# Klasifikavimo taisyklės

# Klasifikavimo taisyklių sandara



Klasifikatorius $R$ aprašomas viena ar keliomis *"jei ... tai... "* pavidalo taisyklėmis $r_1, r_2, \ldots, r_l$

$$R = (r_1 \vee r_2 \vee \cdots \vee r_l)\,.$$

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.

# Klasifikatoriaus pavyzdys (1)



$$
\begin{array}{ll}
r_1: & (Gyvavedis = ne) \wedge (Skraido = taip) \longrightarrow \textit{paukštis} \\
r_2: & (Gyvavedis = ne) \wedge (Gyvena\ vandenyje = taip) \longrightarrow \textit{žuvis} \\
r_3: & (Gyvavedis = taip) \wedge (Kraujo\ tipas = \textit{šiltas}) \longrightarrow \textit{žinduolis} \\
r_4: & (Gyvavedis = ne) \wedge (Skraido = ne) \longrightarrow \textit{roplys} \\
r_5: & (Gyvena\ vandenyje = kartais) \longrightarrow \textit{varliagyvis}
\end{array}
$$

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

| Pavad. | Kraujo tipas | Odos danga | Gyva-vedis | Gyvena vand. | 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 $r_1$ apima varną, bet "nemato" lokio.

# Klasifikavimo taisyklių apimtis ir tikslumas



**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 $a(r) \cdot \theta(r)$.

# Duomenys apie gyvūnus

| Pavadinimas | Kraujo tipas ($X_1$) | Odos danga ($X_2$) | Gyva-vedis ($X_3$) | 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 |

# Pavyzdys (gyvūnų klasifikavimo tikslumas)



Imties dydis $|E| = 15$. Taisyklės

$$r_3 : (Gyvavedis = taip) \wedge (Kraujo\,tipas = \textit{šiltas}) \longrightarrow \textit{žinduolis}$$

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

$$r_5 : (Gyvena\,vandenyje = kartais) \longrightarrow \textit{varliagyvis}$$

apimtis ir tikslumas yra

$$a(r_5) = \frac{4}{15}\,, \quad \theta(r_5) = \frac{2}{4} = \frac{1}{2}\,.$$

# 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.

**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.*

**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.*

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 $R$.

## Pavyzdys (poromis nesutaikomos taisyklės)

$$R_1:\quad \begin{array}{ll} r_1: & (Gyvavedis = ne) \wedge (Skraido = taip) \longrightarrow \textit{paukštis} \\ r_2: & (Gyvavedis = ne) \wedge (Gyvena\ vandenyje = taip) \longrightarrow \textit{žuvis} \\ r_3: & (Gyvavedis = taip) \wedge (Kraujo\ tipas = \textit{šiltas}) \longrightarrow \textit{žinduolis} \\ r_4: & (Gyvavedis = ne) \wedge (Skraido = ne) \longrightarrow \textit{roplys} \\ r_5: & (Gyvena\ vandenyje = kartais) \longrightarrow \textit{varliagyvis} \end{array}$$

| Pavadinimas | Kraujo tipas | Odos danga | Gyva-vedis | 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 |

$$R_2:\quad \begin{array}{ll} r_1: & (Kraujo\ tipas = \textit{šaltas}) \longrightarrow \textit{ne žinduolis} \\ r_2: & (Kraujo\ tipas = \textit{šiltas}) \wedge (Gyvavedis = taip) \longrightarrow \textit{žinduolis} \\ r_3: & (Kraujo\ tipas = \textit{šiltas}) \wedge (Gyvavedis = ne) \longrightarrow \textit{ne žinduolis} \end{array}$$

Tik klasifikavimo taisyklės $R_2$ yra poromis nesutaikomos ir sudaro pilną rinkinį.

# Klasifikavimo taisyklių tvarka



1. **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.
2. **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 ).
   - 2.1 **Kokybinis rūšiavimas**. Visos taisyklės išdėstomos kokybės mažėjimo tvarka.
   - 2.2 **Rūšiavimas pagal klases**. Vieną klasę priskiriančios taisyklės sąraše stovi greta. Jų tarpusavio išsidėstymas nėra svarbus, nes neturi reikšmės kuri tos pačios klasės taisyklė klasifikuos įrašą.

## Pavyzdys (klasifikavimo taisyklių rūšiavimas)

**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*

# Klasifikavimo taisyklių konstravimo metodai



Nagrinėsime surūšiuotus pagal klases taisyklių rinkinius ir jų sudarymo metodus.

# Klasifikavimo taisyklių konstravimo metodai



Nagrinėsime surūšiuotus pagal klases taisyklių rinkinius ir jų sudarymo metodus.

- **Tiesioginiai metodai** skaido turimą įrašų aibę į mažesnius poaibius taip, kad visus vieno poaibio įrašus klasifikuotų viena taisyklė.

# Klasifikavimo taisyklių konstravimo metodai



Nagrinėsime surūšiuotus pagal klases taisyklių rinkinius ir jų sudarymo metodus.

- **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.

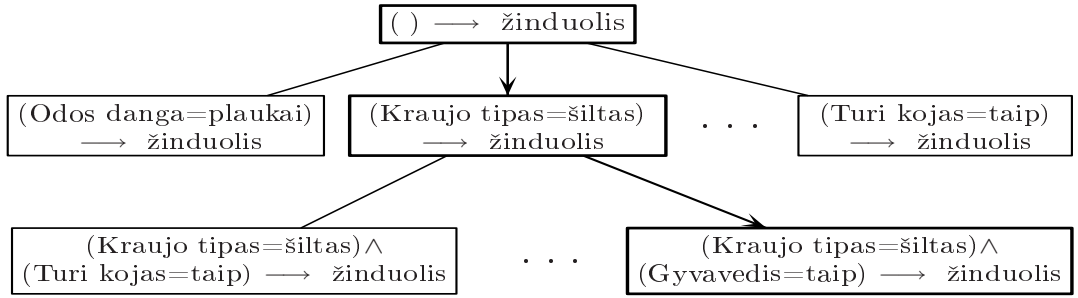
# Nuoseklaus dengimo algoritmas



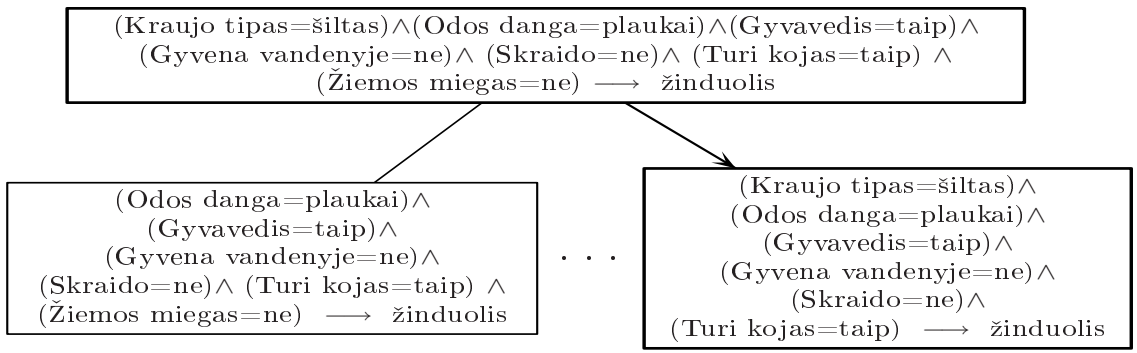
1. Tegul $E$ - mokymo imtis
2. $A_Y := \{y_1, y_2, \ldots, y_k\}$ - surūšiuota visų klasių aibė
3. $r_d = \{() \longrightarrow y_k\}$ - taisyklė "pagal nutylėjimą"
4. $R := \{\}$ - pradinis taisyklių sąrašas
5. **for** $\forall y \in A_Y \backslash \{y_k\}$ **do**
6. **while** netenkinama stabdymo sąlyga **do**
7. $r = \texttt{vienaTaisyklė}(E, y)$
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"

# Dvi klasifikavimo taisyklės konstravimo strategijos

Funkcija `vienaTaisyklė`$(E, y)$ pagal turimą įrašų aibę $E$ ir pasirinktą kokybės matą klasei $y$ konstruoja geriausią taisyklę $r$. Klasei $y$ priklausantys įrašai vadinami teigiamais, o likusieji - neigiamais. Dvi galimos strategijos:




(**a**) Nuo paprasto prie sudėtingo




(**b**) Nuo sudėtingo prie paprasto

# Klasifikavimo taisyklių kokybės matai



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.

**Pavyzdys.** 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.

# Tikėtinumo statistika



Mažos apimties taisyklių eliminavimui galima taikyti statistinius kriterijus. Pavyzdžiui tikėtinumo statistikos

$$L = 2 \sum_{i=1}^{k} n_i \ln \frac{n_i}{e_i} \qquad (1)$$

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 $\chi^2$ 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 (1) sumoje taip pat lygus $0$.

# Tikėtinumo statistika



**Pavyzdys.** Tegul mokymo imtyje yra 60 teigiamų ir 100 neigiamų įrašų. Turime $k = 2$ klases: {+, -}.

$r_1$ : apima 50 teigiamų ir 5 neigiamus įrašus,

$r_2$ : apima 2 teigiamus įrašus ir nė vieno neigiamo įrašo.

$$r_1: \quad e_+ = 55 \cdot \frac{60}{160} = 20,625\,, \quad e_- = 55 \cdot \frac{100}{160} = 34,375\,,$$

$$L(r_1) = 2 \cdot \left(50 \cdot \ln \frac{50}{20,625} + 5 \cdot \ln \frac{5}{34,375}\right) \approx 69,27\,.$$

$$r_2: \quad e_+ = 2 \cdot \frac{60}{160} = 0,75\,, \quad e_- = 2 \cdot \frac{100}{160} = 1,25\,,$$

$$L(r_2) = 2 \cdot \left(2 \cdot \ln \frac{2}{0,75} + 0\right) \approx 3,92\,.$$

Todėl pagal šį kriterijų taisyklė $r_1$ yra geresnė už $r_2$.

# Modifikuotas tikslumas



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

$$\theta_L(r) = \frac{n_+ + 1}{n + k}$$

ir vadinamasis m - įvertis

$$\theta_m(r) = \frac{n_+ + k\, p_+}{n + k}\,,$$

čia $k$ - klasių skaičius; $p_+$ - apriorinė teigiamos klasės tikimybė. Pastebėsime, kad

$$\theta_L(r) = \theta_m(r)\,,\ \text{kai}\ p_+ = \frac{1}{k}\,.$$

# Informacijos prieaugis



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

$$I(r_0, r) = m_+ \cdot \left( \log_2 \frac{n}{n_+} - \log_2 \frac{m}{m_+} \right) = m_+ \cdot \log_2 \frac{\theta(r)}{\theta(r_0)} .$$

Atskiru atveju, kai taisyklės $r_0$ prielaida tuščia $A = (\ )$, informacijos prieaugį žymėsime $I(r)$.

Iš visų taisyklės $r_0$ modifikacijų $r$ reikėtų rinktis tą, kuri labiausiai sumažina neapibrėžtumą, t.y. turi didžiausią informacijos prieaugį $I(r_0, r)$.

# 1R algoritmas (1-Rule, R.C.Holte, 1993)



Tarkime, kad visi imties atributai $X_1, X_2, \ldots, X_k$ yra kategoriniai.

Algoritmo schema

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 $r(y_{\max})$
7. **end for**
8. **end for**
9. $R = \underset{R_i}{\operatorname{argmax}}\, \theta(R_i)$ - randamas geriausias klasifikatorius

# RIPPER algoritmas (W.W.Cohen, 1995)



RIPPER (Repeated Incremental Pruning to Produce Error Reduction) algoritmas generuoja pagal klases surūšiuotą klasifikavimo taisyklių rinkinį (nuo rečiausios klasės iki dažniausios).

1. Kiekvieną taisyklę $r$ RIPPER konstruoja naudodamas "nuo paprasto prie sudėtingo" strategiją. Taisyklė modifikuojama, kol pasidaro tiksli, t.y. $\theta(r) = 1$ . Papildomų konjunktų pasirinkimo kriterijus yra informacijos prieaugio dydis.
2. 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.

# RIPPER algoritmas (W.W.Cohen, 1995)

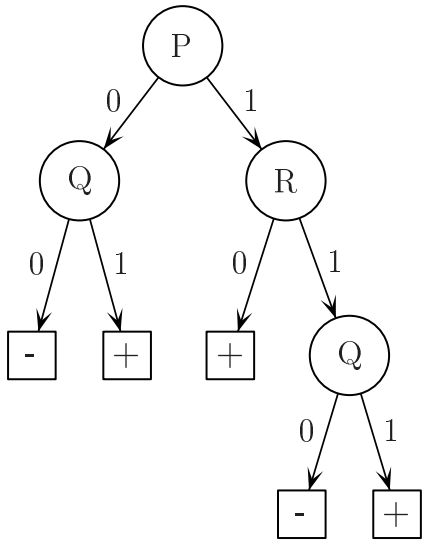


Toliau, pagal nuoseklaus dengimo metodą, iš mokymo imties pašalinami įrašai, kuriuos apima sukonstruotoji taisyklė ir pereinama prie kitos taisyklės generavimo. Beje taisyklė neapima įrašų, kuriuose trūksta bent vieno į taisyklės prielaidą įeinančio atributo reikšmės. RIPPER algoritmas stabdomas, kai 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.

# Netiesioginis klasifikavimo taisyklių konstravimas




$\Longrightarrow$

| Klasifikavimo taisyklės | |
|---|---|
| $r_1:$ | $(P=0) \wedge (Q=0) \longrightarrow$ - |
| $r_2:$ | $(P=0) \wedge (Q=1) \longrightarrow$ + |
| $r_3:$ | $(P=1) \wedge (R=0) \longrightarrow$ + |
| $r_4:$ | $(P=1) \wedge (R=1) \wedge (Q=0) \longrightarrow$ - |
| $r_5:$ | $(P=1) \wedge (R=1) \wedge (Q=1) \longrightarrow$ + |

Galutinai gauname tokį sutvarkytą taisyklių rinkinį

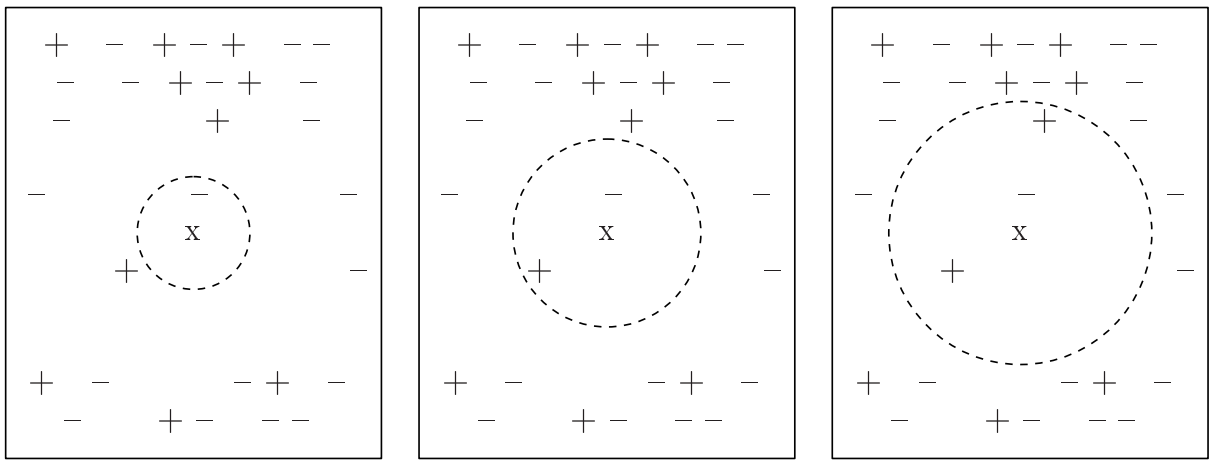
$$(Q=1) \longrightarrow +$$
$$(P=1) \wedge (R=0) \longrightarrow +$$
$$() \longrightarrow -$$

# Artimiausių kaimynų metodas



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.

Rezultatas priklausys ne tik nuo artumo mato pasirinkimo, bet ir nuo sprendimą įtakojančių kaimynų skaičiaus $K$ .

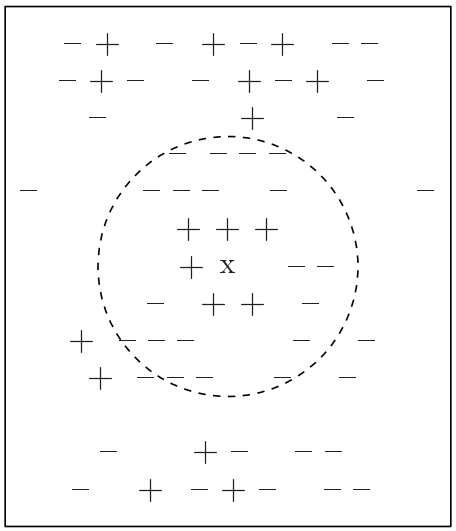



(a) 1 artimiausias kaimynas (b) 2 artimiausieji kaimynai (c) 3 artimiausieji kaimynai

# Artimiausių kaimynų metodas



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.



# Artimiausių kaimynų metodas



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 $w(\mathbf{x})$ . 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.
Jei visi kaimynai vienodai svarbūs, tai $w(\mathbf{x}) \equiv 1$.
Artimiausių kaimynų klasifikatorius įrašą $z$ priskiria klasei $c$ , kurios suma $S(c, E_z)$ yra didžiausia.

# Artimiausių kaimynų klasifikatoriaus algoritmas



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
$$y' = \underset{c}{\operatorname{argmax}}\, S(c, E_z)$$
7. **end for**