Internet a klasifikační metody, 5. přednáška

Kdy je klasifikace srozumitelná uživateli?

volitelný předmět pro magisterské studium

Martin Holeňa


I don‘t know why, but
our deep network estimates your chance
to recover as low as 16 %

O čem to bude?

- Srozumitelné vyjádření klasifikace pomocí jazyka logiky
  - konstrukce pravidel booleovské a fuzzy logiky
  - observační pravidla a způsoby jejich konstrukce
- Získávání klasifikačních pravidel z klasifikačních stromů
  - učení klasifikačních stromů
  - prořezávání klasifikačních stromů

Jak udělat klasifikátor srozumitelný?

- Připomenutí: klasifikátor je zobrazení $F: X \to \{C_1, \ldots, C_m\} | \{1, -1\} | \{1, 0\}, \ldots$

př. $F(x) = \arg \max_{i=1,\ldots,m} \frac{1}{\sqrt{2\pi|\Sigma_i|}} e^{-\frac{1}{2}(x-\beta_i)^\top \Sigma_i^{-1}(x-\beta_i)}, = \begin{cases} 1 \text{ při } \sum_{i \in S} \hat{\alpha}_i c_i \kappa(x, x_i) + \hat{b} \geq 0 \\ -1 \text{ při } \sum_{i \in S} \hat{\alpha}_i c_i \kappa(x, x_i) + \hat{b} < 0 \end{cases}$

Jak udělat klasifikátor srozumitelný?

- Připomenutí: klasifikátor je zobrazení $F: X \to \{C_1, \ldots, C_m\} | \{1, -1\} | \{1,0\}, \ldots$

př. $F(x) = \arg\max_{i=1,\ldots,m} \frac{1}{\sqrt{2\pi|\Sigma_i|}} e^{-\frac{1}{2}(x-\beta_i)^\top \Sigma_i^{-1}(x-\beta_i)} = \begin{cases} 1 \text{ při } \sum_{i \in S} \hat{\alpha}_i c_i \kappa(x, x_i) + \hat{b} \geq 0 \\ -1 \text{ při } \sum_{i \in S} \hat{\alpha}_i c_i \kappa(x, x_i) + \hat{b} < 0 \end{cases}$

  - matematický koncept → málo srozumitelný běžnému uživateli

- Co když $F: X \to \{1,0\}$ *popisuje pravdivost* nějaké $\varphi \to \psi$ nebo $\varphi \leftrightarrow \psi$?
  - pak stačí znát *implikaci* $\varphi(x) \to \psi(x)$ nebo *ekvivalenci* $\varphi(x) \leftrightarrow \psi(x)$
  - *pravidlo logiky*: srozumitelnější než matematická formule

Booleovská a fuzzy pravidla

- *Booleovská* ekvivalence: pravdivost $\|\varphi \leftrightarrow \psi\| = 1$ při $\|\varphi\| = \|\psi\|$,

  implikace: $\|\varphi \rightarrow \psi\| = 1$ při $\|\varphi\| = 0$ nebo při $\|\varphi\| = \|\psi\| = 1$

  - typicky používáno při $\|\varphi\| = 1$, pak $\varphi \rightarrow \psi$ splývá s $\varphi \leftrightarrow \psi$

- *Fuzzy* implikace: $\|\varphi \rightarrow \psi\| = \sup\{t: \|\varphi\| * t \leq \|\psi\|\}$, $*$ – t-norma,

  fuzzy ekvivalence: $\|\varphi \leftrightarrow \psi\| = \|\varphi \rightarrow \psi\| * \|\psi \rightarrow \varphi\|$

  - důležité případy: $\|\varphi\| \leq \|\psi\|$ $(\|\varphi \rightarrow \psi\| = 1)$, $\|\varphi\| = \|\psi\|$ $(\|\varphi \leftrightarrow \psi\| = 1)$

Evoluce klasifikačních pravidel

- *Evoluční algoritmy* − informatická aplikace biologického vývoje

  druhů: výraznější *posilování žádoucích vlastností*

  - nejběžnější typ − genetické algoritmy: křížení + mutace

    genomu − řetězec kódující hodnoty sledovaných proměnných

- Použití na klasifikační pravidla: posilování AC, PR, TPr, TNr, AUC,

  pokud φ, ψ jsou konjunkce → malý počet atomů

Typy populací pravidel

- Všechny evoluční algoritmy jsou populační: paralelní

  pro populaci = soubor problémově závislých jedinců

- *Michiganský* přístup: *jedinec = pravidlo*, hledáme populaci
  - jednoduchý, ale zbytečně dostáváme podobná pravidla
- *Pittsburghský* přístup: jedinec = hledaný *soubor pravidel*
  - evoluce populace souborů: složitá + výpočetně náročná

Použití pravidel při filtraci spamu

- Hlavně filtrace na základě *metainformací*, např.
  - bylo manipulováno s *odesilatelem* | *datem* odeslání
  - hlavní část těla zprávy je *grafika*
- Při kombinování booleovskou disjunkcí: *vysoká FP*, např.
  - např. grafické tělo zprávy → automaticky spam
  - → Łukasiewiczova fuzzy disjunkce pravidel $\varphi \rightarrow \psi$ se stejným ψ

```
body HAS_VIAGRA_ON_BODY eval("[vV][iI?1!][aA][gG][rR][aA]")
describe HAS_VIAGRA_ON_BODY Contains references to viagra in content
score HAS_VIAGRA_ON_BODY 1.5

body HAS_LEVITRA_ON_BODY eval("[lL][eE?1!][vV][iI][tT][rR][aA]")
describe HAS_LEVITRA_ON_BODY Contains references to levitra in content
score HAS_LEVITRA_ON_BODY 1.5

header HAS_VIAGRA_ON_SUBJECT eval_header("Subject","[vV][iI?1!][aA][gG][rR][aA]")
describe HAS_VIAGRA_ON_SUBJECT Contains references to viagra on subject
score HAS_VIAGRA_ON_SUBJECT 1.5

header HAS_LEVITRA_ON_SUBJECT eval_header("Subject","[lL][eE?1!][vV][iI][tT][rR][aA]")
describe HAS_LEVITRA_ON_SUBJECT Contains references to levitra on subject
score HAS_LEVITRA_ON_SUBJECT 1.5

meta HAS_DRUGS_ON_BODY HAS_VIAGRA_ON_BODY & HAS_LEVITRA_ON_BODY
describe HAS_DRUGS_ON_BODY Contains references to viagra and levitra on body
score HAS_DRUGS_ON_BODY +

meta HAS_DRUGS_ON_SUBJECT HAS_VIAGRA_ON_SUBJECT & HAS_LEVITRA_ON_SUBJECT
describe HAS_DRUGS_ON_SUBJECT Contains references to viagra and levitra on subject
score HAS_DRUGS_ON_SUBJECT +
```

Pravidla a doporučovací systémy

- Při kolaborativním filtrování podle chování uživatele
  - uživatel *prošel posloupnost stránek* $\boldsymbol{S}_1,\boldsymbol{S}_2,\ldots,\boldsymbol{S}_n \rightarrow$ doporuč objekt $\boldsymbol{O}$
  - obecněji: uživatel prošel $\boldsymbol{S}_1,\boldsymbol{S}_2,\ldots,\boldsymbol{S}_n \rightarrow$ nabídni další stránku $\boldsymbol{S}_d$
- Jak získávána? − Statistickou *analýzou clickstreamových dat*.
  - odhady pravděpodobností, testování platnosti modelů (hypotéz)
  - z výsledků analýzy konstruujeme pravidla heuristicky

Použití pravidel při detekci malware

- Důvod používání: podezřelé programy nakonec vyhodnocuje *analytik* – chce *rozumět, proč* jsou *podezřelé*
- Pravidel obvykle hodně → odstraňování redundantních nepřesných
- Typicky *hierarchická klasifikace*: 2 vrstvy pravidel
  1. přesná pro normál, citlivá pro malware

Připomínka: binární míry kvality

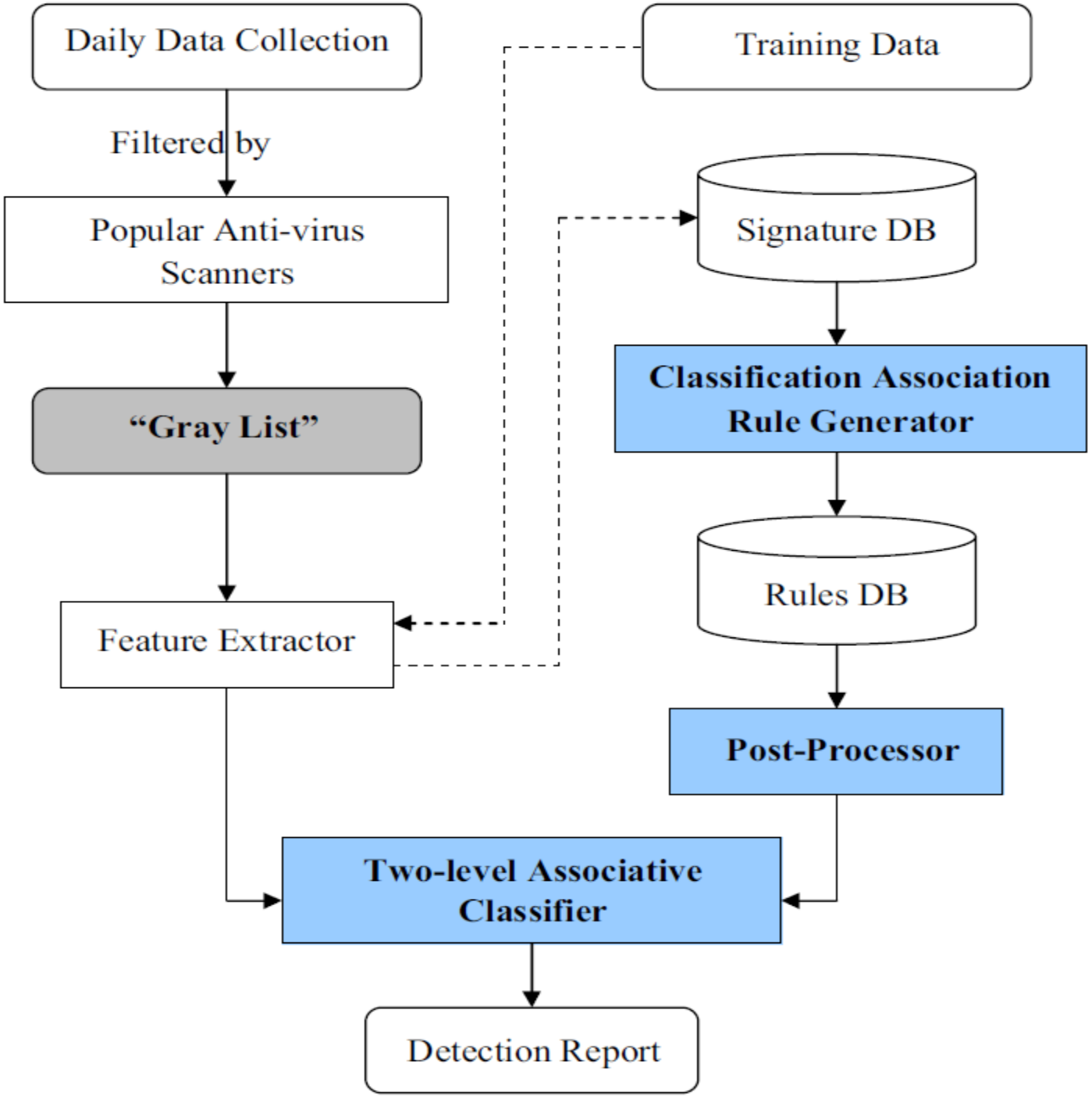
- *Přesnost* (accuracy) $\mathrm{AC} = \frac{\mathrm{TP}+\mathrm{TN}}{n}$
- *Správnost predikce* (precision, predictive value) $\mathrm{PR} = \frac{\mathrm{TP}}{n_{\cdot+}}$
- *Citlivost* (sensitivity), vybavování (recall), detekční poměr $= \mathrm{TPr}$
- Další: *specificita* $= \mathrm{TNr} = \frac{\mathrm{TN}}{n_{-\cdot}} = 1 - \mathrm{FPr}$, F-míra $\mathrm{FM} = 2\frac{\mathrm{PR}*\mathrm{TPr}}{\mathrm{PR}+\mathrm{TPr}}$

  AUC − Area Under Curve (= pod ROC)

- Střední hodnota ceny: $\frac{1}{n}(c_{++}\mathrm{TP}+c_{+-}\mathrm{FN} + c_{-+}\mathrm{FP} + c_{--}\mathrm{TN})$

Použití pravidel při detekci malware

- Důvod používání: podezřelé programy nakonec vyhodnocuje *analytik* – chce *rozumět, proč* jsou *podezřelé*
- Pravidel obvykle hodně → odstraňování redundantních nepřesných
- Typicky *hierarchická klasifikace*: 2 vrstvy pravidel
  1. přesná pro normál, citlivá pro malware
  2. pro rodiny malware, setříděná podle přesnost


Daily Data Collection
Training Data
Filtered by
Popular Anti-virus Scanners
Signature DB
"Gray List"
Classification Association Rule Generator
Rules DB
Feature Extractor
Post-Processor
Two-level Associative Classifier
Detection Report

Observační kalkul pro pravidla z dat

- Booleovská logika popisuje data jen 2 způsoby:
  - „všechny objekty splňují $(\forall)\ \varphi \rightarrow \psi$, resp. $\varphi \leftrightarrow \psi$“: nastává zřídka
  - „existuje objekt splňující $(\exists)\ \varphi \rightarrow \psi$, resp. $\varphi \leftrightarrow \psi$“: slabá informace
- Pro užitečnější popis dat byl navržen (1978)

observační kalkul: má pravidla $\varphi \Omega \psi$, Ω − *zobecněný kvantifikátor*

  - Ω zachycuje výsledek nějaké analýzy dat odpovídajících φ, ψ

Běžné typy observačních pravidel

1. Základem jsou *odhady pravděpodobnosti*, případně podmíněné
   - nejběžnější: *asociační pravidla* $\rightarrow_{\theta,s}$ s konfidencí $\theta \in (0,1)$, podporou $s \in (0,1)$
   - $\|\varphi \rightarrow_{\theta,s} \psi\| = 1$ při $\frac{a}{a+b} \geq \theta \& \frac{a}{a+b+c+d} \geq s$ ($\frac{a}{a+b}$ je nestranný odhad $P(\psi|\varphi)$)
2. Základem jsou *testy hypotéz* (H_0: nulová, H_1: alternativa)
   - např. $\|\varphi \rightarrow_{F,\alpha} \psi\| = 1$ když Fisherův jednostranný test na hladině $\alpha \in (0,0.5)$ zamítá H_0: nezávislost φ, ψ proti H_1: pozitivní závislost φ, ψ

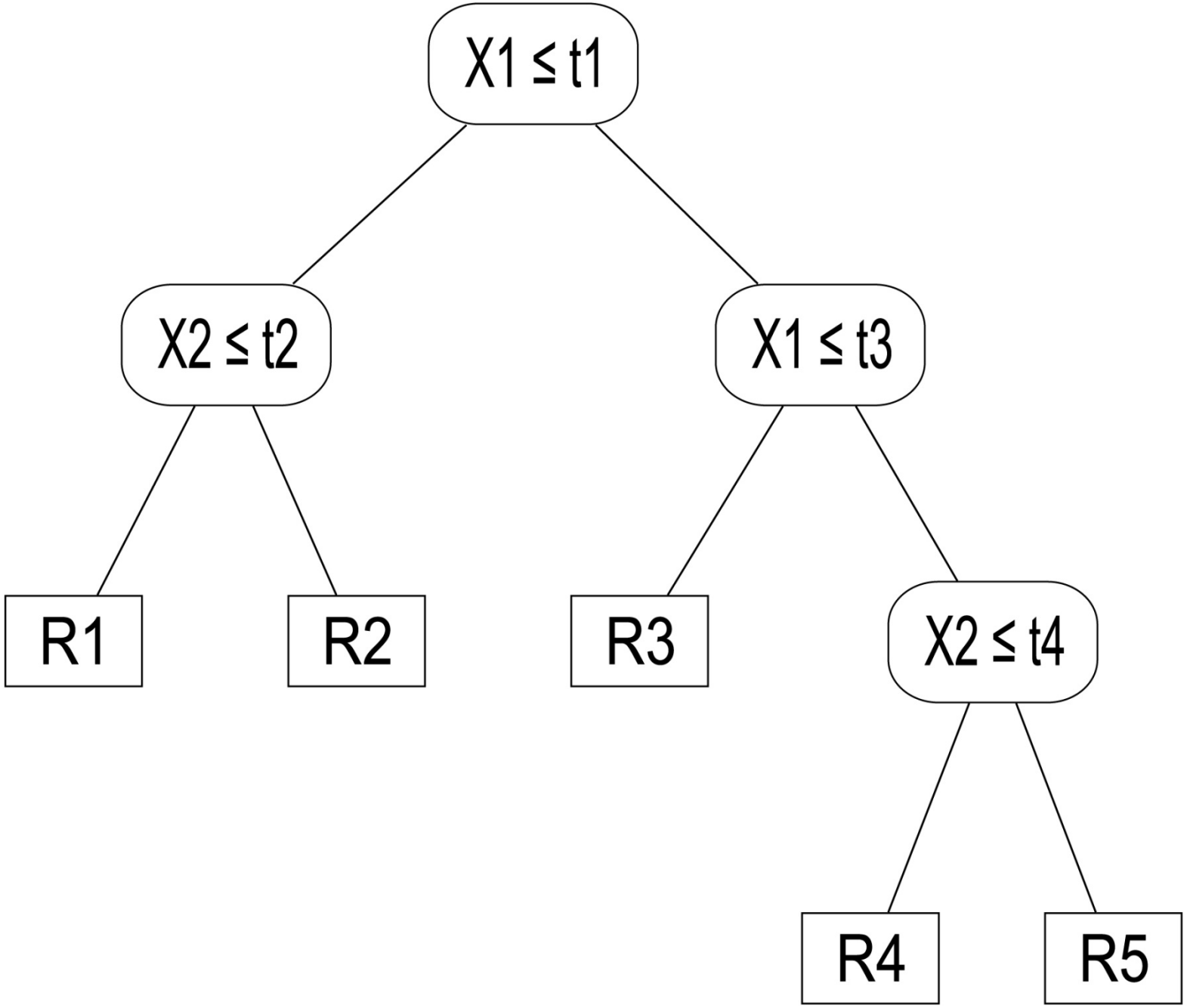
#	ψ	$\neg\psi$
φ	a	b
$\neg\varphi$	c	d

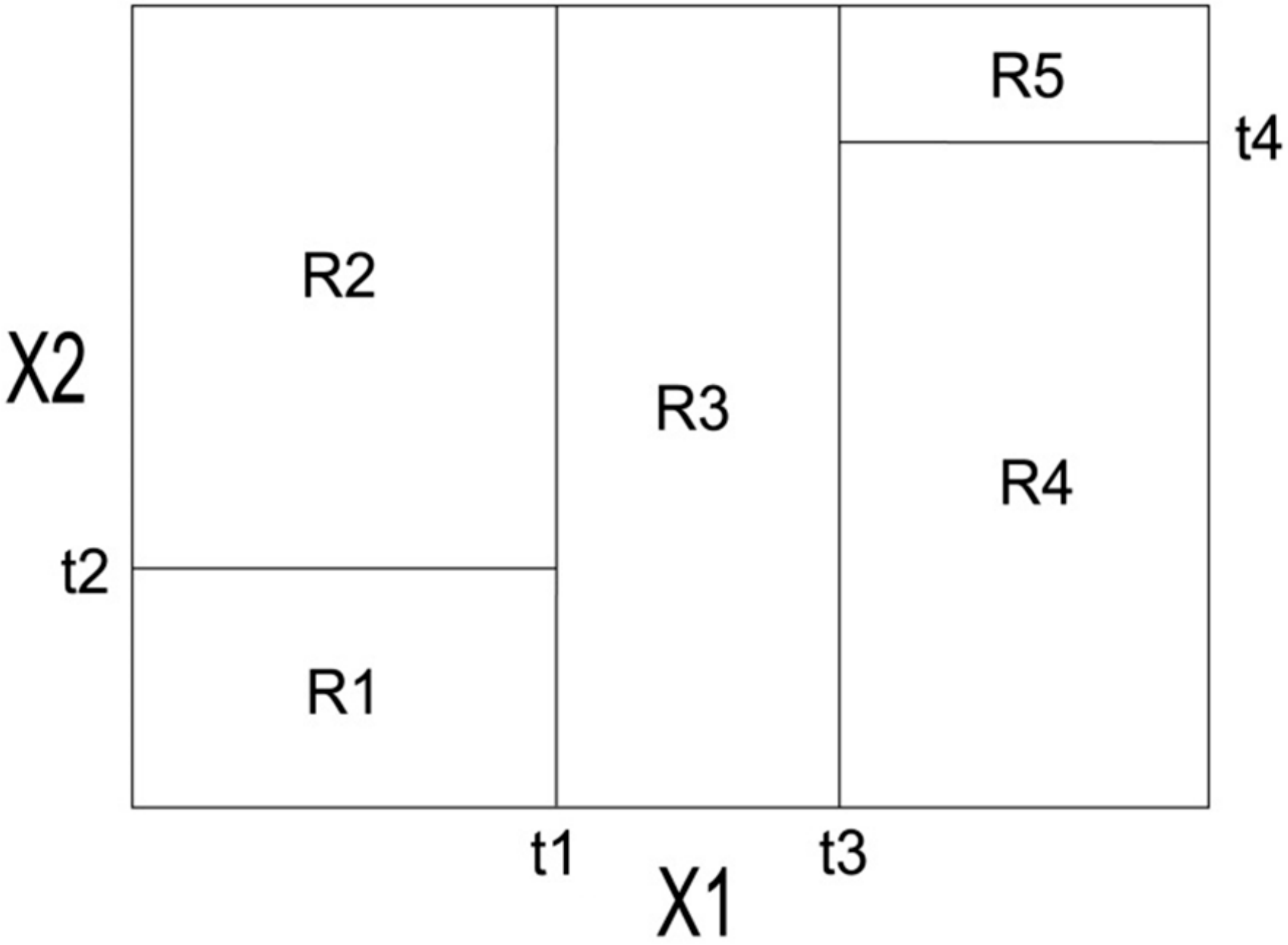
Klasifikační strom a pravidla z něj

- Binární strom, uzly testují booleovské podmínky,

  vstup podle výsledku propadne doleva | doprava

  - podmínky dělí vstupy kolmo na osy


X1 ≤ t1
X2 ≤ t2
X1 ≤ t3
R1
R2
R3
X2 ≤ t4
R4
R5


R5
t4
R2
X2
R3
R4
t2
R1
t1
t3
X1

Klasifikační strom a pravidla z něj

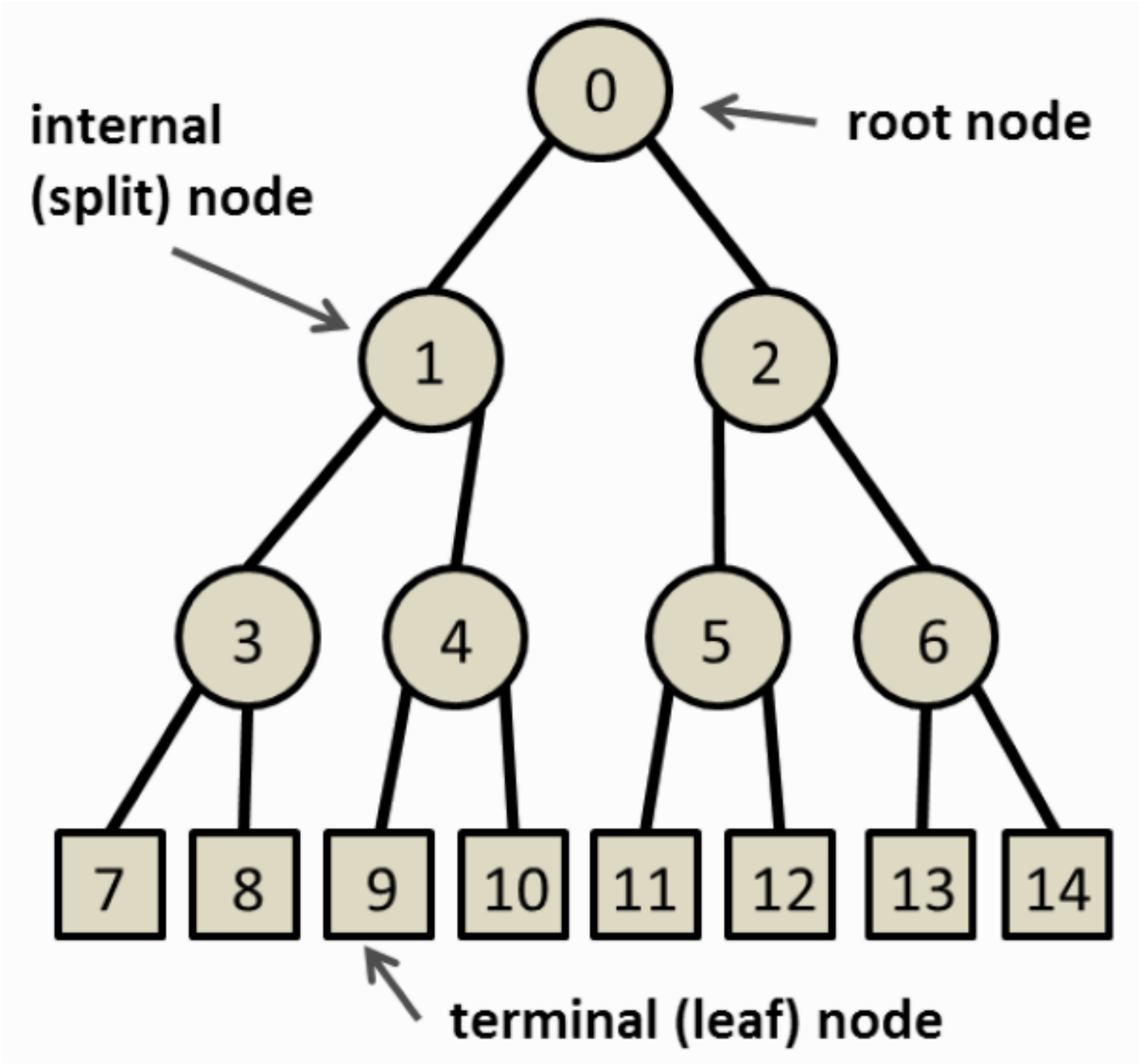
- Binární strom, uzly testují booleovské podmínky,

  vstup podle výsledku propadne doleva | doprava

  - podmínky dělí vstupy kolmo na osy

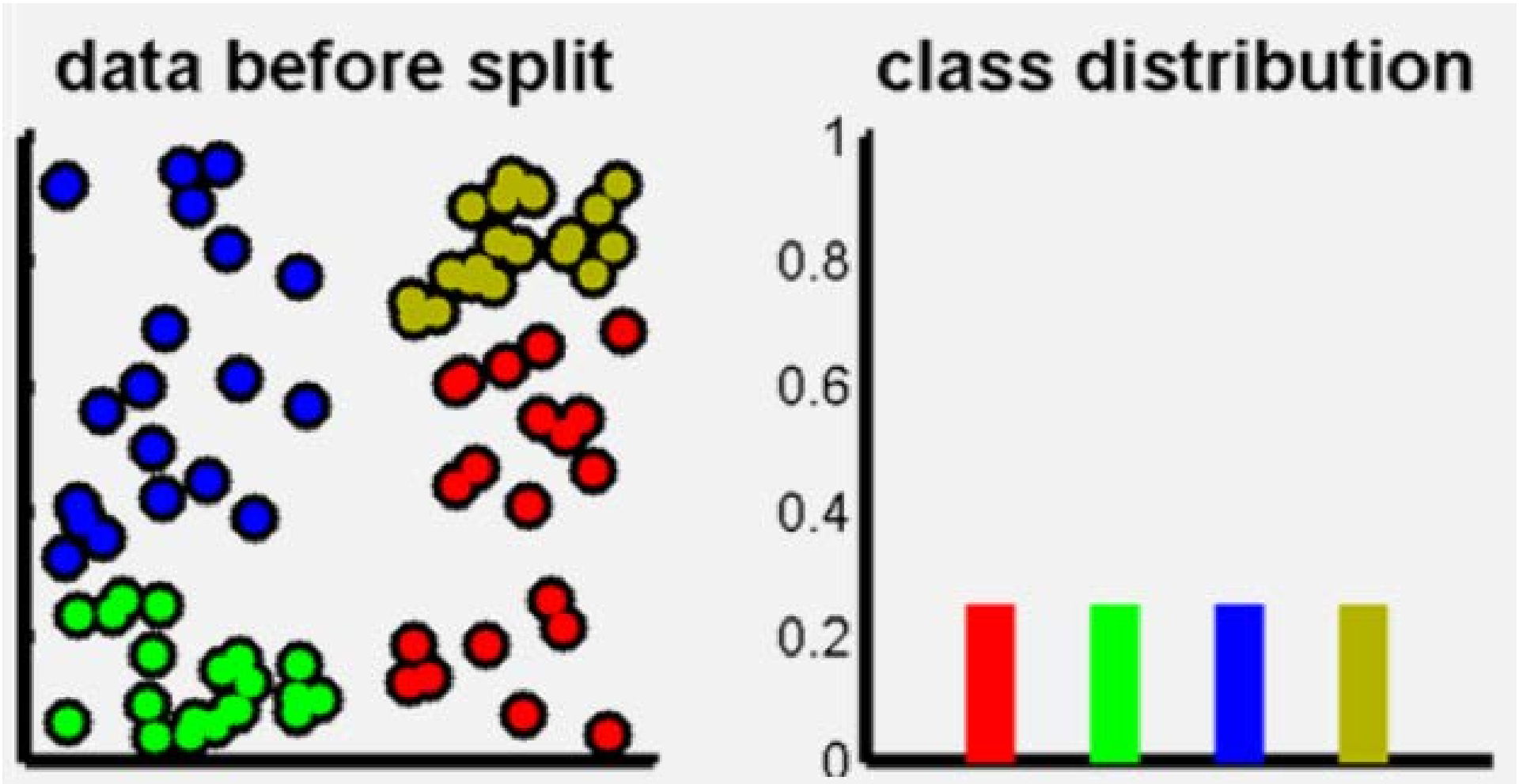
- Pravidla: pro listy − odpovídají jediné třídě

  - *konjunkce podmínek* na cestě kořen → list
  - cesty do *listů téže třídy − disjunkce*

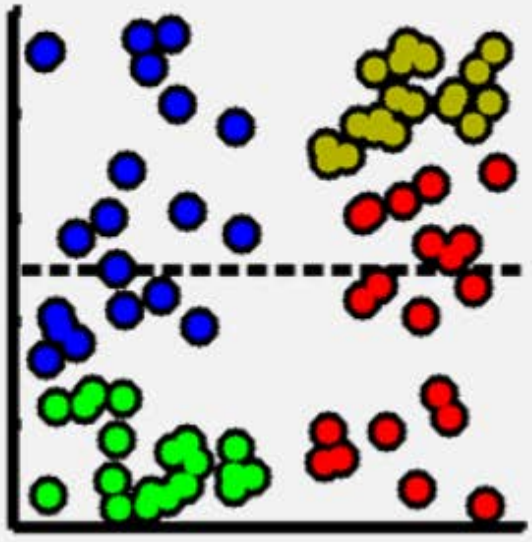

internal
(split) node
root node
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
terminal (leaf) node

Učení klasifikačních stromů

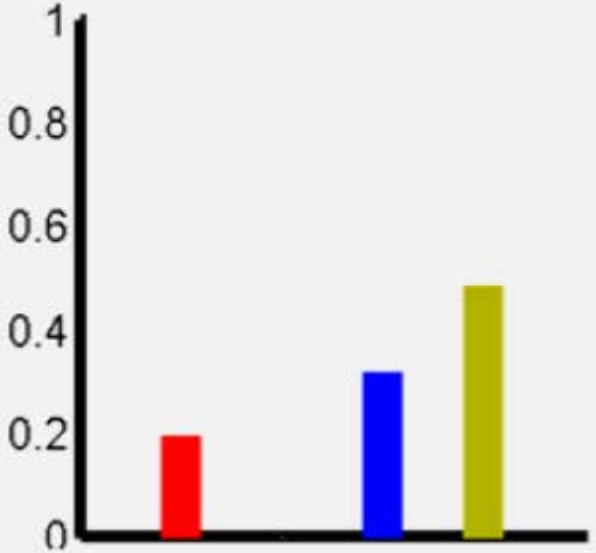
- Dělením v uzlu změníme rozdělení dat


data before split
class distribution
1
0.8
0.6
0.4
0.2
0

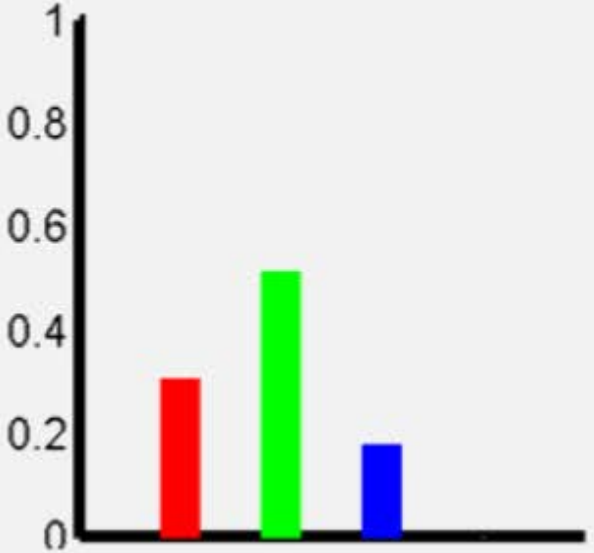
Info Gain = 0.40



top

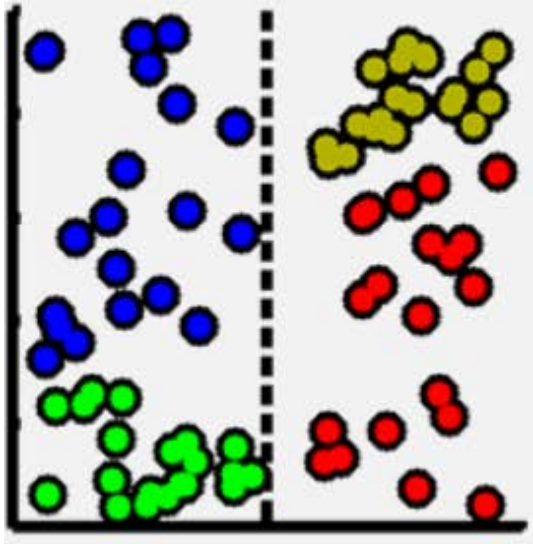


bottom

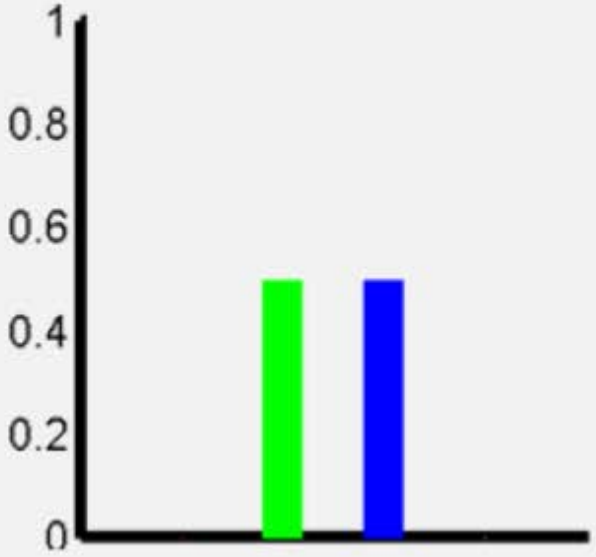


♦

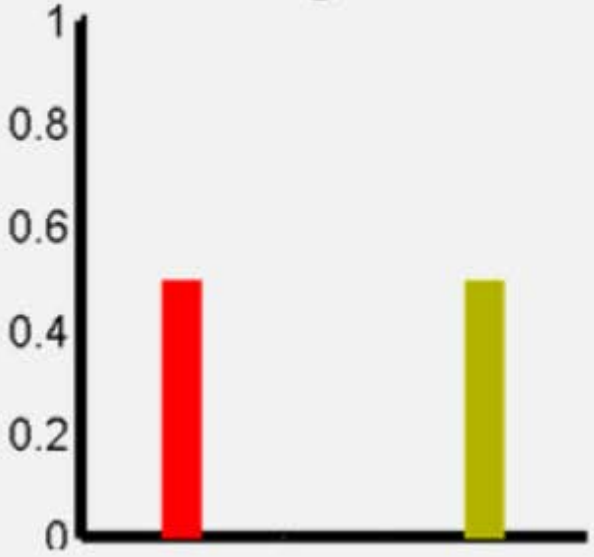
Info Gain = 0.69



left



right

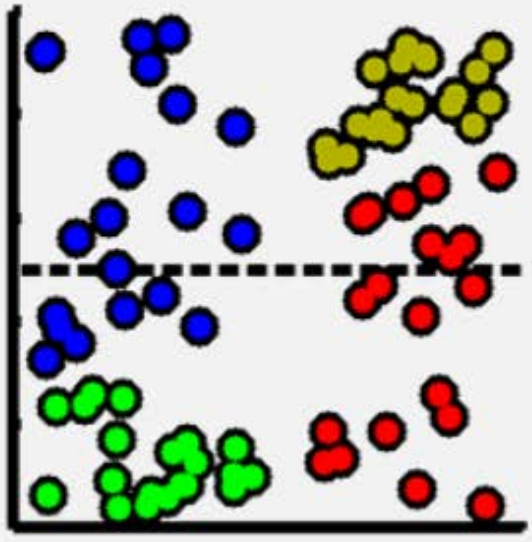


Učení klasifikačních stromů

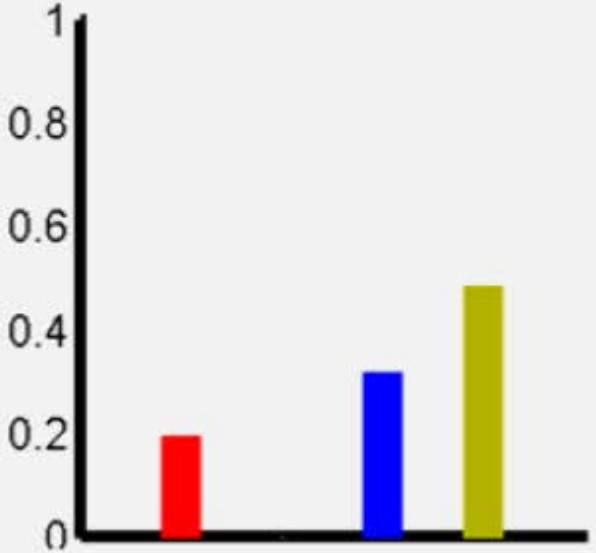
- Dělením v uzlu změníme rozdělení dat
  - princip učení: *optimalizovat dělení* vůči proměnné

    \+ podprostorům spojitých vstupů | podmnožinám diskrétních vstupů

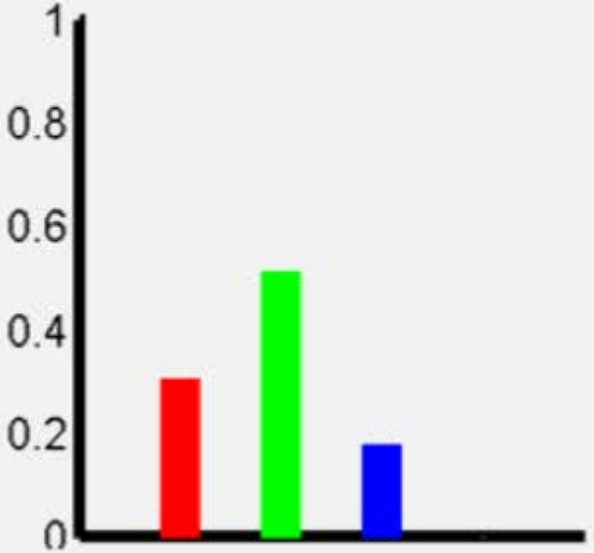
Info Gain = 0.40



top

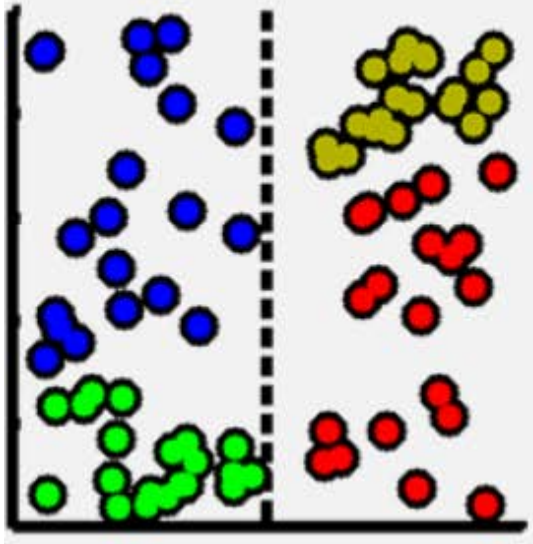


bottom

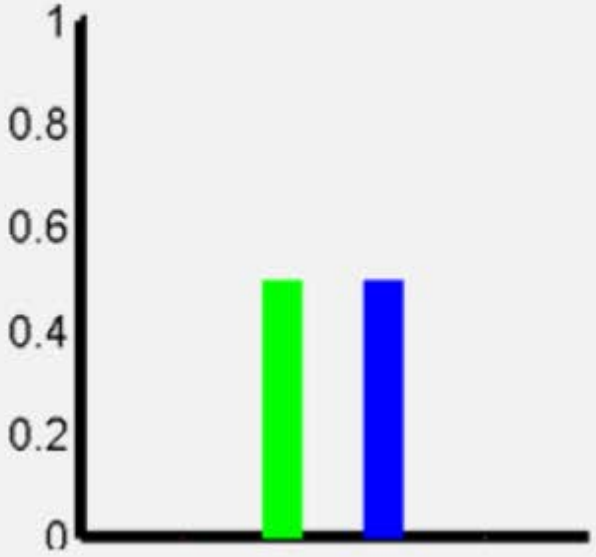


♦

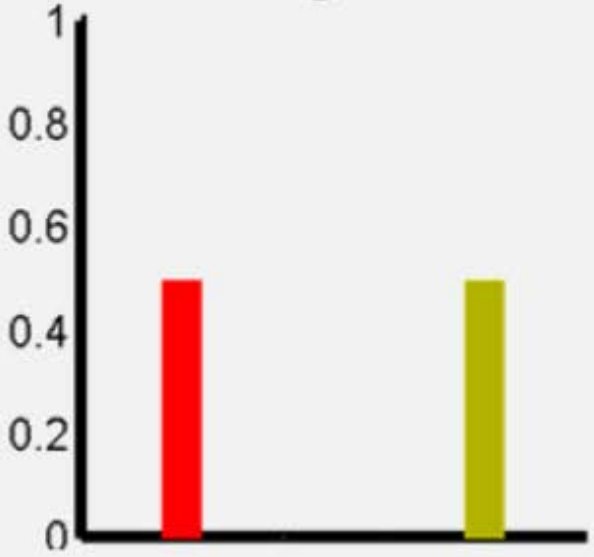
Info Gain = 0.69



left



right



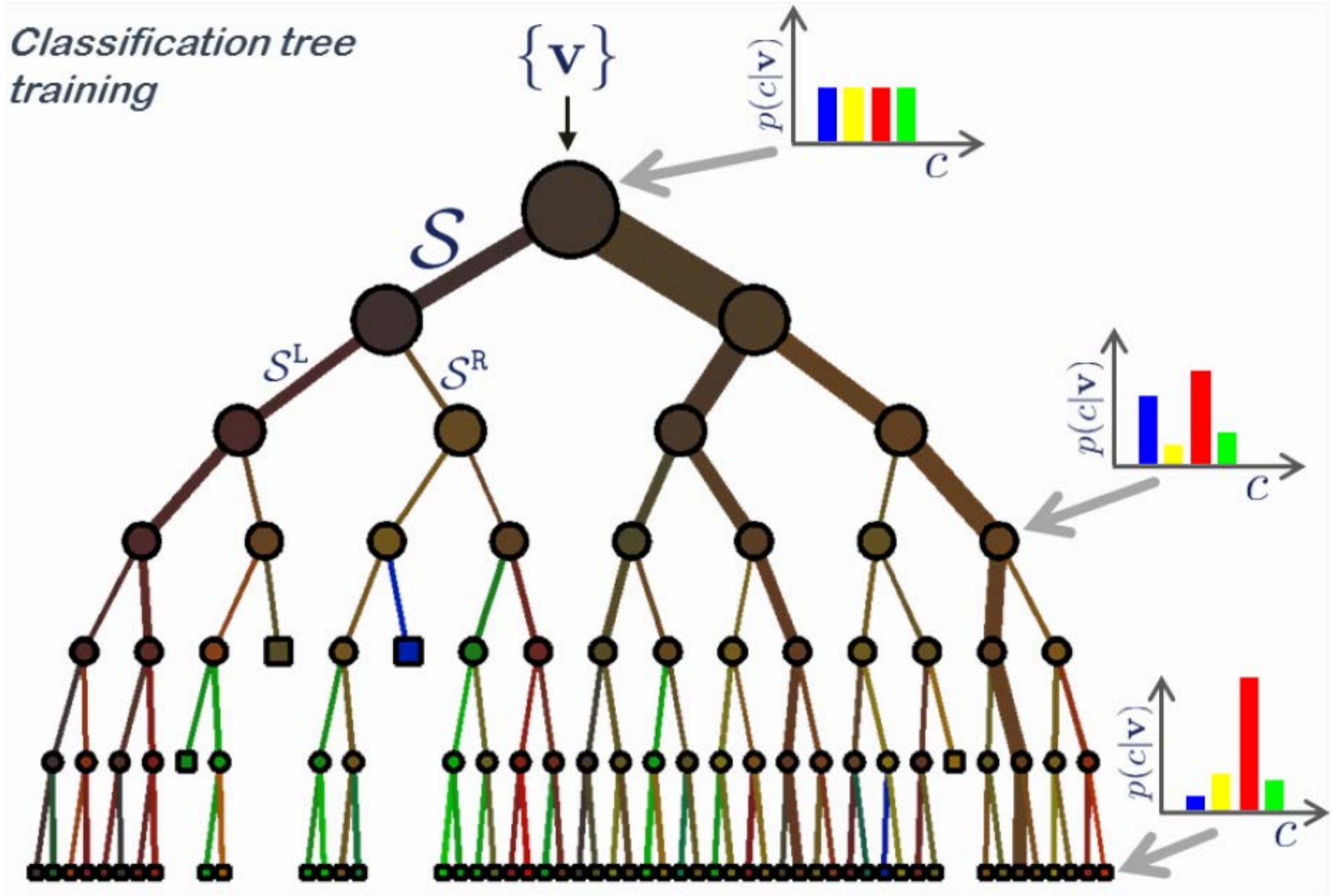
Učení klasifikačních stromů

- Dělením v uzlu změníme rozdělení dat
  - princip učení: *optimalizovat dělení* vůči proměnné

    \+ podprostorům spojitých vstupů | podmnožinám diskrétních vstupů
- Optimum jaké funkce hledáme? 2 nejčastěji používané:
  - *informační zisk* $I(\text{Parent};\text{LChild},\text{RChild}) = H_P - (H_{LC} + H_{RC})$
  - *Giniho index* nečistoty $1 - \sum_{i=1}^{M} P(C_i)^2$

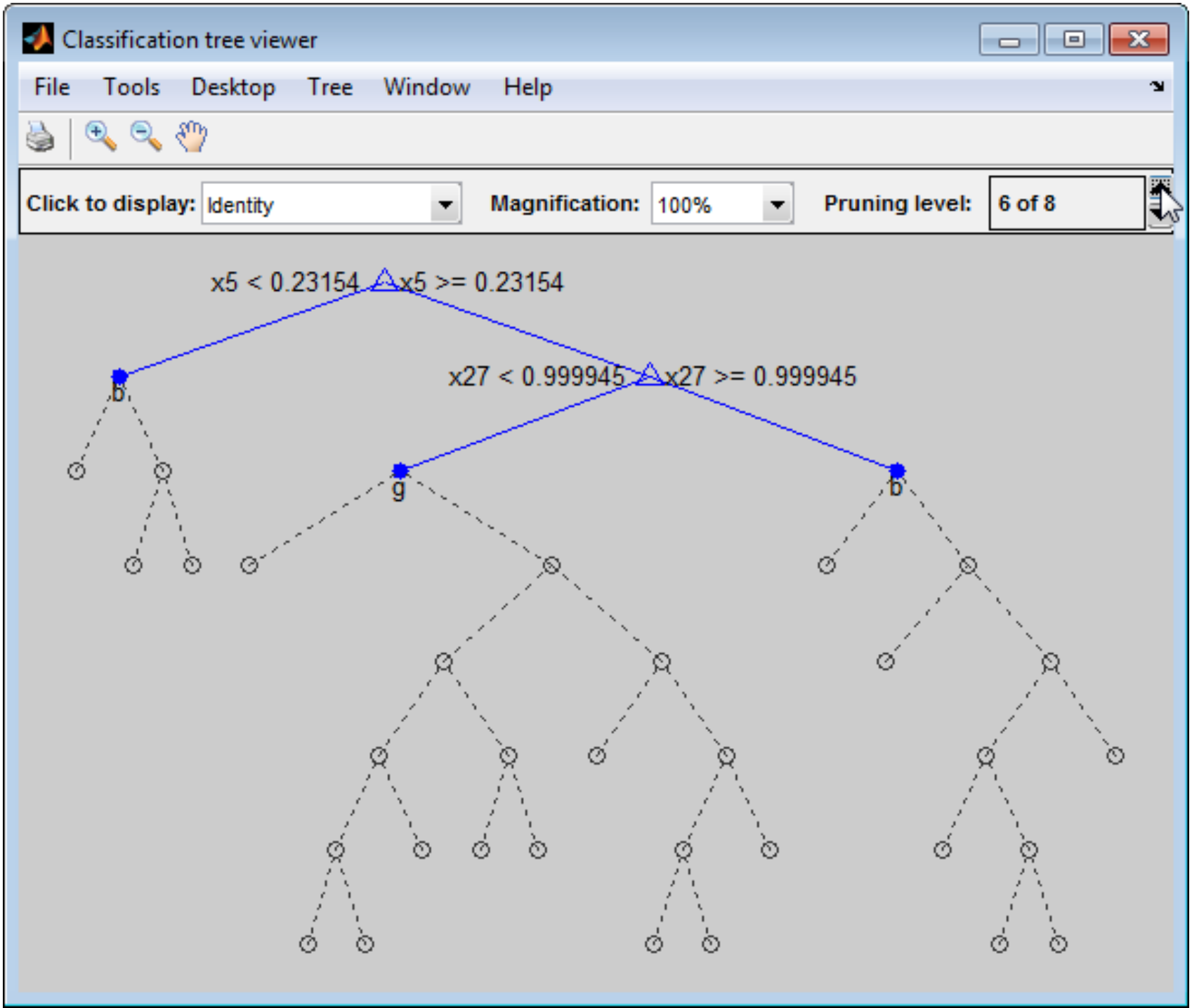
Další vlastnosti klasifikačních stromů

- Klasifikace *nových vstupů*: testováním uzlových podmínek
  - v listech: rozdělení nových = rozdělení učicích


Classification tree training
{v}
p(c|v)
c
S
S^L
S^R
p(c|v)
c
p(c|v)
c

Další vlastnosti klasifikačních stromů

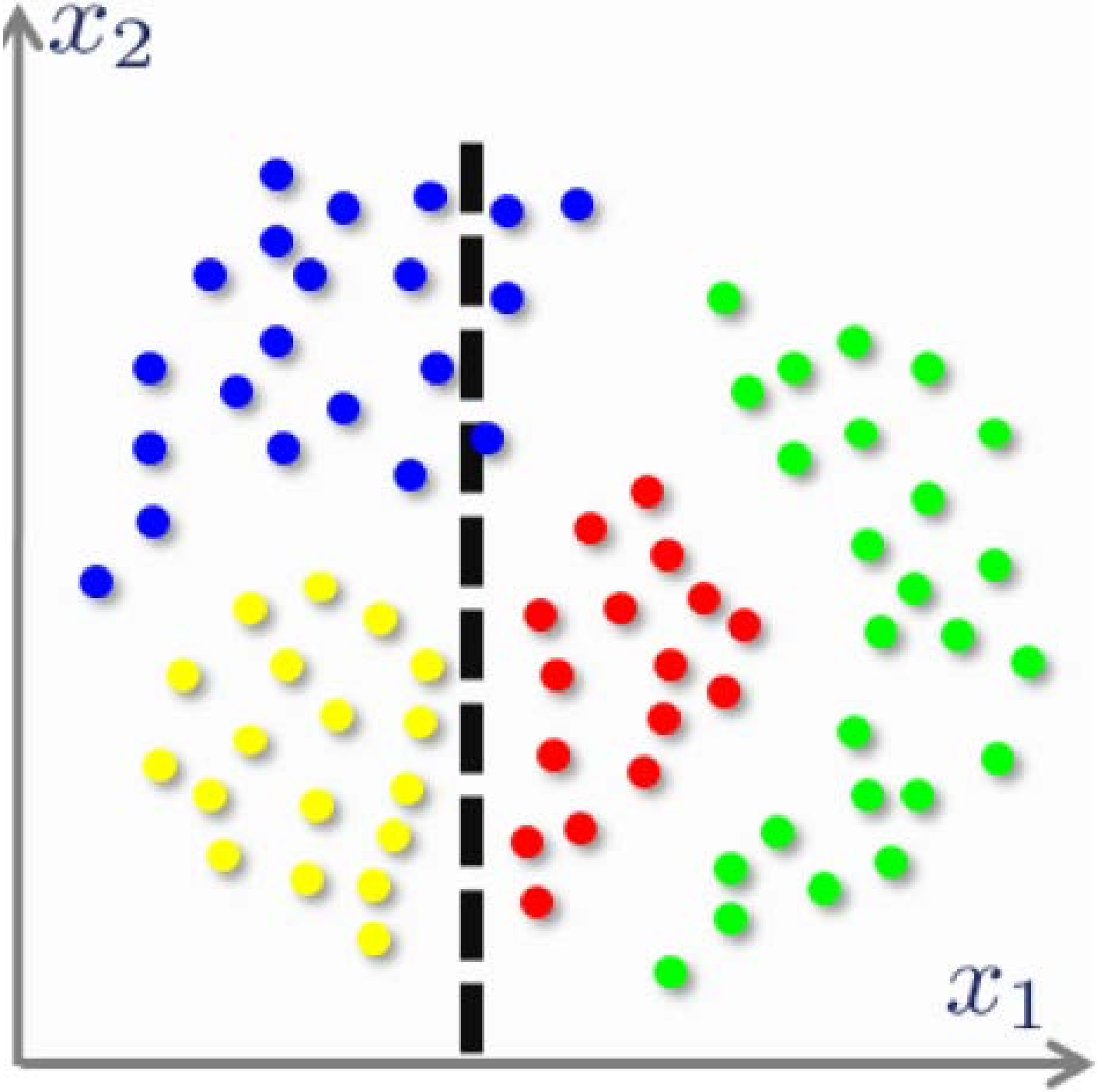
- Klasifikace *nových vstupů*: testováním uzlových podmínek
  - v listech: rozdělení nových = rozdělení učicích
- *Prořezávání* (klestění) stromu − může snížit přeučení

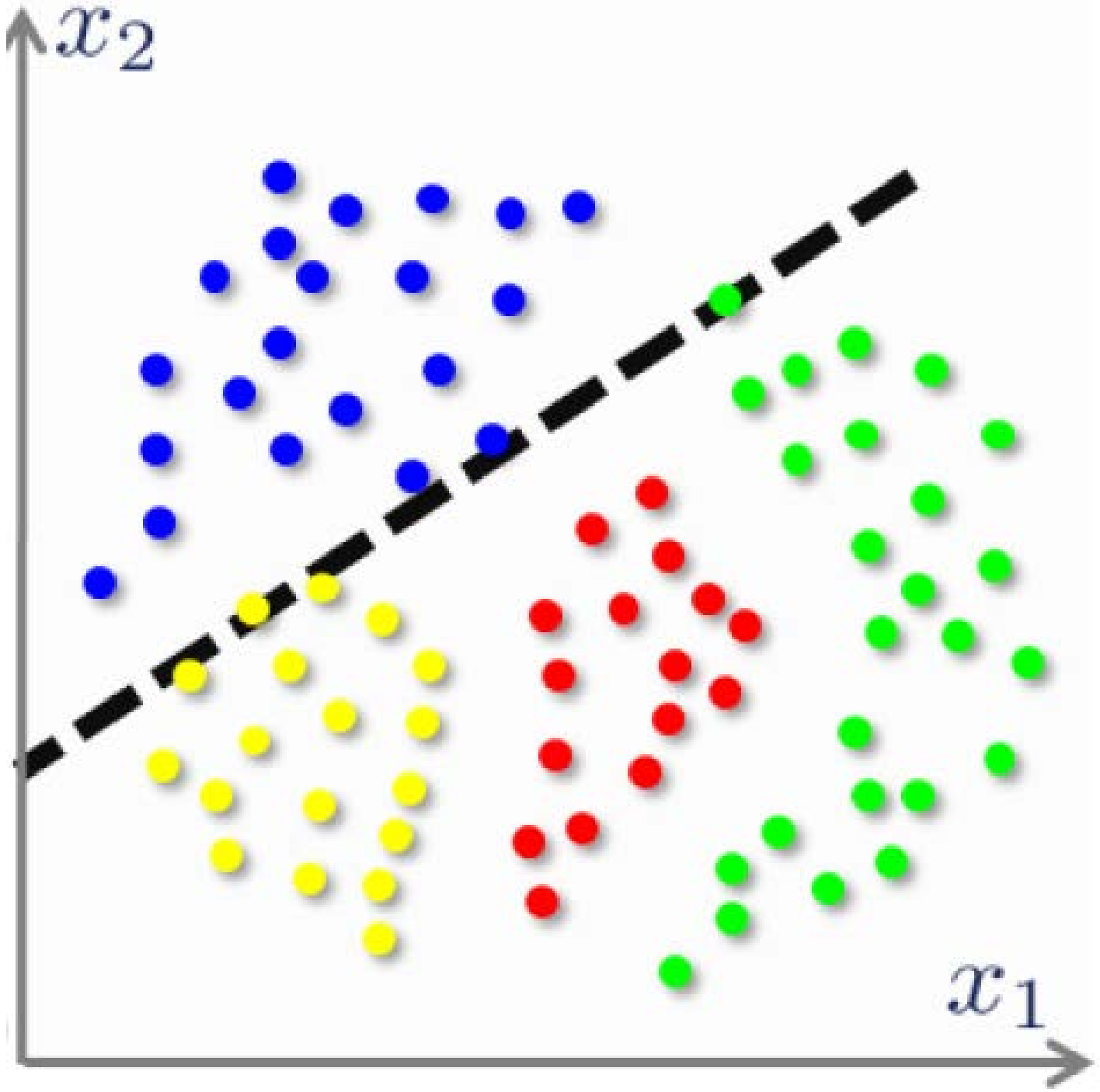

Classification tree viewer
File
Tools
Desktop
Tree
Window
Help
Click to display:
Identity
Magnification:
100%
Pruning level:
6 of 8
x5 < 0.23154
x5 >= 0.23154
x27 < 0.999945
x27 >= 0.999945
b
g
b

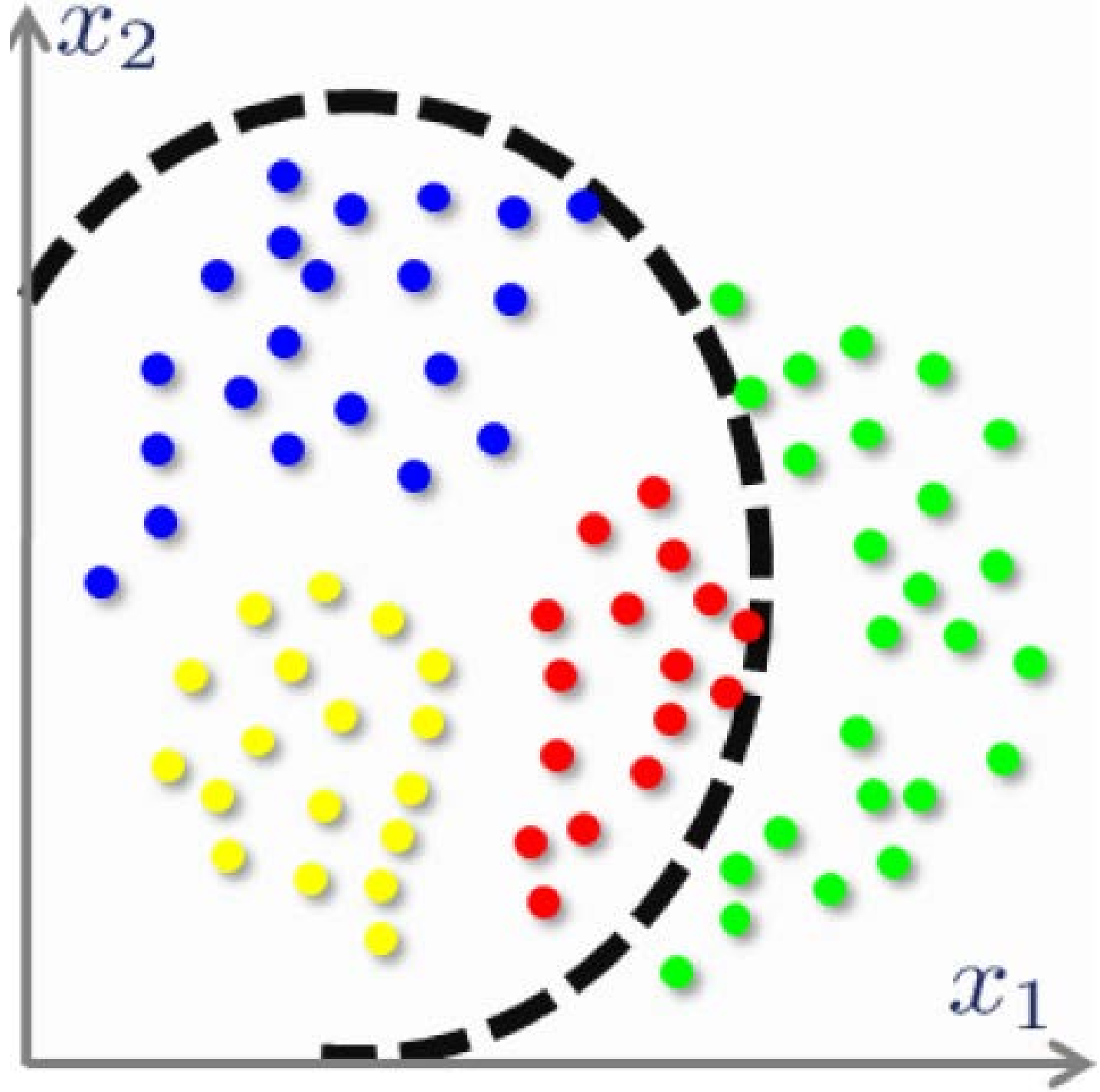
Další vlastnosti klasifikačních stromů

- Klasifikace *nových vstupů*: testováním uzlových podmínek
  - v listech: rozdělení nových = rozdělení učicích
- *Prořezávání* (klestění) stromu − může snížit přeučení
- *Zobecněné dělení vstupů* − místo nadroviny kolmé

  na osu: obecná nadrovina | kvadratická plocha


x_2
x_1


x_2
x_1


x_2
x_1

Další vlastnosti klasifikačních stromů

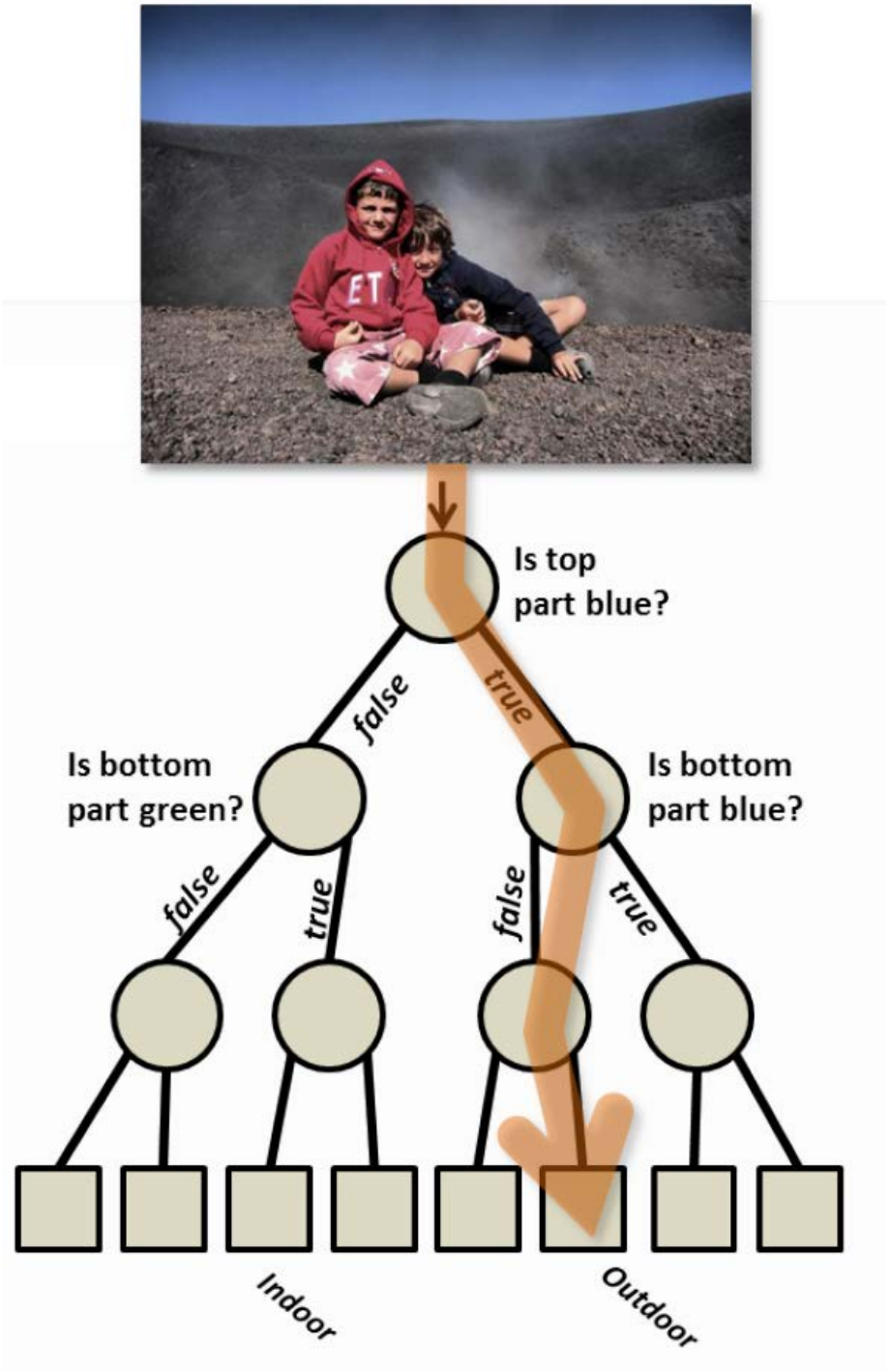
- Klasifikace *nových vstupů*: testováním uzlových podmínek
  - v listech: rozdělení nových = rozdělení učicích
- *Prořezávání* (klestění) stromu − může snížit přeučení
- *Zobecněné dělení vstupů* − místo nadroviny kolmé

  na osu: obecná nadrovina | kvadratická plocha
  - snižuje srozumitelnost pravidel získaných konjunkcí podmínek

Klasifikační stromy při doporučování

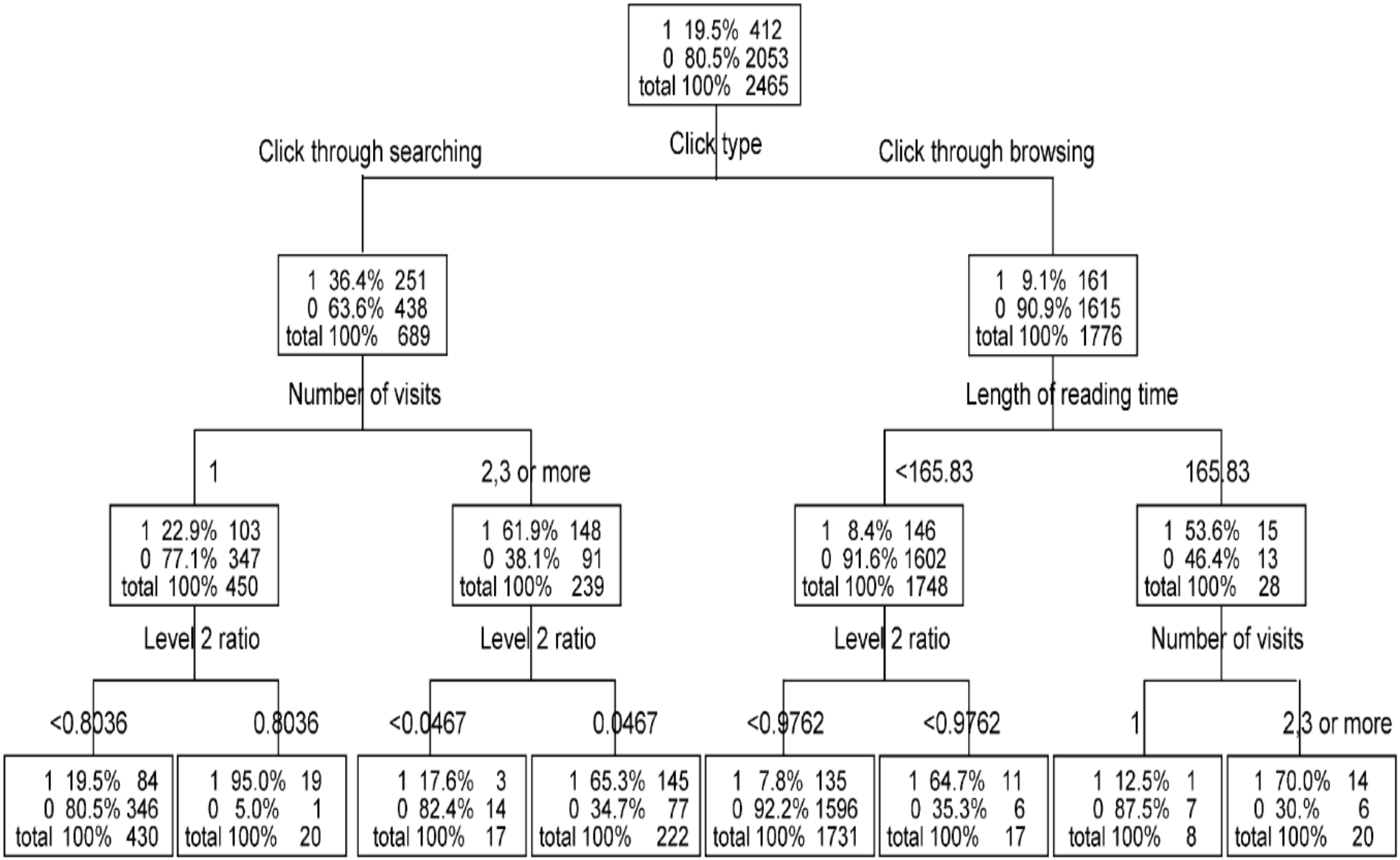
1. obvyklé použití: *klasifikace obrázků* pro doporučování

   - na základě vysokoúrovňových charakteristik (barva pozadí, odlišnost popředí ⋈ pozadí,…) – získány předzpracováním pixelů


Is top part blue?
false
true
Is bottom part green?
Is bottom part blue?
false
true
false
true
Indoor
Outdoor

Klasifikační stromy při doporučování

1. obvyklé použití: *klasifikace obrázků* pro doporučování
   - na základě vysokoúrovňových charakteristik (barva pozadí, odlišnost popředí ⋈ pozadí,…) – získány předzpracováním pixelů
2. obvyklé použití: *predikce zákazníkova akceptování doporučení*
   - dá zboží do košíku?, dokončí nákup?
   - pomocí clickstreamových dat z procházených stránek


1 19.5% 412
0 80.5% 2053
total 100% 2465
Click type
Click through searching
Click through browsing
1 36.4% 251
0 63.6% 438
total 100% 689
1 9.1% 161
0 90.9% 1615
total 100% 1776
Number of visits
Length of reading time
1
2,3 or more
<165.83
165.83
1 22.9% 103
0 77.1% 347
total 100% 450
1 61.9% 148
0 38.1% 91
total 100% 239
1 8.4% 146
0 91.6% 1602
total 100% 1748
1 53.6% 15
0 46.4% 13
total 100% 28
Level 2 ratio
Level 2 ratio
Level 2 ratio
Number of visits
<0.8036
0.8036
<0.0467
0.0467
<0.9762
<0.9762
1
2,3 or more
1 19.5% 84
0 80.5% 346
total 100% 430
1 95.0% 19
0 5.0% 1
total 100% 20
1 17.6% 3
0 82.4% 14
total 100% 17
1 65.3% 145
0 34.7% 77
total 100% 222
1 7.8% 135
0 92.2% 1596
total 100% 1731
1 64.7% 11
0 35.3% 6
total 100% 17
1 12.5% 1
0 87.5% 7
total 100% 8
1 70.0% 14
0 30.% 6
total 100% 20

Další aplikace klasifikačních stromů

- *Filtrace spamu* − hlavně na základě *obsahu*
  - vstupy: pravděpodobnosti některých slov + speciálních znaků (!, ;,…), grafické vlastnosti (počet konsekutivních velkých písmen)
- *Detekce malware* − hlavně na základě *statických* vlastností, tj. struktury + instrukcí programů
  - vstupy: n-gramy kódující profil programu