras kompiuteris...

Jei sukonstruotas medis yra per daug "šakotas", gali pasireikšti nepageidautinas vadinamasis modelio *perteklumo* efektas. Tada medį tenka mažinti, jį sutraukiant. Apie tai ir kalbėsime kitame skyrelyje.

3.3 Klasifikatoriaus charakteristikos

Sprendžiant klasifikavimo uždavinį, skiriamos dvejų tipų klaidos: *mokymo klaidos* ir *modelio klaidos*. Mokymo klaida tai neteisingai klasifikuotų mokymo imties įrašų dalis. Tuo tarpu modelio klaida yra lygi neteisingo naujų įrašų klasifikavimo tikimybei.

3.3.1 Modelio perteklumas

Geras modelis turi ne tik atitikti mokymo imties duomenis, bet ir teisingai klasifikuoti naujus duomenis. Kai mokymo ir modelio klaidos yra didelės, sakome, kad modelis *nepakankamas*. Taip gali atsitikti, pavyzdžiui, kai duomenis klasifikuojantis sprendimų medis yra per mažas. Jį didinant mokymo klaida mažėja. Tačiau modelio klaida, labai išplėtus medį, gali pradėti didėti. Kitaip sakant, "per daug gerai" imties duomenis atitinkančio modelio klaida gali būti didesnė nei paprastesnio modelio su didesne mokymo klaida. Tokiu atveju sakome, kad tas sudėtingesnis modelis yra *perteklus*. Perteklumo priežastys gali

Pavadinimas	Kraujo tipas (X_1)	Gyvavedis (X_2)	Keturkojis (X_3)	Žinduolis (Y)
dygliuotis	šiltas	taip	taip	taip
katė	šiltas	taip	taip	taip
šikšnosparnis	šiltas	taip	ne	ne*
banginis	šiltas	taip	ne	ne*
salamandra	šaltas	ne	taip	ne
komodo varanas	šaltas	ne	taip	ne
pitonas	šaltas	ne	ne	ne
lašiša	šaltas	ne	ne	ne
erelis	šiltas	ne	ne	ne
gupija	šaltas	taip	ne	ne

3.5 lentelė. Žinduolių klasifikavimo mokymo imtis (* - klaida)

būti įvairios: imties nepakankamumas, dalies įrašų iškraipymai, netinkamas modelio konstravimo algoritmo parametrų parinkimas ir t.t. Pavyzdžiui, 3.12 paveiksle pavaizduoti du medžiai, sukonstruoti pagal 3.5 lentelės duomenis. Du šios imties įrašai yra klaidingi: iš

tikrųjų šikšnosparnis ir banginis yra žinduoliai.

Pavadinimas	Kraujo tipas (X_1)	Gyvavedis (X_2)	Keturkojis (X_3)	Žinduolis (Y)
žmogus	šiltas	taip	ne	taip
balandis	šiltas	ne	ne	ne
dramblys	šiltas	taip	taip	taip
tigrinis ryklys	šaltas	taip	ne	ne
vėžlys	šaltas	ne	taip	ne
pingvinas	šaltas	ne	ne	ne
ungurys	šaltas	ne	ne	ne
delfinas	šiltas	taip	ne	taip
echidna	šiltas	ne	taip	taip
šiurpusis nuodadantis	šaltas	ne	taip	ne

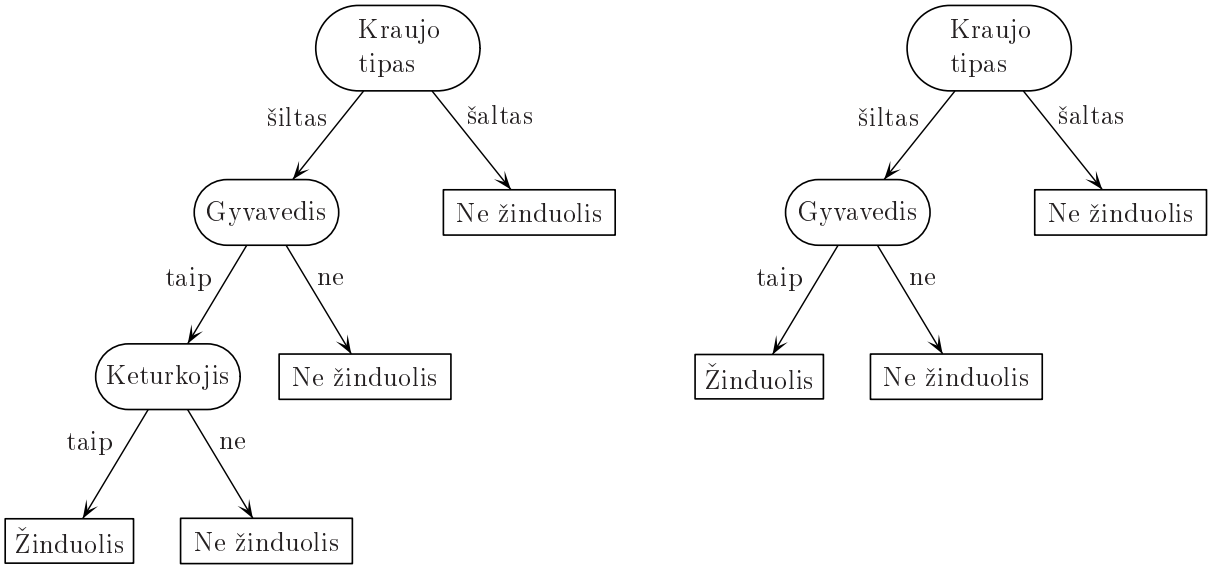
3 .6 lentelė. Žinduolių klasifikavimo kontrolinė imtis

3 .12(a) paveiksle pavaizduotas medis idealiai atitinka turimą mokymo imtį, kitaip sakant modelio M1 mokymo klaida lygi 0. Tačiau medis M1 neteisingai klasifikuoja 3 kontrolinės imties, matomos 3 .6 lentelėje, įrašus. Galime manyti, kad M1 modelio klaida yra apie 30% ! Kodėl taip atsitiko ? Visi trys žmogaus ir delfino atributai

$$X_1(Kraujotipas),\ X_2(Gyvavedis),\ X_3(Keturkojis)$$

turi tokias pat reikšmes, kaip šikšnosparnio ir banginio, kuriems mokymo imtyje klasė pažymėta neteisingai. Trečiasis neteisingai klasifikuotas gyvūnas yra echidna. Tai yra išskirtinis atvejis, nes jis klasifikuojamas kitaip nei panašus mokymo imties įrašas (erelis). Beje, su išskirtiniais atvejais susijusių klaidų dažniausiai išvengti neįmanoma. Todėl nereikėtų tikėtis nulinės modelio klaidos.

Šiek tiek sutraukus M1, gaunamas 3 .12(b) paveiksle pavaizduotas paprastesnis medis M2. Nesunku įsitikinti, kad nors mokymo klaida ir padidėjo (modeliui M2 ji lygi 0,1), tačiau M2 teisingai klasifikuoja 8 iš 10 kontrolinės imties įrašus. Tai rodo modelio M1 perteklumą.



(**a**) Medis M1 (**b**) Medis M2

3 .12 pav. Žinduolių klasifikavimo sprendimų medžiai pagal 3 .5 lentelės duomenis

3.3.2 Modelio klaidos įverčiai

Jau įsitikinome, kad modelio sudėtingumo augimas skatina jo perteklumo atsiradimą. Todėl reikia konstruoti tokio sudėtingumo modelį, kuris duoda mažiausią modelio klaidą. Perfrazavus žinomą A.Enšteino posakį, galėtume pasakyti: modelis turi būti paprastas kiek tik galima, bet ne daugiau.

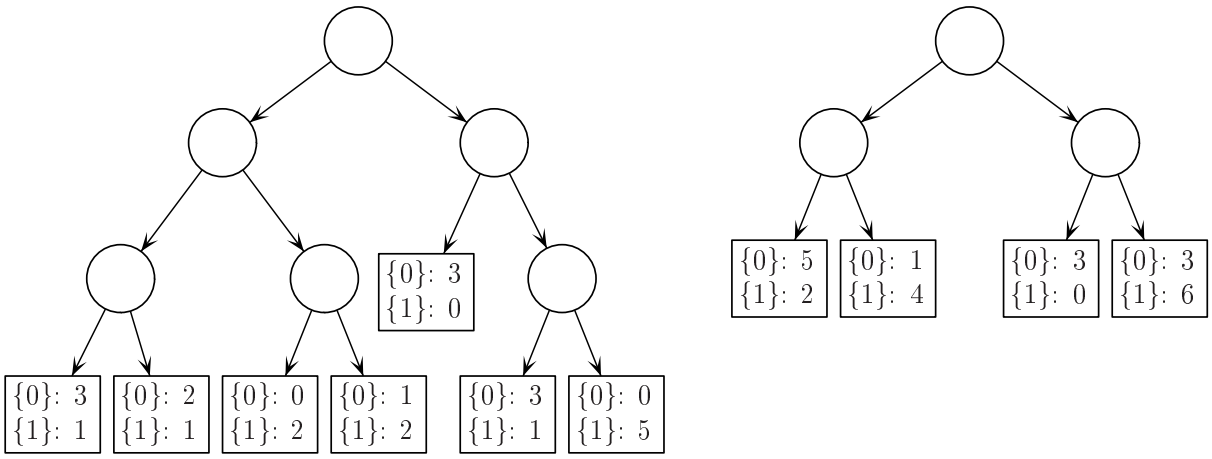
Bet kuris klasifikatoriaus konstravimo algoritmas naudoja tik mokymo imties duomenis ir negali tiksliai numatyti kaip sukonstruotas modelis T "elgsis" su naujais įrašais. Kitaip sakant, mes galime tiesiogiai apskaičiuoti mokymo klaidą $e_m = e_m(T)$, bet ne modelio klaidą $e_M = e_M(T)$. Todėl tenka pasitenkinti e_M įverčiais $\hat{e}_M$. Keletą tokių įverčių radimo metodų čia ir aptarsime.

1. Paprastasis įvertis. Jei mokymo imtis pakankamai tiksliai atspindi visą populiaciją, tai galime manyti, kad mokymo klaida yra apytiksliai lygi modelio klaidai, t.y.

$$\hat{e}_M = e_m \,.$$

Esant tokiai prielaidai, tiesiog konstruojamas mažiausią mokymo klaidą turintis klasifikatorius.

3.3.1 pavyzdys. Nagrinėsime 3 .13 paveiksle pavaizduotus binarinius sprendimų medžius T_1 ir T_2. Pastebėsime, kad T_2 yra paprastesnis, nes jis gautas sutraukus T_1. Tarsime, kad abu medžiai sukonstruoti pagal tą pačią mokymo imtį, kurios įrašai gali priklausyti klasei 0 arba 1. Kiekvienas lapas klasifikuojamas pagal daugumos taisyklę. Turėsime tokius abiejų



Medis T_1 Medis T_2

3 .13 pav. Vienos imties generuoti du sprendimų medžiai

modelių klaidų įverčius

$$\begin{aligned} \hat{e}_M(T_1) &= e_m(T_1) = \frac{4}{24} \approx 0,1667\,, \\ \hat{e}_M(T_2) &= e_m(T_2) = \frac{6}{24} = 0,25\,. \end{aligned}$$

Iš čia išplauktų, kad medžiu T_1 nusakomas klasifikatorius yra geresnis.

Reikia pažymėti, kad paprastasis modelio klaidos įvertis yra retai naudojamas, nes jis neatspindi modelio sudėtingumo, tuo pačiu ir galimo perteklumo.

2. Pesimistinis modelio klaidos įvertis. Šuo atveju modelio klaidos įvertis gaunamas prie mokymo įverčio pridedant tam tikrą modelio sudėtingumo mokestį. Sprendimų medyje toks mokestis gali būti nustatomas kiekvienam lapui. Tegul medžio T lapai yra $V_1, V_2, \ldots, V_l$. Tada

$$\hat{e}_M(T) = e_m(T) + \frac{1}{N(T)} \sum_{j=1}^{l} \delta(V_j)\,.$$

Čia $N(T)$ - mokymo imties įrašų skaičius, o $\delta(V_j)$ žymi sudėtingumo mokestį lapui V_j.

3.3.2 pavyzdys. Rasime pesimistinius modelio klaidos įverčius 3.13 paveiksle pavaizduotiems medžiams. Tegul sudėtingumo mokestis kiekvienam lapui V_j yra $\delta(V_j) = 0,5$. Tada

$$\begin{aligned} \hat{e}_M(T_1) &= \frac{4}{24} + \frac{1}{24} \cdot 7 \cdot 0,5 = 0,3125\,, \\ \hat{e}_M(T_2) &= \frac{6}{24} + \frac{1}{24} \cdot 4 \cdot 0,5 \approx 0,3333\,. \end{aligned}$$

Matome, kad medis T_1 lieka geresnis.

Binariniam medžiui T sudėtingumo mokestis $0,5$ reiškia, kad bet kuris lapas toliau skaidomas, jei tuo padidinamas teisingai klasifikuojamų įrašų skaičius. Iš tikrųjų, skaidydami bet kurį lapą, lapų skaičių didiname vienetu, tuo pačiu padidindami sudėtingumo mokestį dydžiu $\frac{0,5}{N(T)}$. Tačiau sumažėjęs neteisingai klasifikuojamų įrašų skaičius mažina mokymo klaidos įvertį ne mažiau kaip $\frac{1}{N(T)}$. Tai reiškia, kad pesimistinis modelio klaidos įvertis po skaidymo sumažės.

Analogiški samprotavimai rodo, kad pasirinkus lapo sudėtingumo mokestį lygų 1, bus skatinami tik skaidiniai, didinantys teisingai klasifikuojamų įrašų skaičių bent dviem įrašais. Pastebėsime, kad pasirinkus tokį sudėtingumo mokestį, geresniu turėtų būti pripažintas medis T_2, nes

$$\hat{e}_M(T_2) = \frac{10}{24} < \frac{11}{24} = \hat{e}_M(T_1)\,.$$

3. MDL įvertis. Tai dar vienas įvertis, leidžiantis atsižvelgti į modelio sudėtingumą. Jis remiasi informacijos teorijoje naudojama minimalaus aprašymo ilgio (Minimum Description Length) sąvoka. Pateiksime tokią šios sąvokos interpretaciją.

Tegul duomenys nusakomi atributais $\mathbf{X} = (X_1, X_2, \ldots, X_k)$ ir klasės kintamuoju Y. Sudaryta N įrašų imtis $E = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$. Tarkime Jonas žino visą imtį, o jo geras draugas Petras žino tik atributų reikšmes $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. Jonas rašo savo draugui laišką, norėdamas pranešti kaip klasifikuojami imties įrašai. Paprasčiausias būdas - parašyti kokiai klasei priklauso kiekvienas įrašas. Tada pranešimo ilgis L bus $L = O(N)$ bitų. Tačiau yra ir kitas būdas. Jonas gali sukonstruoti kokį nors klasifikatorių T (pavyzdžiui, medį), kuriam užrašyti reikia $L(T)$ bitų. Jei modelio T mokymo klaida lygi 0, tai, žinodamas T, Petras teisingai klasifikuos visus įrašus. Priešingu atveju Jonas dar turės papildomai

pranešti kuriuos įrašus modelis T klasifikuoja neteisingai. Tam reikalingą papildomą informacijos kiekį žymėsime $L(E|T)$. Tad galutinis pranešimo ilgis $L(E,T)$ bus

$$L(E,T) = L(T) + L(E|T)\,. \tag{3.6}$$

Pagal MDL principą renkamės tą modelį T_{min}, kuriam perduoti reikia trumpiausio pranešimo, t.y.

$$T_{min} = \operatorname*{argmin}_T L(E,T)\,.$$

Pastarasis samprotavimas turi gana aiškų tikimybinį pagrindą. Tarkime E ir T žymi imties sudarymo ir modelio parinkimo įvykius. Akivaizdu, kad turi būti konstruojamas pagal turimus duomenis labiausiai tikėtinas modelis. Kitaip sakant, toks kuriam tikimybė $P(T|E)$ didžiausia. Pagal Bajeso formulę

$$P(T|E) = \frac{P(T)\,P(E|T)}{P(E)}\,.$$

Abi šios lygybės puses logaritmuojame ir pakeičiame visų reiškinių ženklus. Turėsime

$$-\log_2 P(T|E) = -\log_2 P(T) - \log_2 P(E|T) + \log_2 P(E)\,.$$

Prisiminę informacijos apibrėžimą 1.3.1 ir jos interpretaciją (žr. 1.3.1 pavyzdį), gausime

$$-\log_2 P(T|E) = L(T) + L(E|T) + \log_2 P(E) = L(E,T) + \log_2 P(E)\,.$$

Pastebėsime, kad $\log_2 P(E)$ nepriklauso nuo T. Todėl iš pastarosios lygybės išplaukia, kad T atžvilgiu maksimizuodami $P(T|E)$, turėsime minimizuoti $L(E,T)$, t.y.

$$\operatorname*{argmax}_T P(T|E) = \operatorname*{argmin}_T L(E,T) = T_{min}\,.$$

Gauname tą patį MDL įvertį.

Praktinis $L(T)$ skaičiavimo būdas priklauso nuo modelio struktūros. Paprasčiau randamas $L(E|T)$ dydis. Neteisingai klasifikuotų įrašų skaičius yra lygus imties dydžiui N padaugintam iš mokymo klaidos $e_m(T)$, be to kiekvienai klaidai identifikuoti reikia $\log_2 N$ bitų. Todėl dažniausiai naudojama tokia $L(E|T)$ išraiška

$$L(E|T) = e_m(T) N \log_2 N\,.$$

3.3.3 pavyzdys. Grįžkime prie 3.13 paveiksle pavaizduotų medžių. Tarkime, kad juos atitinkantys duomenys turi 8 binarinius atributus. Kiekvieną vidinę medžio viršūnę nusakys ją atitinkantis skaidymo atributas. Jis nusakomas $\log_2 8 = 3$ bitais. Šiuo atveju yra tik dvi klasės. Todėl kiekvienas lapas mums kainuos $\log_2 2 = 1$ bitą. Šiek tiek paprastindami situaciją, manysime, kad medžio kaina lygi visų jo viršūnių kainų sumai[1]. Taigi

$$\begin{aligned} L(T_1) &= 6 \cdot 3 + 7 \cdot 1 = 25\,, \\ L(T_2) &= 3 \cdot 3 + 4 \cdot 1 = 13 \end{aligned}$$

Imties E dydis $N = 24$. Todėl

$$\begin{aligned} L(E|T_1) &= \frac{4}{24} \cdot 24 \cdot \log_2 24 \approx 18,34\,, \\ L(E|T_2) &= \frac{6}{24} \cdot 24 \cdot \log_2 24 \approx 27,51\,. \end{aligned}$$

Iš čia pagal MDL įvertį išplaukia, kad geresnis yra medis T_2, nes

$$L(E, T_1) \approx 43,34\,, \quad L(E, T_2) \approx 40,51\,.$$

4. Statistinis įvertis. Pakoreguosime jau išnagrinėtą paprastąjį įvertį. Tegul klasifikatoriaus T mokymo imtyje yra $N(T)$ įrašų. Tarsime, kad kiekvienas duomenų įrašas, nepriklausomai vienas nuo kito, klaidingai klasifikuojamas su tikimybe $p = e_M(T)$. Rasime šios tikimybės, o tuo pačiu ir modelio klaidos, pasikliautinojo intervalo viršutinį rėžį $e_v(T, Q)$. Tai ir bus statistinis modelio klaidos įvertis

$$\hat{e}_M(T) = e_v(T, Q)\,,$$

čia $Q \in (0,\ 1)$ žymi pasikliovimo lygmenį. Praktikoje Q kartais reiškiamas ir procentais. Esamos prielaidos leidžia tvirtinti, kad imties klaidingai klasifikuotų įrašų skaičius S_N yra binominis atsitiktinis dydis, kurio vidurkis ir dispersija yra

$$\mathbf{E}S_N = N(T)p\,, \quad \mathbf{D}S_N = N(T)p(1-p)\,.$$

Iš tikimybių teorijoje žinomos vadinamosios centrinės ribinės teoremos išplaukia, kad

$$P\left(-z_\alpha \le \frac{S_N - \mathbf{E}S_N}{\sqrt{\mathbf{D}S_N}} \le z_\alpha\right) \approx Q\,, \tag{3.7}$$

[1] Iš tikrųjų dar "keletą" bitų reikėtų pridėti viršūnių tarpusavio sąryšiams nusakyti. Tegul tai bus uždavinys skaitytojams.

čia

$$\alpha = \frac{1-Q}{2}\,,$$

o z_α žymi standartinio normalaus skirstinio $1-\alpha$ lygmens kvantilį. Keletą dažniau sutinkamų jo reikšmių galima rasti 3 .7 lentelėje.

Q	0,99	0,95	0,9	0,8	0,75	0,6	0,5	0,4	0,2
$\alpha = \frac{1-Q}{2}$	0,005	0,025	0,05	0,1	0,125	0,2	0,25	0,3	0,4
z_α	2,58	1,96	1,64	1,28	1,15	0,84	0,67	0,52	0,25

3 .7 lentelė. Standartinio normalaus skirstinio $1-\alpha$ lygmens kvantiliai

Įrašę $\mathbf{E}S_N$ ir $\mathbf{D}S_N$ reikšmes, o S_N pakeitę reiškiniu

$$e_m(T)N(T)\,,$$

(3 .7) formulėje esančią nelygybę galime perrašyti taip

$$-z_\alpha \le \frac{e_m(T)-p}{\sqrt{p(1-p)/N(T)}} \le z_\alpha\,.$$

Išsprendę pastarąją nelygybę p atžvilgiu, gausime tokį p intervalo viršutinį rėžį

$$e_v(T,Q) = \left(e_m(T) + \frac{z_\alpha^2}{2N(T)} + z_\alpha\sqrt{\frac{e_m(T)}{N(T)} - \frac{e_m^2(T)}{N(T)} + \frac{z_\alpha^2}{4N^2(T)}}\right)\left(1+\frac{z_\alpha^2}{N(T)}\right)^{-1}. \tag{3 .8}$$

Pastarąjį įvertį galima taikyti ir kaip lapo skaidymo kriterijų.

3.3.4 pavyzdys. 3 .13 paveiksle pavaizduoto medžio T_2 pirmąjį iš kairės lapą pažymėkime V. Matome, kad $N(V)=7$ ir $e_m(V)=2/7$. Pasirinkę pasikliovimo lygmenį $Q=75\%$, rasime lapo V modelio klaidos statistinį įvertį $e_v(V,0.75)$. Šiuo atveju $z_\alpha = z_{0,125} = 1,15$. Įrašę reikalingas reikšmes į (3 .8), gausime[1]

$$e_v(V,0.75) \approx 0,503\,.$$

Medyje T_1 viršūnė V jau išskaidyta į du naujus lapus. Pažymėkime juos V_1 ir V_2. Tada

$$N(V_1)=4\,,\ e_m(V_1)=\frac{1}{4}\,,\ N(V_2)=3\,,\ e_m(V_2)=\frac{1}{3}\,.$$

[1]Iš tikrųjų, taikant (3 .8) įvertį, pageidautina, kad $N(T)$ būtų didesnis

Todėl

$$e_v(V_1, 0.75) \approx 0,537 \;\; e_v(V_2, 0.75) \approx 0,65\,.$$

Jų vidutinė reikšmė ir bus viršūnės V modelio klaidos įvertis po išskaidymo

$$e_v^{'}(V, 0.75) = \frac{4}{7} e_v(V_1, 0.75) + \frac{3}{7} e_v(V_2, 0.75) \approx 0,585\,.$$

Kaip matome, jis yra didesnis už $e_v(V, 0.75)$. Todėl lapo V skaidyti nereikėtų.

3.3.3 Kryžminis patikrinimas

Iki šiol kalbėjome apie klaidų įverčius, kuriuos galima apskaičiuoti modelio konstravimo metu. Tokie įverčiai gali įtakoti tiek modelio pasirinkimą, tiek jo konstravimo algoritmą. Kai modelis jau sukonstruotas, galutinį jo vertinimą atliekame klasifikuodami kontrolinės aibės įrašus. Jei kontrolinė aibė yra pakankamai didelė, tai neteisingai klasifikuotų įrašų dalis (arba jos statistinis įvertis) atspindi tikrąjį modelio klaidos dydį. Daugiau problemų kyla, kai duomenų kiekis ribotas. Tada tenka turimą imtį dalinti į dalis skirtas mokymui ir kontrolei. Dabar ir aptarsime kaip geriau tai padaryti.

Mokymui ir kontrolei skirtų imties dalių proporcijos gali būti įvairios. Dažniausiai du trečdaliai mokymui ir trečdalis kontrolei. Tačiau atsitiktinai arba tiesiog "nesėkmingai" skaidant imtį, gali atsitikti taip, kad sudarytoji mokymo (arba kontrolinė) imtis nebus *reprezentatyvi*. Pavyzdžiui, visi kurios nors klasės (ar klasių) įrašai "sukris" tik į kontrolinę imtį. Vargu ar tokiu atveju pavyks sukonstruoti patikimą klasifikatorių. Kad taip neatsitiktų, sudaromos *sluoksninės* (kitaip *stratifikuotos*) imtys. Tai reiškia, kad klasių pasiskirstymas kiekvienoje dalyje atitinka visos imties proporcijas.

Tačiau net ir sluoksniuojant imties dalis, šis metodas turi akivaizdų trūkumą - didelė duomenų dalis nedalyvauja klasifikatoriaus mokyme. Jo alternatyva yra vadinamasis *kryžminis patikrinimas*. Čia imtis dalijama į k lygių nesikertančių sluoksniuotų dalių. Tada viena dalis, tarkime i - toji, tampa kontroline imtimi, o likusios $k-1$ dalys sudaro mokymo imtį (3 .14 paveiksle $k = 5,\;\; i = 3$).

Pagal šią mokymo imtį konstruojamas modelis ir kontrolinėje imtyje apskaičiuojama jo klaida e_i. Procedūra pakartojama su visais $i = 1, 2, \ldots, k$. Galutinis modelio klaidos įvertis

Mokymas	Mokymas	Kontrolė	Mokymas	Mokymas

3 .14 pav. k - kartinis kryžminis patikrinimas ($k = 5$)

$\hat{e}$ yra klaidų e_i vidurkis

$$\hat{e} = \frac{1}{k} \sum_{i=1}^{k} e_i \, .$$

Toks metodas dar vadinamas k - kartiniu kryžminiu patikrinimu (k - fold cross validation). Dažniausiai jis taikomas su $k = 10$.

Kai k yra lygus visos imties įrašų skaičiui N, gauname vadinamąjį "vieną išmesk" (leave one out) metodą. Jam būdinga didelė ($N - 1$ įrašas) mokymo imtis, tačiau kontrolei kiekvieną kartą paliekamas tik vienas įrašas. Be to, kai N didelis, modelio konstravimo procedūros kartojimas N kartų gali pareikalauti daug laiko.

3.3.4 Pakartotinų imčių metodas

Visą imtį, sudarytą iš N įrašų, pažymėkime E. Iki šiol mūsų nagrinėtos mokymo ir kontrolinės imtys buvo E poaibiai. Kitaip sakant, į mokymo imtį įrašai iš E buvo renkami be grąžinimo. *Pakartotinų imčių* (bootstrap) metodas remiasi grąžintine atranka. Mokymo imtį E_m sudarome N kartų grąžintinai pasirinkdami imties E įrašą. Aišku, kad taip sudaryta imtis turės N įrašų, tarp kurių bus (greičiausiai) ir pasikartojančių. Kontrolinę imtį E_k sudarys nepatekę į mokymo imtį įrašai

$$E_k = E \backslash E_m \, .$$

Kiek gi įrašų turės tokia kontrolinė imtis ? Pastebėsime, kad kiekvienam pradinės imties E įrašui I tikimybė nepatekti į mokymo imtį yra

$$P(I \notin E_m) = \left(1 - \frac{1}{N}\right)^N \, .$$

Tačiau

$$\lim_{N \to \infty} \left(1 - \frac{1}{N}\right)^N = \frac{1}{\mathrm{e}} \approx 0,368 \, .$$

Taigi, pakankamai dideliam N kontrolinėje imtyje turėsime apytiksliai $0,368 \cdot N$, o mokymo imtyje atitinkamai $0,632 \cdot N$ pradinės imties E įrašų. Todėl kartais šis metodas dar vadinamas 0,632 pakartotinų imčių (0,632 bootstrap) metodu.

Tarkime, kad aptartoji imčių sudarymo procedūra pakartojama b kartų. Kiekvieną kartą sukonstruojamas atitinkamas klasifikatorius ir jam apskaičiuojamos mokymo ir kontrolinės imties klaidos $e_m(i)$ ir $e_k(i)$, $i = 1, 2, \ldots, b$. Kaip jau buvo minėta, $e_m(i)$ būtų per daug optimistinis modelio klaidos įvertis. Kita vertus, remtis tik $e_k(i)$ reikšme taip pat nėra gerai, nes kontrolinę imtį E_k sudaro tik apie 36,8% visų duomenų. Todėl pakartotinų imčių metodas siūlo naudoti subalansuotos klaidos vidurkį. Tad galutinis modelio klaidos įvertis $\hat{e}_{boot}$ yra

$$\hat{e}_{boot} = \frac{1}{b} \sum_{i=1}^{b} (0,368 \cdot e_m(i) + 0,632 \cdot e_k(i)) \,.$$

Pakartotinų imčių metodu randami klaidų įverčiai, kai turima mažai duomenų. Tačiau jis (kaip, beje, ir bet kuris kitas) metodas turi ir trūkumų. Pavyzdžiui, tokia, tiesa gana dirbtinė, situacija. Turime visiškai atsitiktinius įrašus, su tikimybe $0,5$ priklausančius vienai iš dviejų klasių. Aišku, kad bet kurio modelio klaidos dydis (įrašo neteisingo klasifikavimo tikimybė) šiuo atveju bus taip pat $0,5$. Konstruojamas pilnai mokymo imtį atsimenantis klasifikatorius. Kitaip sakant, $e_m(i) = 0$. Tuo tarpu $e_k(i) = 0,5$. Todėl

$$\hat{e}_{boot} = 0,368 \cdot 0 + 0,632 \cdot 0,5 = 0,316 \,.$$

Akivaizdžiai optimizmo per daug.

Iš čia išplaukiantis baigiamasis patarimas: kiekvieną įvertį ar metodą reikia taikyti atsižvelgiant į jūsų turimus duomenis, be to, patartina įvertinti sukonstruoto modelio klaidą keliais skirtingais metodais.

3.4 Klasifikavimo taisyklės

Šiame skyrelyje nagrinėsime klasifikatorius, aprašomus viena ar keliomis *"jei ... tai... "* pavidalo taisyklėmis. Pavyzdžiui, vieną iš stuburinių gyvūnų klasifikavimo modelių galima nusakyti 3 .15 paveiksle pavaizduotu penkių taisyklių rinkiniu.

Kaip matome, toks klasifikatorius R nusakomas taisyklių $r_1, r_2, \ldots, r_l$ disjunkcija

$$R = (r_1 \vee r_2 \vee \cdots \vee r_l) \,.$$

$$
\begin{array}{ll}
r_1: & (Gyvavedis = ne) \wedge (Skraido = taip) \longrightarrow paukštis \\
r_2: & (Gyvavedis = ne) \wedge (Gyvena\ vandenyje = taip) \longrightarrow žuvis \\
r_3: & (Gyvavedis = taip) \wedge (Kraujo\ tipas = šiltas) \longrightarrow žinduolis \\
r_4: & (Gyvavedis = ne) \wedge (Skraido = ne) \longrightarrow roplys \\
r_5: & (Gyvena\ vandenyje = kartais) \longrightarrow varliagyvis
\end{array}
$$

3.15 pav. Stuburinių gyvūnų klasifikavimo taisyklių pavyzdys

Todėl taisyklės kartais dar vadinamos disjunktais. Kiekvieną taisyklę r_i sudaro prielaida $P(r_i)$ ir išvada apie klasės kintamojo reikšmę y_i

$$r_i: \quad P(r_i) \longrightarrow y_i\,.$$

Bet kuri prielaida $P(r_i)$ yra sudaryta iš sąlygų atributų $X_1, X_2, \ldots X_k$ reikšmėms

$$P(r_i) = (X_1\ op\ x_1) \wedge (X_2\ op\ x_2) \wedge \cdots \wedge (X_k\ op\ x_k)\,,$$

čia *op* žymi bet kurį santykį iš aibės $\{=, \neq, <, >, \leq, \geq\}$. Sąlygos $(X_j\ op\ x_j)$ vadinamos taisyklės r_i konjunktais.

Jei mokymo imties E įrašo $x \in E$ atributų reikšmės $(x_1, x_2, \ldots x_k)$ tenkina taisyklės r prielaidą $P(r)$, tai sakome, kad taisyklė r *apima* įrašą x. Pavyzdžiui, imkime dviejų stuburinių gyvūnų duomenis

Pavadinimas	Kraujo tipas	Odos danga	Gyva-vedis	Gyvena vandenyje	Skraido	Turi kojas	Žiemos miegas
varna	šiltas	plunksnos	ne	ne	taip	taip	ne
lokys	šiltas	kailis	taip	ne	ne	taip	taip

Nesunku įsitikinti, kad taisyklė r_1 (žr. 3.15 pav.) apima varną, bet "nemato" lokio. Klasifikavimo taisyklės kokybę nulemia jos *apimtis* ir *tikslumas*, nusakantys kiek imties įrašų apima taisyklė ir kiek iš jų teisingai klasifikuoja.

3.4.1 apibrėžimas. *Pagal imties E duomenis sukonstruotos klasifikavimo taisyklės*

$$r: \quad P(r) \longrightarrow y$$

apimtis $a(r)$ *ir tikslumas* $\theta(r)$ *yra lygūs*

$$a(r) = \frac{|P(r)|}{|E|},$$
$$\theta(r) = \frac{|P(r) \cup y|}{|P(r)|}.$$

Čia $|P(r)|$ *- įrašų, kuriuos apima taisyklė* r*, skaičius;* $|P(r) \cup y|$ *- teisingai klasifikuotų, t.y. tenkinančių prielaidą* $P(r)$ *ir priklausančių klasei* y*, įrašų skaičius;* $|E|$ *- imties dydis.*

Pastebėsime, kad teisingai klasifikuotų įrašų dalis visoje imtyje yra lygi sandaugai $a(r) \cdot \theta(r)$.

Pavadinimas	Kraujo tipas (X_1)	Odos danga (X_2)	Gyvavedis (X_3)	Gyvena vandenyje (X_4)	Skraido (X_5)	Turi kojas (X_6)	Žiemos miegas (X_7)	Klasė (Y)
žmogus	šiltas	plaukai	taip	ne	ne	taip	ne	žinduolis
pitonas	šaltas	žvynai	ne	ne	ne	ne	taip	roplys
lašiša	šaltas	žvynai	ne	taip	ne	ne	ne	žuvis
banginis	šiltas	plaukai	taip	taip	ne	ne	ne	žinduolis
varlė	šaltas	nėra	ne	kartais	ne	taip	taip	varliagyvis
komodo varanas	šaltas	žvynai	ne	ne	ne	taip	ne	roplys
šikšnosparnis	šiltas	plaukai	taip	ne	taip	taip	taip	žinduolis
balandis	šiltas	plunksnos	ne	ne	taip	taip	ne	paukštis
katė	šiltas	kailis	taip	ne	ne	taip	ne	žinduolis
gupija	šaltas	žvynai	taip	taip	ne	ne	ne	žuvis
aligatorius	šaltas	žvynai	ne	kartais	ne	taip	ne	roplys
pingvinas	šiltas	plunksnos	ne	kartais	ne	taip	ne	paukštis
dygliuotis	šiltas	dygliai	taip	ne	ne	taip	taip	žinduolis
ungurys	šaltas	žvynai	ne	taip	ne	ne	ne	žuvis
salamandra	šaltas	nėra	ne	kartais	ne	taip	taip	varliagyvis

3 .8 lentelė. Stuburiniai gyvūnai

3.4.1 pavyzdys. Tegul turime 3 .8 lentelėje matomą stuburinių gyvūnų imtį. Nagrinėsime kaip šią imtį atspindi 3 .15 paveiksle pateikiamos klasifikavimo taisyklės r_3 ir r_5. Imties

dydis $|E| = 15$. Taisyklės

$$r_3: \quad (Gyvavedis = taip) \wedge (Kraujo\ tipas = šiltas) \longrightarrow žinduolis$$

apimtis

$$a(r_3) = \frac{5}{15} = \frac{1}{3},$$

o tikslumas $\theta(r_3) = 1$, nes visi penki stuburiniai, kuriuos apima taisyklė r_3, yra žinduoliai. Analogiškai gausime, kad taisyklės

$$r_5: \quad (Gyvena\ vandenyje = kartais) \longrightarrow varliagyvis$$

apimtis ir tikslumas yra

$$a(r_5) = \frac{4}{15}, \quad \theta(r_5) = \frac{2}{4} = \frac{1}{2}.$$

3.4.1 Klasifikavimo taisyklių tarpusavio sąryšiai

Klasifikavimo taisyklių pagrindu sukonstruoto modelio efektyvumas gali priklausyti ne tik nuo taisyklių apimties ir tikslumo, bet ir nuo jų tarpusavio tvarkos. Prisiminkime, pavyzdžiui, 3 .15 paveiksle apibrėžtas taisykles ir pabandykime klasifikuoti tris naujus gyvūnus

Pavadinimas	Kraujo tipas	Odos danga	Gyvavedis	Gyvena vandenyje	Skraido	Turi kojas	Žiemos miegas
lemūras	šiltas	kailis	taip	ne	ne	taip	taip
vėžlys	šaltas	žvynai	ne	kartais	ne	taip	ne
ryklys	šaltas	žvynai	taip	taip	ne	ne	ne

Raskime šiuos įrašus apimančias taisykles.

- Pirmasis gyvūnas (lemūras) tenkina tik taisyklės r_3 prielaidas ir priskiriamas žinduolių klasei.

- Antrąjį gyvūną (vėžlį) apima dvi taisyklės - r_4 ir r_5 ir kiekviena iš jų "bando" priskirti vėžlį skirtingoms klasėms. Tarkime, kad taisykles taikomos paeiliui. Pagal r_4 vėžlys bus pripažintas ropliu. Tačiau, sukeitus r_4 ir r_5 vietomis, vėžlys "taps" varliagyviu pagal r_5. Gauname prieštaravimą, kurį būtina pašalinti.

- Trečiasis gyvūnas (ryklys) yra "nepažįstamas" visoms penkioms taisyklėms. Taip neturėtų būti. Reikia, kad klasifikatorius galėtų patikimai klasifikuoti bet kurį įrašą.

Pateiktasis pavyzdys iliustruoja dvi svarbias klasifikavimo taisyklių rinkinio savybes.

3.4.2 apibrėžimas. *Klasifikatoriaus $R = (r_1 \vee r_2 \vee \cdots \vee r_l)$ taisyklės vadinamos poromis nesutaikomomis, jei bet kokį įrašą gali apimti tik viena iš R taisyklių.*

3.4.3 apibrėžimas. *Sakome, kad klasifikatoriaus $R = (r_1 \vee r_2 \vee \cdots \vee r_l)$ taisyklės sudaro pilną rinkinį, jei bet kurį galimą įrašą apima bent viena R taisyklė.*

Kaip matėme, 3 .15 paveiksle pateiktos taisyklės neturi nė vienos iš šių savybių.

$$
\begin{array}{ll}
r_1: & (Kraujo\ tipas = šaltas) \longrightarrow ne\ žinduolis \\
r_2: & (Kraujo\ tipas = šiltas) \wedge (Gyvavedis = taip) \longrightarrow žinduolis \\
r_3: & (Kraujo\ tipas = šiltas) \wedge (Gyvavedis = ne) \longrightarrow ne\ žinduolis
\end{array}
$$

3 .16 pav. Pilnas, poromis nesutaikomų klasifikavimo taisyklių rinkinys

Jei tarsime, kad *Kraujo tipas* ir *Gyvavedis* yra binariniai kintamieji, tai 3 .16 paveiksle apibrėžtos klasifikavimo taisyklės $R = (r_1 \vee r_2 \vee r_3)$ yra poromis nesutaikomos ir sudaro pilną rinkinį. Tai reiškia, kad bet kurį įrašą apima lygiai viena taisyklė.
Kai klasifikatoriaus $R = (r_1 \vee r_2 \vee \cdots \vee r_l)$ taisyklių rinkinys nėra pilnas, tada prie esamų taisyklių papildomai prijungiama

$$r(y_d): \quad () \longrightarrow y_d\,,$$

priskirianti klasei y_d visus įrašus, kurių neapima taisyklės R. Taip papildytas klasifikatorius $R' = (r_1 \vee r_2 \vee \cdots \vee r_l \vee r(y_d))$ jau, aišku, bus pilnas. Klasė y_d kartais dar vadinama klase "pagal nutylėjimą". Ji dažniausiai pasirenkama taip, kad taisyklės $r(y_d)$ tikslumas būtų didžiausias

$$y_d = \operatorname*{argmax}_y \theta(r(y))\,.$$

Kitaip sakant, y_d yra taisyklių R neapimtų įrašų daugumos klasė.
Taikant taisykles, kurios nėra poromis nesutaikomos, galimi klasifikavimo prieštaravimai. Kaip matėme, tada rezultatas kartais priklauso ir nuo to kuria tvarka taisyklės yra taikomos. Skiriami du šios problemos sprendimo būdai.

1. **Sutvarkytas taisyklių rinkinys.** Kiekvienai taisyklei nustatomas prioriteto indeksas ir visos taisyklės vienareikšmiškai surūšiuojamos prioritetų mažėjimo tvarka. Toks sutvarkytas klasifikavimo taisyklių rinkinys kartais dar vadinamas *sprendimų sąrašu* (decision list). Tada kiekvienas įrašas klasifikuojamas pagal jį apimančią didžiausio prioriteto taisyklę. Prioriteto indeksų nustatymo kriterijai gali būti įvarūs. Jie gali priklausyti nuo taisyklių apimties, tikslumo, aprašymo ilgio ar net nuo tvarkos, kuria taisyklės buvo sukonstruotos ir įtrauktos į rinkinį.

2. **Nesutvarkytas taisyklių rinkinys.** Įrašas klasifikuojamas pagal visas jį apimančias rinkinio taisykles. Kiekviena tokia taisyklė, priskirdama įrašą klasei y, tuo pačiu "balsuoja" už šią klasę. Galutinai įrašas priskiriamas daugiausiai "balsų" surinkusiai klasei. Kartais taisyklės r_i "balsų" skaičius prieš sumavimą dauginamas iš svertinio koeficiento β_i, priklausančio nuo taisyklės apimties ir tikslumo. Todėl bendruoju atveju "balsavimo" procedūros formalus aprašymas galėtų būti toks. Tegul $R = (r_1 \vee r_2 \vee \cdots \vee r_l)$ ir

$$\delta_{ij} = \begin{cases} \beta_i\,, & \text{jei taisyklė } r_i \text{ įrašą priskiria klasei } y_j\,, \\ 0\,, & \text{priešingu atveju.} \end{cases}$$

   Jei

$$M = \underset{j}{\operatorname{argmax}} \left(\sum_{i=1}^{l} \delta_{ij} \right)\,,$$

   tai įrašas patenka į klasę y_M .

Vienareikšmiškai negalima pasakyti kurio tipo rinkinys yra pranašesnis. Nesutvarkytas taisyklių rinkinys nepriklauso nuo taisyklių pasirinkimo tvarkos, todėl tokio modelio konstravimo algoritmas yra paprastesnis (nereikia nustatinėti taisyklių prioritetų). Tačiau sutvarkyto taisyklių rinkinio pagrindu sukonstruotas klasifikatorius yra greitesnis, nes kiekvieno įrašo klasifikavimui reikalingas tikrinamų prielaidų skaičius šiuo atveju mažesnis.
Toliau nagrinėsime tik sutvarkytus klasifikavimo taisyklių rinkinius. Kaip jau minėjome, naudojami įvairūs taisyklių prioriteto apibrėžimai, o tuo pačiu ir rūšiavimo būdai. Sąlyginai juos galima padalinti į dvi grupes: *kokybinis rūšiavimas* ir *rūšiavimas pagal klases.* Kuo jos skiriasi paaiškėja, pažiūrėjus į 3.17 paveiksle pateiktą pavyzdį.

Kokybinis klasifikavimo taisyklių rūšiavimas
$(Odos\ danga = plunksnos) \wedge (Skraido = taip) \longrightarrow paukštis$
$(Kraujo\ tipas = šiltas) \wedge (Gyvavedis = taip) \longrightarrow žinduolis$
$(Kraujo\ tipas = šiltas) \wedge (Gyvavedis = ne) \longrightarrow paukštis$
$(Gyvena\ vandenyje = kartais) \longrightarrow varliagyvis$
$(Odos\ danga = žvynai) \wedge ((Gyvena\ vandenyje = ne) \longrightarrow roplys$
$(Odos\ danga = žvynai) \wedge ((Gyvena\ vandenyje = taip) \longrightarrow žuvis$
$(Odos\ danga = nėra) \longrightarrow varliagyvis$

Klasifikavimo taisyklių rūšiavimas pagal klases
$(Odos\ danga = plunksnos) \wedge (Skraido = taip) \longrightarrow paukštis$
$(Kraujo\ tipas = šiltas) \wedge (Gyvavedis = ne) \longrightarrow paukštis$
$(Kraujo\ tipas = šiltas) \wedge (Gyvavedis = taip) \longrightarrow žinduolis$
$(Gyvena\ vandenyje = kartais) \longrightarrow varliagyvis$
$(Odos\ danga = nėra) \longrightarrow varliagyvis$
$(Odos\ danga = žvynai) \wedge ((Gyvena\ vandenyje = ne) \longrightarrow roplys$
$(Odos\ danga = žvynai) \wedge ((Gyvena\ vandenyje = taip) \longrightarrow žuvis$

3.17 pav. Klasifikavimo taisyklių rūšiavimo būdų palyginimas

Kokybinis rūšiavimas. Apibrėžiamas koks nors taisyklės kokybės matas (pavyzdžiui, tikslumas) ir visos taisyklės išdėstomos kokybės mažėjimo tvarka. Toks klasifikatorius garantuoja, kad įrašą visada klasifikuos "geriausia" jį apimanti taisyklė. Vienas iš tokios schemos trūkumų yra tas, kad sudėtingai interpretuojamos sąrašo apačioje esančios taisyklės. Pavyzdžiui, imkime 3.17 paveikslo viršutinės lentelės ketvirtąją taisyklę. Ji užrašoma labai paprastai

$$(Gyvena\ vandenyje = kartais) \longrightarrow varliagyvis\,.$$

Tačiau, atsižvelgiant į pirmąsias tris taisykles, jos interpretacija yra sudėtingesnė: jei gyvūnas neplunksnuotas arba neskraido, yra šiltakraujis ir kartais gyvena vandenyje, tai jis - varliagyvis. Aišku, kad didelio sąrašo apačioje esančių taisyklių "iššifravimas" taps sudėtingu loginiu uždaviniu.

Rūšiavimas pagal klases. Šiuo atveju tą pačią klasę priskiriančios taisyklės sąraše stovi greta. Jų tarpusavio išsidėstymas nėra svarbus, nes neturi reikšmės kuri tos pačios klasės taisyklė klasifikuos įrašą. Tai šiek tiek supaprastina taisyklių interpretaciją. Skirtingų klasių taisyklių tarpusavio padėtis priklauso nuo klasės "kokybės". Kitaip sakant, reikia apibrėžti ne konkrečios taisyklės, o tam tikrą klasę priskiriančių taisyklių poaibio kokybės matą. Kuo matas didesnis, tuo aukščiau sąraše stovės visos šiam poaibiui priklausančios taisyklės. Todėl akivaizdu, kad aukštesnio rango klasė įgyja pranašumą, nes irašas pirmiausiai bandys "patekti" į aukščiusią klasę. Į tai reikia atsižvelgti pasirenkant kokybės matą. 3 .17 paveikslo apatinėje lentelėje matomo pavyzdžio taisyklės išdėstytos pagal tokią stuburinių gyvūnų klasių tvarką: { *paukščiai, žinduoliai, varliagyviai, ropliai, žuvys* }. Beje pastebėsime, kad šis taisyklių sąrašas, griežtai kalbant, nėra pilnas. Pavyzdžiui,būtų neaišku kuriai klasei priskirti kailinį šaltakraujį vandens gyventoją. Žinoma, klausimas ar toks iš viso egzistuoja !

Detaliau panagrinėsime surūšiuotus pagal klases taisyklių rinkinius ir jų sudarymo metodus. Klasifikavimo taisyklių konstravimo metodai skirstomi į *tiesioginius* ir *netiesioginius*. Tiesioginiai metodai skaido turimą įrašų aibę į mažesnius poaibius taip, kad visus vieno poaibio įrašus klasifikuotų viena taisyklė. Netiesioginiai metodai klasifikavimo taisyklėmis supaprastintai užrašo kitus, sudėtingesnius, klasifikavimo modelius.

3.4.2 Tiesioginis klasifikavimo taisyklių konstravimas

Tiesioginiam klasifikavimo taisyklių konstravimui dažnai yra naudojams *nuoseklaus dengimo* metodas (žr. 3 .18 pav.). Paeiliui kiekvienai klasei generuojamos taisyklės, kol jos tenkina pasirinktą kokybės kriterijų. Taigi, gaunamas pagal klases surūšiuotas taisyklių sąrašas. Kad jis būtų pilnas, paskutiniajai klasei "pagal nutylėjimą" priskiriami į kitas klases nepatekę įrašai. Kurios klasės taisyklės generuojamos pirmiausiai gali priklausyti nuo daugelio faktorių. Pavyzdžiui, nuo klasės populiarumo (dažnio imtyje) ar neteisingo klasifikavimo pasekmių (sakoma, kad geriau išteisinti kaltą, nei nuteisti nekaltą).
Algoritmas pradeda nuo tuščio taisyklių sąrašo R. Svarbiausią darbą atlieka funkcija `vienaTaisyklė`(E, y). Ji pagal turimą įrašų aibę E ir pasirinktą kokybės matą klasei y konstruoja geriausią taisyklę r. Visi klasei y priklausantys įrašai vadinami teigiamais, o

1. Tegul E - mokymo imtis
2. $A_Y := \{y_1, y_2, \ldots, y_k\}$ - surūšiuota visų klasių aibė
3. $r_d = \{() \longrightarrow y_k\}$ - taisyklė "pagal nutylėjimą"
4. $R := \{\}$ - pradinis taisyklių sąrašas
5. **for** $\forall y \in A_Y \backslash \{y_k\}$ **do**
6. **while** netenkinama stabdymo sąlyga **do**
7. $r =$ `vienaTaisyklė`(E, y)
8. Pašalinti iš E įrašus, kuriuos apima taisyklė r
9. $R = R \vee r$ - sąrašo apačioje pridedama taisyklė r
10. **end while**
11. **end for**
12. $R = R \vee r_d$ - sąrašo apačioje pridedama taisyklė "pagal nutylėjimą"

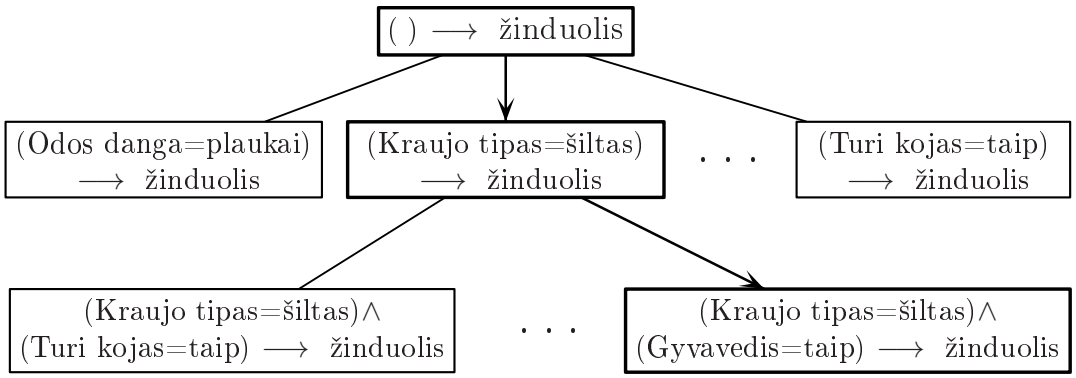
3.18 pav. Nuoseklaus dengimo algoritmas

likusieji - neigiamais. Aišku siekiama, kad r apimtų kuo daugiau teigiamų ir kuo mažiau neigiamų įrašų. Sukonstruotoji taisyklė r įrašoma sąrašo R apačioje, o visi įrašai, kuriuos ji apima, iš aibės E pašalinami. Procedūra tęsiama tol, kol kiekviena naujai sukonstruota taisyklė r yra pakankamai kokybiška. Procesas taip pat stabdomas, jei aibėje E nebelieka teigiamų įrašų arba taisyklės tampa per daug sudėtingos. Tada pereinama prie kitos klasės. Optimalios taisyklės radimas yra sudėtinga problema. Dažniausiai funkcija `vienaTaisyklė`(E, y) generuoja kokią nors pradinę taisyklę r_0 ir toliau ją tobulina, kol gauna reikalaujamos kokybės taisyklę r. Sutinkamos dvi tokio tobulinimo strategijos, kurias sąlyginai galima pavadinti *nuo paprasto prie sudėtingo* ir *nuo sudėtingo prie paprasto*. Pirmuoju atveju imama paprasta pradinė taisyklė

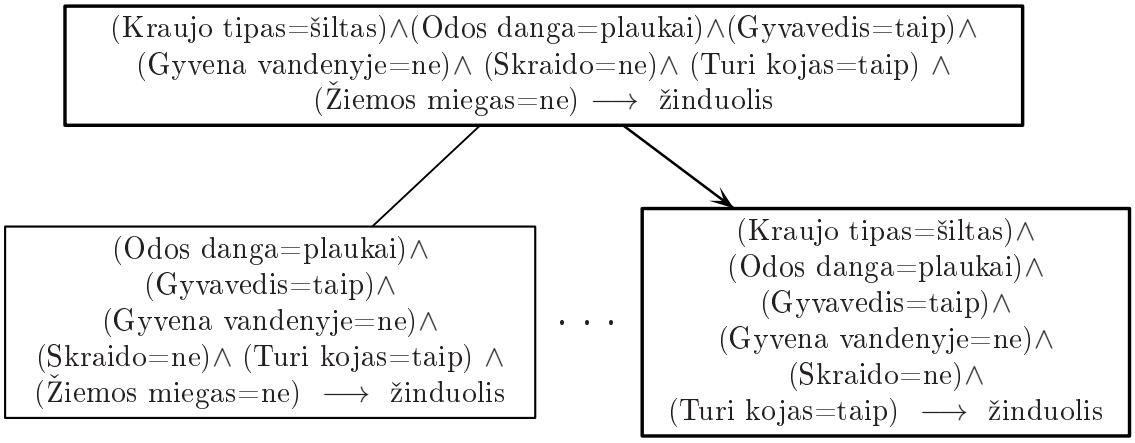
$$r_0 : \quad () \longrightarrow y\,,$$

apimanti visą įrašų aibę E. Ji nėra tiksli. Todėl toliau tobulinama, palaipsniui pridedant naujus konjunktus, kol pasiekiama norima taisyklės kokybė. 3.19(a) paveiksle iliustruojami du žinduolių klasifikavimo taisyklės tobulinimo procedūros žingsniai. Iš pradžių pasirenka-

mas konjunktas *(Kraujo tipas=šiltas)*. Toliau, išanalizavus visas galimybes, taisyklės prielaida papildoma konjunktu *(Gyvavedis=taip)* ir t.t. Šis procesas nutraukiamas, kai patenkinama stabdymo sąlyga, pavyzdžiui, naujai pridėtas konjunktas nebepagerina taisyklės.



(**a**) Nuo paprasto prie sudėtingo



(**b**) Nuo sudėtingo prie paprasto

3 .19 pav. Dvi klasifikavimo taisyklės konstravimo strategijos

Kai taikoma strategija *nuo sudėtingo prie paprasto*, pradinė taisyklė r_0 sudaroma pagal kurį nors atsitiktinai paimtą teigiamą įrašą. Toliau, didinant taisyklės apimtį, palaipsniui mažinamas konjunktų skaičius. Prastinimo procesas stabdomas, kai, pavyzdžiui, taisyklė pradeda apimti neigiamus įrašus. 3 .19(b) paveikslo pavyzdyje pasirinktas pirmasis teigia-

mas 3 .8 lentelės įrašas (žmogaus duomenys) ir pavaizduotas vienas prastinimo žingsnis: pašalintas konjunktas *(Žiemos miegas=ne)*.

Abi šios strategijos generuoja nebūtinai optimalią taisyklę, nes nagrinėjamos toli gražu ne visos galimos konjunktų kombinacijos. Galima praplėsti paieškos aibę. Eiliniame žingsnyje rinkime ne vieną, o k geriausių taisyklių ir kiekvieną iš jų modifikuokime prijungdami (arba išbraukdami) konjunktus. Iš visų taip modifikuotų taisyklių vėl renkame k geriausių ir pereiname prie naujos iteracijos.

Lieka aptarti galimus klasifikavimo taisyklių kokybės matus, kurie leistų palyginti taisykles ir nuspręsti kuriuos konjunktus prijungti (ar pašalinti) kiekvienoje algoritmo iteracijoje. Atrodytų tinkamiausias matas yra taisyklės tikslumas, atspindintis teisingai klasifikuotų įrašų dalį. Tačiau esminis jo trūkumas yra taisyklės apimties ignoravimas. Pavyzdžiui, tegul mokymo imtyje yra 60 teigiamų ir 100 neigiamų įrašų. Nagrinėkime dvi taisykles

r_1 : apima 50 teigiamų ir 5 neigiamus įrašus,

r_2 : apima 2 teigiamus įrašus ir nė vieno neigiamo įrašo.

Antroji taisyklė yra tikslesnė, nes

$$\theta(r_1) = \frac{50}{55} \approx 0,909\,, \quad \theta(r_2) = \frac{2}{2} = 1\,.$$

Tačiau vargu ar kas tvirtins, kad ji yra geresnė, nes apima tik 2 įrašus.

Apibrėšime keletą subtilesnių taisyklių palyginimo būdų.

1. Mažos apimties taisyklių eliminavimui galima taikyti statistinius kriterijus. Pavyzdžiui tikėtinumo statistikos
$$L = 2 \sum_{i=1}^{k} n_i \ln \frac{n_i}{e_i} \tag{3 .9}$$
reišmė parodo kiek klasifikatorius skiriasi nuo atsitiktinio klasifikavimo. Čia k - klasių skaičius; n_i - taisyklės apimamų i - tosios klasės įrašų skaičius[1]; e_i - prognozuojamas i - tosios klasės įrašų skaičius, taikant atsitiktinį klasifikavimą. Statistika L turi χ^2 skirstinį su $k-1$ laisvės laipsniu. Didelės L reikšmės rodo, kad teisingai klasifikuotų įrašų yra ženkliai daugiau, nei galėtume "atsitiktinai pataikyti". Kitaip sakant, kuo L reikšmė didesnė, tuo taisyklė geresnė.

[1] Jei kuris nors $n_i = 0$, tai atitinkamas dėmuo (3 .9) sumoje taip pat lygus 0.

Grįžkime prie mūsų pavyzdžio. Turime $k = 2$ klases: {+, -}. Rasime $L(r_1)$. Kadangi r_1 apima 55 įrašus, tai

$$e_+ = 55 \cdot \frac{60}{160} = 20,625\,, \quad e_- = 55 \cdot \frac{100}{160} = 34,375\,.$$

Taigi

$$L(r_1) = 2 \cdot \left(50 \cdot \ln \frac{50}{20,625} + 5 \cdot \ln \frac{5}{34,375}\right) \approx 69,27\,.$$

Analogiškai skaičiuojame ir antrąjai taisyklei:

$$e_+ = 2 \cdot \frac{60}{160} = 0,75\,, \quad e_- = 2 \cdot \frac{100}{160} = 1,25\,,$$

ir

$$L(r_2) = 2 \cdot \left(2 \cdot \ln \frac{2}{0,75} + 0\right) \approx 3,92\,.$$

Todėl pagal šį kriterijų taisyklė r_1 yra geresnė už r_2.

2. Tegul taisyklė r apima n įrašų, tarp kurių yra n_+ teigiamų. Apibrėšime du modifikuotus tikslumo matus, priklausomus nuo taisyklės r apimties. Tai yra Laplaso įvertis

$$\theta_L(r) = \frac{n_+ + 1}{n + k} \tag{3.10}$$

ir vadinamasis m - įvertis

$$\theta_m(r) = \frac{n_+ + k\,p_+}{n + k}\,, \tag{3.11}$$

čia k - klasių skaičius; p_+ - apriorinė teigiamos klasės tikimybė. Pastebėsime, kad

$$\theta_L(r) = \theta_m(r)\,, \text{ kai } p_+ = \frac{1}{k}\,.$$

Kitaip sakant, tolygiai pasiskirsčiusioms klasėms šie matai ekvivalentūs. Mažos apimties taisyklėms, jie yra artimi teigiamos klasės tikimybei, nes

$$\theta_L(r) = \frac{1}{k}\,, \quad \theta_m(r) = p_+\,, \quad \text{kai } n = 0\,.$$

Kai taisyklės apimtis yra didelė, abu įverčiai darosi artimi taisyklės tikslumui

$$\theta(r) = \frac{n_+}{n}\,.$$

Rasime taisyklių r_1 ir r_2 Laplaso ir m - įverčius. Teigiamos klasės apriorinė tikimybė yra $p_+ = 60/160 = 0,375$, klasių skaičius $k = 2$. Įstatę reikiamas reikšmes į (3 .10) ir (3 .11), turėsime

$$\begin{aligned} \theta_L(r_1) &= \frac{50+1}{55+2} \approx 0,895\,, \quad \theta_L(r_2) = \frac{2+1}{2+2} = 0,75\,, \\ \theta_m(r_1) &= \frac{50+2\cdot 0,375}{55+2} \approx 0,89\,, \quad \theta_m(r_2) = \frac{2+2\cdot 0,375}{2+2} = 0,6875\,. \end{aligned}$$

Matome, kad pagal abu įverčius taisyklė r_1 geresnė.

3. Dar vienas klasifikavimo taisyklių palyginimo kriterijus remiasi *informacijos prieaugio* sąvoka. Tegul taisyklė

$$r_0: \quad A \longrightarrow +$$

apima n_+ teigiamų ir $n-n_+$ neigiamų įrašų. Jos prielaidą papildome nauju konjunktu B. Tegul naujoji taisyklė

$$r: \quad A \wedge B \longrightarrow +$$

apima m_+ teigiamų ir $m-m_+$ neigiamų įrašų. Gaunamas informacijos prieaugis yra

$$I(r_0, r) = m_+ \cdot \left(\log_2 \frac{n}{n_+} - \log_2 \frac{m}{m_+} \right) . \tag{3 .12}$$

Atskiru atveju, kai taisyklės r_0 prielaida tuščia $A = (\ )$, informacijos prieaugį žymėsime $I(r)$. Tada n yra imties dydis, o n_+ - visų teigiamų įrašų skaičius.

Be to, prisiminę klasifikavimo taisyklės tikslumo apibrėžimą 3.4.1 , informacijos prieaugio išraišką (3 .12) galime parašyti ir taip

$$I(r_0, r) = m_+ \cdot \log_2 \frac{\theta(r)}{\theta(r_0)}\,. \tag{3 .13}$$

Informacijos prieaugis $I(r_0, r)$ reiškia sumanį neapibrėžtumo pokytį visiems taisyklės r apimamiems teigiamiems įrašams. Iš tikrųjų, taisyklei r_0 imties įrašo teisingo klasifikavimo tikimybė yra lygi jos tikslumui

$$P(T_0) = \theta(r_0) = \frac{n_+}{n}\,.$$

Todėl įvykio T_0 informacija (arba neapibrėžtumas) yra (žr. 1.3.1 apibrėžimą)

$$I(T_0) = \log_2 \frac{1}{P(T_0)} = \log_2 \frac{n}{n_+}\,.$$

Analogiškai randamas likęs neapibrėžtumas modifikuotai taisyklei r

$$I(T) = \log_2 \frac{1}{P(T)} = \log_2 \frac{m}{m_+}\,.$$

Iš čia ir gauname, kad $I(r_0, r) = m_+ \cdot (I(T_0) - I(T))\,.$

Todėl iš visų nagrinėjamų taisyklės r_0 modifikacijų r reikėtų rinktis tą, kuri labiausiai sumažina neapibrėžtumą, t.y. turi didžiausią informacijos prieaugį $I(r_0, r)\,.$

Informacijos prieaugio taikymą iliustruosime, dar kartą palygindami taisykles r_1 ir r_2. Pagal (3 .13) formulę

$$\begin{aligned} I(r_1) &= 50 \cdot \log_2 \frac{50/55}{60/160} \approx 63,877\,, \\ I(r_2) &= 2 \cdot \log_2 \frac{2/2}{60/160} \approx 2,83\,. \end{aligned}$$

Taigi ir pagal šį kriterijų taisyklė r_1 yra geresnė.

Pagal vieną ar kitą strategiją sukonstruota taisyklė gali būti pertekli, t.y. turėti didelę modelio klaidą (modelio klaidos įverčiai nagrinėjami 3.3 skyrelyje). Tuo atveju taisyklė redukuojama, jos prielaidoje atsisakant kai kurių konjunktų. Jei tokia redukcija sumažina modelio klaidą, taisyklė keičiama paprastesne.

3.4.3 1R algoritmas

Bene paprasčiausias, tačiau dažnai gana efektyvus klasifikavimo taisyklių generavimo algoritmų yra vadinamasis 1R algoritmas (1-Rule, R.C.Holte, 1993). Tarkime, kad visi imties atributai $X_1, X_2, \ldots, X_k$ yra kategoriniai. Algoritmo schema labai paprasta. Ji pateikiama 3 .20 paveiksle.

Kiekvienam atributui X_i, sudaromas pilnas, poromis nesutaikomų taisyklių rinkinys R_i, susidedantis iš tiek taisyklių, kiek skirtingų reikšmių x įgyja X_i. Kiekvienos taisyklės prielaida yra $(X_i = x)$, o išvada lygi reikšmės x daugumos klasei. Po to iš visų sukonstruotų klasifikatorių $R_1, R_2, \ldots, R_k$ išrenkamas tas, kurio tikslumas didžiausias.

Papildomai 1R algoritmas dirba ir su trūkstamomis bei skaitinėmis atributų reikšmėmis. Trūkstamąją atributo reikšmę (ją žymėsime '?') algoritmas traktuoja tiesiog kaip dar vieną atributo reikšmę.

1. Tegul $X_1, X_2, \ldots, X_k$ - imties atributai, A_Y - visų klasių aibė
2. **for** *kiekvienam atributui* X_i **do**
3. $R_i := \{\}$ - pradinis taisyklių sąrašas
4. **for** *kiekvienai atributo* X_i *reikšmei* x **do**
5. $r(y): \ (X_i = x) \ \longrightarrow \ y\,, \quad y \in A_Y\,,$
$y_{\max} = \underset{y}{\operatorname{argmax}}\, \theta(r(y))$ - daugumos klasė
6. $R_i = R_i \vee r(y_{\max})$ - sąrašo apačioje pridedama taisyklė $r(y_{\max})$
7. **end for**
8. **end for**
9. $R = \underset{R_i}{\operatorname{argmax}}\, \theta(R_i)$ - randamas geriausias klasifikatorius

3 .20 pav. 1R algoritmas

Skaitiniai kintamieji yra diskretizuojami. Taikomas labai paprastas diskretizavimo metodas. Visas intervalas dalijama į intervalus, kurių galai randasi viduryje tarp dviejų gretimų reikšmių. Visi įrašai su atributo reikšmėmis iš kurio nors intervalo priskiriami to intervalo daugumos klasei. Jei kintamasis įgyja labai daug skirtingų reikšmių, tai galime gauti daug smulkių intervalų, turinčių po vieną įrašą. Tai, savo ruožtu, įtakotų modelio perteklumo atsiradimą. Todėl mažai įrašų turintys gretimi intervalai jungiami, kol jungtinio intervalo daugumos klasei priklausančių įrašų skaičius pasiekia tam tikrą skaičių n_{diskr}. Šis skaičius yra algoritmo parametras. Algoritmo autoriaus siūloma reikšmė $n_{\text{diskr}} = 6$.

Prisiminkime 2.1.1 skyrelio pavyzdį apie oro sąlygas, kurioms esant "Žvaigždžių" komanda žaidžia parodomąsias rungtynes. Papildysime 2 .2 lentelėje pateiktus duomenis vienu "pamirštu" įrašu, turinčiu trūkstamą atributo *Oras* (X_1) reikšmę. Naujoji imtis pavaizduota 3 .9 lentelėje.

Atsižvelgę į tai, kad imtis nedidelė, atributų X_2 ir X_3 diskretizavimui imsime $n_{\text{diskr}} = 3$. Pirmiausiai surūšiuojame įrašus pagal *Temperatūrą*.

X_2	18	18	19	20	21	21	22	22	22	24	24	27	27	28	29
Y	ne	taip	ne	taip	taip	taip	ne	taip	ne	taip	taip	ne	taip	taip	ne

	Oras (X_1)	*Temperatūra* (X_2)	*Drėgnumas* (X_3)	*Vėjuota* (X_4)	*Žaisti* (Y)
1	saulėta	29	85	FALSE	ne
2	saulėta	27	90	TRUE	ne
3	debesuota	28	86	FALSE	taip
4	lietinga	21	96	FALSE	taip
5	lietinga	20	80	FALSE	taip
6	lietinga	18	70	TRUE	ne
7	debesuota	18	65	TRUE	taip
8	saulėta	22	95	FALSE	ne
9	saulėta	21	70	FALSE	taip
10	lietinga	24	80	FALSE	taip
11	saulėta	24	70	TRUE	taip
12	debesuota	22	90	TRUE	taip
13	debesuota	27	75	FALSE	taip
14	lietinga	22	91	TRUE	ne
15	?	19	92	TRUE	ne

3 .9 lentelė. Oro duomenys su trūkstama reikšme

Daliname visų X_2 reikšmių intervalą į dalis taip, kad kiekvienoje dalyje (išskyrus paskutinę) dažniausiai sutinkamų Y reikšmių skaičius būtų ne mažesnis už 3. Aišku, dalinimo taškus imsime tarp skirtingoms klasėms priklausančių X_2 reikšmių. Gausime

X_2	18	18	19	20	21	21	22	22	22	24	24	27	27	28	29
Y	ne	taip	ne	taip	taip	taip	ne	taip	ne	taip	taip	ne	taip	taip	ne

Trečiajame intervale turime po du abiejų klasių įrašus. Todėl daugumos klasę renkamės atsitiktinai. Tegul tai bus $Y = ne$. Kadangi pirmųjų dviejų intervalų daugumos klasė $Y = taip$ yra ta pati, tai, nekeisdami taisyklių prasmės, galime juos sujungti. Tad galutinis skaidinys bus

X_2	18	18	19	20	21	21	22	22	22	24	24	27	27	28	29
Y	ne	taip	ne	taip	taip	taip	ne	taip	ne	taip	taip	ne	taip	taip	ne

Todėl gausime tokias klasifikavimo pagal *Temperatūrą* taisykles

$$\begin{aligned} r_{21}: &\quad (\textit{Temperatūra} \leq 25,5) \longrightarrow \textit{Žaisti=taip} \\ r_{22}: &\quad (\textit{Temperatūra} > 25,5) \longrightarrow \textit{Žaisti=ne} \end{aligned}$$

Jų tikslumai

$$\theta(r_{21}) = \frac{7}{11}\,, \quad \theta(r_{22}) = \frac{2}{4}\,.$$

Bendrasis klasifikatoriaus pagal *Temperatūrą* tikslumas yra

$$\theta(r_{21} \vee r_{22}) = \frac{9}{15} = 0,6\,.$$

Atributo X_3 (*Drėgnumas*) diskretizavimo skaidinys randamas analogiškai

X_3	65	70	70	70	75	80	80	85	86	90	90	91	92	95	96
Y	taip	ne	taip	taip	taip	taip	taip	ne	taip	taip	ne	ne	ne	ne	taip

Todėl klasifikatorių pagal *Drėgnumą* sudaro trys taisyklės

$$\begin{aligned} r_{31}: &\quad (\textit{Drėgnumas} \leq 82,5) \longrightarrow \textit{Žaisti=taip} \\ r_{32}: &\quad (\textit{Drėgnumas} \in (82,5\,;95,5]) \longrightarrow \textit{Žaisti=ne} \\ r_{33}: &\quad (\textit{Drėgnumas} > 95,5) \longrightarrow \textit{Žaisti=taip} \end{aligned} \tag{3.14}$$

Jo tikslumas

$$\theta(r_{31} \vee r_{32} \vee r_{33}) = \frac{12}{15} = 0,8\,.$$

Pagal 1R algoritmą taip analizuojami visi keturi klasifikatoriai. Rezultatai pateikiami 3.10 lentelėje.

Matome, kad tiksliausias klasifikatorius yra pagal *Drėgnumą*. Todėl (3.14) ir bus 1R algoritmo sukonstruotas klasifikavimo taisyklių rinkinys.

	Atributai	Taisyklės	Taisyklių tikslumas	Klasifikatorių tikslumas
1	X_1 $(Oras)$	$saulėta \longrightarrow ne$	3/5	11/15
		$debesuota \longrightarrow taip$	4/4	
		$lietinga \longrightarrow taip$	3/5	
		$? \longrightarrow ne$	1/1	
2	X_2 $(Temperatūra)$	$\leq 25,5 \longrightarrow taip$	7/11	9/15
		$> 25,5 \longrightarrow ne$	2/4	
3	X_3 $(Drėgnumas)$	$\leq 82,5 \longrightarrow taip$	6/7	12/15
		$\in (82,5\,;95,5] \longrightarrow ne$	5/7	
		$> 95,5 \longrightarrow taip$	1/1	
4	X_4 $(Vėjuota)$	$FALSE \longrightarrow taip$	6/8	10/15
		$TRUE \longrightarrow ne$	4/7	

3 .10 lentelė. 1R algoritmas oro duomenims su trūkstama reikšme

3.4.4 RIPPER algoritmas

Vienas iš tiesioginio klasifikavimo taisyklių konstravimo algoritmų yra vadinamasis RIPPER algoritmas (Repeated Incremental Pruning to Produce Error Reduction, William W. Cohen, 1995). Jis gerai veikia, kai duomenys yra "triukšmingi", nes dar modelio konstravimo etape naudoja kontrolinę imtį. Algoritmas generuoja pagal klases surūšiuotą klasifikavimo taisyklių rinkinį.

Tegul

$$A_Y = \{y_1, y_2, \ldots, y_k\}$$

- pagal dažnius surūšiuota visų klasių aibė. Rečiausiai mokymo imtyje sutinkama klasė yra y_1 , o dažniausiai - y_k. Taisyklės generuojamos nuoseklaus dengimo metodu. Visų pirma klasei y_1 priklausantys įrašai pavadinami teigiamais, o likusieji - neigiamais ir konstruojamos taisyklės teigiamiems įrašams atskirti. Toliau pereinama prie klasės y_2 ir taip toliau, kol lieka tik y_k. Pastaroji klasė tampa klase "pagal nutylėjimą", t.y. jai priskiriami į kitas klases nepatekę įrašai.

Kiekvieną taisyklę r RIPPER konstruoja naudodamas "nuo paprasto prie sudėtingo" strategiją. Taisyklė modifikuojama, prielaidoje pridedant naujus konjunktus, kol pasidaro tiksli, t.y. $\theta(r) = 1$. Papildomų konjunktų pasirinkimo kriterijus yra informacijos prieaugio (3 .12) dydis. Po to taisyklė r redukuojama, priklausomai nuo to kaip ji klasifikuoja kontrolinės imties įrašus. Tam naudojamas toks kokybės matas

$$\gamma(r) = \frac{v_+ - v_-}{v_+ + v_-}\,,$$

čia v_+ (v_-) yra teigiamų (neigiamų) kontrolinės imties įrašų, kuriuos apima taisyklė r , skaičius. Taisyklė redukuojama, jei po redukcijos šis matas padidėja. Redukuojama , pirmiausiai šalinant paskutinius konjunktus. Tarkime, kad taisyklės r prielaidoje yra m konjunktų:

$$r:\ (A_1 \wedge A_2 \wedge \cdots \wedge A_m) \ \longrightarrow\ +\,.$$

Nagrinėkime redukuotas taisykles

$$r_i:\ (A_1 \wedge A_2 \wedge \cdots \wedge A_{m-i}) \ \longrightarrow\ +\,, \quad i = 1, 2, \ldots, m-1\,,$$

ir $r_m:\ () \ \longrightarrow\ +$. Tegul r' yra didžiausią γ reikšmę turinti redukuota taisyklė:

$$r' = \operatorname*{argmax}_{r_i} \gamma(r_i)\,.$$

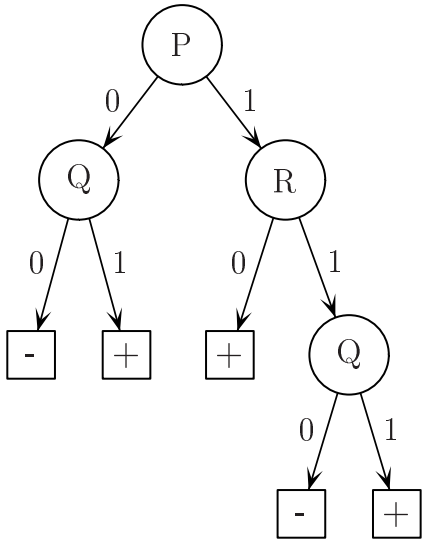
Jei $\gamma(r') > \gamma(r)$, tai taisyklė r pakeičiama redukuotaja taisykle r'. Pastebėsime, kad mokymo imtyje taisyklės r' tikslumas gali būti ir mažesnis už 1, t.y. $\theta(r') < \theta(r) = 1$. Toliau, pagal nuoseklaus dengimo metodą, iš mokymo imties pašalinami įrašai, kuriuos apima sukonstruotoji taisyklė ir pereinama prie kitos taisyklės generavimo. Beje taisyklė neapima įrašų, kuriuose trūksta bent vieno į taisyklės prielaidą įeinančio atributo reikšmės. RIPPER algoritmas stabdomas, kai tenkinama bent viena iš žemiau išvardintų sąlygų.

1. Mokymo imtyje nebėra teigiamų įrašų.
2. Naujai prijungiama taisyklė padidintų sąrašo aprašymo ilgį ne mažiau kaip d bitų (pagal nutylėjimą $d = 64$).
3. Naujai prijungiamos taisyklės tikslumas kontrolinėje imtyje mažesnis už 0,5 .

Baigus konstruoti taisyklių sąrašą, priklausomai nuo algoritmo realizacijos, kartais dar papildomai optimizuojamas visas sąrašas.

3.4.5 Netiesioginis klasifikavimo taisyklių konstravimas

Šiame skyrelyje trumpai aptarsime sprendimų medžio ir klasifikavimo taisyklių tarpusavio priklausomybę. Aišku, kad kiekvienas kelias medyje nuo šaknies iki lapo nusako klasifikavimo taisyklę. Jos prielaidą sudarantys konjunktai atitinka medžio vidinių viršūnių sąlygas, o išvada sutampa su lapo atstovaujama klase. Be to, visus sprendimų medžio lapus atitinkančios taisyklės yra poromis nesutaikomos ir sudaro pilną rinkinį. Taip sukonstruotą taisyklių rinkinį dažniausiai galima supaprastinti. Pavyzdžiui, 3 .21 paveiksle pavaizduotas medis ir visus jo lapus atitinkančių taisyklių rinkinys.



$\Longrightarrow$

Klasifikavimo taisyklės		
$r_1:$	$(P=0)\wedge(Q=0)$	$\longrightarrow$ -
$r_2:$	$(P=0)\wedge(Q=1)$	$\longrightarrow$ +
$r_3:$	$(P=1)\wedge(R=0)$	$\longrightarrow$ +
$r_4:$	$(P=1)\wedge(R=1)\wedge(Q=0)$	$\longrightarrow$ -
$r_5:$	$(P=1)\wedge(R=1)\wedge(Q=1)$	$\longrightarrow$ +

3 .21 pav. Sprendimų medis ir jo generuotos klasifikavimo taisyklės

Pastebėsime, kad teigiamos klasės taisyklių rinkinys $R_+ = (r_2 \vee r_3 \vee r_5)$ visus įrašus, kuriems $Q = 1$, priskiria teigimai klasei. Todėl R_+ galime supaprastinti. Naujasis teigiamos klasės taisyklių rinkinys bus sudarytas iš dviejų taisyklių $R'_+ = (r'_2 \vee r_3)$, čia

$$r'_2: \quad (Q=1) \longrightarrow +\,.$$

Neigiamoji klasė tada tampa klase "pagal nutylėjimą". Galutinai gauname tokį sutvarkytą taisyklių rinkinį

$$(Q=1) \longrightarrow +$$
$$(P=1)\wedge(R=0) \longrightarrow +$$
$$() \longrightarrow -$$

Šios taisyklės jau nėra poromis nesutaikomos, tačiau gerokai paprastesnės ir lengvai interpretuojamos.

Deja, didesnius sprendimų medžius atitinkančių klasifikavimo taisyklių rinkinių optimizavimo uždavinys yra gana sudėtingas. Todėl dažniausiai naudojami algoritmai vienokiais ar kitokiais būdais modifikuoja pradinį taisyklių rinkinį, tačiau ne visada pasiekia optimalų rezultatą. Vienas iš tokių yra vadinamasis *C4.5rules* algoritmas, generuojantis sprendimų medį atitinkančias klasifikavimo taisykles. Algoritmas susideda iš dviejų etapų.

1. *Taisyklės konstravimas.* Panašiai kaip ir 3 .21 paveikslo pavyzdyje, pradinis rinkinys sudaromas iš visus medžio lapus atitinkančių taisyklių. Toliau jos redukuojamos, atsisakant kai kurių prielaidos konjunktų. Tegul

$$r: \quad A \longrightarrow y$$

yra viena iš pradinių taisyklių. Nagrinėkime visas supaprastintas taisykles

$$r': \quad A' \longrightarrow y\,,$$

čia A' gaunama išmetus iš A kurį nors vieną konjunktą. Iš visų taisyklių r' išrenkame tą, kuri turi mažiausią statistinį klaidos įvertį (3 .8) su iš anksto pasirinktu pasikliovimo lygmeniu Q (pagal nutylėjimą $Q = 0,5$). Tegul

$$r_{\min} = \operatorname*{argmin}_{r'} e_v(r', Q)\,.$$

Pradinė taisyklė r keičiama taisykle $r_{\min}$, jei

$$e_v(r_{\min}, Q) < e_v(r, Q)\,.$$

Ši procedūra kartojama, kol modifikuotos taisyklės statistinis klaidos įvertis mažėja. Taip rekonstruojamos visos pradinio rinkinio taisyklės. Jei po rekonstrukcijos atsiranda sutampančių taisyklių, tai paliekama tik viena iš jų.

2. *Rinkinio tvarkymas.* Taisyklės rūšiuojamos pagal klases. Randamas kiekvienos klasės y_i taisyklių rinkinio R_i aprašymo ilgis (3 .6) su papildomu parametru g

$$L_g(E, R_i) = gL(R_i) + L(E|R_i)\,.$$

Kuo daugiau klasifikatorius R_i turi nenaudojamų mokymo imties E atributų, tuo mažesnė yra g reikšmė. Dažniausiai $g = 0,5$. Klasės rūšiuojamos aprašymo ilgio didėjimo tvarka. Galutiniame taisyklių sąraše aukščiausiai stovi trumpiausiai aprašomos klasės taisyklės.

Taisyklių rinkinio rekonstrukcijos procedūra yra gana lėta, nes reikia skaičiuoti kiekvienos nagrinėjamos taisyklės tikslumą bei vienokį ar kitokį klaidos įvertį. Kai imtis didelė, tai nėra taip paprasta. Todėl dažniausiai tiesioginiai klasifikavimo taisyklių konstravimo metodai yra efektyvesni už netiesioginius.

3.5 Artimiausių kaimynų metodas

Bendroji klasifikavimo uždavinio sprendimo schema, pavaizduota 3 .1 paveiksle, susideda iš dviejų dalių:

1) modelio konstravimas pagal turimus duomenis (indukcija);
2) įrašo klasifikavimas, taikant sukonstruotą modelį (dedukcija).

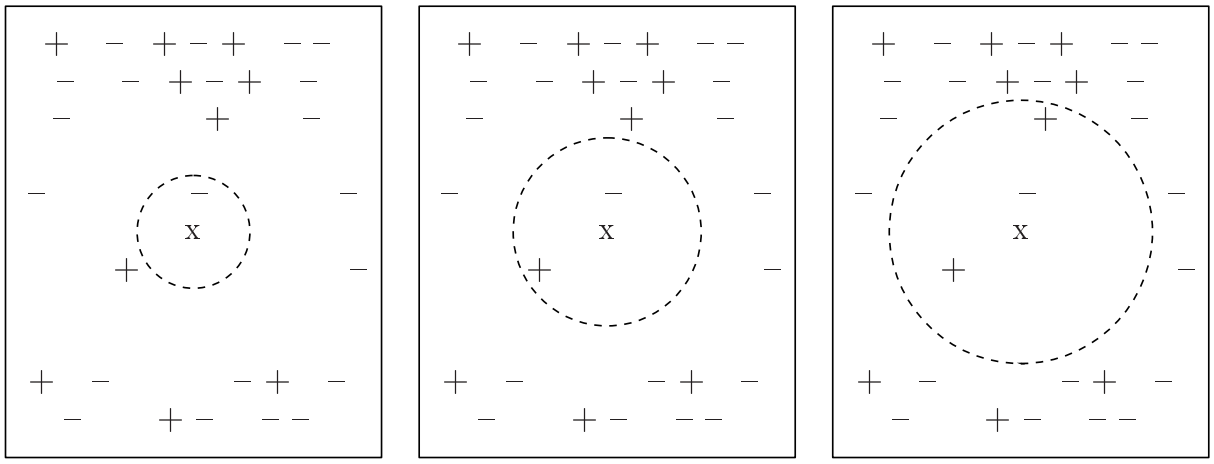
Mūsų jau nagrinėtuose klasifikavimo algoritmuose abi dalys buvo atskirtos. Iš pradžių buvo generuojamas modelis (klasifikatorius), kuris paskui bet kada naudojamas norimo įrašo klasifikavimui. Taip veikia tiek sprendimų medžiais, tiek klasifikavimo taisyklėmis besiremiantys algoritmai. Tokie algoritmai kartais dar vadinami stropiais.

Galima kalbėti ir apie vadinamuosius tingius algoritmus. Pastarieji nieko nedaro iš anksto. Jie "pasijudina" tik kai prisireikia klasifikuoti kokį nors įrašą. Kitaip sakant, abu aukščiau minėti klasifikavimo uždavinio sprendimo žingsniai atliekami kartu. Pats paprasčiausias tokio metodo taikymo pavyzdys yra mechaniškos atminties klasifikatorius (rote classifier). Jis atsimena visą mokymo imtį ir klasifikuoja įrašą tik tuo atveju, jei randa šio įrašo atributų reikšmes tiksliai atitinkantį mokymo imties įrašą. Akivaizdus tokio klasifikatoriaus trūkumas - maža "atpažįstamų" įrašų aibė (ypač kai turima maža mokymo imtis).

Vienas iš šio trūkumo neutralizavimo būdų yra toks. Iš mokymo imties išrenkame K įrašų, kurie yra labiausiai "panašūs" į norimą klasifikuoti įrašą (*K artimiausių kaimynų*). Tiriamasis įrašas klasifikuojamas atsižvelgiant į tai, kokioms klasėms priklauso kaimynai (pavyzdžiui, priskiriamas kaimynų daugumos klasei). Tokio samprotavimo pateisinimui

prisiminkime posakį: "If it walks like a duck, quacks like a duck and looks like a duck, then it's probably a duck"[1]. Įrašų panašumas nustatomas, pasirinkus tinkamą artumo matą (žr. 2.3.4 skyrelį).

Rezultatas priklausys ne tik nuo artumo mato pasirinkimo, bet ir nuo sprendimą įtakojančių kaimynų skaičiaus K . Pavyzdžiui, 3 .22 paveiksle matome 1, 2 ir 3 artimiausius kaimynus apskritimo centre simboliu 'x' pavaizduotam įrašui z. Jei $K = 1$ (3 .22(a) pav.), tai



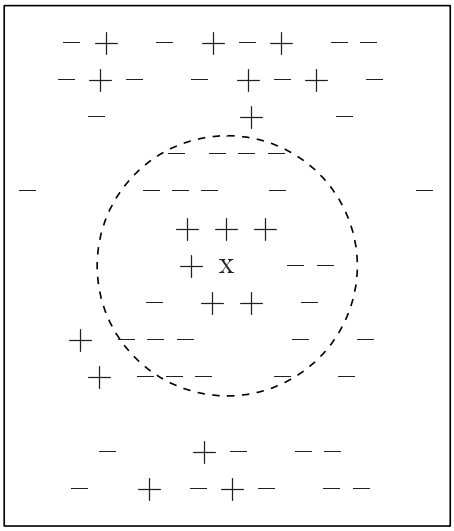
(a) 1 artimiausias kaimynas (b) 2 artimiausieji kaimynai (c) 3 artimiausieji kaimynai

3 .22 pav. Įrašo x artimiausieji kaimynai

z priskiriamas neigiamai klasei, nes šiai klasei priklauso jo artimiausias kaimynas. Du artimiausi kaimynai (3 .22(b) pav.) priklauso skirtingoms klasėms. Todėl z atsitiktinai priskiriamas vienai iš jų, kai $K = 2$. Jei $K = 3$ (3 .22(c) pav.), tai z priskiriamas teigiamai klasei, nes du iš trijų artimiausių kaimynų yra teigiami.

Aišku, kad per mažas kaimynų skaičius K gali sukelti klasifikatoriaus perteklumą dėl galimų mokymo imties duomenų iškraipymų. Kita vertus, per didelis K taip pat nėra gerai, nes tada sprendimą gali įtakoti "tolimi" kaimynai (žr. 3 .23 pav.). Nepageidaujamą "tolimų" kaimynų įtaką galima apriboti tinkamai parinkus kaimynų svorio koeficientus. Tegul $z = (\mathbf{x}'; y')$ yra klasifikuojamas įrašas (klasė y' nežinoma) , $(\mathbf{x}; y) \in E$ - mokymo imties E įrašas. Pasirinkę norimą atstumo matą $d(\mathbf{x}', \mathbf{x})$, sudarome K artimiausių įrašo z kaimynų sąrašą $E_z \subset E$. Kiekvienam kaimynui $(\mathbf{x}; y) \in E_z$ priskiriame neneigiamą svorį $w(\mathbf{x})$.

[1](Angl.) Jei vaikšto kaip antis, kvaksi kaip antis ir atrodo kaip antis, tai greičiausiai ir yra antis.



3.23 pav. Klasifikavimas pagal daug kaimynų

Tada svertinis klasei c priklausančių kaimynų skaičius bus

$$S(c, E_z) = \sum_{\substack{(\mathbf{x};y)\in E_z \\ y=c}} w(\mathbf{x})\,.$$

"Tolimų" kaimynų įtaka sumos $S(c, E_z)$ dydžiui sumažės, jei svorio koeficientai bus monotoniškai mažėjančios atstumo funkcijos, pavyzdžiui,

$$w(\mathbf{x}) = \mathrm{e}^{-d(\mathbf{x}',\mathbf{x})}\,.$$

Jei visi kaimynai vienodai svarbūs, tai $w(\mathbf{x}) \equiv 1$. Tada suma $S(c, E_z)$ bus lygi klasei c priklausančių kaimynų skaičiui. Klasifikatorius įrašą z priskiria klasei c , kurios suma $S(c, E_z)$ yra didžiausia.

Artimiausių kaimynų metodo realizacijos algoritmo schema pateikiama 3.24 paveiksle.

Kaip matėme, artimiausių kaimynų klasifikatorius nekonstruoja bendro modelio. Todėl kiekvieną kartą tiriamasis įrašas klasifikuojamas iš naujo analizuojant visą mokymo imtį. Ši schema yra lankstesnė, bet kita vertus tokio tipo klasifikatoriai dirba lėčiau.

3.6 Bajeso klasifikatoriai

Dažnai atributų reikšmės vienareikšmiškai nenusako įrašo klasės, nes tai gali priklausyti nuo daugelio papildomų faktorių. Pavyzdžiui, du vienodo amžiaus ir vienodą skaičių cigarečių

1. Tegul E - mokymo imtis, K - artimiausių kaimynų skaičius
2. **for** kiekvienam klasifikuojamam įrašui $z = (\mathbf{x}'; y')$ **do**
3. randami atstumai $d(\mathbf{x}', \mathbf{x})$ tarp z ir $(\mathbf{x}; y) \in E$
4. sudaromas K artimiausių įrašo z kaimynų sąrašas $E_z \subset E$
5. visiems kaimynams $(\mathbf{x}; y) \in E_z$ priskiriami svoriai $w(\mathbf{x}) \geq 0$
6. įrašas z priskiriamas svertinės daugumos klasei
$$y' = \operatorname*{argmax}_c S(c, E_z)$$
7. **end for**

3 .24 pav. Artimiausių kaimynų klasifikatoriaus algoritmas

per dieną surūkantys vyrai nebūtinai abu serga plaučių vėžiu. Tačiau didinant surūkytų cigarečių skaičių, tikimybė priklausyti vėžininkų klasei didėja. Kitaip sakant, reikia rasti klasės tikimybę, priklausomai nuo turimų atributų reikšmių. Čia dažnai praverčia Bajeso teorema, siejanti apriorines ir aposteriorines įvykių tikimybes.

3.6.1 Bajeso formulė ir jos taikymas klasifikavimui

Tarkime, kad atsitiktinių dydžių X ir Y galimų reikšmių aibės yra A_X ir A_Y. Tegul $B_x \subset A_X$, $B_y \subset A_Y$. Atsitiktiniams įvykiams $\{X \in B_x\}$ ir $\{Y \in B_y\}$ Bajeso formulė (1 .3) atrodo taip

$$P(Y \in B_y | X \in B_x) = \frac{P(Y \in B_y) P(X \in B_x | Y \in B_y)}{P(X \in B_x)} .$$

Kai poaibiai B_x ir B_y nėra svarbūs arba jie bus aiškūs iš konteksto, Bajeso formulę rašysime trumpiau

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} . \tag{3 .15}$$

Tikimybė $P(Y)$ vadinama *apriorine*, o sąlyginė tikimybė $P(Y|X)$ - *aposteriorine*, nes ją skaičiuojant atsižvelgta į naują informaciją apie X .

Analogišką išraišką gautume (3 .15) lygybėje X pakeitę imties atributų vektoriumi $\mathbf{X}$

$$\mathbf{X} = (X_1, X_2, \ldots, X_k) .$$

Tegul Y šiuo atveju žymi klasės kintamąjį. Tada apriorinė tikimybė yra lygi klasės tikimybei, kurią galime įvertinti tos klasės mokymo imties įrašų dalimi. Svarbesnės yra aposteriorinės tikimybės $P(Y|\mathbf{X})$. Tarkime, kad pagal mokymo imties duomenis radome šias tikimybes visoms galimoms $\mathbf{X}$ ir Y reikšmių kombinacijoms. Tada tiriamąjį įrašą $(\mathbf{x}', y')$ reikėtų priskirti tai klasei, kurios aposteriorinė tikimybė didžiausia, t.y.

$$y' = \underset{y}{\operatorname{argmax}}\, P(Y = y|\mathbf{X} = \mathbf{x}')$$

Prisiminkime 3.2.1 pavyzdį ir 3 .3 lentelėje pateiktus duomenis apie 10 banko klientų. Kiekvienas iš 10 įrašų sudarytas iš 3 atributų $\mathbf{X} = (X_1, X_2, X_3)$ ir klasės kintamojo Y (`Mokus klientas`) reikšmių. Pirmieji du atributai - X_1 (`Namų valda`) ir X_2 (`Šeimyninė padėtis`) yra kategoriniai, o trečiasis - X_3 (`Metinės pajamos`) - skaitinis. Kiekvienas klientas gali būti mokus ($Y = taip$) arba nemokus ($Y = ne$). Tarkime, banko paskolos paprašė vedęs, neturintis namų valdos klientas, kurio metinės pajamos 60 tūkstančių litų. Bankui aktualus klausimas: ar toks klientas bus mokus? Vienareikšmiškas atsakymas vargu ar įmanomas. Todėl formuluokime klausimą taip: atsižvelgiant į informaciją apie ankstesnius banko klientus, kas labiau tikėtina - naujasis klientas bus mokus ar nemokus? Pabandykime pastarąjį klausimą "matematizuoti". Naujojo kliento atributų reikšmės yra

$$\mathbf{x}' = (ne,\ vedęs,\ 60)\,.$$

Taigi atsakymą į suformuluotąjį klausimą nulems aposteriorinių tikimybių tarpusavio santykis. Jei

$$P(Y = taip\,|\,\mathbf{X} = \mathbf{x}') > P(Y = ne\,|\,\mathbf{X} = \mathbf{x}')\,, \tag{3.16}$$

tai manysime, kad klientas bus mokus.

Norint tiesiogiai skaičiuoti aposteriorines tikimybes, gali prireikti labai didelių mokymo imčių net ir esant mažam atributų skaičiui. Pavyzdžiui, turima banko klientų imtis neleidžia betarpiškai skaičiuoti nė vienos iš (3.16) tikimybių, nes imtyje nėra nė vieno įrašo, tenkinančio sąlygą $\{\mathbf{X} = \mathbf{x}'\}$. Todėl reikia supaprastinti aposteriorinių tikimybių skaičiavimą (tiksliau - jų palyginimo procedūrą). Čia ir padės Bajeso formulė. Mūsų atveju (3.15) lygybę galime perrašyti taip

$$P(Y = y\,|\,\mathbf{X} = \mathbf{x}') = \frac{P(Y = y)\,P(\mathbf{X} = \mathbf{x}'\,|\,Y = y)}{P(\mathbf{X} = \mathbf{x}')}\,. \tag{3.17}$$

Pastebėsime, kad pastarosios lygybės dešinėje pusėje esančios trupmenos vardiklis nepriklauso nuo y . Todėl aposteriorinių tikimybių tarpusavio santykį apsprendžia skaitiklyje esančios klasės apriorinė tikimybė $P(Y = y)$ ir sąlyginė tikimybė $P(\mathbf{X} = \mathbf{x}' \,|\, Y = y)$.
Kaip jau minėjome, natūralus apriorinės tikimybės $P(Y = y)$ įvertis yra klasei y priklausančių mokymo imties įrašų dalis. Sudėtingiau įvertinti tikimybę $P(\mathbf{X}|Y)$. Aptarsime du iš galimų būdų: vadinamąjį naivųjį Bajeso metodą ir Bajeso tinklus.

3.6.2 Naivusis Bajeso klasifikatorius

Atsitiktiniai dydžiai (atributai) X_1 ir X_2 yra nepriklausomi, jei

$$P(X_1, X_2) = P(X_1)P(X_2)\,.$$

Analogiškai X_1 ir X_2 vadinami nepriklausomais su sąlyga, kad žinomas atsitiktinis dydis Y, jei

$$P(X_1, X_2|Y) = P(X_1|Y)P(X_2|Y)\,. \tag{3.18}$$

Šios sąvokos nėra ekvivalenčios. Pavyzdžiui, tokie žmogaus atributai kaip ūgis ir gebėjimas rašyti aišku yra priklausomi, nes dar nemokantys rašyti vaikai yra "mažaūgiai". Tačiau, jei žinomas žmogaus amžius, tai tikimybė, kad jis moka rašyti, jau nepriklauso nuo ūgio.
Sąlyginę nepriklausomybę galima apibrėžti ir kitaip

$$P(X_1|X_2, Y) = P(X_1|Y)\,. \tag{3.19}$$

Nesunku įsitikinti, kad šie apibrėžimai ekvivalentūs. Pavyzdžiui, iš (3.19) ir sąlyginės tikimybės apibrėžimo išplaukia

$$\begin{aligned} P(X_1, X_2|Y) &= \frac{P(X_1, X_2, Y)}{P(Y)} = \frac{P(X_1, X_2, Y)}{P(X_2, Y)} \cdot \frac{P(X_2, Y)}{P(Y)} \\ &= P(X_1|X_2, Y) \cdot P(X_2|Y) = P(X_1|Y) \cdot P(X_2|Y)\,. \end{aligned}$$

Naivusis Bajeso metodas taikomas esant prielaidai, kad žinomos klasės įrašo atributai yra nepriklausomi, t.y.

$$P(\mathbf{X}|Y = y) = \prod_{i=1}^{k} P(X_i|Y = y)\,. \tag{3.20}$$

Taigi įrašo su atributų reikšmėmis $\mathbf{x}' = (x_1, x_2, \ldots, x_k)$ klasifikavimui reikalingas aposteriorines tikimybes (3 .17) , visoms klasėms y galime išreikšti taip

$$P(Y = y \,|\, \mathbf{X} = \mathbf{x}') = P(Y = y) \prod_{i=1}^{k} P(X_i = x_i|Y = y)\, \alpha\,, \tag{3 .21}$$

čia daugiklis

$$\alpha = \frac{1}{P(\mathbf{X} = \mathbf{x}')}$$

nepriklauso nuo y. Todėl reikės maksimizuoti tik dešinėje (3 .21) lygybės pusėje esančią apriorinės tikimybės $P(Y = y)$ ir salyginių tikimybių $P(X_i = x_i|Y = y)$ sandaugą. Kategoriniams kintamiesiems X_i tikimybės $P(X_i = x_i|Y = y)$ prilyginamos santykiui

$$P(X_i = x_i|Y = y) = \frac{n_y(i)}{n_y}\,. \tag{3 .22}$$

Čia n_y - klasei y priklausančių imties įrašų skaičius, $n_y(i)$ - skaičius tų klasės y įrašų, kuriems $X_i = x_i$. Pavyzdžiui, pagal 3 .3 lentelėje pateiktus duomenis

$$P(X_1 = taip|Y = taip) = \frac{5}{7}\,, \quad P(X_2 = viengungis|Y = ne) = \frac{2}{3}\,.$$

Tolydžiųjų skaitinių kintamųjų atveju galimi du sąlyginių tikimybių $P(X_i = x_i|Y = y)$ skaičiavimo būdai.

1. Skaitinį kintamąjį X_i galima diskretizuoti. Tokio diskretizavimo metodai nagrinėjami 2.3.2 skyrelyje. Tada sąlyginė tikimybė prilyginama santykiui (3 .22). Tik šiuo atveju $n_y(i)$ reiškia skaičių tų klasės y įrašų, kurių atributo X_i reikšmė patenka į tą patį dalinį intervalą kaip ir x_i .
2. Tarkime, kad atsitiktinio dydžio X_i sąlyginė tankio funkcija su sąlyga $Y = y_j$ yra $p_i(x|j)$. Pasirinkę mažą teigiamą konstantą ε tikimybę $P(X_i = x_i|Y = y_j)$ pakeisime tokiu jos įverčiu

$$P(x_i \le X_i \le x_i + \varepsilon|Y = y_j) = \int_{x_i}^{x_i+\varepsilon} p_i(x|j)\, dx \approx \varepsilon \cdot p_i(x_i|j)$$

Įrašę šiuos įverčius į (3 .21) matome, kad konstanta ε neįtakoja aposteriorinių tikimybių tarpusavio santykio. Todėl skaičiavimuose tikimybes $P(X_i = x_i|Y = y_j)$ kartais

tiesiog pakeisime tankio funkcijos reikšmėmis $p_i(x_i|j)$. Dažniausiai pasirenkamas normaliojo dėsnio tankis, priklausantis nuo dviejų parametrų: vidurkio μ_{ij} ir dispersijos σ_{ij}^2

$$p_i(x|j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp\left\{-\frac{(x-\mu_{ij})^2}{2\sigma_{ij}^2}\right\}. \qquad (3.23)$$

Jei parametrai μ_{ij} ir σ_{ij}^2 yra nežinomi, jie keičiami atitinkamais įverčiais: imties vidurkiu $\bar{x}_{ij}$ ir dispersija s_{ij}^2 .

3.6.1 pavyzdys. Pagal 3.3 lentelėje pateiktą imtį rasime aposteriorinių tikimybių įverčius ir patikrinsime ar teisinga (3.16) nelygybė. Tuo pačiu klasifikuosime naują įrašą, kurio atributai yra $\mathbf{x}' = (ne,\ vedęs,\ 60)$.

Pirmiausiai pagal (3.23) formulę įvertinsime skaitinio atributo X_3 (`Metinės pajamos`) sąlygines tikimybes. Kai $Y = ne$, gausime tokius atributo X_3 vidurkio ir dispersijos įverčius

$$\begin{aligned}
\bar{x}_{31} &= \frac{47+42+45}{3} = \frac{134}{3} \approx 44,67\,,\\
s_{31}^2 &= \frac{(47-134/3)^2+(42-134/3)^2+(45-134/3)^2}{3-1} = \frac{19}{3} \approx 6,33\,,\\
s_{31} &= \sqrt{\frac{19}{3}} \approx 2,52\,.
\end{aligned}$$

Taigi gauname tokį sąlyginės tikimybės įvertį

$$\begin{aligned}
P(X_3 = 60|Y = ne) &\approx \frac{1}{\sqrt{2\pi}\,2,52} \exp\left\{-\frac{(60-44,67)^2}{2\cdot 6,33}\right\}\cdot\varepsilon\\
&\approx 5,45\cdot 10^{-10}\cdot\varepsilon\,.
\end{aligned}$$

Kai $Y = taip$, analogiški skaičiavimai rodo, kad

$$\begin{aligned}
\bar{x}_{32} &= \frac{65+50+\cdots+37}{7} = \frac{387}{7} \approx 55,29\,,\\
s_{32}^2 &\approx 753,90\,,\\
s_{32} &\approx 27,46\,,\\
P(X_3 = 60|Y = taip) &\approx \frac{1}{\sqrt{2\pi}\,27,46} \exp\left\{-\frac{(60-55,29)^2}{2\cdot 753,90}\right\}\cdot\varepsilon\\
&\approx 0,015\cdot\varepsilon\,.
\end{aligned}$$

Kategorinių atributų X_1 (`Namų valda`) ir X_2 (`Šeimyninė padėtis`) sąlygines tikimybes įvertinsime atitinkamais santykiniais dažniais. Gautieji rezultatai kartu su imties duomenimis pavaizduoti 3.25 paveiksle. Klasių apriorinių tikimybių įverčiai yra

	X_1	X_2	X_3	Y
1	taip	viengungis	65	taip
2	ne	vedęs	50	taip
3	taip	viengungis	35	taip
4	taip	vedęs	60	taip
5	ne	išsiskyręs	47	ne
6	ne	vedęs	30	taip
7	taip	išsiskyręs	110	taip
8	ne	viengungis	42	ne
9	taip	vedęs	37	taip
10	ne	viengungis	45	ne

Tikimybė	Įvertis
$P(X_1 = ne \vert Y = ne)$	1
$P(X_1 = taip \vert Y = ne)$	0
$P(X_1 = ne \vert Y = taip)$	2/7
$P(X_1 = taip \vert Y = taip)$	5/7
$P(X_2 = viengungis \vert Y = ne)$	2/3
$P(X_2 = vedęs \vert Y = ne)$	0
$P(X_2 = išsiskyręs \vert Y = ne)$	1/3
$P(X_2 = viengungis \vert Y = taip)$	2/7
$P(X_2 = vedęs \vert Y = taip)$	4/7
$P(X_2 = išsiskyręs \vert Y = taip)$	1/7
$P(X_3 = 60 \vert Y = ne)$	$5,45 \cdot 10^{-10} \cdot \varepsilon$
$P(X_3 = 60 \vert Y = taip)$	$0,015 \cdot \varepsilon$

3 .25 pav. Banko klientų imtis ir jos atributų sąlyginės tikimybės

$$P(Y = ne) = \frac{3}{10}\,, \quad P(Y = taip) = \frac{7}{10}\,.$$

Abiejų aposteriorinių tikimybių įverčius rasime pritaikę (3 .21) formulę. Taigi

$$\begin{aligned} P(Y = ne \,|\, \mathbf{X} = \mathbf{x}') &= P(Y = ne) \cdot P(X_1 = ne \vert Y = ne) \cdot P(X_2 = vedęs \vert Y = ne) \\ &\quad \cdot P(X_3 = 60 \vert Y = ne) \cdot \alpha \\ &\approx 0,3 \cdot 1 \cdot 0 \cdot 5,45 \cdot 10^{-10} \cdot \varepsilon \cdot \alpha = 0\,, \end{aligned}$$

ir

$$\begin{aligned} P(Y = taip \,|\, \mathbf{X} = \mathbf{x}') &= P(Y = taip) \cdot P(X_1 = ne \vert Y = taip) \cdot P(X_2 = vedęs \vert Y = taip) \\ &\quad \cdot P(X_3 = 60 \vert Y = taip) \cdot \alpha \\ &\approx 0,7 \cdot \frac{2}{7} \cdot \frac{4}{7} \cdot 0,015 \cdot \varepsilon \cdot \alpha \approx 0,0017 \cdot \varepsilon \cdot \alpha\,. \end{aligned}$$

Matome, kad $P(Y = taip \,|\, \mathbf{X} = \mathbf{x}') > P(Y = ne \,|\, \mathbf{X} = \mathbf{x}')$. Todėl nagrinėjamą įrašą priskiriame klasei $Y = taip$, t.y. labiau tikėtina, kad vedęs, neturintis namų valdos klientas, kurio metinės pajamos 60 tūkstančių litų, bus mokus.

Ką tik išnagrinėtame pavyzdyje galime pastebėti vieną potencialią problemą, kuri kyla turint mažą imtį. Kaip matėme, pirmosios sąlyginės tikimybės $P(\mathbf{X} = \mathbf{x}' \,|\, Y = ne)$ įvertis buvo lygus 0, nes $P(X_2 = vedęs|Y = ne) = 0$. Dabar tarkime, kad $P(X_1 = ne|Y = taip)$ vietoje $2/7$ lygi 0. Tada gautume, kad ir antrosios sąlyginės tikimybės $P(\mathbf{X} = \mathbf{x}' \,|\, Y = taip)$ įvertis būtų lygus 0. Tai reiškia, kad įrašas lieka neklasifikuotas. Tokių problemų išvengsime,

Tikimybė	m-įvertis
$P(X_1 = ne\|Y = ne)$	3/4
$P(X_1 = taip\|Y = ne)$	1/4
$P(X_1 = ne\|Y = taip)$	7/20
$P(X_1 = taip\|Y = taip)$	13/20
$P(X_2 = viengungis\|Y = ne)$	1/2
$P(X_2 = vedęs\|Y = ne)$	1/6
$P(X_2 = išsiskyręs\|Y = ne)$	1/3
$P(X_2 = viengungis\|Y = taip)$	3/10
$P(X_2 = vedęs\|Y = taip)$	1/2
$P(X_2 = išsiskyręs\|Y = taip)$	1/5

3 .11 lentelė. Sąlyginių tikimybių m-įverčiai

jei vietoje (3 .22) sąlyginių tikimybių vertinimui naudosime vadinamąjį m-įvertį

$$P(X_i = x_i|Y = y) = \frac{n_y(i) + mp}{n_y + m}\,. \tag{3 .24}$$

Čia p - apriorinė atributo reikšmės $X_i = x_i$ tikimybė klasėje y. Jei atributas X_i įgyja m_i skirtingų reikšmių, tai dažniausiai pasirenkama $p = 1/m_i$. Neneigiamas parametras m (jis kartais dar vadinamas ekvivalenčios imties dydžiu) valdo įverčio padėtį tarp nemodifikuoto santykio $n_y(i)/n_y$ ir apriorinės tikimybės p . Kuo didesnis m tuo įvertis yra artimesnis p . Kai $m = 0$, gauname (3 .22) įvertį. Parametro m pavadinimas paaiškinamas taip: jei klasėje y prie turimų n_y įrašų pridėsime m fiktyvių įrašų, tarp kurių bus mp tenkinančių sąlygą $X_i = x_i$, tai naujasis jų santykis ir taps lygus m-įverčiui (3 .24).

Palyginsime 3 .25 paveikslo dešinėje lentelėje esančius X_1 ir X_2 sąlyginių tikimybių įverčius su jų modifikuotais analogais. (3 .24) formulėje imsime $m = 3$. Tegul $p = 1/2$ pirmajam

atributui (nes jis dvireikšmis) ir $p = 1/3$ antrajam. Gautuosius m-įverčius matome 3 .11 lentelėje.

Iš naujo apskaičiavę aposteriorines tikimybes, turėsime

$$\begin{aligned} P(Y = ne \,|\, \mathbf{X} = \mathbf{x}') &\approx 0,3 \cdot \frac{3}{4} \cdot \frac{1}{6} \cdot 5,45 \cdot 10^{-10} \cdot \varepsilon \cdot \alpha \approx 0,2 \cdot 10^{-10} \cdot \varepsilon \cdot \alpha\,, \\ P(Y = taip \,|\, \mathbf{X} = \mathbf{x}') &\approx 0,7 \cdot \frac{7}{20} \cdot \frac{1}{2} \cdot 0,015 \cdot \varepsilon \cdot \alpha \approx 0,0018 \cdot \varepsilon \cdot \alpha\,. \end{aligned}$$

Matome, kad įrašo klasifikavimas nepasikeitė, tačiau išvengėme nulinių tikimybių. Todėl mažoms imtims sąlyginių tikimybių m-įvertis (3 .24) yra geresnis už paprastą santykį (3 .22).

3.6.3 Bajeso tinklai

Naivusis Bajeso metodas taikomas, kai imties kintamieji $(X_1, X_2, \ldots, X_k, Y)$ tenkina gana griežtą atributų sąlyginio nepriklausomumo reikalavimą (3 .20). Šiame skyrelyje nagrinėsime lankstesnį modelį, toleruojantį galimą kintamųjų priklausomybę.

Tarkime, kad imties kintamieji vaizduojami grafo viršūnėmis, o bet kurių dviejų kintamųjų tarpusavio priklausomybę (jei tokia yra) atspindi šias viršūnes jungianti briauna.

3.6.1 apibrėžimas. *Sakysime, kad kintamieji sudaro Bajeso tinklą, jei tenkinamos dvi sąlygos:*

1) kintamuosius vaizduojantis grafas yra orientuotas ir neturi ciklų,

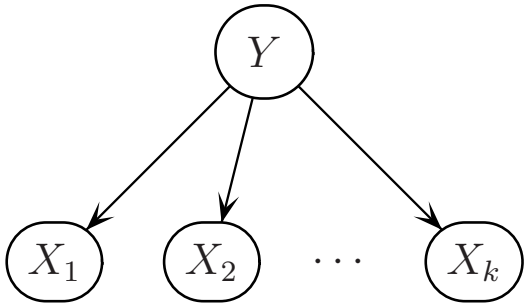
2) jei žinomi viršūnės tėvai, tai pati viršūnė nepriklauso nuo visų likusių viršūnių, išskyrus jos palikuonis.

Pastebėsime, kad (3 .20) sąlygą tenkinantys kintamieji sudaro Bajeso tinklą. Juos atitinkantis grafas pavaizduotas 3 .26 paveiksle. Iš praeito skyrelio žinome, kad tokio grafo viršūnė Y nusakoma apriorinėmis , o likusios viršūnės salyginėmis tikimybėmis. Šį pastebėjimą galima apibendrinti bet kokiam Bajeso tinklui.

Bajeso tinklas nusakomas viršūnėse esančių kintamųjų skirstiniais. Jei viršūnė turi tėvus, tai ją atitinkantis skirstinys yra sąlyginis. Priešingu atveju - nesąlyginis.

Taigi, Bajeso tinklo klasifikatoriaus konstravimas susideda iš dviejų etapų.

1. Nustatoma tinklo struktūra. Tam reikia nurodyti priklausomus kintamuosius.
2. Pagal turimą imtį randami visų viršūnių skirstiniai.



3 .26 pav. Sąlyginis atributų nepriklausomumas

Panagrinėkime tokį pavyzdį. Tegul informacija apie pacientą nusakoma šešiais binariniais kintamaisiais $(X_1, X_2, X_3, X_4, X_5, Y)$:

X_1 - `Rytinė mankšta`,

X_2 - `Sveikas maistas`,

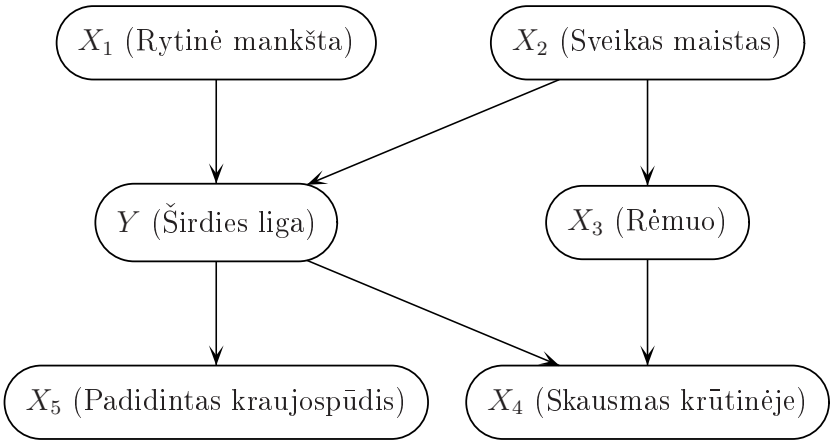
X_3 - `Rėmuo`,

X_4 - `Skausmas krūtinėje`,

X_5 - `Padidintas kraujospūdis`,

Y - `Širdies liga`.

Kiekvienas iš kintamųjų įgyja reikšmę 1, jei atitinkamas požymis yra, ir 0 priešingu atveju. Atsižvelgiant į medikų nuomonę nustatyta šių požymių tarpusavio priklausomybė nusakoma 3 .27 paveiksle pavaizduotu grafu. Pavyzdžiui, matome, kad rytinė mankšta įtakoja širdies



3 .27 pav. Bajeso tinklas ligonių duomenims

ligų riziką, bet neturi įtakos sveikos mitybos įpročiams ar rėmens atsiradimui.

Tarkime, kad pagal turimos imties duomenis 70% pacientų rytais sportuoja ir tik 25% sveikai maitinasi. Todėl tėvų neturinčių kintamųjų X_1 ir X_2 skirstinius nusakys lygybės

$$P(X_1 = 0) = 1 - P(X_1 = 1) = 0,3 \quad \text{ir} \quad P(X_2 = 0) = 1 - P(X_2 = 1) = 0,75\,. \qquad (3.25)$$

Likusios keturios viršūnės turi tėvus. Todėl pagal tos pačios imties duomenis joms randami atitinkami sąlyginiai skirstiniai. Tokiems skirstiniams reikalingų salyginių tikimybių įverčių skaičiavimo būdai buvo aptarti 3.6.2 skyrelyje. Jų nekartodami tarsime, kad kintamųjų X_3, X_4, X_5 ir Y sąlyginiai skirstiniai yra tokie, kaip pavaizduota 3.28 paveikslo lentelėse.

	X_3	
X_2	0	1
0	0,15	0,85
1	0,8	0,2

		X_4	
X_3	Y	0	1
0	0	0,9	0,1
0	1	0,4	0,6
1	0	0,6	0,4
1	1	0,2	0,8

	X_5	
Y	0	1
0	0,8	0,2
1	0,15	0,85

		Y	
X_1	X_2	0	1
0	0	0,25	0,75
0	1	0,45	0,55
1	0	0,55	0,45
1	1	0,75	0,25

3.28 pav. Sąlyginiai skirstiniai ligonių duomenims

3.27 paveiksle pavaizduotas grafas, (3.25) lygybės ir 3.28 paveiksle pateikiami skirstiniai pilnai nusako Bajeso tinklą, leidžiantį klasifikuoti pacientus pagal nagrinėjamų atributų reikšmes. Panagrinėsime keletą tokio klasifikatoriaus taikymo pavyzdžių.

1. Kokia tikimybė, kad pacientas serga širdies liga? Jokios papildomos informacijos apie pacientą neturime.

   Pagal pilnosios tikimybės formulę (1.2) ieškomoji tikimybė yra

$$\begin{aligned} P(Y=1) &= \sum_{i,j\in\{0,1\}} P(Y=1|X_1=i, X_2=j)\, P(X_1=i, X_2=j) \\ &= \sum_{i,j\in\{0,1\}} P(Y=1|X_1=i, X_2=j)\, P(X_1=i)\, P(X_2=j) \\ &= 0,75\cdot 0,3\cdot 0,75 + 0,55\cdot 0,3\cdot 0,25 + 0,45\cdot 0,7\cdot 0,75 \\ &\quad +0,25\cdot 0,7\cdot 0,25 = 0,49\,. \end{aligned}$$

Taigi turimi duomenys rodo, kad su 49% tikimybe atsitiktinis pacientas serga širdies liga.

2. Kaip aukštas kraujospūdis įtakoja riziką susirgti širdies liga?

   Palyginsime tikimybes

$$P(Y=1|X_5=1) \quad \text{ir} \quad P(Y=0|X_5=1) = 1 - P(Y=1|X_5=1)\,.$$

   Tuo tikslu pirmiausiai rasime tikimybę $P(X_5=1)$. Vėl pritaikę pilnosios tikimybės formulę, turėsime

$$\begin{aligned} P(X_5=1) &= \sum_{i\in\{0,1\}} P(X_5=1|Y=i)\,P(Y=i) \\ &= 0,2\cdot 0,51 + 0,85\cdot 0,49 = 0,5185\,. \end{aligned}$$

   Dabar pagal Bajeso formulę (3 .15) gausime

$$\begin{aligned} P(Y=1|X_5=1) &= \frac{P(X_5=1|Y=1)\,P(Y=1)}{P(X_5=1)} \\ &= \frac{0,85\cdot 0,49}{0,5185} \approx 0,8033\,. \end{aligned}$$

   Todėl

$$P(Y=0|X_5=1) \approx 1 - 0,8033 = 0,1967\,.$$

   Tai rodo, kad aukštas kraujospūdis ženkliai padidina širdies ligos riziką.

3. Turintis aukštą kraujospūdį pacientas reguliariai mankštinasi ir valgo sveiką maistą. Kaip pasikeis širdies ligos prognozė, lyginant su prieš tai išnagrinėta situacija.

   Dabar reikalingą aposteriorinę tikimybę galime išreikšti taip

$$\begin{aligned} & P(Y=1|X_5=1,X_2=1,X_1=1) \\ &= \frac{P(X_5=1|Y=1,X_2=1,X_1=1)\,P(Y=1|X_2=1,X_1=1)}{P(X_5=1|X_2=1,X_1=1)} \\ &= \frac{P(X_5=1|Y=1)\,P(Y=1|X_2=1,X_1=1)}{\sum_{i\in\{0,1\}} P(X_5=1|Y=i)P(Y=i|X_2=1,X_1=1)} \\ &= \frac{0,85\cdot 0,25}{0,2\cdot 0,75 + 0,85\cdot 0,25} \approx 0,5862\,. \end{aligned}$$

Tikimybė, kad pacientas neserga bus

$$P(Y = 0|X_5 = 1, X_2 = 1, X_1 = 1) \approx 1 - 0,5862 = 0,4138\,.$$

Matome, kad mankšta ir sveika mityba sumažina galimybę susirgti širdies liga.

Konstruojant Bajeso tinklą, reikia žinoti kintamųjų tarpusavio sąryšius. Todėl taikant šį modelį reikia gerai žinoti duomenų prigimtį. Be to, esant dideliam kintamųjų skaičiui, tinkamos tarpusavio sąryšio schemos paieška gali pareikalauti daug pastangų ir laiko.

4 Kontroliuojamo mokymo uždaviniai: skaitinė prognozė

Nagrinėsime duomenis, kurių visi nepriklausomi kintamieji (atributai)

$$\mathbf{X} = (X_1, X_2, \ldots, X_k)$$

ir priklausomas (klasės) kintamasis Y yra skaitiniai. Tokių duomenų n įrašų imtis yra

$$E = \{(\mathbf{x}_i, y_i)\,,\ i = 1, 2, \ldots, n\}\,.$$

Jos i - tasis įrašas susideda iš k atributų reikšmių $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$ ir priklausomo kintamojo Y reikšmės y_i. Aptarsime skaitinės prognozės (kitaip dar vadinamus *regresijos*) modelius, leidžiančius kintamojo Y reikšmes prognozuoti pagal žinomas atributų vektoriaus $\mathbf{X}$ reikšmes. Iš tikrųjų yra konstruojamas kintamojo Y įvertis (modelis)

$$\hat{Y} = f(\mathbf{X})\,, \tag{4.1}$$

stengiantis, kad jis kuo tiksliau atspindėtų imties duomenis. Aišku sunku tikėtis, kad modelis tiksliai atitiks visus įrašus, t.y. $f(\mathbf{x}_i) = y_i$ visiems $i = 1, 2, \ldots, n$. Kiek įvertis atitinka tikrąjį kintamąjį Y nusako vadinamoji klaidų (nuostolių) funkcija $L(Y, f(\mathbf{X}))$. Konstruojamas toks modelis (4.1), kad klaidų funkcijos reikšmių $L(y_i, f(\mathbf{x}_i))$ suma visoje imtyje būtų kuo mažesnė. Rezultatas žinoma priklauso tiek nuo klaidų funkcijos pasirinkimo, tiek ir nuo funkcijos f analizinių savybių.

4.1 Paprastosios tiesinės regresijos modelis

Tarkime, kad imtis E turi tik $k = 1$ atributą $X = X_1$, t.y. aibė E sudaryta iš n plokštumos taškų

$$E = \{(x_i, y_i)\,,\ i = 1, 2, \ldots, n\}\,.$$

4.1.1 pavyzdys. Žinoma kurortinio miestelio keliolikos vasaros dienų vidutinė dienos temperatūra (C°) ir vietos restorane suvalgytų ledų kiekis (kg.). Duomenys pateikti 4.1 lentelėje. Kokia priklausomybė tarp temperatūros (atributas X) ir ledų kiekio (kintamasis Y)? 4.1 paveiksle visi 24 imties E įrašai pavaizduoti plokštumos taškais. Jų išsidėstymas rodo tarp X ir Y gana aiškiai išreikštą tiesinę priklausomybę.

Temperatūra (X)	25	26	24	26	24	26	22	23	27	20	20	22
Ledai (kg.) (Y)	116	120	115	119	115	118	111	113	121	108	109	110
Temperatūra (X)	28	22	23	23	28	24	26	29	25	25	25	24
Ledai (kg.) (Y)	122	113	113	114	123	116	119	125	118	119	117	116

4.1 lentelė. Ledų pardavimas

Jei turimos imties vaizdas yra panašus į tokią "ledų" imtį, tai natūralu nagrinėti tiesinį modelį

$$f(X) = a + bX\,. \tag{4.2}$$

4.1.1 Regresijos tiesė

Pasirinkę kvadratinę klaidų funkciją

$$L(Y, f(X)) = (Y - f(X))^2\,,$$

rasime tiesinio modelio (4.2) koeficientų a ir b įverčius, minimizuojančius suminę klaidą

$$S(a,b) = \sum_{i=1}^{n} L(y_i, f(x_i)) = \sum_{i=1}^{n} (y_i - a - bx_i)^2\,. \tag{4.3}$$

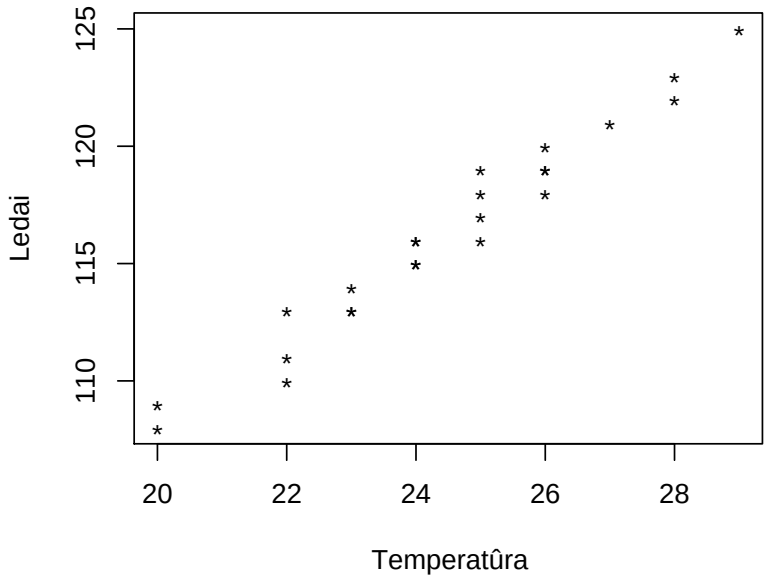
Matematine prasme uždavinys nėra sudėtingas. Reikia rasti dviejų kintamųjų funkcijos $S(a,b)$ minimumo tašką. Atsakymą gausime prilyginę nuliui šios funkcijos dalines išvestines ir išsprendę tiesinių lygčių sistemą

$$\begin{cases} \dfrac{\partial S(a,b)}{\partial a} &= -2\displaystyle\sum_{i=1}^{n}(y_i - a - bx_i) = 0\,, \\ \dfrac{\partial S(a,b)}{\partial b} &= -2\displaystyle\sum_{i=1}^{n}(y_i - a - bx_i)x_i = 0\,. \end{cases} \tag{4.4}$$

Gautasis sprendinys yra

$$\hat{b} = \frac{\displaystyle\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\displaystyle\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{s_y}{s_x} \cdot r\,, \tag{4.5}$$

$$\hat{a} = \bar{y} - \hat{b}\bar{x}\,, \tag{4.6}$$



4 .1 pav. Ledų pardavimo priklausomybė nuo temperatūros

čia

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i\,, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i\,, \quad s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2\,, \quad s_y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i-\bar{y})^2$$

žymi vidurkio ir dispersijos įverčius, o simetriškas x ir y atžvilgiu reiškinys

$$r = \frac{1}{s_x s_y (n-1)}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y}) \tag{4 .7}$$

yra kintamųjų X ir Y koreliacijos koeficiento

$$\rho(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sqrt{\mathbf{D}X\cdot\mathbf{D}Y}} = \frac{\mathbf{E}XY - \mathbf{E}X\mathbf{E}Y}{\sqrt{\mathbf{D}X\cdot\mathbf{D}Y}}$$

įvertis. Mažiausią funkcijos $S(a,b)$ reikšmę įprasta žymeti SSE (Sum of the Squared Error[1]) Taigi

$$SSE = S(\hat{a},\hat{b}) = \sum_{i=1}^{n}(y_i - \hat{a} - \hat{b}x_i)^2\,.$$

Aprašytasis modelio parametrų $\hat{a}$ ir $\hat{b}$ parinkimo metodas vadinamas *mažiausių kvadratų metodu.*

[1] (Angl.) Paklaidų kvadratų suma

4.1.1 apibrėžimas. *Lygtis*

$$\hat{y}(x) = \hat{a} + \hat{b}x\,,$$

kurios koeficientai $\hat{a}$ *ir* $\hat{b}$ *apibrėžti* (4.6) *ir* (4.5) *lygybėmis, vadinama regresijos tiesės lygtimi.*

$$\hat{e}_i = y_i - \hat{y}(x_i) = y_i - \hat{a} - \hat{b}x_i$$

vadinama i *- tąja liekamąja paklaida,* $i = 1, 2, \ldots, n$ *.*

Įrašę $\hat{a}$ ir $\hat{b}$ išraiškas, regresijos tiesės lygtį galime pertvarkyti taip

$$\hat{y}(x) = \bar{y} + \frac{s_y}{s_x} \cdot r \cdot (x - \bar{x})\,. \tag{4.8}$$

Grįžkime prie 4.1.1 pavyzdžio apie ledų prekybą. Pagal 4.1 lentelėje pateiktus duomenis paeiliui randame

$$\begin{aligned}
\bar{x} &\approx 24,458\,, \quad s_x^2 \approx 5,563\,,\\
\bar{y} &= 116,25\,, \quad s_y^2 = 19,5\,,\\
r &\approx 0,982\,,\\
\hat{b} &\approx 1,838\,, \quad \hat{a} \approx 71,284\,.
\end{aligned}$$

Taigi regresijos tiesės lygtis yra

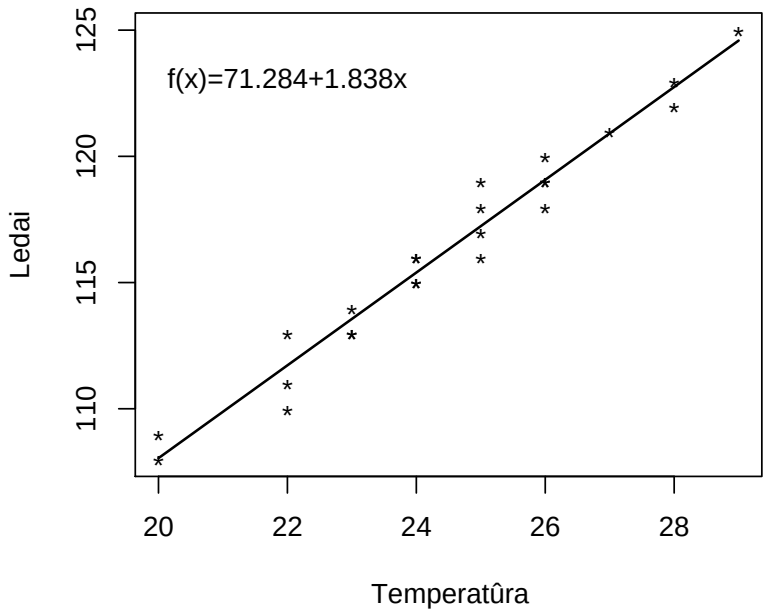
$$\hat{y}(x) = 71,284 + 1,838x\,. \tag{4.9}$$

Iš 4.2 paveiksle pateikto grafiko matome, kad ji pakankamai tiksliai atspindi turimus duomenis. Todėl galime prognozuoti, kad esant , pavyzdžiui, 21 laipsnio temperatūrai, parduodamas ledų kiekis bus maždaug

$$\hat{y}(21) = 71,284 + 1,838 \cdot 21 = 109,882\,\text{kg}.$$

Kai regresijos modelio kintamuosius sieja dvipusė priklausomybė, galima rasti tų pačių duomenų *atvirkštinės regresijos tiesę.* Turint (4.8) išraišką, tai visai nesunku padaryti. Kaip matėme, koreliacijos koeficientas r yra simetriškas kintamųjų atžvilgiu. Todėl pakanka (4.8) lygybėje x ir y sukeisti vietomis. Gausime tokią atvirkštinės regresijos tiesės lygtį

$$\hat{x}(y) = \bar{x} + \frac{s_x}{s_y} \cdot r \cdot (y - \bar{y})\,. \tag{4.10}$$



4 .2 pav. Ledų pardavimo regresijos tiesė

Nors formaliai mes visada galime užrašyti abiejų regresijos tiesių lygtis, tačiau reikia nepamiršti sprendžiamo uždavinio specifikos. Vėl prisiminkime mūsų kurortinio miestelio restoraną. Jei restorano savininkas rytojui užsakys 110 kilogramų ledų, tai vargu ar toks užsakymas kaip nors paveiks rytojaus meteorologinę situaciją visame miestelyje. Taigi šiuo atveju prasminga yra tik (4 .8) lygtis. Dabar tarkime, kad mūsų "prognozė" nukreipta ne į ateitį, o priešingai - į praeitį. Jei vartydami restorano buhalterinius dokumentus, bandysime atspėti koks oras buvo prieš metus, tai kaip tik pravers (4 .10) lygtis.

Kita klaida gali atsirasti mechaniškai taikant modelį nepriklausomo kintamojo reikšmėms už stebimo intervalo ribų. Pavyzdžiui, pagal (4 .9) lygtį, esant 20 laipsnių šalčiui, bus parduota

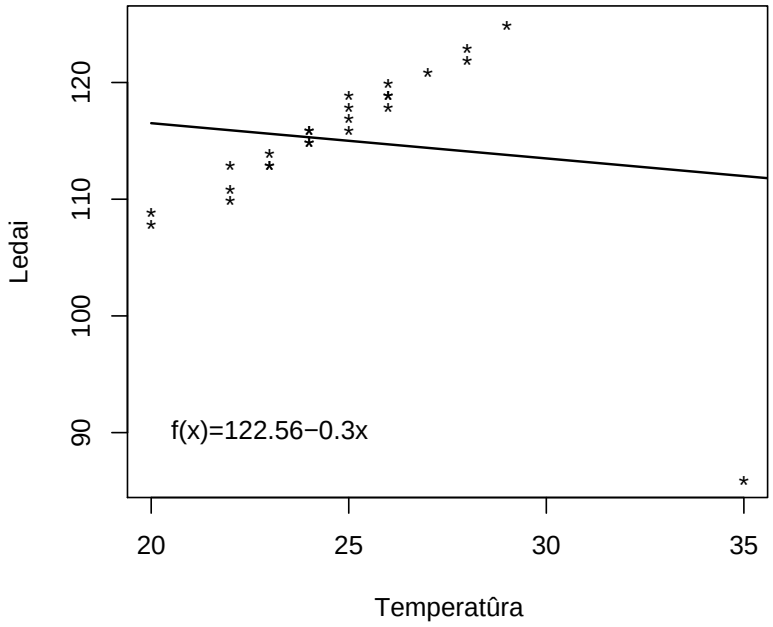
$$\hat{y}(-20) = 71,284 + 1,838 \cdot (-20) = 34,524\,\text{kg}.$$

ledų! Vargu ar kas tuo patikės.

Galimi ir kitokie "nesusipratimai" neatsargiai interpretuojant regresijos modelį. Aptarsime keletą charakteristikų, padedančių pagrįsti modelio tinkamumą sprendžiamam uždaviniui.

4.1.2 Regresijos modelio charakteristikos

Net ir vienas, labai nuo kitų besiskiriantis įrašas (x_j, y_j) gali radikaliai pakeisti regresijos tiesę. Tarkime, kad 4.1 lentelės duomenys yra papildyti: dieną, kai temperatūra siekė 35 lapsnius karščio, suvalgyti 86 kilogramai ledų. Kaip šiuo atveju atrodo imties duomenys ir regresijos tiesė bei jos lygtis, matome 4.3 paveiksle. Nors visi, išskyrus vieną, stebėjimai



4.3 pav. Ledų pardavimo regresijos tiesė duomenims su išskirtimi

rodo, kad kylant temperatūrai ledų suvartojama daugiau, regresijos tiesė yra mažėjanti. Kažin, ar tokia regresijos tiesė tinka prognozėms. Tokie besiskiriantys nuo kitų duomenų įrašai vadinami *išskirtimis.* Išskirtys - tai savotiški įtartini taškai. Išskirtis nebūtinai iš esmės pakeičia parametrų įverčius.

Taigi reikia išmokti rasti duomenų išskirtis ir po to išsiaiškinti, ką su tomis išskirtimis daryti. Iš pradžių aptarsime, kaip sprendžiama antroji problema.

Tyrėjas didesnį dėmesį turėtų skirti toms išskirtims, kurios labai keičia modelio elgesį. Ar išskirtis "kenksminga", paprasčiausiai nustatyti sudarant regresijos tiesės lygtį su išskir-

timi ir be jos. Ką daryti, suradus parametrų įverčius keičiančias išskirtis? Pirmiausiai rekomenduojama patikrinti, ar įrašuose nėra klaidos. Pavyzdžiui, gali paaiškėti, kad turėjo būti ne (35,90), o (35,190). Kartais įrašas atskleidžia visiškai nebūdingą situaciją ar stiprų kokio nors pašalinio faktoriaus poveikį. Pavyzdžiui, dėl orų permainas lydėjusio viesulo išvakarėse sutrikdytas elektros tiekimas. Minėtais atvejais išskirtis galima pašalinti. Dažnai apie išskirtį jokios informacijos neturime. Tad neišsiaiškinus, kaip atsirado duomenų išskirtis, jos negalima šalinti. Tuomet rekomenduojama imtį papildyti naujais įrašais ir tyrimą kartoti. Nepamirškime, kad regresijos modeliai tinka ne visiems duomenims. Gali tiesiog paaiškėti, kad tiesinės regresijos modelis neatitinka stebimų kintamųjų elgesio.

Minėtame pavyzdyje išskirtis buvo langvai pastebima. Tačiau kaip ją rasti bendruoju atveju?

Taikomi įvairūs išskirčių nustatymo kriterijai. Susipažinsime su trimis iš jų, kai išskirtis nustatoma pagal: 1) įrašo įtakos indeksą; 2) standartizuotąją liekaną; 3) Kuko matą.

1. *Įrašo įtakos indeksas* įvertina tik nepriklausomo kintamojo reikšmę (ar toli nuo $\bar{x}$ yra x_j). Įrašo (x_j, y_j) įtakos indeksas

$$h_j = \frac{1}{n} + \frac{(x_j - \bar{x})^2}{(n-1)s_x^2}\,.$$

Kuo regresijos tiesės lygties koeficientai labiau priklauso nuo įrašo, tuo jo įtakos indeksas didesnis. Dažniausiai vadovaujamasi tokia taisykle:

$$\textit{Įrašą } (x_j, y_j) \textit{ laikome išskirtimi, jei } h_j > \frac{4}{n}\,.$$

2. *Standartizuotosios liekanos* yra liekamųjų paklaidų $\hat{e}_j$ z-standartizuotos reikšmės (žr. 2.3.1 skyrelį)

$$SR_j = \hat{e}_j \sqrt{\frac{n-2}{(1-h_j)SSE}}\,.$$

Standartizuotųjų liekanų imties vidurkis lygus nuliui, o imties dispersija - vienetui. Todėl, pagal tikimybių teorijoje žinomą "trijų sigma" taisyklę

$$\textit{Įrašą } (x_j, y_j) \textit{ laikome išskirtimi, jei } |SR_j| > 3\,.$$

3. *Kuko matas* atsižvelgia ir į standartizuotąją liekaną ir į įrašo įtakos indeksą. Kuko matas

$$D_j = \frac{(SR_j)^2 h_j}{2(1-h_j)}\,.$$

Kaip matyti iš šios formulės, Kuko matas didelis tada, kai didelis įrašo įtakos indeksas arba didelė standartizuotoji liekana. Supaprastinta, tačiau gana gera taisyklė yra tokia

$$\textit{Įrašą } (x_j, y_j) \textit{ laikome išskirtimi, jei } D_j > 1\,.$$

Kita svarbi tiesinės regresijos modelio charakteristika yra *determinacijos koeficientas*, nusakantis nepriklausomo ir priklausomo kintamųjų priklausomybės stiprumą.
Jau anksčiau susidūrėme su paklaidų kvadratų suma SSE.

$$SSE = \sum_{i=1}^{n} \hat{e}_i^2\,.$$

Apibrėšime dar dvi panašias sumas. Paklaidą $\hat{e}_i = y_i - \hat{y}(x_i)$ galima išskaidyti į dvi sudėtines dalis:

$$\hat{e}_i = y_i - \bar{y} + \bar{y} - \hat{y}(x_i)\,.$$

Abi šios lygybės puses pakėlę kvadratu, turėsime

$$\hat{e}_i^2 = (y_i - \bar{y})^2 + (\bar{y} - \hat{y}(x_i))^2 + 2(y_i - \bar{y})(\bar{y} - \hat{y}(x_i))\,. \tag{4.11}$$

Rasime paskutiniojo dėmens sumą pagal visus i. Iš regresijos tiesės lygties (4.8) ir (4.5) išraiškos išplaukia, kad

$$\bar{y} - \hat{y}(x_i) = -\hat{b}(x_i - \bar{x})\,,$$

ir

$$\begin{aligned}\sum_{i=1}^{n}(y_i - \bar{y})(\bar{y} - \hat{y}(x_i)) &= -\hat{b}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) = -\hat{b}^2\sum_{i=1}^{n}(x_i - \bar{x})^2 \\ &= -\sum_{i=1}^{n}(\hat{y}(x_i) - \bar{y})^2\,.\end{aligned}$$

Todėl sumuodami abi (4.11) lygybės puses pagal i, gausime sąryšį

$$\sum_{i=1}^{n}\hat{e}_i^2 = \sum_{i=1}^{n}(y_i - \bar{y})^2 - \sum_{i=1}^{n}(\hat{y}(x_i) - \bar{y})^2\,,$$

kurį sutrumpintai galime parašyti taip

$$SSE = SST - SSR\,; \tag{4.12}$$

čia

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2\,, \quad SSR = \sum_{i=1}^{n}(\hat{y}(x_i) - \bar{y})^2\,, \quad SSE = \sum_{i=1}^{n}(y_i - \hat{y}(x_i))^2\,. \tag{4.13}$$

SST (Total Sum of Squares) vadinama visa kvadratų suma, SSR (Regression Sum of Squares) - regresijos kvadratų suma. Pastebėsime, kad $SST = (n-1)s_y^2$.

SST įvertina, kaip stebimos Y reikšmės išsibarsčiusios apie tiesę $y = \bar{y}$. SSE įvertina, kaip stebimos Y reikšmės išsibarsčiusios apie regresijos tiesę $\hat{y}(x) = \hat{a} + \hat{b}x$. SSR - tai kvadratų suma, rodanti, kiek regresijos tiesė skiriasi nuo tiesės $y = \bar{y}$.

4.1.2 apibrėžimas. *Regresijos modelio determinacijos koeficientu vadinamas santykis*

$$R^2 = \frac{SSR}{SST}\,.$$

Aišku, kad determinacijos koeficientas neviršija vieneto: $R^2 \leq 1$. Tarkime, kad imties duomenys idealiai atitinka regresijos tiesės lygtį, t.y. visi įrašus (x_i, y_i) atitinkantys taškai priklauso regresijos tiesei. Tuo atveju $SSE = 0$, $SSR = SST$ ir $R^2 = 1$. Dabar tarkime, kad regresijos tiesės lygtis visiškai netinkama prognozei, t.y. $SSR = 0$. Tuomet $R^2 = 0$.

Determinacijos koeficientas plačiai naudojamas kaip regresijos modelio tinkamumo indikatorius. Didesnis determinacijos koeficientas reiškia, kad įrašai yra labiau koncentruoti apie regresijos tiesę.

Dažniausiai reikalaujama, kad būtų tenkinama nelygybė $R^2 \geq 0,25$. Jeigu $R^2 < 0,25$, labai abejotina, ar tiesinės regresijos modelis tinka.

Determinacijos koeficientas glaudžiai susijęs su vadinamuoju Pirsono koreliacijos koeficientu r, apibrėžtu (4.7) formule. Paprastosios tiesinės regresijos (vieno nepriklausomo kintamojo) atveju iš (4.8) išplaukia

$$R^2 = \frac{SSR}{SST} = \frac{1}{(n-1)s_y^2}SSR = \frac{1}{(n-1)s_y^2}\left(\frac{s_y}{s_x}\cdot r\right)^2 \sum_{i=1}^{n}(x - \bar{x})^2 = r^2\,.$$

Taigi įrodėme tokį teiginį.

4.1.1 teorema. *Paprastosios tiesinės regresijos atveju Pirsono koreliacijos koeficiento modulis yra lygus kvadratinei šakniai iš determinacijos koeficiento*

$$|r| = \sqrt{R^2}\,.$$

Jo ženklas sutampa su $\hat{b}$ ženklu.

Atkreipiame dėmesį, kad taip yra tik paprastosios tiesinės regresijos atveju. Daugiamatį atvejį aptarsime kitame skyrelyje.

4.2 Daugialypė tiesinė regresija

Daugialypės tiesinės regresijos modelis yra paprastosios regresijos modelio apibendrinimas, kai nepriklausomų kintamųjų yra daugiau nei vienas.

4.2.1 Daugialypės tiesinės regresijos modelis

Nagrinėsime (4.1) modelį, kai f yra tiesinė funkcija

$$f(\mathbf{X}) = \beta_0 + \sum_{j=1}^{k} \beta_j X_j \,.$$

Užrašysime šią funkciją matricų sandaugos pavidalu. Vektorių $\mathbf{X}$ papildysime vienetine koordinate, o modelio koeficientų vektorių žymėsime $\boldsymbol{\beta}$, t.y.

$$\mathbf{X} = (1, X_1, X_2, \ldots, X_k)\,, \quad \boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \ldots, \beta_k)\,.$$

Dabar mūsų modelio išraiška mažai kuo skirsis nuo (4.2)

$$f(\mathbf{X}) = \boldsymbol{\beta} \cdot \mathbf{X}^T\,; \tag{4.14}$$

čia, kaip įprasta, $\mathbf{X}^T$ žymi transponuotą matricą (šiuo atveju sudarytą iš vieno stulpelio). Modifikavus atributų vektorių $\mathbf{X}$, imties įrašai

$$E = \{(\mathbf{x}_i, y_i)\,,\ i = 1, 2, \ldots, n\}\,.$$

bus sudaryti iš atributų reikšmių vektoriaus $\mathbf{x}_i = (1, x_{i1}, x_{i2}, \ldots, x_{ik})$ ir priklausomo kintamojo Y reikšmės y_i. Visų atributų reikšmių $n \times (k+1)$ matricą, sudarytą iš vektorių $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ koordinačių, žymėsime $\mathfrak{X}$, t.y.

$$\mathfrak{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1k} \\ 1 & x_{21} & x_{22} & \ldots & x_{2k} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nk} \end{pmatrix}.$$

Modelio koeficientus $\boldsymbol{\beta}$ rasime mažiausių kvadratų metodu, t.y. minimizuodami suminę klaidą

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 = \sum_{i=1}^{n} (y_i - \boldsymbol{\beta} \cdot \mathbf{x}_i^T)^2 \,. \tag{4.15}$$

Tegul $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$ yra priklausomo kintamojo reikšmių stulpelis. Naudodami įvestus žymėjimus, funkciją $S(\boldsymbol{\beta})$ galime įšreikšti matricų sandauga

$$S(\boldsymbol{\beta}) = (\mathbf{y} - \boldsymbol{\mathfrak{X}}\boldsymbol{\beta}^T)^T (\mathbf{y} - \boldsymbol{\mathfrak{X}}\boldsymbol{\beta}^T) \,.$$

Diferencijuodami pagal $\boldsymbol{\beta}$, gausime tokį (4.4) lygčių sistemos analogą

$$(\mathbf{y}^T - \boldsymbol{\beta}\boldsymbol{\mathfrak{X}}^T)\boldsymbol{\mathfrak{X}} = \mathbf{0} \,; \tag{4.16}$$

čia $\mathbf{0} = (0)_{1\times(k+1)}$ - eilutė iš $k+1$ nulio.

Jei matrica $\boldsymbol{\mathfrak{X}}^T\boldsymbol{\mathfrak{X}}$ neišsigimusi, tai (4.16) lygtis turi vienintelį sprendinį

$$\hat{\boldsymbol{\beta}} = \mathbf{y}^T \boldsymbol{\mathfrak{X}} (\boldsymbol{\mathfrak{X}}^T \boldsymbol{\mathfrak{X}})^{-1} \,. \tag{4.17}$$

Tai ir yra (4.14) modelio koeficientai, minimizuojantys suminę klaidą (4.15). Iš čia, (4.1) ir (4.14) išplaukia tiesinės daugialypės regresijos lygtis:

$$\hat{y}(\mathbf{x}) = \hat{\boldsymbol{\beta}} \cdot \mathbf{x}^T \,; \tag{4.18}$$

čia $\mathbf{x} = (1, x_1, x_2, \ldots, x_k)\,, \quad \hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)\,.$

Ne visuomet kintamuosius sieja tiesinė priklausomybė. Tuomet reikia rinktis kito tipo modelį arba kintamuosius transformuoti. Transformuoti kintamuosius siūloma tada, kai pačių kintamųjų priklausomybė yra netiesinė, tačiau jų transformacijų priklausomybė - tiesinė. Pavyzdžiui, ekonomistas gali spėti, jog naudingumo funkciją (U) su kapitalo (K) bei darbo (D) resursais sieja tokia priklausomybė:

$$U = aK^b D^c \,. \tag{4.19}$$

Koeficientai a, b, c nežinomi ir juos reikia įvertinti. Tokiai priklausomybei sudaryti regresijos modelį ganėtinai sudėtinga. Tačiau išlogaritmavę (4.19), gauname tiesinę naujų kintamųjų

$$X_1 = \ln K\,, \quad X_2 = \ln D \quad \text{ir} Y = \ln U$$

priklausomybę

$$\ln U = \ln a + b \ln K + c \ln D \,.$$

Taigi galėsime tirti tiesinės regresijos modelį

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \,.$$

Panašiai sprendžiama problema, kai į modelį reikia įtraukti, X_1^2, X_2^2 arba $X_1 X_2$.
Kartais kintamųjų negalima transformuoti taip, kad tiesinė regresija tiktų. Tuo atveju galima taikyti netiesinę regresiją. Plačiau jos nenagrinėsime, tačiau pažymėsime, kad daugumoje statistinių programinių paketų ji yra realizuota.

4.2.2 Determinacijos ir koreliacijos koeficientai

Skaitiškai vertinant nepriklausomų kintamųjų įtaką Y įgyjamoms reikšmėms, skaičiuojamas determinacijos koeficientas, koreguotasis determinacijos koeficientas ir daugialypės koreliacijos koeficientas.

1. *Daugialypės determinacijos koeficientas* apibrėžiamas taip pat kaip ir vieno nepriklausomo kintamojo atveju. Tai yra santykis

$$R^2 = \frac{SSR}{SST} \,.$$

Čia

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2 \,, \quad SSR = \sum_{i=1}^{n} (\hat{y}(\mathbf{x}_i) - \bar{y})^2 \,.$$

Determinacijos koeficientas $R^2 \leq 1$. Galima sakyti, kad kuo R^2 reikšmė didesnė, tuo daugiau informacijos apie Y suteikia kintamieji $X_1, X_2, \ldots, X_k$. Taigi tuo geriau tinka ir pasirinktasis regresijos modelis.
Tačiau, jei nepriklausomų kintamųjų skaičius k nedaug skiriasi nuo įrašų skaičiaus n, tai vien todėl determinacijos koeficientas gali būti arti vieneto. Todėl į R^2 rekomenduojama atsižvelgti tik tada, kai k daug mažesnis už n. Kitais atvejais skaičiuojamas koreguotasis determinacijos koeficientas.

2. *Koreguotasis determinacijos koeficientas* R^2_{adj} yra lygus

$$R^2_{adj} = 1 - \frac{n-1}{n-k-1}(1 - R^2) \,.$$

Matome, kad jo reikšmė priklauso ir nuo imties dydžio n ir nuo nepriklausomų kintamųjų skaičiaus k. Be to, mažiems R^2 koreguotasis determinacijos koeficientas gali įgyti ir neigiamas reikšmes. Koreguotojo determinacijos koeficiento interpretacija lieka ta pati: kuo jis didesnis, tuo geriau Y reikšmes aprašo regresijos modelyje esančių nepriklausomų kintamųjų elgesys.

3. *Daugialypės koreliacijos koeficientas* R - tai tiesiog kvadratinė šaknis iš determinacijos koeficiento, t.y $R = \sqrt{R^2}$. Pastebėsime, kad skirtingai nuo Pirsono dviejų kintamųjų koreliacijos koeficiento (4.7), jis negali būti neigiamas, nes $0 \leq R \leq 1$. Šių koreliacijos koeficientų tarpusavio sąryšį paprastosios tiesinės regresijos atveju nusako 4.1.1 teorema. Daugialypės koreliacijos koeficientas parodo, kaip stipriai prognozuojamas kintamasis priklauso nuo visų nepriklausomų kintamųjų. Žinoma, kalbama apie tiesinę priklausomybę, aprašytą (4.14) modeliu. Tačiau didelis daugialypės koreliacijos koeficientas dar nereiškia, kad prognozei visi nepriklausomi kintamieji yra svarbūs ir naudingi.

5 Nekontroliuojamo mokymo uždaviniai

5.1 Asociacijos taisyklės

Asociacijos taisyklės yra panašios struktūros kaip ir klasifikavimo taisyklės. Tik šiuo atveju nėra akivaizdaus klasės kintamojo. Todėl tiek asociacijos taisyklės prielaida tiek jos išvada gali būti sudaryta iš daugelio atributų.

5.1.1 Pirkėjo krepšelio uždavinys

Charakteringas asociacijų paieškos pavyzdys yra vadinamasis pirkėjo krepšelio uždavinys: analizuojant prekybos centro pardavimų duomenis, stengiamasi nustatyti kokios prekės dažniausiai perkamos kartu. 5.1 lentelėje pateiktas tokio uždavinio duomenų pavyzdys. Kiekviena eilutė sudaryta iš įrašo (pirkinio) numerio ir elementų (prekių) sąrašo. 5.1

Nr.	Prekės
1	{duona, pienas}
2	{duona, degtukai, alus, sūris}
3	{pienas, degtukai, alus, sultys}
4	{duona, pienas, degtukai, alus}
5	{duona, pienas, degtukai, sultys}

5.1 lentelė. Penkių pirkėjų pirktos prekės

lentelės duomenys rodo, kad pirkėjai, perkantys degtukus, dažniausiai perka ir alų. Tokį pastebėjimą atspindinti asociacijos taisyklė yra

$$\{\text{degtukai}\} \longrightarrow \{\text{alus}\}.$$

Prekybos centro vadybininkams tokia informacija padeda racionaliau išdėstyti prekes, organizuoti reklamines akcijas ir t.t.

Dažnai pirkėjo krepšelio duomenys užrašomi binariniu pavidalu, kaip parodyta 5.2 lentelėje. Kiekviena parduotuvėje turima prekių rūšis vaizduojama atskiru binariniu kintamuoju,

Nr.	duona	pienas	degtukai	alus	sūris	sultys
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

5 .2 lentelė. Binariniai pirkėjų duomenys

įgyjančiu reikšmę 1, jei atitinkama prekė pateko į "pirkėjo krepšelį". Pastebėsime, kad šie kintamieji yra asimetriniai, nes nėra labai svarbu kokių prekių pirkėjas nepirko.

5.1.2 Asociacijos taisyklių apimtis ir tikslumas

Turėdami galvoje pirkėjo krepšelio pavidalo duomenis, tiksliau suformuluosime atributų asociacijų paieškos uždavinį. Prekių rūšis vadinsime elementais. Tegul aibę $I = \{i_1, i_2, \ldots, i_n\}$ sudaro visi galimi nagrinėjamų duomenų elementai. Visų I poaibių aibę žymėsime $\mathcal{P}(I)$. Tada turimoje N įrašų imtyje

$$T = \{t_1, t_2, \ldots, t_N\}$$

kiekvienas įrašas $t_i \in \mathcal{P}(I)$ reiškia tam tikrų elementų rinkinį. Įrašų, į kuriuos įeina elementų rinkinys X, skaičius $\sigma(X)$ vadinamas rinkinio X *dažniu* :

$$\sigma(X) = |\{t_i \mid X \subset t_i,\ t_i \in T\}| \ .$$

Pavyzdžiui, 5 .1 lentelėje pateiktoje imtyje rinkinio { alus, degtukai} dažnis lygus 3.

5.1.1 apibrėžimas. *Asociacijos taisykle* $X \longrightarrow Y$ *vadinsime implikaciją*

$$X \subset t \Longrightarrow Y \subset t \, .$$

Čia $X, Y \in \mathcal{P}(I),\ X \cap Y = \emptyset;\ t$ *- bet kuris duomenų aibės įrašas.*

Dažniausiai naudojami asociacijos taisyklių "gerumo matai" yra *apimtis* ir *tikslumas*. Jų apibrėžimas atitinka analogiškas klasifikavimo taisyklių charakteristikas.

5.1.2 apibrėžimas. *Pagal imties T duomenis sukonstruotos asociacijos taisyklės*

$$r: \quad X \longrightarrow Y$$

apimtis $a(r)$ ir tikslumas $\theta(r)$ yra lygūs

$$\begin{aligned} a(r) &= \frac{\sigma(X \cup Y)}{|T|}, \\ \theta(r) &= \frac{\sigma(X \cup Y)}{\sigma(X)}. \end{aligned}$$

Čia $|T| = N$ - įrašų skaičius imtyje T.

Pavyzdys. Pagal 5.1 lentelės duomenis sukonstruota asociacijos taisyklė

$$r: \quad \{\text{duona}, \text{degtukai}\} \longrightarrow \{\text{alus}\}.$$

Pirmiausiai randame taisyklės prielaidos ir jungtinio elementų rinkinio dažnius. Kadangi $\sigma(\{\text{duona}, \text{degtukai}\}) = 3$ ir $\sigma(\{\text{duona}, \text{degtukai}, \text{alus}\}) = 2$, tai

$$a(r) = \frac{2}{5}, \quad \theta(r) = \frac{2}{3}.$$

Kuo didesnis taisyklės $X \longrightarrow Y$ tikslumas, tuo didesnė tikimybė radus pirkėjo krepšelyje X, rasti ir Y. Kitaip sakant, tikslumas yra sąlyginės tikimybės $P(Y|X)$ įvertis. Tačiau gerai taisyklei nepakanka didelio tikslumo. Maža taisyklės $X \longrightarrow Y$ apimtis rodo, kad elementų rinkinys $X \cup Y$ retai pasitaiko pirkėjo krepšelyje. Todėl, net ir būdama tiksli, tokia taisyklė nekels didelio vadybininkų pasitikėjimo. Juk jei iš 10000 pirkėjų vienas nusipirko duonos ir televizorių, tai tikrai nereiškia, kad rytoj visi perkantys duoną, pirks ir televizorių! Todėl dažniausiai sutinkama tokia asociacijos taisyklių radimo uždavinio formuluotė.

Asociacijos taisyklių generavimo uždavinys. *Pagal turimą imtį reikia rasti asociacijos taisykles r, kurių apimtis $a(r) \geq a_{\min}$ ir tikslumas $\theta(r) \geq \theta_{\min}$, čia $a_{\min}$ ir $\theta_{\min}$ yra iš anksto pasirinktieji apimties ir tikslumo rėžiai.*

"Paprasčiausias" šio uždavinio sprendimo būdas - patikrinti visas galimas asociacijos taisykles. Tačiau jų skaičius auga eksponentiškai. Jei imtyje sutinkamų elementų yra $|I| = n$, tai iš viso galima sukonstruoti

$$R(n) = 3^n - 2^{n+1} + 1 \tag{5.1}$$

asociacijos taisyklių $X \longrightarrow Y$ su netuščiais elementų rinkiniais X ir Y. Net labai mažai parduotuvei, prekiaujančiai tik 20 pavadinimų prekėmis, šis skaičius yra $R(20) = 3484687250$. Akivaizdu, kad reikia ieškoti efektyvesnių metodų.

Pirmiausiai galime atskirti taisyklės apimties ir tikslumo skaičiavimus. Pastebėsime, kad taisyklės $X \longrightarrow Y$ apimtis priklauso tik nuo rinkinio $X \cup Y$ dažnio. Pavyzdžiui, taisyklės

$$\{1,2\} \longrightarrow \{3\}, \quad \{1,3\} \longrightarrow \{2\}, \quad \{2,3\} \longrightarrow \{1\},$$
$$\{3\} \longrightarrow \{1,2\}, \quad \{2\} \longrightarrow \{1,3\}, \quad \{1\} \longrightarrow \{2,3\}$$

yra vienodos apimties, priklausančios nuo rinkinio $\{1, 2, 3\}$ dažnio. Jei šis rinkinys imtyje retai pasitaiko, iš karto galime atmesti visas 6 taisykles, net neskaičiuodami jų tikslumo. Todėl dauguma algoritmų asociacijos taisykles generuoja dviem etapais:

1. **Dažnų rinkinių radimas**. Randami visi elementų rinkiniai X, kurių santykinis dažnis imtyje T yra ne mažesnis už pasirinktą taisyklės apimties rėžį $a_{\min}$, t.y. $\sigma(X) \geq N \cdot a_{\min}$.

2. **Taisyklių generavimas**. Iš rastųjų dažnų rinkinių konstruojamos asociacijos taisyklės, kurių tikslumas ne mažesnis už $\theta_{\min}$. Tokias taisykles vadinsime *tvirtomis*.

Daugiau skaičiavimų paprastai reikalauja pirmasis etapas - dažnų rinkinių paieška.

5.1.3 Dažnų elementų rinkinių paieška

Jei duomenų elementų aibės I dydis yra n, tai bet kuris iš $2^n - 1$ netuščių jos poaibių yra potencialus dažnas rinkinys. Visų jų dažnių radimui reikėtų $O\left(N2^n \max_i |t_i|\right)$ palyginimo operacijų. Todėl labai svarbu kiek įmanoma sumažinti potencialių dažnų rinkinių skaičių, neskaičiuojant jų dažnio. Gana efektyviai tai galima padaryti remiantis vadinamuoju *Apriori* principu. Jis labai paprastas.

***Apriori* principas**. *Bet kuris dažno elementų rinkinio poaibis taip pat yra dažnas.*

Jis išplaukia iš 5.1.1 teoremoje suformuluotos akivaizdžios dažnio *antimonotoniškumo* savybės.

5.1.1 teorema. *Jei $X \subset Y \subset I$, tai $\sigma(X) \geq \sigma(Y)$.*

Tiesioginė *Apriori* principo išvada, kuria ir remiasi tuo pačiu vardu pavadintas algoritmas, yra savotiška "rėčio" taisyklė:

jei elementų rinkinys X nėra dažnas, tai ir bet kuris jį apimantis rinkinys taip pat nėra dažnas.

Pirmiausiai nagrinėjami 1 elemento rinkiniai. Išmetami visi, turintys mažus dažnius, o po to ir visi daugiau elementų turintys rinkiniai, apimantys išmestuosius 1 elemento rinkinius. Toliau nagrinėjami likusieji 2 elementų rinkiniai ir t.t.

Pavyzdys. Prisiminkime 5.1 lentelės duomenis. Tegul apimties rėžis yra $a_{min} = 0,6$. Tai reiškia, kad dažnais bus pripažįstami rinkiniai, sutinkami imtyje ne mažiau kaip $5 \cdot 0,6 = 3$ kartus. Visus tokius rinkinius rasime atlikę tris iteracijas, kiekvieną kartą išmesdami "retus" rinkinius (5.1 pav. jie nuspalvinti pilkai). Pagal *Apriori* principą potencialūs dažni rinkiniai

Elementas	Dažnis
alus	3
duona	4
sultys	2
degtukai	4
sūris	1
pienas	4

$\Rightarrow$

2 elem. rinkinys	Dažnis
{alus,duona}	2
{alus,degtukai}	3
{alus,pienas}	2
{duona,degtukai}	3
{duona,pienas}	3
{degtukai,pienas}	3

$\Downarrow$

3 elem. rinkinys	Dažnis
{duona,degtukai,pienas}	2

5.1 pav. *Apriori* principo taikymas

yra tik tie, kurių visi poaibiai taip pat dažni. Todėl tokių kandidatų, kuriuos reikia patikrinti skaičiuojant jų dažnį, lieka tik

$$\binom{6}{1} + \binom{4}{2} + 1 = 6 + 6 + 1 = 13.$$

To tarpu bendras rinkinių, turinčių ne daugiau kaip 4 elementus, skaičius yra

$$\binom{6}{1} + \binom{6}{2} + \binom{6}{3} + \binom{6}{4} = 6 + 15 + 20 + 15 = 56.$$

Vadinasi šiuo atveju *Apriori* algoritmas leido sumažinti kandidatų skaičių daugiau kaip 4 kartus.

Dažnų rinkinių generavimo algoritmo schemą nusako 5.2 paveiksle pavaizduotas pseudokodas.

1. $k := 1$
2. $F_k := \{i \mid i \in I,\ \sigma(\{i\}) \geq N \cdot a_{\min}\}$ - randami dažni 1 elemento rinkiniai
3. **repeat**
4. $k = k + 1$
5. $C_k = \texttt{genApriori}(F_{k-1})$ - potencialūs dažni rinkiniai iš k elementų
6. **for** $\forall\ t \in T$ **do**
7. $C_k(t) = \{c \mid c \in C_k,\ c \subset t\}$ - įrašui t priklausantys C_k rinkiniai
8. **for** $\forall\ c \in C_k(t)$ **do**
9. $\sigma(c) = \sigma(c) + 1$ - dažnio prieaugis
10. **end for**
11. **end for**
12. $F_k = \{c \mid c \in C_k,\ \sigma(c) \geq N \cdot a_{\min}\}$ - dažni rinkiniai iš k elementų
13. **until** $F_k = \emptyset$
14. Rezultatas $= \cup F_k$

5.2 pav. *Apriori* algoritmas dažnų rinkinių generavimui

Svarbiausią darbą atlieka funkcija $\texttt{genApriori}(F_{k-1})$. Ji pagal turimą $k-1$ elemento dažnų rinkinių aibę F_{k-1}, remiantis *Apriori* principu, generuoja visus galimai dažnus k elementų rinkinius. Toliau (6 - 11 eilutės) skaičiuojami nustatytų kandidatų į dažnus rinkinius dažniai. Priklausomai nuo konkrečios realizacijos, šie algoritmo žingsniai gali būti įvairiai optimizuojami. Tačiau bet kuriuo atveju galima paminėti tokius algoritmo efektyvumą įtakojančius faktorius.

1. **Apimties rėžis**. Didinant apimties rėžį, mažėja kandidatų į dažnus rinkinius ir dažnų rinkinių skaičius. Tuo pačiu greitėja ir pats algoritmas.

2. **Elementų skaičius (dimensija)**. Didesniam elementų skaičiui reikia daugiau sąnaudų jų dažnių skaičiavimui. Be to, didesnės dimensijos imtyse ir dažnų rinkinių gali būti daugiau.

3. **Imties dydis**. Skaičiuojant dažnius, yra tikrinami visi imties įrašai. Todėl akivaizdu, kad didinant įrašų skaičių, laiko sąnaudos didėja.

4. **Vidutinis įrašo dydis**. Įrašams ilgėjant, algoritmas lėtėja. Tai pasireiškia dvejopai. Visų pirma - ilgesni įrašai paprastai turi daugiau dažnų rinkinių. Antra - rinkinių paieška ilgesniuose įrašuose užima daugiau laiko.

5.1.4 Asociacijos taisyklių generavimas

Aptarsime kaip generuojamos asociacijos taisyklės r, sudarytos iš dažno k elementų rinkinio Y poaibių ir tenkinančios tikslumo sąlygą $\theta(r) \geq \theta_{\min}$. Tokias taisykles galima sudaryti skaidant Y į du nesikertančius poaibius $X \subset Y$ ir $Y \setminus X$:

$$r: \quad X \longrightarrow Y \setminus X\,. \tag{5.2}$$

Jos apimtis ir tikslumas yra

$$a(r) = \frac{\sigma(Y)}{N}\,, \quad \theta(r) = \frac{\sigma(Y)}{\sigma(X)}\,.$$

Kadangi visi dažno rinkinio Y poaibiai taip pat yra dažni ir jų dažniai jau buvo apskaičiuoti pirmajame algoritmo etape, tai tikslumas randamas nesunkiai. Be to, $a(r) \geq a_{\min}$. Viena problema: tokių rinkinio Y skaidinių, o tuo pačiu ir galimų taisyklių, labai daug. Jei $X \neq \emptyset$ ir $X \neq Y$, tai (5.2) pavidalo taisyklių galima sudaryti $2^k - 2$. Šį kandidatų į tikslias taisykles skaičių padės sumažinti 5.1.2 teorema. Pasirodo kai kuriais atvejais galima iš anksto palyginti dviejų asociacijos taisyklių tikslumus.

5.1.2 teorema. *Jei $A \subset B \subset C \subset I$, tai asociacijos taisyklė*

$$r_1: \quad A \longrightarrow C \setminus A$$

yra ne tikslesnė už taisyklę

$$r_2: \quad B \longrightarrow C \setminus B\,,$$

t.y., teisinga nelygybė $\theta(r_1) \leq \theta(r_2)$.

Įrodymas. Pagal 5.1.2 apibrėžimą asociacijos taisyklių r_1 ir r_2 tikslumai yra lygūs

$$\theta(r_1) = \frac{\sigma(A \cup (C \setminus A))}{\sigma(A)} = \frac{\sigma(C)}{\sigma(A)}, \quad \theta(r_2) = \frac{\sigma(B \cup (C \setminus B))}{\sigma(B)} = \frac{\sigma(C)}{\sigma(B)}.$$

Dėl dažnio antimonotoniškumo $\sigma(A) \geq \sigma(B)$. Todėl $\theta(r_1) \leq \theta(r_2)$.

□

Pastarosios teoremos teiginys palengvina tikslių taisyklių paiešką.

Apriori algoritmas paiešką pradeda nuo taisyklių, kurių išvados yra 1 elemento rinkiniai. Paliekamos tos, kurių tikslumas ne mažesnis už $\theta_{\min}$, o likusios pašalinamos. Tada pereinama prie taisyklių, kurių išvados yra 2 elementų rinkiniai ir t.t.

Pavyzdys. Tarkime $Y = \{abcd\}$ yra dažnas rinkinys ir

$$\theta(\{acd\} \longrightarrow \{b\}) \geq \theta_{\min}, \quad \theta(\{abd\} \longrightarrow \{c\}) \geq \theta_{\min},$$
$$\theta(\{bcd\} \longrightarrow \{a\}) < \theta_{\min}, \quad \theta(\{abc\} \longrightarrow \{d\}) < \theta_{\min}.$$

Galima sudaryti $\binom{4}{2} = 6$ taisykles, kurių išvadoje bus 2 elementų rinkinys. Iš 5.1.2 teoremos išplaukia, kad penkios iš šių taisyklių

$$\{cd\} \longrightarrow \{ab\}, \quad \{bd\} \longrightarrow \{ac\}, \quad \{bc\} \longrightarrow \{ad\},$$
$$\{ac\} \longrightarrow \{bd\}, \quad \{ab\} \longrightarrow \{cd\}$$

nėra tikslios - jų tikslumas mažesnis už $\theta_{\min}$. Tad lieka apskaičiuoti tik taisyklės

$$\{ad\} \longrightarrow \{bc\}$$

tikslumą.

Apriori algoritmo antrojo etapo pseudokodas, aptartuoju būdu generuojantis tvirtas asociacijos taisykles, pateikiamas 5.3 paveiksle.

Čia naudojami pirmajame etape (žr. 5.2 pav.) rastieji dažni elementų rinkiniai ir jų dažniai. Taisyklės generuojamos iteracijomis pagal elementų skaičių taisyklės išvadoje. Fiksavus dažną k elementų rinkinį f, palaipsniui randamos tikslios taisyklės, kurių išvados

1. Tegul F_k - dažnų k elementų rinkinių aibė
2. **for** $\forall\ f \in F_k\,,\ k \geq 2$ **do**
3. $H_1 := \{\{i\} \mid i \in f\}$ - galimos išvados iš 1 elemento rinkinių
4. $m := 0$
5. **repeat**
6. $m = m + 1$
7. **if** $m > 1$ **then**
8. $H_m = \texttt{genTaisApriori}(f, H_{m-1})$ - galimos išvados iš m elementų
9. **end if**
10. **for** $\forall\ h \in H_m$ **do**
11. $\theta = \sigma(f)/\sigma(f \setminus h)$
12. **if** $\theta \geq \theta_{\min}$ **then**
13. generuojama taisyklė $(f \setminus h) \longrightarrow h$
14. **else**
15. rinkinys h šalinamas iš H_m
16. **end if**
17. **end for**
18. **until** $((H_m = \emptyset) \vee (m \geq k - 1))$
19. **end for**

5.3 pav. *Apriori* algoritmas asociacijos taisyklių generavimui

turi nuo 1 iki $k - 1$ elemento. Kiekvieną kartą netikslios taisyklės išvada iš galimų išvadų aibės eliminuojama (12 -16 eil.). Pradedant $m = 2$, galimų m elementų išvadų aibės H_m formavimui naudojama ankstesnėje iteracijoje sudaryta tikslių taisyklių išvadų aibė H_{m-1}. Tai atlieka funkcija $\texttt{genTaisApriori}(f, H_{m-1})$ (8 eil.). Ji susiaurina galimų išvadų aibę, kai kurias iš jų iš anksto atmesdama, remiantis 5.1.2 teorema.

5.1.5 Asociacijos taisyklių vertinimas

Dauguma asociacijos taisyklių generavimo metodų remiasi taisyklės apimties ir tikslumo įverčiais. Tačiau ar visada taisyklės, kurių apimtis ir tikslumas viršija nustatytus rėžius, yra geros ir atspindi objektyvius dėsningumus ? Panagrinėkime tokį pavyzdį.

5.1.1 pavyzdys. Didelėje elektronikos parduotuvėje iš 10000 pirkėjų 6000 pirko kompiuterinius žaidimus, 7500 pirko DVD filmus, o 4000 pirko ir žaidimus ir filmus. Pasirinkus apimties ir tikslumo rėžius $a_{\min} = 0,3$ ir $\theta_{\min} = 0,6$, buvo suformuluota asociacijos taisyklė:

$$r_1 : \quad X = \{\text{kompiuteriniai žaidimai}\} \longrightarrow Y = \{\text{DVD filmai}\} \tag{5.3}$$

Ši taisyklė yra tvirta, nes

$$a(r_1) = \frac{4000}{10000} = 0,4 > 0,3 \quad \text{ir} \quad \theta(r_1) = \frac{4000}{6000} \approx 0,67 > 0,6\,.$$

Tačiau 5.3 taisyklė nėra logiška. Iš tikrųjų, 75% pirkėjų perka DVD filmus. Bet tik 67% kompiuterinių žaidimų pirkėjų perka ir DVD filmus. Vadinasi prekės X buvimas krepšelyje net sumažina Y tikimybę !

Pavyzdys rodo, kad reikalingi ir alternatyvūs taisyklės kokybės matai. Apibrėšime keletą iš jų.

Tegul asociacijos taisyklė

$$r : \quad X \longrightarrow Y \tag{5.4}$$

gauta pagal imties $T = \{t_1, t_2, \ldots, t_N\}$ duomenis. Jos *įtakos faktorius* lygus

$$ITF(r) = ITF(X,Y) = \frac{N \cdot \theta(r)}{\sigma(Y)}\,. \tag{5.5}$$

Nesunku pastebėti, kad įtakos faktorius $ITF(X,Y)$ iš tikrųjų yra tikimybių santykio

$$\frac{P(Y \mid X)}{P(Y)}$$

įvertis. Todėl jį galima interpretuoti taip:

$$ITF(X,Y) \begin{cases} = 1, \text{jei } X \text{ ir } Y \text{ nepriklausomi;} \\ > 1, \text{jei tarp } X \text{ ir } Y \text{ yra tiesioginė koreliacija;} \\ < 1, \text{jei tarp } X \text{ ir } Y \text{ yra atvirkštinė koreliacija.} \end{cases}$$

Elektronikos parduotuvės pavyzdyje (5.3) taisyklės įtakos faktorius yra

$$ITF(r_1) = \frac{10000 \cdot 4000}{6000 \cdot 7500} \approx 0,89 < 1\,.$$

Tai rodo, kad tarp kompiuterinių žaidimų ir DVD filmų pirkimų yra atvirkštinė koreliacija: perkant vieną, mažėja tikimybė pirkti kitą.

Dažnai (5.4) pavidalo taisyklių vertinimui naudojami 2.3.4 skyrelyje jau nagrinėti dviejų binarinių vektorių panašumo koeficientai. Kai kurie iš jų buvo pateikti 2.10 lentelėje.

Apibrėžkime du, su asociacijos taisykle (5.4) susijusius, vektorius

$$\mathbf{x} = (x_1, x_2, \ldots, x_N)\,, \quad \mathbf{y} = (y_1, y_2, \ldots, y_N)\,,$$

kurių i - tosios koordinatės yra rinkinių X ir Y patekimo į įrašą t_i indikatoriai. Pavyzdžiui, $x_i = 1$, jei $X \subset t_i$, ir $x_i = 0$ - priešingu atveju.

Gerą taisyklę atitinkantys vektoriai $\mathbf{x}$ ir $\mathbf{y}$ yra "panašūs", t.y. jų panašumo koeficientas turėtų būti kuo didesnis. Sąryšį tarp $\mathbf{x}$ ir $\mathbf{y}$ nusako dažnių lentelė

$x_i \backslash y_i$	0	1	
0	k_{00}	k_{01}	k_{0+}
1	k_{10}	k_{11}	k_{1+}
	k_{+0}	k_{+1}	N

Pastebėsime, kad

$$k_{1+} = \sigma(X)\,, \quad k_{+1} = \sigma(Y)\,, \quad k_{11} = \sigma(X \cup Y)\,.$$

Tad ir kiti dažnių lentelės koeficientai jau nesunkiai nusakomi rinkinių X ir Y dažniais. Be to,

$$\mathbf{x} \cdot \mathbf{y} = k_{11}\,, \quad \|\mathbf{x}\|^2 = k_{1+}\,, \quad \|\mathbf{y}\|^2 = k_{+1}\,.$$

Todėl 5.3 lentelėje pateikiami asociacijos taisyklės (5.4) įverčiai, išreikšti vektorių $\mathbf{x}$ ir $\mathbf{y}$ dažnių lentelės elementais pagal (5.5) ir 2.10 lentelės formules.

Naudojami ir kitokie asociacijos taisyklių kokybės matai ([7], 6.7 sk.). Dažnai galutinis taisyklės vertinimas grindžiamas ir subjektyviais kriterijais, būdingais tai žmonių veiklos

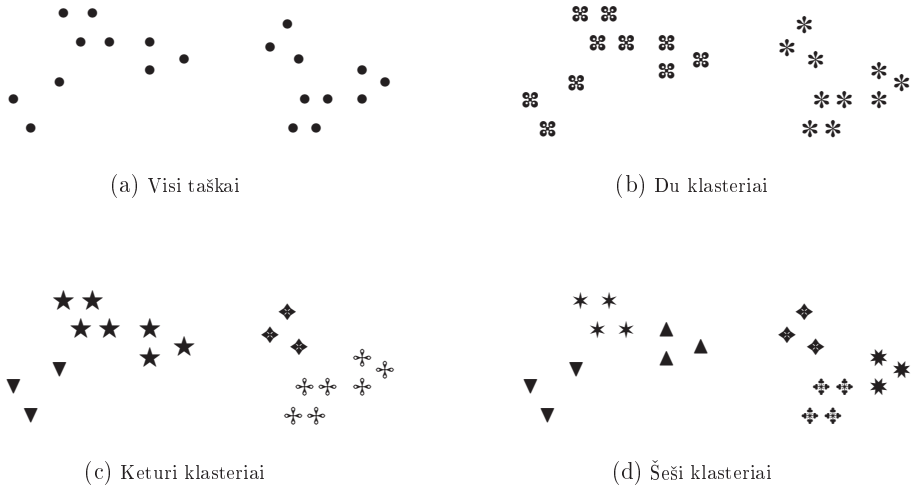
Pavadinimas	Apibrėžimas
Įtakos faktorius	$\frac{N \cdot k_{11}}{k_{1+}k_{+1}}$
Žakardo	$\frac{k_{11}}{k_{1+} + k_{+1} - k_{11}}$
Kosinusas	$\frac{k_{11}}{\sqrt{k_{1+}k_{+1}}}$
Koreliacijos	$\frac{k_{11}k_{00} - k_{10}k_{01}}{\sqrt{k_{1+}k_{+1}k_{0+}k_{+0}}}$

5 .3 lentelė. Asociacijos taisyklių kokybės matai

sričiai, kurios duomenys yra nagrinėjami. Trumpai sakant, kaip visada galutinis žodis priklauso vartotojui. Jis sprendžia, kurios taisyklės yra prasmingos ir atspindi jam aktualius, turimuose duomenyse glūdinčius sąryšius.

5.2 Klasterinė analizė

Taikydami klasterinę analizę, remiantis turimais duomenimis, nustatome objektų panašumą ir suskirstome juos į *klasterius*. Tikslaus klasterio apibrėžimo nėra, tačiau reikia skirti sąvokas - "grupė" ir "klasteris". Klasteris - panašių objektų grupė. Bendru atveju pasakyti ką reiškia panašūs objektai - neįmanoma. Pavyzdžiui, Jonas ir Petras yra aistringi krepšinio sirgaliai, bet priklauso skirtingoms politinėms partijoms. Ar jie panašūs ?!
Klasterinėje analizėje objektų panašumas nusakomas skaitiniais panašumo matais. Jų pasirinkimas, aišku, priklauso nuo turimų duomenų prigimties ir sprendžiamo uždavinio. Kai kurie objektų artumo matai buvo aptarti 2.3.4 skyrelyje.
Pastebėsime, kad skirstydami objektus į klasterius, dažniausiai net nežinome, kiek klasterių turimoje duomenų aibėje realiai egzistuoja (ir ar išvis egzistuoja). Uždavinio komplikuotumą iliustruoja 5 .4 paveiksle pavaizduota situacija. Kaip matome, 20 plokštunos taškų,



(a) Visi taškai (b) Du klasteriai

(c) Keturi klasteriai (d) Šeši klasteriai

5.4 pav. Taškų aibės klasterizavimo būdai

įvairiai skaidydami, galime suskirstyti į du, keturis ar šešis klasterius (priklausomybę klasteriams vaizduoja skirtingos taškų žymės).

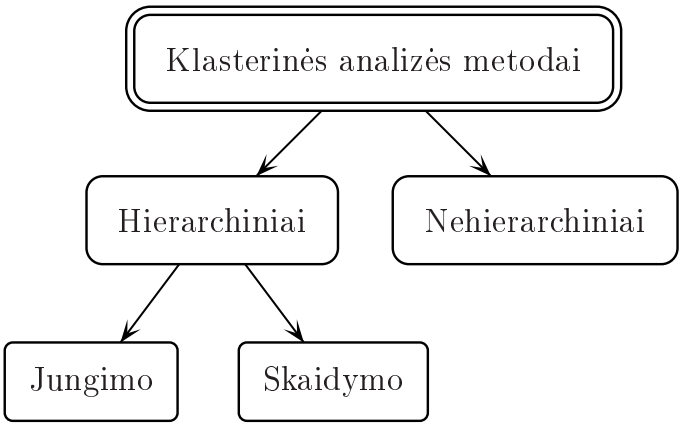
Klasterinėje analizėje imtis neturi klasės kintamojo, pagal kurio reikšmes būtų galima patikrinti ("kontroliuoti") ar žinomas mokymo imties įrašas pateko į "teisingą" klasterį. Todėl, skirtingai nuo klasifikavimo, klasterizavimo uždaviniai priskiriami nekontroliuojamo mokymo uždavinių kategorijai.

5.2.1 Klasterinės analizės metodų klasifikacija

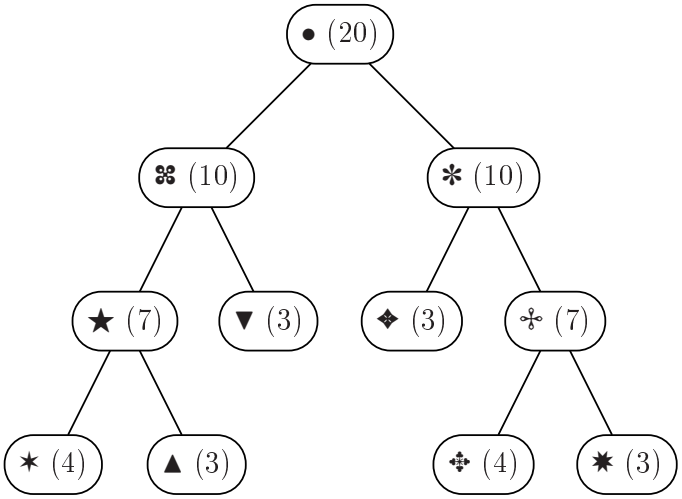
Yra daug klasterių sudarymo metodų. Jie skiriami pagal tai, kaip parenkami panašumo matai, atstumo tarp klasterių nustatymo kriterijai bei kokia skirstymo į klasterius strategija. Gana dažnas yra 5.5 paveiksle parodytoje schemoje vaizduojamas skirstymas pagal galimą klasterių tarpusavio išsidėstymą.

Skiriamos dvi pagrindinės klasterinės analizės metodų klasės - hierarchiniai ir nehierarchiniai.

Hierarchiniai metodai. Jų rezultatai nusako klasterių tarpusavio hierarchiją, t.y. visi objektai laikomi vienu dideliu klasteriu, kurį sudaro mažesni klasteriai, šiuos savo ruožtu dar mažesni ir t.t. Kitaip sakant, gautieji klasteriai sudaro medį, kurio šaknyje yra visų objektų klasteris, o lapuose - patys mažiausi klasteriai. Pavyzdžiui, 5.4 paveiksle pavaizduoti plokštumos taškų klasteriai sudaro medį (žr. 5.6 pav.).



5 .5 pav. Klasterizavimo metodų klasifikacija



5 .6 pav. Klasterių medis

Čia kiekvienas klasteris (medžio viršūnė) vaizduojamas jį sudarančių taškų žyme ir taškų skaičiumi.

Priklausomai nuo to kaip "auginamas" klasterių medis, hierarchiniai metodai skirstomi į jungimo ir skaidymo metodus. Jungimo metodai konstruoja medį nuo lapų, t.y. smulkius klasterius jungia vis į stambesnius, kol galų gale lieka vienas. Skaidymo metodai "augina" medį nuo šaknies - vienintelį klasterį nuosekliai skaido į dalis.

Nehierarchiniai metodai. Taikydami hierarchinius metodus, nustatome bendrą visų klasterių tarpusavio priklausomybių struktūrą ir tik po to sprendžiame, koks klasterių

skaičius tinkamas. Nehierarchiniai metodai tiesiog skaido turimą duomenų aibę į K nesikertančių poaibių (klasterių) taip, kad artimi objektai patektų į vieną poaibį. Pavyzdžiui, bet kuris iš 5.4 (a) - (d) paveikslų vaizduoja nehierarchinės klasterizacijos rezultatą, kai $K = 1, 2, 4, 6$ atitinkamai. Tokie metodai paprastai taikomi tada, kai iš anksto žinomas (pasirenkamas) klasterių skaičius K.

Detaliau panagrinėsime vieną hierarchinių metodų klasę - jungimo metodus bei vieną dažniausiai naudojamų nehierarchinių metodų - K-vidurkių metodą.

5.2.2 Jungimo metodai

Tegul $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ - duomenų (objektų) aibė, $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$ - atstumas tarp objektų $\mathbf{x}_i$ ir $\mathbf{x}_j$. Visi atstumai sudaro $N \times N$ simetrinę *atstumų matricą* $(d_{ij})_{N\times N}$. Hierarchiniai jungimo metodai, pradedant pavieniais objektais, nuosekliai jungia du artimiausius klasterius, kol lieka tik vienas klasteris. Tokio algoritmo schema pavaizduota 5.7 paveiksle.

1. Turime N klasterių po 1 objektą
2. Apskaičiuojame atstumų matricą $(d_{ij})_{N\times N}$
3. **repeat**
4. Nustatome du artimiausius klasterius U ir V
5. U ir V sujungiami į vieną klasterį $U \cup V$
6. Transformuojama atstumų matrica
7. **until** Lieka tik vienas klasteris

5.7 pav. Jungimo metodo algoritmas

Atstumų matrica transformuojama (6 eilutė) taip:

1) išbraukiame eilutes ir stulpelius, atitinkančius klasterius U ir V,

2) pridedame eilutę ir stulpelį su atstumais tarp $U \cup V$ ir likusiųjų klasterių.

Esminė algoritmo operacija - atstumo tarp dviejų klasterių skaičiavimas (4 ir 6 eilutės). Priklausomai nuo algoritmo modifikacijos, sutinkami įvairūs šio atstumo apibrėžimai. Daž-

niausiai naudojami atstumai $d(U,V)$ tarp klasterių $U \subset X$ ir $V \subset X$ pateikti 5.4 lentelėje. Klasterio centro sąvoka priklauso nuo objektų atstumo apibrėžimo. Pastebėsime, kad $\mathbf{x}_U$ ne

Atstumas	$d(U,V)$ formulė
Artimiausio kaimyno	$d(U,V) = \min\limits_{\mathbf{x}\in U,\, \mathbf{x}'\in V} d(\mathbf{x},\mathbf{x}')$
Tolimiausio kaimyno	$d(U,V) = \max\limits_{\mathbf{x}\in U,\, \mathbf{x}'\in V} d(\mathbf{x},\mathbf{x}')$
Vidutinis	$d(U,V) = \dfrac{1}{\lvert U\rvert \cdot \lvert V\rvert} \sum\limits_{\mathbf{x}\in U} \sum\limits_{\mathbf{x}'\in V} d(\mathbf{x},\mathbf{x}')$
Centrų	$d(U,V) = d(\mathbf{x}_U, \mathbf{x}_V)$, $\mathbf{x}_U, \mathbf{x}_V$ - klasterių U ir V centrai

5.4 lentelė. Klasterių U ir V artumo matai

visada yra klasterio U objektas. Pavyzdžiui, kai $X \subset \mathbb{R}^p$, o atstumas $d(\mathbf{x}_i,\mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|$ - Euklido metrika, centru dažniausiai laikomas "vidutinis" klasterio U objektas

$$\mathbf{x}_U = \frac{1}{|U|}\sum_{\mathbf{x}\in U} \mathbf{x}\,, \quad |U| \;-\; \text{klasterio } U \text{ objektų skaičius.} \tag{5.6}$$

Šiuo atveju, be 5.4 lentelėje išvardintųjų, gali būti skaičiuojamas ir vadinamasis Ward'o atstumas

$$d_W(U,V) = \frac{|U|\cdot|V|}{|U|+|V|}\|\mathbf{x}_U - \mathbf{x}_V\|^2\,.$$

Jam būdinga tai, kad sujungus du artimiausius klasterius, visada yra minimizuojama objektų atstumų nuo jų klasterių centrų kvadratų suma. Tačiau tai nereiškia, kad pastarasis atstumas - pats geriausias. Neturint išankstinės informacijos apie tiriamus duomenis, sudėtinga pasakyti kuris metodas geriausias. Patartina objektų klasterizavimui taikyti ne vieną, o kelis metodus su skirtingais klasterių artumo matais. Plačiau apie vieno ar kito metodo pasirinkimo kriterijus galima paskaityti, pavyzdžiui, [7] knygos 8.3 sk. ir [1] knygos II d. 8.4 sk.

5.2.3 *K* - vidurkių metodas

Vienas iš hierarchinių klasterinės analizės metodų trūkumų - skaičiavimams naudojamos atstumų matricos dydžio priklausomybė nuo objektų skaičiaus N. Net ir atsižvelgus į tai, kad ji simetrinė ir pagrindinėje įstrižainėje turi nulius, tenka skaičiuoti $(N^2 - N)/2$ jos elementus. Todėl didesnės objektų aibės dažnai klasterizuojamos nehierarchiniais metodais. Vienas iš tokių ir yra K-vidurkių metodas. Jis plačiai taikomas ir kartu gana paprastas. Trumpas klasterizacijos algoritmo pseudokodas pateikiamas 5.8 paveiksle. Konkreti algo-

1. Pasirekame K pradinių klasterių centrų
2. **repeat**
3. Objektai suskirstomi į K klasterių, kiekvieną objektą priskiriant artimiausiam centrui
4. Perskaičiuojami klasterių centrai
5. **until** Klasterių centrai nebekinta

5.8 pav. K-vidurkių metodo algoritmas

ritmo realizacija priklauso nuo turimų duomenų tipo ir naudojamo objektų atstumo mato. Kaip jau buvo minėta, nuo to priklauso ir klasterio centro sąvoka.

Aptarsime algoritmo variantą, kai klasterizuojami objektai yra Euklido erdvės vektoriai, t.y. $X \subset \mathbb{R}^p$, $d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|$, o klasterio $U \subset X$ centras $\mathbf{x}_U$ yra U priklausančių vektorių vidurkis, apibrėžiamas (5.6) formule. Toks centro pasirinkimas turi matematinį pagrindą.

Tegul objektų aibė X skaidoma į K nesikertančių klasterių

$$X = C_1 \cup C_2 \cup \cdots \cup C_K \,.$$

Natūralu manyti, kad esant gerai klasterizacijai, objektai turi būti kuo arčiau klasterių centrų. Vadinasi tikslo funkcija, kurią reikia minimizuoti, parenkant klasterių centrus, yra

$$SSE = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{x}_{C_i}\|^2 \,.$$

Nesunku įrodyti, kad SSE įgyja mažiausią reikšmę, kai $\mathbf{x}_{C_i}$ yra lygus klasterio C_i vektorių vidurkiui, apskaičiuotam pagal (5 .6) formulę.

Vienas iš K-vidurkių metodo trūkumų - klasterių skaičių reikia nustatyti iš anksto. Tuo pačiu tarsi primetama tam tikra duomenų struktūra, nebūtinai sutampanti su objektyviai egzistuojančia. Su tuo susijusi ir pradinių K centrų parinkimo problema (1 algoritmo eilutė). Pavyzdžiui, būtina atsižvelgti į galimas išskirtis duomenų aibėje, stengtis kad pradiniai centrai nepriklausytų vienam klasteriui. Naudojamos įvairios pradinių centrų parinkimo strategijos: nuo atsitiktinio parinkimo iki preliminarios hierarchinės klasterizacijos į K klasterių (žr. [7], 8.2 sk.).

Literatūra

[1] V.Čekanavičius, G.Murauskas. *Statistika ir jos taikymai, I,II*, TEV, Vilnius, 2000-2002.

[2] D.Hand, H.Mannila and P.Smyth. *Principles of Data Mining*, The MIT Press, 2001.

[3] T.Hastie, R.Tibshirani, J.Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, 2001, 2009.

[4] Mehmed Kantardzic. *Data Mining: Concepts, Models, Methods, and Algorithms*, John Wiley & Sons, 2003.

[5] R.Lapinskas. *Įvadas į statistiką su* **R**, Vilniaus universitetas, paskaitų konspektas - http://www.mif.vu.lt/katedros/eka/medziaga/Lapinskas.pdf

[6] David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, 2003.

[7] Pang-Ning Tan, Michael Steinbach, Vipin Kumar. *Introduction to Data Mining*, Addison Wesley, 2005.

[8] V.Stakėnas. *Informacijos kodavimas*, Vilnius: VU, 1996.

[9] V.Stakėnas. *Kodai ir šifrai*, Vilnius: TEV, 2007.

[10] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[11] Ian H. Witten , Eibe Frank, Mark A. Hall. *Data Mining: Practical machine learning tools and techniques*, Third Edition, Morgan Kaufmann, San Francisco, 2011.