9. Strojové učení

Strojové učení

- Učení s učitelem (Supervised learning)
- Učení bez učitele (Unsupervised learning)
- Posilované/motivované učení (Reinforcement learning)

Pozn.: Učení je přirozenou vlastností umělých neuronových sítí, jejichž popis je jednou z hlavních náplní magisterského předmětu SFC (Soft-Computing).

Učení s učitelem (supervised learning)

- Tvorba rozhodovacích stromů (Decision Trees Building)
- Hledání prostoru verzí (Version Space Search)
- Rozpoznávání/klasifikace obrazů (Pattern Recognition)

Rozhodovací stromy

- slouží ke klasifikaci objektů na základě hodnot jejich vlastností
- jsou vytvářeny ze známé množiny příkladů (učení pozorováním, dolování dat/znalostí)

Příklad:

Z předchozích zkušeností, zapsaných jako jednotlivé položky tabulky, která je uvedená na dalším snímku, má bankovní úředník posoudit riziko úvěru pro nového klienta.

	Historie úvěrů	Dluh	Ručení	Příjem	Risk úvěru
1	špatná	vysoký	žádné	< 15	vysoký
2	neznámá	vysoký	žádné	15 – 35	vysoký
3	neznámá	nízký	žádné	15 – 35	přiměřený
4	neznámá	nízký	žádné	< 15	vysoký
5	neznámá	nízký	žádné	> 35	nízký
6	neznámá	nízký	adekvátní	> 35	nízký
7	špatná	nízký	žádné	< 15	vysoký
8	špatná	nízký	adekvátní	> 35	přiměřený
9	dobrá	nízký	žádné	> 35	nízký
10	dobrá	vysoký	adekvátní	> 35	nízký
11	dobrá	vysoký	žádné	< 15	vysoký
12	dobrá	vysoký	žádné	15 – 35	přiměřený
13	dobrá	vysoký	žádné	> 35	nízký
14	špatná	vysoký	žádné	15 – 35	vysoký

Tabulka obsahuje pouze necelou polovinu možných položek, jejichž počet je dán součinem všech možných hodnot jednotlivých podmínkových atributů:

$$m = \prod_{i=1}^{k} h_k = 3 \cdot 2 \cdot 2 \cdot 3 = 36$$

Na následujícím snímku je uvedena část úplné tabulky.

	Historie úvěrů	Dluh	Ručení	Příjem	Risk úvěru
1	špatná	vysoký	žádné	< 15	vysoký
2	špatná	vysoký	žádné	15 – 35	vysoký
3	špatná	vysoký	žádné	> 35	?
4	špatná	vysoký	adekvátní	< 15	?
5	špatná	vysoký	adekvátní	15 – 35	?
6	špatná	vysoký	adekvátní	> 35	?
7	špatná	nízký	žádné	< 15	vysoký
8	špatná	nízký	žádné	15 – 35	?
9	špatná	nízký	žádné	> 35	?
10	špatná	nízký	adekvátní	> 35	přiměřený
11	špatná	nízký	adekvátní	< 15	?
12	špatná	nízký	adekvátní	15 – 35	?
13	neznámá	vysoký	žádné	< 15	?
14	…..	…..	…..	…..	?

Počet všech možných „mapovacích“ funkcí je dán obecně výrazem

$$N = h^m$$

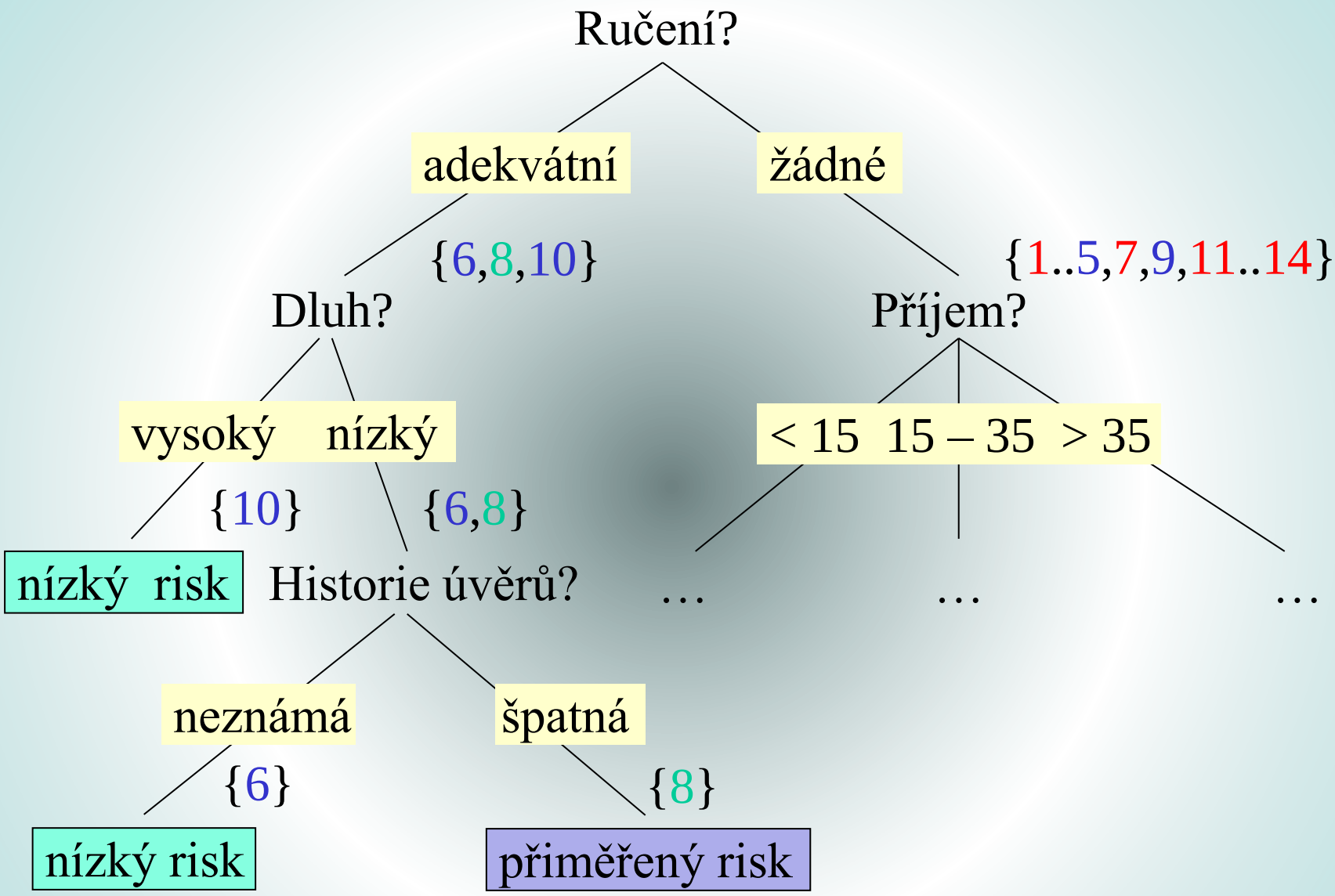
kde *h* značí počet hodnot rozhodovacího atributu, tj. pro uváděný příklad

$$N = 3^{36} \cong 1.5 \cdot 10^{17}$$

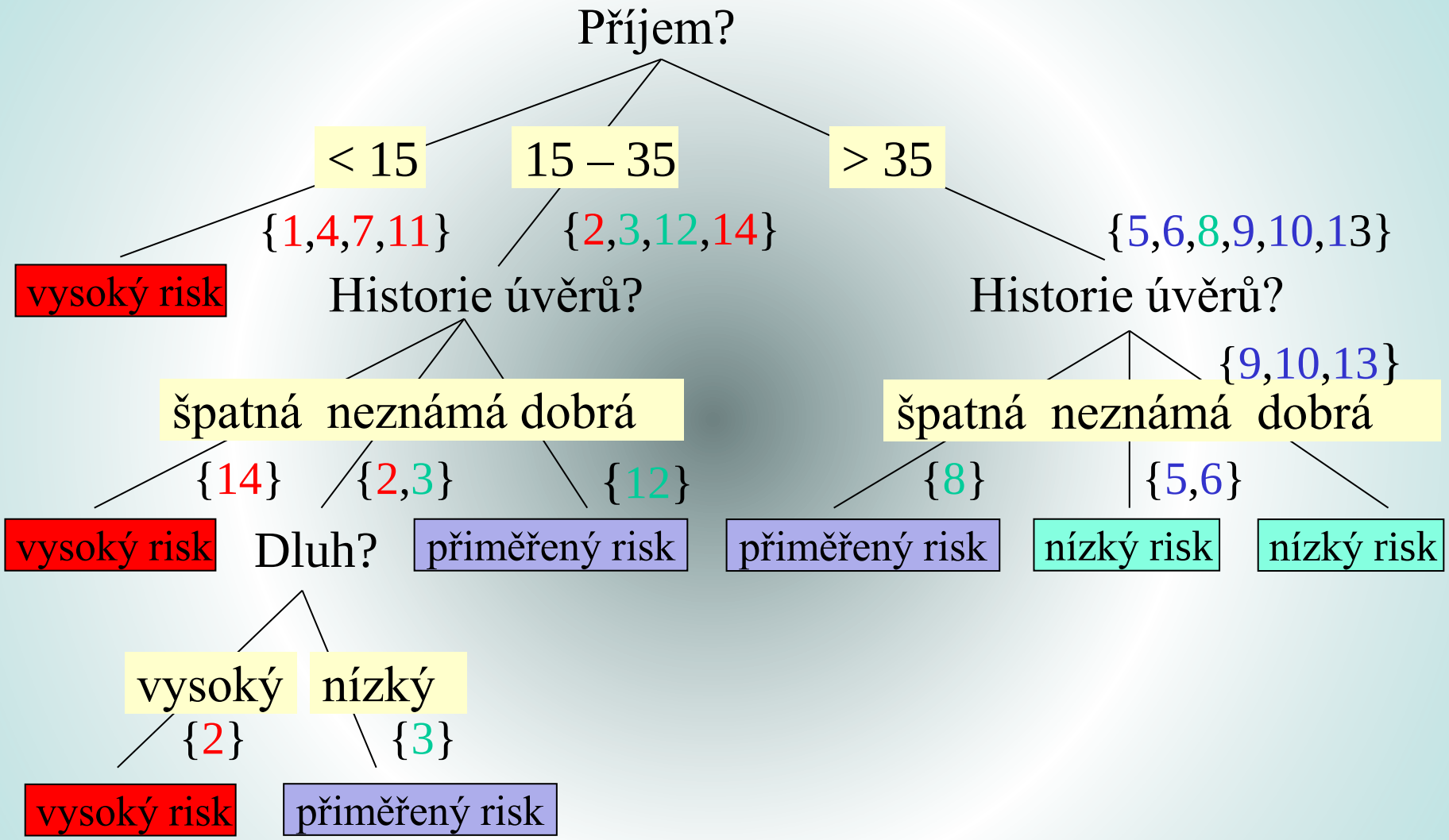
Algoritmus Induce-tree

Vstupní parametry: množina příkladů MP, množina vlastností MV

- Patří-li všechny prvky množiny příkladů MP do stejné třídy, vraťte listový uzel označený touto třídou, jinak pokračujte.
- Je-li množina vlastností MV prázdná, vraťte listový uzel označený disjunkcí všech tříd, do kterých patří prvky v množině příkladů MP, jinak pokračujte.
- Vyberte vlastnost V_i, odstraňte ji z množiny vlastností MV a učiňte ji kořenem aktuálního stromu (nechť MV_{-i} je množina vlastností MV bez vlastnosti V_i).
- Pro každou hodnotu H_j vybrané vlastnosti V_i:
  - Vytvořte novou větev stromu označenou hodnotou H_j (nechť podmnožina příkladů MP_{vihj} je množina všech prvků množiny příkladů MP, které mají hodnotu H_j vlastnosti V_i),
  - volejte rekurzivně Induce-tree s parametry MP_{vihj} a MV_{-i},
  - připojte vrácený strom/uzel k této větvi.


Ručení?
adekvátní
žádné
{6,8,10}
{1..5,7,9,11..14}
Dluh?
Příjem?
vysoký
nízký
< 15
15 – 35
> 35
{10}
{6,8}
nízký risk
Historie úvěrů?
…
…
…
neznámá
špatná
{6}
{8}
nízký risk
přiměřený risk

Optimální rozhodovací strom:



Algoritmus ID3

Entropie (míra neurčitosti) zprávy M s n možnými odpověďmi $\{m_1, m_2, \ldots, m_n\}$, jejichž pravděpodobnosti výskytu jsou označeny jako $p(m_i)$, je dána výrazem:

$$E(M) = \sum_{i=1}^{n} - p(m_i) \log_2 p(m_i) \qquad [bit]$$

Je zřejmé, že „příspěvek“ k entropii zprávy M odpovědí s nulovou pravděpodobností ($p(m_j) = 0$) je také nulový, tj. že

$$\lim_{p(m_j) \to 0} p(m_j) \log_2 p(m_j) = 0$$

a že stejným „příspěvkem“ k entropii zprávy M přispívá i odpověď jistá ($p(m_k) = 1$)

$$p(m_k) \log_2 p(m_k) = 1 \cdot \log_2 1 = 0$$

Entropie je maximální pro rovnoměrné rozložení pravděpodobností možných odpovědí

$$p(m_i) = \frac{1}{n}, \quad i = 1 \ldots n$$

$$E(M) = -\sum_{i=1}^{n} \frac{1}{n} \log_2 \frac{1}{n} = -\log_2 \frac{1}{n} = \log_2 n$$

a minimální (nulová) pro jedinou, předem známou odpověď m_k

$$p(m_k) = 1$$

$$p(m_i) = 0 \quad pro \quad i \# k$$

$$E(M) = -\sum_{i=1, i\#k}^{n} p(m_i) \log_2 p(m_i) - p(m_k) \log_2 p(m_k) = 0$$

Pro předchozí příklad (tabulku) zřejmě platí:

p(vysoký_risk) = 6/14
p(přiměřený_risk) = 3/14
p(nízký_risk) = 5/14

$$E(tabulka) = -\frac{6}{14}\log_2\left(\frac{6}{14}\right) - \frac{3}{14}\log_2\left(\frac{3}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 1.531$$

Vlastnost V, která má k hodnot, rozděluje množinu trénovacích příkladů c do k podmnožin, z nichž každá obsahuje c_i příkladů. Informace potřebná k dokončení stromu, bude-li jeho kořenem vlastnost V (tj. entropie stromu s větvemi podle hodnot vlastnosti V), je dána vztahem

$$E(V) = \sum_{i=1}^{k} \frac{c_i}{c} E(c_i)$$

a odpovídající informační zisk pak vztahem

$$zisk(V) = E(c) - E(V)$$

Pro V = *prijem* (<15, 15-35, >35):

$c1$= {1,4,7,11}, $c2$ = {2,3,12,14}, $c3$ = {5,6,8,9,10,13}

$$E(prijem) = \frac{4}{14}\left(-\frac{4}{4}\log_2\left(\frac{4}{4}\right)\right) + \frac{4}{14}\left(-2\frac{2}{4}\log_2\left(\frac{2}{4}\right)\right) +$$

$$+\frac{6}{14}\left(-\frac{5}{6}\log_2\left(\frac{5}{6}\right) - \frac{1}{6}\log_2\left(\frac{1}{6}\right)\right) = 0.564$$

a tedy

$$zisk(prijem) = E(tabulka) - E(prijem) =$$

$$= 1.531 - 0.565 = 0.966$$

Pro V = *historie úvěrů* (neznámá, špatná, dobrá):

$c1$= {2,3,4, 5,6}, $c2$ = {1,7, 8,14}, $c3$ = {9,10,11,12,13}

$$E(historie\_uveru) = \frac{5}{14}\left(-2\frac{2}{5}\log_2\left(\frac{2}{5}\right)-\frac{1}{5}\log_2\left(\frac{1}{5}\right)\right)+$$
$$+\frac{4}{14}\left(-\frac{3}{4}\log_2\left(\frac{3}{4}\right)-\frac{1}{4}\log_2\left(\frac{1}{4}\right)\right)+$$
$$+\frac{5}{14}\left(-\frac{3}{5}\log_2\left(\frac{3}{5}\right)-2\frac{1}{5}\log_2\left(\frac{1}{5}\right)\right)=1.265$$

$$zisk(historie\_uveru) = E(tabulka)-E(historie\_uveru) =$$
$$=1.531-1.265=0.266$$

Informační zisk pro zbývající vlastnosti:

$$zisk\left(dluh\right) = 0.063$$
$$zisk\left(ruceni\right) = 0.206$$

Nejvyšší informační zisk má rozvětvení podle hodnot vlastnosti *prijem*.

Pro větev *prijem* < 15 (c = {1,4,7,11} ⇒ hotovo ($E(c)$ = 0))

Pro větev *prijem* = 15 – 35 (c = {2,3,12,14} ⇒ $E(c)$ = 1):

V = *historie_uveru* (neznámá, dobrá, špatná):

c_1= {2,3}, c_2 = {12}, c_3 = {14}, *zisk* = 0.500

V = *dluh* (vysoký, nízký)

c_1= {2,12,14}, c_2 = {3} *zisk* = 0.311

V = *ruceni* (adekvátní, žádné):

c_1= {}, c_2 = {2,3,12,14} *zisk* = 0

Nejvyšší informační zisk má rozvětvení podle hodnot vlastnosti *historie_uveru*.

Pro větev *prijem* > 35 ($c = \{5,6,8,9,10,13\} \Rightarrow E(c) = 0.65$):

V = *historie_uveru* (neznámá, dobrá, špatná):

$c_1 = \{5,6\}$, $c_2 = \{9,10,13\}$, $c_3 = \{8\}$, *zisk* = 0.650

V = *dluh* (vysoký, nízký)

$c_1 = \{10,13\}$, $c_2 = \{5,6,8,9\}$ *zisk* = 0.109

V = *ruceni* (adekvátní, žádné):

$c_1 = \{6,8,10\}$, $c_2 = \{5,9,13\}$ *zisk* = 0.191

Nejvyšší informační zisk má opět rozvětvení podle hodnot vlastnosti *historie_uveru*.

Pro větev *prijem* = 15 – 35, *historie_uveru* = špatná

(c = {14} $\Rightarrow$ hotovo ($E(c)$ = 0)

Pro větev *prijem* = 15 – 35, *historie_uveru* = dobrá

(c = {12} $\Rightarrow$ hotovo ($E(c)$ = 0)

Pro větev *prijem* = 15 – 35, *historie_uveru* = neznámá

(c = {2,3} $\Rightarrow$ $E(c)$ = 1):

V = *dluh* (vysoký, nízký)

c_1= {2}, c_2 = {3} *zisk* = 1

V = *ruceni* (adekvátní, žádné):

c_1= {}, c_2 = {2,3} *zisk* = 0

Nejvyšší informační zisk má rozvětvení podle hodnot vlastnosti *dluh*.

Pro větev *prijem* > 35, *historie_uveru* = špatná

(c = {8} $\Rightarrow$ hotovo ($E(c)$ = 0))

Pro větev *prijem* > 35, *historie_uveru* = dobrá

(c = {9,10,13} $\Rightarrow$ hotovo ($E(c)$ = 0))

Pro větev *prijem* > 35, *historie_uveru* = neznámá

(c = {5,6} $\Rightarrow$ hotovo ($E(c)$ = 0))

Pro větev *prijem* = 15 – 35, *historie_uveru* = neznámá, *dluh* = vysoký

(c = {2} $\Rightarrow$ hotovo ($E(c)$ = 0))

Pro větev *prijem* = 15 – 35, *historie_uveru* = neznámá, *dluh* = nízký

(c = {3} $\Rightarrow$ hotovo ($E(c)$ = 0))

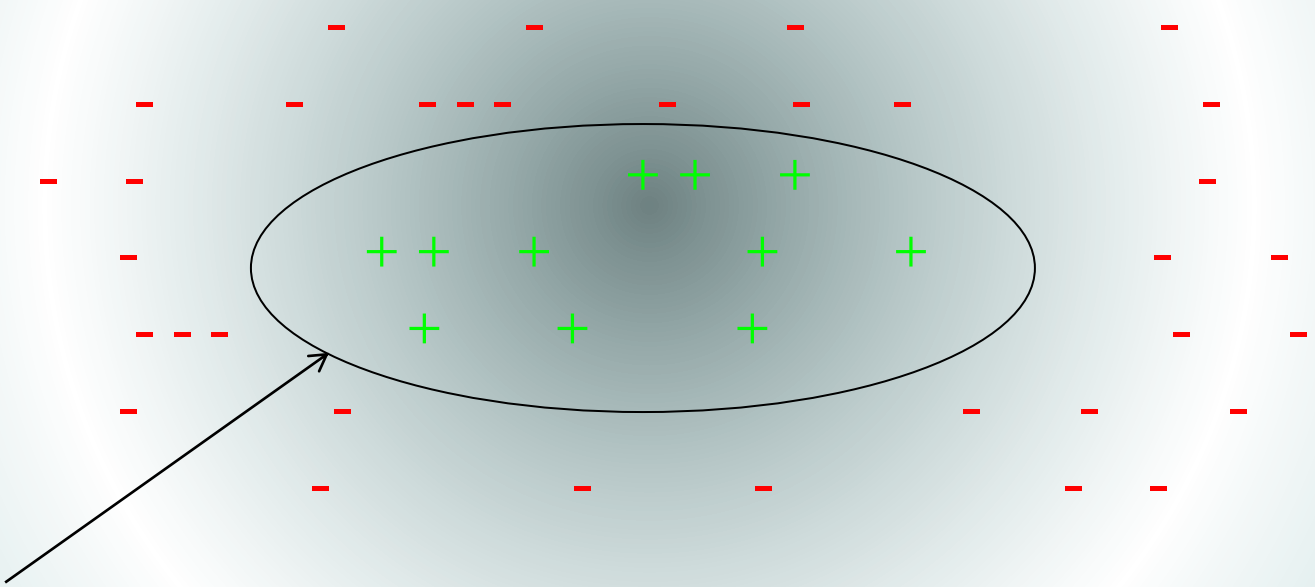
Vytvořený strom lze snadno převést na pravidla:

1. Má-li žadatel příjem nižší než 15 tis. Kč, je risk úvěru vysoký.
2. Má-li žadatel příjem mezi 15 až 35 tis. Kč, pak
   a) je-li historie splácení jeho úvěrů špatná, je risk úvěru vysoký.
   b) je-li historie splácení jeho úvěrů dobrá, je risk úvěru přiměřený.
   c) je-li historie splácení jeho úvěrů neznámá, pak
      - je-li jeho dluh vysoký, je risk úvěru vysoký.
      - je-li jeho dluh nízký, je risk úvěru přiměřený.
3. Má-li žadatel příjem vyšší než 35 tis. Kč, pak
   a) je-li historie splácení jeho úvěrů špatná, je risk úvěru přiměřený.
   b) je-li historie splácení jeho úvěrů dobrá, je risk úvěru nízký.
   c) je-li historie splácení jeho úvěrů neznámá, je risk úvěru nízký.

Hledání prostoru verzí

Podobně jako u rozhodovacích stromů, jde opět o učení na základě příkladů.

Prostor hypotéz/pojmů (kladné příklady + , záporné příklady -):



Prostor verzí: Množina hypotéz, které akceptují všechny pozitivní příklady a které vylučují záporné příklady.

Příklad: Uvažujme pojmy se třemi atributy (velikost, barva, tvar), které mohou nabývat hodnot:

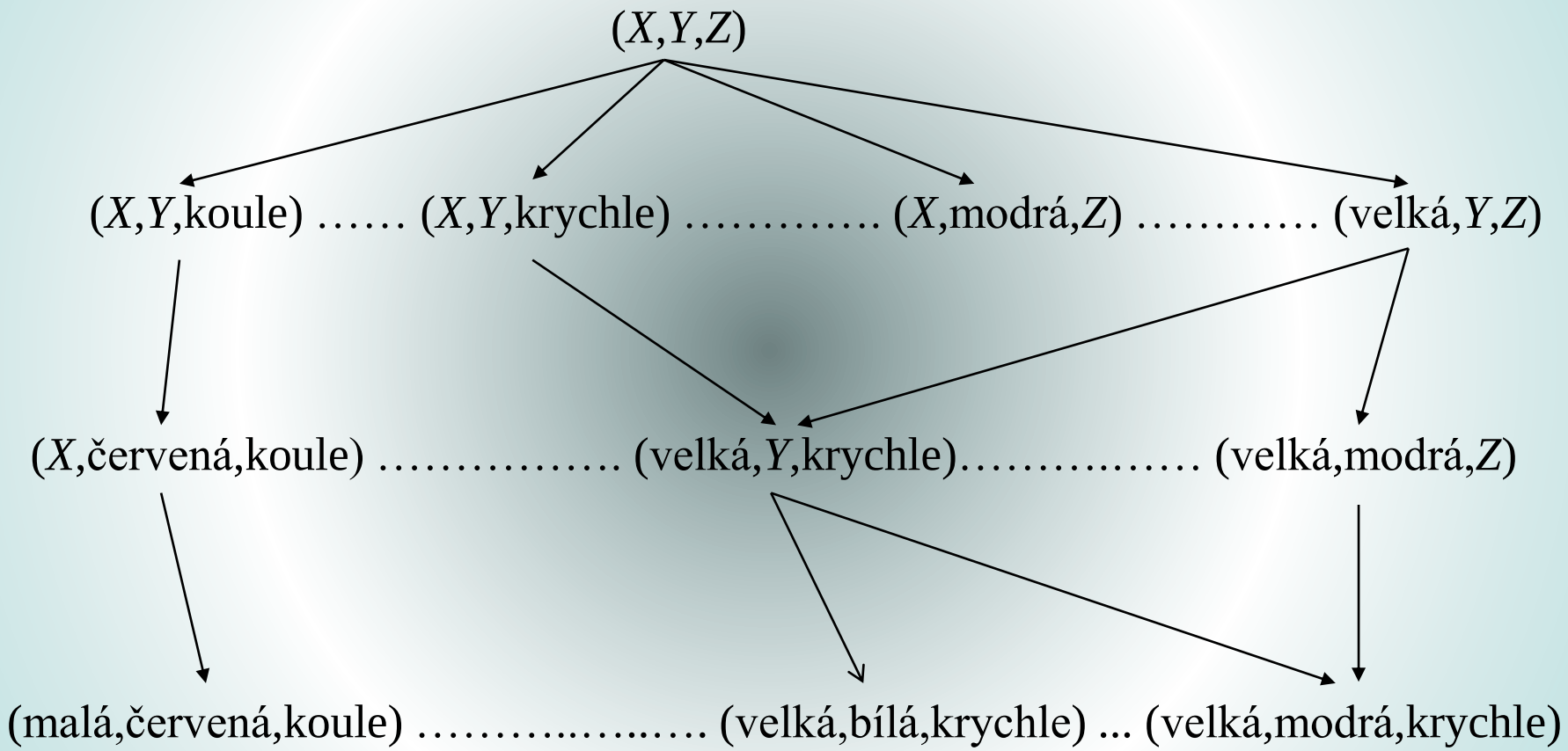
Velikost = {malá,velká}
Barva = {červená,bílá,modrá}
Tvar = {koule,kvádr,krychle}

Prostor hypotéz obsahuje v tomto případě celkem 18 specifických pojmů (každý atribut má přiřazenu konkrétní hodnotu) a 30 obecnějších hypotéz (kdy hodnotou některého atributu je proměnná, resp. kdy hodnotou některých atributů jsou proměnné) – viz následující snímek.

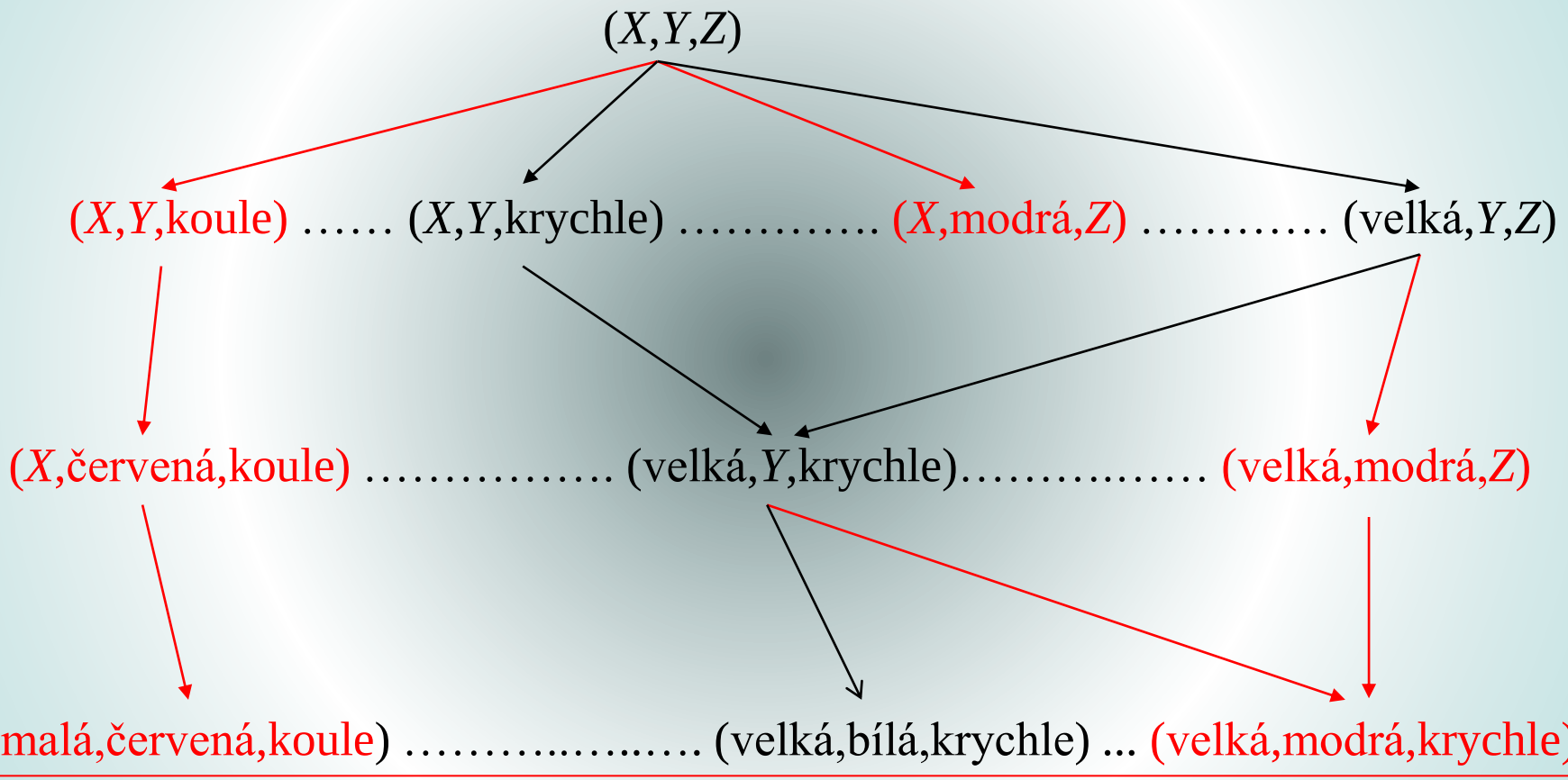
Při hledání prostoru verzí se pak používají dvě operace:

- Zobecňování (nahrazení konstanty proměnnou)
- Specializace (nahrazení proměnné konstantou)

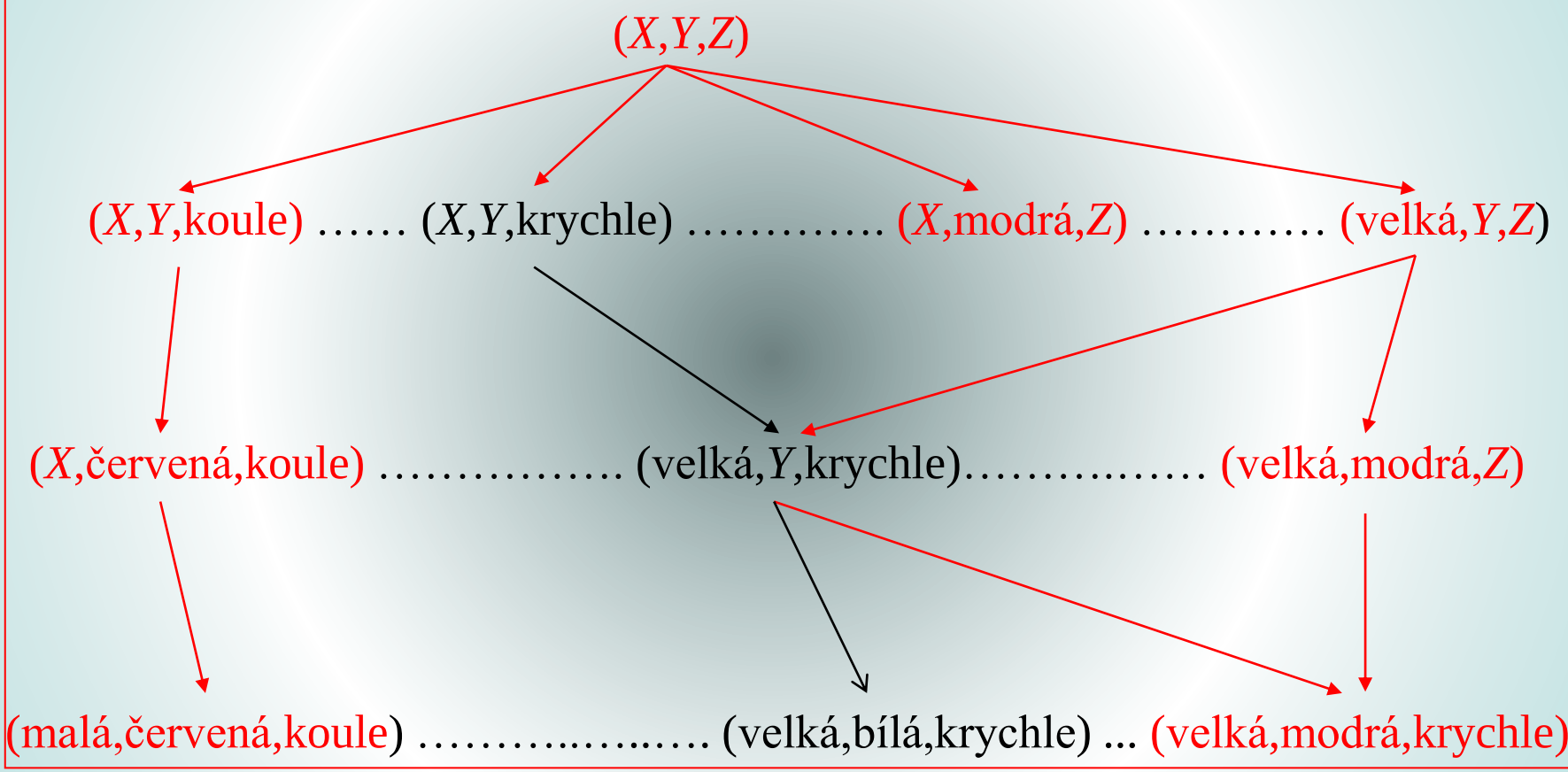
Prostor všech možných hypotéz pro daný příklad (od nejobecnějšího po nejvíce specifické pojmy):



Po pozitivním příkladu, například (velká,bílá,krychle) pak prostor hypotéz/verzí musí být redukován na:



a po následujícím negativním příkladu, například (velká,bílá,koule) pak musí být tento prostor dále redukován na:



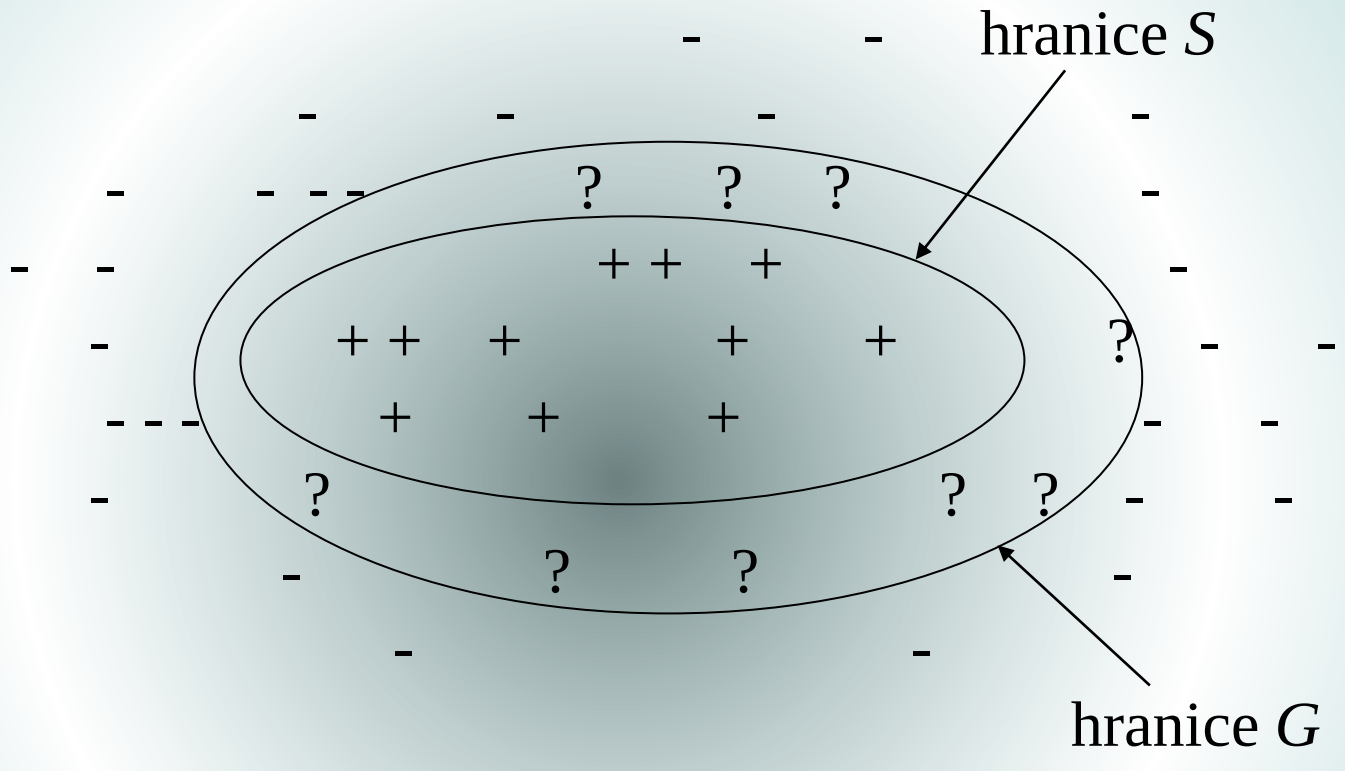
K požadované redukci prostoru verzí lze použit několik přístupů, nejznámější z nich je tzv. algoritmus eliminace kandidátů.

Algoritmus Candidate eliminations

- Vytvořte dvě množiny *G* (General) a *S* (Specific), do množiny *G* vložte nejobecnější hypotézu a do množiny *S* první kladný příklad.
- Pro každý další příklad *p* z trénovací množiny:
  - je-li *p* kladným příkladem, pak
    - z množiny *G* odstraňte všechny hypotézy, které nelze unifikovat s příkladem *p*,
    - všechny hypotézy v množně *S*, které nelze unifikovat s příkladem *p*, nahraďte jejich nejvíce specifickými zobecněními, které lze unifikovat s příkladem *p*,
    - odstraňte z množiny S všechny hypotézy, které jsou:
      - obecnější než jiné hypotézy v této množině,
      - obecnější než nějaké hypotézy v množině G.

- je-li *p* záporným příkladem pak
  - pokud lze příklad *p* unifikovat s nějakou hypotézou v množině *S*, pak tuto hypotézu z množiny *S* odstraňte,
  - každou hypotézu v množině *G*, kterou lze unifikovat s příkladem *p*, nahraďte jejími nejvíce zobecněnými specializacemi, které nelze unifikovat s příkladem *p*,
  - odstraňte z množiny *G* všechny hypotézy, které jsou:
    - více specifické, než jiné hypotézy v množině *G*,
    - více specifické, než nějaké hypotézy v množině *S*.
- jestliže *G* = *S* a obě množiny přitom obsahují jedinou/stejnou hypotézu, pak algoritmus našel pojem, který je konzistentní se všemi příklady a končí úspěchem (výsledkem učení je právě tento pojem).

Význam *S* a *G*:



Každý pojem, který by byl obecnější než nějaký pojem v *G*, by zahrnoval některé negativní příklady,
každý pojem, který by byl specifičtější než nějaký pojem v *S*, by vylučoval některé pozitivní příklady.

Konkrétní příklad:

Trénovací množina příkladů pro pojem ***míč***
(kladné příklady / záporné příklady):

1. (malá,červená,koule)
2. (malá,modrá,kvádr)
3. (velká,červená,koule)
4. (velká,červená,krychle)
5. (malá,modrá,koule)
6. (malá,bílá,kvádr)

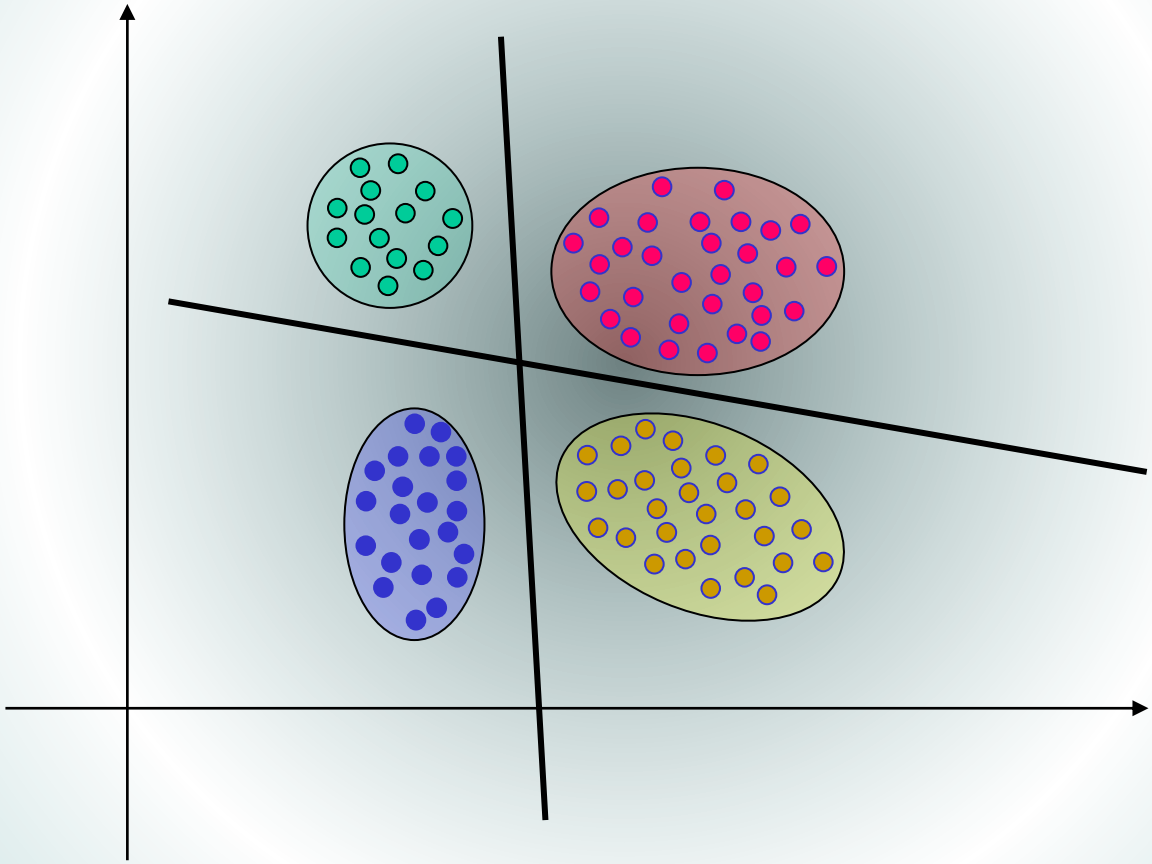
Candidate elimination:

$G = \{(X,Y,Z)\}$

1. (malá,červená,koule) $S = \{(\text{malá,červená,koule})\}$
2. (malá,modrá,kvádr) $G = \{(\text{velká},Y,Z), (X,\text{červená},Z), (X,\text{bílá},Z), (X,Y,\text{koule}), (X,Y,\text{krychle})\}$
3. (velká,červená,koule) $G = \{(X,\text{červená},Z),(X,Y,\text{koule})\}$, $S = \{(X,\text{červená,koule})\}$
4. (velká,červená,krychle) $G = \{(\text{malá,červená},Z), (X,\text{červená,kvádr}), (X,\text{červená,koule}), (X,Y,\text{koule})\} \Rightarrow$ $G = \{(\text{malá,červená},Z), (X,\text{červená,kvádr}), (X,Y,\text{koule})\}$
5. (malá,modrá,koule) $S = \{(X,Y,\text{koule})\}$, $G = \{(X,Y,\text{koule})\}$
6. (malá,bílá,kvádr) $S = \{(X,Y,\text{koule})\}$, $G = \{(X,Y,\text{koule})\}$

$S = G$ a obě množiny obsahují jedinou hypotézu $(X,Y,\text{koule}) \Rightarrow$ míč se nepozná podle velikosti, ani barvy, ale tvarem musí být kulatý.

Učení bez učitele (unsupervised learning)



Shlukování (clustering)

Existuje řada různých algoritmů, nejznámější z nich je algoritmus ***k – means clustering*** (shlukování do *k* shluků).

Algoritmus je založen na předpokladu, že *n*-rozměrné vektory $\vec{x} = (x_1, x_2, \ldots, x_n)$, resp. koncové body těchto vektorů, tvoří v *n*-rozměrném prostoru shluky a že každý shluk *i* je reprezentován prototypem (vektorem „těžiště“ shluku) $\vec{w}_i$.

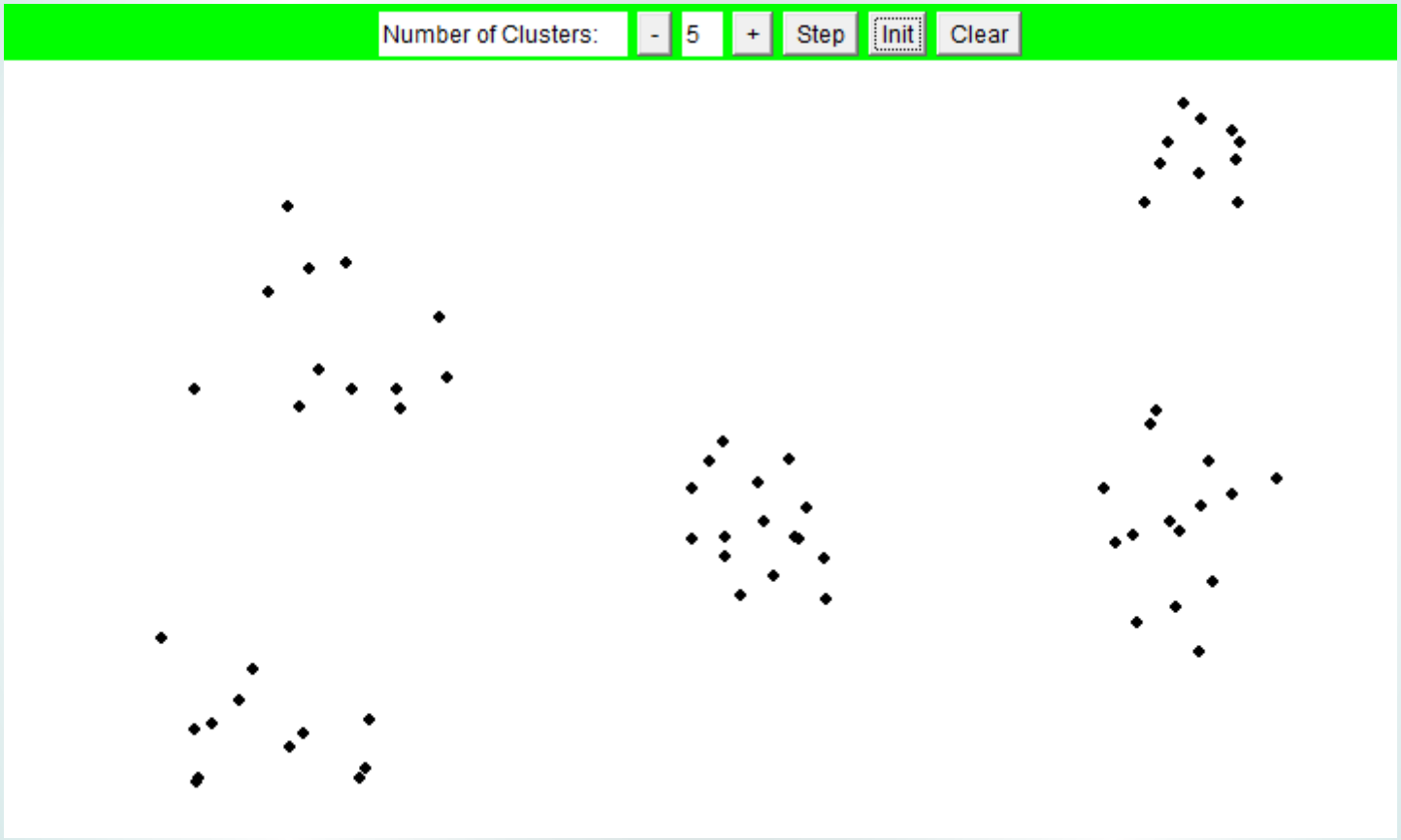
Algoritmus dále předpokládá, že do *k* shluků má rozdělit *p* vektorů z trénovací množiny $T = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_P\}$.

Pozn.: Na výsledek učení má vliv použitá metrika – Euklidovská, Hammingova, atd.

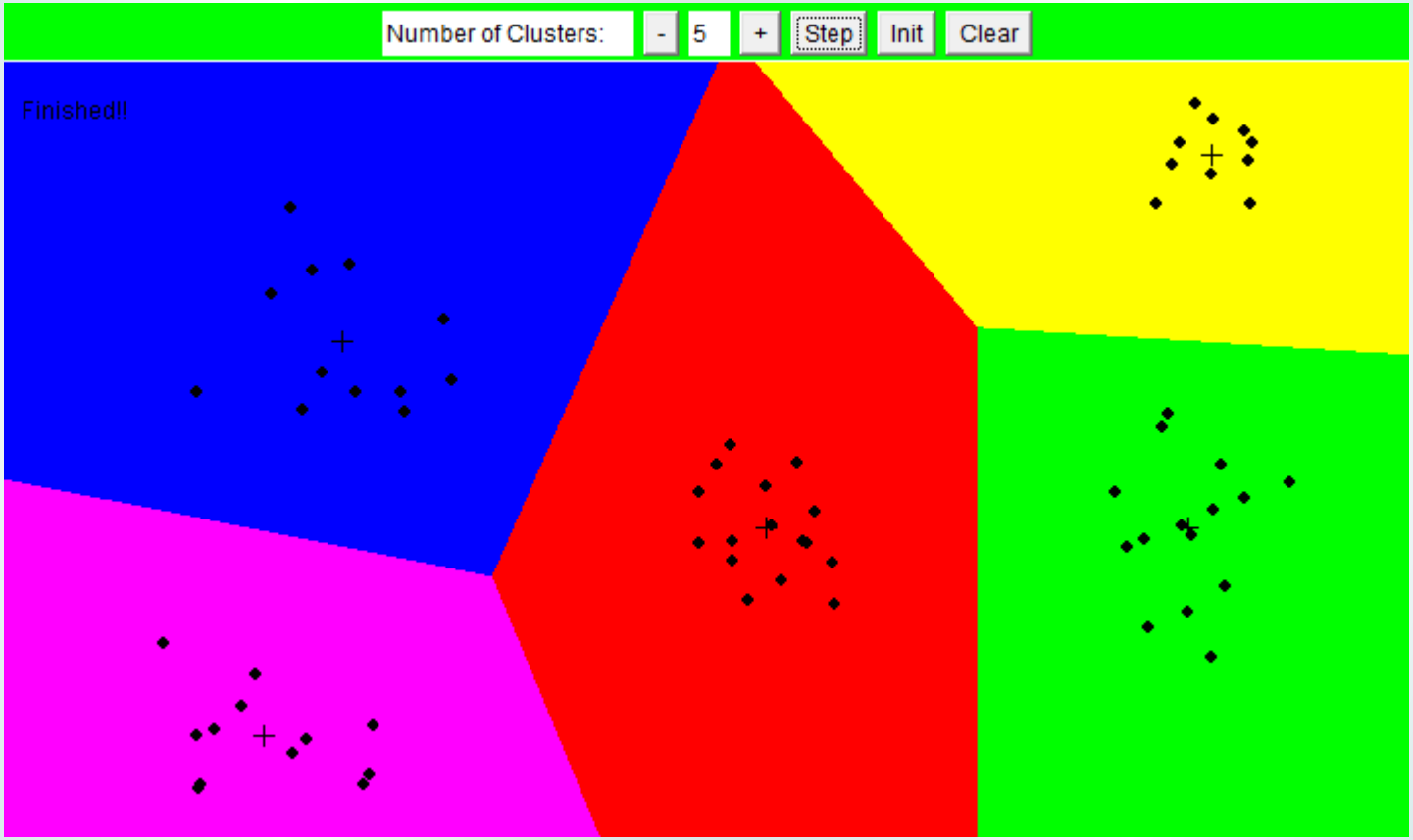
Algoritmus k – means clustering

1. Inicializujte k prototypů $\vec{w}_j$ (použijte například náhodně vybrané, ale různé vektory $\vec{x}_p$ z trénovací množiny P vektorů, tj. $\vec{w}_j = \vec{x}_p, \quad j \in \langle 1,k \rangle, \quad p \in \langle 1,P \rangle$).
2. Každý vektor $\vec{x}_p$ z trénovací množiny přiřaďte do shluku C_j, $j \in <1,k>$, jehož prototyp $\vec{w}_j$ má od vektoru $\vec{x}_p$ nejmenší vzdálenost, tj.
$$\left|\vec{x}_p - \vec{w}_j\right| \leq \left|\vec{x}_p - \vec{w}_i\right| \qquad i \in \langle 1,k \rangle$$
3. Pro každý shluk C_j $j \in <1,k>$ přepočítejte prototyp $\vec{w}_j$ tak, aby byl těžištěm koncových bodů všech vektorů, které jsou k tomuto shluku právě přiřazené (nechť n_j je počet těchto vektorů):
$$\vec{w}_j = \frac{\sum_{\vec{x}_i \in C_j} \vec{x}_i}{n_j}$$
5. Pokud byl některý vektor přeřazen k jinému shluku, vraťte se na bod 2, jinak činnost algoritmu ukončete.

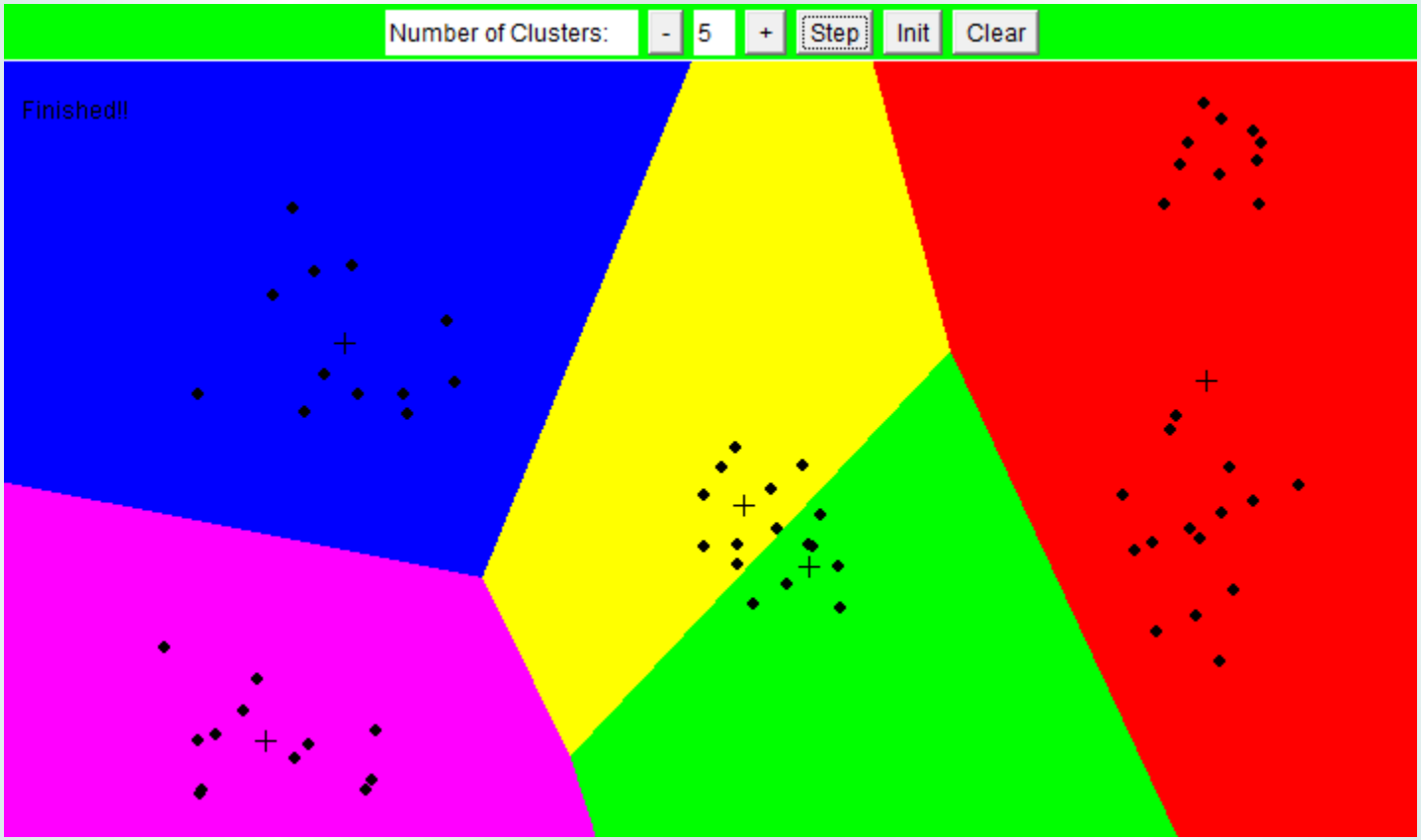
Příklad:



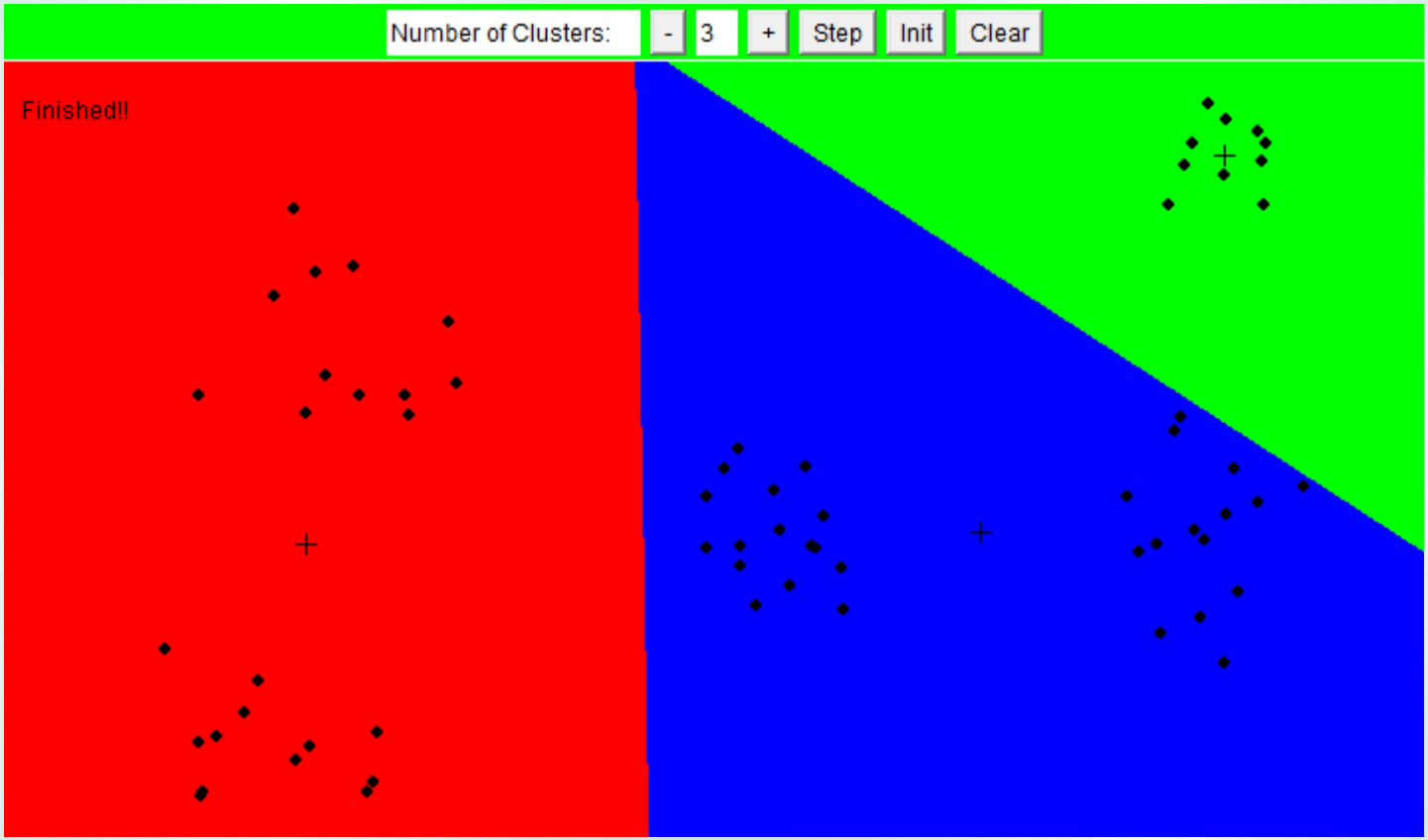
Výsledek získaný metodou k-means:



Nevhodně zvolené počáteční prototypy mohou vést k jiným a ne zcela očekávaným výsledkům:



Možné rozdělení do menšího požadovaného počtu shluků:



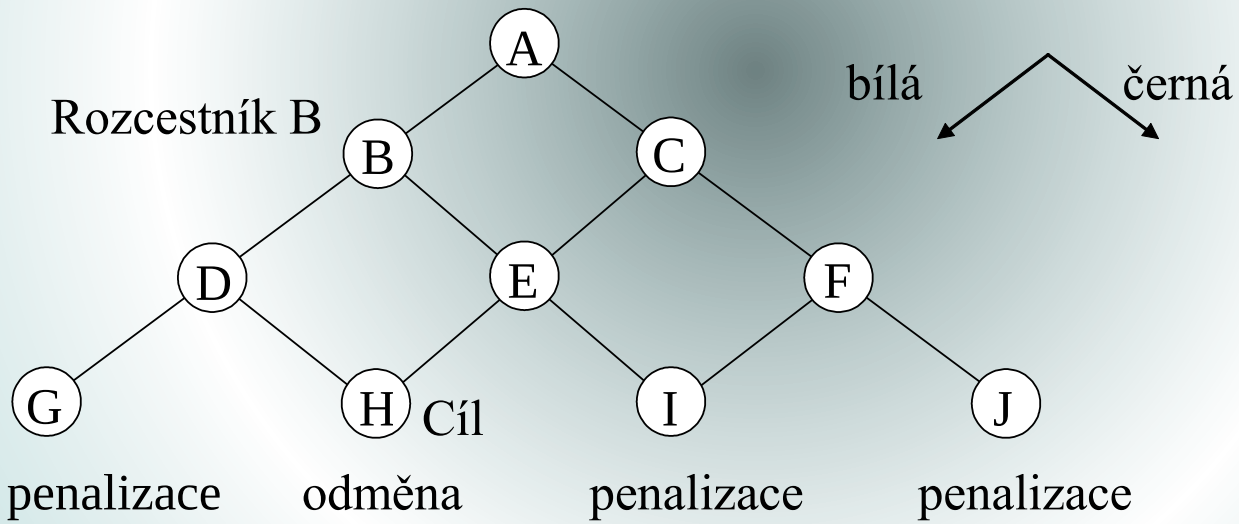
Posilované/motivované učení (reinforcement learning)

- Policy-only learning
- TD – learning (Temporal Diference learning)
- Q – learning (Quality learning)

Agent ohodnocuje své akce na základě penalizací či odměn v koncových stavech, resp. na základě svých ocenění stavů/akcí získaných vlastními předchozími zkušenostmi.

Policy-only learning

Uvažujme nejprve jednoduchý graf, ve kterém z každého uzlu vedou maximálně dvě cesty a hledejme (optimální) cestu z počátečního do cílového uzlu. Každý rozcestník (signpost) obsahuje schránku/box s černými a bílými kameny, které se využívají k určení směru cesty:



- Na začátku učení obsahuje schránka každého rozcestníku stejný počet černých a bílých kamenů ($N_{bílé} = N_{černé}$).
- V průběhu učení jsou pravděpodobnosti pohybu agenta na každém rozcestníku dány výrazy:

$$p_{vlevo} = \frac{N_{bílé}}{N_{bílé} + N_{černé}}$$

$$p_{vpravo} = \frac{N_{černé}}{N_{bílé} + N_{černé}}$$

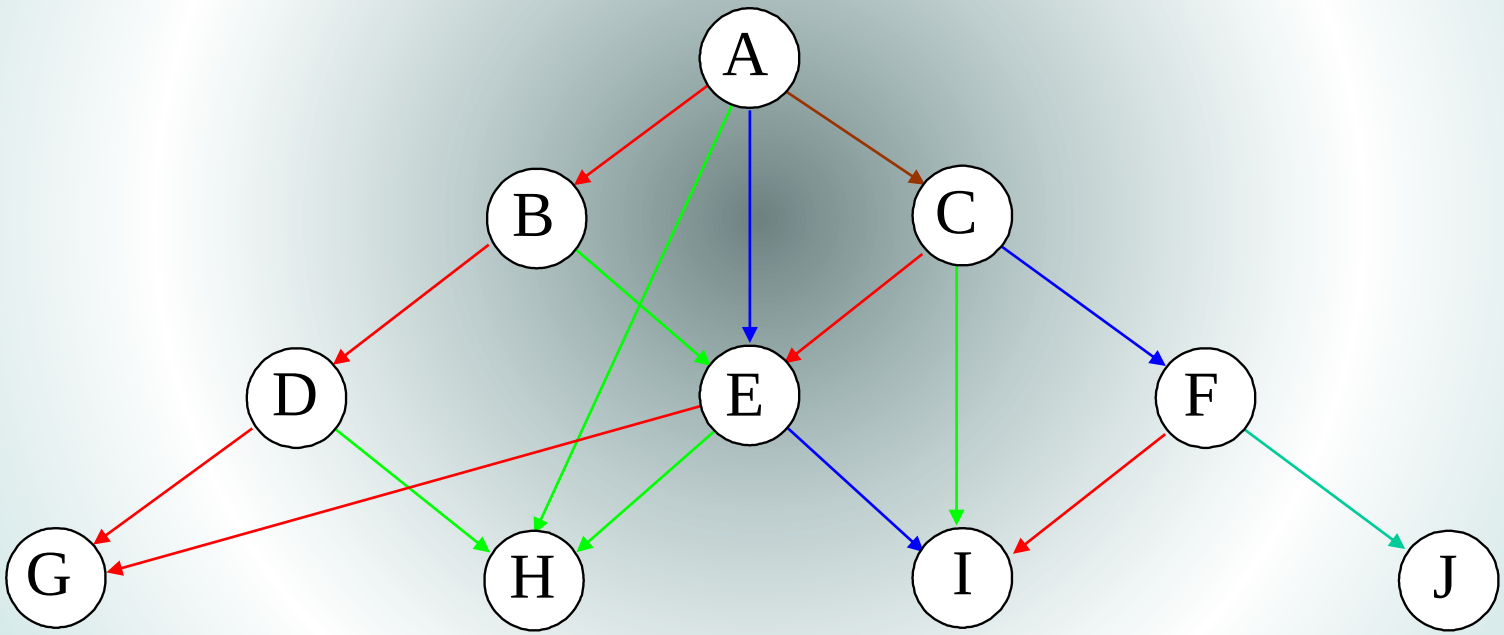
- Vyhodnocení cesty a odměna/penalizace:

Cesta vedla do cíle H $\Rightarrow$ odměna (přidání kamenů odpovídajících barev do boxů všech rozcestníků na příslušné cestě)

Cesta vedla do G, I, J $\Rightarrow$ penalizace (odebrání kamenů odpovídajících barev z boxů všech rozcestníků na příslušné cestě)

Pravděpodobnost výběru optimální cesty do H se zvyšuje, pravděpodobnost výběru ostatních cest se snižuje !!!

Rozšíření popsaného principu posilovaného učení na problémy s více možnými cestami je snadné – použijí se pouze různobarevné kameny. Barvy mohou být zcela libovolné, musí však být jednoznačně přiřazeny k některé výchozí hraně každého uzlu):



Metoda Policy-only je sice jednoduchá, ale její použití může vést k několika problémům:

- V případě obecného grafu, kdy se lze vracet do již dříve vyšetřovaných uzlů, je nutné těmto „návratům“ zabránit.
- Ohodnocení rozcestníků se mění až při dosažení koncových stavů (těmito jsou obecně i konce slepých cest).
- Stejnými vahami se ohodnocují všechny akce provedené na jedné cestě, přičemž správná ohodnocení mohou být značně rozdílná.
- V paměti se musí ukládat úplné cesty, což může být u rozsáhlých úloh problematické.

TD learning

Metoda postupně ohodnocuje během náhodné procházky jednotlivé stavy *s* (**stav *s* je vždy bezprostředním předchůdcem aktuálního stavu *s'***) pomocí vztahu:

$$U^{\pi}(s) = U^{\pi}(s) + \alpha\left(r(s) + \gamma U^{\pi}(s') - U^{\pi}(s)\right),$$

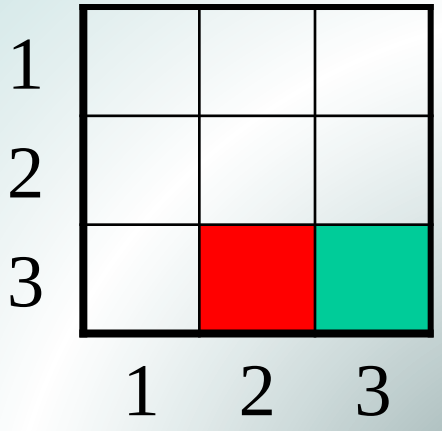
kde značí:

$U^{\pi}(s)$	ohodnocení (*utility*) stavu *s* při použití strategie pohybu π (ze stavu *s* do stavu *s'*).
$r(s)$	odměnu (*reward*) za dosažení stavu *s*.
γ	koeficient určující vliv ohodnocení stavu *s'* na ohodnocení předcházejícího stavu *s*.
α	koeficient učení, jehož hodnota může klesat s počtem průchodů stavem *s*.

Algoritmus TD learning

1. Zvolte hodnoty koeficientů α a γ ($0 < \alpha \leq 1$; $0 < \gamma \leq 1$) a vynulujte ohodnocení $U^{\pi}(s)$ všech stavů. Dále vynulujte počítadlo procházek $p = 0$ a nastavte jejich maximální počet p_{max}. Nastavte *start* $\rightarrow s$.
2. Generujte nový stav *s'* s použitím strategie π.
3. Je-li stav *s'* cílovým stavem stavem, pak mu přiřaďte hodnotu $U^{\pi}(s') = r(s')$.
4. Vypočítejte novou hodnotu stavu pomocí vztahu:
   $$U^{\pi}(s) = U^{\pi}(s) + \alpha\left(r(s) + \gamma U^{\pi}(s') - U^{\pi}(s)\right)$$
5. Je-li stav *s'* cílovým stavem, pak $p+1 \rightarrow p$, *start* $\rightarrow s$, jinak $s' \rightarrow s$.
6. Je-li $p < p_{max}$, pak se vraťte na bod 2.

Příklad:



Nechť $r(3,2) = -1$ (penalizace v nežádoucím cíli), $r(3,3) = 1$ (odměna v žádoucím cíli) a všechna ostatní $r(i,j) = 0$.

Jednotlivé náhodné procházky mohou startovat odkudkoliv, pro jednoduchost předpokládejme, že všechny budou startovat ze stavu (1,1). Dále nechť $\gamma = 0.9$ a $\alpha = 0.1$.

Uvažujme následujících pět náhodných procházek:

1. $(1,1) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (3,3)$
2. $(1,1) \rightarrow (1,2) \rightarrow (2,2) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (3,3)$
3. $(1,1) \rightarrow (2,1) \rightarrow (1,1) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (3,3)$
4. $(1,1) \rightarrow (2,1) \rightarrow (3,1) \rightarrow (3,2)$
5. $(1,1) \rightarrow (2,1) \rightarrow (2,2) \rightarrow (3,2)$

Nejprve se všechna ohodnocení $U(i,j)$ vynulují.

První náhodná procházka $(1,1) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (3,3)$:

$U(1,1) = U(1,1) + \alpha \cdot (r(1,1) + \gamma \cdot U(1,2) - U(1,1)) = 0$

$U(1,2) = U(1,2) + \alpha \cdot (r(1,2) + \gamma \cdot U(1,3) - U(1,2)) = 0$

$U(1,3) = U(1,3) + \alpha \cdot (r(1,3) + \gamma \cdot U(2,3) - U(1,3)) = 0$

$U(2,3) = U(2,3) + \alpha \cdot (r(2,3) + \gamma \cdot U(3,3) - U(2,3)) =$

$= 0 + 0.1 \cdot (0 + 0.9 \cdot 1 - 0) = 0.09;$ $U(3,3) = r(3,3) = 1$ (cílový stav)

	1	2	3
1	0	0	0
2	0	0	0.09
3	0	0	1

Při druhé náhodné procházce $(1,1) \rightarrow (1,2) \rightarrow (2,2) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (3,3)$ se změní pouze ohodnocení stavů (1,3) a (2,3):

$$U(1,1) = U(1,1) + \alpha \cdot (r(1,1) + \gamma \cdot U(1,2) - U(1,1)) = 0$$

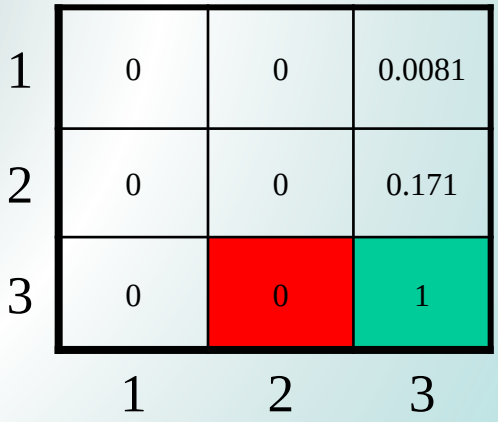
$$U(1,2) = U(1,2) + \alpha \cdot (r(1,2) + \gamma \cdot U(2,2) - U(1,2)) = 0$$

$$U(2,2) = U(2,2) + \alpha \cdot (r(2,2) + \gamma \cdot U(1,2) - U(2,2)) = 0$$

$$U(1,2) = U(1,2) + \alpha \cdot (r(1,2) + \gamma \cdot U(1,3) - U(1,2)) = 0$$

$$U(1,3) = U(1,3) + \alpha \cdot (r(1,3) + \gamma \cdot U(2,3) - U(1,3)) =$$
$$= 0 + 0.1 \cdot (0 + 0.9 \cdot 0.09 - 0) = 0.0081$$

$$U(2,3) = U(2,3) + \alpha \cdot (r(2,3) + \gamma \cdot U(3,3) - U(2,3)) =$$
$$= 0.09 + 0.1 \cdot (0 + 0.9 \cdot 1 - 0.09) = 0.171$$



Při třetí náhodné procházce (1,1) → (2,1) → (1,1) → (1,2) → (1,3) → (2,3) → (3,3) se změní ohodnocení stavů (1,2), (1,3) a (2,3):

$$U(1,1) = U(1,1) + \alpha \cdot (r(1,1) + \gamma \cdot U(2,1) - U(1,1)) = 0$$

$$U(2,1) = U(2,1) + \alpha \cdot (r(2,1) + \gamma \cdot U(1,1) - U(2,1)) = 0$$

$$U(1,1) = U(1,1) + \alpha \cdot (r(1,1) + \gamma \cdot U(1,2) - U(1,1)) = 0$$

$$U(1,2) = U(1,2) + \alpha \cdot (r(1,2) + \gamma \cdot U(1,3) - U(1,2)) =$$
$$= 0 + 0.1 \cdot (0 + 0.9 \cdot 0.0081 - 0) = 0.000729$$

$$U(1,3) = U(1,3) + 0.1 \cdot (r(1,3) + 0.9 \cdot U(2,3) - U(1,3)) =$$
$$= 0.0081 + 0.1 \cdot (0 + 0.9 \cdot 0.171 - 0.0081) =$$
$$= 0.002268$$

$$U(2,3) = U(2,3) + 0.1 \cdot (r(2,3) + 0.9 \cdot U(3,3) - U(2,3)) =$$
$$= 0.171 + 0.1 \cdot (0 + 0.9 \cdot 1 - 0.171) =$$
$$= 0.2439$$

1	0	0.000729	0.002268
2	0	0	0.2439
3	0	0	1
	1	2	3

Při čtvrté náhodné procházce $(1,1) \rightarrow (2,1) \rightarrow (3,1) \rightarrow (3,2)$ se změní ohodnocení stavů (3,2) a (3,1):

$$U(1,1) = U(1,1) + \alpha \cdot (r(1,1) + \gamma \cdot U(2,1) - U(1,1)) = 0$$

$$U(2,1) = U(2,1) + \alpha \cdot (r(2,1) + \gamma \cdot U(3,1) - U(2,1)) = 0$$

$$U(3,2) = r(3,2) = -1;$$

$$U(3,1) = U(3,1) + \alpha \cdot (r(3,1) + \gamma \cdot U(3,2) - U(1,3)) =$$
$$= 0 + 0.1 \cdot (0 + 0.9 \cdot (-1) - 0) = -0.09$$

1	0	0.000729	0.002268
2	0	0	0.2439
3	-0.09	-1	1
	1	2	3

Při páté náhodné procházce $(1,1) \rightarrow (1,2) \rightarrow (2,2) \rightarrow (3,2)$ se změní ohodnocení stavů (1,1), (1,2) a (2,2):

$$U(1,1) = U(1,1) + \alpha \cdot (r(1,1) + \gamma \cdot U(1,2) - U(1,1)) =$$
$$= 0 + 0.1 \cdot (0 + 0.9 \cdot 0.000729 - 0) = 0.00006561$$

$$U(1,2) = U(1,2) + \alpha \cdot (r(1,2) + \gamma \cdot U(2,2) - U(1,2)) =$$
$$= 0.000729 + 0.1 \cdot (0 + 0.9 \cdot 0 - 0.000729) = 0.0006561$$

$$U(2,2) = U(2,2) + \alpha \cdot (r(2,2) + \gamma \cdot U(3,2) - U(2,2)) =$$
$$= 0 + 0.1 \cdot (0 + 0.9 \cdot (-1) - 0) = -0.09$$

1	0,0000651	0.0006561	0.002268
2	0	-0,09	0.2439
3	-0.09	-1	1
	1	2	3

Po naučení se pak přechází z libovolného (necílového) stavu do jeho sousedního stavu, který má nejvyšší hodnotu (ale alespoň stejnou hodnotu jako tento stav).

Problém může nastat v případě, kdy stav s penalizací je snadněji dosažitelný, než stav s odměnou. Pak může dojít k takovému ohodnocení jednotlivých stavů, že z některého stavu není dosažitelný žádný jiný stav s vyšším ohodnocením.

Například předchozí problém po 1000 náhodných procházkách může vést k ohodnocení, kdy z počátečního stavu nelze přejít do žádného jiného stavu:

Pozn.: Tento problém se dá někdy odstranit zvýšením počtu procházek, nebo/a snížením koeficientu α.

1	-0.135254	-0.185655	-0.086196
2	-0.205577	-0.419297	0.231171
3	-0.220843	-1	1
	1	2	3

Pro porovnání stejný příklad, ale s překážkou mezi stavy (1,2) a (2,2). Příklad možného ohodnocení stavů po 1000 náhodných procházkách:

1	-0.097	-0.071	-0.045
2	-0.191	-0.384	0.046
3	-0.379	-1	1
	1	2	3

Žádný problém již nenastává a z libovolného stavu lze přejít přes stavy se zvyšujícím se ohodnocením až do žádoucího cílového stavu (tj. stavu s odměnou).

Q learning

Metoda je podobná metodě TD learning, místo hodnocení stavů však hodnotí akce v těchto stavech. K jejich hodnocení používá vztah:

$$Q(s,a) = Q(s,a) + \alpha\left(r(s) + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$$

kde $Q(s,a)$ označuje ohodnocení akce a provedené ve stavu s a kde výraz

$$\max_{a'} Q(s',a')$$

označuje maximální hodnotu z ohodnocení všech akcí a', které je možné provést ve stavu s'.
Význam ostatních symbolů je stejný jako u metody TD.

Algoritmus Q learning

1. Zvolte hodnoty koeficientů α a γ ($0 < \alpha \leq 1$; $0 < \gamma \leq 1$) a vynulujte ohodnocení $Q(s,a)$ všech akcí a ve všech stavech s. Dále vynulujte počítadlo procházek $p = 0$ a nastavte jejich maximální počet p_{max}. Nastavte $start \rightarrow s$.
2. Vyberte akci a, která povede k přechodu ze stavu s do stavu s'.
3. Je-li stav s' cílovým stavem, pak $Q(s,a) = r(s')$.
4. Jinak vypočítejte novou hodnotu akce a ve stavu s pomocí vztahu:
$$Q(s,a) = Q(s,a) + \alpha\left(r(s) + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$$
5. Je-li stav s' cílovým stavem, pak $p+1 \rightarrow p$, $start \rightarrow s$, jinak $s' \rightarrow s$.
6. Je-li $p < p_{max}$, pak se vraťte na bod 2.

Dále uvažujme stejný příklad jako u metody TD v němž jsou k označení akcí a pro pohyby nahoru, doprava, dolů a doleva použity symboly L, U, R, D (Left, Up, Right, Down). Pro možnost srovnání předpokládejme stejné procházky a stejné koeficienty α a γ.

Nejprve se všechna ohodnocení $Q(s,a)$ vynulují.

Po první náhodné procházce $(1,1) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (3,3)$ se změní ohodnocení akce $Q((2,3),D)$, která vede do stavu (3,3):

$Q((2,3),D) = r(3,3) = 1$

Po druhé náhodné procházce $(1,1) \rightarrow (1,2) \rightarrow (2,2) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (3,3)$ se změní ohodnocení akce $Q((1,3),D)$:

$Q((1,3),D) = 0 + 0.1 \cdot (0 + 0.9 \cdot 1 - 0) = 0.09$

Po třetí náhodné procházce $(1,1) \rightarrow (2,1) \rightarrow (1,1) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (3,3)$ se změní ohodnocení akcí $Q((1,2),\mathrm{R})$ a $Q((1,3),\mathrm{D})$:

$Q((1,2),\mathrm{R}) = 0 + 0.1 \cdot (0 + 0.9 \cdot 0.09 - 0) = 0.0081$
$Q((1,3),\mathrm{D}) = 0.09 + 0.1 \cdot (0 + 0.9 \cdot 1 - 0.09) = 0.171$

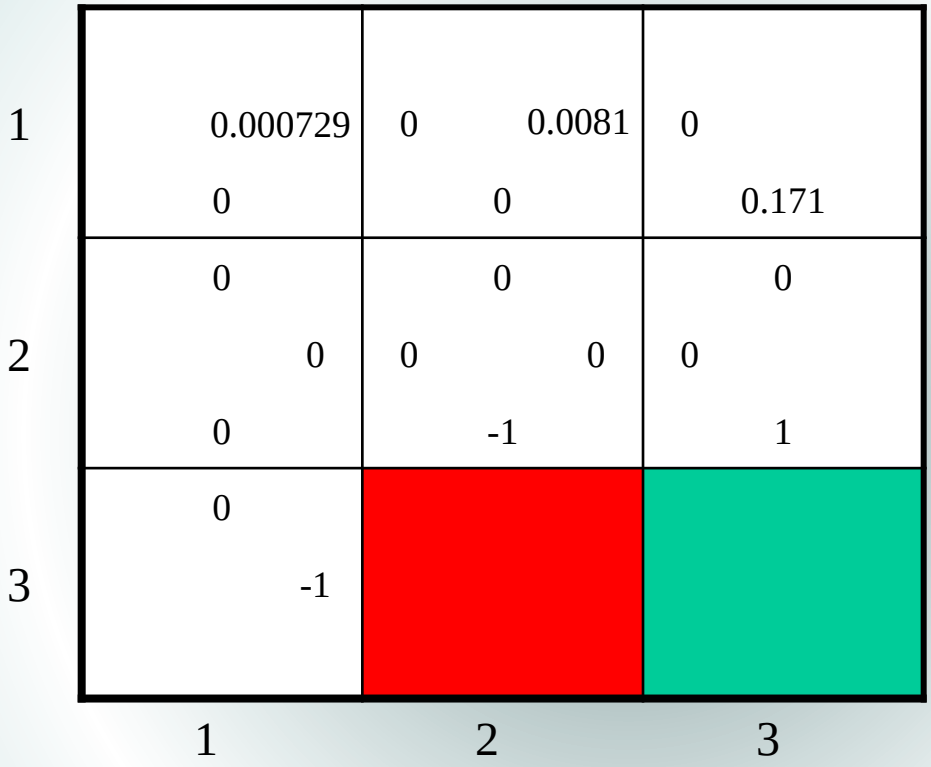
Po čtvrté náhodné procházce $(1,1) \rightarrow (2,1) \rightarrow (3,1) \rightarrow (3,2)$ se změní akce $Q((3,1),\mathrm{R})$, která vede do stavu $(3,2)$:

$Q((3,1),\mathrm{R}) = r(3,2) = -1$

Po páté náhodné procházce $(1,1) \rightarrow (1,2) \rightarrow (2,2) \rightarrow (3,2)$ se změní ohodnocení akcí $Q((1,1),\mathrm{R})$ a $Q((2,2),\mathrm{D})$:

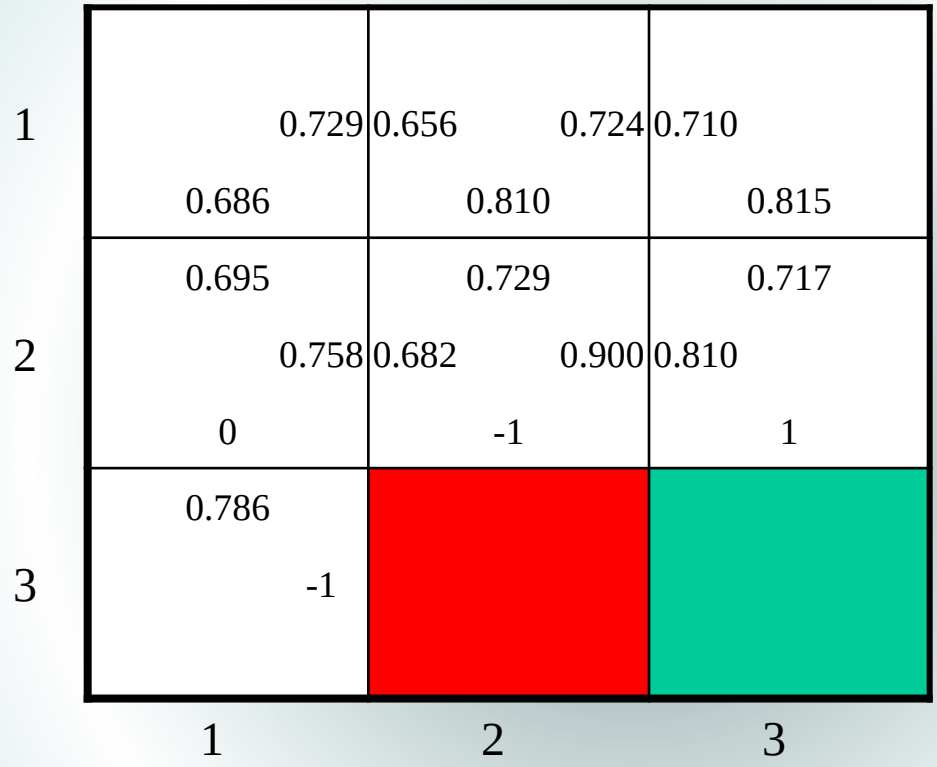
$Q((1,1),\mathrm{R}) = 0 + 0.1 \cdot (0 + 0.9 \cdot 0.0081 - 0) = 0.000729$
$Q((2,2),\mathrm{D}) = r(3,2) = -1$

Ohodnocení akcí po pěti předchozích procházkách:



Po naučení se pak přechází z libovolného stavu do jeho sousedního stavu akcí, která má nejvyšší hodnotu.

Příklad možného ohodnocení akcí po 1000 náhodných procházkách:



K žádnému problému nedochází!