Symbolické metody učení z příkladů

Jiří Kléma

Katedra kybernetiky,
FEL, ČVUT v Praze

http://ida.felk.cvut.cz

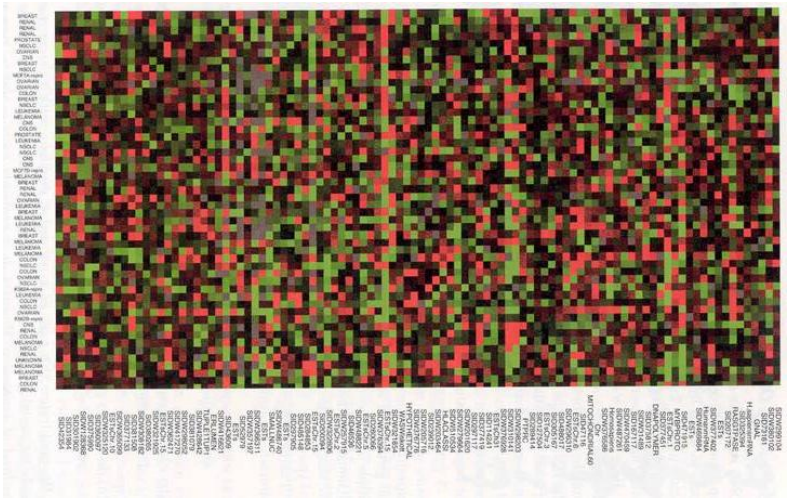
Plán přednášky

- Zaměření 1: učení z příkladů
  - motivace, formulace problému,
  - prediktivní a deskriptivní modely,
- Zaměření 2: symbolické metody učení
  - stromové prediktivní modely
    - rozhodovací stromy
      - TDIDT algoritmus, diskretizace, prořezávání,
    - regresní stromy
      - namísto rozhodnutí numerická předpověď,
  - pravidlové prediktivní modely
    - algoritmy AQ, CN2
  - pravidlové deskriptivní modely
    - asociační pravidla, algoritmus APRIORI.

Učení z příkladů – motivace

- Vytváření počítačových programů, které se zdokonalují se zkušeností
- Typicky na základě analýzy dat, tj. indukcí z příkladů
  - $(\vec{X}, Y)$ – s učitelem – předpověz Y – regrese, klasifikace
  - $\vec{X}$ – bez učitele – najdi podobné příznakové vektory – shlukování

Příklad 1: microarray data
najdi geny s podobnou expresí – shlukování



Příklad 2: digitalizované znaky
rozlišuj rukou psaná čísla – klasifikace

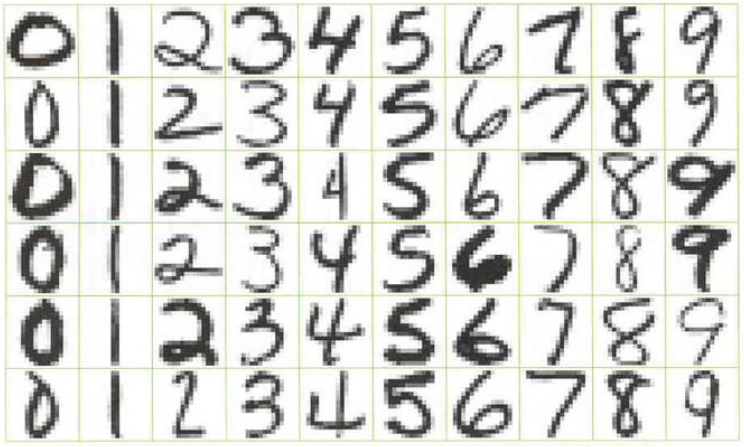


FIGURE 1.2. *Examples of handwritten digits from U.S. postal envelopes.*

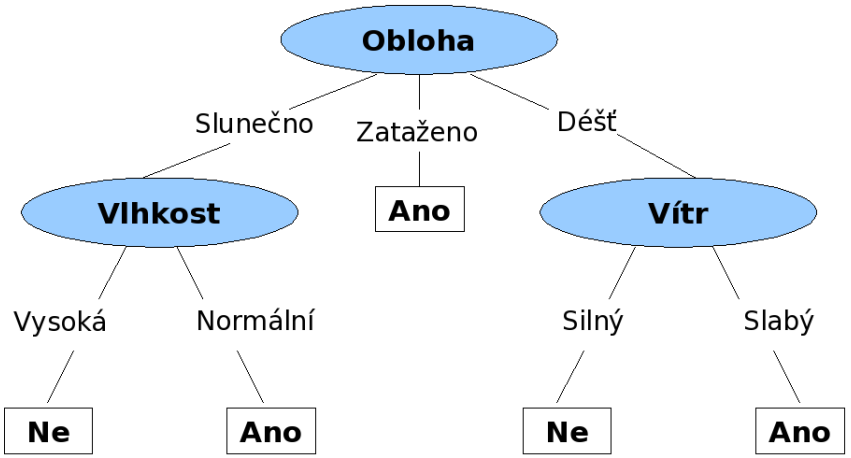
Typy učení

- **Symbolické strojové učení**
  - znalost vyjádřena ve formě symbolických popisů učených konceptů,
  - typicky algoritmy tvořící množiny pravidel zachycující vztah mezi atributy a třídou, tj. nezávislými proměnnými a proměnnou cílovou,
  - propozicionální i relační (induktivní logické programování),
- konekcionistické učení
  - znalost uchovávána ve formě sítě propojených neuronů s váženými synapsemi a prahovými hodnotami,
- pravděpodobnostní/statistické učení
  - modelem je např. distribuční funkce nebo funkce pstní hustoty,
  - statistické regresní modely (lineární, logistické), SVMs,
- další metody
  - např. simulovaná evoluce a genetické algoritmy (analogie s přírodními procesy a darwinistickou teorií přežití nejsilnějších).

Rozhodovací stromy

- Aproximují diskrétní cílovou veličinu
- Aproximace funkcí reprezentovanou stromem
- Rozhodovací strom je disjunktem konjunkcí mezi omezeními hodnot atributů
- Instance klasifikujeme dle hodnot atributů
- Reprezentace
  - vnitřní uzly testují vlastnosti atributů,
  - větve odpovídají možným hodnotám,
  - listy přiřazují klasifikaci, tj. konkrétní hodnotu cílové veličiny.

$$(\text{Obloha} = \text{Slunečno} \wedge \text{Vlhkost} = \text{Normální}) \vee$$
$$(\text{Obloha} = \text{Zataženo}) \vee$$
$$(\text{Obloha} = \text{Déšť} \wedge \text{Vítr} = \text{Slabý})$$



Hrát tenis/golf? Trénovací příklady.

Den	Obloha	Teplota	Vlhkost	Vítr	Tenis/golf
D1	Slunečno	Vysoká	Vysoká	Slabý	Ne
D2	Slunečno	Vysoká	Vysoká	Silný	Ne
D3	Zataženo	Vysoká	Vysoká	Slabý	Ano
D4	Déšť	Střední	Vysoká	Slabý	Ano
D5	Déšť	Nízká	Normální	Slabý	Ano
D6	Déšť	Nízká	Normální	Silný	Ne
D7	Zataženo	Nízká	Normální	Silný	Ano
D8	Slunečno	Střední	Vysoká	Slabý	Ne
D9	Slunečno	Nízká	Normální	Slabý	Ano
D10	Déšť	Střední	Normální	Slabý	Ano
D11	Slunečno	Střední	Normální	Silný	Ano
D12	Zataženo	Střední	Vysoká	Silný	Ano
D13	Zataženo	Vysoká	Normální	Slabý	Ano
D14	Déšť	Střední	Vysoká	Silný	Ne

TDIDT – indukce shora dolů

- TDIDT = Top-Down Induction of Decision Trees,
- konkrétní algoritmus: ID3 (Quinlan, 1986)

dáno: S - trénovací množina,
A - množina atributů,
C - množina tříd,

if $\forall s \in S$ patří do stejné třídy $c \in C$
then jde o list, označ jej třídou c
else
jde o uzel, vyber pro něj ‘‘nejlepší’’ rozhodovací atribut $a \in A$
(s hodnotami $v_1, v_2, \ldots, v_n$),
rozděl trénovací množinu S na $S_1, \ldots, S_n$ podle hodnot atributu a,
rekurzivně vytvářej podstromy $T_1, \ldots, T_n$ pro $S_1, \ldots, S_n$
(přechodem na začátek algoritmu s $S = S_i$),
výsledný strom T je tvořen podstromy $T_1, \ldots, T_n$.

- **který atribut a je nejlepší?**

Entropie

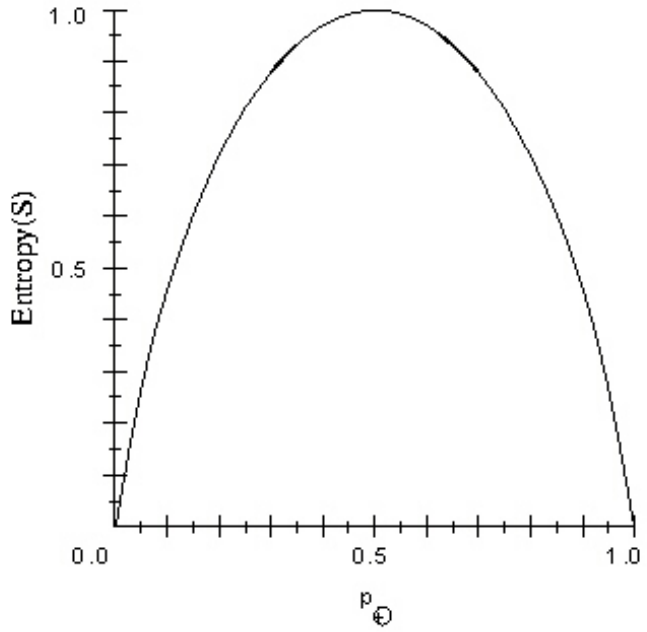
- S – vzorek trénovacích příkladů,
- p^+ (p^-) je zastoupení pozitivních (negativních) příkladů v S
  - zastoupení = relativní četnost $\sim$ pravděpodobnost,
- $H(S)$ – informační entropie trénovací množiny
  - nejmenší průměrný počet bitů nutných k zakódování zprávy o třídě libovolného $s \in S$,
- teorie informace
  - kód optimalní délky přiřazuje $-\log_2 p$ bitů zprávě s pravděpodobností p
- průměrný počet bitů potřebný k zakódování "+" nebo "–" náhodného prvku S

$$H(S) = -p^- \log_2 p^- - p^+ \log_2 p^+$$

- obecně pro c různých tříd:

$$H(S) = -\sum_{c \in C} p_c \log_2 p_c$$

Entropie



- Entropie jako funkce booleovské klasifikace vyjádřené relativní četností pozitivních příkladů
  - četnost negativních příkladů je redundantní informací $(p^- = 1 - p^+)$,
- entropie je mírou neuspořádanosti v souboru příkladů,
- ideální rozhodovací nástroj (klasifikátor) minimalizuje neuspořádanost, tj. entropii.

Informační zisk

- $Gain(S, a)$ nebo $IG(S, a)$ – heuristické kritérium
  - očekávaná redukce entropie za předpokladu rozdělení množiny S dle atributu A,
  - míra efektivity atributu pro klasifikaci trénovacích dat,
  - přímé použití entropie, pouze přechod od minimalizačního k maximalizačnímu kritériu

$$Gain(S, a) = Entropy(S) - \sum_{v \in Values(a)} \frac{|S_v|}{|S|} Entropy(S_v)$$

  - $Values(a)$ – možné hodnoty atributu a,
  - S_v - podmnožina S, pro kterou a má hodnotu v,
  - cílem je minimalizovat počet testů nutných k oddělení tříd (homogenizaci),
  - a v důsledku generovat co **nejmenší strom**.

Hrát tenis? výpočet zisku

- $Values(Vitr) = \{Slaby, Silny\}$
  - $S = [9+, 5-]$, $E(S) = 0.940$
  - $S_{slaby} = [6+, 2-]$, $E(S_{slaby}) = 0.811$
  - $S_{silny} = [3+, 3-]$, $E(S_{silny}) = 1.0$
- $Gain(S, Vitr) = E(S) - \frac{8}{14} \times E(S_{slaby}) - \frac{6}{14} \times E(S_{silny}) = 0.940 - \frac{8}{14} \times 0.811 - \frac{6}{14} \times 1.0 = 0.048$
- $Values(Obloha) = \{Slunecno, Zatazeno, Dest\}$
  - $S = [9+, 5-]$, $E(S) = 0.940$
  - $S_{slunecno} = [2+, 3-]$, $E(S_{slunecno}) = 0.968$
  - $S_{zatazeno} = [4+, 0-]$, $E(S_{zatazeno}) = 0$
  - $S_{dest} = [3+, 2-]$, $E(S_{dest}) = 0.968$
- $Gain(S, Obloha) = E(S) - \frac{5}{14} \times E(S_{slunecno}) - \frac{5}{14} \times E(S_{dest}) = 0.940 - 0.694 = \mathbf{0.246}$,
- $Gain(S, Vlhkost) = 0.151$,
- $Gain(S, Teplota) = 0.029$.

Další heuristiky

- Informační zisk (Gain, IG)
  - nezohledňuje jak široce a rovnoměrně atribut data dělí (extrémní příklad: unikátní index),
- Poměrný zisk (Gain Ratio, GR)
  - využívá doplňkovou informaci o dělení (split information)
  - penalizuje tím atributy jako např. datum

$$SI(S,a) = -\sum_{v \in Values(a)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$$

$$GR(S,a) = \frac{Gain(S,a)}{SI(S,a)}$$

- další - Gini index, Mantarasova heuristika, heuristiky vážící cenu získání atributu (Nunez)

$$GainCost(S,A) = \frac{2^{Gain(S,A)} - 1}{(Cost(A)+1)^w}$$

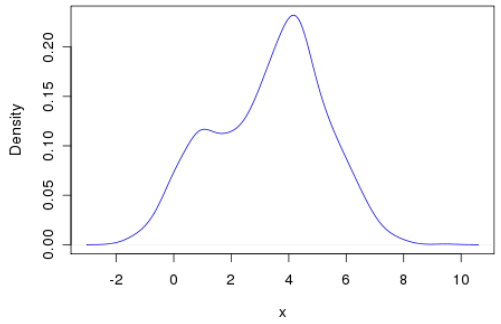
  - $w \in [0,1]$ – relativní důležitost Cost vs. Gain.

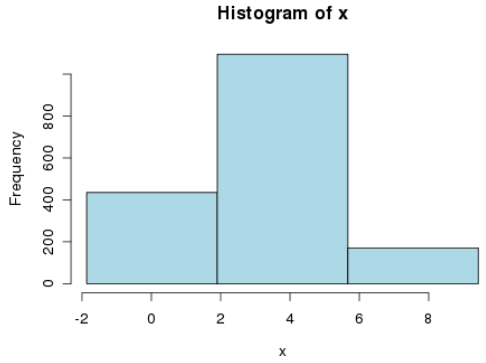
Spojité atributy

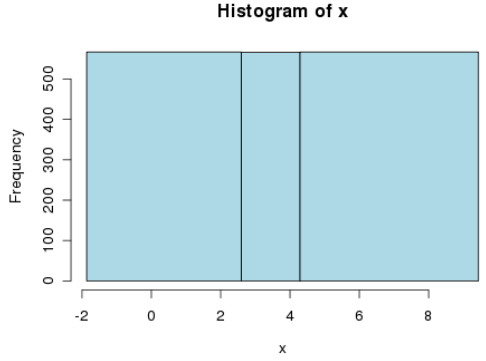
- TDIDT založen na dělení do rozumného počtu větví,
- spojité atributy musí být diskretizovány,
- kategorizace diskretizačních přístupů
  - s učitelem vs. bez učitele (supervised, unsupervised)
    - využíváme znalost cílové veličiny?
  - globální vs. lokální
    - globální – proveď diskretizaci před aplikací učícího se algoritmu na všech datech,
    - lokální – diskretizuj průběžně a pouze na aktuálních instancích,
    - výhody a nevýhody: přímý kontext vs. citlivost na šum,
  - práce s diskrétními atributy
    - uvažuje algoritmus uspořádané hodnoty atributů?
    - pokud ne, lze tomu přizpůsobit proces učení
      - převeď každý diskretizovaný atribut (k hodnot) na množinu k-1 binárních atributů, i-tá hodnota diskretizované veličiny = i-1 binárních atr. 0, zbytek 1
- TDIDT – s učitelem, lokální, diskrétní atributy neuspořádané.

Diskretizace bez učitele

- stejná šíře intervalů (equal-width binning)
  - děl na intervaly o stejné šířce,
  - šířka=(max-min)/počet_intervalů,
  - distribuce instancí může být nerovnoměrná,
- stejná četnost instancí (equal-depth binning)
  - založeno na rozložení hodnot atributu,
  - někdy nazývána vyrovnáním histogramu
    - uniformní (plochý) histogram,
- manuální, např. odvozené od histogramu.

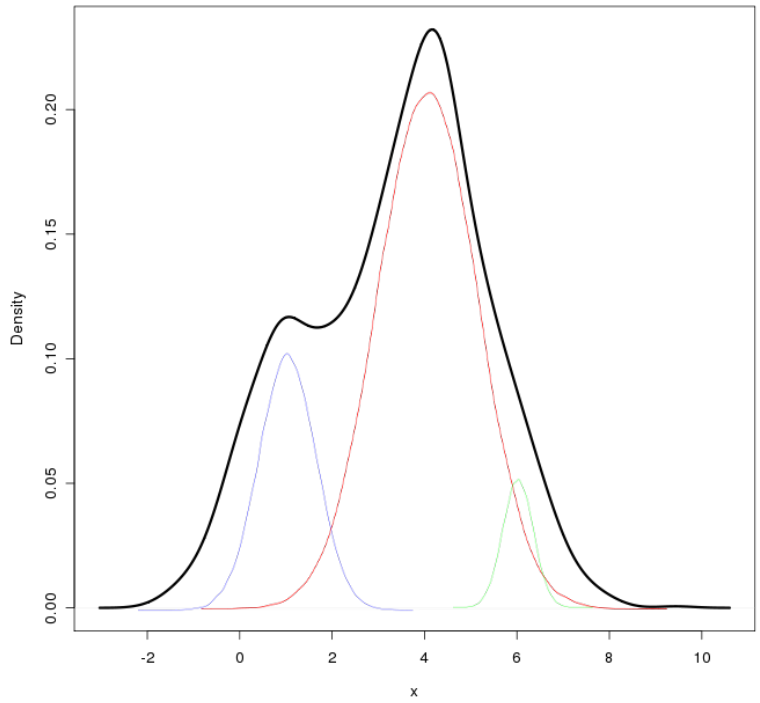






Diskretizace s učitelem

- často založená na entropii
- např. **binarizace**
  - definuj (dynamicky) nový booleovský atribut
  - a_c je true iff $a > c$,
  - jediná otázka – volba prahové hodnoty c,
  - zvol práh optimalizující IG heuristiku
    - setřiď všechny instance podle a,
    - najdi sousedící příklady lišící se klasifikací,
    - kandidát na c
      - střed mezi příslušnými hodnotami a,
    - zvol nejlepší c.



Diskretizace s učitelem

- generalizace binarizace = dělení do více intervalů
  - R-binarizace – a je použit vícekrát na stejné cestě od kořene k listu stromu.
  - zobecněná binarizace
    * najdi optimální dělení (IG je použit jako hodnotící fce),
    * opakuj až do splnění ukončovací podmínky,
- ukončovací podmínka?
  - princip minimální délky popisu (minimum description length – MDL)
    * nejlepší teorie je ta, která minimalizuje svoji délku současně s délkou zprávy nutnou k popsání výjimek z teorie,
    * nejlepší generalizace je ta, která minimalizuje počet bitů nutných ke komunikaci generalizace a všech příkladů, z nichž je vytvořena.
- diskretizace založená na chybovosti (namísto entropie)
  - minimalizuje počet chyb,
  - snadno se počítá ale nelze dosáhnout sousedních intervalů se stejnou třídou (označením).

Poznámky k tvorbě DT a ID3

- Jak obtížné je nalézt optimální strom?
  - uvažujme m instancí popsaných n binárními atributy
  - optimalita – strom minimalizuje počet testů $=$ uzlů
  - jde o NP-úplný problém
    - pro algoritmus bez orákula hypotézový prostor roste superexponenciálně s počtem atributů,
    - nelze prohledávat úplně,
    - je třeba řešit heuristicky/lokálním prohledáváním,
- Hladové prohledávání
  - žádné zpětné řetězení při prohledávání, TDIDT/ID3 udržuje jedinou průběžnou hypotézu,
  - nutně konverguje pouze k lokálnímu optimu,
- ID3 strategie upřednostňuje menší stromy před většími
  - jak? atributy s vysokým IG jsou blíže ke kořeni,
  - induktivní zaujetí – **Occamova břitva**
    - "Plurality should not be assumed without necessity." or KISS: "Keep it simple, stupid."
  - nejjednodušší strom pravděpodobně nebude obsahovat neužitečná omezení, tj. testy.

Prořezávání DT

- Přeučení (overfitting) DT
  - složité stromy odrážející singularity v trénovacích datech,
  - statistická volba atributů při stavbě stromu zajišťuje robustnost vůči šumům
    - přesto může být omezující rozklad na zcela homogenní podmnožiny příkladů,
  - řešení: prořezávání (pruning),
- prahové prořezávání jednoduše zdola nahoru
  - definuj prahovou přesnost ac_0,
  - if $ac = 1 - e/n > ac_0$ then prune,
  - (n = počet příkladů v uzlu, e = počet chyb = počet příkladů z minoritních tříd),
- prořezávání založené na očekávané chybě
  - aproximuj očekávanou chybu pokud daný uzel N (pokrývající množinu příkladů S) prořízneme Laplaceův odhad chyby: $E(S) = \frac{e+T-1}{n+T}$, T = počet tříd,
  - aproximuj "záložní chybu" v následnících N předpokládající, že prořezání neprovedeme $BackedUpE(S) = \sum_{l \in children(N)} p_l E(S_l)$,
  - je-li očekávaná chyba menší než záložní chyba prořež N.

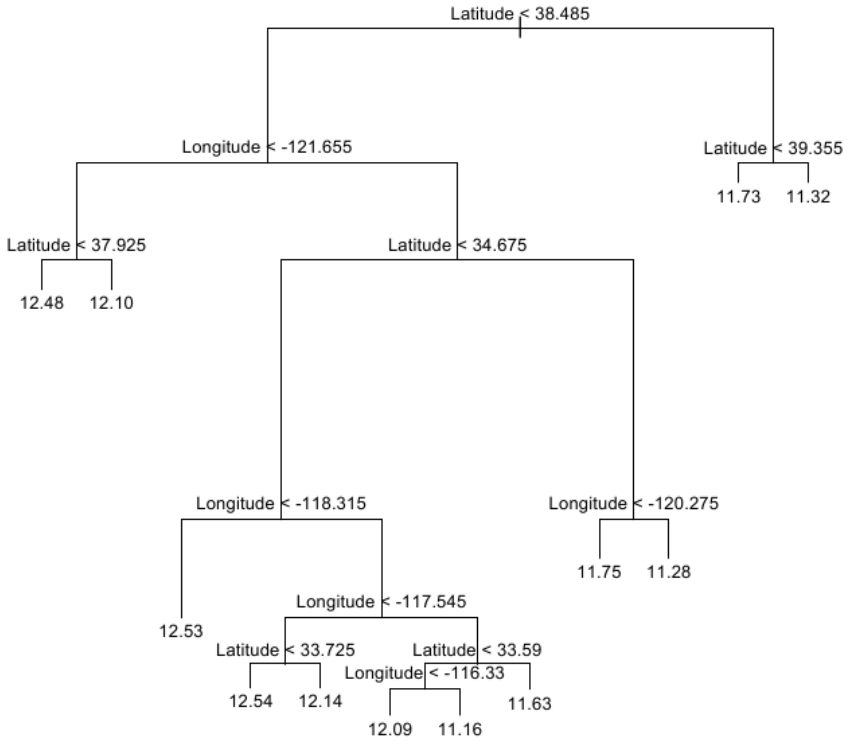
Vhodné problémy pro DT

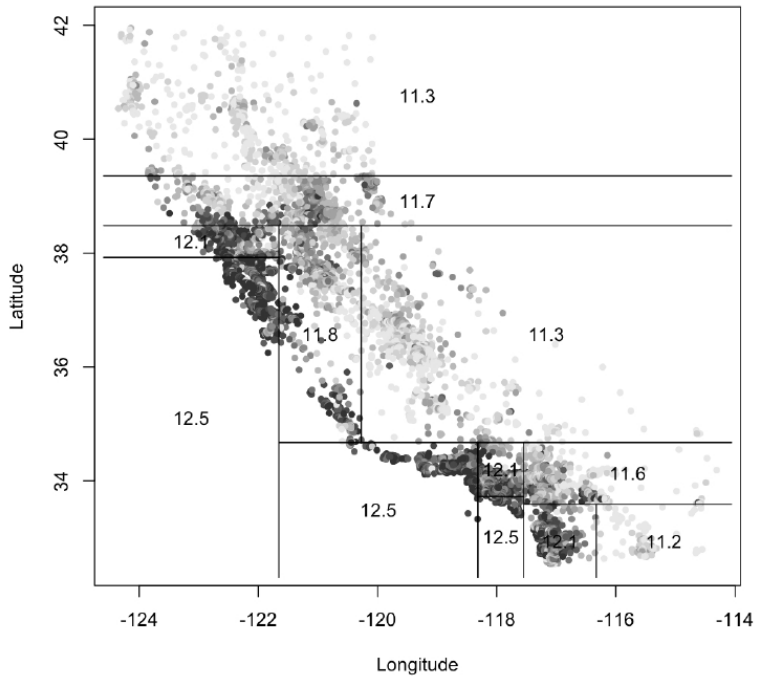
- Instance jsou reprezentovány páry atribut-hodnota (tj. data mají tvar obdélníkové tabulky)
  - ne vždy, viz multirelační data,
- cílová funkce je diskrétní
  - pro spojité cílové funkce regresní stromy,
- disjunktivní hypotézy jsou přijatelným výstupem,
- srozumitelná klasifikace – DT = **white box**,
  - strom rozumné velikosti je přehledný pro lidi,
- šum a chybějící hodnoty
  - data mohou obsahovat chybné klasifikace a hodnoty atributů,
  - v datech se mohou vyskytnout chybějící hodnoty atributů.

Regresní stromy

- učení se spojitou třídou
- CART (Breiman, Friedman)
  - v listech hodnoty (namísto ID tříd u DT)
- standardní odchylka jako chybová míra
  - míra její redukce určuje nejlepší atribut

$$\delta_{err} = sd(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} sd(S_v)$$





CART model mapy ceny domů v Kalifornii, ©Carnegie Mellon University, Course 36-350, Data Mining

Regresní stromy

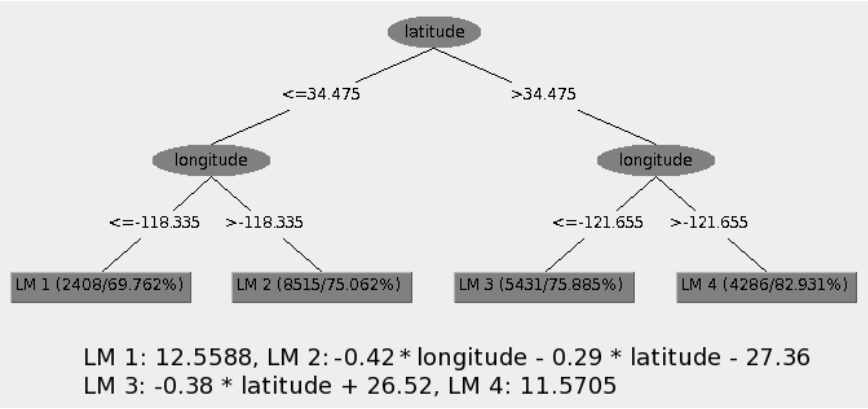
- M5 (Quinlan)
  - v listech multivariační lineární modely
    - schopnost extrapolace,
  - model = stromový, po částech lineární
  - další vylepšení
    - nečistota = standardní odchylka z regresního odhadu,
    - lineární model sestavený v uzlu zohledňuje pouze atributy použité v daném podstromu.
    - zjednodušení – eliminuj atributy málo vylepšující model

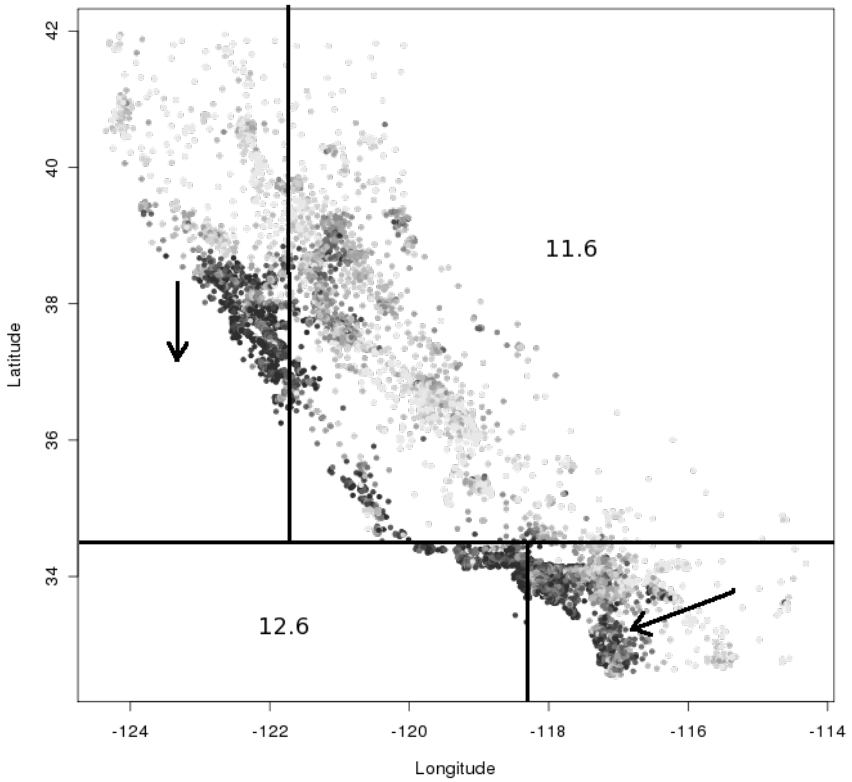
$$acc = \frac{n+\lambda}{n-\lambda}\frac{1}{n}\sum_{i=1}^{n}|f_i^{real} - f_i^{pred}|$$

      - λ . . . počet parametrů v modelu, f_i . . . hodnota cílové veličiny ve vzorku i,
      - extrémní případ . . . pouze konstanta (pak shoda s CART),

- prořezávání
  - porovnej přesnost lineárního modelu s přesností podstromu,
  - je-li model lepší pak prořež,
- vyhlazování
  - hodnota predikovaná v listu je přizpůsobena dle odhadů v uzlech na cestě od kořene.

Regresní stromy

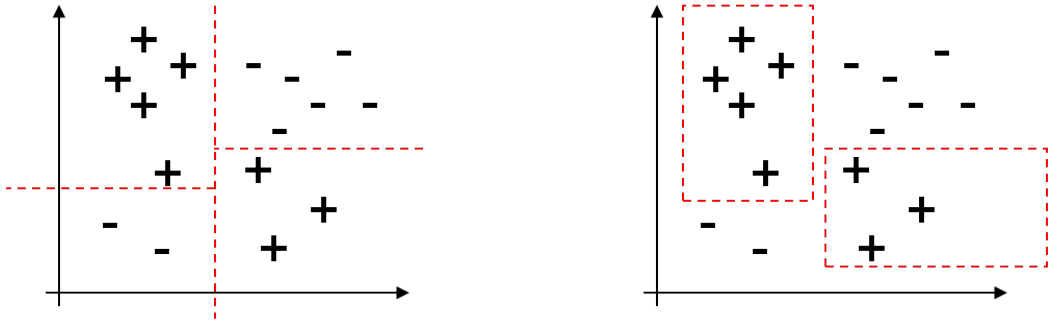




M5 model mapy ceny domů v Kalifornii, WEKA

Rozhodovací pravidla

- další metoda aproximace diskrétních cílových veličin,
- aproximační fce ve tvaru množiny (seznamu) rozhodovacích pravidel,
- **jestliže** <**podmínka**> **pak** <**třída**>
  - selektor = jednoduchý test hodnoty atributu,
  - např. <Obloha=Slunečno>, {=, ?, >, ?} typicky relační operátory,
  - podmínka (complex) = konjunkce selektorů,
- pravidla mají stejnou expresivitu jako stromy
  - stromy - "rozděl a panuj" $\longrightarrow$ dělení, čistě shora dolů,
  - pravidla - "odděl a panuj" $\longrightarrow$ pokrytí, konstrukce oběma směry.



AQ algoritmus

- Michalski – algoritmus pro pokrytí množiny

dáno: S - trénovací množina,
C - množina tříd,

$\forall c \in C$ opakuj
$S_c = P_c \vee N_c$ (pozitivní a negativní příklady)
bázePravidel(c)={}
opakuj {najdi množinu pravidel pro c}
R=find-one-rule($P_c \vee N_c$)
(najdi pravidlo pokrývající nějaké pozitivní příklady
a žádný negativní příklad)
bázePravidel(c)+=R
P_c-=pokryto(R,P_c),
dokud naplatí P_c={}

- **find-one-rule** procedura je kritickým krokem,
- přístup zdola-nahoru $\longrightarrow$ generalizace.

AQ algoritmus

- find-one-rule
  - zvol náhodně pozitivní příklad (zrno)
  - generuj všechny maximální generalizace zrna
    - pokrývá co nejvíce pozitivních příkladů
    - nepokrývá žádný negativní příklad
  - vyber nejlepší generalizaci (preferenční heuristika)

ID	a1	a2	a3	Třída
1	x	r	m	+
2	y	r	n	+
3	y	s	n	+
4	x	s	m	-
5	z	t	n	-
6	z	r	n	-

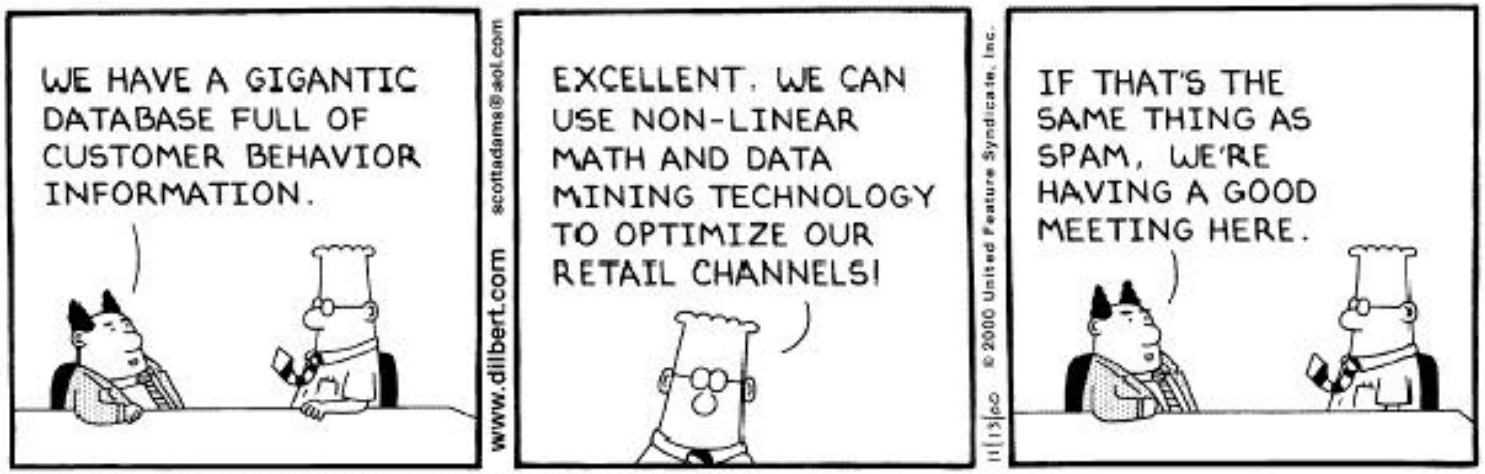
zrno 1: příklad 2
if (a1=y) & (a2=r) & (a3=n) then +
g1: if (a1=y) then +
zrno 2: příklad 1
if (a1=x) & (a2=r) & (a3=m) then +
g1: if (a1=x) & (a2=r) then +
g2: if (a2=r) & (a3=m) then +

Deskriptivní modely

- slouží ke **zhuštěnému popisu dat**, zjednodušeně zachycují obecné závislosti,
- kategorizace popisných modelů
  - na co se soustředí – tvoří globální model dat?
    - vyhledávání dominantních struktur
      - detekce podskupin, segmentace, shlukování, asociace,
    - vyhledávání nugetů, detekce odchylek – podvodné operace, síťové útoky, závadné www stránky,
  - jaký typ modelů využívají?
    - pravděpodbnostní modely – popis dat pomocí pravděpodobnostního rozdělení,
      - parametrické, neparametrické, směsi rozdělení,
    - **symbolické** modely – data interpretují konceptuálně na základě pojmů a jejich vztahů,
      - grafy, pravidla, taxonomie, logické vazby,
      - charakteristika: zřetelně a lidsky srozumitelně vyjadřují znalost,
    - kombinované modely
      - mj. grafické pstní modely – bayesovské sítě, markovské modely,
  - s jakými vstupními daty pracují?
    - číselná data, symbolická data, texty,
    - atributová reprezentace, relační databáze,
    - časová a sekvenční data.

Použití deskriptivních modelů

- privátní sektor
  - banky, pojišťovny, obchodní firmy,
  - snížení nákladů, zvýšení prodejů, průzkum trhu, odhalení podvodů,
- veřejný sektor
  - veřejná správa, lékařství, zpravodajské služby,
  - efektivita, zamezení ztrát a podvodům.



Asociační pravidla

- Association Rules (ARs)
- Definice
  - jednoduchá tvrzení o spoluvýskytu událostí v datech,
  - pravděpodobnostní charakter – nemusí platit vždy,
- Způsob zápisu a význam
  - if **Ant** then **Suc**,
  - alternativní zápis: Ant $\Rightarrow$ Suc,
  - antecedent (Ant) a sukcedent (Suc) definují obecné události v datech,
  - událost – jasně definovatelný jev, který buď nastává nebo nenastává,
  - z extenzivního popisu (dat) generujeme zhuštěný a přehledný popis – znalost.
- Příklady asociačních pravidel
  - doporučení pro nákup knih (Amazon):
    {Castaneda: **Učení Dona Juana**} $\Rightarrow$ {Hesse: **Stepní vlk** & Ruiz: **Čtyři dohody**}
  - vztah mezi rizikovými faktory a onemocněním v lékařství (Stulong):
    {**pivo** $\geq$ 1litr/den & **destiláty**=0} $\Rightarrow$ {not(**srdeční onemocnění**)}

Asociační pravidla – pojmy

- Položky (items): $I = \{I_1, I_2, \ldots, I_m\}$
  - binární atributy nebo výsledky aplikace relačních operátorů,
- Transakce (transactions): $D = \{t_1, t_2, \ldots, t_n\}$, $t_i \subseteq I$
  - příklady, objekty,
- Množiny položek (itemsets): $\{I_{i1}, I_{i2}, \ldots, I_{ik}\} \subseteq I$
  - analogie podmínky, současná platnost více položek,
- Podpora množiny položek: (relativní) četnost transakcí obsahujících danou množinu položek
- Frekventovaná (velká) množina položek:
  - četnost výskytů dané množiny položek je vyšší než zvolený práh,
- Asociační pravidlo (AR): implikace Ant $\Rightarrow$ Suc, kde Ant, Suc $\subseteq$ I a Ant $\cap$ Suc $= \emptyset$,
- Podpora (support) AR, s: poměr resp. počet transakcí obsahujících Ant $\cup$ Suc
  - pozn: podpora Ant $\Rightarrow$ Suc je stejná jako podpora Ant $\cup$ Suc,
- Spolehlivost (confidence) AR, α:
  - poměr mezi podporou AR (Ant $\cup$ Suc) a podporou jeho antecedentu (Ant),
  - vždy menší nebo rovna 1.

AR – problém vyhledávání

- Jsou dány:
  - množina položek $I = \{I_1, I_2, \ldots, I_m\}$,
  - databáze transakcí $D = \{t_1, t_2, \ldots, t_n\}$
    - kde $t_i = \{I_{i1}, I_{i2}, \ldots, I_{ik}\}$, a $I_{ij} \in I$,
  - minimální podpora s_{min},
  - minimální spolehlivost α_{min}.
- Problém vyhledávání asociačních pravidel:
  - nalézt všechna pravidla Ant $\Rightarrow$ Suc s podporou $s \geq s_{min}$ a spolehlivostí $\alpha \geq \alpha_{min}$.
- Realizaci lze rozdělit na 2 kroky:
  - najdi všechny frekventované (velké) (pod)množiny položek,
  - generuj z nich pravidla.

Př.: analýza nákupního košíku

- Cíl: zvýšit prodej a omezit náklady, tj. najdi položky často kupované společně,

Transakce	Položky
t_1	Chleba, Džem, Máslo
t_2	Chleba, Máslo
t_3	Chleba, Mléko, Máslo
t_4	Pivo, Chleba
t_5	Pivo, Mléko

- I = {Pivo, Chleba, Džem, Mléko, Máslo},
- Příklad pravidla: Chleba $\Rightarrow$ Máslo,
  - Ant={Chleba}$\in \{t_1, t_2, t_3, t_4\}$, s_{ant}=4/5=80%,
  - Ant $\cup$ Suc={Chleba,Máslo} $\in \{t_1, t_2, t_3\}$, podpora AR je s=3/5=60%,
  - Spolehlivost AR je $\alpha = s/s_{ant}$=75%.

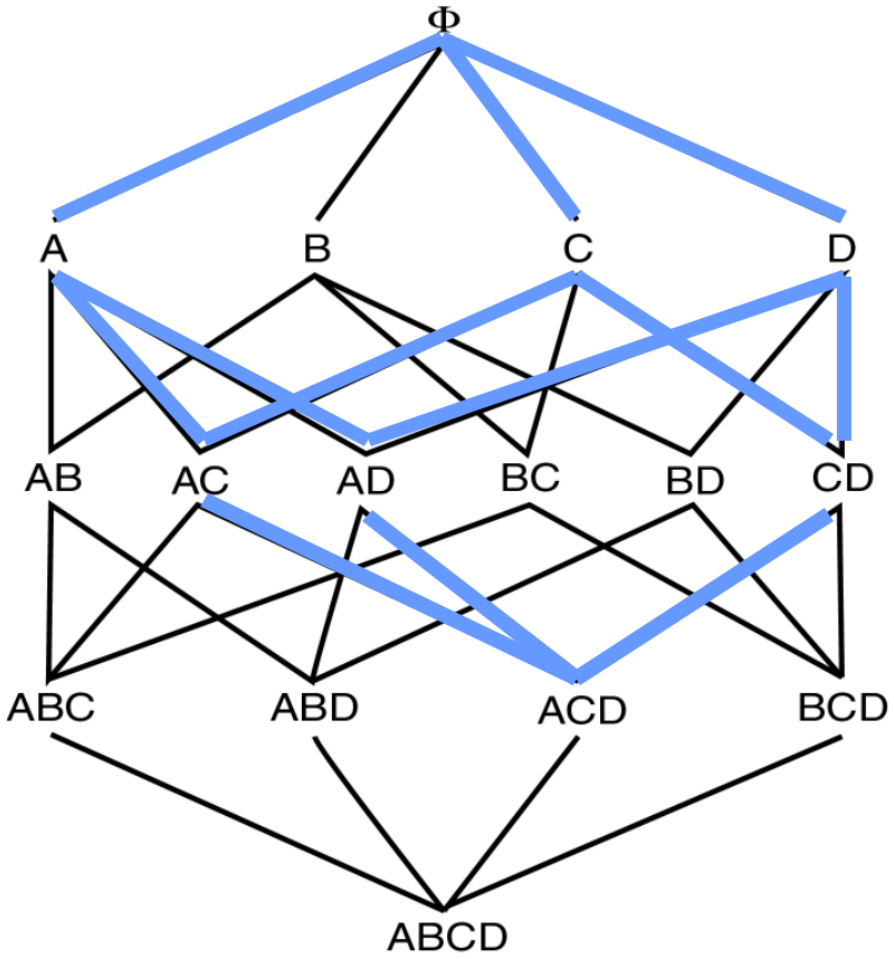
- Další pravidla a jejich parametry:

Ant $\Rightarrow$ Suc	s [%]	α [%]
Chleba $\Rightarrow$ Máslo	60	75
Máslo $\Rightarrow$ Chleba	60	100
Pivo $\Rightarrow$ Chleba	20	50
Máslo $\Rightarrow$ Džem	20	33
Džem $\Rightarrow$ Máslo	20	100
Džem $\Rightarrow$ Mléko	0	0

Vyhledávání velkých množin položek – krok 1

- Úplné prohledávání prostoru množin položek
  - pro m binárních položek existuje $3^m - 1$ množin položek,
  - pro N atributů, každý z nich má K kategorií existuje $(1 + K)^N - 1$ množin položek,
  - složitost roste **exponenciálně** s počtem položek (atributů),
- Algoritmus APRIORI
  - využívá charakteristické vlastnosti velkých množin položek:
    Každá podmnožina velké množiny položek je velká.
  - my ale postupujeme zdola nahoru – od podmnožin k nadmnožinám
    proto princip **kontrapozitivity** (přemístění) v logice
    $(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$
  - antimonotónní vlastnost se převádí na monotónní vlastnost, důsledek:
    Pokud množina položek není velká, žádná z jejích nadmnožin není velká.
  - kandidátské množiny položek
    - potenciálně velké – o všech jejich podmnožinách je známo, že jsou velké.

Vyhledávání velkých množin položek – ilustrace



APRIORI algoritmus [Agrawal et al., 1996]

Apriori:
  C_1 = $\forall$ kandidátské množiny položek velikosti 1 v I;
  L_1 = $\forall$ velké množiny položek velikosti 1 (podpora $\geq$ s_{min});
  i = 1;
  repeat
    i = i + 1;
    C_i = Apriori-Gen(L_{i-1});
    Počítej podporu C_i a vytvoř L_i;
  until žádná velká množina položek nenalezena ($L_i = \emptyset$);
  L = $\bigcup L_i$, $\forall$ i

Apriori-Gen(L_{i-1}):
  $C_i = \emptyset$
  pro $\forall$ dvojice množin položek $Comb_p$, $Comb_q \in L_{i-1}$:
    pokud se shodují v i-2 položkách pak přidej $Comb_p \cup Comb_q$ do C_i
  pro $\forall$ množiny položek Comb z C_i:
    pokud jakákoli podmnožina Comb o délce i-1 $\notin$ L_{i-1} pak odstraň Comb.

Aplikace APRIORI – příklad analýzy nákupního košíku

Transakce	Položky
t_1	Chleba, Džem, Máslo
t_2	Chleba, Máslo
t_3	Chleba, Mléko, Máslo
t_4	Pivo, Chleba
t_5	Pivo, Mléko

- Vstupní parametry: s_{min}=30% (α_{min}=50% – bude použito až v dalším kroku)

i	C_i	L_i
1	{Pivo}, {Chleba}, {Džem} {Mléko}, {Máslo}	{Pivo}, {Chleba} {Mléko}, {Máslo}
2	{Pivo, Chleba}, {Pivo, Mléko} {Pivo, Máslo}, {Chleba, Mléko} {Chleba, Máslo}, {Mléko, Máslo}	{Chleba, Máslo}

Generování pravidel z velkých množin položek – krok 2

```
Vstupy:
  I, D, L, α_min;

Výstup:
  R; % pravidla splňující s_min a α_min

AR-Gen:
  R = ∅;
  pro ∀ l ∈ L proveď:
    pro ∀ x ⊂ l taková, že x ≠ ∅ a x ≠ l proveď:
      jestliže s(l)/s(x) ≥ α_min, pak R = R ∪ {x ⇒ (l-x)}
```

- Příklad analýzy nákupního košíku
  - Vstupní parametry: L={Chleba, Máslo} (generováno pro s_{min}=30%), α_{min}=50%
  - Výstup: R={Chleba ⇒ Máslo: s=60%, α=75%, Máslo ⇒ Chleba: s=60%, α=100%}

Př.: průchod studiem

- Cíl: zjistit, zda skutečný průchod studiem odpovídá předpokladům a doporučením
- Předměty: RZN (Reprezentace znalostí), PAH (Plánování a hry), VI (Výpočetní inteligence), MAS (Multi-agentní systémy), SAD (Strojové učení a analýza dat), AU (Automatické uvažování)

Transakce	Položky
t_1	RZN
t_2	VI, SAD, AU
t_3	PAH, AU
t_4	PAH, VI, AU
t_5	PAH, MAS
t_6	VI, AU
t_7	PAH, SAD
t_8	PAH, VI, MAS, AU
t_9	PAH
t_{10}	PAH, VI, AU

Transakce	Položky
t_{11}	AU
t_{12}	RZN, PAH, VI, SAD, AU
t_{13}	PAH, VI, MAS, AU
t_{14}	VI, SAD, AU
t_{15}	PAH, AU
t_{16}	SAD, AU
t_{17}	RZN, PAH, SAD
t_{18}	PAH, VI, MAS, AU
t_{19}	PAH
t_{20}	PAH, VI, MAS, AU

APRIORI krok – s_{min}=20%, resp. 4

i	C_i	L_i
1	{RZN}, {PAH}, {VI} {MAS}, {SAD}, {AU}	{PAH}, {VI}, {MAS} {SAD}, {AU}
2	{PAH, VI}, {PAH, MAS}, {PAH, SAD} {PAH, AU}, {VI, MAS}, {VI, SAD} {VI, AU}, {MAS, SAD}, {MAS, AU} {SAD, AU}	{PAH, VI}, {PAH, MAS} {PAH, AU}, {VI, MAS} {VI, AU}, {MAS, AU} {SAD, AU}
3	{PAH, VI, MAS}, {PAH, VI, AU} {PAH, MAS, AU}, {PAH, SAD, AU} {VI, MAS, AU}, {VI, SAD, AU} {MAS, SAD, AU}	{PAH, VI, MAS} {PAH, VI, AU} {PAH, MAS, AU} {VI, MAS, AU}
4	{PAH, VI, MAS, AU}	{PAH, VI, MAS, AU}
5	$\emptyset$	$\emptyset$

AR-Gen krok – α_{min}=80%

PAH, VI: PAH ⇒ VI α=50%, VI ⇒ PAH α=70%
(PAH & VI současně 7krát, PAH 14 krát, VI 10 krát)

PAH, MAS: PAH ⇒ MAS 36%, **MAS ⇒ PAH** 100%

PAH, AU: PAH ⇒ AU 57%, AU ⇒ PAH 57%

VI, MAS: VI ⇒ MAS 40%, **MAS ⇒ VI** 80%

VI,AU: **VI ⇒ AU** 100%, AU ⇒ VI 71%

MAS, AU: **MAS ⇒ AU** 80%, AU ⇒ MAS 29%

SAD, AU: SAD ⇒ AU 66%, AU ⇒ SAD 29%

PAH, VI, MAS: PAH & VI ⇒ MAS 57%, **PAH & MAS ⇒ VI** 80%,
VI & MAS ⇒ PAH 100%

PAH, VI, AU: **PAH & VI ⇒ AU** 100%, **PAH & AU ⇒ VI** 88%,
VI & AU ⇒ PAH 70%

PAH, MAS, AU: **PAH & MAS ⇒ AU** 80%, PAH & AU ⇒ MAS 50%,
MAS & AU ⇒ PAH 100%

VI, MAS, AU: **VI & MAS ⇒ AU** 100%, **MAS & AU ⇒ VI** 100%,
VI & AU ⇒ MAS 40%

PAH, VI, MAS, AU: **PAH & VI & MAS ⇒ AU** 100%, PAH & VI & AU ⇒ MAS 57%,
VI & MAS & AU ⇒ PAH 100%, PAH & VI ⇒ MAS & AU 57%, atd.

APRIORI – výhody a nevýhody

- Výhody
  - efektivně využívá monotónní vlastnosti velkých množin položek,
  - obecně stále exponenciální složitost, ale zvládnutelný výpočet při:
    - vhodné volbě s_{min} a α_{min},
    - řídkých datech (v praxi spíše platí).
  - snadná implementace včetně paralelizace,
  - pro kolerovaná data s velkým počtem velkých množin položek může být dále vylepšen
    - použití zahuštěné reprezentace.
- Nevýhody
  - předpokládá residentní umístění databáze transakcí v paměti,
  - vyžaduje až m (počet položek) průchodů databází,
    - přístup lze urychlit použitím hashovacích stromů,
    - počet průchodů lze také snížit sloučením dvou následných velikostí do jednoho kroku,
    - za to zaplatíme větším počtem kandidátských množin, ale ...

Zápis vztahu mezi Ant a Suc čtyřpolní tabulkou

- Čtyřpolní tabulka = 4-fold table (4FT),
  - a, b, c, d → počty transakcí splňujících podmínky.

4FT	Suc	¬Suc	$\sum$
Ant	a	b	r=a+b
¬Ant	c	d	s=c+d
$\sum$	k=a+c	l=b+d	n=a+b+c+d

- Ne vždy je spolehlivost smysluplným kvantifikátorem
  - u často platných sukcedentů je **implikační** charakter spolehlivosti zavádějící,
  - i nezávislé množiny položek vykazují vysokou spolehlivost,
  - příklad čtyřpolní tabulky (s=45%,α=90%, Ant a Suc nezávislé):

450	50	500
450	50	500
900	100	1000

Příklady alternativních kvantifikátorů

- Spolehlivost lze v kroku AR-Gen nahradit libovolnou funkcí nad hodnotami čtyřpolní tabulky:
  - **Zdvih** (lift, above-average) je míra zlepšení přesnosti defaultní predikce pravé strany (spolehlivost dělená obecným podílem příkladů pokrytých sukcedentem)
    - lift=an/rk
  - **Páka** (leverage) je podíl příkladů, které jsou pravidlem (tedy Ant i Suc) pokryté dodatečně nad rámec počtu příkladů pokrytých za předpokladu nezávislosti Ant a Suc.
    - leverage=1/n(a-rk/n)
  - **Přesvědčivost** (conviction) je podobná zdvihu, ale uvažuje příklady nepokryté Suc pravidla, tím pádem musí pracovat s převráceným poměrem uvažovaných četností.
    - conviction=rl/bn

450	50	500
450	50	500
900	100	1000

s=0.45, α=0.9,
zdvih=1, páka=0,
přesvědčivost=1

10	1	11
90	899	989
100	900	1000

s=0.01, α=0.91,
zdvih=9.09, **páka=0.01**,
přesvědčivost=9.9

450	50	500
50	450	500
500	500	1000

s=0.45, α=0.9,
zdvih=1.8, páka=0.2,
přesvědčivost=5

Asociační pravidla – shrnutí

- Základní nástroj deskriptivního dolování dat
  - obecně identifikace jakéhokoli častého spoluvýskytu událostí v datech,
  - identifikace podskupin, odhalování skrytých závislostí, extrakce znalostí.
- Praktické využití
  - nejenom analýza nákupního košíku!!!
  - obecně jakákoli data atributového typu
    - lékařství, průmyslová měření, časo-prostorová data, . . . ,
  - nutností je pouze transakční převod dat, tj. přechod na binární atributy
    - nejčastěji dichotomizací (postupným tříděním do 2 skupin),
    - pro numerické veličiny navíc diskretizací,
    - v úvahu připadá i kódování (minimalizuje počet položek, ale snižuje přehlednost výstupu),
    - př. atribut teplota ve st. Celsia
      - diskretizace: $\{(\text{-}\infty,0\rangle \equiv$ nízká, $(0,15\rangle \equiv$ střední, $(15, \infty) \equiv$ vysoká$\}$,
      - dichotomizace: $\{i_1 \equiv$ t=nízká, $i_2 \equiv$ t=střední, $i_3 \equiv$ t=vysoká$\}$,

Doporučené doplňky – zdroje přednášky

- Mařík & kol.: **Umělá inteligence.**
  - díl I., kapitola Strojové učení,
  - díl IV., kapitola 11., Strojové učení v dobývání znalostí,
- Berka: **Dobývání znalostí z databází.**
  - kniha Academia, široký přehled,
- Quinlan: **Induction of Decision Trees.**
  - starší článek (1986) s příkladem z přednášky,
  - http://www.di.unipi.it/~coppola/didattica/ccp0506/papers/ML_1.1.81_Quinlan.pdf,
- Agrawal, Srikant: **Fast Algorithms for Mining Association Rules.**
  - asociační pravdila, APRIORI algoritmus,
  - rakesh.agrawal-family.com/papers/vldb94apriori.pdf.