Quo vadis, data mining?

Luboš Popelínský
Fakulta informatiky, Masarykova universita
Botanická 68a, 602 00 Brno
Email: `popel@fi.muni.cz`
URL: `http://www.fi.muni.cz/usr/popelinsky`

Abstrakt

Jsou diskutovány perspektivní směry v dolování v datech - data mining. Je uvedena charakteristika data mining systémů 2. generace. Další kapitoly se týkají induktivních dotazovacích jazyků, systému *Data Surveyor* a hledání asociačních pravidel. Závěr se zmiňuje o možných vazbách data mining na datové sklady a OLAP.

Promising trends in data mining are discussed. Characteristics of 2nd generation data mining systems are introduced. Following sections concern inductive query languages, *Data Surveyor* data mining system and association rules. In the conclusion we discuss possibilities and a reality of mining algorithms in data warehouses and OLAP.

1 Vyhledávání znalostí v databázích a data mining

Narůstající množství dat uložených v databázích spolu s potřebou jejich hlubší analýzy vyústily ve zrod řady nových produktů a disciplín. Jednou z nich je **vyhledávání znalostí v databázích** (knowledge discovery in databases, KDD) [1, 9, 7] (z české literatury odkazujeme na [16, 17, 20, 6] a popis aplikací a systémů [5, 15]). Je to proces, v němž se původní (relační) data převádějí do jiného jazyka, který abstrahuje od nedůležitého nebo nepřesného. Ponechává jen to, co je potenciálně užitečné pro odpověď na konkrétní otázku (které skupiny řidičů jsou nejvíce rizikové) nebo pro řešení daného problému (jak snížit výdaje na léčebnou péči). Výsledkem (tedy znalostí) může být například „jen" výběr řádků typických pro dannou relaci, inteligentní grafické zobrazení, statistická charakteristika dat nebo formule v některém logickém kalkulu.

Celý proces KDD se obvykle dělí do pěti etap. První etapou je *selekce* dat, jež mají být popsána, následovaná jejich *předzpracováním*, (především minimalizace šumu, odstranění odlehlých hodnot a zpracování chybějících hodnot). Výsledkem *transformace* takto předzpracovaných dat může již být znalost např. ve formě trojrozměrného grafu. Vizualisační systémy jako *MineSet* `http://www.sgi.com/Products/software/MineSet/` od Silicon Graphics nebo *DataSplash* `http://tioga.cs.berkeley.edu:8000/tioga/` slouží právě k tomu. Další etapou je dolování v datech - **data mining**, o němž je tento příspěvek. *Intepretace* a/nebo *aplikace výsledků* je závěrečným krokem.

2 Nová generace

Data mining je vlastním jádrem celého procesu získávání znalostí. Označují se tak algoritmy, které z předzpracovaných dat odvodí jejich požadovanou charakteristiku.

Imielinski a Mannila [14] dělí data mining systémy do dvou generací. Současné systémy označované jako **systémy data mining 1. generace** (viz např. [16]) většinou slouží k odvození jen jediného typu znalosti. Typickým představitelem jsou systémy pro nalezení pravidla nebo pravidel, která popisují uživatelem definovanou množinu dat. Pravidla jsou tvaru **if** *Podmínka* **then** *Důsledek*, kde *Podmínka* je výrokem nad atributy relace a *Důsledek* je nejčastěji hodnota některého atributu. (např. **if** *Město=Ostrava a Typ=dobrovolní_hasiči* **then** *Výjezd=do_3_minut*). Vznikají spojením existujících induktivních (statistických nebo učících se) systémů s databázovým systémem. Data z databáze jsou exportována do induktivního systému a dále se zpracovávají jen tato exportovaná data. Vzhledem k malé schopnosti komunikace s databází a neschopnosti zpracovat velké (tj. reálné) objemy dat se jim někdy skromněji říká *file mining.*

Systémy 2. generace budou mnohem těsněji spojeny se systémem řízení báze dat(SŘBD). Opravdu dospělé data mining systémy podle [12] musí umět odvodit různé typy znalostí a poskytovat metody nejen pro popis zvolené množiny dat, ale i pro porovnávání, hledání zajímavých podmnožin, předpovědi trendů aj. Jejich součástí je dotazovací jazyk. Efektivně pracují v interaktivním režimu. Zvládnout i „špatně" strukturovaná data (hypertext, prostorová i multimediální data, aktivní databáze, WWW).

I když se omezíme jen na dolování v relačních databázích, nezbývá (pokud nemá data mining skončit u nepříliš užitečných prototypů) než vyřešit především následující úkoly:

1. vytvořit formální definici *induktivního dotazovacího jazyka* ;
2. vyvinout algoritmy pro *optimalizaci dotazů* ;
3. a také *efektivní* algoritmy pro *výběr* .

Induktivním dotazovacím jazyk je takový prostředek, který s komfortem alespoň na úrovni SQL dovolí uživateli dolovat znalosti, aniž by musel znát nějaké konkrétní algoritmy data mining. Jazyk DMQL (viz dále) je jedním z prvních. Dosud totiž neexistuje žádný jazykový standard (jako SQL-API) určený pro data mining, který by omezil třídu dotazů a umožnil jejich optimalizaci.

Optimalizace dotazů je nejdůležitějším úkolem. Data mining dotaz se ve skutečnosti skládá z posloupnosti jednodušších (např. SQL) dotazů. To, který z těchto elementárních dotazů bude položen, závisí na odpovědi na (některé) předcházející. Musíme tedy optimalizovat vždy několik dotazů, podle toho, jak na sebe navazují nebo mohou navazovat.

Kromě výše uvedeného by systém 2.generace měl umět manipulovat se svými vlastními objekty (např. pravidly), tj. nejen je vytvořit, ale i efektivně ukládat, vyhledávat v induktivní databázi(tj. v databázi pravidel) a rušit neplatná a/nebo redundantní pravidla. Takový KDDMS (Knowledge and Data Discovery Management System [14]) musí samozřejmě mít těsnou vazbu na SŘBD, aby byl schopen reagovat na dotazy typu „najdi protipříklady daného pravidla".

■

V dalším textu nás nebudou zajímat jednoúčelové systémy. Zaměříme se na nadějné přístupy k dolování v relačních databázích, které splňují (alespoň některé) z nahoře uvedených charakteristik systémů 2. generace. V následujících odstavcích nejprve uvedeme induktivní dotazovací jazyk DMQL, který používá *DBMiner* `http://db.cs.sfu.ca/DBMiner/` [11], následník systému *DBLearn* [10] . Asociační pravidla (zobecnění funkčních závislostí), která byla (poprvé pod tímto názvem) uvedena v [2] a asociace v geografických databázích jsou námětem dalších kapitol. *Data Surveyor* [13] je jeden z prvních systémů využívající pro data mining specializovaného databázového stroje, který umožňuje optimalizaci dotazů.

3 Induktivní dotazovací jazyky

V této sekci uvedeme příklad induktivního dotazovacího jazyka, který je vyvíjen v rámci projektu *DBMiner* [11]. Úplná syntaxe jazyka je uvedena v appendixu.

DBMiner nabízí grafické uživatelské rozhraní a induktivní dotazovací jazyk DMQL. Jazyk DMQL dovoluje definovat 5 typů pravidel.

Generalizace vytvoří ze zadané relace novou relaci, která obsahuje reprezentativní řádky vstupní relace pro zadanou míru zajímavosti. *Charakteristické pravidlo* popisuje danou relaci pomocí atributů uvedených ve frázi RELATED TO < seznam atributů a/nebo agregátů > (např charakteristika graduovaných studentů informatiky, kteří se narodili v Kanadě, pomocí atributů `gpa`, `birth_place`, `address`). *Klasifikační pravidla* klasifikují instance relace pomocí zadaných atributů (např. klasifikace `gpa` studentů informatiky narozených v Kanadě podle místa narozeí a adresy). *Asociační pravidla* (viz níže) jsou další možností (např. pravidla pro skupinu studentů informatiky narozených v Kanadě pro atributy `gpa, birth_place, address`).

Posledním z typů pravidel jsou *diskriminační pravidla* charakterizující, jak se danná relace liší od jiné (jiných). Mějme relační databázi s relacemi student, course, grading

```
student(name, ano, status, major, gpa, birth_date,
        birth_place, address)
course(cno, title, department)
grading(sno, cno, instructor, semester, grade)
```

Pro tuto databází potom diskriminační pravidlo pro srovnání graduovaných a negraduovaných studentů informatiky narozených v Kanadě pomocí atributů `gpa, birth_place, address` bude vypadat takto:

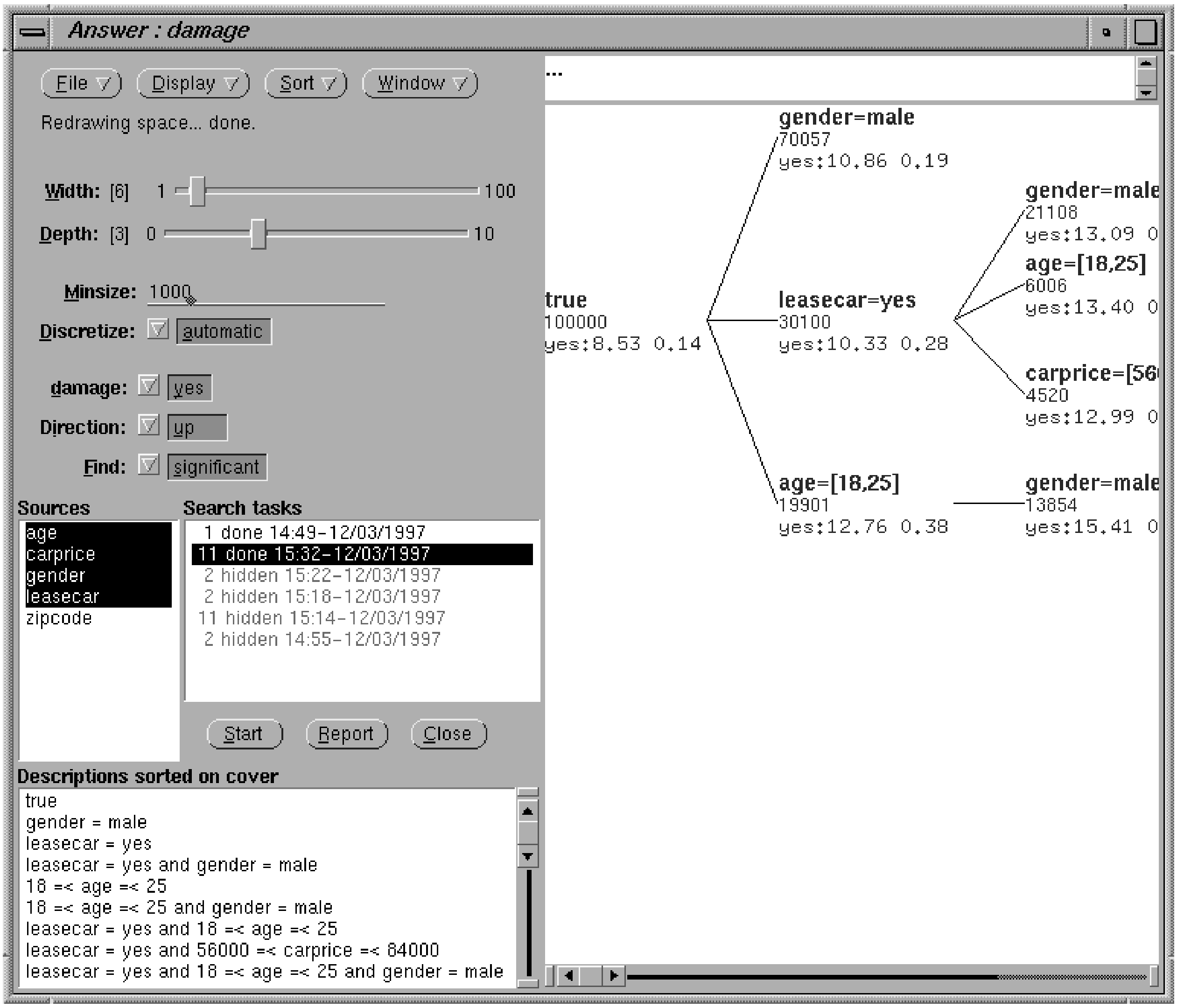
```
FIND DISCRIMINANT RULE
FOR cs_grads
    WITH status= "graduate"
IN CONTRAST TO cs_undergrads
    WITH status= "undergraduate"
RELATED TO gpa, birth_place, address, count(*)%
FROM student
WHERE major="cs" AND birth_place="Canada"
```

Následující příkaz induktivního dotazovacího jazyka $M-SQL$ [14] vygeneruje všechna pravidla pro relaci `T`, v nichž je na levé straně atribut `Age` (v podmínce na pravé straně libovolná kombinace atributů) a která popisují více než 1000 řádků s důvěryhodností větší než 65% . Jde tu opět o asociační pravidla.

```
SELECT
FROM   Mine(T): R
WHERE  R.Consequent = (Age = *)
R.Support    > 1000
R.Confidence > 0.65
```

Induktivní dotaz zadává vlastně skupinu SQL dotazů, z jejichž odpovědí se zobrazí jen ty zajímavé, viz parametry `Support` a `Confidence`).

4 *Data Surveyor*



Obr. 1 Hledání závislostí v databázi řidičů

Podstatou systému [13, 19] je dvouúrovňová architektura. Samotný *Data Surveyor* generuje induktivní dotazy. Na ně odpovídá databázový modul *MONET* optimalizovaný pro paralelní výpočty sloupcově orientovaných operací. Všechna data jsou nejprve transformována do binárních asociativních tabulek (*BAT* - Binary Association Table) tak, že každému atributu původní relace odpovídá jedna tabulka. Každá tabulka se skládá ze sloupce s identifikačním číslem původního záznamu v relaci a z hodnoty konkrétního atributu. Pro relaci s n atributy se tedy vytvoří n BAT-tabulek.

Z operací, které *MONET* umí zpracovat, využívá *Data Surveyor* čtyři

`AB.select(Tl,Th)`	$\{ab \mid ab \in AB \wedge b \geq Tl \wedge b \leq Th\}$
`AB.select(T)`	$\{ab \mid ab \in AB \wedge b = T\}$
`semijoin(AB,CD)`	$\{ab \mid ab \in AB, \exists cd \in CD \wedge a = c\}$
`AB.histogram`	$\{bf \mid ab \in AB, f \text{ je frekvence } b\}$

Data Surveyor buduje rozhodovací strom metodou shora dolů, tj. počáteční nejobecnější hypotéza se postupně specializuje přidáváním dalších podmínek. V každém kroku se dále rozvíjí jen určitý maximální počet cest (beam search), které splňují podmínku dostatečné podpory v datech a podmínku důvěryhodnosti. Díky strategii top-down je možné optimalizovat dotazy na *MONET* tak, že se v každém kroku použijí výsledky kroku předchozího. Navíc některé výpočtu mohou být prováděny, a $MONET$ je i provádí, paralelně.

Obrázek ukazuje zpracování databáze řidičů. Hledá se závislost mezi havárií ridiče a atributy relace `pohlaví, věk, cena_auta, na_leasing`. V každém uzlu je pod rozšiřující podmínkou (např `age=[18,25]` uveden absolutní počet záznamů v databázi (6006), pro něž podmínka `lease=car` $\wedge$ `age=[18,25]` platí.

$KESO$ `http://nathan.gmd.de/projects/ml/keso.html` projekt navazuje na *Data Surveyor*. V rámci tohoto ESPRIT projektu se vyvíjí nástroj data mining pro statistické úřady některých evropských zemí. Na projektu se kromě holandských Data Distilleries a Database Research Group CWI podílí též univesita v Helsinkách a GMD Bonn.

5 Hledání asociací

Asociační pravidla [2, 3] jsou znalosti typu *„90 % spotřebitelů, kteří nakupují zboží A a B, kupují i zboží C a D"* . Obecně je můžeme zapsat

$W \Rightarrow B[card(W), \frac{card(WB)}{card(W)}]$,

kde W, B jsou podmnožiny atributů dané relace nad doménou $\{0,1\}$, $card(W)$, *podpora(support)* je počet řádků relace, v nichž mají atributy z množiny W hodnotu 1. $\frac{card(WB)}{card(W)}$, *důvěryhodnost(confidence)* pravidla udává relativní četnost řádků relace s jedničkami v atributech W a B mezi všemi řádky s jedničkami v W. Chybějící položky se ignorují.

Základní úlohou je nalezení všech asociačních pravidla, kde

$support \geq minsup \wedge confidence \geq minconf$,

kde $minsup$, $minconf$ jsou dané konstanty.

Algoritmus popsaný v [3] má téměř lineární složitost pro náhodně generované databáze. Jeho hlavní schéma je následující.

Jsou-li A,B,C,D, ... atributy relace, potom

1. najdi všechny *velké* množiny položek, tj. takové pro něž $support \geq minsup$;
2. jsou-li $ABCD, AB$ velké množiny, a $\frac{support(ABCD)}{support(AB)} > minconf$
   generuj pravidlo
   $AB \Rightarrow CD[support(ABCD), \frac{support(ABCD)}{support(AB)}]$.

Metody pro hledání asociačních pravidel neuvažují velikost položek (např. množství daného druhu zboží nakoupeného v rámci transakce). Otevřeným problémem zůstává vyhledávání asociací v databázích se složitějšístrukturou. Není ani uspokojivě řešena optimalizace dotazů pro hledání pravidel s předem daným požadavkem na výskyt některé položky v předpokladu a/nebo v důsledku.

6 Asociace v geografických datech

V geografických databázích lze hledat asociační pravidla podobným postupem jako v transakčních databázích. Následující příklad uvádí dotaz na asociace mezi místem kursů golfu "`Golf Courses`" a objekty jako jsou jezera, silnice, parky, školy apod. Predikát `CLOSE_TO` určuje, zda dva objekty jsou blízko sebe.

```
CLOSE_TO(Washington.geo, Washington_Golf_Courses.geo, "3 km")
```

tedy určuje, že nás zajímají jen takové geografické objekty z `Washington.geo`, které jsou nejdále 3 kilometry od golfového hřiště. Dotaz je zapsán v jazyce GMQL, který je rozšířením DMQL.

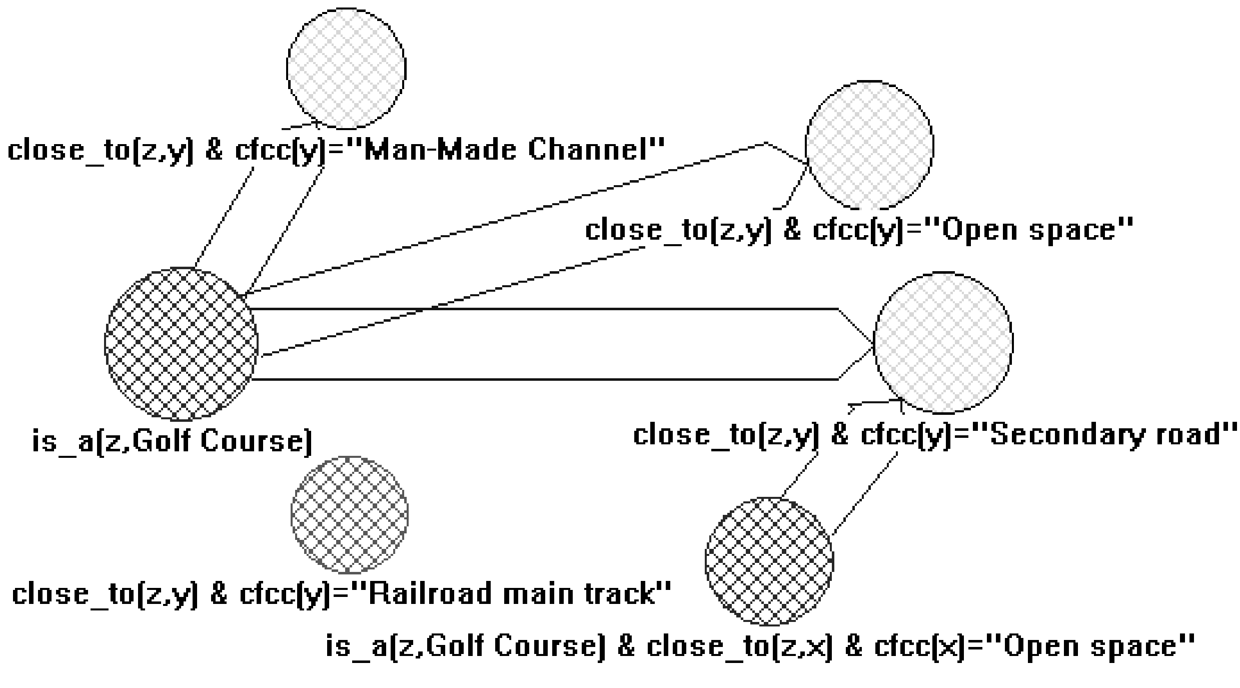
```
MINE SPATIAL ASSOCIATIONS AS "Golf Courses"
WITH RESPECT TO Washington.geo, Washington_Golf_Courses.geo, CFCC
FROM Washington_Golf_Courses, Washington
WHERE CLOSE_TO(Washington.geo, Washington_Golf_Courses.geo, "3 km")
        AND Washington.CFCC  "Golf Course"
SET SUPPORT THRESHOLD 0.4
SET CONFIDENCE THRESHOLD 0.5
```

Systémem *GeoMiner* byla pro st8t Wahington nalezena následující pravidla

```
isa(X, "Golf Course") -> closeto(X, "Man-Made Channel") (61%, 61%).
isa(X, "Golf Course") & closeto(X, "Secondary road")
                      -> closeto(X, "Open space") (64%, 78%).
```

Golfové kursy se tedy konají vetšinou u umělých kanálů. Je-li poblíž místa kursu vedlejší cesta, rozhodně bude nedaleko volný prostor(což příliš nepřekvapí).

Asociační pravidla mohou být kromě toho zobrazena asociačním grafem. Kruhy označují podmínku, přičemž kruh je tím větší čím více objektů podmínku splňuje. Šipky spojují tělo a hlavu pravidla (levou a pravou stranu). Šířka šipky odpovídá parametru *support* (první číslo v závorce), její délka je úměrná důvěryhodnosti pravidla (druhý parametr).



Obr. 2 Graf asociačních pravidel

7 Data mining, datové sklady, OLAP

Datový sklad (data warehouse) je potenciálním zdrojem dat i pro data mining. OLAP (On-Line Analytical Processing) [4] je vlastně (jednoduchým) systémem pro dolování agregovaných charakteristik dat. Data mining systém nad OLAPem může nacházet zajímavé závislosti podél libovolné dimenze ve vícedimenzionální tabulce a v různé hloubce abstrakce (samozřejmeě v závislosti na poskytnuté znalosti o doméně).

Skupina produktů *Fractal Data Mining System* od firmy *Cross/Z International Inc.)* [8] umí efektivně, tj. nezávisle na počtu řádků relace, dolovat v datových skladech. Počet proměnných(použitých sloupců relace) nesmí však být větší než 12. (Podobné číslo se uvádí jako horní mez pro OLAPy). Počet kombinací všech hodnot může být až 10^9. Systém pracuje ve dvou fázích. V první fázi jsou z datového skladu získány všechny výsledky se všemi kombinacemi hodnot všech požadovaných sloupců. Pro relaci se sloupci `pohlaví, stát, věk, příjem` systém klade dotazy typu *„Kolik mužů na Aljašce je ve věku 26 let"*, 27 let atd. pro všechny kombinace hodnot. Pro sloupec `příjem` se vypočte součet a průměr a tyto hodnoty slouží jako základ pro výpočet podobnosti. Výsledná databáze je poté fraktálově komprimována. Dotazy se vyhodnocují přímo na této komprimované databázi. Tento postup vede k extrémnímu snížení objemu dat.

Algoritmy data mining jsou schopny objevit sice složitější znalosti než současné OLAPy nebo relační OLAPy, ale zatím ne pro skutečně velké databáze. O integraci obou technologií se hodně mluví. V současné době, pokud je autorovi známo, ale nenabízí žádná z významných databázových firem žádný takový produkt.

8 Závěr

Některé zajímavé a důležité otázky jsme dosud nezmínili. Jak je to s bezpečností dat a nebezpečností jejich zobecnění? Jak využít vydolovaných znalostí například pro optimalizaci SQL dotazů? Dnešní systémy pracují převážně ve výrokové logice. Jsou vůbec reálně implementovatelné systémy, ktereé pracují s predikátovou logikou 1. řádu?

Data mining není jen samotný induktivní dotaz a jeho efektivní vyhodnocení. Je to spíš iterativní proces, kdy postupně zmenšujeme množinu pravidel, dokud nedostaneme uspokojivý výsledek. Od budoucích SŘBD je rozumné očekávat podporu tohoto procesu.

Poděkování

Autor děkuje recenzentům příspěvku a Jano Šefránkovi za cenné připomínky. Tento text vznikl rozšířením příspěvku na konferenci COFAX'97.

Literatura

[1] Data Mining, Communications of ACM, Vol. 39 No. 11, Nov. 1996.

[2] Agrawal R., Imielinski T., Swami A.: Mining Association Rules between Sets of Items in Large Databases. In: Proc. of ACM SIGMOD Conf. on Management of Data, 207-216, Washington D.C.

[3] Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo A.I.: Fast Discovery of Association Rules. In: [9]

[4] Bannister F.E.: OLAP - A Question of Definition. In: Sborník DATASEM'96, str.211-222, Brno, 1995.

[5] Berka P., Ivánek J.: Automatizované získávání znalostí z dat o kupónové privatizaci. In: Sborník DATASEM'95, str.179-188, Brno, 1995.

[6] Berka P.: Recognizing Reliability of Discovered Knowledge. 1st Eu. Symp. on Principles of Data Mining and Knowledge Discovery , Trondheim, Norsko 1997

[7] Chen M.-S., Han J., Yu P.S.: Data Mining: An Overview from Database Perspective. IEEE Transactions on Knowledge and Data Engineering, 1997.

[8] Edelstein H.: Mining Data Warehouses. Information Week 8. ledna 1996 `http://techweb.cmp.com/iwk`

[9] Fayyad U.M., Piatetski-Shapiro G., Smyth P., Uthurusamy R.(eds.): Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press 1996.

[10] Han J., Cai Y., Cercone N.: Knowledge Discovery in Databases: An Attribute-Oriented Approach. In: Sborník konference VLDB'92, str. 547-559, Vancouver 1992, Kanada.

[11] Han et al.: DMQL: A Data Mining Query Language for Relational Databases. In: ACM-SIGMOD'96 Workshop on Data Mining

[12] Han J.: Data Mining: Where Is It Heading?. Int'l Conf. on Data Engineering ICDE'97 (panelová diskuse), Birmingham, Anglie 1997.

[13] Holsheimer M., Siebes A.: Data mining: the search of knowledge in databases. Technická zpráva CS-R9406, CWI Amsterodam, 1994.
(viz též Holsheimer M., Kersten M., Siebes A v [9])

[14] Imelinski T. Mannila H.: A Database Perspective on Knowledge Discovery. In [1].

[15] Klement P.: ECCE - systém objevující znalosti v databázích. In: Sborník DATASEM'96, str. 295-204, Brno 1996.

[16] Popelínský L.: Data mining: Hledání znalostí v databázích. In: Sborník DATASEM'95, str. 87-94, Brno 1995.

[17] Rauch J.: Získávání znalostí z databází a GUHA. In: Sborník DATASEM'95, str.151-162, Brno 1995.

[18] Rauch J.: Logical Calculi for Knowledge Discovery in Databases 1st Eu. Symp. on Principles of Data Mining and Knowledge Discovery, Trondheim, Norsko 1997

[19] Siebes A.: Data Mining and the *KESO* Project. In: SOFSEM'97, Milovy, CZ, 1997.

[20] Soukeník K.: Hledání závislostí s ohledem na aktualizace databáze. In: Sborník DATASEM'96, str.305-314, Brno 1996.

A Syntaxe jazyka DMQL

DMQL ::=

USE DATABASE <databáze>
{USE HIERARCHY <hierarchie> FOR <atributy>
<specifikace pravidla>
RELATED TO < seznam atributů a/nebo agregátů>
FROM < seznam relací>
[WHERE < podmínka >]
[ORDER BY < seznam >]
{ < kritérium zajímavosti > }

<specifikace pravidla> ::=

GENERALIZE DATA [INTO <jméno relace>] |
FIND CHARACTERISTIC RULES [AS <jméno pravidla>] |
< specifikace charakteristického pravidla>
FIND CLASSIFICATION RULES [AS <jméno pravidla>] |
ACCORDING TO <atributy>
FIND ASSOCIATION RULES [AS <jméno pravidla>]

<specifikace diskriminačního pravidla> ::=

FIND DISCRIMINANT RULES [AS < jméno pravidla >]
FOR < třída1 > < podmínka1 >
FROM < seznam relací 1 >
IN CONTRAST TO < třída2 > < podmínka2 >
FROM < seznam relací 2 >
{ IN CONTRAST TO ... }

< kritérium zajímavosti> ::=

WITH [< typ >]
THRESHOLD = < prahová hodnota >
[FOR < seznam atributů >]

< typ > ::= SUPPORT | CONFIDENCE | NOISE | REDUNDANCY