

FUZZY LOGIKA V DATABÁZOVÝCH SYSTÉMECH

FUZZY LOGIC IN THE DATABASE SYSTEMS

Ondřej Volráb

Anotace:

Příspěvek se zabývá fuzzy logikou a možnostmi její podpory v moderních databázových systémech. V rámci řešené problematiky jsou uvedeny jak přístupy vhodné pro klasické relační, tak i přístupy vhodné pro čistě objektové databáze.

Klíčová slova:

fuzzy, fuzzy logika, relační databáze, objektová databáze, Gemstone

Abstract:

This paper is dedicated to fuzzy logic and some possibilities of its integration with the modern database systems. There are presented some concepts suitable for classic relational databases and some concepts suitable for pure object oriented databases.

Keywords:

fuzzy, fuzzy logic, relational database, object database, Gemstone

ÚVOD

Databázové systémy dnes představují jeden z nejčastěji využívaných zdrojů informací. Vyskytují se doslova na každém kroku a uchovávají v sobě obrovské množství údajů, které je potřeba umět zpracovat a transformovat do podoby, která uspokojí konkrétní informační potřeby. Častý problém však představuje vágní formulace informační potřeby, která naráží na omezené možnosti klasických technologií pro získávání dat (např. jazyk SQL). Jako jedno z řešení se nabízí možnost využití fuzzy logiky při konstrukci databázových dotazů.

FUZZY LOGIKA

Fuzzy logika se poprvé objevila v roce 1965 v článku, který publikoval profesor Lotfi A. Zadeh. Tehdy došlo k definici základního pojmu fuzzy logiky, a to tzv. fuzzy množiny. Termín fuzzy je do češtiny nejčastěji překládán jako neostrý, matný, mlhavý, neurčitý či vágní. Vyjadřuje snahu fuzzy teorie o pokrytí reality v její nepřesnosti a neurčitosti.

Klasická teorie množin připouští pouze dva stavy prvku vůči množině. Prvek do množiny buďto zcela patří nebo vůbec nepatří. Fuzzy logika tento koncept rozšiřuje. Fuzzy množina je taková množina, která kromě úplného členství a žádného členství připouští rovněž členství částečné. Míra členství daného prvku ve fuzzy množině je vyjádřena stupněm příslušnosti. Hodnota stupně příslušnosti každého prvku je získána pomocí tzv. funkce příslušnosti.

Význam fuzzy logiky lze vnímat ve dvou rovinách. První rovinou je potřeba uchopit a pracovat s nepřesnými či mlhavými daty. Druhou rovinou je celkový přístup k popisu reálného světa. Při používání přesných popisů totiž dochází k idealizování skutečnosti a tedy odklonu od reality.

Klasický přístup vede ke snaze popsat skutečnost pouze dvěma stavy (patří/nepatří do

množiny). Pokud není možné situaci jednoznačně rozhodnout, pak je řešený problém dekomponován a opět posouzen z hlediska jediných dvou možných stavů. V situaci, kdy již není možné či účelné problém dále dekomponovat, dochází k odklonu od reality. Z uvedeného vychází i princip inkompatibility, který vyslovil profesor Zadeh:

„S rostoucí složitostí systému klesá naše schopnost formulovat přesné a významné vlastnosti o jeho chování, až je dosáhnuta hranice, za kterou jsou přesnost a relevantnost prakticky vzájemně se vylučující jevy.“

ZAVEDENÍ FUZZY MNOŽIN

S mlhavými pojmy se lze v přirozeném jazyce setkat na každém kroku: blízko obchodu, rychlé auto, mladý člověk atd. Klíčovou otázkou je, co je ještě možné zařadit do dané množiny a co již ne. Je-li člověk ve věku 20 let považován za mladého, proč není mladý také ten, kterému je 21? Problém spočívající v hraničních bodech je možné řešit zavedením fuzzy množin.

Každému prvku je přiřazen stupeň příslušnosti, který vyjadřuje míru členství prvku ve fuzzy množině. Stupeň příslušnosti nabývá hodnot z intervalu $<0,1>$. Čím vyšší (nižší) je stupeň příslušnosti, tím více (méně) platí, že prvek náleží do fuzzy množiny. Jinými slovy vyjadřuje míru relevantnosti prvku v dané množině. Hodnota 0 vyjadřuje stav, kdy prvek do množiny vůbec nepatří a hodnota 1 stav, kdy prvek do množiny zcela patří. Tyto hraniční stavy jsou ekvivalentní se stavy v klasické teorii množin.

FUZZY PŘÍSTUP V RELAČNÍCH DATABÁZOVÝCH SYSTÉMECH

Standard pro práci s relačními databázovými systémy dnes představuje jazyk SQL. Jedním z jeho klíčových rysů je tříhodnotová logika, se kterou pracuje. Prostřednictvím hodnot true, false a null je vymezen způsob, jakým lze sestavovat a vyhodnocovat podmínky v dotazech nad relačními tabulkami.

Problém při užití jazyka SQL pro nalezení určité množiny dat je zřejmý. V první řadě je to problematická konstrukce podmínek, které nelze jednoduše vyhodnotit jako pravdivé resp. nepravdivé. Především však může nastat problém s interpretací získaného výsledku. Množina, která je dotazem generována totiž nenese žádnou informaci o tom, jak moc jednotlivé prvky odpovídají počátečnímu požadavku transformovaného na SQL dotaz. Podstatně nepříjemnější může být skutečnost, že se do výsledku nedostanou takové prvky, které nesplnily zadané podmínky jen velmi těsně. V dotazu s podmínkou typu

A and B and C and ... and Z

budou ve výsledku jen takové prvky, které splní všechny podmínky zároveň. Pokud by bylo potřeba získat rovněž prvky, které nesplňují maximálně jednu z podmínek, situace se začne komplikovat. Malý počet dílčích podmínek umožňuje vytvoření patřičných kombinací, avšak se vzrůstající složitostí vztahů a jejich počtu se stává tento přístup nepoužitelný. Ještě složitější situace by nastala, pokud by prvek nemusel splňovat dvě či více dílčích podmínek a pokud by bylo potřeba rozlišit jejich váhy.

Východiskem pro řešení nastíněných problémů může být aplikace fuzzy logiky v prostředí relačních databází. V principu jde o přiřazení stupně příslušnosti prostřednictvím vhodně definované funkce příslušnosti. Výsledná množina je pak fuzzy relací, která nese informace o tom, do jaké míry odpovídají jednotlivé prvky zadaným kritériím. Tato míra je vyjádřena stupněm příslušnosti od 0 – vůbec neodpovídá do 1 – odpovídá zcela.

Fuzzy relace R_f je podmnožina kartézského součinu domén $D_1, \dots, D_n$, kde jedno D_i je DU — interval $<0,1>$. Ostatní sloupce jsou důležité pro výpočet sloupce DU . Hodnota libovolné n -tice t říká, jak hodně tato n -tice náleží do fuzzy relace R_f . Pro porovnání lze uvést, že obvyklou relaci je možné považovat za fuzzy relaci, kde ve sloupci DU je stupeň

roven vždy jedné pro každou n-tici.

Uplatnění fuzzy teorie nad daty uloženými v relační databázi vyžaduje, aby měl uživatel k dispozici prostředky pro vyjádření pojmů s nejasnými hranicemi a mohl je použít při konstrukci svých dotazů. Nejčastěji jsou takové prostředky v oblasti relačních databázových systémů umístěny do:

- a) aplikační vrstvy,
- b) nadstavby nad databázovým systémem,
- c) vrstvy uložených procedur.

Předností umístění do aplikační vrstvy je vyšší uživatelský komfort. Naopak nevýhodou je omezená možnost měnit předdefinované pohledy a typy dotazů. Velmi častým řešením jsou proto speciální nadstavby nad databázovým systémem, které představují jakýsi filtr mezi uživatelem a samotnými daty. Jejich hlavním úkolem je interpretace rozšíření dotazovacího jazyka a sestavování odpovědí. Příklad dotazu v modifikované verzi jazyka SQL (označované jako Fuzzy SQL) může vypadat následovně:

```
select NAME, AGE, INCOME
from EMPLOYEE
where (AGE = middle)
      and (INCOME >= high)
```

Nejnižší úroveň je představována přidáním podpory fuzzy výrazových prostředků do samotného SRBD. Výhodou je relativní volnost při sestavování dotazů. V moderních relačních databázových systémech jsou pro tyto účely využívány především uložené procedury (např. v Oracle PL/SQL, DB2 SQL PL, PL/pgSQL). Konkrétní dotaz vybírající podmnožinu velmi malých nebo mladých studentů může vypadat následovně:

```
select s.* from student s
where fuzzy.op_or(
      fuzzy.m_very(student_pravidla.maly(vyska)),
      student_pravidla.mlady(vek)) > 0;
```

Základní operace nad fuzzy množinami (sjednocení, průnik, rozdíl atd.) je možné definovat obecně, avšak funkce přiřazující jednotlivým entitám stupeň příslušnosti je potřeba definovat pro každý typ entity zvlášť. To znamená, že pokud by bylo potřeba v databázi zoologické zahrady identifikovat všechna zvířata s nadměrnou hmotností, pak by bylo potřeba nejprve znát všechny druhy zvířat a nadefinovat pro ně funkce příslušnosti do množiny „zvířata s nadváhou“ (každý druh má přirozeně jiné hranice pro posouzení hmotnosti). Nedostatky tohoto přístupu jsou zřejmé. Funkce příslušnosti jsou definovány odděleně od vlastních typů entit, což může vést k obtížné udržitelnosti a přenositelnosti datového modelu. Obtížné je rovněž sdílení jednotlivých funkcí příslušnosti mezi více typy entit. Konstrukce dotazů se navíc stává s narůstajícím počtem typů entit značně nepřehledná. Naznačená úskalí je možné eliminovat využitím některého plně objektového databázového stroje.

FUZZY PŘÍSTUP V OBJEKTOVÝCH DATABÁZOVÝCH SYSTÉMECH

Objektové databáze se poprvé objevují přibližně v polovině 80. let, kdy začala vznikat potřeba co nejelegantnějšího ukládání datových struktur z objektových programovacích jazyků. Počáteční výkonnostní deficit ve srovnání s relačními databázemi byl postupně smazán a dnes lze hovořit o plnohodnotných alternativách. Moderní objektové databázové systémy samozřejmě podporují stovky konkurenčních uživatelů, optimistické/pesimistické řízení transakcí i dnes velmi žádanou práci v clusteru.

U objektových databází je potřeba rozlišovat datový model, na kterém jsou založeny. Rozšířenější variantou jsou databáze založené na relačním datovém modelu, který je postupně rozšiřován o objektové vlastnosti (např. Oracle 10g). Plné využití objektového přístupu však umožňuje teprve databáze založená na čistě objektovém datovém modelu (např. Gemstone,

Versant, ObjectStore). Objektový datový model je vhodnější především v situacích, kdy jsou do databáze ukládány značně strukturovaná data, která jsou vzájemně v mnoha vztazích. V takovém případě je možné využít principů známých z objektových programovacích jazyků, mezi které patří:

- Schopnost vytvářet množiny (kolekce) objektů, jejichž prvky mohou být objekty různých datových typů (tříd).
- Skládání objektů mezi sebou a využívání principu zapouzdření jejich vnitřní struktury.
- Odvozování nových tříd objektů od existujících pomocí mechanismu dědění.
- Definování atributů odvozených z ostatních datových složek objektu.
- Polymorfní chování objektů umožňující tvořit dotazy nad množinami objektů rozdílných tříd.

Pro využití fuzzy logiky v prostředí objektových databází je potřeba, stejně jako v případě relačních databází, zvolit způsob jakým budou jednotlivým objektům přiřazovány stupně příslušnosti do fuzzy množin. Díky objektovému datovému modelu je možné ponechat definici funkce příslušnosti přímo ve třídě objektů, kterých se týká. To přináší celou řadu výhod jako zapouzdření specifických detailů, vyšší míru přenositelnosti a snadnější udržitelnost. Vytvořená datová struktura navíc zůstává do budoucna otevřená. Zpřístupnění nových tříd v rámci fuzzy dotazů je redukováno na doplnění protokolu zpráv pro výpočet jednotlivých funkcí příslušnosti. Díky vazbám typu skládání a dědění lze udržívat funkce příslušnosti společně více třídám na jednom místě a zároveň ponechat specifickým třídám možnost jejich překrytí. Hlavním přínosem naznačeného přístupu je získání vzájemně polymorfních tříd objektů. Následující příklady z prostředí databáze Gemstone jsou napsány v jazyce Smalltalk DB a demonstrují možnosti sestavování fuzzy dotazů v čistě objektové databázi.

```
Zvirata select: [:z | z.nadvaha > 0.7].
```

Kolekce objektů Zvirata obsahuje instance různých tříd, tj. různé druhy zvířat. Díky polymorfismu (zprava nadvaha) je pro každé zvíře (resp. každý druh) vypočtena hodnota příslušnosti do fuzzy množiny dle specifické funkce. Výsledkem dotazu je kolekce zvířat, která splňují podmínku nadváhy z více jak 70 %.

Operace nad získanými fuzzy množinami lze demonstrovat na dalším příkladu. V tomto případě půjde o průnik dvou fuzzy množin, kde $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$.

```
Zvirata select: [:z | (z.nadvaha min: (z.mlady)) > 0.7].
```

Výsledná kolekce bude obsahovat zvířata, která splňují z více jak 40 % podmínku, že jsou mladá a mají nadváhu.

ZÁVĚR

Zavedení podpory fuzzy přístupu při konstrukci databázových dotazů značně rozšiřuje využitelnost uchovávaných dat. V prvé řadě dochází k odbourávání kompromisní transformace vágně definovaných dotazů na soustavy striktně ohraničených podmínek. Prvky v rámci získané fuzzy množiny lze navíc filtrovat a třídit dle míry jejich shody s požadavky.

Uvedené přístupy pro zavádění podpory fuzzy logiky do databázových systémů naznačují, že pro tyto účely jsou vhodnější databáze založené na čistě objektovém datovém modelu. Jedná se především o transparentnost, kompaktnost a schopnost polymorfního chování objektů. V neposlední řadě je třeba zdůraznit skutečnost, že konstrukce fuzzy dotazů v prostředí objektových databází obvykle nevyžaduje žádná rozšíření dotazovacího jazyka, jelikož je zcela postačující obecný princip zasílání zpráv mezi objekty.

Literatura:

- LACKO, Luboslav. Oracle – Správa, programování a použití databázového systému. Přeložil Marek Kocan, Michal Brůha. 1. vyd. Praha: Computer Press, 2002. 464 s. ISBN 80-7226-699-3.
- MERUNKA, V. Objekty v databázových systémech. ČZU Praha, Praha 2002. ISBN 80-213-0318-2
- MERUNKA, V., VOSTROVSKÝ, V. Databázové systémy, 2. vyd. Praha: Provozně ekonomická fakulta ČZU v Praze, 1999. 168 s. ISBN 80-213-0543-6.
- POLÁK, J., MERUNKA, V., CARDA, A. Umění systémového návrhu, 1. vyd. Praha: Grada, 2003. 196 s. ISBN 80-247-0424-2.
- SCIENTA INTELLIGENCE. FuzzySQL [online]. c2004, poslední aktualizace 2004-05-12 [cit. 2004-05-31]. Dostupné z: <<http://scianta.com/products/fuzzysql.htm>>.
- STEPHENS, Scott. Go fuzzy with Oracle [online]. c1995-2004, poslední aktualizace 2004-05-17 [cit. 2004-05-31]. Dostupné z: <<http://asia.cnet.com/builder/architect/db/0,39009328,39179658,00.htm>>.

Adresa autora:

Ing. Ondřej Volráb
Katedra informačního inženýrství
Provozně ekonomická fakulta
Česká zemědělská univerzita v Praze
E-mail: volrab@pef.czu.cz