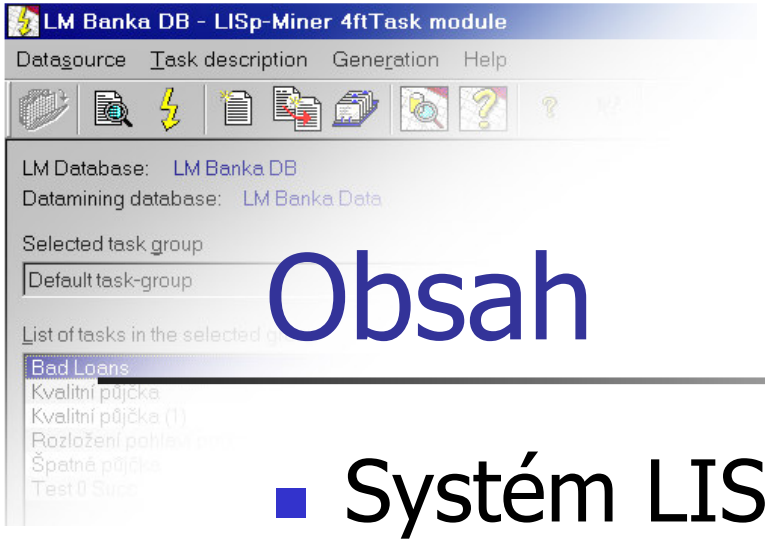
Projekt LISp-Miner

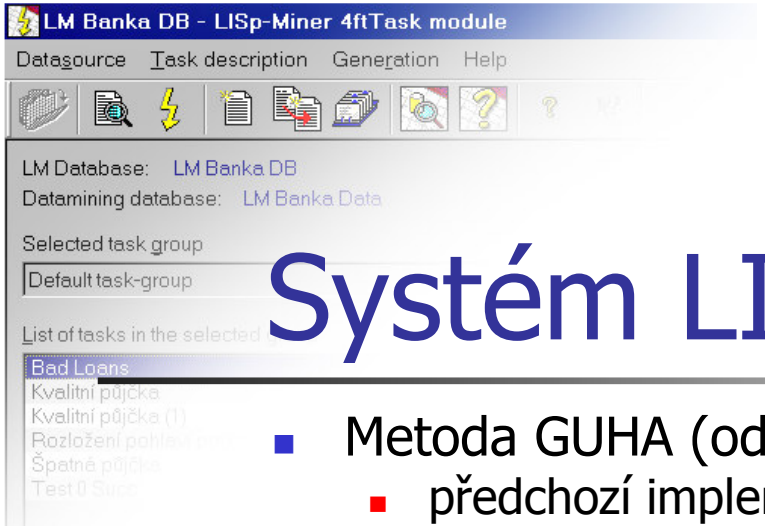
http://lispminer.vse.cz

M. Šimůnek



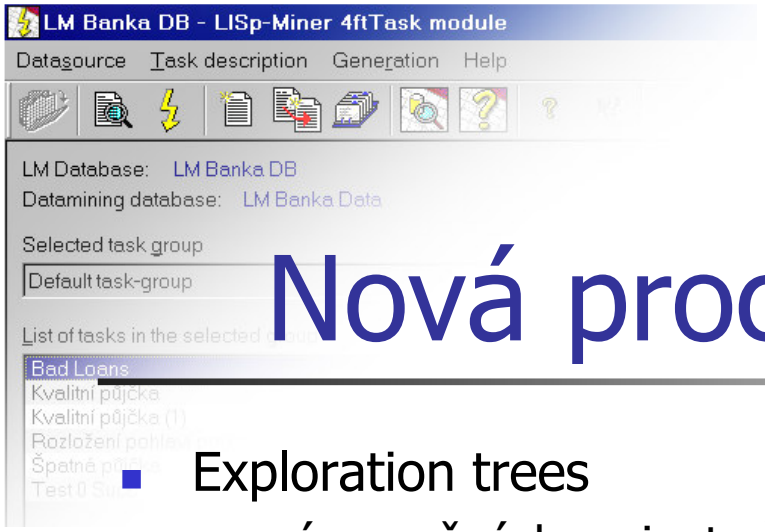
Obsah

- Systém LISp-Miner
- Vývoj systému v dlouhém období
- ETree-Miner



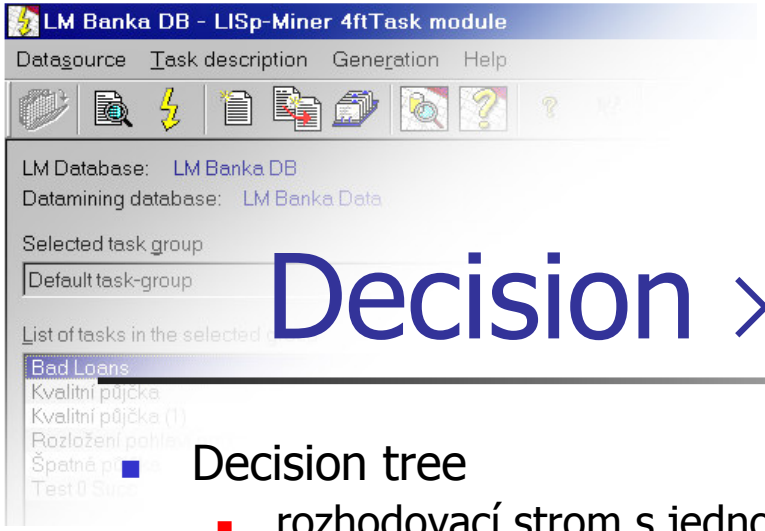
Systém LISp-Miner

- Metoda GUHA (od roku 1966)
  - předchozí implementace (IBM 370, PC-GUHA...)
- Akademické prostředí
- http://lispminer.vse.cz
- Cíle
  - použití ve výuce
  - použití pro výzkum v oblasti DM
  - použití pro řešení reálných DM úloh
  - využíván navazujícími projekty (SEWEBAR, EverMiner…)
- Modulární a vrstvená architektura
- Důraz na rychlost výpočtu
  - generování a verifikace co nejrychlejší (i pro složité zadání úloh)
- Vývoj od roku 1996
  - cca 1 mil. programových řádků, vývojové prostředí MSVC++



Nová procedura ETree-Miner

- Exploration trees
  - více možných variant větvícího atributu
  - vzniká „les" klasifikující s lepší přesností než jeden každý individuální strom
  - výpočetně náročné, protože generovaných a prověřovaných stromů obrovské množství
- Využití prostředků implementovaných v systému LISp-Miner
  - bitové řetězce
  - 4ft-cedent (generování pod-stromů)
  - kontingenční tabulky ⇔ confusion matrix
- GUHA-procedura
  - procházení celého stavového prostoru možných stromů a výběr všech, které splňují kritéria zadání

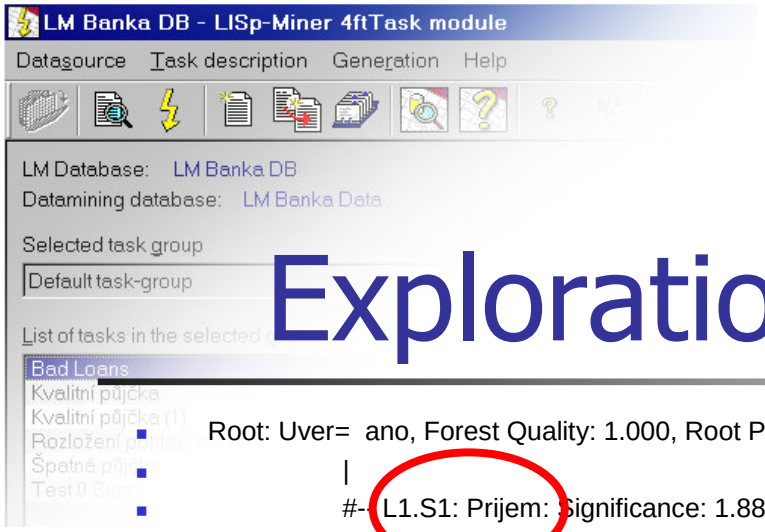


Decision × Exploration Tree

- Decision tree
  - rozhodovací strom s jednoduchou strukturou *kořen – uzly – listy*
  - Top-down induction (TDIDT)
    - např. ID3
  - možnost klasifikace případů

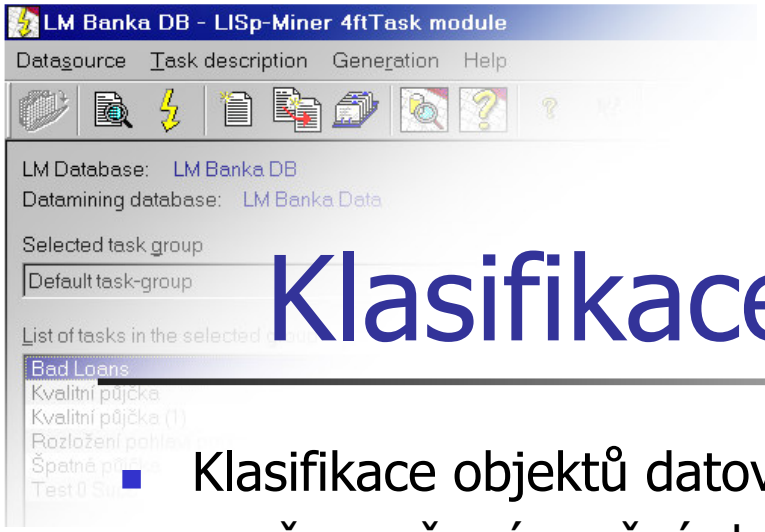
```
Root: Uver= ano, Tree Quality: 0.917, Root Purity: 0.667 (8 from 12)
  |
  #– L1.B1: Prijem= nízký: Uver= ne, Branch Quality: 0.857, Node Purity: 0.571 (4 from 7)
  | |
  | #– L1.B1.L2.B1: Konto= nízké: Uver= ne, Node Purity: 1.000 (2 from 2)
  | |
  | #– L1.B1.L2.B2: Konto= střední: Uver= ne, Node Purity: 0.667 (2 from 3)
  | |
  | #– L1.B1.L2.B3: Konto= vysoké: Uver= ano, Node Purity: 1.000 (2 from 2)
  |
  #– L1.B2: Prijem= vysoký: Uver= ano, Node Purity: 1.000 (5 from 5)
```

- Exploration tree
  - namísto výběru jednoho nejlepšího atributu pro větvení (*split*) jsou postupně vybrány všechny s dostatečnou kvalitou
    - nyní podle χ^2 testu
  - popisuje velké množství možných variant rozhodovacích stromů
  - zvolením vždy právě jednoho z možných větvících atributů vznikne jedna varianta rozhodovacího stromu
- Možnost zadání 4ft-podmínky
  - vzniká „pod-strom" – větev pro podmnožinu záznamů datové matice (pouze těch splňujících 4ft-podmínku)



Exploration Tree (ETree)

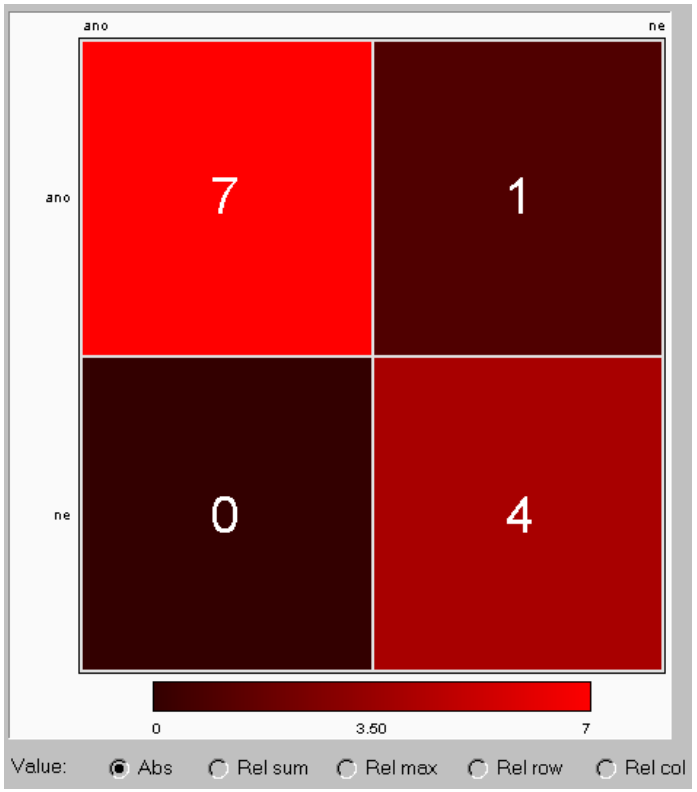
```
Root: Uver=  ano, Forest Quality: 1.000, Root Purity: 0.667 (8 from 12)
|
#-- L1.S1: Prijem: Significance: 1.888, Branch Frequency: 12, Branch Quality: 1.000
|  |
|  #-- L1.S1.B1: Prijem= nízký: Uver=  ne, Node Purity: 0.571 (4 from 7)
|  |  |
|  |  #-- L1.S1.B1.L2.S1: Konto: Significance: 2.233, Branch Frequency: 7, Branch Quality: 1.000
|  |  |  |
|  |  |  #-- L1.S1.B1.L2.S1.B1: Konto=  nízké: Uver=  ne, Node Purity: 1.000 (2 from 2)
|  |  |  |
|  |  |  #-- L1.S1.B1.L2.S1.B2: Konto=  střední: Uver=  ne, Node Purity: 0.667 (2 from 3)
|  |  |  |
|  |  |  #-- L1.S1.B1.L2.S1.B3: Konto=  vysoké: Uver=  ano, Node Purity: 1.000 (2 from 2)
|  |  |
|  |  #-- L1.S1.B1.L2.S2: Nazamestnany: Significance: 0.940, Branch Frequency: 7, Branch Quality: 1.000
|  |  |  |
|  |  |  ...
|  |
|  #-- L1.S1.B2: Prijem= vysoký: Uver=  ano, Node Purity: 1.000 (5 from 5)
|
#-- L1.S2: Konto: Significance: 1.875, Branch Frequency: 12, Branch Quality: 1.000
|  |
#-- L1.S3: Nezamestnany: Significance: 1.532, Branch Frequency: 12, Branch Quality: 0.950
|  |
```

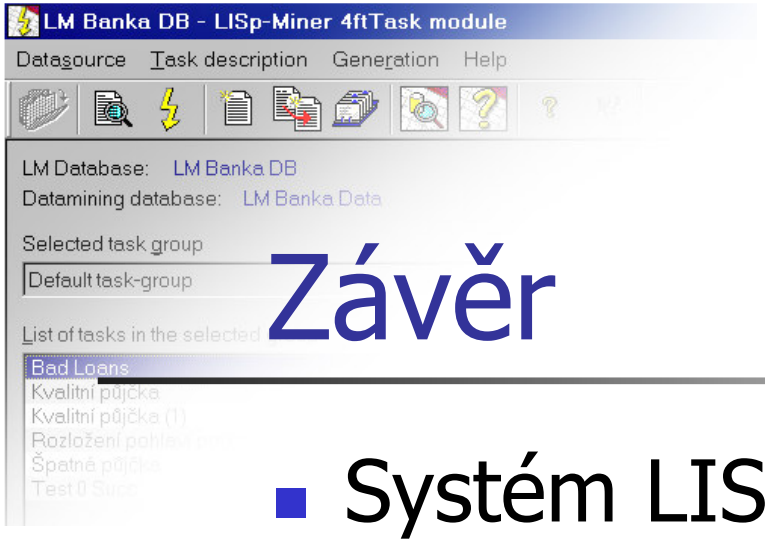


Klasifikace

- Klasifikace objektů datové matice
  - červeně zvýrazněné chyby
- *Confusion matrix*
  - absolutní i relativní četnosti úspěšné a neúspěšné klasifikace

#	Prijem	Uver	Prediction	Ok
1	vysoký	ano	ano	1
2	vysoký	ano	ano	1
3	nízký	ne	ne	1
4	nízký	ano	ne	
5	nízký	ano	ne	
6	nízký	ne	ne	1
7	vysoký	ano	ano	1
8	vysoký	ano	ano	1
9	nízký	ne	ne	1
10	vysoký	ano	ano	1
11	nízký	ne	ne	1
12	nízký	ano	ne	





Závěr

- Systém LISp-Miner
- Využití v projektu SEWEBAR
- Další vývoj

http://lispminer.vse.cz

DOPORUČOVÁNÍ ATRIBUTŮ PŘI ZÍSKÁVÁNÍ ASOCIAČNÍCH PRAVIDEL Z DAT POMOCÍ ROZHODOVACÍCH STROMŮ

KIZI, VŠE v Praze

Radek Škrabal

26. 4. 2012

Představení

- Radek Škrabal
  - FEL ČVUT (2006 – 2010)
  - FIS VŠE (2010 – 2012)
  - DP: Vizualizace asociačních pravidel
  - Vývojář webových aplikací
  - Kontakt: radek@skrabal.me

Plán

- Úvod (5 min)
  - Aplikace AR Builder
- Dolování asociačních pravidel (5-10 min)
  - Integrace background knowledge
- Doporučování atributů (20-25 min)
  - Motivace
  - Definice
  - Řešení
- Závěr

Úvod – Aplikace AR Builder

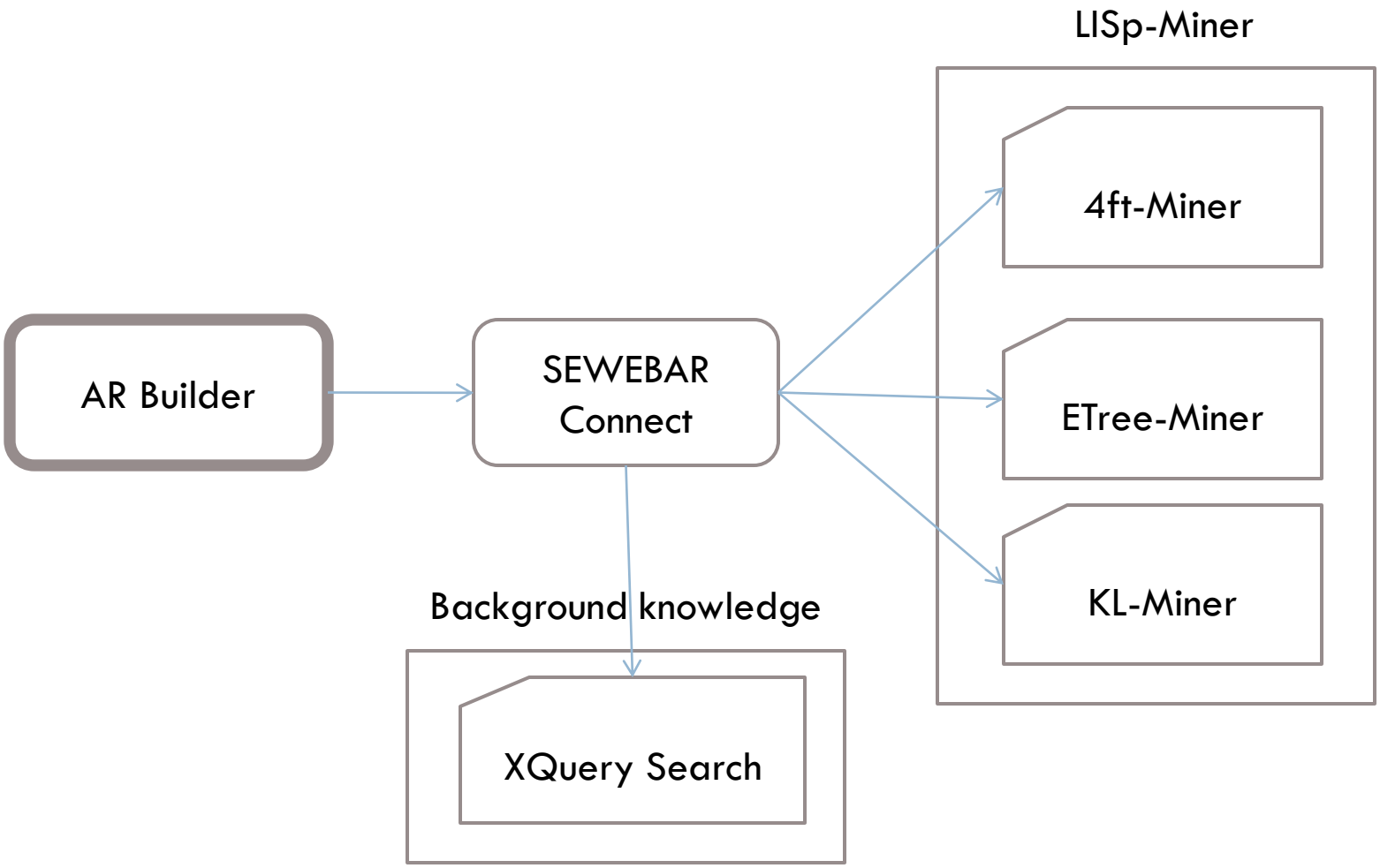
Úvod – Aplikace AR Builder

- Webová aplikace
  - HTML 5, PHP 5, MooTools
- Dolování asociačních pravidel
  - Interaktivní přístup, rychlá odezva
  - Tvorba kvalitních analytických zpráv
  - Využití existujících nástrojů (LISp-Miner, XQuery Search)

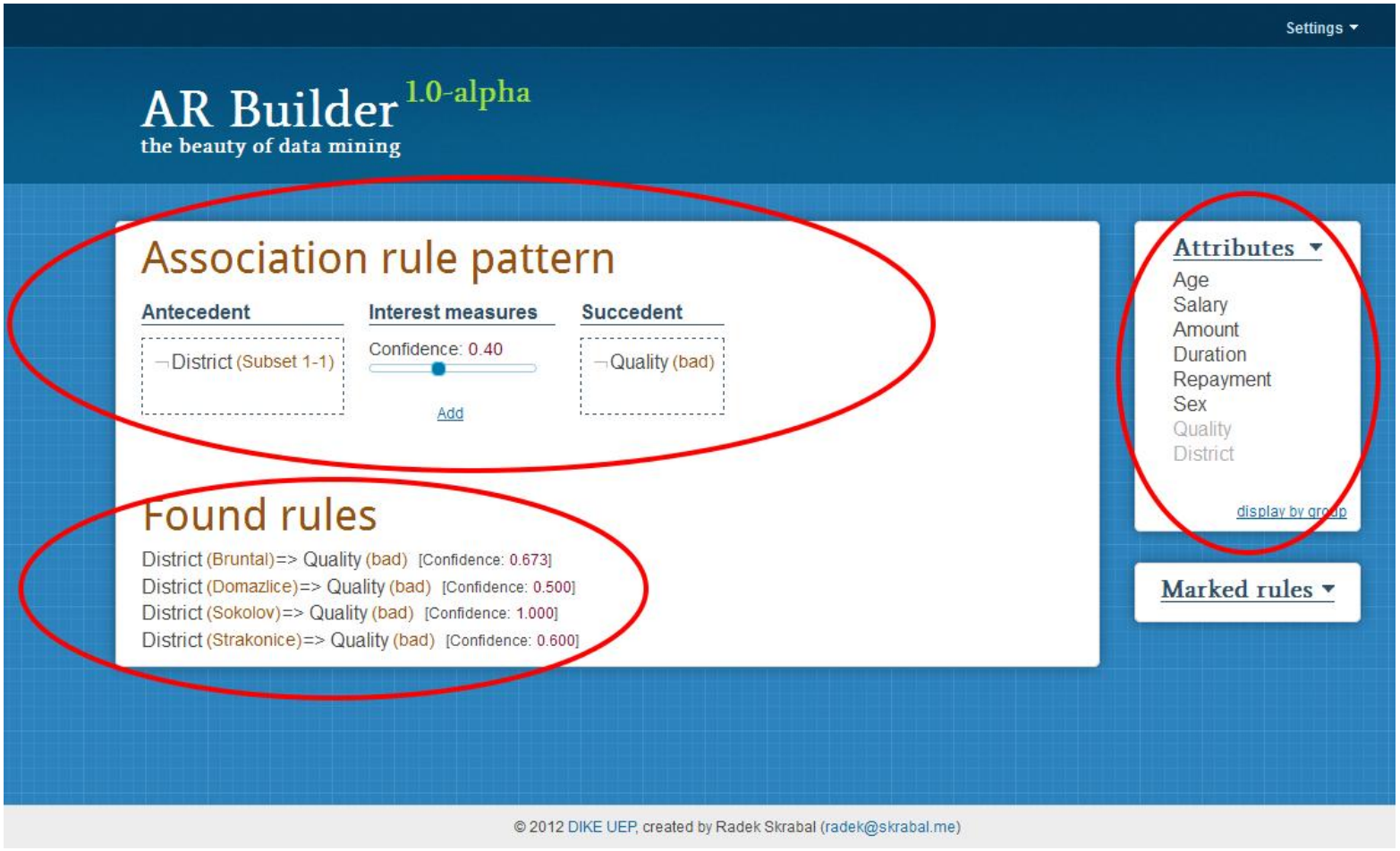
AR Builder – XML konfigurace

- DataDescription – popis dat
- FeatureList – restrikce zadání
  - Restrict
  - Expert
- FrequencyAnalysis – K x L polní tabulka četností

AR Builder - Integrace



AR Builder - UI



Dolování asociačních pravidel

Dolování asociačních pravidel

- Problémy
  - Nesetříděné množství pravidel
  - Chybí interaktivita
- Řešení
  - Interaktivní proces dolování
  - Doporučení vhodného atributu
  - Označení zajímavých pravidel
  - Integrace background knowledge

Integrace background knowledge

District (Bruntal)=> Quality (bad) [Confidence: 0.673]

District (Domazlice)=> Quality (bad) [Confidence: 0.500]

District (Sokolov)=> Quality (bad) [Confidence: 1.000]

District (Strakonice)=> Quality (bad) [Confidence: 0.600]

Integrace background knowledge

District (Bruntal)=> Quality (bad) [Confidence: 0.673]

District (Domazlice)=> Quality (bad) [Confidence: 0.500]

District (Sokolov)=> Quality (bad) [Confidence: 1.000]

District (Strakonice)=> Quality (bad) [Confidence: 0.600]

Integrace background knowledge

- Zatím v teoretické rovině
- Vyhledávání doménových znalostí (XQuery Search)
  - Předem indexované
- Typy dotazů
  - Je pravidlo obecně známé?
  - Je pravidlo v rozporu s obecně známým?
  - …

Integrace background knowledge

Typické workflow:

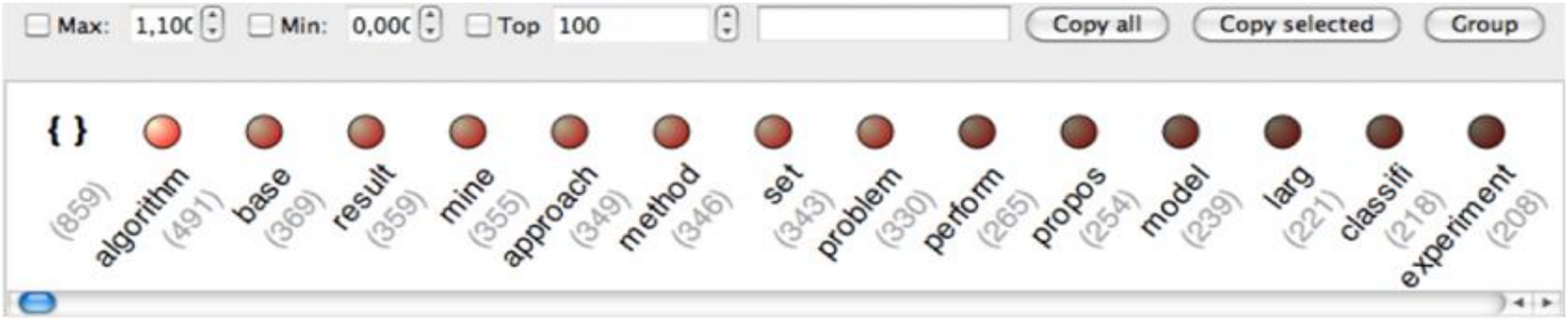
1. Dolování asociačních pravidel
2. Pro každé asociační pravidlo
    1. Dotaz na doménové znalosti
    2. Skrytí, pokud je nezajímavé
    3. Označení, pokud je zajímavé
3. Zpět na krok 1, dokud není uživatel spokojen
4. Uložení do analytické zprávy

Praktická ukázka

Doporučování atributů

Motivace

- MIME Framework
  - University of Antwerp, Belgium
  - Úspěch na ACM SIGKDD 2011
  - Best Pattern Extension
    - Nejlepší rozšíření zvoleného vzoru
    - Vzor – itemset (apriori)



Definice

Uvažujme asociační pravidlo ve tvaru:

$$Ant \approx Succ \mid Cond$$

kde *Ant, Succ* a *Cond* jsou kombinace literálů a vztah $\approx$ mezi *Ant a Succ* je definován pomocí (zobecněného) kvantifikátoru.

Definice

Nechť A je množina atributů. Nechť $Ant \cap Succ = \emptyset$.

Rozšířené asociační pravidlo pak definujme jako:

$$Ant \wedge val(a) \approx Succ \mid Cond$$

kde $a \in A$; $val(a)$ je literál z atributu a; a není obsažen v Ant, $Succ$ ani $Cond$.

Vhodnost atributu

Hledáme atribut $a \in A$, který dokáže posílit vztah $\approx$ mezi *Ant* a *Succ* definovaný (zobecněným) kvantifikátorem.

Pro vhodnost atributu definujme funkci *qual*:

$$0 <= qual(a) <= 1$$

Pro perspektivní rozšíření pravidla budeme uvažovat atributy $qual(a) >= q_{min}$.

Uvažovaná řešení

- Statistické
  - Souvislost uvažovaného atributu s cílovým
  - Fisherův test, metriky vycházející z entropie
- Vytvoření modelu
  - Využití popisných metrik
  - Explorační stromy

ETree Miner

- Nová GUHA procedura
- Výhody
  - 1 běh pro všechny atributy
- Nevýhody
  - Citlivé nastavení zadání (exp. složitost)
  - Jen 1 cílový atribut
  - V současnosti jen Chi-Square kvantifikátor

ETree Miner

Počet generovaných stromů:

$$NT = k \cdot \prod_{l=1}^{lmax} k^{vl}$$

kde *k* je počet větvících atributů, *lmax* je maximální hloubka stromu a *vl* je počet uzlů v hloubce *l*.

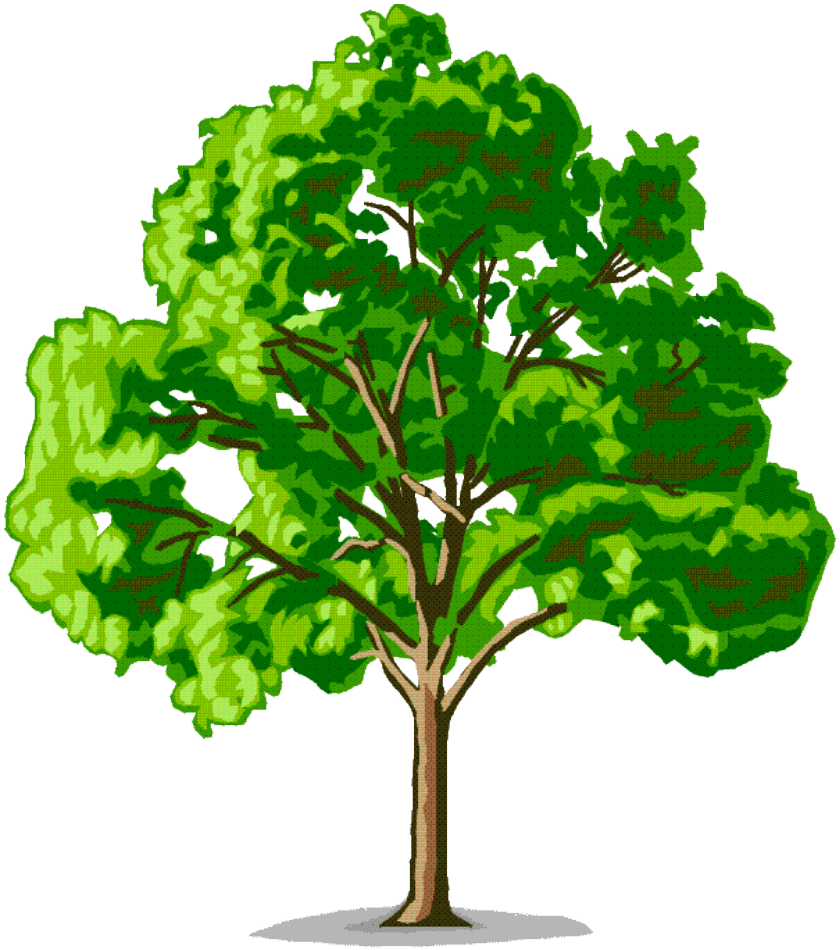
ETree Miner

Chi-Square kritérium pro test signifikance:

$$\chi^2 = \sum_i \sum_j \frac{a_{ij} - \frac{r_i \cdot s_j}{n}}{\frac{r_i \cdot s_j}{n}}$$

kde a_{ij} je počet příkladů i-té hodnoty atributu *A* a j-té hodnoty cílového atributu, r_i je počet příkladů majících i-tou hodnotu atributu *A*, s_j je počet příkladů majících j-tou hodnotu cílového atributu a *n* je počet příkladů.

Doporučení atributu z ETree



Algoritmus

1. Sestavení zadání ETree úlohy
2. Spuštění 1 běhu úlohy
3. Výběr větví dle koeficientu v Succ
4. Výběr split atributů
5. Pokud neexistuje odpovídající atribut -> konec
6. Pro každý split atribut
    1. Výpočet hodnoty qual
    2. Seřazení sestupně podle qual
7. Doporučení atributu s max(qual)

ETree Miner - zadání

Split attributes

- Seznam atributů, které nejsou v *Ant* ani *Succ*

Condition

- Celý *Ant*

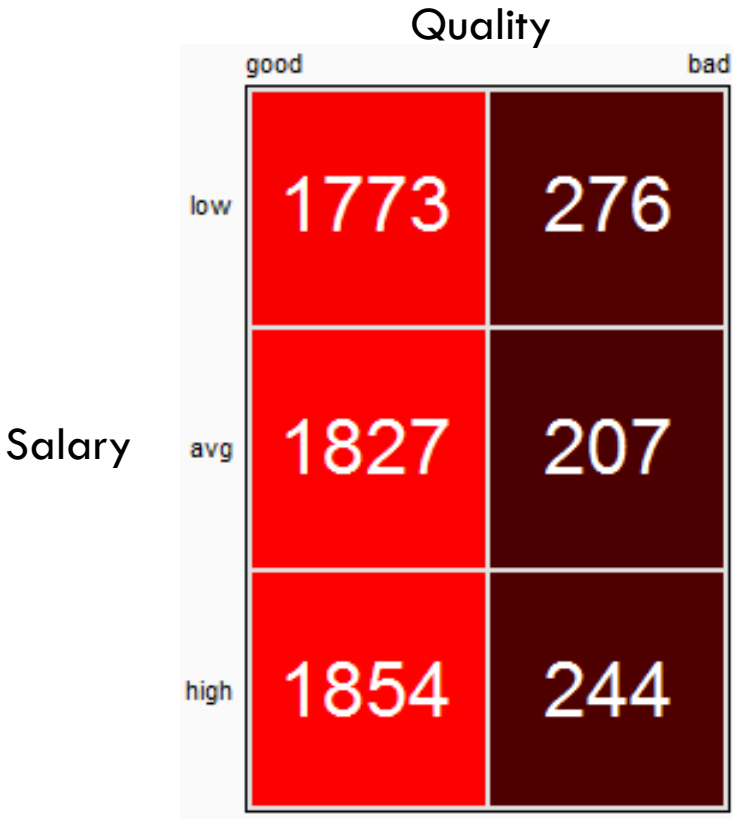
Class attribute

- Atribut v *Succ*

ETree Miner - zadání

Minimal node purity, Minimal tree quality

- Využití KL-Miner (frekvenční analýza)



$fr_1 = 1773 / 2049 = 0.865$
$fr_2 = 1827 / 2034 = 0.898$
$fr_3 = 1854 / 2098 = 0.884$

Salary (high)
$NP = fr_3$

Salary (*)
$NP = avg(fr_1, fr_2, fr_3) + stdev(fr_1, fr_2, fr_3) = \underline{0.894}$

$TQ = median(fr_1, fr_2, fr_3) = \underline{0.884}$

ETree Miner - zadání

Maximal tree depth

- Experimenty s 1-2

Maximal nr. of split attributes

- Experimenty s 3-6
- Vyžaduje další studium

Doporučení atributu – 1. krok

Výběr správných větví ETree

- Na základě koeficientu v *Succ*
  - Typ *One category* – výběr jen těch větví, které správně klasifikují tuto kategorii
  - Typ * – všechny větve (zjednodušení)

Doporučení atributu – 2. krok

Výběr split atributů z ETree

- Informace pro každý split atribut
  - signifikance – Chi-Square

- Výpočet qual (normalizace)

  $$qual = signifikance \,/\, \max(signifikance_{max}, 12)$$

- Seřazení sestupně podle qual

Doporučení atributu – 3. krok

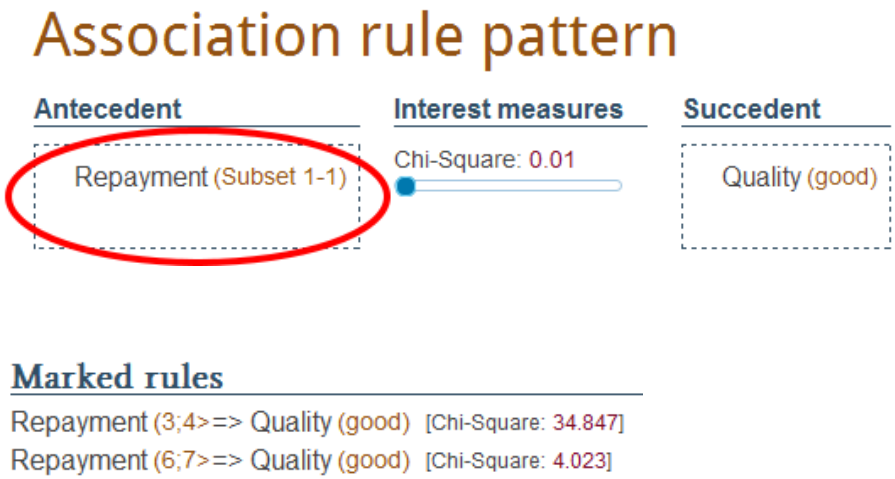
Jaký z atributů tedy doporučit?

- Atribut s max(qual)

Podle čeho měřit jeho vhodnost?

- qual > 0.75 – doporučený
- qual > 0.40 – částečně doporučený, nemusí vždy přispět k posílení vztahu ≈ mezi *Ant* a *Succ*

Fixace zadání



- Expert může přijít o zajímavé informace
- Doporučená změna koeficientu
  - Repayment (3,4> nebo Repayment (6;7>

Praktická ukázka

Závěr

Závěr

- AR Builder
  - Interaktivní přístup
  - Dolování asociačních pravidel
    - Analytické zprávy
  - Integrace background knowledge
  - Doporučení vhodného atributu
  - Mnoho námětů na další rozšíření
    - Implementace dalších kvantifikátorů, doporučování konkrétních kategorií atributů, apod.

Děkuji za pozornost